Report on Assignment 1

Assignment 1a

In the first of the assignment we were told to clean the data from GoogleApp data from Kaggle. There were two files, namely, the googleplaystore.csv and the googleplaystore_user_review.csv, in the kaggle repository.

Both the csv files were instructed to be cleaned. The googleplaystore.csv and the googleplaystore_user_review.csv were read into python using Pandas. Jupyter Notebook have been used to clean the data.

The Cleaning Process

In the googleplaystore.csv file it contained the fields, App name, Category, Rating, Reviews (in numerical form), Size, Installs, Type, Price, Content Rating, Genres, Last Update, CurrentVersion, and Android Version. In the cleaning process, the following fields have been omitted by dropping them using pandas:

- Price: This field have been dropped because none of the application listed in the dataset did not have any price and that is why it was insignificant to keep that as either as an entity nor as an attribute since it did not play any role in the google app rating.

- Content Rating: This field have been dropped since there is already an attribute on rating the application, in numerical form.

- Genres: This have been dropped since the Category field already covers the same purpose. The genre of the application and the Category of the application is the same for which reason it had been dropped.

- Type: This field is similar to price, Because, it signified whether the application was free or not. Since they were all free that is why there was no good reason to keep that field.

Duplicate records were also handled by omitting them from the googleplaystore.csv dataset and records that contain null values (given by Nan) have also been ommitted (dropped) from the dataset. The null values are removed because they are considered to be noisy data and can hamper the accuracy of the result of the aim the research where the dataset will be used.

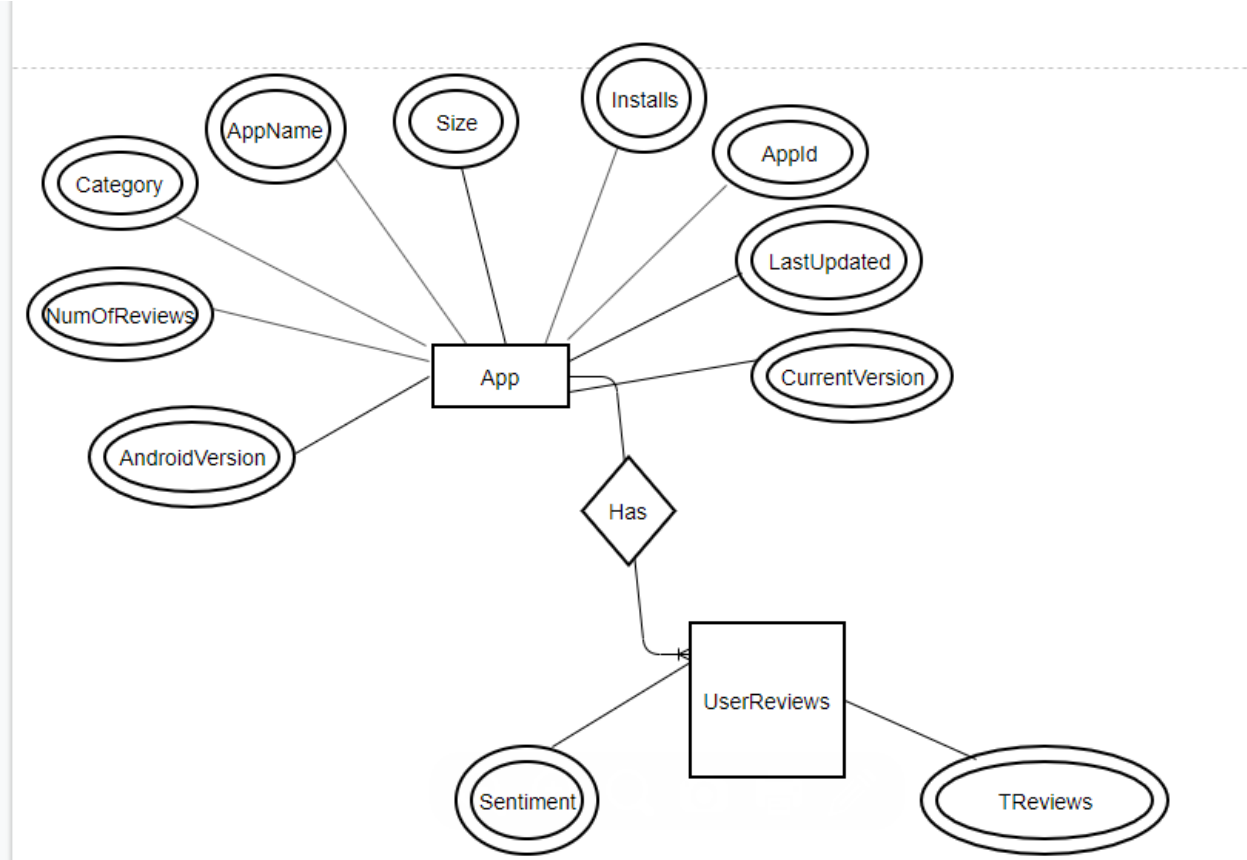
In the cleaning process of the googleplaystore_user_review.csv file, the null values and the duplicate values have been dropped for the same reason as stated above.

- Sentiment_Polarity: This field refers to user's sentiment towards the application which ranges from 1-0 where the numbers towards the 1 means it is positive and toward the 0 means it is negative. It is a float value. It is insignificant since there is already a field on Sentiment which states the same output.

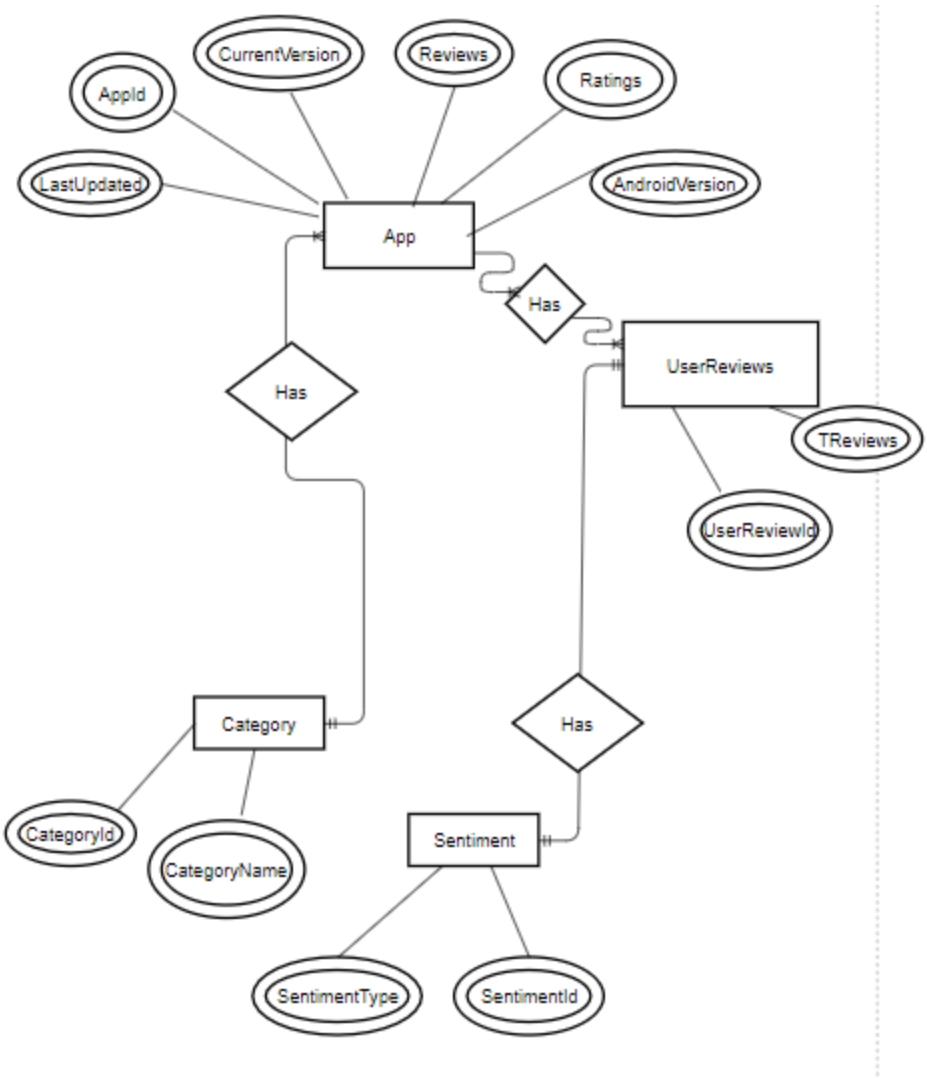
- Sentiment_Subjectivity: It is the personal view point of the user toward the application. This is also very similar to the Sentiment for which this have been removed as well.

- App: The App name have been removed since the App name have been made into an entity which have been taken from the googleplaystore.csv file. This will be explained laer in this section on the planning of the Entity-Relationship diagram.

The ERD diagram planning



In the above ERD (Entity-Relationship diagram), there are two entities, namely, App and UserReviews. The App has been considered as an entity from the googleplaystore.csv which has been normalized to 1NF. Each cell in the dataset had only one value. All the entries of the column are of the same datatype. This not the normalized form. That is, the above ERD have been further normalized which will be discussed in the paragraph below.



In the above diagram, the ERD has been normalized from 1NF to 3NF. BY normalizing, the ERD has four enitities, namely, App, UserReviews, Sentiment and Category. Since, the Sentiment values are either “Positive” or “Negative”, that is why it is best to put them into a table and give it an ID (SentimentId). The “Category” also has been seperated into a different table and so has been “UserReviews”. Since, both “Category” and “UserReviews” have a set of common values which can be held by an ID. An application can have many UserReviews since it has many users but one user can review one application at a time. The user cannot give multiple reviews to the same application. An application can fall under only one category and not multiple categorey. However, each Category can have multiple application under its category. And each UserReview can have one “Sentiment” since the user can have only one sentiment to an application.

Assignment 1b

In this assignment it had many many parts to it. The data would have to be scraped from the website of Dalhousie University and inserted into xml files and then that would be fed into the database. And then make an ERD after normalizing the data.

The data have been scraped using Beautiful Soup which is a library in Python. Python have been chosen for this assignment due to previous exposure to this langauge. After scraping the data from the necessary web links they were then fed into an xml file. The xml file was created by coding it in oython and the scraped data was put into the xml file in the form of strings as shown in Figure 4.

```
from xml.dom import minidom
import os

from urllib.request import urlopen as uReq
from bs4 import BeautifulSoup as soup

dal_faculties_url = 'https://www.dal.ca/academics/faculties.html'
#opening up connection and grabbing the page
uClient = uReq(dal_faculties_url )
page_html = uClient.read()
uClient.close()
#html parser
page_soup = soup(page_html, "html.parser")
faculties_container = page_soup.findAll("div", attrs={"class":"text parbase section"})
num_faculties = print(len(faculties_container))

```

Figure 3. Scraping Method

```

f = open("Dal_Faculty.xml", "w+", encoding = 'utf-8')
i = 0
f.write("<table name = \"Dal_Faculty\"> ")
for x in faculties_container[1:]:
    facultyName = x.find('a').text
    i = i+1
    f.write("<record><column name=\"facultyId\">" + str(i) + "</column>")
    f.write("<column name=\"facultyName\">" + facultyName.replace('&', '') + "</column></record>" )

f.write("</table>")

```

Figure 4. Xml File creation and data Insertion

The issue faced during scraping the data was that some web pages could not be scraped as the tags were in javascript and some had privacy concerns like faculty members' information for which it could not be scraped.

The scraped data is fed into the xml file as shown in Figure 4 by means of creating strings in the format of xml files.

The data is then fed in to the MySql Workbench where the ERD has been made for the entities which have been chosen.

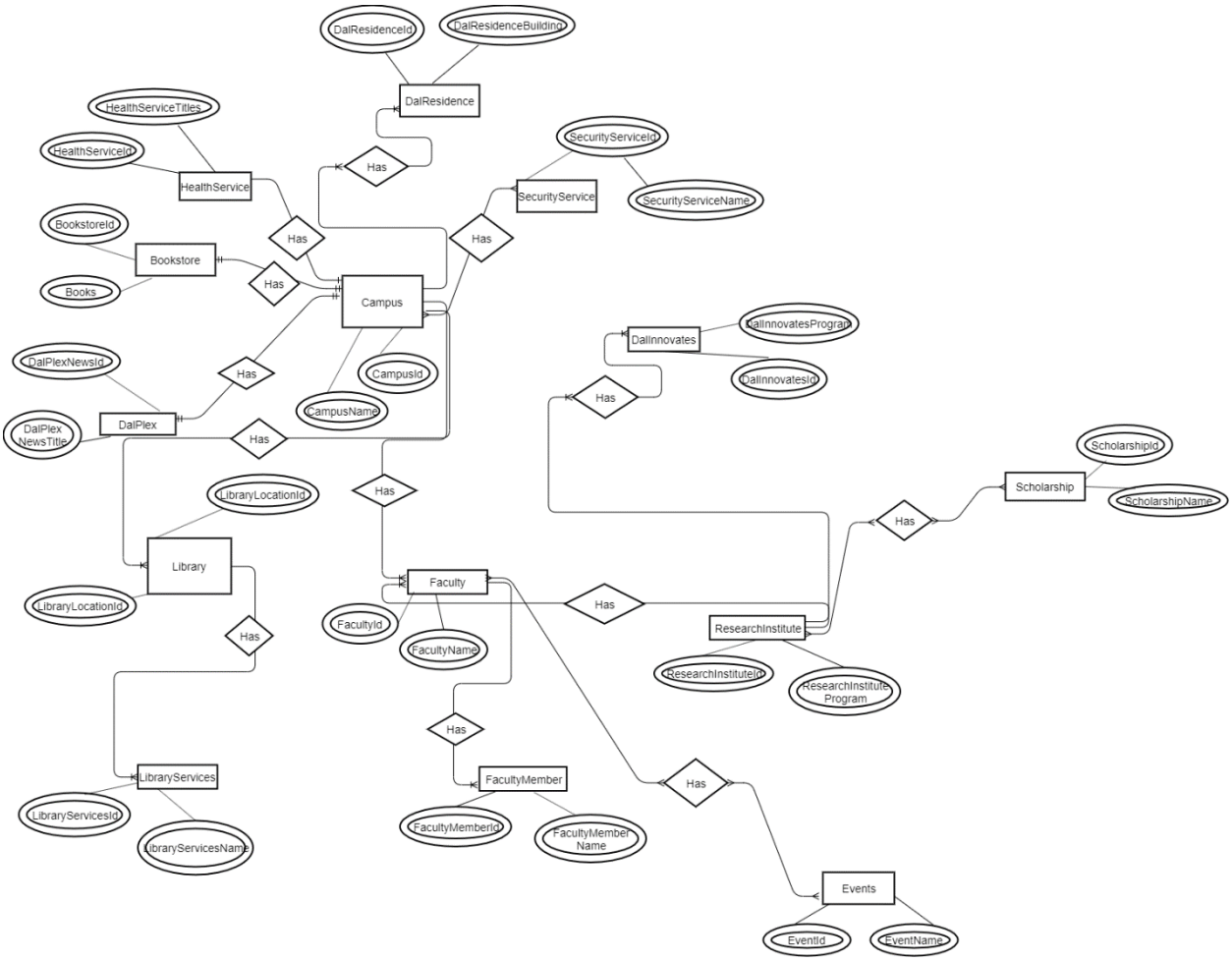


Figure 5. The ERD of Dalhousie

The main entities of the ERD are Campus, Faculty, ResearchInstitute, and Library. The Campus has its own enitities which are SecurityService, HealthService, BookStore, DalPlex. The Faculty entity has FacultyMember as its own entity and FacultyMember has Events as its own Entity.

Since the There are two Campus and only one has Dalplex that means each campus have one DalPlex that is why it is a one-to many relation from Campu to Dalplex. For the DalPlex entity, the attributes, DalplexNews and DalplexNewsId have been taken. As for the Campus the attributes are CampusId and CampusName. Each Campus have one Bookstore for which it is also a one-to-one relation. Each Campus have many residences for which the relation from Campus to DalResidence is one-to-many. Each Campus have many Security services for which the relation from Campus to SecurityServices is one-to-many relation. Each Campus have many Libraries for which it is a one-to-many relation.

The Library has a weak entity, LibraryServices because with the Library there would not be any Library services. Each Faculty has many FacultyMembers for which it has a relation of one-to-many and the FacultyMember entity is weak type. Each Faculty has many Events and that has a relation one-to-many. The ResearchInstitute provides many scholarships for which it has a one-to-many relation since Dalhousie has one research institute. DalInnovates is supported by the ResearchInstitute of Dalhousie university for which it is a one-to-one relation. The ResearchInstitute has many researches from different faculties for which the relation for the ResearchInstitute to the Faculty is one-to-relation.

The weak entities are LibraryServices, Events, SecurityServices, Bookstore, DalPlex, DalResidence as they are all dependent on the entity. LibraryServices is dependent on the Library. The SecurityServices, Bookstore, DalPlex and HealthService are all dependent on the Campus and that is why they are weak entities. The Events is a weak entity as it depends on the entity Faculty.

The ERD has not been normalized any further after creating the intial ERD. Because it is already in 3NF. As each entity has two attributes which is their Id and the name being the other attribute.

There were no design issues as I have kept it very simple and there are no overlapping either.