</talentlabs>

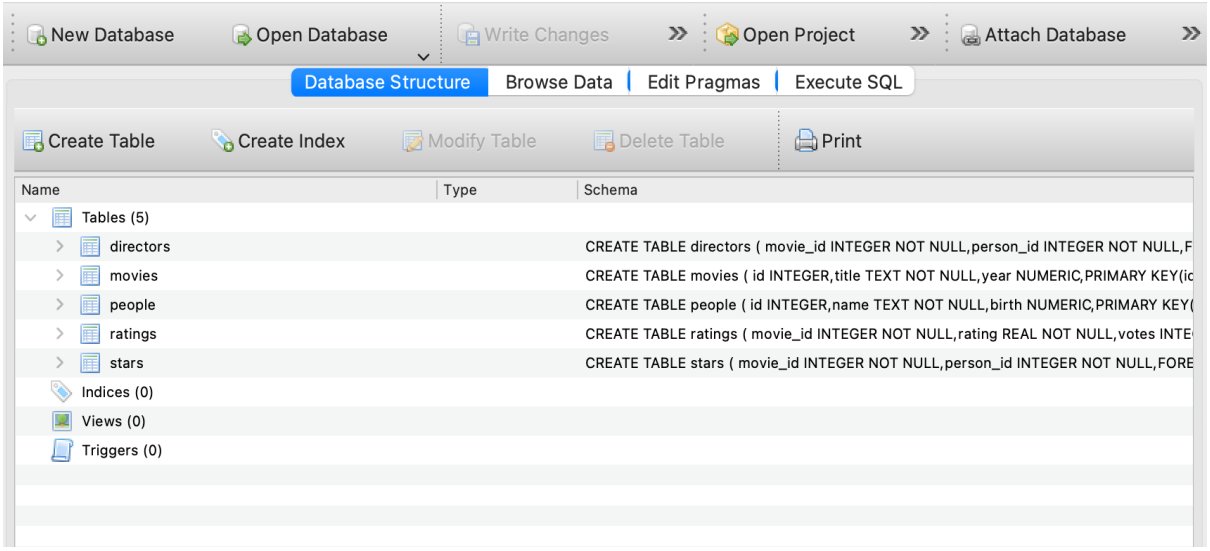
Lab Assignment 6 - Complex SELECT statements

Instructions

1. Answer the below question in the boxes.
2. Please submit the assignment through TalentLabs Learning System.

Open the Movies database

Follow the step illustrated in Chapter 3 to open the Movies database using DB Browser for SQLite. You should see 5 tables in the database.



</talentlabs>

Understanding the database

1. Study the table schema and the data in the “people” and “directors” table and describe the relation between the tables “people” and “directors”

People table consists of:

i. id
ii. name
iii. birth

Directors table consists of:

i. movie_id
ii. person_id

‘id’ column in People table corresponds with ‘person_id’ in Directors table

2. Study the table schema and the data in the “movies” and “directors” table and describe the relation between the tables “movies” and “directors”

Movies table consists of:

i. id
ii. title
iii. year

Directors table consists of:

i. movie_id
ii. person_id

‘id’ column in Movies table corresponds with ‘movie_id’ in Directors table

Query Exercises

1. Write a SQL query to obtain the movie_id who is directed by “Joris Ivens” without using WITH keyword
   Expected Output: a table with a single column for the movie_id of the director’s movie.

```
SELECT movie_id
FROM directors
WHERE person_id IN (
 SELECT id
 FROM people
 WHERE name LIKE 'Joris Ivens'
)
```

2. Write a SQL query to obtain the movie title who is directed by “Joris Ivens”
   Expected Output: a table with a single column for the movie title of the director’s movie.

```
SELECT title
FROM movies
WHERE id IN (
      SELECT movie_id
      FROM directors
      WHERE person_id IN (
         SELECT id
         FROM people
         WHERE name LIKE 'Joris Ivens')
)
```

3. Organize and rewrite the SQL query of Q1 using WITH keyword
   Expected Output: The SQL query in WITH keyword

```
WITH joris_id AS (
   SELECT id
   FROM people
   WHERE name LIKE 'Joris Ivens'
)

SELECT movie_id
FROM directors
WHERE person_id IN (
    SELECT id
    FROM joris_id
)
```

4. Write a SQL query to show each person’s name and whether the person is born before 1970, born in 1970, born after 1970
   Expected Output: The SQL query fulfilling the required data

```
   SELECT name,
     CASE
     WHEN birth < 1970 THEN 'born before 1970'
     WHEN birth = 1970 THEN 'born in 1970'
     WHEN birth > 1970 THEN 'born after 1970'
     END AS birth_period
   FROM people
```

5. Write a SQL query to count the number of people in the “people” table by each birth year.
   Expected Output: The SQL query fulfilling the required data. Note that having the NULL birth year on the query result is normal.

```
SELECT birth AS year,
COUNT(birth) AS number_of_people
FROM people
GROUP BY birth
```

6. Write a SQL query to count the number of directors by each birth year. Only the years with more than 500 directors born are interested.
   Expected Output: a table with two columns for the birth year and count of directors.

```
SELECT birth AS birth_year, COUNT(id) AS
count_of_directors
FROM people
WHERE id IN (
SELECT person_id FROM directors)
GROUP BY birth
HAVING COUNT(id) > 500
```

- End of Assignment -