**ST. XAVIER’S COLLEGE**

**(Affiliated to Tribhuvan University)**

Maitighar, Kathmandu



**Database Management System Lab Assignment #6z**

**Submitted by:**

Bishal Pandey

013BSCCSIT016

**Submitted to:**

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

**Date of Submission:** September 10, 2015

**Join**

Join is a special form of cross product of two tables. It is a binary operation that allows combining certain selections and a Cartesian product into one operation. 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**

In theta join we apply the condition on input relation(s) and then only those selected rows are used in the cross product to be merged and included in the output. It means that in normal cross product all the rows of one relation are mapped/merged with all the rows of second relation, but here only selected rows of a relation are made cross product with second relation. It is denoted as under: -RX S

* **Natural join**

A natural join is a join 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

A RIGHT OUTER JOIN

A right outer join is one of the [join operations](https://docs.oracle.com/javadb/10.8.3.0/ref/rrefsqlj29840.html" \l "rrefsqlj29840) that allow you to specify a join clause. it preserves the unmatched rows from the second (right) table, joining them with a null in the shape of the first (left) table. a left outer join b is equivalent to b right outer join a, with the columns in a different order.

**Syntax**

[***TableExpression***](https://docs.oracle.com/javadb/10.8.3.0/ref/rreftableexpression.html#rreftableexpression) **RIGHT [ OUTER ] JOIN** [***TableExpression***](https://docs.oracle.com/javadb/10.8.3.0/ref/rreftableexpression.html#rreftableexpression)

**{**

**ON *booleanExpression* |**

[***USING clause***](https://docs.oracle.com/javadb/10.8.3.0/ref/rrefsqljusing.html#rrefsqljusing)

**}**

# LEFT OUTER JOIN operation

A left outer join is one of the [join operations](https://docs.oracle.com/javadb/10.8.3.0/ref/rrefsqlj29840.html" \l "rrefsqlj29840) that allow you to specify a join clause. it preserves the unmatched rows from the first (left) table, joining them with a null row in the shape of the second (right) table.

## Syntax

[***TableExpression***](https://docs.oracle.com/javadb/10.8.3.0/ref/rreftableexpression.html#rreftableexpression) **LEFT [ OUTER ] JOIN** [***TableExpression***](https://docs.oracle.com/javadb/10.8.3.0/ref/rreftableexpression.html#rreftableexpression)

**{**

**ON booleanExpression |**

[***USING clause***](https://docs.oracle.com/javadb/10.8.3.0/ref/rrefsqljusing.html#rrefsqljusing)

**}**

# INNER JOIN operation

An INNER JOIN is a [JOIN operation](https://docs.oracle.com/javadb/10.8.3.0/ref/rrefsqlj29840.html#rrefsqlj29840) that allows you to specify an explicit join clause.

## Syntax

[***TableExpression***](https://docs.oracle.com/javadb/10.8.3.0/ref/rreftableexpression.html#rreftableexpression) **[ INNER ] JOIN** [***TableExpression***](https://docs.oracle.com/javadb/10.8.3.0/ref/rreftableexpression.html#rreftableexpression)

**{**

**ON booleanExpression |**

[***USING clause***](https://docs.oracle.com/javadb/10.8.3.0/ref/rrefsqljusing.html#rrefsqljusing)

**}**

**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**.

Additional operations are −

* Set intersection
* Assignment
* Natural join
* **Assignment operation**

The equal sign (=) is the only Transact-SQL assignment operator. In the following example, the @MyCounter variable is created, and then the assignment operator sets @MyCounter to a value returned by an expression.

DECLARE @MyCounter INT;

SET @MyCounter = 1;

The assignment operator can also be used to establish the relationship between a column heading and the expression that defines the values for the column. The following example displays the column headings First Column Heading and Second Column Heading. The string xyz is displayed in the First Column Heading column heading for all rows. Then, each product ID from the Product table is listed in the Second Column Heading column heading.

**Division operation**

The division operation will return a Relation R(X) that includes all tuples t[X] in R(Z) that appear in R1 in combination with every tuple from R2(Y), where Z = X ∪ Y. The division operator is symbolized by:

* R1(Z) ∻ R2(Y)

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:

SELECT surname, forenames

FROM employee X

WHERE NOT EXISTS

(SELECT ‘X’

FROM employee y

WHERE NOT EXISTS

(SELECT ‘X’

FROM employee z

WHERE x.empno = z.empno

AND y.surname = z.surname))

ORDER BY empno

**Additional operations**

Arithmetic operators can perform arithmetical operations on numeric operands involved. Arithmetic operators are addition(+), subtraction(-), multiplication(\*) and division(/). The + and - operators can also be used in date arithmetic.

Syntax

SELECT <Expression>[arithmetic operator]<expression>...

FROM [table\_name]

WHERE [expression];

* + Set –intersection operation
  + Natural join operation