Lab 3.6 - Complex SELECT statements

## Instructions

1. Answer the below question in the boxes.
2. Please submit the assignment after you finish.

## Open the Movies database

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



## 

## 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”

|  |
| --- |
| Director table contain person\_id  Which is is linked to the id in people table to identify who is director in people table and obtain the director’s name |

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

|  |
| --- |
| Directors table contain movie\_id  Which is linked to the id in movie table to identify who dircect the movie |

## 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 = "Joris Ivens"  ) |
|  |

1. 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.

|  |
| --- |
| WITH  joris\_id AS  (  SELECT id FROM people WHERE name = "Joris Ivens"  ),  joris\_movie\_id AS  (  SELECT movie\_id FROM directors WHERE person\_id IN (SELECT id FROM joris\_id)  )  SELECT title FROM movies WHERE id IN (SELECT movie\_id FROM joris\_movie\_id) |

1. 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 = "Joris Ivens"  )  SELECT movie\_id FROM directors WHERE person\_id IN (SELECT id FROM joris\_id) |

1. 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'  ELSE MIssing  END AS birth\_period  FROM people |

1. 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, count(birth) AS number\_of\_people FROM people GROUP BY birth |

1. 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(birth) AS count\_of\_directors FROM people WHERE id IN (SELECT person\_id FROM directors)  GROUP BY birth HAVING COUNT(birth) > 500 |

**- End of Assignment -**