

## Kap8.Sammenføring av tabeller.

- Select med kolonner fra flere tabeller
- Sammenføyingsbetingelser *join*
- Ytre sammenføyingsbetingelser *outer join*
- Sammenføring "til seg selv"
- Mengde-operatorer

## Sammenføring

En SELECT-setning som leser fra flere tabeller, kaller vi en sammenføring:

```
select kolonnenavn, kolonnenavn  
from tabellnavn, tabellnavn
```

Vi skriver bare flere tabellnavn i FROM-delen av SELECT-setningen vår. Det kan være to eller flere tabellnavn.

## Like kolonnenavn

Det kan tenkes at samme kolonnenavn forekommer i flere av de tabellene vi leser. Da må vi skrive tabellnavnet som prefiks til kolonnenavnet for at det skal være klart hvilken vi mener.

SELECT tabellnavn.kolonnenavn osv.

For å spare skrivearbeid, kan vi gi tabellene midlertidige merker som gjelder i denne SELECT-setningen.

```
SELECT X.kolonnenavn, Y.kolonnenavn  
FROM tabellnavn X, tabellnavn Y
```

## Sammenføyingsbetingelser

EKSEMPEL:

```
SELECT VL.LEVNR, L.NAVN, VL.PRIS  
FROM VARE_LEV VL, LEV L  
WHERE L.LEVNR = VL.LEVNR
```

Dette kalles for en sammenføyingsbetingelse fordi vi tester på likhet mellom kolonner i de to tabellene. Vi er nødt til skrive denne testen i WHERE-betingelsen vår. Vi får den ikke automatisk. Det er ikke noe krav at de kolonnene som skal være like har samme navn. Men som oftest har de samme type informasjon, ellers har det ingen hensikt å teste på likhet.

De kolonnene som skal være like, kan vi kalle for sammenføyingsbegreper. Som oftest er det nøkkelbegreper som brukes slik. Det kan godt være flere kolonner som skal være like, eller det kan være virtuelle kolonner.

## Ytre sammenføring

I forrige eksempel fikk vi ikke med alle rader i LEV tabellen. De som har et LEVNR som ikke finnes i VARE\_LEV-tabellen, får ikke klaff i WHERE-betingelsen og kommer ikke med. Men vi kan få dem med likevel, ved å føye til(+) bak kolonne-navnet til den kolonnen som mangler verdier, i dette tilfelle LEVNR i VARE\_LEV-tabellen. De verdiene som skulle vært hentet fra VARE\_LEV-tabellen returneres som NULL.

```
SELECT VL.LEVNR, L.NAVN, VL.VAREN, VL.PRIS  
FROM   VARE_LEV VL, LEV L  
WHERE  L.LEVNR = VL.LEVNR(+)
```

Dette kalles for en YTRE SAMMENFØYING.

## Kartesiske produkt

Tar vi bort sammenføyingsbetingelsen, får vi et KARTESISK PRODUKT. Da vil vi få alle radene i den ene tabellen kombinert med alle radene i den andre.

```
SELECT VL.LEVNR, L.NAVN, VL.VAREN, VL.PRIS  
FROM VARE_LEV VL, LEV L
```

Vanligvis ønsker vi ikke kartesiske produkter, men ofte får vi dem av vanvare ved at vi ikke har tilstrekkelige sammenføyingsbetingelser. Spesielt ved sammenføyning av mange tabeller, er det fort gjort å miste oversikten.

 Huskeregel:

Ved sammenføyning av N tabeller, må vi ha minst N-1 sammenføyingsbetingelser for å unngå kartesiske produkter.

GIR SAMME  
RESULTAT

```
Select * from ansatt
      where lonn < any (select lonn from ansatt
                       where funksjon = 'SELGER');
```

```
.....where lonn < (select max(lonn)
                    from ansatt .....
```

GIR SAMME  
RESULTAT

```
select * from ansatt
      where lonn < all (select lonn from ansatt
                       where funksjon = 'SELGER');
```

```
.....where lonn < (select min(lonn)
                    from ansatt....
```

## Sammenføring til seg selv

En tabell kan slås sammen med seg selv. Maskinen vil da "late som" om den har to like tabeller, og så gjøre sammenføring på disse.

Siden begge tabellene heter det samme og har de samme kolonnene, må vi bruke midlertidige merker for å holde dem fra hverandre.

Det er vanlig å bruke sammenføring til seg selv på tabeller som inneholder to kolonner med lik type informasjon.



## Eksempel: Ansatt - Sjef

Vi kan f.eks. tenke oss en ansatt-tabell med informasjon om ansnr, navn,..., sjefnr, etc, - der sjefen igjen er ansatt i dette firmaet:

```
SELECT ANS.ANSNR, ANS.ENAVN, ANS.SJEFNR, SJEF.ENAVN  
FROM ANSATT ANS, ANSATT SJEF  
WHERE ANS.SJEFNR = SJEF.ANSNR
```

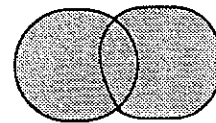
Sammenføyings-betingelsen behøver ikke nødvendigvis være likhet. Hvis ønskelig kan også andre operatorer brukes.

WHERE-betingelsene kan også inneholde andre betingelser enn selve sammenføyings-betingelsen.

## Mengde - operatorene

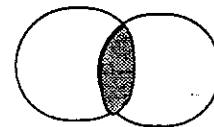
### ☞ UNION

- Union, dubletter undertrykkes



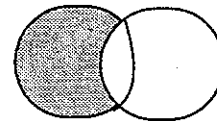
### ☞ INTERSECT

- Snitt, bare dublettene



### ☞ MINUS

- differanse, det som er med i den første tabellen, men ikke i den andre.



☞ UNION ALL  
- som ~~dubletter~~ union  
dubletter undertrykkes, ikke



### SYNTAKS:

```
SELECT-setning
mengde-operator
SELECT-setning
```

## Eksempel: UNION

```
SELECT VARENR FROM VARE  
UNION  
SELECT VARENR FROM VARE_LEV
```

SELECT-setningene må gi sammenlignbare resultater, dvs. samme antall kolonner, og kolonner av samme type.

Finn alle bestillinger som ikke har bestillingslinjer.

ved hjelp av:

1. Mengdeoperator
2. Ikke sammenholdt delspørring
3. Sammenholdt delspørring

1. `SELECT b.bestnr FROM bestilling b  
MINUS  
SELECT bl.bestnr FROM bestlinje bl;`

2. `SELECT b.bestnr  
FROM bestilling b  
WHERE b.bestnr NOT IN (SELECT bl.bestnr FROM bestlinje bl);`

3. `SELECT b.bestnr  
FROM bestilling b  
WHERE b.bestnr NOT IN (SELECT bl.bestnr FROM bestlinje bl  
WHERE bl.bestnr=b.bestnr);`

eller:

`SELECT b.bestnr  
FROM bestilling b  
WHERE b.bestnr NOT EXISTS (SELECT 'OK' FROM bestlinje bl  
WHERE bl.bestnr=b.bestnr);`