

Kapittel 2

Spørringer mot én tabell

1

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 gruppere data (beregne sum, gjennomsnitt og lignende).
- Kunne bruke funksjoner og operatører.

2

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

insert
create user
create index
select
delete
create table
update

3

Databasetabellen Ansatt

AnsattNr	Etternavn	Fornavn	AnsattDato	Stilling	Lønn
1	Veum	Varg	01.01.2001	Løpegutt	483 000.00
2	Stein	Trude	10.10.2009	DBA	570 700.00
3	Dudal	Inger-Lise	24.12.2017	Sekretær	599 000.00
4	Hansen	Hans	23.08.2015	Programmerer	625 000.00
5	Bjørnsen	Henrik	01.01.2019	Tekstforfatter	675 000.00
6	Gredelin	Sofie	18.05.2017	Underdirektør	925 850.00
7	Zimmermann	Robert	17.05.2004	Regnskapsfører	675 000.00
8	Nilsen	Lise	03.04.2021	Direktør	975 340.00
11	Fosheim	Katinka	13.09.2020	Selger	720 000.00
13	Lovles	Ada	12.08.2014	Programmerer	684 250.00
16	Ibsen	Bjørnstjerne	02.01.2017	Tekstforfatter	646 000.00
17	Fleknes	Marve	17.05.2018	Lagerleder	620 120.00
20	Felgen	Reodor	12.12.2010	Sykkelreparatør	579 500.00
23	Karius	Jens	13.12.2020	Salgssekretær	580 390.00
29	Wirkola	Gabriel	21.04.2023	Sekretær	555 000.00

4

Eksempelspørring

SELECT AnsattNr, Etternavn, Lønn
FROM Ansatt
WHERE Lønn < 580000

Spørreresultat:

AnsattNr	Etternavn	Lønn
1	Veum	kr 483 000.00
2	Stein	kr 570 700.00
20	Felgen	kr 579 500.00
29	Wirkola	kr 555 000.00

5

Fra tabell(er) til spørreresultat

- En **utvalgsspørring** tar tabeller som «innndata» 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)?

6

Utførelse av utvalgsspøringer

```
SELECT AnsattNr, Etternavn, Lønn
FROM Ansatt
WHERE Lønn < 580000
```

AnsattNr	Etternavn	Fornavn	AnsattDato	Stilling	Lønn
1	Veum	Varg	01.01.2001	Løpegutt	483 000.00
2	Stein	Trude	10.10.2009	DBA	570 700.00
3	Dudal	Inger-Lise	24.12.2017	Sekretær	599 000.00
...	...	...	...	...	...
20	Felgen	Reodor	12.12.2010	Sykkelreparatør	579 500.00
23	Karius	Jens	13.12.2020	Salgssekretær	580 390.00
29	Wirkola	Gabriel	21.04.2023	Sekretær	555 000.00

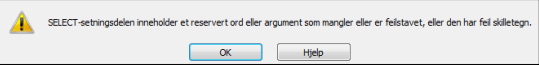
Substitusjon og evaluering

- 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 AnsattNr, Etternavn, Lønn
FROM Ansatt
WHERE Lønn < 580000 AND
      UPPER(Stilling) = 'SEKRETÆR'
```

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 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 < 580000
```

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

13

Logiske operatører AND, OR og NOT

- Brukes for å bygge sammensatte betingelser:

AND	TRUE	FALSE
TRUE	TRUE	FALSE
FALSE	FALSE	FALSE

```
(Lønn > 580000) AND
(Stilling = 'Sekretær')
```

OR	TRUE	FALSE
TRUE	TRUE	TRUE
FALSE	TRUE	FALSE

```
(Lønn < 600000) OR
(Lønn > 800000)
```

NOT	TRUE	FALSE
	FALSE	TRUE

```
NOT (Lønn <= 600000)
```

14

Bruk parenteser!

- Vi ønsker å finne selgere/sekretærer som tjener mer enn 580.000 kroner i året.

```
SELECT *
FROM Ansatt
WHERE Lønn > 580000
AND Stilling = 'Selger'
OR Stilling = 'Sekretær'
```

- 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?

15

Operatorprioritet

- (unær minus, f.eks. -3)
- * / %
- + - (binære operatører, f.eks. 2+2)
- < <= > >= = <>
- NOT
- AND
- OR

16

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:** 22.03.1982 < 07.02.1996 < 31.12.2018
- 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	01.01.2001	Løpegutt	483 000.00
2	Stein	Trude	10.10.2009	DBA	570 700.00
20	Felgen	Reodor	12.12.2010	Sykkelreparatør	579 500.00
6	Gredelin	Sofie	18.05.2017	Underdirektør	925 850.00

17

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.

18

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' !

19

Uttrykk i SELECT-delen

- 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.

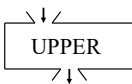
20

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.



21

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
```

22

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)>50000)

23

Dato og tid

- Varighet i ansettelsesforhold (antall dager):

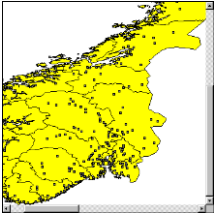
```
SELECT Etternavn,
       YEAR(AnsattDato),
       AnsattDato - CURDATE() AS AntDager
FROM Ansatt
```
- Nå-tid: **CURDATE**
- Trekke ut deler av en dato: **YEAR, MONTH, HOUR**
- Sammenligne datoer: **<, >, <=, >=, =**
- Navn på funksjoner varierer noe fra system til system.

24

Gruppering og mengdefunksjoner

Ola Nordmann bor i Hansegata 3, Andebu og tjener 620.000 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.



25

To typer av kolonner

AnsattNr	Etternavn	Fornavn	Stilling	Lønn
1	Veum	Varg	Selger	400 000
2	Stein	Trude	Programmerer	500 000
3	Dudal	Inger-Lise	Sekretær	400 000
4	Hansen	Hans	Programmerer	600 000
5	Bjørnsen	Henrik	Selger	600 000
6	Gredelin	Sofie	Direktør	900 000
7	Zimmermann	Robert	Programmerer	700 000

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

26

(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

27

Mengdefunksjoner

- Gjennomsnittslønn:
SELECT AVG(Lønn) AS SnittLønn
FROM Ansatt

 →

SnittLønn
585 714

1 rad!
1 kolonne!

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

28

COUNT og nullmerker

- Antall ansatte (= antall rader):
SELECT COUNT(*) FROM Ansatt
- Antall ansatte med registrert lønn:
SELECT COUNT(Lønn) FROM Ansatt
- COUNT(Lønn) teller ikke rader med nullmerke i Lønn.
- Et **feilaktig** forsøk på å beregne gjennomsnitt:
SELECT SUM(Lønn)/COUNT(*) FROM Ansatt
- Bruk **AVG(Lønn)** i stedet!

29

Gruppering

- Finn gjennomsnittslønn for hver stillingskategori.

Stilling	AVG(Lønn)

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

30

Spørreresultat

- 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	400 000
Programmerer	600 000
Selger	500 000
Direktør	900 000

31

Gruppebetingelse (HAVING)

- Hva blir resultatet her?

```
SELECT Stilling, AVG(Lønn), COUNT(*)  
FROM Ansatt  
GROUP BY Stilling  
HAVING AVG(Lønn) > 480000
```
- **HAVING** på grupper svarer til **WHERE** på rader.
- Merk: Det er ikke lov å bruke mengdefunksjoner i **WHERE**-betingelser.

32

Gruppebetingelser og radbetingelser

- Betingelser på rader og på grupper:

```
SELECT Stilling, AVG(Lønn), COUNT(*)  
FROM Ansatt  
WHERE Year(AnsattDato) > 2000  
GROUP BY Stilling  
HAVING AVG(Lønn) > 480000
```
- Hvorfor kan ikke de to betingelsene slås sammen?
- Hvordan kan DBHS utføre spørringen?

33

Oppbygging av utvalgsspørringer

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

SELECT	Hvilke kolonner skal med?
FROM	Hva heter tabellen?
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...

34