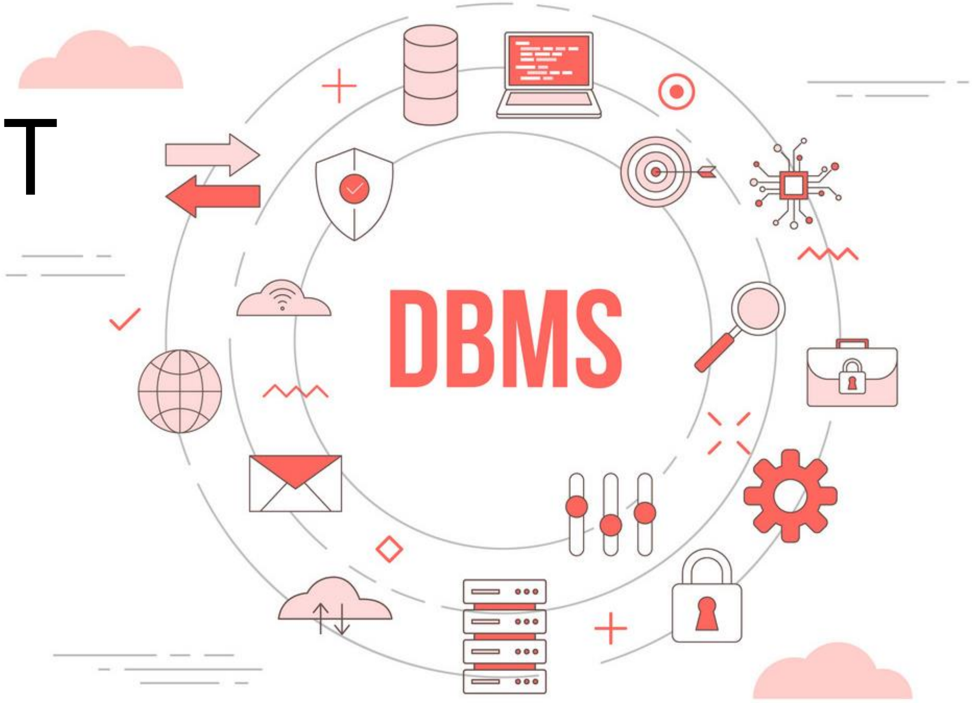
DATABASE MANAGEMENT SYSTEMS

JOIN



Dr. Jay Sarraf

School of Computer Engineering

KIIT Deemed to be University

JOIN (⋈)

JOIN (⋈)

In DBMS, a join is a binary operation that allows you to combine a join product and a selection in a single statement. The purpose of creating a join condition is to assist you in combining data from two or more relational tables. In DBMS, relational tables are linked together using main and foreign keys.

The join operation connects two relations by combining tuples from two relations that meet a predefined criterion. The criterion is based on characteristics of the relations to be connected.

Types of Join

- **Inner Joins:** Theta, Natural, EQUI
- **Outer Join:** Left, Right, Full
- **Self Join**

INNER JOIN (⋈)

INNER JOIN (⋈)

Inner join is used to return rows from both tables that meet the specified criterion. It is the most often used join operation and may be thought of as the default join-type.

Types of Inner Join

- Theta (⋈ θ)
- Natural (⋈)
- EQUI (⋈ =)

Theta Join (⋈ θ)

Theta Join (⋈ θ)

Theta Join joins two tables depending on the condition indicated by theta.

For all comparison operators, Theta joins is compatible. It is represented by the symbol (⋈ θ) . A Theta join is the general case of a JOIN operation.

The comparison operator can be any of these: <,>, $\underline{<}$, $\underline{>}$, = and ≠.

Syntax: $A \bowtie_{\theta} B$

General Form:

$R \bowtie_{\theta} S = \pi \text{ all } (\sigma_{\theta} (R \times S))$

Cont…

STUDENTS

SROLL	SNAME	SAGE	CITY
S1	Avril	20	BBSR
S2	Byril	19	CTC
S3	Cyan	20	BBSR

ENROLLMENT

SROLL	SNAME	FID	CID
S1	Avril	F1	C1
S2	Byril	F2	C2
S3	Cyan	F3	C3
S1	Avril	F2	C2
S2	Byril	F2	C1
S1	Avril	F3	C3
S3	Cyan	F1	C1

Query:

Find the details of all students who have taken enrolment in C1 and age less then 20.

(STUDENTS ⋈ student.sroll = enrollment.sroll AND enrollment.cid = ‘c1’ AND student.age <20 ***ENROLLMENT)***

EQUI Join ($\bowtie_{=}$)

EQUI Join ($\bowtie_{=}$)

Equi join is the same as Theta Join, with the exception that it only employs the equivalence criteria when joining two tables.

The = operator is use for comparison.

Syntax: $A \bowtie_{=} B$

General Form:

$$R \bowtie_{=} S = \pi \text{ all } (\sigma_{\theta} (R \times S))$$

Cont…

STUDENTS

SROLL	SNAME	SAGE	CITY
S1	Avril	20	BBSR
S2	Byril	19	CTC
S3	Cyan	20	BBSR

ENROLLMENT

SROLL	SNAME	FID	CID
S1	Avril	F1	C1
S2	Byril	F2	C2
S3	Cyan	F3	C3
S1	Avril	F2	C2
S2	Byril	F2	C1
S1	Avril	F3	C3
S3	Cyan	F1	C1

Query:

Find the details of all students who have taken enrolment in C1 and age is equal to 20.

(STUDENTS ⋈ student.sroll = enrolment.sroll AND enrolment.cid = ‘c1’ AND student.age = 20 ***ENROLLMENT)***

NATURAL Join (⋈)

NATURAL Join (⋈)

Natural Join makes no use of any comparison operators. The attributes in this form of join should have the same name and domain. There should be at least one common attribute between two relations in Natural Join. It conducts selection creating equality on the characteristics that exist in both relations and removes the duplicates.

Syntax: $A \bowtie_{\theta} B$

General Form:

$R \bowtie S = \pi_{all} (R \bowtie S)$

Cont…

STUDENTS

SROLL	SNAME	SAGE	CITY
S1	Avril	20	BBSR
S2	Byril	19	CTC
S3	Cyan	20	BBSR

ENROLLMENT

SROLL	SNAME	FID	CID
S1	Avril	F1	C1
S2	Byril	F2	C2
S3	Cyan	F3	C3
S4	Dev	F4	C4
S5	Eva	F5	C5

Query:

Find the details of students who have taken enrolment in course 1.

STUDENTS ⋈ ENROLLMENT

SROLL	SNAME	SAGE	CITY	FID	CID
S1	Avril	20	BBSR	F1	C1
S2	Byril	19	CTC	F3	C3
S3	Cyan	20	BBSR	F4	C4

OUTER JOIN

OUTER JOIN

Depending on the criterion, the outer join in Relational algebra returns all the characteristics of both tables. If an attribute value is not present for any of the tables, it returns NULL in the table attribute's relevant row.

Types of Outer Join

- LEFT OUTER (⟕)
- RIGHT OUTER (⟖)
- FULL OUTER (⟗)

Database Management Systems

Types of Join

LEFT OUTER Join (⟕)

LEFT OUTER Join (⟕)

The resultant relation contains all of the tuples from the Left relation, R. If there are tuples in R that do not have a corresponding tuple in the Right relation S, the S-attributes of the resultant relation are set to NULL.

Syntax: A ⟕ B

All information from the left relation is present in the result of the left outer join.

Database Management Systems

Types of Join

Cont...

STUDENTS

SROLL	SNAME	SAGE	CITY
S1	Avril	20	BBSR
S2	Byril	19	CTC
S3	Cyan	20	BBSR
S6	Ferry	19	RKL

ENROLLMENT

SROLL	SNAME	FID	CID
S1	Avril	F1	C1
S2	Byril	F2	C2
S3	Cyan	F3	C3
S4	Dev	F4	C4
S5	Eva	F5	C5

Query:

Find the details of students who have taken enrolment and who have not.

STUDENTS ⟕ ENROLLMENT

SROLL	SNAME	SAGE	CITY	FID	CID
S1	Avril	20	BBSR	F1	C1
S2	Byril	19	CTC	F2	C2
S3	Cyan	20	BBSR	F3	C3
S6	Ferry	19	RKL	Null	Null

π s.sroll, e.cid (STUDENTS ⟕ ENROLLMENT)

SROLL	CID
S1	C1
S2	C2
S3	C3
S6	Null

RIGHT OUTER Join (⟖)

RIGHT OUTER Join (⟖)

The resultant relation contains all of the tuples from the Right relation, S. If there are tuples in S that do not have a corresponding tuple in R, the R-attributes of the resultant relation are set to NULL.

Syntax: A ⟖ B

All information from the right relation is present in the result of the right outer join.

Cont…

STUDENTS

SROLL	SNAME	SAGE	CITY
S1	Avril	20	BBSR
S2	Byril	19	CTC
S3	Cyan	20	BBSR
S6	Ferry	19	RKL

ENROLLMENT

SROLL	SNAME	FID	CID
S1	Avril	F1	C1
S2	Byril	F2	C2
S3	Cyan	F3	C3
S4	Dev	F4	C4
S5	Eva	F5	C5

Query:

Find the details of students who have taken enrolment and who have not.

STUDENTS ⟖ ENROLLMENT

SROLL	SNAME	FID	CID	SAGE	CITY
S1	Avril	F1	C1	20	BBSR
S2	Byril	F2	C2	19	CTC
S3	Cyan	F3	C3	20	BBSR
S4	Dev	F4	C4	Null	Null
S5	Eva	F5	C5	Null	Null

π ***s.sroll, s.sage e.cid (STUDENTS ⟖ ENROLLMENT)***

SROLL	SAGE	CID
S1	20	C1
S2	19	C2
S3	20	C3
S4	Null	C4
S5	Null	C5

FULL OUTER Join (⟗)

FULL OUTER Join (⟗)

The generated relation contains all of the tuples from both participating relations. If neither relation has any matching tuples, their respective mismatched attributes are set to NULL.

Syntax: A ⟗ B

All information from the both the relation is present in the result of the Full outer join.

Cont…

STUDENTS

SROLL	SNAME	SAGE	CITY
S1	Avril	20	BBSR
S2	Byril	19	CTC
S3	Cyan	20	BBSR
S6	Ferry	19	RKL

ENROLLMENT

SROLL	SNAME	FID	CID
S1	Avril	F1	C1
S2	Byril	F2	C2
S3	Cyan	F3	C3
S4	Dev	F4	C4
S5	Eva	F5	C5

Query:

Find the details of students who have taken enrolment and who have not.

STUDENTS ⟗ ENROLLMENT

SROLL	SNAME	SAGE	CITY	FID	CID
S1	Avril	20	BBSR	F1	C1
S2	Byril	19	CTC	F2	C2
S3	Cyan	20	BBSR	F3	C3
S4	Dev	Null	Null	F4	C4
S5	Eva	Null	Null	F5	C5
S6	Ferry	19	RKL	Null	Null

SELF Join

SELF JOIN

A self-join is a join that may be used to connect two tables. As a result, it is a unary relationship. Each row of the table is attached to itself and all other rows of the same table in a self-join. As a result, a self-join is mostly employed in databases to integrate and compare rows from the same table.

Syntax: $A \bowtie A_{copy}$

Nevertheless, doing a self-join generates uncertainty since we must use the same table name again. To prevent these situations, we must utilize alias names for the same tables.

Database Management Systems

Types of Join

Cont…

STUDENTS

SROLL	SNAME	SAGE	CITY
S1	Avril	20	BBSR
S2	Byril	19	CTC
S3	Cyan	20	BBSR
S6	Ferry	19	RKL

STUDENTS 2

SROLL	SNAME	SAGE	CITY
S1	Avril	20	BBSR
S2	Byril	19	CTC
S3	Cyan	20	BBSR
S6	Ferry	19	RKL

Query:

Find the details of students who belong to the same city.

STUDENTS* ⋈ *STUDENTS 2

Database Management Systems

Types of Join

Cont…

STUDENTS ⋈ **s1.sroll = s2.sroll AND** ***s1.city = s2.city*** ***STUDENTS2***

S1. SROLL	S1. SNAME	S1. SAGE	S1. CITY	S2. SROLL	S2. SNAME	S2. SAGE	S2. CITY
S1	Avril	20	BBSR	S3	Cyan	20	BBSR
S3	Cyan	20	BBSR	S1	Avril	20	BBSR

Cont…

ENROLLMENT1

SROLL	SNAME	FID	CID
S1	Avril	F1	C1
S2	Byril	F2	C2
S3	Cyan	F3	C3
S1	Avril	F2	C2
S2	Byril	F2	C1

ENROLLMENT2

SROLL	SNAME	FID	CID
S1	Avril	F1	C1
S2	Byril	F2	C2
S3	Cyan	F3	C3
S1	Avril	F2	C2
S2	Byril	F2	C1

Query:

Find the details of students who have enrolled to at least 2 courses.

ENROLLMENT1 $\bowtie$ ***ENROLLMENT2***

Cont…

ENROLLMENT 1 ⋈ E1.sroll = E2.sroll AND ***E1.cid < > E2.cid*** ***ENROLLMENT2***

E1. SROLL	E1. SNAME	E1. FID	E1. CID	E2. SROLL	E2. SNAME	E2. FID	E2. CID
~~S1~~	~~Avril~~	~~F1~~	~~C1~~	~~S1~~	~~Avril~~	~~F1~~	~~C1~~
~~S1~~	~~Avril~~	~~F1~~	~~C1~~	~~S2~~	~~Byril~~	~~F2~~	~~C2~~
~~S1~~	~~Avril~~	~~F1~~	~~C1~~	~~S3~~	~~Cyan~~	~~F3~~	~~C3~~
S1	Avril	F1	C1	S1	Avril	F2	C2
~~S1~~	~~Avril~~	~~F1~~	~~C1~~	~~S2~~	~~Byril~~	~~F2~~	~~C1~~
~~S2~~	~~Byril~~	~~F2~~	~~C2~~	~~S1~~	~~Avril~~	~~F1~~	~~C1~~
~~S2~~	~~Byril~~	~~F2~~	~~C2~~	~~S2~~	~~Byril~~	~~F2~~	~~C2~~
~~S2~~	~~Byril~~	~~F2~~	~~C2~~	~~S3~~	~~Cyan~~	~~F3~~	~~C3~~
~~S2~~	~~Byril~~	~~F2~~	~~C2~~	~~S1~~	~~Avril~~	~~F2~~	~~C2~~
S2	Byril	F2	C2	S2	Byril	F2	C1
~~S3~~	~~Cyan~~	~~F3~~	~~C3~~	~~S1~~	~~Avril~~	~~F1~~	~~C1~~
~~S3~~	~~Cyan~~	~~F3~~	~~C3~~	~~S2~~	~~Byril~~	~~F2~~	~~C2~~
~~S3~~	~~Cyan~~	~~F3~~	~~C3~~	~~S3~~	~~Cyan~~	~~F3~~	~~C3~~
~~S3~~	~~Cyan~~	~~F3~~	~~C3~~	~~S1~~	~~Avril~~	~~F2~~	~~C2~~
~~S3~~	~~Cyan~~	~~F3~~	~~C3~~	~~S2~~	~~Byril~~	~~F2~~	~~C1~~
S1	Avril	F2	C2	S1	Avril	F1	C1

Cont…

ENROLLMENT 1 ⋈ **E1.sroll = E2.sroll AND** ***E1.cid < > E2.cid ENROLLMENT2***

E1. SROLL	E1. SNAME	E1. FID	E1. CID	E2. SROLL	E2. SNAME	E2. FID	E2. CID
~~S1~~	~~Avril~~	~~F2~~	~~C2~~	~~S2~~	~~Byril~~	~~F2~~	~~C2~~
~~S1~~	~~Avril~~	~~F2~~	~~C2~~	~~S3~~	~~Cyan~~	~~F3~~	~~C3~~
~~S1~~	~~Avril~~	~~F2~~	~~C2~~	~~S1~~	~~Avril~~	~~F2~~	~~C2~~
~~S1~~	~~Avril~~	~~F2~~	~~C2~~	~~S2~~	~~Byril~~	~~F2~~	~~C1~~
~~S2~~	~~Byril~~	~~F2~~	~~C1~~	~~S1~~	~~Avril~~	~~F1~~	~~C1~~
S2	Byril	F2	C1	S2	Byril	F2	C2
~~S2~~	~~Byril~~	~~F2~~	~~C1~~	~~S3~~	~~Cyan~~	~~F3~~	~~C3~~
~~S2~~	~~Byril~~	~~F2~~	~~C1~~	~~S1~~	~~Avril~~	~~F2~~	~~C2~~
~~S2~~	~~Byril~~	~~F2~~	~~C1~~	~~S2~~	~~Byril~~	~~F2~~	~~C1~~

Cont…

ENROLLMENT 1 ⋈ E1.sroll = E2.sroll AND ***E1.cid < > E2.cid ENROLLMENT2***

E1. SROLL	E1. SNAME	E1. FID	E1. CID	E2. SROLL	E2. SNAME	E2. FID	E2. CID
S1	Avril	F1	C1	S1	Avril	F2	C2
S2	Byril	F2	C2	S2	Byril	F2	C1
S1	Avril	F2	C2	S1	Avril	F1	C1
S2	Byril	F2	C1	S2	Byril	F2	C2