

Sie können an dieser Stelle direkt SQL-Befehle eingeben und ausführen, Scripts starten oder bestimmte Voreinstellungen vornehmen. Geben Sie am Prompt jetzt folgenden Befehl ein

```
SELECT * FROM DEPT;
```

und drücken Sie RETURN. SQL*Plus wird daraufhin eine Fehlermeldung ausgeben, die besagt, dass die Tabelle DEPT nicht existiert. Diese Aussage ist allerdings nicht ganz richtig, denn die Tabelle existiert schon, allerdings unter einem anderen Benutzer.

Wir sind an dieser Stelle wieder an der Benutzerverwaltung angelangt. Neue Benutzer erhalten per Voreinstellung lediglich das Recht sich an der Datenbank anzumelden. Beenden Sie jetzt SQL*Plus durch die Eingabe von *exit* am Prompt. Starten Sie das Programm jetzt neu und melden Sie sich als *scott* an. Das entsprechende Passwort ist *tiger*. Geben Sie jetzt noch einmal die obige Befehlszeile ein. Man kann sich als Anwender auch direkt neu anmelden. SQL*Plus muss also nicht zwingend beendet werden. Dazu gibt man am SQL-Prompt die folgende Anweisung ein:

```
connect <Benutzername>/<Benutzerpaßwort>
```

Um sich als Benutzer *scott* direkt anzumelden, wäre also folgendes einzugeben:

```
connect scott/tiger
```

Es erscheint Abbildung 3-4 auf dem Bildschirm.

```
+ Oracle SQL*Plus
Datei Bearbeiten Suchen Optionen Hilfe
SQL> select * from dept;

  DEPTNO DNAME          LOC
-----
10 ACCOUNTING        NEW YORK
20 RESEARCH          DALLAS
30 SALES              CHICAGO
40 OPERATIONS        BOSTON

SQL>
```

Abbildung 3-4: Generierte Abfrage in SQL*Plus

Damit haben Sie, ohne es vielleicht zu merken, Ihren ersten SQL-Befehl abgesetzt. Mit der Zeile

```
select * from dept;
```

haben Sie das Datenbanksystem angewiesen, den gesamten Inhalt der Tabelle *dept* auszugeben. Anhand des Ausdrucks erkennen Sie, dass sich die Tabelle aus insgesamt drei Spalten (DEPTNO, DNAME und LOC) zusammensetzt und insgesamt vier Sätze beinhaltet.

Ein SQL-Befehl wird unter SQL*Plus durch ein Semikolon beendet. Dadurch ist es möglich, komplexere Befehle in mehr als einer Zeile zu schreiben (Abbildung 3-5).

```
+ Oracle SQL*Plus
Datei Bearbeiten Suchen Optionen Hilfe
SQL> select
2 *
3 from
4 dept
5 ;

  DEPTNO DNAME          LOC
-----
10 ACCOUNTING        NEW YORK
20 RESEARCH          DALLAS
30 SALES              CHICAGO
40 OPERATIONS        BOSTON

SQL> |
```

Abbildung 3-5: Mehrzeilige Eingaben

Oftmals ist es interessant, solche Ergebnisse nicht nur einmal vom Programm ausgeben zu lassen, sondern in Textdateien zu speichern. Zu diesem Zweck kennt SQL*Plus den Befehl SPOOL. Über die folgende Zeile weisen Sie das Programm an, die Ergebnisse in einer Textdatei auf dem Laufwerk C abzulegen:

```
SPOOL C:\PROTOKOL.TXT
```

Über die Anweisung SPOOL OFF schalten Sie die Protokollfunktion wieder ab; alle folgenden Ausgaben werden lediglich auf dem Bildschirm ausgegeben. In der entsprechenden Datei finden Sie daraufhin folgenden Inhalt:

```
SQL> SELECT * FROM DEPT;
```

```
  DEPTNO DNAME          LOC
-----
10 ACCOUNTING        NEW YORK
20 RESEARCH          DALLAS
30 SALES              CHICAGO
40 OPERATIONS        BOSTON
```

```
SQL> SPOOL OFF
```

Hinweis:

Das Ein- und Ausschalten der Spool-Funktionalität erreichen Sie auch über das Menü *Datei / Spool / Spool Datei*.

Hierüber erhält der Anwender ebenso die Möglichkeit, eigene Skripte zu generieren. Das folgende Beispiel zeigt ein solches Script:

```
spool zaehler.sql
SELECT 'SELECT COUNT(*) FROM '||owner||'.'||table_name||';'
FROM ALL_TABLES;
spool off
```

select owner, table_name from all-tables

Die obigen Anweisungen generieren ein Script, in dem für jede Tabelle eine Zeile erzeugt wird. In jeder dieser Zeilen wird jeweils die Anzahl Datensätze gezählt.

3.1.2 Script-Verarbeitung

Eine weitere interessante und vor allem nützliche Eigenschaft von SQL*Plus ist die Möglichkeit, Scripts ablaufen zu lassen. Weiter vorne in diesem Kapitel wurde schon erwähnt, dass man Scripts auch als Kommandozeilenparameter beim Start von SQL*Plus mit übergeben kann. Jetzt soll noch einmal genau die Funktion und vor allem die Syntax der Script-Verarbeitung erläutert werden.

Ein solches Script ist nichts weiter als eine Sammlung von SQL-Anweisungen. Eine einzelne Anweisung wird - wie auch in SQL*Plus - durch Semikolon abgeschlossen. Damit ist ein Script vergleichbar mit einem Quelltext einer anderen Programmiersprache wie beispielsweise Turbo Pascal oder C. Erzeugen Sie jetzt eine neue Textdatei mit dem Notepad von Windows und tragen Sie dort die schon bekannte Befehlszeile

```
SELECT * FROM DEPT;
```

ein. Speichern Sie danach diese Datei unter dem Namen *SCRIPT1.SQL* auf dem Laufwerk C. In SQL*Plus starten Sie dieses Script jetzt über den Befehl *START*:

```
START C:\SCRIPT1.SQL
```

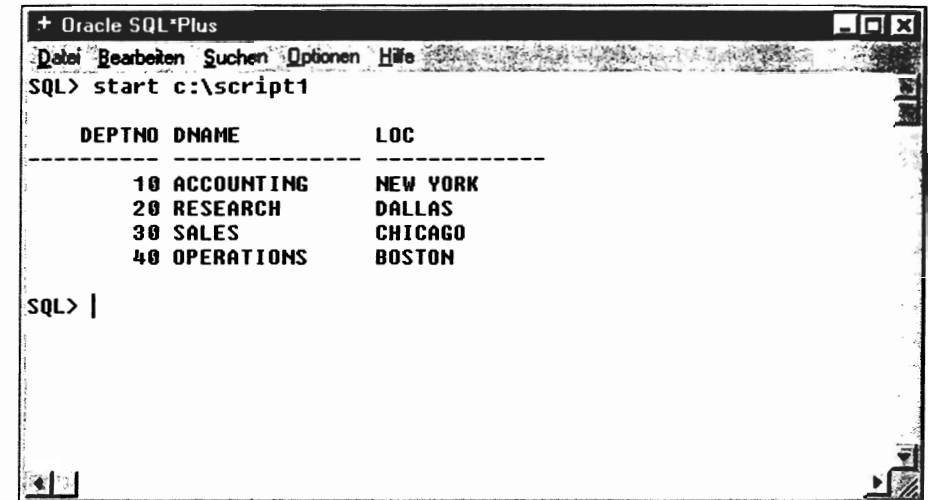


Abbildung 3-6: Script-Ausführung in SQL*Plus

Hinweis:

Alternativ zu dem Befehl *START* stehen Ihnen die Befehle *GET* und *RUN* zur Verfügung. Über *GET* wird die Script-Datei geladen, über *RUN* wird sie ausgeführt. Als Synonym für die Anweisung *START* kann auch ein „@“ eingegeben werden.

Das Ergebnis ist aber das gleiche. Erzeugen Sie ein weiteres Script, diesmal allerdings mit der Bezeichnung *SCRIPT2.SQL*. In diese Datei tragen Sie die folgende Zeile ein:

```
START C:\SCRIPT1.SQL
```

Starten Sie jetzt über die Zeile

```
START C:\SCRIPT2.SQL
```

das soeben erstellte Script. Wie Sie sehen, erfolgt eine analoge Ausgabe (Abbildung 3-7). Es ist also auch möglich, ein Script durch ein anderes aufzurufen.