Application on Analysis and Query Firing in Mini Database Engine Based on Relational Algebra

Anoop Singh
M.Tech-Computer Science
Dept.’ Of CSE, CCSIT,
TMU Moradabad, Uttar Pradesh India

Nitin Kr. Verma
Course Co-ordinator
Dept’t of CSE, CCSIT
TMU Moradabad, Uttar Pradesh India

Abstract—The purpose of this application is analysis the query execution time and process the query parsing based on Relational Algebra. This database Engine is coded in java language which is platform independent. And it will also support Android operating System. It will take less amount of space on disk. This database engine will support the maximum number of data types like int, char, date, varchar, boolean, etc. It will take less execution time to process SQL Queries. In this we can see the all process of execution and decomposition all query evaluation process, and parsing by parser. Many popular database engines like Oracle, DB2, Apache Derby etc do not support Android Operating System. Maximum number of popular database engines take large amount of space on disk therefore small resources contained device can not use these as backend. Some data types like BOOLEAN, etc do not supported by database engines. The execution time for SQL queries taken maximum by heavy database engine.

Keywords—Parsing, Decomposition, SQL, Relational Algebra, Query, DBMS.

I. INTRODUCTION

A. Motivation

Database management systems have emerge as a fairly standard part of any company back office. Typical examples of DBMS include accounting, human resource, customer support system etc. So, we developed an application to manage a database that supports the rules of relational model in DBMS. This motivated me to develop lightweight, easy to use application for database management.

B. Problem Statement

A lot of database engines are available in the market like ORACLE, MySQL, Sybase, IBM DB2 and many more. I decided to develop a lightweight, easy to use application in comparison to these heavy applications. It provides user a platform to execute simple SQL queries on command prompt and determine RELATIONAL ALGEBRA conversion of its SQL queries. For this, we first need to have a parser that will check and validate the grammar of query and verifies the fields and expressions defined in the query. Then i need to process these tokens and project result on command prompt.

II. OBJECTIVE

The Development of a Database engine, based on the relational model that implements its own database storage format and supports SQL statements. The Database engine will be developed in Java. The Engine has a Command interpreter interface for the user. All SQL statements are entered via a command interpreter. I analysis execution time to process SQL Queries. In this i can see the all process of execution and decomposition to all query evaluation process and pointing to parsing query and Maintaining the Integrity of the Specifications.

III. EXHAUSTIVE LITERATURE SURVEY

Introduction

A Database is a logically coherent collection of data with some inherent meaning. A database is designed, built, and populated with data for a specific purpose. A database management system (DBMS) is a collection of programs that enables users to create and maintain a database. The DBMS is hence a general purpose software system that facilitates the processor of defining, constructing, and manipulating database for various applications. A Database Engine is the software in a DBMS that parses and executes SQL statements and accesses the physical data. It includes centralized data management, data independence, and systems integration. Here all access to the data is through the DBMS providing a key to effective data processing.

It is responsible for storing, retrieving, and updating the data. A program which performs the database management function on data held in a database. This is to say it can manipulate, add, delete, search, select and store data following simple users commands.
A. Why does use database?

There are three main features of a database management system that make it attractive to use a DBMS in preference to a more conventional software. It includes

- Centralized data management,
- Data independence, and
- System integration.

Here all access to the data is through the DBMS providing a key to effective data processing. The database management system provides the interface between the application programs and the data.

B. The Relational Model

This model was formally introduced by “Dr. E. F. Codd” in 1970 and has evolved since then. It represents data in the form of two-dimension tables. Each table represents some real world entity. Most popular commercial and open source databases currently in use are based on the relational model.

C. Introduction to SQL

SQL (Structured Query Language) is a database computer language designed for the retrieval and management of data in relational database management systems (RDBMS), database schema creation and modification, and database object access control management.

SQL is a standard interactive and programming language for querying and modifying data and managing databases. Although SQL is both an ANSI and an ISO standard, many database products support SQL with proprietary extensions to the standard language. The core of SQL is formed by a command language that allows the retrieval, insertion, updating, and deletion of data, and performing management and administrative functions. SQL also includes a Call Level Interface (SQL/CLI) for accessing and managing data and databases remotely.

There are also some other features that set SQL apart from those that merely implement features that are part of relational algebra or calculus.

IV. RESEARCH TOOL AND METHODOLOGY

A Database Engine is the software in a DBMS that parses and executes SQL statements and accesses the physical data. It includes centralized data management, data independence, and systems integration. Here all access to the data is through the DBMS providing a key to effective data processing.

These activities are performed in order as shown below:

A. SQL Query Parser and Decomposer

SQL, the Structured Query Language, is a mature, powerful, and versatile relational query language. It is based largely on relational algebra and tuple relational calculus. Relational algebra, introduced by E. F. Codd in 1972, provided the basics concepts behind computing SQL syntax[1].

B. Java JDK and BlueJ

I have used JAVA as the programming language for the development of my application. It must have minimum J2SE v1.4 (JDK 1.4) or later installed on system. Here BlueJ is used as programming environment. BlueJ is a Java™ development environment specifically designed for teaching at an introductory level. It was designed and implemented by the BlueJ team at Deakin University, Melbourne, Australia, and the University of Kent at Canterbury, UK.
C. Query Parser
- Used a Class like ZqlParser.java.
  ➢ Analyses query with rule of grammar.
  ➢ Validates attributes and relations
- Application
  ➢ Scans a SQL query.
  ➢ Validates its Syntax.

Performs Syntax checking and verification. Parses query into token and analyses them with rule of grammar. It checks whether attributes and relations are defined in the database[7].

Thus it involves:-
- Scanning
- Parsing
- Validating

For this purpose, we need to develop a pushdown automata. It will scan the query, and check the grammar of the query with the grammar defined with database of PDA[15].

![Pushdown Automata for Query Parsing](image)

- **Firstly**, it will scan the syntax of the grammar.
- **Secondly**, it will determine the operation that is to be performed i.e. select, delete, insert or update

Followings are some typical usages of parse tree.

![Typical usage of Parse Tree](image)

D. Query Decomposer
- Used some Classes like ZQuery.java, ZInsert.java, ZUpdate.java and ZDelete.java.
- Decomposes query into tokens.

E. Query Processor
- Used a Class like ZExec.java and process the tokens.
- Project the output on command line

F. Implemented Result
- Execution flow of Query Parser, Query Decomposer and Query Processor is given below-

![Execution working flow of all respective java classes module for required result](image)
A. Summary
The aim was to built a Database Engine based on Relational Model. But, i have to reduce to my work and i finally come with an application that processes quiet large number SQL queries like SELECT, INSERT, UPDATE, DELETE and evaluation of some complicated expressions on single table. The application is built in JAVA and i have used BlueJ as development environment. I have used ZQL parser and modified its code as per MY project requirements.

B. Application
- Persistent storage of data.
- Controlled redundancy.
- Support queries of SELECT, DELETE, INSERT and UPDATE.
- Representation of some complex expressions like AND, OR, NOT, BETWEEN operators on Single table.

REFERENCES


