UKE 5

Gruppe 4

Plan

- Grunnleggende SQL med eksempler
- Lab:Oblig eller ukesoppgaver

Slik logger man seg på databasene:

Fjerinnlogge inn på serverne på ifi:

- ssh -YC brukernavn@login.ifi.uio.no

- Logge inn på filmdatabasen:

psql -h dbpg-ifi-kurs03 -U brukernavn -d fdb

- Northwind-databasen:

psql -h dbpg-ifi-kurs03 -U brukernavn -d northwind

- Deres private database:

psql -h dbpg-ifi-kurs03 -U brukernavn -d brukernavn

Spørringer

SELECT -attributter / kolonner

FROM - tabeller

WHERE -betingelser>– velger(selekterer) alt som oppfyller kriteriet/betingelsene

Tabell: film(filmid, title, prodyear)

SELECT filmid FROM film

gir oss bare filmid

SELECT filmid, title FROM film

gir oss filmid og title

SELECT * FROM film

gir oss alle attributtene/kolonnene

SELECT DISTINCT

fjerner duplikater

SELECT med aggregering

Kan også bruke aggregering I select-delen.

- sum – summen
- avg – gjennomsnitt
- max – maksimum
- min – minimum
- count – teller alle rader (ikke null verdier)
- count(*) teller alle rader inkludert null verdier

Eksempler (filmid, title, prodyear):

– finne antall filmid i film-tabellen:

SELECT count(filmid)

– Finne den/de nyeste filmene i databasen (høyest tall):

SELECT max(prodyear)

film(filmid, title, prodyear): finn navn på alle filmene som er produsert i perioden 2000-2002

```
SELECT title FROM film

WHERE prodyear > 1999 AND prodyear < 2003;
```

Søke i tekst

- Bruker LIKE og %

SELECT title FROM film

WHERE title LIKE 'Harry Potter'; vil gi oss akkurat filmen(e) med den tittelen.

SELECT * FROM table_name WHERE column_name LIKE '%delstreng’;

'%Harry Potter', tittelen slutter på «Harry Potter»

'Harry Potter%', tittelen starter på "Harry Potter"

'%Harry Potter%', kommer noe før og etter 'Harry Potter'

NULL

NULL-verdier er ukjente verdier. Kan ikke bruke "=" eller "LIKE" for å

finne disse. Man bruker:

• IS NULL og IS NOT NULL

Eksempel:

```
SELECT * from film
WHERE prodyear IS NULL
LIMIT 3;
```

- Da får vi alle filmene der prodyear mangler

JOINS

- INNER JOIN
- NATURAL JOIN
- SELF JOIN

Flere JOINS senere

INNER JOINS

Filmtabellen har denne fremmednøkkelen:

film(filmid) --> filmitem(filmid)

Med INNER JOIN så joiner man på en condition.

```
SELECT * from film f
INNER JOIN filmitem fi ON f.filmid = fi.filmid
WHERE prodyear = 2011;
```

NATURAL JOIN

Joiner på alle kolonner med likt navn.

film(filmid) --> filmitem(filmid),

Her ser vi at filmid har samme navn I begge tabellene, så her kunne vi brukt

natural join.

```
SELECT * FROM film f
NATURAL JOIN filmitem fi
WHERE prodyear = 2011;
```

Fjerner vekk de dupliserte kolonnene

I resultatet av naturlig join vil det aldri finnes kolonner med likt navn

Self Join

Self join kombinerer rader fra samme tabell basert på en relasjon mellom kolonnene i tabellen.

Opprett to eller flere alias av samme tabell innenfor en SELECT-setning

Nestede spørrringer

```
SELECT <kolonner>
FROM (
    SELECT <kolonner>
    FROM <tabell>
    WHERE <betingelse>
    ) AS tab
WHERE <betingelse>
```

Eksempel

For å finne antall unike kombinasjoner av land og by for alle kunder:

```
SELECT count(*)
FROM (SELECT DISTINCT country , city FROM customers ) AS d
```

Nyttige kommandoer

- \q – logger ut fra PostgreSQL
- \d – lister opp tabeller (relasjoner)
- \d <tabellnavn> – lister opp kolonner og skranker for tabellen
- \i – leser fila som input
- \pset format wrapped – fikser dårlig output

Kjøre fil: psql -h dbpg-ifi-kurs03 –d >database< -a –f >filnavn<