



INF130

Spørringer mot en tabell (Enkle spørringer)

2

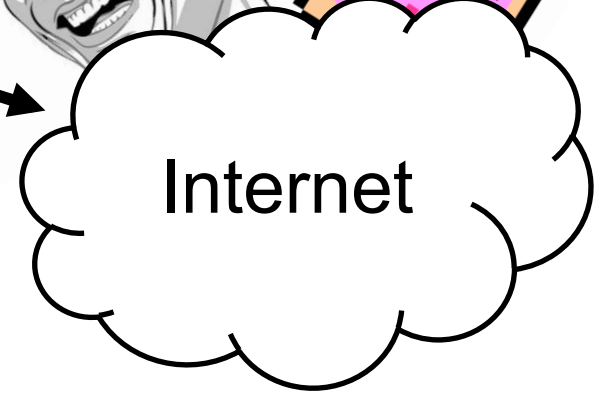
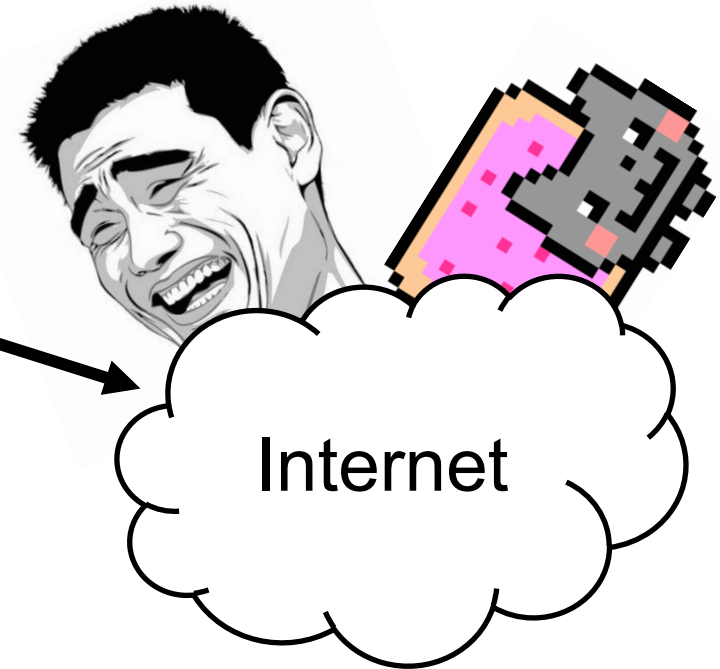
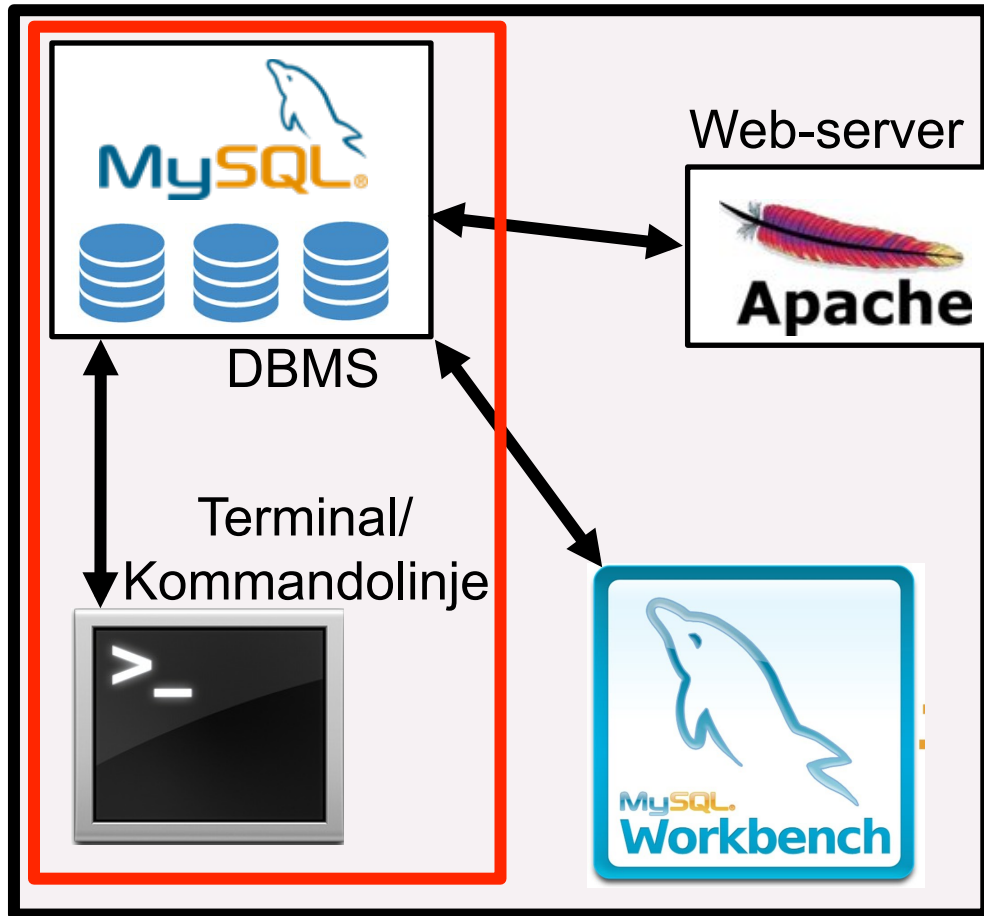
Spørringer mot én tabell

Læringsmål

- Kjenne til viktige bruksområder for databasespråket SQL.
- Kunne bruke SQL utvalgsspørringer for å hente ut data fra én tabell.
- Kunne velge ut rader og kolonner.
- Kunne sortere radene med hensyn på bestemte kolonner.
- Kunne aggregere data (beregne sum, gjennomsnitt og lignende).
- Kunne bruke funksjoner og operatorer.

Programmvaremiljø

Datamaskin



Demonstrasjoner

- Lage databaser
- Lage tabeller
- Import av bokbasen til MySQL
- Import av databasen University for obligatorisk oppgave
- Se på strukturer av databaser og tabeller
- Sette inn data i tabeller

Hovedgrupperinger av SQL oppgaver

- Opprette nye tabeller og struktur
- Jobbe med tabeller
 - Sette inn nye rader
 - Slette rader
 - Oppdatere rader
- Hente data
 - Filtrere
 - Sortere
 - Gruppere
 - Koble
- Brukeradministrasjon
- Effektivisere søk



A word cloud of SQL keywords. The words are arranged in a roughly rectangular shape, with 'select' being the largest and most prominent word in the center. Other large words include 'create', 'insert', 'update', 'delete', 'user', 'index', and 'table'. The words are in various shades of brown and orange, with some words like 'create' and 'table' appearing vertically on the right side.

insert
create user
create index
select
delete
create table
update

Lage database og tabell (kortversjon)



Databasetabellen Ansatt

AnsNr	Fornavn	Etternavn	Adresse	PostNr	Fødselsdato	Kjønn	Stilling	Årslønn
1	Georg	Barth	Kringsjågrenda 3F	3841	1978-10-20	M	Lagerleder	526000.00
2	Gunnlaug	Angeltveit	Langmyrgrenda 9	3800	1965-03-29	K	Markedssjef	559300.00
3	Morgan	Dalland	Jansbergveien 19	3830	1970-01-10	M	Innkjøper	583000.00
6	Vilde	Aksnes	Minister Ditleffs vei 44	3810	1973-10-11	K	Databaseadministrator	602800.00
7	Henriette	Brobakken	Stubberud Sognsvann 1	3800	1967-10-01	K	Daglig leder	725000.00
8	Synøve	Bakketun	Vassøyveien 7	3840	1981-05-15	K	Kundebehandler	450500.00
9	Ragnvald	Allum	Utsikten 4	3812	1988-03-07	M	Kundebehandler	421500.00
11	Oliver	Abrahamsen	Tarjei Vesaas'' vei 3A	3812	1985-01-20	M	Lagermedarbeider	406000.00
13	Oda	Cappelen	Norheimskeiken 12	3800	1987-02-28	K	Produktutvikler	567900.00
16	Andrine	Ebbesen	Kristianias gate 9	3800	1984-12-27	K	Regnskapssekretær	462900.00
17	Karl Anton	Hoff	Furustia 3	3840	1993-08-03	M	Kundebehandler	410700.00
18	Johanna	Li	Krogsteinveien 101A	3841	1992-05-17	K	Kundebehandler	416200.00

Eksempelspørring

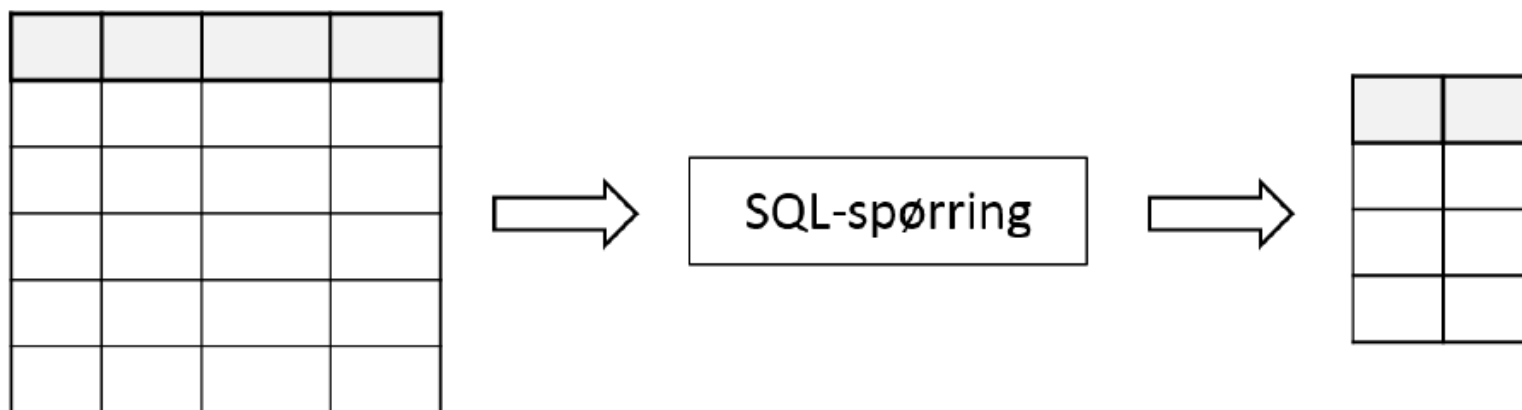
```
SELECT AnsNr, Etternavn, Årslønn  
FROM   Ansatt  
WHERE  Årslønn < 450000
```

Spørreresultat:

AnsNr	Etternavn	årslønn
9	Allum	421500.00
11	Abrahamsen	406000.00
17	Hoff	410700.00
18	Li	416200.00

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.



Utførelse av utvalgsspørringer

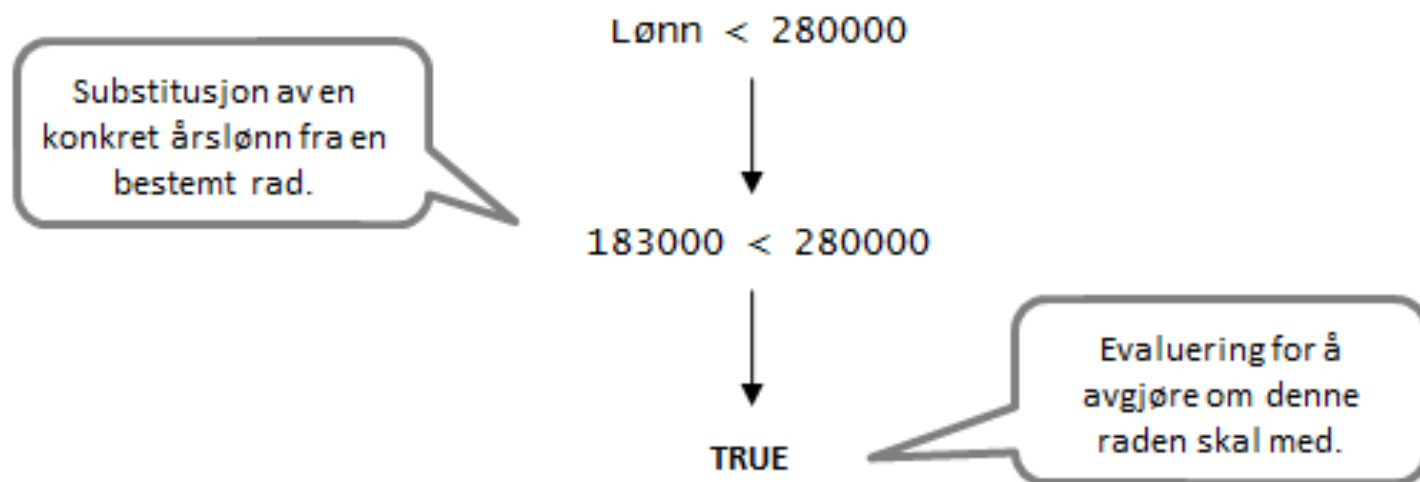
```
SELECT AnsNr, Etternavn, Årslønn  
FROM   Ansatt  
WHERE  Årslønn < 450000
```



AnsNr	Etternavn	fornavn	Stilling	årslønn
1	Barth	Georg	Lagerleder	526000.00
2	Angeltveit	Gunnlaug	Markedssjef	559300.00
3	Dalland	Morgan	Innkjøper	583000.00
6	Aksnes	Vilde	Databaseadministrator	602800.00
7	Brobakken	Henriette	Daglig leder	725000.00
8	Bakketun	Synøve	Kundebehandler	450500.00
9	Allum	Ragnvald	Kundebehandler	421500.00
11	Abrahamsen	Oliver	Lagermedarbeider	406000.00
13	Cappelen	Oda	Produktutvikler	567900.00
16	Ebbesen	Andrine	Regnskapssekretær	462900.00
17	Hoff	Karl Anton	Kundebehandler	410700.00
18	Li	Johanna	Kundebehandler	416200.00

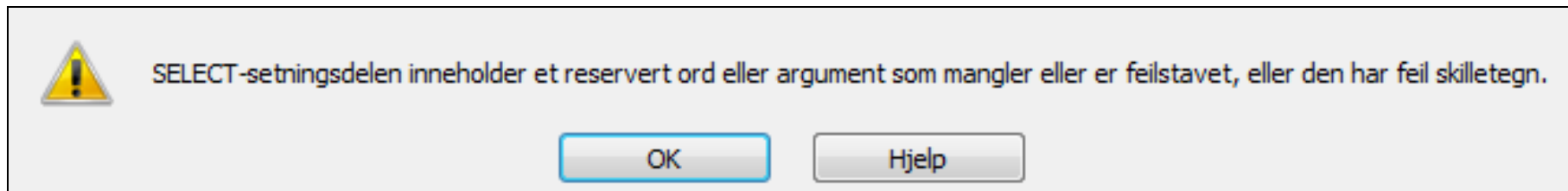
Substitusjon og evaluering

- DBHS utfører spørringen ved å gå gjennom tabellen rad for rad.
- For hver rad blir konkrete verdier satt inn for kolonnenavn (substitusjon).
- Deretter blir uttrykket evaluert (til true eller false).



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**.



- **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!

Byggeklossene 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 AnsNr, Etternavn, Årslønn
FROM   Ansatt
WHERE  Årslønn < 450000 AND
       UPPER(Stilling) = 'KUNDEBEHANDLER'
```

Utvalgsspøringer i SQL (SELECT)

- I første omgang: Enkle spørringer mot 1 tabell
 - Velge ut kolonner
 - Velge ut rader
 - Sortere rader med hensyn på en kolonne
 - Legge til kalkulerte kolonner
 - Gruppere og aggregere data (sum, gjennomsnitt osv.)

Velge ut kolonner

- Når vi kun er interessert i noen av kolonnene:

```
SELECT AnsNr, 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 Årslønn < 450000
```

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

Logiske operatører AND, OR og NOT

- Brukes for å bygge sammensatte betingelser:

(Årslønn > 450000) **AND**
(Stilling = 'kundebhandler')

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

NOT (Lønn <= 300000)

AND	true	false
true	true	false
false	false	false

OR	true	false
true	true	true
false	true	false

NOT	true	false
	false	true

Bruk parenteser!

- Vi ønsker å finne kundebehandlere/selgere som tjener mer enn 450.000 kroner i året.

```
SELECT *  
FROM   Ansatt  
WHERE  Lønn > 450000  
AND    stilling = 'Kundebehandler'  
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?

Operatorprioritet

1. $-$ (unær minus, f.eks. -3)
2. $*$ $/$ $\%$
3. $+$ $-$ (binære operatorer, f.eks. $2+2$)
4. $<$ $<=$ $>$ $>=$ $=$ $<>$
5. **NOT**
6. **AND**
7. **OR**

Sortering og sammenligninger

- For å sortere må vi kunne sammenligne.
 - **Tall**: ... $-2 < -1 < 0 < 1 < 2 < 3 < \dots$
 - **Bokstaver**: $a < b < c < \dots$
 - **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 AnsNr, Etternavn  
FROM Ansatt  
ORDER BY Etternavn
```

- Flere sorteringskriterier:

```
SELECT Etternavn, Stilling, Årslønn  
FROM Ansatt  
ORDER BY Stilling ASC, Årslønn 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 AnsNr, Etternavn,  
       Årslønn/12 AS LønnPrMåned  
FROM   Ansatt
```

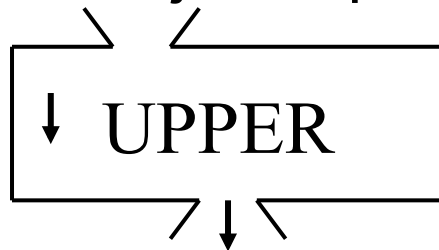
- Ikke nødvendig å lagre både årslønn og månedslønn!
- Lønn/12 er et **uttrykk**. For å få en meningsfull overskrift i utskriften gir vi denne kolonnen et navn.

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

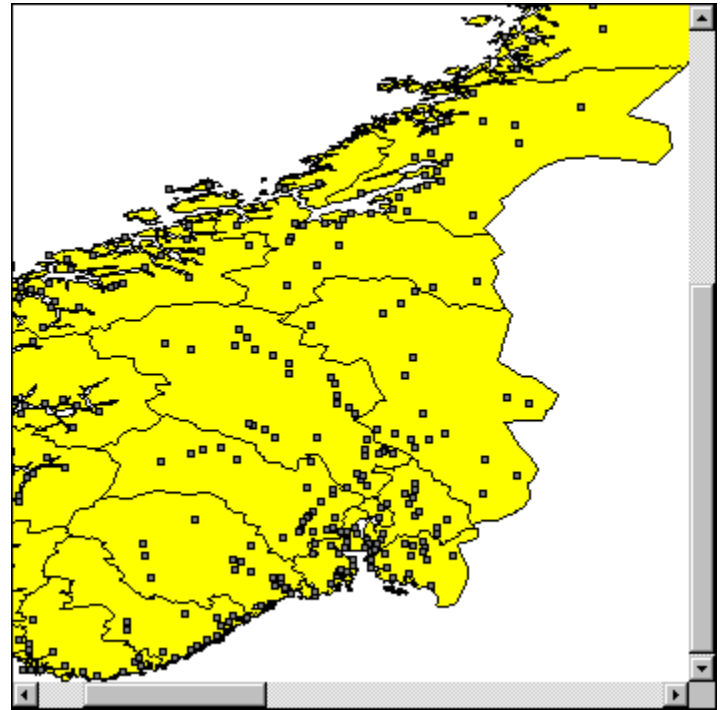
- Varighet i ansettelsesforhold (antall dager):

```
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.

Gruppering og mengdefunksjoner

Ola Nordmann bor i Hansegata 3, Andebu og tjener 320.000 kr. pr. år.

- Zoomer vi langt nok inn i kartet kan vi vise egenskaper ved Ola.
- Zoomer vi ut ønsker vi **aggregerte data**: Total inntekt, gjennomsnittsinntekt fordelt på kjønn, alder, osv.



To typer av kolonner

AnsattNr	Etternavn	Fornavn	Stilling	Lønn
1	Veum	Varg	Selger	300 000
2	Stein	Trude	Programmerer	400 000
3	Dudal	Inger-Lise	Sekretær	300 000
4	Hansen	Hans	Programmerer	500 000
5	Bjørnsen	Henrik	Selger	500 000
6	Gredelin	Sofie	Direktør	800 000
7	Zimmermann	Robert	Programmerer	600 000

- Kolonnen Lønn: Beregne sum, gjennomsnitt, ...
- Kolonnen Stilling: Danne «grupper», f.eks.
 - Gjennomsnitt for hver stillingskategori (gruppe)

(Vanlige) funksjoner og mengdefunksjoner

- Operasjoner på **enkeltverdier**, f.eks.:
 - UPPER: Tekst til store bokstaver
 - YEAR: Trekk ut årstallet fra dato

- Operasjoner på **verdisamlinger**:
 - AVG**(kolnavn) gjennomsnitt
 - SUM**(kolnavn) sum
 - COUNT**(*) antall
 - MIN**(kolnavn) minimum
 - MAX**(kolnavn) maksimum

Mengdefunksjoner

- Gjennomsnittslønn:

```
SELECT AVG(ÅrsLønn) AS SnittLønn  
FROM Ansatt
```




SnittLønn
485 714

1 kolonne!

1 rad!

- Hva med **totale** lønnsutgifter?
- Eller gjennomsnittslønn **for programmererne**?

COUNT og nullmerker

- Antall ansatte (= antall rader):

```
SELECT COUNT(*) FROM Ansatt
```

- Antall ansatte med registrert lønn:

```
SELECT COUNT(ÅrsLønn) FROM Ansatt
```

- COUNT(Lønn) teller ikke rader med nullmerke i Lønn.

- Et **feilaktig** forsøk på å beregne gjennomsnitt:

```
SELECT SUM(ÅrsLønn)/COUNT(*) FROM Ansatt
```

- Bruk **AVG(Lønn)** i stedet!

Gruppering

- Finn gjennomsnittlønn for hver stillingskategori.

```
SELECT stilling, AVG(ÅrsLønn)  
FROM Ansatt  
GROUP BY stilling
```


- Stilling har kun få ulike verdier!
- Grupperingskolonner må være med i resultatet.
- Gruppering brukes ofte sammen med mengdefunksjoner.

Spørresresultat

- For å finne **gjennomsnittslønn fordelt på stilling** må vi først danne **grupper**: selgerne, programmererne, ...
- Så må vi beregne gjennomsnittslønn for hver gruppe.
- Resultatet får 4 rader fordi det er 4 forskjellige verdier i Stilling.

Stilling	Lønn
Sekretær	300 000
Programmerer	500 000
Selger	400 000
Direktør	800 000

Gruppebetingelse (HAVING)

- Hva blir resultatet her?

```
SELECT Stilling, AVG(ÅrsLønn),  
COUNT(*)  
FROM Ansatt  
GROUP BY Stilling  
HAVING AVG(ÅrsLønn) > 450000
```

- **HAVING** på grupper svarer til **WHERE** på rader.
- Merk: Det er ikke lov å bruke mengdefunksjoner i **WHERE**-betingelser.

Gruppebetingelser og radbetingelser

- Betingelser på rader og på grupper:

```
SELECT stilling, AVG(ÅrsLønn),  
COUNT(*)  
FROM Ansatt  
WHERE Year(AnsattDato) > 1990  
GROUP BY stilling  
HAVING AVG(ÅrsLønn) > 280000
```

- Hvorfor kan ikke de to betingelsene slås sammen?
- Hvordan kan DBHS utføre spørringen?

Oppbygging av utvalgsspørringer

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

SELECT	Hvilke kolonner skal med?
FROM	Fra hvilke tabeller?
WHERE	Hvilke rader skal med?
GROUP BY	Gruppere på hvilke kolonner?
HAVING	Hvilke grupper 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...

Øvinger

- Obligatorisk oppgave 1
 - Innlevering innen fredag 23. februar kl 16.00
 - Uke 8
 - Arbeid i din kollokviegruppe og lever **felles** på Canvas
- Øvingsoppgave 3 Import av data
- Utfør kommandoer fra læreboka

Noen demonstrasjoner

- Plenumsgjennomgangen mandag
Tabellen “vare” fra læreboka
- Vi skal se på noen eksempler
 - MySQL terminal/Kommandolinje