A. Initial ERD:

Refer ERD_1 and ERD_2 [6]

B. Data Extraction and data cleaning:

Urllib2, BeautifulSoup, validator_collection, re, sys, spacy, csv, selenium, chromedriver - /usr/bin/chromedriver - libraries used [1] [2]

Regarding cleaning the Google Play Reviews data, I've considered the following points: [4][5]

- remove app name length with more than 50;
- remove multiple categories;
- consider genres as tags;
- price has free too which is equivalent to 0

Construction of xml structure, format used in is reference to MySQL manual.

Disclaimer

Name: Karthikk Tamil Mani
#ID: B00838575
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.

C. Data Insertion:

[3]

D. Final Data Modelling:

In MySQL Workbench I was not able to construct overlapping/disjoint subtypes, hence attaching only the subtype EERD here:

Refer EERD

Design Issues:

Time-variant data:
From the final EERD after normalization, time-variant data design issues are resolved. Staff's have Active and Inactive values, so that even if they move out for 1 term or left the job, we will still have the data. Courses also are combined with terms so as to avoid any issues.

Another issue is due to the data set available. I'm not able to have a primary key for the Staff since there are no values to uniquely identify the staff such as email , SIN or so on. So I'm using a composite key (FIRST_NAME, LAST_NAME) as the primary key even so I'm not able to uniquely identify the Staff. This design issue cannot be resolved since only name and department is to be scrapped.

Also there are no fan traps. Course in linked with term and Staff is linked with both the course and the term. Also Exams are linked with Course and Date. So as far as I see, there are no fan traps in this design.

E. Normalization:

Before Normalization, the table Program was not in normalized form. Removing redundancies, Department will have a separate table and the table Program would have satisfied 1NF and maintains atomicity. Now the table Program will have partial dependencies, PROGRAM_LENGTH and CAMPUS which is dependent on the PROGRAM_NAME which is candidate key. Hence to satisfy the 2NF, these partial dependencies are removed and Program and Faculty are separated. Finally, it will have 3 tables Faculty (containing data such as Faculty of Computer Science and so on), Department (Some faculties will have subdivisions as Department) and the Program table(which will have information regarding the undergraduate or masters program). On analyzing these tables which are in 2NF, there were no transitive dependencies and hence 3NF was achieved.

Similarly, all tables are checked for 3NF and is achieved.

F. SQL Query:

(Query after Normalization)

select FACULTY_NAME from Faculty where FACULTY_ID = (select FACULTY_ID from Program where PROGRAM_TYPE = 'undergraduate' group by FACULTY_ID order by count(FACULTY_ID) desc limit 1);

select DEPARTMENT_NAME from Department where DEPARTMENT_ID = (select DEPARTMENT_ID from Staff where LAST_NAME like ‘A%’ group by DEPARTMENT_ID order by count(DEPARTMENT_ID) desc limit 1);

References:

[1]https://spacy.io/
[2] https://github.com/insightindustry/validator-collection
[3] https://dev.mysql.com/doc/refman/5.5/en/load-xml.html
[4] https://support.google.com/googleplay/android-developer/answer/113469?hl=en
[5] https://blog.prioridata.com/the-complete-guide-to-app-store-categories
[6] https://www.vertabelo.com/blog/chen-erd-notation/