Assignment - 3 : CS3563 : Introduction to DBMS - II

Harsh Agarwal	Vishwak Srinivasan	Sahil Yerawar
cs15btech11018	cs15btech11043	cs15btech11044

1 Introduction

The objective of this assignment is to run some non-trivial queries on the tables designed, built and populated in the last 2 assignments. This report will summarize the results obtained from queries for which there was data available. The queries are available in the file `2_Assgn3.sql`.

There have been some modifications to the existing table structures and assumptions have been made while designing the queries. This report will also summarize these modifications / assumptions to the reader.

2 Modifications and Assumptions

2.1 Modifications

Not many modifications have been made from the tables constructed in the previous assignment. Considering the size of the `ROLE` table which is about 21.5 million rows, we have decided to add a new column `IsMovie`. We found this helpful for some of our queries, and the presence of this new column indicates whether a given role of a particular person in a particular picture (Movie / TV Series) is a movie role or a TV series role.

In the previous assignment, we had procured awards dataset (specifically for Oscars) from Kaggle datasets. But the content of this dataset is presently unstructured and is not able to provide consistent information for various types of awards. So, after further searching in IMDB website, we have found out that for each award organization and year, we have the data about all of this awards encoded in the form of `JSON`. So we have scraped this `JSON` from IMDB's web source page and wrote a Python script to extract the award information relevant to our previously defined data schema. We now have 11,697 rows as opposed to the previous 1,734 rows.

2.2 Assumptions

We have made some assumptions for these queries, since we didn't have related data present during query design:

Query 1, 11, 12

The assumption made here is that the `ROLE` table also has the value of "Singer" to denote singers. We also don't have the Grammy awards dataset, hence we have assumed that there are rows in the table `AWARDS` where the column `AwardOrganization` is "Grammy".

Query 10, 20

We do not have data regarding either the role of assistant director or the duration of it. So we have assumed that there exists a new table called `DIRECTOR_EXPERIENCE` which consists of 4 fields:

- `PersonID` : ID of the director
- `MentorID` : ID of the director's mentor
- `StartYear` : Start Year of working under the mentor
- `EndYear` : End Year of working under the mentor

The creation of this table enables us to perform the above two queries easily.

Query 14

We do not have data from the IMDB database for episode-wise ratings. Instead we have data for TV Series ratings overall. The assumption made for this question is that the ratings data has episode-wise ratings. By the design decisions presented in the previous assignment, every TV Series is a row in the table `PICTURE` with a `NULL` value in the `ParentPicture` column and with a unique `PictureID`, whereas TV episodes are rows in the table `PICTURE` with non `NULL` values in the `ParentPicture` field referencing them back to the TV series they belong to.

Query 15

For testing of this query, we have used the film "The Red Orchestra" as we didn't have any crew information for "The Fault in our Stars".

Query 19

The assumption made here is that the `ROLE` table also has the value of "actress" to denote actresses in the column `role`.

3 Results obtained

Out of 20 queries, we are able to present the results for 13 of them. These queries are Query 2, 3, 4, 5, 6, 7, 8, 9, 13, 15, 16, 17 and 18. The results are available at this link. The number of rows for each of the results are tabulated below:

Query 2	20620	Query 3	1	Query 4	269	Query 5	28
Query 6	2	Query 7	3339	Query 8	1	Query 9	9
Query 13	110	Query 15	12554	Query 16	253000	Query 17	15506
Query 18	10						

For the queries for which results where not possible to be obtained, we verified syntax using SQLFiddle. We have the screenshots of the verification at this link.