

Zur Benutzung des Datenbank-Systems ORACLE

Jürgen Kalinski
Informatik III, Universität Bonn
email: cully@cs.uni-bonn.de

27. Oktober 1998

1 Präliminarien

Der Aufruf von ORACLE setzt voraus, daß spezifische Systemvariablen korrekt belegt sind. Deshalb sind die folgenden Zeilen der `.cshrc`-Datei hinzuzufügen.

```
switch ( 'uname -n' )
case stannum*:
    source /opt/bin/oraclerc
    breaksw
default:
    breaksw
endsw
```

Danach kann auf dem Rechner `stannum` (und nur dort!) in der im folgenden beschriebenen Weise mit ORACLE gearbeitet werden.

2 SQL*Plus

Der Befehl `sqlplus` startet die SQL*Plus-Shell mit ihrem recht simpel gehaltenen Oberflächenkomfort. Es erfolgt eine Abfrage des ORACLE-Benutzernamens und des ORACLE-Benutzerkennworts:

```
Enter user-name:
Enter password:
```

Anschließend erscheinen die Meldungen:

```
Connected to:
Oracle8 Enterprise Edition Release 8.0.3.0.0 - Production
With the Partitioning and Objects options
PL/SQL Release 8.0.3.0.0 - Production
```

Hinter dem `SQL>`-Prompt können nun SQL-Befehle eingegeben werden. Man beachte, daß SQL-Befehle sich über mehrere durch `Return` abgeschlossene Zeilen erstrecken können. Sie gelten erst durch die Eingabe eines Semikolons als beendet!

```
SQL> select * <Return>
2 from <Return>
3 user_tables; <Return>
```

Falls statt des `SQL>`-Prompts eine Fehlermeldung wie

```

ERROR: ORA-01034: ORACLE not available
ORA-07429: msgsg: shmget() failed to get segment.
SVR4 Error: 2: No such file or directory

```

ausgegeben wird, so liegt ein Systemfehler vor. Hier bleibt einem nichts anderes übrig als den Dialog mit 'Control-D' abzubrechen und eine E-Mail an `cully` zu schicken.

Die SQL*Plus-Schnittstelle besitzt den Charme eines Bauhaus-Stuhls: Hier ist kein Schnörkel zuviel. Eine Eingabe wie

```

SQL> create tabel person (
  2 id number(6,0),
  3 name text,
  4 gehalt number(6,2)
  5 );

```

wird mit der Meldung

```

      *
ERROR at line 1:
ORA-00901: invalid CREATE command

```

beantwortet. Einfache textuelle Ersetzungen des vorangegangenen Kommandos lassen sich mit 'c/<alter Text>/<neuer Text>' durchführen (*change*). Das erste Auftreten des alten Textstücks wird durch das neue ersetzt, und die veränderte Kommandozeile ausgegeben.

```

SQL> c/tabel/table
1* create table person (

```

Es können mehrere solcher Veränderungen vorgenommen werden. Das modifizierte SQL-Kommando wird mit 'r' gestartet (*run*). Es wird dann noch einmal vollständig ausgegeben und die Bearbeitung gestartet.

```

SQL> r
  1 create table person (
  2 id number(6,0),
  3 name text,
  4 gehalt number(6,2)
  5* )
name text,
      *
ERROR at line 3:
ORA-00902: invalid datatype

```

```

SQL> c/text/varchar2(20)
  3* name varchar2(20),
SQL> r
  1 create table person (
  2 id number(6,0),
  3 name varchar2(20),
  4 gehalt number(6,2)
  5* )

```

Table created.

```

SQL> insert into person
  2 values(600306, 'Schnaufelberger', 2896.75);

```

1 row created.

SQL> select * from person;

ID	NAME	GEHALT
600306	Schnaufelberger	2896.75

SQL> drop table person;

Table dropped.

Grössere Änderungen eines zuvor eingegebenen Befehls können auch in einem Editor (z.B. emacs, vi) vorgenommen werden, der mit `edit` aufgerufen wird.

Selbstverständlich müssen SQL-Befehle nicht grundsätzlich interaktiv eingegeben werden. Befehlsfolgen können in einer Datei stehen, deren Bearbeitung dann mit `start <Dateiname>` angestossen wird. Man beachte aber auch hier stets, daß SQL-Befehle mit einem Semikolon abgeschlossen werden müssen.

Das Schema einer Relation kann man sich mit

```
desc <Relationsname>
```

anzeigen lassen. Abschließend sei erwähnt, daß nach der Eingabe von

```
spool <Protokolldatei>
```

alle Ein- und Ausgaben einer interaktiven Sitzung protokolliert werden, bis SQL*Plus verlassen oder der Protokollmodus mittels `spool off` beendet wird.

3 Bemerkungen

Umfangreiche Online-Dokumentation zu ORACLE kann mit jedem HTML-Browser unter

```
/usr/datasystems/oracle/doc01
```

gelesen werden.