**ST. XAVIER’S COLLEGE**

**(Affiliated to Tribhuvan University)**

Maitighar, Kathmandu



**Database Management System**

**Assignment # 6**

**Submitted by:**

Anu Kadel  
013BSCCSIT007

**Submitted to:**

|  |  |
| --- | --- |
| **Er. Sanjay Kumar Yadav**  Lecturer  St. Xavier’s College |  |

**Date of Submission:** September 13, 2015

**LAB ASSIGNMENT #6**

**JOIN**

**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.

The join operation forms a Cartesian product of its two arguments, performs a selection forcing equality on those attributes that appear in both relation schemas, and finally removes duplicate attributes

**THETA JOIN**

Theta join combines tuples from different relations provided they satisfy the theta condition. The join condition is denoted by the symbol **θ**.

A theta-join is any Cartesian product that's filtered by a condition which compares values from both Tables. That is, the general theta-join form is:

<Table\_1.Column> relator <Table\_2.Column>

where the relator is almost always "=", as in this example:

Sellers.seller\_name = Sales.seller\_name

This special case of theta-join — where the relation is equality — is called an equijoin.

**NATURAL JOIN**

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.

A natural join is a joint operation that creates an implicit join clause for you based on the common columns in the two tables being joined. Common columns are columns that have the same name in both tables. A natural join can be an inner join, a left outer join, or a right outer join. The default is inner join.

If the SELECT statement in which the NATURAL JOIN operation appears has an asterisk (\*) in the select list, the asterisk will be expanded to the following list of columns (in this order):

* All the common columns
* Every column in the first (left) table that is not a common column
* Every column in the second (right) table that is not a common column

An asterisk qualified by a table name (for example, COUNTRIES.\*) will be expanded to every column of that table that is not a common column.

**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.

SELECT column\_name(s)  
FROM table1  
RIGHT JOIN 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.

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.

SELECT column\_name(s)  
FROM table1  
INNER JOIN table2  
ON table1.column\_name=table2.column\_name;

[3]

**RENAME OPERATION:**

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**.

The REMANE operator returns an existing relation under a new name. This is a Mechanism used for renaming both relations and attributes, this renaming is highly useful when one has to compare a set of tuples of the same relation with other tuples of that relation. It has the following general format :

old\_name as new-name

**ASSIGNMENT OPERATION:**

The assignment operator is one of the most intuitive to use. It assigns a value to a variable. The only confusion in using this operator could stem from its overloading. All RDBMS overload this operator with an additional function — comparison — in the SQL.

SET Variable Name = expression

Variable name can be any valid variable created in sql server. Assignment operator sets value to variable returned by expression.

SELECT FieldName = expression

Field name can be any name. Assignment operator sets value to variable returned by expression.

**DIVISION OPERATION:**

The division operator is the most difficult to implement in SQL as no SQL command is given for division operation. The division operator would be seen as the opposite of the Cartesian product operator; just as in standard math, the relation between division and multiplication. Therefore a series of current SQL commands have to be utilized in implementation of the division operator. An example of the SQL implementation of division operator:

Letr(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.

**ADDITIONAL OPERATION:**

**1.Set Intersection Operation:**

The set intersection operation finds tuples in both the relations. It is denoted as ∩.

**2.Natural Join Operation:**

Natural join (\bowtie) is a binary operator that is written as (*R* \bowtie*S*) where *R* and *S* are relations. The result of the natural join is the set of all combinations of tuples in *R* and *S* that are equal on their common attribute names. For an example consider the tables *Employee* and *Dept* and their natural join:

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

**REFERENCES:**

**[1]** [Relational Algebra System for Oracle and Microsoft SQL Server](http://www.cse.fau.edu/%7Emarty#RADownload)

**[2]** <https://en.wikipedia.org/wiki/Rename_%28relational_algebra%29>

**[3]**<http://www.w3schools.com/sql/sql_join_right.asp>

**[4]** <http://www.tutorialspoint.com/sql_certificate/using_the_set_operators.htm>

**[5]** <https://msdn.microsoft.com/en-us/library/ms188343.aspx>