

PL/SQL Developer 7.0

Руководство пользователя

Январь 2006



allroundautomations

Contents

CONTENTS	2
1. ВВЕДЕНИЕ	7
2. УСТАНОВКА	11
2.1 СИСТЕМНЫЕ ТРЕБОВАНИЯ	11
2.2 УСТАНОВКА НА РАБОЧЮЮ СТАНЦИЮ	11
2.3 УСТАНОВКА НА СЕРВЕР	11
2.4 УСТАНОВКА УПРАВЛЯЕМАЯ СКРИПТОМ	12
2.5 УДАЛЕНИЕ PL/SQL DEVELOPER	12
3. НАПИСАНИЕ ПРОГРАММ	13
3.1 СОЗДАНИЕ ПРОГРАММЫ	13
3.2 ИЗМЕНЕНИЕ ПРОГРАММЫ	15
3.3 КОМПИЛИРОВАНИЕ ПРОГРАММЫ	15
3.4 ПОДСТАНОВОЧНЫЕ ПЕРЕМЕННЫЕ	16
4. ТЕСТИРОВАНИЕ ПРОГРАММ	17
4.1 СОЗДАНИЕ ТЕСТОВОГО СКРИПТА	17
4.2 ВЫПОЛНЕНИЕ ТЕСТОВОГО СКРИПТА	18
4.3 ТИПЫ ПЕРЕМЕННЫХ	19
4.4 СОХРАНЕНИЕ ТЕСТОВЫХ СКРИПТОВ	20
4.5 ОТСЛЕЖИВАНИЕ ОШИБОК ВРЕМЕНИ ВЫПОЛНЕНИЯ	21
4.6 СОСТОЯНИЯ ПАКЕТОВ И JAVA СЕССИЙ	21
4.7 ПРОСМОТР НАБОРА РЕЗУЛЬТАТОВ	22
4.8 ПРОСМОТР DBMS_OUTPUT	22
4.9 ПРОСМОТР НТР ВЫВОДА	22
4.10 ОТЛАДКА	22
4.11 ТРАССИРОВКА ИСПОЛНЕНИЯ	26
4.12 РЕГРЕССИОННОЕ ТЕСТИРОВАНИЕ	27
5. ОПТИМИЗАЦИЯ	28
5.1 ИСПОЛЬЗОВАНИЕ ОКНА ПЛАНА ВЫПОЛНЕНИЯ	28
5.2 АВТОМАТИЧЕСКИЙ СБОР СТАТИСТИКИ	29
5.3 ПРОФИЛИРОВЩИК PL/SQL	31
5.4 ТРАССИРОВКА SQL	32
6. НЕЗАПЛАНИРОВАННЫЙ SQL ЗАПРОС	33
6.1 ИСПОЛЬЗОВАНИЕ SQL ОКНА	33
6.2 ДЕЙСТВИЯ С ТАБЛИЦЕЙ РЕЗУЛЬТАТОВ	34
6.3 РЕЖИМ ЗАПРОСА ПО ОБРАЗЦУ	38
6.4 СВЯЗАННЫЕ ЗАПРОСЫ	39
6.5 ПОДСТАНОВОЧНЫЕ ПЕРЕМЕННЫЕ	40
6.6 ОБНОВЛЕНИЕ БАЗЫ ДАННЫХ	41
6.7 ПРОСМОТР И РЕДАКТИРОВАНИЕ СТОЛБЦОВ XMLTYPE	42
6.8 ПРЯМОЙ ЭКСПОРТ ЗАПРОСА	42
6.9 СОХРАНЕНИЕ SQL СКРИПТОВ	42
6.10 СОЗДАНИЕ СТАНДАРТНЫХ ЗАПРОСОВ	43
7. КОМАНДНОЕ ОКНО	44
7.1 ВВОД SQL ОПЕРАТОРОВ И КОМАНД	44
7.2 РАЗРАБОТКА КОМАНДНЫХ ФАЙЛОВ	45
7.3 ПОДДЕРЖИВАЕМЫЕ КОМАНДЫ	46

8.	СОЗДАНИЕ И ИЗМЕНЕНИЕ НЕ PL/SQL ОБЪЕКТОВ.....	49
8.1	РЕДАКТОР ОПРЕДЕЛЕНИЯ ТАБЛИЦЫ	49
8.2	РЕДАКТОР ОПРЕДЕЛЕНИЯ ПОСЛЕДОВАТЕЛЬНОСТИ	60
8.3	РЕДАКТОР ОПРЕДЕЛЕНИЯ СИНОНИМОВ	60
8.4	РЕДАКТОР ОПРЕДЕЛЕНИЯ БИБЛИОТЕКИ	61
8.5	РЕДАКТОР ОПРЕДЕЛЕНИЯ ДИРЕКТОРИИ	61
8.6	РЕДАКТОР ОПРЕДЕЛЕНИЯ ЗАДАНИЯ	62
8.7	РЕДАКТОР ОПРЕДЕЛЕНИЯ ОЧЕРЕДИ	62
8.8	РЕДАКТОР ОПРЕДЕЛЕНИЯ ТАБЛИЦЫ ОЧЕРЕДИ	63
8.9	РЕДАКТОР ОПРЕДЕЛЕНИЯ ПОЛЬЗОВАТЕЛЯ	64
8.10	РЕДАКТОР ОПРЕДЕЛЕНИЯ РОЛИ	67
8.11	РЕДАКТОР ОПРЕДЕЛЕНИЯ ПРОФИЛЯ	68
8.12	РЕДАКТОР ОПРЕДЕЛЕНИЯ СВЯЗИ БАЗЫ ДАННЫХ	68
9.	ДИАГРАММЫ	69
9.1	СОЗДАНИЕ ДИАГРАММ	69
9.2	СОХРАНЕНИЕ И ОТКРЫТИЕ ФАЙЛА ДИАГРАММЫ	72
9.3	ОБНОВЛЕНИЕ ДИАГРАММЫ	72
10.	ОТЧЕТЫ.....	73
10.1	СТАНДАРТНЫЕ ОТЧЁТЫ	73
10.2	ПОЛЬЗОВАТЕЛЬСКИЕ ОТЧЁТЫ.....	74
10.3	ПЕРЕМЕННЫЕ	75
10.4	ДОВОДКА МАКЕТА.....	80
10.5	БИБЛИОТЕКА СТИЛЕЙ.....	87
10.6	ОПЦИИ	88
10.7	МЕНЮ «ОТЧЁТЫ».....	90
11.	ГРАФИКИ	91
12.	ПРОЕКТЫ.....	94
12.1	СОЗДАНИЕ НОВОГО ПРОЕКТА	94
12.2	СОХРАНЕНИЕ ПРОЕКТА	95
12.3	ДОБАВЛЕНИЕ В ПРОЕКТ ФАЙЛОВ.....	95
12.4	ДОБАВЛЕНИЕ ОБЪЕКТОВ БАЗЫ ДАННЫХ В ПРОЕКТ	95
12.5	РАБОТА С ЭЛЕМЕНТАМИ ПРОЕКТА	96
12.6	КОМПИЛЯЦИЯ ПРОЕКТА.....	96
13.	ТО-ДО ЭЛЕМЕНТЫ.....	98
13.1	СОЗДАНИЕ То-До ЭЛЕМЕНТА	99
13.2	РЕДАКТИРОВАНИЕ То-До ЭЛЕМЕНТА.....	99
13.3	ЗАКРЫТИЕ То-До ЭЛЕМЕНТА	100
13.4	УДАЛЕНИЕ То-До ЭЛЕМЕНТА.....	100
14.	ОКНА, СЕССИИ БАЗЫ ДАННЫХ И ТРАНЗАКЦИИ	101
14.1	РЕЖИМ СЕССИЙ	101
14.2	ВЫПОЛНЕНИЕ В РЕЖИМАХ МНОГО СЕССИЙ ИЛИ ДВЕ СЕССИИ	101
15.	ПРОСМОТР ОБЪЕКТОВ.....	102
15.1	ИСПОЛЬЗОВАНИЕ НАВИГАТОРА	102
15.2	ФИЛЬТРЫ НАВИГАТОРА	108
15.3	ПАПКИ НАВИГАТОРА	109
16.	ПРЕДПОЧТЕНИЯ.....	111
16.1	ORACLE - СОЕДИНЕНИЕ	112
16.2	ORACLE - ОПЦИИ	113
16.3	ORACLE - ОТЛАДЧИК	114

16.4	ORACLE - ВЫВОД.....	115
16.5	ORACLE - ТРАССИРОВКА.....	116
16.6	ORACLE - ПРОФИЛИРОВЩИК.....	116
16.7	ORACLE – ИСТОРИЯ СОЕДИНЕНИЙ.....	118
16.8	ORACLE – ПОДСКАЗКИ.....	120
16.9	ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС – ОПЦИИ.....	122
16.10	ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС – ПАНЕЛЬ ИНСТРУМЕНТОВ.....	124
16.11	ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС – НАВИГАТОР.....	124
16.12	ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС - РЕДАКТОР.....	126
16.13	ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС – ШРИФТЫ.....	128
16.14	ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС – АССИСТЕНТ КОДИРОВАНИЯ.....	129
16.15	ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС – НАСТРОЙКА КЛАВИАТУРЫ.....	130
16.16	ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС – ВНЕШНИЙ ВИД.....	131
16.17	ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС – ДАТА/ВРЕМЯ.....	132
16.18	ТИПЫ ОКОН – ПРОГРАММНОЕ ОКНО.....	133
16.19	ТИПЫ ОКОН – ОКНО SQL.....	135
16.20	ТИПЫ ОКОН – ТЕСТОВОЕ ОКНО.....	137
16.21	ТИПЫ ОКОН - ОКНО ПЛАНА.....	138
16.22	ИНСТРУМЕНТЫ – СРАВНЕНИЕ.....	138
16.23	ИНСТРУМЕНТЫ – ГЕНЕРАТОР ДАННЫХ.....	139
16.24	ИНСТРУМЕНТЫ – To-Do СПИСОК.....	140
16.25	ИНСТРУМЕНТЫ – ЗАПОМИНАНИЕ КОМАНД.....	140
16.26	ФАЙЛЫ - КАТАЛОГИ.....	141
16.27	ФАЙЛЫ – РАСШИРЕНИЯ.....	142
16.28	ФАЙЛЫ - ФОРМАТ.....	143
16.29	ФАЙЛЫ - РЕЗЕРВНОЕ КОПИРОВАНИЕ.....	144
16.30	ФАЙЛЫ – HTML/XML.....	145
16.31	ПРОЧЕЕ – ПЕЧАТЬ.....	146
16.32	ПРОЧЕЕ – ОБНОВЛЕНИЯ & НОВОСТИ.....	147
16.33	НАБОРЫ ПРЕДПОЧТЕНИЙ.....	148
17.	ИНСТРУМЕНТЫ.....	150
17.1	НАВИГАТОР.....	150
17.2	НАЙТИ ОБЪЕКТЫ БАЗ ДАННЫХ.....	150
17.3	КОМПИЛИРОВАТЬ ИНВАЛИДНЫЕ ОБЪЕКТЫ.....	152
17.4	ЭКСПОРТИРОВАТЬ ТАБЛИЦЫ.....	153
17.5	ИМПОРТИРОВАТЬ ТАБЛИЦЫ.....	156
17.6	ЭКСПОРТИРОВАТЬ ОБЪЕКТЫ ПОЛЬЗОВАТЕЛЯ.....	158
17.7	ИМПОРТЁР ТЕКСТА.....	159
17.8	ODBC ИМПОРТЕР.....	163
17.9	ГЕНЕРАТОР ДАННЫХ.....	165
17.10	СРАВНИТЬ ОБЪЕКТЫ ПОЛЬЗОВАТЕЛЯ.....	169
17.11	СРАВНИТЬ ДАННЫЕ ТАБЛИЦЫ.....	171
17.12	МОНИТОР СОБЫТИЙ.....	174
17.13	СЕССИИ.....	175
17.14	ИНСТРУМЕНТЫ, ОПРЕДЕЛЯЕМЫЕ ПОЛЬЗОВАТЕЛЕМ.....	177
17.15	МЕНЕДЖЕР ТЕСТОВ.....	182
18.	РЕДАКТОР.....	186
18.1	ФУНКЦИИ ВЫБОРА.....	186
18.2	ВЫДЕЛЕНИЕ СТОЛБЦА.....	186
18.3	НАПРАВЛЯЮЩАЯ.....	187
18.4	АССИСТЕНТ КОДИРОВАНИЯ.....	187
18.5	ЗАПОМИНАНИЕ КОМАНД.....	188

18.6	СПЕЦИАЛЬНОЕ КОПИРОВАНИЕ	189
18.7	КОНТЕКСТНО-ЗАВИСИМАЯ ПОМОЩЬ	190
18.8	ВСПЛЫВАЮЩЕЕ МЕНЮ ОБЪЕКТА БАЗЫ ДАННЫХ	190
18.9	ПОСТРОЕНИЕ ПЛАНА ВЫПОЛНЕНИЯ	190
18.10	МАКРОСЫ	190
18.11	ЗАКЛАДКИ	192
18.12	ВЫДЕЛЕНИЕ ЦВЕТОМ	192
18.13	РАЗДЕЛЫ КОДА	193
18.14	ИЕРАРХИЯ КОДА	194
18.15	НАВИГАЦИЯ ПО ГИПЕРССЫЛКАМ	194
	КНОПКИ НАВИГАЦИИ	195
18.17	РЕФАКТОРИНГ	195
18.18	ПАНЕЛЬ ПОИСКА	196
19.	РЕДАКТОР МАССИВНЫХ ДАННЫХ	198
19.1	РЕДАКТИРОВАНИЕ ТЕКСТА	199
19.2	РЕДАКТИРОВАНИЕ RTF	199
19.3	РЕДАКТИРОВАНИЕ XML	200
19.4	РЕДАКТИРОВАНИЕ РИСУНКОВ	201
19.5	РЕДАКТИРОВАНИЕ HEX ДАННЫХ	201
19.6	ВЫЗОВ ВНЕШНЕЙ ПРОГРАММЫ ПРОСМОТРА ИЛИ РЕДАКТОРА	203
20.	ПОСТРОИТЕЛЬ ЗАПРОСОВ	205
20.1	СОЗДАНИЕ НОВОГО ОПЕРАТОРА SELECT	205
20.2	ИЗМЕНЕНИЕ СУЩЕСТВУЮЩЕГО ОПЕРАТОРА SELECT	209
20.3	ИЗМЕНЕНИЕ ОПРЕДЕЛЕНИЯ ЗАПРОСА	209
20.4	ПРЕДПОЧТЕНИЯ ПОСТРОИТЕЛЯ ЗАПРОСОВ	211
20.5	МОДУЛИ РАСШИРЕНИЯ ПОСТРОИТЕЛЕЙ ЗАПРОСОВ	212
21.	ФОРМАТИРОВЩИК PL/SQL	213
21.1	ОПРЕДЕЛЕНИЕ ОПЦИЙ	213
21.2	ОПРЕДЕЛЕНИЕ ПРАВИЛ	214
21.3	ИСПОЛЬЗОВАНИЕ ФОРМАТИРОВЩИКА	215
22.	ШАБЛОНЫ	216
22.1	ОКНО ШАБЛОНОВ	216
22.2	ИСПОЛЬЗОВАНИЕ ШАБЛОНОВ	217
22.3	СОЗДАНИЕ И ИЗМЕНЕНИЕ ШАБЛОНОВ	218
23.	СПИСОК ОКОН	223
24.	ДОСКАВЛЕ И ПЛАВАЮЩИЕ ИНСТРУМЕНТЫ	224
25.	АВТОРИЗАЦИЯ	225
25.1	ВКЛЮЧЕНИЕ АВТОРИЗАЦИИ	225
25.2	ЗАДАНИЕ АВТОРИЗАЦИИ	226
25.3	ОТКЛЮЧЕНИЕ АВТОРИЗАЦИИ	227
26.	ФАЙЛОВАЯ СИСТЕМА ORACLE (OFS).....	228
26.1	МЕНЕДЖЕР OFS	228
26.2	ИСПОЛЬЗОВАНИЕ OFS	230
27.	СПРАВОЧНЫЕ СИСТЕМЫ.....	232
27.1	MS HELP ФАЙЛЫ	232
27.2	HTML РУКОВОДСТВА	232
28.	НАСТРОЙКА	236
28.1	ПРЕДПОЧТЕНИЯ	236

28.2	МАКЕТ ОКОН.....	236
28.3	ОПЕРАТИВНАЯ ДОКУМЕНТАЦИЯ	236
28.4	ПАРАМЕТРЫ КОМАНДНОЙ СТРОКИ.....	238
28.5	КЛЮЧЕВЫЕ СЛОВА SQL, PL/SQL, COMMAND, JAVA И XML.....	240
28.6	МОДУЛИ РАСШИРЕНИЯ	240

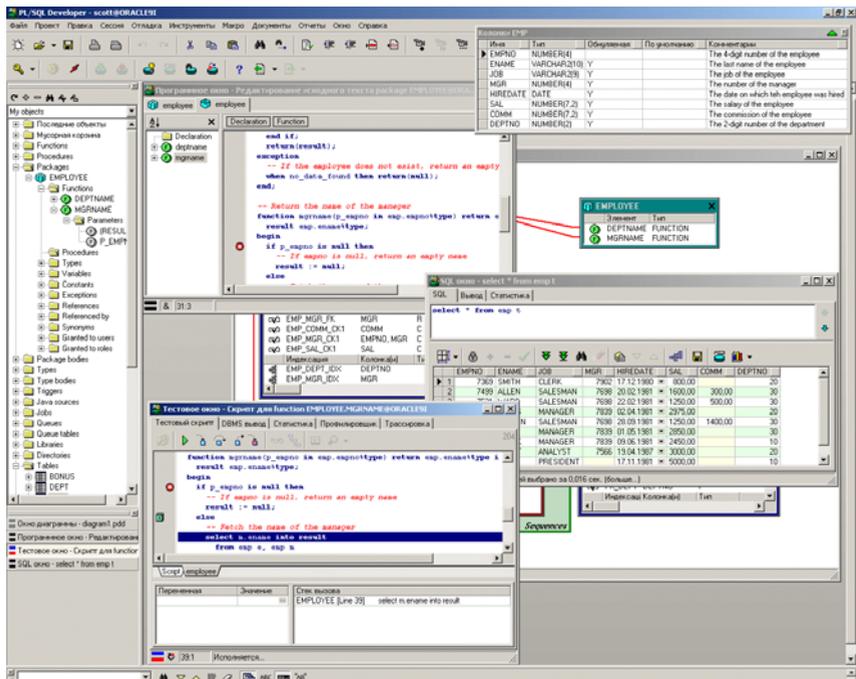
1. Введение

PL/SQL Developer является Интегрированной Средой Разработки ориентированной на хранение программные модули СУБД Oracle. Используя PL/SQL Developer вы можете создавать серверную часть вашего клиент-сервер приложения.

Рассматривая наилучший сценарий, по текущий момент вы могли работать следующим образом:

- Вы используете текстовый редактор для написания программных модулей (процедур, триггеров, и т.д.).
- Вы используете Oracle SQL*Plus для компиляции файлов с исходными текстами.
- Если произошла ошибка компиляции, вам необходимо найти, где она находится в файле с исходным текстом, исправить ее, переключиться обратно на SQL*Plus и перекомпилировать файл, только что бы найти следующую ошибку.
- Вы используете SQL*Plus или клиентскую часть своего приложения для тестирования программного модуля.
- В случае ошибки времени исполнения, вам предстоит опять сложное время обнаружения причины ошибки и ее исправления.
- Вы используете Explain Plan утилиту или tkprof для оптимизации ваших SQL команд.
- Для просмотра или модификации других объектов или данных в вашей базе данных, вы используете SQL*Plus или еще какое-либо средство.

Эти задачи – редактирование, компиляция, исправление, тестирование, отладка, оптимизация и исполнение запросов – могут быть выполнены, не покидая PL/SQL Developer IDE. Более того, PL/SQL Developer предлагает несколько других средств, которые могут быть полезны в ежедневной работе PL/SQL разработчика.



Редактирование

PL/SQL Developer, как и любая другая серьезная среда разработчика, подразумевает что Вы храните ваши исходные файлы на диске. Другие средства позволяют только редактировать исходные тексты в базе данных, но это не позволяет использовать существующие системы контроля версий или схемы развертывания приложений. Исходные файлы могут быть запущены, используя SQL*Plus, так что вы можете развернуть их на любой платформе без использования PL/SQL Developer. Вы можете редактировать несколько файлов одновременно, используя стандартный много документный интерфейс.

Редактор предлагает широкий диапазон видов помощи программисту. Там есть контекстно-зависимая помощь по SQL и PL/SQL командам. Мы все попадали в ситуацию: вы начинаете набирать *substr*, но забыли точное значение параметров. Теперь вы можете просто нажать *F1* и получить необходимый раздел из *Справочного Руководства по SQL*. Аналогично, описания таблиц, представлений и программных модулей могут быть представлены вам ниспадающем окне в редакторе. Ассистент Кодирования, интегрированный в редактор, автоматически отображает информацию об объекте базы данных, по ходу как вы набираете его имя, позволяя вам просмотреть и выбрать элементы из его описания. Для больших пакетов или реализаций типов, программный редактор предлагает древообразное представление исходного текста для упрощенной навигации, выделяет структуру кода и переменные, а так же гипертекстовую навигацию. Построитель запросов позволяет вам графически создавать SELECT команды. Расширяемый набор шаблонов PL/SQL Developer делает простым вставку стандартных SQL и PL/SQL конструкций в ваши программы. Все редакторы используют соответствующую SQL, PL/SQL или SQL*Plus схему цветового выделения, что бы сделать ваш исходный текст более читаемым.

Компиляция и исправление

Не покидая редактора, вы можете компилировать исходный файл. В случае ошибки компиляции редактор перейдет к соответствующей строке исходного текста. Все ошибки компиляции перечислены в списке внизу редактора. Этот список может включать кодовые подсказки, которые могут обозначать наличие стандартных программных ошибок или нарушение определенных пользователем соглашений о поименовании.

Поскольку вы можете компилировать исходный файл, не сохраняя его, вы можете безопасно испробовать различные альтернативы решения проблемы. Редактор помнит тот факт, что вы изменили файл без сохранения или компиляции.

Тестирование & отладка

Для тестирования программного модуля, Вы можете написать тестовый скрипт. Содержимое тестового скрипта является PL/SQL блок, в котором вы можете запрограммировать тестовый код. Любая переменная, которую вы хотите использовать в PL/SQL блоке, может быть объявлена, ей присвоено исходное значение и отображено после исполнения.

Когда вы исполняете тестовый скрипт, может произойти ошибка времени исполнения. В этом случае PL/SQL Developer предоставляет вам стек вызова места возникновения ошибки. Каждая исходная строка, присутствующая в стеке вызова подсвечена, так что вы можете просто отследить причину проблемы.

Если Вы используете Oracle 7.3.4 или более поздний, то можете применять интегрированный в PL/SQL Developer отладчик. Вы сможете пошагово исполнить код, устанавливать точки останова, просматривать / изменять переменные, видеть стек вызова и так далее.

В Oracle8i и более поздней версии вы можете дополнительно использовать возможности *dbms_trace* для регистрации выбранных событий исполнения программы. Это может помочь вам в анализе хода исполнения программы и исключений.

Вывод из вызовов через *dbms_output* и PL/SQL Web Toolkit пакеты автоматически отображается на соответствующей странице в Тестовом Окне.

Для регрессионного тестирования вы можете использовать Менеджер тестов, что бы быстро запускать и проверять набор Тестовых скриптов.

Оптимизация

В оптимизации SQL-операторов модулей вашей программы, очень может помочь утилита Oracle Explain Plan. Поэтому она интегрирована в PL/SQL Developer's IDE. Просто выделите SQL-оператор и нажмите F5, и план выполнения запроса отобразится в отдельном Окне плана выполнения. Потом вы можете изменить оператор, чтобы оптимизировать его план выполнения отдельно от исходного файла и после скопировать его обратно.

Вы можете также просмотреть статистику выполненных SQL операторов и программных модулей PL/SQL. Эта статистика включает в себя данные о времени исполнения, процессорном времени, операциях логического чтения, физического чтения, физической записи и т. д.

В Oracle8i имеется PL/SQL Профилировщик, который даёт вам возможность профилировать ваш PL/SQL код. Для каждой выполненной строки для PL/SQL кода вы можете определить время выполнения, и сколько раз она была выполнена.

Запрос данных

Чтобы запросить данные из базы данных, можно использовать Окно SQL для исполнения любого SQL оператора. Все выполненные операторы хранятся в историческом буфере, так что вы легко сможете выполнить их снова. Любые запрашиваемые результаты удобно отображаются в отдельной таблице, в которой впоследствии можно добавлять, изменять или удалять записи. Таблица результатов может дополнительно использоваться в режиме Запроса по образцу, так что вы легко сможете найти нужную вам информацию.

Чтобы запросить базу данных, вы можете воспользоваться Навигатором объектов. Все нужные свойства объектов базы данных, таких как таблицы, представления, последовательности, функции, процедуры, пакеты, типы и триггеры могут быть отображены, включая любые зависимости между объектами. Для простоты, навигатор построен по принципу дерева, схожего с проводником в Windows.

Исполнение SQL скриптов

PL/SQL Developer включает Командное окно, которое можно использовать для запуска SQL скриптов или выполнения SQL операторов. Вы можете дополнительно выполнять команды, которые очень схожи с SQL*Plus командами, с которыми вы можете быть знакомы.

Создание и изменение определений таблиц

Вы можете легко создавать и изменять описания таблиц, используя SQL-операторы. Заполните описание в диалоговом окне, чтобы изменить колонки, ограничения, индексы, привилегии, хранение информации, комментарии и т. д. Вы можете применить это описание к базе данных, а так же отобразить, изменить и сохранить получившийся SQL.

Диаграммы

Что бы отобразить объекты базы данных вашего приложения или проекта и их связи, вы можете создать диаграммы. Диаграмма может быть использована как для целей документирования, так и как рабочее пространство для работы со связанными объектами.

Создание отчётов

PL/SQL Developer поставляется с набором стандартных отчётов, созданных на основе HTML. Вы можете отобразить эти отчёты в PL/SQL Developer, напечатать или сохранить их как HTML файлы. Вы можете также создать свои собственные отчёты. Отчёты можно сделать доступными через меню отчётов.

Графики

Окно графиков может быть открыто из SQL окна или окна отчетов для получения графического представления запрошенных данных.

Проекты

Чтобы организовать свою работу вы можете использовать концепцию проектов, предложенную в PL/SQL Developer. Проект состоит из некоторого количества файлов и объектов баз данных. Эти объекты легко доступны при помощи Окна элементов проекта, и могут быть скомпилированы одним щелчком мыши.

Инструменты

PL/SQL Developer содержит несколько инструментов, которые могут быть полезны при разработке. Они включают Поиск объекта базы данных, позволяющий вам искать в исходных текстах объектов баз данных, инструмент Компиляция инвалидных объектов, с помощью которого можно быстро скомпилировать объекты, ставшими инвалидными в процессе разработки, инструменты Экспорт и Импорт, инструмент Генератор данных для создания тестовых данных, инструмент Экспорт объектов пользователя для экспорта ЯОД операторов объектов пользователя, инструмент Сравнение объектов пользователя для сравнения определений объектов 2 пользователей, инструмент Информация о сессии, Монитор событий, и инструмент Сравнение данных таблиц для сравнения и уравнивания содержимого таблиц.

В дополнение к этим стандартным инструментам, вы можете определить ваши собственные инструменты и включить их в меню инструментов PL/SQL Developer.

2. Установка

Существует два основных способа установки PL/SQL Developer:

- Установка на рабочую станцию. В этом случае, вы устанавливаете программное обеспечение отдельно на каждую рабочую станцию, на котором оно будет использоваться.
- Установка на сервер. В этом случае, вы устанавливаете программное обеспечение на сервер в папку, к которой есть доступ со всех рабочих станций, на которых оно будет использоваться.

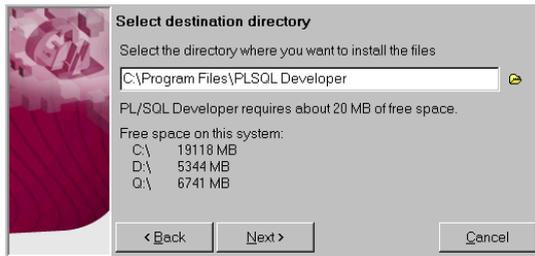
Оба способа будут описаны ниже.

2.1 Системные требования

PL/SQL Developer является 32-битным приложением Windows. Поэтому, на вашем компьютере должна быть установлена операционная система как минимум Windows 95 или Windows NT 4. Чтобы соединиться с базой данных Oracle, PL/SQL Developer необходима 32-битная версия SQL*Net, Net 8, Net 9 или Net 10.

2.2 Установка на рабочую станцию

Чтобы установить PL/SQL Developer на рабочую станцию, запустите программу установки с носителя данных. После этого появится следующий диалог:



Вы можете выбрать директорию для программных файлов, папку в стартовом меню, необходимость помещения ярлыка PL/SQL Developer на рабочий стол и установочные опции. После нажатия кнопки *Finish* на завершающей странице, копируются программные файлы и создаются ярлыки.

2.3 Установка на сервер

Для многопользовательских лицензий можно установить PL/SQL Developer на сервер и обеспечить доступ всем зарегистрированным пользователям. Для этого, вы можете запустить программу установки, также как описано в предыдущем пункте. Вы должны выбрать такую директорию, которая совместно используется всеми пользователями данного программного обеспечения. Во время первого запуска PL/SQL Developer на рабочем компьютере, этот тип установки определяется автоматически.

PL/SQL Developer может быть установлен в директорию, к которой пользователи имеют доступ только для чтения. Однако есть несколько исключений:

- Должна быть возможность записи в подкаталог *Macros*, так как это место, где будут размещены все макросы.

- Должна быть возможность записи в подкаталог Preferences, так как это место, где будут размещены наборы персональных предпочтений. Подробнее, смотрите главу 14.31.

2.4 Установка управляемая скриптом

В дополнение к процедуре установки, описанной выше, вы также можете создать скрипт для безоператорной установки. Для подробной информации прочтите прилагаемый *install.txt* файл.

2.5 Удаление PL/SQL Developer

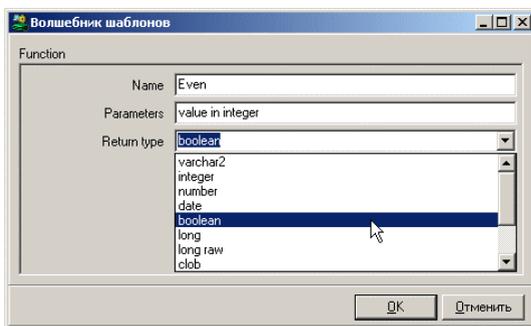
Если по какой-то причине вы захотите удалить PL/SQL Developer, вы сможете это сделать, используя Установку/Удаление программ в Панели управления.

3. Написание программ

В базе данных Oracle существуют пять разных типов хранимых программных модулей: функции, процедуры, пакеты, типы и триггеры. Редактор программ позволяет вам создавать и изменять эти пять типов программных модулей единым способом.

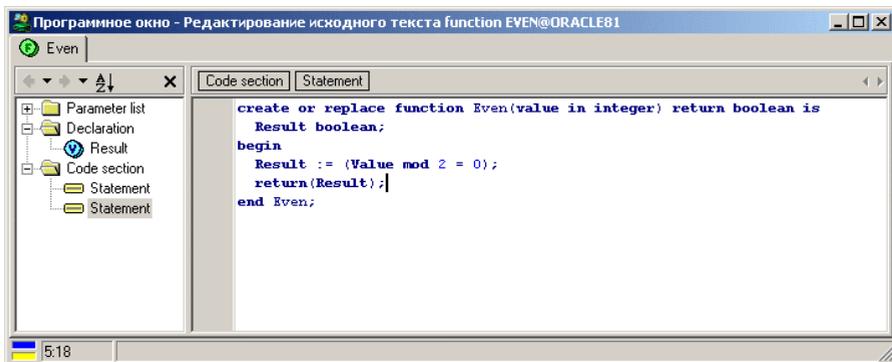
3.1 Создание программы

Чтобы создать новую программу, нажмите кнопку *Создать* на панели инструментов и выберите пункт *Программное окно* и, например, подпункт *Функция*. Теперь вам предложены различные переменные, относящиеся к данному типу программы. В этом случае, создавая функцию, мы вводим ее имя, список параметров и тип возвращаемого значения.



Название этого диалога подсказывает, что вся информация берётся из шаблона. PL/SQL Developer предоставляет набор стандартных шаблонов, которые при необходимости можно изменять. Вы можете также создавать новые шаблоны. Информация об их создании содержится в главе 22.

После того как вы ввели переменные и нажали кнопку *OK*, появляется Окно редактора программ, с шаблонной функцией в нём. Каждая программа, которую вы создаёте в Окне редактора программы, должна начинаться с обычного 'create or replace' SQL синтаксиса. В нашем случае, мы создаём функцию 'Even', и исходный текст, может выглядеть вот так:

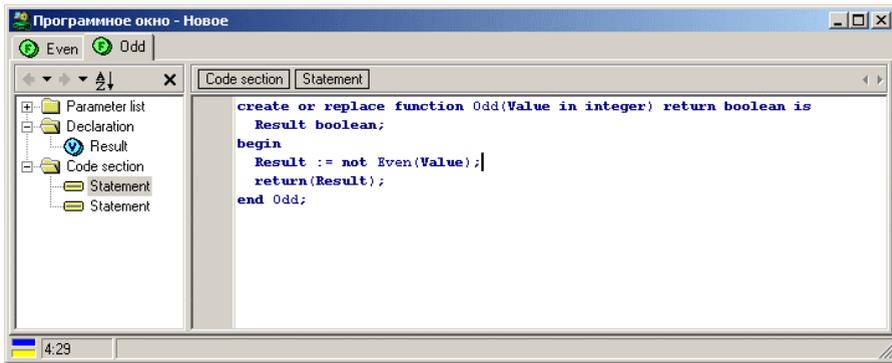


В левой стороне редактора вы видите Разделы кода, которые отображают структуру программного модуля. Это очень полезно при навигации в больших программных модулях, таких как тела пакетов или тела типов. Эта возможность детально описана в главе 18.11.

Над редактором вы видите Иерархию кода, которая отображает иерархию от позиции курсора. В нашем случае она показывает, что вы работаете с разделом *Statement* (оператор `return`), который является частью раздела *Sequence*, входящего в раздел *Function*. Для сложных программ может быть полезным видеть, где именно вы находитесь, и поможет вам выбирать отдельные части PL/SQL кода. Это детально описано в главе 18.14.

Программный файл может содержать более одного программного модуля. Щелкнув правой кнопкой мыши в окне Редактора программ, появляется всплывающее меню, позволяющее добавить или удалить программный модуль. Вы можете переключаться между модулями программы, выбирая соответствующие закладки вверху окна. Именно так вы можете удобно хранить несколько связанных между собой программных модулей в одном исходном файле. Спецификация и тело пакета являются хорошим примером данной возможности.

Программный модуль следует размещать в программном файле после любых других программных модулей, на которые он ссылается. Если вы создаёте функцию 'Odd', которая ссылается на ранее созданную функцию 'Even', то редактор программы будет выглядеть следующим образом:



Вы можете сохранить программный файл, нажав кнопку *Сохранить* на панели инструментов. Появляется диалог сохранения, показывающий набор стандартных расширений для программных файлов:

Тип программы	Расширение
Функция	.fnc
Процедура	.prc
Спецификация и тело пакета	.pck
Спецификация пакета	.spc
Тело пакета	.bdy
Спецификация и тело типа	.typ
Спецификация типа	.tps
Тело типа	.tpb
Триггер	.trg
Исходный текст Java	.jsp

Вы можете изменять эти расширения с помощью предпочтений, как описано в главе 16.27. Если в программный файл входит одна единственная функция, процедура, пакет, тип, триггер или исходный текст Java, то имя файла автоматически получается из имени и типа этого программного модуля. После сохранения файла, его имя отображается в заголовке окна.

Сохранённый программный файл имеет формат, совместимый с Oracle SQL*Plus. Например, исходный файл, содержащий функции 'odd' и 'even' выглядит так:

```
create or replace function Even(Value in integer) return boolean is
  Result boolean;
begin
  Result := (Value mod 2 = 0);
  return(Result);
end Even;
/
create or replace function Odd(Value in integer) return boolean is
  Result boolean;
begin
  Result := not Even(Value);
  return(Result);
end Odd;
/
```

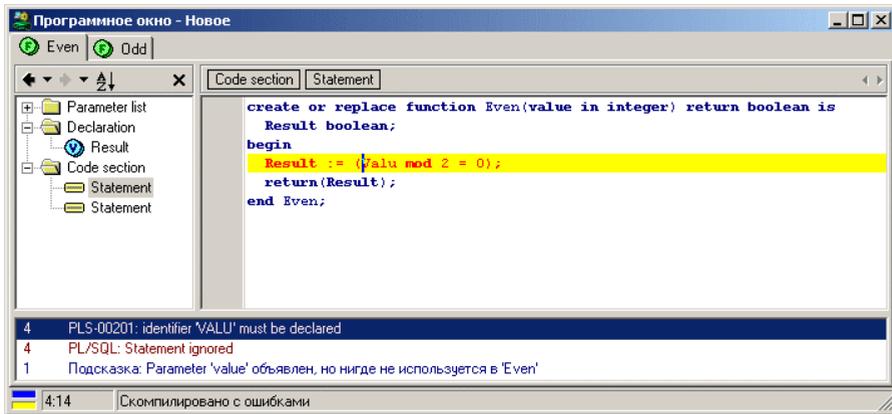
Когда этот файл будет исполняться SQL*Plus, то будут созданы обе функции.

3.2 Изменение программы

Вы можете открыть ранее сохранённый файл, нажав кнопку *Открыть* на панели инструментов и выбрав элемент *Программный файл*. Если файл использовался прежде, вы также можете открыть его, выбрав команду *Открыть повторно* в меню *Файл* или щёлкнув мышью по стрелке рядом с кнопкой *Открыть* на панели инструментов. Когда вы начинаете редактирование, в нижней части окна загорается синий индикатор. Это означает, что файл изменён, но ещё не сохранён. Существует также жёлтый индикатор, показывающий, что файл изменён, но ещё не скомпилирован.

3.3 Компилирование программы

Вы можете скомпилировать программу, нажав на панели инструментов кнопку *Выполнить*. Все программные модули в файле программы компилируются, начиная с первого, независимо от того, какой программный модуль выбран в данный момент. Когда встречается ошибка, компиляция прекращается, и редактор переходит к строке программы, которая вызвала ошибку:



Если в вашей программе несколько ошибок, и вы хотите исправить их, то щёлкните на следующем сообщении об ошибке в нижней части Редактора программы, чтобы перейти к месту её нахождения. Если вы сконфигурировали PL/SQL Developer для использования HTML документации, как это описано в главе 27.2, вы можете, сделав двойной щелчок на ошибке компиляции, отобразить соответствующий параграф в руководстве *Oracle Error Messages*.

Список ошибок компиляции также может содержать подсказки. Подсказки не приводят к аварийному завершению компиляции, но они могут указать на проблему, которая может вызвать ошибку времени выполнения (такие как сравнение с null, функция без возврата значения, неиспользуемый параметр и т.д.). Более подробно о подсказках смотрите главу 16.18.

Если сообщение об ошибке отобразилось после компиляции в окне сообщений, это означает, что оператор `create` был аварийно завершён без компиляции исходного текста программы. В сообщении об ошибке, должна быть объяснена причина. Если, например, допустить синтаксическую ошибку в операторе `'create or replace'`, в окне сообщений будет сказано: «ORA-00900: Инвалидный SQL оператор».

Примечание: Если вы используете Oracle Server 7.2 или более раннюю версию, то сообщение об ошибке компиляции триггера всегда появляется в окне сообщений. Только начиная с Oracle Server 7.3, ошибки компиляции триггера сообщаются также как и в случае с процедурами, функциями, пакетами и типами.

3.4 Подстановочные переменные

Если ваш Программный файл включает подстановочные переменные, подобно тому как они используются в SQL*Plus (начинаются со знака `&`), вы можете нажать кнопку `&` в нижней левой части окна. В исходных текстах программного файла будут осуществлен поиск подстановочных переменных, для которых позже вы можете ввести значение. Если предпочтение *Сохранять состояние окна* включено (см. главу **Ошибка! Источник ссылки не найден.**), значения этих подстановочных переменных будут сохранены и восстановлены если программный файл или объект базы данных будет открыт позже для просмотра или редактирования.

4. Тестирование программ

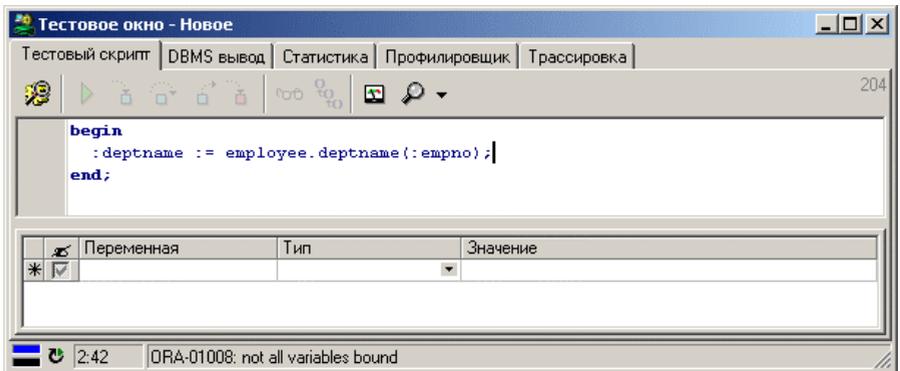
После удачной компиляции программы, вам нужно протестировать её. Чтобы выполнить это, вы можете использовать Тестовые скрипты PL/SQL Developer. Тестовый скрипт позволит вам выполнить один или более программных модулей, определить входные, выходные и входные / выходные переменные, отобразить и присвоить значения переменным. Если во время выполнения Тестового скрипта встречается ошибка времени выполнения, вы можете отобразить исходные тексты программных модулей, которые повлекли за собой ошибку.

Если Вы используете Oracle 7.3.4 или более позднюю версию, вы можете использовать интегрированный отладчик PL/SQL Developer. Вы можете шаг за шагом проверить ваш программный код, установить контрольные точки, просмотреть и изменить переменные, просмотреть стека вызова и тому подобное.

4.1 Создание Тестового скрипта

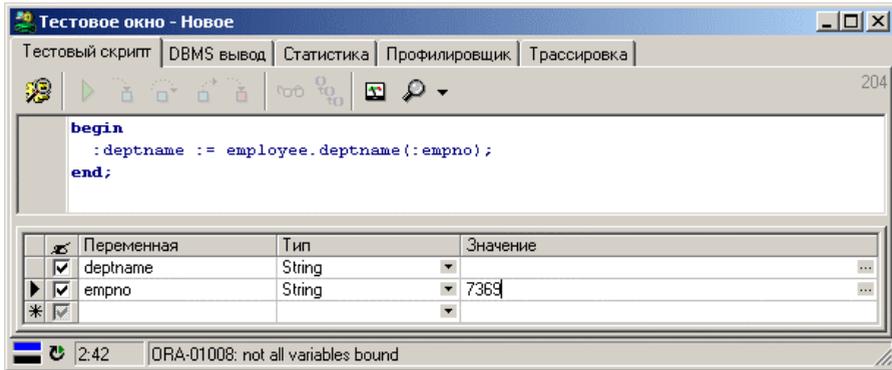
Есть два способа создания Тестового скрипта. Вы можете щёлкнуть правой кнопкой мыши на программном модуле (отдельно стоящей функции или процедуре, пакетной функции или процедуре или методе объектного типа) в Навигаторе и выбрать элемент *Тестировать*. Будет создан новый Тестовый скрипт с переменными для параметров и результата. Для пользы этого руководства, мы собираемся создать Тестовый скрипт вручную с самого начала.

Чтобы создать новый пустой Тестовый скрипт, нажмите кнопку *Создать* на панели инструментов и выберите пункт *Тестовое окно*. Теперь вы можете набрать PL/SQL блок со знакомым "declare...begin...end" синтаксисом в теле Тестового скрипта. В качестве примера, мы собираемся протестировать пакет *employee*, который реализует набор *employee*-функций, используя демонстрационные таблицы *employee / department*. Функция *deptname* выдаёт название отдела, в котором работает служащий, и может быть протестирована при использовании следующего PL/SQL блока:



PL/SQL блок содержит простой вызов функции, которую мы хотим протестировать, и использует переменные *deptname* и *empno*. Нам необходимо присвоить значение переменной *empno*, и после выполнения проверить значение переменной *deptname*, чтобы определить корректность значения возвращенного функцией.

При добавлении двоеточия в качестве префикса к именам этих переменных, они могут быть объявлены в нижней части Тестового скрипта:

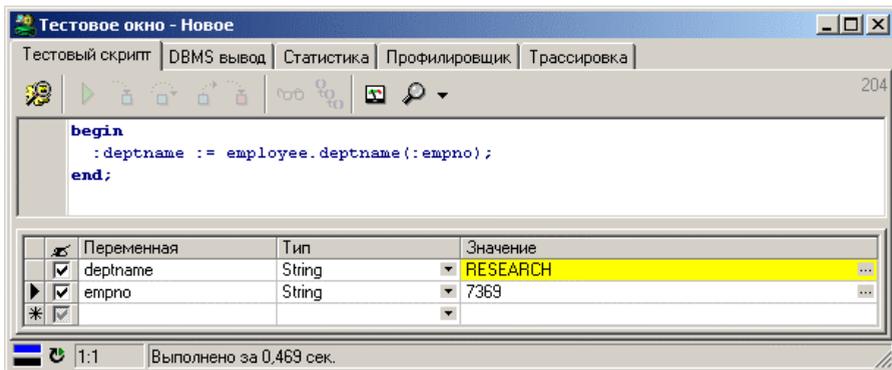


Кнопка *Найти переменные* (U) в левом верхнем углу списка переменных может быть использована для быстрого копирования переменных из исходного текста. После этого, переменной *empno* присваивается целочисленный тип данных и величина 7369 (мистер Смит из исследовательского отдела). Теперь мы готовы к выполнению скрипта.

Вы также можете объявить переменные локально в DECLARE части PL/SQL блока, но у вас не будет возможности изменить или отобразить величины этих переменных. Преимуществом этих локальных переменных, является тот факт, что вы можете использовать типы записей, PL/SQL таблицы, типы и т. д.

4.2 Выполнение Тестового скрипта

Чтобы выполнить Тестовый скрипт, нажмите кнопку *Выполнить* на панели инструментов. После выполнения, значения переменных будут обновлены. Любая переменная, значение которой изменилось, выделяется жёлтым фоном:



Результатом функции является величина 'RESEARCH', так что, очевидно, функция выполняется корректно. В нижней части окна, отображается время выполнения. Эта информация может быть использована, чтобы оптимизировать производительность вашего кода. Так же для целей оптимизации вы можете отобразить статистику выполнения PL/SQL блока, выбрав закладку Статистики. Вы можете также создать отчёт о профилировании для каждой выполненной строки

кода PL/SQL, нажав кнопку *Создать отчет профилировщика* перед выполнением скрипта. После выполнения вы можете переключиться на страницу *Профилировщик*, чтобы просмотреть отчет. Отчеты статистики и профилировщика описаны в 5.2 и 5.3 соответственно.

Вы можете прервать выполнение Тестового скрипта, нажав кнопку *Прервать*, которая особенно полезна, когда программа застревает в бесконечном цикле, или выполнение длится дольше, чем ожидалось. Заметьте, что нажатие кнопки *Прервать* не всегда заканчивается успешно. Если, например, программа ожидает блокировку, то она не будет откликаться на сигнал прерывания.

После выполнения скрипта, транзакция может быть активна, так как выполненные программные модули могли начать транзакцию. В этом случае, на панели инструментов будут включены кнопки *Зафиксировать* и *Откатить*. Чтобы узнать больше о транзакциях, смотрите главу 14.

4.3 Типы переменных

Список типов в секции переменных Тестового окна содержит все типы, которые могут быть использованы. Характеристики этих типов объясняются в данной главе.

Тип	Описание
Integer	Может быть использован для числовых величин в диапазоне от -2^{32} до $2^{32} - 1$.
Float	Может быть использован для других числовых величин
String	Эквивалент данных типа varchar2 в таблице. Может достигать 2000 (в Oracle7) или 4000 (в Oracle8) символов. Ячейка с величиной этого типа данных содержит кнопку, которая вызывает текстовый редактор. Таким образом, вы можете легко отображать и изменять многострочные переменные данного типа.
Date	Тип данных даты и времени
Long	Величина типа данных Long не отображается в таблице, а только выводит надпись <Long>. Нажмите кнопку в ячейке, чтобы вызвать текстовый редактор.
Long Raw	Нажатие кнопки в ячейке с величиной Long Raw, вызовет диалог импорта/экспорта файла, позволяющий вам загрузить или сохранить содержимое этой переменной.
Cursor	Этот тип данных может быть использован там, где вы можете использовать тип переменных cursor в PL/SQL. После выполнения Тестового скрипта, вы можете нажать кнопку ячейки, чтобы отобразить набор результатов в окне SQL.
CLOB, BLOB & BFile	Переменные типа LOB локатор должны быть инициализированы на сервере, перед тем как вы сможете отобразить данные LOB. CLOB вызовет текстовый редактор, BLOB и BFile вызовут диалог импорта/экспорта файла.
PL/SQL String	Используйте этот тип данных для величин PL/SQL varchar2 размером до 32000 символов.
Char	Строковая переменная фиксированной длины (дополненная пробелами)
Substitution	Подстановочные переменные могут быть использованы без ограничений присущих связанным переменным. Имя переменной в тексте PL/SQL блока заменяется её значением, перед тем как текст будет отослан на сервер.
Temporary CLOB &	Эти LOB локаторы могут содержать данные еще до того как они будут переданы на сервер Тестовым скриптом.

BLOB	
------	--

Логические переменные

Выбирая тип переменной в списке, вы заметите, что логический тип данных отсутствует. Это потому, что SQL*Net не поддерживает этот тип данных. Чтобы использовать логическую переменную, вы должны объявить её как переменную типа Integer и применить стандартные Oracle функции *sys.diutil.bool_to_int* и *sys.diutil.int_to_bool*. Эти функции преобразуют null/true/false в null/0/1 и наоборот. Если вы используете функцию *Тестировать* в Навигаторе, то это преобразование будет сгенерировано автоматически.

4.4 Сохранение Тестовых скриптов

Чтобы сохранить Тестовый скрипт, нажмите кнопку *Сохранить* на панели инструментов. Диалог сохранения Тестовых скриптов по умолчанию использует расширение *.lst*, однако, вы можете сменить расширение по умолчанию, используя предпочтение, описанное в главе 16.27. PL/SQL блок, все переменные (имя, тип и значение) а так же наблюдаемые отладочные выражения (смотрите главу 4.10) будут сохранены. Сохранив Тестовый скрипт, вы легко можете заново протестировать программный модуль, если позже вы измените его.

Вы можете открыть ранее сохранённый Тестовый скрипт, нажав на панели управления кнопку *Открыть* и выбрав *Тестовый скрипт*, который создаст новое Тестовое окно. Альтернативно вы можете щёлкнуть правой кнопкой мыши на ранее созданном Тестовом окне и выбрать элемент *Загрузить*, который откроет Тестовый скрипт в существующем Тестовом окне.

Сохранение скрипта в формате SQL*Plus

Кроме способа сохранения описанного выше, вы можете сохранить Тестовый скрипт в формате совместимом с SQL*Plus. Это позволит вам запустить скрипт в среде, в которой невозможно использовать PL/SQL Developer (например, Unix сервер). Для этого, в диалоге выбора файла выберите *SQL*Plus script (*.sql)* как тип файла. Пример скрипта из предыдущей главы был бы сохранён так:

```

rem PL/SQL Developer Test Script

set feedback off
set autoprint off

rem Declare variables
variable result varchar2(2000)
variable p_empno number

rem Set variables
begin
  :result := null;
  :p_empno := 7369;
end;
/

rem Execute PL/SQL Block
begin
  :result := employee.deptname(:p_empno);
end;
/

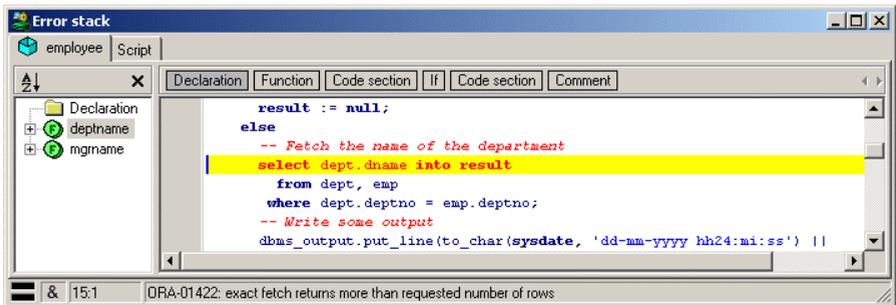
rem Print variables
print result
print p_empno

```

Примечание: Вы не можете открыть скрипт SQL*Plus как Тестовый скрипт. Если вы хотите снова использовать его в Тестовом окне, убедитесь, что сохранили его как Тестовый скрипт!

4.5 Отслеживание ошибок времени выполнения

Всякий раз, когда ваша программа вызывает ошибку времени выполнения, PL/SQL Developer спросит вас, желаете ли вы отобразить исходные тексты стека ошибки. Так образом, вы можете быстро найти причину ошибки. Если бы наша функция *deptname* содержала ошибку, окно стека ошибки выглядело бы так:



В верхней части окна вы видите закладку для каждого программного модуля, имеющего отношение к ошибке времени выполнения. Программные модули были вызваны в порядке слева направо. Перебирая закладки, вы можете легко найти поток исполнения программы, который ведёт к ошибке. В нашем случае он показывает, что Тестовый скрипт вызвал функцию *employee.deptname*. Очевидно, что причина ошибки может быть более сложной и даже быть связанной с триггерами.

Примечание: если вы используете Oracle Server 7.2 или более раннюю версию, исходный текст триггера не будет показан в окне стека ошибки.

Стек ошибки выделяет красным каждую строку, которая имеет отношение к ошибке. Строка последнего программного модуля в стеке является причиной ошибки. Для всех остальных программных модулей, это строка, где был сделан вызов следующего программного модуля.

4.6 Состояния пакетов и Java сессий

Когда вы редактируете, компилируете и тестируете пакеты, состояния пакетов будут удаляться Oracle после каждой компиляции. Значения всех глобальных переменных в пакете будут сброшены, и блок инициализации будет выполнен снова. Это может привести к неожиданным результатам. Всякий раз, когда PL/SQL Developer обнаруживает подобные ситуации, в строке статуса Тестового окна появляется предупреждение: «Предупреждение: ORA-04068: существующее состояние пакета было удалено».

Так же, если вы редактируете и компилируете исходный текст Java, существующее состояние Java сессии будет удалено. PL/SQL Developer также обработает эту ситуацию и отобразит в строке статуса: «Предупреждение: ORA-29549: класс <имя класса> изменён, состояние Java сессии удалено».

4.7 Просмотр набора результатов

В Тестовом скрипте вы не ограничены PL/SQL блоками. Вы также можете выполнять отдельные SQL операторы, включая, SELECT операторы. При выполнении оператора SELECT появится дополнительная страница *Результаты*, которая отображает все выбранные строки.

4.8 Просмотр dbms_output

Для отладки иногда может быть необходимо вывести на экран некоторую информацию из программного модуля. Для этих целей Oracle создал пакет *dbms_output*. При вызове *dbms_output.put_line*, информация помещается в буфер вывода. После выполнения Тестового скрипта, вы можете выбрать в верхней части Тестового окна закладку *DBMS вывод*, чтобы увидеть содержимое буфера вывода. На этой странице вы можете дополнительно установить размер буфера вывода или включить/отключить буферизацию. Установки по умолчанию на странице вывода контролируются предпочтениями вывода как это описано в главе 16.4.

4.9 Просмотр HTTP вывода

Если вы тестируете программные модули, использующие PL/SQL Web Toolkit, то будет отображена страница *HTTP вывод*, если он был сгенерирован:



Если HTTP вывод отсутствует, то эта страница не будет отображена.

4.10 Отладка

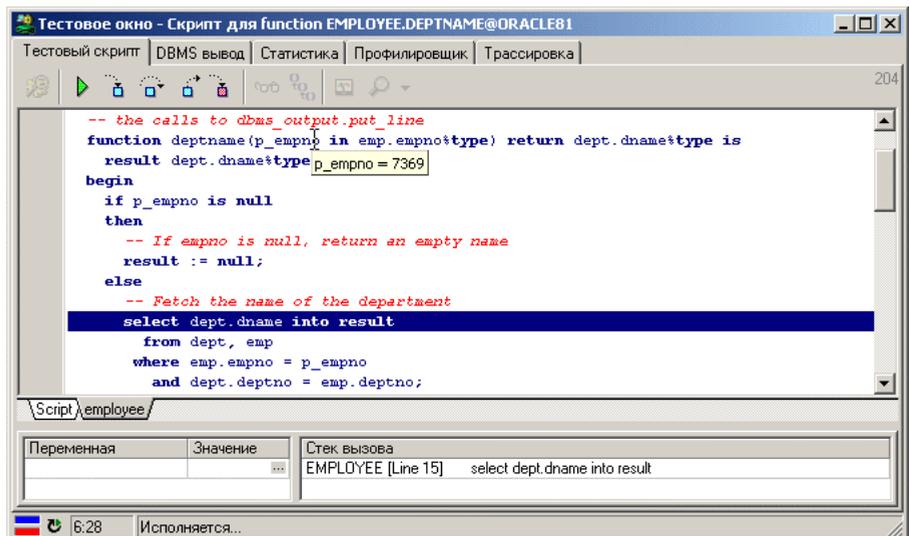
Для тех ошибок программирования, которые действительно трудно отследить, Тестовое окно предлагает интегрированный отладчик. В верхней части окна вы найдёте панель инструментов с функциями, относящимися к отладчику. Чтобы начать отладочную сессию, нажмите вместо кнопки *Выполнить* на основной панели инструментов, кнопку *Выполнить* в левой части отладочной панели инструментов. Теперь будут включены другие кнопки, и вы готовы к отладке.

Управление выполнением

После запуска отладчика, выполнение будет приостановлено перед первым оператором в Тестовом скрипте. После этого, вы можете контролировать выполнение при помощи кнопок на отладочной панели инструментов:

-  Выполнять скрипт до завершения
-  Войти в процедуру, функцию или вызов метода в следующей строке. Если следующая строка содержит операторы обновления, вставки или удаления, которые могут быть причиной срабатывания триггера, вы перейдёте в данный триггер.
-  Пройти следующую строку. Данная команда будет выполнена, но вы не войдете в исходный текст программы.
-  Выйти из текущего программного модуля.
-  Выполнять до тех пор, пока не встретится исключительная ситуация. Выполнение будет приостановлено на той строке, которая вызвала исключительную ситуацию. После следующего шага исключительная ситуация будет сгенерирована.

Всякий раз, когда вы входите в программный модуль, его исходный текст будет автоматически загружен в Тестовое окно. Нижняя часть панели редактора теперь будет содержать закладки для каждого программного модуля, так что вы сможете легко переключаться между ними, чтобы видеть их исходный текст, устанавливать/удалять контрольные точки и так далее. Щёлкнув правой кнопкой мыши на панели редактора, вы можете удалить программный модуль из Тестового окна, если он вас больше не интересует:



Просмотр и изменение значений переменных

Чтобы наблюдать значение переменной в течение отладочной сессии, вы можете просто передвинуть курсор мыши к переменной в исходном тексте программы. Через ½ секунды её значение появится во всплывающем окне. Переменные в PL/SQL блоке Тестового окна никогда не могут быть отображены. Сложные переменные (например, запись и объектные типы) также не

могут быть отображены, однако вы можете просмотреть отдельные поля. Будущие версии Oracle могут снять эти ограничения.

Чтобы задать значения переменной, щёлкните правой кнопкой мыши по её имени в исходном тексте программы и выберите элемент *Изменить переменную* во всплывающем меню. Появляется поле ввода с текущим значением переменной. Вы можете ввести новое значение и нажать *Enter*, чтобы принять его, или *Esc*, чтобы отменить операцию:



Используя это же всплывающее меню, вы также можете добавить переменную в список просмотра, что означает, что после каждого шага отладки, значение переменной будет автоматически отображено и обновлено в контрольном списке в левой нижней части Тестового окна.

Если переменная представляет собой коллекцию (PL/SQL таблица, переменная типа *varray* или вложенная таблица) данных скалярного типа, вы можете просмотреть полный набор данных, щёлкнув на ней правой кнопкой мыши и выбрав элемент *Просмотреть коллекцию* из всплывающего меню.

Заметьте, что значения переменных могут быть отображены и заданы только в том случае, если программные модули скомпилированы с отладочной информацией. Существует предпочтение, позволяющее при каждой компиляции включать отладочную информацию. Вы можете добавлять отладочную информацию вручную, щёлкнув правой кнопкой мыши по программному модулю и выбрав во всплывающем меню элемент *Добавить отладочную информацию*.

Использование точек останова

Точки останова могут быть использованы, чтобы остановить выполнения программы на конкретной строке в вашем PL/SQL коде. Когда выполнение остановлено, вы можете просмотреть и изменить переменные, по шагам пройти код и так далее. Вы можете задать условие для точки останова, в этом случае, выполнение будет остановлено только тогда, когда будет выполнено это условие. Для каждой точки останова вы можете определить сообщение, которое будет отображаться в окне вывода, когда точка останова будет достигнута.

Установка точек останова

Существуют два способа, установки точек останова: в Программном окне или в Тестовом окне. В обоих случаях вам нужно просто щёлкнуть на нужной строке в левом поле редактора. Появится значок точки останова, показывающий, что точка останова установлена на этой строке. Когда вы выполняете Тестовое окно в режиме отладки, выполнение будет остановлено, если встретится одна из точек останова.

Когда вы устанавливаете точку останова в Программном окне, может случиться так, что этот конкретный программный модуль ещё не был скомпилирован в базе данных. Поэтому, подобная точка останова ещё не может быть использована в базе данных. Если вы впоследствии скомпилируете программный модуль, точка останова будет принята, и соответствующий значок будет изменён, чтобы отразить это. Любые ранее установленные в этом программном модуле точки останова, будут иметь силу до тех пор, пока программный модуль не скомпилирован.

Вы не можете устанавливать точки останова в PL/SQL блоке в Тестовом окне.

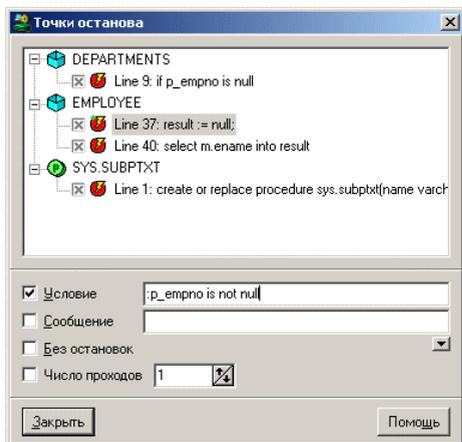
Чтобы удалить точку останова, просто снова щёлкните на значке точки прерывания.

Условия точек останова

Иногда вы устанавливаете точку останова на строке, которая выполняется очень часто, хотя вас интересует только состояние программы при определённых обстоятельствах. В этом случае вы

можете определить условие для точки останова. Исполнения программы будет прерываться, только когда это условие будет удовлетворено.

Чтобы определить условие для контрольной точки, щёлкните правой кнопкой мыши на значке и выберите во всплывающем меню элемент *Изменить точки останова*. Появится следующий диалог:



В этом диалоге вы видите все программные модули, имеющие точки останова. В нем под каждым модулем помещен список принадлежащих ему точек останова. Для каждой точки останова существует флажок, позволяющий включить или отключить её. Секция, расположенная в нижней части окна, отображает следующие поля:

- **Условие** – Флажок включает или выключает условие. Условие должно быть логическим выражением. Когда достигнута строка, содержащая точку останова, выполнение приостанавливается, только если условие истинно. Вы можете использовать в условии любое SQL выражение, вы также можете использовать любую переменную, которая известна в месте нахождения контрольной точки. Это те же самые переменные, которые вы можете просмотреть или установить в течение интерактивной отладки. Перед переменной следует ставить двоеточие. Например, `upper (:ename) = 'SMITH'` является допустимым условием, если переменная `ename` является допустимой в месте расположения контрольной точки.
- **Сообщение** – Флажок включает или отключает сообщение. Когда достигается строка с точкой останова, и если условия выполнены, то сообщение будет помещено на страницу вывода.
- **Без остановок** – Этот флажок полезен только, если вы установили флажок **Сообщение**. Когда установлен флажок **Без остановок**, исполнение не прерывается на строке с точкой останова. Таким образом, вы можете определять точки останова, которые только генерируют сообщения для страницы вывода.
- **Число проходов** – Этот флажок включает или выключает счетчик проходов, который определяет, сколько раз должна быть пройдена строка с точкой останова, до того как исполнение будет прервано. Если, например, счетчик проходов равен 10, исполнение будет прерываться на каждом 10 проходе через строку с точкой останова.

Просмотр стека вызова

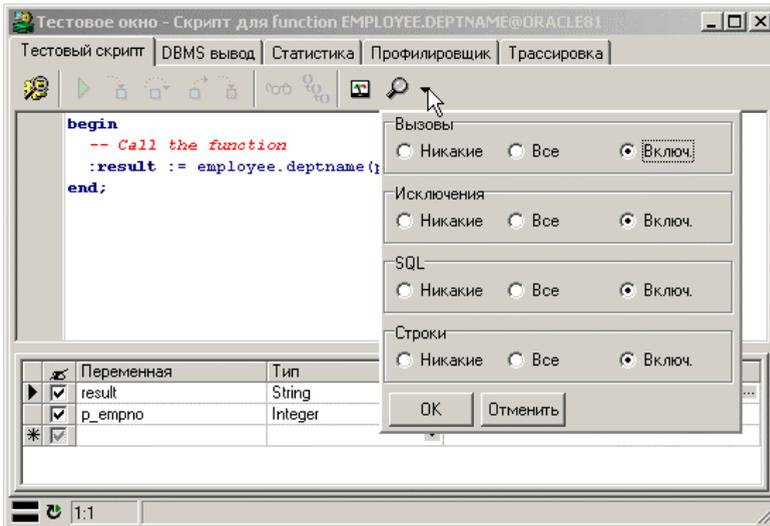
Стек вызова может быть отображен в правой нижней части Тестового окна. Он автоматически обновляется после каждого отладочного шага.

Предпочтения отладчика

В пункте *Предпочтения* меню *Инструменты* вы можете найти секцию предпочтений относящихся к отладчику. Эти предпочтения подробно описаны в главе 16.3.

4.11 Трассировка исполнения

Если вы используете Oracle8i или более позднюю версию, вы можете применить Средства трассировки Тестового окна, чтобы отследить исполнение вашего PL/SQL кода. Вы можете указать события, которые вы хотите отслеживать: вызовы, исключения, SQL или даже каждую выполненную строку вашего PL/SQL кода. Нажмите кнопку *Выбрать уровень трассировки* рядом с кнопкой *Создать трассировочный отчет*, чтобы вывести следующий конфигурационный экран:



Вы можете контролировать, необходимо ли вам отслеживать особые события в каждом программном модуле (*Все*), только в тех программных модулях, которые были скомпилированы с включенной отладочной информацией (*Включ.*) или в никаких (*Никакие*).

Чтобы создать Трассировочный отчёт, просто нажмите кнопку *Создать трассировочный отчет* на панели инструментов Тестового окна и выполните ваш Тестовый скрипт. Чтобы просмотреть текущий отчёт, а также отчёты о предыдущих запусках, вы можете перейти на страницу *Трассировка*.

Комментарий	Время события	Строка модуля	Текст строки модуля	Исключения
PL/SQL Virtual Machine started	22.07.2005 19:58:24			
PL/SQL Virtual Machine started	22.07.2005 19:58:24			
Procedure Call	22.07.2005 19:58:24			
New line executed	22.07.2005 19:58:24			
New line executed	22.07.2005 19:58:24			
New line executed	22.07.2005 19:58:24	9	if p_empno is null	
New line executed	22.07.2005 19:58:24	15	select dept.dname into result	
SELECT DEPT.DNAME FROM DEP	22.07.2005 19:58:24	15	select dept.dname into result	
Exception raised	22.07.2005 19:58:24	15	select dept.dname into result	1403
Exception handled	22.07.2005 19:58:24	29	when no_data_found then return(null);	1403
New line executed	22.07.2005 19:58:24	29	when no_data_found then return(null);	
New line executed	22.07.2005 19:58:24	30	end;	
Return from procedure call	22.07.2005 19:58:24	30	end;	
PL/SQL Virtual Machine started	22.07.2005 19:58:24			

В этом (простом) отчёте вы можете видеть, что оператор SELECT в строке 15 функции *employee.deptname* вызвал исключение 1403 (данные не найдены), которое было обработано в строке 29 в этом же программном модуле.

Для каждого трассировочного события, Вы можете включить в отчёт информацию, такую как описание события, время, программный модуль, номер строки, исключение и т. д. Чтобы настроить это, нажмите кнопку *Предпочтения* в левой верхней части панели инструментов Трассировки. Это вызовет страницу предпочтений трассировки, как это описано в главе 16.5.

Чтобы просмотреть старые трассировочные отчёты, выберите предыдущий трассировочный запуск в списке выбора *Выполнить*. Нажмите кнопку *Удалить запуск*, чтобы удалить выбранный запуск.

За дополнительной информацией о Средствах трассировки Oracle, обратитесь к главе *DBMS_TRACE* руководства *Oracle Supplied Packages Reference*.

4.12 Регрессионное тестирование

Что бы использовать Тестовые скрипты для регрессионного тестирования, вы можете использовать инструмент Менеджер тестов (см. главу 17.15), что бы формировать и запускать Тестовые наборы основанные на Тестовых скриптах.

5. Оптимизация

В оптимизации SQL-операторов ваших программных модулей, очень может помочь утилита Oracle Explain Plan, т. к. она показывает путь выполнения оператора. Поэтому она интегрирована в PL/SQL Developer's IDE.

Чтобы просмотреть ресурсы, используемые SQL оператором или программным модулем PL/SQL, PL/SQL Developer может вывести статистику об его выполнении. Вы можете настроить, какую статистику вы хотите отображать, вы можете включить в неё время исполнения, процессорное время, логические чтения, физические чтения, физические записи и т. д.

Чтобы определить время выполнения каждой отдельной строки кода PL/SQL, вы можете использовать Профилировщик PL/SQL. Эта возможность не доступна в Oracle 8.0 и более ранних версиях.

Наконец, вы можете использовать утилиту Oracle *tkprof*, чтобы получить информацию об использовании ресурсов всеми выполненными в программном модуле SQL операторами, путём включения SQL Trace.

5.1 Использование Окна плана выполнения

Для использования утилиты План выполнения Oracle, нажмите кнопку *Создать* на панели инструментов и выберите *Окно плана выполнения*. Появляется пустое *Окно плана выполнения*. В верхней половине окна вы можете набрать SQL оператор, который вы хотите проанализировать. После нажатия кнопки *Выполнить* на панели инструментов, в нижней половине окна появляется план исполнения.

Цель оптимизатора: First rows

Description	Владелец объекта	Имя объекта	Стоимость	Мощность	Байты
SELECT STATEMENT, GOAL = FIRST_ROWS			2	1	48
NESTED LOOPS			2	1	48
TABLE ACCESS BY INDEX ROWID	SCOTT	EMP	1	1	26
INDEX RANGE SCAN	SCOTT	EMP_PRIMARY_1	1		
TABLE ACCESS BY INDEX ROWID	SCOTT	DEPT	1	82	1804
INDEX UNIQUE SCAN	SCOTT	DEPT_PRIMARY	82		

Select a range of values from an index in ascending order

Теперь вы можете изменить SQL оператор и снова нажать кнопку *Выполнить*, чтобы увидеть, как повлияли внесённые вами изменения. Для большей информации о планах выполнения, вы можете прочитать руководство Oracle *Server Tuning*.

Чтобы увидеть, как сказываются различные цели оптимизатора на плане запроса, выберите соответствующий элемент из списка целей оптимизатора *Цель оптимизатора*.

Используйте кнопки *Первая*, *Предыдущая*, *Следующая* и *Последняя операция* для навигации в плане запроса в порядке следования операций. После того как план определен, будет выделена первая операция.

Утилита Explain Plan для сохранения плана исполнения использует так называемую таблицу плана. Если подобная таблица не доступна для текущего пользователя, PL/SQL Developer спросит вас, хотите ли вы создать эту таблицу в схеме данного пользователя. Чтобы определить, какие колонки таблицы плана вы хотите отображать, и в каком порядке, нажмите кнопку *Предпочтения*. После этого, на экране появится соответствующая страница предпочтений (смотрите главу □).

Примечание: Если вы используете Oracle Server 7.2 или более раннюю версию, колонки «Стоимость» и «Мощность» не доступны в таблице плана. Если вы используете Oracle Server 7.3 или более позднюю версию, и эти разделы не отображаются, вероятно, вам нужно обновить структуру таблицы плана.

Если вы щёлкните правой кнопкой мыши на колонке *Имя объекта*, для выбранного объекта появится всплывающее меню.

Анализ SQL в программном модуле

Большую часть времени, SQL операторы, которые вы хотите проанализировать, находятся в исходном тексте программного модуля. Чтобы проанализировать их, щёлкните левой кнопкой на SQL операторе в Программном редакторе и выберите *Построить план выполнения* из всплывающего меню или нажмите F5. Для выбранного SQL оператора будет создано новое *Окно плана выполнения*, и все PL/SQL переменные будут заменены связанными переменными.

Если вы явно копируете и вставляете SQL оператор, который содержит PL/SQL переменные из программного модуля, и вы выполняете Окно плана выполнения, вы получите ошибку "ORA-00904: Некорректное имя колонки". Утилита Explain Plan ошибочно предполагает, что все эти переменные являются колонками. Поэтому перед каждой переменной следует ставить двоеточие. Если бы мы хотели проанализировать SQL оператор в функции *employee.deptname*, переменные *result* и *p_deptno* были бы изменены следующим образом:

Description	Владелец объекта	Имя объекта	Стоимость	Мощность	Байты
SELECT STATEMENT, GOAL = FIRST_ROWS			2	1	48
NESTED LOOPS			2	1	48
TABLE ACCESS BY INDEX ROWID	SCOTT	EMP	1	1	26
INDEX RANGE SCAN	SCOTT	EMP_PRIMARY_1	1		
TABLE ACCESS BY INDEX ROWID	SCOTT	DEPT	1	82	1804
INDEX UNIQUE SCAN	SCOTT	DEPT_PRIMARY		82	

Теперь утилита Explain Plan знает, какой идентификатор является колонкой, а какой – PL/SQL переменной.

5.2 Автоматический сбор статистики

Когда вы выполняете оператор в SQL окне или в Тестовом окне, PL/SQL Developer автоматически создаёт отчёт о статистике этого выполнения. Единственное условие - вам

необходимо иметь привилегии чтения динамических таблиц `v$session`, `v$statname` and `v$sesstat` (предоставляются через стандартную роль `plustrace`).

Вы можете отобразить статистический отчёт, перейдя на страницу статистики в верхней части SQL окна или Тестового окна. Статистика выполнения функции `employee.deptname` может выглядеть следующим образом:

Имя	Последнее	Итоговое
physical reads	0	0
physical writes	0	0
table scans (short tables)	1	3
table scans (long tables)	0	0
table scan rows gotten	1	3
table scan blocks gotten	1	3
table fetch by rowid	3	6
sorts (memory)	0	0
sorts (disk)	0	0
sorts (rows)	0	0
session logical reads	11	37
CPU used by this session	0	0

Выполнено за 0,016 сек.

Для каждого элемента в Таблице статистики вы можете видеть значение последнего выполнения и итоговое значение для текущей сессии. Вы можете сконфигурировать, какая статистика и когда будет отображаться, установив предпочтение, как это описано в главе 16.2. Набор статистики по умолчанию, наиболее подходящий для настройки вашего SQL и PL/SQL, описан ниже:

Статистика	Значение
CPU used by this session	Использование ЦПУ в сотых секунды
Physical reads	Количество блоков прочитанных с диска
Physical writes	Количество блоков записанных на диск
session logical reads	Количество блоков прочитанных из буфера или с диска
sorts (disk)	Количество сортировок выполненных во временном сегменте
sorts (memory)	Количество сортировок выполненных в памяти
sorts (rows)	Количество отсортированных записей
table fetch by rowid	Количество записей, выбранных по ROWID, обычно в результате доступа к индексу
table scan blocks gotten	Количество блоков прочитанных в результате полного просмотра таблицы
table scan rows gotten	Количество записей прочитанных в результате полного просмотра таблицы
table scans (long tables)	Количество полных просмотров длинных таблиц
table scans (short tables)	Количество полных просмотров коротких таблиц

Какую ещё статистику вы можете включить в отчёт, зависит от версии Oracle Server и не описано в данном руководстве. Информацию об этом вы можете найти в книгах, посвященных настройке Oracle. Также это вкратце описано и в *Oracle Server Reference*.

Статистика может быть экспортирована в CSV файл (Comma Separated Values – значения разделённые запятой), который позже может быть открыт в электронной таблице. Просто щёлкните правой кнопкой мыши на данных статистики, выберите элемент *Экспорт* и далее выберите элемент *CSV файл*. Вы также можете выбрать формат файла TSV, XML или HTML или скопировать этот файл в буфер вырезанного изображения.

5.3 Профилировщик PL/SQL

Профилировщик PL/SQL является очень мощным инструментом, который поможет вам оптимизировать ваш PL/SQL код. Для каждой выполненной строки кода будет определено общее время, максимальное время, минимальное время, среднее время и число вхождений.

Профилировщик доступен в Тестовом окне. Перед выполнением Тестового скрипта нажмите кнопку *Создать отчет Профилировщика* на панели инструментов Тестового окна. Если вы впоследствии выполняете скрипт, вы можете переключиться на страницу *Профилировщик*, чтобы просмотреть отчёт.

Следующий пример отчёта показывает, что выполнение оператора SELECT в функции *employee.deptname* заняло 63 миллисекунды, а 3 вызова *dbms_output.put* заняли 24 миллисекунд:

Модуль	Строк	Полное время	Вхождения	Текст
ANONYMOUS BLOCK	1		10	2
ANONYMOUS BLOCK	4		11	2
ANONYMOUS BLOCK	5		1	1
EMPLOYEE	9		0	1 if p_empno is null
EMPLOYEE	15	63		1 select dept.dname into result
EMPLOYEE	20	24		3 dbms_output.put_line(to_char(sysdate, 'dd-mm-yyyy hh24:miss')
EMPLOYEE	24		1	1 return(result);
EMPLOYEE	28		1	1 end;

Выполнено за 0,016 сек.

По умолчанию страница *Профилировщик* выведет отчёт о последнем запуске. Вы также можете выбрать предыдущие запуски из списка *Запуск*. Список *Модуль* позволяет вам углубиться в детали запуска отдельного программного модуля.

Колонка *Полное время* графически представляет относительное время строки в сравнении со строкой с наибольшим временем. Это позволит вам быстро определить самые дорогостоящие строки. Отчёт можно отсортировать при помощи кнопок сортировки в заголовках колонок.

Если исходная строка выделена красным, то это означает, что с того момента, как был создан профильный отчёт, программный модуль был изменён. И по сравнению с тем моментом, когда был построен профиль, строка кода может быть другой.

Нажатием кнопки *Предпочтения* вы можете изменять различные аспекты макета Отчёта профилировщика. Появится соответствующая страница предпочтений, как это описано в главе 16.6.

Для большей информации о Профилировщике PL/SQL, смотрите главу *dbms_profiler* в руководстве “*Oracle8i Supplied Packages Reference*”.

Примечание: не все платформы предоставляют одинаково точную информацию о времени.

5.4 Трассировка SQL

Вы можете включить SQL трассировку, нажав на панели инструментов кнопку *Создать трассировочный отчет*. После этого любая серверная обработка, вызванная Тестовым или SQL окном, будет зарегистрирована в файле трассировки на сервере базы данных. Единственное условие – значение параметра временной статистики экземпляра базы данных, который вы используете, должно быть установлено в True.

Вы можете отключить SQL трассировку, нажав кнопку *Создать трассировочный отчет* снова.

Вы можете просмотреть информацию о файле трассировки, используя утилиту Oracle *tkprof* на сервере базы данных. Она выдаст отчет о времени выполнения, процессорном времени, вводе/выводе и т. д. для каждого программного модуля. Чтобы узнать больше об утилите Oracle *tkprof*, вы можете прочитать руководство "*Oracle 7 Server Tuning*".

Отчет *tkprof* о выполнении функции *employee.deptname* может выглядеть следующим образом:

```
TKPROF: Release 7.2.2.3.1 - Production on Fri Sep 26 14:59:08 1997
```

```
Copyright (c) Oracle Corporation 1979, 1994. All rights reserved.
```

```
Trace file: ora07087.trc
Sort options: default
```

```
*****
count      = количество исполнений процедуры OCI
cpu         = процессорное время в секундах
elapsed    = время выполнения в секундах
disk        = количество физических чтений буферов с диска
query      = количество буферов полученных для согласованного чтения
current    = количество буферов, полученных в согласованном режиме (обычно для обновления)
rows       = число строк, обработанных вызовом выборки или исполнения
*****
```

```
begin
  :deptname := employee.deptname(:empno);
end;
```

call	count	cpu	elapsed	disk	query	current	rows
Parse	1	0.00	0.00	0	0	0	0
Execute	1	0.01	0.01	0	0	0	1
Fetch	0	0.00	0.00	0	0	0	0
total	2	0.01	0.01	0	0	0	1

```
Misses in library cache during parse: 0
```

```
Optimizer hint: CHOOSE
```

```
Parsing user id: 16
```

```
SELECT DEPT.DNAME
FROM
  DEPT,EMP WHERE EMP.EMPNO = :b1 AND DEPT.DEPTNO = EMP.DEPTNO
```

call	count	cpu	elapsed	disk	query	current	rows
Parse	1	0.00	0.00	0	0	0	0
Execute	1	0.00	0.00	0	0	0	5
Fetch	1	0.06	0.06	4	4	0	1
total	3	0.06	0.06	4	4	0	6

```
Misses in library cache during parse: 0
```

```
Optimizer hint: CHOOSE
```

```
Parsing user id: 16 (recursive depth: 1)
```

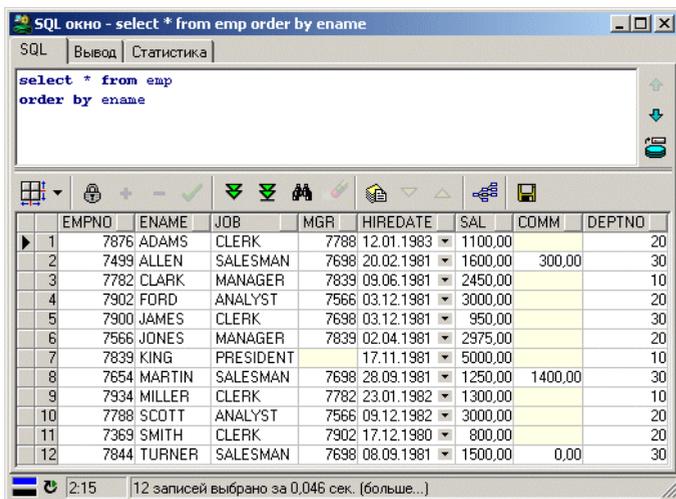
```
*****
```

6. Незапланированный SQL запрос

Часто в течение разработки программы, чтобы проверить SQL оператор, или чтобы просмотреть данные в таблице, или чтобы обновить данные, вам нужно выполнить несколько SQL операторов. Вы можете сделать в PL/SQL Developer, используя окно SQL.

6.1 Использование SQL окна

Чтобы создать новое SQL окно, нажмите на панели инструментов кнопку *Создать* и выберите *SQL окно*. Появится пустое SQL окно, в котором вы можете набрать SQL оператор. Если это оператор SELECT, окно разделяется на две части, и результаты отображаются в таблице в нижней половине. Если мы хотим просмотреть список всех работников в таблице, окно SQL будет выглядеть так:

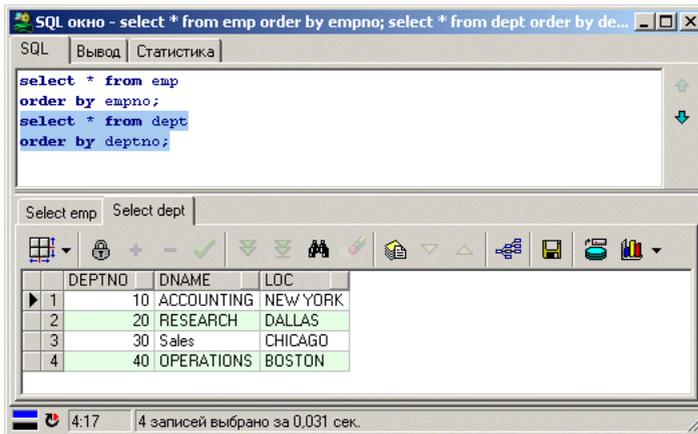


The screenshot shows a window titled "SQL окно - select * from emp order by ename". The window has a menu bar with "SQL", "Вывод", and "Статистика". Below the menu bar is a text area containing the SQL query: "select * from emp order by ename". To the right of the text area are several icons for execution and navigation. Below the text area is a toolbar with various icons. At the bottom of the window is a table with 12 rows and 9 columns. The columns are EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, and DEPTNO. The table contains data for 12 employees. The status bar at the bottom of the window shows "2:15" and "12 записей выбрано за 0,046 сек. (больше...)"

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
▶	1	7876 ADAMS	CLERK	7788	12.01.1983	1100,00		20
	2	7499 ALLEN	SALESMAN	7698	20.02.1981	1600,00	300,00	30
	3	7782 CLARK	MANAGER	7839	09.06.1981	2450,00		10
	4	7902 FORD	ANALYST	7566	03.12.1981	3000,00		20
	5	7900 JAMES	CLERK	7698	03.12.1981	950,00		30
	6	7566 JONES	MANAGER	7839	02.04.1981	2975,00		20
	7	7839 KING	PRESIDENT		17.11.1981	5000,00		10
	8	7654 MARTIN	SALESMAN	7698	28.09.1981	1250,00	1400,00	30
	9	7934 MILLER	CLERK	7782	23.01.1982	1300,00		10
	10	7788 SCOTT	ANALYST	7566	09.12.1982	3000,00		20
	11	7369 SMITH	CLERK	7902	17.12.1980	800,00		20
	12	7844 TURNER	SALESMAN	7698	08.09.1981	1500,00	0,00	30

SQL оператор выполнен, и выбрано 10 записей. Результирующий набор больше, чем 10 записей, это видно по тому, что на панели инструментов набора данных включены обе кнопки *Следующая страница* и *Последняя страница*, а в строке статуса отображено (больше...). Только 10 записей выбрано изначально, т. к. это количество записей, которое может быть отображено в таблице. При нажатии кнопки *Следующая страница*, будут выбраны следующие 10 записей и т.д. При нажатии кнопки *Последняя страница*, будут выбраны все записи.

Если у вас несколько SQL команд в SQL редакторе и вы нажали кнопку *Выполнить*, все команды будут выполнены, и результаты будут отображены на отдельных закладках:



SQL команды должны завершаться знаком точкой с запятой (;) или обратной косой чертой (/), а PL/SQL блоки должны завершаться обратной косой чертой, так же как и в SQL*Plus. Если вы выбираете закладку с результатом, соответствующий текст в SQL редакторе будет подсвечен.

Если вы выделите часть текста в SQL редакторе, будет выполнена только выделенная часть текста. Таким образом, вы можете помещать в редактор несколько операторов и выполнять их один за другим.

Если вы выполняете SQL оператор, завершение которого занимает много времени, вы можете нажать кнопку *Прервать*, чтобы остановить его выполнение. Заметьте, что реакция на нажатие кнопки *Прервать* не всегда будет немедленной. Если, например, оператор ожидает блокировку, то он не отреагирует на сигнал о прерывании. Если вы нажали кнопку *Прервать*, когда окно SQL выбирает записи, оно просто остановится и отобразит уже выбранные строки. Вы можете продолжить выбор записей при помощи кнопок *Следующая страница* и *Последняя страница*.

Заметьте, что предпочтения SQL окна позволяют вам задать для оператора SELECT начальное количество выбранных строк. Это описано в главе 16.8.

В правой части окна вы видите две кнопки, которые позволяют вам перемещаться между всеми SQL операторами, которые вы ввели в SQL окне. Таким образом, вы можете заново выполнить ранее введенные операторы.

В целях оптимизации вы можете отобразить статистику выполнения SQL оператора, выбрав закладку *Статистика*. Статистика объяснена в главе 5.2.

Когда вы распечатываете SQL окно, будут напечатаны SQL оператор и таблица результатов. Выделяя записи в таблице результатов, вы можете ограничить количество записей, которые будут напечатаны.

6.2 Действия с таблицей результатов

Таблица результатов SQL окна может быть обработана несколькими способами. Некоторые типы ячеек имеют специальное поведение, строки, колонки и области ячейки могут быть выделены и напечатаны, колонки могут быть перемещены, строки отсортированы, вы можете переключиться в режим просмотра отдельной записи и т.д.

Как распознать нулевое значение

Нулевые значения выделяются жёлтым фоном, и вы легко сможете отделить их от остальных значений. Также бывает полезным распознать нулевые значения в ячейках, в которых значение переменной не отображается непосредственно, такие типы переменных как Long и LOB. При помощи предпочтений вы можете изменить цвет для выделения ячеек с нулевым значением (смотрите главу 16.8).

Отображение колонок с массивными данными

Величины в колонках типов Long, Long Raw, CLOB, BLOB и BFILE не отображаются в таблице результатов. Вместо этого выводится просто <Long>, <Long Raw>, <CLOB>, <BLOB> и <BFILE>:

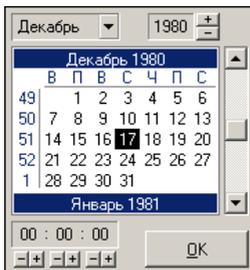
BCC_ADDRESS
<BLOB>
<BLOB>
<BLOB>
PI PR

Когда вы щёлкаете на кнопке в ячейке такой колонки, вызывается Редактор массивных данных, который позволяет вам просматривать или редактировать данные в различных форматах. Для более детального описания смотрите главу 19.

Также вы можете щёлкнуть кнопку в ячейке колонки, чтобы вызвать Редактор массивных данных. Такая кнопка присутствует в ячейке, только если размер колонки больше 20 символов.

Просмотр колонок с данными

В колонке с данными есть кнопка, которая выводит на экран календарь с выделенной текущей датой. Если у даты есть раздел, показывающий время, вы также можете отобразить её в календаре:



Просмотр колонок содержащих временные метки

Колонки содержащие временные метки отображаются в формате, который указан в переменных реестра NLS_TIMESTAMP_FORMAT and NLS_TIMESTAMP_TZ_FORMAT вашего Oracle Home.

Просмотр XML данных

Существует несколько способов разместить XML данные в базе данных: как CLOB (Oracle8i), как XMLTYPE (Oracle9i), или даже как колонки с переменными типов Varchar2 или Long. В любом случае Редактор массивных данных распознает XML формат, если эти данные начинаются со стандартного XML заголовка, и переключится в режим отображения XML формата. Для большей информации смотрите главу 19.

Просмотр колонок содержащих вложенные курсоры

Если вы включаете вложенный курсор в список полей оператора SELECT, значение колонки первоначально будет выведено как <Cursor>. После нажатия кнопки в ячейке появится новое SQL окно с таблицей результатов содержащей результирующий набор курсора. Это может быть использовано при просмотре простых вложенных запросов. Заметьте, что каждое значение типа вложенный курсор неявно приводит к открытию курсора. Таким образом, большой результирующий набор может достигнуть ограничения OPEN_CURSORS и стать причиной ошибки "ORA-01000: превышено максимальное кол-во открытых курсоров".

Выделение столбцов, строк и ячеек

Чтобы выделить строки или столбцы в таблице, просто щёлкните на заголовке строки или колонки и перетащите курсор, чтобы выделить нужное:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL
1	7876 ADAMS	CLERK	7788 12-1-1983	1100,00	
2	7499 ALLEN	SALESMAN	7698 20-2-1981	1600,00	
3	7782 CLARK	MANAGER	7839 9-6-1981	2450,00	
4	7902 FORD	ANALYST	7566 3-12-1981	3000,00	
5	7900 JAMES	CLERK	7698 3-12-1981	950,00	
6	7566 JONES	MANAGER	7839 2-4-1981	2975,00	
7	7839 KING	PRESIDENT	17-11-1981	5000,00	
8	7654 MARTIN	SALESMAN	7698 28-9-1981	1250,00	
9	7934 MILLER	CLERK	7782 23-1-1982	1300,00	
10	7788 SCOTT	ANALYST	7566 9-12-1982	3000,00	

Выделенная колонка может быть перемещена, для этого нужно отпустить кнопку мыши, щёлкнуть снова на заголовке колонки и перетащить выбранную колонку на новое место расположения.

Чтобы выделить определённую группу ячеек, передвиньте курсор к левому краю ячейки, пока не изменится её вид, нажмите кнопку мыши и тащите мышью, чтобы выделить нужные вам ячейки:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL
1	7876 ADAMS	CLERK	7788 12-1-1983	1100,00	
2	7499 ALLEN	SALESMAN	7698 20-2-1981	1600,00	
3	7782 CLARK	MANAGER	7839 9-6-1981	2450,00	
4	7902 FORD	ANALYST	7566 3-12-1981	3000,00	
5	7900 JAMES	CLERK	7698 3-12-1981	950,00	
6	7566 JONES	MANAGER	7839 2-4-1981	2975,00	
7	7839 KING	PRESIDENT	17-11-1981	5000,00	
8	7654 MARTIN	SALESMAN	7698 28-9-1981	1250,00	
9	7934 MILLER	CLERK	7782 23-1-1982	1300,00	
10	7788 SCOTT	ANALYST	7566 9-12-1982	3000,00	

Чтобы выделить все строки и столбцы, нажмите левую верхнюю ячейку или щёлкните правой кнопкой мыши на таблице и выберите во всплывающем меню пункт *Выбрать все*.

Выделенные объекты могут быть копированы или напечатаны как обычно.

Экспорт данных

Существуют несколько способов экспорта данных из результирующей таблицы. После выполнения оператора SELECT, вы можете выделить группу ячеек, как это описано выше, щёлкнуть на ней правой кнопкой мыши и выбрать во всплывающем меню пункт *Экспортировать результаты*. Появится подменю, в котором вы можете выбрать, в каком

формате следует экспортировать данные: CSV (Comma Separated Values – значения, разделённые запятой), TSV (Tab Separated Values – значения, разделённые символом табуляции), HTML или XML. После выбора формата, вы можете указать экспортируемый файл.

Другой способ, вы можете скопировать выделенные объекты в буфер обмена, нажав *Ctrl-C* или щёлкнув правой клавишей мыши на выделенных объектах и выбрав во всплывающем меню пункты *Копировать* или *Копировать с заголовком*. Далее вы можете вставить эти данные в другое приложение, например в электронную таблицу, текстовый редактор или другое.

Чтобы быстро перенести результирующий набор информации в Microsoft Excel, выберите элемент *Копировать в Excel*. Откроется новое окно Excel, и все выделенные данные будут скопированы в него.

Сортировка строк

Чтобы отсортировать в результирующей таблице строки, нажмите кнопку в заголовке колонки, строки которого вы хотите отсортировать:

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL
4	7902	FORD	ANALYST	7566	3-12-1981	3000,00
10	7788	SCOTT	ANALYST	7566	9-12-1982	3000,00
1	7876	ADAMS	CLERK	7788	12-1-1983	1100,00
9	7934	MILLER	CLERK	7782	23-1-1982	1300,00
5	7900	JAMES	CLERK	7698	3-12-1981	950,00
11	7369	SMITH	CLERK	7902	17-12-1980	800,00
3	7782	CLARK	MANAGER	7839	9-6-1981	2450,00
6	7566	JONES	MANAGER	7839	2-4-1981	2975,00
7	7839	KING	PRESIDENT		17-11-1981	5000,00
12	7844	TURNER	SALESMAN	7698	8-9-1981	1500,00

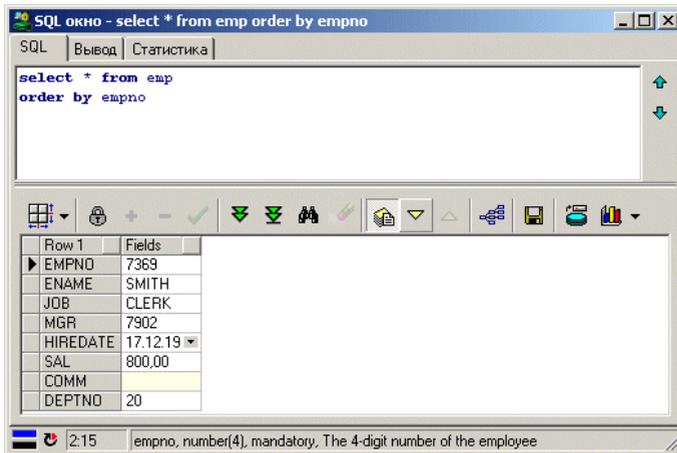
Строки будут отсортированы в восходящем порядке, как показывает кнопка в заголовке. При повторном нажатии кнопки в заголовке, строки будут отсортированы в нисходящем порядке. Если вы нажмёте эту кнопку в третий раз, сортировка будет отменена. Нажатие кнопки в заголовке другой колонки отсортирует строки по этой колонке. Предыдущая колонка сортировки будет использована в качестве колонки вторичной сортировки (обозначен точкой на кнопке заголовка). В приведённом выше примере, столбец *job* является столбцом первичной сортировки, а столбец *hiredate* – столбцом вторичной сортировки.

Заметьте, что сортировка проводится локально, и только для тех строк, которые уже были выбраны. Если вы выбираете дополнительные строки после сортировки результатов, эти новые строки будут добавлены в конце результирующей таблицы без какой-либо сортировки. Для больших результирующих наборов данных локальная сортировка может занять много времени. В этом случае, лучше будет использовать предложение *order by* в операторе SELECT, и позволить Oracle Server выполнить сортировку самому.

Просмотр одной записи

Если результирующий набор данных содержит много столбцов, может быть неудобно, что каждая запись выводится отдельной строкой. Вам приходится прокручивать вперёд-назад, чтобы увидеть связанные колонки (даже, несмотря на то, что вы можете перемещать столбцы), и не возможно просмотреть сразу все колонки записи.

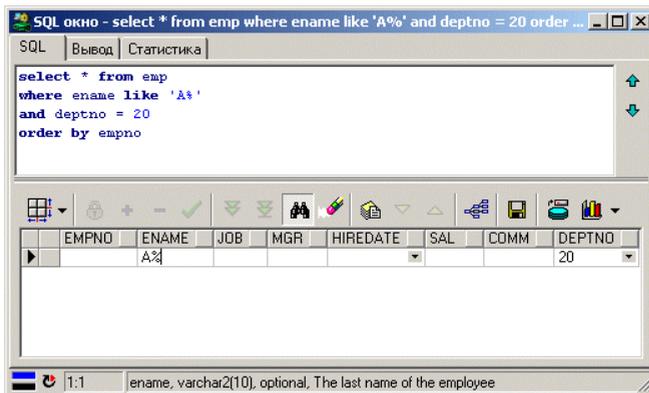
Нажав на панели инструментов таблицы *Однозаписный просмотр*, вы сможете просматривать одну запись в каждый момент времени:



Теперь каждая строка выводит имя и значение одной колонки. Кнопки *Следующая запись* и *Предыдущая запись* на панели инструментов таблицы могут быть использованы для навигации в результирующем наборе данных. Чтобы обратно переключиться в режим Просмотра набора записей, снова нажмите кнопку *Однозаписный просмотр*.

6.3 Режим Запроса по образцу

После выполнения оператора SELECT, вы можете нажать кнопку *Запрос по образцу* над результирующей таблицей, если вы хотите найти нужные вам записи. При нажатии этой кнопки таблица будет очищена, и в ней останется одна пустая запись, где вы можете ввести запрашиваемые вами величины. Если, например, вы хотите найти всех работников департамента 20, с фамилией начинающейся на А, введите А% в столбце *ename* и 20 в столбце *deptno*:



При нажатии кнопки *Выполнить* или повторном нажатии кнопки *Запрос по образцу*, будет выполнен изменённый запрос и выведен отфильтрованный результат. Чтобы продолжить работу с предыдущими запрошенными значениями, нажмите снова кнопку *Запрос по образцу*. Чтобы очистить запрашиваемые значения, нажмите кнопку *Очистить запись*.

Запрашиваемые значения не ограничены одиночными значениями с символами шаблона. Вы можете использовать следующие выражения (в скобках приведены альтернативные выражения)

- value [= value]
- value with wildcards [like value with wildcards]
- > value
- < value
- != value [<> value]
- in (value1, value2, ...)
- between value1 and value2
- null [is null]
- not null [is not null]

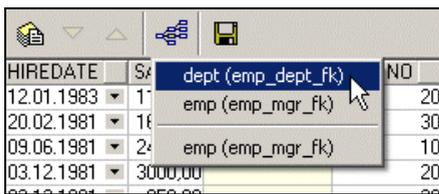
Заметьте, что если вы используете символьное значение, вы можете пропускать кавычки для значений (например, SMITH в SQL тексте автоматически будет преобразовано в 'SMITH'). Для всех остальных выражений вы должны использовать символьное значение, понятное для Oracle Server (например, != 'SMITH'). Это так же применимо для значений типа число или дата. Если вы используете их в выражении, используйте формат понятный для Oracle Server.

Существует несколько предпочтений, которые позволяют вам контролировать установки по умолчанию функциональности Запроса по образцу (например, чувствительность к регистру). Эти предпочтения описаны в главе 16.8.

6.4 Связанные запросы

Просматривая результирующий набор, вы часто хотите запросить связанную таблицу. Например, при просмотре таблицы *dept*, вам, возможно, захочется запросить список всех служащих департамента, или, наоборот, просматривая таблицу *emp*, вы, возможно, захотите запросить название департамента, в котором работает служащий.

В большинстве случаев между родительской и дочерней таблицами будет ограничение внешнего ключа. В этом случае SQL окно может автоматически создавать и выполнять подобные запросы для вас. Если вы находитесь в ячейке записи результирующего набора данных, нажмите на панели инструментов таблицы кнопку *Связанный запрос*. При этом появится всплывающее меню со всеми главными и дочерними таблицами текущего оператора SELECT. Если, например, вы запрашиваете таблицу *emp*, появится следующее всплывающее меню:

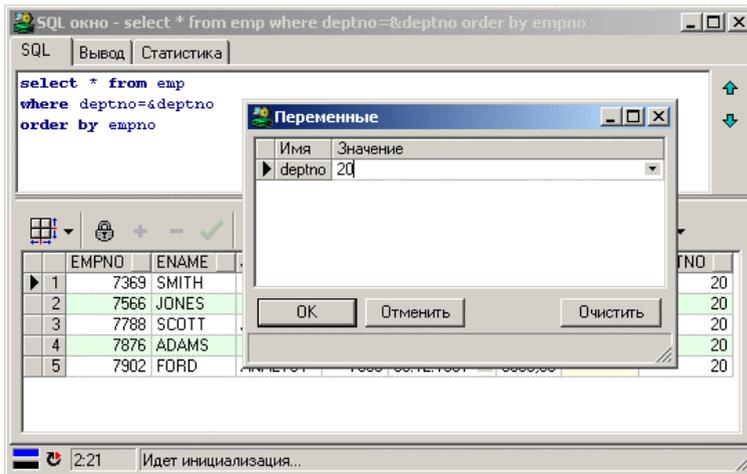


В верхней секции отображены родительские таблицы (и названия внешних ключей), а в нижней – дочерние таблицы. Элемент *dept* создаст запрос о служащих этого департамента (30). Первый (родительский) элемент *emp* создаст запрос о начальнике этого служащего (7839). Второй элемент (дочерний) *emp*, создаст запрос обо всех служащих, находящихся в подчинении у данного служащего.

Связанный запрос будет выполняться в том же SQL окне, до тех пор, пока вы не включите предпочтение *Связанный запрос в новом окне* (смотрите главу 16.8). Если, нажимая кнопку *Связанный запрос*, вы удерживаете клавишу *Ctrl* нажатой, произойдет обратное данному предпочтению.

6.5 Подстановочные переменные

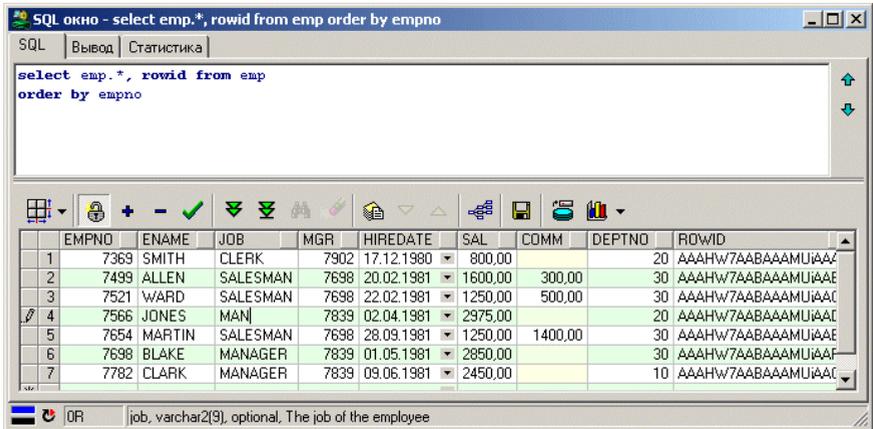
Чтобы разрешить пользовательский ввод при исполнении запроса, вы можете использовать подстановочные переменные в вашем SQL тексте. Наиболее простая форма подстановочной переменной схожа с той, что вы можете использовать в SQL*Plus:



При указании `&deptno` в SQL тексте, у вас будет запрошено значение этой переменной. Это значение будет подставлено в указанную переменную в SQL тексте перед его выполнением. Вы можете указать несколько подстановочных переменных, и вы можете несколько раз указывать одну и ту же подстановочную переменную. В дальнейшем, вы можете определить тип данных, значение по умолчанию, списки выбора, запросы списка, блоки контроля и другие продвинутые опции. Эти особенности также могут быть использованы в Окне отчёта, они подробно описаны в главе 9.

6.6 Обновление базы данных

Чтобы обновить, вставить или удалить записи в базе данных, вы можете выполнить в SQL окне подходящие DML операторы. Вероятно, намного удобнее сделать результирующую таблицу обновляемой, включив `rowid` в список выбора или применив оператор `select ... for update`:



Вам следует помнить, что оператор `select ... for update` заблокирует все выбранные записи, поэтому включение `rowid` может быть в общем случае наилучшим способом сделать результирующую таблицу обновляемой. Если оператор `SELECT` является соединением, столбцы первой таблицы могут быть изменены. Все остальные столбцы будут доступны только для чтения. Столбцы, которым дан алиас, в любом случае не могут быть обновлены.

Если результирующая таблица обновляется, вы можете нажать в правой части результирующей таблицы кнопку *Редактировать данные* и отредактировать записи. Вы можете вставить или удалить записи в таблице, используя кнопки *Вставить запись* или *Удалить запись*. Если вы выбрали несколько записей, все выбранные записи будут удалены.

Изменения результирующей таблицы ничего не изменяют в самой базе данных. Чтобы перенести обновлённые, вставленные или удалённые записи в базу данных нажмите кнопку *Отправить изменения*. Если у вас отключена опция *АвтоФиксация SQL*, то кнопки *Зафиксировать* и *Откатить* на панели будут включены, если начата транзакция. За дополнительной информацией о транзакциях обращайтесь к главе 14.

Редактирование столбцов массивных данных

Как было объяснено в предыдущей главе, чтобы просмотреть значение в различных форматах, вы можете щёлкнуть на кнопке в ячейке столбцов с данными типов `Long`, `Long Raw`, `CLOB`, `BLOB` и `BFILE`. Когда результирующая таблица обновляется, вы можете использовать Редактор массивных данных, чтобы изменить значение столбца.

6.7 Просмотр и редактирование столбцов XMLTYPE

Тип SYS.XMLTYPE не поддерживается Oracle Net 9.0 и более ранними версиями, поэтому у вас не будет возможности прямого доступа к XML данным, которые размещены внутри подобного столбца. Если, например, *xml_text* является столбцом SYS.XMLTYPE, тогда следующий запрос не приведёт к ожидаемому результату:

```
select id, xml_text from xml_table
```

Только столбец *id* будет отображен в результирующем наборе. Чтобы просмотреть XML данные, используйте функцию элемента *getclobval()*:

```
select id, t.xml_text.getclobval() from xml_table t
```

Таким образом, данные типа CLOB могут быть отображены в Текстовом редакторе, и будет автоматически применено выделение XML синтаксиса. Чтобы отредактировать XML данные, сделайте результирующий набор обновляемым, как обычно, включив *rowid*:

```
select id, t.xml_text.getclobval(), rowid from xml_table t
```

Теперь вы можете отредактировать данные типа CLOB и отправить изменённые данные в базу данных. Заметьте, что функции *Просмотр данных* и *Редактирование данных* для таблиц и представлений автоматически применяют эти правила, поэтому самый простой способ просмотреть и отредактировать столбцы XMLTYPE – щёлкнуть правой кнопкой мыши на таблице или представлении, и выбрать соответствующий элемент из всплывающего меню.

6.8 Прямой экспорт запроса

Для запросов с большими результирующими наборами (десятки тысяч записей или более) может быть неудобным сначала запрашивать данные в результирующую таблицу, а потом экспортировать их в файл. Это может занять много времени и отобрать большие ресурсы памяти. В этом случае гораздо более эффективным будет напрямую записать результирующий набор в файл экспорта. Для этого, вместо кнопки *Выполнить* на главной панели инструментов, вы можете нажать кнопку *Экспортировать результаты запроса...* на панели инструментов набора данных. При этом появится всплывающее меню, где вы можете выбрать формат экспортируемых данных - CSV (Comma Separated Values – значения, разделённые запятой), TSV (Tab Separated Values – значения, разделённые символом табуляции), HTML или XML. После выбора формата, вы можете указать файл экспорта, после чего будет выполнен запрос. В этой случае, результирующий набор не будет отображён, а только будет записан в файл экспорта.

6.9 Сохранение SQL скриптов

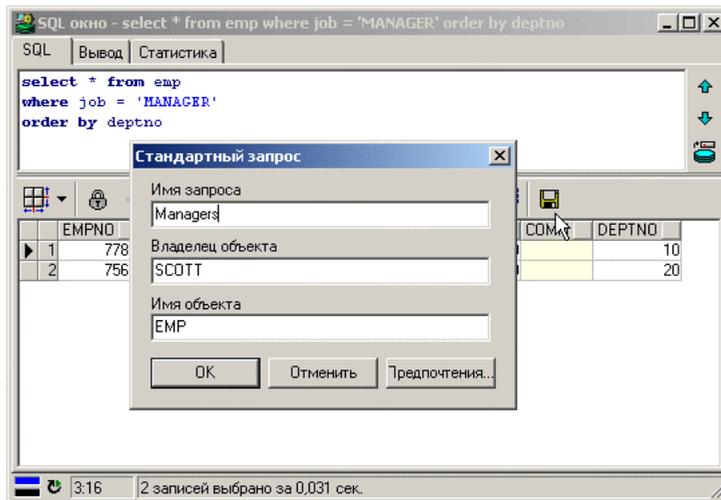
Если вы создали SQL оператор, который впоследствии вы захотите выполнить снова, вы можете сохранить его в SQL скрипте. Для этого, нажмите на панели инструментов кнопку *Сохранить* и введите подходящее имя в диалоге сохранения. По умолчанию для SQL скриптов установлено расширение *.sql*, но вы можете изменить его при помощи предпочтения, описанного в главе 16.27. Заметьте, что только текущий SQL оператор будет сохранён в файл, а не вся история операторов. Чтобы включить историю в сохраняемый файл, щёлкните правой кнопкой мыши в SQL окне и выберите элемент *Сохранить с историей*. В этом случае, все операторы будут разделены слэшем.

Вы можете открыть ранее сохранённый SQL скрипт, нажав на панели инструментов кнопку *Открыть* и выбрав *SQL скрипт*, при этом будет создано новое SQL окно. Также, вы можете щёлкнуть правой кнопкой мыши на ранее созданном SQL окне и выбрать пункт *Загрузить*, который откроет SQL скрипт в существующем SQL окне.

6.10 Создание стандартных запросов

Для большинства таблиц и представлений у вас будут различные стандартные запросы, которые понадобятся вам необходимо часто исполнять. Например: «показать всех служащих, которые являются менеджерами» или «показать суммарную зарплату всех служащих, упорядоченных по номеру отдела». Вы можете сохранить такой запрос, как так называемый «Стандартный запрос», который легко может быть вызван из всплывающего меню для соответствующей таблицы или представления. Это всплывающее меню появится при щелчке правой кнопкой мыши на таблице или представлении в Объектном Навигаторе или в исходном PL/SQL или SQL тексте.

После создания стандартного запроса в SQL окне, вы можете нажать на панели инструментов результирующей таблицы кнопку *Сохранить как стандартный запрос*. Появится следующий диалог:



Вы можете ввести имя запроса, которое будет отображено во всплывающем меню. Вы также можете изменить владельца и имя таблицы или представления, для которого будет отображён стандартный запрос. Чтобы изменить директорию, в которой будут размещены стандартные запросы, нажмите кнопку *Preferences*.

Вместе со стандартным запросом будет сохранена следующая информация:

- SQL текст
- Размер SQL окна
- Режим таблицы: многозаписное отображение или однозаписное отображение

Заметьте, что вы можете использовать подстановочные переменные (смотрите главу 6.5), чтобы сделать стандартный отчёт более гибким.

7. Командное окно

Командное окно позволяет вам выполнять SQL скрипты способом очень схожим с Oracle SQL*Plus. Чтобы создать Командное окно, нажмите на панели инструментов кнопку *Создать* или выберите элемент *Создать* в меню *Файл*. После того как командное окно создано, вы можете вводить SQL и SQL*Plus команды из PL/SQL Developer's IDE:

```

Командное окно - Новое
Диалог | Редактор
Connected to Oracle8i Enterprise Edition Release 8.1.7.0.0
Connected as scott

SQL> select * from emp
  2  where deptno = 20
  3  order by ename;

EMPNO  ENAME      JOB              MGR HIREDATE          SAL          COMM DEPTNO
-----
7876  ADAMS      CLERK            7788 12.01.1983       1100,00          20
7902  FORD       ANALYST          7566 03.12.1981       3000,00          20
7566  JONES      MANAGER          7839 02.04.1981       2975,00          20
7788  SCOTT      ANALYST          7566 09.12.1982       3000,00          20
7369  SMITH      CLERK            7902 17.12.1980        800,00          20

SQL> exec dbms_output.put_line(employee.deptname(7876))

PL/SQL procedure successfully completed

SQL> desc dep
Object dep does not exist.

SQL> desc dept
Name      Type          Nullable Default Comments
-----
DEPTNO   NUMBER(2)
DNAME    VARCHAR2(14) Y
LOC      VARCHAR2(13) Y

SQL> |
  
```

PL/SQL procedure successfully completed in 0,11 seconds

7.1 Ввод SQL операторов и команд

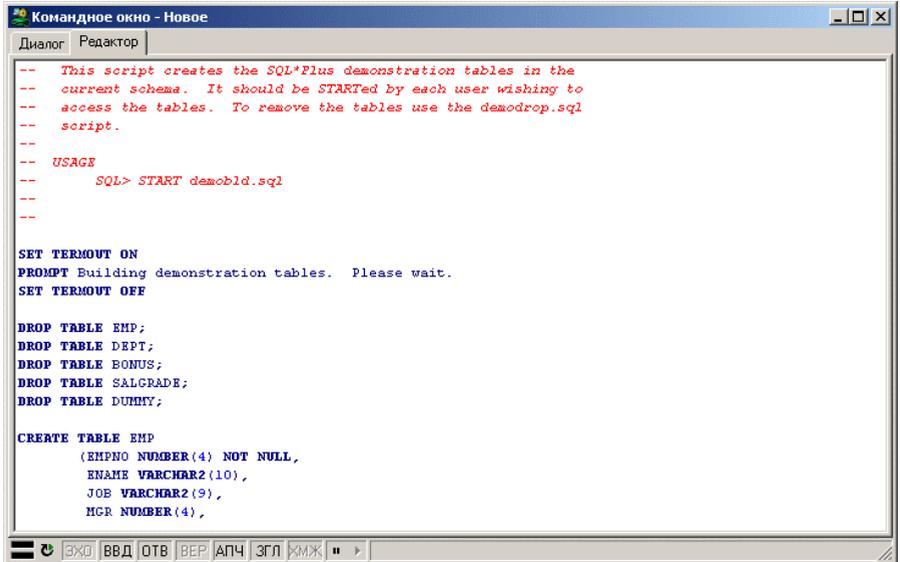
Точно также как и в SQL*Plus, вы можете вводить SQL операторы, разбитые на несколько строк, и заканчивать операторы точкой с запятой или слешем. Чтобы редактировать командную строку, вы можете использовать клавиши управления влево и вправо, и использовать клавиши вверх и вниз, чтобы заново вызвать ранее введенные строки.

При вводе команды *edit*, вы можете использовать обычный текстовый редактор, чтобы редактировать буфер ввода целиком. После редактирования буфера, вы можете выполнить его, введя в командной строке слеш. В редакторе имеется история буфера, со всеми ранее выполненными командами, таким образом, вы можете быстро выполнить измененную команду.

Строка статуса Командного окна показывает статус опций Эхо, Вывод, Ответы, Верификация, Автопечатать, Заголовки и Хронометраж. Также, двойным щелчком мыши можете включить или отключить такую опцию.

7.2 Разработка командных файлов

Чтобы разработать командный файл с несколькими SQL операторами и командами, вам часто нужно отредактировать файл, запустить его, отредактировать его снова, снова запустить его и т. д. Чтобы сделать процесс удобным, в Командном окне имеется встроенный редактор с выделением SQL, PL/SQL и SQL*Plus синтаксиса:



```
-- This script creates the SQL*Plus demonstration tables in the
-- current schema. It should be STARTed by each user wishing to
-- access the tables. To remove the tables use the demodrop.sql
-- script.
--
-- USAGES
--      SQL> START demobld.sql
--
--
SET TERMOUT ON
PROMPT Building demonstration tables. Please wait.
SET TERMOUT OFF

DROP TABLE EMP;
DROP TABLE DEPT;
DROP TABLE BONUS;
DROP TABLE SALGRADE;
DROP TABLE DUMMY;

CREATE TABLE EMP
  (EMPNO NUMBER(4) NOT NULL,
   ENAME VARCHAR2(10),
   JOB VARCHAR2(9),
   MGR NUMBER(4),
```

Открытие, выполнение и сохранение командного файла

Чтобы создать командный файл, переключитесь на страницу *Редактор* и введите команды. Чтобы выполнить команды в редакторе, просто нажмите на панели инструментов кнопку *Выполнить* или нажмите *F8*. Командное окно переключится обратно к странице *Диалог*, и будет выполнять все команды.

Чтобы отредактировать существующий командный файл, нажмите на панели инструментов кнопку *Открыть* и выберите пункт *Командный файл*. При этом командный файл будет загружен в редактор нового Командного окна. Вы также можете щёлкнуть правой кнопкой мыши в редакторе и выбрать пункт *Загрузить* во всплывающем меню. Чтобы сохранить изменённый Командный файл, нажмите на панели инструментов кнопку *Сохранить*.

Пошаговое исполнение командного файла

Если вы нажмёте в строке статуса кнопку *Приостановить выполнение* перед или в течение выполнения, вы сможете шаг за шагом пройти скрипт, управляя этим процессом. Следующая команда будет выделена в Редакторе и будет выполнена, когда вы нажмёте в строке статуса кнопку *Выполнить следующую команду*. Если вы снова нажмёте кнопку *Приостановить выполнение*, после следующего шага выполнение будет продолжено в нормальном режиме. Вы можете принудительно перейти в режим паузы из скрипта, используя команду SET EXECPAUSE.

7.3 Поддерживаемые команды

Помимо всех SQL операторов, командное окно поддерживает следующие стандартные SQL*Plus команды:

Команда	Значение
/	Выполняет SQL буфер
? [Keyword]	Предоставляет SQL помощь по ключевому слову
@[@] [Filename] [Parameter list]	Запускает определённый командный файл с учетом указанных параметров
ACC[EPT] Variable [DEF[AULT] Value] [PROMPT Text NOPR[OMPT]]	Позволяет пользователю вводить значение подстановочной переменной
CL[EAR] [SCR[EEN]]	Очищает экран
CL[EAR] SQL	Очищает SQL буфер
COL[UMN] [Column] [Format] [NEW_VALUE Variable]	Определяет формат столбца, отображает формат столбца или отображает форматы всех столбцов
CON[NECT] [username/password@database]	Устанавливает соединение с базой данных как указанный пользователь
DEF[INE] [Variable] [= Text]	Задаёт подстановочную переменную, отображает переменную или отображает все подстановочные переменные
DESC[RIBE] Object	Даёт описание указанного объекта
DISC[CONNECT]	Отображает текстовый редактор для редактирования SQL буфера
EDIT	Отображает текстовый редактор для редактирования SQL буфера
EXEC[UTE] Procedure	Выполняет указанную процедуру
EXIT [APPLICATION]	Прекращает работу скрипта или закрывает Командное окно. Добавление параметра APPLICATION также приведёт к закрытию PL/SQL Developer.
GET [Filename]	Загружает в редактор командный файл
HOST [Command]	Выполняет команду операционной системы
HELP [Keyword]	Предоставляет SQL помощь по ключевому слову
PAUSE [Message]	Отображает сообщение и переходит в режим паузы, пока пользователь не нажмёт Okay или Cancel
PRI[NT] [Variable]	Отображает значение связанной переменной или всех связанных переменных
PROMPT [Text]	Отображает указанный текст
QUIT [APPLICATION]	Прекращает работу скрипта или закрывает Командное окно. Добавление параметра APPLICATION также приведёт к закрытию PL/SQL Developer.
R[UN]	Выполняет SQL буфер
REM[ARK] [Text]	Строка с комментарием
SET AUTOP[RINT] [ON OFF]	Определяет, отображать автоматически связанные переменные после выполнения SQL оператора или PL/SQL блока.
SET COLSEP [Separator OFF]	Определяет разделитель колонок (по умолчанию = " ").
SET CON[CAT] [Character ON OFF]	Определяет символ, которым завершается ссылка на подстановочную переменную (по умолчанию = .)
SET DEF[INE] [Character ON OFF]	Определяет символ, которым начинается ссылка на подстановочную переменную (по умолчанию =)
SET ECHO [ON OFF]	Определяет, отображена ли в скрипте выполненная команда
SET ESC[APE] [Character ON OFF]	Определяет символ, который экранирует символ, которым начинается ссылка на подстановочную переменную (по умолчанию = \)
SET FEED[BACK] [ON OFF]	Определяет, отображается ли количество затронутых строк SQL оператора

Команда	Значение
SET HEA[DING] [ON OFF]	Определяет, отображаются ли заголовки над столбцами результирующего набора
SET LONG [Width]	Определяет максимальную отображаемую ширину длинного столбца
SET NUM[WIDTH] [Width]	Определяет максимальную отображаемую ширину числового столбца без учета точности
SET PAGES[IZE] [Size]	Определяет количество строк, отображаемых в результирующем наборе, до того как заголовки будут повторены
SET PROMPT [Prompt]	Изменяет стандартную SQL> подсказку. Вы можете использовать не только текст, но и переменные [user], [db], or [connection]. Более того, вы можете включить в команду связанные переменные (set prompt :bind var name).
SET SCAN [ON OFF]	Определяет, следует ли сканировать подстановочные переменные.
SET SERVEROUT[PUT] [ON OFF] [SIZE n]	Определяет, отображается ли вывод вызовов dbms_output.put_line, а также определяет размер буфера вывода
SET SPOOL*DIRECTORY [Directory]	Определяет, в каком каталоге файлы вывода будут сохранены, если SPOOL команда не содержит абсолютного пути.
SET TERM[OUT] [ON OFF]	Определяет, отображается ли вывод выполненных SQL операторов
SET TIMI[NG] [ON OFF]	Определяет, отображается ли хронометрическая информация о выполненных SQL операторах
SET VER[IFY] [ON OFF]	Определяет, отображаются ли подстановочные переменные при их использовании в SQL операторе или PL/SQL блоке.
SHO[W] ERR[ORS] [Type Name]	Отображает ошибки предыдущей компиляции, или для указанного объекта
SHO[W] REL[EASE]	Отображает информацию о версии Oracle для текущего соединения
SHO[W] SQLCODE	Отображает код завершения выполненного SQL оператора
SHO[W] USER	Отображает имя пользователя текущего соединения
SPO[OL] [Filename] [OFF]	Начинает или останавливает запись в буферный файл
STA[RT] [Filename] [Parameter list]	Запускает указанный командный файл, передавая ему указанные параметры
STORE SET [Filename]	Размещает значения всех опций в указанном файле. Вы можете выполнить этот файл позже, чтобы восстановить значения этих опций.
UNDEF[INE] Variable	Отменяет определение указанной подстановочной переменной
VAR[IABLE] [Variable] [Datatype]	Определяет связанную переменную, отображает связанную переменную или отображает все связанные переменные.
WHENEVER [OSERROR SQLERROR] [Action]	Указывает действие при появлении OS или SQL ошибки. Действие может быть либо EXIT, либо CONTINUE. Дополнительно можно указать COMMIT or ROLLBACK.

Все эти команды функционируют так же, как и в SQL*Plus. Следующие команды являются специфическими командами PL/SQL Developer:

Команда	Значение
BEAUT[IFY] File Object	Форматирует указанный файл или объект базы данных, используя текущие правила или правила, указанные

	при помощи команды SET BEAUTIFERRULES.
BROWSE Object	Выбирает объект в Объектном Навигаторе
EDIT Object	Открывает редактируемое окно с определением объекта
EDITD[ATA] Table View	Открывает Окно SQL для таблицы или представления с редактируемым результирующим набором
EXPORT[DATA] Table	Открывает Инструмент экспорта для указанной таблицы
INFO	Выводит информацию о соединении
PROP[ERTIES] Object	Окно свойств для указанного объекта
QUERY[DATA] Table View	Открывает Окно SQL для результирующего набора таблицы или представления, доступного только для чтения.
REC[OMPILE] Object	Перекомпилирует объект
SET BEAUT[IFIERRULES] [File]	Временно использует правила форматирования из указанного файла. Эта команда может использоваться совместно с командой BEAUTIFY.
SET COL[WIDTH] [Width]	Определяет максимальную ширину столбца в результирующем наборе. Если Width = 0, то ширина столбца не ограничена. По умолчанию, ширина столбца – 80.
SET EXEC[PAUSE] [ON OFF]	Приостанавливает выполнение перед следующей командой (ON), или продолжает выполнение со следующей командой (OFF).
SQLPLUS	Вызывает SQL*Plus с текущим файлом.
TEST ProgramUnit	Открывает для указанного программного модуля Тестовое окно со стандартным Тестовым скриптом.
VIEW Object	Открывает доступное только для чтения окно с определением объекта.

8. Создание и изменение не PL/SQL объектов

В течение разработки PL/SQL, вы можете обнаружить, что вам часто нужно создать таблицу, изменить ограничение или индекс, сбросить последовательность и т. д. В PL/SQL Developer имеются несколько функций, которые позволят вам изменять таблицы (и связанные элементы), последовательности, синонимы, библиотеки, директории, пользователей и роли. В этой главе не объясняется функционирование этих объектов, а только объясняется, как вы можете создавать, изменять эти объекты и просматривать их свойства. Для большей информации о функционировании каждого из этих объектов, смотрите документацию по Oracle, такую как “SQL Reference Guide”.

Чтобы создать объект, вы можете либо нажать на панели инструментов кнопку *Создать* и выбрать соответствующий тип объекта или выбрать пункт *Создать* в меню *Файл*. Вы также можете щёлкнуть правой кнопкой мыши на корневой папке объектов определенного типа в Объектном Навигаторе и выбрать во всплывающем меню элемент *Создать*.

Чтобы изменить объект, выберите его в Объектном Навигаторе, щёлкните по нему правой кнопкой мыши и выберите во всплывающем меню элемент *Редактировать*. Вы не можете подобным образом изменить имя объекта, для этого вы должны использовать функцию Объектного Навигатора *Переименовать* (если данный тип объекта поддерживает её). Если вы просто хотите просмотреть объект, выберите во всплывающем меню элемент *Показать*.

В дополнение к этому, вы можете создать копию объекта, выбрав во всплывающем меню Объектного Навигатора элемент *Дублировать*. Появится окно редактора объекта с уже всеми заданными свойствами, за исключением владельца (если применимо) и имени. Подобным образом, вы можете быстро создать дубликат объекта, чтобы исследовать альтернативу, провести рискованный или деструктивный тест или ещё что-либо.

8.1 Редактор определения таблицы

В редакторе определения таблицы имеется 7 страниц для различных аспектов таблицы:

Redaction of table EMP

Общее | Столбцы | Ключи | Проверки | Индексы | Привилегии

Владелец: SCOTT
Имя: EMP

Пересоздать таблицу
Предупреждение: это удалит все данные, триггеры и ссылки внешних ключей

Хранение: SYSTEM

ичное пространство: %свободно: 10, %занято: 40, Нач. транз.: 1, Макс. транз.: 255

1ый экстенд: 64 KB
Следующий экстенд: 64 KB
%увеличение: 50
Мин. экстендов: 1
Макс. экстендов: Нет лимита

Классер: Имя: _____ Колонки: _____

Время жизни: Временная COMMIT не удаляет записи

Организация: Куча Индекс

Комментарии: _____

Применить | Обновить | Закрыть | Помощь | Запрос... | Показать SQL

Внизу редактора вы видите следующие 6 кнопок:

- **Применить** - Применить все изменения, которые вы сделали в редакторе, в базу данных.
- **Обновить** - Перечитывает определение из базы данных, отменяя любые сделанные вами изменения.
- **Закрыть** - Закрывает окно редактора.
- **Помощь** - Показывает онлайн-помощь
- **Запрос** - Вызывает Окно SQL с запросом, которое позволяет вам просматривать и редактировать табличные данные.
- **Показать SQL** - Выводит на экран текстовый редактор с SQL операторами, получившимися в результате изменений, сделанных в редакторе. Если вы не сделали никаких изменений, оно покажет SQL команды создающие объект.

Следующие главы расскажут о различных страницах редактора определения таблицы.

Страница “Общее”

Страница “Общее” показана в предыдущей главе и содержит информацию о владельце таблицы и имени, информацию о размещении, информация о кластерах и комментарии. Для новой таблицы вы можете оставить все свойства пустыми, кроме имени. Значения остальных свойств будут установлены по умолчанию:

- Owner – текущий зарегистрированный пользователь.
- Табличное пространство – табличное пространство установленное по умолчанию для текущего пользователя.
- %Свободно – 10
- %Занято – 40
- Нач. транз. - 1
- Макс. транз. - 255

Значения по умолчанию свойств сегмента (1ый экстенд, следующий экстенд, %увеличение, мин. и макс. экстендов) зависят от соответствующих установленных по умолчанию свойств табличного пространства.

Информация о кластере, информация о хранении и время жизни являются взаимноисключающими, потому что кластер неявно определяет характеристики размещения таблицы, и временные таблицы не могут быть кластеризованы и также имеют неявные характеристики размещения.

Если вы вводите или выбираете имя кластера, то свойства хранения и время жизни станут доступными только для чтения, и можно будет вводить столбцы кластера. Очистка имени кластера даст обратный эффект.

Если вы определили таблицу как временную, то информация о кластере и хранении станет доступной только для чтения, и вы можете определить, будут ли строки существовать после фиксации транзакции.

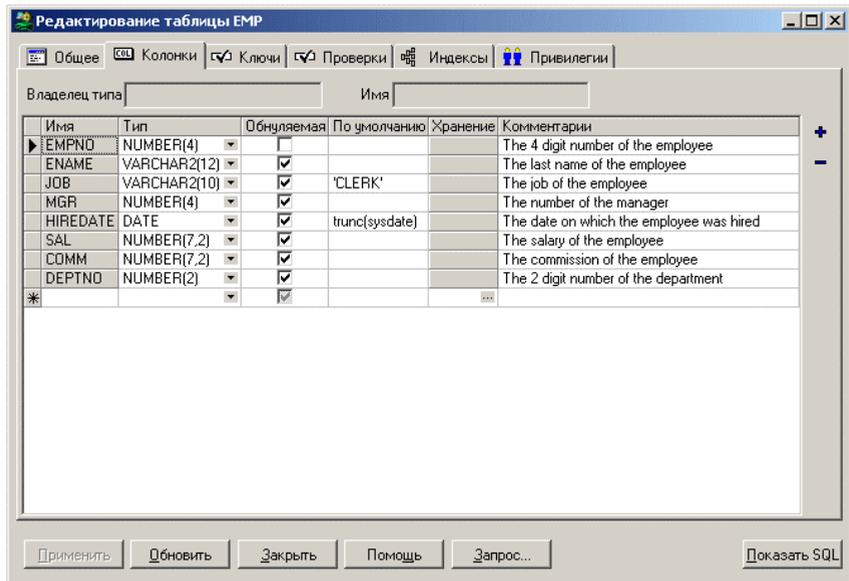
Если вы определяете таблицу как организованную по индексу, вы должны определить первичный ключ. Для таблицы организованной в виде кучи (по умолчанию) это не требуется.

Если вы изменяете существующую таблицу, не все свойства могут быть изменены. Табличное пространство, 1ый экстенд, мин. экстендов, информация о кластере, время жизни и организационные свойства не могут быть изменены для таблицы, которая уже существует в базе

данных. Если вы хотите изменить какие-либо из этих свойств, вам нужно выбрать опцию *Пересоздать таблицу* в верхней части окна. В результате, таблица будет уничтожена и создана заново с новым определением. Все данные, триггеры и ссылки внешних ключей *будут утеряны*, поэтому вам следует быть осторожными при использовании этой опции!

Страница «Колонки»

На странице *Колонки* вы можете просматривать, добавлять, удалять, перемещать и изменять колонки таблицы. Если, вы создаёте объектную таблицу, то на этой странице вы также можете выбрать объектный тип:



Для того чтобы добавить столбец, вы можете нажать кнопку *Вставить колонку*, или вы можете начать набирать информацию нового столбца в последней строке (при этом будет создана новая пустая последняя строка). Столбец *Тип* содержит список самых употребимых типов данных, но вы также можете здесь использовать и другие типы данных. Столбец *По умолчанию* требует, чтобы вы вводили величину в соответствующем формате, например, значения типа string должны быть заключены в кавычки.

Чтобы удалить столбец, нажмите кнопку *Удалить колонку*. При этом будет удалён столбец, с которым вы сейчас работаете, либо все выделенные столбцы. Вы можете выбрать один или несколько столбцов, щёлкнув на заголовке строки. При перетаскивании курсора будет выбран диапазон столбцов. Нажатие клавиши *Control* во время щелчка на заголовке строки позволит вам выбрать несколько отдельных строк.

Чтобы переместить одну или несколько колонок, выберите их, как это описано в предыдущем абзаце. Теперь вы можете щёлкнуть на одном из выбранных заголовков и переместить выбранные колонки в нужное место.

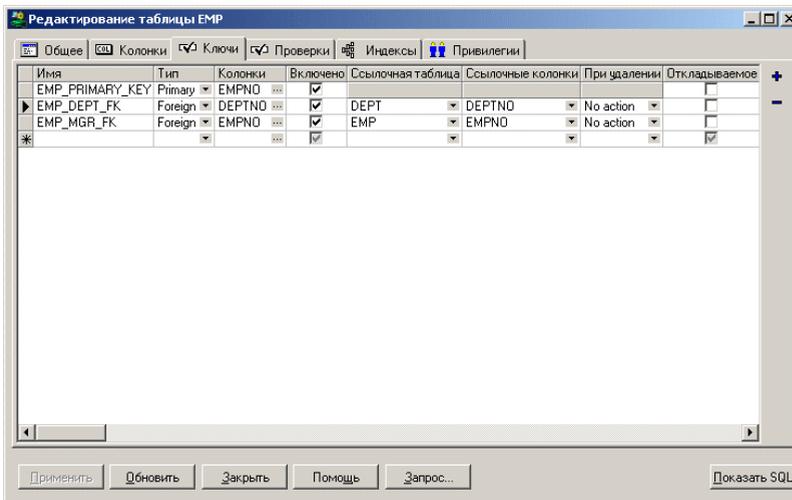
Для существующих таблиц вы не можете изменять имя существующих колонок, если только вы не используете Oracle9.2 или выше. Вы не можете удалять колонки, если только вы не используете Oracle8i и выше. Чтобы преодолеть эти ограничения, вам нужно пометить опцию *Пересоздать таблицу* на странице *Общие*. Назначение этой опции описано в предыдущей главе.

Заметьте, что изменение имени колонки автоматически распространится на колонки входящие в ограничения, индексные колонки и кластерные колонки.

Если вы выбираете владельца типа и имя типа в верхней части страницы, вы создадите объектную таблицу, основанную на этом объектном типе. Имена и типы столбцов теперь определяются и ограничиваются атрибутами верхнего уровня объектного типа. Вы всё ещё можете определить, является ли столбец обнуляемым, значение по умолчанию, его хранение и комментарий. Если вы хотите позже изменить объектный тип, вы должны выбрать опцию *Пересоздать таблицу* на странице *Общее*.

Страница «Ключи»

На странице *Ключи* вы можете просматривать, добавлять, удалять и изменять первичный, уникальные и внешние ключи таблицы:



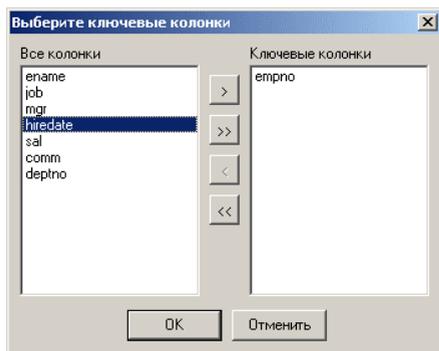
С этой таблицей можно работать таким же образом, как и с таблицей колонок в предыдущей главе. Вы можете добавлять, удалять и изменять ограничения ключей. Перемещение ограничений не даст никакого эффекта, так как они на самом деле не упорядочены.

Заметьте, что информация о хранении индекса, который используется, для того чтобы обеспечить ограничение первичного или уникального ключа, не определена на этой странице. Если ограничение включено, то на странице *Индексы* автоматически будет присутствовать соответствующий индекс с таким же именем. Для этого индекса вы можете определить информацию хранения. Отключение или удаление ограничения первичного или уникального ключа неявно удалит индекс.

В столбце *Type* есть список, который позволит вам выбрать только Первичный, Уникальный или Внешний типы ключа. Вы можете быстро выбрать правильный тип, введя первую букву (P, U или F). Заметьте, что таблица может содержать только одно ограничение первичного ключа, и значение по умолчанию для этого столбца (Первичный или Уникальный) зависит от этого. Изменение типа влияет на столбцы *Referencing table*, *Referencing columns* и *Cascade*, т.к. они действительны только для ограничений внешнего ключа. Изменение типа на Внешний также приведёт к неявному удалению индекса.

Чтобы определить, к каким колонкам применяется ограничение, вы просто можете набрать имена колонок в соответствующей ячейке (разделённые запятыми) или нажать кнопку в ячейке.

Это вызовет диалог выбора колонок, где вы легко сможете добавить, удалить или переместить (переупорядочить) ключевые колонки:

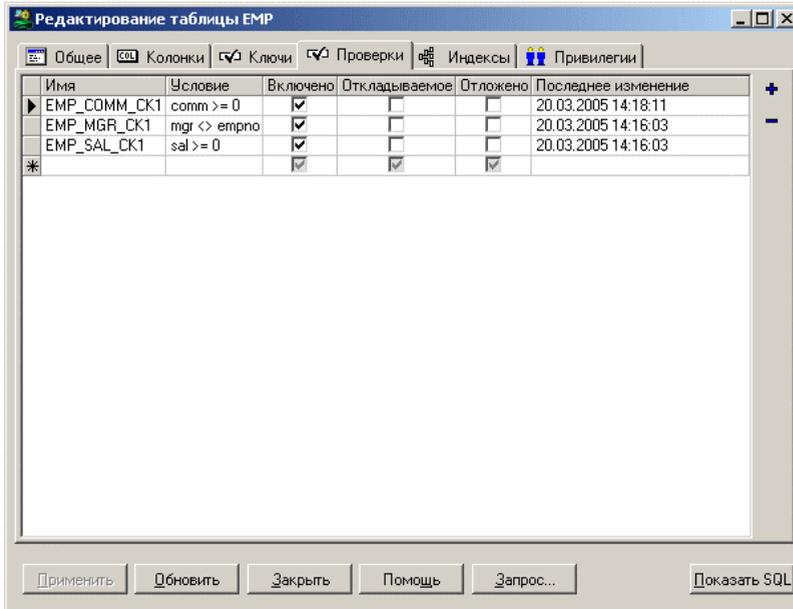


Для ограничений внешних ключей вы можете набрать или выбрать *ссылочную таблицу*. При этом действии будут автоматически выбраны используемые по умолчанию *ссылочные колонки*, основываясь на ограничениях первичного или уникального ключей этой таблицы. В списке выбора для этого столбца отображены все наборы столбцов, которые образуют ограничения первичного или уникального ключа для ссылочной таблицы, и, следовательно, являются единственными кандидатами. Используя список *On Delete*, вы можете определить, какое действие для записей в дочерней таблице должно быть выполнено, когда родительская запись удалена. Действие *Set null* поддерживается в Oracle8i и в более поздних версиях.

Если вы используете Oracle8, вы дополнительно можете определить, является ли ограничение откладываемым, и отложено ли оно первоначально. Эти 2 опции явно связаны между собой: у вас не может быть неоткладываемого ограничения, которое первоначально отложено.

Страница Проверки

На странице *Проверки* вы можете просматривать, добавлять, удалять и изменять проверочные ограничения таблицы:



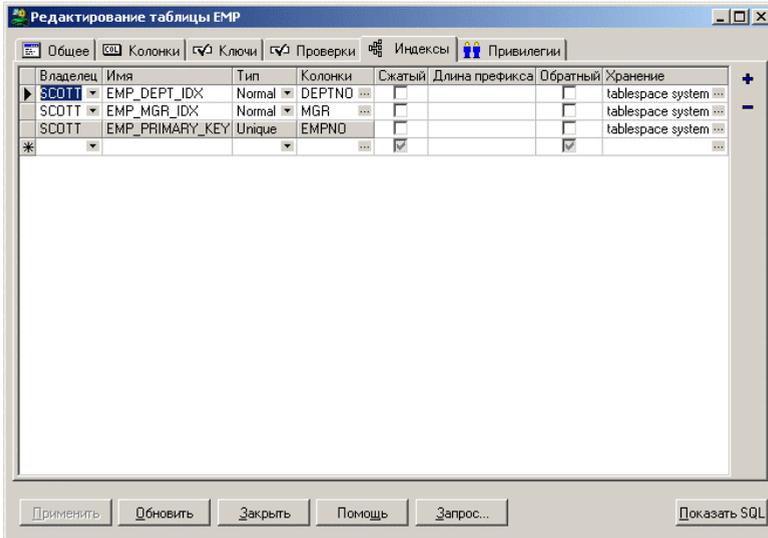
Этой таблицей можно управлять таким же образом, как и любой другой таблицей. Вы можете добавлять, удалять и изменять проверочные ограничения.

Столбец *Условие* должен содержать выражение булевого типа, которое должно принимать значение True (или Null) для всех записей в таблице. Вы не можете применить проверочное ограничение, если одна или несколько записей нарушают условие, до тех пор, пока вы не отключите опцию *Включено*. Проверочное ограничение может быть включено после исправления записей нарушающих условие.

Если вы используете Oracle8 или более позднюю версию, вы дополнительно можете определить, является ли ограничение откладываемым, и отложено ли оно первоначально. Эти 2 опции явно связаны между собой: у вас не может быть неоткладываемого ограничения, которое первоначально отложено.

Страница Индексы

На странице *Индексы* вы можете просматривать, добавлять, удалять и изменять индексы таблицы:



Этой таблицей можно управлять таким же образом, как и любой другой таблицей. Вы можете добавлять, удалять и изменять индексы.

Колонка *Владелец* индекса может быть оставлена пустой, по умолчанию, владельцем будет считаться зарегистрированный на данный момент пользователь. Если вы хотите указать другое имя владельца индекса, просто выберите его из списка.

Тип индекса может быть Normal, Unique или Bitmap. Последний тип индекса доступен только для Oracle7.3 Server и более поздних версий. Заметьте, что индекс типа Unique вы можете создавать, если только каждая строка является на самом деле уникальной. Пока это не так, вы не можете создать этот индекс в базе данных.

Что бы определить *Колонки*, по которым вы хотите проиндексировать таблицу, вы можете просто ввести их имена в нужной ячейке (разделяя запятыми) или нажать кнопку в ячейке. Будет вызван экран выбора колонок, где вы можете добавить, удалить или упорядочить индексные колонки.

Для индексов основанных на функциях, вы можете через запятую указать функции. Для нисходящих индексов, вы можете после имени столбца указать *DESC*.

Вы можете дополнительно включить сжатие ключа для индекса, включая число префиксных столбцов для сжатия, и определить индекс как обратный индекс.

Если это индекс для первичного или уникального ключа, то свойства owner, name, type, и columns не могут быть изменены. Все эти свойства определяются ограничением ключа. При изменении имени ограничения ключа автоматически будет изменено имя соответствующего индекса, при изменении колонок ограничения будут изменены соответствующие колонки для индекса. Имя владельца будет всегда таким же, как и имя владельца таблицы, а индекс будет всегда уникальным.

Параметры *размещения* можно определить или просмотреть, нажав кнопку в ячейке. При этом будет вызван редактор размещения:

Для новых индексов вы не обязаны вводить что-либо из этой информации, так как каждое свойство имеет значение по умолчанию:

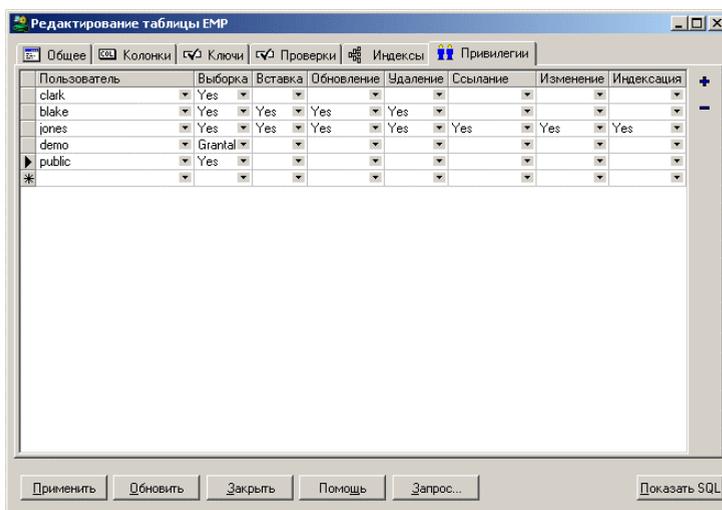
- Табличное пространство – табличное пространство по умолчанию для каждого пользователя.
- %Свободно - 10
- Первоначальное количество транзакционных входов – 2
- Максимальное количество транзакционных входов – 255

Значения по умолчанию свойств сегмента (1ый экстенст, следующий экстенст, %увеличение, мин. экстенстов и макс. экстенстов) зависят от соответствующих значений по умолчанию для табличного пространства. Свойство %занято не применимо к индексам.

Заметьте, что если вы изменяете свойства %свободно, первый экстенст и следующий экстенст существующего индекса, это приведёт к тому, что индекс (и ограничение) будут уничтожен и создан заново, чтобы применить это изменение. Для больших индексов, это может занять время.

Страница Привилегии

На странице *Привилегии* вы можете предоставить/отобразить привилегии на таблицу пользователям и ролям:



Этой таблицей можно управлять таким же образом, как и любой другой таблицей. Вы можете добавлять, удалять и изменять пользователей или роли, имеющие привилегии. Удаление таких пользователей или роли эквивалентно отмене всех индивидуальных привилегий.

Колонка *Пользователь* – это пользователь или роль, которым вы хотите предоставить определённые привилегии. Заметьте, что привилегии, предоставленные роли, часто не будут достаточными для целей разработки хранимых программных модулей. Если пользователь имеет через роль привилегию выбора в таблице и ссылается на эту таблицу, в хранимом программном модуле, компиляция будет завершена по ошибке, потому что данная таблица неизвестна. Пользователю должна быть персонально предоставлена привилегия выбора, если в программном модуле будет использоваться объект, которым он или она владеет.

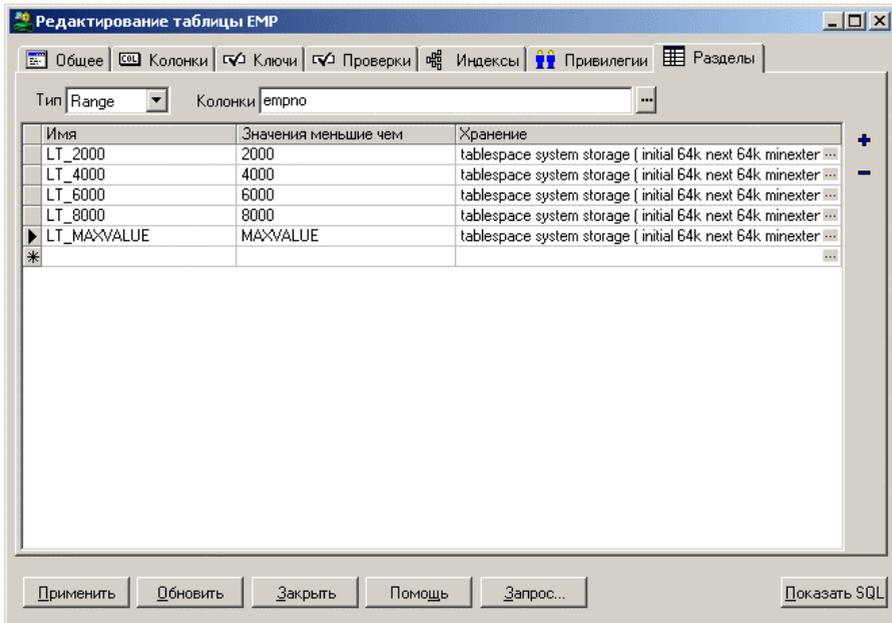
Столбцы индивидуальных привилегий могут иметь 3 значения:

- (Пусто) – Пользователь или роль не имеют привилегии
- Yes - Пользователь или роль имеют привилегию
- Grantable - Пользователь или роль имеют привилегию и могут предоставить её другим пользователям или ролям

Привилегии *Выборка*, *Вставка*, *Обновление* и *Удаление* позволяют имеющему привилегию пользователю совершать эти действия над записями таблицы. Привилегия *Ссылание* позволяет имеющему привилегию пользователю создавать внешний ключ к этой таблице. Привилегия *Изменение* позволяет имеющему привилегию пользователю изменять определение таблицы при помощи команды `Alter table`. Привилегия *Индексация* позволяет имеющему привилегию пользователю создавать индекс для таблицы.

Страница Разделы

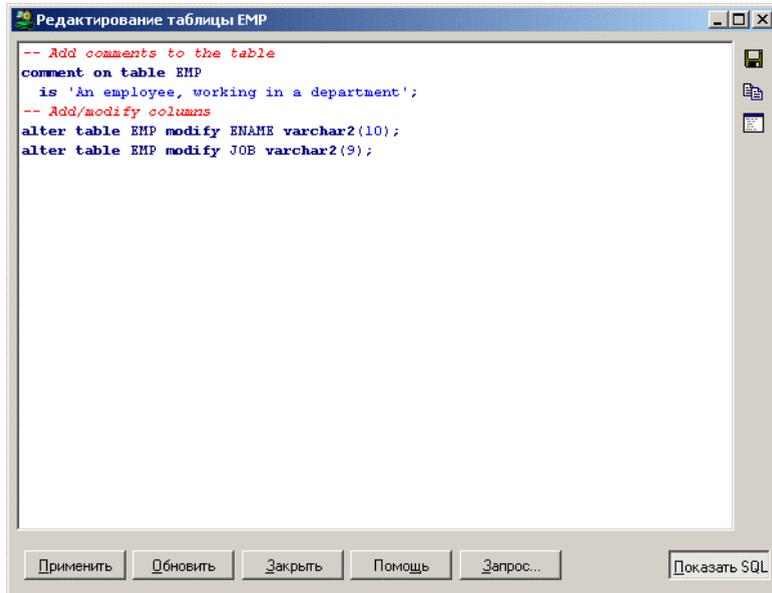
Страница *Разделы* доступна для просмотра, только когда вы создаёте (пересоздаёте) таблицу или редактируете существующую разбитую на разделы таблицу:



В верхней части страницы вы можете выбрать тип разделения *Range*, *Hash* или *List*, а также делящие колонки. В таблице вы можете определить имя, значения колонок (если применимо) и параметры размещения для каждого раздела. Если вы определили несколько делящих столбцов, то вы должны разделить значения запятой.

Просмотр и изменение SQL

После создания таблицы или после изменения существующей таблицы, вы можете просмотреть SQL, нажав кнопку *Показать SQL*. После добавления комментария в таблицу и преобразования столбцов *ename* и *job* в тип данных *varchar2*, результирующий SQL будет выглядеть следующим образом:



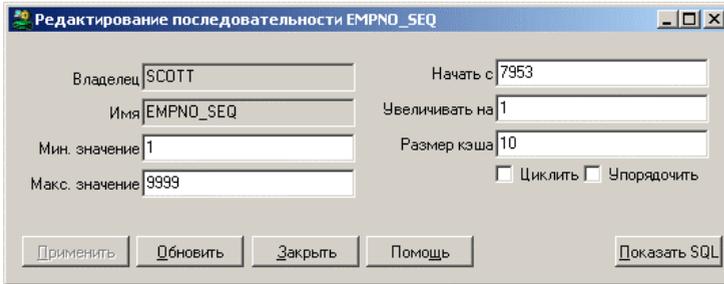
Вы можете сохранить SQL в файл или скопировать его в буфер обмена при помощи соответствующих кнопок в верхнем правом углу окна. Для печати SQL нажмите кнопку *Печать* на панели инструментов. Вы можете дополнительно изменить SQL и применить изменённый SQL. После применения изменённого SQL окно вернётся в режим формы и обновит табличное определение. Изменения, сделанные в SQL, будут немедленно отражены в формах.

Если вы снова нажмёте кнопку *Показать SQL*, то окно вернётся в режим формы, отменив все не применённые изменения, которые вы внесли в SQL.

Если, для того, чтобы просмотреть табличное определение, вы выбрали в Объектном Навигаторе опцию *Показать*, то кнопка *Показать SQL* отобразит CREATE SQL команду для таблицы, без предшествующей этому команды Drop. Таким образом, вы можете сохранять табличное определение в файл или копировать её определение в другую базу данных или другому пользователю.

8.2 Редактор определения последовательности

Редактор определения последовательности позволит вам легко создать или изменить последовательность:



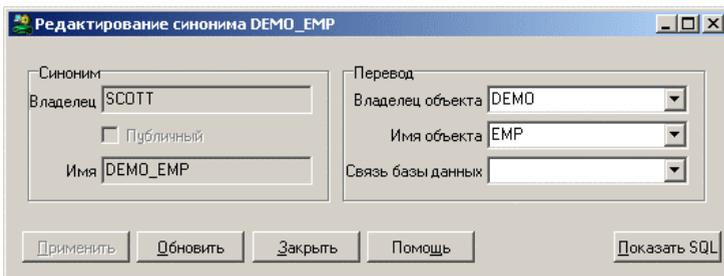
Most properties have a default value, and you only need to supply a name when creating a new sequence:

Большинство свойств имеют значение по умолчанию, вам необходимо только указать имя, когда вы создаёте новую последовательность:

- Владелец – Текущий зарегистрированный пользователь
- Минимальное значение – 1
- Максимальное значение – Неограниченно
- Начальное значение – 1
- Шаг – 1
- Размер кэша – 20
- Циклическость - Выключено
- Упорядоченность – Отключено

8.3 Редактор определения синонимов

Редактор синонимов позволит вам легко создать или изменить синоним:

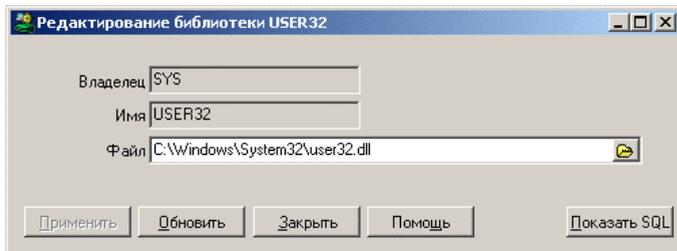


Значением *Владелец* по умолчанию будет имя текущего зарегистрированного пользователя. При включении опции *Публичный* будет создан общедоступный синоним, а поле *Владелец* станет доступным только для чтения. В полях *Владелец объекта* и *Имя объекта* есть списки, для того,

чтобы вы легко могли выбрать объект для синонима. Если поле *Владелец объекта* остаётся пустым, список *Имя объекта* будет отображать все объекты.

8.4 Редактор определения библиотеки

Редактор библиотеки позволит вам легко создать или изменить библиотеку:

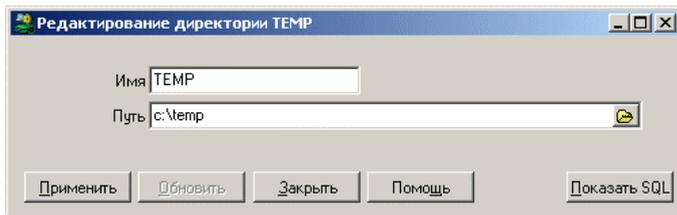


Значением *Владелец* по умолчанию будет имя текущего зарегистрированного пользователя, и вы можете использовать список, для того, чтобы создать библиотеку с другим именем владельца. *Файл* является полной файловой спецификацией библиотеки 3GL, которую вы хотите использовать. Если вы занимаетесь разработкой на сервере базы данных, вы можете использовать кнопку *Выбор файла*, чтобы выбрать динамически связываемую библиотеку. Иначе вы должны ввести имя DLL (Windows) или разделяемый объект (UNIX). Правильность *Файл* не проверяется.

8.5 Редактор определения директории

The directory editor allows you to easily create or modify a directory, which can subsequently be used for BFILE's:

Редактор директории позволит вам легко создать или изменить директорию, которая далее может быть использована для BFILE:



Если вы занимаетесь разработкой на сервере базы данных, вы можете использовать кнопку *Выбор каталога*, чтобы выбрать путь каталога. Правильность пути каталога не проверяется.

8.6 Редактор определения задания

Редактор задания позволит вам создать или изменить задание:

Создание нового задания эквивалентно вызову `dbms_job.submit`. Изменение существующего задания эквивалентно вызову `dbms_job.change`. Для большей информации, смотрите главу *DBMS_JOB* в руководстве пользователя *Supplied Packages Reference*.

Кнопка *Открыть PL/SQL редактор* рядом с полем *Что* вызывает более удобный Редактор PL/SQL с выделением синтаксиса, Ассистентом кодирования и т. д.

8.7 Редактор определения очереди

Редактор очереди позволит вам создать или изменить продвинутую очередь:

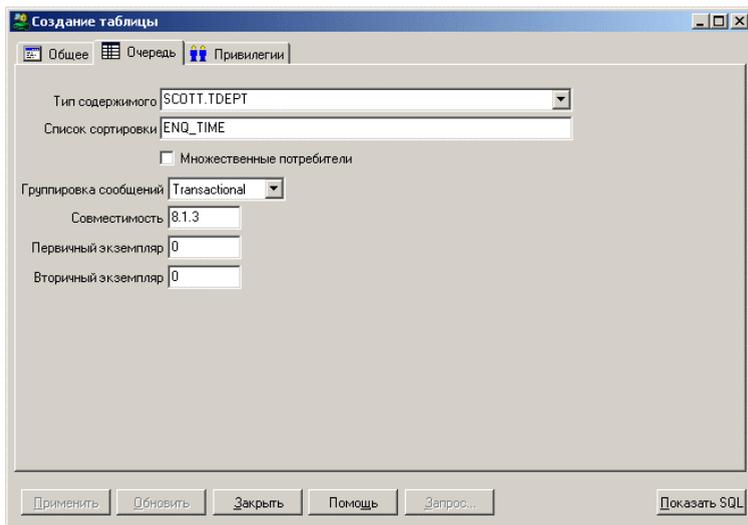
В качестве имени таблицы очереди вы можете выбрать существующую таблицу очереди или нажать кнопку *Создать*, чтобы создать новую таблицу очереди (смотрите главу 8.8).

Для большей информации об Организации продвинутой очереди Oracle, смотрите следующую документацию Oracle:

- *Application Developer's Guide - Advanced Queuing*
- *Supplied PL/SQL Packages and Types Reference*

8.8 Редактор определения таблицы очереди

Редактор таблицы очереди позволит вам создать или изменить таблицу продвинутой очереди:



На странице *Общее* вы можете определить имя, параметры размещения и комментарий к таблице. На странице *Привилегии* вы можете предоставить привилегии на таблицу пользователям или ролям. Смотрите также главу 8.1.

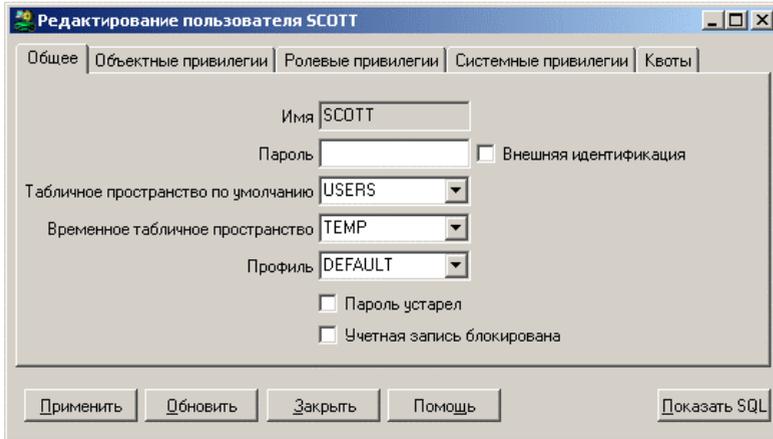
На странице *Очередь* вы можете ввести специфическую информацию таблицы очереди. Для *Тип содержимого* вы можете выбрать существующий объектный тип или *RAW*, чтобы создать не типизированную очередь.

Для большей информации об организации продвинутой очереди Oracle, смотрите следующую документацию Oracle:

- *Application Developer's Guide - Advanced Queuing*
- *Supplied PL/SQL Packages and Types Reference*

8.9 Редактор определения пользователя

Редактор пользователя позволит вам создать или изменить пользователя, его привилегии и квоты табличного пространства:



Заметьте, что для того, чтобы просмотреть или изменить информацию о пользователе, вам необходимы специфические системные привилегии (такие как предоставляемые ролью *DBA*).

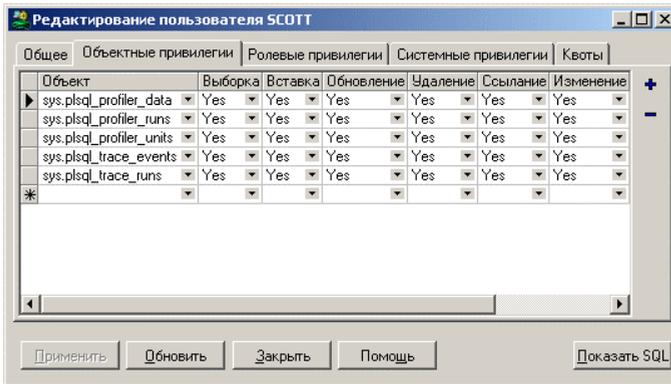
Страница Общее

На главной странице вам, в основном, нужно ввести имя и пароль. Все остальные свойства имеют значение по умолчанию:

- Табличное пространство по умолчанию – SYSTEM
- Временное табличное пространство – SYSTEM
- Профиль – DEFAULT
- Истечение срока действия пароля – Выключено
- Учетная запись заблокирована – Выключено

Если вы изменяете существующего пользователя, текущий пароль не будет отображен. Чтобы изменить его, введите новый пароль или оставьте его пустым, чтобы оставить пароль без изменений. Если пользователь идентифицируется внешне, вы не можете ввести пароль.

Страница Объектные привилегии

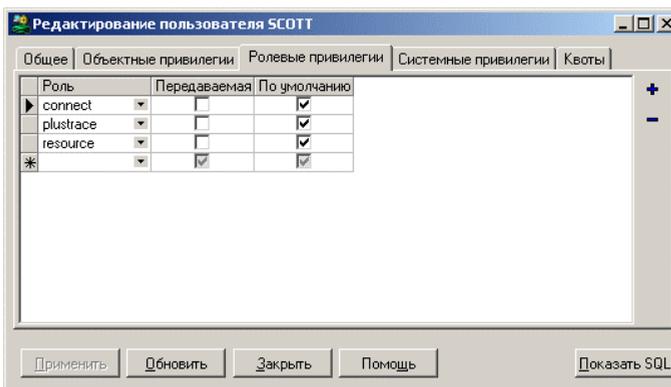


Эта страница позволит вам просматривать и изменять объектные привилегии, предоставленные пользователю. В них не входят привилегии, косвенно предоставленные через роль.

В первом столбце вы можете видеть объект, на который пользователю предоставлены привилегии. Все остальные столбцы отражают специфические привилегии. Каждая привилегия может иметь следующие значения:

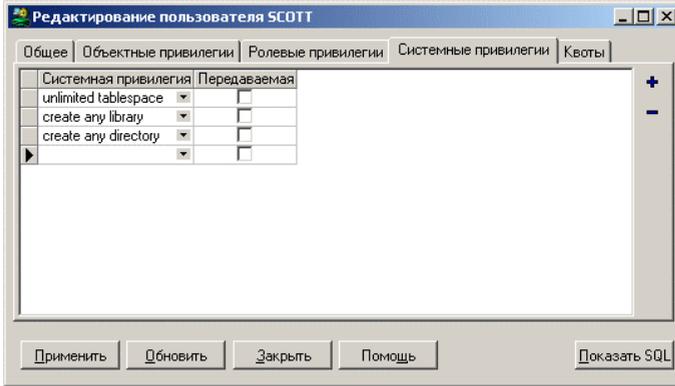
- (Пусто) - Пользователь не имеет привилегии
- Yes - Пользователь имеет привилегию
- Grantable – Пользователь имеет привилегию и может предоставить её другим пользователям или ролям

Страница Рольевые привилегии



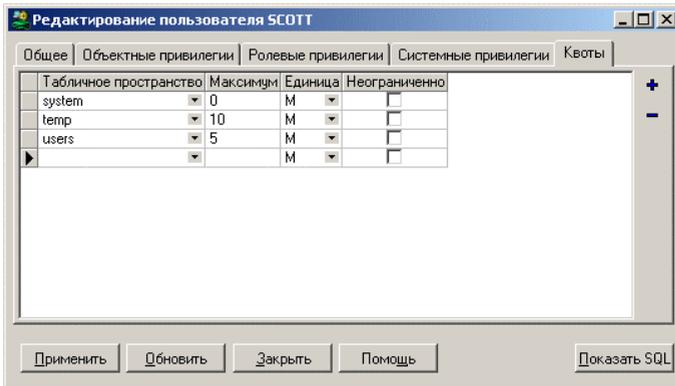
На этой странице отображены роли, предоставленные пользователю. Каждая ролевая привилегия может быть *передаваемой*, то есть пользователь может предоставлять её другим пользователям или ролям. Если ролевая привилегия является по умолчанию, роль будет включена, когда пользователь регистрируется. Если она не является по умолчанию, то ролевая привилегия должна быть включена для сессии выполнением команды `set role`, после того как пользователь зарегистрировался.

Страница Системные привилегии



На этой странице отображены системные привилегии, предоставленные пользователю. Каждая системная привилегия может быть *передаваемой*, то есть пользователь может предоставить её другим пользователям или ролям.

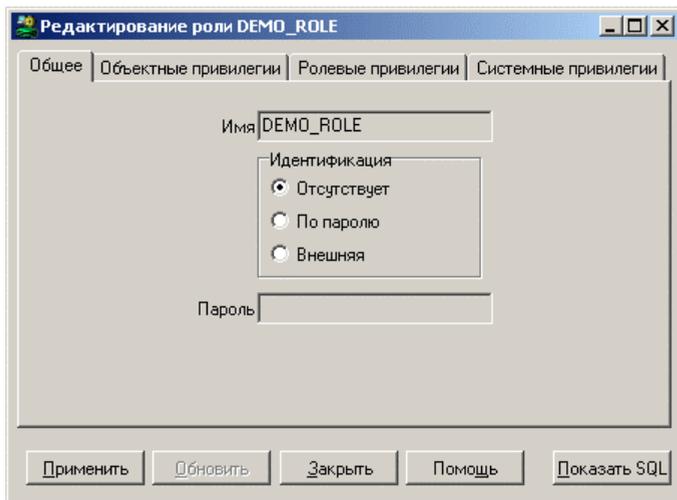
Страница Квоты



На этой странице отображены квоты пользователя на табличное пространство. Вы можете ввести *максимум* (максимальное количество) байт, килобайт или мегабайт, которые могут быть выделены пользователю в табличном пространстве, или вы можете указать, что квота является *неограниченной*.

8.10 Редактор определения роли

Редактор роли позволит вам создать или изменить роль и её привилегии:

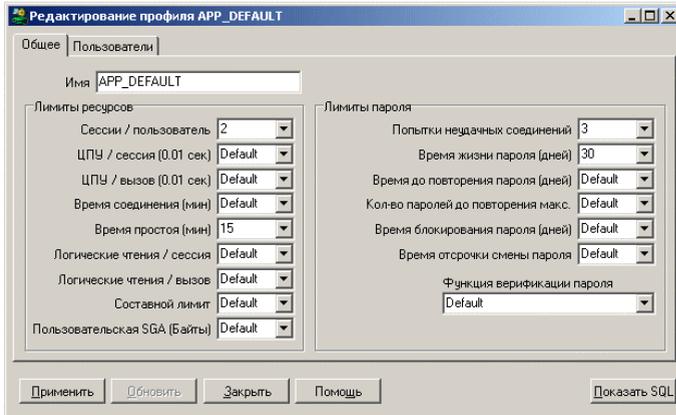


По умолчанию, роль идентифицировать не требуется. Вы также можете определить для роли идентификацию по паролю или внешнюю идентификацию. Идентификация по паролю требует, чтобы вы ввели пароль, и будет полезным, если роль не предоставлена пользователям по умолчанию. Подобная роль должна быть активизирована для пользователя, после того, как он зарегистрировался, что требует введения пароля.

Три другие страницы (Объектные привилегии, Рольевые привилегии и Системные привилегии) работают также, как и соответствующие страницы функции определения пользователя.

8.11 Редактор определения профиля

Редактор профиля позволит вам редактировать пределы ресурсов, пределы паролей и пользователей профиля:



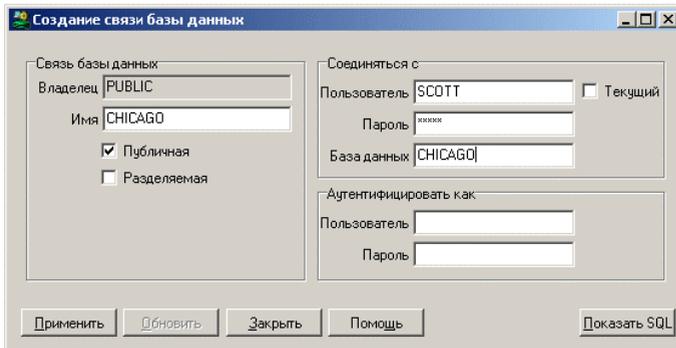
Each limit can be set to *Default*, in which case the value will be inherited from the standard profile DEFAULT, to *Unlimited*, or to a concrete value.

Каждый предел может быть установлен в значение *По умолчанию*, в этом случае значение будет унаследовано от стандартного профиля DEFAULT как неограниченное или конкретное значение.

На странице Пользователи вы можете определить, к кому из пользователей этот профиль был применен. Заметьте, что если вы удалите пользователя из профиля, то его профиль будет неявно изменён на профиль DEFAULT.

8.12 Редактор определения связи баз данных

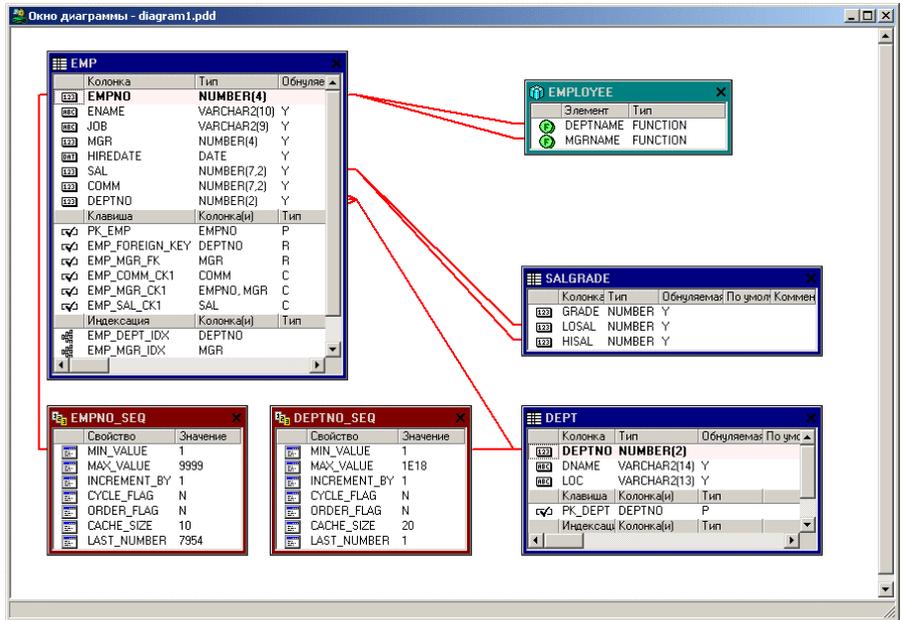
Редактор связи баз данных позволит вам создать и редактировать связь баз данных:



Заметьте, что свойства *Пароль* и *Аутентифицировать как* не будут запрошены из БД, когда вы редактируете существующую связь баз данных. Вы должны будете вводить эту информацию каждый раз, когда редактируете связь баз данных.

9. Диаграммы

Окно диаграмм позволяет вам создавать графическое представление выбранных объектов. Таким образом, вы можете просто отобразить (часть) объекты базы данных вашего приложения или проекта и их связи:



Вы можете включить в диаграмму объекты всех подходящих типов, и включить связи между ними. Связи внешних ключей между таблицами включаются автоматически. Большим диаграммам вы можете придать дополнительную структурированность, используя группы объектов.

Диаграммы могут быть использованы как для целей документирования, так и для создания рабочих пространств. Щелчок правой кнопкой мыши на объекте предоставляет доступ ко всем функциям объекта, а двойной щелчок на объекте вызовет действие по умолчанию для данного типа объекта.

9.1 Создание диаграмм

Что бы создать новую диаграмму, выберите *Окно диаграмм* из подменю *Создать* меню *Файл*. Появится пустое Окно диаграмм. Что бы включить в диаграмму объект базы данных, перетащите его из Объектного навигатора в Окно диаграмм. Вы можете перетащить несколько объектов одновременно.

Что бы передвинуть объект, щелкните мышью на заголовке объекта и перетащите его в нужное место. Что бы изменить размер объекта, вы можете перетащить границу или угол. Что бы удалить объект из диаграммы, нажмите на X в верхнем правом углу объекта. Это влияет только на диаграмму, а объект базы данных не будет удален.

Отображение и скрытие элементов

Что бы отобразить или скрыть необходимый элемент объекта диаграммы, щелкните правой кнопкой мыши на нем и перейдите к подменю *Элементы*. Здесь вы увидите подменю со всеми элементами, которые применимы для выбранного объекта. Для таблицы вы можете выбрать, какие элементы колонки, ключа или индекса вы хотите видеть. Вы так же можете скрыть полностью секцию ключей или индексов.

По умолчанию все элементы будут отображены, но если вы щелкните правой кнопкой мыши на объекте и выберите *Установить по умолчанию*, то набор видимых элементов будет применен ко всем объектам, которые будут добавлены к диаграмме. Вы можете сделать это для всех типов объектов.

Добавление связей

Что бы добавить связь между 2 объектами, щелкните на значке элемента исходного объекта, и перетащите его на элемент целевого объекта. Будет отображена линия между 2 элементами объектов. Вы можете перенести линию к другому элементу, выбрав линию и перенеся выбранную точку вверх или вниз. Что бы удалить линию, выберите ее и нажмите на клавишу *Delete*.

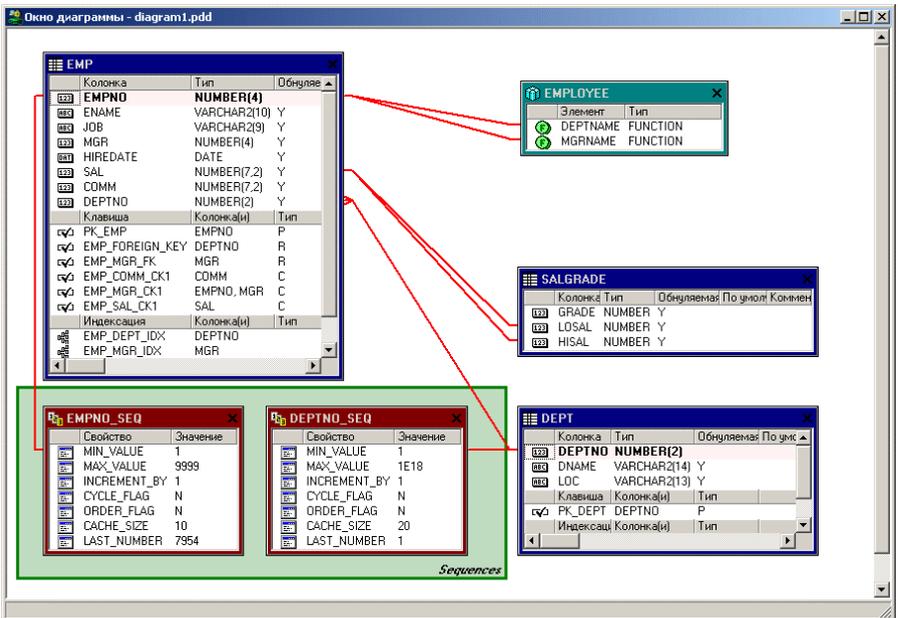
Если вы включите таблицу в диаграмму, связи внешних таблиц с другими таблицами на диаграмме будут включены автоматически. Эти связи могут быть удалены, если необходимо.

По умолчанию связи будут отображены как прямая линии. Вы можете добавить дополнительные изгибы, щелкнув правой кнопкой мыши на линии и выбрав *Вставить изгиб* из всплывающего меню. Вы можете так же нажать клавишу *Ctrl*, перед тем как щелкнуть мышью на линии, что бы вставить изгиб. В результате новая точка выбора будет добавлена к линии.

Что бы создать связь один-ко-многим, щелкните правой кнопкой мыши по концу линии и выберите *Изменить конец линии* из всплывающего меню.

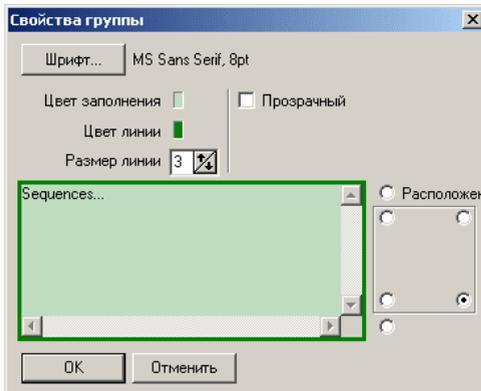
Добавление групп

Что бы добавить большой диаграмме дополнительную структурированность, вы можете создать группы объектов. Группы могут иметь отличные цвета, и могут иметь заголовки в верхней или нижней, левой или правой части:



Что бы добавить группу, щелкните правой кнопкой мыши на свободном месте диаграммы и выберите *Добавить группу* из всплывающего меню. Новая группа появится в этом месте, размеры которой вы можете изменить до необходимого значения. Все объекты, которые полностью покрываются группой, станут неявно частью этой группы. И перемещая группу, вы будете так же перемещать включенные объекты. Если вы нажмете клавишу *Ctrl*, пока перемещаете группу, то включенные объекты не будут перемещаться.

Что бы изменить внешний вид группы, щелкните правой кнопкой мыши на ней и выберите элемент *Свойства* из всплывающего меню. Вы можете так же сделать двойной щелчок по группе. Появится следующий диалог:



Вы можете изменить шрифт, цвет заливки, цвет линии и размер линии группы, и сделать группу прозрачной. Вы можете так же ввести заголовок группы и указать положение заголовка.

Что бы сделать внешний вид группы по умолчанию, щелкните правой кнопкой мыши на ней и выберите *Установить по умолчанию*. Это будет применено ко всем группам, которые будут добавлены к диаграмме.

Что бы удалить группу из диаграммы, выберите ее и нажмите клавишу *Delete*. Это удалит группу, но не включенные в нее объекты.

9.2 Сохранение и открытие файла диаграммы

Что бы сохранить диаграмму в файл, выберите пункт *Сохранить* из меню *Файл*. Диаграмма будет сохранена в файле с расширением *.pdd*, который может быть открыт позже. Что бы задать стандартный путь расположения диаграммных файлов, перейдите к странице предпочтений *Каталоги* и задайте соответствующую опцию.

9.3 Обновление диаграммы

Диаграмма является статическим представлением базы данных. Если объекты базы данных изменятся, вам будет необходимо обновить диаграмму, что бы отобразить изменения. Что бы сделать это, щелкните правой кнопкой мыши на объекте диаграммы и выберите из всплывающего меню *Обновить из БД*. Что бы обновить все объекты диаграммы, щелкните правой кнопкой мыши на ней и выберите *Обновить все из БД*.

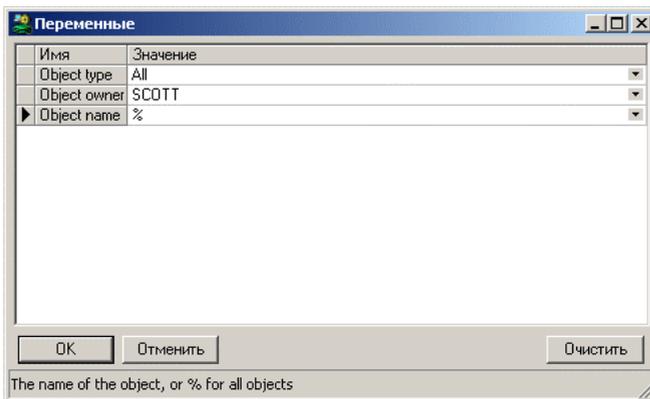
Если объект удален из базы данных, вам будет необходимо так же удалить его из диаграммы. Если объект был переименован в базе данных, вам будет необходимо удалить объект из диаграммы и добавить его с новым именем.

10. Отчеты

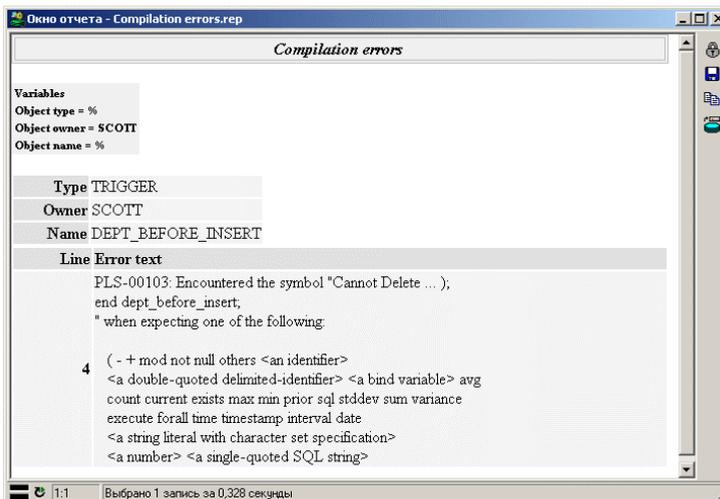
Чтобы запустить отчеты на основе ваших прикладных данных или словаря Oracle, вы можете использовать встроенную в PL/SQL Developer функцию *Отчеты*. Существует набор стандартных отчетов, и вы легко можете создать свои собственные пользовательские отчеты. Эти пользовательские отчеты могут быть сохранены в отчетном файле, и в свою очередь могут быть включены в меню отчетов. Это позволяет очень легко запускать свои собственные, часто используемые отчеты.

10.1 Стандартные отчеты

Стандартные отчеты доступны через главное меню *Отчеты*. Если, например, вы выберете отчет *Ошибки компиляции*, у вас сначала будет запрошен объектный тип, владелец и имя, для которого вы хотите просмотреть текущие ошибки компиляции:



После ввода соответствующих значений и нажатия кнопки *ОК*, отчет будет отображен:



Результатом будет HTML документ, отображённый, в зависимости от предпочтений (смотрите главу 16.8), внутренней PL/SQL Developer программой для просмотра HTML или в Internet Explorer ActiveX управляющим элементом.

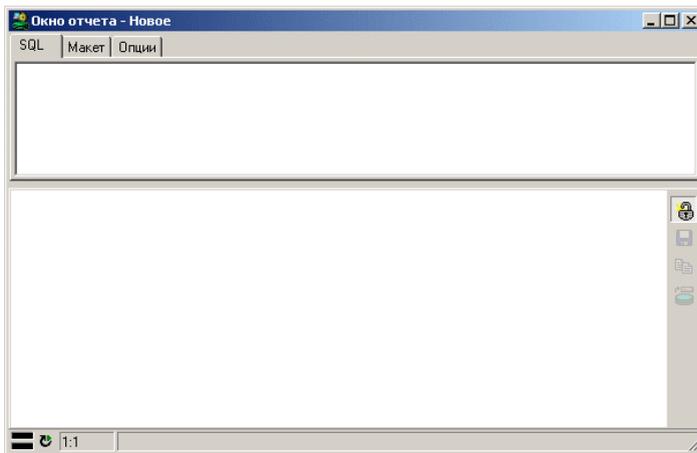
Вы можете непосредственно напечатать отчёт, нажав на панели инструментов кнопку *Печать*, или вы можете сохранить отчёт в формате HTML, нажав в правой части Окна отчёта кнопку *Сохранить результат*. Кнопка *Копировать как HTML* копирует результаты отчёта в буфер обмена. Кнопка *Экспортировать результаты* позволит вам экспортировать результаты в форматы CSV, TSV или XML или экспортировать результаты непосредственно в Excel.

Кнопка *Редактировать отчет* обеспечивает доступ к определению отчёта, если отчёт не заблокирован.

Для дополнительных опций вы можете щёлкнуть правой кнопкой мыши на результатах.

10.2 Пользовательские отчёты

Пользовательские отчёты очень просто создать. Сначала, вы создаёте новый, пустой отчёт нажав на панели инструментов кнопку *Создать*, или выбрав из всплывающего меню пункт *Окно отчета*. Появится пустое Окно отчёта:



В редакторе *SQL* вы можете ввести для вашего отчёта единственный SQL оператор SELECT. Для главных/детальных отчётов вы можете использовать связанный оператор SELECT (это будет описано позже). Давайте начнём с простой и хорошо нам известной таблицы *emp*, введём оператор SELECT и нажмём на панели инструментов кнопку *Выполнить*:

The screenshot shows a window titled "Окно отчета - select * from emp order by deptno, ename". The window contains a SQL editor with the query `select * from emp order by deptno, ename`. Below the editor is a table with the following data:

Empno	Ename	Job	Mgr	Hiredate	Sal	Comm	Deptno
7782	CLARK	MANAGER	7839	09.06.1981	2450,00		10
7839	KING	PRESIDENT		17.11.1981	5000,00		10
7934	MILLER	CLERK	7782	23.01.1982	1300,00		10
7876	ADAMS	CLERK	7788	12.01.1983	1100,00		20
7902	FORD	ANALYST	7566	03.12.1981	3000,00		20
7566	JONES	MANAGER	7839	02.04.1981	2975,00		20
7788	SCOTT	ANALYST	7566	09.12.1982	3000,00		20
7369	SMITH	CLERK	7902	17.12.1980	800,00		20

The status bar at the bottom indicates "Выбрано 13 записей за 0,234 секунды".

Это очень простой отчёт, все свойства которого установлены по умолчанию. В следующих главах будет объяснено, как вы можете усовершенствовать функциональные возможности отчёта.

10.3 Переменные

Очень часто, перед запуском ваших отчётов, в них будет требоваться, дополнительно, ввести некоторые данные. Давайте предположим, что вы хотите ограничить отчёт из предыдущей главы сотрудниками только одного отдела. В этом случае вам нужно использовать подстановочную переменную в тексте SQL:

```
select * from emp
where deptno = &department
order by ename
```

Когда этот отчёт запущен, перед выполнением оператора SELECT у пользователя запрашивается значение для *department*. Это, конечно, не исключает возможности ошибки. Что если пользователь ничего не введёт? Оператор вызовет исключительную ситуацию *ORA-00936: missing expression* (пропущено выражение). Что если пользователь введёт значение, которое не является числом? Или число, которое не существует в таблице *dept*? Чтобы предотвратить эти ситуации, вы можете использовать опции параметров, описанные в следующих главах. Эти опции должны быть указаны в скобках, точно также как атрибуты в документе HTML или XML.

Опция Name

Имя переменной используется как подсказка в форме ввода переменной. Поэтому вам следует убедиться, что оно объясняет смысл переменной. Name – единственная опция, которую не нужно указывать в скобках, если она является единственной опцией. Если имя содержит пробелы или другие специальные символы, заключите его в двойные кавычки. Два следующих примера эквивалентны друг другу:

```
select * from emp
where deptno = &"Department number"
order by ename
```

and

```
select * from emp
where deptno = &<name="Department number">
order by ename
```

Опция Hint

Помимо опции Name, вы можете дополнительно указать текст подсказки. Этот текст будет отображён в строке статуса формы ввода переменной, когда у этой переменной есть фокус:

```
select * from emp
where deptno = &<name="Department number"
             hint="Only employees from this department will be listed">
order by ename
```

Опция Type

Опция Type ограничивает информацию, которая может быть введена пользователем, а также контролирует то, как значение будет вставлено в результирующий текст SQL:

```
select * from emp
where deptno = &<name="Department number"
             type="integer">
order by ename
```

В этой ситуации, пользователь может ввести только значения, которые являются действительными целочисленными значениями. Введённое значение будет вставлено в текст SQL, как оно есть. Однако для переменных string типа, вы захотели бы, чтобы значение было заключено в кавычки. Рассмотрим следующий оператор SELECT:

```
select * from emp
where ename = &<name="Employee name"
             type="string">
```

Если вы введёте *SCOTT* (без кавычек) в форме ввода переменной, результирующий SQL будет выглядеть так:

```
select * from emp
where ename = 'SCOTT'
```

Корректными являются типы: *none*, *string*, *integer*, *float*, and *date*. Для значений типа дата проверка не осуществляется, поэтому пользователь может дополнительно вводить выражения типа дата, например *sysdate*, *trunc(sysdate, 'MM')* и так далее.

Опция Required

Если вам необходимо чтобы значение для некоторой переменной было введено, установите значение её опции *required* в "yes" или "true":

```
select * from emp
where deptno = &<name="Department number"
             required="yes">
order by ename
```

Теперь отчёт может быть запущен, только если значение номера отдела задано.

Опция Default

Чтобы представить значение по умолчанию для переменной, установите опцию *default*:

```
select * from emp
where deptno = &<name="Department number"
             default="10">
order by ename
```

Значение по умолчанию также может быть оператором SELECT. Этот оператор SELECT должен возвращать только одну строку с одной колонкой. Например:

```
select * from emp
where deptno = &<name="Department number"
             default="select min(deptno) from dept">
order by ename
```

В этом случае значением по умолчанию будет наименьший номер отдела.

Опция Ifempty

В качестве альтернативы требованию переменной быть заданной или указания значения по умолчанию, вы можете указать значение *ifempty* для случая, когда пользователь не задал значение:

```
select * from emp
where ename like &<name="Employee name"
               type="string"
               ifempty="%">
```

Если пользователь не определил значение, будет использован знак процента, и будут найдены все сотрудники. Эта опция не может быть установлена вместе с опцией *required*.

Опция Uppercase

Если у вас имеется переменная строкового типа, и вы хотите преобразовать её значение в верхний регистр, установите значение опции *uppercase* в "yes" или "true":

```
select * from emp
where ename = &<name="Employee name"
             type="string"
             uppercase="yes">
```

Значение отображается так, как оно было введено, но оно будет преобразовано в верхний регистр в результирующем тексте SQL.

Опция Readonly

Переменная может быть определена, как доступная только для чтения, указав *readonly="yes"*. Пользователь будет видеть переменную и её значение, но не сможет изменить его.

Опция Hidden

Вы можете скрыть переменную, указав *hidden="yes"*.

Опция List

Опция *list* предоставляет пользователю список всех возможных значений. Вы можете указать разделённый запятыми список элементов:

```
select * from emp
where deptno = &<name="Department number"
              list="10, 20, 30, 40">

order by ename
```

Это позволит пользователю выбрать один из четырёх элементов или ввести другое значение. В этом случае имеет больший смысл использовать оператор SELECT:

```
select * from emp
where deptno = &<name="Department number"
              list="select deptno from dept order by deptno">

order by ename
```

Вы можете дополнительно снабдить каждый элемент описанием:

```
select * from emp
where deptno = &<name="Department number"
              list="10, ACCOUNTING,
                  20, RESEARCH,
                  30, SALES,
                  40, OPERATIONS"
              description="yes">

order by ename
```

Теперь каждый элемент сопровождается описанием, и добавлена опция *description*. Этот список будет только отображать описания, а действительное значение будет использовано в результирующем тексте SQL.

Также вы сможете сделать то же самое с оператором SELECT с 2 полями:

```
select * from emp
where deptno = &<name="Department number"
      list="select deptno, dname from dept order by dname"
      description="yes">
order by ename
```

Чтобы ограничить пользователя только элементами в списке, установите опцию *restricted*:

```
select * from emp
where deptno = &<name="Department number"
      list="select deptno, dname from dept order by dname"
      description="yes"
      restricted="yes">
order by ename
```

Теперь пользователь может выбирать значения только из списка и не сможет вводить любые другие значения вручную.

Элементы в списке могут иногда зависеть от значения другой переменной. Представьте, что вы хотите создать отчёт, который показывает все столбцы пользовательской таблицы. Таблица идентифицируется владельцем и именем. Для выбора владельца вы можете использовать список всех пользователей в базе данных. Для выбора имени таблицы вы можете запросить вид *all_tables* для выбранного владельца:

```
select * from all_tab_columns
where owner = &<name="Owner"
      type="string"
      list="select username from all_users order by username">
and table_name = &<name="Table"
      type="string"
      list="select table_name from all_tables
      where owner = :owner
      order by table_name">
order by column_id
```

Как вы можете видеть, оператор SELECT для второй (таблицы) списка обращается к первой переменной через связанную переменную *:owner*.

Заметьте, что имя связанной переменной не может быть длиннее 30 символов и не может содержать пробелы или другие специальные символы. Пробелы в имени переменной будут преобразованы в символы подчёркивания. Если у переменной было имя *Owner of the table*, то связанная переменная будет *:owner_of_the_table*.

Также заметьте, что если значение переменной остаётся пустым, то зависимый список также будет пустым. В этой ситуации ни один запрос не будет выполнен.

Опция Checkbox

Если пользователь может выбрать одну из двух различных возможностей, вы можете использовать опцию *checkbox*. Эта опция должна сопровождаться отмеченными и неотмеченными значениями:

```
select * from emp
order by hiredate &<name="Descending sort order"
      checkbox="desc, asc">
```

Несмотря на это, вы должны всегда определять 2 значения, одно из них может быть пустым:

```
select * from emp
order by hiredate &<name="Descending sort order"
```

```
checkbox="desc,">
```

Так как порядок сортировки по умолчанию является возрастающим, эти 2 примера являются эквивалентными.

Опции Prefix и Suffix

Если значение переменной пустое, то это может подразумевать, что некоторые фиксированные части текста SQL также должны быть опущены. Рассмотрим пример, когда вы хотите задать дополнительную сортировку столбца. Если столбец не указан, то фразу *order by* следует опустить. В этом случае вы можете указать этот текст как *префикс* или *суффикс* переменной. Если значение переменной пустое, префикс и суффикс будут опущены:

```
select * from emp
&<name="Descending sort column"
list="empno, ename, hiredate"
prefix="order by "
suffix=" desc">
```

Если значение не указано для этой переменной, результат будет также пустым:

```
select * from emp
```

Если колонка указана, то результат будет таким:

```
select * from emp
order by empno desc
```

Обратите внимание на пробелы после префикса и до суффикса. Эти пробелы важны, потому что префикс, значение и суффикс конкатенируются в том виде, в каком они указаны!

Символ Escape

Если вы хотите использовать в тексте SQL символ `&`, который не должен быть интерпретирован как подстановочная переменная, используйте вместо него двойной символ `&`. В следующем примере будут найдены все сотрудники отдела 'R&D':

```
select * from emp
where emp.deptno in (select dept.deptno from dept
                    where dname = 'R&&D')
order by empno desc
```

Если в тексте вместо 'R&&D' вы использовали 'R&D', у вас будет запрошена переменная D.

10.4 Доводка макета

После открытия нового Окна отчёта, ввода оператора SELECT и выполнения отчёта, все свойства макета будут установлены по умолчанию. Это будет простой табличный макет со стандартными значениями свойств шрифтов, стиля таблицы, цветов и так далее. Чтобы усовершенствовать стандартный макет, переключитесь на страницу *Макет*. Для нового отчёта эта страница будет выглядеть следующим образом:

The screenshot shows the 'Report Designer' window with the 'Макет' (Layout) tab selected. The layout configuration table is as follows:

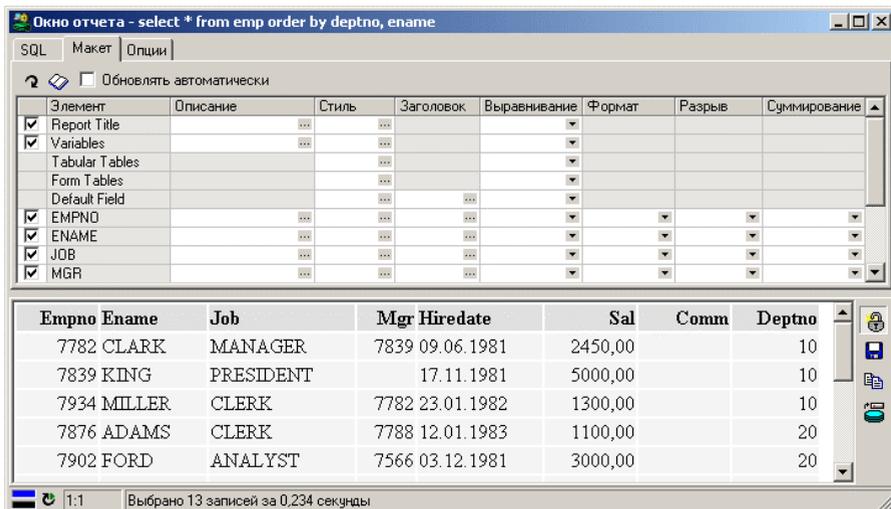
Элемент	Описание	Стиль	Заголовок	Выравнивание	Формат	Разрыв	Суммирование
<input checked="" type="checkbox"/>	Report Title	▼			
<input checked="" type="checkbox"/>	Variables				
	Tabular Tables	▼			
	Form Tables	▼			
	Default Field	▼			

The report preview table is:

Empno	Ename	Job	Mgr	Hiredate	Sal	Comm	Deptno
7782	CLARK	MANAGER	7839	09.06.1981	2450,00		10
7839	KING	PRESIDENT		17.11.1981	5000,00		10
7934	MILLER	CLERK	7782	23.01.1982	1300,00		10
7876	ADAMS	CLERK	7788	12.01.1983	1100,00		20
7902	FORD	ANALYST	7566	03.12.1981	3000,00		20
7566	JONES	MANAGER	7839	02.04.1981	2975,00		20
7788	SCOTT	ANALYST	7566	09.12.1982	3000,00		20

At the bottom of the window, the status bar shows '1:1' and 'Выбрано 13 записей за 0,234 секунды'.

Как вы можете видеть, вы можете определять различные свойства макета для всех элементов макета (заголовков отчёта, переменные и поля). Если вы хотите определить свойства одного отдельного поля, сначала вы должны нажать на панели инструментов кнопку *Обновить список полей*:



Теперь все поля результирующего набора оператора SELECT будут включены в таблицу макета, и вы сможете указать свойства макета для каждого отдельного поля. Если вы оставите свойства поля *Стиль*, *Заголовок* или *Выравнивание* пустыми, будут использованы соответствующие свойства из *Default Field* (Поле по умолчанию).

После изменения свойства макета, вы можете выполнить отчёт снова, чтобы просмотреть, как это отразилось на результатах. Альтернативно, вы можете включить на панели инструментов опцию *Обновить автоматически*, в этом случае каждое изменение немедленно отразится на результатах.

В следующих главах будут детально описаны свойства макета.

Отображение

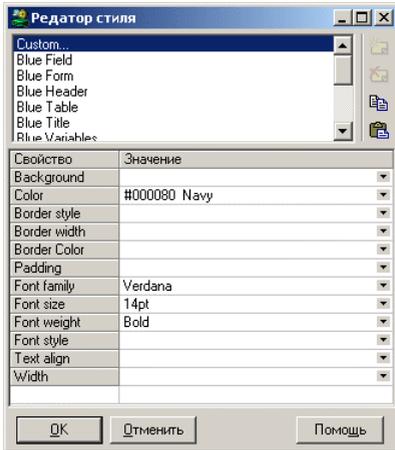
Флажок в самом левом столбце таблицы макета показывает, будет ли отображён элемент или нет. Это позволит вам скрыть заголовок отчёта или переменные и позволит вам контролировать, следует ли отображать специфические поля или нет.

Описание

Описание *Report Title* (Заголовка отчёта) будет отображено в верхней части отчёта и после каждого разрыва страницы. Описание *Variables* (Переменные) будет отображено над переменными.

Для отдельных полей вы можете определить описание, чтобы отменить стандартное имя поля. Это описание будет использовано в заголовке столбца.

Стиль



Стиль контролирует внешний вид элементов макета. Нажмите кнопку в ячейке (...), чтобы вызвать редактор стиля:

В списке в верхней части Редактора стиля вы можете видеть все стандартные стили и *Пользовательский* стиль. Вы можете выбрать стандартный стиль из Библиотеки стилей (смотрите главу 10.5), или же вы можете выбрать пользовательский стиль, изменив различные свойства для текущего элемента макета.

Заметьте, что эта система стилей основана на стандарте Cascading Style Sheets (CSS). Свойства, которые вы видите в Редакторе стиля, одни из наиболее часто вами используемых, но вы также можете добавить ниже стандартных свойств и другие свойства CSS и значения. Для большей информации о Cascading Style Sheets смотрите <http://www.w3.org/TR/REC-CSS2>.

Кнопки *Копировать* и *Вставить* позволят вам копировать свойства стиля, так чтобы вы быстро могли получить новый стиль из уже существующего.

Если вы ничего не задаете в качестве стиля полей, то будет использован стиль *Default Field* из Библиотеки стилей. Вы можете переопределить это, задав стиль элемента *Default Field* макета. Вы можете отменить этот стиль по умолчанию, указав соответствующий стиль. Заметьте, что каждый заголовок поля наследует свой стиль от стиля *Default Header*, так что вам нужно переопределять только те свойства, которые вы хотите изменить. Если, например, вы определяете стиль *Default Field* как тип шрифта - Verdana, размер - 14 точек, цвет - синий и хотите отобразить столбец *EMPNO* в стиле тип шрифта - Verdana, размер - 14 точек, цвет - красный, вам всего лишь надо задать цвет на уровне *EMPNO*. Тип и размер шрифта будет автоматически наследованы от уровня *Default Field*.

Стили по умолчанию различных видов элементов макета описаны в главе 10.5.

Заголовок

Точно также как вы можете устанавливать стиль для данных отдельных полей, вы можете устанавливать стиль для их заголовков. Если вы ничего не определяете, то будет использован стиль *Default Field* из Библиотеки стилей. Вы можете переопределить это, указав стиль элемента *Default Field* макета. Вы можете переопределить этот стиль по умолчанию, указав соответствующий стиль. Заметьте, что каждый заголовок поля наследует свой стиль от стиля *Default Header*, так что вам нужно переопределять те свойства, которые вы хотите изменить.

Выравнивание

Вы можете выровнять элементы макета следующими способами:

- Left – Элемент будет выровнен по левому краю внутри его ячейки
- Right – Элемент будет выровнен по правому краю его внутри его ячейки
- Center – Элемент будет отцентрирован внутри его ячейки
- Default – Числа будут выровнены по правому краю, все остальные элемента – по левому
- None – Не указано никакого выравнивания

Заметьте, что вы также можете установить свойство *Выравнивание текста* для элемента макета. Оно будет иметь старшинство перед опцией выравнивания, которую вы можете указать в таблице макета. Также заметьте, что эффект от выравнивания может зависеть от свойства *Ширина* стиля элемента макета.

Формат

По умолчанию формат поля контролируется типом данных, масштабом и точностью. Очень часто вы будете использовать вычисляемые или агрегируемые поля, в этом случае масштаб и точность неизвестны. В этих ситуациях вы можете указать формат.

Свойство Формат влияет на поля с датами и числовые поля. Для числовых полей вы можете использовать 9 для цифр с незначущими нулями, 0 для нормальных цифр, G для тысячного разделителя, D для десятичного разделителя и E для научной системы обозначений. Вы можете дополнительно указать текст в двойных или одиночных кавычках. Несколько примеров:

Format	Value = 1234	Value = 5	Value = 0.1
999G990D00	1,234.00	5.00	0.10
9G990	1,234	5	0
0000	1234	0005	0000
0"%"	1234%	5%	0%
0D000e+00	1.234e+03	5.000e+00	1.000e-01
0D000e-0	1.234e3	5.000e0	1.000e-1

Для полей с датами вы можете использовать стандартные форматы Windows для даты и времени:

Разрыв

Свойство break макета позволит вам структурировать результаты ваших отчётов. Предположим, что вы хотите отобразить отделы и их сотрудников. Вы можете создать для этого следующий запрос:

```
select d.*, e.*
from emp e, dept d
where e.deptno = d.deptno
order by d.deptno, e.empno
```

При этом, конечно, появится простой табличный отчёт:

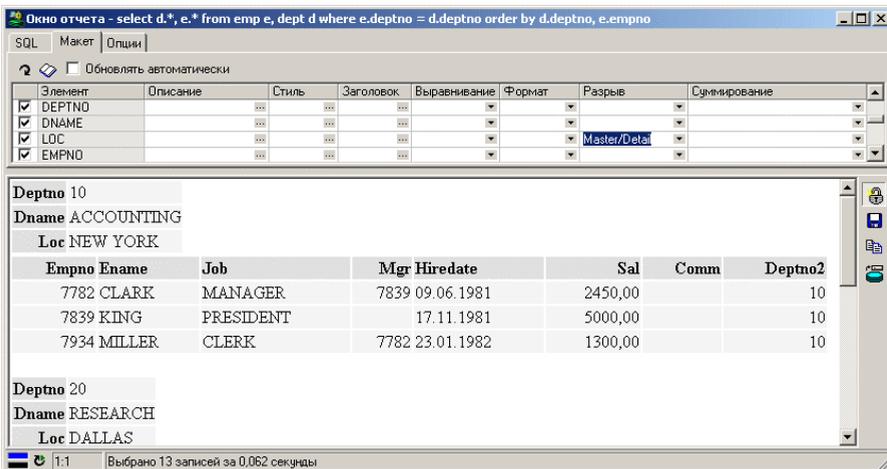
Deptno	Dname	Loc	Empno	Ename	Job	Mgr	Hiredate	Sal	Comm	Deptno2
10	ACCOUNTING	NEW YORK	7782	CLARK	MANAGER	7839	09.06.1981	2450,00		10
10	ACCOUNTING	NEW YORK	7839	KING	PRESIDENT		17.11.1981	5000,00		10
10	ACCOUNTING	NEW YORK	7934	MILLER	CLERK	7782	23.01.1982	1300,00		10
20	RESEARCH	DALLAS	7369	SMITH	CLERK	7902	17.12.1980	800,00		20
20	RESEARCH	DALLAS	7566	JONES	MANAGER	7839	02.04.1981	2975,00		20
20	RESEARCH	DALLAS	7788	SCOTT	ANALYST	7566	09.12.1982	3000,00		20
20	RESEARCH	DALLAS	7876	ADAMS	CLERK	7788	12.01.1983	1100,00		20
20	RESEARCH	DALLAS	7902	FORD	ANALYST	7566	03.12.1981	3000,00		20
30	SALES	CHICAGO	7499	ALLEN	SALESMAN	7698	20.02.1981	1600,00	300,00	30
30	SALES	CHICAGO	7521	WARD	SALESMAN	7698	22.02.1981	1250,00	500,00	30

Указав *Break* для столбца *LOC*, вы добьётесь несколько большей структурированности вашего отчёта:

Deptno	Dname	Loc	Empno	Ename	Job	Mgr	Hiredate	Sal	Comm	Deptno2
10	ACCOUNTING	NEW YORK	7782	CLARK	MANAGER	7839	09.06.1981	2450,00		10
			7839	KING	PRESIDENT		17.11.1981	5000,00		10
			7934	MILLER	CLERK	7782	23.01.1982	1300,00		10
20	RESEARCH	DALLAS	7369	SMITH	CLERK	7902	17.12.1980	800,00		20
			7566	JONES	MANAGER	7839	02.04.1981	2975,00		20
			7788	SCOTT	ANALYST	7566	09.12.1982	3000,00		20
			7876	ADAMS	CLERK	7788	12.01.1983	1100,00		20
			7902	FORD	ANALYST	7566	03.12.1981	3000,00		20
30	SALES	CHICAGO	7499	ALLEN	SALESMAN	7698	20.02.1981	1600,00	300,00	30
			7521	WARD	SALESMAN	7698	22.02.1981	1250,00	500,00	30

Повторение значений полей до (и включая) *LOC* будет подавлено. Вкратце: каждая отдельная запись, будет отображена лишь один раз. В этом случае необходимо, чтобы записи были отсортированы по отделам (столбец *DEPTNO*). Если вы хотите повторять заголовок после каждого разрыва, используйте опцию *Break + Header*.

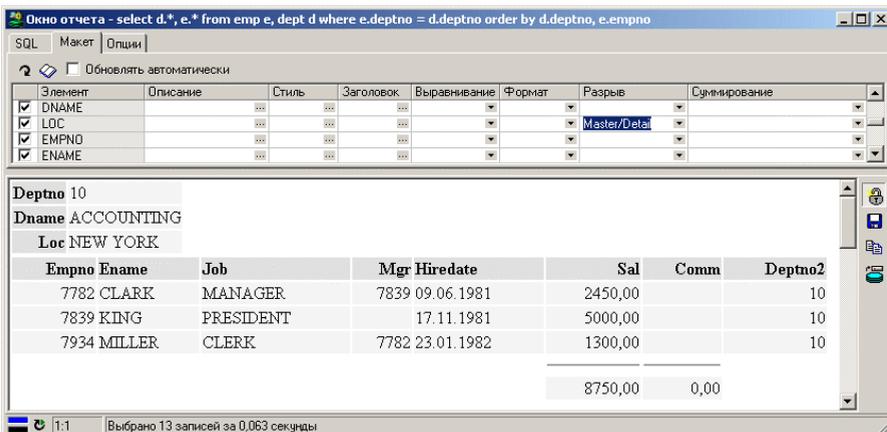
Альтернативно вы можете выбрать Главный/Детальный макет:



В вашем отчёте имеются многократные разрывы. Фраза *ORDER BY* запроса должна гарантировать, что результирующий набор отсортирован по всем столбцам разрыва.

Суммирование

Свойство «суммирование» позволяет вам включать в отчет сумму для поля:



Вы можете указать, что сумма должна быть сосчитана на уровне разрыва, на уровне отчета или на обоих уровнях.

Заголовки страниц

Чтобы определить заголовок страницы, вам сначала нужно установить размер страницы (смотрите главу 10.6). Название отчёта будет повторено после каждого разрыва страницы и может включать в себя следующие переменные:

- OSUser – Пользователь операционной системы.
- DBUser – Пользователь, подсоединённый к базе данных.
- Database – База данных, к которой подсоединён пользователь.
- Date – Текущая дата
- Time – Текущее время.
- Page – Текущая страница

Чтобы включить подобную переменную в заголовок страницы, поставьте перед ней знак `&`. Например:

```
All employees (&dbuser@&database)
```

Если вы запускаете отчёт, будучи подключённым как пользователь *scott* к базе данных *chicago*, то отчёт будет выглядеть следующим образом:

```
All employees (scott@chicago)
```

Как только отчёт отформатирован как HTML, вы можете дополнительно включить код HTML. Чтобы включить рисунок, вы можете включить тэг ``:

```
All employees <img src=logo.gif>
```

Чтобы включить выровненную по левому краю, отцентрированную и выровненную по правому краю информацию, вы можете использовать таблицу, состоящую из одной строки с 3 ячейками, и добавить информацию в соответствующую ячейку:

```
<table width="100%"> <tr> <td>&dbuser</td> <td align="center">All  
Employees</td> <td align="right">&page</td> </tr> </table>
```

Это будет отображено следующим образом:

Scott	All Employees	1
-------	---------------	---

Заметьте, что вы можете использовать эти переменные и код HTML в описании всех элементов макета, но они являются самыми полезными для названия отчёта.

10.5 Библиотека стилей

Вместо того, чтобы определять и переопределять одни и те же стили снова и снова, вы можете нажать на панели кнопку *Библиотека стилей*, чтобы создавать и изменять свои собственные стили. Если, например, вы хотите, чтобы все поля, содержащие выражения SQL были отображены в стиле тип шрифта - Courier New, размер - 12 точек, цвет – синий, просто определите стиль SQL с этими свойствами в библиотеке и примените его, где это требуется.

Существует набор имён стилей, которые имеют специальное значение:

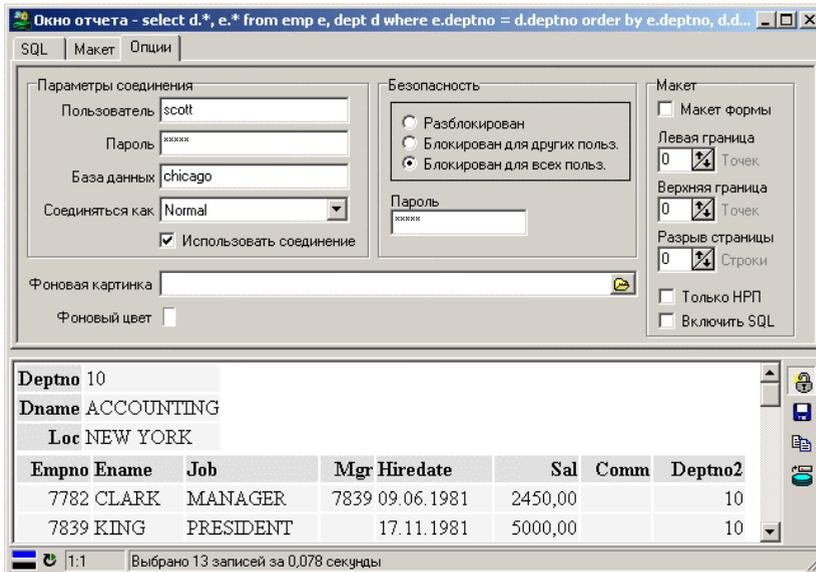
Имя стиля	Описание
Название отчёта по умолчанию	Будет применён к названию отчёта
Переменные по умолчанию	Будет применён к переменным
Таблицы по умолчанию	Будет применён ко всем таблицам (например, детальные записи)
Таблицы формы по умолчанию	Будет применён ко всем таблицам формы (например, главные записи)
Стиль поля по умолчанию	Будет применён ко всем полям, у которых стиль не определён
Заголовок поля по умолчанию	Будет применён ко всем заголовкам полей, у которых стиль не определён

Это стили по умолчанию, так что вы можете переопределить их в отдельном отчёте, определив стиль для соответствующих элементов макета.

Заметьте, что все используемые стили (стандартные и пользовательские) хранятся в файле отчёта. Когда отчёт запущен, и нужно применить стиль, PL/SQL Developer попытается найти его в библиотеке стилей. Когда она не доступна, будет использован стиль из файла отчёта. Это гарантирует, что ваш отчёт всегда будет отображаться корректно, но стили всё равно можно будет изменить.

10.6 Опции

Вы можете указывать различные опции для вашего отчёта:



Параметры соединения

Параметры соединения определяют, как будет запускаться запрос отчёта. Если, например, необходимо, чтобы ваш отчет выполнялся с использованием учётной записи *SYS*, вы можете ввести это имя в поле *Username*. Вы можете дополнительно ввести *Пароль* или можете оставить его пустым. Если вы оставите его пустым, у вас будет запрошен пароль во время выполнения отчёта. Вы также можете указать *Базу данных*, если отчёт следует всегда выполнять в одной и той же базе данных. Альтернативно вы можете ввести * в поле базы данных, которое показывает, что отчёт следует выполнять в текущей базе данных.

Опция *Использовать соединение* (текущее) позволит вам отключать или включать Параметры соединения для отчёта. Если она отключена, то будет использовано текущее соединение PL/SQL Developer.

Безопасность

Опции *Безопасность* позволяют вам блокировать определение отчёта (SQL, Макет и Опции), так чтобы пользователь не мог просмотреть или изменить его. Это может быть полезно, если Параметры соединения указывают привилегированную учётную запись (такую как *SYS*) с паролем. Если пользователь может изменить SQL, он или она может выполнить любой оператор с использованием учётной записи *SYS*, если отчёт не заблокирован.

Для опции *Безопасность* может быть выбрано одно из следующих значений:

- Разблокирован – Любой пользователь может изменять определение отчёта.
- Блокирован для других польз. – Только вы можете изменять определение отчёта, если вы знаете *Пароль* отчёта, или если вы подсоединены в PL/SQL Developer с той же учётной записью, что определена в Параметрах соединения.

- Блокирован для других польз. – У вас всегда будет запрашиваться *Пароль* прежде чем вы сможете изменить определение отчёта, независимо от того, как вы подсоединены в PL/SQL Developer.

Если отчёт заблокирован, то файл отчёта (.rep) будет зашифрован, так что его нельзя будет просмотреть или изменить в текстовом редакторе.

Опции макета

- Макет формы – По умолчанию, все записи отображаются в виде таблицы, исключение составляют мастер записи, которые отображаются в виде формы. После включения этой опции все записи будут отображены в виде формы. Это будет полезным, если каждая запись содержит много длинных полей и/или, которые не могут быть отображены одной строкой.
- Левая граница – Левая граница отчёта в пикселях. Эта опция работает только, если для просмотра HTML вы используете Internet Explorer.
- Верхняя граница – Верхняя граница отчета в пикселях. Эта опция работает только, если для просмотра HTML вы используете Internet Explorer.
- Разрыв страницы – Число строк, после которых должен произойти разрыв страницы. Заголовок отчёта будет повторен после каждого разрыва страницы.
- Только НРП – Когда эта опция включена, все пробелы в данных будут преобразованы в НРП (NBSP - прим. переводчика) (неразрывные пробелы) в HTML выводе. Когда эта опция отключена, будут использованы нормальные пробелы. Это может повлиять на макет, потому что программы для просмотра HTML будут отображать последовательные пробелы как один пробел.
- Включить SQL – Когда эта опция включена, текст SQL запроса для отчёта будет включён в HTML вывод.

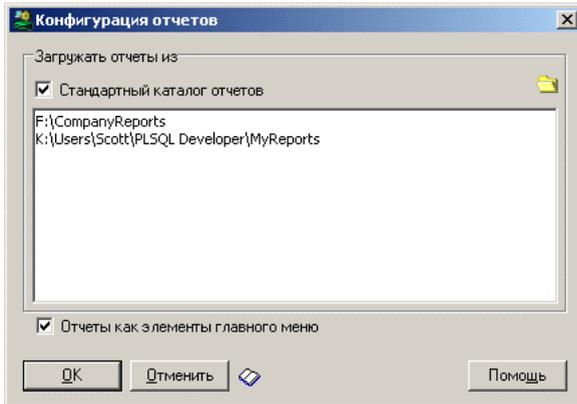
Другие опции

- Фоновая картинка – Картинка (.jpg или .gif), которая должна быть отображена на фоне этого отчёта. Вы можете использовать кнопку *Select file*, чтобы найти ваш файл рисунка.
- Цвет фона – Цвет, который должен быть отображён на фоне этого отчёта.

10.7 Меню «Отчёты»

По умолчанию, пункт меню *Отчёты* включён в главное меню и содержит только стандартные отчёты. Эти стандартные отчёты находятся в подкаталоге Reports установочного каталога PL/SQL Developer. Меню является прямым отображением структуры этого каталога. При выборе пункта меню этот отчёт будет выполнен.

Вы можете добавить или удалить отчёты из этого каталога, но вы также можете сконфигурировать меню отчётов при помощи элемента *Конфигурировать отчёты* из меню *Инструменты*. При этом появится следующий диалог:



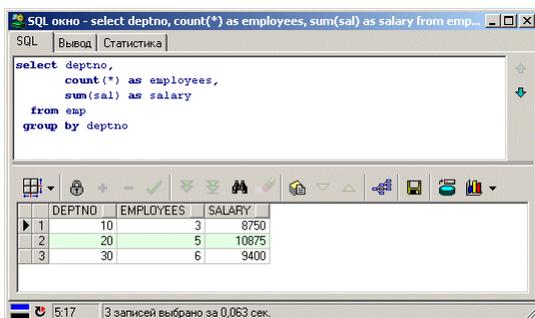
Опция *Стандартный каталог отчетов* показывает, должно ли меню *Отчёты* включать в себя подкаталог Reports из установочного каталога PL/SQL Developer. Этот каталог содержит стандартные отчёты. Поле редактора под этой опцией содержит список каталогов, которые следует включить. Вы можете ввести каталоги вручную или можете выбрать каталог, нажав справа кнопку *Включить каталог в список*.

Заметьте, что файлы отчётов и подкаталоги будут объединены в меню *Отчёты*. Вам следует избегать одинаковых имён подкаталогов и файлов отчётов внутри этого списка каталогов, в противном случае, вы получите неоднозначные пункты меню в меню *Отчёты*.

Библиотека стилей также доступна из этого диалога конфигурации отчёта (смотрите главу 10.5).

11. Графики

Что бы быстро отобразить графическое представление данных колонки запрошенных в SQL окне или Окне отчета, вы можете использовать Окно графика. Рассмотрим следующий запрос в SQL окне:

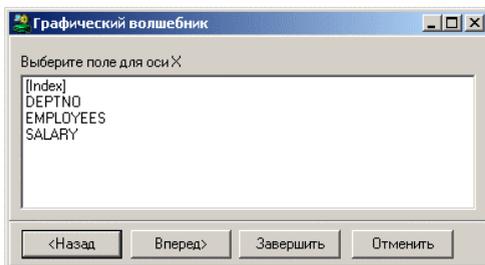


Что бы увидеть график зарплат департамента, вы можете нажать кнопку *График*:

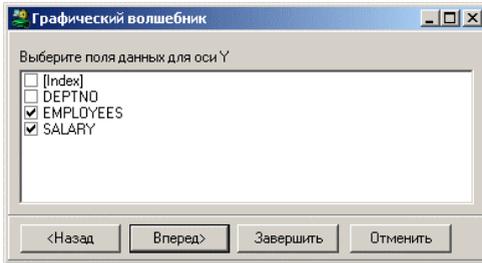


Такая же кнопка доступна и в Окне отчета. Она запустит либо *Волшебник графика* как описано ниже, либо вы можете выбрать предварительно определенный стиль графика или шаблон, и соответствующие колонки будут выбраны для осей X и Y.

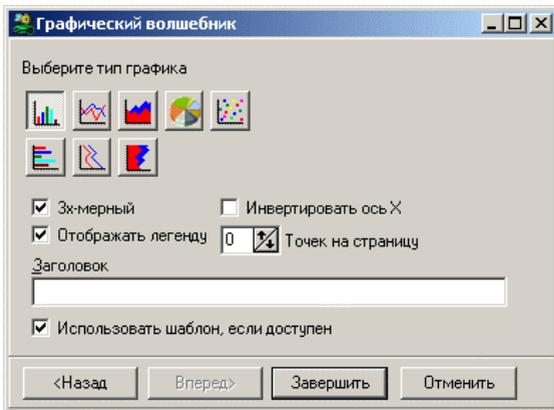
Волшебник графика позволит вам сначала выбрать данные для оси X:



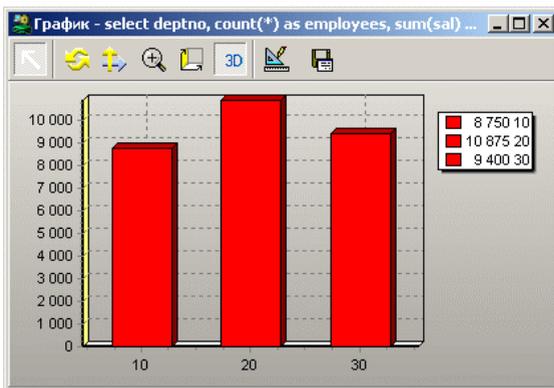
Это отобразит все колонки и элемент *[Index]*, который полезен, если у вас нет значения для оси X, но вы хотите отобразить данные на оси Y в порядке их появления. В этом случае мы хотим отобразить зарплату в зависимости от номера департамента. Поэтому мы выбираем DEPTNO и нажимаем кнопку *Дальше*, что бы выбрать данные для оси Y:



Выбрав *SALARY* для оси Y, вы можете нажать кнопку *Далее*, что бы выбрать свойства различных графических стилей:



Нажав *Завершить*, будет отображено результирующее Окно графика:



Панель инструментов Окна графика предоставляет следующие функции:

- Вращать – Нажмите левую кнопку мыши и двигайте мышью, что бы вращать график вокруг осей X и Y.

- Двигать - Нажмите левую кнопку мыши и двигайте мышью, что бы двигать график в окне. Эта функция всегда доступна, если нажать правую кнопку мыши. Независимо от того, какая функции панели инструментов активна в данный момент.
- Увеличение - Нажмите левую кнопку мыши и двигайте мышью, что бы увеличить или уменьшить график.
- Глубина - Нажмите левую кнопку мыши и двигайте мышью, что бы увеличить или уменьшить глубину графика.
- 3D – Переключает график между 2х и 3х мерным отображением.
- Редактировать – Редактировать данные и макет графика.
- Сохранить как шаблон – Сохраняет макет графика как шаблон для будущих графиков, использующих те же самые имена колонок. Если вы создадите график позже и выберете *Из шаблона* из всплывающего меню кнопки График, тогда макет из этого шаблона будет использован.

Вы можете дополнительно напечатать график или скопировать его в буфер обмена, нажав соответствующие кнопки на главной панели инструментов.

12. Проекты

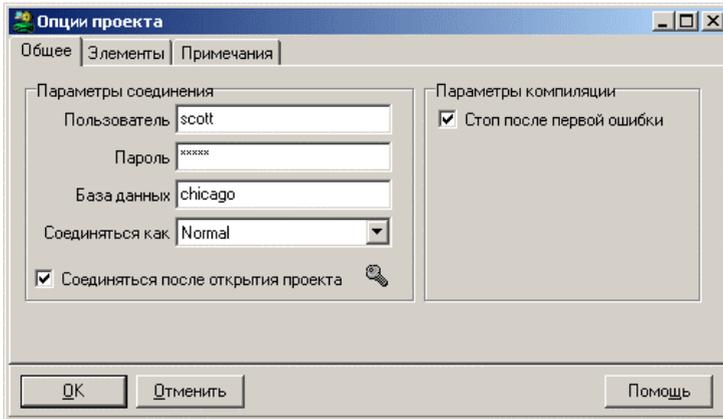
Чтобы организовать свою работу вы можете использовать встроенную концепцию проектов PL/SQL Developer. Проект состоит из коллекции исходных файлов, объектов базы данных, примечаний и опций. Это позволит вам работать в рамках специфического для задачи набора элементов, вместо базы данных или схемы целиком. Это делает простым поиск нужных вам элементов проекта, компиляцию всех элементов проекта или перемещение проекта из одного местонахождения или базы данных в другое местонахождение или другую базу данных.

Проект хранит конфигурацию вашего рабочего стола. Если вы повторно открываете проект, все элементы, которые были открыты, во время предыдущего закрытия проекта, будут открыты в том же положении. Для этого должно быть включено предпочтение *Автосохранение рабочего стола*. Эта информация будет храниться не в файле определения проекта (*Project.prg*), а в файле рабочего стола проекта (*Project.dsk*).

Примечание: для работы с проектами должно быть включено предпочтение *Использовать проекты*. В противном случае, главное меню не будет содержать пункта *Проект*. Для всех пунктов меню *Project*, также существуют соответствующие кнопки на панели инструментов, которые могут быть добавлены на панель инструментов при помощи страницы предпочтений *Панель инструментов* (смотрите главу 16.10).

12.1 Создание нового проекта

Чтобы создать новый проект, выберите пункт *Создать* из меню *Проект*. При этом появится диалог Опции проекта:



На странице *Общие* вы можете определить, как вы хотите подсоединяться, когда открыт этот проект. Вы также можете определить, хотите ли вы продолжать компиляцию, после того как встретилась ошибка компиляции.

Страница *Элементы* позволит вам просматривать элемента проекта. Это могут быть исходные файлы программных модулей PL/SQL, Тестовые скрипты, Отчёты и так далее. Элементом проекта также может быть объект, который хранится в базе данных, и который не имеет исходного файла. Элементом проекта так же может быть файл, такой как MS Word документ или HTML документ и т.д. Когда проект только создан, список элементов будет пустым.

Страница *Примечания* позволит вам вести примечания проекта. Эти примечания могут содержать любые данные, такие как план работ, проектные решения, дизайн и так далее.

12.2 Сохранение проекта

После установки опций, вы можете сохранить проект, выбрав в меню *Проект* пункт *Сохранить*. При этом будет создан файл проекта с расширением .prj. Файл проекта содержит спецификации элементов, опции и примечания. Поэтому вам нужно сохранять проект всякий раз, когда вы добавляете или удаляете элемент, изменяете опцию или изменяете примечания.

При сохранении проекта исходные файлы, включённые в проект, не будут сохранены.

12.3 Добавление в проект файлов

Есть два способа, что бы добавить файл в ваш проект:

Если файл уже открыт в IDE, вы легко можете выбрать в меню *Проект* пункт *Добавить в проект*. Для программных файлов (которые содержат исходный текст программных модулей), пункт неявно включает компиляцию. Это означает, что когда вы делаете или строите проект, этот файл будет включён в процесс компиляции. Для других файлов, таких как Тестовые скрипты, Скрипты SQL, Файлы отчётов и так далее компиляция включена не будет.

Если файл не открыт в IDE, вы можете нажать кнопку *Добавить файл(ы)* на панели инструментов проекта и выбрать файлы для добавления.

Заметьте, что путь файла по возможности хранится относительно местоположения файла проекта. Например, если файл находится в том каталоге, что и файл проекта, то путь сохранён не будет. Если файл хранится в подкаталоге файла проекта, будет сохранён только подкаталог. Если файл находится в каком-либо другом месте, то путь будет сохранён полностью. Это делает проект легко «переносимым» из одного каталога в другой.

12.4 Добавление объектов базы данных в проект

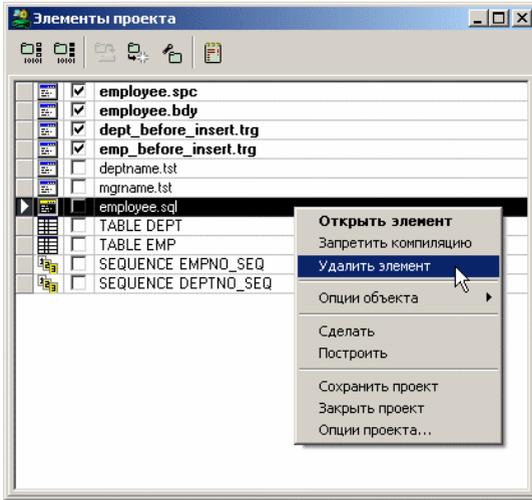
Чтобы добавить объект базы данных в проект, щёлкните правой кнопкой мыши в Объектном навигаторе и выберите во всплывающем меню элемент *Добавить в проект*. Точно так же, как и с Программными файлами, для программного модуля PL/SQL будет включена компиляция. Другие объекты, такие как таблицы и последовательности не могут быть скомпилированы.

Альтернативно, вы можете перетащить и бросить объект базы данных из Объектного навигатора в окно «Элементы проекта» (смотрите ниже).

Если текущий пользователь владеет объектом базы данных, то владелец не будет сохранён в файле проекта. Если другой пользователь владеет объектом, то владелец будет сохранён как часть спецификации объекта. Это делает проект переносимым от одного пользователя к другому.

12.5 Работа с элементами проекта

После добавления элементов в проект, для работы с ними вы можете выбрать в меню *Проект* пункт *Элементы проекта*. Окно «Элементы проекта» может выглядеть вот так:



Компиляция включена для первых 4 Программных файлов. Для 2 Тестовых скриптов, скрипта SQL, таблицы и последовательности компиляция не включена.

Вы можете держать окно «Элементы проекта» постоянно открытым, чтобы иметь быстрый и лёгкий доступ ко всем элементам проекта. Это окно является частью рабочего стола проекта, и будет заново открыто, когда заново будет открыт проект.

Чтобы редактировать элемент, вы можете просто дважды щёлкнуть на нём. Если элемент является исходным файлом, то он будет открыт в соответствующем окне (Программное окно, Тестовое окно и так далее). Если же элемент является объектом базы данных, то будет открыт соответствующий редактор с определением объекта. Если элемент является внешним файлом, то соответствующее приложение будет запущено (например, Acrobat Reader для PDF файла).

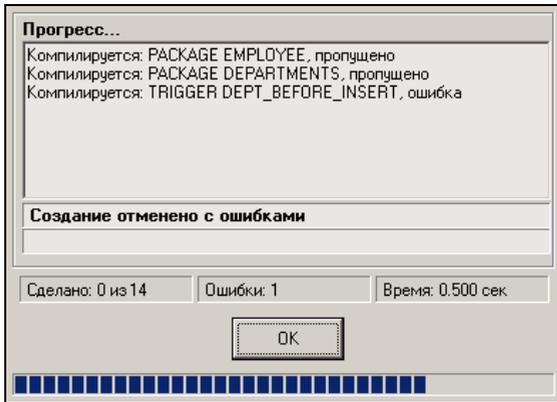
Альтернативно, вы можете щёлкнуть правой кнопкой мыши на элементе, чтобы вызывать всплывающее меню, как показано выше. Здесь вы можете запретить/разрешить компиляцию для элемента, или же вы можете удалить элемент из проекта. Если элемент является объектом базы данных, у вас также будет доступ к всплывающему объектному меню, как подменю.

Чтобы изменить порядок элементов проекта, вы можете щёлкнуть на заголовке строки и перетащить его на новое местоположение. Порядок важен не только визуально, но он также определяет порядок компиляции.

12.6 Компиляция проекта

Чтобы скомпилировать проект, вы можете выбрать в меню *Проект* пункты *Построить* или *Сделать*.

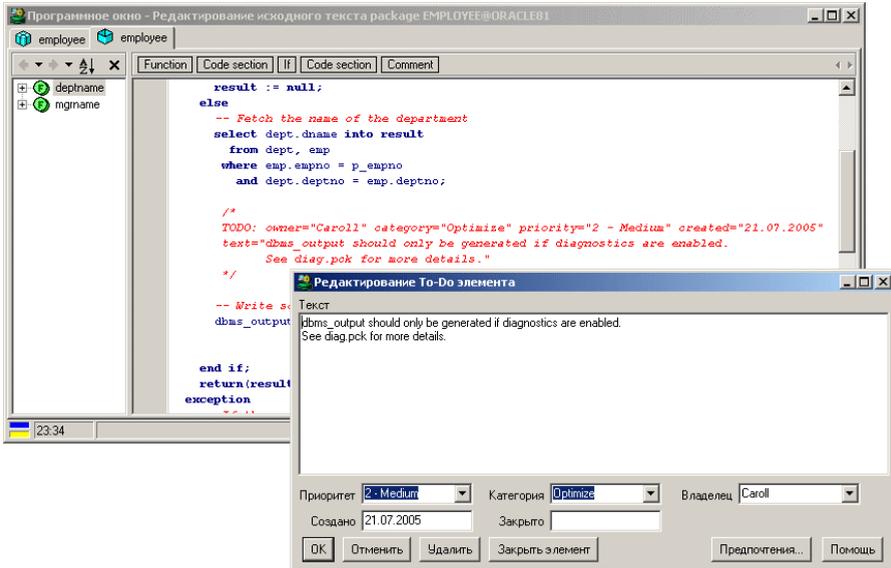
Функция *Построить* просто скомпилирует все элементы проекта, для которых разрешена компиляция. Элементы будут скомпилированы в том порядке, в котором они появляются в списке элементов проекта. Если компиляция элемента заканчивается ошибкой, опция проекта *Стоп после первой ошибки* определяет, будет ли продолжена компиляция проекта или нет:



Функция *Сделать* скомпилирует только те элементы, которые были изменены со времени предыдущей компиляции. Если Программный модуль PL/SQL включён как объект базы данных, и является неактуальным, он также будет перекомпилирован. Временная метка компиляции хранится в файле рабочего стола проекта. Элементы, для которых разрешена компиляция, но являются неактуальными на данный момент времени, в окне процесса компиляции будут перечислены как *Пропущенные*.

13. To-Do элементы

Вы можете использовать To-Do элементы в любом исходном файле, для того чтобы сделать быструю заметку, что необходимо что-либо сделать в этом исходном файле, чтобы позже обратиться к этой информации при помощи To-Do списка. Эти To-Do элементы помещены в исходный файл как комментарий в специфическом формате, в соответствующем местоположении:

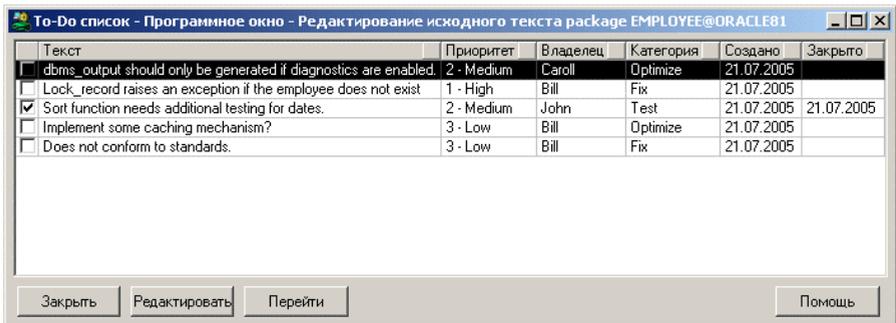


В этом примере вы можете видеть комментарий в исходном файле (начинается с текста *TODO:*) и Редактор To-Do элемента, где вы можете ввести текст, приоритет, категорию владельца, дату создания и дату закрытия. Содержимое списков полей «приоритет», «категория» и «владелец» могут быть определены при помощи предпочтений (смотрите главу 16.24).

К To-Do элементам можно обратиться при помощи редактора To-Do элемента, в котором очень удобно создавать, редактировать, просматривать или удалять To-Do элемент. Вы также можете редактировать комментарий прямо в исходном файле, но не изменяя формат.

Так как To-Do элементы хранятся в исходном файле как комментарии, то они всегда будут соответствовать его текущему состоянию. Независимо от того, откуда взят исходный файл: из базы данных, из файловой системы или является старой версией из системы контроля версий.

Чтобы обратиться ко всем To-Do элементам конкретного исходного файла, вы можете вызвать To-Do список:



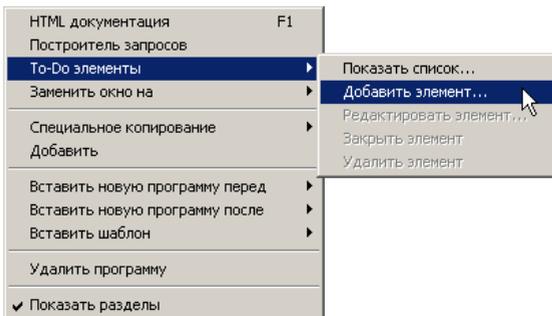
Здесь вы можете быстро найти To-Do элемент и поместить его в исходный файл, дважды щёлкнув на нём или нажав кнопку *Перейти*.

To-Do список можно отсортировать, щёлкнув мышью по заголовку столбца.

Нажмите правую кнопку мыши на списке, что бы скопировать, напечатать, или экспортировать To-Do элементы.

13.1 Создание To-Do элемента

Чтобы создать To-Do элемент, вам сначала нужно поместить курсор на подходящем месте в исходном файле. Сюда будет вставлен комментарий, поэтому очень важно, чтобы это было рядом с тем местом в исходном файле, с которым нужно будет работать. Дальше, вы можете щёлкнуть правой кнопкой мыши на этом местоположении и выбрать в подменю *To-Do элементы* всплывающего меню команду *Добавить элемент*:



Появится упомянутый выше редактор To-Do элемента, и вы сможете ввести информацию для нового To-Do элемента. После нажатия кнопки *OK* комментарий будет вставлен в то место, где находится курсор.

13.2 Редактирование To-Do элемента

Чтобы отредактировать существующий To-Do элемент, вы можете либо щёлкнуть правой кнопкой мыши на соответствующем комментарии в исходном файле и выбрать в подменю *To-Do элементы* команду *Редактировать элемент* или выбрать пункт *Показать список*, если вы не находитесь в непосредственной близости от Элемента рабочего плана, который вы хотите

редактировать. В этом случае, вы можете выбрать To-Do элемент из списка и нажать кнопку *Редактировать*.

Теперь вы можете изменять различные свойства To-Do элемента и нажать кнопку ОК, чтобы обновить комментарий в исходном файле.

13.3 Заккрытие To-Do элемента

To-Do элемент можно закрыть, щёлкнув правой кнопкой мыши на комментарии в исходном файле и выбрав в подменю *To-Do элементы* команду *Заккрыть элемент*. Альтернативно вы можете вызвать редактор To-Do элемента и нажать кнопку *Заккрыть*.

13.4 Удаление To-Do элемента

To-Do элемент можно удалить, щёлкнув правой кнопкой мыши на комментарии в исходном файле и выбрав в подменю *To-Do элементы* команду *Удалить элемент*. Альтернативно, вы можете вызвать редактор To-Do элемента и нажать кнопку *Удалить*.

Вы также можете просто удалить комментарий из исходного файла.

14. Окна, сессии базы данных и транзакции

PL/SQL Developer предоставляет многосессионную / многопоточковую среду.

Многосессионность означает, что хотя вы только один раз зарегистрировались в PL/SQL Developer, отдельные сессии базы данных могут быть использованы для Тестовых окон, Окон SQL и Командных окон. Кроме того, компиляции программных модулей будут происходить в отдельных сессиях базы данных. К тому же, компиляция является оператором DDL и неявно фиксирует текущую транзакцию. Это может привести к нежелательным эффектам, если бы компиляция происходила в общей с другими окнами сессии.

Многопоточность означает, что выполнение Тестовых окон, Окон SQL и Командных окон может происходить одновременно, без блокировки друг друга. Преимущество этого заключается в том, что вы можете продолжать делать другую работу, пока выполняются массивные операторы SQL. А так же вы можете прервать или удалить сессию, можете протестировать блокирующее поведение ваших программных модулей и так далее.

14.1 Режим сессий

Многопоточность подразумевает, что вам нужна много сессионная среда. А так же Oracle Server может выполнять только один оператор в каждый момент времени в одной сессии базы данных. Таким образом, операторы могут быть запущены одновременно, если они запускаются в разных сессиях. Не всегда условия разработки позволяют использовать неограниченное количество сессий, так что может быть установлено предпочтение (смотрите главу 16.2), которое контролирует количество сессий, используемых PL/SQL Developer. Это предпочтение называется «Режим сессий» и может принимать одно из следующих значений:

- Много сессий – Каждое Тестовое окно, SQL окно и Командное окно будет иметь свою собственную сессию, а для компиляций будет использоваться другая сессия. Это наиболее гибкий режим, но он ведёт к самому большому количеству сессий базы данных. Другой возможный недостаток заключается в том, что обновления сделанные в Окне X будут видны в Окне Y, только после того, как они будут зафиксированы.
- Двойная сессия – Тестовые окна, SQL окна и Командные окна будут использовать одну и ту же сессию, и для компиляций будет использоваться другая сессия. Недостаток этого режима, заключается в том, что нельзя выполнять несколько окон одновременно.
- Одиночная сессия – Все окна и все компиляции используют одну и ту же сессию, делая сложным управление транзакцией. В этом режиме отладчик отключён. Используйте этот режим, только если вы ограничены использованием одной сессии базы данных.

14.2 Выполнение в режимах Много сессий или Две сессии

Когда вы выполняете Тестовое окно, SQL окно или Командное окно в режимах Много сессий или Две сессии, на панели инструментов будет включена кнопка *Прервать*, чтобы вы могли в любой момент прервать выполнение. Прерывание не всегда может быть удачным, например, в случае блокирующей ситуации. В этом случае, вы можете, если у вас есть такая привилегия, использовать пункт *Уничтожить* в меню Сессия.

Если вы передвинете курсор мыши к выполняемому в данный момент окну, его вид изменится, для того, чтобы показать, что оно выполняется, и вы не можете вносить никаких изменений в содержимое этого окна.

После того, как выполнение закончено, и, если начата транзакция, будут включены кнопки *Зафиксировать* и *Откатить*. В режиме Много сессий, у каждого окна будет своя собственная сессия, и следовательно, своя транзакция. Кнопки *Зафиксировать* и *Откатить* будут отражать статус транзакции самого верхнего окна.

15. Просмотр объектов

Когда вы разрабатываете программный модуль для базы данных Oracle, он всегда будет обращаться к другим объектам. Если только это не будет случай, когда у программного модуля нет права вообще находиться в этой базе данных. Чтобы написать такой программный модуль, вам нужно точно знать, как определены объекты базы данных. Так что вопросы, приведённые ниже, будут очень знакомы программисту Oracle:

- Какой тип данных у этого столбца?
- Каков первичный ключ этой таблицы?
- Какое в действительности имя у этой пакетной функции?
- Какое значение по умолчанию у этого параметра?
- Какие условия вызывают срабатывание этого триггера?
- Какие объекты вызывают функцию, над которой я работаю?
- И так далее...

Вся эта информация представлена в словаре Oracle и доступна в PL/SQL Developer при использовании Навигатора.

15.1 Использование Навигатора

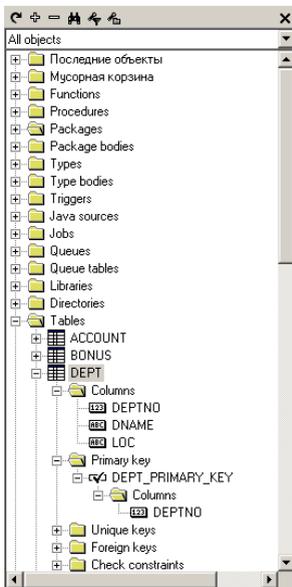
Вы можете вызвать Навигатор, выбрав в меню *Инструменты* пункт *Навигатор*. Навигатор показывает все основные типы объектов базы данных, которые могут быть использованы в разработке программных модулей. Вы можете перемещаться по этому дереву так же, как вы используете Проводник в Windows для навигации в папках. При открытии папки *Таблицы*, будут отображены все таблицы, доступные для текущей сессии. При открытии папки отдельной

таблицы, будут отображены все свойства, относящиеся к этой таблице. При открытии папки *Внешние ключи*, будут отображены все внешние ключи для этой таблицы и так далее. Папка *Последние объекты* содержит все объекты, которые вы использовали в PL/SQL Developer в последнее время.

Если вы когда-либо открывали папку, то информация, запрошенная из словаря, будет сохранена в памяти, чтобы сделать последующий доступ как можно более быстрым. Чтобы Навигатор запросил заново эту информацию, после того, как что-нибудь изменилось, вам нужно выбрать изменённый объект и нажать в левой верхней части окна кнопку *Обновить данные*. Вы также можете обновить объект, выбрав во всплывающем меню соответствующий пункт. Чтобы обновить весь Навигатор полностью, удерживайте клавишу *Ctrl*, когда нажимаете кнопку *Обновить данные*.

Вы можете перетаскивать и бросать объекты из Объектного навигатора в рабочее пространство. При этом появится редактор с определением объекта. Вы можете установить предпочтение, чтобы контролировать, какую конкретно функцию следует выполнить (смотрите главу 16.11).

Вы также можете перетащить и бросить объект из Объектного навигатора в редактор. При этом появится всплывающее меню, содержимое которого зависит от объекта. Вы можете



скопировать имя (если необходимо, с указанием владельца), свойства, описание и другой нужный текст в редактор, в нужное вам место. Если вы перетаскиваете и бросаете папку в

редактор, все имена объектов в папке будут скопированы. Эти имена объектов будут разделены запятыми. Вы можете установить предпочтение, чтобы контролировать, как будут размещены имена объектов в одну строку или во множество строк (смотрите главу 16.12).

При нажатии правой кнопки мыши на значке объекта, появится чувствительное к контексту всплывающее меню, которое позволит вам применять к этому объекту другие функции. Функции из этого всплывающего меню будут объяснены в следующих параграфах.

Вы можете выбрать несколько объектов, удерживая клавиши *Control* или *Shift* в течении выбора. Если выбрано несколько объектов, то всплывающее объектное меню покажет только те элементы, которые применимы ко всем выбранным объектам, и функция перетаскивания и помещения будет работать для всех объектов одновременно.

Вы можете нажать *Ctrl* - и *Ctrl* +, что бы быстро выровнять ширину навигатора по его содержанию.

Предпочтения Навигатора (смотрите главу 16.11) позволят вам определять, какие папки будут видимыми, и в каком порядке они будут отображены. Они также позволят вам определять максимальное количество объектов, которые могут быть видимыми в папке *Последние объекты*. Кроме того, вы можете определять действие двойного щелчка на объекте навигатора. По умолчанию, при двойном щелчке будет развёрнут соответствующий узел, но вы также можете определить, чтобы при двойном щелчке появлялся редактор объектного определения или окно свойств.

Создание нового объекта

Чтобы создать новый программный модуль, таблицу, вид, последовательность или синоним, щёлкните правой кнопкой мыши на подобном объекте или на корневой папке этого типа объекта и выберите во всплывающем меню элемент *Создать*.

При использовании функции *Создать* для программного модуля, будет создано новое Программное окно. Первоначальное содержимое будет взято из специального шаблона, как это описано в главе 22.

Представления могут быть созданы в SQL окне. Первоначальное содержимое которого, также как и для программных модулей, будет взято из специального шаблона.

Таблицы, последовательности и синонимы будут отображены в специальном окне редактора. Редакторы таблиц, последовательностей и синонимов описаны в главе 8.

Дублирование объекта

Вы можете дублировать таблицу, последовательность или синоним, выбрав во всплывающем меню элемент *Дублировать*. При этом будет вызвано окно редактора со всеми свойствами выбранного объекта, за исключением имени. Это может быть полезным, если вы хотите создать такую же таблицу для тестирования.

Копирование объекта или папки

Чтобы копировать объект или имена объектов в папку, выберите пункт *Копировать разделенное запятыми*. При этом имя (имена) будут помещены в буфер обмена. Вы можете установить предпочтение, чтобы контролировать, будут ли имена объектов размещены в одну строку или в несколько строк (смотрите главу 16.12).

Просмотр свойств объекта

Чтобы просмотреть свойства конкретного объекта, выберите во всплывающем меню элемент *Свойства*. Текущие свойства, которые вы видите, зависят от типа объекта и версии вашего Oracle Server.

Свойство	Значение
OWNER	SCOTT
CONSTRAINT_NAME	DEPT_FOREING_KEY
CONSTRAINT_TYPE	R
TABLE_NAME	DEPT
R_OWNER	SCOTT
R_CONSTRAINT_NAME	DEPT_PRIMARY_KEY
DELETE_RULE	NO ACTION
STATUS	ENABLED
DEFERRABLE	NOT DEFERRABLE
DEFERRED	IMMEDIATE
VALIDATED	VALIDATED
GENERATED	USER NAME
LAST_CHANGE	21.07.2005 13:25:21

Свойства могут быть отображены в модальном окне, которое вы можете разместить в любом месте экрана. Вы можете оставить окно свойств видимым (оно будет оставаться поверх других окон) и продолжить редактирование в другом окне. Вы можете дополнительно разворачивать и сворачивать окно, нажимая кнопку в правом верхнем углу окна. Подобным образом, вы можете легко создать пространство для других вещей и оставить свойства доступными.

Просмотр описания объекта

Таблицы, виды, функции, процедуры, пакеты, типы и методы могут быть «описаны» способом, схожим с SQL*Plus, для этого выберите во всплывающем меню пункт *Describe*. Описание будет отображено в модальном окне, как упомянутое ранее окно свойств.

Для таблиц и представлений все столбцы и их соответствующие свойства отображаются в окне свойств. Для функций, процедур и методов отображаются параметры и их свойства. Для пакетов функции, процедуры, типы, переменные, константы и исключительные ситуации отображаются. Для типов методы и атрибуты отображаются.

Имя	Тип	Обнуляемая	По умолчанию	Комментарии
EMPNO	NUMBER(4)			The 4 digit number of the employee
ENAME	VARCHAR2(12)	Y		The last name of the employee
JOB	VARCHAR2(10)	Y	'CLERK'	The job of the employee
MGR	NUMBER(4)	Y		The number of the manager
HIREDATE	DATE	Y	trunc(sysdate)	The date on which the employee was hired
SAL	NUMBER(7,2)	Y		The salary of the employee
COMM	NUMBER(7,2)	Y		The commission of the employee
DEPTNO	NUMBER(2)	Y		The 2 digit number of the department

Вы можете выбрать строки, столбцы или диапазон ячеек в окне описания. Чтобы выбрать строки или столбцы, просто щёлкните на заголовке строки или столбца и перетащите указатель мыши, чтобы произвести выделение. Чтобы выбрать определённый диапазон ячеек, передвиньте указатель мыши к левому краю ячейки, чтобы форма курсора изменилась, нажмите кнопку мыши и тащите ее, чтобы произвести выбор. Выбранный массив ячеек может быть скопирован или напечатан. Чтобы создать копию разделённых запятыми выбранных ячеек, щелкните правой кнопкой мыши на окне и выберите во всплывающем меню соответствующий пункт. Чтобы экспортировать описание в CSV, TSV, HTML или XML-файл, выберите соответствующий элемент в подменю *Экспортировать*. Чтобы напечатать содержимое окна описания, выберите элемент *Печатать*.

При двойном щелчке на ячейке в окне описания, содержимое ячейки будет скопировано в редактор самого верхнего окна.

Чтобы отсортировать строки в окне описания, нажмите кнопку в заголовке столбца, в котором вы хотите отсортировать строки.

Просмотр определения объекта

Все типы объектов имеют некоторое определение. Чтобы просмотреть это определение, вы можете выбрать во всплывающем меню пункт *Показать*. Этот элемент доступен для всех программных модулей, представлений, таблиц, последовательностей и синонимов. Если вы выберете пункт *Показать* для элемента таблицы, такого как столбец или индекс, то будет отображено определение таблицы и выбран соответствующий элемент.

Программные модули будут отображены в Программном окне, представления будут отображены в Окне SQL, а таблицы, последовательности и синонимы будут отображены в специальном окне редактора. Редакторы таблицы, последовательностей и синонимов описаны в главе 8.

Редактирование определения объекта

Вы можете редактировать определение объекта, выбрав во всплывающем меню элемент *Редактировать*. Вы можете отключить эту функцию, при помощи предпочтения, как описано в главе 16.2.

Переименование объекта

Таблицы, виды, последовательности и синонимы можно переименовать, выбрав во всплывающем меню пункт *Переименовать*. Если вы используете Oracle 9.2 или более позднюю версию, вы также можете переименовывать столбцы. Появится диалог со старым именем, которое вы можете изменить и применить изменения.

Удаление объекта

Иногда бывает необходимо удалить объект. Он может стать не нужным, или нужно изменить его тип с одного на другой (например, с функции на процедуру). Чтобы выполнить это, просто выберите во всплывающем меню пункт *Удалить*.

Просмотр связанного объекта

У большинства объектов есть объекты, на которые либо они ссылаются, либо данные объекты ссылаются на них. Выбрав элемент *Browse*, вы можете найти связанный объект в Навигаторе.

Перекомпилирование объекта

Объекты часто ссылаются на другие объекты в базе данных. Когда изменяется объект, на который ссылаются, ссылающийся объект может стать невалидным. В этом случае, в значке объекта появляется красная метка. Чтобы перекомпилировать невалидный объект, выберите во всплывающем меню пункт *Перекомпилировать*. Если компиляция проходит успешно, то метка пропадает. Если происходит сбой компиляции, вы можете просмотреть ошибки, выбрав во всплывающем меню элемент *Просмотреть*.

Добавление исходного текста объекта в редактор

Исходный файл программного модуля можно добавить в Программное окно или в Тестовое окно, выбрав элемент *Вставить исходный текст в редактор*.

В Программное окно исходный текст будет добавлен на новой странице после текущего программного модуля. Таким образом, вы можете создать Программный файл, который содержит несколько программных модулей, которые уже существуют в базе данных.

В Тестовое окно исходный текст будет добавлен в течение отладочной сессии после текущего выбранного исходного текста. Таким образом, вы можете включить исходный текст в Тестовое

окно и впоследствии установить или удалить точки разрыва, до того, как вы непосредственно перейдёте в программный модуль.

Добавление отладочной информации к объекту

Если вы хотите в течение отладочной сессии просмотреть или установить значения переменных, необходимо, чтобы объект, который содержит эти переменные, содержал и отладочную информацию. Вы можете вручную добавить отладочную информацию, выбрав пункт *Добавить отладочную информацию*. У этого пункта меню имеется пометка, показывающая содержит ли объект отладочную информацию или нет. В секции предпочтений отладчика имеется опция, позволяющая автоматически добавлять отладочную информацию, когда программный модуль компилируется, в этом случае вам только нужно использовать этот пункт для программных модулей, которые не находятся в разработке.

Перекомпиляция ссылающихся объектов

Когда вы изменяете объект, на который ссылаются другие программные модули, то эти ссылающиеся программные модули станут недействительными. Их перекомпиляция, которую легко можно произвести, выбрав элемент *Перекомпилировать ссылающиеся объекты*, либо исправит эти объекты, либо укажет на несовместимости. При этом появится окно Компиляция недействительных объектов, которое будет немедленно выполнено. Больше информации об этом инструменте вы можете найти в главе 17.3.

Тестирование программного модуля

Чтобы протестировать программный модуль (отдельно стоящую функцию или процедуру, пакетную функцию или процедуру или метод объектного типа), вы можете использовать Тестовые скрипты, которые детально объяснены в главе 4. В общем случае Тестовый скрипт будет содержать простой вызов функции или процедуры. Вам нужно установить входные параметры и просмотреть выходные параметры и результаты функции после выполнения Тестового скрипта. Чтобы создать подобный Тестовый скрипт, вы можете выбрать в Навигаторе элемент *Тестировать*.

Выбор данных из таблиц и представлений

Чтобы просмотреть данные таблицы или представления, выберите во всплывающем меню пункт *Запросить данные*. После этого будет создано SQL окно с SELECT командой возвращающей данные таблицы или представления, выбранных в навигаторе. Если у вас включено предпочтение SQL окна *АвтоИсполнение запросов*, тогда оператор SELECT будет выполнен немедленно.

Редактирование данных таблицы

Если вы хотите отредактировать данные в таблице, то, когда таблица выбрана в Навигаторе, вы можете выбрать пункт *Редактировать данные*. Будет создано окно SQL с оператором SELECT, который включает в себя rowid. Если у вас включено предпочтение SQL окна *АвтоИсполнение запросов*, тогда оператор SELECT будет выполнен немедленно. После того, как записи будут возвращены, для SQL окна немедленно будет включен режим редактирования.

Экспорт таблиц

Чтобы экспортировать определение и данные таблицы, выберите элемент *Экспортировать данные*. При этом для выбранной таблицы появится инструмент *Экспорт таблиц*. За детальной информацией обратитесь к главе 17.4.

Включение и выключение триггеров и ограничений

Можно включать или отключать триггеры и ограничения в Объектном Навигаторе, выбирая во всплывающем меню элементы *Включить* или *Выключить*. Пиктограммы отключённых триггеров и ограничений отображаются светло серым цветом.

Запуск задания

Вы можете запустить задания, созданные через пакет *dbms_job*, выбрав во всплывающем меню пункт *Выполнить*.

Включение и отключение постановки в очередь и исключения из очереди

Для очередей вы можете включать и отключать операции постановки в очередь и исключения из очереди, выбирая соответствующий пункт во всплывающем меню. Пометка перед элементом меню показывает, включена ли в данный момент операция.

Добавление объекта в папку, определённую пользователем

Чтобы добавить объект в папку, определённую пользователем (смотрите главу 15.3), выберите элемент *Добавить в папку*, потом выберите в подменю папку.

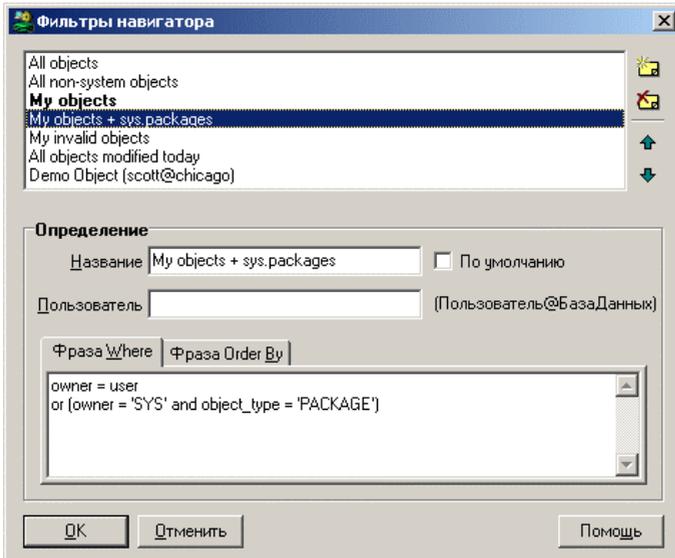
Восстановление или очистка объекта

Если вы используете Oracle10g или более позднюю версию, и если у вас включена соответствующая функция в базе данных, то вы можете восстанавливать или очищать таблицы которые были ранее удалены. Для объекта в *Мусорной корзине* вы можете выбрать пункт *Очистить*, что бы окончательно удалить таблицу и ее данные. Или вы можете выбрать пункт *Восстановить*, что бы восстановить таблицу под ее исходным именем или под новым именем.

15.2 Фильтры Навигатора

Вы можете определить фильтры, которые будут определять, какие из объектов в системных папках Объектного Навигатора будут видимыми, и в каком порядке эти объекты будут отображены. В верхней части Навигатора находится список, который позволит вам переключаться между различными фильтрами

Фильтры Навигатора можно определить, выбрав в меню Инструменты пункт *Фильтры навигатора* или нажав над Навигатором кнопку *Фильтры*. В этом диалоге вы можете изменять, создавать и удалять фильтры:



Фильтр определяется следующими свойствами:

- Описание.
Это описание отображается в списке в верхней части Навигатора.
- Пользователь.
Определяет, для кого из пользователей, этот фильтр применим. Если вы оставите это поле пустым, то этот фильтр будет доступен для всех пользователей Oracle. Если вы указываете пользователя и базу данных (например, scott@chicago), то этот фильтр будет доступен, только если вы зарегистрированы как этот пользователь.
- По умолчанию.
Если вы включите эту опцию, выбранный фильтр будет активирован при каждом вашем запуске PL/SQL Developer. Вы можете определить фильтр по умолчанию для каждого пользователя. Фильтры, определённые по умолчанию, выделены полужирным шрифтом.
- Фраза Where.
Фильтр определяет оператор SELECT для представлений *all_objects* (или *dba_objects*). Чтобы ограничить результирующий набор этого оператора SELECT, может быть использовано предложение where. Вы можете использовать следующие столбцы:
 - owner – владелец объекта

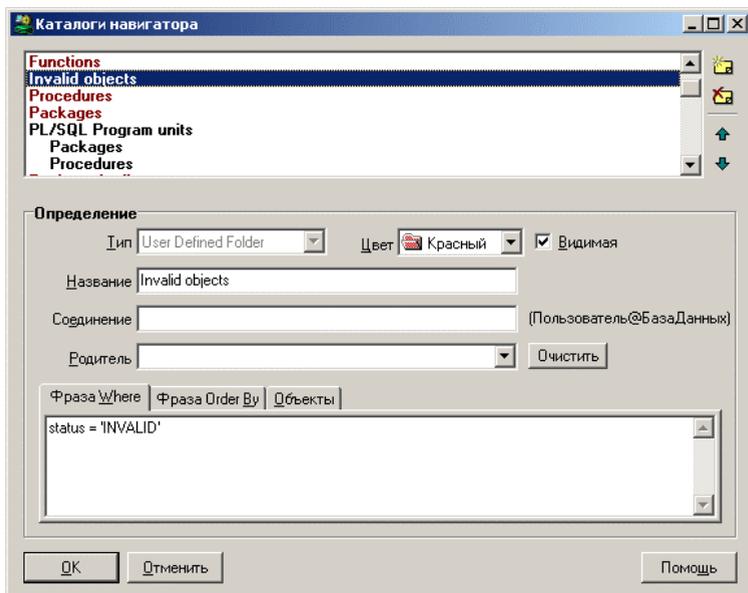
- `object_name` – имя объекта
 - `object_type` – тип объекта (TABLE, VIEW, PACKAGE и так далее)
 - `status` – статус объекта (VALID или INVALID)
 - `created` – дата/время, когда был создан объект
 - `last_ddl_time` – дата/время, когда объект в последний раз был скомпилирован
- Фраза `Order by`.
Чтобы упорядочить объекты в Навигаторе, используйте столбцы из вида `all_objects`.

15.3 Папки Навигатора

Папки Объектного Навигатора можно настроить несколькими способами. Вы можете изменить порядок папок в навигаторе, цвет каждой папки, иерархию папок, также вы можете скрыть специфические папки, основываясь на вашем текущем соединении.

Вы также можете определить ваши собственные папки и заполнить их объектами. Вы можете либо явно добавить специфические объекты в папку, либо определить запрос, используемый для её заполнения, либо воспользоваться двумя этими способами одновременно. Когда папка будет открыта в Объектном Навигаторе, запрос будет выполнен.

Можно определить папки Навигатора, выбрав в меню Инструменты пункт *Папки навигатора*. В этом диалоге вы можете изменять, создавать и удалять папки:



В верхней части этого диалога вы можете видеть все папки и их иерархию. Системные папки обозначаются красным цветом, а папки определённые пользователем – чёрным. К системным папкам применяются некоторые ограничения.

Справа от списка папок у вас имеется 4 кнопки, позволяющие добавлять новую папку на место текущей выбранной папки, удалять папку или перемещать её вверх или вниз по списку. Порядок в Объектном Навигаторе такой же, как и в этом списке.

В нижней части вы можете изменять определение текущей выбранной папки:

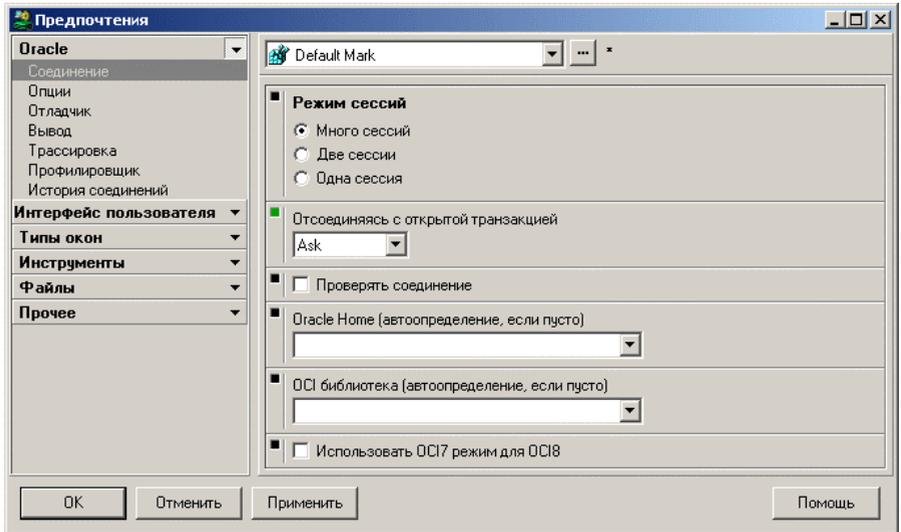
- **Type** – Показывает тип папки: Определённая пользователем или Системная
- **Color** – Выберите цвет папки. Подпапки объектов унаследуют тот же цвет.
- **Visible** – Отключите эту опцию, чтобы сделать папку невидимой в Объектном Навигаторе. Эта опция может быть использована, чтобы скрыть системные папки, и отменяет опцию Connection.
- **Description** – Описание папки в Объектном Навигаторе. Нельзя изменить для системных папок.
- **Connection** – Определяет, для кого из пользователей Oracle будет отображена эта папка. Если вы оставите эту опцию пустой, папка будет отображаться для всех пользователей. Если вы укажете пользователя и базу данных (например, `scott@chicago`), папка будет видима только в случае, если вы зарегистрированы как этот пользователь. Вы можете использовать подстановочные символы `?` и `*`, чтобы выделять нескольких пользователей. Например, `*@prgd`, для всех пользователей в базе данных `prgd`.
- **Parent** – Чтобы иерархически разместить одну папку внутри другой, вы можете выбрать родительскую папку из списка. Чтобы очистить родительскую папку, нажмите кнопку Очистить.
- **Where clause** – Для папок основанных на запросах может быть использована фраза `where`, чтобы ограничить выбор объектов. В приведённом выше примере, в качестве фразы `where` указано `STATUS = 'INVALID'`, чтобы заполнить содержимое папки всеми невалидными объектами.
- **Order By clause** – Для папок основанных на запросах может быть использована фраза `order by`, чтобы определить порядок сортировки выбранных объектов.
- **Objects** – Для папок заполняемых вручную, вы можете определить список объектов. Вы можете разместить объект на каждой строке в формате `owner.name`. Например: `SCOTT.EMP`. Заметьте, что часто более удобным бывает добавлять отдельные объекты Объектного Навигатора, щёлкнув на нём правой кнопкой мыши и выбрав во всплывающем меню Добавить в папку.

Фразы `where` и `order by` могут включать в себя любое SQL выражение и любой столбец представления *all objects*. Это работает так же как и в случае фильтров навигатора.

Заметьте, что фильтры не влияют на содержимое папок, заполняемых вручную.

16. Предпочтения

Инструмент *Предпочтения* позволяет вам установить для PL/SQL Developer различные предпочтения. Когда вы выберете этот пункт, появится следующий диалог:



В левой части окна вы можете выбрать страницу предпочтений - одну из 6 категорий (Oracle, User Interface, Window Types, Tools, Files и Other).

В верхней части окна вы можете выбрать набор предпочтений. По умолчанию вы определяете персональные предпочтения для вашего пользователя Windows (в экранном снимке сверху для пользователя *Mark*), но вы также можете определить персональные предпочтения для специфического соединения Oracle. Если вы когда-либо используете подобное соединения, чтобы зарегистрироваться в базе данных, то будет использован соответствующий набор предпочтений. В дальнейшем, ваш системный администратор, может определить предпочтения по умолчанию на глобальном уровне и на уровнях соединения Oracle. Для детальной информации о наборах предпочтений, смотрите главу 16.33.

Под наборами предпочтений вы можете видеть предпочтения для текущей выбранной страницы предпочтений (на экранном снимке выше - для страницы *Oracle – Соединение*). Для каждого предпочтения будет видно текущее значение. Зелёный индикатор слева от предпочтения показывает, что значение установлено в текущем наборе предпочтений. Если индикатор чёрного цвета – значение унаследовано из другого набора (поместите курсор мыши на индикатор, чтобы увидеть имя набора предпочтений). Чтобы вернуть предпочтение к его унаследованному значению (другими словами, удалить его из текущего набора предпочтений), вы можете щёлкнуть на зелёном индикаторе.

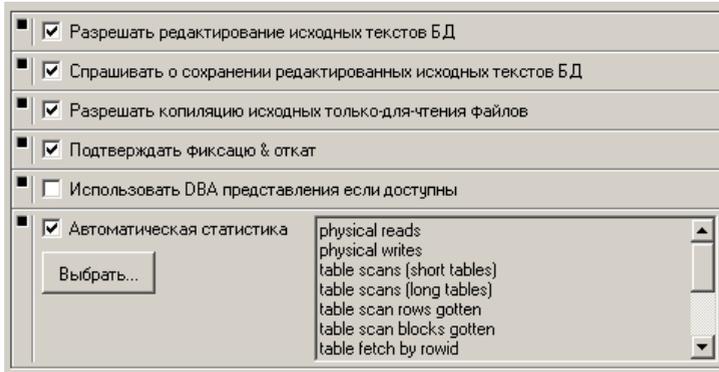
В следующих главах будет детально описана каждая страница предпочтений.

16.1 Oracle - Соединение

- **Режим сессий.**
Контролирует количество отдельных сессий базы данных, которые будет использовать PL/SQL Developer. Эта функция детально описана в главе 14.
- **Отсоединяясь с активной транзакцией.**
Когда сессия отсоединяется от системы и у этой сессии есть активная транзакция, то действие по умолчанию в этом случае – фиксация этой транзакции. При помощи этого предпочтения, вы сможете определить, что такая активная транзакция должна быть откатена, или что у вас должно быть запрошено подтверждение.
- **Проверять соединение.**
Когда эта опция включена, PL/SQL Developer будет каждые 60 секунд проверять, существует ли ваше соединение с базой данных. Если ваша сессия была уничтожена, или сервер был выключен, PL/SQL Developer автоматически отсоединится от системы.
- **Oracle Home.**
По умолчанию, PL/SQL Developer будет использовать на вашем ПК основной Oracle Home. Вы можете определить основной Oracle Home при помощи Oracle Home Selector (стандартная утилита Oracle). Вы можете указать PL/SQL Developer, использовать специфический Oracle Home, выбрав его из этого списка.
- **OCI библиотека.**
PL/SQL Developer будет использовать самые свежие из установленных версий SQL*Net или Net8. Если эта версия SQL*Net или Net8 вызывает проблемы, вы можете указать PL/SQL Developer использовать другую версию. Имя DLL будет выглядеть примерно так ora72.dll (SQL*Net 2.2), ora73.dll (SQL*Net 2.3), oci.dll (Net8 8.1.x) и так далее. Эти DLL хранятся в директории Oracle bin.
- **Использовать OCI7 режим для OCI8.**
Некоторые выпуски Net8 (8.0.3 и 8.0.4) содержат ошибки, которые могут вызвать проблемы в PL/SQL Developer. При установке этого предпочтения вместо функций Net8 будут использованы более корректные функции SQL*Net 2. Один из примеров ошибки Net8, при описании процедур и функций эта программа сообщает неверные параметрические режимы. Недостатком этого предпочтения является то, что вы не

можете запрашивать столбцы BLOB, CLOB and BFILE, и обернутые (wrapped) пакеты не могут быть описаны.

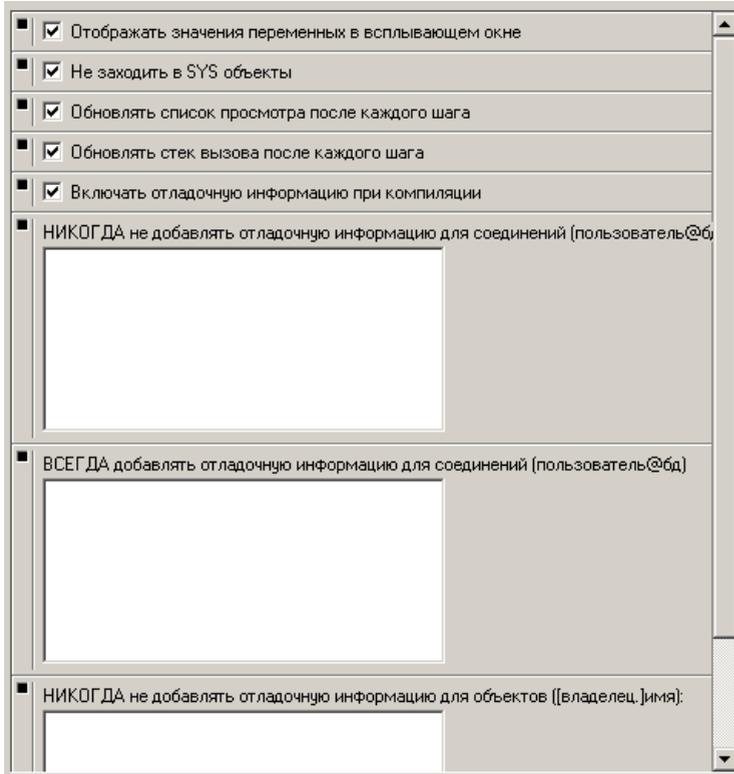
16.2 Oracle - Опции



- Разрешать редактирование исходных текстов БД.
Эта опция контролирует возможность редактирования исходного текста программного модуля или представления полученного прямо из Объектного Навигатора, или же требует открыть его из файловой системы.
- Спрашивать о сохранении отредактированных исходных текстов БД.
Если вы редактируете исходный текст программного модуля, загрузив его из словаря базы данных (щёлкнув правой кнопкой мыши на программном модуле и выбрав во всплывающем меню пункт *Редактировать*), вы можете, отключив эту опцию, предотвратить запрос о необходимости сохранить эти изменения в файл. Если вы работаете с программным модулем, который ранее был открыт из файловой системы, вас всегда будут спрашивать, сохранить ли эти изменения.
- Разрешать компиляцию исходных только-для-чтения файлов.
Если исходный файл доступен только для чтения, его нельзя компилировать, когда он загружен в Окно программного модуля. Это предпочтение полезно при использовании программы Version Control System, которая часто оставляет в рабочей директории доступную только для чтения копию исходного файла. Подобную копию, нельзя компилировать, так как это может помешать работе с другими элементами проекта.
- Подтверждать фиксацию & откат.
Когда вы фиксируете или откатываете транзакцию, нажимая соответствующие кнопки, у вас будет запрошено подтверждение. При отключении этой опции транзакция будет немедленно зафиксирована или откатчена без подтверждения.
- Использовать DBA представления, если доступны.
Если установлено это предпочтение, PL/SQL Developer попытается использовать представления DBA (*dba_objects*, *dba_triggers* и так далее) вместо видов ALL (*all_objects*, *all_triggers* и так далее). Представления DBA могут возвращать больше информации, чем виды ALL, если у разработчика имеются некоторые системные привилегии.
- Автоматическая статистика.
Эта опция контролирует, должен ли для каждого выполненного оператора SQL и Тестового скрипта генерироваться статистический отчёт. Если вы отключите эту опцию, статистический отчёт будет пустым. При нажатии кнопки *Выбрать...*, будет открыт список

со всей статистикой, доступной в базе данных, к которой вы на данный момент подключены. Вы можете выбрать, какую статистику вы хотите включить в статистический отчёт. Если у вас нет доступа к динамическим таблицам, кнопка Выбрать... будет отключена.

16.3 Oracle - Отладчик



- Отображать значения переменных во всплывающем виде.
Если вы отключите это предпочтение, значения переменных не будут автоматически отображены во всплывающем виде, когда вы поместите курсор в Тестовом окне на имени переменной. Вместо этого вы должны выбрать пункт Set Variable во всплывающем меню после щелчка правой кнопкой мыши на имени переменной.
- Не заходить в SYS объекты.
Если у вас есть привилегии просмотра объекта SYS, то отладчик может войти в такой программный модуль при отключении этой опции. Обычно, вы не хотите так делать, поэтому эта опция отключена по умолчанию.
- Обновлять список просмотра после каждого шага.
Если вы отключите эту опцию, вы можете вручную обновлять список просмотра, нажав на отладочной панели инструментов Тестового окна кнопку *Обновить список просмотра*. Это может ускорить отладку на медленных системах.

- Обновлять стек вызов после каждого шага.
Если вы отключите эту опцию, вы можете вручную обновлять стек вызова, нажав на отладочной панели инструментов Тестового окна кнопки *Обновить стек вызова*. Это может ускорить отладку на медленных системах.
- Включать отладочную информацию при компиляции.
Когда эта опция включена, каждая компиляция в PL/SQL Developer будет автоматически включать отладочную информацию, так что значения переменных в течение отладочной сессии всегда можно просматривать и устанавливать. Если вы отключите эту опцию, вы можете вручную добавить отладочную информацию, используя Навигатор.
- НИКОГДА не добавлять отладочную информацию для соединений.
Для этого списка шаблонов соединений, отладочная информация никогда не будет автоматически добавлена, независимо от предпочтения «Включать отладочную информацию при компиляции». Это может быть полезно, если вы хотите быть уверенными, что вы никогда не добавите отладочную информацию в промышленные базы данных или для некоторых пользователей. Вы можете использовать знакомый синтаксис `user@database`, где вы можете использовать подстановочные символы * и ? (например, `sys@*` для пользователя `sys` любой базы данных).
- ВСЕГДА добавлять отладочную информацию для соединений.
Для этого списка шаблонов соединений, отладочная информация всегда будет автоматически добавляться, независимо от предпочтения «Добавлять отладочную информацию во время компиляции». Это может быть полезно, если вы хотите быть уверенным, что вы всегда добавите отладочную информацию в разрабатываемые базы данных или для некоторых пользователей. Вы можете использовать знакомый синтаксис `user@database`, где вы можете использовать подстановочные символы * и ? (например, `*@devdb` для всех пользователей в базе данных `devdb`).
- НИКОГДА не добавлять отладочную информацию для объектов.
Для этого списка объектов никогда не будет добавлена отладочная информация. Некоторые версии Oracle Server в особых ситуациях во время добавления отладочной информации могут вызвать ошибки. В таких случаях в течение компиляции вы будете видеть ошибки ORA-00600, ORA-03113 или PLS-00801. Если описанная проблема возникает, вы можете добавить соответствующий объект в этот список (`owner.name`), пока Oracle не предоставит заплатку.

16.4 Oracle - Вывод

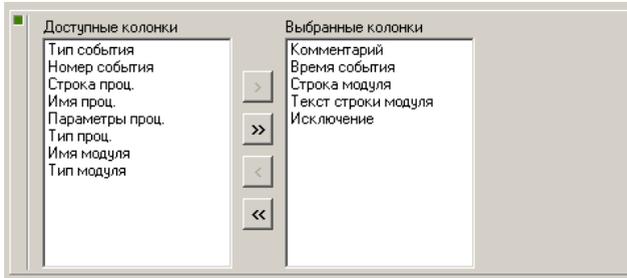


<input checked="" type="checkbox"/> Включен
<input checked="" type="checkbox"/> Очищать перед исполнением
Размер буфера 10000 

- Включен.
В SQL окнах и в Тестовых окнах, обращения к `dbms_output.put_line` будут буферизованы и отображены после выполнения скрипта. Это предпочтение может быть переопределено на странице вывода, каждого отдельного SQL окна и Тестового окна.
- Очищать перед исполнением.
Перед исполнением SQL окна или Тестового окна, текущая страница вывода будет очищена. Таким образом, страница вывода всегда будет содержать вывод, относящийся только к последнему исполнению.

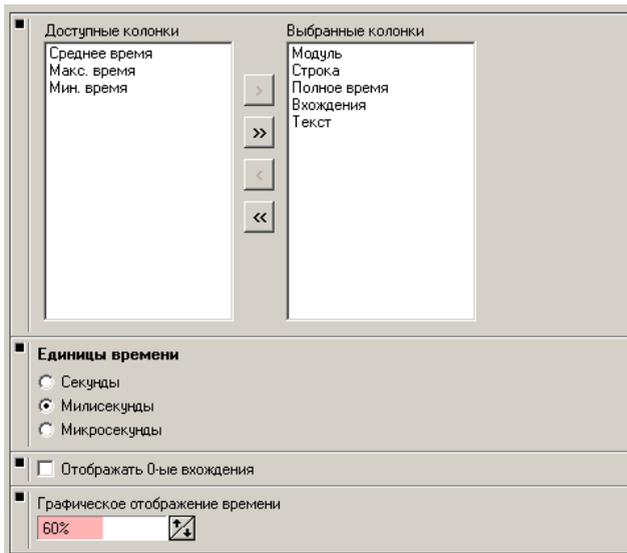
- Размер буфера.
Размер в байтах буфера *dbms_output*. Если в течение одного выполнения число выведенных байт превышает размер буфера, то возникает исключительная ситуация.

16.5 Oracle - Трассировка



Эти предпочтения контролируют страницу *Трассировка* Тестового окна (смотрите главу 4.11). Вы можете контролировать, какие столбцы из таблицы трассировки будут отображены и в каком порядке.

16.6 Oracle - Профилировщик

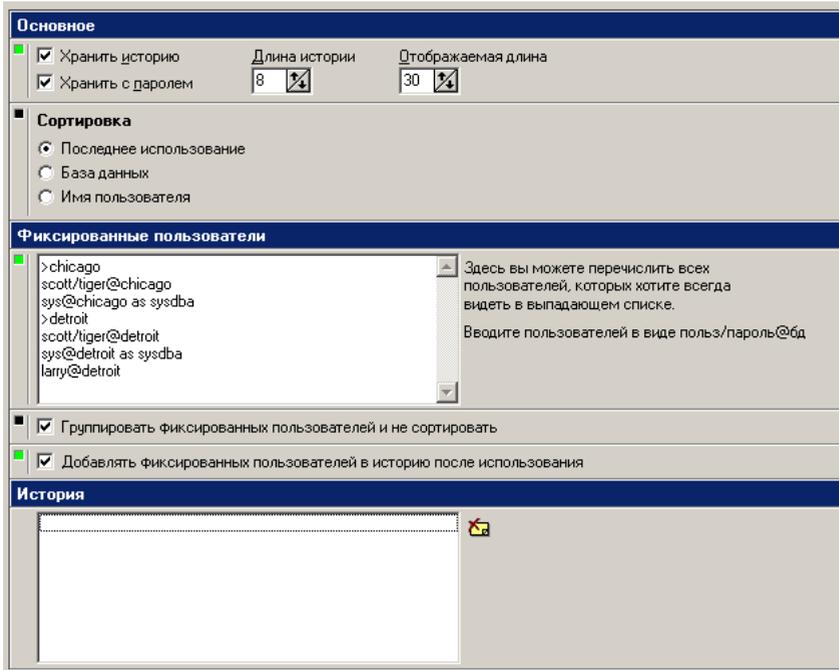


Эти предпочтения контролируют страницу *Профилировщик* Тестового окна (смотрите главу 5.3).

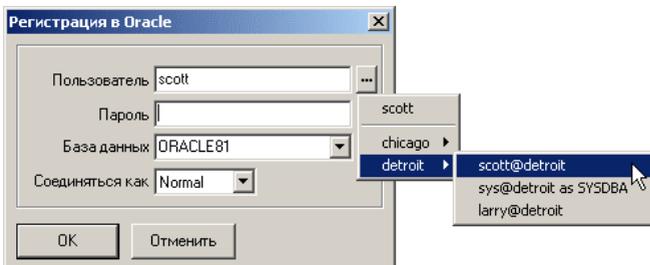
- Колонки.
Контролирует, какие колонки из профильной таблицы следует отображать, и в каком порядке.

- **Единицы времени.**
Определяет единицы измерения для столбцов *Полное время*, *Макс. время*, *Мин. время* и *Среднее время*.
- **Отображать 0-ые вхождения.**
Определяет, будут ли отображены строки отчёта с 0-ым количеством исполнений.
- **Графическое отображение времени.**
Столбец *Полное время* будет графически отображать относительное значение. Вы можете контролировать яркость цветов этой области или сделать её невидимой.

16.7 Oracle – История соединений



Если вы включите предпочтение *Хранить историю*, PL/SQL Developer будет сохранять имя пользователя, пароль и информацию базы данных в *Истории соединений*. Когда вы запустите PL/SQL Developer позже, диалог соединения позволит вам быстро выбрать ранее использованную учётную запись:



Из соображений безопасности, возможно, вы не захотите сохранять пароли в истории, даже, несмотря на то, что PL/SQL Developer зашифрует их. Можно предположить, что кто-нибудь, имеющий доступ к PL/SQL Developer на вашем ПК, может подсоединиться к базе данных, используя ваши учётные записи. В этом случае, вам следует отключить опцию *Хранить с паролем*. Предпочтения *Длина истории* и *Отображаемая длина* контролируют, количество хранящихся соединений (самое старое будет удалено из истории) и количество соединений,

отображаемых на экране. Опция *Сортировка* контролирует, как соединения будут отсортированы в списке выбора.

Заметьте, что даже если вы выбрали сохранение паролей в истории соединений, вы всё ещё можете предотвратить это, удерживая клавишу *Control* во время нажатия клавиши *Return* или кнопки *OK* в диалоге соединения. Таким образом, вы можете предотвратить запоминание паролей привилегированных учётных записей (например, SYS или SYSTEM).

Дополнительно, в секции *Фиксированные пользователи* вы можете определить фиксированные учётные записи, с паролем или без него. Эти фиксированные учётные записи, будут всегда отображаться и никогда не будут удалены автоматически. А также вы можете сопоставить пункты подменю для этих фиксированных пользователей, добавив *>menuname*. В примере, приведённом ниже, подменю будет создано для *chicago* и *detroit*:

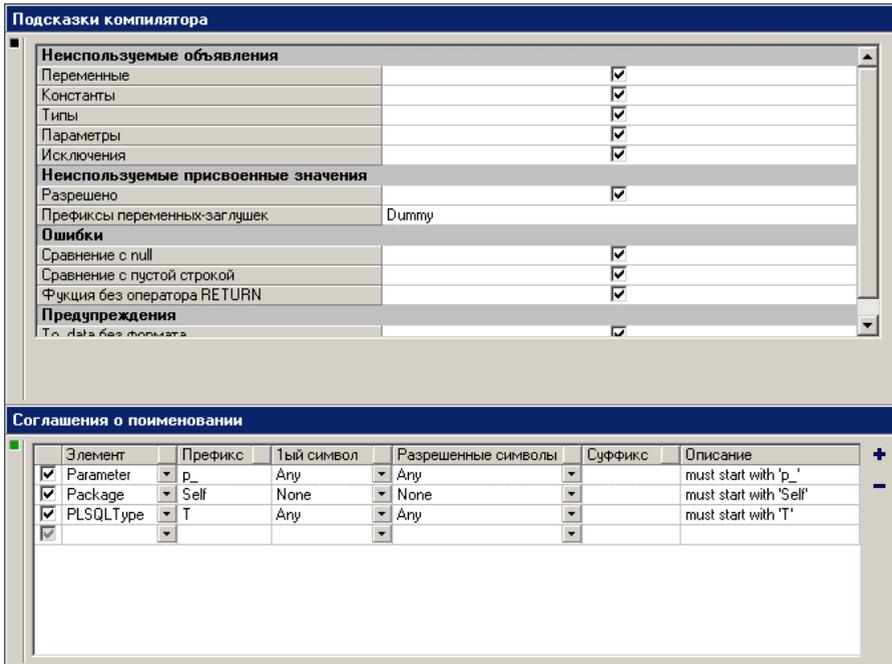
```
>chicago
scott/tiger@chicago
sys@chicago as sysdba
>detroit
scott/tiger@detroit
sys@detroit as sysdba
larry@detroit
```

В этом случае подменю было создано для базы данных, но вы можете создавать его для любой категории, какой хотите. Например, *Development*, *Test*, *Production*.

Если включена опция *Группировать фиксированных пользователей и не сортировать*, фиксированные пользователи будут отображены в отдельной от истории секции и не будут отсортированы. Если включена опция *Добавлять фиксированных пользователей в историю после использования*, то при выборе фиксированного пользователя во время соединения, он также будет включен в историю.

В секции *История* вы можете видеть, какие пользователей были добавлены в историю после использования в PL/SQL Developer. Чтобы удалить элемент, выберите его и нажмите рядом со списком кнопку *Удалить элемент*.

16.8 Oracle – Подсказки



Подсказки компилятора

Вы можете включить и выключить каждую отдельную подсказку, и управлять другими свойствами подсказок, когда это возможно. Объяснение каждой подсказки и свойства отображается под списком подсказок.

Соглашения о именовании

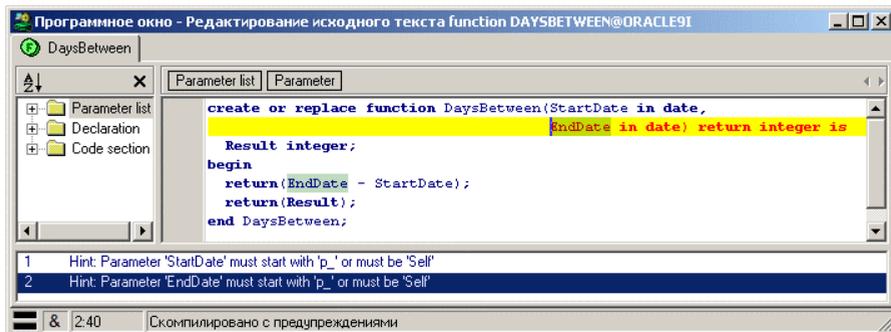
В этом списке вы можете определить соглашения о именовании, которые будут проверяться когда программный модуль компилируется, или когда вызвана функция *Показать подсказки компилятора*. Подсказка будет отображена для каждого элемента в исходном тексте программного модуля, который не соответствует соглашениям о именовании.

Для каждого соглашения о именовании вы можете определить тип элемента (параметр, переменная, и т.д.), необходимый префикс, возможные значения для первого и последующего символов (после префикса), необходимый суффикс, и описание подсказки, которое должно быть отображено, когда соглашение нарушено.

Описание должно быть задано в форме, как показано в примере выше. Полное сообщение подсказки будет: <Элемент> <Имя> <Описание>. Например: Параметр 'StartDate' должен начинаться с 'p'.

Если элемент должен соответствовать одному из многих соглашений о именовании, тогда вы можете просто добавить несколько строк для одного и того же типа элемента. Описания всех строк будет выведены, разделенные 'или'. Для примера выше сообщение подсказки параметра будет:

Параметр 'StartDate' должен начинаться с 'p' или должен быть 'Self'



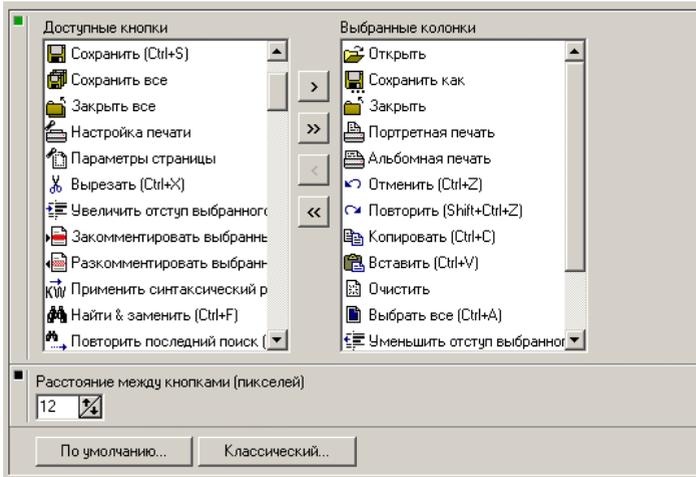
16.9 Пользовательский интерфейс – опции

<input checked="" type="checkbox"/>	Автосохранение имени пользователя
<input type="checkbox"/>	Автосохранение рабочего стола
<input type="checkbox"/>	Использовать внутренний HTML просмотрщик
<input type="checkbox"/>	Окно с HTML справкой: поверх всех окон
<input type="checkbox"/>	Использовать многорядные закладки
<input type="checkbox"/>	Отображать полный путь файла в заголовках окон
<input checked="" type="checkbox"/>	Использовать проекты

- **Автосохранение имени пользователя.**
Эта опция может быть включена для автоматического сохранения имени пользователя каждой сессии PL/SQL Developer. Когда вы перезапускаете PL/SQL Developer, последнее имя пользователя будет указано по умолчанию в диалоге соединения.
- **Автосохранение рабочего стола.**
Когда эта опция включена, все файлы, которые были открыты, когда вы вышли из PL/SQL Developer, будут автоматически открыты при повторном запуске PL/SQL Developer. Текущие открытые Командное окно или Окно Explain Plan, которые не связаны с файлом, также будут восстановлены. А также, будут восстановлены размеры главного окна и Объектного навигатора.
- **Использовать внутренний HTML просмотрщик.**
По умолчанию, для отображения справочной документации HTML, PL/SQL Developer будет использовать внутреннюю программу для просмотра HTML. Если вы используете Microsoft Internet Explorer 4 или более позднюю версию, вы можете отключить эту опцию и вместо этого использовать HTML компоненты Microsoft. При этом во всплывающем меню у вас появятся несколько дополнительных функций, а также будут унаследованы предпочтения, сделанные в Internet Explorer.
- **Окно с HTML справкой: поверх всех окон.**
Если эта опция включена, окно с HTML справкой будет оставаться наверху до тех пор, пока вы не закроете его или не минимизируете. Если она отключена, окно будет представлено отдельным заданием, со своей собственной кнопкой в области задач Windows. Таким образом, вы можете перенести на передний план главное окно PL/SQL Developer или Справочное окно, куда вам это нужно.
- **Использовать многорядные закладки.**
Если окно имеет несколько страниц с закладками, это предпочтение контролирует, можно ли разбить закладки на несколько рядов, в том случае, когда закладки не помещаются в один ряд. Если эта опция выключена и происходит переполнение, появятся 2 кнопки прокрутки. Пока это только работает для Программного окна.
- **Отображать полный путь файла в заголовках окон.**
По умолчанию, в заголовке окна отображается только имя файла. Если эта опция включена, то будет отображён полный путь.
- **Использовать проекты.**
Элемент главного меню *Проект* будет видимым, только если данная опция включена. Если вы не используете проекты, отключите её. Тогда она не будет занимать место в главном меню. Для более детальной информации о проектах смотрите главу 11.

- DSA диалог.
Когда вы выбираете опцию 'Не показывать это сообщение снова' в окне сообщения, строка будет добавлена к DSA секции в ваших предпочтениях. Если позже вы захотите снова включить это сообщение, нажмите кнопку *DSA диалог*, что бы отредактировать эту секцию. Появится текстовый редактор, который позволит вам удалить соответствующую строку.

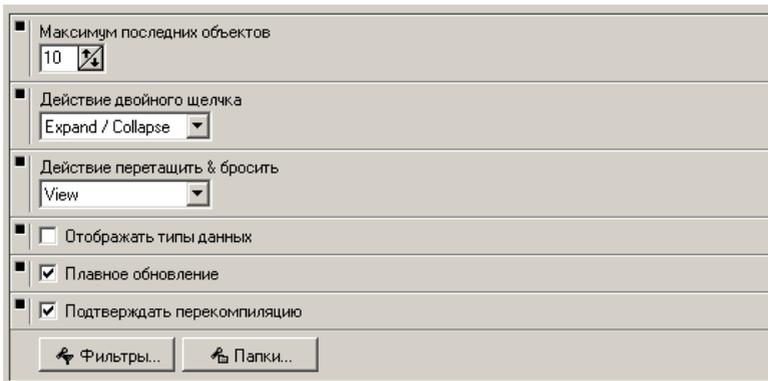
16.10 Пользовательский интерфейс – Панель инструментов



Вы можете контролировать, какие кнопки должны быть видимыми, а также их порядок. Чтобы разделить группы связанных кнопок, используйте элемент *Разделитель*. Чтобы создать под предыдущей панелью инструментов новую, можно использовать элемент *Новая панель*.

Кнопка *По умолчанию* возвращает панель инструментов к её состоянию, принятому по умолчанию. Кнопка *Классический* возвращает панель инструментов к состоянию, как в PL/SQL Developer 3. Чтобы изменить расстояние между кнопками, задайте значение *Расстояние между кнопками*.

16.11 Пользовательский интерфейс – Навигатор



Эти предположения контролируют внешний вид и поведение Объектного Навигатора (смотрите главу 15).

- **Максимум последних объектов.**
Контролирует количество объектов, которые будут видны в папке *Последние объекты*.

Наиболее давно использованный объект будет удален, когда число отображаемых объектов превысит это значение.

- Действия двойного щелчка и перетаскивания.
Контролирует, какое действие будет выполнено при двойном щелчке по объекту в навигаторе или когда вы перетаскиваете его из навигатора в рабочее пространство. Вы можете указать действие по умолчанию и переопределить или наследовать это действие по умолчанию для каждого типа объектов в списке.
-
- Отображать типы данных.
Когда эта опция включена, в навигаторе будут отображены типы данных в столбцах (таблицы и виды), атрибуты (объекты) и параметры (функции и процедуры).
- Плавное обновление.
В некоторых системах эта опция может замедлить операцию обновления Навигатора (например, в случаях изменения фильтра навигатора, регистрации другого пользователя и так далее). В этом случае, вам следует отключить её, однако это может привести к небольшому мерцанию экрана в процессе обновления Навигатора.
- Подтверждать перекомпиляцию.
Когда это предпочтение включено, то после перекомпилирования программного модуля в Объектном Навигаторе, появится сообщение об удачной или неудачной перекомпиляции. Когда оно отключено, вы будете видеть сообщение, только когда компиляция заканчивается ошибкой.

Кнопки *Фильтры* (смотрите главу 15.2) и *Папки* (смотрите главу 15.3) вызовут соответствующий диалог.

16.12 Пользовательский интерфейс - Редактор

Отступ																																				
<input checked="" type="checkbox"/> Автоматически	Шаг (символов) 2 <input type="text"/>																																			
Табуляция & перенос																																				
<input checked="" type="checkbox"/> Умная табуляция	Размер табуляции 2 <input type="text"/>																																			
<input type="checkbox"/> Умное заполнение																																				
<input type="checkbox"/> Использовать символ табуляции																																				
<input type="checkbox"/> Перенос строк	Видимая граница 0 <input type="text"/>																																			
<input type="checkbox"/> Разрешать позиционирование курсора после конца строки																																				
Выделение синтаксиса																																				
<input checked="" type="checkbox"/> Включено																																				
<table border="0"> <thead> <tr> <th></th> <th>Жирно</th> <th>Курсив</th> <th>Цвет Текст</th> <th>Цвет Фон</th> </tr> </thead> <tbody> <tr> <td><input checked="" type="checkbox"/> Ключевые слова</td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> <td>■</td> <td>■</td> </tr> <tr> <td><input checked="" type="checkbox"/> Комментарий</td> <td><input type="checkbox"/></td> <td><input checked="" type="checkbox"/></td> <td>■</td> <td>■</td> </tr> <tr> <td><input checked="" type="checkbox"/> Строки</td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td>■</td> <td>■</td> </tr> <tr> <td><input checked="" type="checkbox"/> Числа</td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td>■</td> <td>■</td> </tr> <tr> <td><input checked="" type="checkbox"/> Символы</td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td>■</td> <td>■</td> </tr> <tr> <td><input checked="" type="checkbox"/> Пользовательские</td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> <td>■</td> <td>■</td> </tr> </tbody> </table>		Жирно	Курсив	Цвет Текст	Цвет Фон	<input checked="" type="checkbox"/> Ключевые слова	<input checked="" type="checkbox"/>	<input type="checkbox"/>	■	■	<input checked="" type="checkbox"/> Комментарий	<input type="checkbox"/>	<input checked="" type="checkbox"/>	■	■	<input checked="" type="checkbox"/> Строки	<input type="checkbox"/>	<input type="checkbox"/>	■	■	<input checked="" type="checkbox"/> Числа	<input type="checkbox"/>	<input type="checkbox"/>	■	■	<input checked="" type="checkbox"/> Символы	<input type="checkbox"/>	<input type="checkbox"/>	■	■	<input checked="" type="checkbox"/> Пользовательские	<input checked="" type="checkbox"/>	<input type="checkbox"/>	■	■	
	Жирно	Курсив	Цвет Текст	Цвет Фон																																
<input checked="" type="checkbox"/> Ключевые слова	<input checked="" type="checkbox"/>	<input type="checkbox"/>	■	■																																
<input checked="" type="checkbox"/> Комментарий	<input type="checkbox"/>	<input checked="" type="checkbox"/>	■	■																																
<input checked="" type="checkbox"/> Строки	<input type="checkbox"/>	<input type="checkbox"/>	■	■																																
<input checked="" type="checkbox"/> Числа	<input type="checkbox"/>	<input type="checkbox"/>	■	■																																
<input checked="" type="checkbox"/> Символы	<input type="checkbox"/>	<input type="checkbox"/>	■	■																																
<input checked="" type="checkbox"/> Пользовательские	<input checked="" type="checkbox"/>	<input type="checkbox"/>	■	■																																
<input type="text" value="Ключевые слова пользователя..."/>																																				
<input type="checkbox"/> Регистр ключевого слова	Unchanged <input type="text"/>																																			
Прочие																																				
<input type="checkbox"/> Отображать номера строк																																				
<input type="checkbox"/> Выделять редактируемую строку																																				
<input type="checkbox"/> Переходить в начало / конец файла, когда текст не найден																																				
<input checked="" type="checkbox"/> Выделять скобки																																				
<input type="checkbox"/> Вставлять разделенные запятой значения одной строкой																																				

Отступ

- **Автоматически.**
Определяет, будет ли редактор автоматически делать отступ после нажатия клавиши Enter.
- **Шаг (символов).**
Количество символов, которое редактор будет отступать при увеличении или уменьшении отступа выделенного текста.

Табуляция и перенос

- **Умная табуляция.**
Заставляет редактор переходить при нажатии клавиши Tab к позиции, относительно предыдущей строки.
- **Размер табуляции.**
Определяет количество символов, которые будут использоваться при табуляции.

- Умное заполнение.
Редактор будет заменять пробелы символами табуляции, если это возможно.
- Использовать символ табуляции.
Когда это предпочтение отключено, для табуляций и отступов будут использоваться только пробелы.
- Перенос строк.
Строки, ширина которых больше, чем у окна, будут продолжены на следующей строке.
- Разрешить позиционирование курсора после конца строки.
Когда это предпочтение включено, вы можете позиционировать курсор после окончания строки. Когда вы вставляете туда текст, перед этим строка будет дополнена пробелами.
- Видимая граница.
Отобразит в указанной позиции вертикальную линию. 0 = отключено.

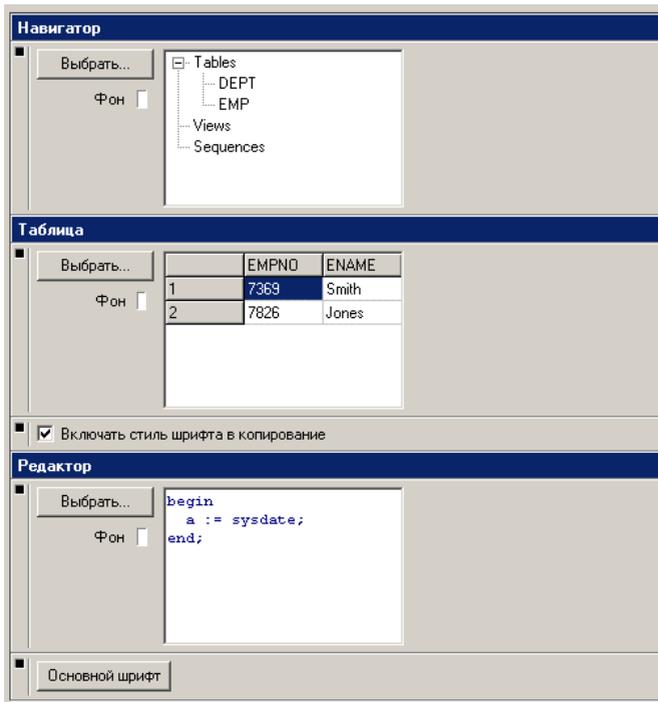
Выделение синтаксиса

Эта группа предпочтений контролирует выделение синтаксиса, используемое в редакторах. Вы можете включить или отключить выделение синтаксиса и определить стиль и цвет ключевых слов, комментариев, строк, чисел и символов (:=, =>, || и так далее). Для ключевых слов вы можете дополнительно контролировать, следует ли их преобразовать в верхний регистр, в нижний регистр или начинать с заглавной буквы. Вы также можете определить пользовательское выделение синтаксиса, а также можете определить слова, которые следует выделять. При нажатии кнопки *Ключевые слова пользователя* появится редактор, который позволит вам определить эти слова.

Прочее

- Отображать номера строк.
Включение этой опции делает номера строк в левой части Программных редакторов и Тестовых окон видимыми.
- Выделять редактируемую строку.
Контролирует, будет ли выделена строка, на которой находится курсор.
- Переходить в начало / конец файла, когда текст не найден.
Когда вы используете функцию *Найти*, и искомый текст не найден, поиск будет продолжен с начала файла (при поиске в прямом направлении) или с конца файла (при поиске в обратном направлении).
- Выделять скобки.
Когда это предпочтение включено, и курсор находится на открывающей или закрывающей скобке, то редактор выделит пару скобок.
- Цвет выделения.
Задает цвет для всех выделений, таких как скобки, переменные и т.д.
- Цвет найденного.
Задает цвет найденного при помощи панели поиска.
- Вставлять разделенные запятой значения одной строкой.
Контролирует, будут ли несколько выделенных в Объектном Навигаторе или Окне описаний элементов вставлены в редактор в одну строку или в несколько строк.

16.13 Пользовательский интерфейс – Шрифты



Предпочтение Шрифты определяет шрифт навигатора, шрифт редактора и шрифт, используемый в различных таблицах, таких как окно SQL, окно свойств и так далее. Нажатием кнопки *Выбрать* вы можете изменить имя, стиль и цвет шрифта. Вы также можете установить основной шрифт для всех диалогов.

16.14 Пользовательский интерфейс – Ассистент Кодирования

Ассистент Кодирования

- Автоматически активизировать
- Задержка (мс):
- Стиль кодирования:
- Использовать оригинальный регистр, если возможно
- Описывать пользователей
 - Таблицы
 - Представления
 - Последовательности
 - Функции
 - Процедуры
 - Пакеты
 - Типы
- Описывать контекст
 - Минимум символов:
- Описывать стандартные функции (например to_char)

Ассистент Кодирования автоматически отображает информацию об объекте базы данных, как только вы ввели его имя (смотрите главу 18.4). Эта страница предпочтений позволит вам определить поведение этой функции.

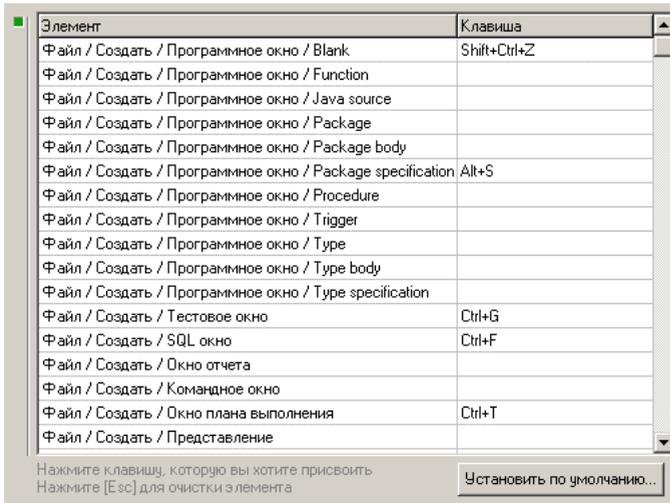
- Автоматически активизировать.
Ассистент Кодирования может быть вызван автоматически после некоторой задержки (смотрите ниже). Вы также можете выбрать режим ручной активации Ассистент Кодирования при помощи функциональной клавиши.
- Задержка.
Количество миллисекунд, которое редактор будет ожидать перед тем, как отобразит список Ассистента Кодирования.
- Стиль кодирования.
Указывает, как элементы должны быть вставлены в редактор, когда вы их выбираете:
 - Умный – Ассистент Кодирования проанализирует описываемый объект, чтобы определить стиль.
 - Начинать слово с прописной буквы – Первая буква каждого слова (разделенных символом подчёркивания) будет заглавной.
 - Нижний регистр – Все символы будут преобразованы в нижний регистр.
 - Верхний регистр – Все символы будут преобразованы в верхний регистр.
- Использовать оригинальный регистр, если возможно.
Когда эта опция включена, то, если возможно, Ассистент Кодирования определит регистр идентификатора, используя исходный текст хранящийся в Oracle Dictionary. Это применимо ко всем программным модулям и их элементам (параметрам, типам и так далее) и для колонок представлений. А так же переопределяет предпочтение Стиля кодирования, описанное выше. Если оригинальный регистр не может быть определён, будет применено

предпочтение Стиля кодирования.

Возможно, для улучшения производительности, вы захотите отключить эту функцию.

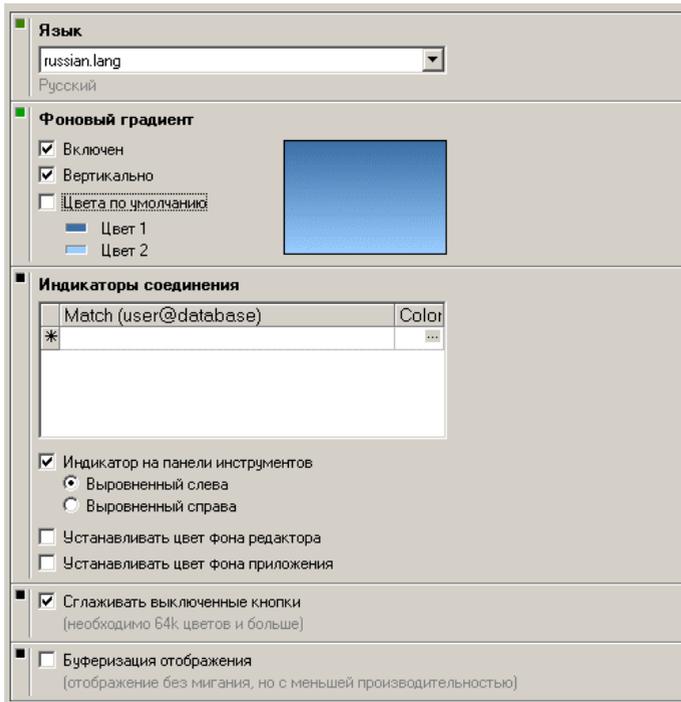
- **Описывать пользователей.**
Определяет, будут ли приведены в списке объекты, принадлежащие пользователю, когда вы вводите имя пользователя и точку. Если эта опция включена, вы можете дополнительно определить, какие объекты вы хотите включить в список.
- **Описывать контекст.**
Определяет, следует ли Ассистенту Кодирования описывать контекст текущего пользователя, редактора и программного модуля.
- **Минимум символов.**
Определяет, сколько букв слова необходимо ввести, чтобы автоматически было вызвано контекстное описание. Заметьте, что вы всегда можете вручную вызвать Ассистента Кодирования, даже если заданное количество символов не введено.
- **Описывать стандартные функции.**
По умолчанию, Ассистент Кодирования будет описывать стандартные функции, такие как `to_char`, `add_months` и так далее. Если эти функции вам хорошо знакомы, вы можете отключить эту опцию.

16.15 Пользовательский интерфейс – Настройка клавиатуры



На этой странице вы можете определить ваше собственное назначение клавиш для всех функций PL/SQL Developer. Просто выберите в списке функцию и нажмите комбинацию клавиш, которую вы хотите использовать для этой функции. Чтобы удалить горячую клавишу для данной функции, выберите её в списке и нажмите клавишу *Esc*.

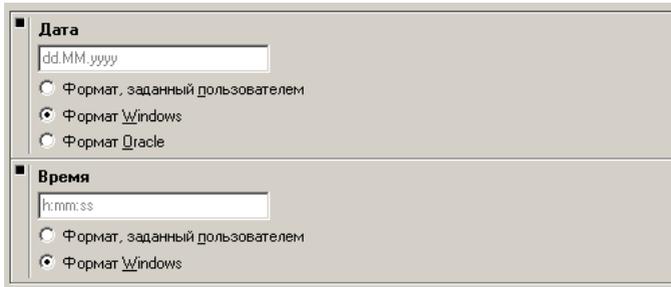
16.16 Пользовательский интерфейс – Внешний вид



- **Язык.**
Если у вас установлена многоязыковая версия PL/SQL Developer, то здесь вы можете выбрать язык. Все надписи и сообщения после этого будут отображаться на этом языке.
- **Фоновый градиент.**
Позволяет вам определить цвет и градиент рабочего пространства PL/SQL Developer. Если вы не хотите использовать градиент, а только один цвет, установите одно и то же значение для Цвет 1 и Цвет 2.
- **Индикаторы соединения.**
Эти опции позволят вам обеспечить некоторую визуальную индикацию использования специфических соединений. Вы можете определить список шаблонов соединений в формате user@database и связать с ними определённый цвет. Этот цвет будет присвоен индикатору в левом или правом краю панели инструментов, фону редактора или фону приложения (отменяя градиент фона).
Шаблон соединения может содержать подстановочные символы * и ?. В приведенном выше примере, фон редактора и индикатор с правого края панели инструментов будут красными, когда любой пользователь установил соединение с базой данных PRODDb. Он будет жёлтым, если пользователь SYS установил соединение с любой базой данных. Первый элемент в списке имеет преимущество, поэтому индикаторы, после подключения пользователя SYS к базе данных PRODDb, будут красными.

- Сглаживать выключенные кнопки.
Когда это предпочтение включено, выключенные кнопки будут отображаться сглаженными. Когда оно отключено, для выключенных кнопок используется стандартный внешний вид. Для 256-цветных мониторов для выключенных кнопок всегда применяется стандартный вид.
- Буферизация отображения.
Мерцание экрана будет меньше при перемещении или изменении размеров окон, но скорость вывода на экран может стать немного меньше.

16.17 Пользовательский интерфейс – Дата/Время



На этой странице вы можете определить формат даты и времени, которые будут использованы в PL/SQL Developer.

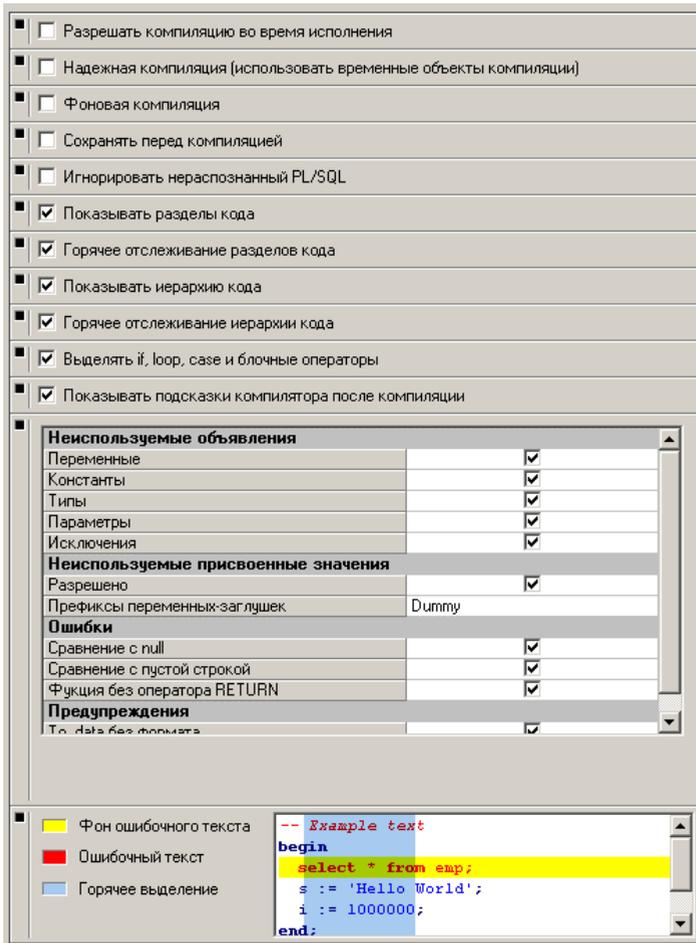
Формат даты может быть определён одним из 3 способов:

- Формат, заданный пользователем.
Явно указанный формат даты. Используйте те же правила, как и в случае региональных настроек Windows.
- Формат Windows.
Формат, определённый в региональных настройках Windows.
- Формат Oracle.
Формат, определённый вашим Oracle окружением (NLS_DATE_FORMAT).

Формат времени может быть определён одним из 2 способов:

- Формат, заданный пользователем.
Явно указанный формат времени. Используйте те же правила, как и в случае региональных настроек Windows.
- Формат Windows.
Формат, определённый в региональных настройках Windows.

16.18 Типы окон – Программное окно



Эти предпочтения определяют поведение Программного окна (смотрите главу 3).

- Разрешать компиляцию во время исполнения.
В мульти или двух сессионном режиме вы имеете техническую возможность компилировать программные модули, пока вы запускаете Тестовые скрипты или SQL скрипты. Это может привести к блокировкам, которые могут быть предотвращены при помощи этого предпочтения.
- Надежная компиляция (использовать временные объекты компиляции).
Когда это предпочтение включено, каждая компиляция сначала будет протестирована с использованием временного объекта компиляции. Если эта тестовая компиляция прошла успешно, основной объект будет скомпилирован. Это может быть полезным, если вы компилируете объекты в среде, где невалидные объекты могут вызвать проблемы.

- **Фоновая компиляция.**
Когда это предпочтение включено, компиляция будет выполняться в фоновом потоке, с использованием отдельной сессии. Таким образом, вы можете продолжать другую работу, пока происходит компиляция. Это потребует дополнительной сессии базы данных для каждого Программного окна. Заметьте, что это предпочтение работает только, если для предпочтения Режим сессий (смотрите главу 16.1) установлено значение Много сессий.
- **Сохранять перед компиляцией.**
Если вы хотите быть уверенным, что ваш объект базы данных и исходный файл в файловой системе будут синхронизированы, вы можете включить эту опцию.
- **Игнорировать нераспознанный PL/SQL.**
Когда это предпочтение включено, Программное окно будет игнорировать все SQL команды, которые не начинаются с “create or replace <program unit type>”. Другие команды SQL or SQL*Plus в программном файле будут проигнорированы.
- **Показывать разделы кода.**
Это предпочтение указывает, будет ли отображаться панель с Разделами кода. Для большей информации смотрите главу 18.11.
- **Горячее отслеживание разделов кода.**
Контролирует, будет ли PL/SQL код автоматически выделяться в редакторе, когда курсор мыши находится над элементом в панели Разделов кода.
- **Показывать иерархию кода.**
Это предпочтение контролирует, будет ли отображена панель с Иерархией кода. Для большей информации смотрите главу □.
- **Горячее отслеживание иерархии кода.**
Контролирует, будет ли PL/SQL код автоматически выделяться в редакторе, когда курсор мыши находится над элементом в панели Иерархии кода.
- **Выделять переменные.**
Когда курсор находится на переменной, все появления этой переменной будут выделены. Это позволяет просто увидеть все места, где переменная использована, и какой эффект может быть от ее изменения. Если эта опция выключена, вам будет необходимо использовать пункт *Найти сопоставления* из меню *Правка*.
- **Выделять if, loop, case и блочные операторы.**
Когда курсор находится на ключевых словах if / then / else / elsif / end if, loop / end loop, for / loop / end loop, while / loop / end loop, case / when / then / else / end case и declare / begin / end. Если эта опция выключена, вам будет необходимо использовать пункт *Найти сопоставления* из меню *Правка*.
- **Показывать подсказки компилятора после компиляции.**
Подсказки компилятора предоставляют информацию о неиспользованных объявлениях, неиспользованных присвоениях, потенциальных ошибках и предупреждениях. Когда это предпочтение включено, эти подсказки автоматически отображаются после компиляции, после ошибок компиляции (если они присутствуют). Если вы отключите эту опцию, вам надо будет явно выбрать в меню *Инструменты* или на панели инструментов пункт *Показывать подсказки компилятора*.
- **Фон ошибочного текста.**
Выберите цвет фона, которым будет выделяться текст, содержащий ошибку компиляции.
- **Текст ошибки.**
Выберите цвет текста, содержащего ошибку компиляции.

- Горячее выделение.
Выберите фоновый цвет выделенного текста для горячего отслеживания разделов кода, иерархии кода, ключевых слов if, loop, case и блочных операторов.
- Сохранять состояние окна.
Когда включена, позиция, размер, закладки, цветовые пометки и подстановочные переменные Программного окна будут сохраняться в отдельном .pfl файле и восстанавливаться когда файл или объект базы данных повторно открывается для просмотра или редактирования. Расположение этих файлов так же может быть задано.

16.19 Типы окон – Окно SQL

<input type="checkbox"/>	АвтоФиксация SQL
<input type="checkbox"/>	АвтоФиксация посланных записей
<input checked="" type="checkbox"/>	АвтоИсполнение запросов
<input type="checkbox"/>	АвтоВыбор команды
<input checked="" type="checkbox"/>	Спрашивать про сохранение новых окон
<input type="checkbox"/>	Связанный запрос в новом окне
<input type="checkbox"/>	to_char для числовых полей
<input type="checkbox"/>	to_char для полей дат
<input checked="" type="checkbox"/>	Отображать информацию словаря БД в таблице результата
<input type="checkbox"/>	Цвет ячейки с Null значением
<input type="checkbox"/>	Записей на странице
<input checked="" type="radio"/>	Автоматически определяемое
<input type="radio"/>	Фиксированное
	<input type="text" value="100"/> %
<input type="radio"/>	Все записи
<input type="checkbox"/>	Запрос по образцу
<input type="checkbox"/>	Не учитывать регистр
<input type="checkbox"/>	Автоматическое содержание
<input type="checkbox"/>	Игнорировать время
<input type="checkbox"/>	Макет чисел
<input type="radio"/>	Выровненное по левому краю
<input type="radio"/>	Выровненное по правому краю
<input checked="" type="radio"/>	Форматированное
<input type="radio"/>	Форматированное с тысячным разделителем

Эти опции определяют поведение SQL окна (смотрите главу 6).

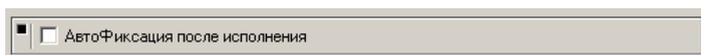
- АвтоФиксация SQL.
Эта опция контролирует, будут ли автоматически зафиксированы операторы DML, выполненные в SQL окне.
- АвтоФиксация посланных записей.
Эта опция контролирует, будут ли автоматически зафиксированы записи, посланные из результирующей таблицы.
- АвтоИсполнение запросов.
Если эта опция включена, то SQL окно созданное после выбора *Запросить данные* или

Редактировать данные для таблицы или представления, будет немедленно выполнено. Если эта опция выключена, вы можете изменить сгенерированный оператор SELECT перед его выполнением.

- АвтоВыбор команды.
Если у вас в редакторе SQL имеется несколько операторов SQL (разделённых точками с запятой), то SQL окно перед выполнением автоматически выделит оператор, на котором находится курсор. Таким образом, вам не придётся самим выделять оператор.
- Спрашивать про сохранение новых окон.
Когда вы закрываете SQL окно, которое было только что создано, обычно вам будет задан вопрос, хотите ли вы сохранить его в файл. Если вы отключите эту опцию, это подтверждение будет опущено. Если вы изменили ранее открытый SQL файл, то, несмотря на это предпочтение, вам будет всегда задаваться вопрос о необходимости сохранения этих изменений.
- Связанный запрос в новом окне.
Когда установлено это предпочтение, то для выполнения связанного запроса (смотрите главу 6.4) будет создано новое окно SQL.
- to_char для числовых полей.
Когда это предпочтение включено, числовые поля будут преобразованы в символьные величины на сервере. NLS параметры сессии будут определять формат этих величин. PL/SQL Developer не будет осуществлять никакого дополнительного форматирования или проверки числовых величин.
- to_char для полей дат.
Когда это предпочтение включено, поля дат будут преобразованы в символьные величины на сервере. NLS параметры сессии будут определять формат этих величин. PL/SQL Developer не будет осуществлять никакого дополнительного форматирования или проверки дат, и окно выбора даты не будет доступно во время редактирования полей дат.
- Отображать информацию словаря БД в таблице результата.
Когда это предпочтение включено, в результирующей таблице будет отображена следующая словарная информация:
 - В строке статуса будут отображены тип данных столбца, опциональность и комментарий.
 - Для столбцов с проверочными ограничениями, которые проверяют на вхождение значения в список (например, col in (value1, value2, ...)) будет отображён список выбора.
 - Для столбцов с ограничениями внешних ключей, ссылающихся на небольшие таблицы с количеством записей меньше 1000, будет отображён список выбора.
- Цвет ячейки с Null значением.
Этот цвет будет использоваться для обозначения Null значений. Так легче отличить Null значения от значений, состоящих только из пробелов, и значений, которые не отображаются непосредственно в таблице (например, LOB's and Longs).
- Записей на странице.
Это предпочтение определяет, сколько записей может быть первоначально выбрано для оператора SELECT, а также каждый раз, когда вы нажимаете в результирующей таблице Окна SQL кнопку *Следующая страница*:
 - Автоматически определяемое – Размер результирующей таблицы определяет, сколько записей будет выбрано.

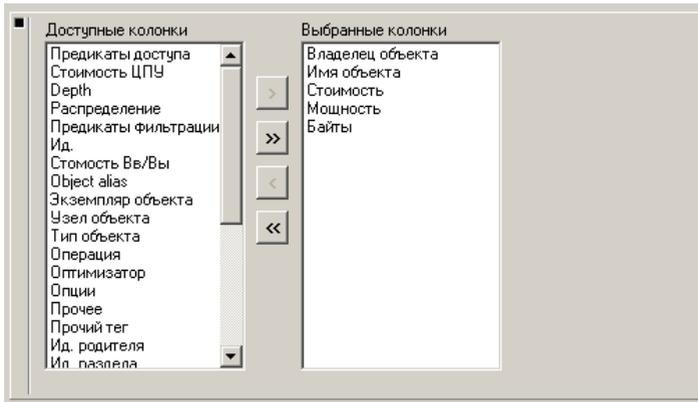
- Фиксированное – Будет выбрано указанное число записей.
- Все записи – Все записи будут немедленно выбраны после исполнения запроса.
- Макет чисел.
Это предпочтение указывает, как будут отображаться числовые столбцы:
 - Выровненное по левому краю – Значения отображаются выровненными по левому краю, без какого-либо форматирования.
 - Выровненное по правому краю – Значения отображаются выровненными по правому краю, без какого-либо форматирования.
 - Форматированное – Значение отображаются, выровненными по правому краю в формате, который соответствует точности и шкале поля.
 - Отформатировано с тысячным разделителем – Схоже с предыдущей опцией, но также включает в себя тысячный разделитель (например, 1,277.65)
- Запрос по образцу.
Эти предпочтения контролируют поведение SQL окна по умолчанию во время режима QBE (смотрите главу 6.3):
 - Не учитывать регистр – Запрашиваемые значения символьных полей не чувствительны к регистру.
 - Автоматическое содержание – Символьные поля должны содержать только запрашиваемые значения.
 - Игнорировать время – Временная часть полей дат игнорируется, если запрашиваемые значения не содержат времени.

16.20 Типы окон – Тестовое окно



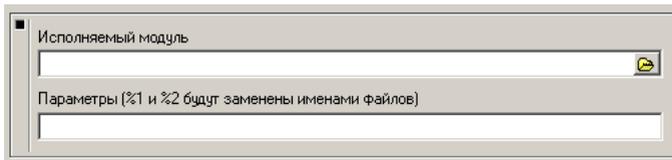
- АвтоФиксация после исполнения.
Эта опция указывает, будут ли автоматически фиксироваться транзакции после исполнения Тестовых скриптов.

16.21 Типы окон - Окно плана



Эти предпочтения указывают, какие столбцы следует отображать в Окне плана (смотрите главу 5.1), и в каком порядке они должны появляться.

16.22 Инструменты – Сравнение



На этой странице предпочтений для инструмента *Сравнить объекты пользователя* вы можете выбрать внешнюю программу сравнения (смотрите главу 17.10).

16.23 Инструменты – Генератор данных

Общие предпочтения

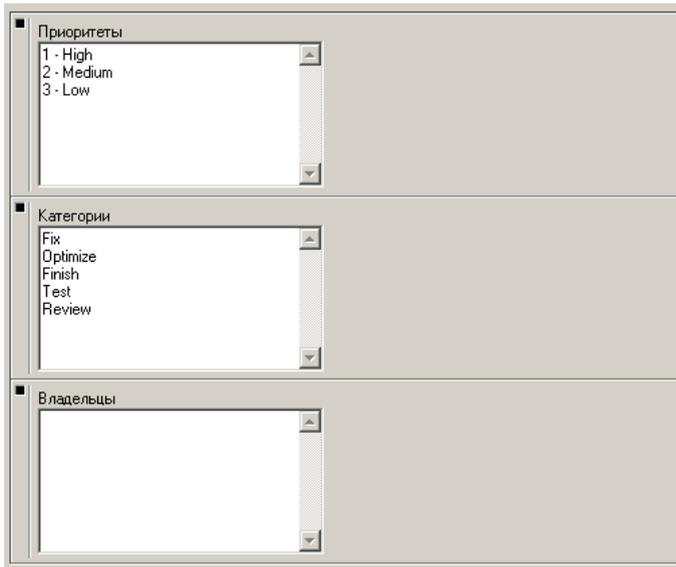
- Формат дат
[Empty text box]
- Фиксировать каждые...
0 [Spin box]
- Задержка (мСек)
0 [Spin box]
- Количество элементов
10..20 [Text box]

Пользовательские определения

Элемент	Значение
*	

Эта страница предпочтений позволит вам определять глобальный набор значений по умолчанию для инструмента *Генератор данных*. Эти значения по умолчанию могут быть переустановлены для каждого определения генератора данных. Для большей информации о значении этих предпочтений смотрите главу 17.8.

16.24 Инструменты – То-До список



Приоритеты

- 1 - High
- 2 - Medium
- 3 - Low

Категории

- Fix
- Optimize
- Finish
- Test
- Review

Владельцы

Эти предпочтения управляют значениями списков выбора *Priority*, *Category* and *Owner* в Редакторе То-До элемента. Для приоритетов может быть полезным начинать каждое значение с номера, так что список может быть отсортирован по приоритетам. Для более детальной информации смотрите главу 13.

16.25 Инструменты – История команд



Глубина памяти (команд)

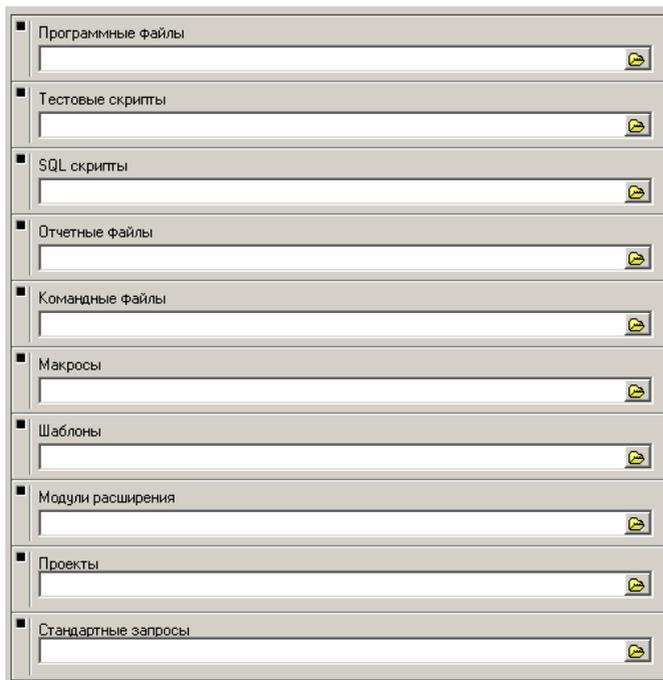
200

Каталог для файла запоминания (PLSRecall.dat)

C:\Windows\Application Data\PLSQL Developer\

На этой странице вы можете определить максимальное количество команд в буфере запоминания команд и каталог, в котором его следует хранить. Для более детальной информации смотрите главу 18.5.

16.26 Файлы - Каталоги



- **Программные файлы.**
Определяет исходный каталог при открытии или сохранении Программных файлов. Если вы оставите этот каталог пустым, вы всегда будете начинать с каталога, в котором вы последний раз открывали или сохраняли Программный файл.
- **Тестовые скрипты.**
Определяет исходный каталог при открытии или сохранении Тестовых скриптов. Если вы оставите этот каталог пустым, вы всегда будете начинать с каталога, в котором вы последний раз открывали или сохраняли Тестовый скрипт.
- **Скрипты SQL.**
Определяет исходный каталог при открытии или сохранении скриптов SQL. Если вы оставите этот каталог пустым, вы всегда будете начинать с каталога, в котором вы последний раз открывали или сохраняли скрипт SQL.
- **Файлы отчётов.**
Определяет исходный каталог при открытии или сохранении файлов отчётов. Если вы оставите этот каталог пустым, вы всегда будете начинать с каталога, в котором вы последний раз открывали или сохраняли файл отчёта.
- **Командные файлы.**
Когда вы открываете командный файл из меню или, введя *@file*, этот каталог будет использован по умолчанию. Если вы оставите этот каталог пустым, будет использован рабочий каталог.

- **Макросы.**
Этот каталог будет использован для хранения ваших макро библиотек. Если вы оставите этот каталог пустым, то будет использован подкаталог *Macro* в каталоге PL/SQL.
- **Шаблоны.**
Этот каталог будет использован для хранения шаблонов программных файлов. Если вы оставите этот каталог пустым, то в каталоге PL/SQL будет использован подкаталог *Template*.
- **Модули расширения.**
Этот каталог будет использован для поиска модулей расширения. Если вы оставите этот каталог пустым, то в каталоге PL/SQL будет использован подкаталог *PlugIn*.
- **Проекты.**
Определяет исходный каталог при открытии или сохранении Проектов. Если вы оставите этот каталог пустым, вы всегда будете начинать с каталога, в котором вы последний раз открывали или сохраняли Проект.
- **Стандартные запросы.**
Определяет каталог, в котором будут храниться стандартные запросы. Если вы оставите этот каталог пустым, стандартные запросы будут храниться в каталоге «Прикладные данные» вашего профиля.
- **OFS Расположения**
Определяет файл с перечнем OFS расположений (ofs.ldr). Кнопка *OFS менеджер* позволяет вам вызвать OFS менеджер для просмотра или редактирования этого файла. См. так же главу 26.1.

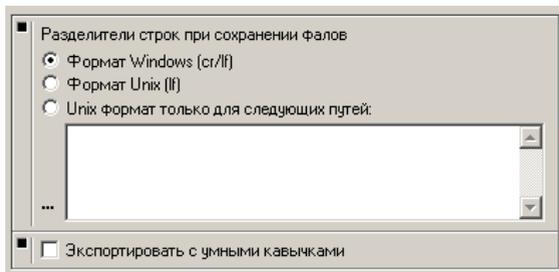
16.27 Файлы – Расширения

<input type="checkbox"/>	По умолчанию	sql
<input type="checkbox"/>	Другие	
Отчетные файлы		
<input type="checkbox"/>	По умолчанию	rep
<input type="checkbox"/>	Другие	
Тестовые скрипты		
<input type="checkbox"/>	По умолчанию	tst
<input type="checkbox"/>	Другие	
Командные файлы		
<input type="checkbox"/>	По умолчанию	pdc
<input type="checkbox"/>	Прочие	
Зарегистрировать типы файлов...		

На этой странице вы можете определить расширения по умолчанию для всех типов файлов. При сохранении или загрузке подобного файла, и если вы не указали расширение в файловом диалоге, будет принято расширение, указанное по умолчанию. Кроме того, для каждого типа файла, вы можете дополнительно указать «другие» расширения, которые будут отображены в файловых диалогах. Несколько расширений разделяйте запятыми.

Вы можете, нажав кнопку *Зарегистрировать типы файлов*, что бы связать эти расширения по умолчанию с PL/SQL Developer. Если вы дважды щёлкните на файле с таким расширением в Проводнике Windows, будет автоматически запущен PL/SQL Developer. Появится диалог, который позволит вам выбрать, какое из расширений вы хотите связать с PL/SQL Developer.

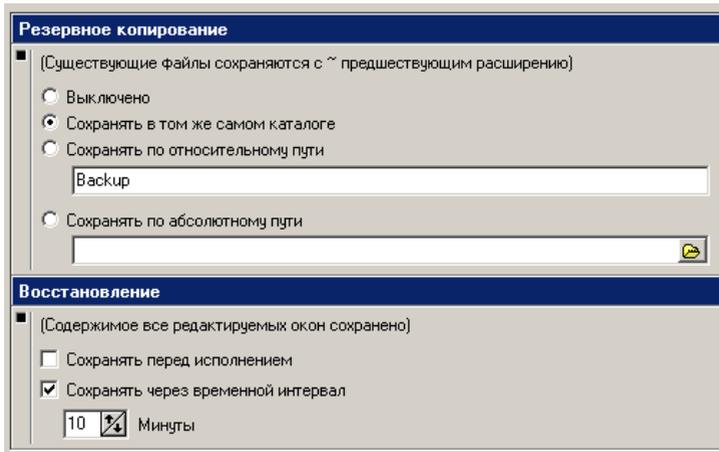
16.28 Файлы - Формат



Опция *Разделители строк* позволяет задать, как строки будут завершаться в текстовых файлах. Строка может быть завершена либо в стиле Windows парой символов CR/LF, либо в стиле Unix/Linux единственным символом LF. Вы можете дополнительно указать пути, где файлы хранятся в Unix формате, и в формате Windows иначе. Вы можете указать несколько каталогов на отдельных строках, и выбрать каталог, нажав на кнопку *Добавить каталог (...)*. Обратите внимание, что не только файлы, сохраненные в этих каталогах будут отформатированы в Unix стиле, но так же файлы, сохраненные в подкаталогах этих каталогов.

Экспорт с умными кавычками контролирует, как значения будут экспортироваться в CSV формате. Когда выключено, все значения будут заключены в кавычки. Когда включено, только те значения, которым это необходимо, будут заключены в кавычки. Это значения, содержащие запяты или кавычки.

16.29 Файлы - Резервное копирование



Эта страница предпочтений содержит опции для резервного копирования и восстановления ваших исходных файлов.

Создание файлов резервных копий

Перед тем, как PL/SQL Developer сохранит файл, оригинальный файл будет сохранён как файл резервной копии. Этот файл резервной копии имеет то же самое имя, но расширение будет начинаться с символа ~ (например, employee.pck будет сохранён как employee.~pck). Таким образом, у вас всегда будет одно предыдущее поколение файлов.

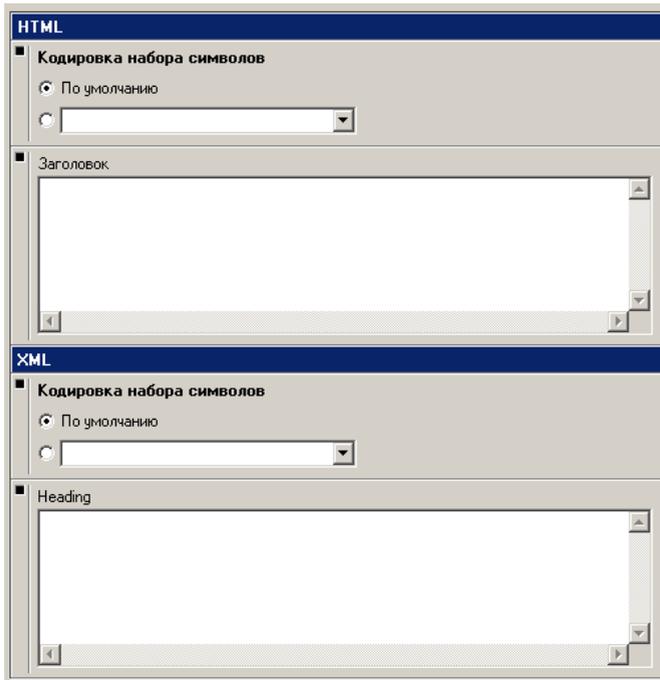
Все файловые диалоги, позволяющие вам открыть файл, будут иметь фильтр "Backup files (*.~*)", который позволит вам легко открыть файл резервной копии.

Восстановление

В некоторых случаях, PL/SQL Developer может сохранять все открытые файлы во временное место хранения. В случае некоторых проблем (например, сбоя электропитания, отказ программного обеспечения и так далее), PL/SQL Developer позволит вам восстановить эти файлы, во время его следующего запуска. Вы можете указать следующие события:

- Сохранять перед выполнением.
Перед выполнением Программного окна, Тестового окна, Командного окна, Окна SQL или Окна отчёта все открытые файлы будут сохранены во временном каталоге.
- Сохранять через временной интервал.
Через указанный временной интервал, все открытые файлы будут сохраняться во временном каталоге.

16.30 Файлы – HTML/XML



Эти предпочтения позволяют вам контролировать вывод HTML и XML, сгенерированный функциями **Окно отчёта** и **Экспорт**:

- **Кодировка символов.**
Позволяет вам определить кодировку набора символов HTML или XML документов. Это должно соответствовать набору символов вашей среды Oracle, так чтобы данные интерпретировались корректно.
- **Заголовок.**
Этот текст будет вставлен в <HEAD> секцию HTML файла или в стандартный XML заголовок файла XML.

16.31 Прочее – Печать

Общие	
<input type="checkbox"/>	Показывать диалог печати
<input type="checkbox"/>	Печатать номера строк
<input type="checkbox"/>	Переносить строки во время печати
<input type="checkbox"/>	Черно-белая печать
<input type="checkbox"/>	Печатать фон
<input checked="" type="checkbox"/>	Печатать SQL текст в SQL окне
Таблица	
<input checked="" type="checkbox"/>	Печатать линии
<input checked="" type="checkbox"/>	Инвертировать заголовки
Заголовок	
<input type="checkbox"/>	Название
<input type="checkbox"/>	Дата
Подвал	
<input type="checkbox"/>	Номера страниц

Общие

- Показывать диалог печати.
При нажатии кнопки *Печать*, сначала будет отображён диалог печати.
- Печатать номера строк.
Во время печати исходного кода SQL или PL/SQL, будут напечатаны номера строк.
- Переносить строки во время печати.
Строки, ширина которых больше, чем у окна, будут продолжены на следующей строке.
- Чёрно-белая печать.
Весь печатный вывод будет чёрно-белым. При отключении этой опции печать будет с использованием цветов или оттенков серого.
- Печатать фон.
Когда эта опция включена, будет печататься цвет фона.
- Печатать текст SQL в Окне SQL.
Определяет, печатать ли текст SQL перед результатами запроса во время печати содержимого окна SQL.

Таблица

- Печатать линии.
Определяет, печатать ли линии таблицы.
- Инвертировать заголовки.
Определяет, печатать ли заголовок таблицы инвертированным цветом.

Заголовок

- Название.
Когда эта опция включена, на каждой странице слева от заголовка будут напечатаны тип и имя файла.
- Дата.
Когда эта опция включена, на каждой странице справа от заголовка будут напечатаны текущая дата и время.

Подвал

- Номера страниц.
Когда эта опция включена, в центре подвала будет указан номер страницы.

16.32 Прочее – Обновления & Новости

Обновления

Интервал проверки онлайнных обновлений

Запуск приложения

Ежедневно

Еженедельно

Ежемесячно

Вручную

Никогда (выключено)

Проверить сейчас... Последнее обновление: 11-10-2004 14.03.2005

Новости

Интервал проверки онлайнных новостей:

Запуск приложения

Ежедневно

Еженедельно

Ежемесячно

Вручную

Никогда (выключено)

Проверить сейчас... Последние новости: 11-08-2004 29.06.2005

Настройки Rgou

Включен

Адрес Порт

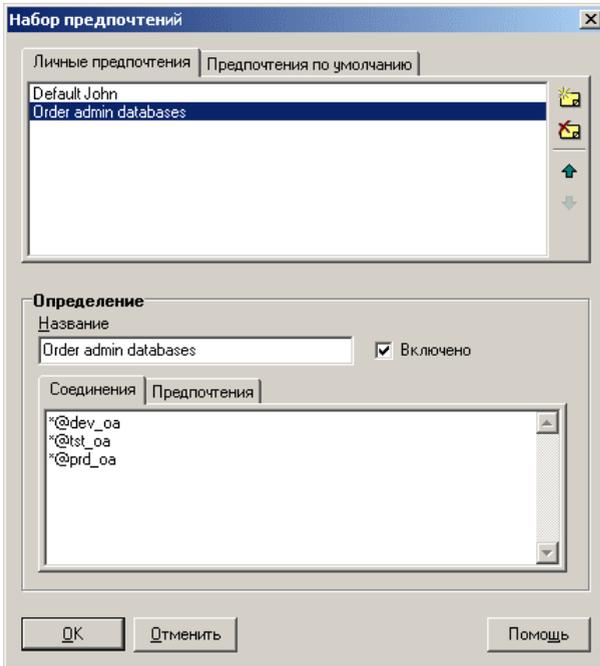
Пользователь Пароль

Спрашивать пароль

Позволит вам контролировать, будет ли PL/SQL Developer и когда проверять веб сайт Allround Automations на наличие обновлений и новостей о PL/SQL Developer.

16.33 Наборы предпочтений

Как уже упоминалось в главе 16, вы можете определять наборы предпочтений на нескольких уровнях. Чтобы определить предпочтение, нажмите рядом со списком наборов предпочтений кнопку *Конфигурировать наборы предпочтений*. Появится следующий диалог:



Личные предпочтения определены для текущего пользователя Windows (в данном случае *John*). *Предпочтения по умолчанию* могут быть определены системным администратором и предоставлены всем пользователям Windows либо на глобальном уровне, либо на уровне соединения.

Персональные предпочтения

По умолчанию, все пользователи имеют набор предпочтений *Default <user>*. Этот тот набор предпочтений, который будет доступен вам по умолчанию. Чтобы создать дополнительный набор предпочтений для одного или нескольких соединений Oracle, нажмите справа от списка предпочтений кнопку *Создать*. Теперь вы можете ввести описание и добавить одно или несколько соединений. Для каждого соединения вы можете использовать подстановочные символы, чтобы сопоставить несколько пользователей или баз данных. В приведённом выше примере, набор предпочтений *Order admin databases* определён пользователем *John* для баз данных *dev_oa*, *tst_oa* и *prd_oa*. Когда *John* подсоединяется к одной из этих баз данных, используется этот набор предпочтений. Если предпочтение не определено в этом наборе, то будет использовано соответствующее предпочтение из пользовательского набора предпочтений, указанного по умолчанию.

Предпочтения по умолчанию

Системный администратор может определить глобальные предпочтения по умолчанию для всех пользователей и соединений, и, дополнительно, может определить набор предпочтений по умолчанию для одного или нескольких соединений. Чтобы сделать это, переключитесь на страницу *Предпочтения по умолчанию* и нажмите рядом со списком предпочтений кнопку *Создать*. Если вы не введёте никаких *Соединений*, новый набор предпочтений станет для всех пользователей набором по умолчанию. Если вы введёте одно или несколько соединений, набор предпочтений станет набором по умолчанию для этих соединений.

Старшинство набора предпочтений

Если предпочтение определено на нескольких уровнях, самый специфический уровень будет иметь старшинство:

1. Личный набор предпочтений для соединения (наиболее высокий по старшинству)
2. Личный набор предпочтений пользователя Windows
3. Набор предпочтений для соединения по умолчанию
4. Глобальный набор предпочтений по умолчанию (самый низкий по старшинству)

Файлы и каталоги наборов предпочтений

Набор предпочтений хранится в отдельном файле в подкаталоге *Preferences* установочного каталога PL/SQL Developer. Внутри каталога предпочтений, персональные предпочтения хранятся в подкаталоге с именем пользователя Windows. Например:

```
C:\Program Files\PLSQL Developer\Preferences\John
```

Предпочтения, указанные по умолчанию, хранятся в подкаталоге *Default*. Например:

```
C:\Program Files\PLSQL Developer\Preferences\Default
```

Если вы устанавливаете PL/SQL Developer на файловый сервер, вы должны предоставить доступ с возможностью чтения/записи к каталогу *Preferences* и доступ с возможностью чтения к подкаталогу *Preferences\Default*. Альтернативно, вы можете использовать параметр командной строки *PREFPATH*, чтобы указать каталог для пользовательских предпочтений:

```
plsqldev.exe prefpath=u:\userdata
```

Если пользователь Windows *John* использует PL/SQL Developer, его персональные предпочтения будут храниться в *u:\userdata\john*. Предпочтения, указанные по умолчанию, всегда будут храниться в подкаталоге *Preferences\Default* внутри установочного каталога PL/SQL Developer.

Чтобы быть уверенными, что все пользователи используют одинаковый путь для предпочтений, вы можете установить этот параметр в файле *params.ini* (смотрите главу 28.4).

17. Инструменты

В меню *Инструменты* доступны несколько инструментов. Они будут описаны в этой главе.

17.1 Навигатор

Инструмент *Навигатор* детально описан в главе 15. Вы можете показать или скрыть навигатор в меню *Инструменты*, выбрав пункт *Навигатор*. Вы также можете определить из этого меню *Фильтры навигатора*, которые также описаны в главе 15.

17.2 Найти объекты баз данных

Этот инструмент позволит вам найти объекты баз данных, используя специфические критерии, такие как имя объекта, тип, владелец или статус, также как и текстовый критерий поиска для ЯОД объекта:

Имя	Тип	Скомпилирован
DEPARTMENTS	PACKAGE BODY	20.01.2005 20:58:12
DEPT	TABLE	21.07.2005 13:25:21
EMP	TABLE	20.03.2005 14:25:02
EMPLOYEE	PACKAGE BODY	12.06.2005 12:14:26

Вы можете ввести критерий поиска в 3 верхних секциях этой формы.

Поиск текста

Опции в этом диалоге более или менее схожи с диалогом “Найти” в текстовом редакторе, и позволяют вам искать независимо от регистра, искать только отдельные слова и искать, используя регулярные выражения. Если вы не будете использовать вый поиск текста, тогда будут найдены все объекты, которые удовлетворяют другим критериям.

Свойства объектов

Здесь вы можете ввести следующие критерии:

- **Фильтр навигатора** – Ограничивает объекты согласно критерию выбранного фильтра.
- **Владелец** – Владелец объекта. Вы можете использовать подстановочные символы SQL (например, sys%). Значение по умолчанию – `<CURRENT USER>`, оно ограничит поиск объектами текущего подключённого пользователя.
- **Имя** – Имя объекта. Вы можете использовать подстановочные символы SQL (например, %common).
- **Статус** – Ограничивает поиск валидными или невалидными объектами.
- **Создано после/до** – Ограничивает поиск объектами, созданными после и до указанной даты. Чтобы включить или отключить этот критерий, используйте переключатель слева от поля даты.
- **Изменено до/после** – Ограничивает поиск объектами, изменёнными после и до указанной даты. Чтобы включить или отключить этот критерий, используйте переключатель слева от поля даты.

Типы объектов

Здесь вы можете выбрать набор типов объектов, которые вы хотите найти.

Результаты поиска

После нажатия кнопки *Найти*, операция поиска будет проведена в фоновом режиме, и будет заполнен список результатов поиска.

Щёлкнув правой кнопкой мыши на результатах поиска, вы увидите всплывающее меню объекта. Первый пункт в этом меню, называется *Найти текст*, откроет исходный текст в Программном окне и переместит вас к первому местонахождению искомого текста. Также вы можете вызвать функцию *Найти текст* двойным щелчком на объекте в результатах поиска.

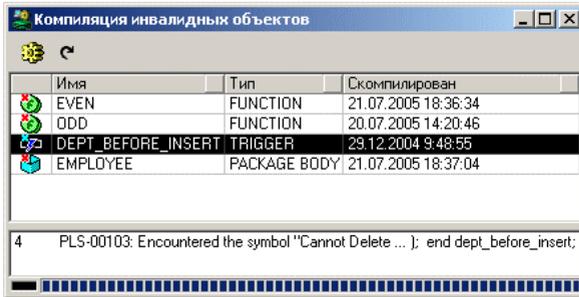
Чтобы сохранить результаты поиска, вы можете щёлкнуть на них правой кнопкой мыши и выбрать подходящий формат из подменю *Экспортировать*.

Сохранение часто используемых критериев поиска

Чтобы сохранить часто используемые критерии поиска, вы можете нажать кнопку *Сохранить* и указать каталог и имя файла. Вы можете повторно вызвать эти критерии поиска позже, нажав кнопку *Загрузить*.

17.3 Компилировать инвалидные объекты

С помощью этого инструмента вы можете легко перекомпилировать все инвалидные объекты в базе данных. Когда вы выберете его, появится диалог, в котором будут отображены все инвалидные объекты:



Заметьте, что текущий активный Фильтр навигатора контролирует, какие инвалидные объекты будут включены. Нижняя секция диалога будет отображать все ошибки компиляции, которые ранее были обнаружены для выбранного объекта.

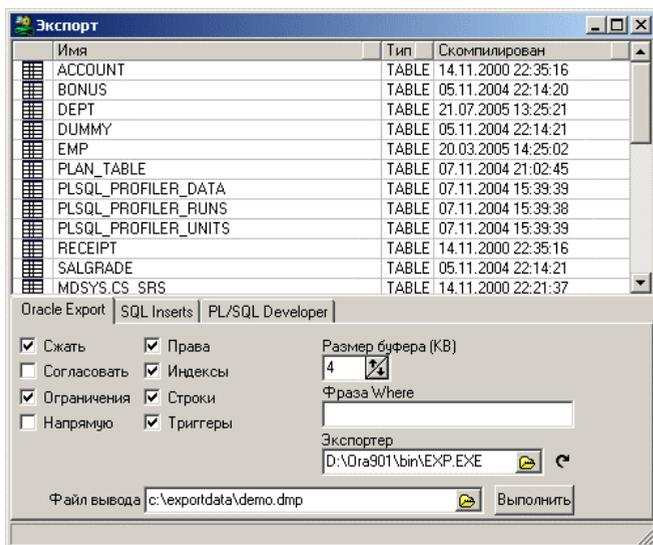
Чтобы перекомпилировать объекты, нажмите на панели инструментов кнопку *Компилировать инвалидные объекты* или *Выполнить* на главной панели инструментов. Зависимости объектов будут проанализированы, чтобы определить, в каком порядке следует проводить компиляцию. Значки слева от списка объектов будут показывать, какие объекты были скомпилированы успешно. Секция ошибок компиляции после завершения компиляции будет обновлена.

Чтобы обновить список инвалидных объектов, нажмите на панели инструментов кнопку *Обновить список объектов*. После этого список будет содержать все инвалидные на данный момент объекты.

В списке объектов вы можете щёлкнуть правой кнопкой мыши на объекте, чтобы обратиться к его всплывающему меню.

17.4 Экспортировать таблицы

Инструмент Экспортировать таблицы позволит вам экспортировать одно или несколько определенных таблиц и их данные в файл, так что вы могли позже импортировать таблицы. После запуска инструмента Экспортировать таблицы, вы можете выбрать таблицы, которые вы хотите экспортировать, выбрать метод экспорта (Oracle Export, SQL Inserts, or PL/SQL Developer) и установить различные опции, применимые к методу экспорта:



Функция экспорта анализирует зависимости внешних ключей, которые могут существовать между таблицами, и размещает таблицы в файле вывода в корректном порядке.

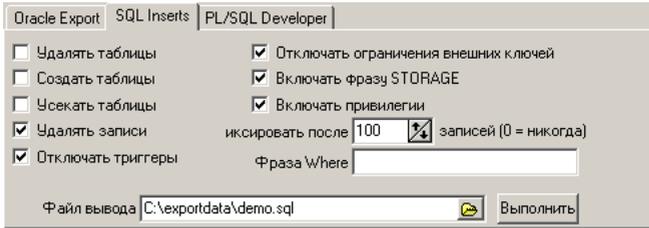
Oracle Export метод

Экранный снимок в предыдущей главе показывает опции метода Oracle Export. Этот метод позволяет указать опции, используемые в командной строке утилиты экспорта Oracle (смотрите справочник “Oracle Server Utilities”), и в дальнейшем запустит ее, чтобы выполнить экспорт. Поле *Экспортёр* позволит вам выбрать определённую версию утилиты экспорта Oracle. По умолчанию, будет использована самая свежая версия из текущего Oracle Home.

После окончания экспорта, появится новая страница *Протокол*. Эта страница протокол работы, созданный утилитой экспорта.

Преимуществами этого метода, являются скорость и совместимость файла вывода. К тому же, вы можете импортировать его при помощи утилиты импорта Oracle куда угодно.

SQL Inserts метод



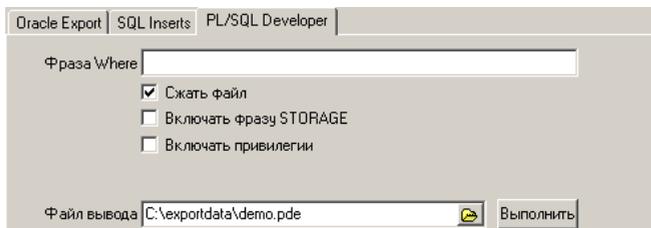
Этот метод экспорта создаёт стандартный скрипт SQL с операторами INSERT и (необязательно) операторами DDL, заново создающими таблицы. Эта страница содержит следующие опции:

- Удалять таблицы – Сгенерированный SQL скрипт содержит операторы, что бы удалить таблицы перед тем, как они будут созданы заново и загружены. В этом случае, опции *Создать таблицы*, *Усекать таблицы* и *Удалять записи* недоступны и имеют неявное значение.
- Создать таблицы – Сгенерированный SQL скрипт содержит операторы, что бы создать таблицы перед тем, как они будут загружены. Это включает ограничения, индексы и права.
- Усекать таблицы – Сгенерированный SQL скрипт будет содержать команды усечения таблиц (TRUNCATE TABLE), чтобы освободить таблицы от записей перед тем, как они будут загружены. Эта опция быстрее, чем *Delete records*, но она не может быть использована, если существуют ограничения внешних ключей.
- Удалять записи – Сгенерированный SQL скрипт будет содержать операторы DELETE, чтобы освободить таблицы от записей перед загрузкой.
- Отключать триггеры – Сгенерированный скрипт SQL отключит все триггеры таблиц перед их загрузкой, и впоследствии включит их обратно. Это может улучшить производительность, и иногда бывает необходимым, если триггеры содержат проверки, которые не применимы в процессе импорта.
- Отключать ограничения внешних ключей – Сгенерированный SQL скрипт отключит все внешние ключи таблиц перед их загрузкой, и впоследствии включит их обратно. Это может быть необходимым в случае ссылающихся самих на себя внешних ключей. Внешние ключи между разными таблицами не приведут к конфликтным ситуациям, потому что таблицы экспортируются в корректном порядке. Отключение внешних ключей также может улучшить производительность.
- Включать фразу STORAGE – Сгенерированный SQL скрипт будет включать оригинальную информацию о размещении, такую как имена табличных пространств и начальные размеры для операторов создания таблиц. Эта информация может различаться для баз данных, поэтому она не всегда может быть применимой.
- Включать привилегии – Сгенерированный SQL скрипт будет включать в себя информацию об объектных привилегиях, предоставленных другим пользователям и ролям. Очевидно, когда объекты созданы в другой базе данных, эти пользователи и роли должны существовать.
- Фиксировать после NNNN записей – Контролирует, сколько записей будет вставлено сгенерированным SQL скриптом перед исполнением команды COMMIT. Если значение равно нулю, то COMMIT будет выполнен только один раз в конце SQL скрипта. Для массивных экспортных файлов и небольших сегментов отката, будет необходимо указать соответствующее значение.

- Фраза Where – Будут экспортированы только те записи, которые удовлетворяют фразе Where. Фраза Where должно быть применима ко всем выбранным таблицам!

Преимущества этого метода заключаются в универсальности скрипта SQL и в возможности редактирования результатов при помощи текстового редактора.

PL/SQL Developer метод



Этот метод экспорта создаёт сжатый файл экспорта с табличными определениями и данными. Формат файла является внутренним форматом PL/SQL Developer. Страница содержит следующие опции:

- Фраза Where – Будут экспортированы только те записи, которые удовлетворяют фразе Where. Фраза Where должно быть применима ко всем выбранным таблицам!
- Сжать файл – При отключении этой опции сжатие файла будет отключено. Это может улучшить производительность, если данные не могут быть качественно сжаты (например, для данных типа LONG RAW или BLOB, которые уже сжаты в базе данных)
- Включать фразу STORAGE – Сгенерированный SQL скрипт будет включать оригинальную информацию о размещении, такую как имена табличных пространств и начальные размеры для операторов создания таблиц. Эта информация может различаться для баз данных, поэтому она не всегда может быть применимой.
- Включать привилегии – Сгенерированный SQL скрипт будет включать в себя информацию об объектных привилегиях, предоставленных другим пользователям и ролям. Очевидно, когда объекты созданы в другой базе данных, эти пользователи и роли должны существовать.

Преимущества этого метода являются существенное уменьшение размера файла экспорта, то, что все опции откладываются до фазы импорта, и то, что вы можете выбрать таблицы во время импорта.

17.5 Импортировать таблицы

Инструмент Импортировать таблицы позволит вам импортировать определения таблиц и данные из файла, который ранее был экспортирован при помощи инструмента Экспортировать таблицы, описанного в предыдущей главе. Точно также как и для инструмента Экспортировать таблицы, существует три метода импорта файла, каждый для своего собственного формата файла.

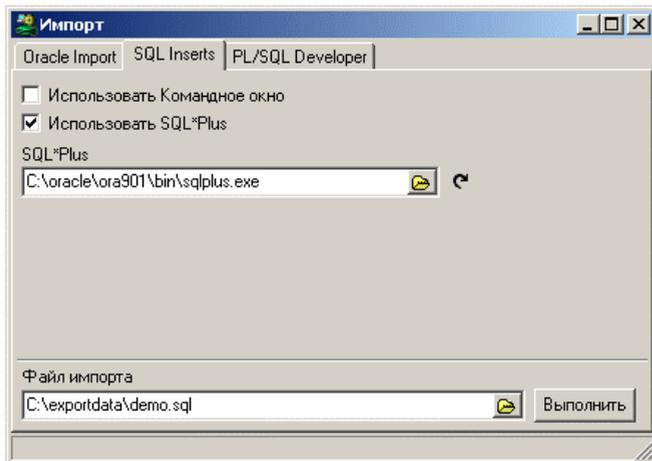
Oracle Import метод



Этот метод использует дамп файл, созданный утилитой экспорта Oracle, и запускает утилиту импорта Oracle для импорта данных. Вы можете указывать различные опции, относящиеся к опциям командной строки утилиты импорта (смотрите руководство “Oracle Server Utilities”). Поле *Импортер* позволит вам выбрать определённую версию утилиты импорта Oracle. По умолчанию, будет использована самая свежая версия из текущего Oracle Home.

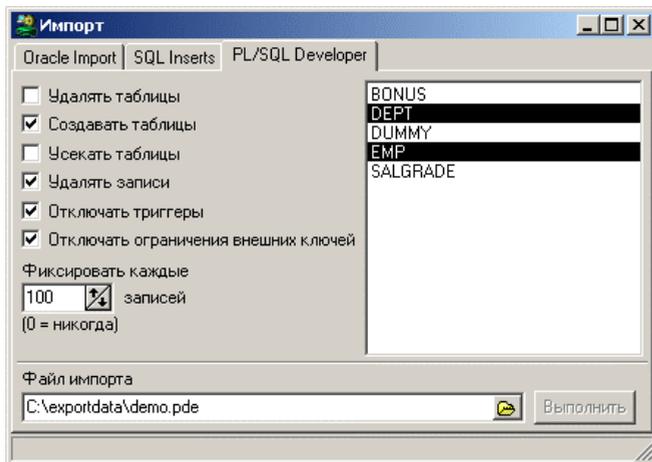
После окончания импорта, будет отражена новая страница *Протокол*. Эта страница содержит протокол работы, созданный утилитой импорта.

SQL Inserts метод



Этот метод выполняет скрипт SQL, который был сгенерирован инструмент Экспортировать Таблицы. Вы можете выбрать: запускать для выполнения скрипта SQL*Plus или использовать внутреннее Командное окно. Чтобы использовать SQL*Plus, вы можете выбрать специфический исполняемый модуль SQL*Plus. По умолчанию, будет использована самая свежая версия из текущего Oracle Home.

PL/SQL Developer метод

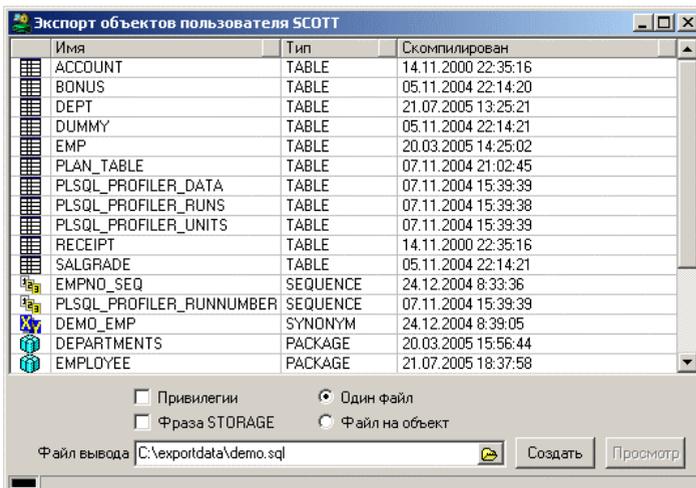


После выбора файла импорта, который был экспортирован методом PL/SQL Developer, вы можете выбрать для импорта одну или несколько таблиц из списка. Следующий опции могут быть указаны:

- Удалять таблицы – Удалять таблицы, перед тем как они будут созданы заново и загружены. В этом случае опции *Создавать таблицы*, *Усекать таблицы*, и *Удалять записи* будут отключены и имеют неявное значение.
- Создавать таблицы – Создавать таблицы, перед тем, как они будут загружены. Это включает в себя ограничения, индексы и предоставленные привилегии.
- Усекать таблицы – Усекать таблицы перед их загрузкой. Эта опция быстрее, чем *Удалять записи*, но она не может быть использована, если существуют ограничения внешних ключей.
- Удалять таблицы – Удалять все записи в таблицах перед их загрузкой.
- Отключать триггеры – Отключает все триггеры таблицы перед их загрузкой и включает их впоследствии. Это может улучшить производительность, и, иногда, может быть необходимым, если триггеры содержат проверки, не применимые к процессу импорта.
- Отключать ограничения внешнего ключа – Отключает все внешние ключи перед их загрузкой и включает их впоследствии. Это может быть необходимым для внешних ключей, ссылающихся самих на себя. Внешние ключи между различными таблицами не приведут к конфликтам между таблицами, потому что таблицы экспортируются в корректном порядке. Отключение внешних ключей также может улучшить производительность.
- Фиксировать каждые NNNN записей – Контролирует, сколько записей будет вставлено перед исполнением команды COMMIT. Если значение равно нулю, то COMMIT будет выполнен только один раз в конце SQL скрипта. Для массивных экспортных файлов и небольших сегментов отката, будет необходимо указать соответствующее значение.

17.6 Экспортировать объекты пользователя

Чтобы экспортировать операторы DDL (Data Definition Language) всех объектов пользователя, вы можете использовать инструмент *Экспортировать объекты пользователя*. Этим способом вы сможете легко пересоздать объекты для другого пользователя или в другой базе данных. После выбора в меню *Инструменты* пункта *Экспортировать объекты пользователя*, все объекты текущего пользователя будут представлены в таблице:



После выбора файла вывода, вы можете нажать кнопку *Создать*, чтобы экспортировать объекты. Если в таблице не выбраны какие-либо определённые объекты, то будут экспортированы все объекты. Вы можете выбирать объекты, щёлкая на них, и, используя клавиши *Control* и *Shift* для выбора нескольких объектов или диапазона объектов.

Файл вывода, является скриптом SQL, который совместим с Oracle SQL*Plus и с Командным окном PL/SQL Developer. После операции экспорта, вы можете нажать кнопку *Просмотр*, чтобы открыть сгенерированный файл в Командном окне.

Выбрав другого *Пользователя*, вы можете экспортировать объекты принадлежащие другому пользователю, если привилегии вашего текущего пользователя позволяют это.

Если вы хотите предоставить объектные привилегии другим пользователям и ролям, то можете включить опцию *Привилегии*. Очевидно, когда объекты будут создаваться в другой базе данных, эти пользователи и роли должны существовать.

Аналогично, вы можете включить опцию *Фраза STORAGE*, чтобы включить информацию о размещении, такую как имена табличных пространств и первоначальные размеры. Для разных баз данных эта информация может различаться, поэтому она не всегда может быть подходящей.

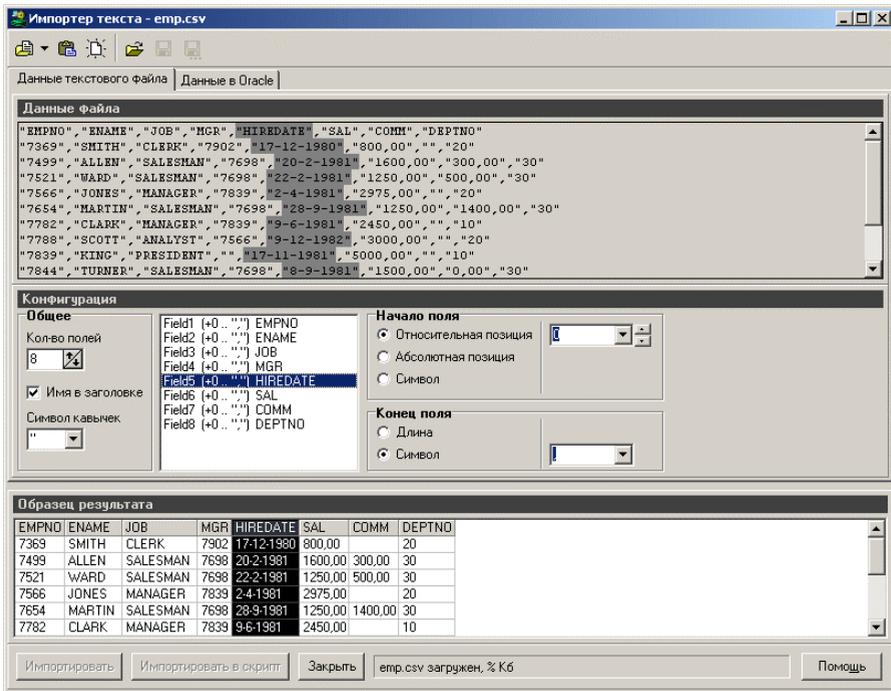
Опция *Включить пользователя* указывает, будут ли имена объектов начинаться с имени пользователя (например, SCOTT.EMP вместо EMP) в результирующем выходном файле.

Опции *Один файл* и *Файл на объект* указывают, либо один SQL скрипт, содержащий DDL, будет создан для всех объектов, либо каждый объект будет экспортирован в отдельный файл (с именем объекта в качестве имени файла, и .sql в качестве расширения). В этом случае файл вывода будет содержать последовательность обращений к этим отдельным объектным файлам. Все файлы будут записаны в каталог файла вывода.

17.7 Импортёр текста

При помощи импортёра текста вы можете импортировать файлы ASCII в базу данных. Поддерживаются большинство строкоориентированных форматов, такие как с полями разделёнными запятыми или знаками табуляции. Импортёр автоматически попытается определить формат файла, так что в большинстве случаев вам не нужно будет ничего определять, а только выбрать файлы, выбрать таблицы и всё.

Когда вы откроете импортёр текста и загрузите текстовый файл, вы получите примерно следующее:



Импортер загружает первые 100 строк из файла, эти исходные данные отображаются в верхней секции “Данные файла”. Средняя секция позволит вам создавать определение текстового файла. Нижняя секция “Образец результат” имеет таблицу с данными в том виде, в котором они будут импортированы.

Панель инструментов содержит кнопки для выбора текстового файла или вставки текста из буфера обмена. Файлы могут быть виртуально неограниченного размера, так как они не читаются в памяти целиком. Кнопки открытия и сохранения определения позволяют вам использовать определение снова.

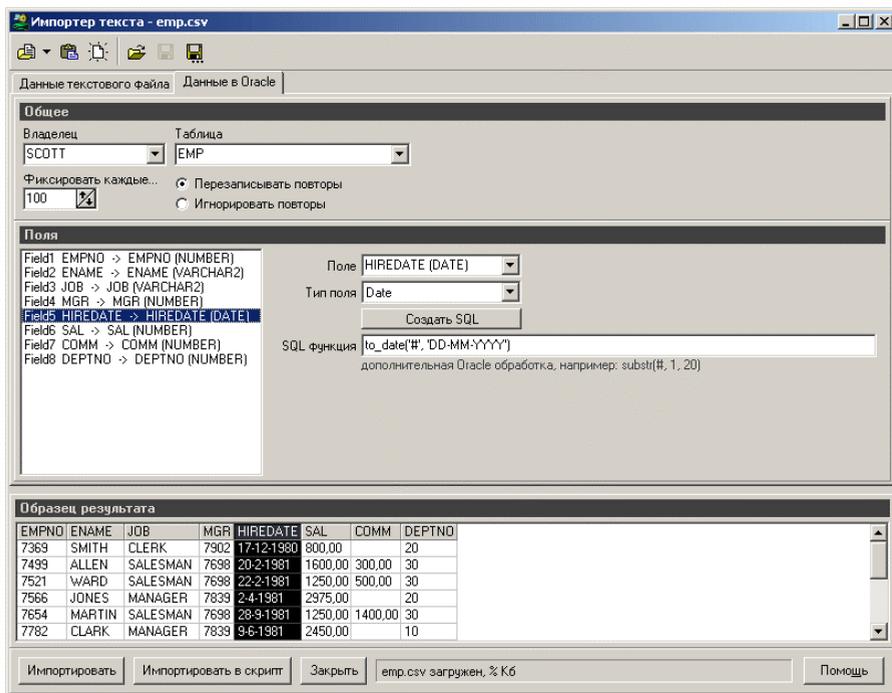
Как только вы загрузили файл, конфигурация определяется автоматически, но, если она не корректна, вы можете сами создать или изменить её. Установите корректное число для “Кол-во полей” и выберите поле из списка полей, чтобы задать его определение. В секции “Данные файла” данные выбранного поля будут выделены, позволяя вам тем самым проверить корректность определения поля. Чтобы сконфигурировать импорт текста, у вас есть следующие опции:

- **Общее – Кол-во полей**
Число полей в записи.
- **Общее – Имя в заголовке**
Показывает, содержит ли первая запись имена полей.
- **Общее – Символ кавычек**
Показывает строковые символы, обычно двойные или одиночные кавычки.
- **Начало поля – Относительная позиция**
Поле начинается с позиции, указанной относительно окончания предыдущего поля. 0 указывает на то, что поле начинается там, где заканчивается предыдущее поле.

- Начало поля – Абсолютная позиция
Показывает, что поле начинается с фиксированной позиции.
- Начало поля – Символ
Показывает, что поле начинается с конкретного символа.
- Конец поля – Длина
Показывает, что поле имеет указанную фиксированную длину.
- Конец поля – Символ
Показывает, что поле оканчивается конкретным символом.

Для стандартных разделённых запятыми файлов (csv) все поля имеют относительную начальную позицию 0, и символом окончания поля должна быть запятая. Окончание строки (сг/lf или lf, или сг) также будут показывать окончание поля, так что, для последнего поля в строке, символ окончания не имеет значения.

Когда у вас имеется корректное определение файла, вам нужно выбрать таблицу Oracle и указать какие текстовые поля должны быть импортированы и в какие Oracle поля. Вы можете это сделать на второй странице:



Верхняя секция имеет несколько важных общих параметров. Секция “Поля” позволит вам связать поля текстового файла и поля Oracle. Нижняя секция служит для предварительного просмотра результатов.

Вы можете установить следующие общие предпочтения:

- Таблица
Таблица Oracle (или представление), в которую вы хотите импортировать данные. После выбора таблицы, импортер попытается автоматически определить поля и их типы. Это будет работать только в том случае, если текстовый файл имеет заголовок.

- Фиксировать каждые...
Показывает, после какого количества записей вы хотите фиксировать транзакцию. Если вы установите для этого предпочтения 0, все данные будут зафиксированы после полного окончания импорта.
- Перезаписывать повторы
Дублирующиеся записи будут обновлены в базе данных
- Игнорировать повторы
Дублирующиеся записи будут проигнорированы.

В секции “Поля” вы видите список с полями из текстового файла. Для каждого поля вы можете установить следующее:

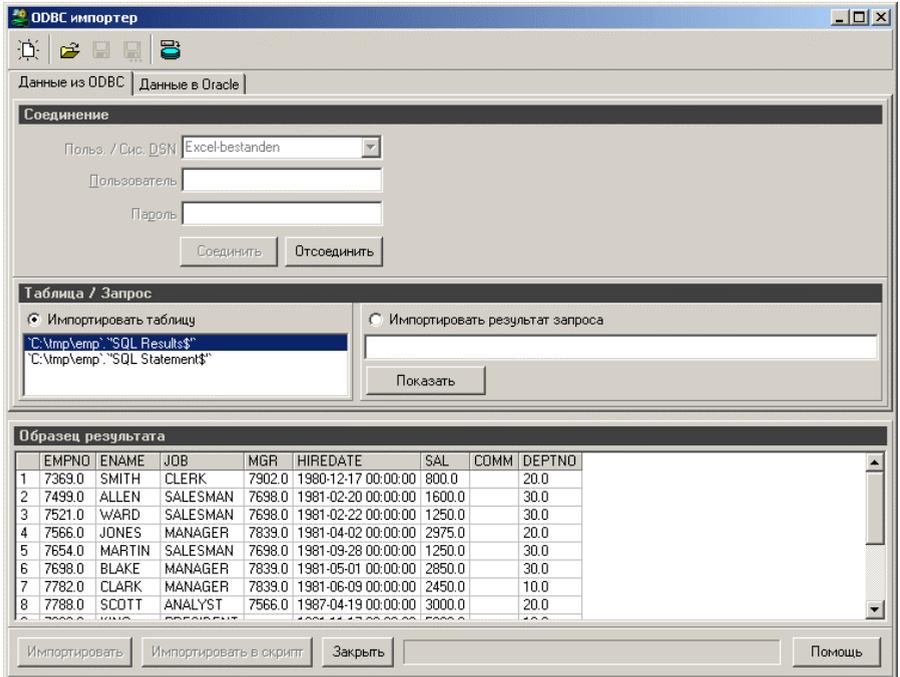
- Поле
Поле Oracle, которое вы хотите связать с текстовым полем. Если вы не хотите импортировать это поле, вы можете оставить эту опцию пустой.
- Тип поля
Основные типы полей: String, Number или Date.
- SQL функция
Эта опция позволит вам определить дополнительную SQL обработку. Для полей дат функция to_date добавляется автоматически. Это очень мощная опция, позволяющая вам конвертировать импортируемые данные. Вы можете ввести всё, что может быть обработано Oracle. Чтобы указать данные, вы можете добавить знак #. Для полей дат кнопка “Create SQL” заполнит это поле функцией to_date.

Когда определение завершено, вы можете сохранить его. На панели инструментов имеются кнопки сохранения и загрузки файлов определений. Текстовый импортёр запомнит, какой файл определения был использован для какого текстового файла, и в следующий раз, когда вы откроете этот текстовый файл, будет автоматически загружен файл определения.

У вас имеются две кнопки импорта – “Импортировать” и “Импортировать в скрипт”. Первая кнопка начнёт импорт данных в выбранную таблицу. Вторая кнопка создаст SQL скрипт с операторами insert.

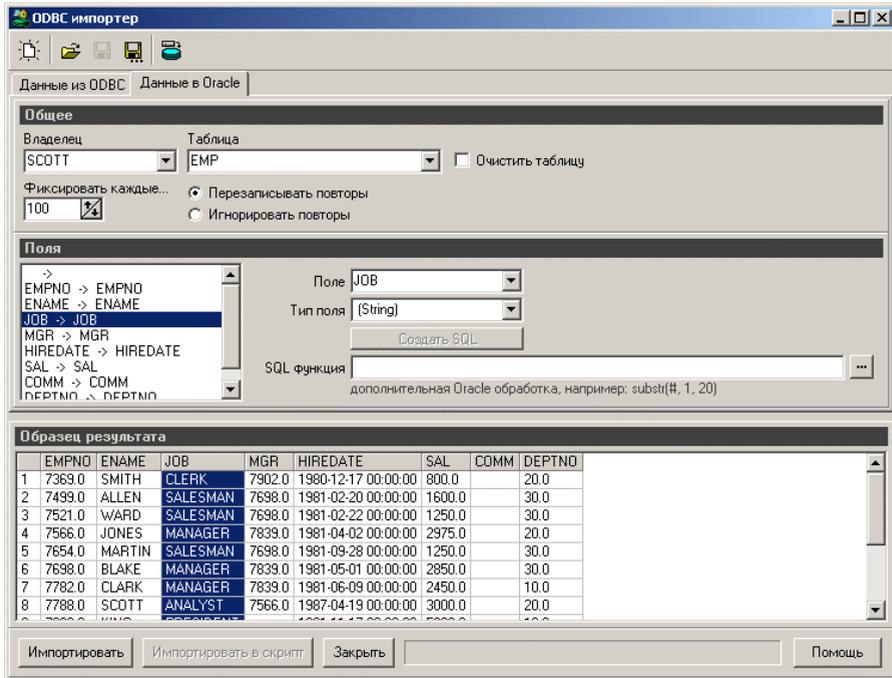
17.8 ODBC импортер

The ODBC Importer tool allows you to import data from any ODBC data source into an Oracle table: ODBC импортер позволяет вам импортировать данные из любого ODBC источника данных в Oracle таблицу:



На первой странице ODBC импортера вам необходимо выбрать ODBC источник данных, и, если необходимо, задать имя пользователя и пароль для этого источника данных. Далее вы можете соединиться, после чего список таблиц будет заполнен именами всех таблиц источника данных. Выбрав таблицу, *Предварительный просмотр результата* будет заполнен данными. Вместо выбора таблицы, вы можете так же задать запрос, выбрав опцию *Импортировать результат запроса* и введя текст запроса. Это может быть полезно, если вы хотите ограничить записи или колонки, или если необходимы данные из нескольких таблиц.

Задав спецификацию источника, вы можете переключиться на страницу *Данные в Oracle*, что бы выбрать целевую таблицу и указать соответствие колонок:



В разделе *Общие* вы можете выбрать владельца и таблицу, в которую записи должны быть импортированы. Вы можете так же выбрать, хотите ли вы очистить таблицу перед импортом, выбрать интервал сохранения (0 = сохранить в конце), и необходимо ли игнорировать дублирующиеся записи или перезаписывать их.

В разделе *Поля* вы можете задать соответствие полей ODBC источника данных (слева) полям Oracle таблицы (справа). Для каждой колонки вы можете дополнительно указать SQL функцию. Например, если вы хотите преобразовать строковое поле в верхний регистр, введите *upper(#)*. Знак решетки будет заменен данными поля для каждой записи, и результирующее выражение будет подставлено в INSERT команду.

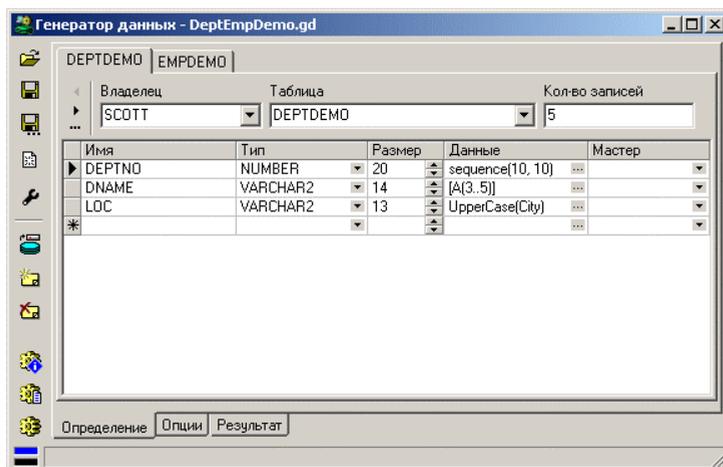
После задания соответствия полей, вы можете нажать кнопку *Импорт*, что бы импортировать данные в Oracle таблицу. Если вы нажмете клавишу *Ctrl* перед тем как нажать эту кнопку, будет выведен пример INSERT команды, так что вы сможете проверить соответствие полей и указанные вами SQL функции.

Что бы позже повторно использовать это определение импорта, вы можете нажать кнопку *Сохранить определение* панели инструментов. Это определение может быть открыто позже, если вы нажмете на кнопку *Открыть определение*.

17.9 Генератор данных

Генератор данных позволит вам создавать демонстрационные и тестовые данные. Это может быть очень полезным, если вы хотите протестировать приложения и посмотреть, как они будут работать с большими объемами данных. Вы найдёте Генератор данных в меню Инструменты.

В основном, определение состоит из одной или нескольких таблиц, некоторого количества записей, которые вы хотите сгенерировать и определений полей данных. Кнопки наверху слева позволяют вам открыть и сохранить определение.



Выше вы видите демонстрационную программу deptemp, которая создаёт данные для таблиц deptdemo и empdemo, схожим с хорошо известными нам таблицами dept и emp. На этой странице вы найдёте следующие опции:

- Таблица – Имя таблицы.
- Кол-во записей – Число записей, которые вы хотите сгенерировать. Это может быть не только число, но и диапазон, например 10..100.
- Имя – Имя поля.
- Тип – Тип данных поля.
- Размер – Размер поля (когда необходимо). Для числовых полей это будет масштаб, точность.
- Данные – Определение данных для поля (смотрите ниже).
- Мастер – Если эта таблица является детальной для другой таблицы, вы можете установить мастер таблицу. Для каждой мастер записи будет сгенерировано указанное число детальных записей.

Чтобы добавить существующую таблицу, вы можете использовать кнопку “Добавить существующую таблицу”. Вы также можете перетащить & бросить таблицу из Объектного навигатора.

Определение данных

Определение данных определяет сгенерированные данные. Если вы хотите создать простые символы, вы можете ввести определения символов в квадратных скобках: [data]

Данные могут представлять собой сочетание следующих предопределённых наборов:

- a: a..z (символы нижнего регистра)

- A: A..Z (символы верхнего регистра)
- @: A..Z and a..z (все символы)
- #: A..Z and a..z and 0..9 (все символы и числа)
- *: #33..#126 (все символы ASCII)
- 0: 0..9 (все числа)
- 1: 1.. (все числа кроме 0)
- 9: 0..9 (все числа)

Например:

[Aaa00] чтобы сгенерировать строковые данные, такие как: Gxe21, Lly05 и т. д.

Вы также можете добавлять текст в одиночных кавычках.

Например:

[AA '-' 1000] чтобы сгенерировать строковые данные, такие как: CX-4903, SY-1044 и т. д.

Пробелы будут игнорироваться в определении, если только они не заключены в кавычки.

Если вы хотите, чтобы символ был повторён некоторое количество раз, вы можете добавить это число в скобках после символа (n). Вы также можете добавить диапазон чисел (min..max).

Например:

[Aa(5..15) '-' Aa(8..20)] даст результат, такой как: "Masfae Qwwecdsadif"

Литеральный текст в определении не требуется заключать в скобки. Другими словами, ['hello'] и 'hello' эквивалентны друг другу. Текст, введённый без кавычек, интерпретируется как функция.

Доступны несколько специальных функций:

- Signal(Min, Max, Delta, Noise)
Возвращает данные технических измерений (например, температуры). Min и Max определяют диапазон, Delta – максимальное изменение, вы также можете добавить некоторую погрешность. Пример: Signal(-10, 20, 0.1, 0.1)
- Random([Min], Max)
Возвращает случайные числа между Min и Max. Если указан только Max, для Min будет установлено значение 0. Для полей дат вы можете вводить даты для Min и Max.
- Sequence(Start, [Inc], [WithinParent])
Возвращает упорядоченные числа, начиная со Start и увеличивая на Inc (которое по умолчанию равно 1). Для детальной таблицы вы можете дополнительно указать ключевое слово WithinParent, чтобы показать, что для каждой родительской записи последовательность должна быть сброшена в начальное значение.
- List('item'(weight), 'item'(weight), ...)
Возвращает элемент случайно выбранный из списка. Чтобы некоторые элементы появлялись чаще, чем остальные, в скобках можно добавить вес.
Например: List('CLERK'(50), 'SALESMAN'(30), 'MANAGER'(10))
- List(select statement)
Такая же, как и предыдущая функция List, но элементы возвращаются при помощи SQL оператора SELECT.
- Text([Style], MaxCharacters, [WordsPerLine, LinesPerParagraph])
Эта функция возвращает текст. Опциональный параметр Style может принимать значения LoremIpsum (по умолчанию), English, German или Japanese. Он также может представлять собой набор символов, например, [aA], в этом случае, слова генерируются из указанного набора.
MaxCharacters определяет максимальный размер сгенерированного текста, а опции WordsPerLine и LinesPerParagraph определяют размеры строки и абзаца. Указанный размер может быть либо числом, либо диапазоном (min..max).

- `File(path, path, ...)`
Случайным образом выбирает файл из заданных путей и вставляет его содержимое. Эта опция позволит вам загрузить двоичные данные (такие как изображения) в базу данных. Путь может содержать подстановочные символы, например, `d:\images*.bmp`.

Вы можете изменять текстовые результаты предыдущих функций, используя следующие функции:

- `Uppercase()`
- `LowerCase()`
- `InitCaps()`

Например: `InitCaps(List(select ename from emp))`

Также, для генерирования более или менее реальных данных, существует несколько предопределённых наборов данных. Вы можете использовать следующие определения:

- `Firstname` – Основной список имён
- `Lastname` – Основной список фамилий
- `Company` – Названия компаний (случайный список существующих компаний)
- `Address1` – Адресная строка 1
- `Address2` – Адресная строка 2
- `ZIP` – код ZIP
- `City` – Город
- `State` – Штат
- `Country` – Страна
- `Email` (относится к `Firstname`, `Lastname` и `Country`)

Также доступны несколько примеров определённых пользователем наборов данных

- `Components.Code` – General article items: Article code
- `Components.Description` – Article description (Computer parts)
- `Components.Price` – Article price
- `Elements.Name` – Chemical elements (name)
- `Elements.Symbol` – Chemical elements (symbol)

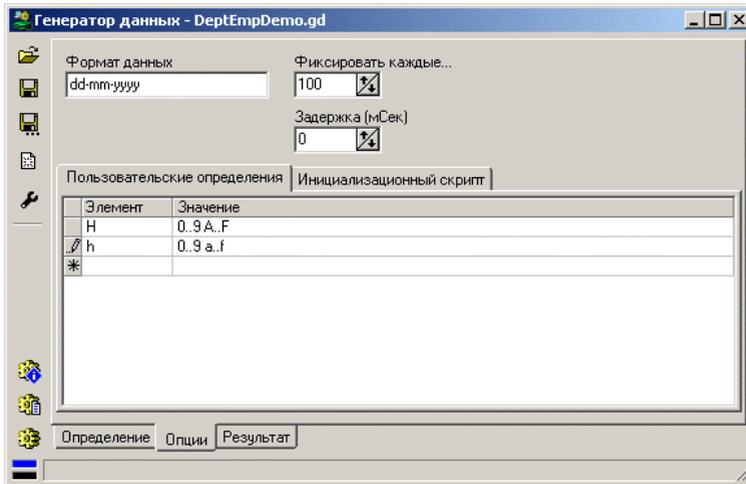
Эти наборы данных могут быть найдены в каталоге `DataGenerator\UserData` как `elements.txt` и `components.txt`. Если вы хотите, вы можете добавить свои собственные наборы. Просто добавьте файл, в котором первая строка содержит имена полей, разделенных запятыми и заключенных в квадратные скобки. Вы можете использовать данные из вашего файла, указав имя файла.расширение, как в двух предыдущих примерах.

Все упомянутые выше функции и данные могут быть добавлены вместе, например: `Random(10..99) + '-' + [A(4)]`

Знак `+` является опциональным, но там должен быть как минимум пробел в качестве разделителя.

Опции

Страница опций позволит вам устанавливать некоторые предпочтения, относящиеся к определению генератора данных. Вы можете указывать формат даты (например, как для функции Random). Вы также можете указать, после какого количества записей вы хотите зафиксировать транзакцию (установите 0, чтобы зафиксировать транзакцию после окончания генерации). Задержка полезна только при некоторых тестах в режиме реального времени, где вы хотите вставлять данные с определённой скоростью.



Секция “Пользовательские определения” позволит вам определять или переопределять символьные наборы в определении [data]. Элемент всегда является одиночным символом, а Значение представляет собой один или несколько разделённых пробелами диапазонов символов, таких как: A..Z a..z 0..9 #200..#220. Например, чтобы определить шестнадцатеричный символьный набор, вы можете указать H, как элемент, и 0..9 A..F – как значение.

Секция “Инициализационный скрипт” позволит вам указывать операторы SQL, которые следует выполнить перед тем, как данные будут вставлены в базу данных. Обычно, это будет оператор создания таблицы, или усечения таблицы, или выбора сегмента таблицы и т. д. Несколько операторов следует разделять точками с запятой.

Общие опции, которые будут использованы по умолчанию для всех генераторов данных, могут быть определены нажатием кнопки *Общие предпочтения*, находящейся слева. Обратитесь к главе 16.23 за дополнительной информацией. Если соответствующее предпочтение в определении генератора данных также установлено, то оно будет иметь приоритет.

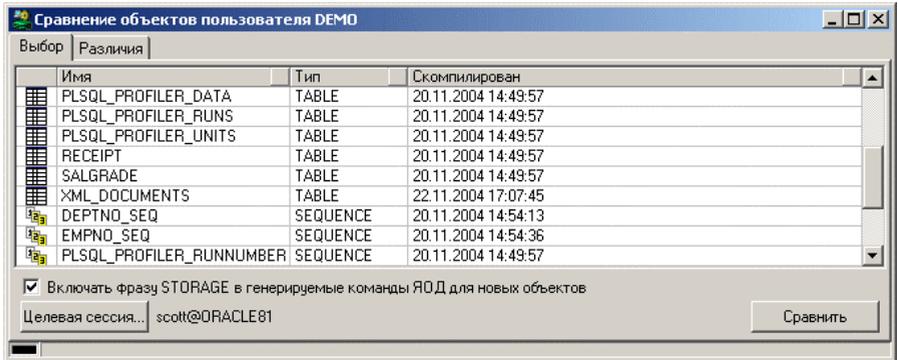
Генерирование данных

Для генерирования фактических данных слева внизу доступны три кнопки:

- Начать тестовый прогон – При помощи этой кнопки данные будут сгенерированы, и результаты будут отображены в таблице на странице *Результат*. Щелчком правой кнопки мыши на таблице, вы сможете экспортировать эти результаты в различных форматах.
- Создать данные в виде SQL – Генерирует данные как SQL скрипт. Это будет работать, только если вы не добавляете данные из файлов, используя функцию File.
- Создать данные в БД – Генерирует данные в базу данных Oracle.

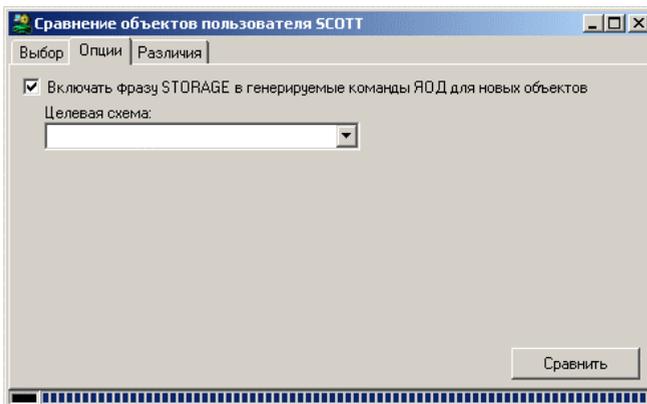
17.10 Сравнить объекты пользователя

После внесения изменений в табличные определения, представления, программные модули и т. д., может быть полезным, распространить эти изменения на других пользователей базы данных. Это может быть другая среда разработки или тестирования. Чтобы сравнить объекты пользователя, где ведется разработка, с объектами другого пользователя, вы можете использовать функцию *Сравнить объекты пользователя* из меню *Инструменты*. При этом появится следующий диалог:



На странице *Выбор* вы можете выбрать объекты, которые вы хотите сравнить. После выбора этих объектов, вы можете нажать кнопку *Целевая сессия*, чтобы выбрать пользователя и базу данных, которые вы хотите сравнить. После этого будет включена кнопка *Сравнить*, нажав которую, начнется операция сравнения.

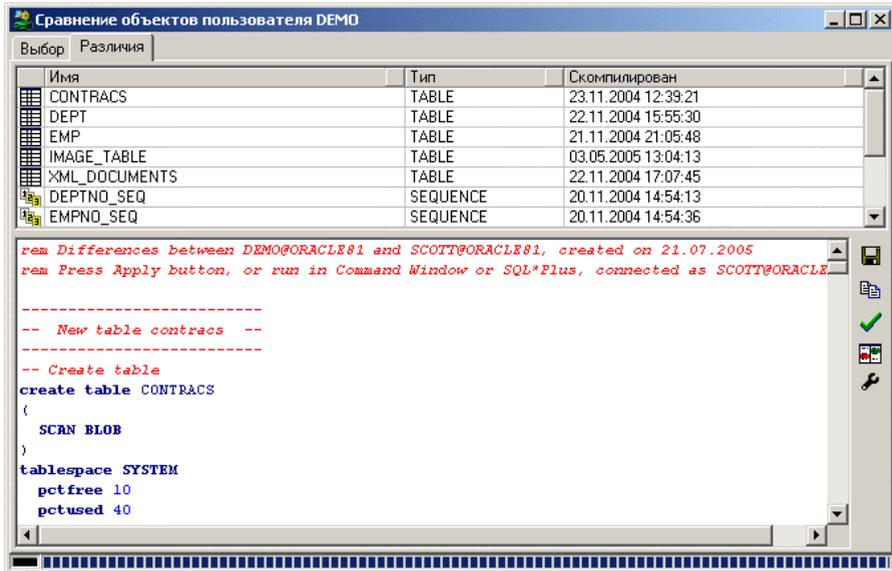
На странице *Опции* вы можете задать следующие опции:



Вы можете выбрать опцию *Включать фразу STORAGE...*, чтобы включить для новых объектов информацию о размещении, такую как имена табличных пространств и первоначальные размеры. Эта информация может различаться для разных баз данных, поэтому она не всегда может быть корректной.

Целевая схема может быть задана, если объекты в целевой сессии принадлежат пользователю отличному, от целевого пользователя, который присоединен в данный момент. Когда операция

сравнения будет завершена, диалог переключится на страницу *Различия*, которая отобразит список всех различающихся объектов:



Объекты этого списка отсортированы в порядке зависимости. Под списком различающихся объектов целевого пользователя, вы увидите SQL команды, которые необходимо выполнить, чтобы эти объекты стали идентичными соответствующим объектам текущего пользователя. Если не выбрано ни одного объекта, то будут отображены SQL команды для всех объектов. Если выбран один или несколько объектов, SQL команды будут отображены только для них. В приведённом выше примере, отсутствующее проверочное ограничение *EMP_MGR_EMPNO* было добавлено в таблицу *EMP*.

Кнопка *Показать различия*, строка за строкой, покажет отличия между старым и новым исходным текстом объекта. Это может быть полезным, чтобы просмотреть изменения, сделанные в Программных модулях, или поможет вам определить, почему некоторый оператор DDL был сгенерирован для других типов объектов. *Конфигурировать внешнее средство сравнения* позволит вам сконфигурировать инструмент сравнения, который должен применяться. По умолчанию будет использоваться утилита *ExamDiff*, для которой существует так же версия Pro (смотрите в меню *Info* утилиты *ExamDiff* пункт *About*). Смотрите также главу 16.22.

Теперь, чтобы выполнить этот SQL в целевой сессии, вы можете нажать кнопку *Применить SQL в целевой сессии*. Альтернативно, вы можете сохранить SQL в файл, нажав кнопку *Сохранить SQL в файл*, или же вы можете скопировать его в буфер обмена, нажав кнопку *Копировать SQL в буфер обмена*.

При сравнении объектов, игнорируются следующие свойства:

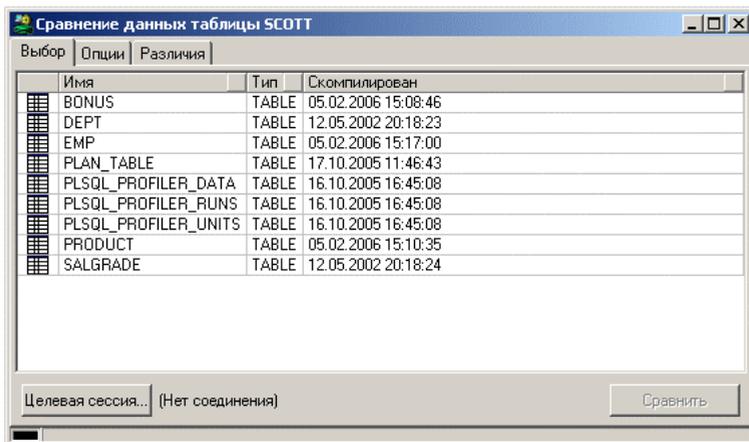
- Параметры хранения – Такие свойства таблиц и индексов как *next extent* и *pct free* не участвуют в сравнении.
- Ограничения со сгенерированными системой именами – Эти ограничения для двух пользователей будут иметь разные имена, поэтому они не могут быть сравнены. Однако если таблица является новой для целевой сессии, эти ограничения будут сгенерированы.

- Свойства создания таблицы – Свойства, которые потребуют пересоздания таблицы, будут проигнорированы.
- Данные таблицы – Чтобы экспортировать данные, используйте функцию *Экспортировать таблицы* (смотрите главу 17.4).
- Значения последовательностей – Текущее значение последовательности приравнивается к данным.

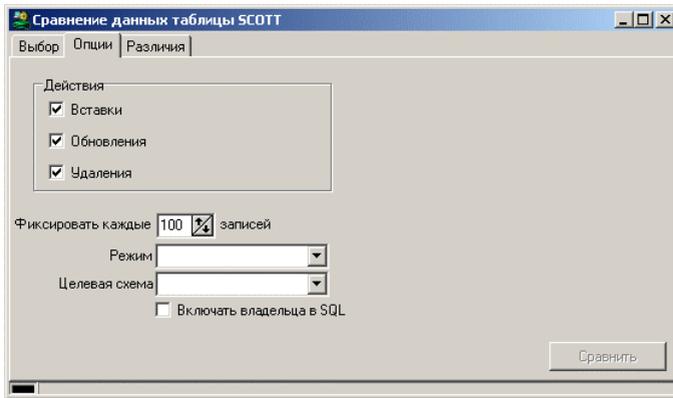
17.11 Сравнить данные таблицы

После вставки, изменения или удаления записей в одной или более таблицах в течении разработки, может быть полезно перенести эти изменения такой же таблице, но принадлежащей другому пользователю. Это может быть перенос из базы данных разработки в тестовую базу данных, или в рабочую базу данных или передача другому разработчику проекта.

Что бы выполнить это, вы можете использовать инструмент *Сравнить данные таблицы* из меню *Инструменты*. Это отобразит следующий диалог:



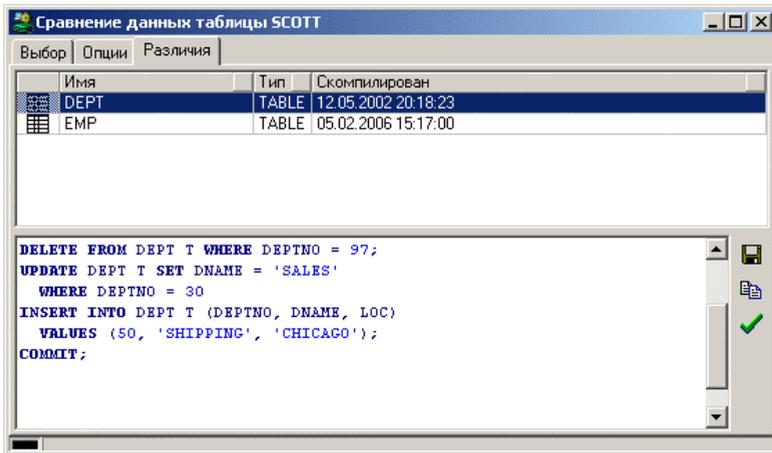
На странице *Выбор* вы можете выбрать таблицы, которые вы хотите сравнить. После это, вы можете нажать на кнопку *Целевая сессия...*, что бы задать пользователя и базу данных, которые вы хотите сравнить. Это делает доступной кнопку *Сравнить*, которую вы можете нажать, что бы запустить операцию сравнения. Но вы так же можете сначала перейти на страницу *Опции*:



На этой странице вы указываете различные опции процесса сравнения.

- **Действия** – Вы можете выбрать те различные действия, которые будут выполняться в целевой сессии. Если, например, вы опустите удаление, будут выполняться только вставки и обновления.
- **Фиксировать каждые N записей** – Показывает, после какого количества записей вы хотите фиксировать транзакцию. Если вы установите для этого предпочтения 0, все данные будут зафиксированы после полного окончания процесса.
- **Режим** – Выберите Генерировать скрипт, что бы сгенерировать SQL скрипт, включающий команды вставки, обновления, удаления и фиксации транзакции. Вы можете сохранить этот скрипт позже и запустить его в Командном окне или в SQL*Plus. Выберите Обновлять базу данных, что бы сразу же переносить изменения в целевую сессию.
- **Целевая схема** – Если таблицы расположены в схеме отличной от пользователя целевой сессии, вы можете здесь указать эту схему. Пользователю целевой сессии необходимы права на таблицы, выбранные в целевой схеме.
- **Включать владельца в SQL** – Эта опция применима только в режиме генерация скрипта, она указывает, начинать ли имена таблиц в сгенерированном SQL с имени пользователя. Позже это позволит вам соединиться под другим пользователем, когда вы будете выполнять скрипт, и перенести изменения в необходимую схему.

После выбора необходимых опций, вы можете нажать кнопку *Сравнить*, что бы запустить процесс сравнения. По завершению *Различия* будут отображены:



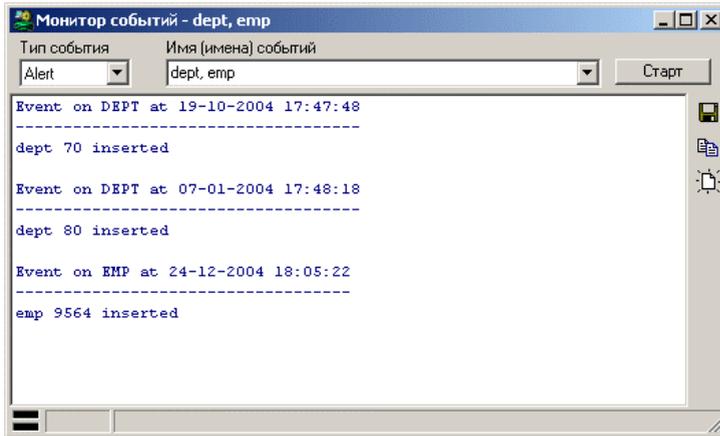
В режиме обновления, эта страница будет отображать количество записей вставленных, обновленных и удаленных в каждой таблице.

В режиме скрипта, вы можете просмотреть все различия и сохранить результирующий скрипт в файле, нажав на кнопку *Сохранить SQL* справа. Заметьте, что вы можете ограничить набор таблиц в скрипте, выбрав необходимые таблицы на панели сверху. Вы можете так же скопировать SQL в буфер обмена или применить выбранные изменения в целевой сессии, нажав соответствующие кнопки справа.

17.12 Монитор событий

Монитор событий позволит вам перехватывать и просматривать сообщения `dbms_pipe` или сигналы `dbms_alert`, сгенерированные программными модулями в других сессиях. Просто укажите *Тип события* (Сигнал или Канал), *Имя (имена)* сигнала или канала и нажмите кнопку *Старт*.

На следующем экранном снимке изображён Монитор событий, который ожидает сигнал с именем 'DEPT' или 'EMP':



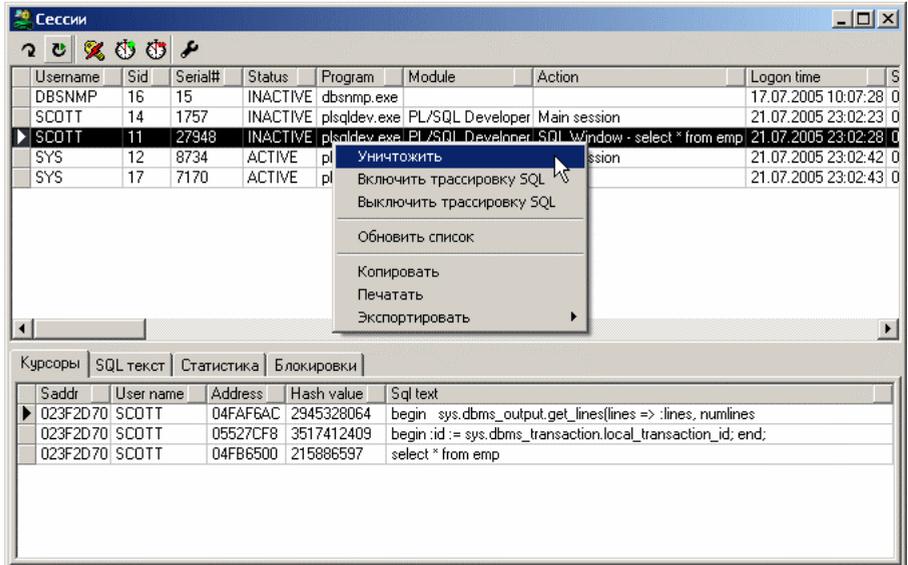
Когда такое событие наступает, имя канала или сигнала и временная метка записываются в страницу вывода, сопровождаемую информацией о событии:

- Канал: отображаются все элементы сообщения
- Сигнал: отображается значение параметра `message` при вызове `dbms_alert.signal`

Для событий сигналов вы можете указать несколько имён, разделённых двоеточиями. Для событий каналов, вы можете указать только одно имя канала.

17.13 Сессии

Инструмент *Сессии* отображает все сессии в экземпляре базы данных, к которому вы сейчас подсоединены:



В верхней части окна вы увидите информацию о сессиях, таких как имя пользователя, SID и серийный номер (который идентифицирует сессию), статус и т. д. Вы можете отсортировать объекты в списке сессий, щёлкнув на кнопку в заголовке столбца. Например, чтобы быстро найти все активные сессии, щёлкните на кнопке в заголовке столбца статуса.

Вы можете так же выбрать *Фильтр сессий* на панели инструментов, что бы ограничить список сессий или ограничить набор отображаемых колонок. Фильтр так же может определять порядок сессий и колонок.

Если вы щёлкните правой кнопкой мыши на сессии, появится всплывающее меню со следующими пунктами:

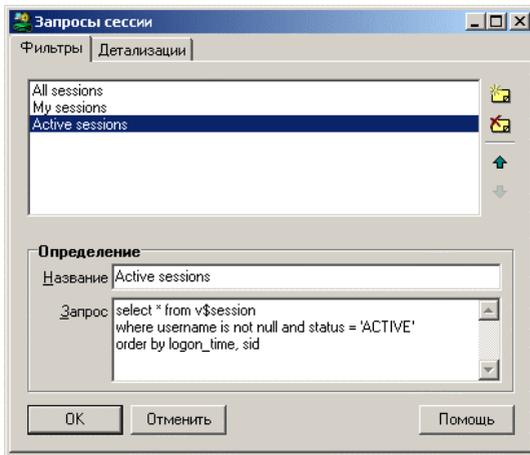
- Уничтожить – Позволит вам отключить выбранную сессию.
- Включить трассировку SQL – Включает для выбранной сессии трассировку SQL. Эта опция будет отключена, если ваша версия Oracle Server не поддерживает эту функцию, или если у вас нет привилегий на пакет `dbms_system`.
- Отключить трассировку SQL – Отключает для выбранной сессии трассировку SQL.
- Обновить список – Обновляет список сессий.
- Копировать – Копирует список сессий в буфер обмена.
- Печатать – Печатает список сессий.
- Экспортировать – Экспортирует список сессий в формате CSV, TSV HTML или XML.

Функции Обновить, Уничтожить, Включить трассировку SQL и Выключить трассировку SQL также доступны через панель инструментов в верхней части окна. Дополнительно, панель

инструментов включает кнопку *Таймер периодического обновления*. Когда она нажата, список сессий и сессионная информация будут периодически обновляться. Вы можете определить период обновления, щёлкнув на ней правой кнопкой мыши.

Нижняя половина окна содержит несколько страниц с информацией о выбранной сессии. Информация на этих страницах может быть обновлена, скопирована или напечатана, для этого нужно щёлкнуть правой кнопкой на списке и выбрать соответствующий пункт во всплывающем меню. Для большей информации об отдельных столбцах отображённых в списке, смотрите справочник “*Oracle Server Reference*”.

Вы можете определить ваши собственные запросы для страниц сессионной информации или изменить стандартные запросы. Вы можете создать, изменить или удалить Запросы сессии для списка сессий и страниц детализации сессии. При этом появится следующий диалог:



В верхней части вы можете переключаться между запросами *Фильтров* и *Детализаций*. Фильтры контролируют, что будет отображаться в списке сессий, и детализации контролируют, что будет отображаться на страницах детализации.

В правой части вы увидите четыре кнопки для создания нового запроса, удаления запроса или перемещения запроса вверх или вниз по списку. Когда вы создаёте или изменяете запрос, вы должны указать следующую информацию:

Фильтры

Название отображается в списке выбора фильтра на панели инструментов.

Запрос является SELECT командой, которая выбирает информацию о сессиях из главного списка сессий. Вы можете использовать этот запрос что бы:

1. Ограничить набор отображаемых сессий (например, только активные сессии).
2. Задать порядок отображения сессий.
3. Задать какие колонки отображать, и в каком порядке.

Детализации

Название выводится в заголовках страниц детализации сессии. Позиция в списке определяет позицию страницы в окне сессии.

Запрос является SQL оператором SELECT, который выбирает сессионную информацию. Этот оператор SELECT обычно должен содержать связанную переменную *:sid*. И перед исполнением запроса ее значение будет установлено в SID выбранной сессии. Более того, вы можете использовать любые из столбцов представления *v\$session*, такие как *:username*, *:sql_address* или *:sql_hash_value*. Если во время запуска ваш запрос приводит к ошибке, то после выполнения она будет отображена на соответствующей странице.

Что бы соединить значение(я) нескольких записей детального запроса сессии, вы можете использовать подсказку */* concatenate */* в тексте SQL. Например:

```
select sql_text from v$sqltext_with_newlines
where address = hexoraw(:sql_address)
and hash_value = :sql_hash_value
order by piece
/* concatenate */
```

Значения *sql_text* колонки всех записей возвращенных этим запросом будут соединены, и отображены как одно значение в детальной таблице сессии.

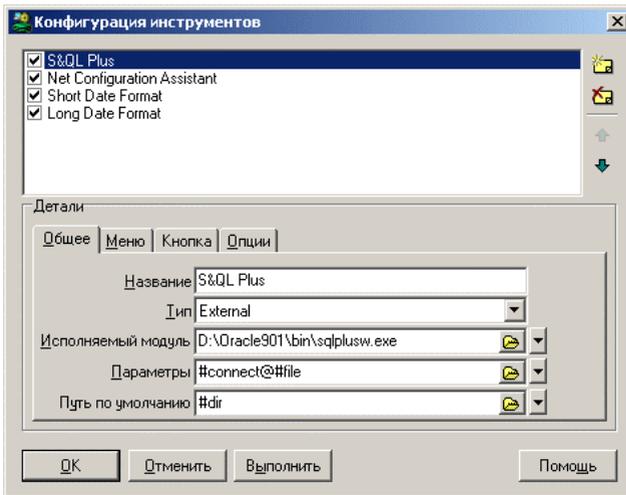
17.14 Инструменты, определяемые пользователем

Вы можете интегрировать с PL/SQL Developer IDE ваши собственные инструменты. Эти инструменты могут быть добавлены как пункты меню для запуска внешних приложений или для выполнения скриптов SQL или PL/SQL в текущей сессии. Вы можете передать приложению в качестве параметров имя файла и строку соединения.

Хорошим примером этой возможности может быть SQL*Plus. Вы можете добавить пункт меню, запускающий SQL*Plus, и, используя соответствующие настройки, обеспечить автоматическое подключение SQL*Plus и даже исполнение текущего открытого файла.

Другим примером является установка NLS параметров в специфический набор значений. Вы можете создать скрипт, исполняющий операторы «alter session», и добавить его в меню, откуда он может быть вызван для сессии Окна SQL, Тестового окна и т. д.

Чтобы настроить внешние инструменты, вам необходимо выбрать пункт *Конфигурировать инструменты* в меню *Инструменты*. Появится следующий диалог:



Четыре кнопки в правом верхнем углу позволяют вам вставлять и удалять элементы, а также перемещать элементы вверх или вниз по списку. Если во время нажатия кнопки *Создать*, вы будете удерживать клавишу Ctrl, то вновь созданный элемент будет скопирован из текущего выбранного элемента.

Для выполнения выбранного инструмента может быть использована кнопка *Выполнить*. Если вы нажмете эту кнопку вместе с клавишей ctrl, то будет выведено сообщение с информацией (с подставленными параметрами) о том, что должно быть выполнено.

Список показывает все сконфигурированные инструменты, а нижняя половина показывает конфигурацию выбранного инструмента. Конфигурация разбита на три секции:

- **Общее** – для определения исполняемого модуля / скрипта, который должен быть запущен, и его параметров.
- **Меню** – для определения места, где соответствующий пункт меню должен появиться
- **Кнопка** – для определения картинки и описания кнопки панели инструментов
- **Опции** – для некоторых дополнительных настроек инструмента

Далее мы рассмотрим конфигурирование внешних инструментов, на примере SQL*Plus.

Страница “Общее”

Первое, что нужно сделать, после того, как вы при помощи кнопки *Создать* создали новый элемент, это определить тип инструмента. Если вы выберете *External*, инструмент будет запущен как внешнее приложение. Это подходит для SQL*Plus. Если вы выберете *Session*, то инструмент запустит SQL скрипт для текущей сессии.

Далее на странице **Общее** вам нужно ввести описание. Описанием является имя, которое будет отображено в меню и в списке. Вы можете ввести & перед символом, который вы хотите сделать горячей клавишей (S&QL*Plus, чтобы получить SQL*Plus). Если в качестве описания вы вводите –, будет отображена разделительная линия.

Третьей важной вещью является имя программы или скрипта, которые необходимо выполнить. В качестве внешнего приложения, вы можете ввести любой исполняемый файл, или даже, если хотите, документный файл, в этом случае будет запущено связанное приложение. Кнопка *Выбор каталога* открывает файловый диалог, позволяющий вам выбрать файл. Большинство инструментов Oracle хранится в каталоге Oracle bin, и SQL*Plus может быть найден по адресу <oracle_home>\bin\sqlplusw.exe.

Этих трёх шагов достаточно, чтобы SQL*Plus запускался из меню. Вы можете дополнительно определить параметры, которые должны быть переданы приложению, а также вы можете определить путь по умолчанию. Вы можете использовать путь, указанный по умолчанию, как альтернативу полному пути выполняемого или файлового параметра. Для SQL*Plus вы могли бы добавить строку соединения (#connect) и ссылку на файл, который необходимо выполнить (#file).

Маленькая кнопка со стрелкой вниз справа позволит вам выбирать переменную, которую вы можете вставить в какое-либо из полей. Все эти переменные относятся к текущему соединению и открытому файлу. Перед исполнением инструмента, они будут заменены соответствующей информацией. Поддерживаются следующие переменные:

Переменная	Значение
#file	Представляет в текущем окне имя файла (без пути)
#path	Имя файла с указанием пути
#dir	Каталог файла в текущем окне
#object	Выбранный объект навигатора (например, SCOTT.EMP)
#otype	Тип выбранного объекта навигатора (например, TABLE)
#owner	Владелец объекта (SCOTT)
#oname	Имя объекта (EMP)
#connect	Полная строка соединения текущего соединения PL/SQL Developer (например, scott/tiger@demo)
#username	Имя пользователя
#password	Пароль
#database	База данных

При этом SQL*Plus не любит полный путь с возможными пробелами как файловый параметр, вот почему #dir указывается как каталог по умолчанию, чтобы SQL*Plus передавалось только имя файла.

Сессионные инструменты

Для сессионных инструментов вам нужно указать SQL скрипт, который может содержать несколько операторов SQL или PL/SQL блоков. Операторы SQL могут быть разделены точками с запятой или слешами, PL/SQL блоки должны оканчиваться слешем. Это тот же самый синтаксис, как и в SQL*Plus.

Example 1 – Germany.sql:

```
alter session set nls_language = 'german';
alter session set nls_territory = 'germany';
```

Example 2 – SpecialRole.sql:

```
begin
  dbms_session.set_role(role_cmd => 'special_role identified by &Password');
end;
/
```

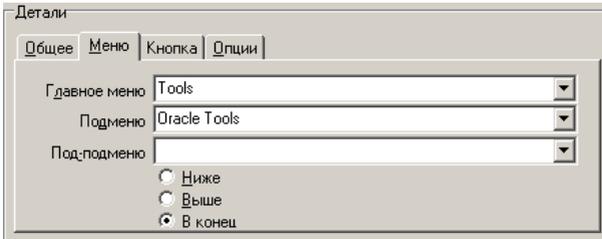
Заметьте, что для того, чтобы сделать скрипт более гибким, вы можете добавить в него подстановочные переменные. Перед запуском инструмента, у вас будут запрошены значения

этих переменных. Например, у вас будет запрошен пароль перед запуском приведённого выше скрипта SpecialRole.

Если вы хотите использовать в скрипте какие-либо из указанных параметров, вы должны использовать &1, &2 и т. д. Эти подстановочные переменные командной строки у вас не будут запрошены.

Страница “Меню”

По умолчанию все сконфигурированные инструменты добавляются в нижнюю часть меню Инструменты. Если вы хотите создать их в каком-либо другом месте или даже в их собственном основном меню, вы можете использовать страницу Меню:



Если вы хотите, чтобы ваш пункт меню создавался в другом месте главного меню, вы можете выбрать пункт меню из списка Главное меню, или же, если вы хотите создать новый пункт в главном меню, вы можете ввести новое имя пункта главного меню.

Если вы указали пункт главного меню, вы можете использовать Подменю и Под-подменю, чтобы указать точное место, где вы хотите создать новый пункт меню.

Чтобы указать, где должен помещаться новый пункт меню, относительно указанного пункта меню, вы можете использовать три кнопки (Ниже, Выше & В конец). Если вы указываете новое (под)меню, вам всегда следует пользоваться кнопкой ‘В конец’, потому что кнопки ‘Ниже’ и ‘Выше’ имеют значение, если вы обращаетесь к существующему (под)меню.

Пример 1:

Main menu	Tools
Sub menu	Oracle tools
Sub-sub menu	
Location	At End

При этом в основном меню Инструменты будет создано подменю Oracle tools. Подобное подменю может быть неплохой идеей, если вы собираетесь добавить несколько инструментов, относящихся к Oracle.

Пример 2:

Main menu	Oracle
Sub menu	
Sub-sub menu	
Location	At End

При этом будет создан пункт Oracle в главном меню. Если вы используете инструменты регулярно и хотите, чтобы они всегда были «под рукой», вам, скорее всего, будет необходимо создать новый пункт в главном меню.

Если вы хотите добавить ваш инструмент первым пунктом в меню Инструменты, вам нужно ввести:

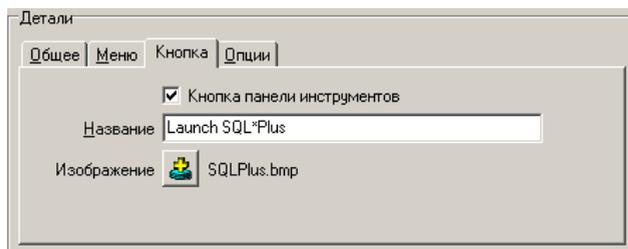
Main menu	Tools
Sub menu	Preferences...

Sub-sub menu	
Location	Above

При этом над существующим пунктом меню Предпочтения будет создан новый пункт.

Страница “Кнопка”

Вы можете определить, следует ли и каким образом, включать внешний инструмент в панель инструментов:

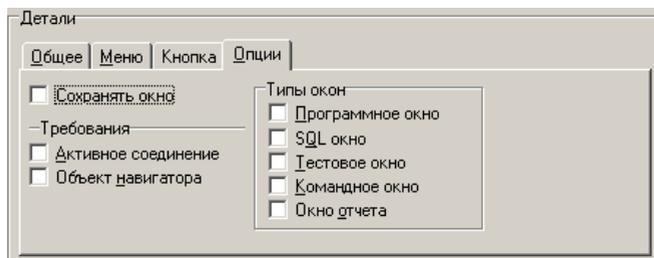


- Кнопка панели инструментов
 Когда эта опция включена, внешний инструмент может быть включён в панель инструментов.
- Название
 Название будет отображено в качестве подсказки, когда вы удерживаете курсор мыши над кнопкой панели инструментов. Если вы оставите это описание пустым, будет использовано описание внешнего инструмента (как показано в меню).
- Изображение
 Чтобы выбрать файл рисунка Windows (*.bmp) для кнопки панели инструментов, нажмите кнопку Изображение. Желательно, чтобы размер рисунка был 20 x 20 пикселей. Заметьте, что PL/SQL Developer всегда будет загружать файл рисунка из первоначального местонахождения, поэтому вам не следует перемещать или переименовывать этот файл без изменения соответствующего внешнего инструмента.

 PL/SQL Developer содержит набор стандартных файлов рисунков, из которого вы можете выбрать подходящий. Эти файлы хранятся в установочном каталоге PL/SQL Developer в подкаталоге Icons. Это каталог по умолчанию для диалога выбора изображения.

Страница “Опции”

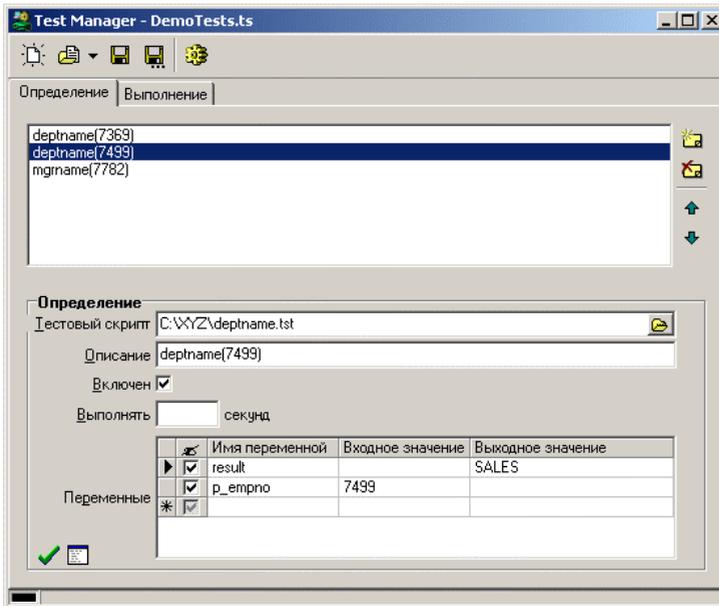
Страница Опции позволит вам делать следующие настройки:



- **Сохранить окно**
Когда установлена эта опция, перед выполнением инструмента активное окно будет сохранено. Вам следует устанавливать эту опцию, когда инструмент получает файл в качестве параметра, и вы хотите быть уверенным, что инструмент использует текущие данные.
- **Активное соединение**
Если инструмент применим только если PL/SQL Developer подключен к БД (если вы укажете строку соединения как параметр), вам следует включить опцию Активное соединение. При этом пункт меню будет выключен, если PL/SQL Developer не подключен к БД.
- **Объект навигатора**
Установите эту опцию, если инструмент применяется к объектам, выбранным в навигаторе.
- **Типы окон**
Если инструмент применим к специфическому окну, в этой секции вы можете это определить. Если выбран один или несколько типов, пункт меню будет включен, только если активное окно одного из указанных типов.

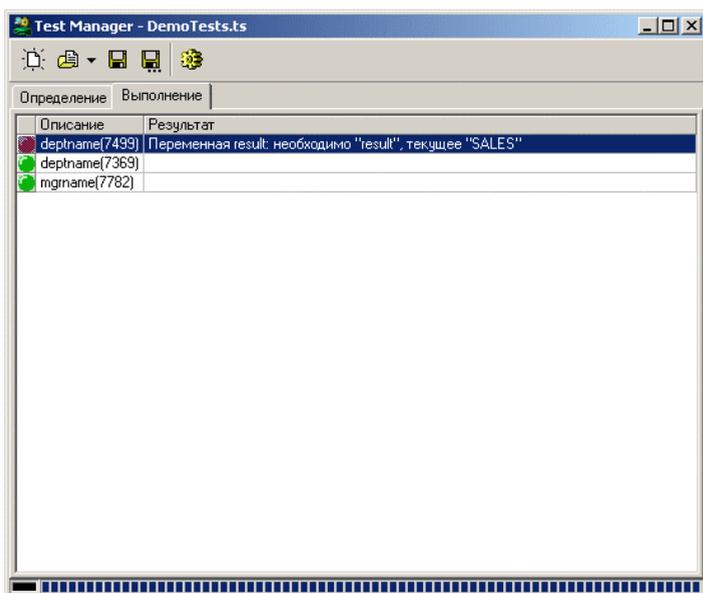
17.15 Менеджер тестов

Что бы выполнить регрессионное тестирование ваших хранимых программных модулей Oracle, вы можете использовать Менеджер тестов. Он позволяет вам определить Тестовый набор, который является коллекцией Тестовых скриптов с входными переменными и необходимым выводом. Менеджер дополнительно позволяет вам указать необходимую производительность:



В этом примере Тестовый набор состоит из 3 Тестовых скриптов. Для выбранного скрипта *DeptName.ts*, переменной *p_empno* будет присвоено значение 7499 на входе, и после исполнения переменная *result* должна иметь значение *SALES*.

Запустив Тестовый набор, вы сможете быстро обнаружить, работают ли модули тестируемой программы корректно и/или с необходимой производительностью:



Все Тестовые скрипты завершившиеся некорректно будут иметь красный индикатор и будут помещены в начале. В этом случае скрипт *DeptName(7499)* завершился некорректно, так как спецификация требует, что бы результатом было значение “SALES” а не “Sales”.

Создание Тестового набора

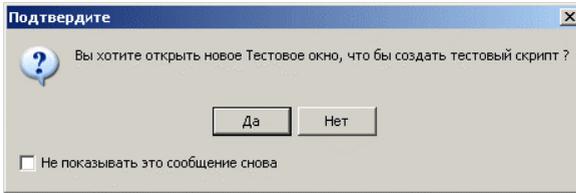
Что бы создать Тестовый набор, выберите пункт *Менеджер тестов* из меню *Инструменты*. Появится пустой экран Менеджера тестов. Вверху Менеджера тестов вы видите список Тестовых скриптов, где вы можете включить скрипты в набор, удалить скрипты из набора или изменить порядок скриптов. Внизу вы видите определение выбранного скрипта.

Перед созданием нового Тестового скрипта, будет хорошей идеей сохранить Тестовый набор, нажав на панели инструментов кнопку *Сохранить Тестовый набор*. Все новые Тестовые скрипты будут созданы по умолчанию в том же самом каталоге, где находится Тестовый набор, и будет хорошей идеей - использовать отдельный каталог для каждого Тестового набора, так что всеми связанными тестовыми элементами можно будет управлять из этого отдельного каталога.

Добавление Тестового скрипта к Тестовому набору

Есть 2 метода добавить Тестовый скрипт в Тестовому набору:

1. Перетащить процедуру или функцию из Объектного навигатора в Менеджер тестов. Будет открыто новое Тестовое окно с вызовом этого программного модуля, и переменными для всех параметров и возвращаемого значения. Сохраните Тестовый скрипт, задав некоторое описательное имя, предпочтительно в том же самом каталоге, как и Тестовый набор.
2. Нажмите кнопку *Создать* справа от списка Тестовых скриптов. Появится следующий диалог:



Нажмите *Да*, что бы создать новый, пустой Тестовый скрипт. Вам будет необходимо вручную запрограммировать Тестовый скрипт и сохранить его, перед тем как он может быть использован в Тестовом наборе.

Нажмите *Нет*, если вы хотите добавить существующий Тестовый скрипт в Тестовому набору.

После того, как Тестовый скрипт сохранен, вы можете задать его определение в Тестовом наборе:

- **Тестовый скрипт** – Имя файла Тестового скрипта. Оно будет уже заполнено, если вы создали и сохранили новый Тестовый скрипт. Если вы хотите добавить существующий Тестовый скрипт, нажмите кнопку *Выбрать* справа от этого поля.
- **Описание** – Функциональное описание скрипта. Если вы запускаете один скрипт несколько раз с различными значениями входных переменных, вам следует включить это в описание, что бы можно было быстро распознать это в списке.
- **Включен** – Используйте эту опцию, что бы включить или выключить скрипт в Тестовом наборе. Только включенные скрипты будут исполнены во время тестового прогона, а выключенные - пропущены.
- **Выполнять N секунд** – Необходимая производительность может быть указана как максимальное количество секунд (допускаются десятичные доли). Если вы оставите это поле пустым, производительность не будет тестироваться.
- **Переменные** – Для каждой переменной в Тестовом скрипте, введите имя, входное значение и требуемое выходное значение. Входное значение будет передано Тестовому скрипту перед исполнением. Требуемое выходное значение будет проверено после исполнения. Если вы оставите выходное значение пустым, оно не будет тестироваться. Что бы проверить на пустое значение, введите слово “null”. Что бы проверить вывод dbms_output, введите “dbms_output” в качестве имени переменной и введите требуемый текст в выходном значении.

Внизу слева вы видите 2 кнопки, которые позволяют вам быстро запустить и проверить выбранный Тестовый скрипт, или открыть его в Тестовом окне для редактирования.

Дублирование Тестового скрипта

Очень часто вам необходимо запустить один и тот же Тестовый скрипт, но с различными значениями входных и выходных переменных. Что бы упростить это, вы можете выбрать существующий Тестовый скрипт в Тестовом наборе и, удерживая клавишу *Ctrl* нажатой во время нажатия кнопки *Создать*. Новый созданный элемент Тестового набора унаследует все свойства от выбранного элемента, таким образом, вам будет только необходимо изменить описание и значения переменных.

Исполнение Тестового набора

Что бы выполнить Тестовый набор, нажмите кнопку *Выполнить* на панели инструментов. Менеджер тестов переключится на страницу *Исполнение*, и каждый включенный Тестовый скрипт будет выполнен в отдельном Тестовом окне. Неудачные завершения Тестовых скриптов

будут отмечены красным индикатором, и будут помещены в начало списка. Удачные скрипты будут отмечены зеленым индикатором, и будут помещены в конец списка.

Что бы определить причины неудачи, щелкните правой кнопкой мыши на скрипте и выберите *Отладить* из всплывающего меню. Будет создано Тестовое окно с Тестовым скриптом, а значения входных переменных будут установлены соответственно. Теперь вы можете отладить Тестовый скрипт, что бы определить причину ошибки.

Щелкните правой кнопкой мыши на результатах тестового прогона, что бы скопировать их в буфер обмена, напечатать или экспортировать в различные форматы.

18. Редактор

Каждое окно, которое позволяет вам редактировать какой-либо исходный текст SQL или PL/SQL, использует один и тот же редактор. Этот редактор имеет много специальных функций, которые помогут немного облегчить работу программиста.

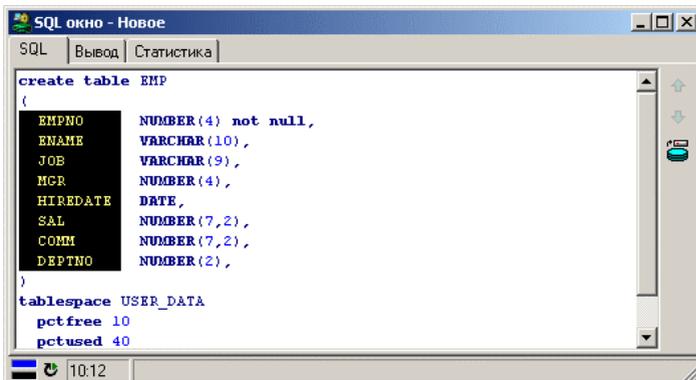
18.1 Функции выбора

Редактор позволит вам выполнять различные функции над выделенным текстом. Эти функции доступны при помощи пункта *Выделение* в меню *Правка* и применимы, только в том случае, если вы выделили текст. При помощи предпочтения *Настройка клавиатуры* вы можете задать функциональные клавиши для этих функций.

- Увеличить отступ / Уменьшить отступ
Количество символов зависит от предпочтений редактора. Если выделение существует, то вы можете использовать клавиши *Tab* и *Shift-Tab* для этих двух функций.
- Верхний регистр / Нижний регистр
- Закомментировать / Разкомментировать
Использует */** и **/*, чтобы быстро закомментировать выделение.
- Применить регистр
Если у вас установлено значение предпочтения Регистр ключевого слова – Uppercase, Lowercase или Init Caps, то эта функция применит этот стиль ко всем ключевым словам в выделенном тексте.
- Сортировать
Строки выделения будут отсортированы в восходящем порядке. Если вы вызовете функцию сортировки снова, выделение будет отсортировано в нисходящем порядке.
- Пометить цветом
Применяет текущий маркерный цвет в выделенному тексту (см. так же главу **Ошибка!** [Источник ссылки не найден.](#)).

18.2 Выделение столбца

Вы можете выделить текстовый столбец, удерживая во время выделения клавишу *Alt*:



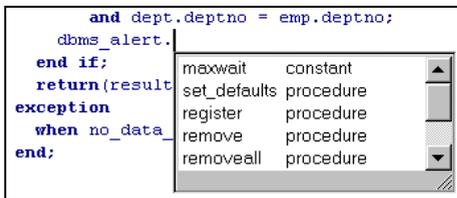
Как обычно, вы можете применить к выделению следующие функции - копировать, вставить, удалить, перевести в верхний регистр, перевести в нижний регистр, увеличить (уменьшить) отступ и закомментировать (Разкомментировать). Если вы используете функцию Применить регистр, и ключевое слово заходит за границу столбца, то оно не будет обработано.

18.3 Направляющая

Когда вы редактируете сложный код с несколькими операторами if/then/else и с циклами, которые занимают несколько страниц кода PL/SQL, то может быть полезным установить в редакторе вертикальную направляющую. Для этого просто поместите курсор в нужном горизонтальном местоположении и выберите в меню *Правка* пункт *Направляющая*. Этот пункт меню действует как переключатель, так что, чтобы убрать направляющую, вы можете использовать его ещё раз.

18.4 Ассистент кодирования

Ассистент кодирования – очень мощная функция, которая помогает при вводе вашего SQL или PL/SQL текста. Например, когда вы введёте *dbms_alert*. и немного подождёте, редактор отобразит список элементов пакета *dbms_alert*:



Когда вы продолжите вводить имя процедуры *dbms_alert*, список будет сокращаться. Например, если после *dbms_alert.* вы введёте *r*, в списке останутся только процедуры *register*, *remove* и *removeall*. Если, когда отображён Ассистент кодирования, вы нажмёте клавишу enter, то имя текущего выделенного пункта будет вставлено в исходный текст. Для навигации в Ассистенте кодирования и выбора нужного пункта вы также можете использовать клавиши со стрелками. При нажатии клавиши *Escape* Ассистент кодирования будет скрыт.

Ассистент кодирования будет помогать при работе со следующими типами объектов:

Object type	Elements
Package	Procedures, functions, types, variables, constants and exceptions
Function, Procedure	Parameters, inserted with named notation (param =>)
Table, View	Columns
Sequence	nextval and currval
User	Objects owned by the user
Cursor variable	Fields of the select statement of the cursor
Record type variable	Fields of the record type
Table%rowtype variable	Columns of the table
Object type	Attributes (for default constructor)
Object type variable	Attributes and methods
Object type column	Attributes and methods
Collection variable	Collection methods (first, next, limit, and so on)

Ассистент кодирования также может помочь вам ввести имена объектов (таблиц, пактов и т. д.), ключевые слова и идентификаторов PL/SQL, которые имеют значение в текущем контексте. Например, если вы введёте в редакторе SQL *pro* и подождёте, Ассистент кодирования отобразит ключевые слова *procedure* и *profile* вместе с объектами базы данных, начинающимися на *pro*. Контекст зависит от текущего подключённого пользователя (который определяет имена доступных объектов), типа редактора (который определяет доступные ключевые слова) и программного модуля, который вы редактируете (который определяет доступные идентификаторы PL/SQL).

Ассистент кодирования может быть вызван автоматически и/или вручную, и время, по прошествии которого Кодовый помощник автоматически активизируется после ввода имени объекта, может быть определено предпочтением. В дальнейшем, вы можете определить, хотите ли вы описывать пользователя, и, если да, то, какие объектные типы вы хотите включить. Можно указать описание контекста и минимальное число, которое нужно ввести, перед тем, как оно может быть активировано. Все эти предпочтения описаны в главе 16.14.

18.5 История команд

Когда вы успешно выполните оператор в SQL окне, Командном окне или Окне отчёта, этот оператор будет помещён в глобальный буфер повторного вызова команд. Вы можете повторно вызвать в редакторе ранее выполненные операторы, выбрав в меню *Правка* пункт *История команд* или нажав *Ctrl-E*. При этом появится список выбора, подобный данному:

Время	Пользователь	Команда
21.07.2005 23:00:37	scott@ORACLE81	select * from emp
21.07.2005 22:58:15	scott@ORACLE81	select * from dept
21.07.2005 22:57:49	demo@ORACLE81	select * from emp
14.07.2005 22:31:14	scott@ORACLE81	select t.*, t.rowid from emp t
14.07.2005 22:30:54	scott@ORACLE81	select t.rowid, t.* from emp t
19.06.2005 14:52:00	scott@ORACLE81	begin dbms_output.put_line(employee.deptname(7876)); en
19.06.2005 14:51:24	scott@ORACLE81	select * from emp where deptno = 20 order by ename
19.06.2005 14:46:56	scott@ORACLE81	select * from emp where job = 'MANAGER' order by deptno
19.06.2005 14:45:03	scott@ORACLE81	select t.*, t.rowid from emp t order by ename
19.06.2005 14:39:40	scott@ORACLE81	select * from emp order by ename
19.06.2005 14:38:35	scott@ORACLE81	select * from emp where ename like 'A%' and deptno = 20 o
19.06.2005 14:38:22	scott@ORACLE81	select * from emp where ename like 'A%' order by ename
19.06.2005 14:30:19	scott@ORACLE81	select t.*, t.rowid from test t
19.06.2005 14:30:13	scott@ORACLE81	select * from test t
19.06.2005 14:28:59	scott@ORACLE81	select * from dummy t

Команды отображены в порядке выполнения, начиная с самой последней выполненной. В списке вы увидите время, пользователя, текст команды (или его часть). Теперь вы можете выбрать команду и дважды щёлкнуть по ней (или нажать *Enter*), чтобы вставить ее в редактор на ту позицию, в которой был вызван список.

Заметьте, что команды, которые включают в себя пароли, не помещаются в глобальный список повторного вызова операторов из соображений безопасности.

Вы можете отсортировать список по времени, пользователю или тексту команды, нажимая соответствующие кнопки в заголовке списка. Чтобы отобразить только те операторы, которые были выполнены текущим подсоединённым пользователем, щёлкните правой кнопкой мыши на списке и выберите во всплывающем меню пункт *Текущий пользователь*. В этом же всплывающем меню вы можете переключиться обратно на *Все пользователи*, а также вы можете удалить команду или скопировать текст в буфер обмена.

Чтобы найти какую-либо команду, вы можете вызвать функцию поиска, используя соответствующую комбинацию клавиш (по умолчанию *Ctrl-F*) или щёлкнув правой кнопкой мыши на списке и выбрав из всплывающего меню пункт *Найти команду*. Функция поиска

начнёт поиск со следующей после текущей выбранной команды и продолжит его с первой команды, если она не нашла подходящий.

Кроме того, всплывающее меню содержит функции удаления выбранной команды из списка, копирования ее в буфер обмена или экспорта всего списка в файл CSV.

Буфер повторного вызова может содержать до 200 операторов, которые будут храниться в каталоге прикладных данных вашего профиля Windows. Вы можете изменить максимальное количество операторов при помощи предпочтений (смотрите главу 16.25).

18.6 Специальное копирование

Если вы используете для написания SQL и PL/SQL кода PL/SQL Developer, то вы можете впоследствии захотеть использовать другие инструменты, такие как язык программирования 3GL, вам может понадобиться код в немного другом формате. Давайте предположим, что вы написали и протестировали в PL/SQL Developer оператор SQL, такой как этот:

```
select deptno, sum(sal) mgr_sal from emp
where job = 'MANAGER'
group by deptno
order by mgr_sal desc
```

Если, например, вы захотите использовать этот оператор в Borland Delphi, вам может понадобиться такой формат:

```
SQL := 'select deptno, sum(sal) mgr_sal from emp' + #13#10 +
'where job = 'MANAGER'' + #13#10 +
'group by deptno' + #13#10 +
'order by mgr_sal desc';
```

Для этого вы можете использовать функцию *Специальное копирование* из меню *Правка* или из всплывающего меню после щелчка правой кнопкой мыши в редакторе. Эта функция имеет подменю, которое показывает все форматы специального копирования, которые определены. После выбора формата, преобразованный код помещается в буфер обмена, так что вы можете вставить его в редактор соответствующего инструмента.

Форматы специального копирования определены в каталоге SpecialCopy установочного каталога PL/SQL Developer. Вы можете изменить ранее определённые форматы копирования или добавить новые форматы копирования. Просто добавьте текстовый файл с расширением .copy, который содержит переменную для первой строки кода PL/SQL (<line_1>), последней строки кода PL/SQL (<line_N>) и переменную для всех остальных строк (<line_*>). Пример для Borland Delphi будет выглядеть следующим образом:

```
;PL/SQL Developer SpecialCopy definition for Borland Delphi
;<line_1> for first line
;<line_*> for all other lines
;<line_N> for last line
;
SQL := '<line_1>' + #13#10 +
'<line_*>' + #13#10 +
'<line_n>';
```

Первой строке должно предшествовать задание переменной SQL, а также её должна сопровождать пара CR/LF. Последней строке не требуется пары CR/LF, но необходимо завершить её точкой с запятой. Все остальные строки необходимо сопровождать CR/LF.

Вы можете опустить <line_1> и <line_n>, если они такие же, как и <line_*>.

В некоторых языках для специальных символов вам необходимо использовать управляющие последовательности. Например, в C++ вы будете использовать в качестве символа табуляции \t (ASCII code 9). Чтобы определить эту управляющую последовательность, используйте ключевое слово #define:

```
#define char(9) = \t
#define \ = \\
String("<line_1>\n") +
String("<line_*>\n") +
String("<line_n>");
```

Чтобы указать, что вы хотите удалить из результата все избыточные пустые символы (пробелы, символы табуляции и переносы), вы можете дополнительно использовать "#define compress".

Заметьте, что имя .сору файла будет включено в меню, так что вам следует использовать понятные имена файлов.

18.7 Контекстно-зависимая помощь

Большинство программистов часто обращаются к информации, относящейся к функциям, которые они используют в своих программах. Чтобы найти эту информацию максимально быстро, надо поместить курсор на ключевое слово в исходном тексте и нажать *F1*, и будет начат поиск этого ключевого слова в оперативных руководствах Oracle.

Более подробно о Справочной системе PL/SQL Developer вы можете прочитать в главе 26.

18.8 Всплывающее меню объекта базы данных

Когда ваша программа обращается к объекту базы данных, очень часто вам будет необходимо получить некоторую информацию от этого объекта или вы захотите проделать какие-либо действия над ним. Возможно, вы захотите узнать его описание или свойства. Для таблицы или представления вы, может быть, захотите запросить или отредактировать данные. Для функции или процедуры, возможно, вы захотите протестировать некоторый сценарий.

Если вы щёлкните в редакторе правой кнопкой мыши на имени объекта, вы увидите то же самое всплывающее меню, которое вы видели, когда щёлкали правой кнопкой мыши на имени объекта в Объектном Навигаторе. Поэтому, все функции, применимые к этому объекту, непосредственно доступны из редактора.

18.9 Построение плана выполнения

Чтобы вызвать Окно плана выполнения для SQL оператора в редакторе, выделите текст оператора, щёлкните на нём правой кнопкой мыши и выберите во всплывающем меню пункт *Построить план выполнения*. Если текст не выбран, в Окно плана выполнения будет помещён весь исходный текст. Если оператор SQL содержит переменные PL/SQL, вы должны изменить оператор в Окне плана выполнения, как это описано в главе 5.1.

18.10 Макросы

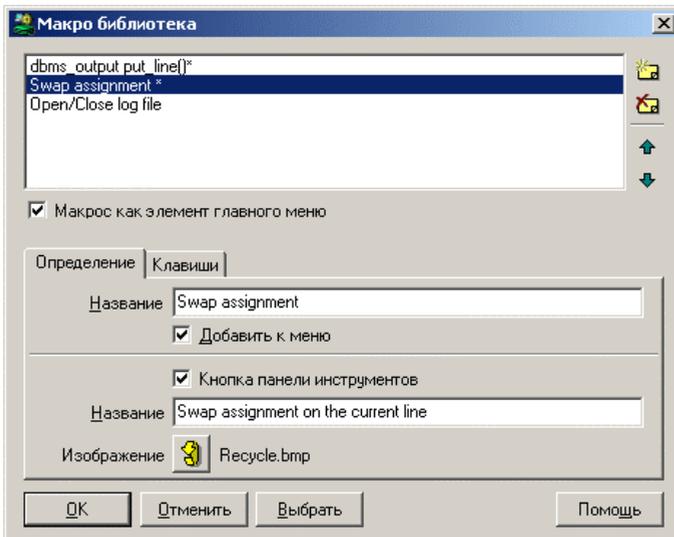
Редактор обладает мощной функцией *масго*, которая позволяет быстро и легко записывать и воспроизводить макрос, которая может быть полезна для автоматизации некоторых повторяющихся задач.

Записанный макрос может быть сохранен в Библиотеке макросов, чтобы позже быть выполненным. Макросы в библиотеке обычно будут некоторыми периодически повторяющимися задачами (например, поменять местами части оператора присвоения: a := b; ->

b := a;) или могут содержать отрывки кода. Например: почему постоянно вспоминать, как должен быть объявлен курсорный тип, когда вы можете набрать один раз и записать для дальнейшего использования?

Чтобы записать макрос, нажмите *F11* или выберите в меню *Макро* из меню *Инструменты* пункт *Запись*. Теперь панель инструментов отображает мигающий значок рекордера, показывающий, что в данный момент вы записываете макрос. Все нажатия клавиш будут записаны для дальнейшего воспроизведения, что подразумевает, что в течение записи макроса вам не следует пользоваться мышкой. При повторном нажатии *F11* запись макроса будет закончена. Для воспроизведения макроса нажмите *F12*.

Чтобы поместить текущий записанный макрос в Библиотеку макросов, выберите в меню *Макро* пункт *Библиотека*. При этом будет открыт диалог *Макро библиотека*:



Чтобы добавить новый макрос с только что записанным определением, нажмите кнопку *Создать*. Вы можете ввести описание для макроса и указать, следует ли включать этот макрос в меню *Макро*. Таким образом, вы можете сделать некоторые макросы более доступными, а также при помощи предпочтения *Настройка клавиатуры* вы можете дополнительно присвоить комбинацию клавиш любому макросу. Если включена опция *Макрос как элемент главного меню*, то меню *Макро* будет находиться в главном меню. Если опция выключена, меню *Макро* будет находиться в меню *Инструменты*.

Когда включена опция *Кнопка панели инструментов*, макрос может быть дополнительно помещен на панель инструментов. Когда вы будете удерживать курсор над кнопкой панели инструментов, соответствующее *Название* будет отображаться в качестве подсказки. Если вы оставите название пустым, то будет использовано основное название макроса (как отображается в библиотеке и в меню). Чтобы выбрать в качестве кнопки на панели инструментов файл точечного рисунка Windows (*.bmp), нажмите кнопку *Изображение*. Желательно, чтобы размер рисунка был 20 x 20 пикселей. Заметьте, что PL/SQL Developer всегда будет загружать файл точечного рисунка из первоначального местонахождения, поэтому вам не следует перемещать или переименовывать этот файл без изменения соответствующего макроса. PL/SQL Developer поставляется с набором стандартных файлов точечных рисунков, из которого вы можете

выбирать. Эти файлы хранятся в установочном каталоге PL/SQL Developer в подкаталоге Icons. Для диалога выбора рисунка этот каталог указан по умолчанию.

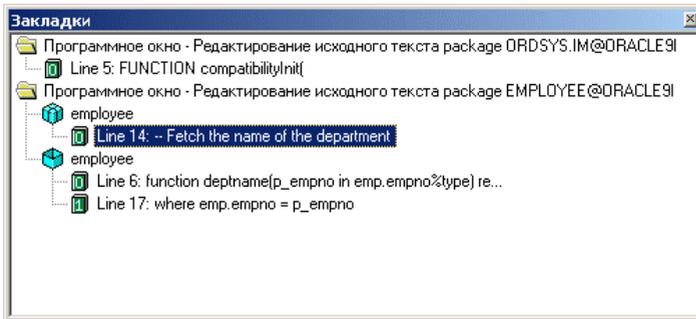
На странице *Клавиши* вы можете просмотреть и изменить действия с клавиатурой для макроса.

Чтобы выполнить хранимый макрос, откройте Макро библиотеку и дважды щёлкните на имени нужного вам макроса. Выбранный макрос может быть выполнен нажатием клавиши *F12*. Когда вы выберете в меню макрос, он будет немедленно выполнен и, нажатием *F12*, он может быть выполнен снова.

18.11 Закладки

Что бы пометить положение курсора в редакторе, вы можете добавить закладку. Просто нажмите Ctrl-Kn (где n – цифра от 0 до 9), что бы добавить закладку в текущем положении. Зеленая метка появится слева от редактора с соответствующим номером. Что бы перейти к этой закладке позже, нажмите Ctrl-Qn. Вы так же можете использовать подменю *Установить закладку* и *Перейти к закладке* из меню *Правка*.

Пункт *Список закладок* меню *Правка* отобразит окно с глобальным иерархическим списком закладок во всех редакторах:



В этом дереве вы видите окна, редакторы и строки, где закладки установлены. Щелкнув мышью на закладке, окно и редактор перейдут на передний план, а редактор перейдет к этой закладке. Вы можете закрепить список закладок, что бы сделать его доступным все время, и что бы он не закрывал другие окна.

18.12 Выделение цветом

В редакторе вы можете выделить цветом необходимый фрагмент. Что бы сделать это, выберите фрагмент, который вы ходите пометить, и нажмите кнопку *Выделение цветом* на панели инструментов:



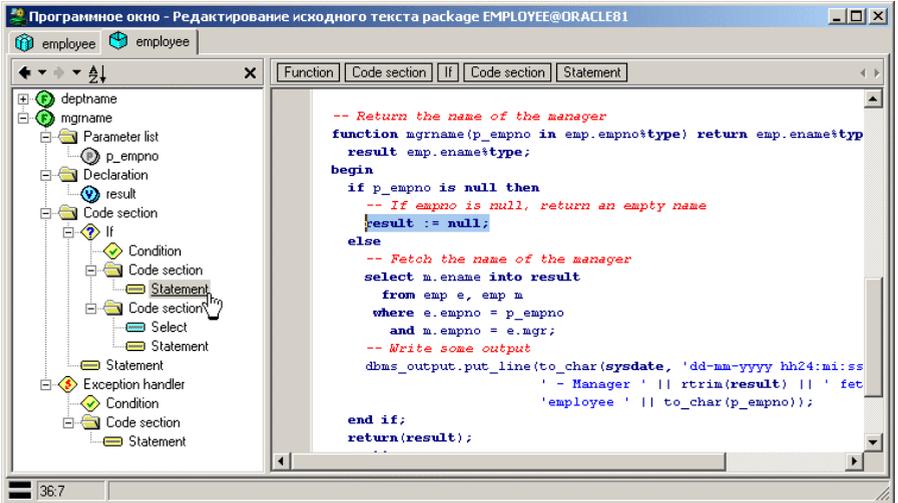
Цвет фона выбранного текста сменился на текущий цвет выделения:

```
-- Fetch the name of the manager
select m.ename into result
from emp e, emp m
where e.empno = xa
and m.empno = e.mgr;
```

Что бы сменить цвет выделения, нажмите на стрелку справа от кнопки и выберите новый цвет. Что бы отменить выделение цветом, поместите курсор внутри выделенного текста и нажмите кнопку *Выделение цветом*.

18.13 Разделы кода

Функция Разделы кода относится только к программному редактору. Она отображает всю структуру программного модуля. Эта функция наиболее полезна при работе с массивными телами пакетов и типов, которые могут занимать тысячи строк PL/SQL кода. Разделы кода отображаются в виде дерева с левой стороны от Программного окна:



Разделы кода отслеживают ваше текущее местоположение в исходном коде. Во время вашего перемещения по исходному тексту, выбранный элемент будет показывать, где вы находитесь в данный момент.

Если вы поместите мышь над элементом в Разделах кода, соответствующий исходный текст будет выделен в редакторе, как это показано выше.

Щелкнув элемент Разделов кода, вы немедленно перейдете к его объявлению. Двойной щелчек элемента развернет или свернет его. Если вы щёлкните на нём правой кнопкой мыши, появится всплывающее меню. Оно позволит вам выбрать в редакторе соответствующий исходный текст, вырезать его или копировать в буфер обмена, вставить его из буфера обмена, преобразовать его комментарий, развернуть все дочерние элементы или создать для него Тестовый скрипт (если это публичный программный модуль).

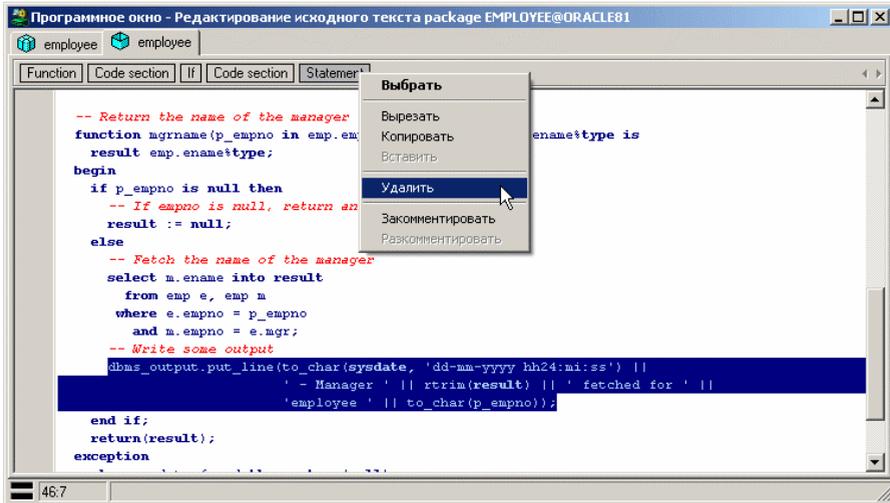
Разделы могут быть отсортированы в алфавитном порядке нажатием кнопки *Сортировать разделы* в верхней части панели разделов. Отсортировано будут только основные разделы: параметры, локальные переменные и тому подобное отсортированы не будут. Кнопка будет оставаться нажатой, и повторное нажатие вернёт содержимое к первоначальному порядку.

Чтобы скрыть Разделы кода, вы можете нажать в верхнем правом углу кнопку *Закрыть разделы*. Вы также можете щёлкнуть правой кнопкой мыши в редакторе и выбрать пункт *Показать разделы*, чтобы скрыть или отобразить их или выбрать в меню *Инструменты* пункт *Разделы кода*.

По умолчанию Разделы кода являются видимыми. Чтобы изменить это свойство, вы можете закрыть Разделы кода в текущем Программном окне и выбрать в меню *Окно* пункт *Сохранить макет*.

18.14 Иерархия кода

Иерархия кода относится только к Программному редактору и показывает иерархическую структуру PL/SQL кода в местонахождении курсора. Она находится над редактором:



В приведённом выше примере курсор находится рядом с Оператором `dbms_output` внутри Кодовой секции внутри Функции внутри Тела пакета. Если вы будете перемещать мышью по элементам Иерархии кода, будет выделен соответствующий исходный текст. При щелчке на элементе будет выбран соответствующий исходный текст.

Если вы щёлкните на элементе правой кнопкой мыши, появится всплывающее меню, которое позволит вам выбрать в редакторе соответствующий исходный код, вырезать его или скопировать в буфер обмена, вставить его из буфера обмена или преобразовать его в комментарий.

18.15 Навигация по гиперссылкам

В вашем PL/SQL коде часто будут встречаться ссылки на элементы, которые определены в другом месте. Например, вы можете сослаться на PL/SQL тип внутри того же тела пакета или внутри спецификации того же пакета или внутри другого пакета. То же самое возможно для вызова функции или процедуры, переменных, видов и так далее.

Чтобы легко найти объявление в Программном окне, вы можете поместить указатель мыши над такой ссылкой, нажать клавишу `Ctrl` и нажать левую кнопку мыши. После удерживания клавиши `Ctrl`, идентификатор под указателем мыши станет светло-синего цвета и подчеркнутым, повторяя вид гиперссылок HTML документа:

```

-- Created : 17-12-1999 17:19:26
-- Purpose : Lock an employee record
procedure lockrecord(p_empno in emp.empno%type) is
dummy varchar2(1);
begin
if p_empno != g_lastrecord.empno then
select null into dummy from emp
where empno = p_empno
for update;
end if;
end lockrecord;

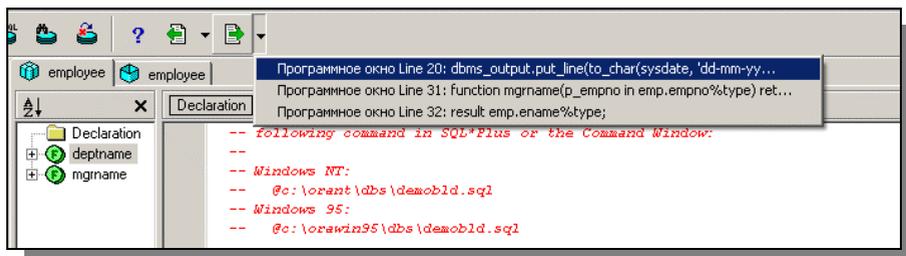
```

Если объявление находится в том же самом исходном тексте, PL/SQL Developer будет перемещаться внутри него. После этого поиск будет идти в оставшихся программных модулях внутри того же Программного окна. Далее поиск будет продолжен во всех других открытых Программных окнах. И, наконец, для объекта, который содержит объявление, PL/SQL Developer попытается открыть новое Программное окно или SQL окно и перейдет туда.

По умолчанию навигацию по гиперссылкам будет искать объявление в спецификации пакета или типа. Чтобы открыть вместо спецификации тело пакета или типа, вы можете щёлкнуть правой кнопкой на гиперссылке.

18.16 Кнопки навигации

Когда вы переходите к абсолютному местоположению внутри редактора какого-либо окна, PL/SQL Developer создаст закладку для предыдущего местоположения. К этим закладкам можно будет обращаться при помощи кнопок Назад и Вперед на главной панели инструментов:



Закладки будут добавлены, когда вы перемещаетесь по Разделам кода, используете функцию Найти, щёлкаете на ошибке компиляции, перемещаетесь в начало или конец редактора, используете навигацию по гиперссылкам, открываете новое окно и так далее.

18.17 Рефакторинг

Функция рефакторинга позволяет вам быстро реорганизовать ваш PL/SQL код. Она работает для выделенного кода, или, если выделение отсутствует, для текущей команды. Щелкнув правой кнопкой мыши на команде или выделении, вам будут доступны следующие функции в соответствующем подменю:

- Переименовать элемент
Переименовывает текущую переменную, параметр, константу или программный модуль. Как объявление, так и все ссылки будут переименованы.
- Выделить процедуру
Если программный модуль стал слишком большим, вы можете выделить фрагмент и преобразовать его в отдельную процедуру. Все переменные, используемые как внутри и так и вне выделения, будут преобразованы в параметры. Все переменные, используемые только

внутри выделения, будут перенесены из текущего программного модуля в новый программный модуль. И выделенный фрагмент будет замещен вызовом нового программного модуля.

- **Выделить локальную константу**
Если некоторое выражение необходимо преобразовать в локальную константу, то вы можете выделить его и указать имя константы. Локальная константа будет создана внутри текущей подпрограммы. Ее тип будет определен по выражению. Все вхождения выражения в текущей подпрограмме будут заменены на имя константы.
- **Выделить глобальную константу.**
Если некоторое выражение необходимо преобразовать в глобальную константу, то вы можете выделить его и указать имя константы. Глобальная константа будет создана внутри текущего пакета. Ее тип будет определен по выражению. Все вхождения выражения в текущем пакете будут заменены именем константы.
- **Заменить присваивание на инициализацию**
Если присваивание локальной переменной является ее инициализацией, вы можете перенести его в объявление переменной. Вы можете щелкнуть правой кнопкой мыши на команде или выделить несколько команд.

Если результат рефакторинга вас не устраивает, вы можете просто его отменить.

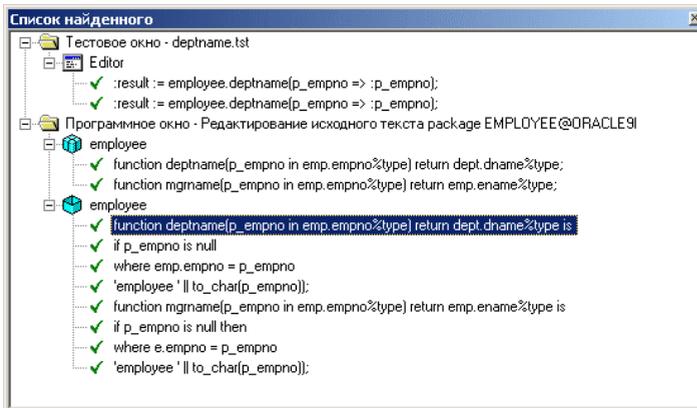
18.18 Панель поиска

Панель поиска доступна из меню Правка и может быть отображена как плавающий инструмент, или как закрепленный вверху или внизу рабочей области. Будучи закрепленной, панель будет постоянно доступной для выполнения поисков по нескольким редакторам и окнам:



После ввода искомого слова, вы можете нажать кнопку *Найти* или нажать *Enter*. Результат поиска будет подсвечен в редакторе(ах), и вы можете нажать кнопки *Перейти к следующему / предыдущему вхождению*. Вы можете нажать *Esc*, что бы перевести фокус ввода из панели в редактор. Нажмите кнопку *Скрыть вхождения*, что бы скрыть подсвеченные результаты поиска в редакторах.

Кнопка *Список поиска* отобразит окно с иерархическим представлением результатов поиска:



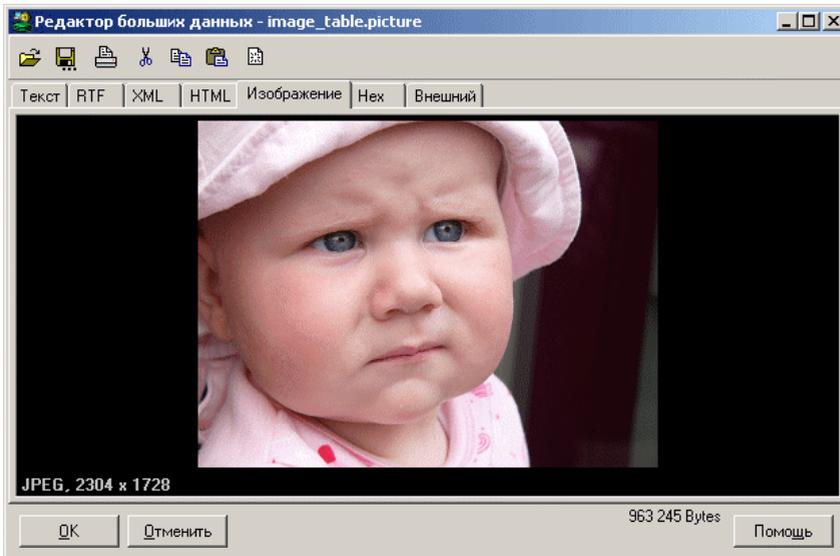
В этом дереве вы можете видеть окна, редакторы и строки, где результаты поиска были найдены. Щелкнув мышью на результате поиска, окно и редактор перейдут на передний план, а редактор перейдет к результату поиска и подсветит его. Вы можете закрепить список поиска, что бы сделать его доступным все время, и что бы он не закрывал другие окна.

На панели поиска вы дополнительно можете управлять такими опциями, как искать во всех редакторах или только в текущем, искать с учетом регистра или без, искать слова целиком или нет, и использовать регулярные выражения или нет.

19. Редактор массивных данных

Если вам понадобится просмотреть или отредактировать столбцы или значения переменных типов long, long raw, CLOB, BLOB or BFILE, будет вызван Редактор массивных данных. Он может быть дополнительно использован для редактирования данных varchar2. Редактор массивных данных анализирует данные и попытается отобразить их в корректном формате. Он может отображать значения как текст, как данные в форматах RTF, XML, HTML, рисунок (bmp, jpg, gif, tiff и другие) и Hex (шестнадцатеричные) данные. Более того, он может запустить внешнее приложение (например, MS Word или Acrobat Reader) для просмотра или редактирования данных.

При вызове Редактора массивных данных для рисунка вы увидите следующее:



Панель инструментов в верхней части редактора включает в себя следующие кнопки:

- Открыть – Открывает в редакторе содержимое файла.
- Сохранить – Сохраняет содержимое редактора в файл.
- Печать – Печатает содержимое редактора в текущем выбранном формате.
- Вырезать – Помещает содержимое редактора в буфер обмена в текущем выбранном формате и далее очищает (обнуляет) содержимое редактора.
- Копировать – Помещает содержимое редактора в буфер обмена в текущем выбранном формате.
- Вставить – Вставляет данные буфера обмена в редактор в текущем выбранном формате.
- Очистить – Очищает (обнуляет) содержимое редактора.

Под панелью инструментов вы можете выбрать страницу с корректным форматом содержимого редактора. Несколько форматов могут быть доступны для некоторых данных. Например,

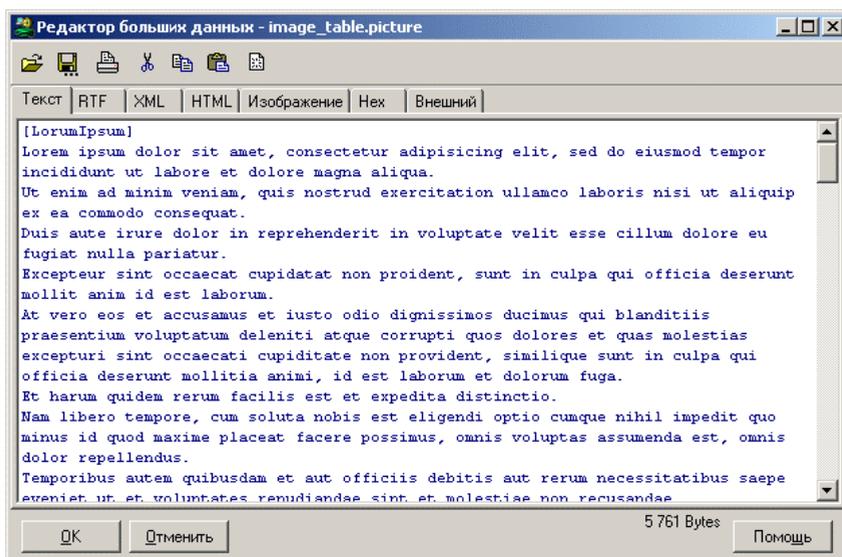
документ HTML может быть отображён либо в формате HTML, либо в текстовом формате. Все данные могут быть отображены в формате Hex (бинарный формат).

Когда вы вносите изменения в содержимое редактора определенного формата, то для просмотра этих изменений вы можете переключиться в другой формат. Например, при просмотре документа HTML вы можете переключиться в Текстовый формат, чтобы внести изменения, и вернуться в формат HTML, чтобы просмотреть их.

После нажатия кнопки ОК ваши изменения будут переданы функции, из которой был вызван Редактор массивных данных.

19.1 Редактирование текста

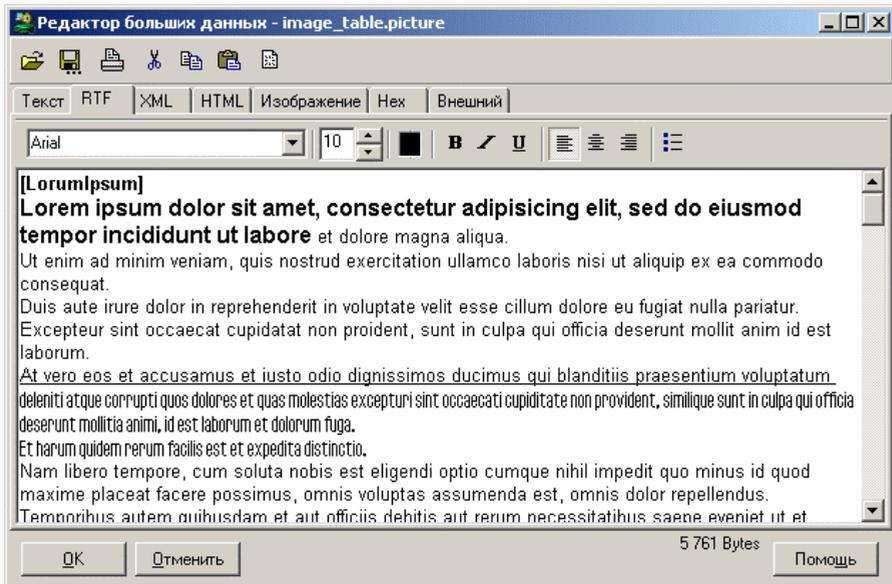
Для просмотра или редактирования любого вида текстовых данных может быть использован текстовый редактор:



Он также может быть использован для редактирования текста XML и HTML. При переключении на страницу XML или HTML будут немедленно показаны этих изменений.

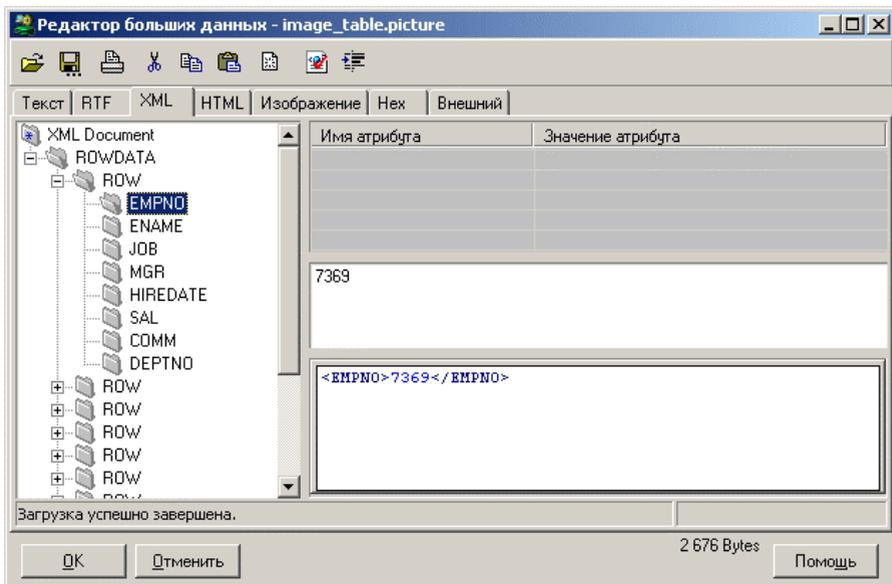
19.2 Редактирование RTF

Для просмотра и редактирования текста в формате Rich Text Format может быть использован редактор RTF. Он позволит вам легко изменять тип шрифта, его размер, цвет, стиль и выравнивание:



19.3 Редактирование XML

Во время просмотра содержимого XML, редактор первоначально переключится на страницу XML:

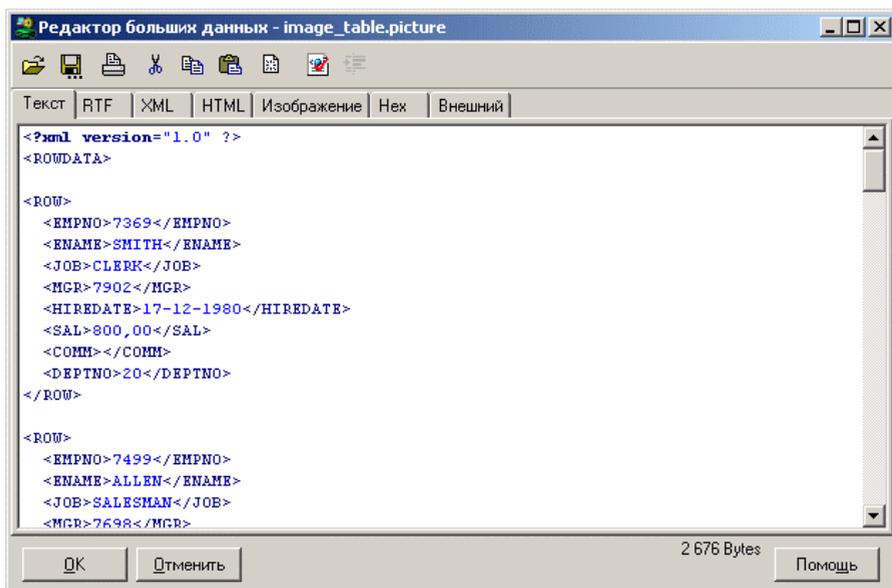


Здесь вы сможете увидеть структуру документа XML в виде дерева и отредактировать отдельные элементы и атрибуты.

При нажатии кнопки *Разбор XML* документ XML будет проанализирован, и будет представлен отчёт об ошибках. Для этого требуется Net8 8.1 или более поздняя версия для того, чтобы у пользователя был доступ к типу SYS.XMLTYPE или пакету XMLPARSER.

Чтобы применить к документу стандартные правила форматирования, вы можете нажать на панели инструментов кнопку *Форматировать XML*.

Если вы переключитесь на страницу текстового формата для XML документа, будет использовано соответствующее выделение синтаксиса.



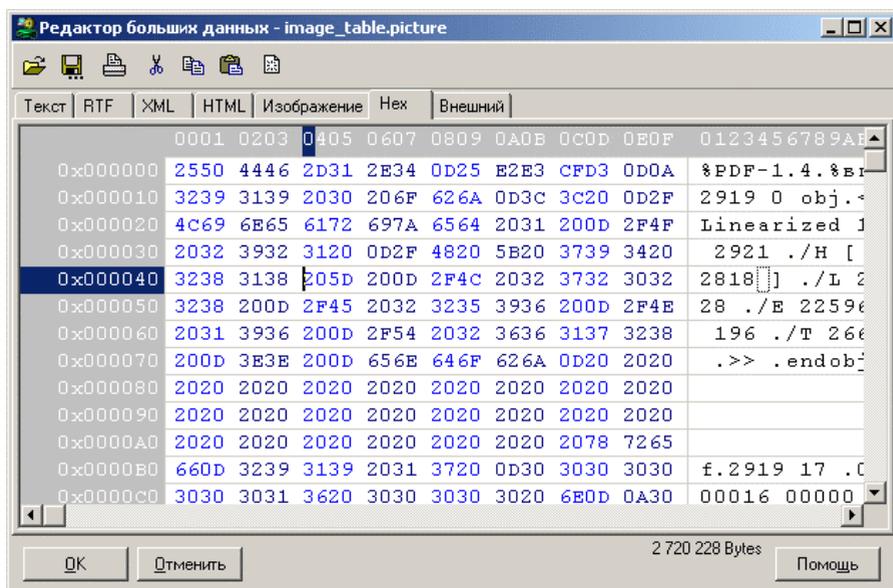
19.4 Редактирование рисунков

Редактор рисунков может отображать рисунки самых распространённых форматов, таких как JPG, BMP, GIF и так далее. Если рисунок больше, чем редактор может отобразить, рисунок будет сжат. Это не изменит содержимое, и является только визуальным эффектом. Под рисунком вы увидите формат рисунка и его размеры в пикселях.

Чтобы изменить рисунок, вы можете либо нажать на панели инструментов кнопку *Открыть*, чтобы загрузить его из файла, либо скопировать рисунок из другого источника в буфер обмена и нажать кнопку *Вставить*. Заметьте, что при копировании/вставке через буфер обмена, формат рисунка будет изменён на BMP. Чтобы убедиться, что первоначальный формат сохранён, вам следует загрузить его из файла.

19.5 Редактирование Hex данных

Чтобы просмотреть или отредактировать содержимое редактора в шестнадцатеричном формате, вы можете переключиться на страницу Hex:

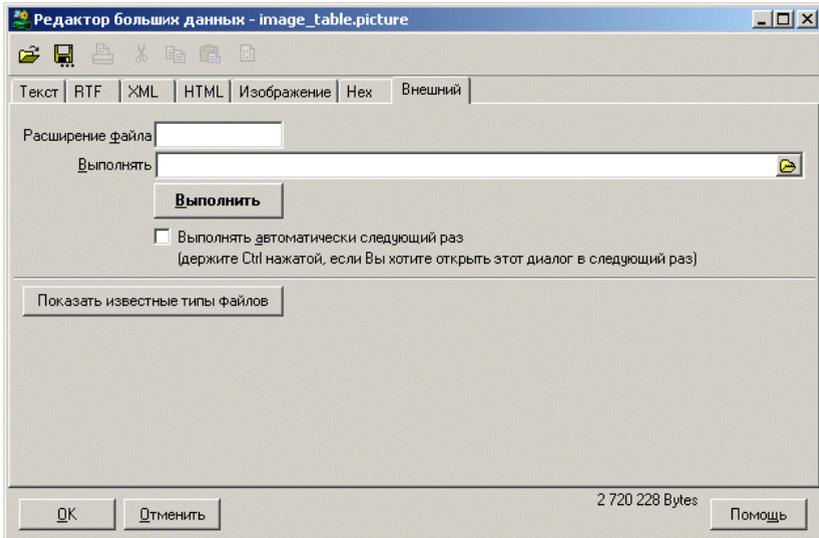


Слева вы видите 8 столбцов со словами (16 битов) шестнадцатеричных данных. Справа вы видите текстовое изображение шестнадцатеричных данных. Вы можете отредактировать шестнадцатеричное и текстовое изображения.

Если вы щёлкните правой кнопкой мыши на редакторе шестнадцатеричных данных, вы можете выбрать другой макет столбцов.

19.6 Вызов внешней программы просмотра или редактора

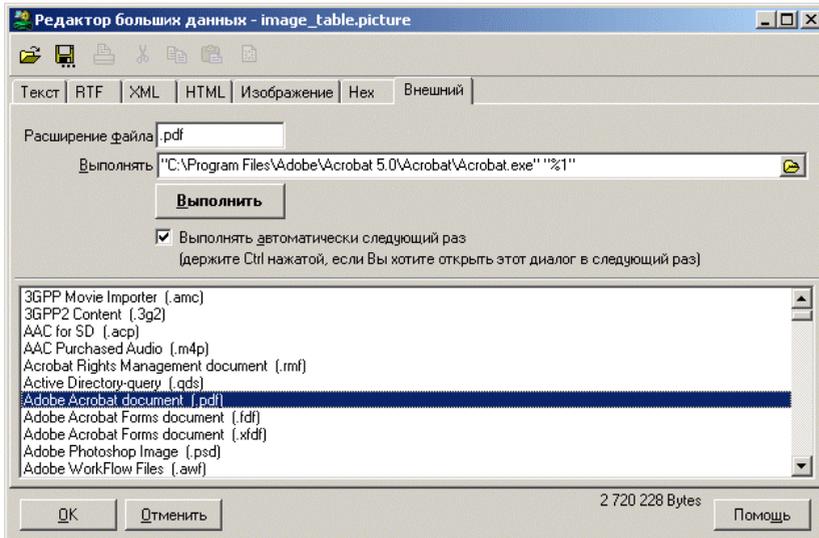
Для многих массивных объектов, которые хранятся в базе данных, может быть необходимым вызов внешних программ просмотра или редактирования. Например, если в столбце BLOB находится содержимое файла Acrobat PDF (Portable Document File), то будет необходимо вызвать Acrobat Reader. Когда вы вызываете Редактор массивных данных для подобного содержимого, вы будете немедленно переключены на страницу Текст или Hex. После этого вы можете переключиться на страницу Внешний:



Здесь вы можете определить, какая программа должна быть выполнена. Во-первых, вам необходимо определить расширения файла. Перед выполнением внешней программы, временный файл с содержимым редактора будет сохранён с данным расширением. Далее, вам нужно определить, какая программа должна быть выполнена. После этого, чтобы просмотреть содержимое во внешней программе, вы можете нажать кнопку *Выполнить*. Если вы измените файл во внешней программе и сохраните его, то Редактор массивных данных обнаружит эти изменения и позволит вам импортировать эти изменения обратно в редактор.

Во время просмотра данных столбцов, связь между таблицей/столбцом и внешней программой будет сохранена для вашего следующего просмотра. Если вы дополнительно включите опцию *Выполнять автоматически следующий раз*, Редактор массивных данных немедленно вызовет внешнюю программу. Чтобы отменить этот автоматический режим, во время вызова редактора удерживайте нажатой клавишу *Ctrl*.

Большинство программ будет иметь свои собственные типы файлов, и, вместо определения расширения и программы вручную, вы можете просто выбирать их из списка. Сначала нажмите кнопку *Показать известные типы файлов*. После этого выберите корректный тип файла и дважды щёлкните на нём. Для файла PDF это будет *Adobe Acrobat File*:



Если вы щёлкните правой кнопкой мыши на списке типов файла, вы сможете отсортировать его по расширению или описанию (по умолчанию).

20. Построитель запросов

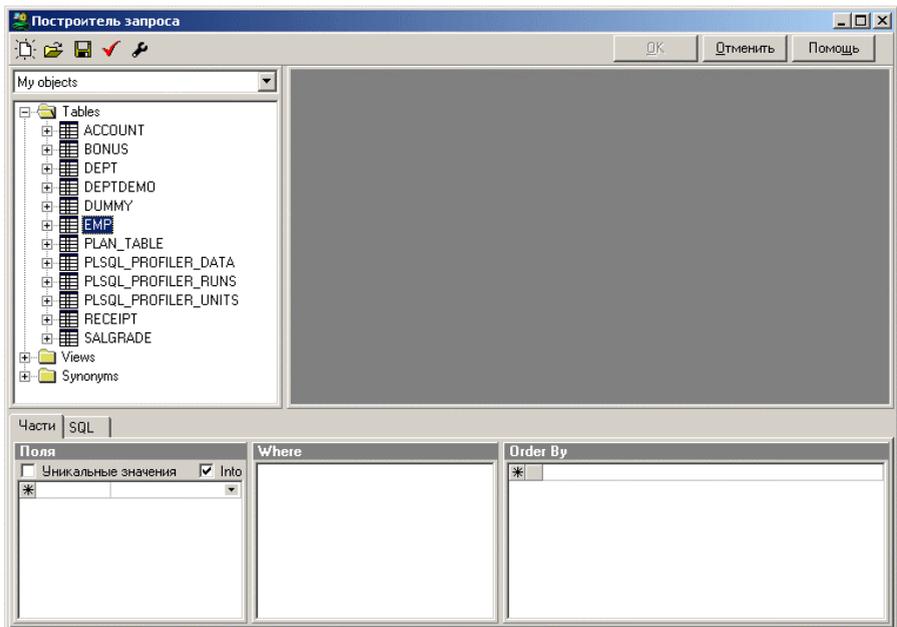
Построитель запросов позволит вам создавать и изменять операторы select в ваших исходных файлах PL/SQL и SQL. Он обеспечивает простой интерфейс “point and click” чтобы составить список полей и таблиц, предложения where и order by оператора select.

20.1 Создание нового оператора select

Для этого примера предположим, что мы хотим создать соединение таблиц dept и emp и показать некоторые столбцы из обеих таблиц:

```
select e.empno, e.ename, d.deptno, d.dname
from emp e, dept d
order by e.empno
```

Чтобы создать новый оператор select, поместите курсор редактора в то местоположение в тексте, куда вы хотите его вставить и нажмите на панели инструментов кнопку *Построитель запросов* (или выберите в меню *Инструменты* пункт *Построитель запросов*). При этом появится пустая форма Построителя запросов:

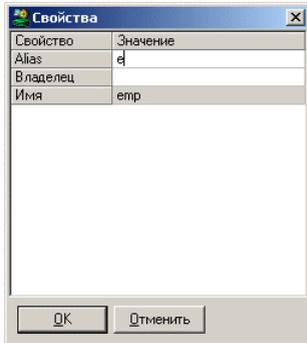


С левой стороны вы видите Объектный навигатор, содержащий только таблицы и представления. Справа вы видите рабочую область, которая сейчас пуста. Внизу вы видите 3 панели со списком полей (и, опционально, список элементов *into*) и фразами where и order by. Нижняя секция содержит страницу, которая позволит вам переключаться к фактическому тексту SQL, который будет создан для текущего определения запроса.

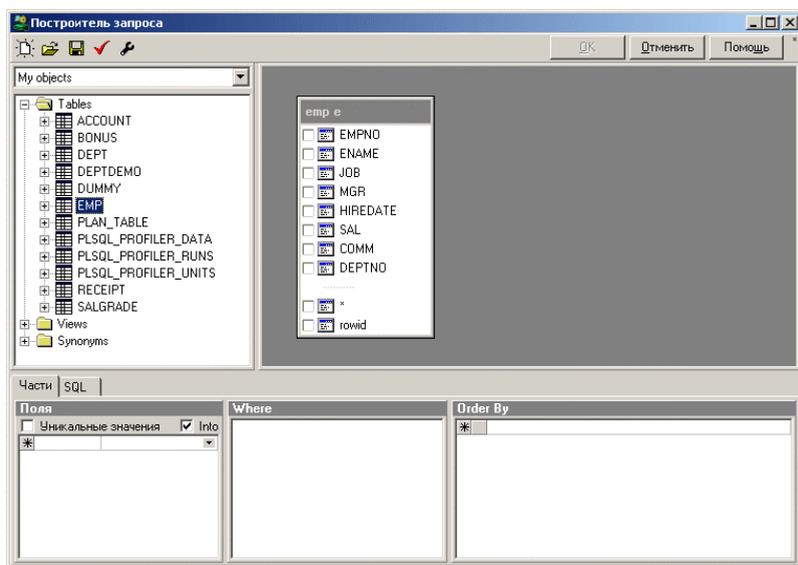
Вверху окна вы видите панель инструментов с кнопками *Создать* для создания нового запроса, *Загрузить* для загрузки из файла ранее сохранённого определения запроса, *Сохранить* для сохранения текущего определения запроса в файл, *Проверить* для проверки текущего

определения запроса и *Предпочтения*. Кнопка *OK* вернёт вас либо к PL/SQL редактору, либо к SQL редактору и вставит текст SQL, который вы построили.

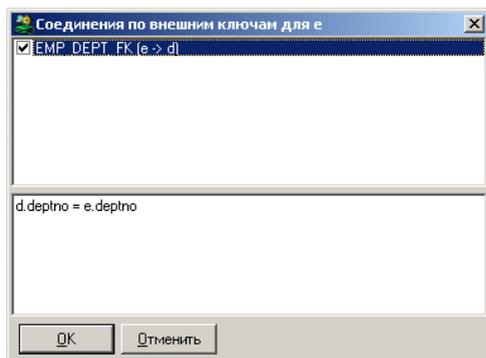
Чтобы включить таблицы или представления в оператор *select*, перетащите их из Объектного навигатора в рабочую область. Каждый раз, когда вы добавляете новую таблицу, появляется диалог *Table Properties*:



Здесь вы можете указать алиас таблицы. Для нашего примера мы решили использовать 'e' для таблицы *emp*:

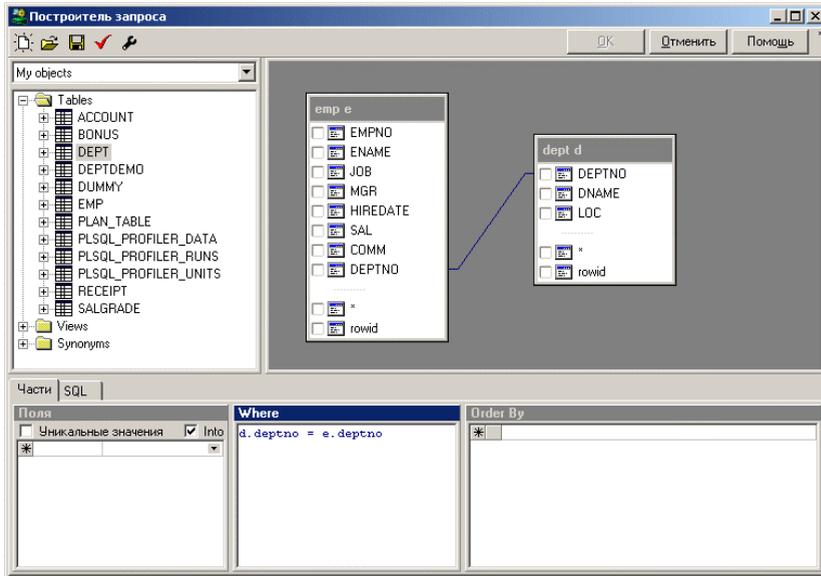


После добавления подобным способом таблицы *Dept*, вам будет задан вопрос, хотите ли вы объединить 2 таблицы, используя столбцы внешнего ключа с именем *EMP_FOREIGN_KEY*:

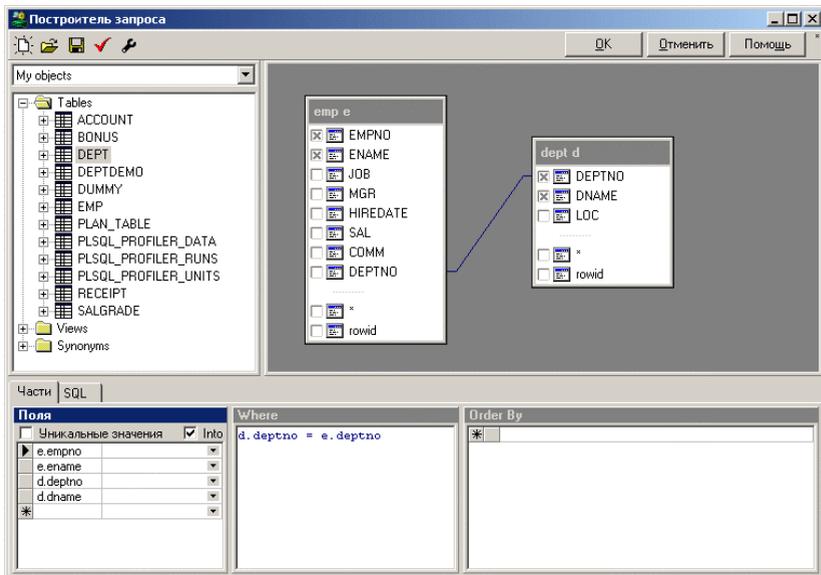


После включения этой опции, в нижней половине формы вы увидите новое условие соединения.

Теперь таблица *Dept* добавлена, и в рабочей области отображено соединение:

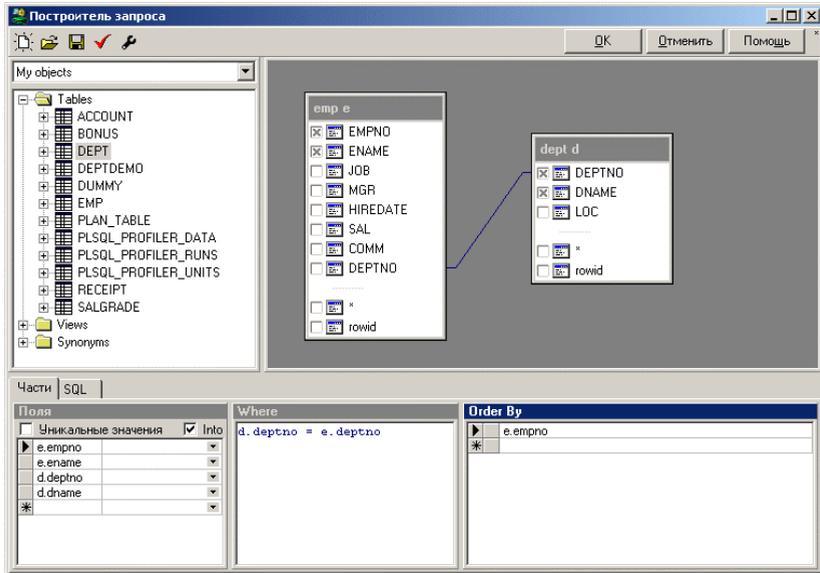


Что нам необходимо сделать, чтобы включить столбцы в список полей оператора select? Просто щёлкнуть на переключателе рядом с именами столбцов. Столбцы будут отображены на панели *Поля*:



Если вы хотите создать оператор PL/SQL `select .. into`, вам также понадобится указать элементы для списка `Into`, находящегося справа от полей. Список выбора отобразит переменные PL/SQL, которые определены в коде PL/SQL, куда вы вставляете оператор. В данном случае, мы не хотим создавать список `Into`, так что мы оставим эти элементы пустыми.

Чтобы создать предложение `order by`, перетащите столбец `empno` на панель `Order By`:



Теперь, чтобы вставить оператор SQL в редактор, вы можете нажать кнопку `OK`.

20.2 Изменение существующего оператора select

Чтобы изменить в вашем исходном PL/SQL или SQL коде существующий оператор `select`, просто щёлкните правой кнопкой мыши на операторе `select` и выберите во всплывающем меню пункт *Построитель запросов*. Оператор `select` будет выделен в редакторе целиком, и будет отображён в *Построителе запросов*. Теперь вы можете изменить определение запроса, как это описано в предыдущей главе.

20.3 Изменение определения запроса

В следующих параграфах описано, как вы можете менять определение запроса.

Изменение списка полей

Чтобы добавить поле из таблицы в рабочей области в запрос, вы можете щёлкнуть на переключателе рядом со столбцом, или же вы можете перетащить столбец в список полей. Альтернативно, если список полей имеет фокусировку, вы можете дважды щёлкнуть на столбце.

Вы также можете перетащить столбец или папку столбцов из Объектного навигатора в список полей. Под таблицей находятся папки столбцов, ограничений и индексов.

Вы можете ввести имена столбцов и другие выражения прямо в списке полей.

Чтобы удалить поле из списка полей, щёлкните на нём правой кнопкой мыши и выберите из всплывающего меню *Удалить выбранные элементы*. Если вы выберете несколько полей, вы можете удалить их за один раз.

Вы можете переместить поля в другое местоположение, выбрав их и перетащив их на их новое местоположение.

Если вы хотите иметь в результирующем наборе только различные значения, включите соответствующую опцию над списком полей.

Использование алиасов полей

Чтобы использовать алиас для поля, просто введите алиас после выражения, точно также, если бы вы вводили оператор select вручную.

Изменение фразы where

Фраза where представлена нормальным текстовым редактором, где вы можете вводить условия. Чтобы включить столбец, вы можете перетащить его из таблицы в рабочую область, или же, если предложение where имеет фокус, дважды щёлкните на столбце.

Чтобы добавить условие соединения, основанное на определении внешнего ключа, щёлкните правой кнопкой на таблице в рабочей области и выберите из всплывающего меню пункт *Внешние ключи*. Вы не можете использовать эту функцию, чтобы удалить существующее условие соединения, но, вместо этого, вы можете удалить соответствующий текст в редакторе.

Изменение фразы order by

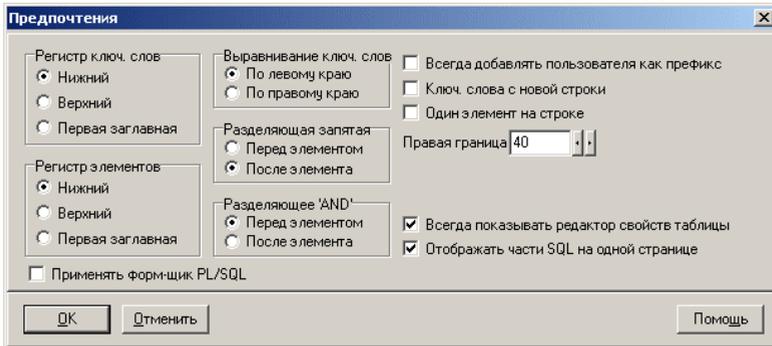
Фразой order by можно управлять таким же образом, как и списком полей, за исключением того, что вы не можете использовать переключатели рядом с именами столбцов. Чтобы задать элементу восходящий или нисходящий порядок, вы можете щёлкнуть на кнопке рядом с элементом, которая отобразит соответствующую стрелку.

Использование синонимов для таблиц и представлений

Чтобы использовать таблицу или представление через синоним, щёлкните правой кнопкой на рабочей области и выберите из всплывающего меню пункт *Добавить таблицу*. Введите имя синонима в свойстве *Имя*, задайте алиас, если это необходимо, и нажмите кнопку *OK*.

20.4 Предпочтения Построителя запросов

Нажав на панели инструментов кнопку *Предпочтения*, вы можете изменить поведение Построителя запросов. Появится следующая форма:



Большинство этих предпочтений влияют на макет сгенерированного оператора select:

- Регистр ключевых слов**
 Определяет, как ключевые слова SQL (select, from, into, where, order by) должны быть вставлены в сгенерированный оператор SQL.
- Регистр элементов**
 Определяет, как элементы (table names, column names) должны быть вставлены в сгенерированный оператор SQL.
- Выравнивание ключевых слов**
 Определяет, как будут выровнены ключевые слова SQL – по правому или по левому краю.
- Разделяющая запятая**
 Определяет, будут ли запятые поставлены перед следующим элементом или после текущего элемента. Это представляет интерес, прежде всего в случае, когда включена опция *Один элемент на строке*.
- Разделяющее 'AND'**
 Определяет, будет ли во фразе where оператор 'AND' помещён перед следующим условием или же после текущего условия.
- Всегда добавлять пользователя как префикс**
 Когда эта опция включена, имена таблиц будут начинаться с владельца (например, вместо emp – scott.emp). Когда она отключена, имена таблиц будут начинаться с владельца, только если они не принадлежат текущему пользователю.
- Ключевые слова с новой строки**
 Когда эта опция включена, ключевое слово SQL начинает новую строку, и все элементы помещаются на последующие строки с отступом в 2 символа. Если опция отключена, элементы будут следовать непосредственно за ключевым словом на той же строке.
- Один элемент на строке**
 Когда эта опция включена, каждый элемент будет размещаться на отдельной строке. Когда она отключена, элементы будут располагаться на одной строке, пока не будет достигнута *Правая граница*.

- **Правая граница**
Определяет максимальную длину каждой строки, когда отключено предпочтение *Один элемент на строке*.
- **Всегда показывать редактор свойств таблицы**
Когда включена эта опция, то после добавления новой таблицы в определение запроса всегда будет отображаться редактор свойств таблицы. Когда эта опция отключена, свойство будет отображаться только в том случае, если новая добавляемая таблица уже используется в запросе, и поэтому у нее должен быть алиас.
- **Отображать части SQL на одной странице**
Определяет, будут ли 3 части SQL (список field, фраза Where, фраза Order By) отображаться на одной странице. Если эта опция выключена, каждая часть будет отображаться на отдельной странице, следующей за страницей текста SQL.

20.5 Модули расширения строителей запросов

Если Модуль расширения строителя запросов установлен, тогда стандартная функция *Построитель запросов* вызовет этот модуль расширения. Щелкнув правой кнопкой мыши на кнопке *Построитель запросов*, вы сможете выбрать строителя запросов из списка все установленных строителей запросов. Последний использованный строитель запросов будет вызван по умолчанию, когда вы позже снова вызовете функцию *Построитель запросов*.

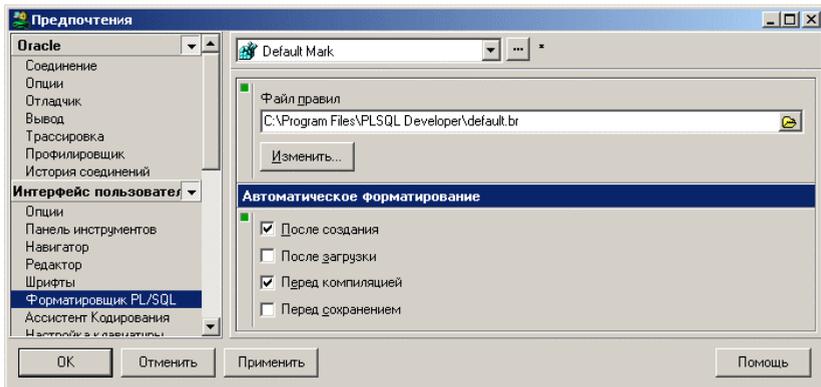
21. Форматировщик PL/SQL

Если вы разрабатываете и поддерживаете код PL/SQL в составе команды с несколькими разработчиками, будет очень полезным, если все будут использовать одинаковый стиль кодирования. Это благоприятно скажется и на качестве, и на производительности. Для этого PL/SQL Developer включает в себя Форматировщик PL/SQL.

Всё, что вам нужно сделать, это установить опции Форматировщика PL/SQL и, возможно, определить стиль кодирования при помощи правил Форматировщика PL/SQL (вы также можете просто адаптировать правила по умолчанию). После этого ваш PL/SQL код будет автоматически отформатирован во время вашей работы или будет явно отформатирован после вызова форматировщика.

21.1 Определение опций

Чтобы определить опции Форматировщика PL/SQL, выберите соответствующий пункт из меню *Правка*. Появится следующий диалог предпочтений:



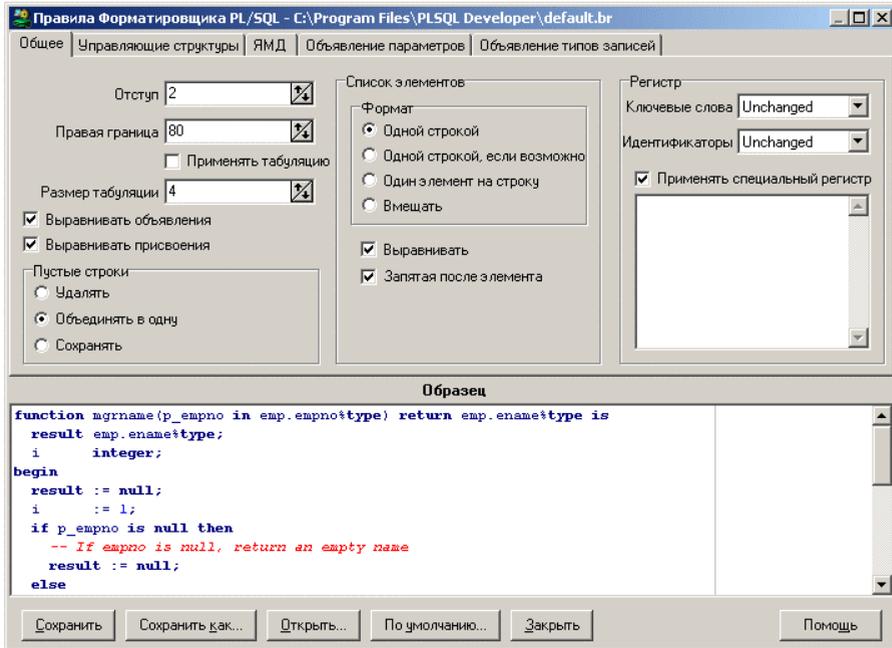
Файл правил определяет правила, которые будут соблюдаться во время форматирования вашего кода. Вы можете оставить эту опцию пустой, чтобы принять правила, установленные по умолчанию при помощи файла *default.br* в установочном каталоге PL/SQL Developer. Для большей информации об этих правилах смотрите следующую главу.

Более того, вы сможете определить, когда ваш код PL/SQL будет автоматически отформатирован:

- **После создания** - Как только создан новый программный файл из шаблона или оператор DML перетащен из Объектного навигатора и помещён в Программное окно.
- **После загрузки** – После того, как программный файл загружен из файловой системы. Это не повлияет на код, который открыт из базы данных Oracle.
- **Перед компиляцией** – Когда вы компилируете программу в Программном окне. Это гарантирует, что все PL/SQL тексты в базе данных Oracle соответствует правилам.
- **Перед сохранением** – Перед тем, как программный файл будет сохранён в файловой системе.

21.2 Определение правил

Если вы нажмете в диалоге Предпочтения кнопку *Изменить*, вы сможете просмотреть или отредактировать правила выбранного файла:



После внесения изменений в эти правила, чтобы сохранить эти изменения вы можете нажать кнопки *Сохранить* or *Сохранить как*. Для просмотра или редактирования, вы также можете открыть другой файл правил, также, вы можете вернуться к правилам, установленным по умолчанию, для этого нажмите соответствующую кнопку. Если вы сохраните файл под новым именем или откроете другой файл, вам нужно выбрать этот файл правил в диалоге Предпочтения, чтобы сделать эти правила эффективными.

На странице *Общее* вы можете изменить правила, которые применяются к основным аспектам вашего PL/SQL кода. Все изменения, которые вы вносите, немедленно будут отображены в области окна *Образец* в нижней части диалога, и большинство опций являются очевидными. Следующие опции нуждаются в некотором объяснении:

- **Отступ** – Число пробелов, которые будут вставлены в начале для вложенных структур, таких как begin/end, if/then/else, циклы и так далее.
- **Правая граница** – Когда необходимо перенести код на следующую строку, правое поле будет использовано как направляющая. Могут возникнуть ситуации, когда текст может заходить за пределы правой границы, например, когда используются длинные строковые переменные.
- **Применять табуляцию** – Когда эта опция включена, результирующий код будет содержать символы табуляции (глава 9) для отступов. Когда она отключена, будут использованы пробелы.

- **Размер табуляции** – Число символов, которое представляет символ табуляции. Это полезно не только при включённой опции *Применять табуляцию*, но также может быть полезным для корректного выравнивания комментариев, которые содержат символы табуляции.
- **Пустые строки** – Вы можете определить, что пустые строки необходимо удалять, что форматировщик будет вставлять пустые строки по своим собственным правилам, и что группы повторяющихся пустых строк будут объединены в одну пустую строку, или все пустые строки будут сохранены.
- **Список элементов** – Эти правила применимы ко всем спискам элементов, которые не входят в категории указанные на других страницах.
- **Регистр** – Определяет регистр ключевых слов и идентификаторов PL/SQL. Опция *Применять специальный регистр* позволит вам определить список слов, регистр которых будет сохранен. Если, например, вы включите в этот список *DBMS_Output*, то это слово будет приведено именно к такому виду в вашем PL/SQL коде.

Другие страницы содержат правила, которые применимы к определённым аспектам вашего PL/SQL кода. Эти правила очевидны и их действие продемонстрировано в области предварительного просмотра.

21.3 Использование форматировщика

Если вы не выбрали никакие форматировочные события в диалоге Предпочтения, вы можете отформатировать ваш код, выбрав в меню *Правка* пункт *Форматировщик PL/SQL*, нажав соответствующую кнопку на панели инструментов, или используя комбинацию клавиш, которая назначена для него. По умолчанию, весь код в текущем редакторе PL/SQL будет отформатирован. Если в коде присутствует критическая синтаксическая ошибка, он не будет отформатирован, и вы получите сообщение об ошибке.

Альтернативно, вы можете выбрать завершённую часть PL/SQL кода (например, локальная функция или процедура или оператор DML) и вызвать форматировщик.

22. Шаблоны

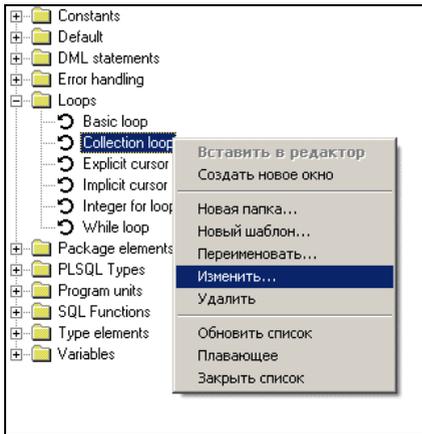
Когда вы создаёте новый программный модуль, его первоначальное содержимое основано на шаблоне, который содержит predefined текст и переменные. Когда вы выбираете шаблон, у вас будут запрошены значения этих переменных, которые будут потом подставлены в текст шаблона.

Шаблоны также могут быть использованы для вставки частей кода SQL или PL/SQL в существующий исходный текст. Вы могли бы, например, создать шаблон для курсорного цикла, и использовать его, когда вам будет нужно запрограммировать подобную управляющую структуру.

Шаблоны являются текстовыми файлами, находящимися в каталоге *Template* и имеющими расширение *.tpl*. Шаблоны predefined для всех типов программных модулей, также как и для основных SQL и PL/SQL конструкций. Вы можете изменять и создавать шаблоны для ваших собственных специфических требований.

22.1 Окно шаблонов

Окно шаблона отображает иерархическую структуру каталога шаблонов. Это позволит легко организовать ваши шаблоны. Когда вы запускаете PL/SQL Developer в первый раз, Окно шаблонов будет закреплено под Объектным навигатором. Таким образом, оно всегда доступно и никогда не затрудняет просмотр других окон документа:



Существует набор predefined шаблонов, они организованы в следующих папках шаблонов:

- Константы – Шаблоны для объявления PL/SQL констант.
- По умолчанию – Шаблоны с содержимым по умолчанию Окна SQL, Тестового окна, Командного окна или Окна Explain Plan. Она также содержит шаблон для нового представления. Вероятно, вы возможно не будете использовать эти шаблоны напрямую, но вы можете содержать их здесь.
- DML операторы– Шаблоны для DML операторов, которые вы можете использовать в PL/SQL.
- Циклы – Шаблоны для различных видов циклов PL/SQL.

- Элементы пакетов – Шаблоны для элементов, которые могут быть использованы в спецификациях или телах пакетов.
- PL/SQL Типы– Шаблоны для всех типов, которые могут быть объявлены в PL/SQL.
- Программные модули – Это специальная папка, которая содержит шаблоны для новых программных модулей. Когда вы создаёте новый программный модуль, будет использован шаблон из этой папки с таким же именем, как и название типа программного модуля.
- SQL Функции– Шаблоны со всеми SQL функциями, такими как add_months, substr и так далее. Эти шаблоны могут быть полезны, если вы не знаете точное имя или значение параметров для некоторой функции.
- Элементы типов – Шаблоны для элементов, которые могут быть использованы в спецификациях или телах типов.
- Переменные – Шаблоны для объявлений PL/SQL переменных.

По умолчанию, Окно шаблонов закреплено под Объектным навигатором, но вы можете сделать его плавающим окном, щёлкнув на нём правой кнопкой мыши и выбрав из всплывающего меню пункт *Плавающее*. Чтобы закрепить окно снова, повторите эту процедуру.

Вы можете закрыть Окно шаблонов, выбрав во всплывающем меню пункт *Закрыть список*. Чтобы сделать его снова видимым, выберите в меню *Инструменты* пункт *Шаблоны*.

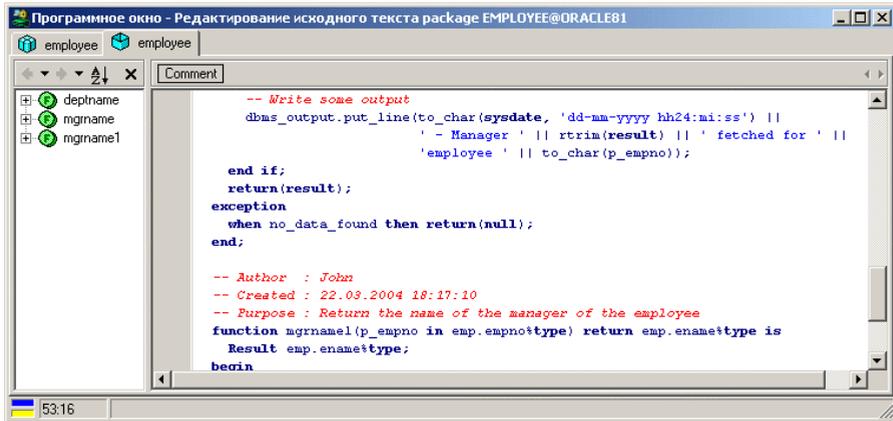
Вы можете сохранить этот макет для следующего запуска PL/SQL Developer, используя пункт *Сохранить макет* меню Окно, который сохранит текущую ситуацию. Более подробно эта функция описана в главе 28.2.

22.2 Использование шаблонов

Чтобы вставить шаблон в существующий исходный текст, вы можете просто дважды щёлкнуть на нём. Если шаблон содержит переменные, которые должны быть указаны пользователем, то перед вводом текста появится форма ввода. Если, например, вы дважды щёлкните на шаблоне *Function implementation*, появится следующая форма ввода:

Function Implementation	
Purpose	Return the name of the manager of the employee
Name	mgrname1
Parameters	p_empno in emp.empno%type
Return type	emp.ename%type

После заполнения формы и нажатия кнопки *OK*, шаблонный текст будет вставлен в текущую позицию курсора в исходном тексте:



Вы также можете перетащить шаблон из Окна шаблонов и поместить его в некоторую позицию в редакторе. Эта позиция также будет определять сдвиг шаблонного текста. Аналогично, вы можете щёлкнуть правой кнопкой мыши на местоположении в редакторе и выбрать нужный шаблон в подменю *Шаблоны* всплывающего меню.

22.3 Создание и изменение шаблонов

Чтобы изменить существующий шаблон, щёлкните на нём правой кнопкой мыши и выберите из всплывающего меню пункт *Изменить*. При этом появится текстовый редактор с текстом шаблона. Этот текст содержит полную спецификацию шаблона, включая переменные, запросы и так далее. Формат текста шаблона описан в следующей главе.

Чтобы создать новый шаблон, щёлкните правой кнопкой мыши на папке, в которой вы хотите создать его и выберите во всплывающем меню пункт *Новый шаблон*. Сначала, у вас будет запрошено имя шаблона, после этого появится текстовый редактор шаблона. Вы также можете создать новую папку, выбрав из всплывающего меню пункт *Новая папка*.

Шаблонный текст

Спецификация шаблона содержит символьный текст и переменные. Символьный текст будет просто скопирован в исходный файл без какого-либо изменения. Рассмотрим шаблон *Commit*, который содержит только символьный текст *commit*. Если дважды щёлкнуть на этом шаблоне, текст *commit* будет вставлен в исходный текст программы, в то место, где находится курсор. Квадратные скобки указывают на переменную в шаблоне (смотрите ниже), так что, если вы хотите использовать квадратные скобки как символьный текст, то вместо них используйте двойные квадратные скобки. Например: `[[option]]`

Пользовательские переменные

Шаблонные переменные, являются кодами в шаблонном тексте, которые будут заменены подстановочным текстом. Этот подстановочный текст может быть неявной величиной, как текущая дата или имя пользователя, или может быть указано пользователем. Эти пользовательские переменные появятся в форме ввода, когда шаблон будет вызван. Следующий пример определяет переменные *Name* и *Type* для шаблона функции:

```

create or replace function [Name] return [Type] is
begin
    return(result);
end [Name];

```

Когда вызван этот шаблон, пользователь может определить значения для переменных *Name* и *Type*. Как вы видите, переменная *Name* используется дважды. Хотя у пользователя она будет запрошена только один раз, оба ее вхождения она будет заменена одним и тем же значением.

Следующая функциональность доступна для пользовательских переменных:

- Значение по умолчанию.
Значение по умолчанию для переменной может быть просто помещено после имени переменной. Чтобы определить `varchar2`, как тип функции по умолчанию, вы можете использовать следующее объявление:
[Type = varchar2]
- Ограничивающий список.
Чтобы ограничить возможные значения для переменной, вы можете определить её как список переменных. Чтобы ограничить типы функции значениями `varchar2`, `number` и `date`, вы можете использовать следующее объявление:
[Type = varchar2, *number, date]
Значение, которому предшествует звёздочка, будет использоваться по умолчанию.
- Открытый список.
Чтобы предоставить пользователю список возможных значений, а также позволяющий и другие значения для переменной, используйте ... в качестве последней переменной в списке:
[Type = varchar2, number, date, ...]
- Список с описаниями.
Вместо использования одного и того же текста для элементов списка и подстановочного текста, вы также можете использовать описание для каждого отдельного элемента. Для этого после описания укажите значение, отделенное двоеточием:
[Level = Write No Database State:WNPS, Read No Database State:RNDS, ...]
- Флажок.
Для переменных, которые фактически являются опциями, вы можете использовать флажок. Поместите слеш между значениями выключенного (слева) и включённого (справа) флажка. Следующая переменная вставит текст *for each row*, когда флажок отключён, и вставит *for each statement*, когда флажок включён:
[Statement level? = for each row / for each statement]

Если вы хотите использовать в подстановочном тексте специальные символы (такие как запятые, скобки и так далее), вы можете поместить текст между двойными кавычками.

Далее приведён пример шаблона для триггера, использующий только пользовательские переменные:

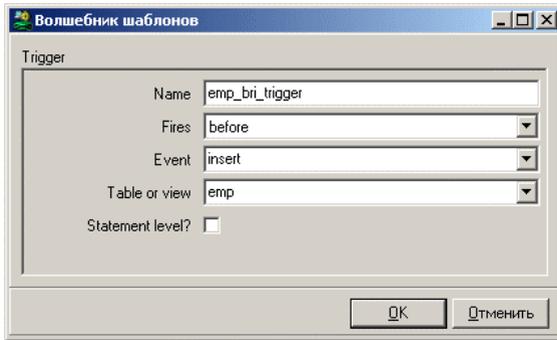
```

create or replace trigger [Name]
[Fires = before, after, instead of] [Event = insert, update, delete, ...]
on [Table or view]
[Statement level? = for each row/for each statement]
declare
    -- local variables here
begin
    ;
end [Name];

```

Заметьте, что вы можете создавать несколько программных модулей в одном шаблоне, разделяя их строкой с символом слеш (/). Таким образом, вы можете создать шаблоны для спецификации и тела пакета в одном программном файле.

Когда используется этот шаблон, у пользователя будут запрошены значения переменных следующим образом:



Неявные переменные

Помимо пользовательских переменных, вы можете дополнительно использовать неявные переменные. Подстановочный текст для этих переменных не указан пользователем, но определен системой (дата, имя пользователя) или разработчиком шаблона (запросы, текст).

Системные переменные

Определены следующие 4 системные переменные:

- `$OSUSER` – Имя пользователя Операционной системы.
- `$DBUSER` – Имя текущего подсоединённого пользователя базы данных.
- `$DATE` – Текущая дата.
- `$TIME` – Текущее время.

Первые 2 строки следующего примера шаблона вставляют пользователя Windows и текущие дату/время в исходный файл:

```
-- Author : $OSUSER
-- Created : $DATE $TIME
-- Purpose : [Purpose]
procedure [Name] is
begin
;
end [Name];
```

Заметьте, что для системных переменных в тексте шаблона не указываются квадратные скобки, но имя переменной начинается со знака \$.

Если вы хотите использовать в результирующем тексте первоначальное имя системной переменной вместо подстановочного значения, поместите перед именем второй знак \$.
Например:

```
-- $$Date$ $Revision$
```

Результирующий текст будет \$Date\$ \$Revision\$.

Позиция курсора

Переменная позиции курсора определяет, где в редакторе будет помещён текстовый курсор после того, как шаблонный текст вставлен в редактор. Просто поместите [#] в желаемой позиции:

```
loop
  [#]
end loop;
```

Переменные запроса

Вы можете использовать запросы, чтобы заполнить списки выбора в шаблонах. В следующем примере определён запрос *seq_query*, который впоследствии будет использован как список значений для переменной *Sequence*:

```
[$QUERY seq_query =
  select lower(object_name) from user_objects
  where object_type = 'SEQUENCE'
  order by object_name]
select [Sequence=$seq_query,...].nextval into [Variable name] from dual;
```

Переменная *Sequence* является открытым списком, потому что запрос сопровождается “...”, при этом подразумевается, что другие значения могут быть введены пользователем вручную. Результаты запроса можно легко просмотреть как разделённые запятыми списки значений.

Включённый и исключённый текст

Вы можете включить или исключить текст из шаблона, в зависимости от значения другой переменной. В следующем примере, слово *where* будет добавлено, если *Search condition* введено пользователем:

```
select [Item list]
  into [Variable list]
  from [Table list]
  [+Search condition=where] [Search condition];
```

В результате, пользователю не придётся вводить ‘where’ в условии поиска. Чтобы исключить часть текста, вместо синтаксиса *+Variable name*, используйте синтаксис *-Variable name*.

Текстовые переменные

Вы можете определять текстовые переменные в шаблоне и ссылаться на эти переменные в других частях шаблона. Это особенно полезно, если вы хотите условно включить массивные части текста. В следующем примере, включается блокировка исключительной ситуации, если это указано пользователем:

```
[$TEXT exception_block=  
exception  
  when no_data_found then ...  
  when too_many_rows then ...  
  when others then ...  
end;]  
select [Item list]  
  from [Table list]  
  into [Variable list]  
  where [Search condition];  
[Exception block =/$exception_block]
```

Значки шаблонов

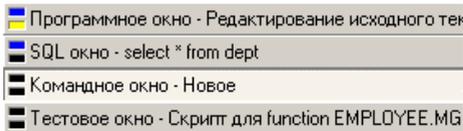
Дерево в Окне шаблонов отображает значок для каждого шаблона. Сначала Окно шаблонов ищет файлы рисунков с таким же именем, как и шаблон. Для шаблона Cursor.tpl будет отображён Cursor.bmp из того же каталога, если он присутствует. Если файл рисунка не найден, то поиск будет продолжен в каталогах ниже текущего каталога. Если файл рисунка всё равно не найден, то в текущем каталоге и в каталогах ниже файл будет искаться default.bmp.

Файлы рисунков для значков шаблонов должны иметь размер 16x16 пикселей с 16 цветами.

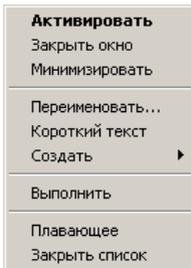
23. Список окон

Список окон позволит вам легко перемещаться между несколькими окнами документов. Эта функция особенно полезна, если вы любите работать с максимизированными окнами документов, и у вас есть много открытых окон, подобный список облегчит навигацию.

По умолчанию, Список окон закреплён под Объектным навигатором и Окном шаблонов. Он видим, только если открыто одно или несколько окон, и показывает названия и статус этих окон:



Если вы щёлкните на одном из этих элементов, будет активировано соответствующее окно. Если вы щёлкните правой кнопкой мыши на элементе Списка окон, будет отображено следующее всплывающее меню:



Из этого всплывающего меню, вы можете активизировать, закрыть или изменить размер окна на нормальный или минимизированный. Вы также можете закрыть окно, щёлкнув с нажатой клавишей shift, на элементе в списке.

Более того, вы можете переименовать окна. Это может быть полезным, если вы только что создали новое окно и хотите дать ему описательное имя вместо, например, *SQL окно – Новое*.

Пункт меню *Создать* позволит вам создать новое Программное окно, SQL окно, Тестовое окно, Командное окно или Окно плана выполнения.

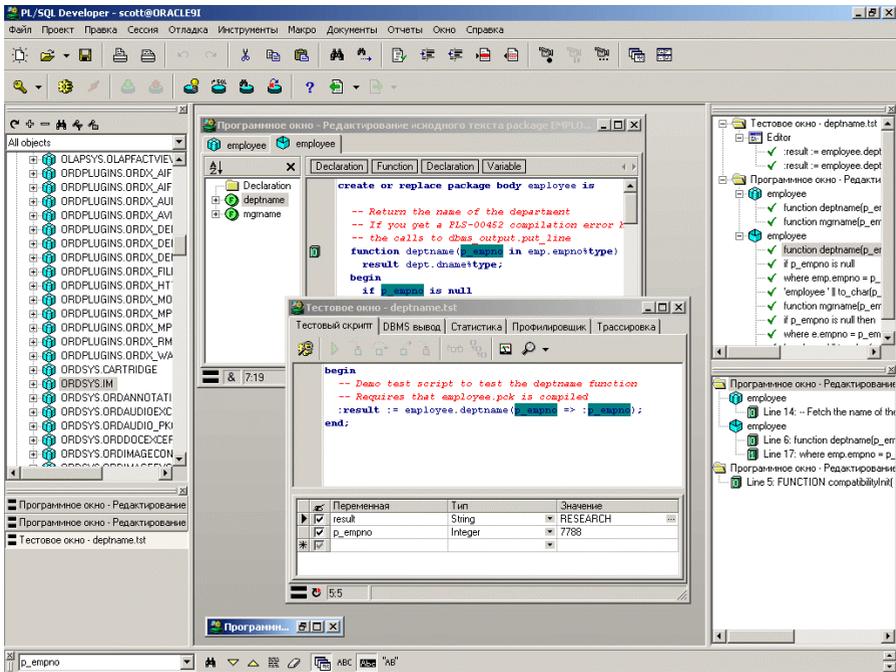
Чтобы выполнить данное окно, вы можете выбрать пункт меню *Выполнить*.

Вы можете сделать Список окон плавающим окном, выбрав пункт *Плавающее*. Чтобы закрепить окно снова, повторите эту же процедуру. Чтобы закрыть список окно, выберите пункт *Закрыть список*. Чтобы сделать Список окон снова видимым, выберите в меню *Инструменты* пункт *Список окон*. Вы можете сохранить этот макет для следующего вашего запуска PL/SQL Developer, используя в меню *Окно* пункт *Сохранить макет*, который сохранит текущую ситуацию. Более подробно эта функция описано в главе 28.2.

24. Закрепленные и плавающие инструменты

Следующие инструменты могут быть закреплены или могут быть представлены в виде плавающих окон:

- Навигатор объектов
- Список окон
- Список шаблонов
- Список закладок
- Панель поиска
- Результаты поиска



Что бы закрепить инструмент, просто перетащите его влево, вправо, вверх или вниз рабочей области. Как только вы подойдете близко к краю рабочей области, будет отображена рамка, указывающая место закрепления. Если другой инструмент уже закреплен в этом месте, вы можете перетащить новый инструмент выше, ниже, левее, правее или внутрь другого инструмента. Если вы перетащите один инструмент внутрь другого, они станут доступными в этом же самом месте, но на разных страничках (см. Список окон и Список шаблонов на экранном снимке выше).

Что бы сделать инструмент плавающим, просто перетащите его из закрепленной позиции в плавающую позицию.

25. Авторизация

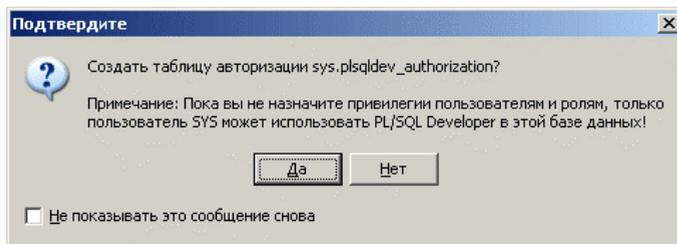
После установки PL/SQL Developer все пользователи могут использовать все функциональные возможности PL/SQL Developer в пределах системных привилегий и объектных привилегий, предоставленных пользователю Oracle, который подключён к базе данных. Например, если пользователь Oracle не имеет системной привилегии *create user*, пользователь PL/SQL Developer может запустить в PL/SQL Developer функцию *Создать пользователя*, но в итоге получит сообщение об ошибке от Oracle “ORA-01031, insufficient privileges”.

Вы можете явно авторизовать все необходимые функции PL/SQL Developer для определённых пользователей и ролей Oracle. В базе данных разработок вы позволите всем пользователям все функциональные возможности, тогда как в промышленной базе данных, вы, вероятно, отключите почти у всех пользователей все функции, которые могут изменить базу данных или займут слишком ресурсов и повлияют на производительность.

Предоставляя привилегии PL/SQL Developer ролям, вы можете настроить авторизацию для определённых групп людей. Вы можете использовать существующие роли, которые неявно соответствуют группам пользователей (такие как DBA и RESOURCE) или вы можете создать роли специально для групп пользователей PL/SQL Developer.

25.1 Включение авторизации

Информация об авторизации хранится в таблице *plsqldev_authorization* в схеме *sys*. Пока эта таблица в базе данных не существует, все пользователи авторизованы для использования всех функциональных возможностей. Если вы подсоединены к базе данных, где авторизация ещё не включена, и выбираете в меню *Файл* пункт *Авторизация*, вы получите следующий вопрос:

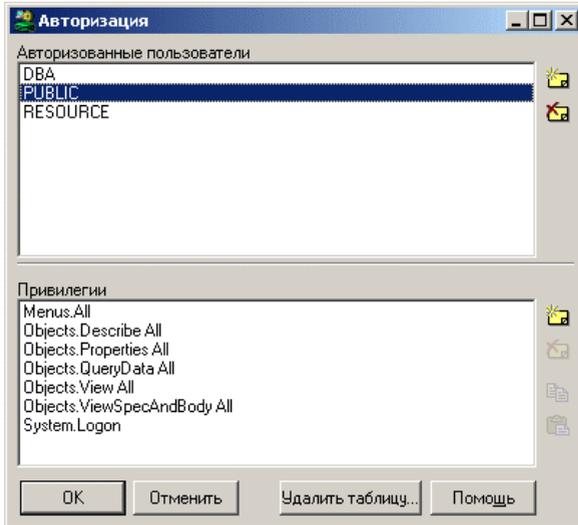


Если вы выберете *Да*, то будет создана пустая таблица авторизации. Заметьте, вы должны быть подсоединены как DBA, чтобы создать эту таблицу. Как только вы создадите одну или более привилегий в этой таблице, авторизация будет включена, и только привилегированные пользователи смогут использовать PL/SQL Developer а этой базе данных.

По умолчанию, только владелец таблицы (*sys*) или другой DBA может изменять таблицу или обновлять её содержимое. Все пользователи имеют привилегию выбора данных из этой таблицы.

25.2 Задание авторизации

Чтобы определить авторизацию, вы должны быть подсоединены как DBA и должны запустить в меню *Файл* функцию *Авторизация*. Появится следующий диалог:



В верхней части окна вы увидите всех пользователей и роли, для которых предоставлены одна или несколько привилегий. В нижней части – вы увидите конкретные привилегии, предоставленные для текущего выбранного пользователя или роли.

Чтобы добавить пользователя или роль, обладающих привилегиями, нажмите справа от списка пользователей или ролей, обладающих привилегиями, кнопку *Добавить пользователя*. При этом появится список потенциальных пользователей и ролей. Вы можете выбрать одного или несколько (во время выбора удерживайте нажатой либо клавишу Ctrl, либо – Shift) пользователей или ролей, и чтобы добавить их, нажмите кнопку OK. В данный момент, эти новые пользователи или роли не будут иметь ни каких привилегий.

Чтобы удалить пользователя или роль и их привилегии, нажмите кнопку *Удалить пользователя*. Это не повлияет на текущего пользователя или роль в базе данных, но просто удалит её привилегии из таблицы авторизации.

Чтобы предоставить привилегии, выберите пользователя или роль из списка и нажмите справа от этого списка кнопку *Добавить привилегию*. Будет отображён список всех привилегий PL/SQL Developer, где вы сможете выбрать одну или несколько привилегий. Существует 3 категории привилегий:

- **Системные привилегии** – системные функции PL/SQL Developer (например, Соединиться).
- **Объектные привилегии** – Специфические привилегии объекта базы данных (например, Drop Table).
- **Привилегии меню** – Привилегии на использование функций меню PL/SQL Developer (например, Инструменты > Сессии).

Примечание: Если пользователь не имеет привилегий *System.Logon* или *All*, он или она не могут использовать PL/SQL Developer на текущем экземпляре базы данных.

Вы можете выбрать отдельные привилегии или выбрать из иерархического набора. Например:

- **All** – Предоставить все привилегии PL/SQL Developer.
- **Objects.All** – Предоставить все привилегии PL/SQL Developer, относящиеся к объектам базы данных.
- **Objects.Rename All** – Предоставить все привилегии PL/SQL Developer, относящиеся к переименованию объектов базы данных (переименовать таблицу, вид, последовательность и синоним).
- **Objects.Rename Table** – Предоставить привилегию PL/SQL Developer на переименование таблицы.

Заметьте, что для объектов базы данных, пользователь или роль Oracle всё ещё должны иметь необходимую системную привилегию на проведение операции. Предоставление *Objects.Drop User* всё ещё требует системной привилегии *drop user*.

Также заметьте, что если пользователь имеет системную привилегию *Drop User*, он всё ещё может выполнять команду *drop user* из скрипта SQL или PL/SQL.

25.3 Отключение авторизации

Чтобы отключить авторизацию, вы можете удалить таблицу *sys.plsqldev_authorization* или удалить все записи из нее. Чтобы временно отключить авторизацию, вы можете либо удалить привилегии на таблицу, либо переименовать её (сделать её невидимой для пользователей PL/SQL Developer). Чтобы позже снова включить авторизацию, вы можете снова предоставить привилегии выбора данных.

26. Файловая система Oracle (OFS)

OFS позволяет вас хранить все ваши файлы (исходные тексты, скрипты, отчеты и т.д.) в базе данных Oracle. Это имеет то преимущество, что как объекты базы данных, так и файлы, работающие с этими объектами, могут быть сохранены в одной и той же базе данных, и доступны для всех пользователей, имеющих права доступа к этой базе данных. Более того, это гарантирует согласованность объектов базы данных и файлов, и резервное копирование и восстановление не нарушает согласованность.

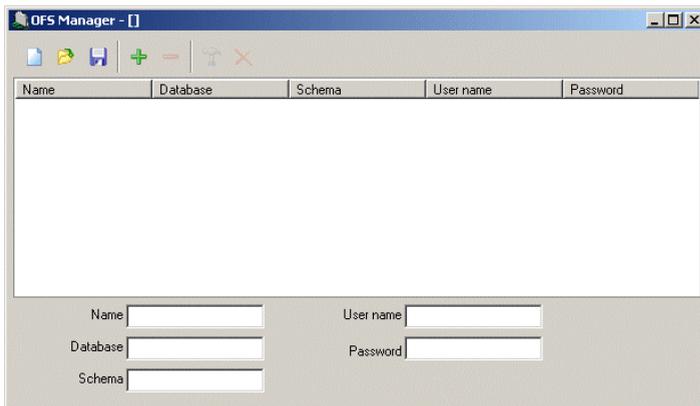
Файлы могут быть сохранены и открыты обычным способом, используя стандартный диалог открытия/сохранения файлов.

26.1 Менеджер OFS

Перед тем, как кто-либо сможет использовать OFS из PL/SQL Developer, вам необходимо воспользоваться Менеджером OFS, что бы сформировать Каталог OFS расположений с одним или более OFS расположениями. Для каждого расположения вам необходимо установить объекты базы данных OFS.

Создание Каталога OFS расположений

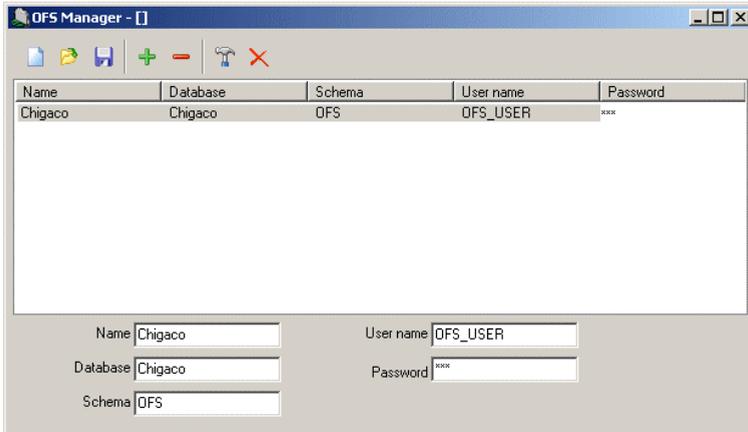
Что бы запустить Менеджер OFS, перейдите к *Предпочтениям* PL/SQL Developer и выберите страницу *Каталоги*. Внизу этой страницы вы можете найти кнопку запуска Менеджера OFS. После его запуска вы увидите следующий экран:



Это пустой Каталог OFS расположений, в который вы можете добавить расположения, и который вы можете сохранить в файле Каталога OFS расположений (.ldf). Этот файл используется PL/SQL Developer что бы просмотреть каталог, присоединиться к расположению и просмотреть файлы.

Добавление OFS расположения

Что бы добавить OFS расположение, нажмите на кнопку *Добавить расположение (+)*. Теперь вы можете ввести следующую информацию:



Name	Database	Schema	User name	Password
Chigaco	Chigaco	OFS	OFS_USER	****

Name: User name:
 Database: Password:
 Schema:

- **Название** – Будет отображаться в списке расположений диалога выбора файлов.
- **База данных** – База данных, где будут храниться OFS файлы.
- **Схема** – Схема, которая содержит объекты базы данных OFS (таблицы, пакеты и т.д.). Этому пользователю необходимы привилегии для создания своих объектов: роль *resource* и системная привилегия *query rewrite*,
- **Имя пользователя** – Имя Oracle пользователя, который будет использован для доступа к OFS. Если вы не укажете имя пользователя OFS, тогда пользователю, использующему OFS, будет необходимо указать имя Oracle пользователя и пароль, или доступ к OFS будет осуществлен под текущим Oracle пользователем PL/SQL Developer.
- **Пароль** – Пароль Oracle пользователя, который будет использован для доступа к OFS. Применим только, если поле *Имя пользователя* заполнено.

После создания одного или более расположений, вы можете сохранить Каталог OFS расположений, нажав на кнопку *Сохранить*. Если вы сохраните файл в каталоге PL/SQL Developer под именем *OFS.ldf*, тогда PL/SQL Developer, будет автоматически выбирать этот каталог. Иначе пользователю PL/SQL Developer будет необходимо указать в предпочтениях корректный файл Каталога OFS расположений.

Установка объектов базы данных OFS

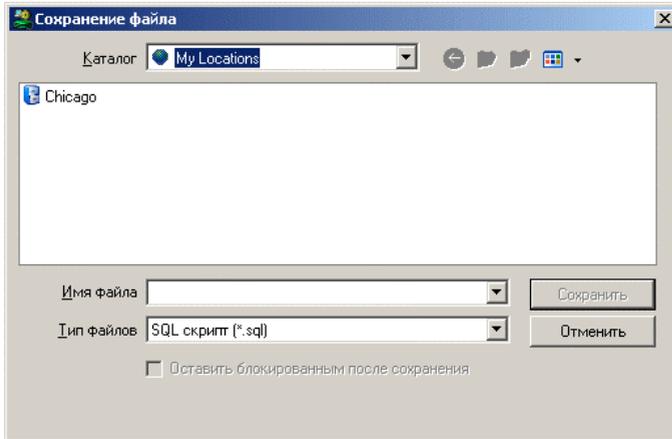
После добавления к каталогу расположения, вам необходимо установить его объекты базы данных. Если OFS схема и (необязательно) OFS пользователь пока еще не существуют в базе данных, вам необходимо сначала создать их:

```
create user OFS identified by <password>;
grant connect, resource, query rewrite to OFS;
create user OFS_USER identified by <password>;
grant connect to OFS_USER;
```

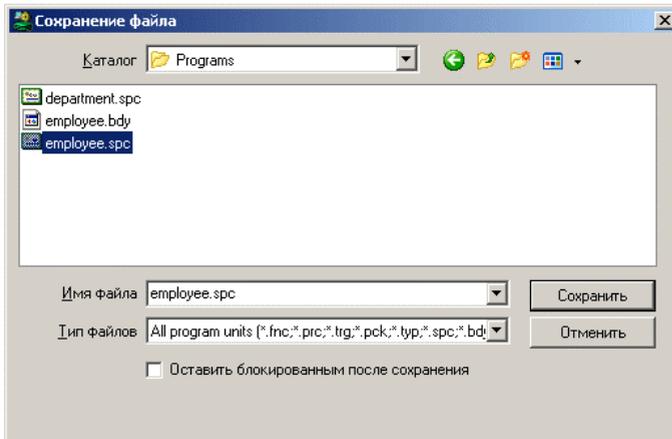
Далее вам необходимо установить объекты базы данных OFS, нажав на кнопку *Установить объекты базы*. После ввода пароля пользователя схемы, объекты базы данных будут созданы, и размещение готово для использования.

26.2 Использование OFS

После создания Каталога OFS размещений, установки размещений и обеспечения доступности файла каталога для PL/SQL Developer, вы можете использовать пункт *OFS сохранить как* в меню *Файл*, что бы сохранить файл в OFS. Появится следующий диалог:



Вам сначала необходимо открыть расположение, после чего вы можете сохранить файл или создать каталог:



Что бы открыть файл в OFS, вы можете использовать пункт *OFS открыть* в меню *Файл*, или использовать меню *Открыть повторно*. OFS файлы имеют путь OFS:\<Размещение>\<Путь> (например OFS:\Chicago\Programs\Employee.bdy). Открытый OFS файл может быть сохранен стандартной функцией *Сохранить*. Вы можете использовать функцию *Сохранить как*, что бы сохранить OFS файл в стандартной файловой системе. Функция *OFS сохранить как* может быть использована, что бы сохранить файл в OFS под другим именем или в другом каталоге.

Когда вы сохраняете или открываете файл, вы можете выбрать опцию *Блокировать*, что бы заблокировать файл. Другие пользователи не смогут блокировать или переписать файл заблокированный вами.

Щелкнув правой кнопкой мыши на файле в диалоге выбора файлов, вы сможете выбрать из всплывающего меню опции *Вырезать*, *Копировать*, *Вставить*, *Удалить* и *Переименовать* файл. Вы так же можете их этого меню просмотреть и изменить свойства объекта. Вы можете изменить следующие свойства – статус *Только-для-чтения*, статус *Блокированно* и статус *Сжато*. Файлы могут быть сжаты только в Oracle10g и позднее, так как для сжатия используется пакет *utl_compress*, который доступен только начиная с 10g.

27. Справочные системы

Вы можете интегрировать PL/SQL Developer со справочными файлами и руководствами, которые поставляются с программным обеспечением Oracle Server. В этой главе описывается, как настроить и использовать эти справочные системы.

27.1 MS Help файлы

До Oracle 7.2 все сетевые справочники поставлялись в формате MS Help. Эти справочные файлы могут быть использованы для предоставления в редакторе чувствительной к контексту помощи, либо щелчком правой кнопкой мыши на слове, либо выбором в меню *Help* пунктов *SQL help* или *PL/SQL help*, либо нажатием комбинации клавиш, которую вы установили для этих двух пунктов.

PL/SQL Developer ищет справочные файлы в следующих двух каталогах:

1. В каталоге, где находится PL/SQL Developer.
2. В каталоге, где находится SQL*Plus.

Предполагается, что имя справочного файла SQL будет *sqlhelp.hlp*, а справочного файла PL/SQL – *plshelp.hlp*. Если эти справочные файлы не доступны, но у вас есть какие-либо другие справочные файлы, которые вы хотели бы использовать для справки по SQL или PL/SQL, вы можете поместить эти файлы в каталог PL/SQL Developer и переименовать их, как это упомянуто выше.

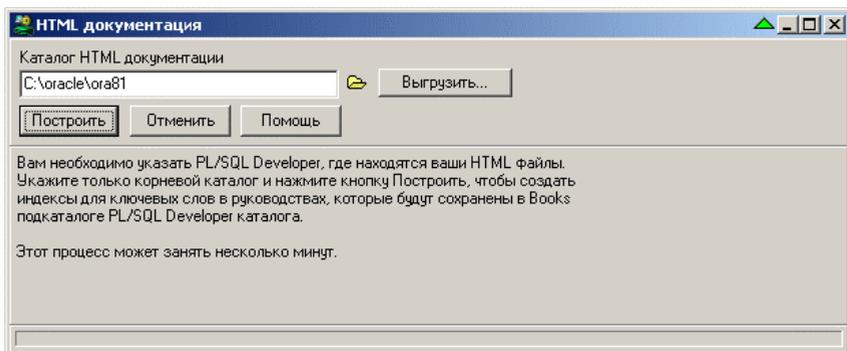
27.2 HTML руководства

Начиная с Oracle 7.3, оперативные руководства поставляются в форматах Acrobat Reader и HTML. Вы можете настроить PL/SQL Developer, чтобы он использовал HTML руководства при нажатии правой кнопкой мыши на слове в редакторе, при выборе в меню *Справка* пункта *HTML документация* или при нажатии *F1*. Более того, в случае исключительной ситуации Oracle, вы сможете нажимать на кнопку *Помощь* в сообщении об ошибке. Эта кнопка отобразит соответствующий параграф в руководстве *Oracle Error Messages*. При двойном щелчке на ошибке компиляции в программном окне, также будет отображена эта информация.

Примечание: сетевая документация некоторых версий Oracle Server содержит файлы HTML размером более 1 Мегабайта. Загрузка этих файлов может занять очень много времени. В этом случае, предлагается использовать сетевую документацию той версии Oracle Server, где главы разделены на меньшие по размеру файлы. Oracle имеет файлы меньших размеров в версии 8.1.5.

Настройка HTML документации

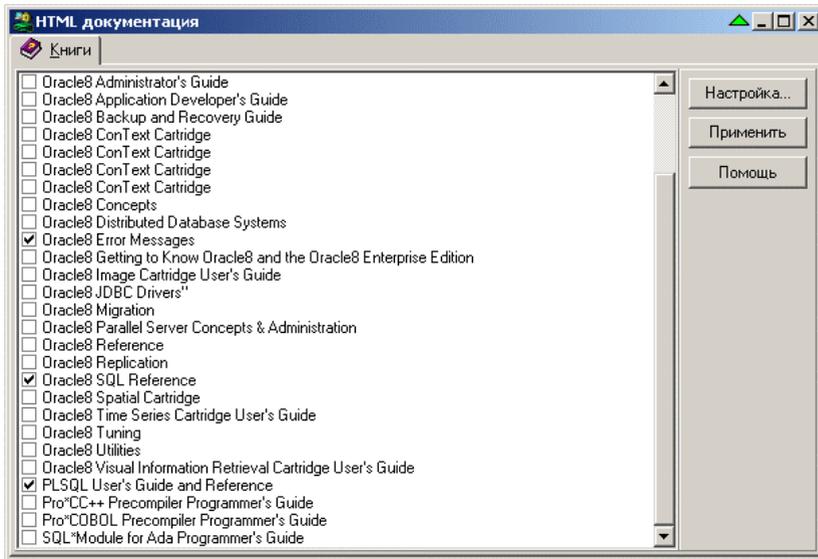
Первый раз, когда вы вызываете руководства HTML одним из способов, описанных выше, вам нужно указать PL/SQL Developer, где находятся ваши HTML файлы:



Просто укажите корневой каталог и нажмите кнопку *Построить*, чтобы создать индексы для ключевых слов в этих руководствах, которые будут храниться в подкаталоге *Books* каталога PL/SQL Developer. Этот процесс может занять несколько минут.

Если у вас нет доступной HTML документации Oracle, вы можете нажать кнопку *Выгрузить*. При этом вы будете перемещены на локальную html страницу, где вы сможете выбрать подходящую версию Oracle Server и скачать zip файл с HTML документацией. Это скачивание требует наличия регистрации в Oracle Technology Network. Если вы не зарегистрированы в OTN, то сначала вы должны бесплатно зарегистрироваться.

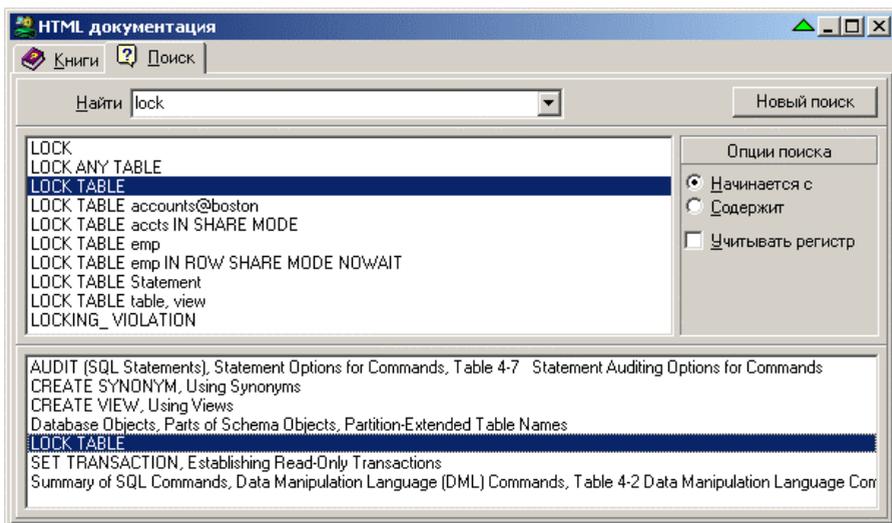
После того, как составление этого индекса закончено, вам будет предоставлен список найденных руководств. Теперь вы можете выбрать книги, которые вы хотите использовать для помощи чувствительной к контексту. По умолчанию в этот выбор будут входить руководства *Error Messages*, *SQL Reference* и *PL/SQL Reference*, которые все содержат уместную в течение разработки PL/SQL информацию. Руководство *Error Messages* требуется, если в случае исключительной ситуации или ошибки компиляции Oracle, вы хотите иметь автоматически доступными *причину* и *действие*. Текущее руководство Oracle7 некорректно индексируется для использования этой функциональной возможности, а версия Oracle8 работает без проблем.



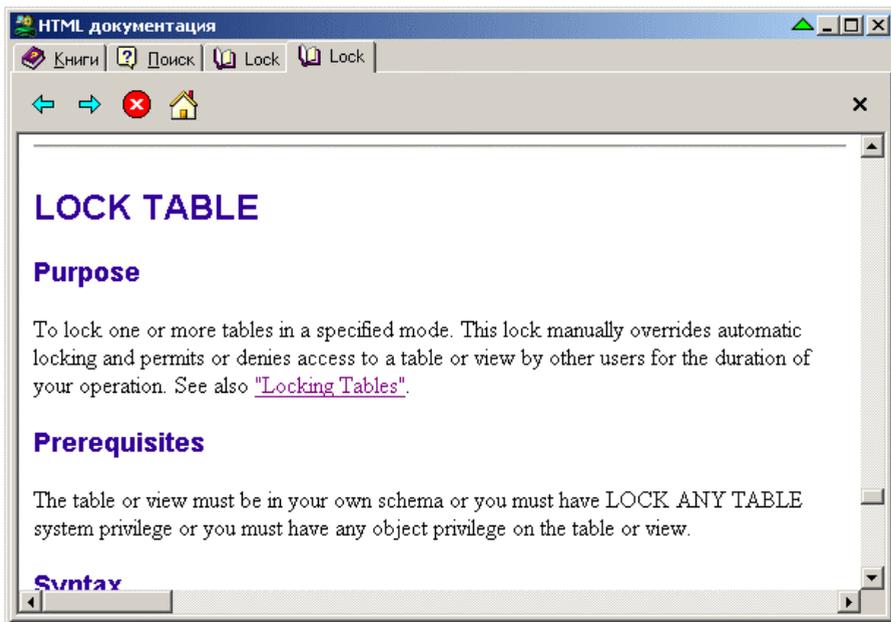
Чтобы выбор стал эффективным, нажмите кнопку *Применить*. После этого страницу поиска, которая обсуждалась в предыдущей главе, будет активной. Нажав кнопку *Настроить*, вы можете просканировать различные каталоги на наличие HTML руководств.

Использование HTML руководств

После вызова HTML руководств, вы можете ввести ключевое слово, которое вы хотите найти на странице *Поиск*. Два списка показывают ключевые слова, которые удовлетворяют условиям поиска, и разделы в руководствах, где встречаются эти ключевые слова:



Используя панель *Опции поиска*, справа от списка ключевых слов, вы можете уточнить критерии поиска. Далее, вы можете дважды щёлкнуть на элементе в списке разделов в нижней части страницы поиска, чтобы отобразить соответствующий параграф в HTML руководстве:



Страница результатов поиска отображает страницу HTML, по которой, как обычно, вы можете перемещаться при помощи гиперссылок. Чтобы перемещаться в предыдущее или следующее местоположение, которое вы посетили ранее, чтобы отменить загрузку страницы или чтобы перейти на первоначальную страницу, вы можете использовать кнопки, расположенные в верхней части диалога.

Вы можете вернуться обратно на страницу *Поиск* для поиска других ключевых слов или перейти на страницу *Книги*, чтобы выбрать другие книги. При каждом поиске создаётся новая страница результатов, так что вы можете иметь доступными несколько результатов поиска. Чтобы закрыть страницу результатов, нажмите справа сверху кнопку *Закрыть страницу*. Чтобы закрыть все страницы, нажмите на странице поиска кнопку *Новый поиск*.

По умолчанию, окно HTML в PL/SQL Developer IDE будет оставаться поверх других окон, так что информация остаётся видимой. Чтобы временно скрыть окно, вы можете использовать зелёную кнопку в правой верхней части окна, чтобы свернуть окно.

28. Настройка

Некоторые аспекты PL/SQL Developer могут быть настроены, чтобы отвечать вашим персональным требованиям. Эти аспекты описаны в следующих параграфах.

28.1 Предпочтения

В меню *Инструменты* находится пункт *Предпочтения*, пункт позволит вам установить для PL/SQL Developer различные предпочтения. Эти предпочтения подробно описаны в главе 16.

28.2 Макет окон

Существуют различные настройки макетов окон, которые могут быть адаптированы под ваши требования. При выборе в меню *Окно* пункта *Созрнить макет* эти настройки будут сохранены. Настройки макета включают в себя:

- Размер и местоположение окна приложения.
- Наличие и размер Объектного навигатора.
- Наличие и размер Разделов кода в Программном окне.
- Наличие, статус (плавающее или закреплённое) и местоположение Списка окон и Окна шаблона.
- Размер Программного окна, Тестового окна, Окна SQL, Окна Explain Plan, Командного окна, Окна отчёта, Окна компиляции инвалидных объектов, Окна поиска объектов базы данных, Окна экспорта пользовательских объектов и Окна сравнения пользовательских объектов.

Когда вы сохраняете макет, будет сохранён размер последнего открытого окна. Если окно некоторого типа отсутствует, настройки размеров этого окна будут оставлены без изменения.

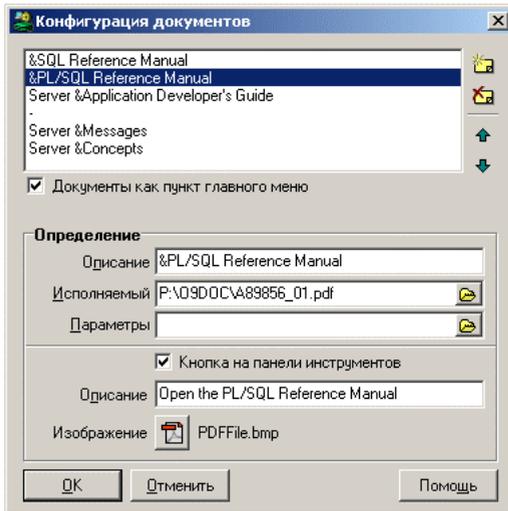
28.3 Оперативная документация

Большинство документации Oracle поставляется в виде оперативной документации. Со временем, у этих документов стало несколько форматов. На данный момент существуют MS Help файлы, Oracle Book файлы, Adobe PDF файлы и HTML файлы. В будущем, Oracle может выбрать для использования другой формат.

Кроме того, у вас могут быть какие-либо корпоративные стандарты и проектные документы в формате Word, которые время от времени будут нужны вам.

Все эти оперативные документы могут быть интегрированы в PL/SQL Developer's IDE, путём включения их в меню *Документы*. Таким образом, они могут быть доступны вам при щелчке по ним мышкой.

Чтобы настроить оперативные документы, выберите в меню *Инструменты* пункт *Конфигурировать документы*. Для определения описания каждого документа и способа его просмотра, появится следующий диалог:



Справа вы видите четыре кнопки для создания нового документа, удаления документа и перемещения документа по меню вверх или вниз. Когда вы создаёте или изменяете документ, вы должны указать следующую информацию:

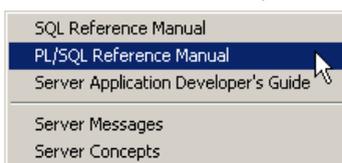
Описание отображается в меню. Чтобы создать горячую клавишу для пункта меню документа, поставьте перед буквой в описании значок &. Если вы хотите использовать этот значок в описании, то вместо него пользуйтесь двойным значком &. Если вы введёте только '-', это станет разделителем в меню. Таким образом, вы можете разделить документы на логические группы.

В поле Исполняемый должен быть указан полный путь программы, которую вы хотите использовать для просмотра документа. Если документ является зарегистрированным типом файла (таким как MS Help файл или HTML файл), вы можете ввести здесь путь для документа.

Поле Параметры может быть использовано для передачи информации исполняемому модулю, если он является программой. Это поле, по меньшей мере, должно включать в себя путь к документу, но также может быть использовано для передачи опций программе.

Когда включена опция *Документы как пункт главного меню*, меню *Документы* находится в главном меню. Если она не включена, оно будет находиться в меню *Инструменты*.

В приведённом выше примере руководство PL/SQL Reference Manual является файлом PDF (P:\O9DOC\A89856_01.pdf), который является зарегистрированным типом файла, так что вы можете ввести только имя файла в поле Исполняемый. Буква 'P' используется как горячая клавиша в меню. Меню *Документы* будет выглядеть следующим образом:



Когда включена опция *Кнопка на панели инструментов*, документ может быть дополнительно включён в панель инструментов. Когда вы поместите курсор мыши над кнопкой на панели инструментов, в качестве подсказки появится соответствующее *Описание*. Если вы оставите это описание пустым, то будет использовано основное описание документа (как показано в меню). Чтобы выбрать файл рисунка Windows (*.bmp), нажмите кнопку *Изображение*. Желательно, чтобы размер рисунка был 20 x 20 пикселей. Заметьте, что PL/SQL Developer всегда загружает файл рисунка из первоначально места размещения, так что вам не следует перемещать или переименовывать этот файл без изменения соответствующего документа. PL/SQL Developer поставляется с набором стандартных файлов рисунков, из которых вы можете выбирать. Эти файлы находятся в установочном каталоге PL/SQL Developer в подкаталоге Icons. Этот каталог является каталогом по умолчанию диалога выбора рисунка.

28.4 Параметры командной строки

В командной строке PL/SQL Developer могут быть использованы следующие параметры. Ярлык, установленный по умолчанию, не включает в себя никаких параметров, а только запускает из установочного каталога PL/SQL Developer файл `plsqdev.exe`.

Набор параметров, установленных по умолчанию, может быть определён при помощи файла `params.ini` из установочного каталога PL/SQL Developer. Вы можете редактировать файл в текстовом редакторе, например, в Notepad. Файл содержит объяснение каждого параметра. Если параметр определён в `params.ini` и в командной строке, то командная строка имеет приоритет.

userid

Каждый раз, когда вы запускаете PL/SQL Developer, он будет запрашивать у вас пароль. Чтобы избежать этого, вы можете указать параметр `userid` со знакомым вам форматом `username/password@database`:

```
plsqdev.exe userid=scott/tiger@chicago
```

Заметьте, что вы также можете использовать следующий ключ системного реестра для регистрации по умолчанию:

```
HKEY_CURRENT_USER\Software\Allround Automations\PL/SQL Developer\Logon
```

Здесь вы можете добавить Username, Password и Database.

Преимущество последнего метода состоит в том, он позволяет вам автоматически регистрироваться после двойного щелчка на зарегистрированном файле PL/SQL Developer.

nologon

Параметр `nologon` подавляет диалог регистрация, который появляется при запуске PL/SQL Developer. Этот параметр не требует никаких дополнительных аргументов:

```
plsqdev.exe nologon
```

oraclehome

Вы можете указать в командной строке имя Oracle Home, таким образом, переопределив главный Oracle Home (который используется по умолчанию) и предпочтение Oracle Home:

```
plsqdev.exe oraclehome=ora817
```

Заметьте, что это имя Oracle Home, как указано в ключе ORACLE_HOME_NAME в системном реестре. Это имя также будет использоваться и при выборе Oracle Home. Оно не является именем секции системного реестра или именем каталога.

dontadjustpath

PL/SQL Developer не будет временно изменять PATH для инициализации Oracle Net:

```
plsqldev.exe dontadjustpath
```

nosplash

Параметр nosplash подавляет отображение заставки во время запуска PL/SQL Developer. Он не требует каких-либо дополнительных параметров:

```
plsqldev.exe nosplash
```

noplugins

Указание параметра noplugins подавляет загрузку каких-либо плагинов:

```
plsqldev.exe noplugins
```

library

Указывает местоположение библиотеки стилей для отчётов:

```
plsqldev.exe library=p:\standard.lib
```

prefpath

Указывает путь для персональных наборов предпочтений. Например:

```
plsqldev.exe prefpfpath=u:\userdata
```

Для более подробной информации смотрите главу 16.33.

commandfile

Запускает указанный командный файл в Командном окне. Например:

```
plsqldev.exe userid=scott/tiger commandfile="u:\sql scripts\demo_build.sql"
```

Эта команда сначала подключится к базе данных и потом запустит командный файл *demo_build.sql*.

viewobject

Отображает указанный объект базы данных. Параметр требует, чтобы вы также указали *userid* параметр. Например:

```
plsqldev.exe userid=scott/tiger@chicago viewobject=emp
```

Эта команда сначала подключится к базе данных, а потом покажет определение таблицы *EMP*. Заметьте, что спецификация объекта может также включать в себя владельца (например, *SCOTT.EMP*).

editobject

Редактирует указанный объект базы данных. Параметр требует, чтобы вы также указали *userid* параметр. Например:

```
plsqldev.exe userid=scott/tiger@chicago editobject=emp
```

Эта команда сначала подключится к базе данных, а потом покажет определение таблицы *EMP*. Заметьте, что спецификация объекта может также включать в себя владельца (например, *SCOTT.EMP*).

registry

Что бы использовать различные наборы IDE установок, вы можете использовать параметр *registry*. Например:

```
plsqldev.exe registry=home
```

Эта команда обяжет IDE загружать и сохранять установки (размеры и позиции окон, историю файлов, историю соединений и т.д.) под именем *home*.

28.5 Ключевые слова SQL, PL/SQL, Command, Java и XML

Ключевые слова, которые подсвечиваются в редакторах SQL, PL/SQL, а также Редакторе команд, могут быть определены в файлах ключевых слов. В каталоге, куда установлен PL/SQL Developer, вы сможете найти текстовые файлы *sql.kwf*, *plsql.kwf*, *command.kwf*, *java.kwf* и *xml.kwf*. Вы можете изменять, добавлять или перемещать ключевые слова в подходящие секции, используя текстовый редактор, например Notepad.

28.6 Модули расширения

Вы можете создавать свои собственные модули расширения, чтобы расширить функциональные возможности PL/SQL Developer. Функции модуля расширения могут быть добавлены в меню PL/SQL Developer и могут выполнять различные действия: обращаться к базам данных, Объектному навигатору, текущему окну и редактору и так далее.

Модули расширения являются DLL, которые предоставляют для PL/SQL Developer специфический интерфейс. Чтобы создать модуль расширения, вы можете использовать любой язык программирования, который позволяет создавать DLL. Когда модуль расширения помещен в каталог PL/SQL Developer'a Plug-In, он будет автоматически загружен из него. Это делает распространение вашего модуля расширения очень лёгким.

Вы можете создать модули расширения для своего собственного или для общего использования. Некоторые стандартные модули расширения доступны на веб-сайте Allround Automations (<http://www.allroundautomations.com/plsqldev.html>). Здесь также доступны модули расширения, разработанные другими пользователями.

Документация по модулям расширения и их примеры доступны в подкаталоге *PluginDoc* корневого каталога PL/SQL Developer.