

Kap2.Lesing av data

- SELECT KOMMANDO.
- ENDRING AV KOLONNEOVERSKRIFTER.
- SELEKSJON AV RADER.
- SORTERING.

SELECT KOMMANDO

SELECT setninger brukes for å lese data fra databasen.

SYNTAKS

Obligatorisk:



SELECT

FROM

WHERE *AND OR*

CONNECT BY

Valgfritt



GROUP BY

HAVING

ORDER BY

☞ Legg merke til rekkefølgen..

Enkleste SELECT

```
SELECT * FROM tabellnavn
```

Leser hele tabellen, samtlige rader og kolonner.

```
SELECT kolonnenavn, kolonnenavn FROM tabellnavn
```

Leser samtlige rader i tabellen, men bare de kolonner som er angitt.

EKSEMPEL:

```
SELECT ENAVN, ANSNR FROM ANSATT
```

eller `select enavn "Etternavn på ansatt" from ansatt;`

☞ OBS! Komma mellom kolonnenavnene!

Endring av kolonneoverskrifter.

Vi kan “døpe om” kolonnene når de skrives ut. Det gjøres ved å skrive et nytt navn bak kolonnenavnet i SELECT-listen:

```
SELECT kolonnenavn nyttnavn, kolonnenavn nyttnavn  
FROM tabellnavn
```

Vi velger selv hvilke kolonner og hvor mange kolonner som skal navnes om på denne måten. Det nye navnet kalles for et “midlertidig merke”. Det erstatter ikke det virkelige navnet i databasen, men kommer ut som kolonne-overskrift i denne ene SELECT-setningen.

EKSEMPEL:

```
SELECT ENAVN NAVN, ANSNR NR  
FROM ANSATT
```

Legg merke til plasseringen av kommaet! Se på eksemplet på forrige side. Ta bort kommaet og se hva som hender da.

Seleksjon av rader

Vi kan velge hvilke rader i tabellen som skal leses, ved å skrive en WHERE-betingelse i SELECT-setningen.

WHERE-betingelsen kan teste kolonner mot konstantverdier, mot andre kolonner, eller mot uttrykk. Flere betingelser kan kombineres med AND og OR. Husk AND utføres før OR.

IKKE skriv WHERE flere ganger!

Det vanligste er å teste på likhet, men det finnes også en rekke andre muligheter.

☞ = lik

☞ != ikke lik

☞ > større enn

☞ >= større eller lik

☞ < mindre enn

☞ <= mindre eller lik

WHERE eksempler

```
SELECT * FROM ANSATT  
WHERE LONN = 11000
```

```
WHERE LONN > 12000
```

```
WHERE LONN < ANSNR
```

```
WHERE LONN > ANSNR * 2
```

```
WHERE 1 > 2
```

```
WHERE ANSNR = 7876 AND SJEFNR > 7800
```

```
WHERE LONN < 13000 OR SJEFNR = 7698
```

```
WHERE LONN < 13000 OR SJEFNR = 7698 AND  
LONN > 14600
```

```
WHERE LONN < 13000 OR (SJEFNR = 7698 AND  
LONN > 14600)
```

```
WHERE (LONN < 13000 OR SJEFNR = 7698) AND  
LONN > 14600
```

Test på NULL

Skal vi teste på NULL-value, må dette gjøres med IS NULL eller IS NOT NULL.

Kolonner med NULL-value gir ikke tilslag på noen andre tester!

- ☞ NULL er ikke lik noen verdi (heller ikke 0 eller blank)
- ☞ NULL er ikke forskjellig fra noen verdi.
- ☞ NULL er ikke større enn noen verdier.
- ☞ NULL er ikke mindre enn noen verdier.
- ☞ To kolonne verdier som er NULL er ikke like, og de er heller ikke forskjellige.

ved NULL summering : NVL(LONN, 0)

EKSEMPEL:

```
SELECT * FROM ANSATT WHERE SJEFNR IS NULL
```

Andre tester

BETWEEN og NOT BETWEEN tester om en verdi befinner seg innenfor eller utenfor et intervall. Grenseverdiene er en del av intervallet.

IN og NOT IN tester om en verdi finnes eller ikke finnes i en liste av verdier.

EKSEMPLER:

```
SELECT *  
FROM ANSATT  
WHERE LONN BETWEEN 9000 AND 11000
```

Inkluderer grenseverdier.

```
SELECT *  
FROM ANSATT  
WHERE ANSNR NOT IN (7698,7782,7788)
```


Søking på tekst

Ved søking på konstantverdier som ikke er numeriske, må konstanten stå mellom enkeltapostrofer: 'tekst'

EKSEMPEL

```
SELECT *  
FROM ANSATT  
WHERE ENAVN = 'DONALD'
```

string → *char*
Vi kan teste på "jokertegn" (% og _) ved å bruke LIKE og NOT LIKE. (Altså ikke likhetstegn!)

EKSEMPEL:

```
SELECT *  
FROM ANSATT  
WHERE ENAVN LIKE '%SEN'
```

 Det er forskjell på store og små bokstaver!

Sortering

SELECT-setninger presenterer ikke radene i noen spesiell rekkefølge. Ønsker vi en rekkefølge, må vi føye til ORDER BY og en liste over kolonner det skal sorteres på. For hver kolonne kan vi velge synkende rekkefølge ved å føye til DESC.

EKSEMPLER :

```
SELECT * FROM ANSATT
ORDER BY ANSNR
eller
ORDER BY ANSNR, LONN
eller
ORDER BY ANSNR,LONN DESC
```

Man kan bruke kolonnenes plassering i SELECT-listen som spesifikasjon av sorteringen:

```
ORDER BY 1,5 desc
```