UNIT 2

Lecture 17

Relational Algebra

Joins (Part – I)

Cartesian Product (×)

If Relation R contains m tuples and p attributes and

Relation S contains n tuples and q attributes

then

Relations $\mathbf{R \times S}$ contains m*n tuples and p+q attributes.

Cartesian Product ($R \times S$)

R

A	B
1	2
3	4

S

C	D
5	6
7	8

$R \times S$

A	B	C	D
1	2	5	6
3	4	5	6
1	2	7	8
3	4	7	8

Equivalent SQL Query

SQL > select * from R cross join S; **[SQL 99 syntax]**

SQL > select * from R, S;

Cartesian Product ($R \times S$)

R

A	B	C
1	2	3
3	4	5

S

C	D
5	6
7	8

$R \times S$

A	B	C	C	D
1	2	3	5	6
3	4	5	5	6
1	2	3	7	8
3	4	5	7	8

S

D	E	F
1	1	1
2	3	4
5	6	7

$R \times S$

A	B	C	D	E	F
1	2	3	1	1	1
3	4	5	1	1	1
1	2	3	2	3	4
3	4	5	2	3	4
1	2	3	5	6	7
3	4	5	5	6	7

Equivalent SQL Query

SQL > select * from R cross join S; **[SQL 99 syntax]**

SQL > select * from R, S;

Cartesian Product ($R \times S$)

STUDENT

Rollno	Sname	Sem	Branch	Marks	Pno
1	RAM	3	CSE	40	121
2	SHYAM	5	CSE	50	122
3	MOHAN	7	CSE	55	123
4	GOPAL	5	IT	65	121
5	RINKI	3	MECH	40	122
6	PINKI	3	ETC	90	123

PROJECT

Pno	Pname	Duration
121	P1	10
122	P2	20
123	P3	30

STUDENT × PROJECT

Rollno	Sname	Sem	Branch	Marks	Pno	Pno	Pname	Duration
1	RAM	3	CSE	40	121	121	P1	10
2	SHYAM	5	CSE	50	122	121	P1	10
3	MOHAN	7	CSE	55	123	121	P1	10
4	GOPAL	5	IT	65	121	121	P1	10
5	RINKI	3	MECH	40	122	121	P1	10
6	PINKI	3	ETC	90	123	121	P1	10
1	RAM	3	CSE	40	121	122	P2	20
2	SHYAM	5	CSE	50	122	122	P2	20
3	MOHAN	7	CSE	55	123	122	P2	20
4	GOPAL	5	IT	65	121	122	P2	20
5	RINKI	3	MECH	40	122	122	P2	20
6	PINKI	3	ETC	90	123	122	P2	20
1	RAM	3	CSE	40	121	123	P3	30
2	SHYAM	5	CSE	50	122	123	P3	30
3	MOHAN	7	CSE	55	123	123	P3	30
4	GOPAL	5	IT	65	121	123	P3	30
5	RINKI	3	MECH	40	122	123	P3	30
6	PINKI	3	ETC	90	123	123	P3	30

Equivalent SQL Query

SQL > select * from student
cross join project; **[SQL 99 syntax]**

SQL > select * from student, project;

Cartesian Product ($R \times S$)

STUDENT

Rollno	Sname	Sem	Branch	Marks	Pno
1	RAM	3	CSE	40	121
2	SHYAM	5	CSE	50	122
3	MOHAN	7	CSE	55	123
4	GOPAL	5	IT	65	121
5	RINKI	3	MECH	40	122
6	PINKI	3	ETC	90	123

PROJECT

Pno	Pname	Duration
121	P1	10
122	P2	20
123	P3	30

$\Pi_{sname,\ student.pno,\ project.pno,\ pname}$ (STUDENT × PROJECT)

Sname	Pno	Pno	Pname
RAM	121	121	P1
SHYAM	122	121	P1
MOHAN	123	121	P1
GOPAL	121	121	P1
RINKI	122	121	P1
PINKI	123	121	P1
RAM	121	122	P2
SHYAM	122	122	P2
MOHAN	123	122	P2
GOPAL	121	122	P2
RINKI	122	122	P2
PINKI	123	122	P2
RAM	121	123	P3
SHYAM	122	123	P3
MOHAN	123	123	P3
GOPAL	121	123	P3
RINKI	122	123	P3
PINKI	123	123	P3

Equivalent SQL Query

SQL > select sname, student.pno, project.pno, pname from student cross join project;

[SQL 99 syntax]

SQL > select sname, student.pno, project.pno, pname from student, project;

Cartesian Product ($\mathbf{R} \times \mathbf{S}$)

STUDENT

Rollno	Sname	Sem	Branch	Marks	Pno
1	RAM	3	CSE	40	121
2	SHYAM	5	CSE	50	122
3	MOHAN	7	CSE	55	123
4	GOPAL	5	IT	65	121
5	RINKI	3	MECH	40	122
6	PINKI	3	ETC	90	123

PROJECT

Pno	Pname	Duration
121	P1	10
122	P2	20
123	P3	30

$\prod_{\mathbf{sname}\,,\,\mathbf{pno}\,,\;\mathbf{pname}} (\text{STUDENT} \times \text{PROJECT})$

Invalid Query

Equivalent SQL Query

SQL > select sname, pno, pname from student cross join project;

[SQL 99 syntax]

SQL > select sname, pno, pname from student, project;

Theta Join $(R \bowtie_{\theta c} \mathrm{S})$

Where θc can be $=, \neq, <, >, \leq, \geq$

Types of theta joins

1. Equi Join (When $\theta c\ is\ =)$
2. Non – Equi Join (When $\theta c\ is\ other\ than\ =$)

Theta Join $(R \bowtie_{\theta c} S)$ – Equi Join

STUDENT

Rollno	Sname	Sem	Branch	Marks	Pno
1	RAM	3	CSE	40	121
2	SHYAM	5	CSE	50	122
3	MOHAN	7	CSE	55	123
4	GOPAL	5	IT	65	121
5	RINKI	3	MECH	40	122
6	PINKI	3	ETC	90	123

PROJECT

Pno	Pname	Duration
121	P1	10
122	P2	20
123	P3	30

STUDENT $\bowtie_{student.pno = project.pno}$ PROJECT

Rollno	Sname	Sem	Branch	Marks	Pno	Pno	Pname	Duration
1	RAM	3	CSE	40	121	121	P1	10
4	GOPAL	5	IT	65	121	121	P1	10
2	SHYAM	5	CSE	50	122	122	P2	20
5	RINKI	3	MECH	40	122	122	P2	20
3	MOHAN	7	CSE	55	123	123	P3	30
6	PINKI	3	ETC	90	123	123	P3	30

Equivalent SQL Query

```
SQL > select * from student inner join
project on student.pno = project.pno;
                          [SQL 99 syntax]
SQL > select * from student, project
where student.pno = project.pno;
```

Theta Join $(R \bowtie_{\theta c} S)$ – Equi Join

$\Pi_{sname, pname}$ (STUDENT $\bowtie_{student.pno = project.pno}$ PROJECT)

STUDENT

Rollno	Sname	Sem	Branch	Marks	Pno
1	RAM	3	CSE	40	121
2	SHYAM	5	CSE	50	122
3	MOHAN	7	CSE	55	123
4	GOPAL	5	IT	65	121
5	RINKI	3	MECH	40	122
6	PINKI	3	ETC	90	123

PROJECT

Pno	Pname	Duration
121	P1	10
122	P2	20
123	P3	30

Sname	Pname
RAM	P1
GOPAL	P1
SHYAM	P2
RINKI	P2
MOHAN	P3
PINKI	P3

Equivalent SQL Query

```
SQL > select distinct sname, pname from student inner join project on
student.pno = project.pno;                                   [SQL 99 syntax]

SQL > select distinct sname, pname from student, project where student.pno
= project.pno;
```

Using Cartesian Join $(R \times S)$ – Equi Join

STUDENT

Rollno	Sname	Sem	Branch	Marks	Pno
1	RAM	3	CSE	40	121
2	SHYAM	5	CSE	50	122
3	MOHAN	7	CSE	55	123
4	GOPAL	5	IT	65	121
5	RINKI	3	MECH	40	122
6	PINKI	3	ETC	90	123

PROJECT

Pno	Pname	Duration
121	P1	10
122	P2	20
123	P3	30

$$\sqcap_{sname, pname}(\sigma_{student.pno = project.pno}(\text{STUDENT} \times \text{PROJECT}))$$

Sname	Pname
RAM	P1
GOPAL	P1
SHYAM	P2
RINKI	P2
MOHAN	P3
PINKI	P3

Equivalent SQL Query

SQL > select distinct sname, pname from student cross join project where student.pno = project.pno; [SQL 99 syntax]

SQL > select distinct sname, pname from student, project where student.pno = project.pno;

Equivalence Rule

$$\sigma_{\text{condition}}\,(R \times S) \equiv R \bowtie_{\text{condition}} S$$

Theta Join $(R \bowtie_{\theta c} S)$ – Equi Join

STUDENT

Rollno	Sname	Sem	Branch	Marks	Pno
1	RAM	3	CSE	40	121
2	SHYAM	5	CSE	50	122
3	MOHAN	7	CSE	55	123
4	GOPAL	5	IT	65	121
5	RINKI	3	MECH	40	122
6	PINKI	3	ETC	90	123

PROJECT

Pno	Pname	Duration
121	P1	10
122	P2	20
123	P3	30

$\Pi_{sname,\, pno,\, pname}$ (STUDENT $\bowtie_{student.pno \,=\, project.pno}$ PROJECT)

Invalid Query

Equivalent SQL Query

SQL > select distinct sname, pno, pname from student inner join project on student.pno = project.pno; **[SQL 99 syntax]**

SQL > select distinct sname, pno, pname from student, project where student.pno = project.pno;

Theta Join $(R \bowtie_{\theta c} S)$ – Non Equi Join

STUDENT

Rollno	Sname	Sem	Branch	Marks	Pno
1	RAM	3	CSE	40	121
2	SHYAM	5	CSE	50	122
3	MOHAN	7	CSE	55	123
4	GOPAL	5	IT	65	121
5	RINKI	3	MECH	40	122
6	PINKI	3	ETC	90	123

PROJECT

Pno	Pname	Duration
121	P1	10
122	P2	20
123	P3	30

$\Pi_{sname,\, pname}$ (STUDENT $\bowtie_{student.pno \,\neq\, project.pno}$ PROJECT)

Sname	Pname
SHYAM	P1
MOHAN	P1
RINKI	P1
PINKI	P1
RAM	P2
MOHAN	P2
GOPAL	P2
PINKI	P2
RAM	P3
SHYAM	P3
GOPAL	P3
RINKI	P3

Equivalent SQL Query

SQL > select distinct sname, pname from student inner join project on student.pno <> project.pno; **[SQL 99 syntax]**

SQL > select distinct sname, pname from student, project where student.pno <> project.pno;

Theta Join $(R \bowtie_{\theta c} S)$ – Non Equi Join

STUDENT

Rollno	Sname	Sem	Branch	Marks	Pno
1	RAM	3	CSE	40	121
2	SHYAM	5	CSE	50	122
3	MOHAN	7	CSE	55	123
4	GOPAL	5	IT	65	121
5	RINKI	3	MECH	40	122
6	PINKI	3	ETC	90	123

PROJECT

Pno	Pname	Duration
121	P1	10
122	P2	20
123	P3	30

$\Pi_{sname,\, pname}$ (STUDENT $\bowtie_{student.pno \,>\, project.pno}$ PROJECT)

Sname	Pname
SHYAM	P1
MOHAN	P1
RINKI	P1
PINKI	P1
MOHAN	P2
PINKI	P2

Equivalent SQL Query

```
SQL > select distinct sname, pname from student inner join project on
student.pno > project.pno;                                        [SQL 99 syntax]

SQL > select distinct sname, pname from student, project where student.pno >
project.pno;
```

Natural Join $(R \bowtie S)$

STUDENT

Rollno	Sname	Sem	Branch	Marks	Pno
1	RAM	3	CSE	40	121
2	SHYAM	5	CSE	50	122
3	MOHAN	7	CSE	55	123
4	GOPAL	5	IT	65	121
5	RINKI	3	MECH	40	122
6	PINKI	3	ETC	90	123

PROJECT

Pno	Pname	Duration
121	P1	10
122	P2	20
123	P3	30

STUDENT $\bowtie$ PROJECT

Pno	Rollno	Sname	Sem	Branch	Marks	Pname	Duration
121	1	RAM	3	CSE	40	P1	10
121	4	GOPAL	5	IT	65	P1	10
122	2	SHYAM	5	CSE	50	P2	20
122	5	RINKI	3	MECH	40	P2	20
123	3	MOHAN	7	CSE	55	P3	30
123	6	PINKI	3	ETC	90	P3	30

Equivalent SQL Query

SQL > select * from student natural join project; **[SQL 99 syntax]**

Natural Join $(\mathbf{R} \bowtie \mathbf{S})$

STUDENT

Rollno	Sname	Sem	Branch	Marks	Pno
1	RAM	3	CSE	40	121
2	SHYAM	5	CSE	50	122
3	MOHAN	7	CSE	55	123
4	GOPAL	5	IT	65	121
5	RINKI	3	MECH	40	122
6	PINKI	3	ETC	90	123

PROJECT

Pno	Pname	Duration
121	P1	10
122	P2	20
123	P3	30

$\Pi_{sname,\, pname}$ (STUDENT ⋈ PROJECT)

Sname	Pname
RAM	P1
GOPAL	P1
SHYAM	P2
RINKI	P2
MOHAN	P3
PINKI	P3

Equivalent SQL Query

SQL > select distinct sname, pname from student natural join project;
[SQL 99 syntax]

Natural Join $(R \bowtie S)$

STUDENT

Rollno	Sname	Sem	Branch	Marks	Pno
1	RAM	3	CSE	40	121
2	SHYAM	5	CSE	50	122
3	MOHAN	7	CSE	55	123
4	GOPAL	5	IT	65	121
5	RINKI	3	MECH	40	122
6	PINKI	3	ETC	90	123

PROJECT

Pno	Pname	Duration
121	P1	10
122	P2	20
123	P3	30

$\Pi_{sname,\ pno,\ pname}$ (STUDENT $\bowtie$ PROJECT)

Sname	Pno	Pname
RAM	121	P1
GOPAL	121	P1
SHYAM	122	P2
RINKI	122	P2
MOHAN	123	P3
PINKI	123	P3

Equivalent SQL Query

SQL > select distinct sname, pno, pname from student natural join project;
[SQL 99 syntax]

Natural Join $(R \bowtie S)$

STUDENT

Rollno	Sname	Sem	Branch	Marks	Pno
1	RAM	3	CSE	40	121
2	SHYAM	5	CSE	50	122
3	MOHAN	7	CSE	55	123
4	GOPAL	5	IT	65	121
5	RINKI	3	MECH	40	122
6	PINKI	3	ETC	90	123

PROJECT

Pno	Pname	Duration
121	P1	10
122	P2	20
123	P3	30

$\Pi_{sname,\ student.\ pno,\ pname}$ (STUDENT $\bowtie$ PROJECT)

Invalid Query

Equivalent SQL Query

```
SQL > select distinct sname, student.pno, pname from student natural join
project;                    [SQL 99 syntax]
```

Equivalence Rule

If R(A, B, C) and S(C, D, E) are two relations

then

$$\Pi_{R.C, A, B, D, E}(\sigma_{condition}(R \times S)) \equiv R \bowtie S$$

Or

$$\Pi_{R.C, A, B, D, E}(R \bowtie_{condition} S) \equiv R \bowtie S$$

Natural Join $(R \bowtie S)$

STUDENT

Rollno	Sname	Sem	Branch	Marks	Pno
1	RAM	3	CSE	40	121
2	SHYAM	5	CSE	50	122
3	MOHAN	7	CSE	55	123
4	GOPAL	5	IT	65	121
5	RINKI	3	MECH	40	122
6	PINKI	3	ETC	90	123

PROJECT

Pno	Pname	Duration
121	P1	10
122	P2	20
123	P3	30

$$\Pi_{sname,\ pname}(\sigma_{duration\ >\ 15}(STUDENT \bowtie PROJECT))$$

Sname	Pname
SHYAM	P2
RINKI	P2
MOHAN	P3
PINKI	P3

Equivalent SQL Query

SQL > select distinct sname, pname from student natural join project where duration > 15; [SQL 99 syntax]

Natural Join $(\mathbf{R} \bowtie \mathbf{S})$

STUDENT

Rollno	Sname	Sem	Branch	Marks	Pno
1	RAM	3	CSE	40	121
2	SHYAM	5	CSE	50	122
3	MOHAN	7	CSE	55	123
4	GOPAL	5	IT	65	121
5	RINKI	3	MECH	40	122
6	PINKI	3	ETC	90	123

PROJECT

Pno	Pname	Duration
121	P1	10
122	P2	20
123	P3	30

$$\Pi_{\mathbf{sname}}\left(\sigma_{\mathbf{pname}\ =\ '\mathbf{P2}'}\left(\mathbf{STUDENT} \bowtie \mathbf{PROJECT}\right)\right)$$

Sname
SHYAM
RINKI

Equivalent SQL Query

SQL > select distinct sname from student natural join project where pname = ‘P2’; [SQL 99 syntax]

Equi Join $(R \bowtie S)$

STUDENT

Rollno	Sname	Sem	Branch	Marks	Pno
1	RAM	3	CSE	40	121
2	SHYAM	5	CSE	50	122
3	MOHAN	7	CSE	55	123
4	GOPAL	5	IT	65	121
5	RINKI	3	MECH	40	122
6	PINKI	3	ETC	90	123

PROJECT

Pno	Pname	Duration
121	P1	10
122	P2	20
123	P3	30

$$\Pi_{sname}(\sigma_{student.pno = project.pno \wedge pname = 'P2'}(STUDENT \times PROJECT))$$

Or

$$\Pi_{sname}(\sigma_{pname = 'P2'}(STUDENT \bowtie_{student.pno = project.pno} PROJECT))$$

Sname
SHYAM
RINKI

Equivalent SQL Query

```
SQL > select distinct sname from student cross join project where student.pno and project.pno and pname = ‘P2’;
```

[SQL 99 syntax]

```
SQL > select distinct sname from student inner join project on student.pno = project.pno where pname = ‘P2’;
```

Natural Join (R ⋈ S)

STUDENT

Rollno	Sname	Sem	Branch	Marks	Pno
1	RAM	3	CSE	40	121
2	SHYAM	5	CSE	50	122
3	MOHAN	7	CSE	55	123
4	GOPAL	5	IT	65	121
5	RINKI	3	MECH	40	122
6	PINKI	3	ETC	90	123

PROJECT

Pname	Duration
P1	10
P2	20
P3	30

STUDENT ⋈ PROJECT

Rollno	Sname	Sem	Branch	Marks	Pno	Pname	Duration
1	RAM	3	CSE	40	121	P1	10
2	SHYAM	5	CSE	50	122	P1	10
3	MOHAN	7	CSE	55	123	P1	10
4	GOPAL	5	IT	65	121	P1	10
5	RINKI	3	MECH	40	122	P1	10
6	PINKI	3	ETC	90	123	P1	10
1	RAM	3	CSE	40	121	P2	20
2	SHYAM	5	CSE	50	122	P2	20
3	MOHAN	7	CSE	55	123	P2	20
4	GOPAL	5	IT	65	121	P2	20
5	RINKI	3	MECH	40	122	P2	20
6	PINKI	3	ETC	90	123	P2	20
1	RAM	3	CSE	40	121	P3	30
2	SHYAM	5	CSE	50	122	P3	30
3	MOHAN	7	CSE	55	123	P3	30
4	GOPAL	5	IT	65	121	P3	30
5	RINKI	3	MECH	40	122	P3	30
6	PINKI	3	ETC	90	123	P3	30

Equivalent SQL Query

SQL > select distinct sname from student natural join project;

[SQL 99 syntax]

Equivalence Rule

If R(A, B, C) and S(D, E) are two relations

then

$$R \times S \equiv R \bowtie S$$

Difference between Natural join and Equi join

S. NO.	NATURAL JOIN	EQUI JOIN
1.	Natural Join joins two tables based on same attribute name and datatypes.	Equi Join joins two table on the basis of the column which is explicitly specified in the ON clause.
2.	In Natural Join, The resulting table will contain all the attributes of both the tables but keep only one copy of each common column.	In Equi Join, The resulting table will contain all the attribute of both the tables including duplicate columns also.
3.	In Natural Join, common attribute between two relation should be mentioned directly by name.	In Equi Join, duplicate occurrence of attribute should be mentioned by relational_name.attribute_name.

4.	SYNTAX :
	SELECT *
	FROM table1 NATURAL JOIN table2;
		SYNTAX :
		SELECT *
		FROM table1 INNER JOIN table2 ON
		table1.Column_Name = table2.Column_Name;

For Video lecture on this topic please subscribe to my youtube channel.

The link for my youtube channel is

https://www.youtube.com/channel/UCRWGtE76JlTp1iim6aOTRuw?sub_confirmation=1