

sql

February 19, 2024

1 Exercise 3

1.0.1 Group 14

```
[ ]: import sqlite3

dbfile = 'musikk.db'
con = sqlite3.connect(dbfile)

cur = con.cursor()

table_list = [a for a in cur.execute("SELECT name FROM sqlite_master WHERE type_
    ↳='table'")]

print(table_list)
```

```
[('artist',), ('song',), ('album',), ('featuredOn',), ('songOnAlbum',)]
```

- a) Write a query which returns the songID, title, duration, year and artistID for all songs in the database.

```
[ ]: table_list = [a for a in cur.execute("SELECT * FROM sqlite_master WHERE name =_
    ↳'featuredOn'")]
table_list
```

```
[ ]: [('table',
        'featuredOn',
        'featuredOn',
        5,
        'CREATE TABLE featuredOn(\nartistID INTEGER NOT NULL,\nsongID INTEGER NOT
        NULL,\nPRIMARY KEY(artistID, songID),\nFOREIGN KEY(artistID) REFERENCES
        artist(artistID) ON DELETE CASCADE,\nFOREIGN KEY(songID) REFERENCES song(songID)
        ON DELETE CASCADE)')]
```

```
[ ]: a = [a for a in cur.execute("SELECT songID, name, duration, year, artistID FROM_
    ↳song")]
a
```

```
[ ]: [(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) Write a query which returns the name and year of all albums released before 2017.

```
[ ]: b = [a for a in cur.execute("SELECT name, year FROM album WHERE year > 2017")]
      b
```

```
[ ]: [('Future Nostalgia', 2020),
      ('Sucker Punch', 2019),
      ('Sour', 2021),
      ('Scorpion', 2018),
      ('Fine Line', 2019),
      ('Positions', 2020),
      ('thank u, next', 2019),
      ('Good News', 2020),
      ('folklore', 2020)]
```

c) Write a query which returns the name and year of all songs released between 2018 and 2020 (inclusive), ordered by year.

```
[ ]: c = [a for a in cur.execute("SELECT name, year FROM album WHERE (year >= 2018)
      ↳AND (year <= 2020)")]
      c
```

```
[ ]: [('Future Nostalgia', 2020),
      ('Sucker Punch', 2019),
      ('Scorpion', 2018),
      ('Fine Line', 2019),
      ('Positions', 2020),
      ('thank u, next', 2019),
      ('Good News', 2020),
      ('folklore', 2020)]
```

- d) Write a query which returns the artist name and song name for all artists that have featured on a song (i.e. they are not the main artist), ordered by the artist name and song name.

```
[ ]: d= [a for a in cur.execute("""
                                SELECT artist.name AS artistname, song.name as_
                                ↪songname
                                From artist
                                JOIN FeaturedOn ON artist.artistID = featuredOn.
                                ↪artistID
                                JOIN song ON featuredOn.songID = song.songID
                                ORDER BY artist.name ASC
                                """)]
d
```

```
[ ]: [('Beyoncé', 'Savage Remix'),
      ('Chance the Rapper', "I'm the One"),
      ('Drake', 'Work'),
      ('Drake', 'Love Me'),
      ('E-40', 'Saved'),
      ('Future', 'Blasé'),
      ('Future', "Don't Judge Me"),
      ('Future', 'Love Me'),
      ('Gucci Mane', 'Black Beatles'),
      ('Justin Bieber', "I'm the One"),
      ('Lil Wayne', "I'm the One"),
      ('Lil Wayne', 'Love U Better'),
      ('Megan Thee Stallion', 'WAP'),
      ('Nicki Minaj', 'Hot Girl Summer'),
      ('Quavo', "I'm the One"),
      ('Rae Sremmurd', 'Blasé'),
      ('Swae Lee', "Don't Judge Me"),
      ('The-Dream', 'Love U Better'),
      ('Ty Dolla Sign', 'Hot Girl Summer')]
```

- e) Write a query which returns the song name, album name and song year for all songs by Ariana Grande, ordered by year, album name and song name.

```
[ ]: e = [a for a in cur.execute("""
                                SELECT song.name as sn, album.name as an, song.year
                                ↪as sy
                                FROM song
                                INNER JOIN album ON album.artistID = song.artistID
                                INNER JOIN artist ON song.artistID == artist.
                                ↪artistID
                                INNER JOIN songOnAlbum ON songOnAlbum.songID ==
                                ↪song.songID
                                WHERE artist.name = 'Ariana Grande'
                                ORDER BY sy, an, sn
                                """)]
e
```

```
[ ]: [('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) Write a query which, for all songs where Ty Dolla Sign is either the main artist or is featured, returns the name of the main artist and the song name, ordered by the artist name and song name

```
[ ]: f = [a for a in cur.execute("""
                                SELECT sa.name as an, song.name as sn
                                FROM song
                                JOIN artist sa ON song.artistID = sa.artistID
                                LEFT JOIN featuredOn ON song.songID = featuredOn.
                                ↪songID
                                LEFT JOIN artist fa ON fa.artistID = featuredOn.
                                ↪artistID
                                WHERE (sa.name = "Ty Dolla Sign") OR (fa.name = "Ty
                                ↪Dolla Sign")
                                ORDER BY an, sn
                                """)]
f
```

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

- g) Write a query which returns the artist name and song name for all song names that include the sequence of characters “the”.

```
[ ]: g = [a for a in cur.execute("""
                                SELECT artist.name as an, song.name as sn
                                FROM song
                                JOIN artist ON song.artistID = artist.artistID
                                WHERE song.name LIKE '%the%'
                                """)]
g
```

```
[ ]: [('DJ Khaled', "I'm the One")]
```

- h) Write a query which returns the artist name and number of features for the artist in the database with the highest number of features (songs with said artist as the main artist do not count).

(HINT: a HAVING clause might be beneficial to use here.)

```
[ ]: g = [a for a in cur.execute("""
                                SELECT artist.name AS an, COUNT(*) AS num_features
                                FROM featuredOn
                                JOIN artist ON featuredOn.artistID = artist.artistID
                                GROUP BY artist.name
                                ORDER BY num_features DESC
                                LIMIT 1
                                """)]
g
```

```
[ ]: [('Future', 3)]
```

```
[ ]:
```