Kap15. SATSVIS KJØRING I SQL*PLUS

- ❑ Bruk av kommandofiler
- ❑ Brukernavn og passord
- ❑ Feilhåndtering

Kommandofiler

I SQL*Plus kan man si

- START filnavn

Den filen man starter kan inneholde en sekvens av SQL*Plus-kommandoer og SQL-setninger.

Filen leses og kommandoene utføres etterhvert, som om de ble tastet inn fra terminalen.

En slik fil er et SQL*Plus "program".

- ☞ Pass på at SQL-setninger blir avsluttet riktig, med ; eller /

Det er mulig starte en ny fil fra en fil som startes.

Ved oppstart av SQL*Plus vil filen login.sql startes automatisk, dersom den finnes på brukerens katalog. Denne filen utgjør en slags "brukerprofil".

Start av SQL*Plus kommandofiler fra operativsystemet.

Det er mulig å starte kommandofiler direkte fra operativsystemnivå, dvs. uten være inne i SQL*Plus

Det gjøres ved å skrive

- sqlplus @filnavn

når man er på operativsystemnivå.

SQL*Plus vil da starte opp og umiddelbart begynne å lese den angitte fil.

- OBS: Brukernavn/passord må ligge først på filen!

Eller skriv

- sqlplus brukernavn/passord @filnavn

Etter at filen er utført, forblir man inne i SQL*Plus. Hvis det er ønskelig å returnere til operativsystemnivå, må filen avsluttes med EXIT. På denne måten kan SQL*Plus-programmer kjøres i "batch".

Kommandofiler uten brukernavn/ passord.

Ofte ønsker vi ikke at brukernavn og passord skal ligge fast inne i programmet (kommandofilen).

Det unngår vi ved lage en prosedyre i operativsystemets eget kommandospråk. Den skal utføre følgende.

- Ta imot brukernavn/passord, f.eks. som parameter.
- Ta imot kommandofilens navn, f.eks. som parameter.
- Lag en temporær fil som inneholder:
  - Brukernavn/passord
  - START kommandofilnavn
- Start opp SQL*Plus:
  - Sqlplus @tempfil
- Slett den temporære filen.

Parametersubstitusjon

Det går an å bruke tall som navn på substitusjons-variabler. Hvis disse ligger i en fil som startes med START-kommandoen, kan man gi parametre sammen med START-kommandoen. Første parameter blir lagt i variabel 1, annen parameter i variabel 2 osv. Verdier som gis sammen med START, impliserer DEFINE, selv om det bare er en & i SQL-setningen.

EKSEMPEL:

```
- SELECT &2 FROM &1
- SAVE TESTSTART
- START TESTSTART LEV NAVN
- RUN
- UNDEF 1
- UNDEF 2
- RUN
```

Feilhåndtering

Dersom det oppstår en feil i forbindelse med utføringen av en SQL-setning (syntaks-feil,feil data,full tabell,...), vil denne setningen bli rullet tilbake, men tidligere utførte og ikke bekreftede setninger vil bli stående urørte.

SQL*PLUS kommandoen WHENEVER SQLERROR styrer hva som videre vil skje.

- WHENEVER SQLERROR EXIT
  - vil gjøre COMMIT, og deretter avslutte SQL*PLUS. Det kan også gies beskjed til operativsystemet om hvordan SQL*PLUS sesjonen gikk.

☞ Se SQL*PLUS håndboken på EXIT.

- WHENEVER SQLERROR CONTINUE
  - vil fortsette å utføre de neste setningene på kommando-filen.
  - Som ekstra opsjoner til CONTINUE kan brukes COMMIT eller ROLLBACK

☞ Se SQL*PLUS håndboken på WHENEVER

VARIABLER

Vi har tidligere gjennomgått oppretting av variable ved hjelp av DEFINE og ACCEPT.

Vi kan også opprette en variabel ved å benytte COLUMN opsjonen NEW_VALUE.

Eksempel:

COLUMN VARENR NEW_VALUE VARENUMMER

Vi vil da ha fått en variabel som kan refereres til på vanlig måte.

Verdien i VARENUMMER vil være verdien i den siste raden der VARENR ble skrevet ut.

Denne bruken av variabler er egnet for å overføre kolonneverdier fra en SQL-setning til en annen, for eksempel en parameter eller et tall som skal brukes i en summering.