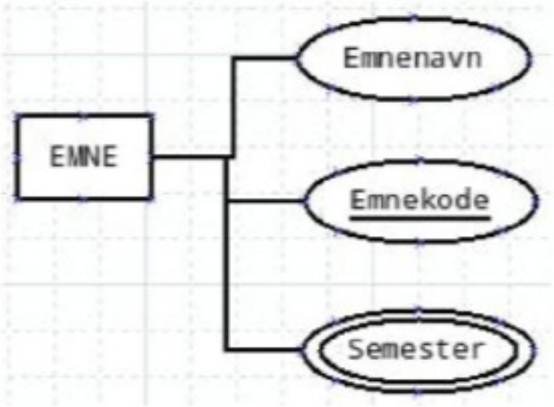
Gruppetime 5 - SQL for beginners

Av Katrine <3



Emne x Semester

Emnekode	Emnenavn	Semester
IN2090	Databaser	H2019
IN2120	Sikkerhet	H2019
IN2010	Algoritmer	H2019
IN2090	Databaser	H2020
IN2120	Sikkerhet	H2020
IN2010	Algoritmer	H2020

Emne(Emnekode, Emnenavn)

Semester(Semester, Emnekode)

Emne

Emnekode	Emnenavn
IN2090	Databaser
IN2120	Sikkerhet
IN2010	Algoritmer

Semester

Semester	Emnekode
H2019	IN2090
H2019	IN2120
H2019	IN2010
H2020	IN2090
H2020	IN2120
H2020	IN2010

Hvordan ta i bruk databasen?

- Dere fikk et brukernavn og passord...
- Det skal dere ikke bruke helt enda :3
- For å komme seg inn på filmdatabasen, bruk
- psql -h dbpg-ifi-kurs01 -U [brukernavnet ditt] -d fdb (fdb står for filmdatabase)

Kommandoer for databasen

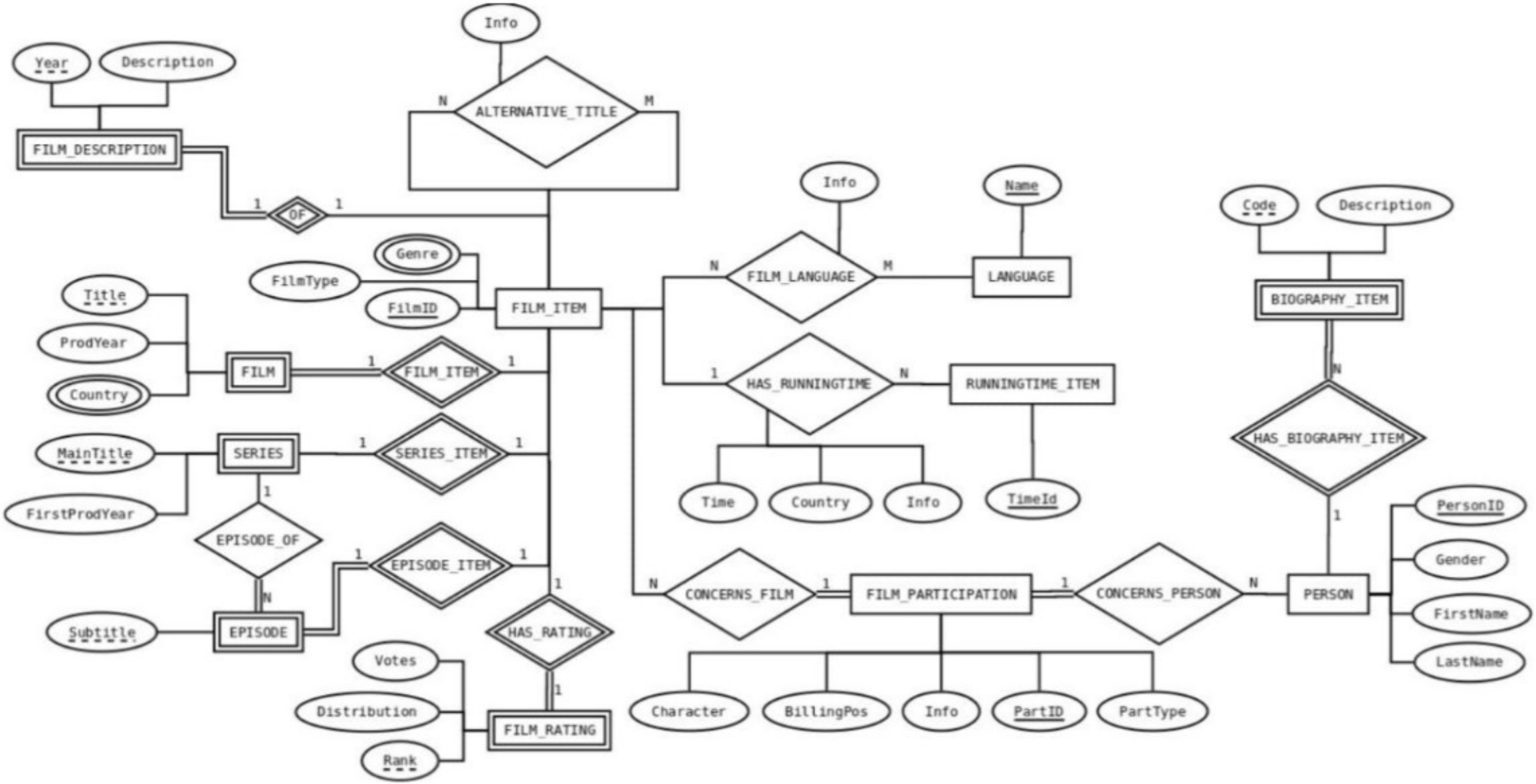
```
fdb=> \d film
      Table "public.film"
  Column  |  Type   | Modifiers
----------+---------+-----------
 filmid   | integer |
 title    | text    | not null
 prodyear | integer |
Indexes:
    "filmpkey" UNIQUE CONSTRAINT, btree (filmid)
    "filmtitleindex" btree (title)
    "filmyearindex" btree (prodyear)
Foreign-key constraints:
    "filmfkey" FOREIGN KEY (filmid) REFERENCES filmitem(filmid)
```

- \q - logg ut
- \h - hjelp til SQL
- \? - SQL kommandoer
- \d - tabellene/relasjonene til databasen
- \d <tabell> - gir infromasjon om den gitte tabellen. Bildet er fra tabellen "film"
- \i <filnavn> - leser den gitte filen som input
- \e - lar deg åpne en teksteditor som du kan skrive i, for så å kjøre scriptet når du lagrer

```
fdb=> \d
                List of relations
 Schema |          Name           | Type  |  Owner
--------+-------------------------+-------+----------
 public | alternativefilmtitle    | table | fdb_user
 public | biographyitem           | table | fdb_user
 public | country                 | table | fdb_user
 public | episode                 | table | fdb_user
 public | film                    | table | fdb_user
 public | filmcharacter           | table | fdb_user
 public | filmcountry             | table | fdb_user
 public | filmdescription         | table | fdb_user
 public | filmgenre               | table | fdb_user
 public | filmitem                | table | fdb_user
 public | filmlanguage            | table | fdb_user
 public | filmlanguageinfo        | table | fdb_user
 public | filmparticipation       | table | fdb_user
 public | filmparticipationinfo   | table | fdb_user
 public | filmrating              | table | fdb_user
 public | genre                   | table | fdb_user
 public | language                | table | fdb_user
 public | person                  | table | fdb_user
 public | runningtime             | table | fdb_user
 public | runningtimeinfo         | table | fdb_user
 public | series                  | table | fdb_user
(21 rows)
```

Hvorfor vi liker ER-diagrammer :)

- På forrige slide så dere at vi kan se hva som er attributter, nøkler og fremmednøkler til en tabell
- Se på bildet
- Lykke til med å skulle huske alt det for alle disse :)
- Heldigvis finnes det en lett måte å holde orden på dette...


Info
Year
Description
ALTERNATIVE_TITLE
N
M
FILM_DESCRIPTION
1
OF
1
Genre
FilmType
FilmID
FILM_ITEM
Info
Name
N
FILM_LANGUAGE
M
LANGUAGE
Code
Description
Title
ProdYear
Country
FILM
1
FILM_ITEM
1
1
HAS_RUNNINGTIME
N
RUNNINGTIME_ITEM
BIOGRAPHY_ITEM
N
HAS_BIOGRAPHY_ITEM
Time
Country
Info
TimeId
MainTitle
FirstProdYear
SERIES
1
SERIES_ITEM
1
1
EPISODE_OF
1
N
EPISODE_ITEM
1
1
N
CONCERNS_FILM
1
FILM_PARTICIPATION
1
CONCERNS_PERSON
N
PERSON
1
PersonID
Gender
FirstName
LastName
Subtitle
EPISODE
HAS_RATING
Votes
Distribution
Rank
FILM_RATING
1
Character
BillingPos
Info
PartID
PartType

Viktige SQL ord for å skrive spørringer for dere

- SELECT -> Henter informasjon, formaterer informasjonen
- FROM -> Tabellen/e informasjonen hentes fra, her dere som oftest vil joine tabeller, med mindre en er implisitt
- WHERE -> Sier hvilke verdier du vil hente ut

```
fdb=> select max(seriesid) AS highest, min(seriesid) AS lowest, avg(s
 highest  | lowest |       average        | number_of |    total
----------+--------+----------------------+-----------+-------------
 15707930 |     43 | 1499332.576225016913 |    446402 | 669305060692
(1 row)
```

SELECT

- Kan skrive inn flere kolonnenavn
- Med AS kan du endre navnet på en kolonne, spesielt nyttig hvis du endrer verdien på kolonnen
- Med || '' || kan du slå sammen kolonner til en kolonne
- select episodeid || '' || seriesid AS kombinert, subtitle from episode;
- Vi kan aggreggere med count, max, min, avg, sum etc
- select max(seriesid) AS high, min(seriesid) AS low, avg(seriesid) AS avg, count(seriesid) AS num, sum(seriesid) AS total from episode;
- Count(*) teller rader
- Med DISTINCT fjernes duplikater

Mentimeter

WHERE

```
fdb=> select * from episode where subtitle SIMILAR TO '%(T|t)he%';
 episodeid | seriesid |
-----------+----------+-------------------------------------------
   3002469 |     1716 | Operation: Steal the Safe (#1.11)
   3794856 |     1755 | The Positive Negative (#3.20)
   6567874 |     1755 | The Catspaw Caper (#5.6)
   3001718 |     2343 | The Black Crowes (#2.29)
     97286 |     2360 | The Way We Were (#6.9)
  15676346 |     2629 | The Twins Visit (#1.39)
   5774725 |     2987 | The Tower (#1.2)
   5774773 |     2987 | The Eye of the Needle (#1.9)
    889569 |     3163 | The Flag Flies Black (#1.17)
    625138 |     3291 | The Bandit of Brittany (#2.21)
    625154 |     3291 | The Ordeal (#1.10)
   1153457 |     3291 | The Champion (#4.20)
```

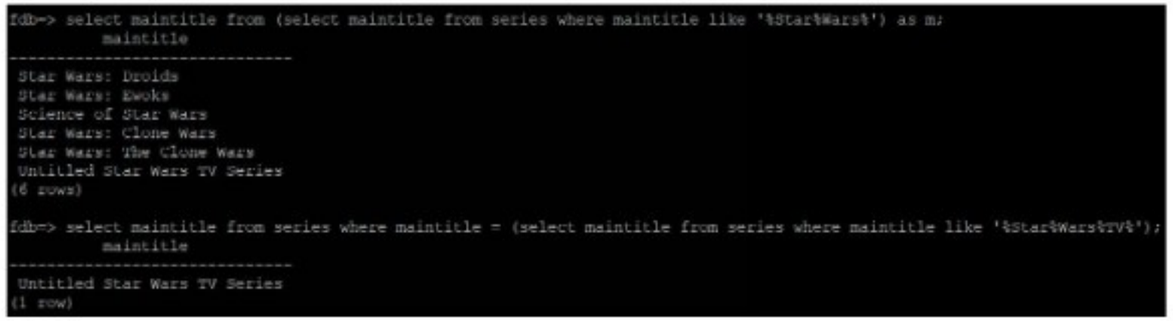
- Her du spesifiserer hva du vil se av informasjon i from av uttrykk
- Den delen som ligner mest på programmering som dere er vant til
- where (seriesid = 43 AND episodeid > 13) OR (seriesid < 100 AND episodeid = 100)
- For å søke på tekst brukes LIKE
- '[streng]%' gir alle stringer som starter med [streng], mens '%[streng]' skal slutte med ordet, og '%[streng]%' skal bare inneholde det
- select * from episode where subtitle like 'The%';
- For å få alt som har "the" i seg, både med små og store bokstaver skriver vi SIMILAR TO '%(T|t)he%'. Da skal vi få alt som inneholder "The" eller "the
- NOT er det samme som alltid, hvor vi vil ha alt som IKKE inneholder den gitte verdien
- For å bruke den ukjente verdien NULL brukes IS NULL for de som ikke har verdi, og NOT NULL for de som har verdi

FROM og JOINS

```
fdb-> select s.maintitle, e.episodeid from episode AS e, series AS s where e.seriesid = s.seriesid;
                                    maintitle                                     | episodeid
----------------------------------------------------------------------------------+-----------
$10,000 Pyramid, The                                                              |   7887002
$10,000 Pyramid, The                                                              |   7887018
$10,000 Pyramid, The                                                              |   7887034
$10,000 Pyramid, The                                                              |   7887050
$10,000 Pyramid, The                                                              |   7887066
$10,000 Pyramid, The                                                              |   7887082
$10,000 Pyramid, The                                                              |   7887098
$10,000 Pyramid, The                                                              |   7887114
$10,000 Pyramid, The                                                              |   7887130
$10,000 Pyramid, The                                                              |   7887146
$10,000 Pyramid, The                                                              |   7887162
$10,000 Pyramid, The                                                              |   7887178
$10,000 Pyramid, The                                                              |   7887194
$10,000 Pyramid, The                                                              |   7887210
$10,000 Pyramid, The                                                              |   7887226
$10,000 Pyramid, The                                                              |   7887242
```

- I FROM kan vi krysse flere tabeller med hverandre
- Vi kan implisitt joine ved å si FROM a, b WHERE a.attributt = b.attributt
- Natural join joiner på ALLE kolonner med likt navn, og fjerner duplikater og tupler som ikke har en ekvivalent i den andre tabellen
- FROM a NATURAL JOIN b
- Dele vil lære flere joins senere

Nøstede spørringer



```
fdb=> select maintitle from (select maintitle from series where maintitle like '%Star%Wars%') as m;
          maintitle
-----------------------------
 Star Wars: Droids
 Star Wars: Ewoks
 Science of Star Wars
 Star Wars: Clone Wars
 Star Wars: The Clone Wars
 Untitled Star Wars TV Series
(6 rows)

fdb=> select maintitle from series where maintitle = (select maintitle from series where maintitle like '%Star%Wars%TV%');
          maintitle
-----------------------------
 Untitled Star Wars TV Series
(1 row)
```

- Nøstede spørringer er hvor du du bruker en annen spørring i spørringen
- Dette kan gjøres i både FROM og WHERE
- select count(maintitle) from (select maintitle from series where maintitle like '%Star%Wars%') as m;
- Note: Vi henter ut en ny tabell, som gjør at vi må gi den et navn
- select maintitle from series where maintitle = (select maintitle from series where maintitle like '%Star%Wars%TV%');
- Note: Den skal sammenligne noe, som betyr at den bare kan returnerer ett svar, ikke en tabell
- Med WITH kan vi bruke samme delspørring om igjen

QUIZ :3

SELECT fornavn FROM person; gir oss...

Mentimeter

SELECT DISTINCT fornavn FROM person; gir oss...

Mentimeter

SELECT fornavn || ' ' || etternavn AS navn FROM person; gir oss...

Mentimeter

SELECT fornavn FROM person where fornavn LIKE '%k%'; gir oss...

Mentimeter

Person har kolonnene fornavn, etternavn og personid.
Ansatt har kolonnene fornavn, etternavn og ansattid.
Hvilke kolonner joines ved NATURAL JOIN

Mentimeter

Hvilke spørringer gir feilmelding?

Mentimeter

✓	✗	✓
SELECT * FROM (SELECT * FROM person);	SELECT * FROM (SELECT * FROM person) AS p;	SELECT * FROM person WHERE fornavn = (SELECT * FROM person);

Leaderboard

Mentimeter

No results yet

Top Quiz participants will be displayed here once there are results!

Spørsmål stilles her :)

Mentimeter