

# Øving 3, TDT4145

## Oppgave 1: Lage SQL-tabeller og legge inn data

- a) Via FOREIGN KEY CONSTRAINTS. Hvis man ON DELETE CASCADE vil den slette alle relasjonene som bruker attributtet. Dette må gjøres i SkuespillerIFilm og SjangerForFilm.

b)

```
1 use task1;
2 CREATE TABLE Regissor(
3     RegissorID INT NOT NULL,
4     Navn VARCHAR(50) NOT NULL,
5     CONSTRAINT RegissorPK PRIMARY KEY(RegissorID)
6 );
7
8 CREATE TABLE Film(
9     FilmID INT NOT NULL,
10    Title VARCHAR(50) NOT NULL,
11    Produksjonar INT NOT NULL,
12    RegissorID INT NOT NULL,
13    CONSTRAINT FilmPK PRIMARY KEY (FilmID),
14    CONSTRAINT FilmFK FOREIGN KEY (RegissorID) REFERENCES Regissor(RegissorID) ON UPDATE CASCADE ON DELETE CASCADE
15 );
16
17 CREATE TABLE Sjanger(
18     SjangerID INT NOT NULL,
19     Navn VARCHAR(50) NOT NULL,
20     Beskrivelse VARCHAR(100) NOT NULL,
21     CONSTRAINT SjangerPK PRIMARY KEY(SjangerID)
22 );
23
24 CREATE TABLE Skuespiller(
25     SkuespillerID INT NOT NULL,
26     Navn VARCHAR(50) NOT NULL,
27     Fodselsar INT NOT NULL,
28     CONSTRAINT SkuespillerPK PRIMARY KEY (SkuespillerID)
29 );
30
31 CREATE TABLE SkuespillerFilm(
32     FilmID INT NOT NULL,
33     SjangerID INT NOT NULL,
34     CONSTRAINT SjangerForFilmPK PRIMARY KEY(FilmID, SjangerID),
35     CONSTRAINT SjangerForFilmFK1 FOREIGN KEY(FilmID) REFERENCES Film(FilmID) ON UPDATE CASCADE ON DELETE CASCADE,
36     CONSTRAINT SjangerForFilmFK2 FOREIGN KEY(SjangerID) REFERENCES Sjanger(SjangerID) ON UPDATE CASCADE ON DELETE CASCADE
37 );
38
39
40
```

```
INSERT INTO Regissor VALUES(1, "Peytod Read");  
INSERT INTO Regissor VALUES(2, "Tom Shadyac");  
INSERT INTO Film VALUES (1, "Yes Man", 2008, 1);  
INSERT INTO Skuespiller VALUES(1, "Jim Carrey", 1962);  
INSERT INTO SkuespillerIFilm VALUES(1, 1, "Carl");
```

c)

d)

```
UPDATE Skuespiller SET Navn = "James Eugene Carrey" WHERE SkuespillerID = 1;
```

e)

```
DELETE FROM Regissor  
WHERE RegissorID = 2;
```

## Oppgave 2: Spørring i SQL

```
#Oppgave A
SELECT FilmID, Title, Produksjonsar, RegissorID FROM Film;

#Oppgave B
SELECT Navn FROM Skuespiller WHERE Fodselsar > 1960;

#Oppgave C
SELECT Navn FROM Skuespiller WHERE Fodselsar > 1979 AND Fodselsar < 1990;

#Oppgave D
SELECT Film.Title, Sif.Rolle AS Rolle FROM Film INNER JOIN SkuespillerFilm AS Sif ON (Film.FilmID = Sif.FilmID) INNER JOIN Skuespiller
AS S ON (Sif.SkuespillerID = S.SkuespillerID) WHERE S.Navn = "Morgan Freeman";

#Oppgave E
SELECT DISTINCT Title FROM Film INNER JOIN Regissor ON (Film.RegissorID = Regissor.RegissorID) WHERE Regissor.Navn
IN (SELECT Navn FROM Film INNER JOIN SkuespillerIFilm AS Sif ON (Film.FilmID = Sif.FilmID)
INNER JOIN Skuespiller AS S ON (S.SkuespillerID = Sif.SkuespillerID)
WHERE Regissor.RegissorID = Film.RegissorID);

#Oppgave F
SELECT COUNT(*) AS Antall FROM Skuespiller WHERE Navn LIKE "C%";

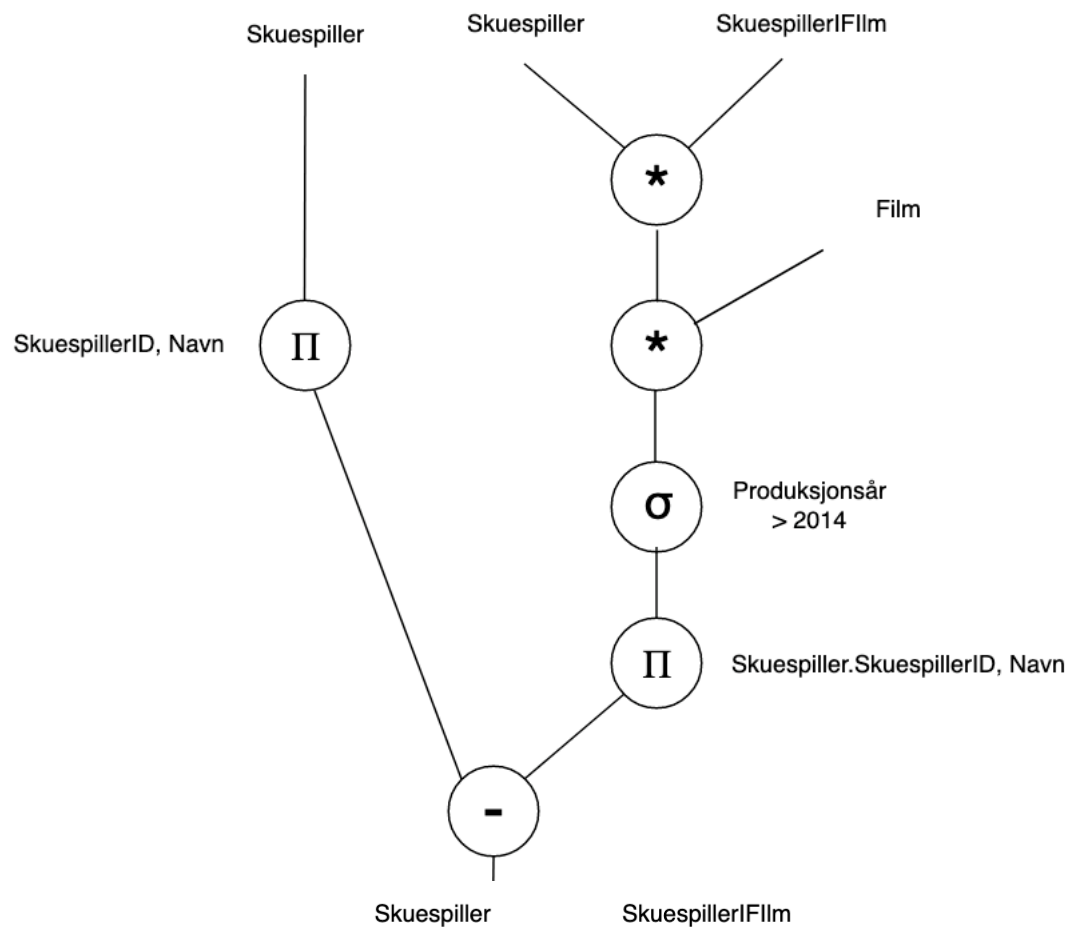
#Oppgave G
SELECT Navn, COUNT(*) AS Antall FROM Sjanger INNER JOIN SjangerForFilm ON (Sjanger.SjangerID = SjangerForFilm.SjangerID) GROUP BY Navn;

#Oppgave H
SELECT S.Navn FROM Skuespiller AS S INNER JOIN SkuespillerFilm AS Sif ON (S.SkuespillerID = Sif.SkuespillerID)
INNER JOIN Film ON (Film.FilmID = Sif.FilmID)
WHERE Film.Title = "Ace Ventura: Pet Detective" AND
S.SkuespillerID NOT IN (SELECT SkuespillerID FROM SkuespillerFilm INNER JOIN Film
ON (SkuespillerIFilm.FilmID = Film.FilmID) WHERE Film.Title = "Ace Ventura: When Nature Calls");

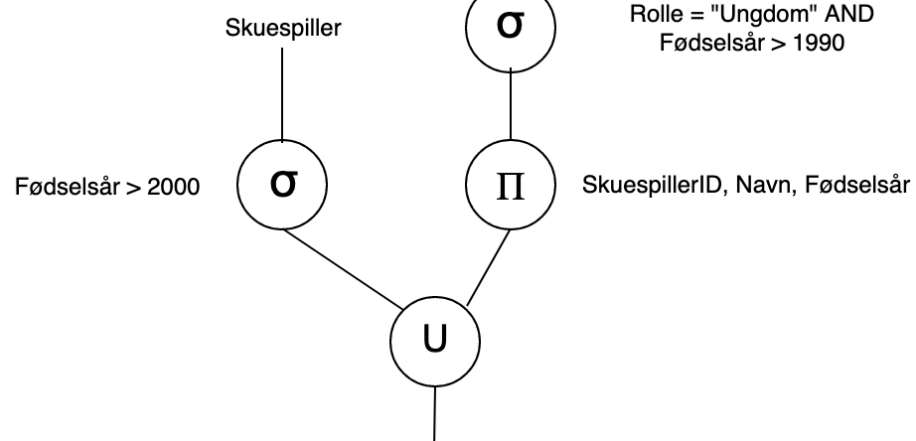
#Oppgave I
SELECT Film.Title, Film.FilmID, AVG(S.Fodselsar) FROM Film INNER JOIN SkuespillerIFilm AS Sif
ON (Film.FilmID = Sif.FilmID) INNER JOIN Skuespiller AS S ON (Sif.SkuespillerID = S.SkuespillerID)
GROUP BY Film.FilmID HAVING AVG(S.Fodselsar) > (SELECT AVG(Fodselsar) FROM Skuespiller ) ORDER BY Film.Title ASC;
```

### Oppgave 3: Flere spørringer i Relasjonsalgebra

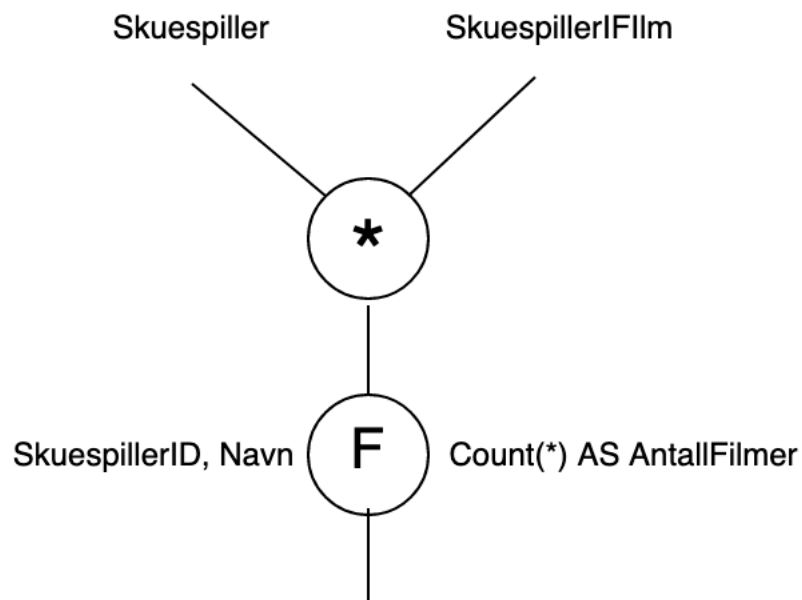
a)



b)



c)



#### Oppgave 4: Introduksjon til normaliseringsteori

- a) To attributter oppdateres for hver tuppel i tabellen som endres.
- b) PersonID -> Navn, Telefonnr, Fakultetskode

Fakultetskode -> Fakultetsnavn, Fakultetsbygg

Deler tabellen slik:

Ansatt(PersonID, Navn, Telefonnr, Fakultetskode)

Fakultet(Fakultetskode, Fakultetsnavn, Fakultetsbygg)

#### Oppgave 5: Funksjonelle avhengigheter, nøkler og tillukning

a)

- 1) A -> A : Ja
- 2) A -> B : Kanskje
- 3) A -> C : Nei
- 4) AB -> C : Nei
- 5) C -> D : Nei
- 6) D -> C : Nei
- 7) Sant
- 8) Kanskje
- 9) Usant
- 10) Kanskje

Gruppe 164

Marius Hauge Tobiassen, Tor Berre

b)  $R = \{A, B, C, D\}$  og  $F = \{A \rightarrow C, B \rightarrow D, ABC \rightarrow D\}$

$$A^+ = AC$$

$$D^+ = D$$

$$BC^+ = BCD$$

$$AB^+ = ABCD$$