

## SQLPlus

SQL\*PLUS ist ein einfaches Tool um SQL Abfragen von einer Oracle Instanz ausführen zu lassen.

### SQL-Plus starten

sqlplus	user/passwort@ORACLE_SID
sqlplus	user/passwort
sqlplus	sys/passwort@ORACLE_SID as sysdba
sqlplus	sys/passwort as sysdba
sqlplus	"/ as sysdba"

Nur Benutzer die SYSDBA oder SYSOPER Rechte haben, können sich an der Oracle Instanz als sysdba oder als sysoper anmelden.

### SQL-Plus Befehle für Anfänger und Profis

SQL-Plus Befehle sind dazu da, die Konfiguration von SQL-Plus zu ändern und zu ergänzen und um einige zusätzliche Funktionen die über ein SELECT, UPDATE und DELETE hinausgehen zur Verfügung zu haben. In den meisten Fällen ist es für den Anwender leichter sich mit einem Datenbanktool wie z.B. TOAD oder dem PLSQL-Developer zu rüsten als sich in den Tiefen von SQL-Plus zu verlieren. Leider ist das nicht immer möglich. Z.B. wenn die Ergebnisse einer SQLPlus Anfrage mit einem Shellscript in awk oder sed weiterverarbeitet werden sollen das in regelmäßigen Abständen per cronjob aufgerufen wird. In dem Fall ist SQL-Plus die richtige Lösung. Für die Verwendung von SQL-Plus muss man mindestens einen Oracle Client installiert haben. Der Aufruf von sqlplus sieht dann folgendermaßen aus.

```
%>sqlplus user/passwort@ORACLE_SID
```

Die ORACLE\_SID ist die ID der Oracle Datenbank die verwendet werden soll. Erhält man bei der Anmeldung den Fehler "ORA-12154: TNS:Der Servicename konnte nicht aufgelöst werden." kann die ORACLE\_SID nicht gefunden werden. Die verwendete ORACLE\_SID muss in der Datei tnsnames.ora eingetragen sein, die sich unter [ORACLE\_INSTALLATION]/network/ADMIN/tnsnames.ora findet. Ein Eintrag mit der SID TESTDB könnte beispielsweise so aussehen.

```
TESTDB_192.168.100.1 =
```

```

(DESCRIPTION =

  (ADDRESS_LIST =

    (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.100.1)(PORT = 1521))

  )

  (CONNECT_DATA =

    (SID = TESTDB)

    (SERVER = DEDICATED)

  )

)

```

Hinweis: Es ist möglich die Datei im Texteditor zu bearbeiten. Besser ist es aber dies nicht zu tun, da "vor allem ältere Oracle Versionen" sehr allergisch auf scheinbar korrekte Einträge reagieren. Am besten ist es, man verwendet ein Tool um hier Einträge zu Ändern.

Wenn man sich nun erfolgreich an der Oracle Datenbank angemeldet hat, erscheint der folgende Bildschirm.

```

%>sqlplus XXX/XXX@TESTDB
SQL*Plus: Release 9.2.0.1.0 - Production on Di Jan 3 13:22:38 2006
Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.
Verbunden mit:
Oracle9i Release 9.2.0.4.0 - Production
JServer Release 9.2.0.4.0 - Production
SQL>

```

Wenn man sich das erste Mal mit SQL-Plus an der Oracle Datenbank anmeldet ist es von Vorteil, wenn man sich erst mal einen Überblick verschafft, ob man richtig angemeldet ist. Hierzu kann man unterschiedliche SQL Befehle nutzen, die in den meisten Fällen schon ausreichend sind.

```
select TABLE_NAME from USER_TABLES;
```

Zeigt die Tabellennamen im eigenen Tablespace an.

```
select DISTINCT TABLESPACE_NAME from USER_TABLES;
```

Zeigt den Tablespace an in dem man sich gerade befindet. Darüber hinaus gibt es noch endlos viele Tabellen in denen man zusätzliche Informationen darüber erfragen kann, die eine Auskunft über Oracle Datenbank geben können. Auf eine komplette Auflistung des Oracle Data Dictionary möchte ich an der Stelle aber verzichten. Wer sich dafür interessiert, kann ja mal nach "Oracle Data Dictionary" googlen. Neben den SQL Befehlen SELECT, INSERT, UPDATE gibt es in SQL-Plus

noch eine Reihe weiterer Befehle, die bei der täglichen Arbeit mit Oracle helfen können. Eine Liste der Befehle erhält man mit

```
SQL> HELP INDEX
```

@	COPY	PAUSE	SHUTDOWN
@@	DEFINE	PRINT	SPOOL
/	DEL	PROMPT	SQLPLUS
ACCEPT	DESCRIBE	QUIT	START
APPEND	DISCONNECT	RECOVER	STARTUP
ARCHIVE LOG	EDIT	REMARK	STORE
ATTRIBUTE	EXECUTE	REPFOOTER	TIMING
BREAK	EXIT	REPHEADER	TTITLE
BTITLE	GET	RESERVED WORDS (SQL)	UNDEFINE
CHANGE	HELP	RESERVED WORDS (PL/SQL)	VARIABLE
CLEAR	HOST	RUN	WHENEVER OSERROR
COLUMN	INPUT	SAVE	WHENEVER SQLERROR
COMPUTE	LIST	SET	
CONNECT	PASSWORD	SHOW	

Mit diesen Befehlen möchte ich nun den Leser im Weiteren an ein paar Beispielen und mit meinen Erfahrungen vertraut machen.

## **SQL-Plus Kommandozeilenparameter**

SQL-Plus hat eine Reihe von Parametern, die direkt beim Start der Anwendung übergeben werden können. Eine Übersicht der SQL-Plus Kommandozeilenparameter kann man mit sich auf der Konsole anzeigen lassen.

```
%>sqlplus -h
```

```
SQL*Plus: Release 9.2.0.1.0 - Production
```

```
Verwendung: SQLPLUS [ [<option>] [<logon>] [<start>] ]
```

```
wobei <option> ::= -H | -V | [ [-L] [-M <o>] [-R <n>] [-S] ]
```

```
<logon> ::= <username>[/<password>][@<connect_string>] | / | /NOLOG
```

```
<start> ::= @<URI>|<filename>[.<ext>] [<parameter> ...]
```

```
"-H" zeigt SQL*Plus-Versions-Banner und Syntax von Verwendung an
```

```
"-V" zeigt SQL*Plus-Versions-Banner an
```

```
"-L" versucht eine einmalige Anmeldung
```

```
"-M <o>" verwendet HTML-Markup-Optionen <o>
```

```
"-R <n>" verwendet eingeschränkten Modus <n>
```

```
"-S" verwendet Ton aus-Modus
```

So ist es unter anderem möglich, SQL-Anweisungen aus einer Datei heraus ausführen zu lassen. Dies ist auch aus einer schon laufenden SQL-Plus Anwendung heraus möglich, aber vor allem der Aufruf

über die Kommandozeilenparameter ist in vielen Fällen eine sehr elegante Möglichkeit um z.B. Oracle Daten in einfachen Shell Skripten zu verwenden. Das sieht dann wie folgt aus:

```
SQL> sqlplus user/passwort@ORACLE_SID @datei.sql
```

In dem Fall wird dann die Datei "datei.sql" ausgeführt. Alle SQL und SQL-Plus Befehle in der Datei werden von oben nach unten abgearbeitet und auf der Konsole ausgegeben. Ist der letzte Befehl in der Datei ein "quit", wird SQL-Plus wieder beendet und das Shell wird weiter ausgeführt. Wenn man die Ausgaben z.B. mit awk weiter verarbeiten zu können, ist es sinnvoll, einige Kleinigkeiten zu beachten:

- Die Verwendung der option "-S" unterdrückt die Informationen beim Starten und Beenden von SQL-Plus
- "SET PAGES 9999" setzt die Größe einer Seite auf 9999 Zeilen. Hierdurch werden unnötige Leerzeilen unterdrückt.
- "SET LIN 9999" setzt die Zeilenlänge auf 9999. Hierdurch wird der Zeilenumbruch verhindert.
- "SET HEADING OFF" unterdrückt die Ausgabe der Kopfzeilen.

## **SQL-Plus Befehlsübersicht**

Die nun folgenden 53 SPL-Plus Befehle sind alle in Oracle 9.2 verfügbaren SQL-Plus Befehle. Bei diesen Befehlen ist es egal, ob sie in Groß- oder in Kleinbuchstaben eingegeben werden. Die meisten Befehle benötigen mindesten einen oder mehr Parameter. Eine detaillierte englisch Hilfe zu einem Befehl kann man sich mit dem SQL-Plus Befehl "HELP" anzeigen lassen. Umfangreiche deutsche Infos zu den Befehlen gibt es wenn man dem Link zu dem SQL-Plus Namen der Liste weiter unten folgt.

```
SQL>HELP [topic]
```

Kommentare werden in SQL-Plus mit einem "-" am Anfang der Zeile Markiert.

<b>SQL-Plus Befehl</b>	<b>Kurzform</b>
<u>@</u>	-
<u>@@</u>	-
<u>/</u>	-

<u>ACCEPT</u>	ACC
<u>APPEND</u>	A
<u>ARCHIVE LOG</u>	-
<u>ATTRIBUTE</u>	-
<u>BREAK</u>	BRE
<u>BTITLE</u>	BTI
<u>CHANGE</u>	C
<u>CLEAR</u>	CL
<u>COLUMN</u>	COL
<u>COMPUTE</u>	COMP
<u>CONNECT</u>	CONN
<u>COPY</u>	-
<u>DEFINE</u>	DEF
<u>DEL</u>	-
<u>DESCRIBE</u>	DESC

<u>DISCONNECT</u>	DISC
<u>EDIT</u>	ED
<u>EXECUTE</u>	EXEC
<u>EXIT</u>	-
<u>GET</u>	-
<u>HELP</u>	-
<u>HOST</u>	HO
<u>INPUT</u>	I
<u>LIST</u>	L
<u>PASSWORD</u>	PASSW
<u>PAUSE</u>	PAU
<u>PRINT</u>	PRI
<u>PROMPT</u>	PRO
<u>QUIT</u>	-
<u>RECOVER</u>	-

<u>REMARK</u>	REM
<u>REPFOOTER</u>	REPF
<u>REPHEADER</u>	REPH
<u>RESERVED WORDS (SQL)</u>	-
<u>RESERVED WORDS (PL/SQL)</u>	-
<u>RUN</u>	R
<u>SAVE</u>	SAV
<u>SET</u>	-
<u>SHOW</u>	SHO
<u>SHUTDOWN</u>	-
<u>SPOOL</u>	SPO
<u>START</u>	-
<u>STARTUP</u>	-
<u>STORE</u>	-
<u>TIMING</u>	TIMI

<u>TTITLE</u>	TTI
<u>UNDEFINE</u>	UNDEF
<u>VARIABLE</u>	VAR
<u>WHENEVER OSERROR</u>	-
<u>WHENEVER SQLERROR</u>	-

## **SQL-Plus @ [at]**

@ Aufruf:

```
@ { uri | file_name[.ext] } [arg ...]
```

@ Beschreibung:

Mit dem @ [at] Befehl können Sie ein SQL Skript ausführen.

Das SQL Skript Das Skript kann vom lokalen Dateisystem oder von einem Webserver aufgerufen werden. In dem SQL Skript können alle in Oracle verwendeten SQL Statements und die in SQL-Plus möglichen SQL-Plus Befehle aufgerufen werden.

Ein SQL Skript kann SQL-Plus nach der Ausführung beenden, wenn in der letzten Zeile des Skripts ein "QUIT" eingegeben wird. Die Verwendung von Kommentaren ist auch möglich. Kommentare beginnen immer mit –, alle folgenden Anweisungen in der Zeile werden nicht ausgeführt.

Parameter können mit @dateiname arg1 arg2 ... angegeben werden. In dem ausgeführten SQL Skript werden die Parameter arg1, arg2, ... mit der Angabe von &1 &2 ... verwendet. Der @ [at] Befehl bietet viele Möglichkeiten, zur Verwendung von selbst definierten Makros und Shell Skripten die auf Daten einer Oracle Datenbank zugreifen.

Weiterführende Informationen zu dem @ [at] Befehl und zu den Möglichkeiten in SQL-Plus finden Sie auf den Webseiten von Oracle.

## **SQL-Plus @@ [at][at]**

@@ Aufruf:

```
@@ file_name[.ext]
```

@@ Beschreibung:

Mit dem @@ [at][at] Befehl können Sie eine SQL Skript aus einem anderen SQL-Skript heraus ausführen. Das mit @@ aufgerufene SQL Skript wird unter dem gleichen Pfad oder URI gesucht, in dem sich das aufrufende SQL-Skript befindet. Alles weitere ist identisch mit dem @ [at] Befehl.

Zusätzliche Infos zu dem @@ [at][at] Befehl finden Sie auf den Webseiten von Oracle.

## **SQL-Plus / [slash]**

/ Aufruf:

/

/ Beschreibung:

Das / [slash] Kommando führt das letzte auf der Kommandozeile eingegebene Skript noch einmal aus, ohne es auf der Konsole auszugeben. Der / [slash] Befehl ist sehr hilfreich wenn es darum geht, einen Befehl in SQL-Plus mehrmals auszuführen. Mehr zum / [slash] in den Online-Dokumentationen der Webseiten von Oracle.

## **SQL-Plus ACCEPT**

ACCEPT Aufruf:

```
ACCEPT variable [NUMBER|CHAR|DATE] [FORMAT format] [DEFAULT default]
[PROMPT text|NOPROMPT] [HIDE]
```

ACCEPT Beschreibung:

Liest eine Zeile aus dem Eingabepuffer (User Eingabe) und speichert sie in einer vorgegebenen Variable.

- **variable**  
Der Name der Variable, in der die Eingabe gespeichert wird. Eine existierende Variable wird überschreiben. Eine nicht existierende Variable wird von SQL-Plus angelegt.
- **NUMBER CHAR DATE**  
Festlegung des Datentyps. Wenn Die Eingabe nicht mit dem vorgegebenen Datentyp übereinstimmt, wird SQL-Plus einen Fehler ausgeben und wiederholt zur Eingabe auffordern. Die max. Länge für den Datentype CHAR ist 240 Zeichen.
- **FORMAT**  
Festlegung des Eingabeformats. Wenn Die Eingabe den Formatvorgaben nicht entspricht, wird ein Fehler ausgegeben und erneut zur Eingabe aufgefordert. Der Format String kann folgendermaßen aussehen:  
Z.B. 'dd/mm/yy', A10 oder 9.999.  
Die Liste der möglichen Formatvorgaben kann unter 'SQL-Plus Formate' nachgelesen werden.
- **DEFAULT**  
Der Default Wert, falls keine Eingabe gemacht wird.
- **PROMPT NOPROMPT**  
Mit PROMPT kann man vor der Eingabeaufforderung einen Text anzeigen lassen. Mit NOPROMPT wird kein Text ausgegeben.
- **HIDE**  
Anzeige der Eingaben unterdrücken. (z.B. für Passwörter)

Auf den Webseiten von Oracle finden Sie weiterführende Infos zum ACCEPT Befehl.

## **SQL-Plus APPEND**

APPEND Aufruf:

```
APPEND text
```

APPEND Beschreibung:

Hängt den mit 'text' angegebenen Text an das Ende der Zeile im SQL Puffer. Unter [www.Oracle.com](http://www.Oracle.com) finden Sie weitere Infos zum APPEND Befehl.

## **SQL-Plus ARCHIVE LOG**

ARCHIVE LOG Aufruf:

```
ARCHIVE LOG {LIST|STOP} | {START|NEXT|ALL|integer} [TO destination]
```

ARCHIVE LOG Beschreibung:

Starten oder Stoppen der automatischen Archivierung von "online redo log files", manuelles Archivieren von vorgegebenen "redo log files" und Anzeigen von Informationen über die "redo log files" ist mit dem ARCHIVE LOG Befehl möglich.

Unter [www.Oracle.com](http://www.Oracle.com) finden Sie immer die aktuellen Neuerungen zum ARCHIVE LOG Befehl.

## **SQL-Plus ATTRIBUTE**

ATTRIBUTE Aufruf:

```
ATTRIBUTE [type_name.attribute_name [option...]]
```

ATTRIBUTE Beschreibung:

Mit ATTRIBUTE kann man "Display" Attribute für ein Objektattribut definieren oder sich die Liste der definierten Attribute anzeigen lassen. Es ist auch möglich sich nur ein einziges Attribut anzeigen zu lassen.

Mögliche Options:

ALI[AS] alias

CLE[AR]

FOR[MAT]

format

LIKE {type\_name.attribute\_name|alias}

ON|OFF

## **SQL-Plus BREAK**

BREAK Aufruf:

```
BREAK [ON report_element [action [action]]] ...
```

BREAK Beschreibung:

Mit dem BREAK Befehl kann man die Formatierung eines Reports in SQL-Plus bestimmen. Ohne Angabe von Parametern wird nur die Aktuelle BREAK Definition ausgegeben. Das "report\_element"

hat folgende Syntax:

{column|expr|ROW|REPORT}

Die "action" hat folgende Syntax:

[SKI[P] n|[SKI[P]]

PAGE] [NODUP[LICATES]]|DUP[LICATES]]

Die vollständige Beschreibung des BREAK Befehls entnehmen Sie bitte dem Dokument: "SQL\*Plus User's Guide and Reference" von Oracle, das auch als PDF bei Oracle erhältlich ist.

"SQL\*Plus User's Guide and Reference"

## **SQL-Plus BTITLE**

BTITLE Aufruf:

BTI[TLE] [printspec [text|variable] ...] [[ON|OFF]

BTITLE Beschreibung:

Dient zur Festlegung des Formates der Titel am Ende jedes Reports. Mit BTITLE kann man sich auch die Liste der BTITLE Definitionen anzeigen lassen.

## **SQL-Plus CHANGE**

CHANGE Aufruf:

C[HANGE] sepchar old [sepchar [new [sepchar]]]

CHANGE Beschreibung:

Mit CHANGE ist es möglich den Text einer Zeile des SQL Puffers zu ändern. Es ist möglich jedes Zeichen als "sepchar" zu verwenden, wenn es sich nicht um einen Buchstaben oder eine Zahl handelt.

## **SQL-Plus CLEAR**

CLEAR Aufruf:

CL[EAR] option ...

CLEAR Beschreibung:

Löschen oder Rücksetzen des Wertes oder einer Einstellung für eine vorgegebene Option:

Die "option" ist Wert aus der folgenden Liste

BRE[AKS]

BUFF[ER]

COL[UMNS]

COMP[UTES]

SCR[EEN]

SQL

TIMI[NG]

## **SQL-Plus COLUMN**

COLUMN Aufruf:

OL[UMN] [{column|expr} [option ...]]

COLUMN Beschreibung:

Mit COLUMN ist es möglich die Eigenschaften für die Anzeige einer vorgegebenen "column" zu ändern. Es sind folgende Datentypen möglich: NCHAR, VARCHAR2 (VARCHAR), NVARCHAR2 (NCHAR VARYING), LONG, CLOB, NCLOB und NUMBER

Die "option" ist einer der folgenden Werte:

- ALI[AS] alias
- CLE[AR]
- ENTMAP {ON|OFF}
- FFOLD\_A[FTER]
- FOLD\_B[EFORE]
- FOR[MAT] format
- HEA[DING] text
- JUSTIFY {L[EFT]|C[ENTER]|C[ENTRE]|R[IGHT]}
- LIKE {expr|alias}
- NEWL[INE]
- NEW\_V[ALUE] variable
- NOPRI[NT]|PRI[NT]
- NUL[L] text
- OLD\_V[ALUE] variable
- ON|OFF
- WRA[PPED]|WOR[D\_WRAPPED]|TRU[NCATED]

Die Liste der möglichen Formate kann unter 'SQL-Plus Formate' nachgelesen werden.

## **SQL-Plus COMPUTE**

COMPUTE Aufruf:

```
COMP[UTE] [function [LAB[EL] text] ... OF {expr|column|alias} ...ON  
{expr|column|alias|REPORT|ROW} ...]
```

COMPUTE Beschreibung:

Berechnet und zeigt Werte in einer Zeile, die mit diversen Funktionen berechnet werden können.

Funktionen des COMPUTE Aufrufs

AVG	Durchschnittswerte	NUMBER
COU[NT]	Null Werte zählen	Alle Datentypen
MIN[IMUM]	Minimum	NUMBER, CHAR, NCHAR, VARCHAR2 (VARCHAR), NVARCHAR2 (NCHAR VARYING)
MAX[IMUM]	Maximum	NUMBER, CHAR, NCHAR, VARCHAR2 (VARCHAR), NVARCHAR2 (NCHAR VARYING)

NUM[BER]	Anzahl der Zeilen	Alle Datentypen
SUM	Summe aller Werte <> null	NUMBER
STD	Standard der Werte <> null	NUMBER
VAR[IANCE]	Abweichungen der Werte <> null	NUMBER

## **SQL-Plus CONNECT**

CONNECT Aufruf:

```
CONN[ECT] [{logon|/}] [AS {SYSOPER|SYSDBA}]]
```

CONNECT Beschreibung:

Verbindung zu einer Oracle Instanz aufnehmen. Wenn kein login angegeben wird, fragt SQL-Plus nach einem Login.

Das "login" hat die folgende Syntax:

```
username[/password][@connect_identifier]
```

## **SQL-Plus COPY**

COPY Aufruf:

```
COPY {FROM database | TO database | FROM database TO database}
{APPEND|CREATE|INSERT|REPLACE} destination_table[(column, column, column,
...)] USING query
```

COPY Beschreibung:

Mit COPY kopiert man Daten von einer Datenbank zu einer anderen. Mit den Parametern APPEND, CREATE, INSERT oder REPLACE kann man festlegen, wie die Daten kopiert werden sollen.

## **SQL-Plus DEFINE**

DEFINE Aufruf:

```
DEF[INE] [variable][[variable = text]
```

DEFINE Beschreibung:

Setzt eine Benutzerdefinierte Variable auf einen Wert (CHAR) Der Aufruf von DEFINE ohne Parameter listet die definierten Variablen auf.

## **SQL-Plus DEL**

DEL Aufruf:

```
DEL [n|n m|n *|n LAST|*|* n|* LAST|LAST]
```

DEL Beschreibung:

Löscht eine oder mehrere Zeilen im SQL Puffer. Mit \* kann man die aktuelle Zeile löschen. Die Eingabe von DEL ohne Parameter löscht die aktuelle Zeile aus dem SQL Puffer.

## **SQL-Plus DESCRIBE**

DESCRIBE Aufruf:

```
DESC[RIBE] [schema.]object[@connect_identifizier]
```

DESCRIBE Beschreibung:

Mit DESCRIBE kann man sich die Spaltendefinitionen der Tabelle, des Views oder sonst. Objektes anzeigen lassen. Im Falle einer Funktion oder einer Procedure wird die Spezifikation der Funktion/Procedure angezeigt.

## **SQL-Plus DISCONNECT**

DISCONNECT Aufruf:

```
DISC[ONNECT]
```

DISCONNECT Beschreibung:

Logout des aktuellen Users. SQL-Plus wird nicht beendet.

**Hinweis: Beim Logout wird ein COMMIT an die Datenbank gesendet**

## **SQL-Plus EDIT**

EDIT Aufruf:

```
ED[IT] [file_name[.ext]]
```

EDIT Beschreibung:

Mit EDIT wird eine Text Editor geöffnet, in dem eine Datei oder der Inhalt des SQL Puffers bearbeitet werden kann. Wenn keine Datei angegeben wird, nimmt SQL-Plus automatisch den SQL Puffer.

## **SQL-Plus EXECUTE**

EXECUTE Aufruf:

```
EXEC[UTE] statement
```

EXECUTE Beschreibung:

Ausführen eines PL/SQL Statements, oder ausführen einer Stored Procedure. Der Return Wert einer Stored Procedure wird in der Bind-Variable :ID gespeichert. Informationen zu Bind-Variablen gibt es in der Doku des VARIABLE Befehls.

## **SQL-Plus EXIT**

EXIT Aufruf:

```
{EXIT|QUIT} [SUCCESS|FAILURE|WARNING|n|variable |:BindVariable]
[COMMIT|ROLLBACK]
```

EXIT Beschreibung:

Der Befehl EXIT beendet SPL-Plus. Ohne Angabe weiterer Parameter wird ein COMMIT ausgeführt. Mit SUCCESS|FAILURE|WARNING|... kann man einen Returncode festlegen.

## **SQL-Plus GET**

GET Aufruf:

```
GET file_name[.ext] [LIST|NOLIST]
```

GET Beschreibung:

Mit GET wird eine Datei aus dem Dateisystem in den SQL Puffer geladen.

## **SQL-Plus HELP**

HELP Aufruf:

```
HELP [topic]
```

HELP Beschreibung:

Mit HELP kann man sich die Liste der SQL Plus Befehle anzeigen lassen oder konkrete Hilfe zu einen SQL-Plus Befehl erhalten.

## **SQL-Plus HOST**

HOST Aufruf:

```
HO[ST] [command]
```

HOST Beschreibung:

HOST command führt den Befehl "command" im System aus, ohne SQL-Plus zu verlassen.

HOST Beispiel:

HOST ls -altr => zeigt alle Dateien in dem Verzeichnis sortiert nach Datum (Neue Dateien zuletzt)

## **SQL-Plus INPUT**

INPUT Aufruf:

```
I[NPUT] [text]
```

INPUT Beschreibung:

Fügt eine oder mehrere Zeilen Text hinter der aktuellen Zeile im SQL Puffer ein.

## SQL-Plus LIST

LIST Aufruf:

```
L[IST] [n|n m|n *|n LAST|*|* n|* LAST|LAST]
```

LIST Beschreibung:

Mit dem LIST Befehl kann man sich eine oder mehrere Zeilen des SQL Puffers anzeigen lassen. Mit LIST ohne Angabe von Optionen werden alle Zeilen des SQL Puffers angezeigt.

n	Zeile n anzeigen.
n m	Die Zeilen von n bis m anzeigen.
n *	Die Zeilen von n bis zur aktuellen Zeile anzeigen
n LAST	Die Zeilen von n bis zur letzten Zeile anzeigen
*	Aktuelle Zeile anzeigen.
* n	Von der aktuellen Zeile bis zur Zeile n anzeigen.
* LAST	Von der aktuellen Zeile bis zur letzten Zeile anzeigen
LAST	Die letzte Zeile anzeigen.

## SQL-Plus PASSWORD

PASSWORD Aufruf:

```
PASSW[ORD] [username]
```

PASSWORD Beschreibung:

Ermöglicht es, dass Passwort zu ändern ohne, dass es auf der Konsole angezeigt wird.

## SQL-Plus PAUSE

PAUSE Aufruf:

```
PAU[SE] [text]
```

PAUSE Beschreibung:

Ausgabe einer Leerzeile gefolgt von einem Text der als Option dem Befehl PAUSE übergeben werden kann. Nach der Ausgabe wartet SQL-Plus auf die Eingabe von [Return] durch den User. Wird kein "text" übergeben, wird nur eine Leerzeile ausgegeben.

## **SQL-Plus PRINT**

PRINT Aufruf:

```
PRI[NT] [variable ...]
```

PRINT Beschreibung:

Zeigt den Wert einer Variable an.

PRINT Beispiel:

SQL>	VARIABLE i NUMBER
SQL>	BEGIN :i := 4711; END;
SQL>	PRINT i i ----- 4711

## **SQL-Plus PROMPT**

PROMPT Aufruf:

```
PROMPT [text]
```

PROMPT Beschreibung:

Zeigt den Text auf dem Display des Nutzers an. Wenn kein Text angegeben wird, wird nur eine Leerzeile ausgegeben.

## **SQL-Plus QUIT**

QUIT Beschreibung:

Weitere Infos zu QUIT sind unter dem SQL-Plus Befehl EXIT zu finden.

## **SQL-Plus RECOVER**

RECOVER Aufruf:

```
RECOVER {general | managed | END BACKUP}
```

RECOVER Optionen: Die Option 'general' hat folgende Syntax:

[AUTOMATIC] [FROM location] { {full\_database\_recovery | partial\_database\_recovery | LOGFILE filename} [ {TEST | ALLOW integer CORRUPTION } [TEST | ALLOW integer CORRUPTION ]...] |CONTINUE [DEFAULT]|CANCEL}

Die Option 'full\_database\_recovery' hat folgende Syntax:

[STANDBY] DATABASE [ {UNTIL {CANCEL | TIME date | CHANGE integer} | USING BACKUP CONTROLFILE} [UNTIL {CANCEL | TIME date | CHANGE integer} | USING BACKUP CONTROLFILE]...]

Die Option 'partial\_database\_recovery' hat folgende Syntax:

{TABLESPACE tablespace [, tablespace]... | DATAFILE datafilename [, datafilename]... | STANDBY {TABLESPACE tablespace [, tablespace]... | DATAFILE datafilename [, datafilename]...} UNTIL [CONSISTENT] [WITH] CONTROLFILE }

Die Option 'managed' hat folgende Syntax:

MANAGED STANDBY DATABASE [ {NODELAY | [TIMEOUT] integer | CANCEL [IMMEDIATE] [NOWAIT]} | [DISCONNECT [FROM SESSION] ] [FINISH [NOWAIT] ] ]

RECOVER Beschreibung:

Führt ein Recovery von einem oder mehreren Tablespaces, einem oder mehreren Datafiles oder der ganzen Datenbank durch.

## **SQL-Plus REMARK**

REMARK Aufruf:

REM[ARK] [text]

REMARK Beschreibung:

Mit REM kann man einen Kommentar kennzeichnen. Zeilen die mit REM[ARK] beginnen werden von SQL-Plus nicht interpretiert. Es ist richtig, das REM am Anfang einer Zeile steht. Es sich nicht möglich Teile einer Zeile mit REM als einen Kommentar zu kennzeichnen.

## **SQL-Plus REPFOOTER**

REPFOOTER Aufruf:

REP[FOOTER] [PAGE] [printspec [text|variable] ...] | [OFF|ON]

REPFOOTER Beschreibung:

Mit REPFOOTER definiert man einen Footer am Ende eines jeden Reports. Mit den Anweisungen in 'printspec' können hier die Vorgaben ausformuliert werden. Bei einem Aufruf von REPFOOTER ohne Parameter, wird die aktuelle Definition des Footers angezeigt.

**printspec:**

COL n

S[KIP] [n]

TAB n

LE[FT]

CE[NTER]

R[IGHT]

BOLD  
FORMAT text

## **SQL-Plus REPHEADER**

REPHEADER Aufruf:

REPH[EADER] [PAGE] [printspec [text|variable] ...] | [OFF|ON]

REPHEADER Beschreibung:

Mit REPHEADER definiert man einen Header am Anfang des Reports. Mit den Anweisungen in 'printspec' können hier die Vorgaben für den Header ausformuliert werden.

Bei einem Aufruf von REPHEADER ohne Parameter, wird die aktuelle Definition des HEADERS angezeigt.

Folgende Möglichkeiten gibt es den Parameter 'printspec' zu definieren.

**printspec:**

COL n

S[KIP] [n]

TAB n

LE[FT]

CE[NTER]

R[IGHT]

BOLD

FORMAT text

## **SQL-Plus RESERVED WORDS (SQL)**

Die reservierten SQL Befehle haben eine spezielle Bedeutung und dürfen nicht für andere Zwecke verwendet werden. Die folgende Liste zeigt die reservierten Wörter in SQL:

ACCESS	DEFAULT	INTEGER	ONLINE	START
ADD	DELETE	INTERSECT	OPTION	SUCCESSFUL
ALL	DESC	INTO	OR	SYNONYM
ALTER	DISTINCT	IS	ORDER	SYSDATE
AND	DROP	LEVEL	PCTFREE	TABLE
ANY	ELSE	LIKE	PRIOR	THEN
AS	EXCLUSIVE	LOCK	PRIVILEGES	TO
ASC	EXISTS	LONG	PUBLIC	TRIGGER
AUDIT	FILE	MAXEXTENTS	RAW	UID
BETWEEN	FLOAT	MINUS	RENAME	UNION
BY	FOR	MLSLABEL	RESOURCE	UNIQUE
CHAR	FROM	MODE	REVOKE	UPDATE
CHECK	GRANT	MODIFY	ROW	USER
CLUSTER	GROUP	NOAUDIT	ROWID	VALIDATE
COLUMN	HAVING	NOCOMPRESS	ROWNUM	VALUES

COMMENT	IDENTIFIED	NOT	ROWS	VARCHAR
COMPRESS	IMMEDIATE	NOWAIT	SELECT	VARCHAR2
CONNECT	IN	NULL	SESSION	VIEW
CREATE	INCREMENT	NUMBER	SET	WHENEVER
CURRENT	INDEX	OF	SHARE	WHERE
DATE	INITIAL	OFFLINE	SIZE	WITH
DECIMAL	INSERT	ON	SMALLINT	

## SQL-Plus RESERVED WORDS (PL/SQL)

Die reservierten Wörter in PL/SQL haben eine spezielle Bedeutung und dürfen nicht für andere Zwecke verwendet werden. Die folgende Liste zeigt die reservierten Wörter in PL/SQL:

ALL	DESC	JAVA	PACKAGE	SUBTYPE
ALTER	DISTINCT	LEVEL	PARTITION	SUCCESSFUL
AND	DO	LIKE	PCTFREE	SUM
ANY	DROP	LIMITED	PLS_INTEGER	SYNONYM
ARRAY	ELSE	LOCK	POSITIVE	SYSDATE
AS	ELSIF	LONG	POSITIVEN	TABLE
ASC	END	LOOP	PRAGMA	THEN
AT	EXCEPTION	MAX	PRIOR	TIME
AUTHID	EXCLUSIVE	MIN	PRIVATE	TIMESTAMP
AVG	EXECUTE	MINUS	PROCEDURE	TIMEZONE_ABBR
BEGIN	EXISTS	MINUTE	PUBLIC	TIMEZONE_HOUR
BETWEEN	EXIT	MLSLABEL	RAISE	TIMEZONE_MINUTE
BINARY_INTEGER	EXTENDS	MOD	RANGE	TIMEZONE_REGION

BODY	EXTRACT	MODE	RAW	TO
BOOLEAN	FALSE	MONTH	REAL	TRIGGER
BULK	FETCH	NATURAL	RECORD	TRUE
BY	FLOAT	NATURALN	REF	TYPE
CHAR	FOR	NEW	RELEASE	UI
CHAR_BASE	FORALL	NEXTVAL	RETURN	UNION
CHECK	FROM	NOCOPY	REVERSE	UNIQUE
CLOSE	FUNCTION	NOT	ROLLBACK	UPDATE
CLUSTER	GOTO	NOWAIT	ROW	USE
COALESCE	GROUP	NULL	ROWID	USER
COLLECT	HAVING	NULLIF	ROWNUM	VALIDATE
COMMENT	HEAP	NUMBER	ROWTYPE	VALUES
COMMIT	HOUR	NUMBER_BASE	SAVEPOINT	VARCHAR
COMPRESS	IF	OCIROWID	SECOND	VARCHAR2
CONNECT	IMMEDIATE	OF	SELECT	VARIANCE
CONSTANT	IN	ON	SEPERATE	VIEW

CREATE	INDEX	OPAQUE	SET	WHEN
CURRENT	INDICATOR	OPEN	SHARE	WHENEVER
CURRVAL	INSERT	OPERATOR	SMALLINT	WHERE
CURSOR	INTEGER	OPTION	SPACE	WHILE
DATE	INTERFACE	OR	SQL	WITH
DAY	INTERSECT	ORDER	SQLCODE	WORK
DECIMAL	INTERVAL	ORGANIZATION	SQLERRM	WRITE
DECLARE	INTO	OTHERS	START	YEAR
DEFAULT	IS	OUT	STDDEV	ZONE
DELETE	ISOLATION			

## **SQL-Plus RUN**

RUN Aufruf:

R[UN]

RUN Beschreibung:

Führt den Befehl aus, der sich momentan im SQL Puffer befindet. Hierbei ist es egal, ob es sich um einen SQL oder um einen SQL-Plus Befehl handelt. Vor der Ausführung wird der Befehl noch einmal angezeigt.

## **SQL-Plus SAVE**

SAVE Aufruf:

```
SAV[E] file_name[.ext] [CRE[ATE]|REP[LACE]|APP[END]]
```

SAVE Beschreibung:

Speichert den Inhalt des SQL Puffers im Filesystem des Betriebssystems ab.

Mit 'file\_name[.ext]' gibt man den Dateinamen an, in dem der SQL Puffer gespeichert werden soll.

Mit der Option CRE[ATE] kann man SQL-Plus anweisen die Datei zu erzeugen, falls sie noch nicht existiert.

Mit der Option REP[LACE] kann man SQL-Plus anweisen die Datei bei Bedarf zu überschreiben.

Um den SQL Puffer an eine existierende Datei anzuhängen kann man die Option APP[END] verwenden.

## **SQL-Plus SET**

SET Aufruf:

```
SET system_variable value
```

SET Beschreibung:

Mit SET kann man in SQL-Plus eine Systemvariable setzen.

Der SET Befehl bietet eine sehr große Auswahl an Möglichkeiten. Details können in dem SQL-Plus Handbuch "SQL\*Plus Users Guide and Reference" von Oracle nachgelesen werden. Dieses Buch gibt es auch als PDF direkt von Oracle. Sie können es bei Bedarf bei Ihrem Oracle Partner anfordern.

## **SQL-Plus SHOW**

SHOW Aufruf:

```
SHO[W] option
```

SHOW Option:

system\_variable  
ALL  
BTI[TLE]  
ERR[ORS] [ {FUNCTION |  
PROCEDURE | PACKAGE | PACKAGE BODY | TRIGGER | VIEW | TYPE | TYPE  
BODY | DIMENSION | JAVA CLASS} [schema.]name]  
LABEL  
LNO  
PNO

REL[EASE]  
REPF[OOTER]  
REPH[EADER]  
PARAMETERS  
[parameter\_name]  
SPOO[L]  
SQLCODE  
TTI[TLE]  
USER

SHOW Beschreibung:

Zeigt unterschiedlichste Informationen an. Es können mit SHOW z.B. Systemvariablen, Fehler von Funktionen oder auch verschiedene Definitionen an. Mehr Infos zu dem SHOW Befehl gibt es in der SQL-Plus Dokumentation "*SQL \*Plus Quick Reference*".

## **SQL-Plus SHUTDOWN**

SHUTDOWN Aufruf:

SHUTDOWN [ABORT | IMMEDIATE | NORMAL | TRANSACTIONAL [LOCAL]]

SHUTDOWN Beschreibung:

Beendet die aktuelle Oracle Instanz. Optional kann zusätzlich die Datenbank geschlossen werden.

**Hinweis: Man benötigt die entsprechenden Rechte um die Oracle Instanz zu beenden**

## **SQL-Plus SPOOL**

SPOOL Aufruf:

SPO[OL] [file\_name[.ext] | OFF | OUT]

SPOOL Beschreibung:

Mit SPOOL kann man alle Eingaben und Ausgaben von SQL-Plus in einer Datei protokollieren und optional auch an einen Drucker schicken.

SPOOL Options:

file_name[.ext]	Mit file_name gibt man die Datei an, in der die Spool-Daten gespeichert werden sollen.
OFF	Mit OFF wird das SPOOLING abgebrochen, ohne den Inhalt in eine Datei zu schreiben.
OUT	Mit OUT wird das SPOOLING beendet und der Inhalt in eine Datei geschrieben.

## SQL-Plus START

START Aufruf:

```
STA[RT] file_name[.ext] [arg ...]
```

START Beschreibung:

Führt die SQL-Plus Befehle in der mit 'file\_name' benannten Datei aus. In der Datei können SQL-Plus und SQL Befehle enthalten sein. Mit den optionalen Parametern arg1 bis argn können der "SQL Skript" noch Argumente übergeben werden.

## SQL-Plus STARTUP

STARTUP Aufruf:

```
STARTUP [FORCE] [RESTRICT] [PFILE=filename] [MOUNT [dbname] | OPEN  
[open_options] [dbname] | NOMOUNT]
```

STARTUP Beschreibung:

Mit dem SQL-Plus Befehl STARTUP kann man eine Oracle Instanz starten und optional auch eine Datenbank mounten und öffnen.

Die möglichen 'open\_options':

```
READ {ONLY | WRITE [RECOVER]} | RECOVER
```

## SQL-Plus STORE

STORE Aufruf:

```
STORE SET file_name[.ext] [CRE[ATE]|REP[LACE]| APP[END]]
```

STORE Beschreibung:

Mit STORE hat man die Möglichkeit, Eigenschaften der aktuellen SQL-Plus Umgebung in einer Datei zu speichern, mit der man später die Eigenschaften der Umgebung wieder reproduzieren kann.

Mit den Parametern APPEND, CREATE, INSERT oder REPLACE kann man festlegen, wie die Daten gespeichert werden sollen.

## **SQL-Plus TIMING**

TIMING Aufruf:

```
TIMI[NG] [START text|SHOW|STOP]
```

TIMING Beschreibung:

Mit dem TIMING Befehl kann man unter SQL-Plus das Timing von Befehlen protokollieren.

START text	Option zum Starten eines Timers. Mit dem 'text' kann man dem Timer einen Namen geben.
SHOW	Zeigt den aktuellen Timer mit den zugehörigen Daten an.
STOP	Zeigt den aktuellen Timer mit den zugehörigen Daten an und löscht den Timer. Falls noch andere Timer laufen, wird der vorher gestartete Timer zum aktuellen Timer.

## **SQL-Plus TTITLE**

TTITLE Aufruf:

```
TTI[TLE] [printspec [text|variable] ...][[OFF|ON]
```

TTITLE Beschreibung:

Mit TTITLE kann man in SQL-Plus einen Titel am Anfang eines jeden Report festlegen. Hierbei kann der Titel und die Formatierung des Titels festgelegt werden.

Mögliche Angaben für 'printspec':

COL	n
S[KIP]	[n]
TAB	n
LE[FT]	
CE[NTER]	
R[IGHT]	
BOLD	
FORMAT text	

## **SQL-Plus UNDEFINE**

UNDEFINE Aufruf:

```
UNDEF[INE] variable ...
```

UNDEFINE Beschreibung:

Löschen einer oder mehrere Benutzervariablen.

## SQL-Plus VARIABLE

VARIABLE Aufruf:

```
VAR[IABLE] [variable [NUMBER|CHAR|CHAR (n)|VARCHAR2 (n) | REFCURSOR]]
```

VARIABLE Beschreibung:

Mit VARIABLE kann man eine Variable definieren, die aus dem PL/SQL Code heraus aufgerufen werden kann.

Ausführliche Infos zu Variablen in PL/SQL findet man im "PL/SQL Users Guide and Reference".

variable	Variablenname festlegen, unter dem Namen ist die Variable aus dem PL/SQL Code heraus aufrufbar.
NUMBER	Eine Variable vom Type NUMBER mit einer festen Länge.
CHAR	Eine Variable vom Type CHAR mit einer festen Länge von einem Zeichen.
CHAR (n)	Eine Variable vom Type CHAR mit einer maximalen Länge von 255 Zeichen.
VARCHAR2 (n)	Eine Variable vom Type VARCHAR2 mit einer maximalen Länge von 2000 Zeichen.
REFCURSOR	Eine Variable vom Type REF CURSOR.

## SQL-Plus WHENEVER OSERROR

WHENEVER OSERROR Aufruf:

```
WHENEVER OSERROR {EXIT [SUCCESS|FAILURE |n|variable|:BindVariable]  
[COMMIT|ROLLBACK]|CONTINUE [COMMIT|ROLLBACK|NONE]}
```

WHENEVER OSERROR Beschreibung:

Mit WHENEVER OSERROR kann man ein Fehlerhandling im Falle eines Fehlers vom Betriebssystem definieren. Es ist z.B. möglich, im Falle eines I/O Fehlers ein ROLLBACK durchzuführen.

## SQL-Plus WHENEVER SQLERROR

WHENEVER SQLERROR Aufruf:

```
WHENEVER SQLERROR {EXIT [SUCCESS|FAILURE|WARNING |n|variable|:BindVariable]  
[COMMIT|ROLLBACK]|CONTINUE [COMMIT|ROLLBACK|NONE]}
```

WHENEVER SQLERROR Beschreibung:

Beendet SQL-Plus, im Falle eines fehlerhaften SQL Befehls oder eines Fehlers in einem PL/SQL Block.

## SQL-Plus Formatstrings

Beschreibung:

In der Tabelle stehen die in SQL-Plus gültigen Bezeichner für einen Formatstring der mit FOR[MAT] *format* angegeben werden kann.

Element	Beispiel	Beschreibung
9	999	Definiert die max. Länge einer Nummer die eingegeben werden kann.
0	0999 9990	Anzeige einer führenden Null oder einer Null statt einem Leerzeichen an der vorgegebenen Stelle.
\$	\$9999	Mit dem Dollar Zeichen wird dem Wert ein \$ Zeichen vorangestellt.
B	B9999	Zeigt eine Leerzeichen statt einer 0.
MI	9999MI	Zeigt eine Minus (-) hinter einem negativen Wert. Wenn der Wert positiv ist, wird ein Leerzeichen angezeigt.
S	S999	Zeigt ein "+" bei positiven und ein "-" bei negativen Werten an.
PR	9999PR	Zeigt einen Negativen Wert in spitzen Klammern. Positive Werte werden mit umhüllenden Leerzeichen dargestellt.
D	99D99	An der Stelle wird ein Dezimalzeichen angezeigt. Es trennt den ganzzahligen Wert von dem Nachkommateil.

G	9G999	“Group Separator” an dieser Position anzeigen.
C	C9999	Anzeige der ISO Währung am Anfang der Zahl.
L	L9999	Anzeige der lokalen Währung am Anfang der Zahl.
, (Komma)	99,99	Zeigt ein Komma an der vorgegebenen Position der Zahl.
. (Punkt)	99.99	Zeigt einen Punkt an der vorgegebenen Position der Zahl.
EEEE	9999EEEE	Wissenschaftliche Notation der Zahl.
RN/rn	RN/rn	Zeigt die Römische Darstellung der Zahl jeweils in Groß- oder in Kleinbuchstaben.
DATE	DATE	Wert als Datum interpretiert.

Oracle is a registered trademark of Oracle Corporation.