Advanced Database - Final Project Individual Journal

Name: Kushal mundra

Day 1

Meet date/time: 29th March | 11:00am to 4:00pm

Today, Amoga and I devoted approximately 4 hours to brainstorming and initiating our final project. After thorough consideration, we decided to develop a database management system (DBMS) with SQL-like commands. For instance, if a user inputs “create table my_table (id,firstn,lastn,number)", the system will generate a CSV file named my_table.csv, containing the specified column headers.

Personally, I focused on writing the code for the insertData function, as well as developing the splittingStrings and tokenizer functions. The insertData function facilitates data insertion into the table after its creation, allowing users to execute commands like "insert into my_table (1,kushal,mundra,9876543210)", if the table is not present it raises an error saying that the table is not created. Meanwhile, the splittingStrings and tokenizer functions serve to dissect user input strings into distinct words, enabling subsequent operations such as table creation, data insertion, deletion, and updating.

Day 2

Meet date/time : 5th April | 9:00am to 2:00pm

On day 2 of the project, I focused on implementing the delete function. My approach to the delete query was to execute: "delete from <csv_table_name> where <condition_column> = <condition_value>." Upon completing the code for this function, I initially believed that our project was finished. However, upon running the code for testing purposes, I encountered a critical issue. Although the delete operation was executing, it was not removing the corresponding data from the index files. This discrepancy arose because our system employs separate files to store IDs and column headings, utilizing binary sort for retrieval. Consequently, deletions were only affecting the main file, leaving the index files untouched—a problem requiring resolution. Interestingly, Amoga encountered a similar issue while working on the update function. Despite our efforts, we were unable to resolve this issue today. We decided to approach the problem with fresh perspectives starting from the next day.

Day 3
Meet date/time : 11th April | 12:00pm to 6:00pm

On day 3, following a discussion with our professor, Prof. Prajish Prasad, we came to the realization that our indexing approach was overly complex and resource-intensive. After deliberation, Amoga and I decided to switch to using B-trees for indexing. Implementing the B-tree class proved to be straightforward and efficient, so we promptly integrated it into our system. I focused on coding the search and delete operations for the B-tree, successfully implementing them along with several auxiliary functions to support their functionality. Additionally, I devised a method for importing files into our DBMS, although I didn't have the opportunity to implement it today. Overall, it was a highly productive day during which we completed a significant portion of our work and gained clarity on the direction for developing our DBMS.

Day 4
Meet date/time : 13th April | 5:00pm to 9:00pm

On day 4, with our B-tree successfully implemented, we shifted our focus to integrating all CRUD (Create, Read, Update, Delete) operations into our system. This involved incorporating SQL queries from our previous codebase for functionalities such as update, delete, search, and insert. I specifically worked on refining the delete, search, import, and select functions, ensuring their seamless integration into our DBMS. We collaborated on defining the structure of the queries and how they should be formatted. Upon running the code, users are prompted with the available queries they can input.

As mentioned earlier, I also completed the implementation of the import function, which facilitates the importation of any CSV file into our program. This feature automatically generates B-trees for the columns, simplifying subsequent CRUD operations. Additionally, we developed a display function to showcase all changes made within the system. Furthermore, we added functionality to print column headings every time, eliminating the need to repeatedly enter ".schema".

Overall, we successfully accomplished all our objectives for the day and conducted thorough testing to ensure the proper functioning of our DBMS.