ST. XAVIER’S COLLEGE

**Maitighar, Kathmandu**

**(Affiliated to Tribhuvan University)**



**Database Management System**

**Theory Assignment #6**

**Submitted By**

Sneha Prasai

013BSCCSIT040

**Submitted To**

Er. Sanjay Kumar Yadav

Lecturer

Department of Computer Science

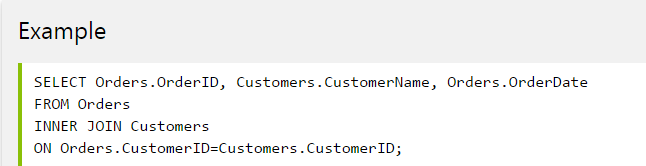
St. Xavier’s College

Maitighar, Kathmandu

JOIN

A SQL **join** clause combines records from two or more tables in a relational database. It creates a set that can be saved as a table or used as it is. A **JOIN** is a means for combining fields from two tables (or more) by using values common to each.

The most common type of join is: **SQL INNER JOIN (simple join)**. An SQL INNER JOIN return all rows from multiple tables where the join condition is met.



Types of join:

1. Theta Join
2. Equi Join
3. Semi Join
4. Natural Join
5. Outer Join

Theta Join:

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:

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.

**INNER JOIN**: Returns all rows when there is at least one match in BOTH tables

**LEFT JOIN**: Return all rows from the left table, and the matched rows from the right table

**RIGHT JOIN**: Return all rows from the right table, and the matched rows from the left table

**FULL JOIN**: Return all rows when there is a match in ONE of the tables

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**.

assignment 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.”

division operation:

additional operation

set-intersection operation:

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.

8. natural join operation:

The natural-join operation is a binary operation on relations r(R) and s(S) that is denoted by the symbol ./. Intuitively, a natural-join “matches” the tuples of r with the tuples of s based on attributes that are both in r and s.

• If we take the relational schemas R and S as sets of attributes, then we can define “attributes that are in both r and s” as R ∩ S = {A1, A2, . . . , An}. With that, we can formally define r ./ s as:

r ./ s = ΠR ∪ S(σr.A1 = s.A1 ∧ r.A2 = s.A2 ∧ ... ∧ r.An = s.An (r × s))

• Note that R ∪ S removes duplicate attribute names, so r ./ s will only have one attribute Ak ∀Ak ∈ R ∩ S.

• Natural join is associative — that is, (a ./ b) ./ c = a ./ (b ./ c).

• When r and s do not have any common attributes — i.e., R ∩ S = ∅ — then r ./ s = r × s.

http://myweb.lmu.edu/dondi/share/db/relational3.pdf