**DBMS - JOINS**

**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.

Various joins are briefly described in the following sections.

# 1.1 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**.

# 1.1.1 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** | | |
| SSID | **Name** | **Std** |
| 101  102 | Alex  Maria | 10  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 |

# 1.2 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 | Maria |

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

# 1.2.1 Types of Natural Join

# 1.2.1.1 Left Join(R 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 | Maria |

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

# 1.2.1.2 Right Join: ( R 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**  **HoD** | | | |
| **A** | **B** | **C** | **D** |
| 100 | Database | 100 | Alex |
| 102 | Electronics | 102 | Maya |
| --- | ---- | 104 | Maria |

# 1.2.1.3 Inner Join: ( R 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**  **HoD** | | | |
| **A** | **B** | **C** | **D** |
| 100 | Database | 100 | Alex |
| 101 | Mechanics | --- | ---- |
| 102 | Electronics | 102 | Maya |
| --- | ----- | 104 | Maria |

**REFERENCES**

https://docs.oracle.com/javadb/10.8.3.0/ref/rrefsqljnaturaljoin.html