Katherine Zeng
Nicole Rosas
Connie Liu

Final Database Project Report

Overview: This final project consists of a database for Korean dramas and related entities such as actors, writers, network/tv stations, and awards (see UML diagram for all objects involved). We chose this idea for our project because we are fans of Korean dramas and have enjoyed many series in the past. We thought that it would be meaningful to create and use a database of kdramas to help users find shows that could possibly cater to their interests and it could even help us choose which drama to watch next.

I README.md

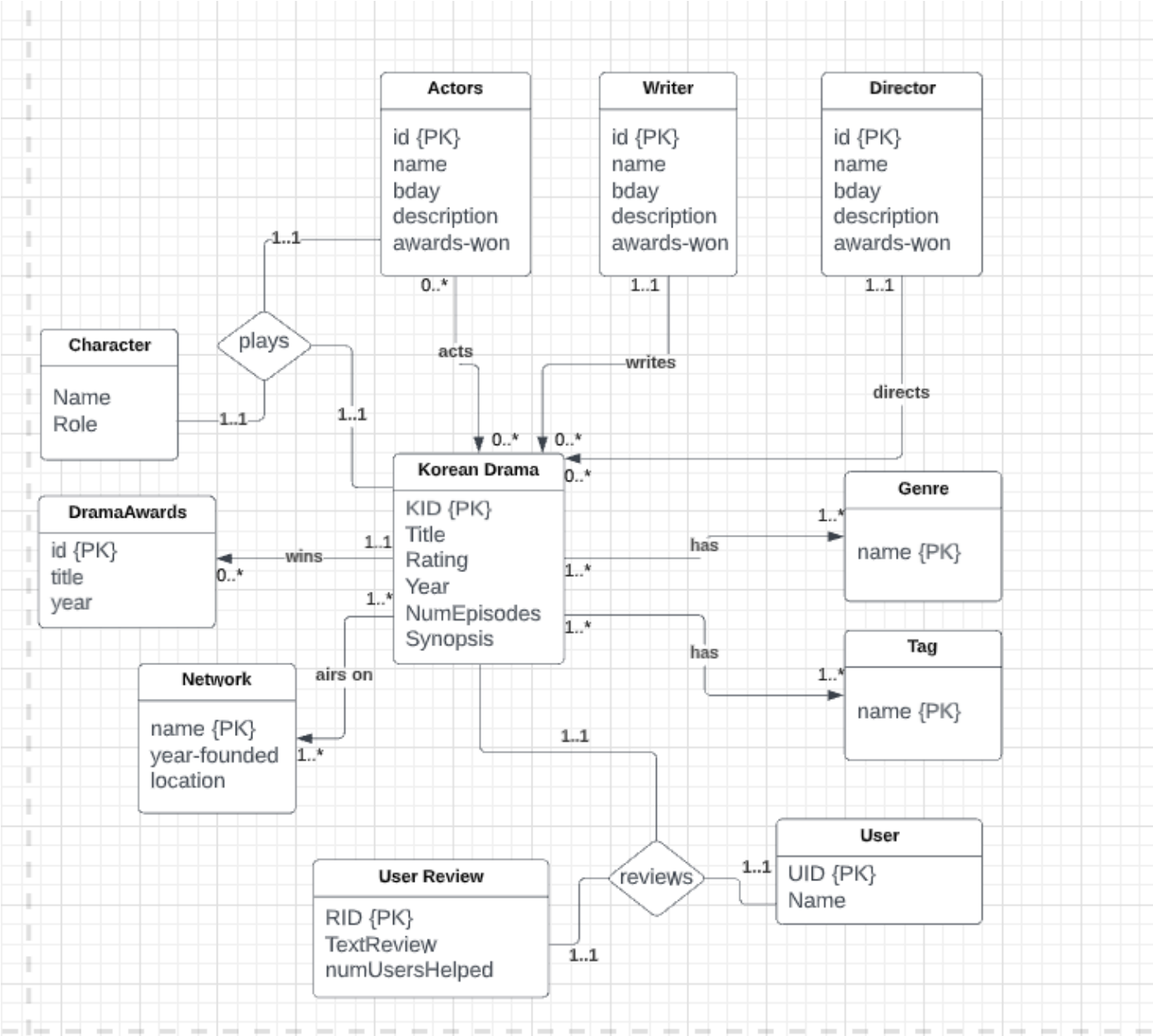
```
KDRAMA database
Katherine Zeng, Connie Liu, Nicole Rosas

Necessary Technology + Files
- MySQL
- dump sql file
- IDE that can run python
- main.py (python file)
- PyMySQL library
- csv files to fill out the database

To run:
1) First download and extract the provided zip file.
2) Set up the database using the dump file and if no data is
present use the given csv files to import the data. Some csv
files have unnecessary columns, just deselect them when
importing
3) Open up the python file (main.py) and begin running, make
sure that PyMySQL is imported before running.
4) Follow the command line prompts and enjoy :)
```

II Technical specifications

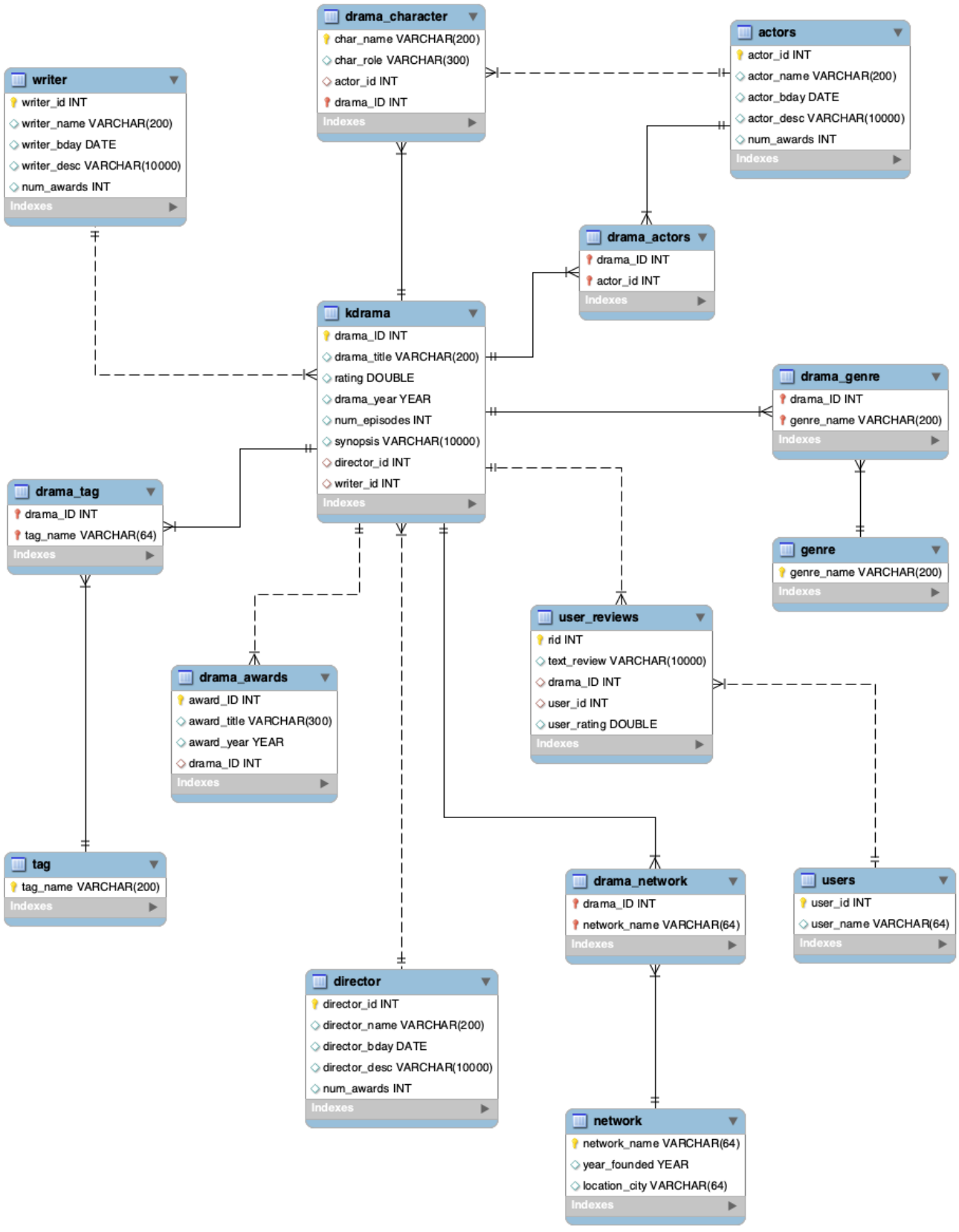
We used MySQL for the database to create the schema, tables, procedures, and functions. We use a python frontend application that is connected to our schema via the pymysql library. Our python application takes in input through the command line in the form of text, prompting users for information related to the operations it can perform.

III Conceptual UML (below image)

Image attached/pasted below. There are 10 separate objects, where “Character” is a weak entity due to depending on the drama for its existence.



IV Logical Design

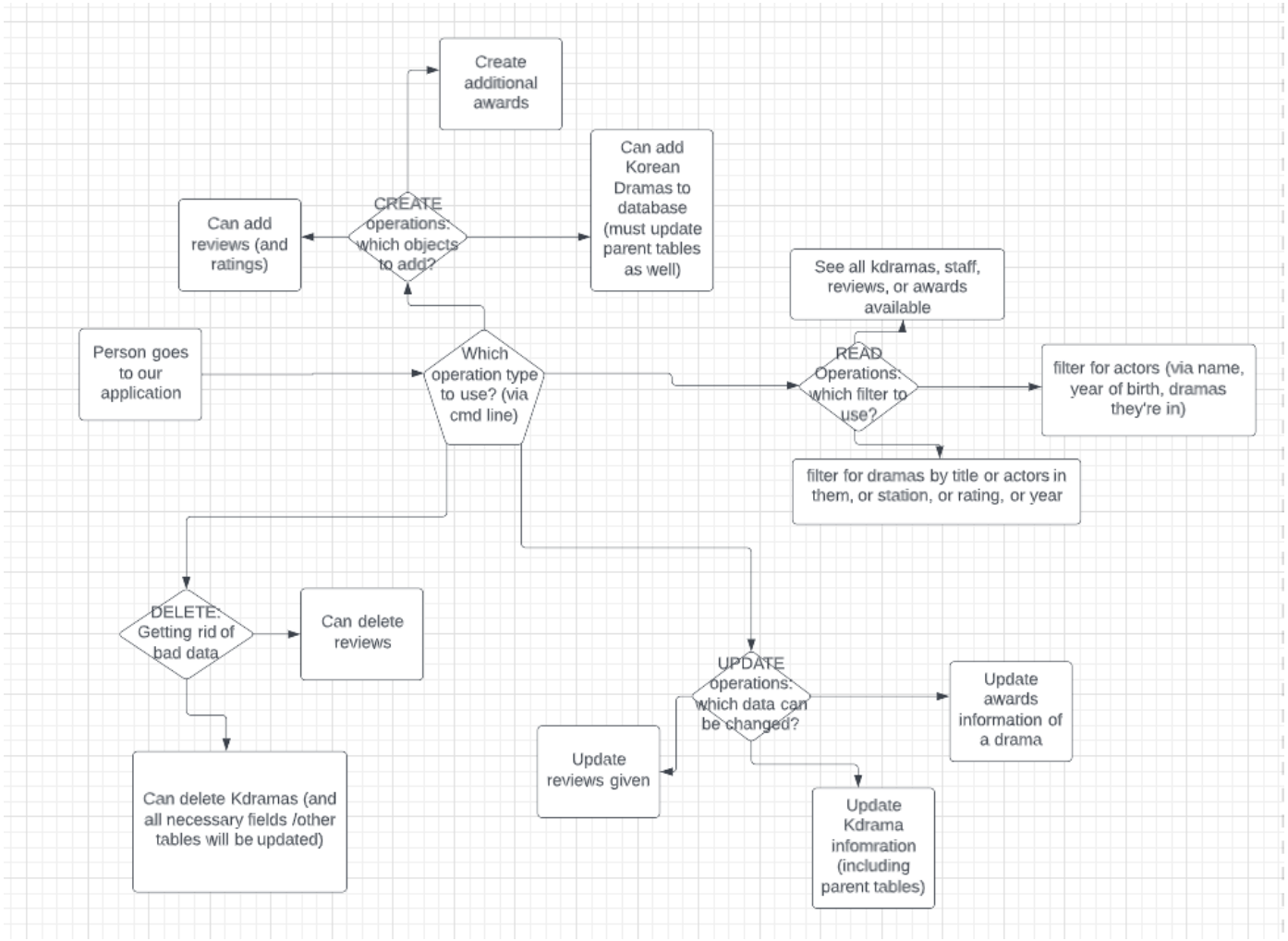


V User flow

The commands users can do to interact with the system include:

- CREATING
    - Korean Dramas (this will also create corresponding writer, director, and network if necessary)
    - Awards
    - Reviews
- READING (selecting)
    - Korean Dramas (we can find KDramas based on criteria such as rating, title, station, year featured, or actors playing in them)
    - Awards
    - Reviews
    - Actors (we can filter for actors based on name, year of birth, or kdramas they play in)
    - We can also select dramas or actors by various filters
        - The program will find the best match with the user's input. For example, if the user typed in "heaven" for the title, the program will return the drama "Move to Heaven"
- UPDATING
    - Korean Dramas
    - Reviews
    - Awards
- DELETING
    - Korean Dramas (all related tables are changed as well)
    - Reviews
- Quit program

A diagram is below:



VI Lessons learned

We learned a lot throughout this class and it was a good experience being able to use our knowledge for a final project. Some of our insights include:

- **Technically**: Throughout this project, we gained more experience in being able to import outside data into our database (as we had to take datasets from other sources in order to get existing kdramas, actors, writers, etc, into our data). We improved upon our skills of creating procedures and functions for our database schema, as we had to create many, many procedures in order to add tuples to our database, delete, etc. We also learned better logical organization/flow because our tables are so interconnected with each other: for example, in order to create a kdrama, we have to first make sure that all the parent tables have the necessary information, and then we also need to add to any child tables (especially for the tables representing many-to-many relationships).

- **Insights:** In general, we gained a lot of experience with time-management and efficient teamwork. Evidently because we are a group of three, we had to figure out how to divide the work in a way that made sense. We realized that there were essentially three main parts to our project: creating the database/getting the data, connecting the database to the python application with the procedures/functions for the db, and the python application itself. Because of this, we had each person take a role. We also made sure to spread the time out as much as we could so that we had efficient time management.

- **Alternative designs**: We actually ended up going with a slightly different design for our database than we started off with. This is because as we were importing data in, we realized that some of our objects did not match up well. For example, we once had both a “review” object and a “rating” object (where the rating object is essentially a bunch of different numbers rating things like music, plot, etc), but we decided to turn the rating object into an overall rating number and include it in our review object. We also originally had “stations” and “providers”, but realized that these two entities are extremely similar and could just be combined as “networks” -- basically different companies that you can watch the dramas from. Lastly, at first we had separate awards for writers/actors/directors, but we realized that this wasn’t as important to our database that focuses on the kdramas themselves, and has the actors, writers, and directors as just additional information to supplement the kdramas. So, we only list the number of awards that the actors, writers, and directors win, but we focus on the awards that dramas win -- we have an object for that.

- **Code issues**: None

VII **Future Work**

- **Planned Uses:** Going forward, we plan to use this database as a tool to help users find top rated dramas to watch that are catered to their interests. They could effectively filter out dramas by searching genres, tags, the name of the drama, actors, etc. The dramas we’ve chosen to input into this database are some of the more well-known dramas (or at the very least, dramas that we like!) so they would definitely serve as good recommendations for anyone looking to get a start into kdramas.

- **Potential areas for added functionality:** Another functionality that we could consider implementing in the future would be to find the duration of each episode and multiply that with the number of episodes for each drama. This way, the database could sort dramas based on their actual length and allow the user to

select shows based on watch-time preferences. Furthermore, we currently only have 20 dramas in our database for convenience, but if this were to be published, we would ideally need more drama tuples and their corresponding actors, directors, writers, etc in our tables.