**ST.XAVIER’S COLLEGE**

MAITIGHAR, KATHMANDU



Database Management System

Assignment #6

Submitted By:

Aashish Raj Shrestha

013BSCCSIT002

2nd year/ 4th semester

Submitted to:

|  |  |
| --- | --- |
| Er. Sanjay Kumar Yadav  Lecturers  Department of Computer Science |  |

**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.

We will briefly describe various join types in the following sections.

**Theta (θ) 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.

Theta join can use all kinds of comparison operators.

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

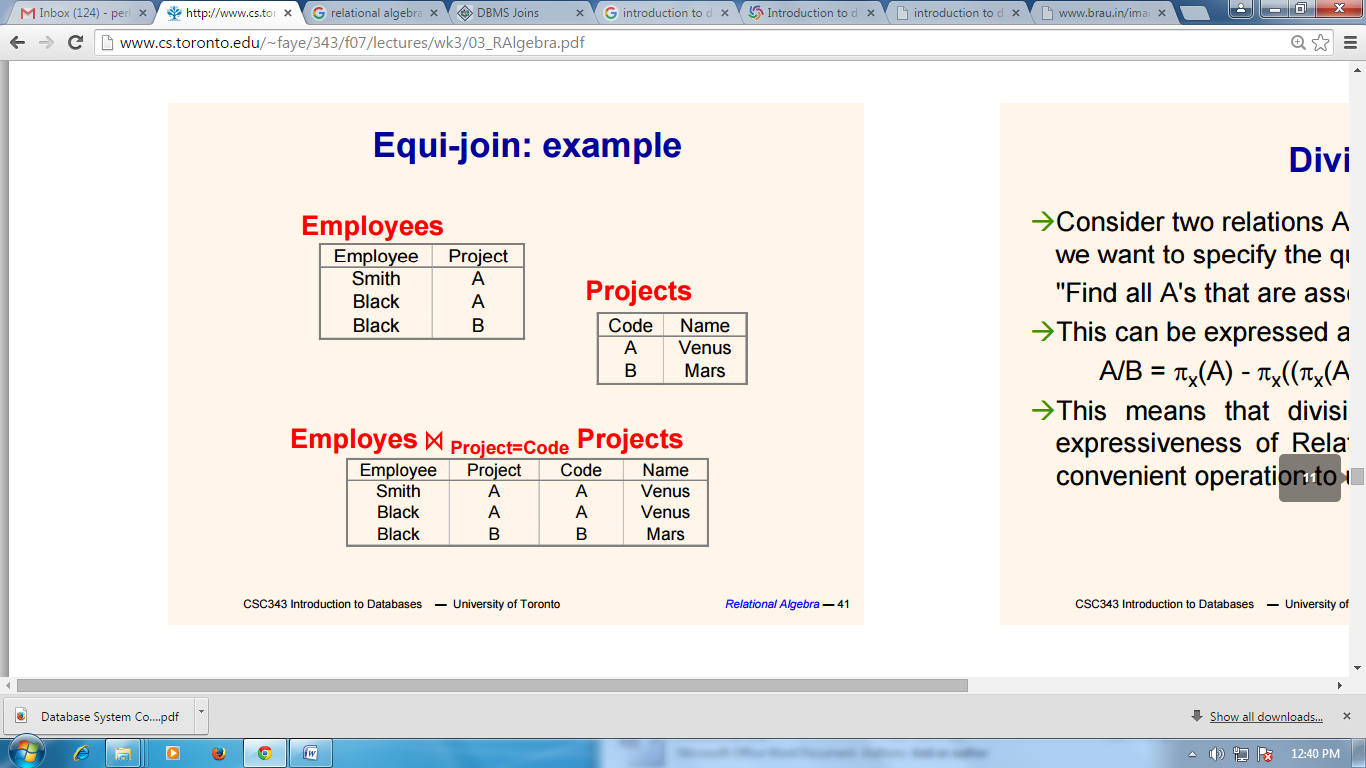
|  |  |
| --- | --- |
| **Subjects** | |
| **Class** | **Subject** |
| 10 | Math |
| 10 | English |
| 11 | Music |
| 11 | Sports |

Student\_Detail −

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Student\_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 |

**Equijoin**

When Theta join uses only **equality** comparison operator, it is said to be equijoin. The above example corresponds to 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.

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 |

**Outer Joins**

Theta Join, Equijoin, and Natural Join are called inner joins. An inner join includes only those tuples with matching attributes and the rest are discarded in the resulting relation. Therefore, we need to use outer joins to include all the tuples from the participating relations in the resulting relation. There are three kinds of outer joins − left outer join, right outer join, and full outer join.

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.

|  |  |
| --- | --- |
| **Left** | |
| **A** | **B** |
| 100 | Database |
| 101 | Mechanics |
| 102 | Electronics |

|  |  |
| --- | --- |
| **Right** | |
| **A** | **B** |
| 100 | Alex |
| 102 | Maya |
| 104 | Mira |

|  |  |  |  |
| --- | --- | --- | --- |
| **Courses Left Outer Join HoD** | | | |
| **A** | **B** | **C** | **D** |
| 100 | Database | 100 | Alex |
| 101 | Mechanics | --- | --- |
| 102 | Electronics | 102 | Maya |

**Right Outer Join: ( R Right Outer Join S )**

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.

|  |  |  |  |
| --- | --- | --- | --- |
| **Courses Right Outer Join HoD** | | | |
| **A** | **B** | **C** | **D** |
| 100 | Database | 100 | Alex |
| 102 | Electronics | 102 | Maya |
| --- | --- | 104 | Mira |

**Full Outer Join: ( R Full Outer Join S)**

All the tuples from both participating relations are included in the resulting relation. If there are no matching tuples for both relations, their respective unmatched attributes are made NULL.

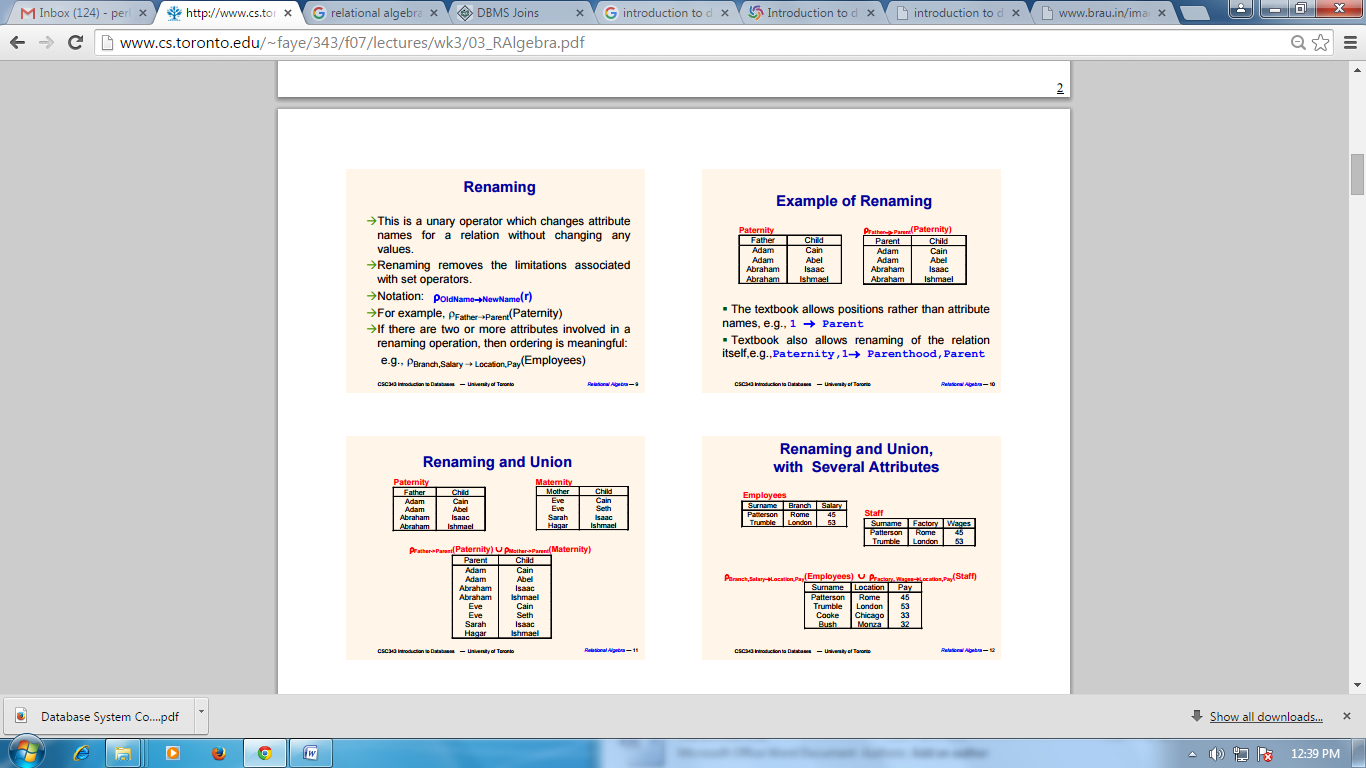
|  |  |  |  |
| --- | --- | --- | --- |
| **Courses Full Outer Join HoD** | | | |
| **A** | **B** | **C** | **D** |
| 100 | Database | 100 | Alex |
| 101 | Mechanics | --- | --- |
| 102 | Electronics | 102 | Maya |
| --- | --- | 104 | Mira |

**Rename Operation (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)

Where the result of expression **E** is saved with name of **x**.



**Division:**

Find items in one set that are related to all of the items in another set.

In a many-to-many relationship there are three tables, A, B, C with C as the table representing the many-to-many key pairs of A and B.

For simple division: What are the 'A\_KEY's to which all 'B\_KEY's belong?

select distinct A\_KEY

from TABLE\_C C

where not exists (

select B\_KEY

from TABLE\_B B

where not exists (

select \*

from TABLE\_C CC

where A.A\_KEY = CC.A\_KEY

and B.B\_KEY = CC.B\_KEY ))

REFERENCES:

[1] University of Toronto. “Relational Algebra*”. CSC343 Introduction to Databases.*  Link: http://www.cs.toronto.edu/~faye/343/f07/lectures/wk3/03\_RAlgebra.pdf

[2] tutorials point Link: http://www.tutorialspoint.com/dbms/database\_joins.htm