

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA , SURATHKAL

An Individual Report on

**PARSER ON IMPLEMENTATION OF RELATIONAL ALGEBRA**

**Submitted** by

**Rahul Kumar Bhojwani - 11CO73**

*Under the guidance of*

**Dr. Ravindranath C**

*August-November, 2013*

**CONTENTS**

1. Introduction and Problem Statement
2. An overview of Relational Algebra
3. Design and Implementation
4. Acknowledgements
5. Problems Faced During implementation
6. Assumptions
7. Screenshots

***INTRODUCTION And PROBLEM STATEMENT***

**PROBLEM STATEMENT :**

*Program to implement all the operations of relation algebra.*

***An overview of Relational Algebra***

The basic set of operations for the relational model is the **relational algebra**. These operations enable a user to specify basic retrieval requests as *relational algebra expressions*. The result of a retrieval is a new relation, which may have been formed from one or more relations. The algebra operations thus produce new relations, which can be further manipulated using operations of the same algebra. A sequence of relational algebra operations forms a **relational algebra expression**, whose result will also be a relation that represents the result of a database query (or retrieval request).

The relational algebra is very important for several reasons.

* First, it provides a formal foundation for relational model operations.
* Second, and perhaps more important, it is used as a basis for implementing and optimizing queries in the query processing and optimization modules that are integral parts of relational database management systems (RDBMSs).
* Third, some of its concepts are incorporated into the SQL standard query language for RDBMSs.

There are many operations in relational algebra, like

* Select
* Project
* Rename
* Union
* Intersection
* Difference
* Natural Join
* Theta Join
* Division Operator

Operators which are successfully implemented here, using c++ are:

Select , Project , Theta join , Natural join , Union.

***Design and implementation***

Idea behind the functions

1. **Select :**

Here the idea is that we take the input condition and the name of the relation from the user and substitute in the attributes present in the input condition with corresponding value present in every tupple of the relation stated, and evaluate the logical expression. If this line evaluates to true, only then the line is printed.

The syntax is given in the help documentation of the program.

1. **Project :**

Here the idea is to take the relation and the set of attribute names to be projected as arguments in the syntax specified in the help documentation, and then it compares the name of the attribute in the file with those specified and if they match only then it is shown.

1. **Rename :**

This is nothing but renaming a file and then using it.

1. **Union, Difference and Intersection :**

This simulates the required operations of the set theory based on the concept of union compatibility and only then executes it else generates an error.

1. **Theta Join, Cartesian Product and Natural Join :**

This simulates the various join operations, and joins the attributes. This can include any number of relations as the input, and is efficient in storing the result in a new relation, but delay occurs in displaying the result.

**PROBLEMS FACED:**

* It is seen that though this method gives extremely fast results for fairly small inputs,as the number of tupples increases, generally beyond 3000 or so, the displaying stops , because of increasing burden over the RAM.The problem is with the displaying.
* Debugging a long code.
* Implementing nested Queries and Nested conditions.
* A fourth problem is with regard to the use of aggregate functions.

**Assumptions:**

1 The input is always in the format given in readme.

2 The file name that is given is already stored in the proper location.

**For a file:**

1 First line contains the no. of columns

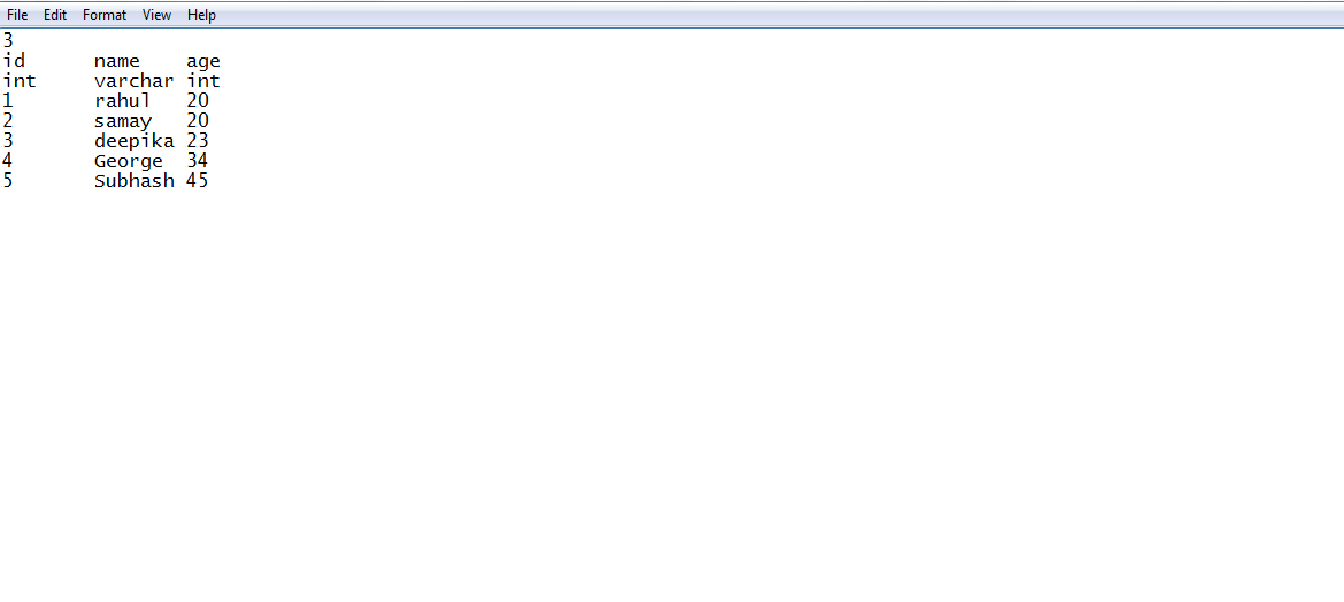
2 Second line contains the data type of those columns

3 Remaining each line represent one tuple.

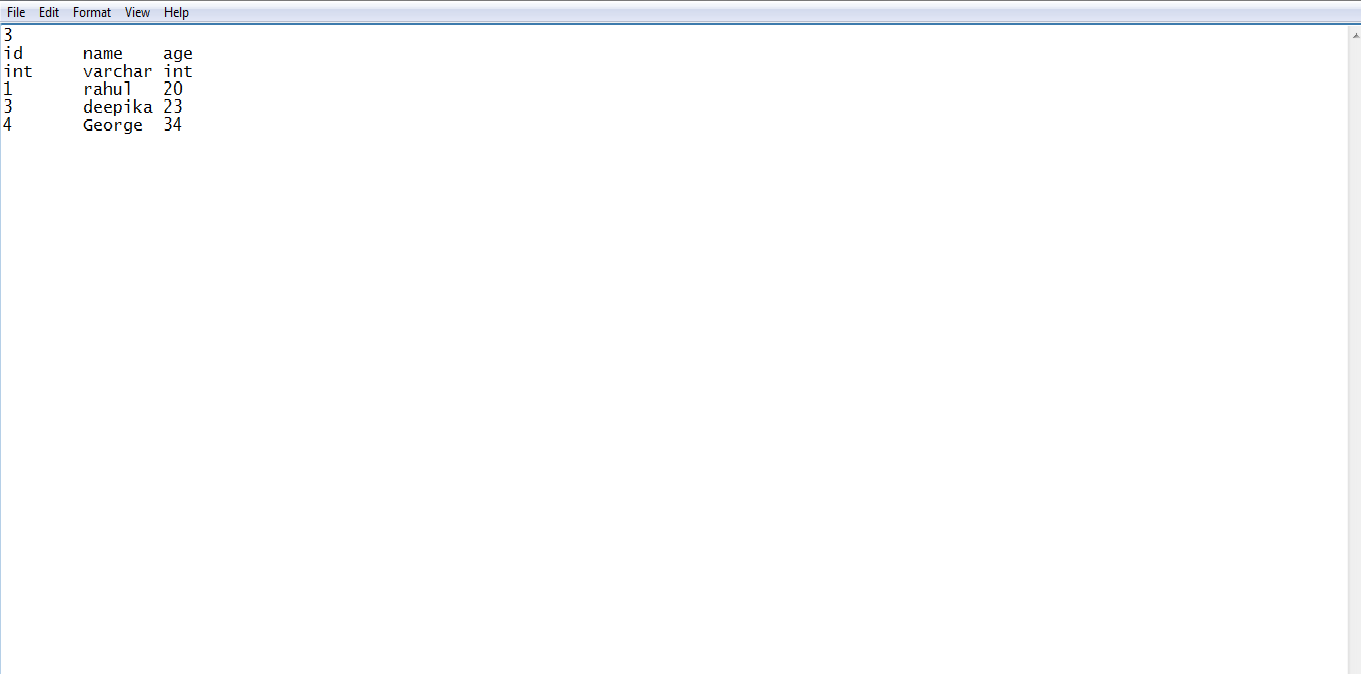
**Screenshots:**

**Query: A = ( Select < ( ( [ name = rahul ] OR [ age = 23 ] ) OR [ id = 4 ] ) > data.txt )**

**Data.txt :**

****

A.txt

****

**ACKNOWLEDGEMENT**

I hereby gratefully acknowledge the immense support and inspiration provided by our course instructor, Dr. Ravindranath Choudhary, Department of Computer Science and Engineering, NITK Surathkal which was instrumental in realizing this project. I would even like to express my sincere gratitude to him for his insightful advice, encouragement, guidance, critics and valuable suggestions throughout the course of this project work.