UNIT 2 Lecture 18 Relational Algebra Joins (Part — II)

R

A	В
1	2
3	4

S

С	D
5	6
7	8

 $\mathbf{R} \times \mathbf{S}$

A	В	С	D
1	2	5	6
3	4	5	6
1	2	7	8
3	4	7	8

 $R \bowtie S$

A	В	С	D
1	2	5	6
3	4	5	6
1	2	7	8
3	4	7	8

Equivalent SQL Query (For Cross Join)

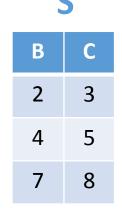
SQL > select * from R cross join S;

SQL > select * from R, S;

[SQL 99 syntax]

Equivalent SQL Query (For Natural Join) SQL > select * from R natural join S;

R		
Α	В	
1	2	
3	4	
E	6	



В	A	С				
2	1	3				
4	3	5				

 $R \bowtie S$

Equivalent SQL Query (For EQUI Join)

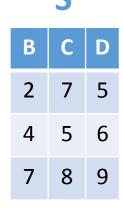
SQL > select R.B, A, C from R inner join S on R.B = S.B;

SQL > select R.B, A, C from R, S where R.B = S.B;

[SQL 99 syntax]

Equivalent SQL Query (For Natural Join) SQL > select * from R natural join S;

R							
A	В	С					
1	2	7					
3	4	3					
5	6	5					



В	С	A	D
2	7	1	5

 $R \bowtie S$

Equivalent SQL Query (For EQUI Join)

SQL > select R.B, R.C, A, D from R inner join S on R.B = S.B and R.C = S.C;

[SQL 99 syntax]

SQL > select R.B, R.C, A, D from R, S where R.B = S.B and R.C = S.C;

Equivalent SQL Query (For Natural Join) SQL > select * from R natural join S;

	R		S				R	M	S
A	В	С	A	В	С		A	В	С
1	2	7	1	2	7		1	2	7
3	4	3	4	5	6				
5	6	5	7	8	9				

Equivalent SQL Query (For EQUI Join)
SQL > select R.A, R.B, R.C from R inner join S
on R.A = S.A and R.B = S.B and R.C = S.C;

[SQL 99 syntax]

SQL > select R.A, R.B, R.C from R, S where R.A = S.A and R.B = S.B and R.C = S.C;

Equivalent SQL Query (For Natural Join) SQL > select * from R natural join S;

Intersection $(R \cap S)$

R				S		$\mathbf{R} \cap$			
	A	В	С	A	В	С	A	В	C
	1	2	7	1	2	7	1	2	7
	3	4	3	4	5	6			
	5	6	5	7	8	9			

Equivalent SQL Query SQL > select * from R intersect select * from S;

Equivalence Rule

If R(A,B,C) and S(A,B,C) then

$$R \cap S \equiv R \bowtie S$$

Outer Joins

1. Left Outer Join (R=⋈S)

Left outer join = Natural join + rows left in the left table.

2. Right Outer Join $(R \bowtie = S)$

Right outer join = Natural join + rows left in the right table.

3. Full Outer Join ($R = \bowtie = S$)

Full outer join = Natural join + rows left in the left table + rows left in the right table.

Left Outer Join ($R = \bowtie S$)

R	R		R S			$\mathbf{R}\bowtie\mathbf{S}$			$\mathbf{R} = \bowtie \mathbf{S}$		
Α	В		В	С	В	A	С		В	A	С
1	2		2	3	2	1	3		2	1	3
3	4		4	5	4	3	5		4	3	5
5	6		7	8					6	5	

Equivalent SQL Query

SQL > select * from R left natural join S;

[SQL 99 Syntax]

SQL > select R.B, A, C from R left join S on R.B = S.B;

[SQL 99 Syntax]

SQL > select R.B, A, C from R, S where R.B = S.B(+);

[Old Syntax]

Right Outer Join ($R \bowtie = S$)

R					
A	В				
1	2				
3	4				
_					

6

В	C
2	3
4	5
7	8

$R \bowtie S$			R	 =	= 5	5	
	В	A	С	В	A	С	
	2	1	3	2	1	3	
	4	3	4	4	3	5	

В	A	С
2	1	3
4	3	5
7		8

Equivalent SQL Query

SQL > select * from R right natural join S;

[SQL 99 Syntax]

SQL > select R.B, A, C from R right join S on R.B = S.B;

[SQL 99 Syntax]

SQL > select R.B, A, C from R, S where R.B(+) = S.B;

[Old Syntax]

Full Outer Join ($R = \bowtie = S$)

A	В
1	2
3	4
5	6

C
J

В	С
2	3
4	5
7	8

$$R \bowtie S$$

В	A	С
2	1	3
4	3	4

$$R \bowtie S$$
 $R = \bowtie = S$

В	Α	С
2	1	3
4	3	5
6	5	
7		8

Equivalent SQL Query

SQL > select * from R full natural join S;

[SQL 99 Syntax]

SQL > select R.B, A, C from R full join S on R.B = S.B;

[SQL 99 Syntax]

SQL > select R.B, A, C from R, S where R.B(+) = S.B(+);

[Wrong Syntax]

Left Outer Join ($R = \bowtie S$)

STUDENT

PROJECT

STUDENT = ⋈ student pno = project pno

PROJECT

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

Pname	Duration
P1	10
P2	20
P3	30
P4	40
	P1 P2 P3

Rollno	Sname	Sem	Branch	Marks	Pno	Pno	Pname	Duration
1	RAM	3	CSE	40	121	121	P1	10
2	SHYAM	5	CSE	50	122	122	P2	20
3	MOHAN	7	CSE	55	123	123	Р3	30
4	GOPAL	5	IT	65	121	121	P1	10
5	RINKI	3	MECH	40	122	122	P2	20
6	PINKI	3	ETC	90	123	123	Р3	30

Equivalent SQL Query SQL > select * from student Left join project on student.pno = project.pno; [SQL 99 syntax]

SQL > select * from student, project where student.pno = project.pno(+);

Right Outer Join ($R \bowtie = S$)

STUDENT

PROJECT

STUDENT ⋈= student pno = project pno PROJECT

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

Pno	Pname	Duration
121	P1	10
122	P2	20
123	Р3	30
124	P4	40

Rollno	Sname	Sem	Branch	Marks	Pno	Pno	Pname	Duration
1	RAM	3	CSE	40	121	121	P1	10
2	SHYAM	5	CSE	50	122	122	P2	20
3	MOHAN	7	CSE	55	123	123	Р3	30
4	GOPAL	5	IT	65	121	121	P1	10
5	RINKI	3	MECH	40	122	122	P2	20
6	PINKI	3	ETC	90	123	123	Р3	30
						124	P4	40

Equivalent SQL Query
SQL > select * from student
right join project on student.pno =
project.pno; [SQL 99 syntax]

SQL > select * from student, project
where student.pno(+) = project.pno;

Full Outer Join ($R = \bowtie = S$)

STUDENT

PROJECT

STUDENT=⋈= student.pno = project.pno PRO	DJECT
--	--------------

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

Pno	Pname	Duration
121	P1	10
122	P2	20
123	Р3	30
124	P4	40

Rollno	Sname	Sem	Branch	Marks	Pno	Pno	Pname	Duration
1	RAM	3	CSE	40	121	121	P1	10
2	SHYAM	5	CSE	50	122	122	P2	20
3	MOHAN	7	CSE	55	123	123	Р3	30
4	GOPAL	5	IT	65	121	121	P1	10
5	RINKI	3	MECH	40	122	122	P2	20
6	PINKI	3	ETC	90	123	123	Р3	30
						124	P4	40

Equivalent SQL Query
SQL > select * from student
full join project on student.pno = project.pno;

[SQL 99 syntax]

SQL > select * from student, project where student.pno(+) = project.pno(+);
[Wrong Query]

Rename Operator (ρ)

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

Syntax:

$$\rho_{s}(R)$$

Example 1: Rename the student table to stud

$$ho_{\mathsf{STUD}}(\mathsf{STUDENT})$$

Example 2 : Rename attribute names

$$\rho_{\text{rn, sn, s, b, m, pn}}$$
 (STUDENT)

Example 3: Rename relation name as well as attibutes

$$ho_{STUD (rn, sn, s, b, m, pn)}$$
 (STUDENT)

Self Join

STUDENT (s1)

STUDENT (s2)

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

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

OUTPUT

Sname RAM GOPAL

$$\sqcap_{s1 \text{ sname}} (\sigma_{s1 \text{ pno}} = s2 \text{ pno} \land s2 \text{ sname} = \text{'RAM'} (\rho_{s1} (STUDENT) \times \rho_{s2} (STUDENT)))$$

Equivalent SQL Query

SQL > select s1.sname from student s1, student s2 where s1.pno = s2.pno and s2.sname = 'RAM';
Or

SQL > select s1.sname from student s1 inner join student s2 on s1.pno = s2.pno where s2.sname = 'RAM'; Or

SQL > select sname from student where pno in (select pno from student where sname = 'RAM');

Self Join

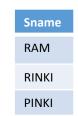
STUDENT (s1)

STUDENT (s2)

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

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

OUTPUT



$$\sqcap_{s1 \text{ sname}} (\sigma_{s1 \text{ sem}} = s2 \text{ sem} \land s2 \text{ sname} = 'RAM' (\rho_{s1} (STUDENT)) \times \rho_{s2} (STUDENT)))$$

Equivalent SQL Query

SQL > select s1.sname from student s1, student s2 where s1.sem = s2.sem and s2.sname = 'RAM';
Or

SQL > select s1.sname from student s1 inner join student s2 on s1.sem = s2.sem where s2.sname = 'RAM';
Or

SQL > select sname from student where sem in (select sem from student where sname = 'RAM');

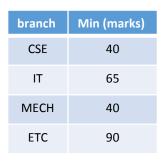
Aggregate Functions (G) and Group by clause

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

Min (marks)
40

Sum (marks) 340



Syntax:

G aggregate fun (R)

Example 1: Display the min marks scored by students.

ල min(marks) (STUDENT)

SQL > select min(marks) from student;

Example 2 : Display the sum of marks obtained by students.

G_{sum(marks)}(STUDENT)

SQL > select sum(marks) from student;

Example 3: Display the branch wise minimum marks

branch \$\mathref{G}_{\text{min(marks)}}\$(STUDENT)

SQL > select branch, min(marks) from student group by branch;

Dinesh Kumar Bhawnani, BIT DURG

Self Join

STUDENT (s1)

STUDENT (s2)

				Ŧ
U	U		U	

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

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



□ sname (STUDENT)

 $\sqcap_{s1,sname}$ ($\sigma_{s1,marks} > s2,marks$ (ρ_{s1} (STUDENT) × ρ_{s2} (STUDENT)))

Equivalent SQL Query

SQL > select sname from student where marks in (select min(marks) from student);

Or

SQL > select sname from student minus

select s1.sname from student s1, student s2 where s1.marks > s2.marks;

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