

Г Л А В А 2

Обзор SQL и SQL*Plus

В этой главе...

- Классификация и распознавание типов SQL-выражений
- Редактирование и запись SQL-выражений
- Использование трех различных интерфейсов SQL, доступных в Oracle
- Написание SQL-выражений для создания таблиц и ввода данных в таблицы
- Применение SQL-выражений для удаления строк, добавления столбцов и удаления таблиц
- Использование расширений SQL*Plus для захвата выхода и форматирования результатов

Введение

Язык структурированных запросов (Structured Query Language — SQL) представляет собой промышленный стандартный язык, используемый для взаимодействия с реляционными базами данных. Впервые SQL был описан в работе доктора Е.Ф. Кодда (E.F. Codd), разработавшего в 1970-х System R и SQL для IBM. На данном стандарте основаны все инструменты доступа к Oracle. Промышленные стандарты для SQL устанавливаются двумя сообществами — ANSI (American National Standards Institute — Национальный институт стандартизации США) и ISO (International Standards Organization — Международная организация по стандартизации). ANSI стандартизовал SQL в 1986 году. В 1989 году ANSI и ISO опубликовали стандарт SQL89 (или SQL1). Позже ANSI пересмотрел первый стандарт и продублировал его как SQL92

(SQL2). Последний стандарт ANSI, SQL99 (SQL3), содержит указания по концепциям объектно-ориентированных баз данных. Следует отметить высокую переносимость SQL — данный язык можно использовать для доступа к большому числу систем баз данных, включая Oracle, DB2 (IBM), SQL Server (Microsoft) и MySQL.

SQL — это не язык программирования. Он предоставляет выражения, используемые для ввода, извлечения, модификации, удаления и отображения данных, хранимых в таблицах базы данных. Используя SQL, вы можете формулировать запросы для немедленного выполнения, создавать различные объекты и выполнять операции по ведению базы данных. За прошедшие годы развилось несколько версий стандарта. Последняя из них, SQL99, полностью распознается системой управления базой данных Oracle. Преимущество стандарта заключается в том, что любые выученные вами команды SQL и навыки, которыми вы овладели при использовании Oracle, пригодятся вам и при работе с другими системами управления реляционными базами данных. Небольшие отличия в реализациях SQL от различных производителей баз данных возможны, но, как правило, они являются крайне незначительными (например, использование для выделения имен не двойных кавычек, а квадратных скобок). Другими словами, изучив сейчас SQL, вы сможете работать с интерфейсами практически всех других систем управления базами данных.

Несмотря на то что в Oracle реализован ANSI SQL, здесь также предлагается богатый набор расширений к SQL, называемых командами SQL*Plus. Хотя расширения SQL*Plus нельзя использовать со всеми системами баз данных на основе SQL, они достаточно полезны при работе с Oracle. Расширения SQL*Plus предлагают множество элементов, облегчающих разработку, в частности, форматирование данных, возвращаемых в ответ на запрос, облеченный в стандартную форму SQL. Еще раз повторим, что в данной книге и в документации, предлагаемой Oracle, эти полезные расширения называются командами SQL*Plus.

Типы команд SQL

Существует пять типов выражений SQL: выражения запроса, DDL-выражения (Data Definition Language — язык определения данных), DML-выражения (Data Manipulation Language — язык манипулирования данными), TC-выражения (Transaction Control — управление транзакциями) и DCL-выражения (Data Control Language — язык управления данными).

Выражения запроса

Выражения запроса позволяют извлекать строки из одной или нескольких таблиц. В качестве примера выражения запроса можно привести `SELECT` (мы придерживаемся принятого в SQL принципа зарезервированных слов — слова, определяющие элементы языка SQL и не используемые в качестве идентификаторов, пишутся прописными буквами).

DDL-выражения

Выражения SQL, принадлежащие к данной группе, позволяют определять структуры (таблицы и представления), образующие базу данных. В данной главе мы будем использовать пять таких выражений: CREATE, ALTER, DROP, RENAME и TRUNCATE. Данные команды позволяют соответственно создавать и определять таблицы или пользователей; модифицировать структуру базы данных; удалять структуру (например, таблицу); изменять ее имя; а также удалять все позиции таблицы.

DML-выражения

Как следует из названия, выражения манипулирования данными позволяют модифицировать содержимое таблиц путем ввода новых данных, изменения существующих данных или удаления из таблицы целых строк данных. DML-выражения INSERT, UPDATE и DELETE мы разберем в следующих главах.

ТС-выражения

Выражения управления транзакциями предлагают инструменты, позволяющие сделать постоянными любые изменения строк таблицы или отменить эти изменения. Команда COMMIT делает постоянными любые изменения, внесенные в строки таблицы. Команда ROLLBACK является обратной — она аннулирует все изменения строк таблицы, сделанные после последнего сохранения. Еще в данную группу входит выражение SAVEPOINT, позволяющее отмечать точку, к которой вы можете вернуться, отменив все внесенные изменения.

DCL-выражения

Выражения языка управления данными определяют, кому разрешен доступ к различным таблицам, кто может входить в систему Oracle, а также какие права доступа и привилегии имеют пользователи различных таблиц базы данных. Для определения прав доступа и привилегий пользователей используется выражение GRANT, а для отмены выбранных прав и привилегий — команда REVOKE. Например, вы можете задать, какой пользователь может извлекать информацию из данных другого пользователя или запретить пользователю создавать таблицы.

“Анатомия” выражения SQL

Выражения SQL применяются для выполнения задач, связанных с базой данных. После командного слова или фразы (например, CREATE) вы включаете фразу, уточняющую действие, которое следует выполнить. Выражения SQL, введенные посредством интерфейса SQL*Plus, могут занимать одну или несколько строк. Для продолжения выражений SQL на следующей строке необходимо нажать клавишу <Enter>. В ре-

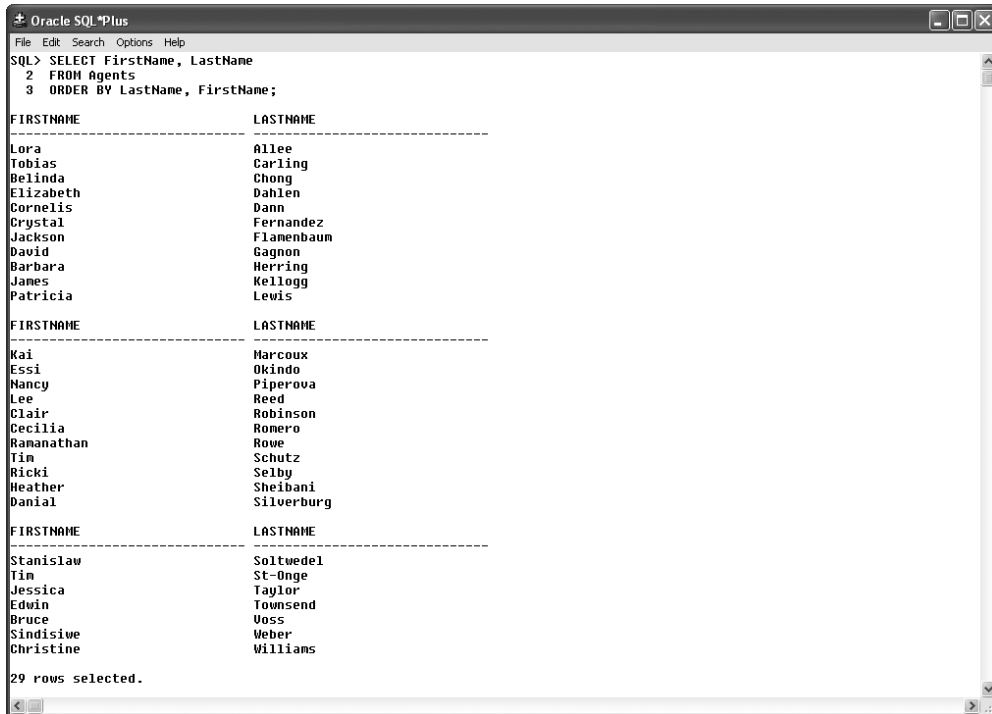


Рис. 2.1. Пример команды SQL, введенной с помощью SQL*Plus for Windows

зультате точка ввода переносится на начало новой строки, и при этом не требуется использовать метку продолжения строки. В любом случае разрыв слова на две строки не допускается. Если вы пользуетесь построчным интерфейсом, подобным тому, что предлагает SQL*Plus, то последовательные строки будут пронумерованы. Выражение SQL не обязательно полностью писать в одну строку, довольно часто выражения SQL в данной книге представляются в несколько строк, каждая из которых содержит отдельную фразу SQL. На рис. 2.1 показан пример выражения SQL SELECT, введенного посредством интерфейса SQL*Plus.

Сокращенная запись слов, входящих в выражение SQL, недопустима (хотя некоторые слова SQL*Plus имеют краткие формы; подробнее об этом — ниже). Слова должны разделяться по крайней мере одним пустым символом (<пробел>, <Tab> или <Enter>). Слова SQL можно набирать прописными или строчными буквами. Ключевые слова SQL (слова, по определению имеющие специальное значение в языке SQL) в данной книге для удобства представлены прописными буквами. Самыми распространенными знаками пунктуации в командах SQL являются запятая (разделение списка имен, например, названий столбцов) и круглые скобки (обособление элементов). Точки применяются в именах объектов для разделения отдельных составляющих, например Inventory.Price. Одинарные кавычки (апострофы) замыкают последовательности литеральных символов (например, даты). В двойные кавычки

берутся строки, используемые, например, как альтернативные имена столбцов. Точка с запятой указывает конец выражения SQL. Ввод точки с запятой и нажатие клавиши <Enter> в SQL*Plus приводит к выполнению команды SQL. Существуют и другие знаки пунктуации, которые мы подробно рассмотрим ниже.

Интерактивная помощь

Один из самых полезных навыков при изучении Oracle связан с получением помощи. Например, вы ввели команду SQL с ошибкой и интерпретатор SQL*Plus отобразил сообщение об ошибке, ее код, номер строки, содержащей ошибку, и отметил положение ошибки в пределах этой строки. На рис. 2.2 показан пример ошибки, порожденной командой CREATE TABLE. Пытаясь создать новую таблицу ContactReason, мы неправильно ввели команду, описывающую структуру таблицы. Некоторые ошибки могут быть очевидными, а другие обнаружить очень трудно. Интерпретатор SQL*Plus указывает, что ошибка находится в строке 2, отображает ее код (ORA-00922) и сообщение “missing or invalid option” (“пропущенная или неверная опция”). Большинство сообщений об ошибках Oracle обычно лаконичны. Если вы не можете сразу определить проблему, помните, что в Интернете вы можете легко найти дополнительную информацию, которая поможет понять и исправить ошибку. Приставка “ORA” в начале кода ошибки Oracle указывает, что ошибку (иногда именуемую исключительной ситуацией) сгенерировала система баз данных. Пятизначное число после ORA, присваиваемое Oracle Corporation, указывает точную природу ошибки — в данном случае, что мы пропустили запятую во второй строке. Звездочка в сообщении об ошибке часто указывает точное положение ошибки в операторе, вызвавшем сбой. Однако в примере на рис. 2.2 звездочка просто отмечает начало выражения или фразу, в которой была обнаружена ошибка. В частности, звездочка указывает, что ошибка произошла при обработке открывающей круглой скобки, за которой следует определение первого столбца таблицы ContactReason. В данном случае сообщение “missing or invalid option” недостаточно конкретно, чтобы быть особенно полезным.

Помощь по сообщениям об ошибках Oracle доступна на сайте OTN (Oracle Technology Network) и других Web-сайтах. На сайте <http://ora-code.com> представлен полный список кодов ошибок ORA по порядку. Возможно, вам придется несколько раз щелкнуть мышью, чтобы найти требуемый код на нескольких страницах перечня, однако это того стоит.

Чтобы получить больше информации о сообщениях об ошибках в стиле ORA, выполните действия, приведенные ниже.

1. Запустите Web-браузер, введите в адресной строке <http://ora-code.com> и нажмите клавишу <Enter>. Появится страница Oracle Database Error Code.
2. Щелкните на ссылке с цифрой 3, чтобы перейти на третью страницу перечня ошибок.

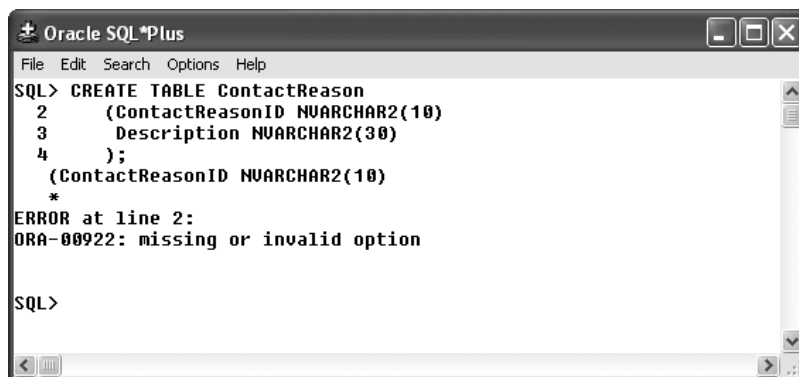


Рис. 2.2. Сообщение об ошибке в команде SQL и соответствующий код ошибки

3. Щелкните на ссылке ORA-00922 (значения увеличиваются слева направо и сверху вниз).
4. Появится полное описание ошибки. По-видимому, Oracle считает, что мы добавили неверный оператор. Внимательно изучая строку SQL, видим, что в конце строки 2 отсутствует запятая. Пока что данную ошибку мы исправлять не будем.
5. Закройте Web-браузер.

Взаимодействие с базой данных

Помимо определения языка SQL, в стандарте SQL99 приводятся указания по выполнению выражений SQL. В частности, данные указания (*связующие стили*) определяют минимальный набор выражений SQL, который должен поддерживаться для определенного связующего стиля. Всего существует четыре метода выполнения: прямое (немедленное) выполнение, связывание модулей, внедренные SQL-выражения и интерфейс уровня вызовов (Call-Level Interface — CLI).

Прямое выполнение позволяет общаться с системой управления базой данных непосредственно через клиентское приложение, подобное SQL командной строки, SQL*Plus или iSQL*Plus. Интерпретатор SQL*Plus проверяет командную строку, чтобы гарантировать, что она не содержит синтаксических ошибок. Если все правильно, то интерпретатор отправляет строку в базу данных для выполнения. Отметим, что команды SQL *нечувствительны* к регистру. Даже если мы будем писать зарезервированные слова SQL прописными буквами, интерпретатор SQL*Plus и база данных Oracle, выполняющая команды, регистр не различают — команды и любые слова могут содержать только прописные, только строчные буквы или и те и другие в произвольной комбинации. Более того, SQL*Plus игнорирует пробелы и разрывы строки в выражении. Пробелы и разрывы строк мы используем исключительно для удобочитаемости.

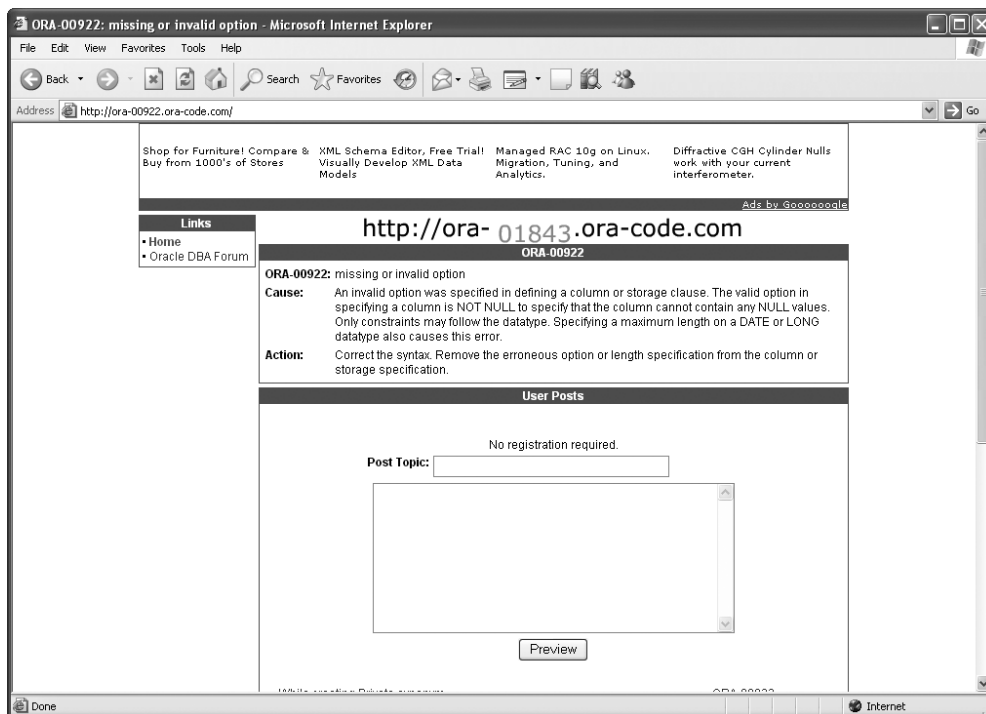


Рис. 2.3. Дополнительная информация о коде ошибки Oracle

Интерпретатор SQL взаимодействует с сервером базы данных, расположенным на том же компьютере или на другом компьютере сети. При прямом выполнении вы вводите выражение SQL непосредственно на клиентском компьютере и, нажимая соответствующую клавишу, отправляете его серверу базы данных. Вскоре сервер возвращает результаты интерфейсу клиента. Если, например, выражение SQL предполагало извлечение данных, интерфейсу клиента будут отправлены строки результата.

Метод связывания модулей позволяет создавать блоки выражений SQL. Готовый блок можно вводить в приложение с помощью программы, именуемой компоновщиком (linker). В результате получается завершенная процедура, содержащая все, что требуется для выполнения ряда операций с базой данных.

Внедренные выражения SQL вводятся непосредственно в базовый язык программирования (например, C++ или COBOL). Препроцессор Oracle изучает выражения SQL на предмет правильности и отделяет их от другого скомпилированного кода на упомянутом языке программирования. Код SQL преобразовывается в форму, которую понимает система баз данных. Внедрив выражения SQL непосредственно в базовый язык программирования, мы позволим программе обращаться к таблицам базы данных посредством этих выражений и извлекать реальные строки данных.

Интерфейс уровня вызовов позволяет вызывать выражения SQL через “дверцу в коде”, передавая выражения SQL непосредственно обрабатывающим подпрограм-

мам. Выражения не компилируются и не подвергаются другой обработке до получения их сервером базы данных (после чего они выполняются непосредственно системой управления базой данных).

Ниже описаны три различных способа организации запросов и модификации баз данных Oracle: через интерфейс командной строки из операционной системы; через клиентский продукт командной строки SQL*Plus от Oracle; и через Web-среду Oracle названную *iSQL*Plus*.

Иногда разница между SQL и SQL*Plus на первый взгляд незаметна, однако они отличаются, и вам следует знать об этом. SQL — это непроцедурный язык, используемый для связи с системой управления реляционной базой данных. SQL*Plus — это разработанная Oracle среда, в которой вы можете выполнять выражения SQL. Позже в данной главе мы расскажем о втором значении термина SQL*Plus — командах SQL*Plus. Команды SQL*Plus — это расширение SQL от Oracle, позволяющее форматировать результаты, отображать определения таблиц, редактировать и сохранять файлы. Подробнее об этом рассказывается в разделе “Использование расширений SQL*Plus”.

Ниже мы рассмотрим использование трех различных методов для входа в систему баз данных Oracle с использованием SQL*Plus на компьютере с ОС Windows. Во-первых, вы научитесь вызывать SQL*Plus из командной строки DOS (Disk Operating System — дисковая операционная система). Во-вторых, узнаете, как использовать более доброжелательное к пользователю средство SQL*Plus for Windows. И в-третьих, вы разберетесь с вызовом *iSQL*Plus*, интерфейсом базы данных Oracle, основанном на Web. Все названные методы имеют достоинства и недостатки. Прочитав данную книгу, вы сможете выбрать интерфейс, оптимальный для ваших потребностей.

Вызов SQL*Plus из командной строки DOS

Если вы запустили Oracle на собственном ПК с операционной системой Windows или используете ПК Windows для взаимодействия с Oracle на сервере, то для взаимодействия с базой данных можете использовать окно DOS или среду SQL*Plus. Поскольку мы рекомендуем использовать средства SQL*Plus, предлагающие множество знакомых по Windows инструментов (например, копирование/вставка и т.д.), возможности командной строки мы рассмотрим кратко.

Вход в Oracle

SQL*Plus можно вызвать из командной строки DOS, после этого войти в Oracle и далее вводить команды SQL. Итак, чтобы открыть окно DOS и вызвать SQL*Plus в операционной системе Windows, выполните следующие действия.


```

C:\WINDOWS\system32\command.com
C:\>sqlplus
SQL*Plus: Release 10.1.0.2.0 - Production on Sat Sep 23 12:40:26 2006
Copyright (c) 1982, 2004, Oracle. All rights reserved.
Enter user-name: redwood
Enter password:
Connected to:
Personal Oracle Database 10g Release 10.1.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options
SQL> help describe
DESCRIBE
Lists the column definitions for a table, view, or synonym,
or the specifications for a function or procedure.
DESCRIBE1 <[schema.]object[@connect_identifier]>
SQL>
    
```

Рис. 2.4. Использование SQL*Plus в окне DOS

1. Щелкните на кнопке Start (Пуск) панели задач, выберите Run (Выполнить), введите command в текстовом окне Open (Открыть) и щелкните на кнопке ОК. Откроется окно DOS.
2. Введите sqlplus и нажмите клавишу <Enter>. Появится запрос имени пользователя.
3. Введите свое имя пользователя и нажмите клавишу <Enter>. Появится запрос пароля.
4. Введите пароль (на экране он не отображается) и нажмите клавишу <Enter>. Если вы все правильно ввели, появится запрос SQL>.
5. Введите help describe и нажмите клавишу <Enter>, чтобы отобразилась справочная информация о заданной команде SQL*Plus (рис. 2.4).
6. Чтобы выйти из Oracle, введите exit и нажмите клавишу <Enter>. Вы вернетесь к запросу DOS C:\>.
7. Чтобы закрыть окно DOS, еще раз введите exit и нажмите клавишу <Enter>.

Ввод и запуск команд SQL

В предыдущем примере вы запустили команду SQL*Plus — одну из команд, определенных Oracle, для расширения стандартного набора команд SQL. Эта команда описывает структуру таблицы Dual, устанавливаемой Oracle автоматически. Перед выполнением указанных ниже действий найдите на компакт-диске файл Ch02SaleStatus.sql и скопируйте его на компьютер. Запомните полный путь к папке, в которую вы его скопировали. Данный файл не стоит записывать в папке My Documents (Мои документы),

104 Глава 2. Обзор SQL и SQL*Plus

поскольку это слишком длинный путь. Сохраните его в корне диска C (C:\) или в папке Temp (C:\Temp\).

Чтобы запустить команду SQL в SQL*Plus, откройте окно DOS.

1. Щелкните на кнопке Start (Пуск) панели задач, выберите Run (Выполнить), введите command в текстовом окне Open (Открыть) и щелкните на кнопке ОК. Откроется окно DOS.
2. Запустите SQL*Plus: введите sqlplus <имя_пользователя>/<пароль> и нажмите клавишу <Enter>, подставив вместо <имя_пользователя> свое имя пользователя, а вместо <пароль> — пароль, разделив их косой чертой. (Помните, что в данном случае пароль *отображается* на экране!)
3. Введите команду SQL*Plus CLEAR SCREEN (прописными или строчными буквами) и нажмите клавишу <Enter>, чтобы очистить экран.
4. После запроса SQL> введите START, нажмите клавишу <пробел>, введите полный путь к файлу (Ch02SaleStatus.sql), обратную косую черту Ch02SaleStatus.sql (использование прописные букв необязательно). После этого нажмите клавишу <Enter>. Если, например, вы записали файл (в данном случае называемый сценарием) в корне диска C, то вам нужно будет набрать следующую строку и нажать клавишу <Enter>:

```
SQL> START C:\Ch02SaleStatus.sql
```

5. В ответ Oracle отобразит подтверждение “Table created” (“таблица создана”).
6. Окно DOS не закрывайте — мы к нему еще вернемся.

Редактирование команд SQL

Если вам требуется изменить только что введенную команду SQL (возможно, нужно исправить ошибку), вызовите редактор, введя в запросе SQL> команду edit. Windows откроет программу Notepad (Блокнот), позволяя изменить команду, а затем записать результат.

Итак, чтобы запустить команду SQL в SQL*Plus, активизируйте открытое ранее окно DOS, а затем выполните нижеприведенные действия.

1. Введите edit и нажмите клавишу <Enter>. Windows откроет программу Блокнот — предопределенный редактор для окна DOS. В окне данной программы откроется сценарий CREATE TABLE, записанный в файле Ch02SaleStatus.sql. Содержимое этого сценария записывается во временный файл afiedt.buf (имя файла предопределено Oracle).
2. Если потребуется, щелкните мышью слева от первой строки перед словом CREATE.
3. Введите --Create a new table и нажмите клавишу <Enter>. Любая строка, начинающаяся с двух минусов, является комментарием, который Oracle игнорирует.

4. Щелкните на слове File (Файл) в строке меню, выберите Exit (Выход) и щелкните на кнопке Yes, чтобы сохранить изменения.
5. Измененное выражение SQL (включая добавленный комментарий) появится ниже сообщения Wrote file afiedt.buf.
6. Чтобы выполнить модифицированное выражение SQL, введите косую черту и нажмите клавишу <Enter>. В ответ Oracle сгенерирует сообщение об ошибке (ORA-00955), поскольку таблица уже существует, а выражение SQL пытается создать ее повторно.
7. Введите exit и нажмите клавишу <Enter>, чтобы выйти из SQL*Plus. Еще раз наберите exit и нажмите клавишу <Enter>, чтобы закрыть окно DOS.

Использование SQL*Plus for Windows

Гораздо легче использовать среду SQL*Plus с помощью SQL*Plus for Windows, поскольку в этом случае мы имеем возможность работать с диалоговым окном Windows и более понятным интерфейсом. Кроме того, в SQL*Plus for Windows легче задавать положения файлов. Если вы можете выбирать, какой интерфейс операционной системы использовать, — DOS- или Windows-версию, — советуем выбрать последнюю.

При входе в Oracle с помощью SQL*Plus for Windows отображается запрос командной строки SQL> и предлагается среда, обладающая описанными ниже характеристиками.

- Принимает вводимые вами выражения SQL и передает их серверу Oracle.
- Интерпретирует команды редактора, форматирования и работы с файлами.
- Форматирует результаты, возвращаемые в ответ на запросы SQL, и отображает отчеты на экране.
- Помнит установленные вами настройки среды (например, число строк между заголовками и ширина строки экрана).
- Предлагает простой построчный редактор, поддерживающий ввод и модификацию набираемых выражений.
- Принимает выражения SQL, записанные в файлах.

Вход в Oracle

Для того чтобы войти в Oracle с помощью SQL*Plus for Windows, выполните следующие действия.

1. Щелкните на кнопке Start (Пуск), выберите All Programs (Все программы) (Windows XP), затем выберите Oracle–OraDb10g_home1, выберите ApplicationDevelopment и щелкните на SQLPlus. Откроется диалоговое окно LogOn.

Совет. Для удобства можно создать ярлык для SQL*Plus, щелкнув правой кнопкой мыши на SQLPlus, выбрав Send To (Отправить), а затем Desktop (create shortcut) (Рабочий стол (создать ярлык)).

2. Введите имя пользователя в текстовом окне User Name и нажмите клавишу <Tab>. Введите пароль в поле Password. Если потребуется, нажмите клавишу <Tab> и, если необходимо, введите строку хоста в текстовом окне Host String.

Совет. Если вы неосторожно нажали клавишу <Enter> после ввода имени пользователя, Oracle закроет диалоговое окно Log On и запросит у вас пароль. Введите его и нажмите клавишу <Enter>.

3. Щелкните на кнопке OK, чтобы войти в базу данных Oracle. Для удобства можете развернуть окно SQL*Plus на весь экран.

Если вы все правильно ввели, SQL*Plus предоставит вам доступ к базе данных Oracle, покажет номер версии SQL*Plus, дату и время, а также уведомление об авторских правах. Кроме того, будет указана база данных Oracle (например, Personal Oracle), с которой вы соединились. В последней строке отображается приглашение командной строки SQL>, свидетельствующее о том, что приложение SQL*Plus готово получать команды. (Окно SQL*Plus не закрывайте — оно нам еще понадобится.)

Если вы получите сообщение об ошибке, например ORA-01017: invalid username/password; logon denied, это означает, что вы ввели неверное имя пользователя или пароль. Спросите у своего администратора базы данных правильное имя и пароль. Если вы увидите сообщение об ошибке с кодом ORA-12154, например ORA-12154: TNS: could not resolve the connect identifier specified, — обратитесь к администратору.

После того как вы успешно вошли в систему и увидели приглашение SQL>, вы начали новый сеанс Oracle. После приглашения вы можете ввести только одну команду SQL*Plus. Для очень больших команд просто нажимайте клавишу <Enter> между вводом слов, чтобы перевести строку. Никакого ограничения на число строк в команде SQL*Plus не существует.

Ввод и запуск команд SQL

После входа в Oracle через интерфейс Windows SQL*Plus вы можете выполнять команды SQL и SQL*Plus.

Чтобы отобразить структуру таблицы SaleStatus и добавить в нее строки данных, выполните следующие действия.

1. После отображенного приглашения SQL введите команду SQL*Plus describe salestatus, которая перечислит названия всех столбцов таблицы и их типы дан-

ных, а затем нажмите клавишу <Enter>. Oracle извлечет и отобразит информацию о таблице `salestatus`. Обратите внимание на то, что названия столбцов написаны прописными буквами (они хранятся именно в таком виде).

2. Введите в таблицу какие-нибудь данные, используя команду `INSERT`. После `SQL>` введите указанные ниже строки, нажав клавишу <Enter> после слова `SaleStatus`. Введите точку с запятой, но *не нажимайте* клавишу <Enter> в конце второй строки.

```
INSERT INTO SaleStatus  
VALUES (101, 'For Sale');
```

3. Нажмите клавишу <Enter>, чтобы отправить завершенное выражение `SQL` к `SQL*Plus`, а затем к `Oracle` для выполнения. `Oracle` выполняет команду, если последним символом, набранным перед нажатием клавиши <Enter>, была точка с запятой (это нужно для работы с одно- или многострочными командами). Появится сообщение “1 row created”, которое означает, что вы успешно добавили данные в таблицу.
4. Введите следующие два выражения `SQL`, нажимая клавишу <Enter> в конце каждой строки, завершающейся точкой с запятой. (Использование прописных букв существенно *только* в данных в одинарных кавычках. Наберите текст точно так, как показано ниже.)

```
INSERT INTO SaleStatus VALUES (102, 'Pending');  
INSERT INTO SaleStatus VALUES (103, 'Sold');
```

Если выражения были введены правильно, после каждого выражения `SQL*Plus` отобразит сгенерированное `Oracle` сообщение — “1 row created” (рис. 2.5).

5. Введите `CLEAR SCREEN` и нажмите клавишу <Enter>, чтобы очистить экран `SQL*Plus` (окно `SQL*Plus` не закрывайте).

Редактирование команд SQL

Команды, набранные в `SQL*Plus`, можно изменить, набрав `edit` в командной строке `SQL>` (процедура аналогична рассмотренной выше для редактирования команд `SQL` в окне `DOS`). Тем не менее при работе с книгой мы рекомендуем использовать другой метод, чтобы вы могли записывать все, что набираете, и спокойно его редактировать. Довольно часто команды `SQL` длинные, что затрудняет их повторный набор для устранения ошибок. Мы считаем, что лучшей альтернативой является создание команд `SQL` и `SQL*Plus` с помощью текстового редактора, подобного Блокноту (использование текстового процессора, подобного `Word`, не рекомендуется).

Набрав команду `SQL`, вы копируете ее в буфер `Windows` <Ctrl+A>, чтобы выделить весь текст, затем выбираете из меню `Copy` (Копировать), активизируете окно `SQL*Plus`, вставляете в него команду и запускаете ее. Если возникнет ошибка, исполь-

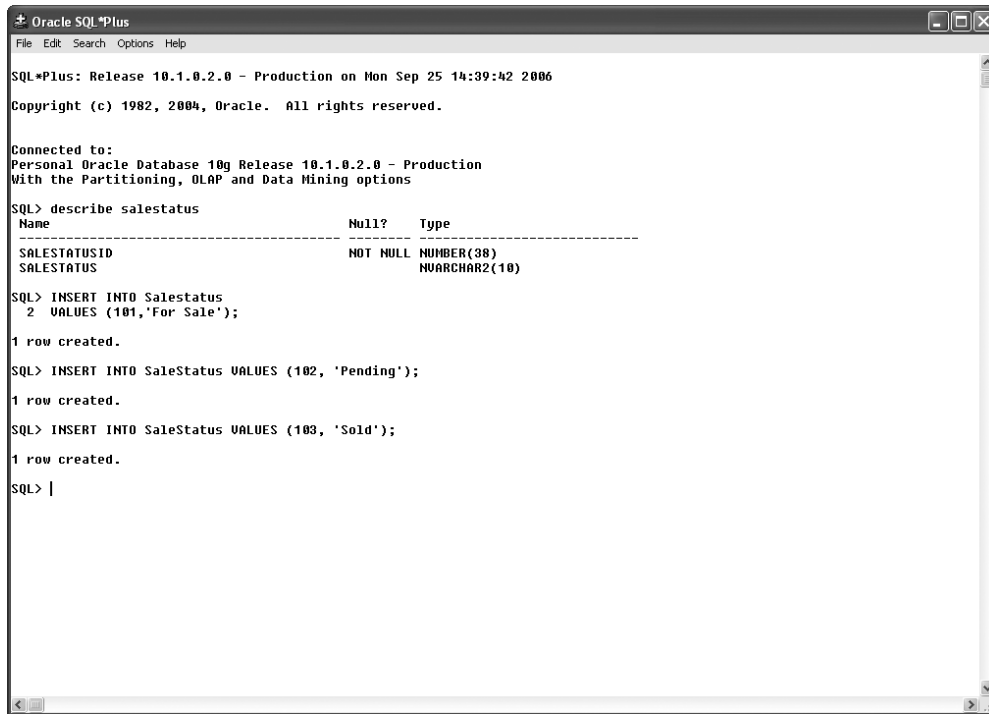


Рис. 2.5. Ввод команды SQL через окно SQL*Plus

зуете Блокнот и исправляете ошибку, а затем копируете текст в SQL*Plus. Преимущество такого подхода заключается в том, что файл Блокнота можно сохранить, а затем по необходимости загрузить в SQL*Plus повторно. Использование команды `Edit` в командной строке SQL*Plus можно назвать более неуклюжим способом. Любые записываемые команды SQL (или группы отдельных команд в одном файле, разделенные точкой с запятой) называются *сценарием*. Вообще, удобно иметь копии многих (если не всех) написанных вами строк SQL как запись структур таблиц и данных, содержащихся в этих таблицах. Когда сценарии безопасно хранятся на диске, вы можете запускать их на другом компьютере с Oracle, открыв SQL*Plus и набрав `start` и полный путь к файлу сценария (включая его название). Ниже мы попрактикуемся это делать.

Для того чтобы отредактировать команду SQL в Блокноте, а затем выполнить ее в SQL*Plus, выполните следующие действия.

1. В окне SQL*Plus введите приведенные ниже команды. Нажмите клавишу `<Enter>` после ввода названия таблицы (`SaleStatus`) и еще раз после точки с запятой во второй строке.

```

INSERT INTO SaleStatus
VALUES (101, 'Unknown');
    
```

Oracle возвратит сообщение об ошибке, указывающее, что система обнаружила нарушение условия уникальности.

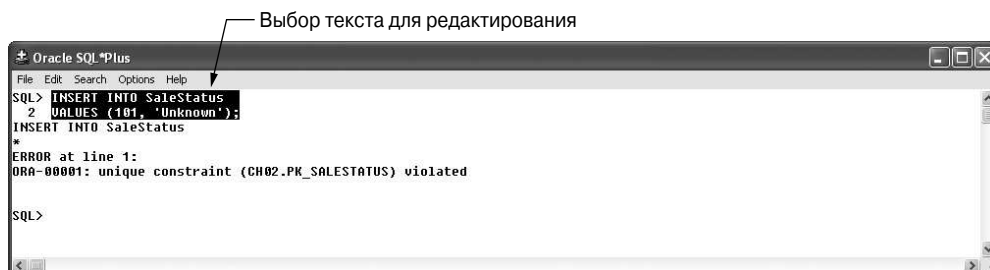


Рис. 2.6. Выбор редактируемого текста

2. Щелкните мышью в начале первой строки и растяните прямоугольник выделения до точки с запятой во второй строке. Приглашение SQL, номера строк, повторенную команду и сообщение об ошибке выделять не нужно (рис. 2.6).
3. Выберите Edit из строки меню и щелкните на позиции Copy.

Совет. Для экономии времени нажмите комбинацию клавиш <Ctrl+C>, чтобы скопировать выбранный текст в буфер.

4. Запустите Блокнот (щелкните на кнопке Start (Пуск), затем выберите AllPrograms (Всепрограммы), Accessories (Стандартные) и Notepad (Блокнот)) или запустите любой другой текстовый редактор.
5. Находясь в программе Блокноте, щелкните на кнопке Edit (Правка), затем выберите Paste (Вставить) (или нажмите комбинацию клавиш <Ctrl+V>). Команда SQL появится в Блокноте. Теперь вы можете исправить ее и записать результат в виде файла.
6. Исправьте команду: измените значение 101 на 104 — это идентификатор новой строки таблицы.
7. Скопируйте исправленную команду в SQL*Plus: нажмите комбинацию клавиш <Ctrl+A>, чтобы выбрать весь текст, нажмите комбинацию клавиш <Ctrl+C>, чтобы скопировать его в буфер, переключитесь на окно SQL*Plus, нажмите <Ctrl+V>, чтобы вставить новый код в окно SQL*Plus, и нажмите клавишу <Enter>, чтобы запустить команду. На этот раз она выполняется правильно и Oracle возвращает подтверждающее сообщение — “1 row created”.
8. Откройте окно Блокнота, откройте меню File (Файл), а затем выберите Save (Сохранить). Дойдите до папки, в которой хотите сохранить файлы сценариев для главы 2, введите Ch2<имя>SaleStatus.sql в окне File name (Имяфайла) (вместо “<имя>” введите свое имя), откройте список Save as type (Тип файла), выберите All Files (Всефайлы), а затем щелкните на кнопке Save.
9. Закройте программу Блокнот. Если вы собираетесь активно вводить команды SQL*Plus, окно Блокнота удобно держать открытым.

Выход из SQL*Plus

Выйти из SQL*Plus можно тремя способами.

- Выберите File в строке меню, а затем Exit.
- Введите Exit в командной строке SQL>.
- Щелкните на значке X в строке заголовка.

При выходе из SQL*Plus Oracle отключает вас и автоматически закрывает соединение с базой данных, т.е. описанная выше последовательность действий — это процедура выхода из Oracle.

Использование iSQL*Plus

Другим способом доступа к Oracle является использование *iSQL*Plus* — основанной на Web среде SQL*Plus. Для обращения к *iSQL*Plus* необходимо ввести URL вида `http://<имя_компьютера>.<имя_домена>:<порт>/isqlplus/`.

В предыдущем примере `<имя_компьютера>` — это имя компьютера, а значение `<порт>` равно 5560. Если вы устанавливали Oracle на собственный компьютер, вы сможете обращаться к *iSQL*Plus* с помощью URL `http://127.0.0.1:5560/isqlplus/`.

Если ни один из указанных URL не подошел, обратитесь к администратору, чтобы он выдал необходимые сведения. На рис. 2.7 показан экран регистрации в *iSQL*Plus* с использованием имени компьютера PURDUE. Имя домена не указывается, поскольку среда *iSQL*Plus* располагается на той же машине, что и база данных Oracle (компьютере PURDUE). Точно так же, как при использовании SQL*Plus, вы вводите имя пользователя, пароль и (возможно) идентификатор соединения, после чего щелкаете на кнопке Login и входите в базу данных Oracle. Если вы не уверены насчет значения идентификатора соединения — обратитесь к преподавателю.

Вход в Oracle

Ну что ж, попробуем войти в Oracle через *iSQL*Plus* — версию SQL*Plus на основе Web.

Для того чтобы войти в Oracle с помощью *iSQL*Plus* и ввести команду SQL, выполните следующие действия.

1. Запустите Web-браузер и введите URL базы данных Oracle. Если вы не уверены насчет значения URL — обратитесь к преподавателю. Отобразится Web-страница *iSQL*Plus*.
2. Введите в текстовом окне Username имя пользователя, нажмите клавишу <Tab> и введите пароль.

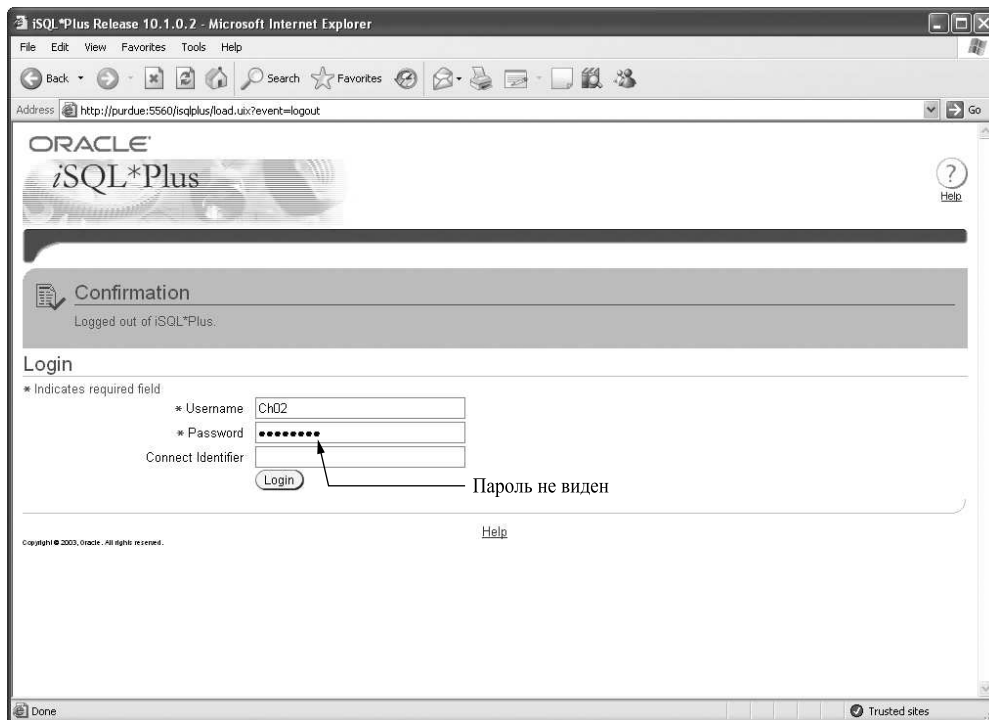


Рис. 2.7. Вход в Oracle с использованием iSQL*Plus

3. Если нужно заполнить поле Connect Identifier, нажмите клавишу <Tab> и введите идентификатор соединения. На рис. 2.7 показана Web-страница iSQL*Plus с выбранным нами именем пользователя (Ch02) и паролем.
4. Щелкните на кнопке Login, чтобы войти в базу данных Oracle. Если вы ввели правильное имя пользователя и пароль, iSQL*Plus отобразит Web-страницу Workspace.
5. Щелкните на окне с пометкой “<Enter> SQL, PL/SQL and SQL*Plus statements”, чтобы поместить в окно точку ввода.

Ввод и запуск команд SQL

Ввод команд SQL и SQL*Plus происходит достаточно понятно. Предлагаемый метод может вам понравиться: он позволяет перемещать точку ввода в пределах команды, вносить изменения, а затем отправлять команду на обработку с помощью кнопки Execute.

Для того чтобы ввести команду SQL и выполнить ее, выполните следующие действия.

112 Глава 2. Обзор SQL и SQL*Plus

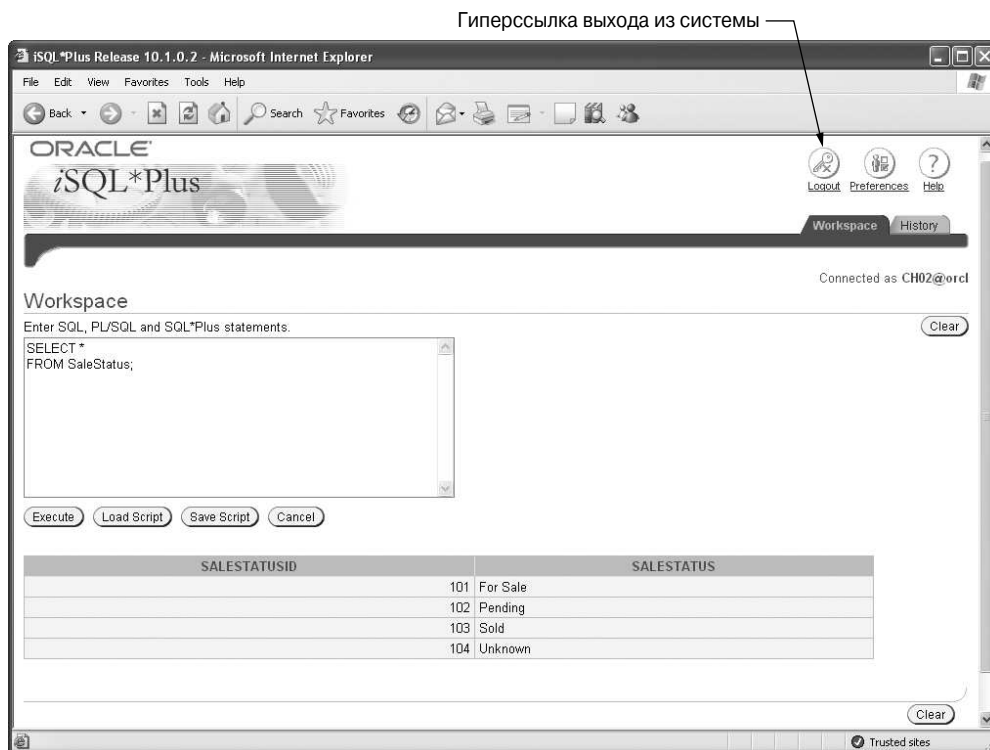


Рис. 2.8. Команда SQL и извлеченные строки в браузере

1. Введите следующее выражение SQL:

```
SELECT *
FROM SaleStatus;
```

2. Щелкните на кнопке Execute, расположенной под окном, в котором вы ввели команду. Результат ее выполнения будет показан внизу окна: появятся строки, записанные в созданной вами таблице SaleStatus (рис. 2.8).

Итак, вы вводите команду в рабочей области, а результат ее выполнения появляется внизу экрана. Если результат занимает больше места, чем может вместить экран, iSQL*Plus отображает кнопку Next Page внизу экрана, щелкнув на которой можно открыть на следующие окна. Среда iSQL*Plus также позволяет выполнять набранные команды, загружать файлы сценариев SQL, записывать сценарии, появляющиеся на экране, а также проверять историю введенных команд (гиперссылка History); кроме того, имеется множество других возможностей. Щелкнув на кнопке Help вверху экрана, вы откроете в окно интерактивной помощи.

Редактирование команд SQL

Команды SQL или SQL*Plus редактируются точно так же, как текст в текстовом редакторе: вы устанавливаете курсор там, где требуется внести изменения, вносите правку, а затем щелкаете на кнопке **Execute**, чтобы выполнить модифицированную команду. Чтобы записать измененный вариант, щелкните на кнопке **Save Script**, а затем выберите, в какой папке сохранить файл сценария.

Чтобы отредактировать команду SQL в окне браузера *iSQL*Plus*, выполните следующие действия.

1. Щелкните мышью между словом `SaleStatus` и заключительной точкой с запятой.
2. Нажмите клавишу `<Enter>`, чтобы перейти на новую строку, а затем введите
`ORDER BY SaleStatus DESC`
3. Щелкните на кнопке **Execute**, чтобы отобразить новые результаты. Возвращаемые строки Oracle сортирует в поле `SaleStatus` в порядке убывания.
4. Запишите новый сценарий как файл: щелкните на кнопке **Save Script**, если появится диалоговое окно, предупреждающее об опасности загрузки файлов из Интернета, щелкните на кнопке **Save**.
5. Дойдите до папки, в которой записаны сценарии главы 2, введите имя `Ch2<имя>ReverseSaleStatus.sql`, в окне **Save as type** выберите **All Files** и щелкните на кнопке **Save**. После этого *iSQL*Plus* запишет ваш файл сценария. Чтобы загрузить его впоследствии, щелкните на кнопке **Load Script**, затем на кнопке **Browse** и введите путь к файлу, который требуется загрузить (позже мы попрактикуемся в этом).
6. Щелкните на кнопке **Clear**, чтобы очистить окно **Workspace**.

Выход из iSQL*Plus

Для того чтобы выйти из *iSQL*Plus*, выполните следующие действия.

1. Щелкните на ссылке **Logout** (вверху и внизу окна), чтобы выйти из Oracle. Oracle отобразит подтверждение, сообщающее, что выход прошел правильно.
2. Закройте Web-браузер.

Выражения SQL

Фундаментальными операциями, выполняемыми через интерфейс SQL*Plus, являются организация запросов к базе данных, создание и удаление различных объектов

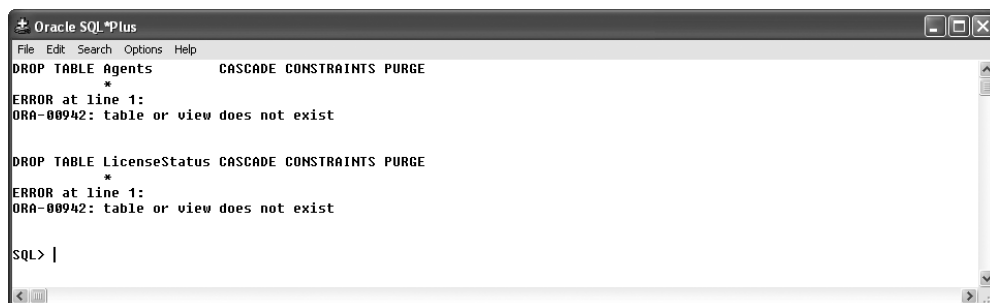


Рис. 2.9. Сообщения об ошибках, генерируемые при выполнении команд `DROP TABLE`

базы данных (включая таблицы), а также добавление, обновление и удаление строк таблиц с помощью команд SQL. Ниже мы рассмотрим структуру всех пяти типов выражений SQL, разберем их использование на примере базы данных Redwood Realty и будем периодически создавать команды SQL для исследования их влияния на базу данных. Чтобы ознакомиться с различными выражениями SQL, вначале нужно запустить сценарий SQL, определяющий две новые таблицы и заполняющий их данными. Создав хотя бы несколько таблиц для базы данных Redwood Realty, можно будет уже будет поэкспериментировать с описанными в книге командами SQL.

Итак, чтобы использовать файл сценария для создания двух таблиц и заполнения их данными, выполните следующие действия.

1. Скопируйте с компакт-диска (папка, соответствующая главе 2) файл сценария `Ch02Agents.sql` и запишите или запомните полный путь к нему (например, `C:\Temp\` или `C:\My Documents\SQL\`).
2. Если требуется, запустите SQL*Plus и введите свое имя пользователя и пароль. Войдите в систему.
3. В командной строке `SQL>` наберите `START <путь>\Ch02Agents.sql`, где `<путь>` — полный путь к файлу сценария.
4. Нажмите клавишу `<Enter>`, чтобы запустить серию команд SQL из файла сценария. В ответ Oracle выведет два сообщения об ошибках (рис. 2.9) под каждой из команд `DROP TABLE`. Не обращайтесь на них внимания, они просто указывают, что вы начинаете работу с двух пустых таблиц.
5. Введите выражение `SQL*Plus CLEAR SCREEN` (можно и строчными буквами) и нажмите клавишу `<Enter>`, чтобы очистить экран SQL*Plus от всех сообщений.
6. Введите `exit` и нажмите клавишу `<Enter>`, чтобы закрыть SQL*Plus и выйти из Oracle.

При выполнении файла сценария создаются две таблицы, `Agents` и `LicenseStatus`, которые затем заполняются данными. Таблица `Agents` содержит 29 строк,

в которых перечисляются агенты по недвижимости, работающие в компании Redwood Realty. Таблица `LicenseStatus` содержит 16 строк с информацией о состоянии лицензий агентов (`Licensed` (“действительна”), `Expired` (“истек срок действия”), `Revoked` (“аннулирована”) и т.д.).

Запуск запросов SQL

Пожалуй, основной задачей языка SQL является *организация запросов* к базе данных и получение ответов на них. Для извлечения информации в SQL применяется команда `SELECT`. Данная команда извлекает информацию из одной или нескольких таблиц. Она задает, какие поля таблицы (включая расчеты в этих полях) извлечь, к каким таблицам обратиться, чтобы найти затребованные поля, какие строки таблицы включить в возвращаемые данные и в каком порядке должны идти строки.

Рассмотрим синтаксис команды `SELECT`. Вообще, *синтаксис* — это правила, касающиеся правописания и грамматики языка; в нашем случае — SQL. Синтаксические правила не допускают вольностей и позволяют получить ответ, только если вы правильно набрали выражение (“правильно” — с точки зрения правил синтаксиса). В данном и последующих примерах, приведенных в данной книге, в квадратных скобках ([и]) указываются необязательные части выражения, а в фигурных ({ и }) приводится список элементов, разделенных вертикальной чертой, один из которых вы должны выбрать.

```
SELECT [DISTINCT|ALL] {*| [column_expression [AS new_name]] [, ...]}
FROM table_name
[JOIN table_name [alias] ON (table_column <operator>table_column)]...
WHERE search_condition]
[GROUP BY column_list] [HAVING search_condition]
[ORDER BY column_list]
```

Возможно, что вначале синтаксис команды `SELECT` покажется вам непонятным. Не беспокойтесь. Вы настолько часто будете использовать команду `SELECT` во множестве различных форм, что быстро привыкнете к ней. Например, с помощью этой команды можно перечислить содержимое таблицы `Agents` (все строки и столбцы):

```
SELECT * FROM Agents;
```

Звездочка указывает, что перечислить нужно все существующие столбцы таблицы, а оператор `FROM` — что Oracle должна вернуть строки из таблицы `Agents`. В результате мы получаем 15 столбцов и 29 строк. Чтобы получить часть данной информации, можно использовать следующий запрос `SELECT`:

```
SELECT firstname, lastname, gender
FROM agents
WHERE gender = 'F'
ORDER BY lastname, firstname;
```

Имена столбцов после оператора `SELECT` указывают на то, что из таблицы `Agents` нас интересуют только имя, фамилия и пол агента. Более того, условие `WHERE` огра-

116 Глава 2. Обзор SQL и SQL*Plus

ничивает показываемые результаты строками с информацией по агентам-женщинам (`gender = 'F'`). Перед отображением результатов оператор `ORDER BY` сортирует строки по фамилии, а затем по имени, если встречаются люди с одинаковой фамилией. Попрактикуемся использовать выражение `SQL SELECT` с интерпретатором *iSQL*Plus*. Прежде всего закройте клиентскую программу *SQL*Plus*, если она еще запущена, введя `exit` в командной строке `SQL>`.

Чтобы с помощью `SELECT` отобразить на экране избранные строки таблицы `Agents`, выполните следующие действия.

1. Запустите Web-браузер, затем *iSQL*Plus*. (Введите имя пользователя и пароль и щелкните на кнопке `Login`.)
2. Установите курсор в окне `Workspace` и введите приведенные ниже четыре строки. Использование прописных букв не критично за исключением буквы “M” в одинарных кавычках (третья строка). Обратите внимание на отсутствие завершающей точки с запятой, поскольку *iSQL*Plus* это не требуется.

```
SELECT firstname, lastname, gender, licensestatusid
FROM agents
WHERE gender = 'M'
ORDER BY licensestatusid, lastname
```

3. Щелкните на кнопке `Execute`, чтобы передать команду Oracle. В ответ возвращается 15 строк (рис. 2.10).
4. Щелкните на кнопке `Logout`. (Данная кнопка есть вверху и внизу экрана.)
5. Закройте браузер. Существуют и другие способы записи команды `SELECT`; их мы рассмотрим в главе 3.

DLL-выражения

В число самых используемых выражений `SQL`, написанных на языке `DDL` (`Data Definition Language` — язык определения данных), входят `CREATE TABLE`, `ALTER TABLE` и `DROP TABLE`. Существуют и другие `DDL`-выражения, которые будут рассмотрены в следующих главах, однако приведенные операторы являются наиболее характерными представителями этой группы.

Создание таблицы

Одним из основных выражений языка определения данных является `CREATE TABLE`. `SQL` поддерживает три типа таблиц: базовые, производные и наблюдаемые таблицы. Вы преимущественно будете работать с *базовыми таблицами* — объектами схемы, содержащими данные (как таблица `Agents`, созданная при запуске файла сценария `Ch02Agents.sql`). *Производные таблицы* представляют собой результаты, возвращае-

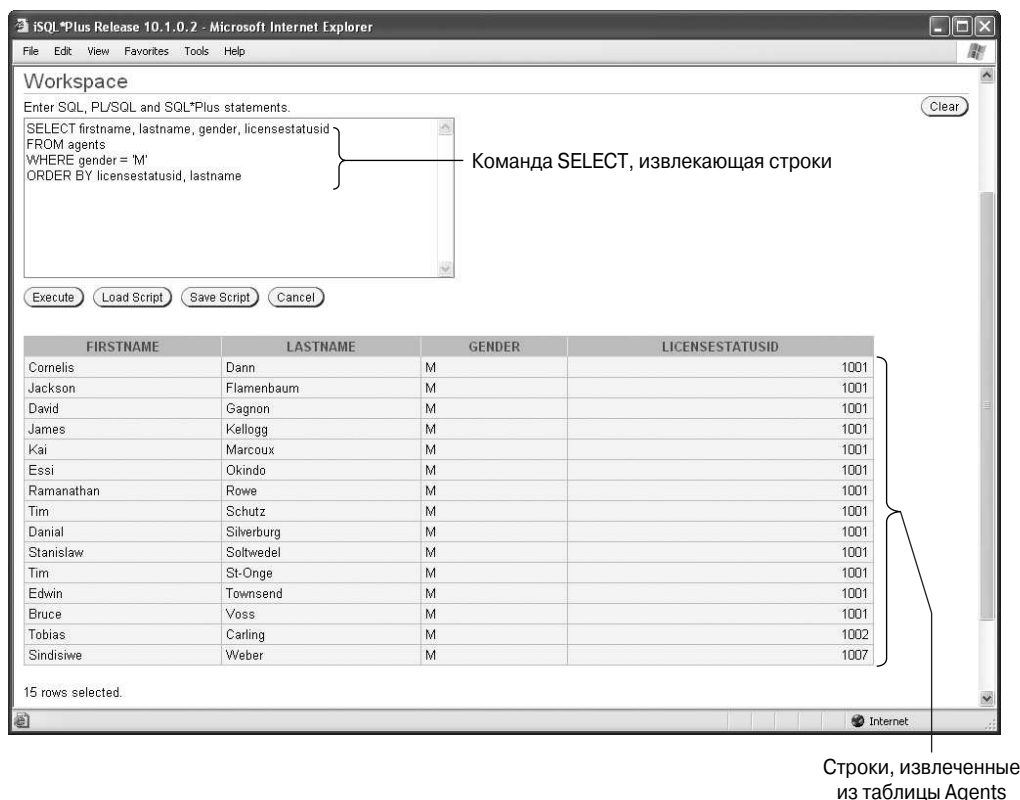


Рис. 2.10. Строки, возвращаемые в ответ на команду SELECT

мые системой баз данных в ответ на запрос (SELECT). *Наблюдаемые таблицы*, подробно рассмотренные в главе 6, являются именованными производными таблицами. Наблюдаемые таблицы обычно представляют собой сложные выражения SELECT, которым присвоено имя, позволяющее обращаться к ним так же, как к обычным базовым таблицам. Ниже мы изучим структуру и использование команды CREATE TABLE для создания базовой таблицы.

Ниже приведен сокращенный синтаксис команды CREATE TABLE, применяющийся при создании базовой таблицы. (Полное описание синтаксиса CREATE TABLE можно найти в руководстве SQL Reference Oracle Corporation.)

```
CREATE TABLE table_name
(column_name type [NOT NULL]
[CONSTRAINT constraint_definition DEFAULT expression]
[,column_name type [NOT NULL]
[CONSTRAINT constraint_definition DEFAULT expression]...]
)
[PRIMARY KEY (column_name [,column_name ...])
[FOREIGN KEY (column_name [,column_name ...])
REFERENCES foreign_table (column_name [,column_name ...])]
```

Приведенная ниже команда CREATE TABLE создает для базы данных Redwood Realty таблицу ContactReason, содержащую два столбца, с помощью которых указывается причина контакта клиента с агентством. На данный момент предусмотрено три причины: Buy (Покупка), Casual (Случайно), Sell (Продажа). Причина Buy означает, что клиент интересуется покупкой дома; Casual — что человек просто “зашел посмотреть” и в текущий момент не планирует операций с недвижимостью; Sell указывает, что соответствующий клиент связался с Redwood Realty для продажи дома. Итак, указанная команда CREATE TABLE создает таблицу ContactReason и определяет ContactReason как первичный ключ (напомним, что первичный ключ единственным образом определяет все строки таблицы).

```
CREATE TABLE ContactReason
  (ContactReason NVARCHAR2(15) NOT NULL,
   Description NVARCHAR2(50),
   CONSTRAINT pk_ContactReason PRIMARY KEY (ContactReason)
);
```

В первой строке таблице присваивается имя. После этого в круглых скобках идет список названий столбцов. Третья строка содержит определение связи для одного столбца. *Условие* (constraint) — это ограничение на значение, которое может иметь столбец. Условие указывает, что первый столбец, ContactReason, является первичным ключом. Следовательно, каждое значение в этом столбце должно быть уникальным для данной таблицы (значения Buy, Casual и Sell уникальны — каждое из них появляется только в одной строке таблицы).

После каждого имени столбца следует тип данных и размер. Тип данных столбца ContactReason равен NVARCHAR2 — строка символов национального языка размером до 15 символов. (*Набор символов национального языка* позволяет записывать информацию из языков, отдельные символы которых нельзя представить в восьмибитовой схеме кодирования.) Необязательный оператор NOT NULL указывает, что значение в этом столбце нельзя пропускать. Второй столбец, Description, позволяет использовать строки символов национального языка размером до 50 символов.

Добавление и отбрасывание столбцов

Одной из удобных особенностей реляционных баз данных является возможность добавлять, изменять и удалять столбцы. Предположим, что таблица сотрудников содержит несколько сотен строк с типичной информацией о персонале: имя, адрес, телефон, дата найма и дата рождения. За последний год (или около того) каждый сотрудник получил для использования в разъездах служебный мобильный телефон. Изначально таблица не содержала столбца, предназначенного для номера мобильных телефонов. Можете ли вы добавить еще один столбец для новой информации? Да! Используйте для этой цели оператор SQL ALTER TABLE. Ниже показана команда, позволяющая добавить в таблицу Employee столбец MobilePhone.


```
ALTER TABLE Employee  
ADD MobilePhone VARCHAR2(14);
```

Oracle добавляет столбец `MobilePhone` в конец таблицы и присваивает всем его позициям значение `NULL`. Позже в нужные строки таблицы можно ввести номера мобильных телефонов. Подобным образом можно модифицировать или удалять существующие столбцы таблицы. Для этого используются соответственно команда `ALTER TABLE` с уточнением `MODIFY` или `DROP COLUMN`. Например, приведенная ниже команда SQL удаляет из таблицы столбец `YearStarted`.

```
ALTER TABLE Employee  
DROP COLUMN YearStarted;
```

Подобным образом указанная ниже команда заменяет текущий тип данных столбца типом `DATE`.

```
ALTER TABLE Employee  
MODIFY YearStarted DATE DEFAULT SYSDATE;
```

Кроме того, оператор `DEFAULT` гарантирует, что в любое незаполненное поле `YearStarted` будет автоматически внесена сегодняшняя дата — значение, возвращаемое функцией `SYSDATE`.

Время от времени требуется удалять нежелательные позиции таблицы. Предположим, например, что вам больше не требуется таблица `CheapWebHost`, содержащая имена и URL дешевых Web-хостов. Вы уже выбрали хост, установили на него систему, поэтому указанная таблица стала бесполезной. Разумеется, можно удалить все строки, однако при этом останется структура таблицы. Чтобы вместе со всеми строками убрать определение таблицы, используйте команду `DROP TABLE`, имеющую следующий синтаксис:

```
DROP TABLE table_name [CASCADE CONSTRAINTS] [PURGE]
```

При удалении таблицы с опцией `CASCADE CONSTRAINTS` таблица вместе со всей содержащейся в ней информацией, другими объектами, ссылающимися на таблицу или зависящими от нее (включая представления, процедуры и присоединенные процедуры; см. ниже), переносится в корзину. При использовании необязательного оператора `PURGE` таблица удаляется, но не помещается в корзину. Вместо этого она безвозвратно удаляется из системы. Например, приведенная ниже команда удаляет таблицу `CheapWebHost` и очищает корзину.

```
DROP TABLE CheapWebHost CASCADE CONSTRAINTS PURGE;
```

DML-выражения

Одной из фундаментальных функций любой базы данных является возможность манипулирования содержащейся в ней информацией. Именно для этого предназначены SQL-выражения языка манипулирования данными (Data Manipulation Language — DML), позволяющие добавлять, модифицировать и удалять данные из таблиц базы

данных. Возможность вставки новых строк данных, модификации ошибочной или изменившейся информации, а также удаления устаревших записей необходима, чтобы информация не устаревала и была точной. Для реализации названных возможностей применяются выражения SQL INSERT, UPDATE и DELETE.

Вставка данных

Команда INSERT применяется для вставки значения в некоторые или во все столбцы (или поля) таблицы. Синтаксис стандартной команды INSERT достаточно прост:

```
INSERT INTO table_name [(column_name [, column_name ...])]
VALUES (value [, value...])
```

При вставке значений во все столбцы таблицы можно использовать следующую форму команды:

```
INSERT INTO table_name
VALUES (value, value, value, ...)
```

Обратите внимание на то, что в предыдущей форме опущен список столбцов таблицы — указываются только значения, которые они получают. Если вам требуется вставить значения во все столбцы, в команде INSERT оператор VALUES должен содержать значения для каждого столбца — ни одно значение пропустить нельзя. Если вы не знаете, какое значение должен иметь столбец, используйте вместо него слово NULL. (NULL — это зарезервированное слово, означающее, что значение неизвестно.) Ниже показано, как с помощью команды INSERT добавить новую строку в таблицу LicenseStatus, зная все значения и порядок, в котором идут поля таблицы.

```
INSERT INTO LicenseStatus
VALUES (1017, 'Reactivated');
```

При использовании данной формы список значений должен удовлетворять следующим условиям.

- Если в команде INSERT не заданы названия столбцов, для каждого столбца должно указываться значение, причем данные значения должны идти в том же порядке, в каком они определены в таблице.
- Неизвестные значения должны представляться значением NULL .

Если требуется ввести лишь несколько значений, то можно в любом порядке перечислить названия столбцов, получающих значения, а затем указать список VALUES, содержащий получаемые значения. Предположим, например, что в таблицу Agents требуется добавить нового агента по недвижимости, а все, что у вас есть, — это присвоенное брокером данному агенту значение AgentID, а также имя и фамилия агента. В таком случае проще всего написать команду INSERT со списком из трех столбцов, за которым в операторе VALUE следуют три значения. Попробуем это сделать с помощью SQL*Plus.

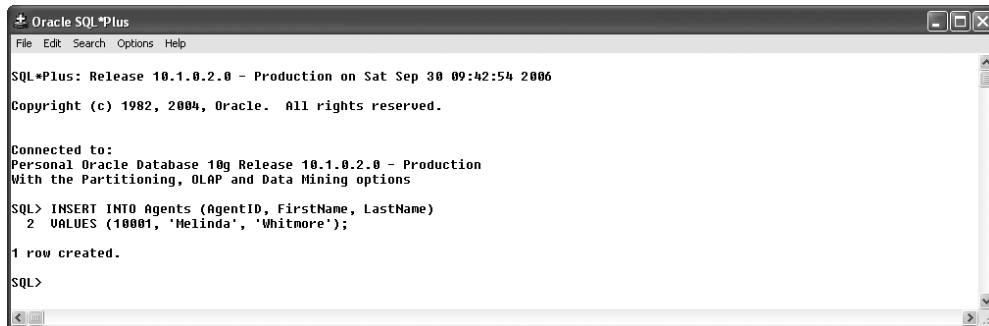


Рис. 2.11. Вставка новой строки и проверка ее значений

Для того чтобы добавить нового агента в таблицу Agents, зная только значения трех столбцов, выполните следующие действия.

1. Откройте SQL*Plus и войдите в Oracle со своим именем пользователя и паролем.
2. В командной строке SQL> наберите следующий текст:

```
INSERT INTO Agents (AgentID, FirstName, LastName)
VALUES (10001, 'Melinda', 'Whitmore');
```

3. Нажмите клавишу <Enter>, чтобы выполнить команду. SQL*Plus проверит команду и передаст ее базе данных Oracle. Oracle ответит сообщением “1 row created”, указывающим, что база данных добавила одну строку (рис. 2.11).
4. Проверьте, что строка действительно была вставлена, затребовав информацию об агентах. Для этого в командной строке SQL> введите следующее:

```
SELECT AgentID, FirstName, LastName
FROM Agents
WHERE AgentID < 10200;
```

5. Нажмите клавишу <Enter>, чтобы выполнить команду. Oracle покажет, что заданному критерию отвечают две строки, одна из которых была введена вами только что. Не закрывайте окно SQL*Plus — мы продолжим работать с ним позже.

Рассмотрим третью форму команды INSERT, использующую оператор SELECT для копирования строк из другой таблицы. В данной форме вместо VALUES применяется оператор SELECT:

```
INSERT INTO table_name (column_name1, column_name2, ...)
SELECT column1, column2, ...
FROM table_name [WHERE search_condition]
```

Здесь вы перечисляете столбцы, в которые требуется вставить значения, а затем с помощью команды SELECT извлекаете аналогичное число столбцов для вставки (тип новых и старых столбцов также должен совпадать). Необязательный оператор

122 Глава 2. Обзор SQL и SQL*Plus

WHERE позволяет фильтровать строки. Например, с его помощью вы можете добавить строки в таблицу Agents, копируя значения из другой таблицы, столбцы которой имеют тот же тип данных, что и столбцы таблицы Agents:

```
INSERT INTO Agents (AgentID, LastName, FirstName)
  SELECT NewAgentNumber, AgentLastName, AgentFirstName
  FROM NewlyLicensedAgents;
```

Помните, что тип данных столбцов таблицы Agent должен быть таким же, как у таблицы NewAgentNumber, кроме того, каждая строка, добавляемая в таблицу, должна иметь уникальный идентификатор нового агента, не совпадающий ни с одним номером в таблице Agents. Как видите, с помощью данного метода удобно копировать большие объемы данных из одной таблицы в другую.

Обновление строк

Вас не должно удивлять, что иногда информацию в таблицах приходится исправлять из-за ошибок ввода данных или обновлять из-за изменения информации. Адреса клиентов могут регулярно изменяться, сотрудников повышают по службе, а наличный товар обновляется постоянно. В связи с этим приходится использовать команду UPDATE, позволяющую модифицировать одно или несколько полей в одной или нескольких строках таблиц Oracle. Синтаксис команды UPDATE выглядит так:

```
UPDATE table_name
  SET column_name = expression [, column_name = expression, ...]
  [WHERE search_condition]
```

Видно, что оператор SET обязательный, а оператор WHERE — нет. В целом вы задаете имя обновляемой таблицы плюс пары “имя столбца–новое значение”. Если вы зададите условие WHERE, обновляться будут только строки, удовлетворяющие условиям поиска.

Предположим, что вам требуется добавить номера мобильного телефона и лицензии агента Melinda Whitmore (информацию, которой вы не располагали при вставке в таблицу соответствующей записи). Для решения этой задачи можно использовать команду UPDATE, найдя нужную запись и обновив требуемые значения.

Для того чтобы обновить три значения для определенного агента по недвижимости, выполните следующие действия.

1. В окне SQL*Plus введите CLEAR SCREEN, чтобы очистить экран от предыдущей информации.
2. Откройте программу Блокнот и введите приведенный ниже код (нажмите клавишу <Enter>, чтобы перейти на следующую строку для трех первых строк, однако *не нажимайте* клавишу <Enter> в конце четвертой строки). Будьте очень внимательны — номер мобильного телефона, имя и фамилия заключаются в одинарные кавычки. Прописные и строчные буквы используйте точно так, как показано ниже.

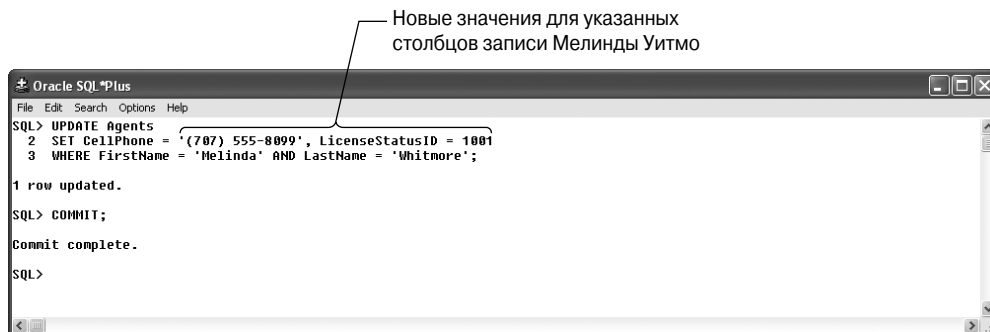


Рис. 2.12. Обновление строки в таблице Agents

(Оператор COMMIT рассмотрен ниже в этой главе.)

```
UPDATE Agents
SET CellPhone = '(707) 555-8099', LicenseStatusID = 1001
WHERE FirstName = 'Melinda' AND LastName = 'Whitmore';
COMMIT;
```

3. В программе Блокнот нажмите комбинацию клавиш <Ctrl+A> чтобы выбрать весь текст, затем нажмите комбинацию клавиш <Ctrl+C>, чтобы скопировать его в буфер.
4. Переключитесь на SQL*Plus, нажмите комбинацию клавиш <Ctrl+V>, чтобы вставить текст в SQL*Plus, и нажмите клавишу <Enter> чтобы выполнить команду COMMIT. Если SQL*Plus укажет на наличие синтаксических ошибок, переключитесь на Блокнот, исправьте нужное место и повторите шаг 3 и 4. Oracle отобразит сообщения: “1 row updated” и “Commit complete” (рис. 2.12). Теперь поля CellPhone и LicenseStatusID записи MelindaWhitmore будут обновлены новыми значениями. (Чтобы проверить внесенные изменения, можете использовать команду SELECT.)
5. Переключитесь на Блокнот, запишите команду SQL, если она вам нужна, и закройте Блокнот. Окно SQL*Plus не закрывайте.

Удаление строк из таблицы

Для удаления строк из таблицы базы данных применяется команда SQL DELETE. С ее помощью можно удалить одну, несколько или все строки таблицы. Впрочем, для удаления всех строк лучше использовать команду TRUNCATE. Разница между командами DELETE и TRUNCATE заключается в том, что в определенных обстоятельствах строки, удаленные с помощью команды DELETE, можно восстановить с помощью команды ROLLBACK, которая описана в следующем разделе. Если же строки были удалены с помощью команды TRUNCATE, то восстановить их невозможно. Кроме

124 Глава 2. Обзор SQL и SQL*Plus

того, команда `TRUNCATE` выполняет определенные дополнительные действия, например, высвобождает место, которое занимала таблица, и т.п. Команда `DELETE` имеет следующий синтаксис:

```
DELETE FROM table_name
[WHERE search_condition]
```

Команда `DELETE FROM` требует, чтобы вы указали таблицу, из которой необходимо удалить строки. Необязательная команда `WHERE` используется для того же, что и команда `SELECT` или `UPDATE`: для указания условий поиска, которым должна удовлетворять удаляемая строка. Если вы пропустите команду `WHERE` (распространенная ошибка новичков), то Oracle удалит все строки таблицы. Отметим, что удалять строки из таблицы можно как с помощью `DELETE`, так и с помощью `TRUNCATE`, хотя ни одна из названных команд не удаляет определение таблицы. Чтобы удалить таблицу и ее строки, применяйте команду `DROP TABLE` (она описана ранее в этой главе).

Для того чтобы удалить строку из таблицы `Agents`, выполните следующие действия.

1. Если вы закрыли SQL*Plus, откройте его и войдите в базу данных, используя ваше имя пользователя и пароль. Введите `CLEAR SCREEN` и нажмите клавишу `<Enter>`, чтобы начать с командной строки SQL и чистого экрана.
2. После `SQL>` введите приведенную ниже команду, чтобы удалить из таблицы `Agents` нового агента (`Melinda Whitmore`), и нажмите клавишу `<Enter>` в конце обеих строк:

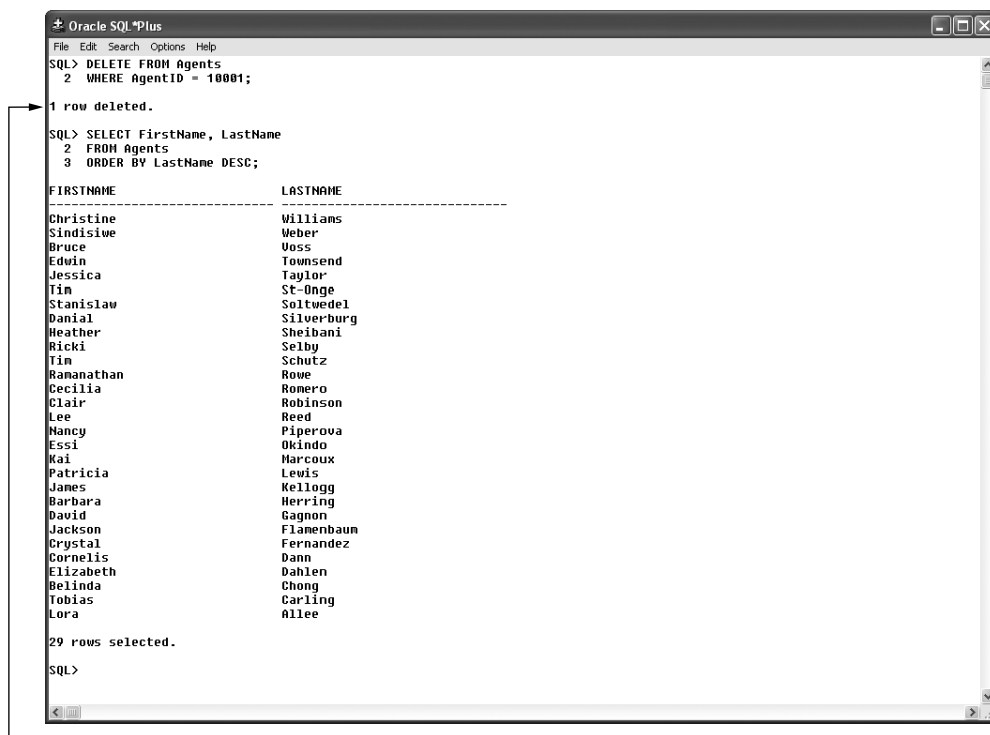
```
DELETE FROM Agents
WHERE AgentID = 10001;
```

Oracle отобразит подтверждающее сообщение “1 row deleted”.

3. Проверьте, что запись `MelindaWhitmore` удалена из таблицы, отобразив на экране имена и фамилии всех агентов. Для этого в командной строке `SQL>` введите следующую команду (рис. 2.13):

```
SELECT FirstName, LastName
FROM Agents
ORDER BY LastName DESC;
```

4. Как видите, запись `MelindaWhitmore` исчезла из списка агентов. (Изображение, которое вы увидите на экране, может отличаться от приведенного в книге и иметь больше заголовков столбцов. Это зависит от настройки среды SQL*Plus, именуемой `LINESIZE`. Подробно `LINESIZE` рассматривается в следующем разделе.) Не закрывайте SQL*Plus. Если вы закроете SQL*Plus, а затем откроете его, когда будете читать следующий раздел, некоторые действия будут выполняться неверно.



Подтверждение удаления указанной строки

Рис. 2.13. Удаление строки и проверка результата

ТС-выражения

Предназначение ТС-команд SQL понятно из названия: ТС — это аббревиатура от Transaction Control (“управление транзакциями”). *Транзакцией* базы данных называется группа выражений SQL, представляющих собой логическую единицу работы. Выражения в транзакциях представляют собой нераздельно связанную группу, отдельные выражения которой должны завершаться успешно. В противном случае необходимо “откатить” весь набор выражений SQL, чтобы база данных вернулась в состояние, предшествовавшее выполнению транзакции. Распространенным примером транзакции является перевод денег с депозита на текущий счет. Одно выражение UPDATE уменьшит количество денег на депозите на величину перевода, а последующая команда UPDATE увеличит на ту же величину количество денег на текущем счете. Если по какой-либо причине одна из команд UPDATE даст неверный результат, фонды затеряются в киберпространстве и в бухгалтерских книгах банка не удастся свести баланс. Выражения управления транзакциями предотвращают такую возможность, связывая две операции в одну транзакцию и требуя, чтобы обе команды UPDATE записывались в базу данных одновременно или же чтобы их эффект обнулялся для возврата базы данных в состояние, предшествовавшее выполнению транзакции.



Рис. 2.14. Пример структуры транзакции

Принятие или отклонение изменений

При выполнении DDL-выражения, например, использовании `CREATE TABLE` для добавления новой таблицы в базу данных, Oracle сразу изменяет базу данных, что видно любому пользователю, обращающемуся к таблице. Для DML-выражений (`INSERT`, `UPDATE` и `DELETE`) это не так, если вы не “включите” специальную переменную SQL*Plus `AUTOCOMMIT`. При “включении” `AUTOCOMMIT` (`SET AUTOCOMMIT ON`) любое изменение, вызываемое DML-выражением, записывается в базу данных сразу же после выполнения.

Чтобы записать результаты одного или нескольких выражений SQL, образующих ряд выражений транзакции, вы фиксируете транзакцию, выполняя выражением `COMMIT`. Чтобы обратить результат одного или нескольких выражений SQL, образующих транзакцию, вы выполняете откат с помощью команды `ROLLBACK`. Любые изменения базы данных, выполненные до команды `ROLLBACK` (от последнего выражения `COMMIT`), отбрасываются. Когда вы отключаетесь от базы данных, Oracle автоматически фиксирует все изменения базы данных, внесенные пользователем. На рис. 2.14 показана одна транзакция, размещенная между командами `COMMIT`.

Напомним, что на предыдущем этапе вы ввели нового агента (Мелинду Уитмо), обновили ее номер мобильного телефона и ввели команду `COMMIT`, чтобы сделать постоянной добавленную и обновленную запись об агенте. После этого вы удалили запись Мелинды из таблицы `Agents`. Если вы не выходили из системы Oracle после выполнения команды `DELETE`, то можете восстановить запись Мелинды. Если же вы вышли из системы после удаления записи, восстановление невозможно, поскольку Oracle выполняет “тихую” команду фиксации транзакций. Ниже описаны действия, необходимые для отмены операции `DELETE` и восстановления записи Мелинды Уитмо в таблице `Agents`.

Итак, чтобы отменить все операции, выполненные после последней команды `COMMIT`, выполните следующие действия.

1. Введите ROLLBACK, затем ; (точка с запятой) и нажмите клавишу <Enter>. Oracle отобразит сообщение “Rollback complete”.
2. В командной строке SQL> введите следующий текст:


```
SELECT LastName, CellPhone
FROM Agents
ORDER BY LastName DESC;
```
3. После этого нажмите клавишу <Enter>, чтобы проверить, что запись Мелинды Уитмо восстановлена. Обратите внимание на то, что ее имя снова появилось в списке агентов (второе сверху) вместе и обновленным номером мобильного телефона. Следовательно, вы отменили команду DELETE — последнюю транзакцию после предыдущего выражения COMMIT.

Поскольку запись Мелинды в таблице Agents нам не нужна, введите представленные ниже команды, чтобы навсегда удалить ее запись из базы данных.

Для того чтобы безвозвратно удалить запись, выполните следующие действия.

1. В командной строке SQL> введите команду.


```
DELETE FROM Agents
WHERE AgentID = 10001;
```
2. Чтобы удаление было безвозвратным, введите команду


```
COMMIT;
```

Oracle ответит сообщением “Commit complete”.
3. Введите Exit и нажмите клавишу <Enter>, чтобы выйти из Oracle и закрыть SQL*Plus.

Установка точек отката

В любом месте транзакции можно установить точку сохранения, которая позволит отменить любые изменения, внесенные после нее. Точки сохранения широко используются в очень длинных транзакциях для минимизации числа команд, которые нужно отменять при сбое транзакции. Для создания точек сохранения применяется команда SAVEPOINT, которая имеет следующий синтаксис:

```
SAVEPOINT savepoint_name
```

В данном выражении savepoint_name — любое значимое имя. Например, можно объявить точку сохранения, если ввести команду

```
SAVEPOINT AfterFinancialUpdates;
```

Если вы выполняете DML-выражения до и после команды SAVEPOINT, состояние базы данных, существовавшее в точке AfterFinancialUpdates, можно восстановить с помощью выражения

```
ROLLBACK TO SAVEPOINT AfterFinancialUpdates;
```

DCL-выражения

В многопользовательской базе данных Oracle доступ пользователей к системе необходимо контролировать. Любое обращение к базе данных можно ограничить — пользователю можно разрешать и запрещать доступ к различным объектам базы данных (таблицы, представления и отчеты). Для обеспечения безопасности базы данных применяются DCL-выражения (Data Control Language — язык управления данными), в которые входят команды SQL, разрешающие и запрещающие доступ к системе и содержащимся в ней объектам. Когда вы создаете таблицы и другие объекты в своей схеме (определяемой именем пользователя), другие пользователи не могут обращаться к этим объектам, если вы явно не разрешите им этого. Кроме того, новые пользователи вообще могут разве что войти в базу данных, если администратор базы данных — самый надежный пользователь системы — не предоставит им привилегии на использование систем и объектов. Для предоставления привилегий применяется команда GRANT, для удаления — команда REVOKE. Хотя вы слабо можете повлиять на привилегии, присвоенные вам, вы можете предоставить другим пользователям привилегии на использование созданных объектов. Соответственно, вы можете и лишить привилегий на собственные объекты.

Существуют два типа привилегий: касающиеся системы и объектов. *Привилегии системы* позволяют пользователю выполнять в Oracle определенные действия (создавать таблицы, представления и т.д.). *Привилегии объектов* разрешают пользователю совершать определенные действия по отношению к объектам базы данных (например, вставлять данные в таблицу другого пользователя).

Предоставление привилегий

Для предоставления привилегий на пользование системой и объектами применяется команда GRANT, которая имеет следующий синтаксис.

```
GRANT privilege [, privilege [, privilege ] ...] | ALL PRIVILEGES
ON object
TO user [IDENTIFIED BY password] [, user2 [, user3]...] | PUBLIC
[WITH GRANT OPTION]
```

В приведенном выражении вместо всех слов, выделенных курсивом, необходимо подставить соответствующие значения. Предположим, например, что ваше имя пользователя Pine и вы владеете таблицей Agents. Вы желаете, чтобы пользователь Elm мог просматривать объекты вашей таблицы Agents, но при этом вы *не хотите*, чтобы этот пользователь мог изменять или удалять записи из таблицы Agents. Для этого необходимо войти в базу данных под именем Pine и ввести следующую команду SQL:

```
GRANT SELECT ON Pine.Agents TO Elm;
```

Лишение привилегий

Любой пользователь, владеющий объектом, может запретить доступ других пользователей к своим (созданным этим пользователем) объектам базы данных. Используемая для этого команда REVOKE имеет следующий синтаксис:

```
REVOKE privilege[, privilege[, privilege] ...] | ALL PRIVILEGES
ON object
FROM user [, user2 [, user3 ]...] | PUBLIC
```

Чтобы лишить пользователя Elm возможности отображать на экране строки таблицы Agents, владельцем которой является Pine, пользователь Pine должен ввести следующую команду SQL:

```
REVOKE SELECT ON Pine.Agents FROM Elm
```

Существует несколько привилегий объектов, которые вы можете предоставить другим пользователям: SELECT, INSERT, UPDATE, DELETE и EXECUTE.

Администраторы базы данных обладают достаточными привилегиями для предоставления другим пользователям разнообразных привилегий системы: CREATE SESSION, CREATE TABLE, CREATE USER, CREATE VIEW, CREATE SYNONYM и т.д. Проверить, какими привилегиями обладаете вы, можно с помощью простой команды SELECT. Войдите в Oracle, используя SQL*Plus, и выполните описанные ниже действия.

Чтобы проверить, какими привилегиями системы вы обладаете, выполните следующие действия.

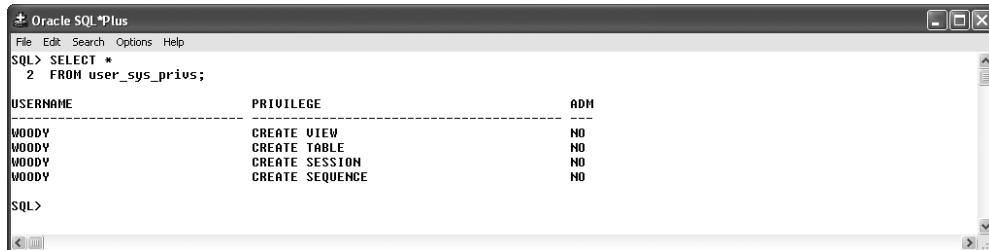
1. Если необходимо, запустите SQL*Plus и войдите в базу данных со своим именем пользователя и паролем.
2. Введите следующую команду (будьте внимательны, не пропустите два символа подчеркивания):

```
SELECT *
FROM user_sys_privs;
```

3. Нажмите клавишу <Enter>, чтобы передать команду базе данных. В ответ Oracle отобразит ваше имя пользователя и перечислит все привилегии, которыми вы обладаете, например, право создавать таблицы.
4. Введите exit и нажмите клавишу <Enter>, чтобы выйти из базы данных Oracle и закрыть SQL*Plus.

В первом столбце возвращаемой таблицы указано ваше имя пользователя, затем идет столбец привилегий (каждая привилегия приведена в отдельной строке). Кроме того, указывается, можете ли вы предоставлять привилегии другим пользователям. На рис. 2.15 показан список системных привилегий, предоставленных пользователю woody. Разумеется, в вашем случае список может быть другим.

Подробно (с примерами) предоставление и лишение привилегий будут рассмотрены в главе 12 “Поддержание безопасности базы данных”.



USERNAME	PRIVILEGE	ADM
WOODY	CREATE VIEW	NO
WOODY	CREATE TABLE	NO
WOODY	CREATE SESSION	NO
WOODY	CREATE SEQUENCE	NO

Рис. 2.15. Список системных привилегий

Использование расширений команд SQL*Plus

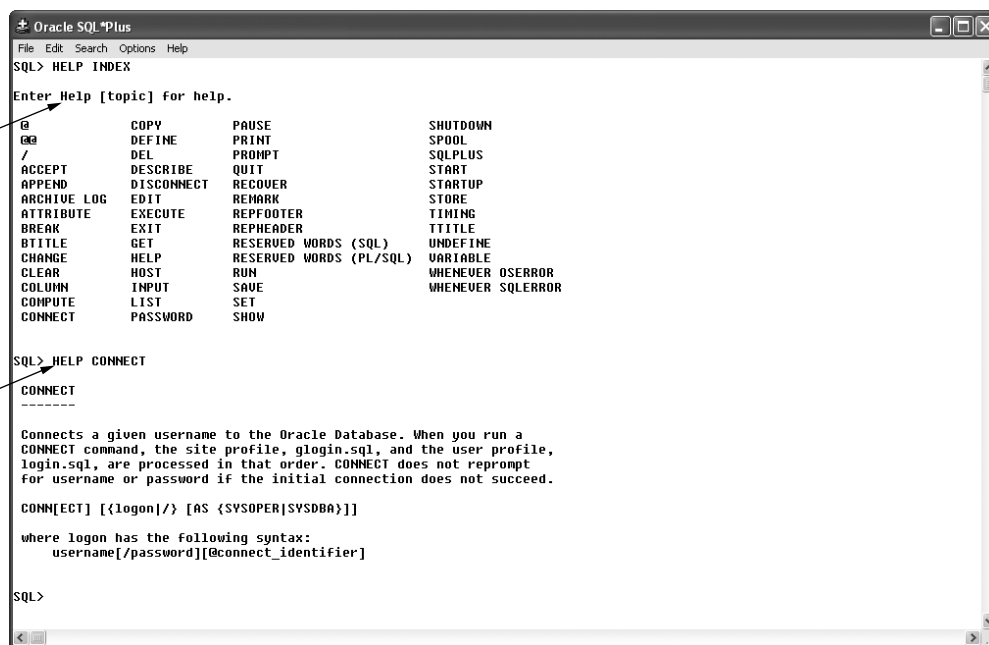
Клиент SQL*Plus от Oracle предлагает несколько *команд SQL*Plus*, которые отличаются от стандартных команд SQL и позволяющих форматировать получаемые столбцы, копировать возвращаемые результаты в буфер, запускать внешние файлы сценариев, отображать структуру таблицы и определения типов данных, менять пароль и соединяться с системой баз данных — всего предлагается около 80 команд. Некоторые из этих команд вы уже использовали при изучении материала данной главы (например, `CLEAR SCREEN`). Все они определены только в среде SQL*Plus (но не обязательно в среде *iSQL*Plus*) и только для Oracle. Одной из команд SQL*Plus, позволяющих экономить время, является команда `START`, которую вы уже использовали ранее для запуска файла сценария, содержащего команды SQL.

Мы не будем описывать все отдельные команды SQL*Plus, а рассмотрим их по категориям или выполняемым функциям. Как вы могли заметить, команды SQL*Plus не заканчиваются точкой с запятой, кроме того, многие команды SQL*Plus допускают сокращенную форму записи (`DESCRIBE` или `DESC`), а для команд SQL такое невозможно. Наконец, некоторые команды SQL*Plus не поддерживаются в *iSQL*Plus*. Следовательно, все примеры, приведенные в конце главы, предполагают использование клиентской программы SQL*Plus.

Примечание. Вы уже знаете, что в SQL*Plus необходимо нажимать клавишу `<Enter>` в конце каждой строки, чтобы перейти на следующую строку или выполнить команду SQL, завершающуюся точкой с запятой. Начиная с этого момента мы часто будем опускать такие подробности, надеясь, что вы их не забудете, т.е. мы будем говорить “введите команду” или “выполните команду”, не уточняя “нажмите клавишу `<Enter>`”.

Для того чтобы вывести на экран список всех команд SQL*Plus, выполните следующие действия.

1. Если необходимо, запустите SQL*Plus и войдите в базу данных, используя свое имя пользователя и пароль.



Отображает помощь по указанной команде SQL*Plus

Перечисляет команды SQL*Plus

Рис. 2.16. Перечень команд SQL*Plus и справка о команде CONNECT

2. Введите `CLEAR SCREEN`, чтобы очистить экран.
3. Введите команду `HELP INDEX` (набрать ее можно и строчными буквами) без точки с запятой, чтобы отобразить список из более чем 50 команд SQL*Plus.
4. Введите `HELP CONNECT` (рис. 2.16).

После ввода выражения `HELP CONNECT` SQL*Plus отобразит информацию об использовании и синтаксисе команды SQL*Plus. Если вы затребуете помощь о команде SQL, например `CREATE TABLE`, SQL*Plus вернет сообщение об ошибке, указывающее, что найти помощь по данной теме не удастся. Помните, что ответы на все вопросы, связанные с SQL, можно найти в сети OTN (Oracle Technical Network). Для решения вопросов, связанных с SQL*Plus, используйте команду SQL*Plus `HELP`.

Описание структуры таблицы

Команда SQL*Plus `DESCRIBE` отображает определения столбцов таблицы, представления или синонима. Кроме того, она приводит спецификацию функции или процедуры (подробнее об этом — в следующих главах). Если вы не можете вспомнить названия определенных столбцов таблицы — выведите на экран список всех их.

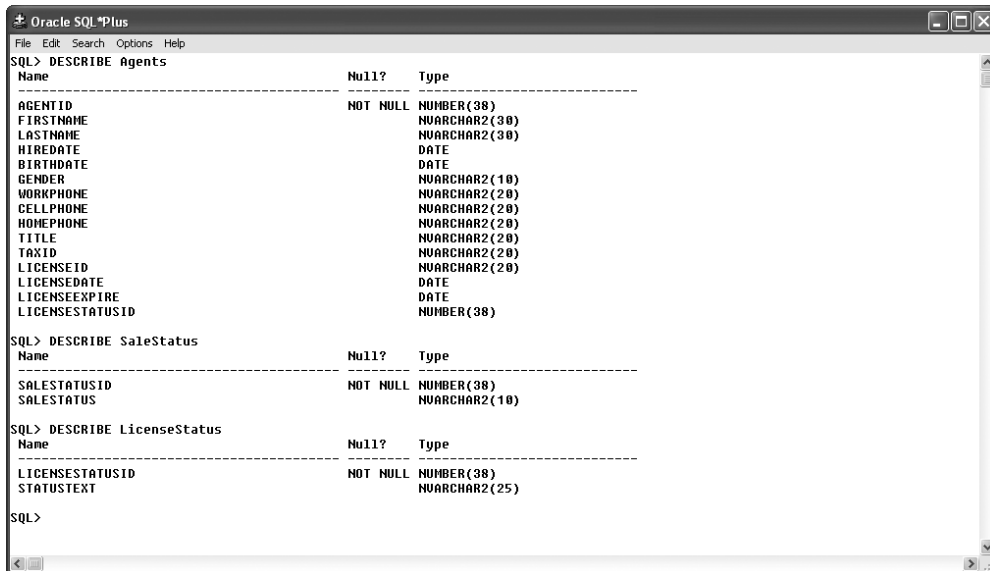


Рис. 2.17. Отображение определений столбцов таблицы

Для того чтобы отобразить названия и типы данных столбцов, выполните следующие действия.

1. В командной строке SQL*Plus выполните команду
`DESCRIBE Agents`
 Oracle отобразит имена и типы данных 15 столбцов таблицы Agents.
2. Введите `DESCRIBE SaleStatus`.
3. Введите `DESCRIBE LicenseStatus` (рис. 2.17). не закрывайте SQL*Plus.

Захват файлов для печати

Иногда требуется записать результаты сеансов SQL*Plus (команды SQL и SQL*Plus, а также результаты их выполнения). Записав команды, выполняемые Oracle, подтверждающие сообщения и результаты запросов в файл, вы можете впоследствии его напечатать. Хотя для решения подобной задачи можно использовать средства Windows (например, нажать клавишу <Print Screen>, чтобы занести в буфер находящееся в текущий момент на экране содержимое окна SQL*Plus), они неудобны, если выход Oracle превышает одну страницу. В связи с этим рекомендуется применять команду SQL*Plus `SPOOL`, позволяющую захватывать в файл весь текст, появляющийся в окне SQL*Plus (как вводимые команды, так и результаты, возвращаемые базой данных).

В *iSQL*Plus* команда `SPOOL` отсутствует. (Если вы попытаетесь выполнить `SPOOL` в *iSQL*Plus*, то получите соответствующее сообщение об ошибке.) `SPOOL` — это “би-стабильная система”, т.е. вы включаете ее для записи, а затем отключаете. Синтаксис команды вызывается с помощью `HELP SPOOL` и выглядит следующим образом:

```
SPOOL [file_name[.ext]] [CREATE | REPLACE | APPEND | OFF | OUT]
```

Любое слово, представленное прописными буквами (за исключением `OFF` и `OUT`), можно сократить до первых трех букв (например, `SPO` вместо `SPOOL`). Чтобы реализовать буферизацию данных в файл, который впоследствии можно напечатать, после команды `SPOOL` указывается полное имя файла (включая путь). После того как вы указали файл, все, что будет вводиться или отображаться на экране, будет записываться в файл. Так будет продолжаться до тех пор, пока вы либо не выйдете из *SQL*Plus*, либо не введете `SPOOL OFF`. Попрактиковаться в использовании данной программы вы сможете в следующем разделе, где рассматриваются команды форматирования.

Команды форматирования

Для форматирования заголовков столбцов и данных, содержащихся в этих столбцах, применяется команда *SQL*Plus* `COLUMN`. Полный ее синтаксис достаточно длинный, поэтому мы приведем только сокращенную версию:

```
COLUMN [{column_name | expression}] [option...]
```

Одними из наиболее важных опций являются `CLEAR`, `FORMAT`, `HEADING`, `JUSTIFY` и `WRAP`. Команда `COLUMN` задает атрибуты отображения столбца, включая текст заголовка, выравнивание столбца, формат числовых данных, а также возможность перехода данных на новую строку, если это требуется. Например, приведенная ниже команда *SQL*Plus* устанавливает характеристики отображения столбца `AgentID` до конца сеанса *SQL*Plus*.

```
COLUMN AgentID FORMAT 99999 HEADING 'Agent ID'
```

В частности, на экране отображается только пять цифр идентификатора агента, а заголовок изменен с принятого по умолчанию (название столбца таблицы) на “Agent ID”. С помощью нескольких строк *SQL*Plus* вы можете установить любое количество названий столбцов и задать множество настроек. Чтобы удалить форматирование, установленное в ходе текущего сеанса *SQL*Plus*, применяется команда `CLEAR COLUMNS`. При ее выполнении удаляется все форматирование столбцов. Отметим, что команда `COLUMN` особенно полезна в часто используемом файле сценария. Разберем подробнее команды *SQL*Plus* `SPOOL` и `COLUMN`.

Для того чтобы скопировать выход окна *SQL*Plus* в файл и отформатировать столбцы, выполните следующие действия.

1. При открытом окне *SQL*Plus* запустите программу Блокнот.
2. В Блокноте введите следующие команды *SQL*Plus* (не нажимайте клавишу <Enter> в конце последней строки):

134 Глава 2. Обзор SQL и SQL*Plus

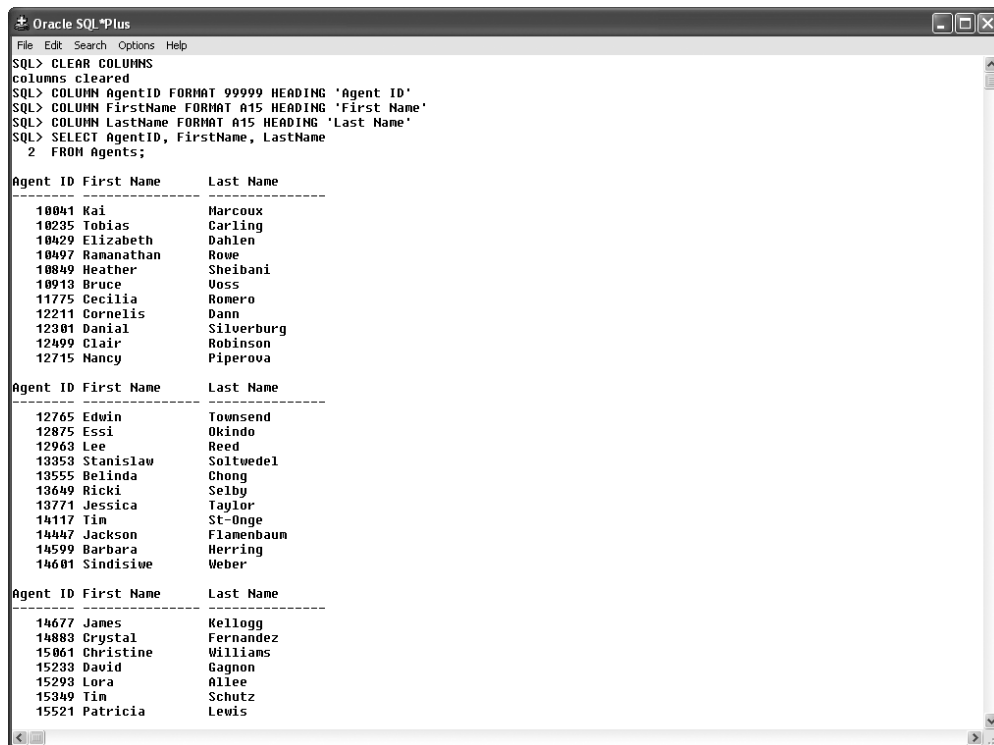


Рис. 2.18. Изменение заголовков столбцов

```

CLEAR SCREEN
CLEAR COLUMNS
COLUMN AgentID FORMAT 99999 HEADING 'Agent ID'
COLUMN FirstName FORMAT A15 HEADING 'First Name'
COLUMN LastName FORMAT A15 HEADING 'Last Name'
SELECT AgentID, FirstName, LastName
FROM Agents;
  
```

3. Запишите данный фрагмент в файле Блокнота Ch2<имя>FormatColumns.sql в удобной для вас папке.
4. Скопируйте весь текст из Блокнота в SQL*Plus. SQL*Plus немедленно интерпретирует команду COLUMN, однако не будет выполнять выражение SQL SELECT (если вы не нажали клавишу <Enter> после точки с запятой в файле программы Блокнот).
5. Нажмите клавишу <Enter>, чтобы выполнить команду SELECT и отформатировать столбцы заданным образом (рис. 2.18).

Модель форматирования A15, указанная после модификатора FORMAT, означает, что столбец является буквенно-цифровым, и ограничивает отображаемый в нем текст

15 позициями. Более длинные записи усекаются справа. Маска формата 99999 означает, что столбец является числовым, а SQL*Plus будет отображать до пяти цифр. Если изменить маску формата, например на 999, в столбце появится запись #####, указывающая, что значения в столбце шире, чем позволяет маска формата. В таком случае увеличьте количество цифр-заполнителей. Далее предположим, что мы задали следующее условие:

```
COLUMN AgentID FORMAT 99,999 HEADING 'Agent ID'
```

Тогда идентификационный номер агента будет идти с запятой между третьей и четвертой значащей цифрой (например, 10,041).

Обратите внимание на то, что заголовки столбцов отличаются от своего обычного формата. Каждый заголовок содержит два слова, написанных не прописными буквами, как принято по умолчанию. Скорее всего, вы увидите, что под заголовками идет всего 11 строк с информацией об агентах. Добавьте к этому две строки заголовка и пустую строку, предшествующую заголовку, и вы получите 14 строк — значение по умолчанию переменной среды SQL*Plus PAGESIZE. Переменная PAGESIZE определяет число строк, отображаемых между верхом заглавия и низом страницы. Вы можете изменить ее значение, добавив в файл сценария выражение SQL*Plus SET PAGESIZE <value>, где <value> — число строк на странице. Максимальное значение PAGESIZE равно 50 000.

Для того чтобы модифицировать файл сценария SQL, выполните следующие действия.

1. Откройте файл сценария FormatColumns в программе Блокнот, щелкните слева от команды CLEAR COLUMNS и нажмите клавишу <Enter>, чтобы создать пустую строку. В этой строке между CLEAR SCREEN и CLEAR COLUMNS введите команду
2. Выберите пункт File (Файл) и Save (Сохранить), чтобы сохранить модифицированный файл SQL*Plus.
3. Скопируйте все строки из программы Блокнот в окно SQL*Plus.
4. В окне SQL*Plus нажмите клавишу <Enter>, чтобы выполнить команду SELECT (рис. 2.19).
5. В SQL*Plus введите следующую команду, чтобы восстановить исходное значение переменной среды PAGESIZE и очистить экран:

```
SET PAGESIZE 14
CLEAR SCREEN
```

6. Закройте окно Блокнота, но не закрывайте SQL*Plus.

Существуют и другие команды форматирования SQL*Plus, с которыми вы еще встретитесь в книге. Подробнее они будут рассмотрены по мере появления.

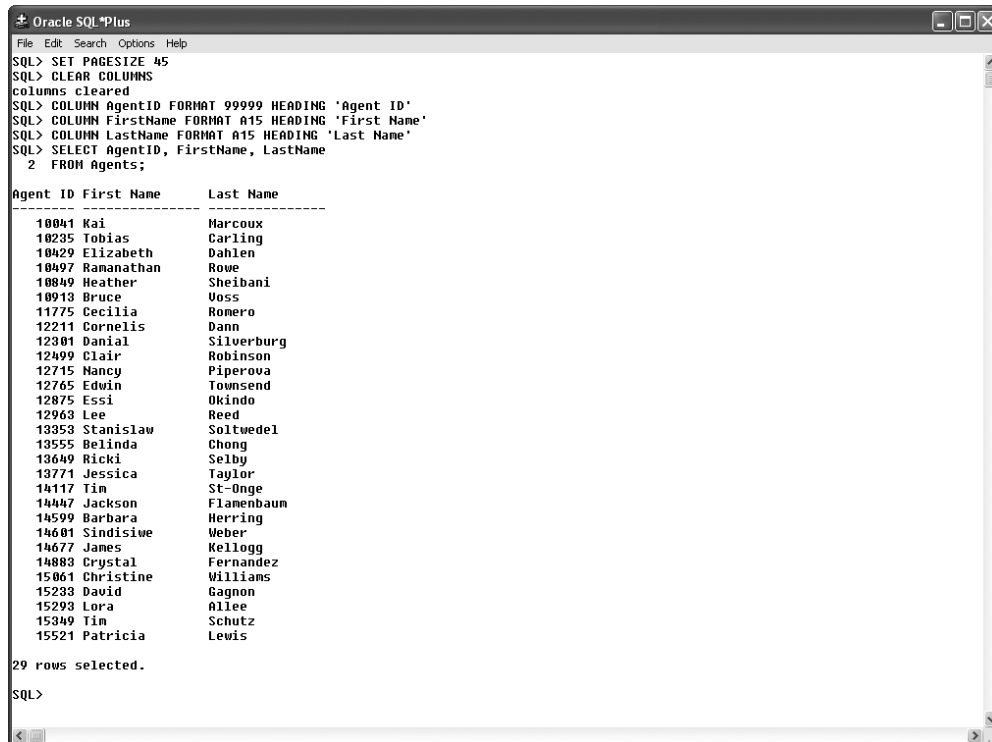


Рис. 2.19. Изменения значения переменной среды PAGESIZE

Настройка среды SQL*Plus

Некоторые переменные среды SQL*Plus (например, PAGESIZE) можно установить навсегда, чтобы их не приходилось задавать при каждом входе в Oracle через SQL*Plus. Например, вам удобно использовать значение PAGESIZE, равное 50, а значение LINE-SIZE (число символов в строке), равное 75. Для настройки среды SQL*Plus применяются команды SQL*Plus из строки меню.

Итак, чтобы модифицировать любую переменную среды SQL*Plus, выполните следующие действия.

1. В окне SQL*Plus (не iSQL*Plus) выберите Options в строке меню и позицию Environment. SQL*Plus откроет диалоговое окно Environment.
2. Прокрутите список Set Options вниз, пока не увидите надпись pagesize. Активируйте ее (щелкните на ней) и установите переключатель Custom на панели Value. Обратите внимание на то, что в текстовом окне появилось значение 14, соответствующее текущему значению размера страницы (вы восстановили это значение, выполняя предыдущие шаги).

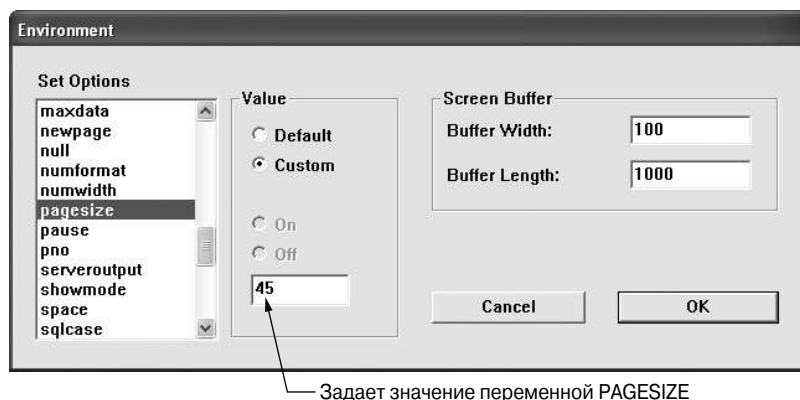


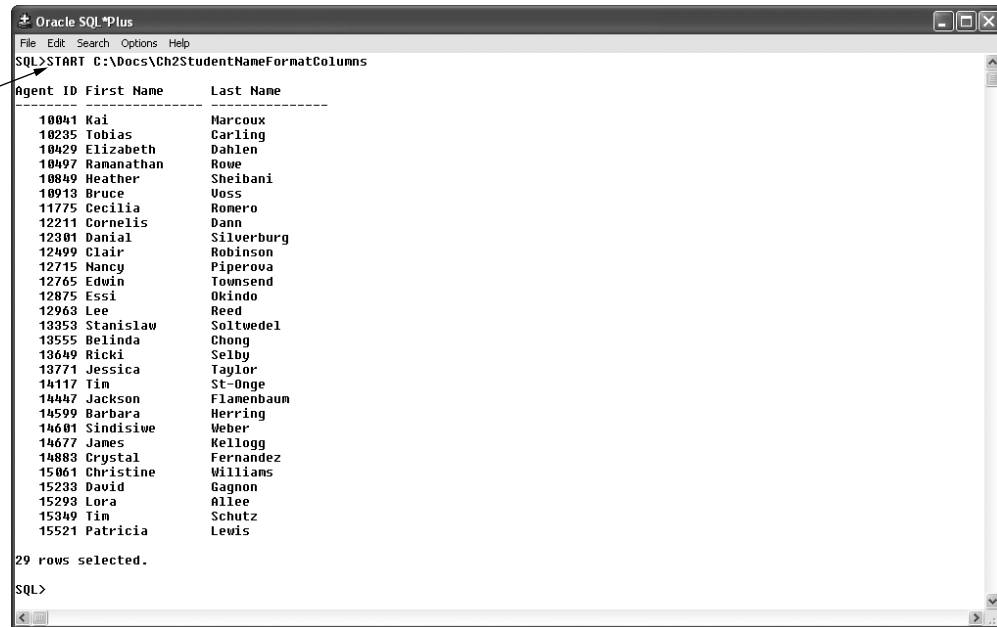
Рис. 2.20. Настройка среды SQL*Plus

3. Дважды щелкните на значении 14, чтобы выбрать всю строку, и введите число 45 (рис. 2.20).
4. Прокрутите окно просмотра вниз и выберите в нем надпись sqlprompt. Щелкните на кнопке Custom. Обратите внимание на то, что в окне в группе Value появилось принятое по умолчанию приглашение командной строки SQL>. Пока что изменять его не будем.
5. Щелкните на кнопке OK, чтобы подтвердить внесенные изменения в размер строки. Диалоговое окно Environment закроется.
6. В командной строке SQL*Plus введите SHOW PAGESIZE и нажмите клавишу <Enter>, чтобы отобразить на экране текущее значение переменной PAGESIZE.
7. Чему теперь равно значение переменной LINESIZE? Чтобы узнать это, введите SHOW LINESIZE и нажмите клавишу <Enter>.

Вообще, можно настроить значение любой переменной SQL*Plus, имя которой отображается в списке Set Options диалогового окна Environment. Кстати, если вам нужен длинный список значений всех переменных среды, введите команду SHOW ALL.

Выполнение файлов сценария

Скоро вы обнаружите, что определенные наборы команды SQL и SQL*Plus приходится запускать довольно часто. Вместо того чтобы постоянно набирать десятки строк кода, можно записать команды SQL*Plus в файле, чтобы их можно было запустить все сразу (почти так же, как вы поступали, когда копировали строки из программы Блокнот в SQL*Plus, а затем запускали весь код). Набор выражений SQL и SQL*Plus в файле называется *сценарием*. Записав сценарий, вы можете открыть весь файл, не открывая программу Блокнот и не занимаясь копированием. Все нужные действия представляются двумя командами SQL*Plus, которые отвечают за загрузку и запуск



Команда START (в вашем случае не видна)

Рис. 2.21. Запуск файла сценария в SQL*Plus

файлов сценария: START и @. Их выполнение даст идентичный результат: загрузку файла сценария и выполнение всех его строк.

Для того чтобы выполнить записанный на диске файл сценария, выполните следующие действия.

1. Найдите папку, содержащую записанный ранее файл сценария Ch2<имя> FormatColumns.sql, и запомните путь к нему.
2. В командной строке SQL*Plus введите следующий код:

```
START <путь>\Ch2<имя>FormatColumns
```

Здесь <путь> — полный путь с указанием диска и папки, в которой записан файл сценария; <имя> — имя, подставленное вами ранее. Например:

```
START C:\My Documents\Ch2ElmFormatColumns
```

3. Нажмите клавишу <Enter>, чтобы выполнить команду START. SQL*Plus загрузит файл сценария и выполнит содержащиеся в нем команды. В результате на экране SQL*Plus отобразится 3 столбца и 29 идентификаторов и имен сотрудников агентства недвижимости — все под одним заголовком (рис. 2.21).

Примечание. Мы удалили первую строку, CLEAR SCREEN, чтобы вы могли увидеть команду START, запускающую файл сценария. В вашем случае ее видно не будет.

Если хотите, вместо слова `START` можете подставить знак `@` — результат от этого не изменится.

Использование переменных

SQL*Plus позволяет определять переменные, которые можно использовать вместо реальных значений в выражениях SQL. Такие переменные, именуемые *подстановочными*, применяются для подстановки значений в выражения SQL. Особенно они полезны в файлах сценариев SQL. Понять, что является переменной, довольно легко — это обычные текстовые строки, которые выглядят так, как названия столбцов, но начинаются с амперсанта (&). Таким образом, предположим, что SQL*Plus выполняет следующее выражение `SELECT`:

```
SELECT AgentID, FirstName, LastName
FROM Agents
WHERE LastName = '&lastnameplease';
```

Здесь SQL*Plus требует, чтобы пользователь предоставил значение “lastnameplease” (ввел фамилию) перед отправкой запроса в Oracle. Используя переменные, очень удобно создавать обобщенные файлы сценариев. Рассмотрим предыдущий пример: кто угодно может затребовать информацию об агенте, указав “в процессе” его фамилию. Рассмотрим на реальном примере, как это действует.

Для того чтобы использовать подстановочную переменную в файле сценария, выполните следующие действия.

1. Переключитесь на SQL*Plus и введите `CLEAR SCREEN` (нажмите клавишу <Enter>), чтобы очистить рабочую поверхность.
2. Откройте программу Блокнот, наберите следующий сценарий SQL, а затем запишите его как текстовый файл с именем `Ch2<имя>SubVar.sql`:

```
SELECT AgentID, FirstName, LastName, BirthDate
FROM Agents
WHERE LastName = '&lastnameplease';
```

3. Вернитесь в SQL*Plus и наберите следующую строку.

```
@<путь>\Ch2<имя>SubVar
```

Здесь <путь> — полный путь с указанием диска и папки, в которой записан файл сценария; <имя> — имя, подставленное вами ранее.

4. Нажмите клавишу <Enter>, чтобы запустить файл сценария. После этого SQL*Plus запросит фамилию.
5. Наберите `Carling` (точно так, как написано) и нажмите клавишу <Enter>. Oracle отобразит избранные столбцы таблицы `Agent` для агента `Carling`.
6. Повторите шаги 3 и 4.
7. После запроса “<Enter> value for lastnameplease” введите `Fernandez`

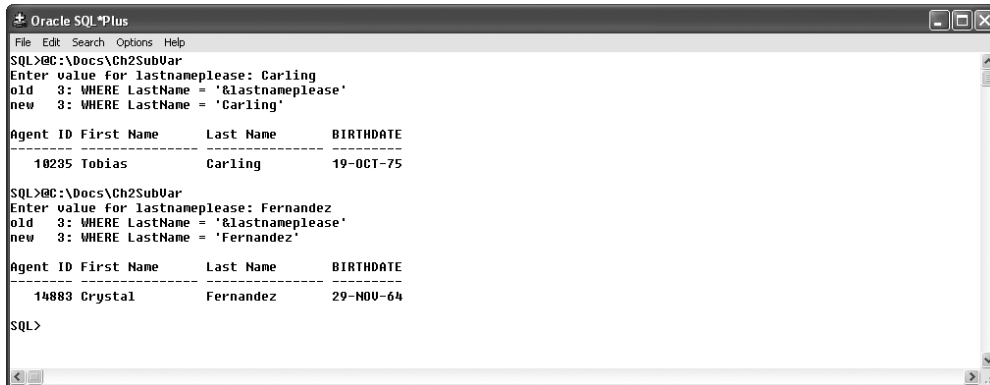


Рис. 2.22. Использование подстановочной переменной в файле сценария

и нажмите клавишу <Enter>. SQL*Plus подставит набранную вами строку вместо переменной и передаст весь запрос Oracle. В ответ Oracle вернет другую строку (рис. 2.22).

8. Выйдите из SQL*Plus: введите Exit и нажмите клавишу <Enter>.
9. Закройте программу Блокнот.

Обзор других команд SQL*Plus

В настоящей книге вам встретятся еще несколько команд SQL*Plus, по которым иногда требуется интерактивная помощь (справка). Помните, что описание и синтаксис команд SQL*Plus можно получить в SQL*Plus, если ввести HELP, а затем имя команды. Вы можете вывести на экран список всех команд SQL*Plus, набрав HELP INDEX. Ниже мы кратко опишем эти команды, не вдаваясь в особые подробности (помимо того, что данные команды полезны, их очень легко понять и использовать).

Команда HOST, недоступная в iSQL*Plus, выполняет команды операционной системы, не покидая SQL*Plus. Вместо HOST в командной строке SQL> вы также можете использовать \$. Обе команды вызывают операционную систему. Если речь идет о Windows, то откроется окно DOS и появится запрос командной строки операционной системы с указанием текущего пути. Чтобы вернуться в SQL*Plus, введите Exit и нажмите клавишу <Enter>. Например, если вам требуется перечислить файлы SQL в текущей папке, вы можете ввести HOST, а затем в командной строке DOS ввести DIR *.SQL. Более того, если вы хотите автоматизировать команду DOS, используйте конструкцию “HOST+команда”. Рассмотрим, например, выполнение следующей команды в SQL*Plus:

```
HOST DEL C:\Docs\myfile.sql
```

SQL*Plus вызывает операционную систему компьютера и указывает ей удалить (DEL) файл myfile.sql из папки C:\Docs\, после чего закрыть окно DOS, практи-

чески сразу вернувшись в SQL*Plus. Ниже мы используем эту возможность в файле сценария. Разумеется, в командной строке операционной системы вы можете выполнять различные команды, включая изменение текущей папки.

Переменные среды `TERMOUT` и `PAGESIZE` также весьма полезны в файле сценария, когда требуется выполнить ряд команд SQL и SQL*Plus. `TERMOUT` применяется, когда на экране не нужны информационные сообщения, подобные “1 row created”. Если ввести команду `SET TERMOUT OFF` (или `SET TERM OFF`), это подавит информацию, выводимую на экран из данного файла сценария (и только из него). Если ввести команду `SET TERM ON`, окно вернется в обычный режим отображения. Как только выполнение сценария, содержащего команду `TERMOUT`, завершается, `TERMOUT` включается автоматически. Другой переменной среды SQL*Plus является `FEEDBACK`, определяющая, будут ли отображаться на экране *информативные* сообщения — “1 row created” или “table dropped”. После ввода команды `SET FEEDBACK OFF` информативные сообщения подавляются, после ввода `SET FEEDBACK ON` — снова активизируются. Если отображение сообщений не включить явно, состояние `FEEDBACK` остается равным `OFF` до конца сеанса SQL*Plus, независимо от того, где оно было отключено (в файле сценария или в командной строке). Ввод `SET PAGESIZE 0` подавляет отображение заголовков столбцов, что бывает полезно, когда результаты запроса выводятся в файл без заглавий столбцов.

В файлы сценариев вы можете включать комментарии. Если комментарий занимает одну строку, он начинается с двух дефисов (`--`). Комментарии в несколько строк начинаются с косой черты и звездочки (`/*`), а заканчиваются звездочкой и косой чертой (`*/`). Кроме того, в начале однострочного комментария можно использовать команду `REM`. По возможности, старайтесь избегать варианта `/* текст */`, поскольку вы можете забыть закрыть комментарий, и тогда остаток сценария будет считаться комментарием.

Команда `PROMPT` выводит на экран сообщение или пустую строку. Особенно она полезна в файле сценария, когда требуется отобразить сообщение о ходе работы, уведомляющее о состоянии файла сценария. Например, после выполнения приведенной ниже команды на экране появится текст, который следует после команды `PROMT`.

```
PROMPT Successfully ran script to delete files.
```

Создание и запуск сценария

Как говорилось ранее, файл сценария — это набор команд SQL и SQL*Plus, применяющихся для выполнения определенной задачи. Напомним, что для запуска сценария необходимо ввести команду `START`, а после нее — путь и имя файла сценария. После этого SQL*Plus загрузит указанный файл и последовательно выполнит все его строки. Для логического завершения данной главы полезно собрать некоторые изученные в ней команды в файл сценария. Кроме того, вы сможете вызвать несколько системных

142 Глава 2. Обзор SQL и SQL*Plus

таблиц, поддерживаемых базой данных, которая содержит имена всех объектов, принадлежащих схеме (пользователю). Вообще, Oracle поддерживает список объектов, принадлежащих текущему пользователю в представлении `user_tables`. Указанная таблица является “таблицей таблиц”, поскольку содержит более 45 столбцов, описывающих характеристики принадлежащих вам таблиц, включая число строк в каждой таблице, процент свободного места и имена таблиц. (Чтобы отобразить на экране список столбцов данной системной таблицы, введите `DESCRIBE user_tables`.) Чуть позже мы напишем сценарий, отображающий имена всех принадлежащих вам таблиц, число строк в этих таблицах и описание столбцов в каждой таблице, — все это из одного файла сценария, который вы сможете периодически запускать по мере изменения базы данных.

Далее вы напишете сценарий, описывающий имена принадлежащих вам (вашей схеме) файлов. Файл сценария будет обращаться к системной таблице, чтобы перечислить имена таблиц вашей учетной записи. На основе данного списка имен вы динамически создадите другой файл сценария, содержащий набор команд `DESCRIBE` (по одной на каждую таблицу, найденную в вашей учетной записи). Наконец, ваш файл сценария выполнит динамически созданный файл сценария, а затем удалит его после выполнения. В конце вы получите отображенный в окне SQL*Plus список имен таблиц и список столбцов, содержащихся в каждой таблице, с указанием типа данных, соотнесенных с каждым столбцом. Не волнуйтесь, если не все строки кода будут вам понятны. В данном случае наша цель — показать, как создавать файлы сценариев, которые затем передаются SQL*Plus для выполнения.

Итак, для того чтобы создать файл сценария и сохранить его, выполните следующие действия.

1. Прежде всего определите, куда вы будете записывать файл сценария. Например, вы можете выделить для этого папку `C:\Temp` или использовать корневую папку `C:\`.
2. Откройте программу Блокнот и введите показанный ниже код. Не забудьте подставить вместо `C:\Docs\` точный путь к папке, получающей файл сценария (это нужно сделать в трех местах; все они указаны на рис. 2.23):

```
REM Describe my tables
CLEAR SCREEN
--Создать и выполнить файл
SET FEEDBACK OFF
SET PAGESIZE 0
SPOOL C:\Docs\Describe.sql
SELECT 'DESCRIBE' || table_name || ';'
FROM user_tables
WHERE table_name NOT LIKE 'BIN%'
ORDER BY 1;
SPOOL OFF
PROMPT
@@C:\Docs\Describe.sql
```

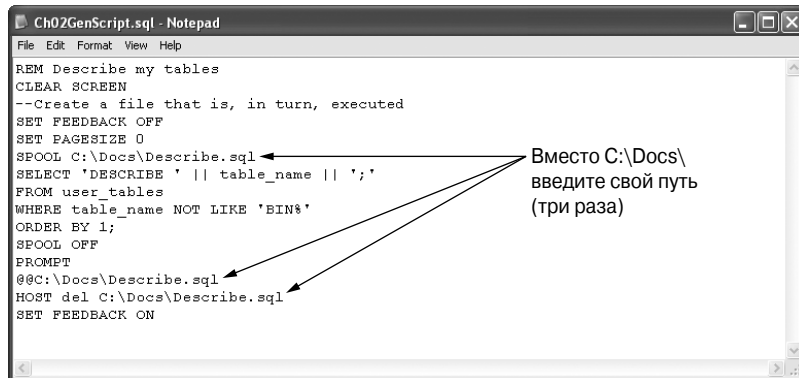



Рис. 2.23. Файл сценария

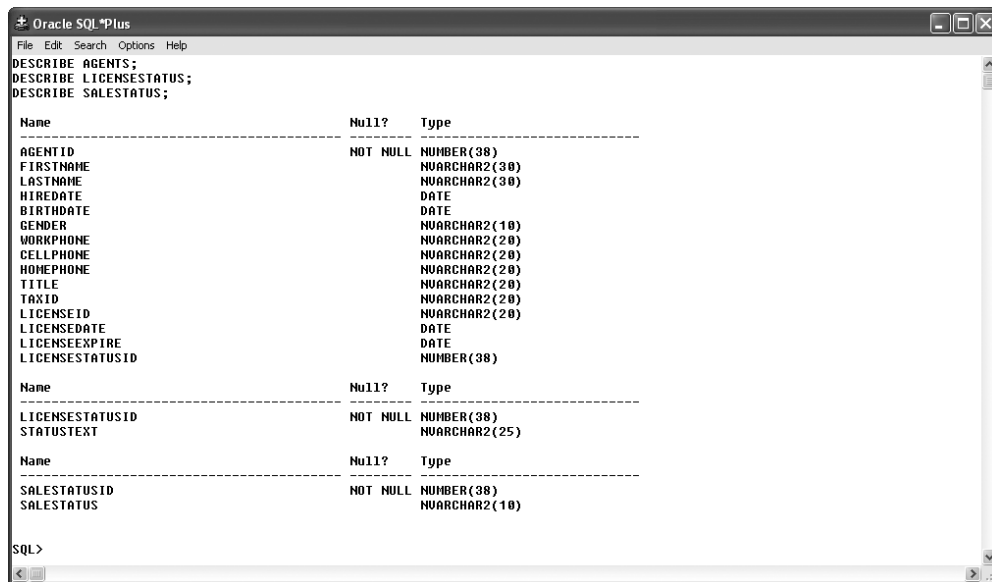
```

HOST del C:\Docs\Describe.sql
SET PAGESIZE 14
SET FEEDBACK ON
    
```

3. Выберите пункт File (Файл), а затем Save (Сохранить). Щелкните в текстовом окне Save as type (Тип файла), выберите All Files (Все файлы), перейдите в папку, выбранную для сохранения файла сценария. Введите Ch2<имя>GenScript.sql (вместо <имя> введите свое имя) в окне File name (Имя файла) и щелкните на кнопке Save (Сохранить). Не закрывайте Блокнот на случай, если SQL*Plus выдаст ошибку и сценарий придется править.
4. Откройте SQL*Plus и войдите в систему, используя свое имя пользователя и пароль. В командной строке SQL введите следующий текст:
START <путь>Ch2<имя>GenScript
Вместо <путь> подставьте путь к файлу сценария, а вместо <имя> — свое имя.
5. Нажмите клавишу <Enter>, чтобы запустить файл сценария. На экране вы должны увидеть то же, что показано на рис. 2.24.
6. Если появятся сообщения об ошибках, переключитесь на программу Блокнот, найдите опечатки, еще раз сохраните файл и повторите п. 4.
7. Закройте Блокнот и выйдите из SQL*Plus.

Резюме

Существует пять категорий выражений языка структурированных запросов (Structured Query Language — SQL): запросы, определения данных (DDL), команды манипулирования данными (DML), выражения управления транзакциями (TC) и команды языка управления данными (DCL). Запросы SQL извлекают информацию из базы данных. Выражения DML INSERT, UPDATE и DELETE модифицируют данные. Ко-



The screenshot shows the Oracle SQL*Plus window with the following content:

```

Oracle SQL*Plus
File Edit Search Options Help
DESCRIBE AGENTS;
DESCRIBE LICENSESTATUS;
DESCRIBE SALESTATUS;

Name                               Null?    Type
-----
AGENTID                            NOT NULL NUMBER(38)
FIRSTNAME                          NVARCHAR2(30)
LASTNAME                           NVARCHAR2(30)
HIREDATE                           DATE
BIRTHDATE                          DATE
GENDER                             NVARCHAR2(10)
WORKPHONE                          NVARCHAR2(20)
CELLPHONE                          NVARCHAR2(20)
HOMEPHONE                          NVARCHAR2(20)
TITLE                              NVARCHAR2(20)
TAXID                              NVARCHAR2(20)
LICENSEID                           NVARCHAR2(20)
LICENSEDATE                         DATE
LICENSEEXPIRE                      DATE
LICENSESTATUSID                    NUMBER(38)

Name                               Null?    Type
-----
LICENSESTATUSID                    NOT NULL NUMBER(38)
STATUSTEXT                          NVARCHAR2(25)

Name                               Null?    Type
-----
SALESTATUSID                       NOT NULL NUMBER(38)
SALESTATUS                          NVARCHAR2(10)

SQL>
  
```

Рис. 2.24. Результат выполнения файла сценария

манды языка определения данных определяют в базе данных структуры (например, таблицы). Выражения управления транзакциями отменяют или фиксируют любые изменения, внесенными последними незафиксированными командами DML. Команды языка управления данными обеспечивают безопасность, разрешая (или запрещая) пользователям обращаться к выбранным объектам базы данных. К базе данных “клиент/сервер” Oracle можно обращаться из операционной системы компьютера, используя клиент Windows SQL*Plus, и из Web-интерфейса iSQL*Plus. Оба метода имеют свои достоинства, однако самым распространенным вариантом является доступ с использованием клиентского приложения SQL*Plus.

Расширения стандартного языка SQL компании Oracle называются выражениями SQL*Plus. Данные команды позволяют форматировать выход, захватывать извлекаемые результаты, запускать файлы сценариев, отображать структуру таблицы и настраивать среду интерфейса клиента SQL*Plus. Чтобы отобразить список данных полезных расширений SQL, введите `HELP INDEX` в командной строке SQL. Чтобы получить дополнительную справку по заинтересовавшей вас команде SQL*Plus, введите `HELP` и имя команды. Последовательности команд SQL и SQL*Plus образуют сценарий SQL. Записывая блоки команд в файл, вы можете впоследствии вызывать эти блоки с помощью единственной команды. Файлы сценариев могут содержать переменные, значения которых задает пользователь во время выполнения сценария. Переменные позволяют отложить определение имен объектов и действий до времени выполнения и создать динамический и настраиваемый сценарий, удовлетворяющий требованиям множества пользователей базы данных.

Основные термины

- Базовая таблица
- Фиксировать
- Условие
- Производная таблица
- Модель формата
- Набор символов национального языка
- NULL
- Запрос
- Откат
- Сценарий
- Команды SQL*Plus
- Подстановочная переменная
- Привилегии системы
- Транзакция
- Наблюдаемая таблица

Повторение пройденного материала

Истина или ложь?

1. Язык структурированных запросов (Structured Query Language — SQL) — это язык программирования.
2. Выражения SQL языка манипулирования данными (Data Manipulation Language — DML) позволяют создавать такие объекты, как таблицы и представления.
3. Взаимодействовать с Oracle непосредственно из операционной системы компьютера гораздо легче, чем использовать SQL*Plus.
4. Для удаления записей из таблицы базы данных применяется команда DELETE FROM.
5. Запись, удаленную из таблицы, можно восстановить, если до выхода из системы или фиксации удаления использовать команду ROLLBACK.

Заполнить пропущенное

1. Выполните выражение SQL _____, чтобы добавить записи в таблицу.
2. Выражение DESCRIBE относится к командам _____, а не к командам SQL.
3. Чтобы удалить таблицу из схемы, используйте команду _____.

146 Глава 2. Обзор SQL и SQL*Plus

4. Чтобы сделать постоянными изменения базы данных, вызванные командами INSERT, DELETE и UPDATE, введите команду SQL _____.
5. Команда SQL*Plus _____, введенная с правильным модификатором, позволяет контролировать формат возвращаемых данных в SQL*Plus.

Варианты ответов

1. Какая из приведенных ниже команд позволяет захватывать в файл все, что появляется на экране SQL*Plus?
 - а) PRINT ON.
 - б) SPOOL <filename>.
 - в) SPOOL OFF.
 - г) CAPTURE <filename>.
2. Какая команда позволяет удалить все форматирование столбцов, установленное в течение сеанса SQL*Plus?
 - а) FORMAT COLUMN OFF.
 - б) COLUMN FORMAT CLEAR.
 - в) CLEAR COLUMNS.
 - г) Ни одна из перечисленных.
3. Какая из приведенных команд позволяет добавить строку в таблицу LicenseStatus (предполагается, что для всех столбцов тип данных предлагаемых значений правильный)?
 - а) INSERT INTO LicenseStatus VALUES (876,'Out of commission');
 - б) INSERT LicenseStatus VALUES (876,'Out of commission');
 - в) INSERT FROM LicenseStatus VALUES (876,'Out of commission');
 - г) CREATE FROM LicenseStatus VALUES (876,'Out of commission');
4. Какая из приведенных команд удаляет запись Danial Silverburg из таблицы Agents Redwood Realty?
 - а) DELETE FROM Agents LastName = 'Silverburg' AND FirstName = 'Danial';
 - б) DROPFROMAgentsWHERELastName = 'Silverburg' AND FirstName = 'Danial';
 - в) DELETE FROM Agents WHERE LastName = 'Silverburg' AND FirstName = 'Danial';
 - г) DELETE Agents WHERE LastName = 'Silverburg' AND FirstName = 'Danial';
 - д) Правильные варианты в) и г).
5. С помощью какой команды можно подавить дальнейшее отображение информации на экране информативных сообщений, подобных “3 rows deleted”?
 - а) SET TERMOUT ON.

ТАБЛИЦА 2.1. Данные, которые требуется ввести в таблицу `ContactReason`

<i>ContactReason</i>	<i>Описание</i>
Buy	Offer to buy a property (“Желает купить недвижимость”)
Casual	General customer probably looking for properties (“Обычный клиент, возможно, интересующийся недвижимостью”)
Sell	Listing to sell a property (“Желает продать недвижимость”)

- б) `SET PAGESIZE 0.`
- в) `SET SPOOL OFF.`
- г) `SET INFORM OFF.`
- д) `SET FEEDBACK OFF.`

Упражнения

Redwood Realty

Администратор базы данных Redwood Realty требует создать в базе таблицу `ContactReason`. В этой таблице должны быть представлены три причины контакта людей с агентством: чтобы купить дом; чтобы продать дом или чтобы связаться с брокером для каких-то других целей. Три данные категории должны храниться в таблице вместе с более длинными описаниями. Три строки, которые вы должны вставить в таблицу `ContactReason`, приведены в табл. 2.1.

1. Войдите в Oracle, используя SQL*Plus. Откройте программу Блокнот и сразу сохраните файл как `Ch2<имя>RedwoodEOC.sql`. Не забудьте щелкнуть на `Save as type` (Тип файла) и выбрать пункт `All Files` (Все файлы).
2. В программе Блокнот введите следующий текст, нажимая клавишу `<Enter>` в конце первых двух строк (После завершающей точки с запятой клавишу `<Enter>` не нажимайте):

```
CREATE TABLE ContactReason
(ContactReason NVARCHAR2(15) PRIMARY KEY,
Description NVARCHAR2(50));
```

3. Скопируйте все четыре строки из программы Блокнот в SQL*Plus. Нажмите клавишу `<Enter>`, чтобы выполнить команду `CREATE TABLE`. База данных Oracle должна вывести на экран сообщение `“Table created”`.
4. Если на предыдущем этапе появились сообщения об ошибках, найдите их и исправьте в программе Блокнот, после чего повторите п. 3.
5. Переключитесь на Блокнот, нажмите клавишу `<Enter>`, чтобы перейти на следующую строку, и введите следующие команды `INSERT` (после последней строки также нажмите `<Enter>`):

148 Глава 2. Обзор SQL и SQL*Plus

```
INSERT INTO ContactReason VALUES ('Buy',
'Offer to buy a property');
INSERT INTO ContactReason VALUES ('Casual',
'General customer probably looking for
properties');
INSERT INTO ContactReason VALUES ('Sell',
'Listing to sell a property');
```

6. Выделите в Блокноте три команды Insert, включая третью точку с запятой и пустую строку после нее. Вставьте данный текст в SQL*Plus. Три команды будут выполнены немедленно.
7. Переключитесь на программу Блокнот, запишите файл, выйдите из программы Блокнот и вернитесь в SQL*Plus.
8. Введите COMMIT; (не пропустите в конце точку с запятой) и нажмите клавишу <Enter>, чтобы сделать постоянной вставку трех указанных строк.
9. В SQL*Plus введите и выполните следующую команду INSERT (между буквами n и t в слове Don't вводятся две одинарные кавычки):

```
INSERT INTO ContactReason VALUES
('Don't know', 'Unknown reason');
```
10. В SQL*Plus введите следующую команду и нажмите клавишу <Enter>, чтобы выполнить ее:

```
SELECT *
FROM ContactReason;
```
11. Введите ROLLBACK; (не забудьте точку с запятой) и нажмите клавишу <Enter>, чтобы отменить последнее действие.
12. Чтобы создать текстовый файл, в который можно записывать информацию и который можно сохранить на диске, выполните в SQL*Plus следующие команды (вместо <путь> подставьте полный путь (например, C:\Main\)):

```
SPOOL <путь>\Ch2ContactReasonList.txt
DESCRIBE ContactReason
SELECT * FROM ContactReason;
SPOOL OFF
```
13. Откройте созданный выше файл Ch2ContactReasonList.txt в программе Блокнот, вставьте сверху новую строку, введите свое имя, запишите файл, напечатайте его, а затем закройте программу Блокнот.
14. Переключитесь на SQL*Plus и удалите буферный файл с помощью следующей команды (вместо <путь> введите реальный путь к файлу):

```
HOST del <путь>\Ch2ContactReasonList.txt
```
15. Чтобы удалить таблицу и ее содержимое, введите следующую команду.

```
DROP TABLE ContactReason;
```
16. Выйдите из SQL*Plus.

Coffee Merchant

Администратор базы данных Coffee Merchant требует, чтобы вы создали несколько таблиц и немного изменили представленные в базе данные. Предполагается, что в данном примере вы будете использовать SQL*Plus и iSQL*Plus. Если по какой-либо причине вы не можете задействовать iSQL*Plus (интерфейс базы данных на основе браузера), просто используйте SQL*Plus. Итак, администратор базы данных желает, чтобы вы создали и заполнили таблицу States, которая после выполнения работы будет содержать данные о 50 штатах США и округе Колумбия. Строка этой таблицы включает двухбуквенное сокращение названия штата, полное название штата, ставку налога, примерную численность населения, размер штата в квадратных милях и URL официального Web-сайта штата (если он есть).

1. Запустите SQL*Plus и войдите в Oracle. Откройте программу Блокнот и сразу сохраните файл под именем Ch2<имя>CoffeeEOC.sql. На забудьте щелкнуть на кнопке Save as type (Тип файла), а затем выбрать AllFiles (Все файлы).
2. Введите в программе Блокнот следующий текст, нажимая клавишу <Enter> в конце каждой строки (не забудьте набрать заключительную закрывающую скобку и точку с запятой):

```
CREATE TABLE States
  (StateID NCHAR(2) PRIMARY KEY,
   StateName NVARCHAR2(20) NOT NULL,
   TaxRate NUMBER(7,4),
   Population INTEGER,
   LandArea INTEGER,
   WebURL NVARCHAR2(50)
  );
```

3. Запишите файл в Блокноте, скопируйте в буфер весь текст до завершающей точки с запятой, но без следующей за ним пустой строки. Закройте программу Блокнот.
4. В SQL*Plus нажмите комбинацию клавиш <Ctrl+V>, чтобы вставить текст, и клавишу <Enter> — чтобы выполнить команды сценария. Oracle ответит сообщением “Table created”.
5. В SQL*Plus введите и выполните следующие выражения SQL:

```
INSERT INTO STATES
  VALUES ('AK', 'Alaska', 0, 643786,
          571951, 'www.state.ak.us');
CLEAR SCREEN
COLUMN StateID FORMAT A7
COLUMN StateName FORMAT A15
COLUMN TaxRate FORMAT 0.9999
COLUMN Population FORMAT 99,999,999
COLUMN LandArea FORMAT 99,999,999
COLUMN WebURL FORMAT A15
SELECT * FROM STATES;
COMMIT;
```

150 Глава 2. Обзор SQL и SQL*Plus

```
DELETE FROM STATES;
SELECT * FROM STATES;
ROLLBACK;
SELECT * FROM STATES;
```

6. Найдите на компакт-диске файл `Ch02PopulateStates.sql` и запишите полный путь к нему. После этого введите в SQL*Plus указанный ниже код, нажав клавишу `<Enter>`, чтобы запустить файл сценария. (Вместо `<путь>` введите путь к файлу сценария.)

```
START <путь>\Ch02PopulateStates.sql
```

Последним сообщением, которая отобразится после ряда строк `1 row created`, должно быть `Commit complete`.

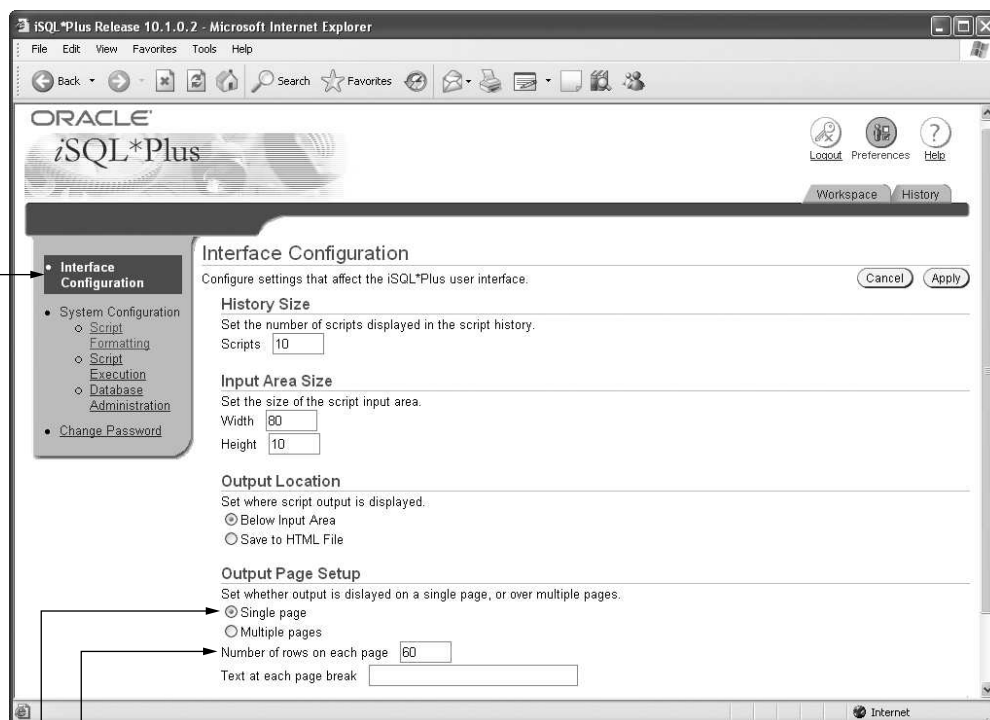
7. Введите `Exit`, чтобы выйти из Oracle и закрыть SQL*Plus.
8. Откройте Web-браузер, откройте *iSQL*Plus* и войдите в Oracle, используя свое имя пользователя и пароль.
9. Щелкните на ссылке **Preferences** возле правого верхнего края экрана, а затем (если потребуется) на ссылке **Interface Configuration** слева. Далее щелкните на кнопке **Single page** под заголовком **OutputPageSetup**, дважды щелкните в текстовом окне **Number of rows on each page** и введите 60 (рис. 2.25). Щелкните на кнопке **Apply**, а затем щелкните на закладке **Workspace** в правом верхней углу страницы.
10. Щелкните в текстовом окне **Workspace** и введите следующий текст, нажимая клавишу `<Enter>` в конце каждой строки:

```
COLUMN Density FORMAT 9,999
SELECT StateID, StateName,
       Population/LandArea AS Density
FROM States
ORDER BY Density DESC;
```

11. Щелкните на кнопке **Execute**, чтобы просмотреть результаты: появится список штатов и округ Колумбия. Наиболее плотно населенные штаты будут идти вверху списка, а менее плотно населенные — внизу.
12. Выберите **File** в строке меню вашего браузера. Выберите **Print** из меню и щелкните на кнопке **Print** диалогового окна **Print**, чтобы напечатать запрос *iSQL*Plus* и результат его выполнения.
13. Щелкните на гиперссылке **Logout**, чтобы выйти из базы данных. Когда появится Web-страница подтверждения выхода, закройте браузер.

Rowing Ventures

Компания Rowing Ventures отвечает за координацию гребных регат в более чем десятке мест всего мира. Помимо надзора за трассой и предоставления места для торговли



Число строк на каждой странице задано равным 60

Кнопка *Single page*

Гиперссылка *Interface Configuration*

Рис. 2.25. Изменение переменной среды iSQL*Plus

и рекламы на гонках, Rowing Ventures хранит массу информации о заездах регаты, участниках, их спонсорах, лодках, выставленных на каждый заезд, номерах и экипажах этих лодок.

Запустите SQL*Plus и войдите в Oracle. После этого на компакт-диске, прилагаемом к этой книге, найдите и запустите файл сценария `Ch02RowingVentures.sql`. Данный файл создает таблицы `Boat` и `Organization`. Откройте файл, предназначенный для буферизации данных, чтобы записать следующие этапы на диск, и выполните команды `DESCRIBE Organization` и `DESCRIBE Boat`, чтобы отобразить на экране названия столбцов указанных таблиц. Если хотите, можете распечатать данный файл, но после этого не закрывайте его. Далее с помощью команды `SELECT` для всех записей выведите на экран следующие поля таблицы `Organization`: `OrganizationID`, `Phone` и поле адреса. Закройте буферный файл и распечатайте его. Очистите экран, затем выведите на него столбцы `BoatID` и `BoatCategory` для всех записей таблицы `Boat`. Удалите первые две записи, задав их значения `BoatID` в команде `DELETE`.

152 Глава 2. Обзор SQL и SQL*Plus

Создайте еще один буферный файл (например, `SPOOL C:\Docs\Myfile.txt`) и выведите на экран столбцы `BoatID` и `BoatCategory` для всех записей в таблице `Boat`. Выполните команду, отменяющую последнее удаление записи. Распечатайте буферный файл. Закройте SQL*Plus.

Broadcloth Clothing

Руководство компании-производителя женской одежды Broadcloth Clothing требует, чтобы вы перечислили содержимое нескольких таблиц корпоративной базы данных, направив эту информацию в файл, который можно напечатать. Итак, прежде всего найдите на компакт-диске файл сценария `Ch02Broadcloth.sql` и запишите путь к нему. Далее запустите SQL*Plus, войдите в Oracle, используя свое имя пользователя и пароль. Запустите буферизацию данных в файл, выполнив следующую команду:

```
SPOOL
<путь>\<имя>Ch02BroadclothSpool.txt
```

Здесь вместо `<путь>` необходимо ввести путь к файлу, а вместо `<имя>` — ваше имя (оно должно быть уникальным для выбранной папки). После этого, используя записанный ранее путь, запустите файл сценария `Ch02Broadcloth.sql`, создающий и заполняющий две таблицы, с которыми вы будете работать в этом упражнении (файл сценария создает таблицы `Factory` и `WorkingConditions`). Используя команду `DESCRIBE`, перечислите для обеих таблиц имена полей и их типы данных. После этого перечислите значения в столбцах `ComplianceID` и `ConditionCategory` для всех строк в таблице `WorkingConditions`. Удалите из таблицы `WorkingConditions` все строки, значение `ComplianceID` которых равно 2003. Снова отобразите те же два столбца для всех строк, используя команду `SELECT`. Отмените удаление и остановите буферизацию в файл. Отредактируйте этот файл в программе Блокнот и добавьте свое имя в первой строке (вставьте в начале файла новую строку). Запишите буферный файл и напечатайте его. Закройте программу Блокнот и введите `Exit` в SQL*Plus, чтобы выйти из SQL*Plus и Oracle.

Запустите *i*SQL*Plus, войдите в базу данных, отобразите структур таблицы `Factory` (имена и типы данных), после чего напечатайте Web-страницу с этими результатами. Щелкните на кнопке `Clear`, чтобы очистить окно `Workspace` и удалить результаты. Затем введите команду `SELECT`, чтобы отобразить все столбцы таблицы `Factory`. Напечатайте запрос и результаты, показанные на Web-странице. Щелкните на кнопке `Clear`, чтобы удалить запрос и результаты. После этого введите и выполните команду `DELETE`, чтобы удалить строки таблицы `Factory`, в которых разница времени по Гринвичу (`GMTDifference`) равна +8 часов. Снова отобразите все строки таблицы `Factory` и напечатайте эту Web-страницу. Щелкните на кнопке `Logout`, чтобы выйти из Oracle, и закройте браузер.