

### Øving 3

#### Oppgave 1:

- a) Kan gjøres ved: CASCADE. Eksempel

```
CREATE TABLE Song(  
  songID INTEGER PRIMARY KEY,  
  artistID INTEGER NOT NULL,  
  name TEXT NOT NULL,  
  duration INTEGER NOT NULL,  
  year INTEGER NOT NULL,  
  FOREIGN KEY (artistID) REFERENCES Artist(artistID)  
    ON UPDATE CASCADE  
    ON DELETE CASCADE);
```

- b) Tolket det som om det skal gjøres for alle som bruker artistID, altså også album. Der alle vil ha CASCADE.

```
sqlite> CREATE TABLE Artist(  
  ...> name TEXT NOT NULL,  
  ...> artistID INTEGER PRIMARY KEY);
```

```
CREATE TABLE Album(  
  albumID INTEGER PRIMARY KEY,  
  name TEXT NOT NULL,  
  year INTEGER NOT NULL,  
  artistID TEXT NOT NULL,  
  FOREIGN KEY (artistID) REFERENCES Artist(artistID)  
    ON UPDATE CASCADE  
    ON DELETE CASCADE  
);
```

```
CREATE TABLE FeaturedOn(  
  artistID INTEGER NOT NULL,  
  songID INTEGER NOT NULL,  
  FOREIGN KEY (artistID) REFERENCES Artist(artistID)  
    ON UPDATE CASCADE  
    ON DELETE CASCADE  
  FOREIGN KEY (songID) REFERENCES Song(songID)  
    ON UPDATE CASCADE  
    ON DELETE CASCADE);  
CREATE TABLE SongOnAlbum(  
  songID INTEGER NOT NULL,  
  albumID INTEGER NOT NULL,  
  FOREIGN KEY (songID) REFERENCES Song(songID)  
    ON UPDATE CASCADE  
    ON DELETE CASCADE  
  FOREIGN KEY (albumID) REFERENCES Album(albumID)  
    ON UPDATE CASCADE  
    ON DELETE CASCADE);
```

```
CREATE TABLE Song(  
  songID INTEGER PRIMARY KEY,  
  artistID INTEGER NOT NULL,  
  name TEXT NOT NULL,  
  duration INTEGER NOT NULL,  
  year INTEGER NOT NULL,  
  FOREIGN KEY (artistID) REFERENCES Artist(artistID)  
    ON UPDATE CASCADE  
    ON DELETE CASCADE);
```

- c)

```
INSERT INTO Artist (name, artistID) VALUES ('DuaLipa', 1);
```

```

INSERT INTO Artist (name, artistID) VALUES ('DaBaby', 2);
INSERT INTO Album (albumID, name, year, artistID) VALUES (1, 'Future Nostalgia', 2022, 1);
INSERT INTO Song (songID, artistID, name, duration, year) VALUES (1, 1, 'Levitating', 203, 2022);
INSERT INTO SongOnAlbum (songID, albumID) VALUES (1, 1);
INSERT INTO FeaturedOn (artistID, songID) VALUES (2, 1);

```

d)

```

UPDATE Artist
SET name = 'Jonathan Lyndale Kirk'
WHERE artistID = 2;

```

e)

```

DELETE FROM Artist
WHERE artistID = 2;

```

## Oppgave 2:

a)

```

sqlite> select songID, name, duration, year, artistID
...> from song;

```

songID	name	duration	year	artistID
1	Saved	178	2015	1
2	Oops!... I Did It Again	221	2000	2
3	Don't Start Now	183	2019	3
4	Strangers	233	2017	4
5	I Went Too Far	294	2016	5
6	Blasé	286	2015	1
7	Hot Girl Summer	199	2019	9
8	drivers license	242	2021	13
9	Black Beatles	291	2016	8
10	Nice for What	204	2018	12
11	Work	219	2016	15
12	I'm the One	288	2017	16
13	WAP	187	2020	11
14	Watermelon Sugar	173	2020	21
15	positions	172	2020	22
16	7 rings	178	2019	22
17	thank u, next	217	2018	22
18	Don't Judge Me	242	2017	1
19	Love U Better	183	2017	1
20	Savage Remix	242	2020	9
21	Cardigan	232	2020	26
22	Butter	265	2021	27
23	good 4 u	178	2021	13
24	Love Me	255	2013	20

b)

```

sqlite> select * from album where year < 2017
...> ;

```

albumID	name	year	artistID
1	Free TC	2015	1
2	Oops!... I Did It Again	2000	2
5	All My Demons Greeting Me as a Friend	2016	5
7	SremmLife 2	2016	8
9	ANTI	2016	15
17	I Am Not a Human Being II	2013	20

c)

```
sqlite> select * from song where year between 2018 and 2020
...> order by year asc
...> ;
```

songID	name	duration	year	artistID
10	Nice for What	204	2018	12
17	thank u, next	217	2018	22
3	Don't Start Now	183	2019	3
7	Hot Girl Summer	199	2019	9
16	7 rings	178	2019	22
13	WAP	187	2020	11
14	Watermelon Sugar	173	2020	21
15	positions	172	2020	22
20	Savage Remix	242	2020	9
21	Cardigan	232	2020	26

- d) Sammenligner kolonnen for artistID i artist-tabell med kolonnen for artistID i featuredOn tabell, samt sammenligner kolonnen for songID i song tabell med kolonnen for songID i featuredOn tabell og henter ut dataen som oppfyller én av de to sammenligningene. Da ønsker vi å hente ut navnet til enten sangen, ELLER artisten for raden som oppfyller en av disse to sammenligningene.

```
sqlite> select distinct artist.name, song.name
...> from artist, song, featuredOn
...> where artist.artistID = featuredOn.artistID and artist.artistID = song.artistID or song.songID = featuredOn.songID
...> and artist.artistID = song.artistID;
```

name	name
Ty Dolla Sign	Saved
Ty Dolla Sign	Blasé
Megan Thee Stallion	Hot Girl Summer
Rae Sremmurd	Black Beatles
Drake	Nice for What
Rihanna	Work
DJ Khaled	I'm the One
Cardi B	WAP
Ty Dolla Sign	Don't Judge Me
Ty Dolla Sign	Love U Better
Megan Thee Stallion	Savage Remix
Lil Wayne	Love Me

- e) Kan sortere på bakgrunn av flere parametere, men vil tro de nevnes i prioritert rekkefølge. Altså vil vi først ta hensyn til year, om det er likt, se på album navnet, og til sist sang navnet.

```
sqlite> select distinct song.name, album.name, song.year
...> from song, album
...> where song.artistID = 22 and album.artistID = 22
...> order by song.year, album.name, song.name;
```

name	name	year
thank u, next	Positions	2018
thank u, next	thank u, next	2018
7 rings	Positions	2019
7 rings	thank u, next	2019
positions	Positions	2020
positions	thank u, next	2020

- f) Ble greiere når jeg tegna det opp før proggingen.

```
sqlite> select distinct song.name, artist.name
...> from song, artist, featuredOn
...> where featuredOn.artistID = 1 and featuredOn.songID = song.songID and artist.name = 'Ty Dolla Sign' or song.artistID = 1 and a
rtist.name = 'Ty Dolla Sign';
```

name	name
Saved	Ty Dolla Sign
Blasé	Ty Dolla Sign
Hot Girl Summer	Ty Dolla Sign
Don't Judge Me	Ty Dolla Sign
Love U Better	Ty Dolla Sign

- g) Henter ut navn på artist og sangen, til de sangtitlene som inneholder 'the'.

```
sqlite> select artist.name, song.name, instr(song.name, 'the')
...> from artist, song
...> where instr(song.name, 'the')>0 and artist.artistID = song.songID;
```

name	name	instr(song.name, 'the')
Drake	I'm the One	5

Generelt vil `instr(kolonne, 'søkeOrd')` returnere int som representerer index for hvor man finner starten på substrangen. Altså vil man få returnert 5 fordi ved index 5 starter substring «st».

```
1 SELECT instr('myteststring','st');
```

Sample Output:

```
instr('myteststring','st')
-----
5
```

Dermed ved å kreve condition der denne integer > 0, sier man altså at vi avhenger av at substrangen faktisk finnes (aka har en index).

- h) Usikker på hvordan jeg må løse 'h'.

### Oppgave 3:

- a) Mener de at det skal opprettes en *ny* tabell med resultatene fra spørringen og at navnet og kolonnene skal hete spesifikke ting? Har i hvert fall fått hentet rett info – mangler rename.

```
sqlite> select Artist.name, Song.name
...> from Artist, Song, SongOnAlbum
...> where Song.songID != SongOnAlbum.songID and Song.artistID = Artist.artistID;
```

name	name
DuaLipa	New Rules
Christina Aguilera	Cruz

- b) Hadde nesten ingenting i databasen fra task 1, så brukte task 2 sin database sisteden..?

```
sqlite> select distinct artist.name, instr(artist.name, 'B')
...> from artist, song, featuredOn
...> where instr(artist.name, 'B') = 1 and song.year >= 2020 and song.artistID = artist.artistID
...> or song.year between 2000 and 2010 and song.artistID = artist.artistID;
```

name	instr(artist.name, 'B')
Britney Spears	1
BTS	1

- c)

### Oppgave 4:

- a) I så fall må vi oppdatere 8 celler minst.
- b) En mulig løsning er å splitte tabellen inn i flere tabeller som ved normaliseringsteorien. Mer spesifikt kunne man delt tabellen inn i følgende deltabeller:

Director  
DirectorID  
 Director name  
 Director Birth

Film  
FilmID  
 name  
 year  
 Smidde nøkkel: DirectorID

#### Oppgave 5:

- a) Følgende kan umulig stemme:
- 2:  $A \rightarrow B$  fordi a3 går til både b4 og b3
- 5:  $C \rightarrow D$  fordi c1 går til både d1 og d2
- 6:  $D \rightarrow C$  fordi d2 går til både c1 og c2
- 8: For at ABC skal kunne være en supernøkkel må  $ABC \rightarrow D$ , noe som ikke stemmer fordi a1b1c1 går til både d1 og d2
- b) Utledet svaret under:

$R = \{A, B, C, D\}$ , der  $F = \{D \rightarrow A, B \rightarrow D, ABD \rightarrow C\}$

$D^+ = D$   
 $= DA$

$BC^+ = BC$   
 $BCD$

$AB^+ = AB$   
 $= ABDC = R$

$BD^+ = BD$   
 $= BDA$   
 $= BDAC = R$

De som har R er  $AB^+$  og  $BD^+$

---