

DBMS - Joins

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

Copyright © tutorialspoint.com

We understand the benefits of taking a Cartesian product of two relations, which gives us all the possible tuples that are paired together. But it might not be feasible for us in certain cases to take a Cartesian product where we encounter huge relations with thousands of tuples having a considerable large number of attributes.

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

$R1 \bowtie_{\theta} R2$

$R1$ and $R2$ are relations having attributes $A1, A2, \dots, An$ and $B1, B2, \dots, Bn$ such that the attributes don't have anything in common, that is $R1 \cap R2 = \Phi$.

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 –

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

Student_detail

SID Name Std Class Subject

101 Alex 10 10 Math

101 Alex 10 10 English

Processing math: 0%

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 ⋈ 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.

100 Database
101 Mechanics
102 Electronics

Right

A B

100 Alex
102 Maya
104 Mira

Courses HoD

A B C D

100 Database 100 Alex
101 Mechanics --- ---
102 Electronics 102 Maya

Right Outer 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 Mira

Full Outer 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 Mira