

Kap4.Data Definisjonsspråk

- Oppretting/sletting av tabeller.
- Oppretting/sletting av andre database-objekter.

Oppretting av tabeller

Obligatorisk:

- ☞ Hvilke tabeller vi skal ha i databasen.
- ☞ Hvilke kolonner de enkelte tabeller skal inneholde.
- ☞ Størrelse og type på de enkelte kolonner.

Valgfri del som sier noe om regler :

- ☞ Lovlige verdier for kolonner.
- ☞ Entydighets-regler.
- ☞ Og flere andre.

← skranke

Oppretting av tabeller.

Valgfri del som sier noe om fysiske implementering:

- ☞ Hvor tabellene skal plasseres.
- ☞ Hvor stor plass tabellene skal ta i utgangspunktet.
- ☞ Hvordan tabellene skal organiseres fysisk.

CREATE TABLE

- Definere tabellene i databasen, og gjøre dem kjent for Data Dictionary.

SYNTAKS:

```
Create Table tabell_navn  
(kolonne1 type(lengde),  
kolonne2 type(lengde),  
..)
```

EKSEMPEL:

```
SQL > Create Table VARE  
2 (varenr char(4) not null,  
3 varenavn varchar2(20),  
4 enhet char(4),  
5 beskrivelse long)
```

Krav til CREATE TABLE:

● TABELLNAVN og KOLONNENAVN:

- Må begynne med bokstav.
- Kan inneholde bokstaver, tall og flg. spesialtegn:
 - \$ # _ (ikke-)
- Kan være opp til 30 posisjoner langt.
- Kan inneholde blanke og hvilke som helst tegn (også Æ, Ø, Å) dersom navnet skrives mellom dobbeltapostrofer: "NAVN".

☞ Reserverte ord kan ikke brukes. For å se hvilke ord dette er, skriv HELP RESERVED_WORDS

Datatyper

TYPE kan være:

NUMBER

- Lagres i et spesielt internt format. Lengde kan være opptil 40 sifre. Lengde og antall desimaler kan angis slik :
(lengde,desimaler) f.eks. number(8,2).

VARCHAR

- variabel lengde. Kan inneholde tekst og tall. Maksimal lengde 2000 tegn.

VARCHAR2 (fins ikke i versjon 6)

- samme som VARCHAR, anbefales brukt i Oracle7. Datatypen vil ikke endre seg i nyere versjoner.

CHAR

- Fast lengde. Kan inneholde tekst og tall. Max lengde 255 tegn.

Dat typer

DATE

- Kan inneholde gyldige datoer og klokkeslett. Lagres i spesielt internt format. Lengde kan ikke angis.

LONG

- Helt spesiell datatype med diverse restriksjoner. Kan inneholde opp til 2 Gigabyte (64 K i versjon 6) . Lengde kan ikke angis.

LONG RAW

- Binære data, audio, video etc. Ellers som Long.

Lagringsplass

- For alle datatyper bortsett fra CHAR, lagrer Oracle i variabelt format, dvs. at enhver kolonne opptar så mye plass som innholdet krever.
- Lengdeangivelsen i CREATE TABLE påvirker ikke den fysiske lagringsplassen, men tjener til inputkontroll.
- Felter som ikke inneholder noen verdi, tar heller ingen plass.
- Felter som inneholder verdi, tar plass:
 - NUMBER: ca 1/2 byte pr. siffer.
 - VARCHAR, VARCHAR2, CHAR og LONG: 1 byte pr. tegn.
 - DATE: alltid 7 bytes.

NULL value

- Felter som ikke har fått noen verdi, tar ingen plass. Vi sier at de har "NULL" verdi.

☞ OBS:

- NULL er ikke 0 (zero).
- NULL er ikke blank (space).

- Vi kan kreve at en kolonne alltid har verdi, ved å fylle til NOT NULL i CREATE TABLE-setningen. Vi sier at kolonnen er obligatorisk.

EKSEMPEL

```
SQL > Create Table VARE  
2 (varenr char(4) not null,  
3 varenavn varchar2(20),  
4 enhet char(4),  
5 beskrivelse long)
```

Data Dictionary

- ORACLE's interne data dictionary vet hvilke tabeller vi har definert, og hvilke kolonner disse inneholder. Vi har mulighet til å hente informasjon ut fra data dictionary:

DESCRIBE tabellnavn

- Den lister opp kolonner i en tabell, uten at SQL-bufferen påvirkes.

- ☞ Data dictionary består av ORACLE-tabeller. Det er mulig å lese disse ved hjelp av SQL-setninger.

EKSEMPLER:

```
SQL > Select *  
2 From USER_TABLES
```

- USER_TABLES er en dictionary-tabell som inneholder informasjon om de tabeller vi selv har definert (med CREATE TABLE).

```
SQL> Select *  
2 From USER_TAB_COLUMNS
```

- USER_TAB_COLUMNS er en dictionary-tabell som inneholder informasjon om kolonnene i de tabellene vi selv har definert.

EKSEMPLER:

```
SQL > Select *  
2 From DICT
```

- DICT er en dictionary-tabell som inneholder en liste over samtlige dictionary-tabeller.

COMMENT

- Det er mulig å legge kommentarer inn i data dictionary. Dette er kun for dokumentasjonsformål og har ingen behandlingmessig betydning. Både tabeller og kolonner kan kommenteres.

SYNTAKS:

```
COMMENT ON TABLE  
tabellnavn IS 'tekst'  
  
COMMENT ON COLUMN  
tabellnavn.kolonnenavn IS 'tekst'
```

EKSEMPEL:

```
SQL>COMMENT ON TABLE  
2 VARE IS 'VAREREGISTERET'  
  
SQL>SELECT COMMENTS  
FROM USER_TAB_COMMENTS  
WHERE TABLE_NAME = 'VARE'
```

ALTER TABLE

- Forandrer tabelldefinisjonen i Data Dictionary.

SYNTAKS(1):

```
ALTER TABLE tabellnavn ADD  
(kolonnenavn type(lengde),  
...  
...)
```

- Definerer inn nye kolonner i tabellen, men setter ikke inn verdier. Dvs. de nye kolonnene får NULL value. Verdier må evt. legges inn senere.

 Kolonner kan ikke fjernes.

ALTER TABLE

SYNTAKS(2):

```
Alter Table tabellnavn MODIFY  
(kolonnenavn type(lengde),  
...  
...)
```

- = Øker (men kan ikke minske) lengden på eksisterende kolonner. Type kan ikke endres.
- Dette kan gjøres mens det er data i tabellen. Det skjer ingen fysisk utvidelse av selve tabellen (unntatt for char-kolonner i ORACLE7)
- Dersom kolonnen er tom, kan lengden minskes og typen endres.

Synonymer

- Tabeller kan ha synonymer. Dette er spesielt nyttig når man vil bruke andres tabeller.

SYNTAKS:

```
Create Synonym synonymnavn  
For tabellnavn
```

```
Create Synonym synonymnavn  
For brukernavn.tabellnavn
```


Endring av navn.

- Med kommandoen RENAME kan vi endre navn på tabeller og synonymer.

SYNTAKS:

```
Rename gammelnavn  
to nyttnavn
```

EKSEMPEL:

```
Rename VARE  
to NY_VARE
```

- Når vi bruker RENAME, forsvinner det gamle navnet. Det kan selvfølgelig benyttes til nye tabeller senere.

☞ Kolonnenavn kan ikke endres.

Fjerning

☞ Man kan fjerne synonymer

☞ Man kan fjerne hele tabeller.

- Når tabellen fjernes forsvinner innholdet av tabellen, det samme gjør tabelldefinisjonen i data dictionary. Synonymer vil bestå, men når vi bruker dem vil vi få feilmelding: 'Synonym-oversetting er ikke gyldig lenger.'

SYNTAKS:

Drop Synonym synonymnavn

Drop Table tabellnavn
{ Cascade Constraints }