# DST2 Final Project Documentation

Team Fake Lovers

1/23/2020

## Team introduction

Team member & contribution

- *Jeff Gui (Yifan)*

  Contribution: database design, data cleaning, query 1 design and coding, documentation (software design, input/output), white-box testing.

- *Alana Cheng (Yuchen)*

  Contribution: database design, query 2 design and coding, documentation (database design, input/output), black-box testing.

Team name: *Fake `Lovers`*

## Design of the software

The software is coded in `Python` with the sorcecode modified from Wanlu's sample code.
There are mainly two parts:

1. A connection between `Python` and the database management system (DBMS), in this case, the `PostgresSQL` that handles our localized, mini IMDB.
2. A graphical user interface (GUI) that guides and collects users' query input, which visualize the query result in a clear and readable way.

### Python - PostgresSQL connection

The Python - PostgreSQL connection is achieved by `psycopg2` package. Some basic methods of the package including DBMS login, query encapsulation and result fetch is implemented in the sofeware.

### Graphical user interface & available queries

The `pygame` package is used to frame a graphical user interface of the software.
After entering the password of PostgreSQL, a window will pop up. As shown in the figure 1, the window is divided into 3 panels from top to bottom: the operation panel, the instruction panel and the output panel. The **operation panel** contains buttons for the user to choose, input the query or specify it if there are multiple meanings (? choose options provided by the result of queries). The **instruction panel** guide the user to make querys step-by-step and indicates errors when the connection to the database fails or no record is found according to the query. The **output panel** visualise the query result. An extra **"restart"** button at the bottom of the window allows the user to make multiple queries in one run.

Two kinds of the query is available. More details about these queries will be mentioned in the *Input* and *Output* sections.

1. **Adjusted movie ratings according to the title.**

Movie rating is an effective information resources for movie recommandation, which is determined by a lot of factors. To dissect the contribution of cast's skills to the movie rating, we query the corresponding records of each actor, actress, composer, writer and director of the movie. We assume that the average rating of the cast's previous works represents his/her professional skill. Therefore, by knowing the release year of the queried movie, we are able to obtain all the previous works of a cast and their average ratings.

Putting the average ratings of all the casts together, we can compute an adjusted rating of the queried movie by taking the mean of the casts' ratings. The adjusted rating makes sense as it represents the overall quality of the group of casts of a certain movie. By comparing the adjusted rating with the original rating of a movie, we may obtain clues about what kinds of industrial or social factors can affect movie ratings. For example, if the adjusted rating is much higher than the original one, there may be a promoting failure, scandals about casts etc. that play the role.

- Although only movie is allowed to query, all types of work is accounted for average rating.
- Cast with no previous works or work with no rating record is ignored.

The software allows querying the adjusted movie ratings according to the input title. The user will be asked to specify or confirm the movie referred in the upcoming steps.

2. **Searching for titles according to cast name and time period.**

   Sometimes, the users would like to find out the movies or series performed by an actor/actress within certain period of time. This query can output all the work that the queried cast is involved within the queried time period. User must specify the cast name, start year and/or end year of the query.

   - If either the record of start year or end year in the database is missing, the sofeware will only consider the other time parameter.

## Design of the database

Four databases, title.basics, title.ratings, title.principals and name.basics from IMDB (https://www.imdb.com/interfaces/) are used for the project. Firstly, we discussed about the type of queries we were going to provide based on the datasets provided by IMBD. Next, we figured out the relationship of all datasets provided and selected essential databases according to our queries (Fig. 2). Datasets are cleaned by using JAVA in order to fit the required input data type of PostgreSQL.

## Input

To begin, press the button on top of the operation panel according to the query. Once entering either query, you can quit and restart the software whenever you like by pressing "restart" button at the bottom.

1. **Adjusted movie ratings according to title** (Fig. 3).
   - Press "movie title" button.
   - If pressed correctly, the button field will be highlighted as shown in the picture below; then type in the movie title (for example: Titanic).
   - Wait until movie records with the queried title are printed. Choose and select the movie title that you refer to. Press "Enter" to finish the input. The software will then compute the adjusted rating.
2. **Searching for titles according to cast name and time period** (Fig. 4). Input staff's full name, start year of the movie/series and/or end year of the movie/series.

## Output

1. **Adjusted movie ratings according to title** (Fig. 5).

The result of adjusted movie rating query is shown in a sheet-like format, with several attributes separated by "|".

- *Name*: name of the cast.
- *Average ratings before the queried movie*: averaged ratings of all works done by the cast previous to the release of the queried movie.
- *ID*: unique ID of the cast in the database.
- *Job category*: job of the cast in the queried movie.

Both the adjusted movie rating, denoted as *"Average previous ratings of all members involved in the queried movie"*, and the *"Original rating of the queried movie"* are shown at the bottom for further comparision and analysis.

2. **Searching for titles according to cast name and time period** (Fig. 6).

   The name of the movie/series with the staff involved in and be played between the input start year and end year.

## Figures

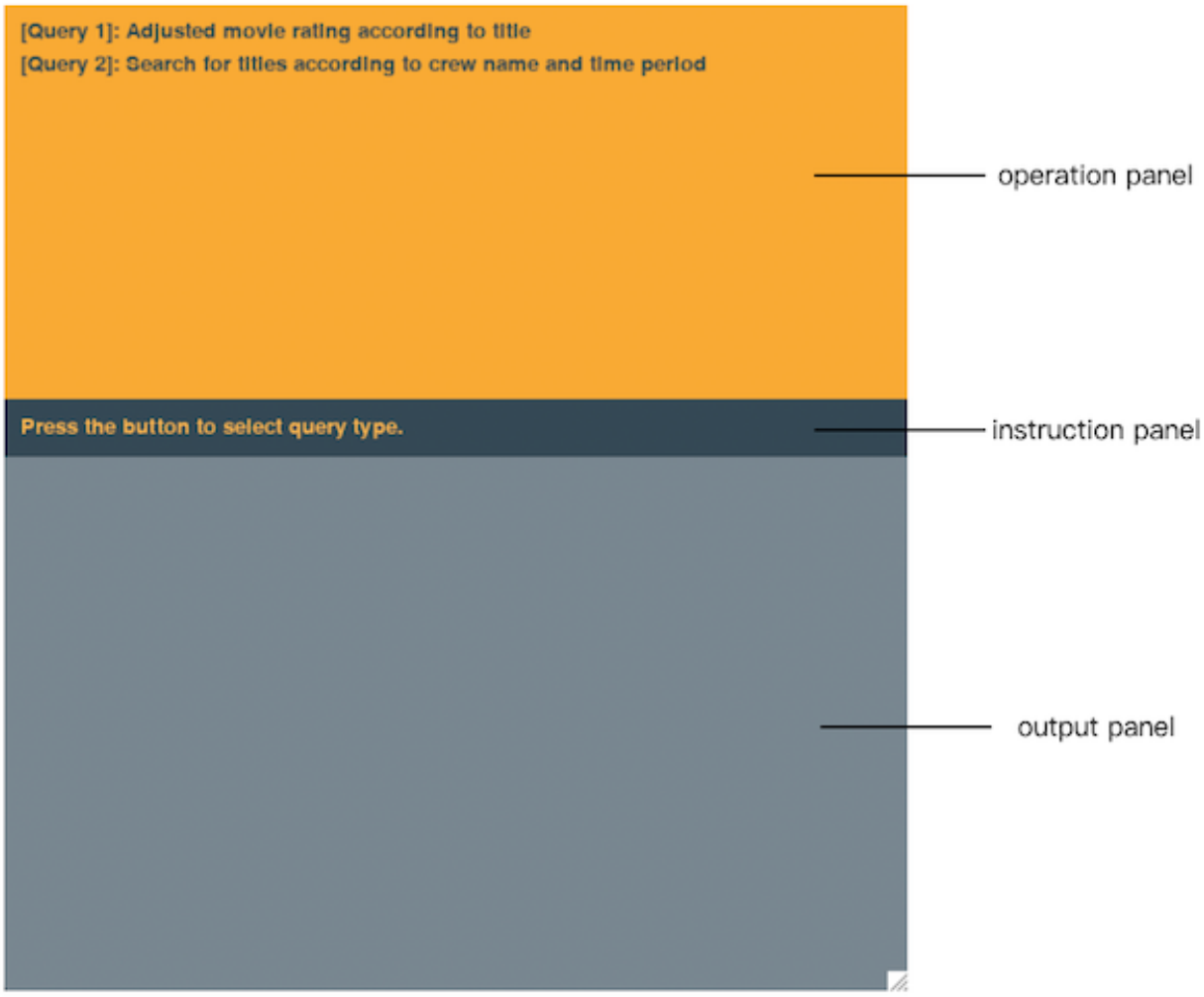


Figure 1: Graphical User Interface

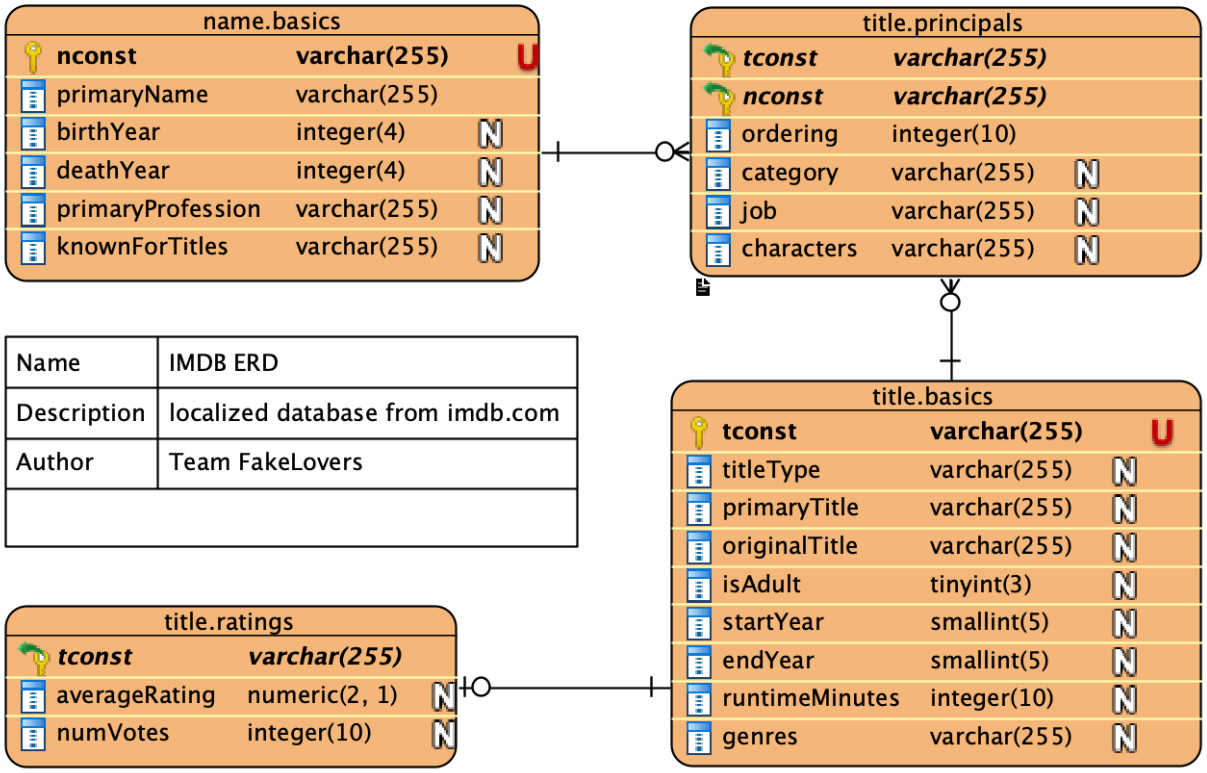


Figure 2: Database ER diagram

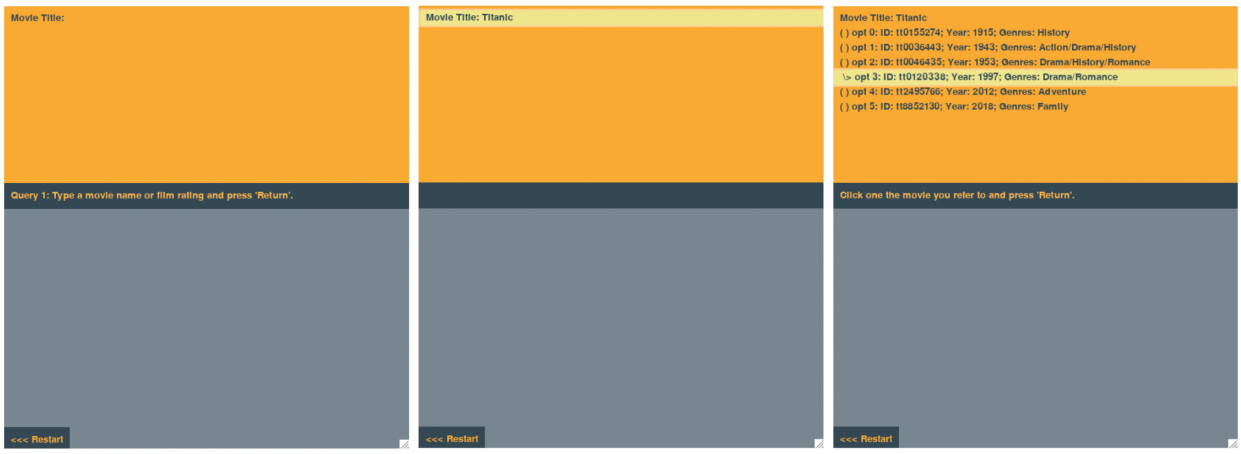


Figure 3: Query 1 input steps (Titanic as an example)

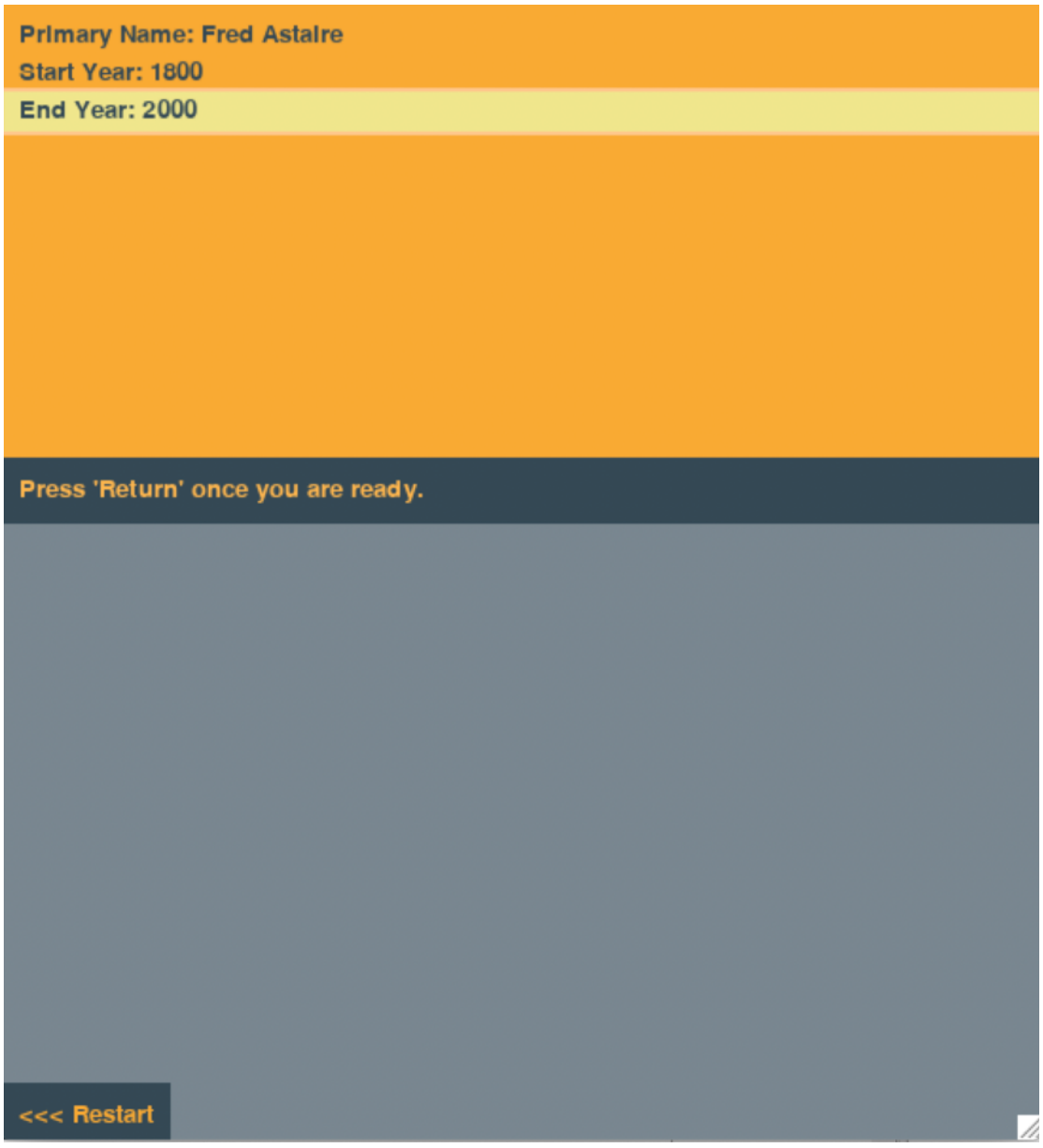


Figure 4: Query 2 input (Fred Astalre as an example)


Figure 5: Query 1 output (Titanic as an example)


Figure 6: Query 2 output (Fred Astalre as an example)