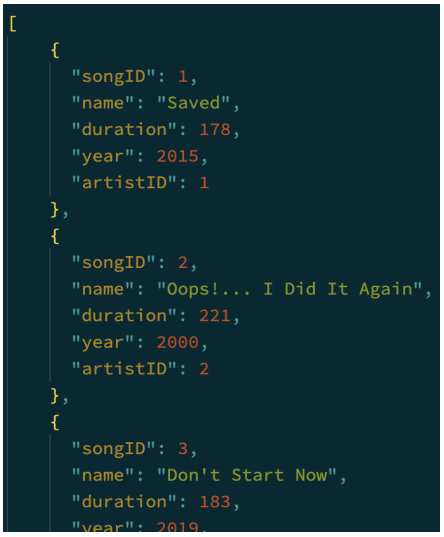
Oppgave 1: SELECT-spørringer i SQL

a)

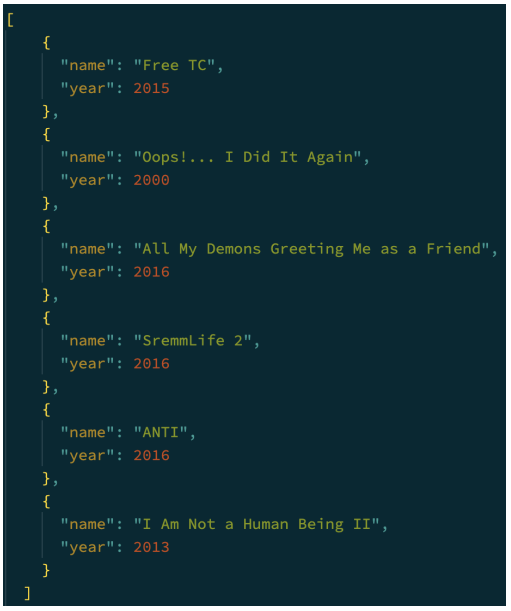
```
Run on active connection | Select block
SELECT
    *
FROM
    song
```



Denne var veldig lang så jeg kuttet den av, men den ga ut 24 objekter.

b)

```
Run on active connection | Select block
SELECT
    name,
    year
FROM
    album
WHERE
    year < 2017
```



c)

```
Run on active connection | Select block
SELECT
    name,
    year
FROM
    song
WHERE
    year BETWEEN 2018
    AND 2020
ORDER BY
    year
```


```
[
    {
      "name": "Nice for What",
      "year": 2018
    },
    {
      "name": "thank u, next",
      "year": 2018
    },
    {
      "name": "Don't Start Now",
      "year": 2019
    },
    {
      "name": "Hot Girl Summer",
      "year": 2019
    },
    {
      "name": "7 rings",
      "year": 2019
    },
    {
      "name": "WAP",
      "year": 2020
    },
    {
      "name": "Watermelon Sugar",
      "year": 2020
    },
    {
      "name": "positions",
      "year": 2020
    },
    {
      "name": "Savage Remix",
      "year": 2020
    },
    {
      "name": "Cardigan",
      "year": 2020
    }
  ]
```


d)

```
Run on active connection | Select block
SELECT
    artist.name AS artist,
    song.name AS song
FROM
    artist
    JOIN featuredOn ON artist.artistID = featuredOn.artistID
    JOIN song ON featuredOn.songID = song.songID
WHERE
    NOT artist.artistID = song.artistID
ORDER BY
    artist,
    song;
```

```
[
    {
      "artist": "Beyoncé",
      "song": "Savage Remix"
    },
    {
      "artist": "Chance the Rapper",
      "song": "I'm the One"
    },
    {
      "artist": "Drake",
      "song": "Love Me"
    },
    {
      "artist": "Drake",
      "song": "Work"
    },
    {
      "artist": "E-40",
      "song": "Saved"
    },
    {
      "artist": "Future",
      "song": "Blasé"
    },
    {
      "artist": "Future",
      "song": "Don't Judge Me"
```


Denne er også lang, fikk ut 19 objekter.

e)


```
[
    {
      "song": "thank u, next",
      "album": "thank u, next",
      "year": 2018
    },
    {
      "song": "7 rings",
      "album": "thank u, next",
      "year": 2019
    },
    {
      "song": "positions",
      "album": "Positions",
      "year": 2020
    }
]
```

f)

```
Run on active connection | Select block
SELECT
    DISTINCT artist.name AS artist,
    song.name AS song
FROM
    artist
    JOIN song ON artist.artistID = song.artistID
    LEFT JOIN featuredOn ON song.songID = featuredOn.songID
WHERE
    artist.name = 'Ty Dolla Sign'
    OR featuredOn.artistID = (
        SELECT
            artistID
        FROM
            artist
        WHERE
            name = 'Ty Dolla Sign'
    )
ORDER BY
    artist.name,
    song.name;
```

```
[
    {
      "artist": "Megan Thee Stallion",
      "song": "Hot Girl Summer"
    },
    {
      "artist": "Ty Dolla Sign",
      "song": "Blasé"
    },
    {
      "artist": "Ty Dolla Sign",
      "song": "Don't Judge Me"
    },
    {
      "artist": "Ty Dolla Sign",
      "song": "Love U Better"
    },
    {
      "artist": "Ty Dolla Sign",
      "song": "Saved"
    }
]
```

g)

```
Run on active connection | Select block
SELECT
    artist.name AS artist,
    song.name AS song
FROM
    artist
    JOIN song ON artist.artistID = song.artistID
WHERE
    LOWER(song.name) LIKE '%the%'
ORDER BY
    artist.name,
    song.name;
```

```
[
    {
      "artist": "DJ Khaled",
      "song": "I'm the One"
    }
]
```

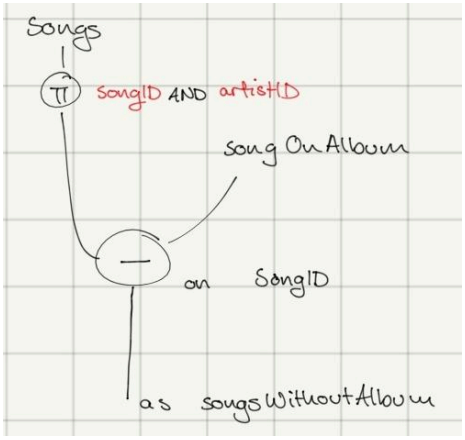
h)

```
Run on active connection | Select block
SELECT
    artist.name AS artist,
    COUNT(featuredOn.songID) AS guestAppearances
FROM
    artist
    JOIN featuredOn ON artist.artistID = featuredOn.artistID
GROUP BY
    artist.name
HAVING
    guestAppearances = (
        SELECT
            COUNT(featuredOn.songID)
        FROM
            featuredOn
        WHERE
            artistID = artist.artistID
    )
ORDER BY
    guestAppearances DESC
LIMIT
    1;
```

```
[
    {
      "artist": "Future",
      "guestAppearances": 3
    }
]
```

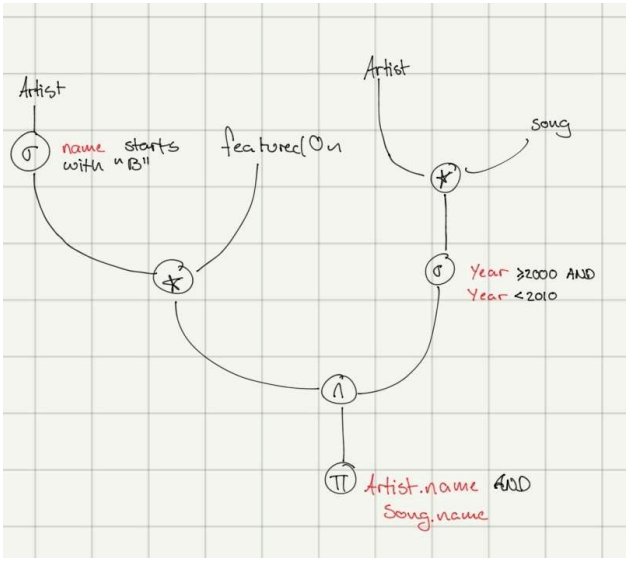
Oppgave 2: Flere spørringer i relasjonsalgebra

a)

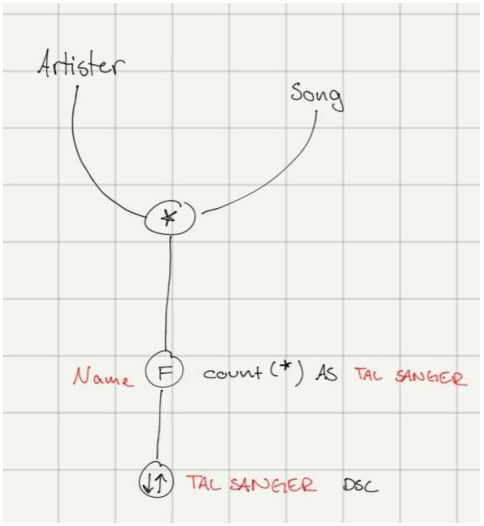


b)

Tegnet som kan se ut som ^ er et union tegn.



c)



Oppgave 3: Introduksjon til normaliseringsteori

a) Dersom vi må endre navnet til regissøren, må vi endre fire celler. Dersom vi må endre fødselsdatoen også, må vi endre fire celler til.

b) Alternativt design innebærer å lage to uavhengige tabeller, en med informasjon om regissør og en med informasjon om film, og en tabell som kobler sammen denne informasjonen

Regissør:

directorID	directorName	directorBirthYear
1	Jacques Táti	1908
2	Harmony Korine	1973
3	Cheryl Dunye	1966

Film:

filmID	name	year
1	PlayTime	1967
2	Mon Ocle	1953
3	Spring Breakers	2012
4	Monsieur Hulot’s Holiday	1953
5	Trafic	1971
6	The Watermelon Woman	1996

FilmAv:

filmID	regissørID
1	1
2	1
3	2
4	1
5	1
6	3

Oppgave 4: Funksjonelle avhengigheter, nøkler og tillukning

Vi vet blant annet at alle rader som har samme verdi for A må ha samme verdi for B, at en supernøkkel er en kombinasjon av attributter som unikt kan identifisere hver rad i en tabell, og at en kandidatnøkkel er en minimal supernøkkel, det vil si den minste mengden av attributter som kan unikt identifisere hver rad

a) Påstander:

1. Stemmer, trivielt.
2. Nei, B har ulike verdier for samme A.
3. Stemmer, C har unike verdier for unike verdier av A.
4. Stemmer, C har unike verdier for unike kombinasjoner av AB.
5. Nei, D har ulike verdier for samme C.
6. Nei, C har ulike verdier for samme D.
7. Ja, siden alle rader i tabellen er unike.
8. Nei, vi har duplikatrader som gir ulike verdier.
9. Nei, samme verdi for D gir ulike verdier for ABC.
10. Ja, ABD gir ulike verdier for D.

b) Tillukninger:

1. $D^+ = DA$
2. $BC^+ = BCD = BCDA = R$
3. $AB^+ = ABD = ABDC = R$
4. $BD^+ = BDA = BDAC = R$

Relasjonen har bare en kandidatnøkkel, $B^+ = BD = BDA = BDAC = R$.