PROJECT REPORT(PHASE-1)

GROUP – 4
Neelkanth Dabhi [B00848532]
Samkit Shah [B00852292]
Prashant Sarvi [B00870599]

Table of Contents

Table of Figures

1. Features of DBMS

Database Management System provides various features along with storing and manipulating data. Few features of DBMS are the following:

- The database maintains data integrity by recording it accurately as observed during the transaction and prevent the loss of sensitive information.
- Enhanced data access and convenience to perform actions on data like read, update, delete, etc.
- Improved data security by giving access to specific authorized users and storing sensitive information in an encoded format.
- Improved data security by giving access to specific authorized users and storing sensitive information in an encoded format.
- Improves end-user efficiency by using tools that transform data into a visual representation and helps the client take quick decisions.

2. Core Components

1. Query Parsing

Checks if any table name class exists or not in the database and after all the validations it generates all the possible relational algebra expression as you can see in the example, and it will forward these expressions to the query optimizer.

2. Query Optimization

As the name itself suggest the query optimizer's job is to choose the leader costly relational algebra expression and convert that into an execution plan. Database statistics help the optimizer to make decisions based on the data.

3. Query Evaluation

It operates on the execution plan provided by the previous phase. It continuously executes all the commands on each row until all the rows are processed. After evaluating instruction on all the rows, it generates the output table and sends it to the user. If there are no records in the output table, it will send still send the rows affected.

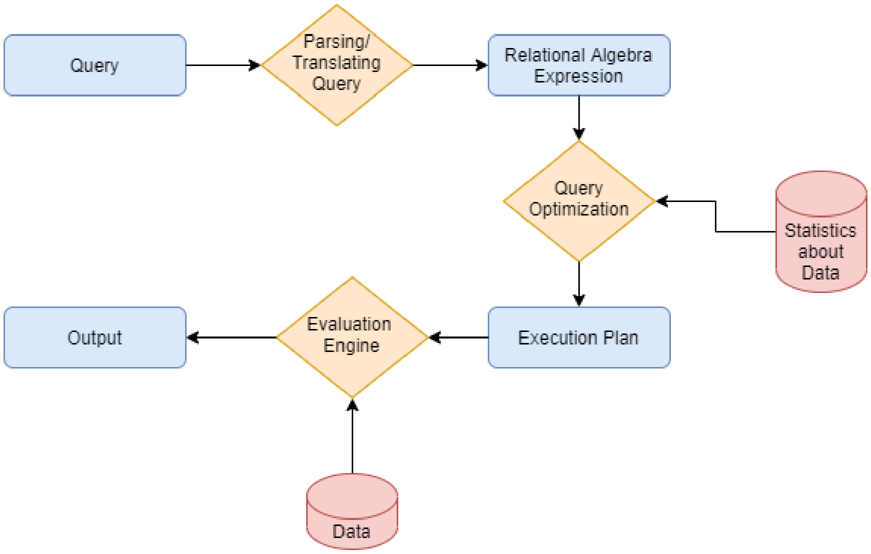


Figure 1 Query Processing

3. Data Structures

Data structures plays an important role in whole DBMS system.The data structure will be mainly used to process the query, store and retrieve data.Data structures will provide an easy and efficient way to store, modify and retrieve the data. We are going to use mainly three types of data structure to perform the tasks.

1. Parse Tree
   - Parse tree will mainly use to validate and optimize the query. The query entered by the user will be translated into an internal representation which is easy to evaluate.
   - Parser will take input as a query string and generate the parse tree. Query could be combination of two or more query, query can contain multiple function, query can have multiple different attributes. So, we need a mechanism which is easy to scale and represent the parent-child relationship for subquery.
2. B+ Tree
   - B – Tree is mainly used to store the data what we have retrieved from the memory.
   - B – Tree is an efficient data structure for the insertion, deletion and lookup with O(log(n)) complexity.
   - Each data is stored in a persistent form on memory. The data is fetched from the memory to database management system and stored in the form of B – Tree.
   - Each data row is associated with the unique index and all the index are parent node of child. Leaf node contains the actual data.
3. Hash Map
   - Hash map stores the data in the key – value form.
   - Actual data is stored in the JSON form on memory. There will be one individual JSON file for single table. The physical location of JSON file is stored in the value and key is a table name.
   - The main benefit of Hash map is that it has O (1) time complexity for put and get operation.
   - In this scenario we just want to do insert and retrieve the entry.
4. Stack
   - Stack is very efficient to insert the data and fetching data from one end.
   - In DBMS system, stack will be used for transaction purpose.
   - When transaction rollback then we need to undo all the operations performed in the particular transaction. So, we need to keep track of operations in the ordered manner.
   - Stack has O (1) time complexity for push and pop operation. It will work in the LIFO manner.

1. Data Storage

Database Management System contains multiple databases, one database contains the multiple table and one table contains multiple rows. We need to store all the data in a persistent way on memory. So, we are going to use JSON format to store the data. There will be one JSON file per table and all the data are stores in the that particular json file. We are using JSON format because it is easy to read and write. Along with that json is language independent.so, we can fetch particular json file into DBMS system and maintain it as a JSON object.

4. Tools and Technologies

There are various programming languages and platform available. We have researched the feasibility for different platforms. After that, we concluded that there are mainly two programming language java and python are feasible. Because it has very good community support and platform independent. But as we need to implement various data structures and develop through scratch then java would be the best option for data structure implementation. Apart from this, Java is object-oriented programming language which can provide us Abstraction, Inheritance, and encapsulation. These all will support to follow ACID properties. After considering all, we decided to consider JAVA as a programming language.

5. Uniqueness

Database replication is a method to preserve data in multiple databases. So, this helps all users share the same level of information and it leads to distributed databases which is helpful in the following ways:

- It provides latency to interrogate or verify the data before it lands in the form of reports.
- Business personnel need not rely on a single database and face issue like downtime.
- It is also helpful in secondary and disaster management techniques.

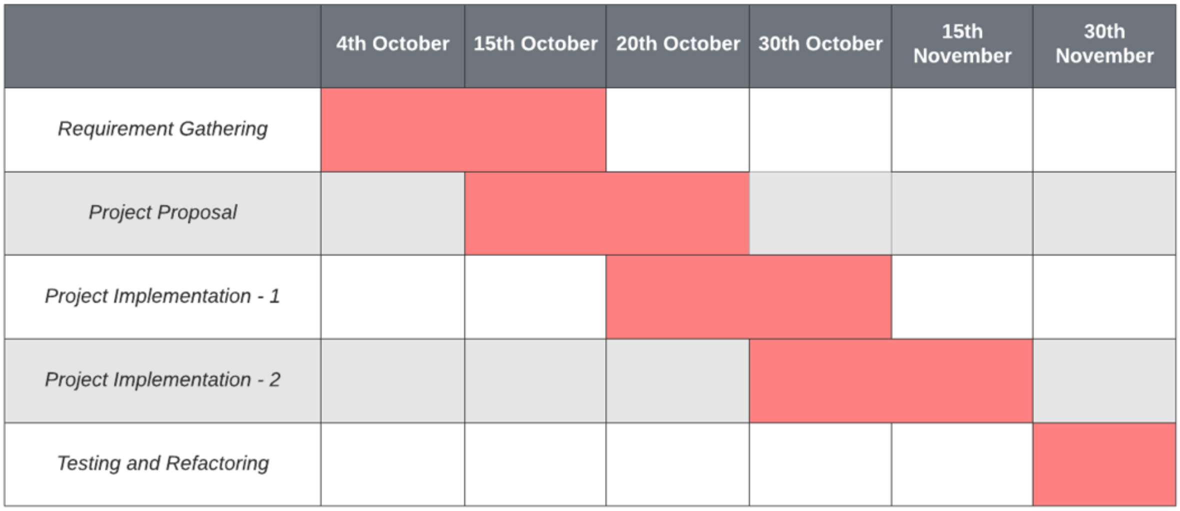
6. Challenges & Solutions

- **Challenge**: To build a parser. It can be prone to errors and must validate the defined syntax properly.
  Solution: To thoroughly understand the phases of compiler and using ANTLR[1]
  which is a powerful parser generator for reading, processing, executing, or translating
  structured text or binary files.

- **Challenge:** Performance. Storing the entire data on a single disk could lead to reduced rate of processed information, degrading the performance.
  Solution: Fragmentation of data which are stored long-time ago and aggregating the relevant data will result in faster I/O related operations.
- **Challenge**: Data Security. The information stored in the database is vulnerable if built-in security is not provided.
  Solution: Passwords and other confidential information can be encrypted and stored in the form of hash values.

7. Project Timeline

The gathering of the requirement was started on 4th October 2020. Each team member had divided the work for aggregation of requirement into equal parts, prepared a proper structure of the requirements in the form of power point presentation and uniformly contributed to the formation of final report. We are optimistic that we will be able to complete testing and refactoring by 30th November 2020.



	4th October	15th October	20th October	30th October	15th November	30th November
Requirement Gathering	🟥	🟥				
Project Proposal		🟥	🟥			
Project Implementation - 1			🟥	🟥		
Project Implementation - 2				🟥	🟥	
Testing and Refactoring						🟥

Figure 2 Gantt chart

8. References

[1] Antlr.org. 2020. *ANTLR*. [online] Available at: https://www.antlr.org/[Accessed 20 October 2020].