CS327E Elements of Databases Final Project Essay

Introduction

In this essay, me and my partner will provide an overview of the Elements of Databases class, and then move on to a discussion of each of the 4 labs we have done. The format will be 1. Lab Overview 2. Core work done in the lab 3. Debugging the Lab 4. Unsolved issues 5. What we learned from the lab. We will tackle all 4 labs using this format in order to delve deep into the machinations of each, and to demonstrate our understanding of the material. This essay will be a way to summarize the overall expected topics, workload, and difficulty level of CS327E Elements of Databases class taught by Professor Shirley Cohen.

So, now let us move on to an overall description and generalization of the class. This class is all about relational databases (we use MySQL) and designing database applications. We use Python (PyMySQL) to create connections to the database and to run the queries to populate the databases and even create a user interface (command line). The class focuses less on the administrative side databases and instead arms students with the knowledge to create theier own databases and how to design them with maximum efficiency. Not only that, but students also learn to initiate queries to gleam information from the databases using the application they have made. Information is provided to students on the best practices to protect their database from malicious SQL injections, as well as giving them a glimpse of Big Data and Data Warehousing practices. Students work on the labs in pairs, with the first 3 labs building up to the Final project (4^{th} lab).

The data to be used for the labs is up to the groups and this approach lets students be excited for the labs as they are allowed to choose a data set they are interested in. The dataset chosen in the

first lab is used in consecutive labs, and each lab builds upon the previous and this progression is something we really enjoyed about this class. This process of taking something as simple as dataset and evolving it into a complex beast of an application by modeling the data, deciding relationships, creating interesting queries to extract information from it etc. really imbues the student with a sense of accomplishment by the time they finish the final project. Also, the advantage of building upon previous knowledge and experience in developing a large project is essential to an aspiring computer expert. The class arms the student with practical knowledge as well as hands on experience in creating their own project essentially with semi-strict guidelines which are always beneficial and never in excess or redundant.

Now that we have outlined the basic format of the labs and provided a general idea of lab progression and development cycle, we will get to the meat of the essay, the labs themselves. The following sections will provide information on the 4 labs we have completed adhering to the previously mentioned format in the hopes that the reader can find valuable insight in the machinations of databases.

Lab 1

In lab 1 our learning objectives were to get acquainted to and work with our partners. The other objectives were to work with real life data (the dataset we choose), create data models that describe our data, design a relational scheme of the database in MySQL and learn how to use Lucid chart (for creating schemas – conceptual and logical models) and gain some experience in using GitHub to share our project as well as manage “issues” which is GitHub’s system to track project progress.

Now let us get into the bread and butter of the project. After completing the pre requisites of the project which were setting up GitHub account with a private repo for the project, signing up to Lucid Charts and getting the MySQL server up and running on our computer, we proceeded to find a data set we were interested in. This took the bulk of our time as it was the single most important part of the lab as we would use this dataset for all consecutive labs. We decided to go with a game information dataset which had information of all games released in recent memory and had information such as genre, sales, reviews, number of reviewers and many other interesting information. To complement this dataset, we also got a simpler dataset of movies with its title, genre and release year. Our intentions of picking these two datasets were to see how many movies were eventually made into games and how many games were made into movies, as well as how the average reviews of these translated games and movies hold up to standalone movies and games. We were also interested in the sales data of these translated games when compared to standalone games. After we had the data set, we focused on creating a conceptual model for our two datasets by deciding on the overarching entity classes and what attributes they would contain. The attributes are generally grouped under these entity classes and they are the columns of information in our datasets. We chose the most interesting attributes we could find such as review score and reviewers for games and grouped them under their appropriate entity classes (the Reviews entity class in this case) and used the very useful Lucid Chart to draw said relationships. We also wrote a “data dictionary” for our data which provides information on the meaning of our entity classes and corresponding attributes so the uninitiated scholar venturing into the abysmal depths of our data can find some solace in their journey. Once we had an idea of how we wanted to proceed with the relationships between the data with respect to achieving our goals previously mentioned, we derived the logical model of our relational schema from the

conceptual model. The logical model is where we decide on SQL specific criteria for our data. For each table (entity) we must choose an attribute to the foreign key of that table and this key should be unique in its entries (so no duplicate entry). For entity classes that we did not find a solution for, the debug protocol was to introduce an auto increment counter that assigns integers from 1 in descending order to serve as primary key. Finally, we created the create table statements in MySQL to actually form the tables that were prescribed under the logical model complete with constraints and foreign keys that dictated relationships between the tables and had no issues in constructing them. This lab gave allowed us to test our knowledge on classes, attributes and relationships between tables and exercise our introductory MySQL skills to construct a functional (though not yet query functional) database. With a solid base in place we moved on to Lab 2, our most challenging endeavor this semester largely thanks to technical issues.

Lab 2

For Lab 2 we needed to make sure that Lab 1 was fully functional, and we needed to reiterate and shift around some of our entity classes and attributes and echo those changes to the logical model and the create table statements from our base conceptual model. Once we were happy with the organization of the database, we began workin on the core of Lab 2. The learning objectives here, in addition to the Lab 1 objectives, were to actually understand and implement the process of populating our database with data from our two datasets. An additional objective was to gain experience in using a connector in python using PyMySQL to connect to our database from the command line. First, we needed to write import scripts in Python that would populate the database by importing the attributes from our datasets. Our dataset was in csv format so we used the appropriate method of importing the data using python that allowed us to

format the data for our tables. We needed to write an import script for each table in our logical model and run these scripts. We also needed to include the time it took for the script to run, how many entries were imported and if there were any errors that we couldn't solve. These errors should be ones that cannot be debugged unless the raw data was altered, for example a duplicate entry in our primary key attribute (they all need to be unique hence the error). Following that we needed to create "rollback scripts" which would delete all populated data before the import scripts are initiated (in order to avoid duplicate entries in our columns). Finally, we had to write a main script that would call all the rollback and import scripts to populate our database with data from our database. While doing this lab, we had technical issues where we could not connect to the database and actually test our scripts, so it was really incomplete when we turned it in. We realized soon after that that was not the only extent of our issues. We realized that we had issues with the coherency in the data and some entity classes, like the Sales entity class we had could be grouped under the Games entity class due to their one to one relationships. We also had an issue where a lot of our primary keys had duplicate data, and since we could not alter raw data we had to come up with a different way to tackle the issue. We realized soon after that we can introduce entity specific auto incremented primary keys like movie_id for the Movies entity game_id for the Games entity. Since we could not actually test our scripts, it left us with an almost unsurmountable amount of work for Lab 3, but somehow, we managed to pull through and complete it. This lab taught us about time management in tackling the project, and the importance of having a working environment set up before tackling the actual heart of the lab. It also taught us how to debug issues that commonly arise in populating the database, and that more often than not, there are more than one way to solve the problem. We learned a great deal about the standards expected in populating databases (i.e rollback scripts run before import scripts) and

gave us hands on experience in dealing with database connectors and the difficulties that commonly arise when using them. While we were unable to successfully implement this lab, we rebounded back in lab 3 and finally produced something we were proud of.

Lab 3

In Lab 3, we started fixing lab 2 after getting feedback from the TA's. The dataset we chose was harder to work with as it was taken from other sources than what the Professor recommended. The dataset that the Professor recommended was chosen so as to complete the lab3 within the time required. From the early start that is from Lab 1, we tried to choose topics which were interesting to us. The professor recommended addition to our original project was a movie dataset, and we realized we could use this in exciting new ways as we highlighted in the introduction.

After fixing the lab 2 work by finally getting our connection to work on a different device and using the MySQL Workbench software, we started working on Lab 3 where we improved upon each Roll-back scripts& import scripts were written for each table. The create_tables.sql was used to create the database. In a database, 5 tables were created named Games, Movies, Publisher and Reviews. Firstly, we wrote db_connect.py script where it established the connection with the database. Then, populate_database.py script was used to run the import & roll-back scripts. Import & Roll-back scripts were used to extract the data from two csv files, Games.csv & Movies.csv files, and importing it to database. Once all the data was imported, we created & ran the query_interface.py scripts. Query_interface.py script ran he program where user was given chance to interact with database using SQL statement that were hard-coded. User can pick a choice (from 1 to 15) to run an SQL query on database and especially, in first five choices, user had to input a specific value to get their desired results. The queries ranged from all

sorts of interesting perspectives such as connecting data from the games and movies tables where the title were the same hence signifying that the game was made into a movie or vice versa. For security purposes (SQL Injections), user input went through a list which contained common possible SQL injection attempt values and if it matched with one of these values, users are asked to input a valid value instead. Otherwise, there are chances that whole database would be compromised through an injection attack.

Lab 3 was more rigorous and time-consuming than Lab-2. But as a team, we believe Lab 3 was well-spent time and lab created the pace for us to complete more rigorous challenging work in the final project that followed soon after. The hard work we put in this lab allowed us to feel confident in tackling the final project and fix any issues we had in our teamwork.

As a team, we decided not to divide the work, but to work on scripts and SQL statements together. In the last lab we tried to divide the work and lack of communication was a clear barrier to successful completion. Problems arose multiple times because it became harder to translate from the files that the TA's have written as our data required a lot more extra optimization. For example, in the movies dataset that we had, the title of the movie and the date of the movie were together under the column title, and we had to debug it to extract only the title and not the date in order to compare the title with games titles. In working together, mistakes became less frequent as well as debugging became lot faster. Both of us were on the same page. We made right decisions by going through Lab 2 work again rather than moving forward. If we would had started doing Lab 3 work first before Lab 2, it would have become quite difficult to fix lab 2 after wards as all Lab 3 work was dependent on Lab 2.

Problems started showing up in Lab 3 was once we started writing and running query_interface.py script. Few of SQL statement was to be hard coded but few of SQL queries

had a requirement to include input and we felt it became little harder for us. After, working for prolonged time, we came to conclusion that input of SQL queries need to be in a single quotes and whole SQL query needs to be inside double string.

Final Project (Lab 4)

The feedback from the Lab 3 was very much improved from the feedback of Lab 2. There was enough good feedback from Lab 3 that made me and my teammate to move forward to Lab 4. Lab 4 became more interesting than the Lab 3 because we had to use an API (Application Program Interface) (we used twitter API) that was to be connected with one of the table in our database. In our case, Movies table was the best option. The best part of the project was, we had to work with Twitter and had to connect with our database. The requirement was to get at least 50 searches from Twitter that then can be joined with Movies table.

Tweet table was created in database by using SQL query and saved it in the file name, extend_database.sql. api_client.py script was created to connect it with twitter and get the results by running the script and saved it in database. We used Rotten Tomatoes as a source to get the tweets with the tag #RottenTomatoes. Rotten Tomatoes is a movie review site with multiple critics reviewing movies as well as reviewers dishing out their scores for the movie. We believe this search fit the overall theme of our project in that it helps give context when comparing reviews of movies that have been made into games and standalone movies with no game release of their own. Once all the data from twitter was saved it in database, query_interface.py was extended to include more options or choices from the user. The extended choices for the user have SQL queries which uses Movies and tweet table to get specified result. Though no input was added for the user this time. We wanted to maintain quality of the scripts

rather than focus on complexity in order to make a product that was simple yet effective and easy to manage. JSON was used in extended_database.sql and json_queries.sql as the results from Twitter that was saved in database was in JSON and therefore JSON was included in two sql files.

The backup of the database was created by mysqldump modulo through command line in windows and was saved in the backup.sql file.

The problem that arose in Lab 4 was with the api_client.py script. There was an error of 429 that use to come up once tweets search reached the limit. The script usually went to sleep mode for 15 minutes, once 15 minutes passed, the script started to run again. Therefore, we could not analyze the full scope of the data and had to wait till 147 of our results tweets.

Lab 3 had more challenging work than Lab 4 but Lab 4 became more interesting because there was less to write the scripts and more result oriented work. It can be said that, Lab 3 was foundation of the Lab 4 and Lab 3 constant work made Lab 4 work easier. For example, extending interface did not take a long time when compared to lab 3 query_interface.py because the requirements of adding SQL queries in python was already learned in Lab3 and it was just revising and reiterating those same principles in Lab 4.

Conclusion

The whole Elements of Databases is based in its core on the labs. All the concepts that were learned in the class were applied in the final group project. Between the fixing the Labs, as a group, there always had been an opportunity and a reason to revisit the concepts to make the group projects better by improving on them using techniques and debugging lessons we learned from the previous labs. Lab 1 became the foundation for all the Labs. Though there was a chance

of getting whole project sequence rendered unsuccessful if the lab1 wasn't fixed. In our case, we learned that the datasets recommended by the professor is the best in all scenarios as those datasets required less time though they were challenging. But, we chose the datasets which were difficult to find in the website Professor provided. Therefore, it took a little more time than required to complete Lab 1 and Lab 2 task. In Lab 3, we started to give more time towards database class so as to complete the lab 3 in the required time. The time that was invested in Lab 3 resulted in making our final project a success.

From this class, we not only became proficient in database modeling, database application development, and database security, we also learned the importance of teamwork and time management in an agile development scenario. We will conclude by saying that this class has been a worthy endeavor for our constant efforts that bore fruit in the form of satisfaction gained from creating a product(application) that we can really and truly be proud of.