

# Relationel Algebra og SQL

## Indholdsfortegnelse

Relationel Algebra og SQL.....	1
Indholdsfortegnelse .....	1
De oprindelige mængdeoperationer .....	2
1. UNION (foreningsmængde) .....	2
2. INTERSECTION (fællesmængde).....	2
3. DIFFERENCE (differensmængde).....	2
4. PRODUKT (kartesisk produkt) .....	3
RESTRIKTION, SELEKTION (udvælgelse) .....	3
PROJEKTION .....	4
JOIN (samling) .....	4
DIVISION.....	5

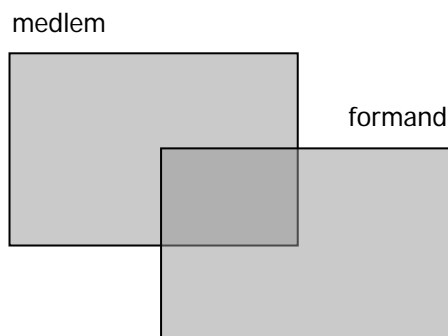
# De oprindelige mængdeoperationer

## 1. UNION (foreningsmængde)

To mængder forenes til en foreningsmængde.

Alle rækker (tupler) i den ene tabel (relation) forenes med alle rækker i den anden tabel. Rækkerne i de to tabeller skal være af samme slags.

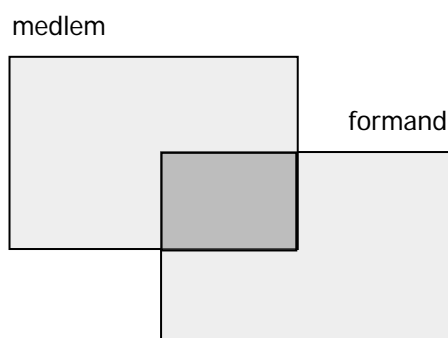
```
SELECT * from medlem
UNION
SELECT * from formand;
```



## 2. INTERSECTION (fællesmængde)

Fællesmængden fra to mængder er de elementer, som findes både i den ene og i den anden mængde.

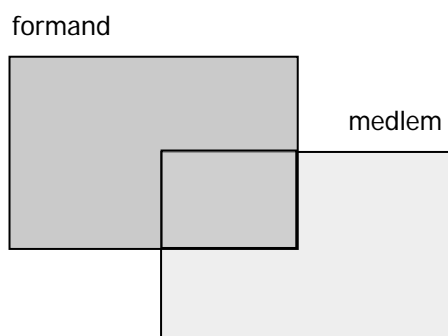
```
SELECT m.afdnr, m.personnr
FROM medlem m, formand f
WHERE m.afdnr = f.afdnr AND
      m.personnr = f.personnr;
```



## 3. DIFFERENCE (differensmængde)

Ved at trække to mængder fra hinanden, fås mængdedifferencen mellem dem. Forekomsterne i begge mængder skal være af samme slags.

```
SELECT *
FROM formand f
WHERE not EXISTS (SELECT *
                  FROM medlem m
                  WHERE m.afdnr = f.afdnr AND
                        m.personnr = f.personnr);
```

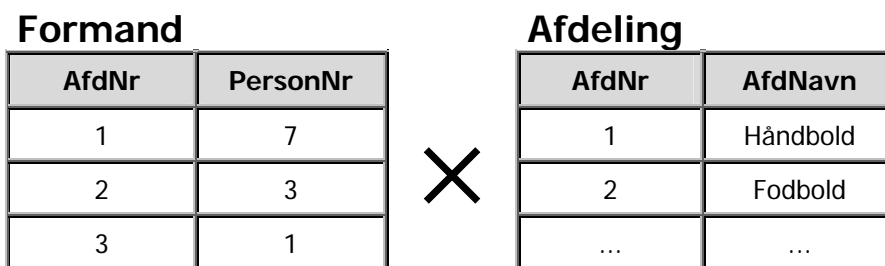


Det modsatte fra eksemplet med fællesmængde. Dvs. at mængdedifferencen udgøres af de formænd som også er medlemmer, men som **ikke** er formænd og medlemmer i samme afdeling.

### 4. PRODUKT (kartesisk produkt)

Produktet mellem to mængder er mængden af **ordnede par**, hvor det ene element stammer fra den ene mængde, og det andet element stammer fra den anden mængde.

```
SELECT *
FROM afdeling, formand;
```



AfdNr	PersonNr	AfdNr	AfdNavn
1	7	1	Håndbold
1	7	2	Fodbold
2	3	1	Håndbold
2	3	2	Fodbold
3	1	1	Håndbold
3	1	2	Fodbold
...	...	...	...

### RESTRIKTION, SELEKTION (udvælgelse)

Finder de forekomster, der opfylder en eller flere logiske betingelser.

```
SELECT *
FROM post
WHERE distrikt LIKE 'O%' AND
      distrikt <> Odense;
```

Selektion


PostNr	Distrikt
6476	Otterup
6565	Ollerup
6477	Ovre

## PROJEKTION

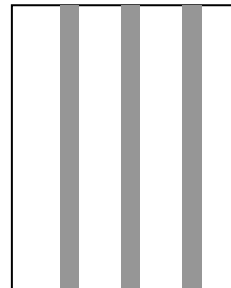
Undersøger de søjler (attributter), der er interesse for.

```
SELECT fornavn
FROM person;
```

### Person

Fornavn
Helge
Tove
Poul
...

### Projektion



## JOIN (samling)

Samling af forekomster fra flere mængder i én mængde. Udtrækket er kombineret med selektion og evt. projektion, dvs. et kartesisk produkt med restriktioner

```
SELECT *
FROM afdeling a, formand f, person p
WHERE a.afdnr = f.afdnr AND
      f.personnr = p.personnr;
```

Den følgende (ækvivalente) SQL-sætning er Access bud på at lave en joinforbindelse:

```
SELECT *
FROM afdeling a INNER JOIN
      (formand f INNER JOIN person p
      ON f.personnr = p.personnr)
ON a.afdnr = f.afdnr;
```

I Access kan man bruge en INNER JOIN-handling i alle FROM-delsætninger. Dette er den mest almindelige type joinforbindelse. Indre joinforbindelser kombinerer poster fra to tabeller, når der er tilsvarende værdier i et felt, der er fælles for begge tabeller.

Man kan bruge INNER JOIN sammen med tabellerne Afdeling og Medlem til at vælge alle medlemmerne i en afdeling. Hvis man derimod vil vælge alle afdelinger (selvom nogle ikke har tilknyttet medlemmer) eller alle medlemmer (selvom nogle ikke er tilknyttet en afdeling), kan man benytte en LEFT JOIN- eller RIGHT JOIN-handling til at oprette en ydre joinforbindelse. Venstre ydre joinforbindelser medtager alle poster fra den første (venstre) af de to tabeller, selvom der ikke findes tilsvarende værdier for poster i den anden (højre) tabel, og man kan anvende en RIGHT JOIN-handling til at oprette en højre ydre joinforbindelse. Denne

medtager alle posterne fra den anden (højre) af de to tabeller, selv om der ikke findes tilsvarende værdier for poster i den første (venstre) tabel.

## DIVISION

Udvælger alle (kvotient) forekomster i en mængde (dividend), der er relateret til samtlige værdier i en anden mængde (divisor).

```
SELECT distrikt, personnr, COUNT(*)  
FROM medlem  
GROUP BY personnr  
HAVING COUNT(*) = (SELECT COUNT(*) FROM afdeling);
```

