

Übung 5: (Zugang zur Oracle Datenbank als Console-Anwendung)

Neben der für die Oracle Datenbank bisher benutzten Anwendung **SQL Developer 2.1** mit graphischer Benutzeroberfläche, gibt es auch eine rein textbasierte Anwendung zum Ausführen von SQL-Befehlen und SQL-Skript-Dateien. Diese so genannte Terminal- oder Console-Anwendung (Programmname **sqlplus**) wird direkt im Terminal-/Console-Fenster ausgeführt.

Das Programm ist für beide Betriebssysteme Linux- oder Windows verfügbar.

Für die Anmeldung sind dem Programm **sqlplus** die Parameter für die Datenbank-Anmeldung zu übergeben:

sqlplus ss12etxyz@ora11g

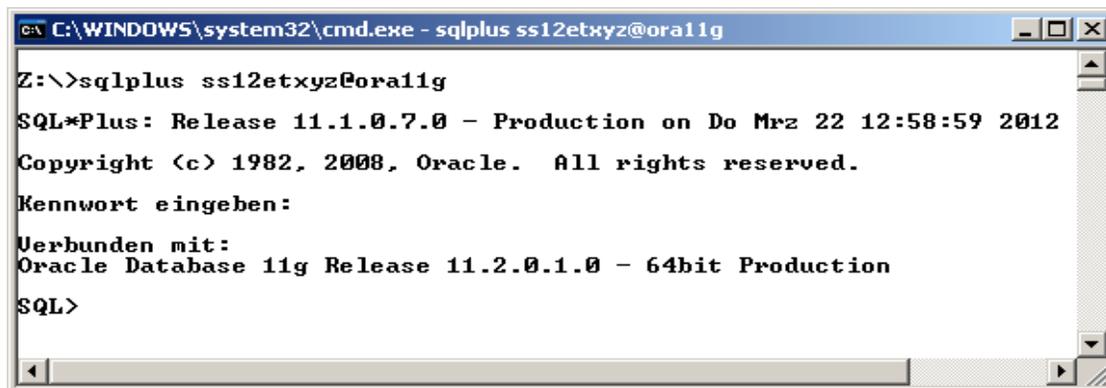
↑
Programm-
name

↑
Benutzer-
name

↑
Kennung
des Oracle-Servers

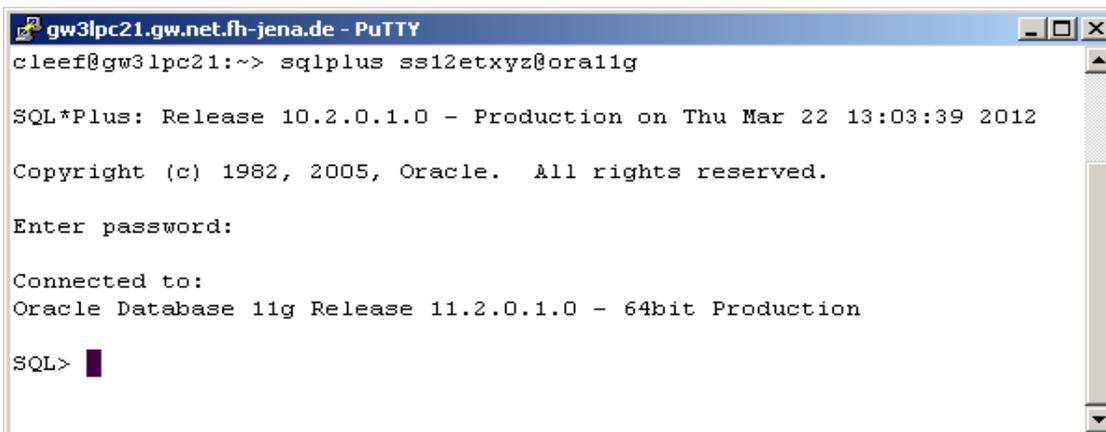
Das Kennwort für den Benutzer wird anschließend angefordert (verdeckte Eingabe), anschließend meldet sich der Datenbank-Server mit dem SQL-Prompt und es können dann SQL-Anweisungen ausgeführt werden:

Eingabeaufforderung (Windows) bzw. Terminal (Linux):



```
C:\WINDOWS\system32\cmd.exe - sqlplus ss12etxyz@ora11g

Z:\>sqlplus ss12etxyz@ora11g
SQL*Plus: Release 11.1.0.7.0 - Production on Do Mrz 22 12:58:59 2012
Copyright (c) 1982, 2008, Oracle. All rights reserved.
Kennwort eingeben:
Verbunden mit:
Oracle Database 11g Release 11.2.0.1.0 - 64bit Production
SQL>
```



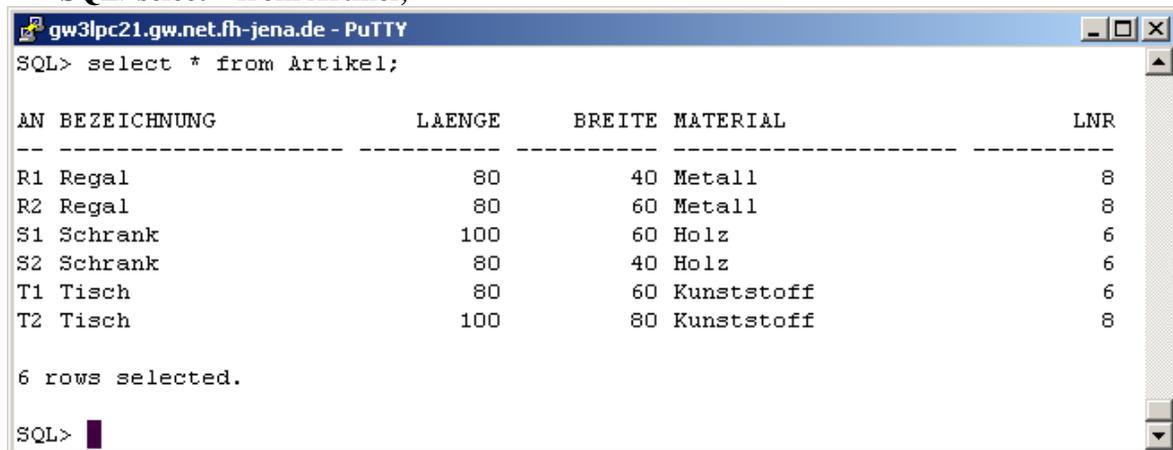
```
gw3lpc21.gw.net.fh-jena.de - PuTTY
cleef@gw3lpc21:~> sqlplus ss12etxyz@ora11g

SQL*Plus: Release 10.2.0.1.0 - Production on Thu Mar 22 13:03:39 2012
Copyright (c) 1982, 2005, Oracle. All rights reserved.
Enter password:
Connected to:
Oracle Database 11g Release 11.2.0.1.0 - 64bit Production
SQL> █
```

Hinweis: Die Eingabe des Kennworts erfolgt ohne Anzeige der eingegebenen Zeichen!

Dies ist die Dialog-Variante des Programms. Hinter dem SQL-Prompt können nun entweder direkt SQL-Anweisungen eingegeben werden (Abschluss einer Anweisung durch ein Semikolon), z.B.

SQL>select * from Artikel;



```
SQL> select * from Artikel;

AN BEZEICHNUNG          LAENGE    BREITE MATERIAL          LNR
-----
R1 Regal                80        40 Metall                8
R2 Regal                80        60 Metall                8
S1 Schrank             100        60 Holz                  6
S2 Schrank             80        40 Holz                  6
T1 Tisch                80        60 Kunststoff            6
T2 Tisch               100        80 Kunststoff            8

6 rows selected.

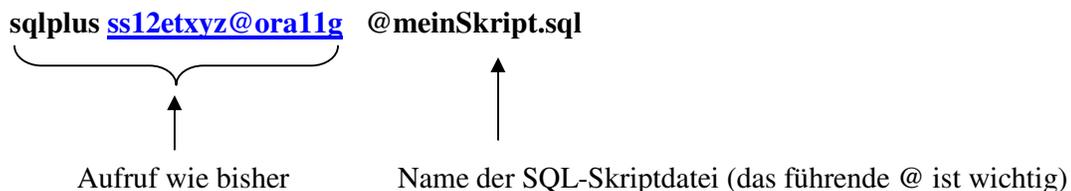
SQL>
```

oder es kann direkt die Ausführung einer SQL-Skriptdatei gestartet werden, dabei wird **wahlweise** das Schlüsselwort **start** oder das Zeichen **@** benutzt, z.B.

SQL>start meinSkript.sql oder **SQL>@meinSkript.sql**

Eine SQL-Skriptdatei ist dabei eine Textdatei (i. A. mit der Datei-Erweiterung sql), die eine Reihe von SQL-Anweisungen enthält.

Es kann aber auch schon beim Programmstart direkt eine SQL-Skriptdatei zur Ausführung angegeben werden:



die direkt nach dem Programmstart automatisch ausgeführt wird. Beim Einsatz von SQL-Skriptdateien gibt es zwei Varianten:

- die SQL-Skriptdatei enthält am Ende eine **exit**-Anweisung
In diesem Fall wird das Programm sqlplus nach Ausführung der SQL-Skriptdatei **beendet** und die **Console** meldet sich wieder zur Eingabe eines **Shell-Kommandos**.
- die SQL-Skriptdatei enthält keine **exit**-Anweisung
In diesem Fall wird das Programm sqlplus nach der Ausführung der SQL-Skriptdatei in der Dialogvariante **fortgesetzt** und **sqlplus** meldet sich wieder SQL-Prompt zur weiteren Eingabe von **SQL-Anweisungen**.

Sofern noch nicht geschehen, kopieren Sie die SQL-Skriptdateien:

create_ORA.sql
insert_ORA.sql
init_LNR_ORA.sql

für den Oracle-Server und

create_MYSQL.sql

insert_MYSQL.sql

init_LNR_MYSQL.sql

für den MySQL-Server aus dem Kurs-Verzeichnis in Ihr Home-Verzeichnis und starten Sie die Console (Shell).

Wechseln Sie dort in Ihr Home-Verzeichnis und führen Sie nacheinander die SQL-Skriptdateien mit **sqlplus** aus:

sqlplus ss12etxyz@ora11g @create_ORA.sql

sqlplus ss12etxyz@ora11g @insert_ORA.sql

sqlplus ss12etxyz@ora11g @init_LNR_ORA.sql

Grundsätzlich können Sie auch bereits beim Programmstart das Kennwort für den Benutzer offen mit angeben – dann entfällt die verdeckte nachträgliche Eingabe, also z.B.

sqlplus ss12etxyz/geheim@ora11g

sqlplus ss12etxyz/geheim@ora11g @create_ORA.sql

Hinweis:

Alle SQL-Skriptdateien enthalten am Ende keine exit-Anweisung, deshalb wird das Programm sqlplus nach der Ausführung nicht beendet, sondern verbleibt in der Dialog-Variante zur Eingabe weiterer Kommandos. Die Exit-Anweisung beendet danach sqlplus.

Einfacher wird die Sache vielleicht, wenn die Dialog-Variante gestartet wird und dann dort nacheinander die Ausführung der drei Skriptdateien gestartet werden.

Übung 6: (Zugang zur MySQL Datenbank als Console-Anwendung)

Auch für die MySQL-Datenbank gibt es eine zu sqlplus analoge Console-Anwendung (Programmname **mysql**). Für die Anmeldung sind auch hier dem Programm **mysql** nach einem Programm spezifischem Schema die erforderlichen Parameter für die Datenbank-Anmeldung zu übergeben:

mysql -hgw3svr32.gw.net.fh-jena.de -uss12etxyz -pgeheim ss12etxyz

↑ Programm-name

↑ Datenbank-Server (Host-Name)

↑ Benutzer-Name

↑ Kennwort

↑ Datenbank (Default Schema)

dabei folgt hinter der Option:

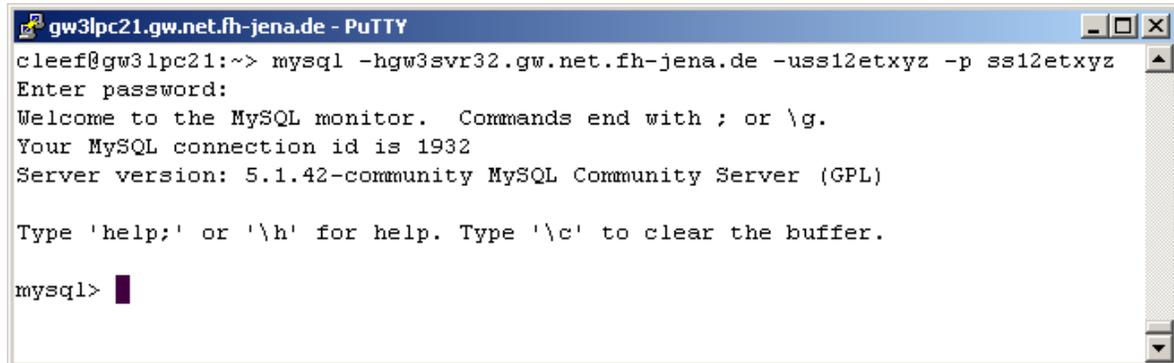
-h der Name (gw3svr32.gw.net.fh-jena.de) des Datenbank-Server im Netzwerk

-u folgt der Benutzername (ss12etxyz)

-p folgt das Kennwort (geheim).

Wenn nur die Option **-p** ohne Angabe des Kennwortes verwendet wird, wird die (verdeckte) Eingabe des Kennwortes nach dem Programmstart explizit angefordert.

Am Ende wird ohne Option ein Datenbank-Name (Default-Schema) angegeben, dieser ist bei uns immer identisch mit dem Benutzernamen:



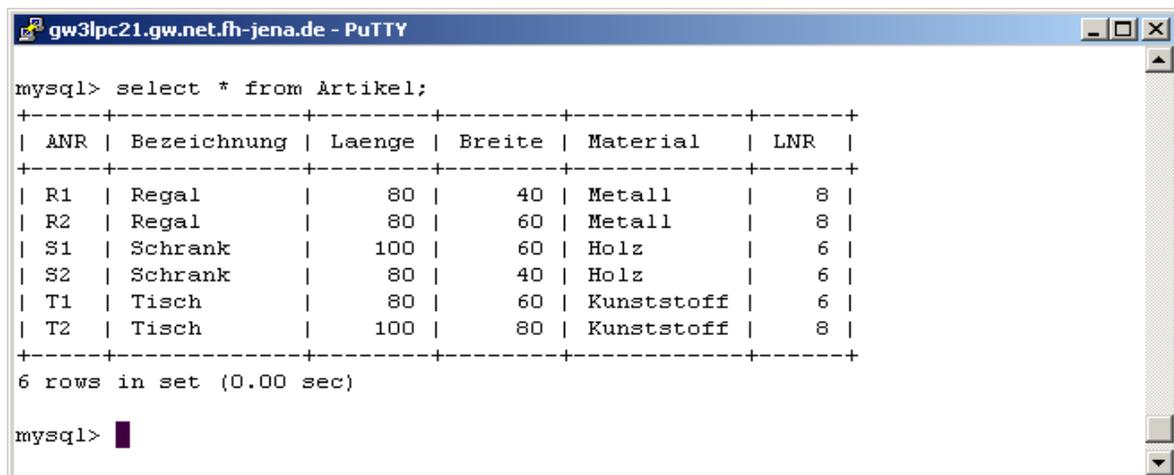
```
gw3lpc21.gw.net.fh-jena.de - PuTTY
cleef@gw3lpc21:~> mysql -hgw3svr32.gw.net.fh-jena.de -uss12etxyz -p ss12etxyz
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1932
Server version: 5.1.42-community MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

Anschließend meldet sich die Console-Anwendung mysql in der Dialog-Variante und es kann wieder eine SQL-Anweisung eingegeben werden (Abschluss einer Anweisung durch ein Semikolon), z.B.

mysql>select * from Artikel;



```
gw3lpc21.gw.net.fh-jena.de - PuTTY

mysql> select * from Artikel;
+-----+-----+-----+-----+-----+-----+
| ANR | Bezeichnung | Laenge | Breite | Material | LNR |
+-----+-----+-----+-----+-----+-----+
| R1 | Regal      | 80     | 40    | Metall  | 8   |
| R2 | Regal      | 80     | 60    | Metall  | 8   |
| S1 | Schrank    | 100    | 60    | Holz    | 6   |
| S2 | Schrank    | 80     | 40    | Holz    | 6   |
| T1 | Tisch      | 80     | 60    | Kunststoff | 6  |
| T2 | Tisch      | 100    | 80    | Kunststoff | 8  |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

oder es kann direkt die Ausführung einer SQL-Skriptdatei gestartet werden, dabei wird das Schlüsselwort **source** (an Stelle von **start** oder **@** bei sqlplus) verwendet, z.B.

mysql>source meinSkript.sql

Auch bei der Console-Anwendung mysql kann bereits beim Programmstart eine SQL-Skriptdatei zur automatischen Ausführung angegeben werden, jedoch sollte jetzt zusätzlich die Option **-f** angegeben werden, da sonst die Ausführung der Skriptdatei beim ersten Fehler beendet wird:

mysql -hgw3svr32.gw.net.fh-jena.de -uss12etxyz -pgeheim -f ss12etxyz < meinSkript.sql

Aufruf mit zusätzlicher Option -f

Name der SQL-Skriptdatei

(das Kleiner-Zeichen vor dem Namen der SQL-Skriptdatei ist dabei zwingend erforderlich!)

Bei der Console-Anwendung mysql wird nach der letzten SQL-Anweisung der Skriptdatei die Anwendung automatisch beendet, eine exit-Anweisung in der Skriptdatei ist nicht zulässig und führt zu einer Fehlermeldung.

Bitte benutzen Sie die SQL-Skriptdateien **create_MYSQL.sql**, **insert_MYSQL.sql** und **init_LNR_MYSQL.sql** (ohne abschließende exit-Anweisung) und erstellen und füllen Sie damit erneut in der MySQL-Datenbank die Basistabellen von Beispiel 1 mit der Console-Anwendung mysql.

Hinweis: Aufgrund eines Fehlers bei der Behandlung der DELIMITER-Anweisung innerhalb einer Skriptdatei, muss nach der Rückkehr in die Dialog-Variante das Begrenzungszeichen mit der zusätzlichen Anweisung **delimiter ;** neu gesetzt werden!

Übung 7: (Einfügen, Löschen und Ändern von Tabellenzeilen)

Diese Übung sollte auf dem **Oracle-Server** und mit der **Anwendung** SQL Developer 2.1 unter Linux durchgeführt werden:

Wichtiger Hinweis:

Für die Oracle Datenbank sind **Änderungen des Inhalts einer Tabelle mit der speziellen SQL-Anweisung: commit; zu bestätigen (siehe Skriptdatei insert_ORA.sql)!**

(a) Erstellen Sie in ihrem Home-Verzeichnis unter dem Namen **create_neu.sql** zunächst eine Kopie der SQL-Skriptdatei **create_ORA.sql** und ändern dann anschließend in **create_neu.sql** die Namen für die Tabellen, z.B. neuer Name

für die Artikel-Tabelle: product

für die Lieferant-Tabelle: vendor

für die liefert-Tabelle: delivery

Beachten Sie dabei, dass die Namen nicht nur in der Zeile create table ..., sondern überall geändert werden müssen (Referenz bei Fremdschlüssel, Constraint-Namen, drop table ...). Da die Constraint-Namen ebenfalls innerhalb eines Schemas eindeutig sein müssen, sind auch hier Änderungen erforderlich.

Führen Sie anschließend die neue SQL-Skriptdatei **create_neu.sql** aus. Danach sollten die neuen Tabellen angelegt, aber noch leer sein.

Fügen Sie alle weiteren SQL-Anweisungen für (b) bis (f) in die Skriptdatei **create_neu.sql** ein und benutzen Sie einzelne Anweisungen aus (g) um nach den einzelnen Änderungen die dann aktuellen Tabelleninhalte zu dokumentieren.

Alle nachfolgenden durchzuführenden Änderungen beziehen sich immer auf diese neu erstellten Tabellen **vendor**, **product** und **delivery**. Die Ausgangstabellen Lieferant, Artikel und liefert bleiben unverändert!

- (b) Füllen Sie alle drei in (a) neu erstellten Tabellen und benutzen Sie dazu die nützliche Variante der insert-Anweisung, bei der die einzufügenden Daten durch eine Select-Anweisung bereitgestellt werden, z.B. für die Tabelle product:

```
insert into product (select * from Artikel);
```

d.h. benutzen Sie die Tabellen Artikel, Lieferant und liefert, um die entsprechenden neu erstellten Tabellen identisch zu füllen. Anschließend gibt es für die drei Tabellen Lieferant, Artikel und liefert in Struktur und Inhalt nahezu identische Kopien vendor, product und delivery. **Bitte beachten Sie dabei die Reihenfolge der insert-Anweisungen, die sich zwangsläufig aus den Fremdschlüsseln ergibt.**

- (c) Benutzen Sie anschließend die values-Variante der insert-Anweisung, um in die neu erstellten Tabellen **product** und **vendor** jeweils eine oder auch mehrere Zeilen neu einzufügen.
- (d) Führen Sie anschließend für den Lieferanten mit der Lieferantenummer LNR = 8 in der neu erstellten Tabelle delivery eine Preisreduzierung um 10 % durch, d.h. die Preise für alle von diesem Lieferanten gelieferten Artikel müssen um 10 % gesenkt werden.
- (e) Nehmen Sie anschließend den Schrank mit der Artikelnummer S1 aus dem Sortiment, d.h. löschen Sie sowohl den Eintrag als Artikel (Tabelle product), als auch alle abhängigen Sätze in den Lieferbedingungen (Tabelle delivery). **Bitte beachten Sie auch hier wieder die Reihenfolge der Lösch-Anweisungen.**
- (f) Fügen Sie anschließend für den (einen) in (c) neu erstellten Lieferanten (Tabelle: vendor) neue Zeilen in die Lieferbedingungen (Tabelle delivery) ein. Dabei soll der neu erstellte Lieferant dieselben Artikel wie der Lieferant mit der LNR = 8 anbieten, aber mit um 10 % erhöhten Preisen.
- (g) Mit den Anweisungen

```
select * from vendor;  
select * from product;  
select * from delivery;
```

kann jederzeit der aktuelle Inhalt der Tabellen ausgegeben werden. Benutzen Sie diese Möglichkeit, um jeweils punktuell vor bzw. nach den Änderungen in (b) bis (f) die Tabelleninhalte anzuzeigen und so vergleichen zu können.

Übung 8: (Beispiele für die Select-Anweisung)

Bearbeiten Sie die folgenden einführenden Beispiele aus der Vorlesung für die Select-Anweisung in Ihrem Schema und variieren Sie dabei auch einzelne Teile, um mit der Grundstruktur der Select-Anweisung vertraut zu werden.

Beispiel 1:

```
Select ANR, Bezeichnung  
from Artikel  
where Material = 'Metall';
```

Beispiel 2:

```
Select LNR, Name, Status  
from Lieferant  
where Status = 10;
```

Beispiel 3:

```
Select LNR, Name, Status  
from Lieferant  
where Status = 10 and Ort = 'Jena';
```

Beispiel 4:

```
Select COUNT(*) AS Anzahl  
from Lieferant;
```

Beispiel 5:

```
Select MIN(Laenge) AS Minimum, MAX(Laenge) AS Maximum  
from Artikel;
```

Beispiel 6:

```
Select ANR, Laenge * Breite AS Flaechе  
from Artikel  
where Bezeichnung > 'Stuhl';
```