CSCI5408 Data Management, Warehousing and Analytics

Assignment 1

**Submitted By:**

Rahul Anand

B00841310

rahul.anand@dal.ca

# Disclaimer

Name: Rahul Anand

#ID: B00841310

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.

# Requirement Gathering

* For the first component of the project, I downloaded the Google App rating dataset from Kaggle. The dataset contains the information about approximately 10,000 applications and a review set of over 70,000 records.
* For the second part of the project, the data set was chosen on the assumption that the application is to be designed for prospective students and professors who wish to join Dalhousie University. Thus, the only entities selected were the ones which will be useful to only the prospective students or professors.

Data Gathering and Cleaning

**App Review Dataset**

* For the first part, I decided to drop some attributes from the *Application* table. The dropped attributes were *‘Type’, ‘Price’, ‘Content’ and ‘Genres’*

**REASON :** The dropped attributes were useless according to me, as they did not contribute a factor in determining success of the application. Most of the data under this column was either Null or garbage value.

* Further, I removed the duplicate rows in the table (the rows which had the same application names) using a combination of small python scripts and MS Excel.
* After that, I generated a unique ID for each application using python, and generated a foreign key in the *Reviews* file, again using python scripts and MS excel
* A separate table *category* was also generated, just so to maintain clarity in the database.

The following table shows the final entities and attributes for application reviews dataset.

|  |  |  |
| --- | --- | --- |
| **Entity** | **Initial Attributes** | **Final Attributes** |
| Application | App  Rating  Reviews  Size  Installs  Type  Price  Content  Genres  Last Updates  Current Ver  Android Ver | App ID  App  Rating  Reviews  Size  Installs  Last Updated  Current Ver  Android Ver |
| Category | None | Category ID  Category Name |
| Review | App  Translated Review  Sentiment  Sentiment Polarity  Sentiment Subjectivity | Review ID  App  Translated Review  Sentiment  Sentiment Polarity  Sentiment Subjectivity |

**Dalhousie Website Dataset**

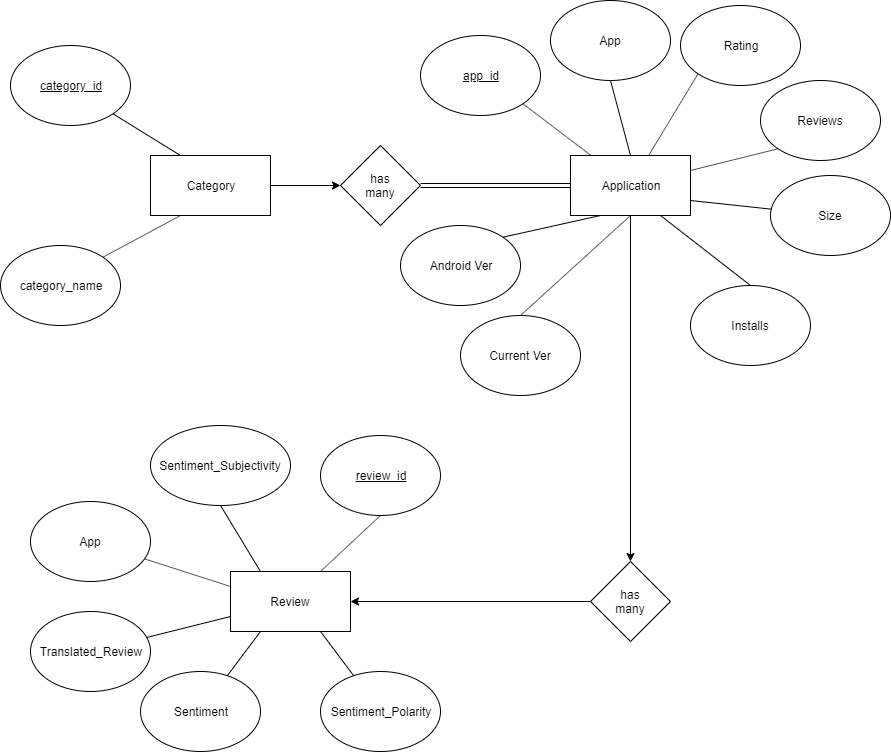
* For the second part, I decided to take 11 entities that I think will be useful for prospective students and faculties.
* I used a python script to scrape data for the entities from Dalhousie official website. The code can be found under *‘scrape.py’* and has 11 functions to scrape data (using BeauifulSoup or mocking API calls) and return XML strings for each of them
* Following is a table of the entities and the attributes I decided to scrape

|  |  |
| --- | --- |
| Entity | Attributes |
| Campus | Campus ID  Campus Name |
| Faculty | Faculty ID  Faculty Name |
| Department | Department ID  Department Name |
| Program | Program ID  Program Type  Program Name |
| Staff | Staff ID  Staff First Name  Staff Last Name |
| Research Offices | Office ID  Office Name |
| Internal Funding | ID  Type of Grant |
| External Funding | ID  Type of Grant |
| Awards And Prizes | ID  Type of Grant |
| Facility Type | Facility ID  Facility Name |
| Facility Sub Type | ID  Facility Sub-Type |

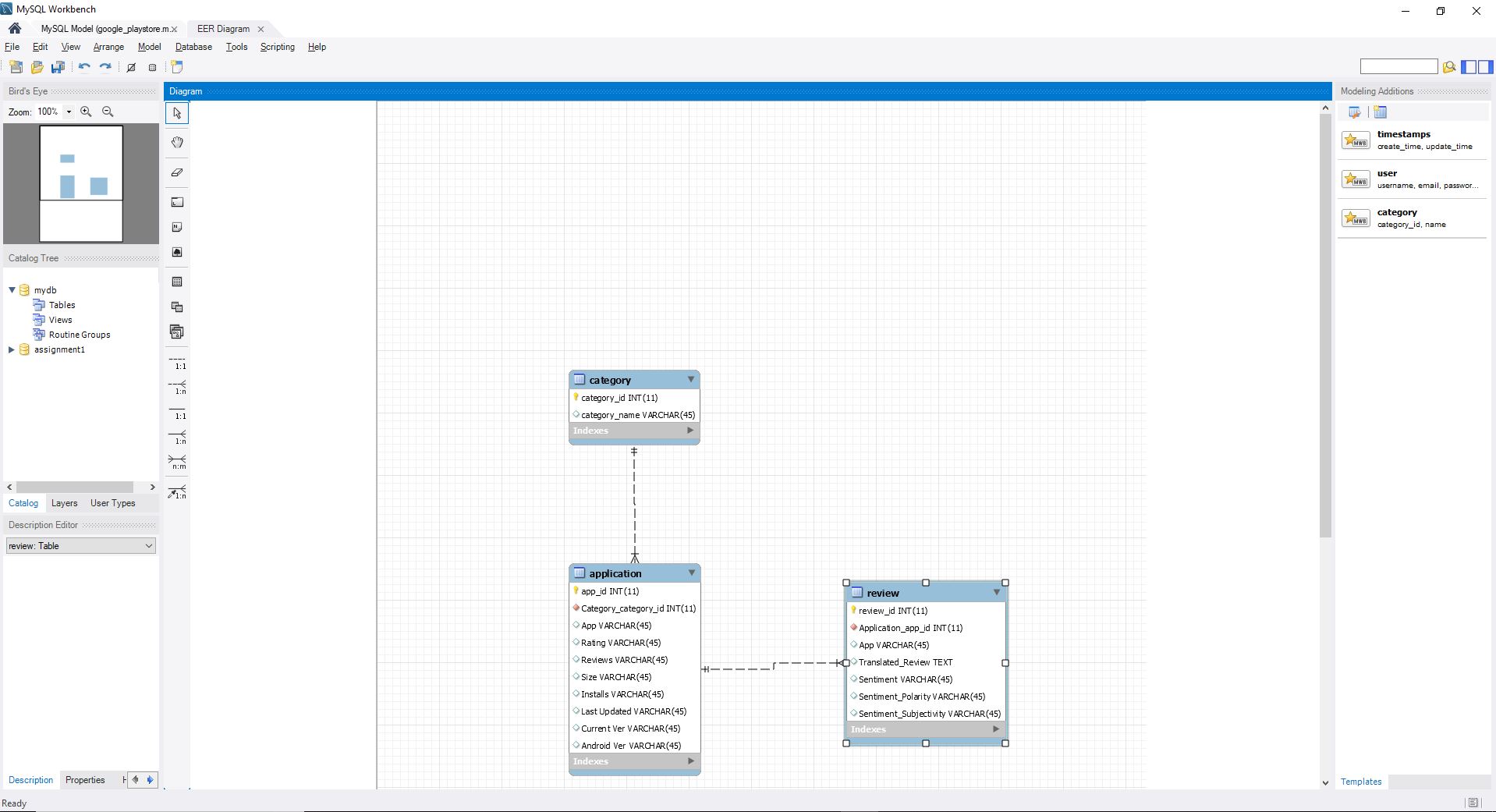
# Data Modelling

**App Review Modelling**

Below are the conceptual and final dataset for the Dalhousie application



**Figure 1: Conceptual ERD google app review**

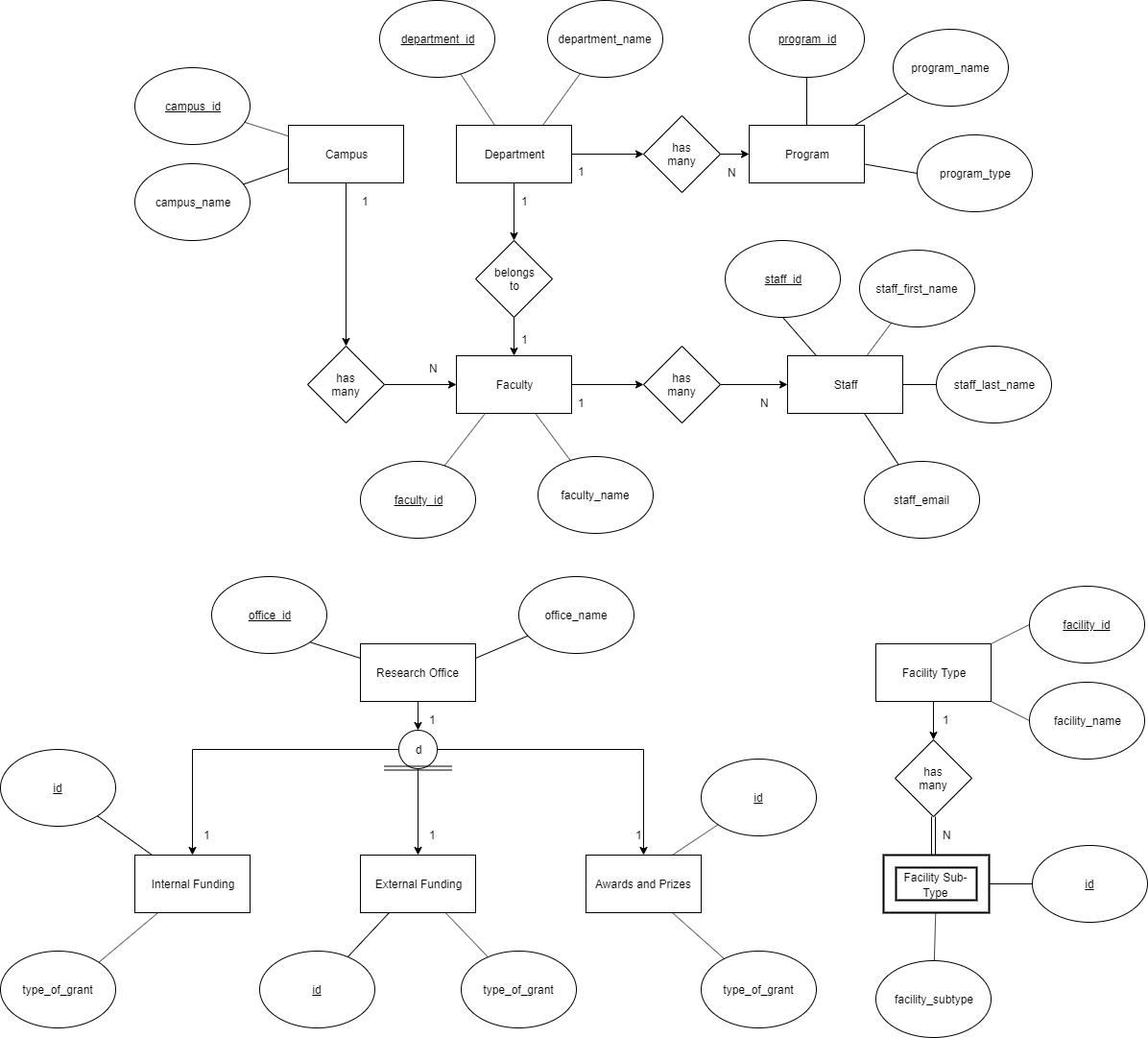


**Figure 2: Final ERD google app review**

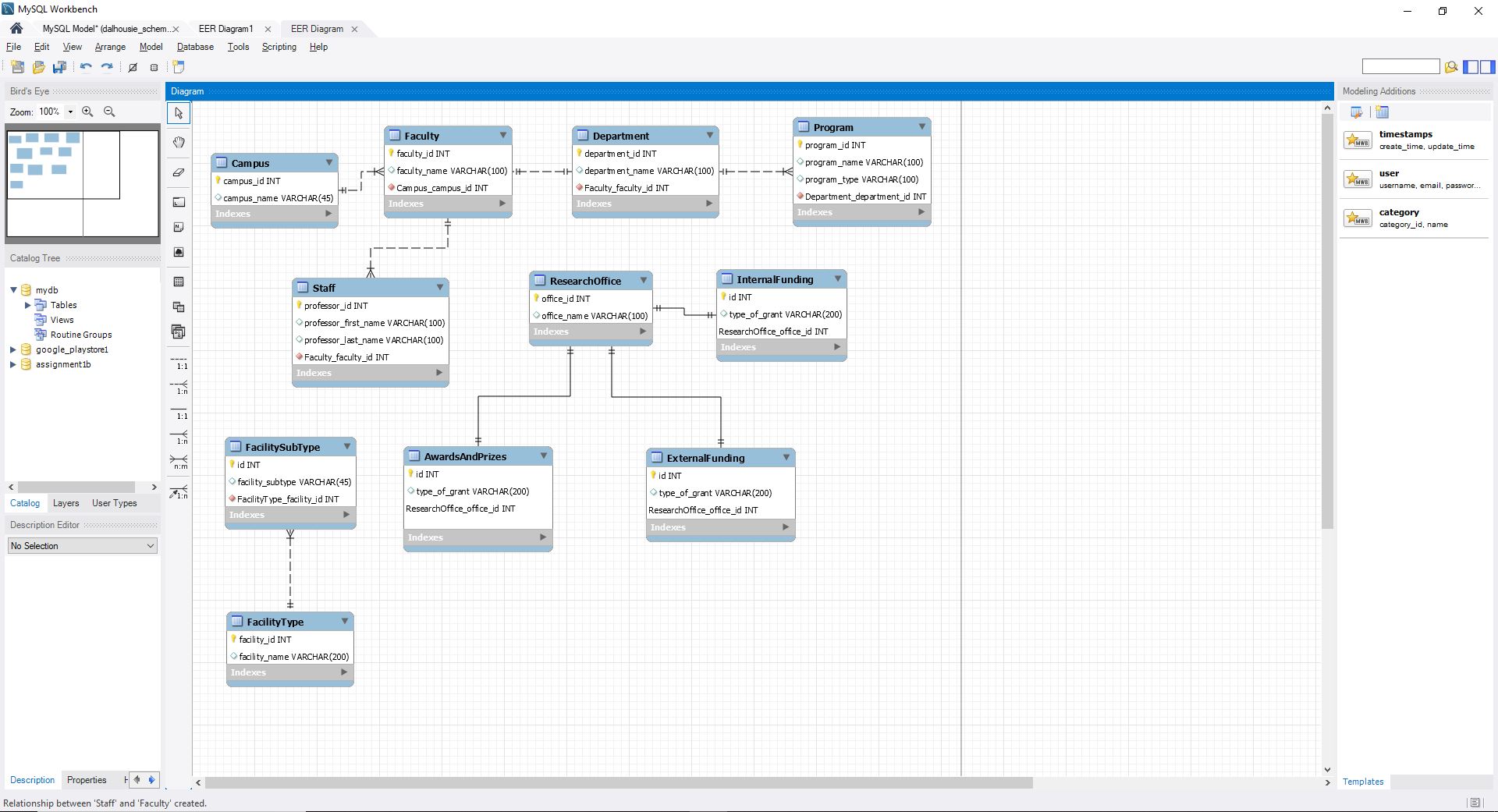
The initial ERD is free from any design issues like “Fan-Trap” or “Chasm” problems. The final ERD is drawn on MySQL workbench and has no partial or transitive dependencies. So this database does not require any further normalization and is in 3NF form

**Dalhousie Website Modelling**

Below are the conceptual and final dataset for the Dalhousie application



**Figure 3: Conceptual ERD Dalhousie**



**Figure 4: Final ERD Dalhousie**

The initial ERD is free from any design issues like “Fan-Trap” or “Chasm” problems. The final ERD is drawn on MySQL workbench and has no partial or transitive dependencies. So this database does not require any further normalization and is in 3NF form

**Note:**

* It was necessary to create separate entity for each type of funding as some public funding can exist even if a research office doesn’t
* Also, the facility subtype is dependent entity on facility type. An example is that swimming facility sub type can’t exist if there is no *Dalplex* facility

# Data Extraction

Hello. The data was scrapped from Dalhousie University’s official website. The code can be found under file *‘scrape.py’.* It contains 11 function, one each for scrapping data for separate entity.

The data for all entities except *‘Staff’* was scrapped using BeautifulSoup. Since the data for *‘Staff’* was generated dynamically on the front end, I mocked an API call to fetch the results.

# Data Loading

**App Review Loading**

The CSV were loaded in tables through MySQL workbench after I generated a column unique ID for the Application CSV and linked both the CSVs with those unique IDs.

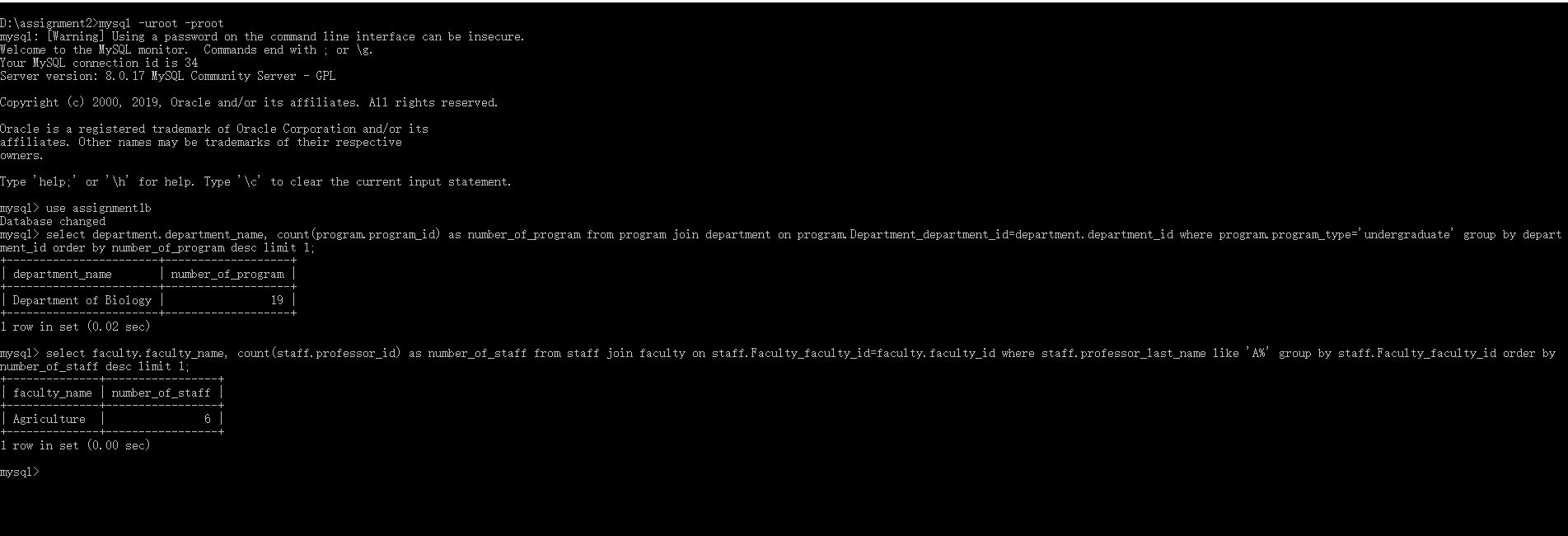
**Dalhousie Website Modelling**

The XMLs were loaded through CLI as workbench did not gave an option to import XML. I used the following commands to get the XMLs to load



# Queries

The dumps for both databases can be found under folder ‘*SQL\_Dumps*’. The answer for Part F of the assignment can be found under ‘*part\_f.sql’*. Following is a screenshot that shows successful results for both queries.



**Figure 5: Query Results**

# References

1. Dataset: "Google Play Store Apps", *Kaggle.com*, 2019. [Online]. Available: https://www.kaggle.com/lava18/google-play-store-apps. [Accessed: 29- Sep- 2019].
2. "MySQL :: MySQL Workbench Manual", *Dev.mysql.com*, 2019. [Online]. Available: https://dev.mysql.com/doc/workbench/en/. [Accessed: 29- Sep- 2019].
3. "MySQL :: MySQL 8.0 Reference Manual :: 13.2.8 LOAD XML Syntax", *Dev.mysql.com*, 2019. [Online]. Available: https://dev.mysql.com/doc/refman/8.0/en/load-xml.html. [Accessed: 29- Sep- 2019].
4. "Beautiful Soup Documentation — Beautiful Soup 4.4.0 documentation", *Crummy.com*, 2019. [Online]. Available: https://www.crummy.com/software/BeautifulSoup/bs4/doc/. [Accessed: 29- Sep- 2019].
5. "Dalhousie University ‑ Halifax, Nova Scotia, Canada", *Dalhousie University*, 2019. [Online]. Available: https://www.dal.ca/. [Accessed: 29- Sep- 2019].