RELATIONAL QUERY LANGUAGES

UNIT 3

Prepared By
Prof. Rutika Patel

Structure of Relational Databases

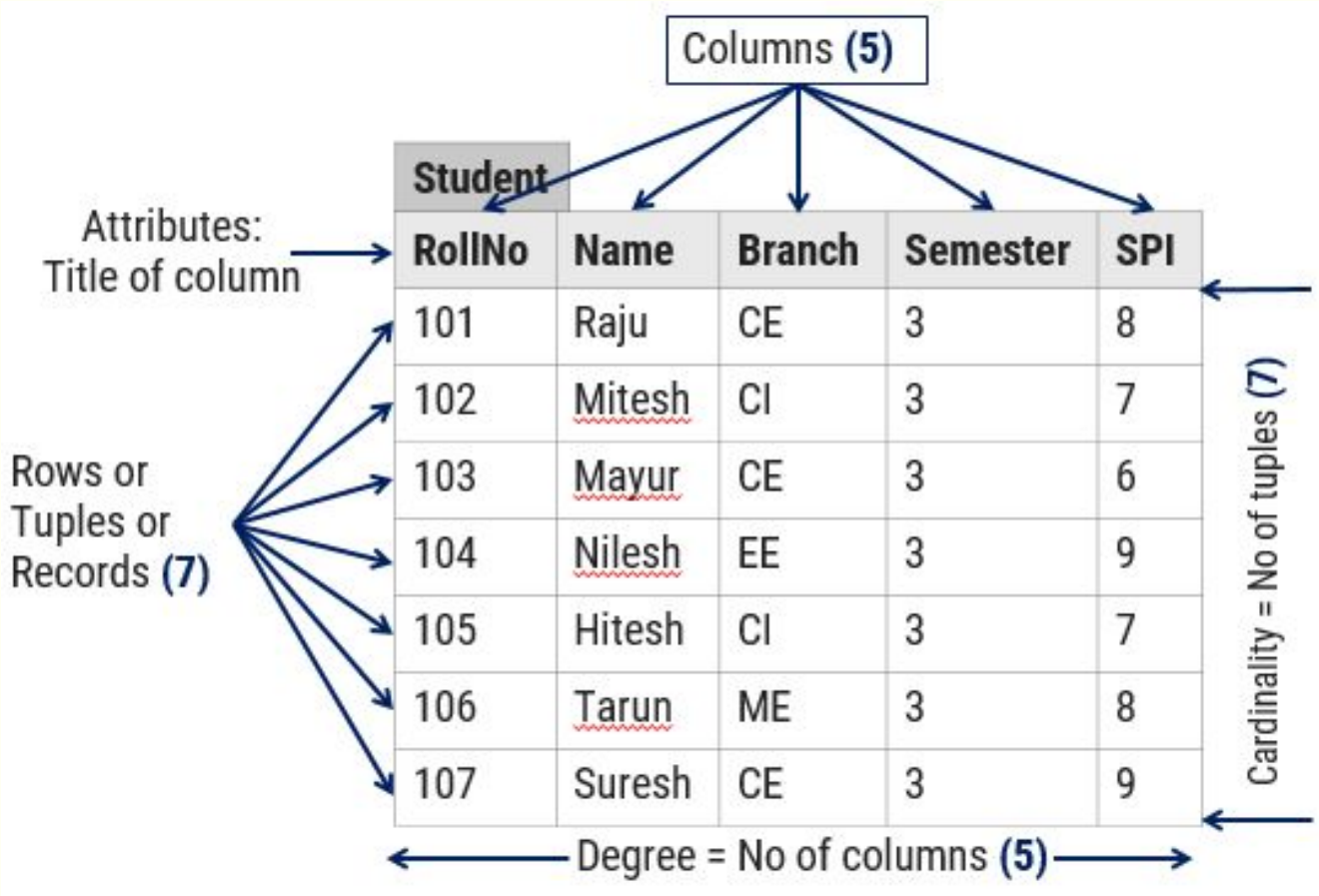


Table (Relation): A database object that holds a collection of data for a specific topic.
Table consist of rows and columns.

Column (Attribute): The vertical component of a table. A column has a name and a particular data type; e.g. varchar, decimal, integer, datetime etc.

Record (Tuple): The horizontal component of a table, consisting of a sequence of values, one for each column of the table. It is also known as row.

A database consists of a collection of tables (relations), each having a unique name.

Domain is a set of **all possible unique values** for a specific column. Domain of Branch attribute is (CE, CI, ME, EE)

Relational Algebra Operations

Operator	Description
Selection	Display particular rows/records/tuples from a relation
Projection	Display particular columns from a relation
Cross Product	Multiply each tuples of both relations
Joins	Combine data or records from two or more tables 1. Natural Join / Inner Join 2. Outer Join 1. Left Outer Join 2. Right Outer Join 3. Full Outer Join
Set Operators	Combine the results of two queries into a single result. 1. Union 2. Intersection 3. Minus / Set-difference
Division	Divides one relation by another
Rename	Rename a column or a table

1) Selection Operator

Symbol	σ (Sigma)
Notation	$\sigma_{condition}$ (Relation)
Operation	**Selects tuples** from a relation that **satisfy a given condition**.
Operators	=,<, >, <=, >=, Λ (AND), V (OR)

Example: Display the detail of students belongs to "CE" Branch.

Answer: $\sigma_{Branch='CE'}$ (Student)

Student

RollNo	Name	Branch	SPI
101	Riha	CE	8
102	Misha	ME	9
103	Nitin	CI	9
104	Vedant	CE	9

Output

RollNo	Name	Branch	SPI
101	Riha	CE	8
104	Vedant	CE	9

Cont…

- **Example:** Display the detail of students belongs to “CE” Branch and having SPI more than 8.

Student

RollNo	Name	Branch	SPI
101	Riha	CE	8
102	Misha	ME	9
103	Nitin	CI	9
104	Vedant	CE	9

- **Answer:** $\sigma_{Branch='CE' \wedge SPI>8}$ (Student) [σ condition (Relation)]

Output

RollNo	Name	Branch	SPI
101	Riha	CE	8
104	Vedant	CE	9

Cont..

Example: Display the detail of students belongs to either “CI” or “ME” Branch.

Student

RollNo	Name	Branch	SPI
101	Riha	CE	8
102	Misha	ME	9
103	Nitin	CI	9
104	Vedant	CE	9

- **Answer:** $\sigma_{Branch='CI' \ V \ Branch='ME'}$ (Student)

Output

RollNo	Name	Branch	SPI
102	Misha	ME	9
103	Nitin	CI	9

Cont..

Example: Display the detail of students whose SPI between 7 and 9.

Student

RollNo	Name	Branch	SPI
101	Riha	CE	8
102	Misha	ME	9
103	Nitin	CI	9
104	Vedant	CE	9

Answer: $\sigma_{SPI>7 \wedge SPI<9}$ (Student)

Output

RollNo	Name	Branch	SPI
101	Riha	CE	8

Projection Operator

Symbol	∏ (Pi)
Notation	$\prod_{attribute\ set}$ (Relation)
Operation	**Selects specified attributes** of a relation.

- It **removes duplicate tuples** (records) from the result.

Example: Display RollNo, Name and Branch of all students.

Answer: $\prod_{RollNo,\ Name,\ Branch}$ (Student)

Student

RollNo	Name	Branch	SPI
101	Raju	CE	8
102	Mitesh	ME	9
103	Nilesh	CI	9
104	Meet	CE	9

Output

RollNo	Name	Branch
101	Raju	CE
102	Mitesh	ME
103	Nilesh	CI
104	Meet	CE

Combined Projection & Selection Operation

Example: Display RollNo, Name & Branch of "ME" Branch students.

Student

RollNo	Name	Branch	SPI
101	Raju	CE	8
102	Mitesh	ME	9
103	Nilesh	CI	9
104	Meet	CE	7

Step: 1 $\sigma_{Branch='ME'}$ (Student)

Output-1

RollNo	Name	Branch	SPI
102	Mitesh	ME	9

Output: $\prod_{RollNo,\ Name,\ Branch}$ ($\sigma_{Branch='ME'}$ (Student))

Output-2

RollNo	Name	Branch
102	Mitesh	ME

Cont..

Example: Display Name, Branch and SPI of students whose SPI is more than 8.

Student

RollNo	Name	Branch	SPI
101	Raju	CE	8
102	Mitesh	ME	9
103	Nilesh	CI	9
104	Meet	CE	7

Step-1: $\sigma_{SPI>8}$ (Student)

Answer: $\prod_{Name, Branch, SPI} (\sigma_{SPI>8}$ (Student))

Output-1

RollNo	Name	Branch	SPI
102	Mitesh	ME	9
103	Nilesh	CI	9

Output-2

Name	Branch	SPI
Mitesh	ME	9
Nilesh	CI	9

Cont…

Example: Display Name, Branch and SPI of students who belongs to "CE" Branch and SPI is more than 7.

Student

RollNo	Name	Branch	SPI
101	Raju	CE	8
102	Mitesh	ME	9
103	Nilesh	CI	9
104	Meet	CE	7

Step-1: $\sigma_{Branch='CE' \wedge SPI>7}$ (Student)

Output-1

RollNo	Name	Branch	SPI
101	Raju	CE	8

Answer: $\prod_{Name, Branch, SPI} (\sigma_{Branch='CE' \wedge SPI>7}$ (Student))

Output-2

Name	Branch	SPI
Raju	CE	8

Cont..

Example: Display Name of students along with their Branch who belong to either “ME” Branch or “CI” Branch.

Student

RollNo	Name	Branch	SPI
101	Raju	CE	8
102	Mitesh	ME	9
103	Nilesh	CI	9
104	Meet	CE	7

Step-1: $\sigma_{Branch='ME' \ V \ Branch='CI'}$ (Student) **Answer:** $\prod_{Name,\ Branch}$ ($\sigma_{Branch='ME' \ V \ Branch='CI'}$ (Student))

Output-1

RollNo	Name	Branch	SPI
102	Mitesh	ME	9
103	Nilesh	CI	9

Output-2

Name	Branch
Mitesh	ME
Nilesh	CI

Cartesian Product / Cross Product

Symbol	X (Cross)
Notation	*Relation-1 (R1)* X *Relation-2 (R2)* **OR** *Algebra-1* X *Algebra-2*
Operation	It will **multiply each tuples** of Relation-1 to each tuples of Relation-2. • Attributes of Resultant Relation = Attributes of R1 + Attributes of R2 • Tuples of Resultant Relation = Tuples of R1 * Tuples of R2

Example: Perform Cross Product between Student and Result.

Ans: (Student) X (Result)

Student

RNo	Name	Branch
101	Riva	CE
102	Kiran	ME

Result

RNo	SPI
101	8
102	9

Output

Student. RNo	Name	Branch	Result. RNo	SPI
101	Riva	CE	101	8
101	Riva	CE	102	9
102	Kiran	ME	101	8
102	Kiran	ME	102	9

If both relations have some attribute with the same name, it can be distinguished by combing **relation-name.attribute-name**.

Cont..

Example: Perform Cross Product between Student and Result. (Consider only selected attributes Student – RNo, Name and Branch, Result – RNo, SPI and BL)

Student

RNo	Name	Branch	Sem
101	Riva	CE	3
102	Kiran	ME	5

Result

RNo	SPI	BL	Rank
101	8	1	2
103	9	0	1

Ans: $\prod_{RNo,\ Name,\ Branch}$ (Student) X $\prod_{RNo,\ SPI,\ BL}$ (Result)

Output

Student. RNo	Name	Branch	Result. RNo	SPI	BL
101	Riva	CE	101	8	1
101	Riva	CE	103	9	0
102	Kiran	ME	101	8	1
102	Kiran	ME	103	9	0

Cont…

Example: Perform Cross Product between Student and Result. Consider only selected tuples Student – Branch='CE' and Sem=3, Result – SPI>7 and BL<1

Student

RNo	Name	Branch	Sem
101	Riva	CE	3
102	Kiran	ME	5
103	Om	CE	3
104	Dhara	CE	5

Result

RNo	SPI	BL	Rank
101	8	1	2
103	9	0	1
105	7	2	3

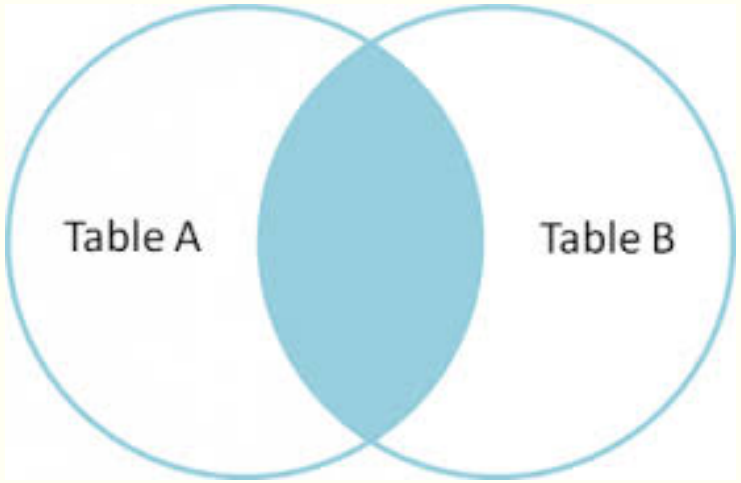
Ans: $\sigma_{Branch='CE' \wedge Sem=3}$ (Student) X $\sigma_{SPI>7 \wedge BL<1}$ (Result)

Output

Student. RNo	Name	Branch	Sem	Result. RNo	SPI	BL	Rank
101	Riva	CE	3	103	9	0	1
103	OM	CE	3	103	9	0	1

Natural Join / Inner Join

Symbol	⋈
Notation	*Relation-1 (R1)* ⋈ *Relation-2 (R2)* **OR** *Algebra-1* ⋈ *Algebra-2*
Operation	Natural join will **retrieve consistent data** from multiple relations. • It **combines records** from different relations that **satisfy a given condition**.



Cont..

- **Steps performed in Natural Join:**

Steps	Description
Step – 1	It performs Cartesian Product
Step – 2	Then it deletes inconsistent tuples
Step – 3	Then it removes an attribute from duplicate attributes

Cont..

Example: Perform Natural Join between Student and Result.

Student

RNo	Name	Branch
101	Riva	CE
102	Kiran	ME

Result

RNo	SPI
101	8
103	9

Steps performed in Natural Join:

Step:1 Perform Cross Product

Student.RNo	Name	Branch	Result.RNo	SPI
101	Riva	CE	101	8
101	Riva	CE	103	9
102	Kiran	ME	101	8
102	Kiran	ME	103	9

Cont..

Step:2 Removes inconsistent tuples

Student.RNo	Name	Branch	Result.RNo	SPI
101	Riva	CE	101	8

Step:3 Removes an attribute from duplicate

RNo	Name	Branch	SPI
101	Riva	CE	8

Result: (Student) ⋈ (Result)

Output

RNo	Name	Branch	SPI
101	Riva	CE	8

Outer Join

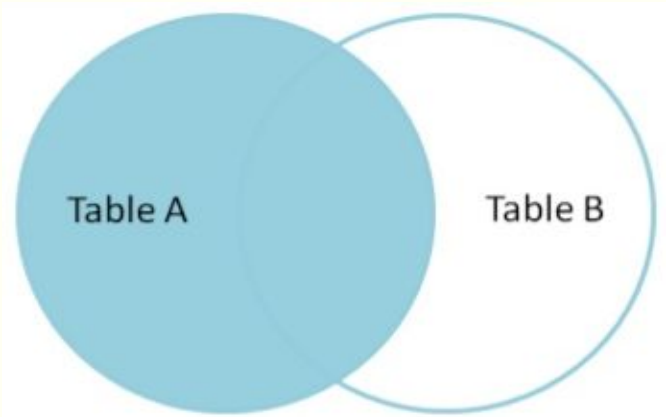
- In natural join some records are missing, if we want that missing records than we have to use outer join.
- Three types of Outer Join

Sr.	Outer Join	Symbol
1	Left Outer Join	
2	Right Outer Join	
3	Full Outer Join	

To perform a Outer Join there must be **one common attribute (column)** between two relations.

1) Left Outer Join

Symbol	⟕
Notation	*Relation-1 (R1)* ⟕ *Relation-2 (R2)* OR *Algebra-1* ⟕ *Algebra-2*
Operation	• Display all the tuples of the left relation even through there is no matching tuple in the right relation. • For such kind of tuples having no matching, the attributes of right relation will be padded with NULL in resultant relation.



- **Example:** Perform Left Outer Join between Student and Result.

Student

RollNo	Name	Branch
101	Raj	CE
102	Meet	ME

Result

RollNo	SPI
101	8
103	9

Answer: (Student) ⟕ (Result)

Output

RollNo	Name	Branch	SPI
101	Raj	CE	8
102	Meet	ME	NULL

- **Example:** Perform Left Outer Join between Student and Result. (Display RollNo, Name and SPI)

Student

RollNo	Name	Branch
101	Raj	CE
102	Meet	ME

Result

RollNo	SPI	BL
101	8	1
103	9	0

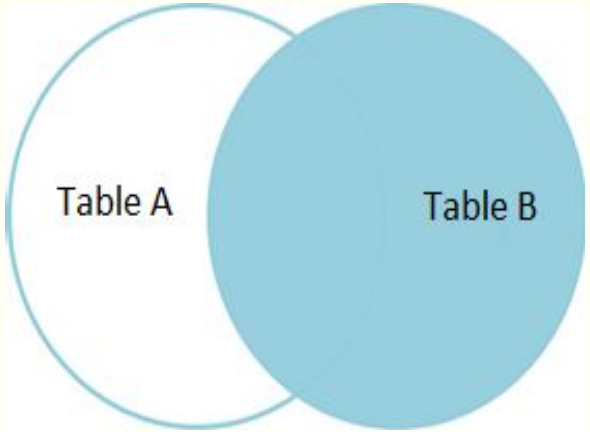
Answer: $\prod_{RollNo,\ Name,\ SPI}$ (Student) $\overline{\bowtie}$(Result)

Output

RollNo	Name	SPI
101	Raj	8
102	Meet	NULL

2) Right Outer Join

Symbol	⟖
Notation	*Relation-1 (R1)* ⟖ *Relation-2 (R2)* OR *Algebra-1* ⟖ *Algebra-2*
Operation	• Display all the tuples of right relation even through there is no matching tuple in the left relation. • For such kind of tuples having no matching, the attributes of left relation will be padded with NULL in resultant relation.



- **Example:** Perform Right Outer Join between Student and Result.

Student

RollNo	Name	Branch
101	Raj	CE
102	Meet	ME

Result

RollNo	SPI
101	8
103	9

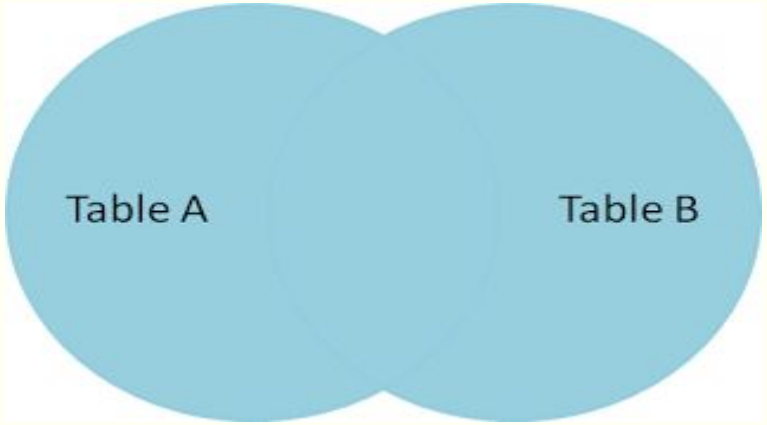
Answer: (Student) ⟖ (Result)

Output

RollNo	Name	Branch	SPI
101	Raj	CE	8
103	NULL	NULL	9

3) Full Outer Join

Symbol	⟗
Notation	*Relation-1 (R1)* ⟗ *Relation-2 (R2)* **OR** *Algebra-1* ⟗ *Algebra-2*
Operation	• Display **all the tuples of both of the relations**. It also pads null values whenever required. (Left outer join + Right outer join). • For such kind of **tuples having no matching**, it will be **padded with NULL** in resultant relation.



- **Example:** Perform Full Outer Join between Student and Result.

Student

RollNo	Name	Branch
101	Raj	CE
102	Meet	ME

Result

RollNo	SPI
101	8
103	9

Answer: (Student) ⟗ (Result)

Output

RollNo	Name	Branch	SPI
101	Raj	CE	8
102	Meet	ME	NULL
103	NULL	NULL	9

Example: Perform Full Outer Join between Student and Result. (Display RollNo, Name and SPI)

Student

RollNo	Name	Branch
101	Raj	CE
102	Meet	ME

Result

RollNo	SPI	BL
101	8	1
103	9	0

Answer: $\prod_{RollNo,\ Name,\ SPI}$ ((Student) ⋈ (Result))

Output

RollNo	Name	SPI
101	Raj	8
102	Meet	**NULL**
103	**NULL**	9

Set Operators

Set operators combine the results of two or more queries into a single result.

Three types of Set operator

Sr No	Set Operator	Symbol
1	Union	U
2	Intersect / Intersection	∩
3	Minus / Set difference	–

Conditions: Set operators will take two or more queries as input, which must be union-compatible.

- Both queries should have **same (equal) number of columns**
- Corresponding **attributes should have the same data type** or **domain**

Conditions to perform Set Operators

Condition 1: Both queries should have same (equal) number of columns.

Student

RNo	Name	Dept	SPI
101	Raj	CE	8
102	Meet	ME	9
103	Jay	CE	9

Faculty

FId	Name	Dept
101	Patel	CE
102	Shah	ME
103	Dave	ME

Student

RNo	Name	Dept
101	Raj	CE
102	Meet	ME
103	Jay	CE

Faculty

FId	Name	Dept
101	Patel	CE
102	Shah	ME
103	Dave	ME

Cont…

Condition 2: Corresponding attributes should have the same data type.

Student

RNo	Name	Dept	SPI
101	Raj	CE	8
102	Meet	ME	9
103	Jay	CE	9

Faculty

FId	Name	Dept	Sub
101	Patel	CE	DS
102	Shah	ME	DBMS
103	Dave	ME	DF

Student

RNo	Name	Dept	SPI
101	Raj	CE	8
102	Meet	ME	9
103	Jay	CE	9

Faculty

FId	Name	Dept	Exp
101	Patel	CE	5
102	Shah	ME	3
103	Dave	ME	4

Cont..

Exercise: Check whether following tables are compatible or not.

- A: (First_name(char), Last_name(char), Date_of_Birth(date))
- B: (FName(char), LName(char), PhoneNumber(number))

- A: (First_name(char), Last_name(char), Date_of_Birth(date))
- B: (FName(char), LName(char), DOB(date))

- Person (PersonID, Name, Address, Hobby)
- Professor (ProfessorID, Name, OfficeAddress, Salary)

Union Operator

Symbol:	U
Notation:	*Relation-1 (R1)* U *Relation-2 (R2)* **OR** *Algebra-1* U *Algebra-2*
Operation:	• It displays all the tuples/records belonging to the first relation (left relation) or the second relation (right relation) or both. • It also eliminates duplicate tuples (tuples present in both relations appear once).

Example: Perform Union between Customer and Employee.

Customer

Name
Raju
Suresh
Meet

Employee

Name
Meet
Suresh
Manoj

Answer: (Customer) U (Employee)

Output

Name
Manoj
Meet
Raju
Suresh

Union Operators Example

Example: Display Name of person who are either employee or customer.

Customer

ID	Name	Balance
1	Raju	10000
2	Suresh	20000

Employee

ID	Name	Dept	Salary
2	Suresh	CE	8000
3	Manoj	ME	9000

Answer: $\prod_{Name}$ (Customer) U $\prod_{Name}$ (Employee)

Output

Name
Manoj
Raju
Suresh

Intersect/ Intersection Operator

Symbol:	∩
Notation:	Relation-1 (R1) ∩ Relation-2 (R2) OR Algebra-1 ∩ Algebra-2
Operation:	• It displays all the tuples/records belonging to both relations. OR • It displays all the tuples/records which are common from both relations.

Example: Perform Intersection between Customer and Employee.

Answer: (Customer) ∩ (Employee)

Customer

Name
Raju
Suresh
Meet

Employee

Name
Meet
Suresh
Manoj

Output

Name
Meet
Suresh

Intersect/ Intersection Operators Example

Example: Display Name of person who are employee as well as customer.

Customer

ID	Name	Balance
1	Raju	10000
2	Suresh	20000

Employee

ID	Name	Dept	Salary
2	Suresh	CE	8000
3	Manoj	ME	9000

Answer: $\prod_{Name}$ (Customer) $\cap$ $\prod_{Name}$ (Employee)

Output

Name
Suresh

Minus/ Set difference Operator

Symbol:	−
Notation:	*Relation-1 (R1)* − *Relation-2 (R2)* **OR** *Algebra-1* − *Algebra-2*
Operation:	It displays all the tuples/records belonging to the first relation (left relation) but not in the second relation (right relation).

Example: Perform Set difference between Customer and Employee

Answer: (Customer) − (Employee)

Customer
Name
Raju
Suresh
Meet

Employee
Name
Meet
Suresh
Manoj

Output
Name
Raju

Minus/ Set difference Operators Example

Example: Display Name of person who are customer but not employee.

Customer

ID	Name	Balance
1	Raju	10000
2	Suresh	20000

Employee

ID	Name	Dept	Salary
2	Suresh	CE	8000
3	Manoj	ME	9000

Answer: $\prod_{Name}$ (Customer) − $\prod_{Name}$ (Employee)

Output

Name
Raju

Division Operator

Symbol:	÷ (Division)
Notation:	Relation1 (R1) ÷ Relation2 (R2) OR Algebra1 ÷ Algebra2
Condition:	Attributes of relation2/algebra2 must be a proper subset of attributes of relation1/algebra1.
Operation:	• The output of the division operator will have attributes = All attributes of relation1 – All attributes of relation2 • The output of the division operator will have tuples = Tuples in relation1, which are associated with the all tuples of relation2.

Division Operator Example

Example: Perform Division operation between Student and Subject.

Answer: (Student) ÷ (Subject)

Student

Name	Subject
Raj	DBMS
Raj	DS
Meet	DS
Meet	DF
Rohit	DBMS
Rohit	DS
Rohit	DF
Suresh	DBMS
Suresh	DF
Suresh	DS

Subject

Subject
DBMS
DS
DF

Output

Name
Rohit
Suresh

Cont..

A

Sno	PNo
S1	P1
S1	P2
S1	P3
S1	P4
S2	P1
S2	P2
S3	P2
S4	P2
S4	P4
S5	P4

B1

PNo
P2

Algebra (A) ÷ (B1)

Output

SNo
S1
S2
S3
S4

B2

PNo
P2
P4

Algebra (A) ÷ (B2)

Output

SNo
S1
S4

B3

PNo
P1
P2
P4

Algebra (A) ÷ (B3)

Output

SNo
S1

B4

PNo
P2
P5

Algebra (A) ÷ (B4)

Output

SNo

Rename Operator

Symbol:	ρ (Rho)
Notation:	$\rho_{A\ (X1,X2....Xn)}$ (Relation)
Operation:	• The rename operation is used to rename the output relation. • The result of rename operator are also relations with new name. • The original relation name can not be changed when we perform rename operation on any relation.

How to use:

$\rho_{x}(E)$

Returns a relation E under a new name X.

$\rho_{A1, A2. \ldots,An}(E)$

Returns a relation E with the attributes renamed to A1, A2, …., An.

$\rho_{x(A1, A2. \ldots,An)}(E)$

Returns a relation E under a new name X with the attributes renamed to A1, A2, …., An.

Cont..

Example Rename table

Student

RNo	Name	CPI
101	Raj	8
102	Meet	9
103	Jay	7

Algebra ρ_{Person} (Student)

Person

RNo	Name	CPI
101	Raj	8
102	Meet	9
103	Jay	7

Example Rename attributes

Student

Rno	Name	CPI
101	Raj	8
102	Meet	9
103	Jay	7

Algebra $\rho_{(RollNo,\ StudentName,\ SPI)}$ (Student)

Student

RollNo	StudentName	SPI
101	Raj	8
102	Meet	9
103	Jay	7

Cont…

Example Rename table and attributes both

Student

Rno	Name	CPI
101	Raj	8
102	Meet	9
103	Jay	7

Algebra $\rho_{Person\ (RollNo,\ StudentName)}\ (\Pi_{RNo,\ Name}\ (Student))$

Person

RollNo	StudentName
101	Raj
102	Meet
103	Jay

Example Rename particular attributes

Student

Rno	Name	CPI
101	Raj	8
102	Meet	9
103	Jay	7

Algebra $\rho_{StudentName\ /\ Name}\ (Student)$

Student

Rno	StudentName	CPI
101	Raj	8
102	Meet	9
103	Jay	7

Aggregate Functions

Symbol:	***g* or G**
Notation:	$g_{\textit{function-name(column), function-name(column), ..., function-name(column)}}$ (Relation)
Operation:	It **takes a more than one value** as input and **returns a single value** as output (result).

Aggregate functions are:

Sum (It **returns the sum (addition)** of the values of a column.)

Max (It **returns the maximum** value for a column.)

Min (It **returns the minimum** value for a column.)

Avg (It **returns the average** of the values for a column.)

Count (It **returns total number** of values in a given column.)

Cont…

Aggregate Functions Example

Student

Rno	Name	Branch	Sem	CPI
101	Ramesh	CE	3	9
102	Mahesh	EC	3	8
103	Suresh	ME	4	7
104	Amit	EE	4	8
105	Anita	CE	4	8
106	Reeta	ME	3	7
107	Rohit	EE	4	9
108	Chetan	CE	3	8
109	Rakesh	CE	4	9

Cont…

- **Example 1:** Find out sum of CPI of all students.

Answer: $g_{sum(CPI)}$ (Student)

Output
sum
73

- **Example 2:** Find out maximum & minimum CPI.

Answer: $g_{max(CPI),\ min(CPI)}$ (Student)

Output	
max	min
9	7

- **Example 3:** Count the number of students.

Answer: $g_{count(Rno)}$ (Student)

Output
count
9

- **Example 4:** Find out average of CPI of all students.

Answer: $g_{avg(CPI)}$ (Student)

Output
avg
8.11

Open source and Commercial DBMS

1. Open Source Database:

- An open-source database is a database where anyone can easily view the source code and this is open and free to download. Also for the community version, some small additional and affordable costs are imposed. Open Source Database provides Limited technical support to end-users. Here Installation and updates are administered by the user. For example: MySQL, MongoDB, SQLite etc

2. Commercial Database :

Commercial database are that which has been created for Commercial Purpose only. They are premium and are not free like Open Source Database. In Commercial Database it is guaranteed that technical support is provided. In this Installation and updates are Administrated by software Vendor. For examples: IBM DB2 etc.

Difference between Open Source Database and Commercial Database

Open Source	Commercial DBMS
• DBMS, which is available in the market at free of cost.	• DBMS, which is available in the market at a certain price.
• The code of open source DBMS product can be viewed, shared or modified by the community.	• The code of commercial DBMS product can not be view, share or modify by the community.
• There are chances of malfunctioning with code as source code is open.	• The security is high and code is not accessible to unauthorized person.
• Examples: MySQL, MongoDB, SQLite etc	• Examples: Microsoft SQL Server, IBM Db2 etc

Questions asked in GTU

1. Define Super key, Primary key, Candidate key and Alternate key.
2. Explain following Relational Algebra Operation with example.
    I. Selection
    II. Projection
    III. Cross Product
    IV. Joins (Inner Join, Outer Joins)
    V. Rename
    VI. Division
    VII. Set operators
3. Explain different aggregate functions with example.

Questions asked in GTU [Relational Algebra]

4. Consider the following relational database, where the primary keys are underlined. Give an expression in the relational algebra to express each of the following queries
   - employee (ssn, name, dno, salary, hobby, gender)
   - department (dno, dname, budget, location, mgrssn)
   - works_on (ssn, pno)
   - project (pno, pname, budget, location, goal)

II. List all pairs of employee names and the project numbers they work on.

III. List out department number, department name and department budget.

IV. List all projects that Raj Yadav works on by project name.

V. List the names of employees who supervise themselves.

Questions asked in GTU [Relational Algebra]

5. Consider the following relational database, where the primary keys are underlined. Give an expression in the relational algebra to express each of the following queries
   - course (course-id, title, dept_name, credits)
   - instructor (id, name, dept_name, salary)
   - section (course-id, sec-id, semester, year, building, room_no, time_slot_id)
   - teaches (id, course-id, sec-id, semester, year)

II. Find the name of all instructors in the physics department.

III. Find all the courses taught in the fall 2009 semester but not in Spring semester.

IV. Find the names of all instructors in the Comp. Sci. department together with the course titles of all the courses that the instructors teach.

V. Find the average salary in each department.

Questions asked in GTU [Relational Algebra]

6. Consider the following relations and write an relational algebra:
   - EMP (empno, ename, jobtitle, managerno, hiredate, sal, commission, deptno)
   - DEPT (deptno, dname, location)

II. Find the Employees working in the department number10, 20, 30 only.

III. Find Employees whose names start with letter A or letter a.

IV. Find Employees along with their department name.

V. Find the Employees who are working in Smith's department

VI. Find the Employees who get salary more than Allen’s salary.

VII. Display employees who are getting maximum salary in each department.

VIII. Find list of employees whose hire date is on or before 1-April-18.

Questions asked in GTU [Relational Algebra]

7. Consider the relational database given below and give an expression in the relational algebra:
   - Employee (person-name, street, city) , Works (person-name, company-name, salary)
   - Company (company-name, city) , Manages (person-name, manager-name)

   II. Find the names of all employees in this database who live in the same city as the company for which they work.

   III. Find the names, street address, and cities of residence of all employees who work for HCL and earn more than $10,000 per annum.

Questions asked in GTU [Relational Algebra]

8. The relational database schema is given below and write the relational algebra expressions for the given queries.
   - employee (person-name, street, city)
   - works (person-name, company-name, salary)
   - company (company-name, city)
   - manages (person-name, manager-name)

   I. Find the names of all employees who work for First Bank Corporation.

   II. Find the names and cities of residence of all employees who work for First Bank Corporation.

   III. Find the names, street address, and cities of residence of all employees who work for First Bank Corporation and earn more than $10,000 per annum.

   IV. Find the names of all employees in this database who do not work for First Bank Corporation.

Thank You...