

# IN2090 – Databaser og datamodellering

## 06 – Views

Leif Harald Karlsen  
leifhka@ifi.uio.no



Universitetet i Oslo

# Views

---

- ◆ Merk at vi nesten aldri er interessert i dataene slik de er lagret
- ◆ Vi må nesten alltid joine tabeller, filtrere vekk rader, projisere vekk kolonner, osv. for å få interessant data ut
- ◆ F.eks. i Filmdatabasen må man joine 3 tabeller for å finne ut hvilken skuespiller som spiller i hvilken film
- ◆ Hvorfor er det slik?
- ◆ Jo, fordi vi ønsker å representere dataene på slik måte at:
  - ◆ vi aldri repeterer data (gjør det enkelt å vedlikeholde, mer effektivt, osv.)
  - ◆ dataene kan brukes på mange forskjellige måter
- ◆ Vi bruker så spørringer for å få ut interessant data
- ◆ Av og til vil en bestemt spørring bli eksekvert veldig ofte
- ◆ Det er da upraktisk å måtte skrive den ut hver gang
- ◆ I slike tilfeller kan man lage et **VIEW**

# Å lage views

- ◆ Et view er egentlig bare en navngitt spørring, og lages slik:

```
CREATE VIEW StudentTakesCourse ( StdName text, CourseName text )
AS
  SELECT S.StdName, C.CourseName
  FROM Students AS S,
       Courses AS C,
       TakesCourse AS T
  WHERE S.SID = T.SID AND C.CID = T.CID
```

- ◆ Et view kan så brukes som om det var en vanlig tabell
- ◆ Men blir beregnet på nytt hver gang den brukes
- ◆ Så et view tar ikke opp noe plass og trengs ikke oppdateres
- ◆ Så,

```
SELECT *
FROM StudentTakesCourse AS s
WHERE s.StdName = 'Anna Consuma'
```



```
SELECT *
FROM (
  SELECT S.StdName, C.CourseName
  FROM Students AS S, Courses AS C,
       TakesCourse AS T
  WHERE S.SID = T.SID AND
       C.CID = T.CID) AS s
WHERE s.StdName = 'Anna Consuma'
```

# Views som abstraksjoner

---

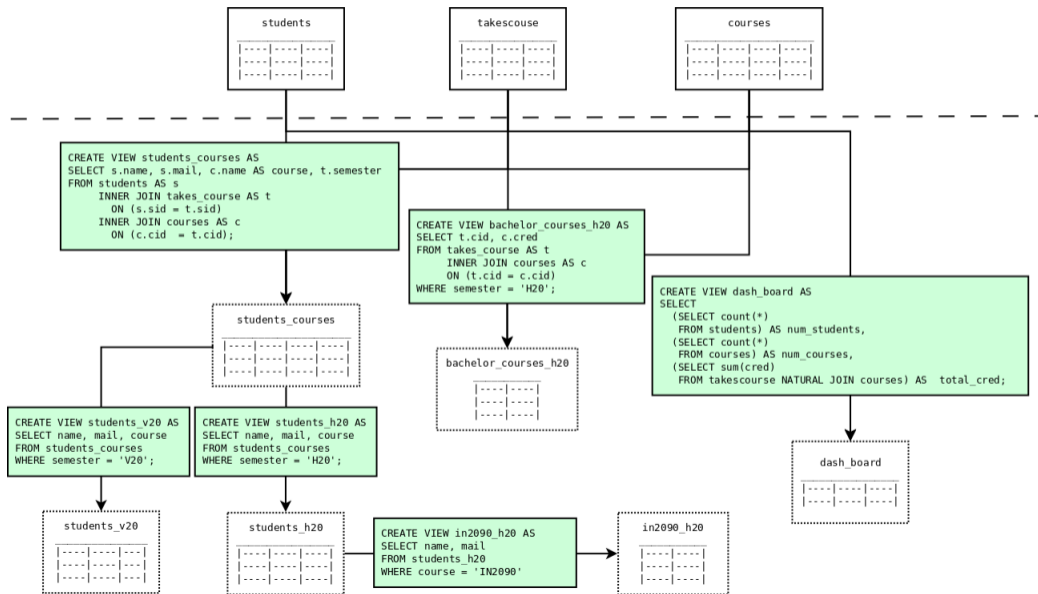
- ◆ Views kan også brukes for å bygge lag med abstraksjoner over tabellene
- ◆ F.eks. gitt følgende tabeller:

sid	name	mail
1	Anna Consuma	anna@mail.no
2	Peter Young	py@uio.no
3	Mary Smith	smith@ifi.no

sid	cid	semester
1	1	h18
1	2	v18
2	3	v18
3	2	v19
3	1	h19

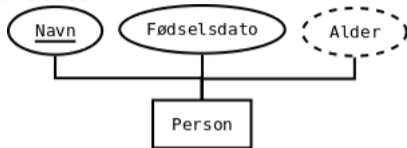
cid	name	cred	lvl
1	Databases	10	B
2	Programming 101	5	B
3	Advanced SQL	10	M

# Views som abstraksjoner



# Views for utledbare verdier

- ◆ I ER har vi utledbare attributter:



- ◆ Med views kan vi introdusere disse attributtene igjen
- ◆ Uten at vi trenger å lagre dem, holde dem oppdatert, osv.

navn	fødselsdato
Anna Consuma	1989-08-17
Peter Young	1991-02-29
Mary Smith	1993-01-01

navn	fødselsdato	alder
Anna Consuma	1989-08-17	30
Peter Young	1991-02-29	28
Mary Smith	1993-01-01	26

```
CREATE VIEW person_alder AS
SELECT navn,
       fødselsdato,
       EXTRACT(year FROM age(current_date, fødselsdato)) AS alder
FROM person
```

# Materialiserte Views

---

- ◆ Dersom et view brukes veldig ofte kan det lønne seg å materialisere det
- ◆ Et materialisert view lagres som en vanlig tabell på disk
- ◆ De er derfor like effektive å kjøre spørringer mot som en vanlig tabell
- ◆ Lages slik:

```
CREATE MATERIALIZED VIEW person_alder AS
SELECT navn,
       fødselsdato,
       EXTRACT(year FROM age(current_date, fødselsdato)) AS alder
FROM person
```

- ◆ Men, den kan enkelt oppdateres når de tabellene den avhenger av oppdateres
- ◆ Dette skjer derimot ikke automatisk, man må kjøre følgende for å oppdatere det:

```
REFRESH MATERIALIZED VIEW person_alder;
```

Takk for nå!

---

Neste video vil se på SQL-scripts og transaksjoner.