

Kapittel 4

Spørringer mot flere tabeller

Læringsmål

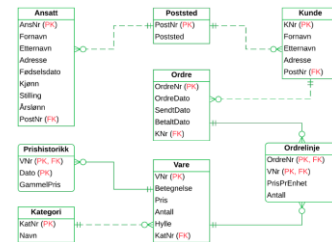
- Forstå behovet for spørringer mot flere tabeller.
- Forstå effekten av de forskjellige måtene å koble tabeller på, og spesielt forskjellen på et kryssprodukt og en likekobling.
- Kunne bruke SQL for å skrive indre likekoblinger, ytre koblinger, egenkoblinger, generelle koblinger og naturlige koblinger.
- Kunne bruke SQL mengdeoperatorer.

Motivasjon

- For å unngå **redundans** («dobbeltlagring») lagrer vi data i flere tabeller.
- Relasjonsdatabaser består av **mange** tabeller.
- Det er logiske sammenhenger mellom data i ulike tabeller.
 - Behov for å «**koble**» data fra flere tabeller.
 - Forutsetter at samme type verdier finnes i flere tabeller.
 - Koblinger er ofte basert på fremmednøkler (men ikke alltid).

Hvilke tabeller må brukes?

- Kundeliste sortert på poststed.
- Totalpris pr. ordre. Det holder å vise ordrenr og pris.
- Salg pr. kunde i hver kategori. Sorter på kundenavn.



Flere tabeller i FROM-delen

- Hva blir resultatet?

SELECT *
FROM Ordre, Kunde

Ordre

Kunde
- Hvilke rader i Ordre skal kobles med hvilke rader i Kunde?
 - Ordre inneholder KNr ...
- Antall **rader** og **kolonner** i resultatet?
- Vi kan generelt hente data fra flere enn 2 tabeller.
- **NB! Spørringen øverst er meningsløs! Forklaring følger...**

Kartesisk produkt (kryssprodukt)

KNr	Navn	OrdreNr	KNr	AnsNr
1	Per	1	1	21
2	Ola	2	2	21
		3	2	28

$$2 \times 3 = 6 !$$

KNr	Navn	OrdreNr	KNr	AnsNr
1	Per	1	1	21
1	Per	2	2	21
1	Per	3	2	28
2	Ola	1	1	21
2	Ola	2	2	21
2	Ola	3	2	28

Likekobling

KNr	Navn	OrdreNr	KNr	AnsNr
1	Per	1	1	21
2	Ola	2	2	21
		3	2	28



KNr	Navn	OrdreNr	KNr	AnsNr
1	Per	1	1	21
2	Ola	2	2	21
2	Ola	3	2	28

- Tar som regel kun med koblingskolonne(r) fra én av tabellene. De inneholder jo de samme verdiene.

Sammensatte navn

- Vi har ofte kolonner med samme navn i flere tabeller.
- I spørringer mot flere tabeller er det ikke alltid entydig hva som menes med et kolonnenavn.
- Bruker sammensatte navn for å fortelle hvilken kolonne vi mener (på formen **tabell.kolonne**), f.eks:
 - Kunde.KNr
 - Kunde.Navn
 - Ordre.KNr

Eksempel på likekobling

- Kunder med sine ordrer.

```
SELECT Kunde.KNr, Etternavn, OrdreNr
FROM Kunde, Ordre
WHERE Ordre.KNr = Kunde.KNr
```
- Likheten i **WHERE** er en **koblingsbetingelse**.
- Spørringen kalles en (indre) **likekobling** (join).
- KNr må prefikses med tabellnavn fordi kolonnenavnet forekommer i begge tabeller.
- Etternavn og OrdreNr kan prefikses, men må ikke.

Fremmednøkler og koblinger

- Vi kobler ofte to tabeller med hensyn på fremmednøkler.
- Det er mulig å koble på kolonner som ikke er fremmednøkler eller primærnøkler.
- Kombinasjoner av ansatte og kunder bosatt samme sted:

```
SELECT *
FROM Ansatt, Kunde
WHERE Ansatt.PostNr = Kunde.PostNr
```
- Av og til blir det feil å koble med hensyn på samtlige fremmednøkler.

Syntaks for likekobling

- Likekoblinger forekommer så hyppig at det er innført en spesiell skrivemåte:

```
SELECT *
FROM Ordre
INNER JOIN Kunde ON Ordre.KNr = Kunde.KNr
```
- Generelt:

```
T1 INNER JOIN T2 ON T1.ko11 = T2.ko12
```
- Rekkefølgen av tabellene spiller ingen rolle.
- Access genererer spørringer med denne syntaksen hvis du bruker utformingsvisning («pek-og-klikk»).
- INNER JOIN er også den anbefalte skrivemåten, men helt i starten er det kanskje lettere å skrive koblingsbetingelser i WHERE?

Synonymer

- For å redusere skrivearbeidet kan man innføre synonymer (kortnavn) for tabellene.

```
SELECT O.OrdreNr, K.KNr
FROM Ordre AS O
INNER JOIN Kunde AS K ON O.KNr = K.KNr
```
- Hvis man innfører synonymer må de brukes !
 - Synonymene O og K må også brukes i **SELECT**-delen, selv om de først blir «introdusert» i **FROM**-delen.
- Oracle bruker ikke AS. I MySQL kan man velge.

Likekobling med ekstrabetingelser

- Kan ha generelle betingelser i tillegg til koblingsbetingelser.

```
SELECT V.VNr, K.Navn
FROM Vare AS V
INNER JOIN Kategori AS K ON V.KatNr = K.KatNr
AND V.Pris > 100
AND K.Navn = 'Fiske'
```

- Hvordan bør DBHS utføre slike spørringer ?
 - Koblingsbetingelser først ?
 - Andre betingelser først ?

13

Datatyper og sammenligninger

- Dette er **meningsløst**:

```
SELECT *
FROM Ansatt AS A
INNER JOIN Ordre AS O ON A.AnsNr = O.OrdreNr
```

- Koblingskolonnene må ha samme datatype, og også inneholde **samme slags** verdier.
- Koblingskolonner trenger imidlertid **ikke** ha samme **navn**.
- Vil DBHS godta spørringen over?

14

Koble flere tabeller enn 2

- Hvilke kunder har kjøpt vare 1014 ?

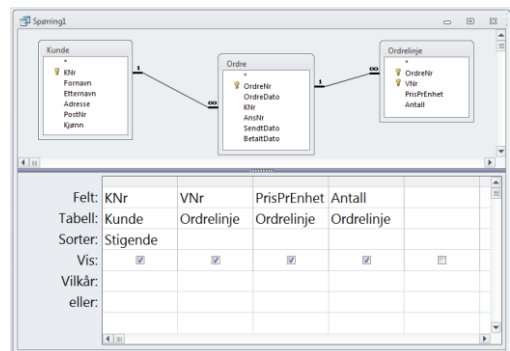
```
SELECT K.*
FROM Kunde AS K
INNER JOIN Ordre AS O ON K.KNr = O.KNr
INNER JOIN Ordrelinje AS OL
ON O.OrdreNr = OL.OrdreNr
WHERE OL.VNr = 1014
```

- Kan også skrives slik:

```
SELECT K.*
FROM Kunde AS K, Ordre AS O, Ordrelinje AS OL
WHERE O.OrdreNr = OL.OrdreNr
AND K.KNr = O.KNr
AND OL.VNr = 1014
```

15

Query By Example



16

Likekobling med gruppering

- Finn antall ordrer pr. kunde:

```
SELECT K.KNr, Etternavn,
COUNT(*) AS AntallOrdrer
FROM Kunde AS K
INNER JOIN Ordre AS O ON K.KNr = O.KNr
GROUP BY K.KNr, Etternavn
```

- Hva om vi kun vil vise kunder med flere enn 10 ordrer ?
- Totalt varekjøp pr. kunde:
 - Hvilke tabeller må kobles?
 - Hva skal vi gruppere på?
 - Hvilken mengdefunksjon må vi bruke?

17

Hvilke kolonner skal med i gruppering?

- Se spørring på forrige lysark: **Etternavn** føres opp etter SELECT fordi vi vil ha det med i utskriften.
- Da tar vi den også med etter GROUP BY, selv om den logisk sett ikke bidrar til (endrer) grupperingen.
- Noen systemer gir feilmelding hvis den ikke er med.


```
SELECT K.KNr, Etternavn
...
GROUP BY K.KNr, Etternavn
```
- I neste spørring er det **feil** å ta med **Betegnelse** fordi dette er en egenskap ved enkeltvarer og ikke hele kategorien.
- Men Pris kan være med etter SELECT selv om den **ikke** står etter GROUP BY, fordi den er parameter til AVG.


```
SELECT KatNr, Betegnelse, AVG(Pris) AS Snitt
FROM Vare
GROUP BY KatNr
```

18

Ytre koblinger (høyre og venstre)

- Indre likekoblinger tar bare med verdier som finnes i begge tabeller. Det er ikke alltid det vi ønsker.
- Vis kunder med tilhørende ordrer. Samtlige kunder skal med.


```
SELECT K.KNr, O.OrdreNr
FROM Kunde AS K
LEFT OUTER JOIN Ordre AS O ON K.KNr = O.KNr
```
- «Venstre» og «høyre» refererer til rekkefølgen i FROM.
- Antall rader i resultatet?

19

Venstre ytre kobling

KNr	Navn	OrdreNr	KNr	AnsNr
1	Per	1008	1	25
2	Ola	1009	2	25
3	Lise	1010	1	28

KNr	Navn	OrdreNr	KNr	AnsNr
1	Per	1008	1	25
2	Ola	1009	2	25
1	Per	1010	1	28
3	Lise			

- Ytre kobling = likekobling + 1 rad for hver uten «match».

20

Generelle koblinger

- Det er mulig å koble med **andre operatoren** enn likhet (=).
- Finn varer som har kostet mer før:


```
SELECT DISTINCT V.VNr
FROM Vare AS V, Prishistorikk AS H
WHERE V.VNr = H.VNr
AND V.Pris < H.Gammelpris
```
- Noen GIS-eksempler med «geografiske operatoren»:
 - Finn byer i Telemark (punkt **INNENFOR** polygon)
 - Finn veier som krysser Mjøsa (linje **KRYSSER** linje)
 - Finn eiendommer som blir berørt av en veiutbygging (polygon **OVERLAPPER** polygon)

21

Egenkoblinger

- Tabeller kan kobles med «seg selv».
- Finn alle kombinasjoner av varer med samme pris:


```
SELECT V1.VareID, V2.VareID, V1.Pris
FROM Vare AS V1, Vare AS V2
WHERE V1.VNr <> V2.VNr
AND V1.Pris = V2.Pris
```
- Tenk slik: DBHS «**dager 2 kopier**» av tabellen Vare, og kobler disse på vanlig måte.

22

Egenkoblinger

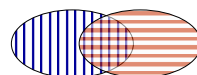
- Egenkoblinger må brukes når vi har en fremmednøkkel som peker på primærnøgkelen i samme tabell.
- Eksempel: Vis ansatte med navn på leder (se side 104).

AnsNr	Fornavn	Etternavn	Leder
1	Georg	Barth	2
2	Gunnlaug	Angeltveit	
3	Morgan	Dalland	7
6	Vilde	Aksnes	8
7	Henriette	Brobakken	2
8	Synøve	Bakketun	2
9	Ragnvald	Allum	7
11	Oliver	Abrahamsen	7
13	Oda	Cappelen	8
16	Andrine	Ebbesen	7

23

Snitt, union og differanse

- En tabell består av en **mengde** rader. Vi har tilgang på standard mengdeoperasjoner:



- Anta Ordre2023 og Ordre2024 inneholder historiske data.
- Hvem har handlet i 2023 og/eller 2024?


```
SELECT KNr FROM Ordre2023
UNION
SELECT KNr FROM Ordre2024
```
- Noen systemer støtter også **INTERSECT** (snitt) og **MINUS/EXCEPT** (differanse).

24