ST. XAVIER’S COLLEGE

**(Affiliated to Tribhuvan University)**

Maitighar, Kathmandu



**Database Management System**

**Lab assignment #6**

**Submitted by:**

Arjun Neupane

013BSCCSIT009

**Submitted to:**

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

Date of submission: 10th September 2015

**1.Theta Join:**

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

Notation:

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 .

|  |  |  |
| --- | --- | --- |
| **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 |

**2. 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 |

Types of natural join:

1. Right Join
2. Left Join
3. Inner Join

**2.1. Right Join:**

This join returns all the rows from the left table in conjunction with the matching rows from the right table. If there are no columns matching in the right table, it returns NULL values.

Implementation:

Select \*

FROM Table1 A RIGHT OUTER JOIN Table2 B On A.Pk = B.Fk;

**2.2. Left outer Join(R Left Outer Join S):**

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

Implementation:

Select \*

FROM Table1 A LEFT OUTER JOIN Table2 B On A.Pk = B.Fk;

**2.3. Inner Join:**

It merges matched rows from two tables. The matching is done based on common columns of tables and their comparing operation. If equality based condition then: EQUI-JOIN performed, otherwise Non-EQUI-Join.

Implementation:

Select A.\*, B.Col1, B.Col2 --But no B.ForeingKeyColumn in Select

FROM Table1 A

INNER JOIN Table2 B On A.Pk = B.Fk;

|  |  |
| --- | --- |
| Inner Join | Outer Join |
| Left outer Join | Right outer Join |

## 3. Rename Operator (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)

It is simply a mechanism to rename both relations and attributes.

Example:

**select distinct** cname, borrower.loan# **as** loan\_id

**from** borrower, loan

**where** borrower.loan# = loan.loan#

**and** bname= ``SFU"[3]

**4. Assignment Operator:**

An assignment operator is the operator used to assign a new value to a variable, property, event or indexer element in C# programming language. Assignment operators can also be used for logical operations such as bitwise logical operations or operations on integral operands and Boolean operands.

Unlike in C++, assignment operators in C# cannot be overloaded directly, but the user-defined types can overload the operators like +, -, /, etc. This allows the assignment operator to be used with those types[4].

**5. Division Operator:**

Division, denoted http://www.cs.sfu.ca/CourseCentral/354/zaiane/material/notes/Chapter3/_7092_tex2html_wrap1375.gif, is suited to queries that include the phrase ``for all''.

Suppose we want to find all the customers who have an account at **all** branches located in Brooklyn.

Strategy: think of it as three steps.

We can obtain the names of all branches located in Brooklyn by

http://www.cs.sfu.ca/CourseCentral/354/zaiane/material/notes/Chapter3/_7092_displaymath1319.gif

Figure 3.19 in the textbook shows the result.

We can also find all cname, bname pairs for which the customer has an account by

http://www.cs.sfu.ca/CourseCentral/354/zaiane/material/notes/Chapter3/_7092_displaymath1320.gif

Now we need to find all customers who appear in http://www.cs.sfu.ca/CourseCentral/354/zaiane/material/notes/Chapter3/_7092_tex2html_wrap1261.gif with **every** branch name in http://www.cs.sfu.ca/CourseCentral/354/zaiane/material/notes/Chapter3/_7092_tex2html_wrap1251.gif.

The divide operation provides exactly those customers:

http://www.cs.sfu.ca/CourseCentral/354/zaiane/material/notes/Chapter3/_7092_eqnarray492.gif

which is simply http://www.cs.sfu.ca/CourseCentral/354/zaiane/material/notes/Chapter3/_7092_tex2html_wrap1381.gif.

Formally,

* Let http://www.cs.sfu.ca/CourseCentral/354/zaiane/material/notes/Chapter3/_7092_tex2html_wrap1383.gif and http://www.cs.sfu.ca/CourseCentral/354/zaiane/material/notes/Chapter3/_7092_tex2html_wrap1385.gif be relations.
* Let http://www.cs.sfu.ca/CourseCentral/354/zaiane/material/notes/Chapter3/_7092_tex2html_wrap1387.gif.
* The relation http://www.cs.sfu.ca/CourseCentral/354/zaiane/material/notes/Chapter3/_7092_tex2html_wrap1409.gif is a relation on scheme http://www.cs.sfu.ca/CourseCentral/354/zaiane/material/notes/Chapter3/_7092_tex2html_wrap1405.gif.
* A tuple http://www.cs.sfu.ca/CourseCentral/354/zaiane/material/notes/Chapter3/_7092_tex2html_wrap1447.gif is in http://www.cs.sfu.ca/CourseCentral/354/zaiane/material/notes/Chapter3/_7092_tex2html_wrap1409.gif if for every tuple http://www.cs.sfu.ca/CourseCentral/354/zaiane/material/notes/Chapter3/_7092_tex2html_wrap1397.gif in http://www.cs.sfu.ca/CourseCentral/354/zaiane/material/notes/Chapter3/_7092_tex2html_wrap1531.gif there is a tuple http://www.cs.sfu.ca/CourseCentral/354/zaiane/material/notes/Chapter3/_7092_tex2html_wrap1401.gif in http://www.cs.sfu.ca/CourseCentral/354/zaiane/material/notes/Chapter3/_7092_tex2html_wrap1345.gif satisfying both of the following:

http://www.cs.sfu.ca/CourseCentral/354/zaiane/material/notes/Chapter3/_7092_eqnarray502.gif

* These conditions say that the http://www.cs.sfu.ca/CourseCentral/354/zaiane/material/notes/Chapter3/_7092_tex2html_wrap1405.gif portion of a tuple http://www.cs.sfu.ca/CourseCentral/354/zaiane/material/notes/Chapter3/_7092_tex2html_wrap1447.gif is in http://www.cs.sfu.ca/CourseCentral/354/zaiane/material/notes/Chapter3/_7092_tex2html_wrap1409.gif if and only if there are tuples with the http://www.cs.sfu.ca/CourseCentral/354/zaiane/material/notes/Chapter3/_7092_tex2html_wrap1291.gif portion **and** the http://www.cs.sfu.ca/CourseCentral/354/zaiane/material/notes/Chapter3/_7092_tex2html_wrap1335.gif portion in http://www.cs.sfu.ca/CourseCentral/354/zaiane/material/notes/Chapter3/_7092_tex2html_wrap1345.gif for **every** value of the http://www.cs.sfu.ca/CourseCentral/354/zaiane/material/notes/Chapter3/_7092_tex2html_wrap1335.gif portion in relation http://www.cs.sfu.ca/CourseCentral/354/zaiane/material/notes/Chapter3/_7092_tex2html_wrap1335.gif.

We will look at this explanation in class more closely.

The division operation can be defined in terms of the fundamental operations.

http://www.cs.sfu.ca/CourseCentral/354/zaiane/material/notes/Chapter3/_7092_displaymath1321.gif

**Reference:**

[1]<https://www.google.com.np/search?q=types+of+join&oq=types+of+join&aqs=chrome..69i57j0l5.5347j0j4&sourceid=chrome&es_sm=122&ie=UTF-8>

[2] <http://stackoverflow.com/questions/17946221/sql-join-and-different-types-of-joins>

[3] <http://www.cs.sfu.ca/CourseCentral/354/zaiane/material/notes/Chapter4/node7.html>

[4] <https://www.techopedia.com/definition/25583/assignment-operator-c>

[5] <http://www.cs.sfu.ca/CourseCentral/354/zaiane/material/notes/Chapter3/node10.html>