Project 2:

CMPE 321, Introduction to Database Systems, Spring 2023

Due: 14.04.2023, 23:59

1 Project Description

The Oscars, also known as the Academy Awards, is an annual ceremony that honors outstanding achievements in the film industry. The awards were first presented in 1929, making it one of the oldest and most prestigious award ceremonies in the world.

The Oscars are presented by the Academy of Motion Picture Arts and Sciences, which was founded in 1927 to advance the arts and sciences of motion pictures. The Academy is composed of over 9,000 members, including actors, directors, producers, and other film industry professionals. Over the years, the Oscars have become an important cultural event, with millions of people tuning in to watch the ceremony each year. The awards are presented in various categories, including Best Picture, Best Director, Best Actor, Best Actress, and Best Screenplay.

Throughout its history, the Oscars have recognized many of the greatest films and performances of all time, and have helped to launch the careers of numerous actors and filmmakers. The ceremony has also been the subject of much controversy and criticism over the years, particularly in regard to issues of diversity and representation in Hollywood. Despite these controversies, the Oscars remain a significant event in the world of film and continue to honor the best and brightest in the industry each year.

Since it is the top event in the industry, many companies want to analyse the results and make predictions for upcoming years. To this end, our company has collected some data and created a database, but the company needs someone to

manage these requirements by automatically retrieving the necessary information from the database. Now it's your turn! Please write queries to fulfill the requirements!

2 Data

The database is provided as an SQLite database file which is named as Film_Selection.db. The database consists of 4 tables: Film, Director, Award, and Genre. Detailed information about the tables is provided below.

- **Film**
  - Film_ID INTEGER,
  - Title TEXT,
  - Director INTEGER,
  - Genre INTEGER,
  - Release_Year TEXT,
  - Budget INTEGER,
  - PRIMARY KEY(Film_ID),
  - FOREIGN KEY(Genre) REFERENCES Genre(Genre_ID),
  - FOREIGN KEY(Director) REFERENCES Director(Director_ID)
- **Director**
  - Director_ID INTEGER,
  - Director_Name TEXT,
  - Favorite_Genre INTEGER,
  - PRIMARY_KEY(Director_ID),
  - FOREIGN_KEY(Favorite_Genre) REFERENCES Genre(Genre_ID)
- **Award**
  - Award_ID INTEGER,
  - Award_Title TEXT,
  - Awarded_Film INTEGER,
  - PRIMARY KEY(Award_ID),
  - FOREIGN_KEY(Awarded_Film) REFERENCES Film(Film_ID)
- **Genre**
  - Genre_ID INTEGER,
  - Type TEXT,

– PRIMARY KEY(Genre_ID)

3 Queries

1. Find the number of films and display as *Film_Count*.
2. List Title, Director, *Release_Year* of the films which are filmed before 2020 (exclusive). Display *Release_Year* in YYYY format. Sort the results by *Release_Year* in ascending order.
3. List all the fields of the films with the minimum *Budget*.
4. List *Director_Name* and *GenreType* of the films with the maximum *Budget*.
5. Find the total price of filming all films based on *Budget*.
6. List *Type* of Genre together with the number of films (display as *Film_Count*) in each *Type*. Sort the results by *Film_Count* in descending order.
7. Insert award whose *Title* is "BU-Best Actor", *Awarded_Film* is "After Sun".
8. Find the directors who have directed films that have won awards in at least three different categories (e.g. "Best Picture", "Best Director", "Best Screenplay").
9. Find the films that were released in the same year as the film "The Godfather", and have a higher budget than it.
10. List all the films directed by directors whose favorite genre is Comedy, and were released between 2000 and 2010.
11. List all the films that were not directed by Martin Scorsese and did not win any awards.
12. Find the director who has directed the film with the highest budget in each year.
13. For each genre, find the director who has won the highest number of awards for their films in that genre.
14. List all the films that were released after 2015, and have not won any awards yet.
15. Find the director who has directed the highest number of films in the database.
16. Find the average budget of films for each director, and list the directors in descending order of their average budgets.
17. For each genre, find the director who has directed the highest number of films in that genre, and has never won any awards for their films in that genre.
18. Find the names of directors who have directed at least one film from each genre type.
19. Return all the fields of all films and all the fields of directors with matching *Favourite_Genre*.

20. List *Director_ID* and *Awarded* which displays TRUE if there is at least one award for the director's film, and displays FALSE otherwise.

    The output of each query is given in the ***output_query-index.txt*** files (e.g. *output1.txt, output2.txt, ... , output20.txt.*

4 Submission

- This project can be implemented either individually or as a team of two people. You are free to change teams in upcoming projects.
- Write each query to a file named **query_index.sql** (e.g. q1.sql,...,q20.sql).
- Include a comment in each file that explains the reasoning that led you to the query in detail.
- The .sql files that do not contain explanatory comments will receive 0 points.
- Put all .sql files into a folder.
- Name the folder as **StudentID1_StudentID2** if you are working as a group.
- Name the folder as **StudentID1** if you are working as an individual.
- Zip the folder for submission and name the .zip file with the same name given to the folder.
- Submit .zip file through **Moodle** until the deadline.
- Each group should submit **one** .zip file.
- Do not include any other files in your .zip file.
- **Any other submission method is not allowed.**
- 10 points will be deducted in case of non-compliance to any of the naming and folder conventions explained above.
- Do not inject your observations into your queries to skip some essential steps! Do not obtain values manually from the database and inject them into the queries! This kind of query will receive 0 points. There will be additional sanctions too since violating this condition will be treated as dishonest behavior.

5 Late Submission Policy

You are allowed a total of 7 late days on the projects with no late penalties applied. You can use these 7 days as you wish. For example, you can submit the first project on time, the second project 3 days late, and then the third project 4 days late. In that case, you will have to submit the fourth project on time. No late submissions for any of the projects will be accepted after you use these 5 extra days. If you change your team, the team's late days used so far will be the maximum number of late days used by any of the team members.