DBMS Joins

We understand the benefits of Cartesian product of two relation, which gives us all the possible tuples that are paired together. But Cartesian product might not be feasible for huge relations where number of tuples are in thousands and the attributes of both relations are considerable large.

Join is combination of Cartesian product followed by selection process. Join operation pairs two tuples from different relations if and only if the given join condition is satisfied.

Following section should describe briefly about join types:

Theta (θ) join

θ in Theta join is the join condition. Theta joins combines tuples from different relations provided they satisfy the theta condition.

Notation:

```
R1 ⋈θ R2
```

R1 and R2 are relations with their attributes (A1, A2, .., An) and (B1, B2,.. ,Bn) such that no attribute matches that is R1 ∩ R2 = Φ Here θ is condition in form of set of conditions C.

Theta join can use all kinds of comparison operators.

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

[*Table: Student Relation*]

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

[*Table: Subjects Relation*]

Student_Detail =

$$\text{STUDENT} \bowtie_{\text{Student.Std = Subject.Class}} \text{SUBJECT}$$

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

[*Table: Output of theta join*]

Equi-Join

When Theta join uses only **equality** comparison operator it is said to be Equi-Join. The above example conrresponds to equi-join

Natural Join (⋈)

Natural join does not use any comparison operator. It does not concatenate the way Cartesian product does. Instead, Natural Join can only be performed if the there is at least one common attribute exists between relation. Those attributes must have same name and domain.

Natural join acts on those matching attributes where the values of attributes in both relation is same.

Courses

CID	Course	Dept
CS01	Database	CS
ME01	Mechanics	ME
EE01	Electronics	EE

[*Table: Relation Courses*]

HoD

Dept	Head
CS	Alex

ME	Maya
EE	Mira

[*Table: Relation HoD*]

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

[*Table: Relation Courses ⋈ HoD*]

Outer Joins

All joins mentioned above, that is Theta Join, Equi Join and Natural Join are called inner-joins. An inner-join process includes only tuples with matching attributes, rest are discarded in resulting relation. There exists methods by which all tuples of any relation are included in the resulting relation.

There are three kinds of outer joins:

Left outer join (R ⟕ S)

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

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

[*Table: Left Relation*]

Right	
A	**B**

100	Alex
102	Maya
104	Mira

[*Table: Right Relation*]

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

[*Table: Left outer join output*]

Right outer join: (R ⟖ S)

All tuples of the Right relation, S, are included in the resulting relation and if there exists 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	Mira

[*Table: Right outer join output*]

Full outer join: (R ⟗ S)

All tuples of both participating relations are included in the resulting relation and if there 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	Mira

[*Table: Full outer join output*]

Source:

http://www.tutorialspoint.com/dbms/database_joins.htm