

Introduksjon til SQL

Leksjon 1

Eksempler på bruk av SQL

- Opprette nye tabeller
- Jobbe med tabeller
 - Sette inn nye rader
 - Slette rader
 - Oppdatere rader
- Spørre
 - Filtrere
 - Sortere
 - Gruppere
 - Koble
- Brukeradministrasjon
- Effektivisere søk



A word cloud of SQL keywords. The words are arranged in a roughly circular pattern. The largest word is 'select' in dark red. Other large words include 'create' (in dark red and orange), 'insert', 'update', 'delete', 'user', 'index', and 'table'. The words are in various shades of red, orange, and brown.

insert
create user
create index
select
delete
create table
update

Databasetabellen Ansatt

| AnsattNr | Etternavn | Fornavn | AnsattDato | Stilling | Lønn |
|----------|------------|--------------|------------|-----------------|------------|
| 1 | Veum | Varg | 01.01.1992 | Løpegutt | 183 000.00 |
| 2 | Stein | Trude | 10.10.2000 | DBA | 270 700.00 |
| 3 | Dudal | Inger-Lise | 24.12.1988 | Sekretær | 299 000.00 |
| 4 | Hansen | Hans | 23.08.2006 | Programmerer | 325 000.00 |
| 5 | Bjørnsen | Henrik | 01.01.2000 | Tekstforfatter | 375 000.00 |
| 6 | Gredelin | Sofie | 18.05.1998 | Underdirektør | 625 850.00 |
| 7 | Zimmermann | Robert | 17.05.1995 | Regnskapsfører | 375 000.00 |
| 8 | Nilsen | Lise | 03.04.2002 | Direktør | 675 340.00 |
| 11 | Fosheim | Katinka | 13.09.2011 | Selger | 420 000.00 |
| 13 | Lovløs | Ada | 12.08.2005 | Programmerer | 384 250.00 |
| 16 | Ibsen | Bjørnstjerne | 02.01.2008 | Tekstforfatter | 346 000.00 |
| 17 | Fleksnes | Marve | 17.05.2009 | Lagerleder | 320 120.00 |
| 20 | Felgen | Reodor | 12.12.2001 | Sykkelreparatør | 279 500.00 |
| 23 | Karius | Jens | 13.12.2001 | Salgssekretær | 280 390.00 |
| 29 | Wirkola | Gabriel | 21.04.2009 | Sekretær | 255 000.00 |

Eksempelspørring

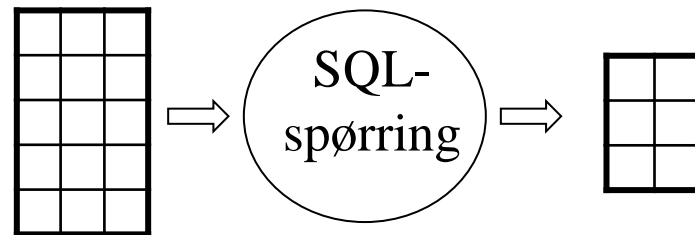
```
SELECT AnsattNr, Etternavn, Lønn  
FROM    Ansatt  
WHERE   Lønn < 280000
```

Spørreresultat:

| AnsattNr | Etternavn | Lønn |
|----------|-----------|---------------|
| 1 | Veum | kr 183 000.00 |
| 2 | Stein | kr 270 700.00 |
| 20 | Felgen | kr 279 500.00 |
| 29 | Wirkola | kr 255 000.00 |

Fra tabell(er) til spørreresultat


- En **utvalgsspørring** tar tabeller som «inndata» og gir som «utdata» et **spørreresultat**, som også er på tabellform.
- I første omgang ser vi på spørringer mot 1 tabell.



- Hvordan blir en utvalgsspørring utført (av DBHS)?

Utførelse av utvalgsspørringer

SELECT **AnsattNr, Etternavn, Lønn**
FROM **Ansatt**
WHERE **Lønn < 280000**



| | AnsattNr | Etternavn | Fornavn | AnsattDato | Stilling | Lønn |
|-----|----------|-----------|------------|------------|-----------------|------------|
| → ✓ | 1 | Veum | Varg | 01.01.1992 | Løpegutt | 183 000.00 |
| → ✓ | 2 | Stein | Trude | 10.10.2000 | DBA | 270 700.00 |
| → | 3 | Dudal | Inger-Lise | 24.12.1988 | Sekretær | 299 000.00 |
| | ... | ... | ... | ... | ... | ... |
| → ✓ | 20 | Felgen | Reodor | 12.12.2001 | Sykkelreparatør | 279 500.00 |
| → | 23 | Karius | Jens | 13.12.2001 | Salgssekretær | 280 390.00 |
| → ✓ | 29 | Wirkola | Gabriel | 21.04.2009 | Sekretær | 255 000.00 |

Datamaskiner og formelle språk

- SQL er et formelt språk:
 - Presise regler for hva som er en lovlig «setning».
 - Små skrivefeil gir **feilmelding** eller **uventet resultat**.



SELECT-setningsdelen inneholder et reservert ord eller argument som mangler eller er feilstavet, eller den har feil skilletegn.

OK

Hjelp

- **Les feilmeldingene!**
 - Det krever trening å tolke feilmeldinger.
 - Feilmeldingene identifiserer ikke alltid feilen helt presist.
- Sjekk at spørreresultatet gir mening:
 - Spørringen kan være **logisk feil** selv om vi ikke får feilmelding!

Byggekllossene i spørrespråket SQL

- SQL-spørringer er bygd opp fra:
 - Reserverte ord (f.eks. **SELECT** og **FROM**)
 - Navn på tabeller og kolonner (f.eks. **Ansatt** og **Etternavn**)
 - Funksjoner (f.eks. **UPPER**) og operatorer (f.eks. **<**)
 - Skilletegn (f.eks. **komma**)

```
SELECT AnsattNr, Etternavn, Lønn
FROM    Ansatt
WHERE   Lønn < 280000 AND
          UPPER(Stilling) = 'SEKRETÆR'
```


Utvalgsspøringer i SQL (SELECT)

- Denne leksjonen: Enkle spørringer mot 1 tabell
 - Velge ut kolonner
 - Velge ut rader
 - Sortere rader med hensyn på en kolonne
 - Legge til kalkulerte kolonner
- Leksjon 3+5: Mer avanserte spørringer
 - Gruppere og aggregere data (sum, gjennomsnitt osv.)
 - Koble data fra flere tabeller
 - ...

Velge ut kolonner

| | | | | | |
|--|--|--|--|--|--|
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

- Når vi kun er interessert i noen av kolonnene:

```
SELECT  AnsattNr, Etternavn  
FROM Ansatt
```

- Når vi vil ha alle kolonnene:

```
SELECT  *  
FROM Ansatt
```

- Hvis vi velger en kolonne kan vi få flere like verdier.
- Slike duplikater kan enkelt fjernes:

```
SELECT  DISTINCT Stilling  
FROM Ansatt
```

Velge ut rader

| |
|--|
| |
| |
| |
| |

- Når vi vil plukke ut rader som oppfyller en gitt betingelse.

```
SELECT *  
FROM Ansatt  
WHERE Lønn < 280000
```

- En **betingelse** er et uttrykk som er sant eller galt.

Logiske operatorer AND, OR og NOT

- Brukes for å bygge sammensatte betingelser:

| AND | true | false |
|-------|-------|-------|
| true | true | false |
| false | false | false |

(Lønn > 280000) **AND**
(Stilling = 'Sekretær')

| OR | true | false |
|-------|------|-------|
| true | true | true |
| false | true | false |

(Lønn < 300000) **OR**
(Lønn > 500000)

| NOT | true | false |
|-----|-------|-------|
| | false | true |

NOT (Lønn <= 300000)

Bruk parenteser!

- Vi ønsker å finne sekretærer/selgere som tjener mer enn 280.000 kroner i året.

```
SELECT *  
FROM   Ansatt  
WHERE  Lønn > 280000  
AND    Stilling = 'Sekretær'  
OR     Stilling = 'Selger'
```

- Kan spørringen tolkes på flere måter?
- Er det flere måter å plassere parenteser på?
- Hva skal gjøres «først»: **AND** eller **OR**?
- Hva gjør DBHS hvis vi **ikke** setter på parenteser?

Sortering og sammenligninger

- For å sortere må vi kunne sammenligne.
 - **Tall**: ... -2 < -1 < 0 < 1 < 2 < 3 < ...
 - **Bokstaver**: a < b < c < ...
 - **Tekster**: adam < anna < anne < anneli < david
 - **Datoer**: 22.03.1978 < 07.02.1992 < 31.12.2014
- Sortering i SQL innebærer å sortere **rader** med hensyn på en bestemt kolonne.
 - Noen av Ansatt-radene sortert etter **Lønn**:

| AnsattNr | Etternavn | Fornavn | AnsattDato | Stilling | Lønn |
|----------|-----------|---------|------------|-----------------|-------------------|
| 1 | Veum | Varg | 01.01.1992 | Løpegutt | 183 000.00 |
| 2 | Stein | Trude | 10.10.2000 | DBA | 270 700.00 |
| 20 | Felgen | Reodor | 12.12.2001 | Sykkelreparatør | 279 500.00 |
| 6 | Gredelin | Sofie | 18.05.1998 | Underdirektør | 625 850.00 |

Sortering i SQL

- Sortert navneliste:

```
SELECT AnsattNr, Etternavn  
FROM Ansatt  
ORDER BY Etternavn
```

- Flere sorteringskriterier:

```
SELECT Etternavn, Stilling, Lønn  
FROM Ansatt  
ORDER BY Stilling ASC, Lønn DESC
```

- **ASC** gir **stigende** sortering og **DESC** gir **synkende**.
- ASC er standard, så første spørring gir stigende sortering.

Jokernotasjon og intervallsøk

- Finn alle med etternavn som begynner på F:

```
SELECT  *  
FROM    Ansatt  
WHERE    Etternavn >= ' F'  
AND      Etternavn <  ' G'
```

- Jokernotasjon er mer elegant her:

```
SELECT  *  
FROM    Ansatt  
WHERE    Etternavn LIKE ' F%'
```

- Hvorfor er **Etternavn = 'F%'** meningsløst?
- Finn alle som har etternavn som slutter på 'sen' !

Kalkulerte kolonner

- Det er lov å bruke uttrykk i SELECT-delen:

```
SELECT AnsattNr, Etternavn,  
        Lønn/12 AS LønnPrMåned  
FROM    Ansatt
```

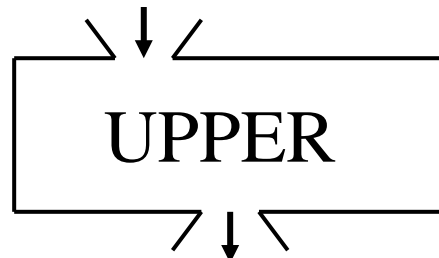
- Ikke nødvendig å lagre både årslønn og månedslønn!
- Lønn/12 er et **uttrykk**. For å få en meningsfull overskrift i utskriften gir vi denne kolonnen et navn.

Funksjoner

- Tekst, f.eks. navn, kan gjøres om til store bokstaver:

```
SELECT AnsattNr, UPPER(Etternavn)  
FROM   Ansatt
```

- UPPER er en **funksjon** (MySQL).
- En funksjon kan ha flere **argumenter**, men bare én **returverdi**.
- Det finnes et stort antall funksjoner, se vedlegg i læreboken og MySQL-dokumentasjonen på nettet.



Teste funksjoner i MySQL

- For å få erfaring med virkemåten til funksjoner kan man skrive uttrykk i en spørring uten FROM-del:

```
SELECT UPPER( ' Hansen' )
```

Svaret blir:

```
HANSEN
```

Kan bruke MySQL som kalkulator:

```
SELECT 2+2
```

Operatorer og uttrykk

■ Operatorer

- Aritmetiske: *****, **/**, **+**, **-**
- Sammenligning: **>**, **<**, **=**, **>=**, **<=**, **<**, **<>**
- Jokernotasjon: **LIKE**
- Test for nullmerke: **IS NULL**
- Boolske: **NOT**, **AND**, **OR**
- Intervalltest: **BETWEEN ... AND ...**

- Vi kan bygge opp uttrykk fra konkrete verdier, navn på kolonner, funksjoner og operatorer:

(Stilling LIKE 'S*') AND ((Lønn/12) > 15000)

Dato og tid

- Varighet i ansettelsesforhold (antall dager):

```
SELECT Etternavn,  
        YEAR(AnsattDato) ,  
        AnsattDato - CURDATE() AS AntDager  
FROM Ansatt
```

- Nå-tid: **CURDATE**
- Trekke ut deler av en dato: **YEAR, MONTH, HOUR**
- Sammenligne datoer: <, >, <=, >=, =
- Navn på funksjoner varierer noe fra system til system.

Oppbygging av enkle utvalgsspørringer

- Enkle utvalgsspørringer (**SELECT**-spørringer) mot **1 tabell** følger dette «mønsteret»:

| | |
|-----------------|-----------------------------|
| SELECT | Hvilke kolonner skal med? |
| FROM | Hva heter tabellen? |
| WHERE | Hvilke rader skal med? |
| ORDER BY | Sortere på hvilke kolonner? |

- Ikke alle delene må være med i alle spørringer.
- Vi skal legge på flere reserverte ord.
- **SELECT** er én blant mange SQL-kommandoer...