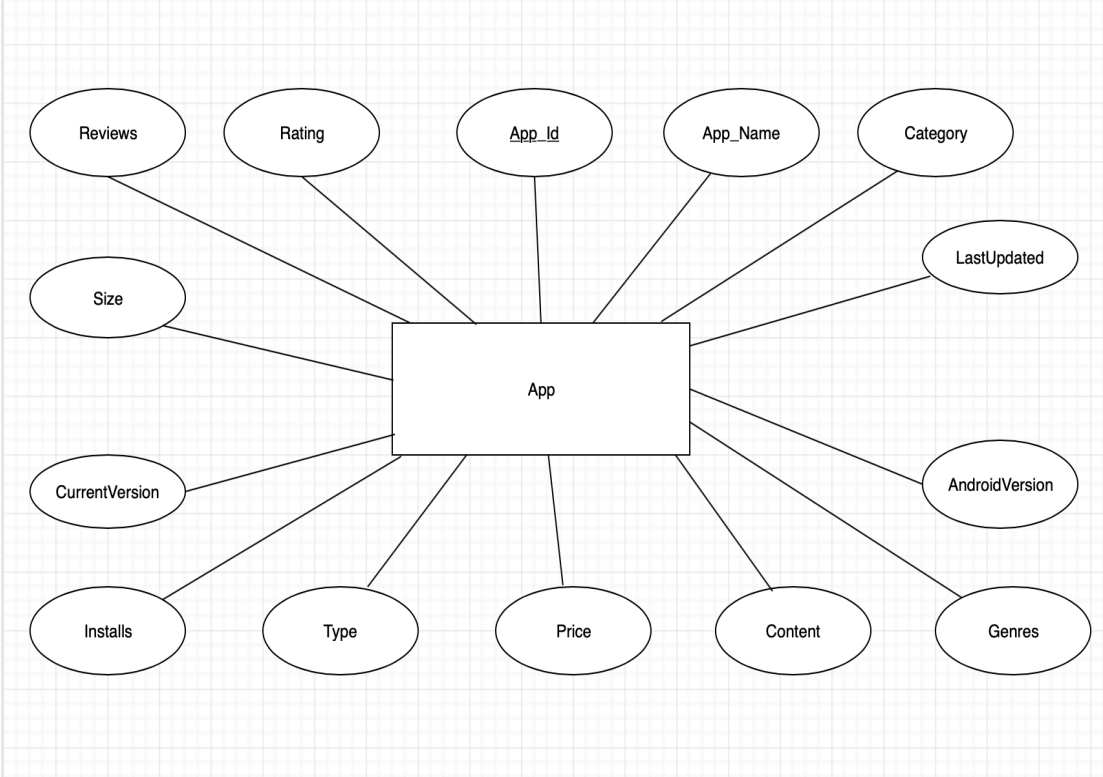
Name: RAVITEJA KASE #ID: B00823644

Disclaimer (Data Scraping)

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.

A. Data Modelling (Initial Design or the Conceptual Model)

1. **Initial data model from Google App Ratings Data:**
   Based on the data available in the googleplaystore.csv, entity and attributes are as follows.[2]
   Entity: App
   Attributes: AppId (PrimaryKey), Category, Rating, Reviews, Size, Installs, Type, Price, Content, Genres, LastUpdated, CurrentVersion, AndroidVersion.

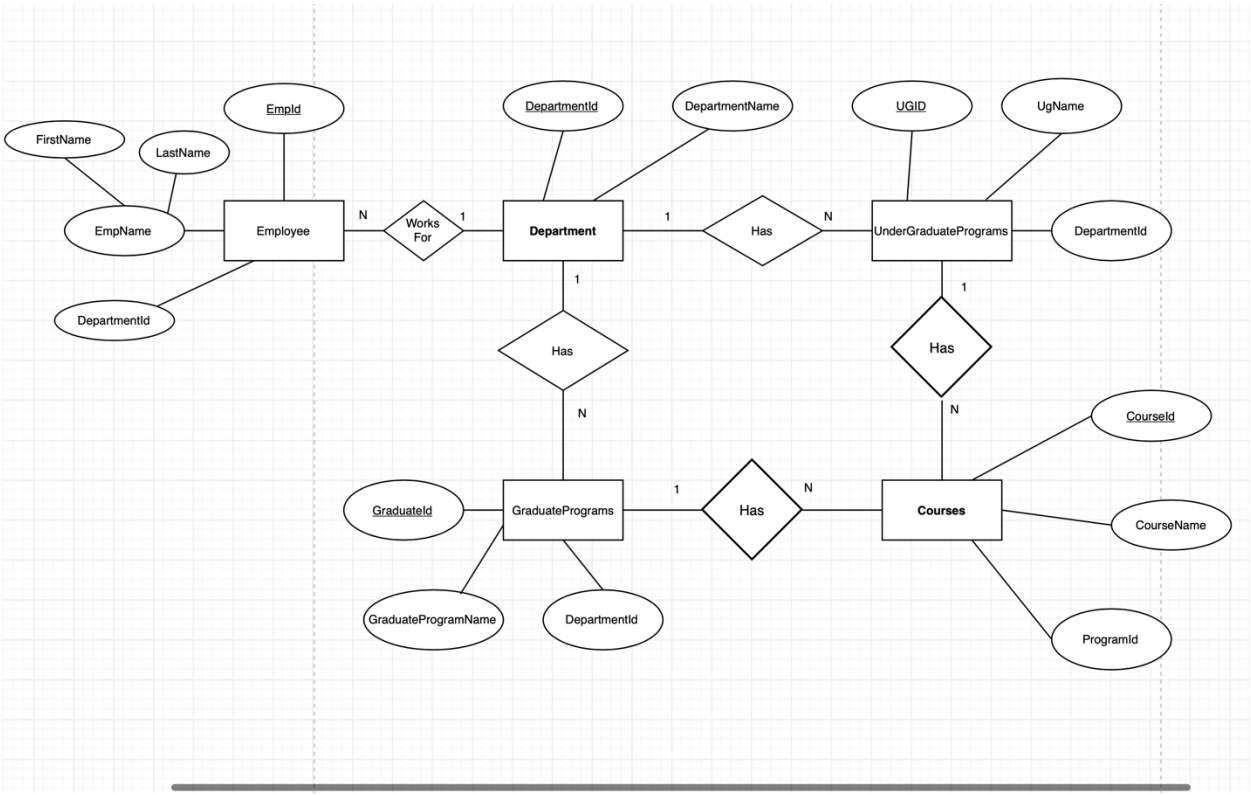


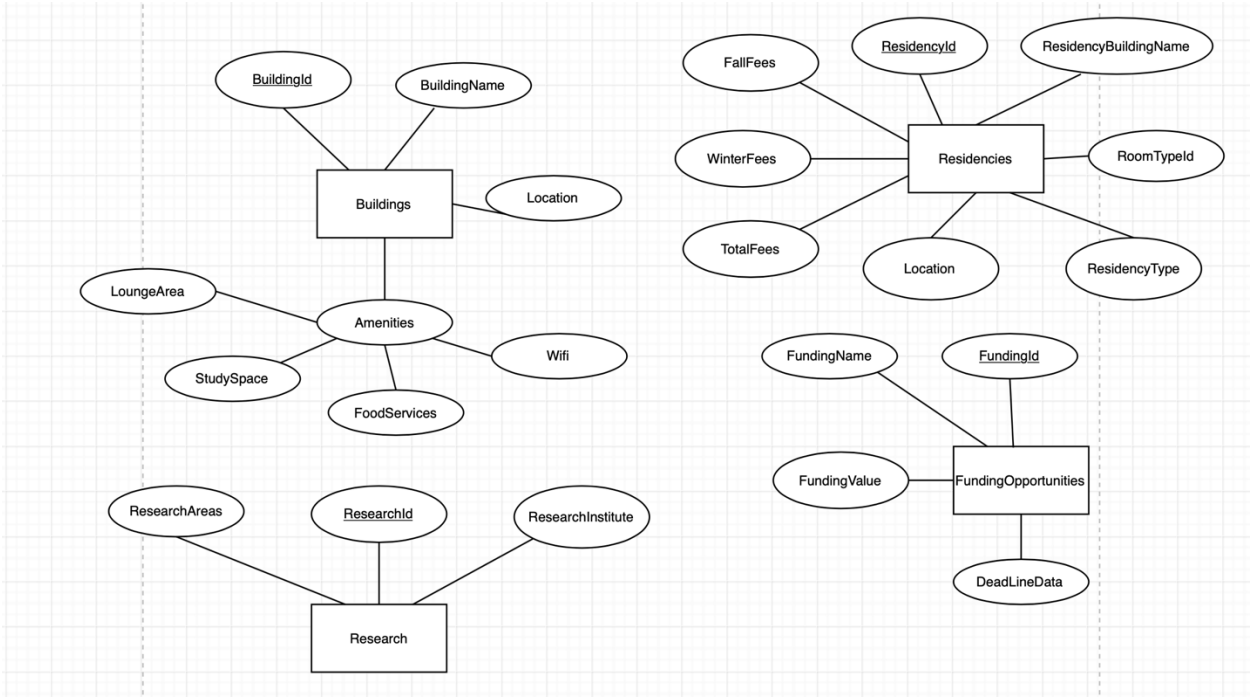
2. **Initial Data model from DAL website:**
   The objective is to design an information system for people who want to join the university as students, faculty member, or staff. Here are the observed entities from the DAL website.[2]

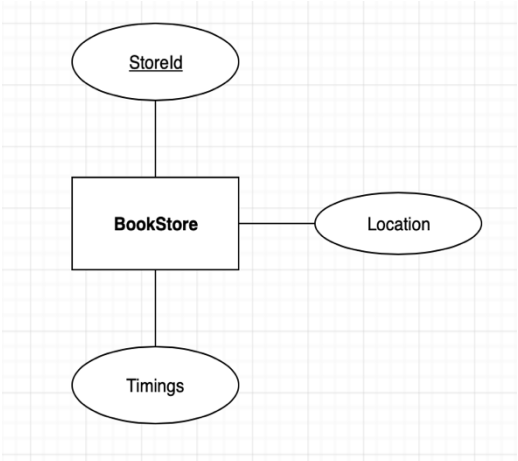
   1. Entity: Department
      Attributes: DepartmentId (Primary Key), DepartmentName
   2. Entity: Employee
      Attributes: EmpId (Primary Key), LastName, FirstName, DepartmentId
   3. Entity: UnderGraduatePrograms
      Attributes: UGID (Primary Key), UgName, DepartmentId
   4. Entity: GraduatePrograms
      Attributes: GraduateID (Primary Key), GraduateProgramName, DepartmentId
   5. Entity: Courses
      Attributes: CourseId (Primary Key) ,CourseName , ProgramId
   6. Entity: Buildings
      Attributes: BuildingId (Primary Key), BuildingName, Location, LoungeArea, StudySpace, FoodServices, Wifi
   7. Entity: Research
      Attributes: ResearchId (Primary Key), ResearchArea, ResearchInstitute
   8. Entity: Residencies

Attributes: ResidencyID (Primary Key), ResidencyBuildingName, RoomTypeId, ResidencyType, Location, FallFees, WinterFees, TotalFees

9. Entity: FundingOpportunities
   Attributes: FundingId (Primary Key), FundingName, FundingValue, DeadlineData
10. Entity: BookStore
    Attributes: StoreId (Primary Key), StoreName, Location







B. Data extraction and Data Collection.

1. Duplicate rows from the googleplaystoe.csv are removed and inserted into the KaggleApp database and into App table.

3. **Script to extract Data from DAL website and store it in XML Format**

Script is in PythonProgram-In_Src_Folder /src/. folder
Script – Dal_information.py
Xml files:

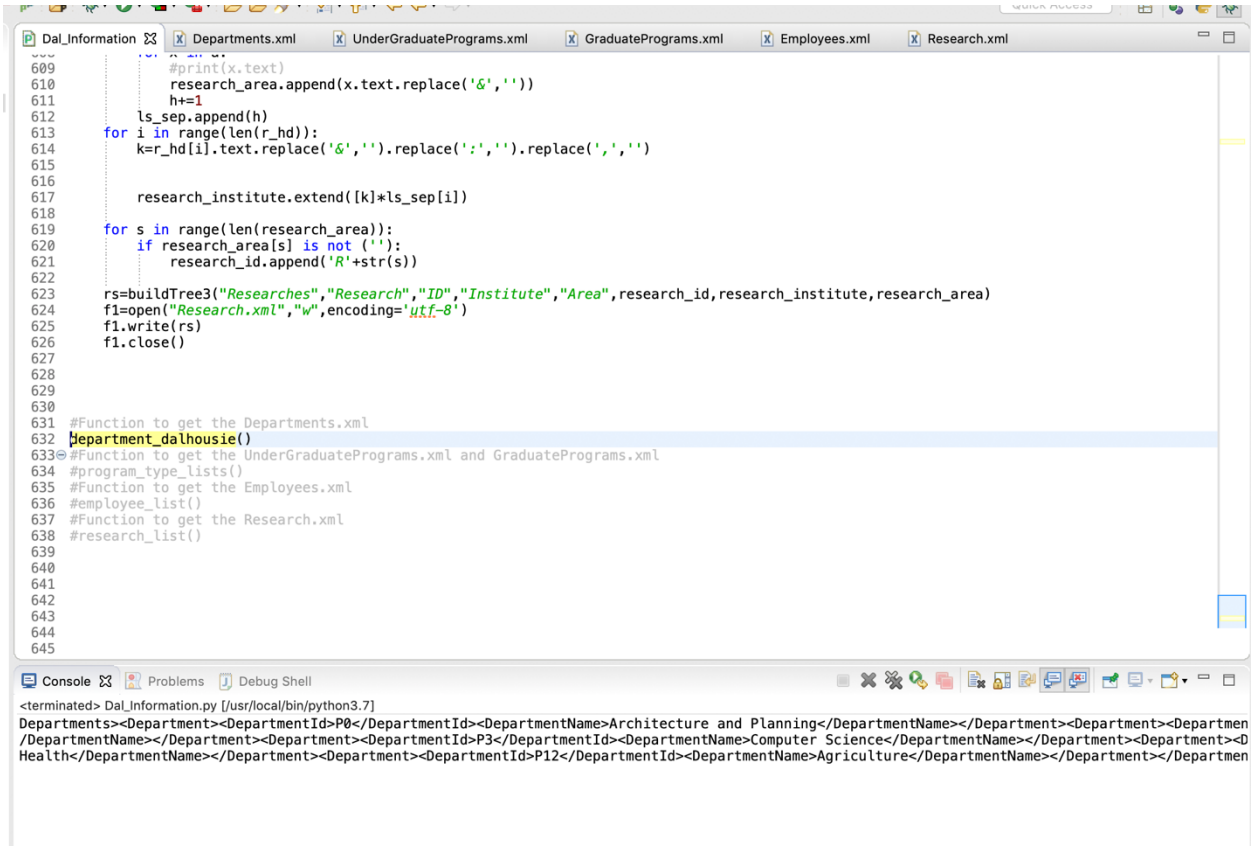
- Departments.xml
- UnderGraduatePrograms.xml
- GraduatePrograms.xml

- Employees.xml
- Research.xml

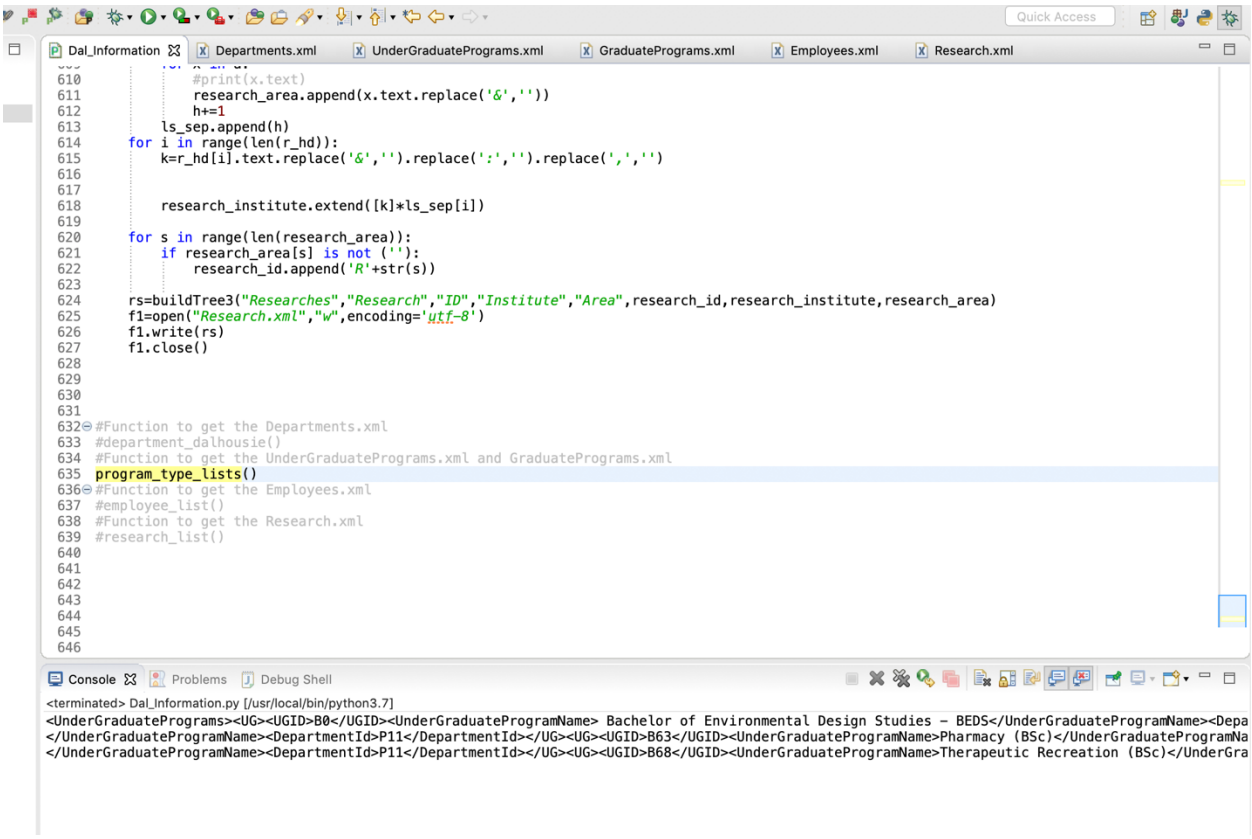
Libraries used:

- import requests
- from bs4 import BeautifulSoup[1]

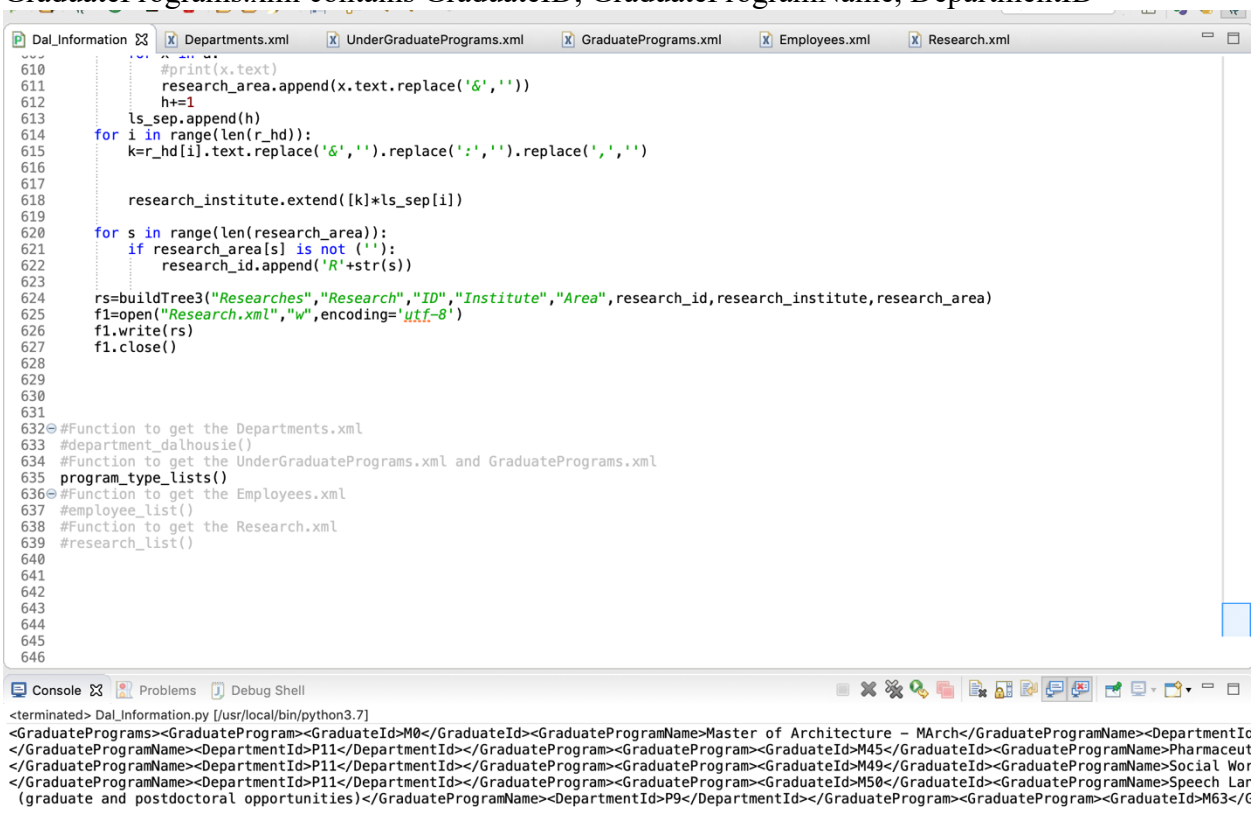
department_dalhousie() - Collects the data for faculties , faculty names are fetched and Ids are auto generated and loaded into Departments.xml file.
Departments.xml contains DepartmentId, DepartmentName



program_type_lists()- Collects the undergraduate and graduate programs from specific faculties and auto generates the UGID, graduated and loads into UnderGraduatePrograms.xml and GraduatePrograms.xml.
For the following faculties, UG and graduate programs are collected:
Architecture and planning, Arts and Science, Computer Science, Dentistry, Engineering, Health, Management, Medicine, Science, Agriculture.
UnderGraduatePrograms.xml contains UGID, UnderGraduateProgramName, DepartmentID



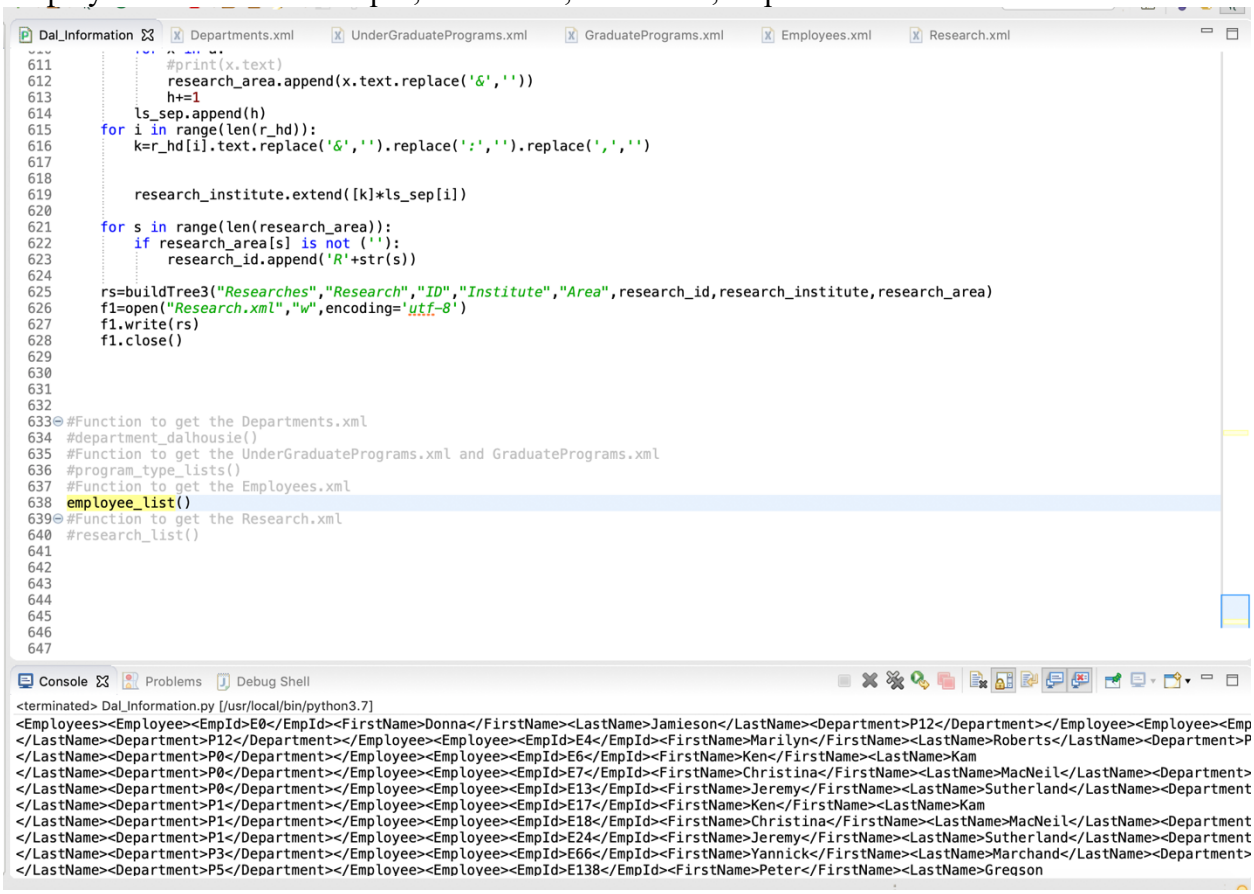
GraduatePrograms.xml contains GraduateID, GraduateProgramName, DepartmentID



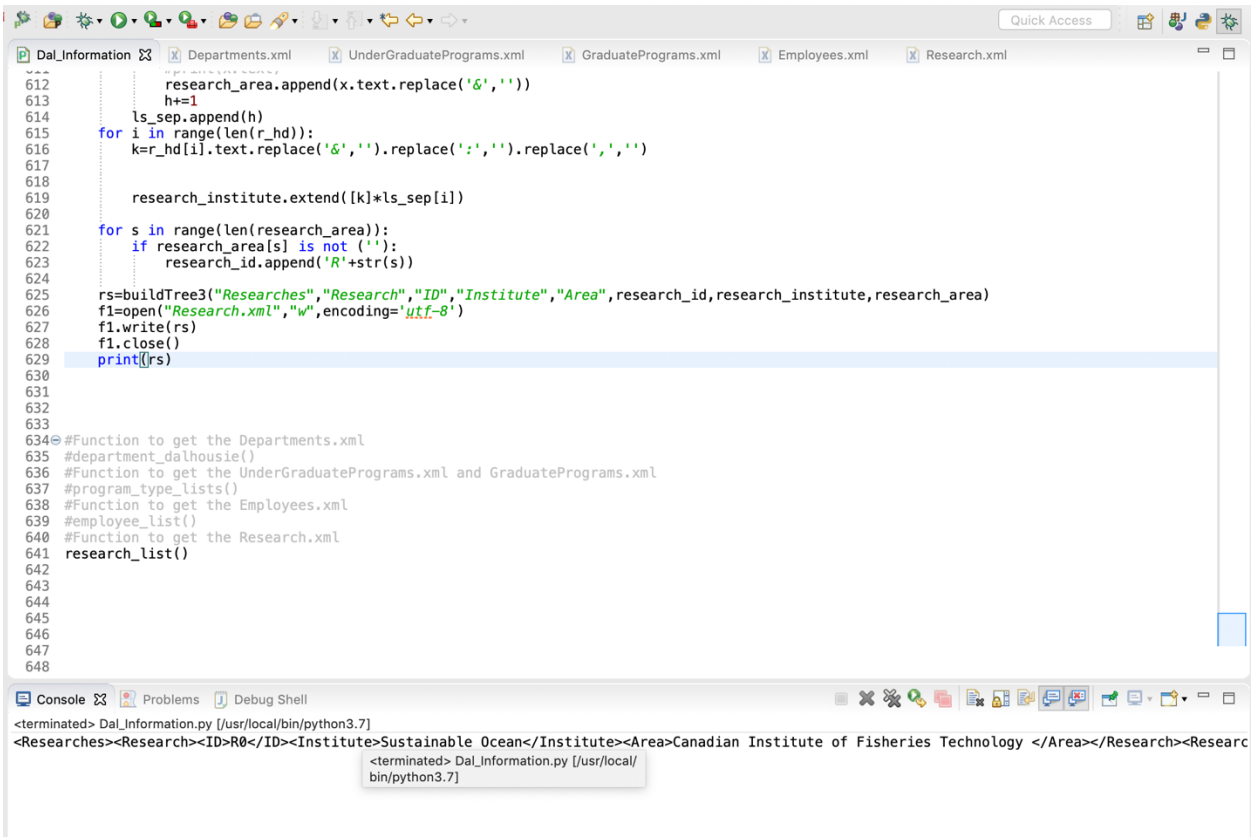
employee_list () -Collects the names of staff in each faculty and auto generates Id for each employee and loads into Employees.xml.
For the following faculties, employee details are collected:

- Staff in Agriculture (Animal science agriculture)
- staff in Architecture and planning
- staff in Arts and science
- faculty-staff in computer science
- faculty-staff in Engineering (Electrical)
- staff in health

Employees.xml contains EmpId, FirstName, LastName, Department



research_list () -Collects the research institutes and research areas done in each center and loads into Research.xml
Research.xml contains ResearchId, ResearchInstitute, ResearchArea



C. Data Insertion:

Dal_1 database is created in MySQL and the following tables were created and the respective xml files are used for insertion.

Department table –(Departments.xml)
Employee table-(Employees.xml)
UnderGraduatePrograms table –(UnderGraduatePrograms.xml)
GraduatePrograms table-(GraduatePrograms.xml)
Research table –(Research.xml)
Course table
Residencies table
FundingOpportunities table
BookStore table
Buildings table

D. Final Data Modeling:

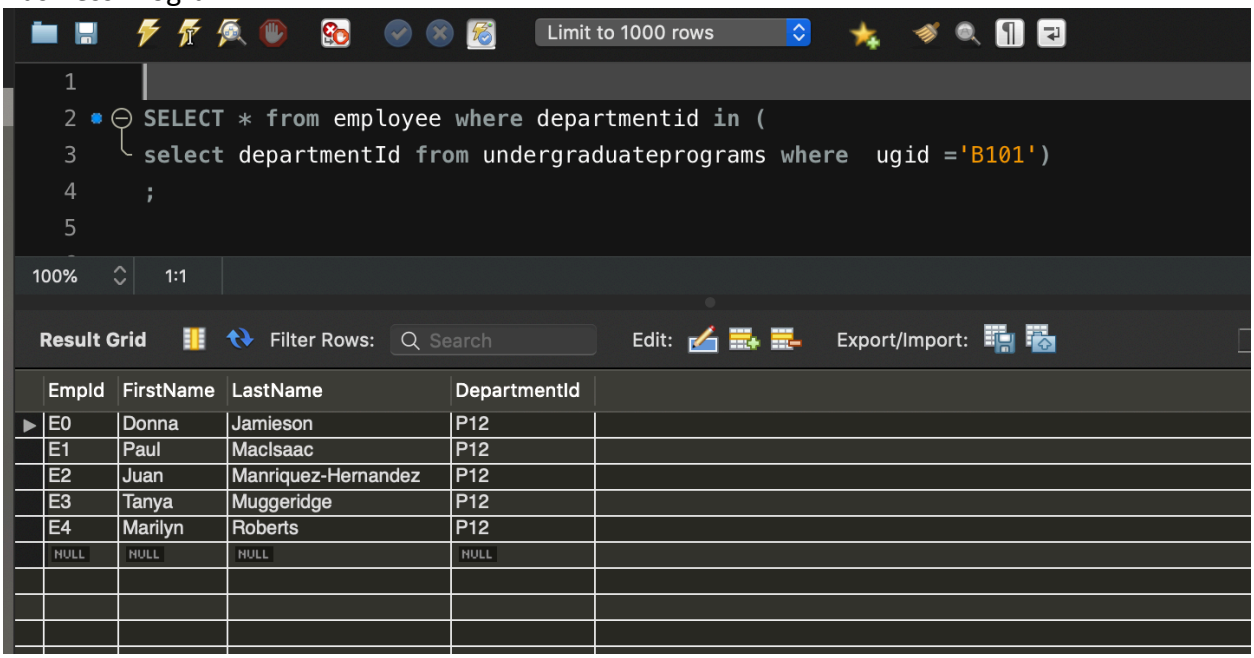
1. Is your initial sketch/design free from any design issues?

No, Fan trap design issue is present.[3]

Department entity is in three 1:M relationships (Employee, UnderGraduatePrograms, GraduatePrograms).

Justification: According to the design, an employee belongs to a specific department. Each department has particular undergraduate programs are graduate programs.

Each employee who works for particular department will work for all the undergraduate programs as well as graduate programs falling under the department. For example, professor can teach for MACS as well as MCS. Similar way administrative staff also works for all the programs in a particular department.
So, it is not required to have one- one mapping for Employee and Undergraduate Programs and Employee and Graduate Programs.
If queried, we can be able to fetch the employee names who work for a particular undergraduate program.
The below query fetches the employee names who work for “International Food Business Program”.



Similarly, if queried we can be able to fetch the undergraduate programs which belong to particular department, where employee works.
NOTE: In the website, faculty and staff list also provided for specific departments. So the design with Fan traps would not cause any problem.

Final Design:

1. Identify the relationships, and cardinality between the entity sets you created.
    - DepartmentId from Department is referred in undergraduatPrgrams. Department has UnderGraduateProgramas [1:M] relationship
    - DepartmentId from Department is referred in GraduatePrograms. Department has GraduatePrograms [1:M] relationship
    - DepartmentId from Department is referred in Employees. Department has Employees [1:M] relationship
    - UGID from UnderGraduatePrograms is referred in Courses as ProgramId UnderGraduatePrograms has courses [1:M] relationship
    - GraduateID from GraduatePrograms is referred in Courses as ProgramId GraduatePrograms has courses [1:M] relationship

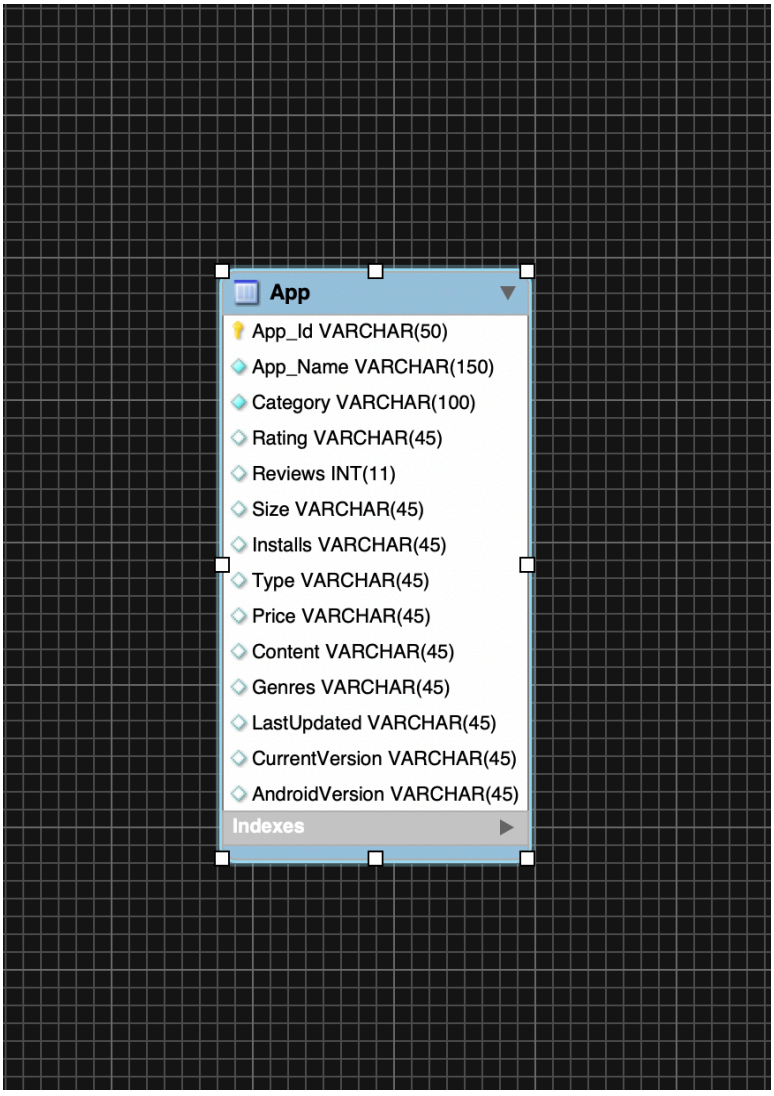
2. Construct an extended ERD. Your ERD should highlight if any overlap/disjoint subtype exists.
   Following ERD is generated and is present in the images folder.ERD doesn't have overlap/disjoint subtypes.
3. The extended ERD should be created using a tool, such as MySQL workbench/ ErWin/ Visio etc.,
   Extended ERD is generated using MySQL workbench.

E. Normalization:

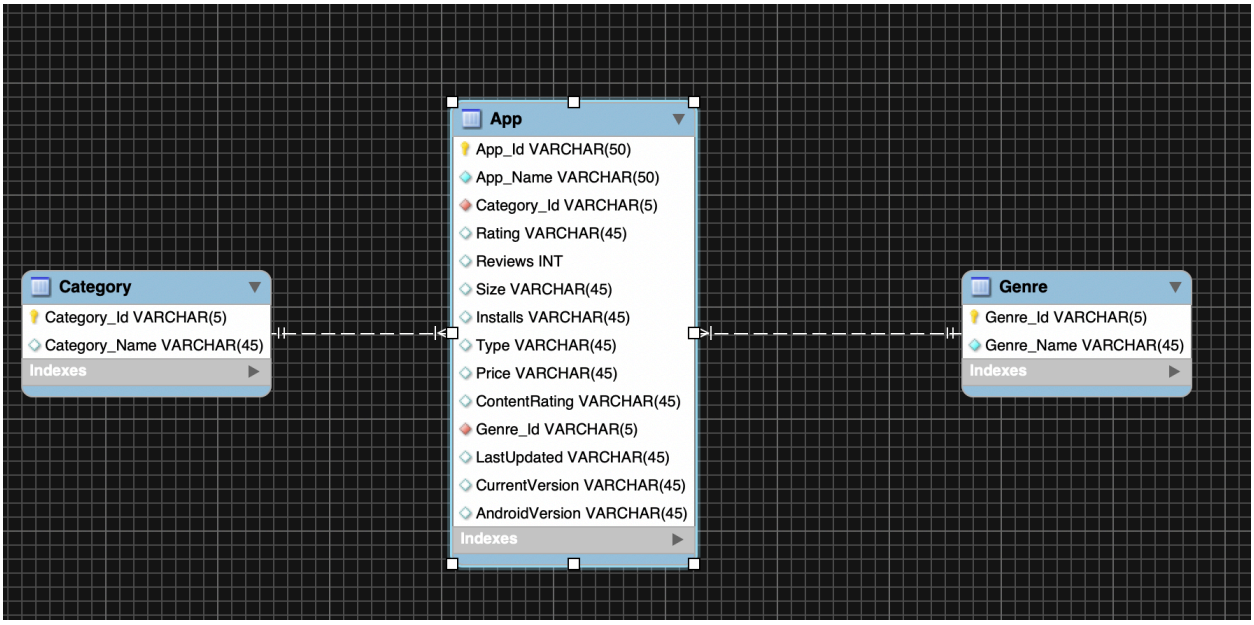
Google Play Store data:

App table is in 2NF with no transitive dependencies. However, the category and genres of app can increase at some point of time. So, App table is broken down into App table, Category table (CategoryId, Category name) and Genre (Genreid, Genre name)

Before Normalization:



After Normalization:



Dalhousie Website:

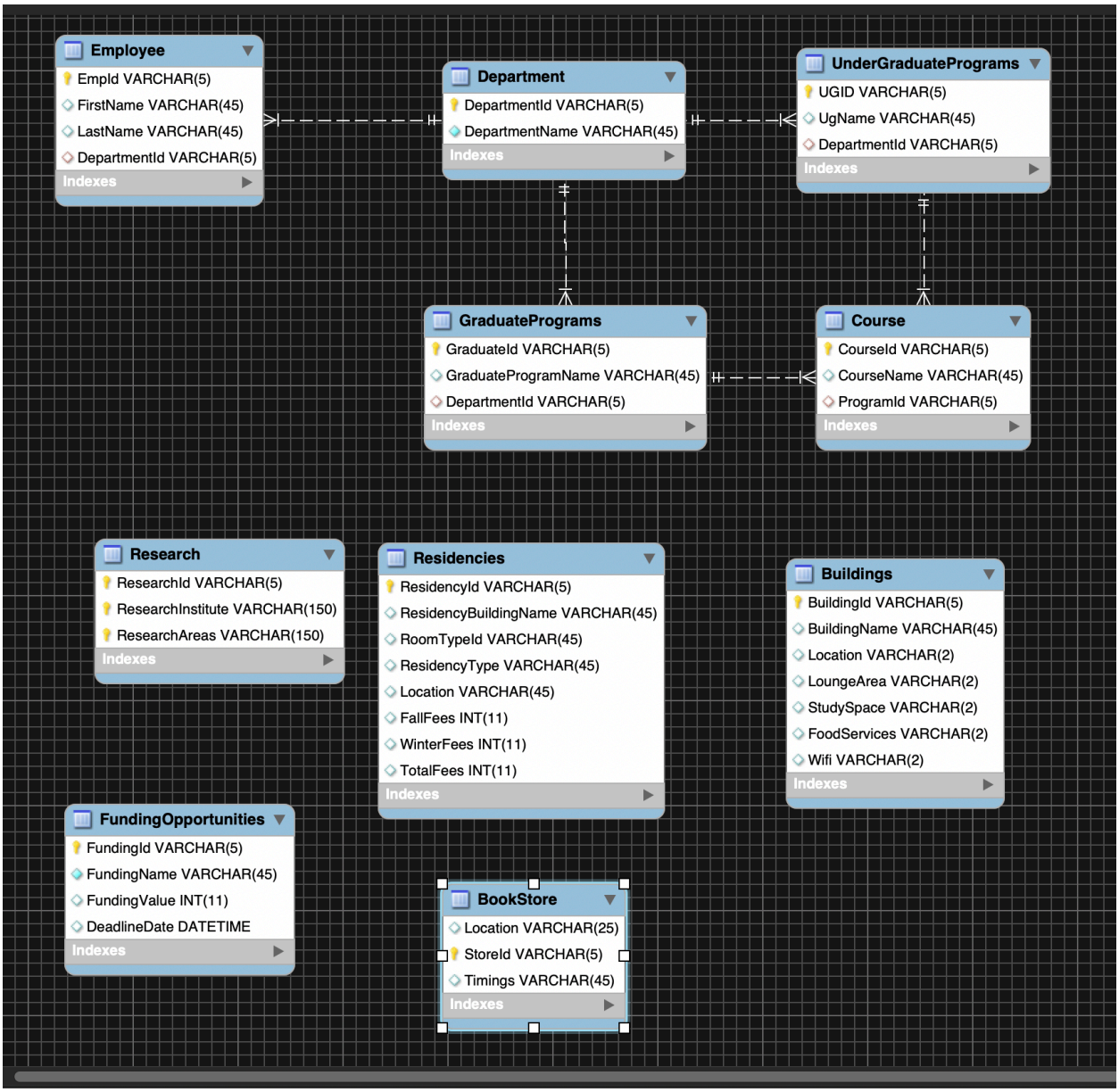
As while data extraction from the website, data was collected accordingly. For example, departments data is collected from and stored. Similarly, data is collected for undergraduate programs, graduate programs, employees table.

Normalization is done for Research table

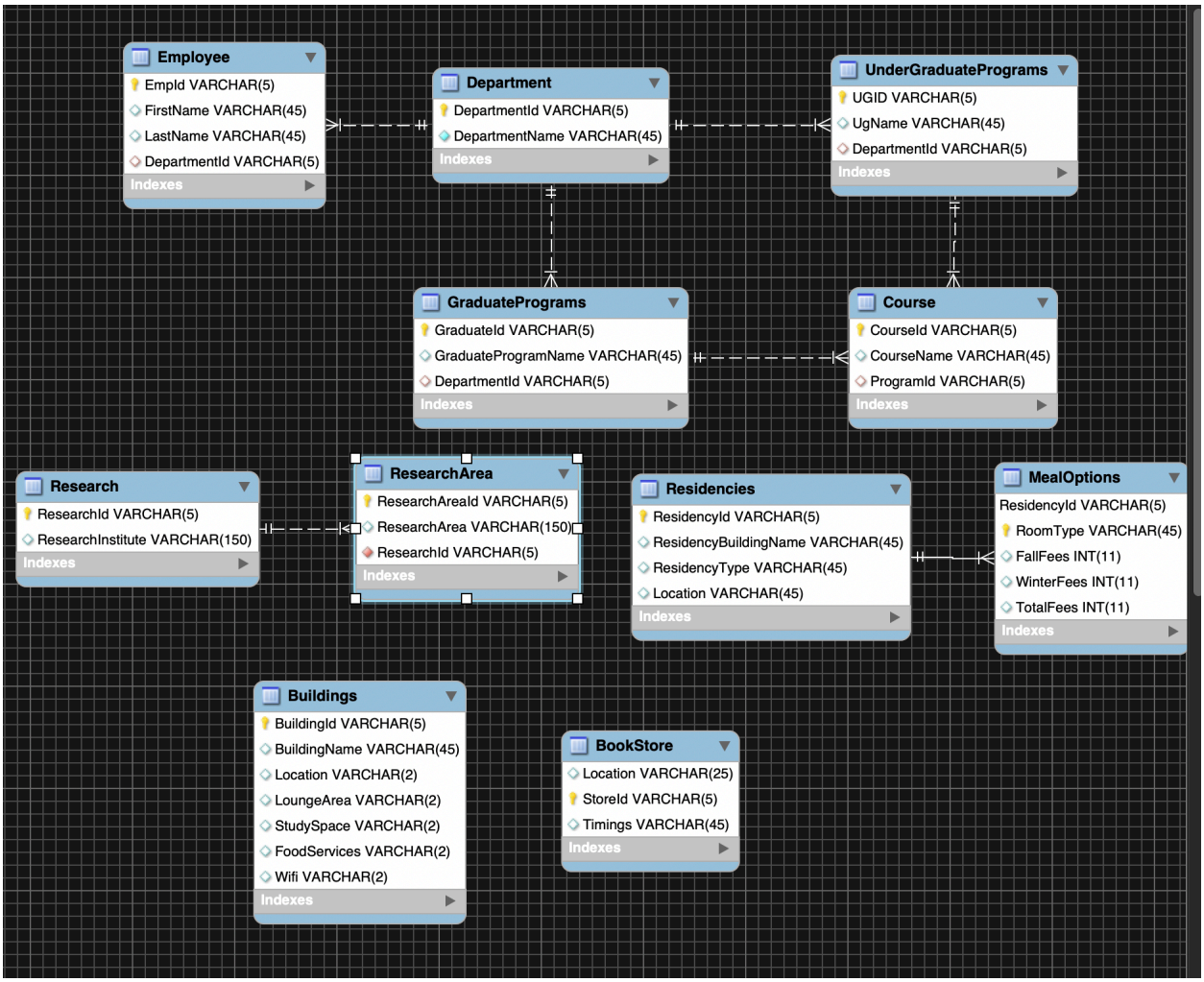
Research table have ResearchId, Research Institute and Research Area. So, research Area is partially dependent on both research Institute and ResearchId.

So, tables were broken down to create a 3NF form. Research table contains Research Id and ResearchInstitute and Research Area table contains Area Id, Research Area and Research ID which is a foreign key and refers Research table.

Before Normalization:



After Normalization:



F. Write SQL Query:

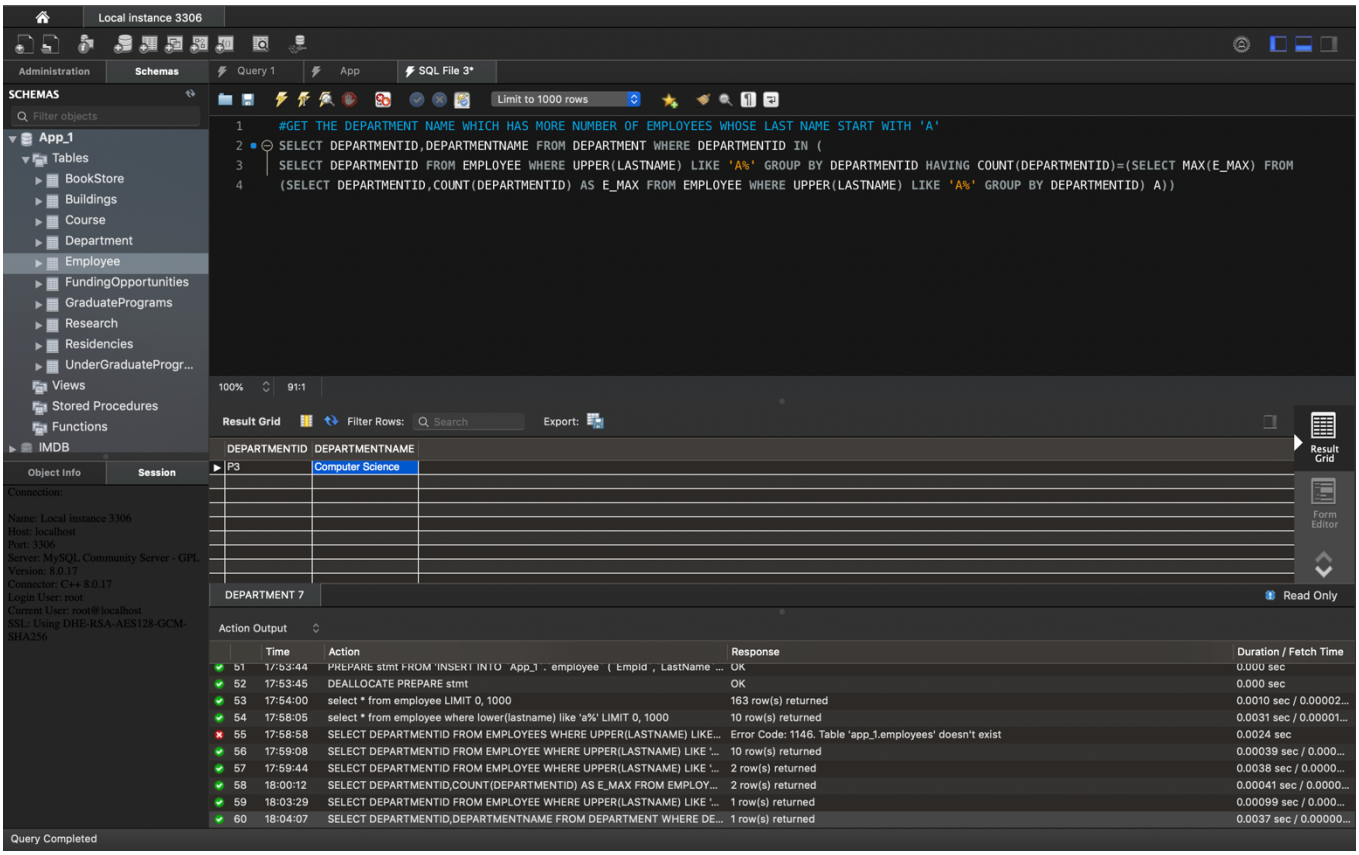
As the queries required Department, Employees , Undergraduate programs , these tables were not required to further normalization , same queries would fetch the results.

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

   #GET THE DEPARTMENT NAME WHICH HAS MORE NUMBER OF EMPLOYEES WHOSE LAST NAME START WITH 'A'

   SELECT DEPARTMENTID,DEPARTMENTNAME FROM DEPARTMENT WHERE DEPARTMENTID IN (
   SELECT DEPARTMENTID FROM EMPLOYEE WHERE UPPER(LASTNAME) LIKE 'A%' GROUP BY DEPARTMENTID HAVING COUNT(DEPARTMENTID)=(SELECT MAX(E_MAX) FROM (SELECT DEPARTMENTID,COUNT(DEPARTMENTID) AS E_MAX FROM EMPLOYEE WHERE UPPER(LASTNAME) LIKE 'A%' GROUP BY DEPARTMENTID) A))

Answer is Computer Science .

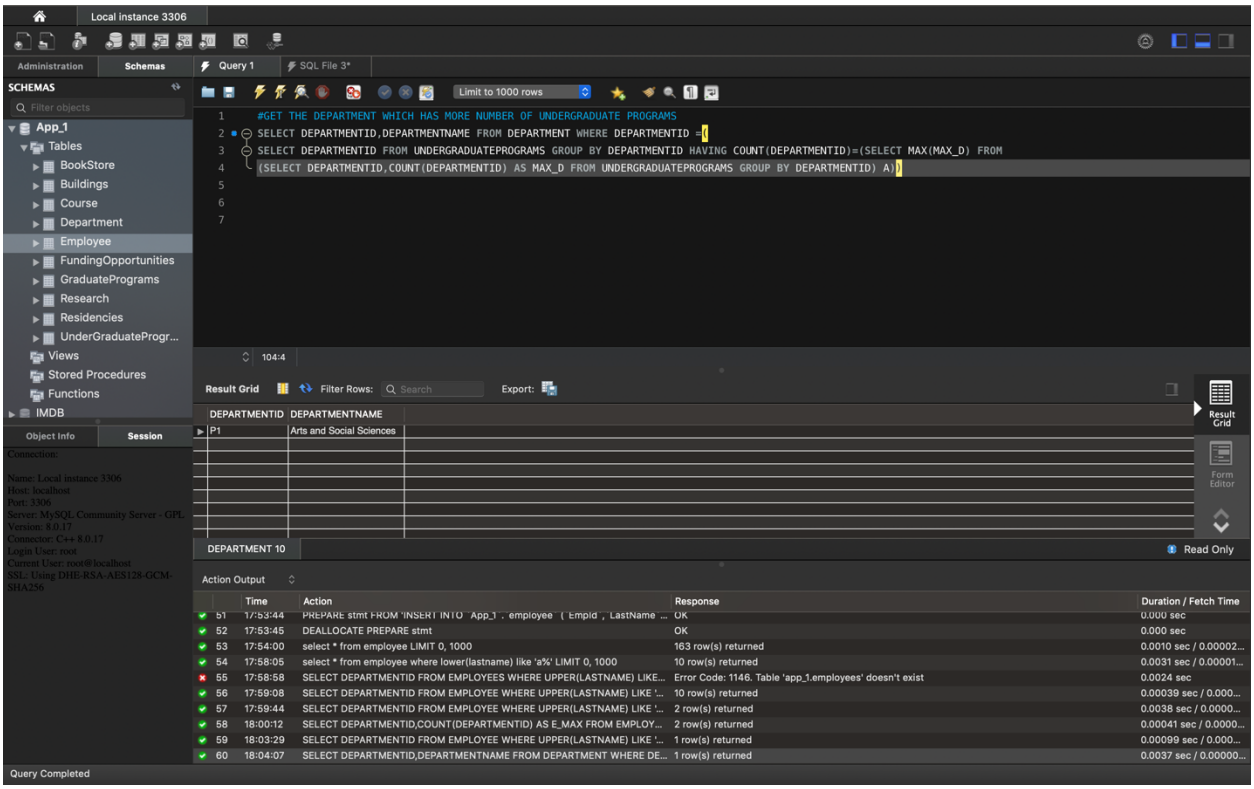


2. Find the name of the department or faculty that has the highest number of undergraduate programs

   #GET THE DEPARTMENT WHICH HAS MORE UNDERGRADUATE PROGRAMS

   SELECT DEPARTMENTID,DEPARTMENTNAME FROM DEPARTMENT WHERE DEPARTMENTID=
   (SELECT DEPARTMENTID FROM UNDERGRADUATEPROGRAMS GROUP BY DEPARTMENTID HAVING COUNT(DEPARTMENTID)=(SELECT MAX(MAX_D) FROM (SELECT DEPARTMENTID,COUNT(DEPARTMENTID)AS MAX_D FROM UNDERGRADUATEPROGRAMS GROUP BY DEPARTMENTID)A));

   Answer is Arts and Social Science



References

[1]. Tutorial: Python Web Scraping Using BeautifulSoup.[Online].
Available: https://www.dataquest.io/blog/web-scraping-tutorial-python/ . [Accessed 20th September 2019]

[2]. Data Model:]Ramez Elmasri, Shamkant B. Navathe, "*Fundamentals of Database Systems*", *7the*

[3]: Fan trap design issue[Online]: DBMS Concepts_ERD_Normalization lecture slide in *Brightspace.*