# UNIT 2

## Lecture 19

## Relational Algebra

## Division Operation

# Intersection example

Display the name of branches in which project 121 and project 122 both are running.

RA : $\Pi_{branch}(\sigma_{pno = 121}(STUDENT)) \cap \Pi_{branch}(\sigma_{pno = 122}(STUDENT))$

```
SQL > select distinct branch from student where pno = 121
      intersect
      select distinct branch from student where pno = 122;
```

## 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 |

| Branch |
|---|
| CSE |
| IT |

| Branch |
|---|
| CSE |
| MECH |

## OUTPUT

| Branch |
|---|
| CSE |

# Intersection example

Display the name of branches in which all the projects are running.

RA : $\Pi_{branch}(\sigma_{pno = 121}(STUDENT))$

$\cap$

$\Pi_{branch}(\sigma_{pno = 122}(STUDENT))$

$\cap$

$\Pi_{branch}(\sigma_{pno = 123}(STUDENT))$

**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 |

| Branch |
|---|
| CSE |
| IT |

| Branch |
|---|
| CSE |
| MECH |

| Branch |
|---|
| CSE |
| ETC |

**OUTPUT**

| Branch |
|---|
| CSE |

# Division Operation (r ÷ s)

- The **division** operation, denoted by ÷, is suited to queries that include the phrase "for all."
- Formally, let r(R) and s(S) be relations, and let S ⊆ R; that is, every attribute of schema S is also in schema R. The relation r ÷ s is a relation on schema R − S (that is, on the schema containing all attributes of schema R that are not in schema S).
- A tuple t is in r ÷ s if and only if both of two conditions hold:
  1. t is in $\Pi_{R-S}(r)$
  2. For every tuple $t_s$ in s, there is a tuple $t_r$ in r satisfying both of the following:
     - **a.** $t_r[S] = t_s[S]$
     - **b.** $t_r[R - S] = t$
- We can define the division operation in terms of the fundamental operations.
- Let r(R) and s(S) be given, with S ⊆ R:

$$r \div s = \Pi_{R-S}(r) - \Pi_{R-S}((\Pi_{R-S}(r) \times s) - \Pi_{R-S,S}(r))$$

# Division Operation (r ÷ s)

**A**

| X | Y |
|---|---|
| X1 | Y1 |
| X1 | Y2 |
| X1 | Y3 |
| X1 | Y4 |
| X2 | Y1 |
| X2 | Y2 |
| X3 | Y2 |
| X4 | Y2 |
| X4 | Y4 |

**B**

| Y |
|---|
| Y2 |

**C**

| Y |
|---|
| Y2 |
| Y4 |

**D**

| Y |
|---|
| Y1 |
| Y2 |
| Y4 |

**A ÷ B**

| X |
|---|
| X1 |
| X2 |
| X3 |
| X4 |

**A ÷ C**

| X |
|---|
| X1 |
| X4 |

**A ÷ D**

| X |
|---|
| X1 |

# Division Operation (r ÷ s)

R

| A | B |
|---|---|
| 1 | 2 |
| 3 | 4 |

S

| C | D |
|---|---|
| 5 | 6 |
| 7 | 8 |

T = R × S

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

T = R × S

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

T ÷ S

| A | B |
|---|---|
| 1 | 2 |
| 3 | 4 |

T ÷ R

| C | D |
|---|---|
| 5 | 6 |
| 7 | 8 |

# Division Operation (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

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

$$\Pi_{branch,\ pno}(STUDENT) \div \Pi_{pno}(PROJECT)$$

## OUTPUT

| Branch |
|---|
| CSE |

**Equivalent SQL Query**

```
SQL > select distinct branch from student s1
where not exists (select pno from project p
where not exists (select branch from student s2
where s1.branch = s2.branch and p.pno = s2.pno));
```

# Division Operation (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

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

$\Pi_{branch,\ pno}(STUDENT) \div \Pi_{pno}(PROJECT)$

## OUTPUT

| Branch |
|---|
| CSE |

**Equivalent SQL Query**

```
SQL >select distinct branch from student s1
where not exists (select pno from project
where pno not in (select s2.pno from student s2, project p
where s1.branch = s2.branch and p.pno = s2.pno));
```

# Division Operation (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

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

$\Pi_{branch,\ pno}(STUDENT) \div \Pi_{pno}(PROJECT)$

## OUTPUT

| Branch |
|---|
| CSE |

**Equivalent SQL Query**

```
SQL >select distinct branch from student s1
where not exists (select
minus
select s2.pno from student s2, project p
where s1.branch = s2.branch and p.pno = s2.pno);
```

# Division Operation (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

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

$$\Pi_{branch,\ pno}(STUDENT) \div \Pi_{pno}(PROJECT)$$

## OUTPUT

| Branch |
|---|
| CSE |

**Equivalent SQL Query**

**SQL >select distinct branch from student group by branch having count(pno) = (select count(pno) from project);**

# Division Operation (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

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

$\Pi_{sem, pno}(STUDENT) \div \Pi_{pno}(PROJECT)$

## OUTPUT

| Sem |
|---|
| 3 |

**Equivalent SQL Query**

**SQL >select distinct sem from student group by sem having count(pno) = (select count(pno) from project);**

# The Assignment Operation (←)

- It is convenient at times to write a relational-algebra expression by assigning parts of it to temporary relation variables.
- The **assignment** operation, denoted by ←, works like assignment in a programming language.
- For e.g., consider the definition of division. We could write $r \div s$ as

$$temp1 \leftarrow \Pi_{R-S}(r)$$

$$temp2 \leftarrow \Pi_{R-S}((temp1 \times s) - \Pi_{R-S,\, S}(r))$$

$$result = temp1 - temp2$$

- The evaluation of an assignment does not result in any relation being displayed to the user. Rather, the result of the expression to the right of the ← is assigned to the relation variable on the left of the ←. This relation variable may be used in subsequent expressions.

# Self Join

## STUDENT (s1)

| 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 |

## 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 |

## OUTPUT

| Marks |
|---|
| RAM |
| RINKI |

$$\Pi_{sname}(STUDENT) - \Pi_{s1.sname}(\sigma_{s1.marks > s2.marks}(\rho_{s1}(STUDENT) \times \rho_{s2}(STUDENT)))$$

**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;**

# Using assignment operator (←)

## STUDENT (s1)

| 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

| Marks |
|---|
| RAM |
| RINKI |

**Equivalent SQL Query**

```
SQL > with r1 as
(select min(marks) as mm
from student)
select sname from student, r1
where marks = mm;
```

$$R1 \leftarrow \rho_{mm}(\mathcal{G}_{min(marks)}(STUDENT))$$

$$RESULT \leftarrow \Pi_{sname}(\sigma_{marks = mm}(STUDENT \times R1))$$

# Using assignment operator (←)

## STUDENT (s1)

| 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

| Marks |
|---|
| RAM |
| RINKI |

**Equivalent SQL Query**

```
SQL > with r1(mm) as
(select min(marks) from
student)
select sname from student, r1
where marks = mm;
```

$$R1(mm) \leftarrow \mathcal{G}_{min(marks)}(STUDENT)$$

$$RESULT \leftarrow \Pi_{sname}(\sigma_{marks = mm}(STUDENT \times R1))$$

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