

Relation Algebra Parser

Ayush Shrivastava

14075014

B.Tech. CSE 3rd Year

Relational Algebra consists of six basic operations:

1. Select σ
2. Project π
3. Rename ρ
4. Union \cup
5. Set Difference $-$
6. Cartesian Product \times

Rest of the operations can be derived from these basic operations.

Setup:

Copy the whole folder. Open terminal inside folder and run

```
g++ dbms.cpp -o out && ./out
```

In RA parser, the table names are allowed to be alphanumeric and attributes are allowed to be alphabets only. No table can have more than one attribute of same name. Also, Tables are implemented a Set, so duplicate records are not supported.

This RA parser supports following data types:

1. Integer
2. Double
3. Char
4. String
5. Bool

- All queries should be given in balanced parenthesis form, else they result in error.

- Select may be used in place of SELECT as this is not case-sensitive. “select”, “sElecT” are also supported.

Same is for PROJECT, RENAME, UNION, CART, SDIFF.

1. SELECT Operation:

The basic structure of SELECT query is

SELECT { Predicate } [Table]

Select may also be used in place of SELECT as this is not case-sensitive.

Predicate has to be specified in Curly Brackets { } and Table has to be specified in Square Brackets []. Whole query should be in Balanced Parenthesis form.

The Predicate part of SELECT operation supports following operations on the above specified data types:

+ - * / < > >= <= == != && || ! (NOT for Bool datatype)

= can also be used in place of ==.

| can also be used in place of ||.

& can also be used in place of &&.

Predicate has to be specified in Infix form and can use only Parenthesis () .

Example Predicates are as follows:

(Age>20 && Name!=”Ayush”) || (Salary/2 > 50000 && isCustomer==True)

Here isCustomer is of type BOOL.

Operations may be performed elements of same data type only.

However, in case of INT and DOUBLE, they may be operated with each other interchangeably.

(Salary > 200.45 && Sex=’F’)|| (3*Age + 30 < 90) || (Salary=SalaryPrevious)

Here, Salary is in INT but may be compared with DOUBLE. Two Variables such as Salary & SalaryPrevious are allowed to be compared.

Sample query for SELECT operation:

SELECT { Age>20 && Sex!=’M’ } [Student]

2. PROJECT Operation:

PROJECT {Attr1, Attr2, Attr3, ... } [Table]

{ } should enclose the attributes and attributes should be separated by comma ‘,’. Table should be provided in []. Projection of attributes not existent in Table will result in error.

3. RENAME Operation:

RENAME {newTableName} [Table]

RENAME {newTableName | Attr1,Attr2,Attr3, ... } [Table]

4. UNION Operation:

UNION [TableA , TableB]

Tables should be separated by a comma and tables should be enclosed in Square Brackets.

Tables should have exactly same schema i.e., names and datatypes of attributes and the order in which they are given should be same, otherwise it results in error.

5. CART Operation:

Cartesian Product has been abbreviated to CART.

CART [TableA , TableB]

TableA and TableB should have completely disjoint schemas. If an user wants to product two tables with same name of attributes, then he/she must rename atleast one table to make this condition satisfy.

6. SDIFF Operation:

Set Difference has been abbreviated as SDIFF.

SDIFF [TableA, TableB]

This returns TableA – TableB.

Few Examples of Nested Queries:

Brackets have to be taken care of in Nested Queries.

```
CART [ SELECT{predicate}[TableA] , PROJECT{Attr1, Attr2, ... }[UNION [TableB, TableC] ] ]
```