Disclaimer (Data Scraping)

Name: Vamsi Gamidi
#ID: B00834696

In assignment 1 of CSCI 5408 course, data scraping is done manually or programmatically from Dalhousie University's website, and it is used only for educational purpose. Sensitive information, such as personal Email, personal contact numbers are not extracted. However, names of instructors, professors, or other staff members available on the Dalhousie University websites are extracted for course (CSCI 5408) related analysis, such as "find how many employees have similar first name etc." The scope of the extracted data usage is limited to the course CSCI 5408 only. The course instructor and the Faculty of Computer Science cannot be held responsible for any misuse of the extracted data.

Queries for questions:

1. Find the name of the department or faculty that has the highest number of employees having last name starting with an "A"

Ans:
```
SELECT Faculty, COUNT(Faculty) FROM mydb.faculty
WHERE Last_Name LIKE "A%"
GROUP BY Faculty
ORDER BY COUNT(Faculty) DESC;
```

2. Find the name of the department or faculty that has the highest number of undergraduate programs

Ans:
```
SELECT faculty, COUNT(program) FROM mydb.undergrad_programs
GROUP BY faculty
ORDER BY count(program) DESC
LIMIT 1;
```

Initial Data Design:

I have completed the initial data design for the based on the business requirements.

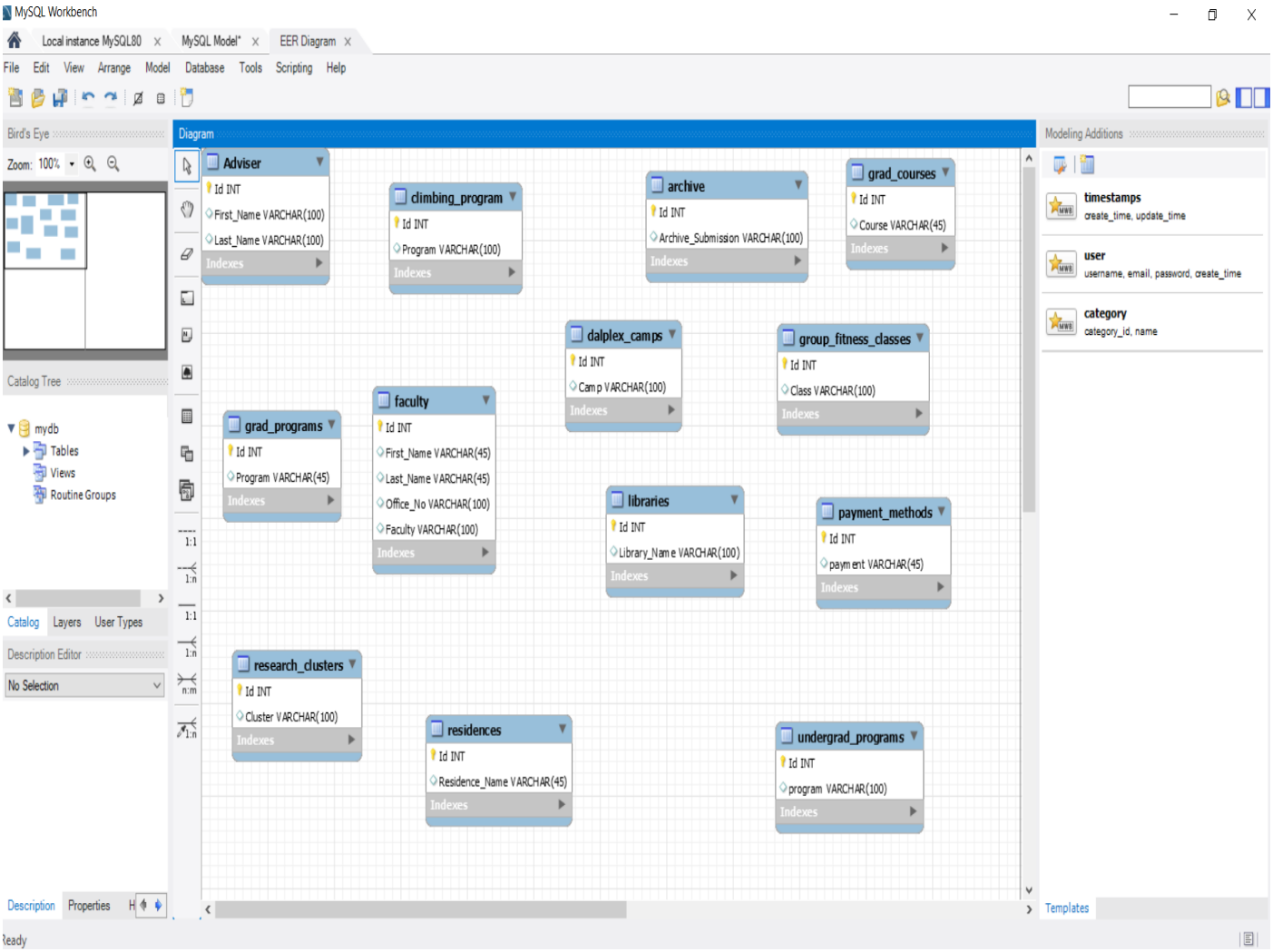

Figure:1 Initial Data Model without for Scraped Entities

Figure:1 shows the initial data model for the entities from https://www.dal.ca/

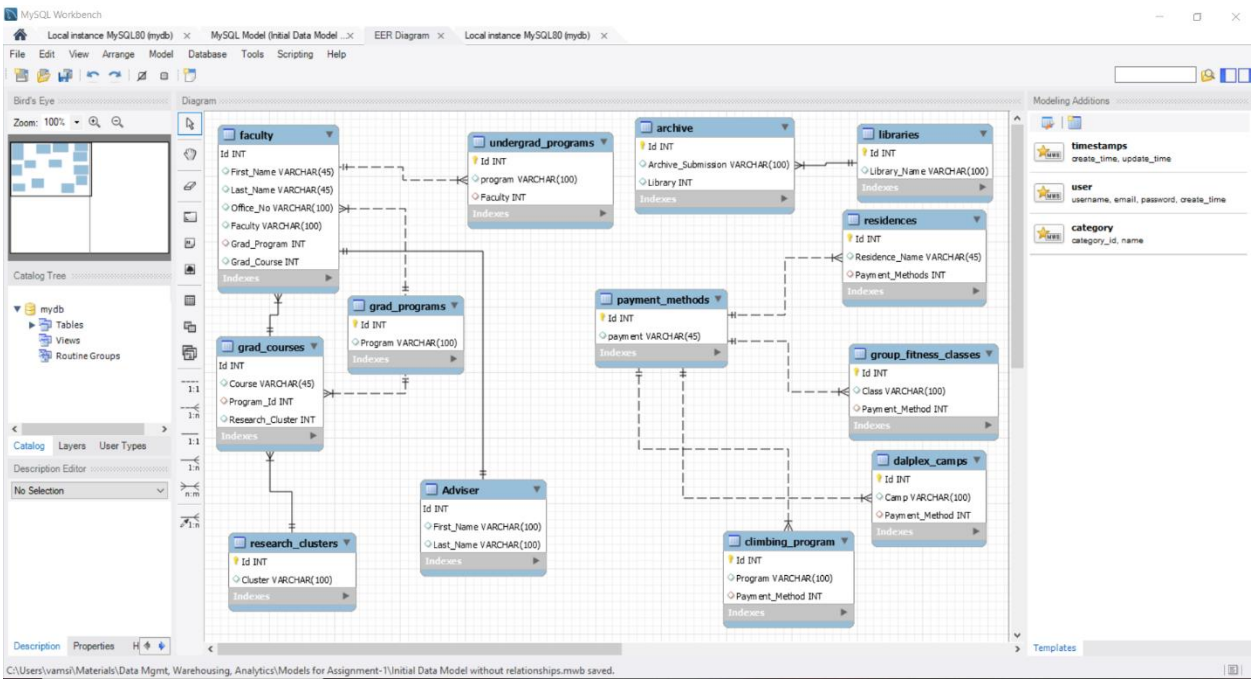

Figure:2 Initial Data Model with relationships among entities

The initial data model has multiple instances of fan trap problem.

Data Extraction and Data Collection:

I have extracted the data from https://www.dal.ca/ and stored required data into XML files using python script (Please refer '1.Web_Scraping.py'). I have also included a sample execution screenshot (Doesn't display any output as the script doesn't print anything but writes date into XML files, please see 'Generated XML Docs' folder).
I have cleaned the data from Google App Rating data (Please refer '2.Data_Cleaning.py').

Data Insertion:

I have imported the data from XML files into the respective tables. Please refer 'Screenshots for Data Insertion Commands and Inserted Data in Tables' folder to see the screenshots for execution of commands and inserted data.

Final Data Model:

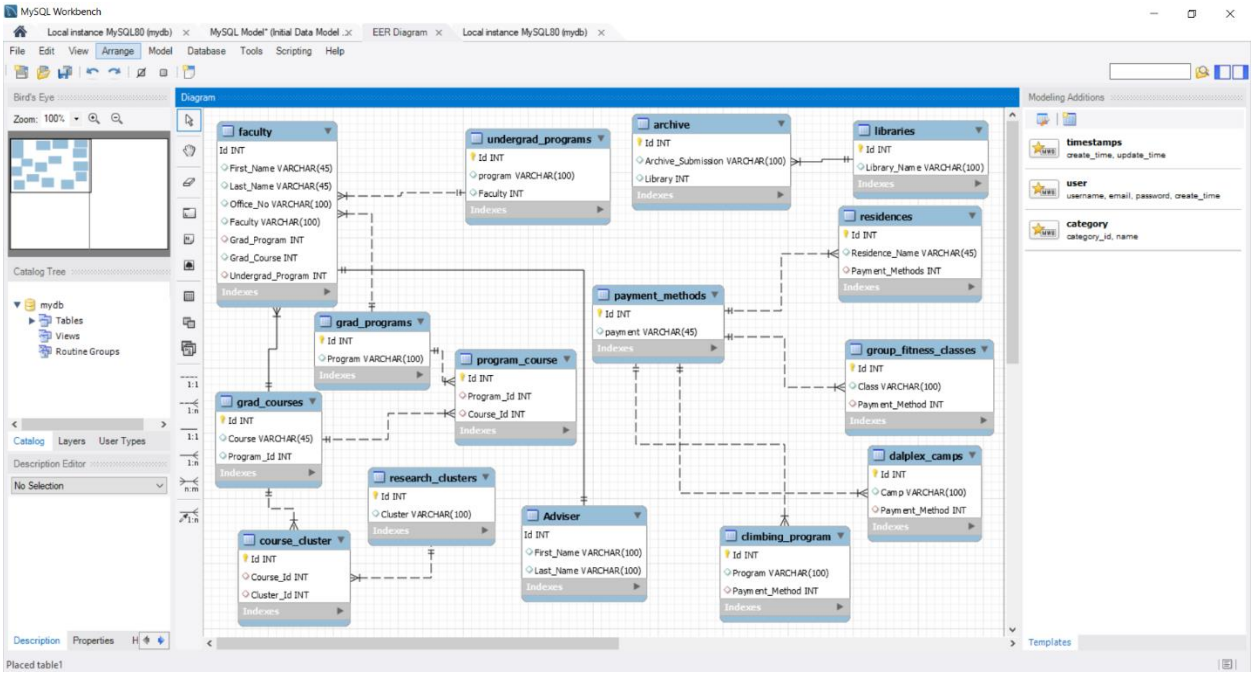

Figure:3 Final Data Model

The fan trap problems which we recognized in initial data model (see figure:2) are being resolved.
Fan trap problem 1: one to many relationships from research_clusters to grad_courses and grad_courses to faculty.

research_clusters → grad_courses → faculty

Solution:
Created a new entity course_cluster with Course_Id and Cluster_Id as foreign keys from grad_courses and research_clusters respectively.

Fan trap problem 2: one to many relationships from grad_programs to grad_courses and grad_courses to faculty

grad_programs → grad_courses → faculty

Solution:
Created a new entity program_course with Program_Id and Course_Id as foreign keys from grad_programs and grad_courses respectively.

Normalization up to 3NF (If possible):

The entities which were created initially are already in 3NF.

Every attribute in the entities is single valued. So, it is in 1NF.
There are no partial dependencies in the entities. So, it is in 2NF.
There are no transitive dependencies in the entities. So, it is in 3NF.

Google App Dataset from kaggle:

As mentioned above, I have cleaned the data from Google App Rating data (Please refer Data_Cleaning.py) and imported to the database. Please see the screenshots of inserted data in 'Screenshots for Data Insertion Commands and Inserted Data in Tables' folder.

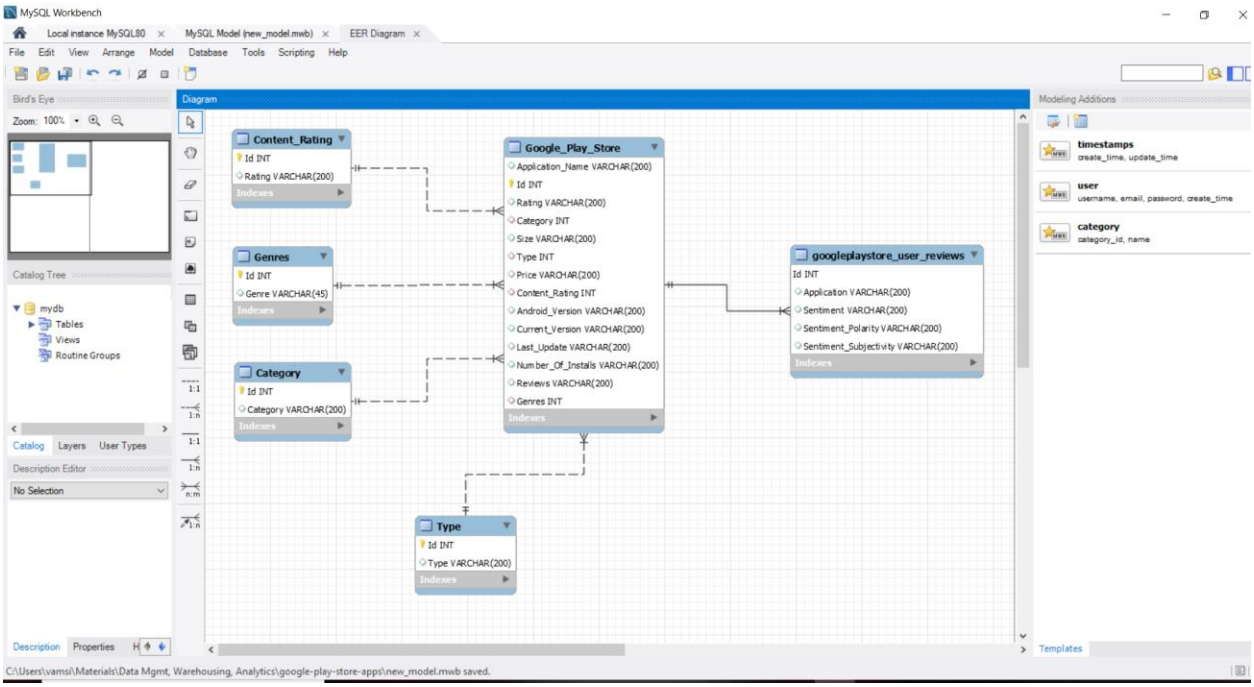

Figure:4 Data model with fan traps

The initial data model (see figure:4) has the following design issues (fan traps):

Fan trap problem 1: one to many relationships from Content_Rating to Google_Play_Store and Google_Play_Store to googleplaystore_user_reviews

Content_Rating → Google_Play_Store → googleplaystore_user_reviews

Solution:
Created a new table App_Content with App_Id and Content_Id as foreign keys from Google_Play_Store and Content_Rating respectively.

Fan trap problem 2: one to many relationships from Genres to Google_Play_Store and Google_Play_Store to googleplaystore_user_reviews

Genres → Google_Play_Store → googleplaystore_user_reviews

Solution:
Created a new table App_Genre with App_Id and Genre_Id as foreign keys from Google_Play_Store and Genres respectively.

Fan trap problem 3: one to many relationships from Category to Google_Play_Store and Google_Play_Store to googleplaystore_user_reviews

Category → Google_Play_Store → googleplaystore_user_reviews
Solution:
Created a new table App_Category with App_Id and Category_Id as foreign keys from Google_Play_Store and Category respectively.

Fan trap problem 4: one to many relationships from Types to Google_Play_Store and Google_Play_Store to googleplaystore_user_reviews

Types → Google_Play_Store → googleplaystore_user_reviews

Solution:
Created a new table App_Type with App_Id and Type_Id as foreign keys from Google_Play_Store and Type respectively.

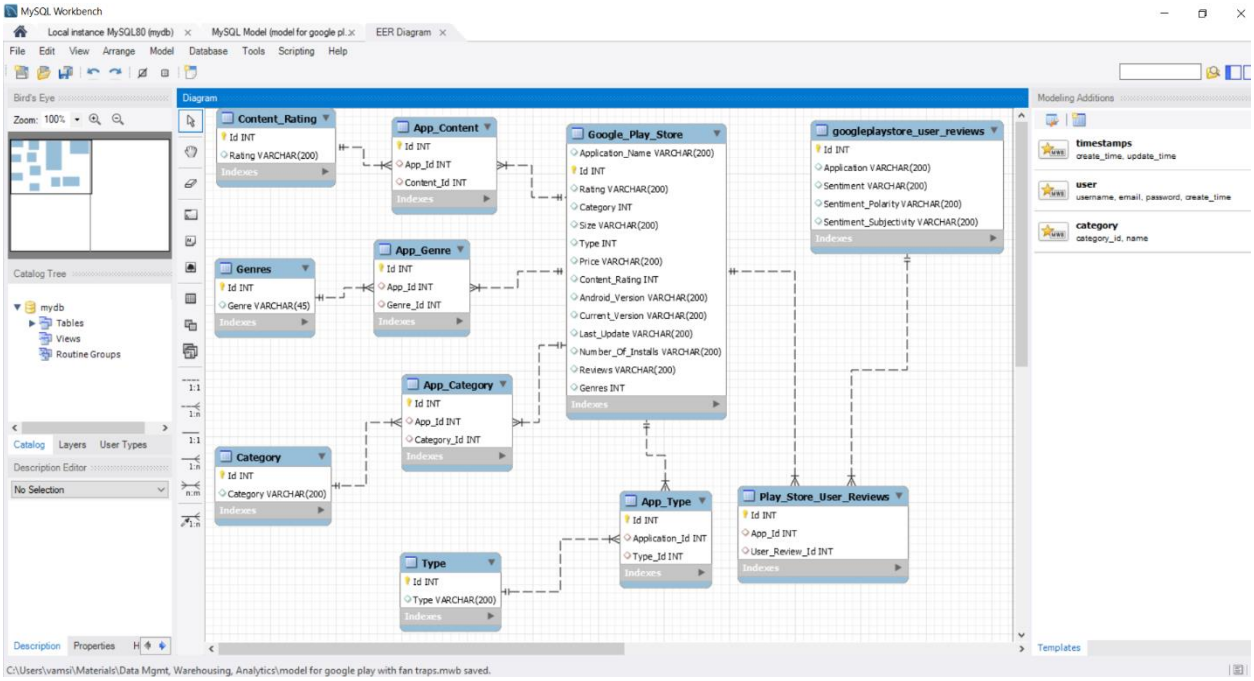

Figure:5 Final Data Model for Google_App Dataset Free from Design Issues

References:

Professional Internet Site:

[1] MySQL Documentation on 'LOAD XML Syntax' [Online]
Available: https://dev.mysql.com/doc/refman/5.5/en/load-xml.html
[Accessed on Sept. 20, 2019]

[2] Omkar S Hiremath's 'A Beginner's guide to learn web scraping with python!'
Last Updated on May 22, 2019 [Online]
Available: https://www.edureka.co/blog/web-scraping-with-python/
[Accessed on Sept. 17, 2019]