ST. XAVIER'S COLLEGE

**(Affiliated to Tribhuvan University)**

Maitighar, Kathmandu



**DATABASE MANAGEMENT SYSTEM**

**THEORY ASSIGNMENT #6**

**SUBMITTED BY:**

Dikita Tuladhar  
013BSCCSIT018

2nd Year/4th Sem

**SUBMITTED TO:**

|  |  |
| --- | --- |
| **Er. Sanjay Kumar Yadav**  Supervisor,  Department of Computer Science  St. Xavier’s College |  |

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

* **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. ANSI-standard SQL specifies five types of join: inner, left outer, right outer, full outer and cross. As a special case, a table (base table, view, or joined table) can join to itself in a self*-*join.

A programmer writes a join statement to identify the records for joining. If the evaluated predicate is true, the combined record is then produced in the expected format, a record set or a temporary table.

There are two types of joins. They are as follows:

* + **Theta join**

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

|  |  |  |
| --- | --- | --- |
| **Student** | | |
| **SID** | **Name** | **Std** |
| 101 | Alex | 10 |
| 102 | Maria | 11 |
|  |  |  |

|  |  |
| --- | --- |
| **Subjects** | |
| **Class** | **Subject** |
| 10 | Math |
| 10 | English |
| 11 | Music |
| 11 | Sports |

Student\_Detail −

STUDENT ⋈Student.Std = Subject.Class SUBJECT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Student\_detail** | | | | |
| **SID** | **Name** | **Std** | **Class** | **Subject** |
| 101 | Alex | 10 | 10 | Math |
| 101 | Alex | 10 | 10 | English |
| 102 | Maria | 11 | 11 | Music |
| 102 | Maria | 11 | 11 | Sports |

* + **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.

Natural join acts on those matching attributes where the values of attributes in both the relations are same.

|  |  |  |
| --- | --- | --- |
| **Courses** | | |
| **CID** | **Course** | **Dept** |
| CS01 | Database | CS |
| ME01 | Mechanics | ME |
| EE01 | Electronics | EE |

|  |  |
| --- | --- |
| **HoD** | |
| **Dept** | **Head** |
| CS | Alex |
| ME | Maya |
| EE | Mira |

|  |  |  |  |
| --- | --- | --- | --- |
| **Courses ⋈ HoD** | | | |
| **Dept** | **CID** | **Course** | **Head** |
| CS | CS01 | Database | Alex |
| ME | ME01 | Mechanics | Maya |
| EE | EE01 | Electronics | Mira |

* + - **Right join**

Return all rows from the right table, and the matched rows from the left table

All the tuples from the Right relation, S, are included in the resulting relation. If there are tuples in S without any matching tuple in R, then the R-attributes of resulting relation are made null.

* + - **Left join**

Return all rows from the left table, and the matched rows from the right table

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.

* + - **Inner join**

Returns all rows when there is at least one match in both tables

An SQL JOIN clause is used to combine rows from two or more tables, based on a common field between them. There are different types of joins available in SQL: inner join: returns rows when there is a match in both 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. it is noted as − 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**

Assignment operator causes the user variable on the left hand side of the operator to take on the value to its right. The value on the right hand side may be a literal value, another variable storing a value, or any legal expression that yields a scalar value, including the result of a query (provided that this value is a scalar value). You can perform multiple assignments in the same set statement. You can perform multiple assignments in the same statement-

Unlike =, the: = operator is never interpreted as a comparison operator. This means you can use: = in any valid SQL statement (not just in set statements) to assign a value to a variable.

* **Division operation**

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**
* **Set-intersection operation**
* **Natural join operation**