

En Kort Introduktion til Oracle

Henrik Bulskov
12. februar 2001
bulskov@ruc.dk

| | | |
|-----|---|---|
| 1 | Start SQL*Plus..... | 1 |
| 1.1 | TELNET..... | 1 |
| 1.2 | WINDOWS SQL PLUS..... | 2 |
| 2 | Kør et SQL-script..... | 3 |
| 3 | Hjælp i SQL*Plus..... | 3 |
| 4 | Editering af kommando..... | 3 |
| 4.1 | EDITERING UNDER WINDOWS..... | 5 |
| 5 | Opret Tabel..... | 5 |
| 6 | Indsæt rækker..... | 6 |
| 7 | Slet Rækker..... | 6 |
| 8 | Drop Tabel..... | 7 |
| 9 | Beskrivelse af elementer i databasen..... | 7 |
| 10 | Spool output til fil..... | 7 |
| 11 | Antallet af rækker i en tabel..... | 8 |
| 12 | Konstanter i forespørgslerne..... | 8 |
| 13 | ROWNUM..... | 8 |
| 14 | Tabellen DUAL..... | 9 |

1 Start SQL*Plus

SQLPlus er en klient program til eksekvering af SQL og PL/SQL kommandoer på en Oracle database server.

Der er flere forskellige måder hvormed en forbindelse til en Oracle database server kan etableres via SQLPlus. Her gives to eksempler der kan benyttes fra en eller flere af de platforme vi har på datalogi.

1.1 Telnet

Oracle database serveren kører på en unix maskine og vil derfor kunne nås via telnet fra en hvilken som helst maskine. Først åbnes en telnet forbindelse til homer¹ og herefter startes SQL*Plus med parametrene

1. sqlplus <username>/<password>@database

eller

2. sqlplus

¹ Hvis man skal i forbindelse med Homer udefra (altså en computer der ikke har et navn der ender med "ruc.dk"), skal man benytte SSH.

(i 2. vil programmet starte med at spørge efter brugernavn og password)

Eksempel:

```
sqlplus scott/tiger@modul2
```

eller

```
sqlplus
```

```
Enter user-name: scott/tiger@modul2
```

(NB: brugernavn, password og database skal indtastes på en gang selvom der kun spørges efter brugernavn, ellers virker det ikke.)

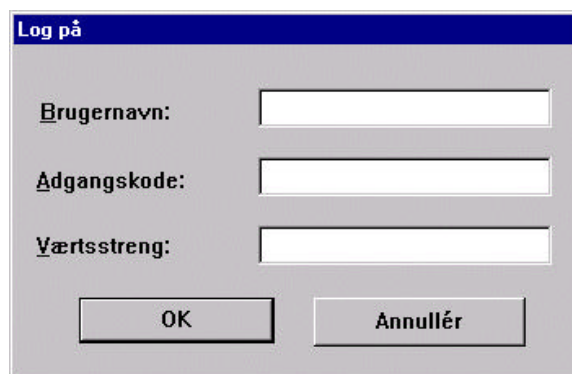
herefter fremkommer SQL*Plus prompten

```
SQL>
```

1.2 Windows SQL Plus

Oracle tilbyder en windows grænseflade til SQLPlus. Den startes ved at vælge start-menu, kurser, database og SQLPlus.

Når programmet startes fremkommer en log-på dialog



The image shows a Windows-style dialog box titled "Log på". It contains three text input fields stacked vertically. The first field is labeled "Brugernavn:", the second "Adgangskode:", and the third "Værtsstreng:". Below the input fields are two buttons: "OK" on the left and "Annullér" on the right.

hvor der indtastes

Brugernavn: brugernavn

Adgangskode: password

Værtsstreng: modul2.

Herefter fremkommer SQL prompten

```
SQL>
```

2 Kør et SQL-script

Troels har lavet et SQL-script, der ligger i kataloget(på homer):

```
/projects/modul2/
```

Et script køres vha. kommandoen @ (snabel-a) eller med kommandoen START.
Opret f.eks. spj-databasen således:

```
SQL> @ /projects/modul2/spj
```

eller

```
SQL> start /projects/modul2/spj
```

programmet går udfra at filen har suffix .sql, hvis ikke andet er angivet.

3 Hjælp i SQL*Plus

Oracles manualer til SQL*Plus, SQL og PL/SQL er tilgængelige online på

```
http://otn.oracle.com/docs/
```

under "Related Features & Product Documentation" vælges **Oracle8i** og her vælges General Documentation, Release 3 (8.1.7).

For at benytte online dokumentationen er det nødvendigt at registrere sig i Oracle Technology Network. (Det er gratis).

4 Editering af kommando

Du kan rette i den kommando, der er i SQL*Plus's buffer. Følgende kommandoer er gode at kende i den forbindelse:

- ?? edit [<filename>] - starter en editor med angivne fil, default er bufferens indhold
- ?? save <filename> - gemmer bufferens indhold i den angivne fil (der tilføjes automatisk suffix .sql)
- ?? list - viser bufferens indhold
- ?? list n - vis n'te linie i bufferen
- ?? list n m - vis n'te til m'te line af bufferen
- ?? input - indsæt ny linie(r) i bufferen (afslut med to "return")
- ?? change /from/to - ændre strengen <from> til strengen <to>

?? append streng - tilføj til enden af linien
?? del - slet linie i bufferen

Eksempel(NB: * viser aktive linie):

```
SQL> list
  1 create table tmp(
  2 i int,
  3 streng varhar2(123),
  4* x number(2,4)
```

```
SQL> list 2
  2* i int,
```

```
SQL> list 2 4
  2 i int,
  3 streng varhar2(123),
  4* x number(2,4)
```

Ret linie 3 "varhar2" til "varchar2"

```
SQL> list 3
  3* streng varhar2(123),
```

```
SQL>change /varhar2/varchar2
  3* streng varchar2(123),
```

Indsæt ny attribut i linie 4

```
SQL> list 3
  3* streng varchar2(123),
```

```
SQL> input dato data,
SQL>
```

```
SQL> list
  1 create table tmp(
  2 i int,
  3 varrchar2(123),
  4 dato data,
  5* x number(2,4)
```

Tilføj tekst til linie

```
SQL> list 2
  2* i int,
```

```
SQL> append primary key
```

2* i int,primary key,

Fjern kommaet mellem int og primary

```
SQL> change /t,p/t p  
2* i int primary key,
```

```
SQL> list  
1 create table tmp(  
2 i int primary key,  
3 varchar2(123),  
4 dato data,  
5* x number(2,4))
```

Når kommandoen "edit" bruges startes default host editoren hvilket bestemmes af din UNIX opsætning(filen: .bashrc+). For at få hjælp til hvordan editoren bruges skrives

```
man vi
```

for vi-editoren eller

```
man joe
```

for joe-editoren på kommandolinien i UNIX eller med kommandoen

```
SQL> host <host kommando>
```

i SQL*Plus. <host>-kommandoen kan bruges til alle UNIX kommandoer fx

```
SQL> host ls
```

for en liste af filer i det aktive katalog.

(Hint: Editoren vi afsluttes med :q! og joe med ctrl-k ctrl-x efter hinanden)

4.1 Editering under windows

På Windows vil kommandoen "edit" i SQLPlus aktivere editoren notepad. Kommandoen kan herefter editeres og når notepad lukkes vil den overføre den nye tekst til SQLPlus. Dette kræver naturligvis at de rettelser der laves gemmes i notepad. Man skal være opmærksom på at SQLPlus ikke kan benyttes når man er i gang med at editere i notepad.

(Hint: Lav rettelserne i notepad og luk den. Herefter vil den spørge om du ville gemme eventuelle ændringer, og vende tilbage til SQLPlus)

5 Opret Tabel

En tabel oprettes på følgende måde

```
CREATE TABLE <table_name> ( <Liste af attributter> );
```

Se spj.sql for eksempler.

6 Indsæt rækker

Når der er oprettet en eller flere tabeller indsætte nye rækker ved

```
INSERT INTO <table_name> VALUES( <Liste af værdier> );
```

Antallet af værdier i listen og antallet af attributter i tabellen skal være ens og data typerne skal også stemme overens. (eller konverteres eksternt i INSERT kommandoen).

Se spj.sql for eksempler.

7 Slet Rækker

Rækker i tabeller fjernes med

```
DELETE <table_name> WHERE <betingelse>;
```

skal alle rækker fjernes bruges

```
DELETE <table_name>;
```

eller

```
TRUNCATE TABLE <table_name>;
```

DELETE og INSERT kommandoer skal bekræftes med

```
COMMIT;
```

men har inden dette mulighed for at fortryde med

```
ROLLBACK;
```

ROLLBACK sætter sessionen tilbage til sidste COMMIT;

Der kan ikke fortrydes ved TRUNCATE, men det går meget hurtigere idet der ikke skal gemmes rollback informationer.

8 Drop Tabel

En tabel fjernes med

```
DROP TABLE <table_name>;
```

Det er ikke muligt at genskabe tabeller efter de er fjernet.

9 Beskrivelse af elementer i databasen

For at få navnene på alle tabeller bruges

```
SELECT <table_name> FROM tabs;  
el. SELECT <table_name> FROM user_tables;
```

og hvis attributterne ønskes beskrevet bruges

```
DESCRIBE <table_name>  
el. DESC <table_name>
```

DESCRIBE er ikke en SQL kommando men en SQL*Plus kommando så der skal ikke afsluttes med semikolon, og det få ikke indflydelse på bufferen.

10 Spool output til fil

Output fra SQL kommandoer kan sendes til en fil på host maskinen.

```
spool tmp.txt  
SELECT <attribut> FROM <table_name>;  
spool off
```

dette vil oprette filen tmp.txt i kataloget hvor SQL*Plus er startet fra og skrive output fra forespørgslen til denne fil.

(NB: Alt mellem "spool" og "spool off" sendes til filen, i eksemplet vil forespørgslen også blive sendt til filen. Derfor HUSK at lukke output igen!.)

11 Antallet af rækker i en tabel

Antallet af elementer i en tabel findes med

```
SELECT COUNT(*) FROM <table_name>;
```

12 Konstanter i forespørgslerne

Man kan benytte konstanter i sine forespørgsler. Hvis man fx vil fjerne alle sine tabeller kan databasen benyttes til at genererer SQL-forespørgsler

```
select 'drop table ||table_name||;' from tabs;
```

```
'DROPTABLE||TABLE_NAME||;'
```

```
-----  
drop table BONUS;  
drop table DEPT;  
drop table DUMMY;  
drop table EMP;  
drop table SALGRADE;
```

operatoren || benyttes til at sammensætte strenge. I ovenstående eksempel sammensættes strengen 'drop table ' med ||table_name|| (tabel navnet der kommer fra forespørgslen) og strengen ';'.

13 ROWNUM

Antallet af rækker der ønskes vist i en forespørgsel kan begrænses ved at benytte ROWNUM. Forespørgslen

```
SQL> select ROWNUM from emp;
```

```
ROWNUM  
-----  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```


udskriver nummeret på rækkerne i svaret. Dette kan benyttes i en betingelse

```
SQL> select ename from emp where rownum < 4;
```

```
ENAME
-----
SMITH
ALLEN
WARD
```

således at der kun vælges 3 rækker. (NB: dette kan ikke benyttes hvis ORDER BY benyttes, og betingelsen skal være en < eller <=).

14 Tabellen DUAL

Tabellen DUAL er en speciel system tabel. Den kan med fordel benyttes hvis man er i tvivl om hvordan forskellige funktioner virker. Er man fx i tvivl om hvor dato konverteringen to_char virker kan dette testes med DUAL

```
SELECT to_char(sysdate, 'DDMMYY') from dual;
```

```
TO_CHAR(SYSDATE,'DDMMYY')
-----
120399
```

```
SELECT to_char(sysdate, 'DD/MM/YYYY HH24:MI:SS') from dual;
```

```
TO_CHAR(SYSDATE,'DD/MM/YYYY HH24:MI:SS')
-----
12/03/1999 02:55:14
```

(SYSDATE giver system datoen).