**ST. XAVIER’S COLLEGE**

**Maitighar, Kathmandu**



DATABASE MANAGEMENT SYSTEM

theory Assignment #6

**Submitted by:**

Narayan Panthi

013BSCCSIT024

**Submitted to:**

Er. Sanjay Kr. Yadav

Lecturer, St. Xavier’s college

Department of Computer Science

**Date of Submission:**

August 10, 2015

**RELATIONAL ALGEBRA**

Relational algebra is a procedural query language, which takes instances of relations as input and yields instances of relations as output. It uses operators to perform queries. An operator can be either **unary** or **binary**. They accept relations as their input and yield relations as their output. Relational algebra is performed recursively on a relation and intermediate results are also considered relations.

The fundamental operations of relational algebra are as follows −

* Select
* Project
* Union
* Set different
* Cartesian product
* Rename

# Joins

**Join** is a combination of a Cartesian product followed by a selection process. A Join operation pairs two tuples from different relations, if and only if a given join condition is satisfied.

Join is used to fetch data from two or more tables, which is joined to appear as single set of data. SQL Join is used for combining column from two or more tables by using values common to both tables. Join Keyword is used in SQL queries for joining two or more tables. Minimum required condition for joining table, is (n-1) where n, is number of tables. A table can also join to itself known as, Self Join. Join are of two types:

* Theta Join
* Natural Join

# Theta (θ) join

Theta R1 ⋈θ R2

R1 and R2 are relations having attributes (A1, A2, .., An) and (B1, B2,.. ,Bn) such that the attributes don’t have anything in common, that is R1 ∩ R2 = Φ.

Theta join can use all kinds of comparison operators.

# Natural Join (⋈)

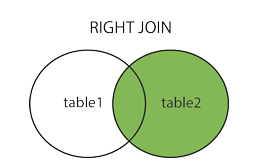
This is the most common and general form of join. If we simply say join, it means the natural join. It is same as equi­join but the difference is that in natural join, the common attribute appears only once. Now, it does not matter which common attribute should be part of the output relation as the values in both are same. Natural join does not use any comparison operator. It does not concatenate the way a Cartesian product does. We can perform a Natural Join only if there is at least one common attribute that exists between two relations. In addition, the attributes must have the same name and domain.

# Right Join

The RIGHT JOIN keyword returns all rows from the right table (table2), with the matching rows in the left table (table1). The result is NULL in the left side when there is no match.

Syntax

SELECT *column\_name(s)*  
FROM *table1*   
RightJOIN *table2*  
ON *table1.column\_name*=*table2.column\_name*;



# Left Join

The LEFT JOIN keyword returns all rows from the left table (table1), with the matching rows in the right table (table2). The result is NULL in the right side when there is no match.

Syntax

SELECT *column\_name(s)*  
FROM *table1*  
LEFT JOIN *table2*

ON *table1.column\_name*=*table2.column\_name*;



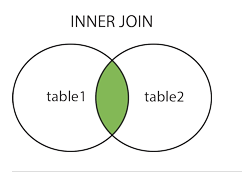
# Inner Join

The INNER JOIN keyword selects all rows from both tables as long as there is a match between the columns in both tables.

Syntax

SELECT *column\_name(s)*  
FROM *table1*  
INNER JOIN *table2*

ON *table1.column\_name*=*table2.column\_name*;



**RENAME OPERATION (Ρ)**

The results of relational algebra are also relations but without any name. The rename operation allows us to rename the output relation. 'rename' operation is denoted with small Greek letter rho ρ.

Notation − ρ x (E)

Where the result of expression E is saved with name of x.

E: relational algebra expression

ρ x (E): returns the result of expression E under the name x.

ρ x (A1, A2, A3… An) (E): returns the result of expression E under the name x with attributes renamed to A1, A2, A3… An.

**ASSIGNMENT OPERATOR**

Particularly with division, that relational algebra feels a lot like programming: there are many steps to some expressions, with intermediate or temporary relations along the way. For this very reason, we have the assignment operation, which works a lot like assignments in a programming language. It is notated with the left-pointing arrow ←:

variable ← E

where E is any relational algebra expression.

• The assignment operation is more of a notational convenience rather than a real relational

operation — it helps human beings with writing out complex relational expressions

in steps so that they can be more easily understood.

• Revisiting the breakdown of the division operation, we can use assignment to rewrite this way:

temp1 ← ΠR−S(r)

temp2 ← ΠR−S((temp1 × s) − ΠR−S,S(r))

r ÷ s = temp1 − temp2

**THE DIVISION OPERATION**

Let r(R) and s(S) be relations **r ÷ s: -** the result consists of the restrictions of tuples in r to the attribute names unique to R, i.e. in the Header of r but not in the Header of s, for which it holds that all their combinations with tuples in s are present in r.  
  
Example:

|  |  |  |
| --- | --- | --- |
| Relation or table "r":- | Relation or table "s":- | Therefore, r ÷ s |
| Code: | Code: | Code: |
| | A | B |  \_\_\_\_\_\_\_\_\_  | a | 1 |  | b | 2 |  | a | 2 |  | p | 3 |  | p | 4 | | | B |  \_\_\_\_  | 2 |  | 3 |  \_\_\_\_ | | A |  \_\_\_\_  | b |  | a |  | p |  \_\_\_\_ |

**ADDITIONAL OPERATION**

• “Additional operations” refer to relational algebra operations that can be expressed in terms of the fundamentals — select, project, union, set-difference, cartesian-product, and rename.

• The compositions of these operations are so lengthy, yet so common, that we define new operations for them, based on the fundamentals. Kind of a mathematical “syntactic sugar.”

**SET-INTERSECTION**

The set-intersection operation is a binary operation on relations r and s that is denoted by the traditional intersection symbol, ∩. r ∩ s results in all tuples t such that (t ∈ r) ∧ (t ∈ s). 1

Set-intersection is defined in terms of set-difference: r ∩ s = r − (r − s)

Thus, set-intersection must follow the same compatibility rules as set-difference: same arity, corresponding domains.

**NATURAL JOIN**

It is a binary operation and a combination of certain selections and a Cartesian product into one operation.

* It is denoted as |X| .
* It is associative.

It forms a Cartesian product of its two arguments.  
Then performs a selection forcing equality on those attributes those appear in both the relations.  
And finally removes duplicates attributes.  
  
r(R): r is a relation with attributes R.  
s(S): s is a relation with attributes S.  
  
If R **∩**S = Ф i.e. they have no attributes in common then **r |X| s = r X s.**

**REFERENCE :**

<http://searchsqlserver.techtarget.com/definition/relational-database-management-system>

<http://cisnet.baruch.cuny.edu/holowczak/classes/3400/relationalalgebra/>