UNIT II

Relational query languages: Relational algebra, Tuple and domain relational calculus, SQL3, DDL and DML constructs, Open source and Commercial DBMS - MYSQL, ORACLE, DB2, SQL server. Relational database design: Domain and data dependency, Armstrong's axioms, Functional Dependencies, Normal forms, Dependency preservation, Lossless design.

Relational Algebra

Sources: Reference Book, NPTEL website, Internet

Outline

- Relational Query Languages
- The Relational Algebra

Relational Query Languages

- Relational query languages use relational algebra to break the user requests and instruct the DBMS to execute the requests. It is the language by which user communicates with the database. These relational query languages can be procedural or non-procedural.
- SQL has its own querying methods to interact with database. But how these queries work in the database? These queries work similar to relational algebra that we have in mathematics. In database we have tables participating in relational algebra.
- **Procedural Query Language**
  - A procedural query language will have set of queries instructing the DBMS to perform various transactions in the sequence to meet the user request. For example, *get_CGPA* procedure will have various queries to get the marks of student in each subject, calculate the total marks, and then decide the CGPA based on his total marks. This procedural query language tells the database what is required from the database and how to get them from the database. Relational algebra is a procedural query language.
- **Non-Procedural Query Language**
  - Non-procedural queries will have single query on one or more tables to get result from the database. For example, get the name and address of the student with particular ID will have single query on STUDENT table. Relational Calculus is a non procedural language which informs what to do with the tables, but doesn't inform how to accomplish this.

Relational Algebra

- Relational algebra is a procedural query language. It takes one or more relations / tables and performs the operation and produce the result. This result is also considered as a new table or relation.
- Relational algebra will have operators to indicate the operations. This algebra can be applied on single relation – called **unary** or can be applied on two tables – called **binary**.
- Relational Algebra works on the whole table at once, so we do not have to use loops etc to iterate over all the rows(tuples) of data one by one.
- Relational Algebra is formal description of how relational database operates.
- It is a procedural query language, i.e. user must define both “how” and “what” to retrieve.
- All we have to do is specify the relation(table) name from which we need the data, and in a single line of command, relational algebra will traverse the entire given table to fetch data for you.

Operators

1. Select: σ
2. Project: $\prod$
3. Cartesian product: **X**
4. Inner Join:
   a) Natural Join:
   b) Theta join: $\bowtie_\theta$
5. Outer join
   a) Left outer join:
   b) Right outer join:
   c) Full outer join or outer join:
6. Additional operator:
   a) Set union: $\cup$
   b) Set intersection : $\cap$
   c) Set difference: --
   d) Division: /
   e) rename: ρ
   f) Assignment: $\leftarrow$

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. Select 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.

Student

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

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

Output

RollNo	Name	Branch	SPI
101	Raju	CE	8
104	Meet	CE	9

Selection Operator [$\sigma_{condition}$ (Relation)]

Example Display the detail of students belongs to "CE" Branch and having SPI more than 8.

Student

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

Answer $\sigma_{Branch='CE' \wedge SPI>8}$ (Student)

Output

RollNo	Name	Branch	SPI
104	Meet	CE	9

Selection Operator [$\sigma_{condition}$ (Relation)]

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

Student

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

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

Output

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

Selection Operator [$\sigma_{condition}$ (Relation)]

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

Student

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

Answer $\sigma_{SPI>8 \wedge SPI<9}$ (Student)

Output

RollNo	Name	Branch	SPI
101	Raju	CE	8

Exercise

- Write down the relational algebra for the student table.
  - Display the detail of students whose RollNo is less than 104.
  - Display the detail of students having SPI more than 8.
  - Display the detail of students belongs to “CE” Branch having SPI less than 8.
  - Display the detail of students belongs to either “CE” or “ME” Branch.
  - Display the detail of students whose SPI between 6 and 9.

Student

RollNo	Name	Branch	SPI
101	Raj	CE	6
102	Meet	ME	8
103	Harsh	EE	7
104	Punit	CE	9

- Write down the relational algebra for the employee table.
  - Display the detail of all employee.
  - Display the detail of employee whose Salary more than 10000.
  - Display the detail of employee belongs to “HR” Dept having Salary more than 20000.
  - Display the detail of employee belongs to either “HR” or “Admin” Dept.
  - Display the detail of employee whose Salary between 1000 and 25000 and belongs to “HR” Dept.

Employee

EmpID	Name	Dept	Salary
101	Nilesh	Sales	10000
102	Mayur	HR	25000
103	Hardik	HR	15000
104	Ajay	Admin	20000

2. Project Operator

- Symbol: ∏ (Pi)
- Notation: ∏ *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 ∏ *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

Exercise

- Write down the relational algebra for the student table.
  - Display RollNo, Name and SPI of all students.
  - Display Name and SPI of all students.
  - Display the Name of all students.
  - Display the Name of all branches.

Student

RollNo	Name	Branch	SPI
101	Raj	CE	6
102	Meet	ME	8
103	Harsh	EE	7
104	Punit	CE	9

- Write down the relational algebra for the employee table.
  - Display EmpID with Name of all employee.
  - Display Name and Salary of all employee.
  - Display the Name of all employee.
  - Display the Name of all departments.

Employee

EmpID	Name	Dept	Salary
101	Nilesh	Sales	10000
102	Mayur	HR	25000
103	Hardik	HR	15000
104	Ajay	Admin	20000

2.5 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

Answer $\Pi_{RollNo, Name, Branch}$ ($\sigma_{Branch='ME'}$ (Student))

Output-2

RollNo	Name	Branch
102	Mitesh	ME

Combined Projection & Selection Operation

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)

Output-1

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

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

Output-2

Name	Branch	SPI
Mitesh	ME	9
Nilesh	CI	9

Combined Projection & Selection Operation

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 $\Pi_{Name, Branch, SPI}$ ($\sigma_{Branch='CE' \wedge SPI>8}$ (Student))

Output-2

Name	Branch	SPI
Raju	CE	8

Combined Projection & Selection Operation

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)

Output-1

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

Answer $\Pi_{Name, Branch}$ ($\sigma_{Branch='ME' \ V \ Branch='CI'}$ (Student))

Output-2

Name	Branch	SPI
Mitesh	ME	9
Nilesh	CI	9

Exercise

- Write down the relational algebra for the student table.
  - Display Rollno, Name and SPI of all students belongs to “CE” Branch.
  - List the Name of students with their Branch whose SPI is more than 8 and belongs to “CE” Branch.
  - List the Name of students along with their Branch and SPI who belongs to either “CE” or “ME” Branch and having SPI more than 8.
  - Display the Name of students with their Branch name whose SPI between 7 and 9.

Student

RollNo	Name	Branch	SPI
101	Raj	CE	6
102	Meet	ME	8
103	Harsh	EE	7
104	Punit	CE	9

- Write down the relational algebra for the employee table.
  - Display the Name of employee belong to “HR” Dept and having salary more than 20000.
  - Display the Name of all “Admin” and “HR” Dept’s employee.
  - List the Name of employee with their Salary who belongs to “HR” or “Admin” Dept having salary more than 15000.
  - Display the Name of employee along with their Dept name whose salary between 15000 and 30000.

Employee

EmpID	Name	Dept	Salary
101	Nilesh	Sales	10000
102	Mayur	HR	25000
103	Hardik	HR	15000
104	Ajay	Admin	20000

3. 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.

Answer (Student) X (Result)

Student

RNo	Name	Branch
101	Raju	CE
102	Mitesh	ME

Result

RNo	SPI
101	8
102	9

Output

Student.RNo	Name	Branch	Result.RNo	SPI
101	Raju	CE	101	8
101	Raju	CE	102	9
102	Mitesh	ME	101	8
102	Mitesh	ME	102	9

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

Cartesian Product / Cross Product Example

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	Raju	CE	3
102	Mitesh	ME	5

Result

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

Answer $\Pi_{RNo, Name, Branch}$ (Student) X $\Pi_{RNo, SPI, BL}$ (Result)

Output

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

Cartesian Product / Cross Product Example

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	Raju	CE	3
102	Mitesh	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

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

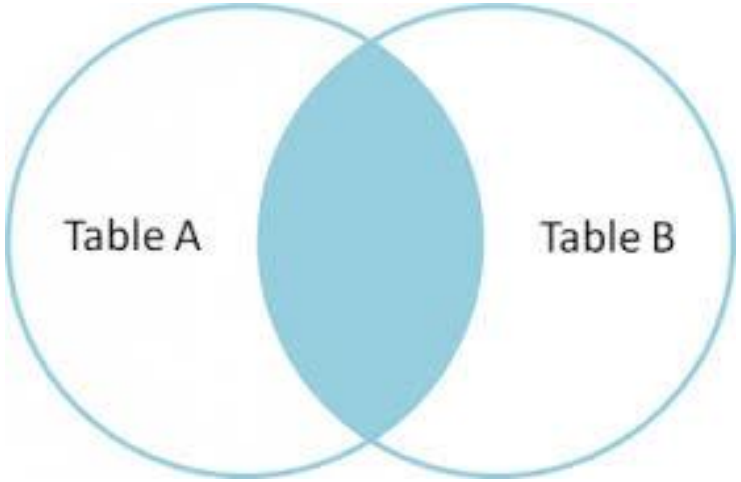
Output

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

4. 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**.

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



- Let *r* and *s* be relations on schemas *R* and *S* respectively.
- Then, $r \bowtie s$ is a relation on schema $R \cup S$ obtained as follows:
  - Consider each pair of tuples t_r from *r* and t_s from *s*.
  - If t_r and t_s have the same value on each of the attributes in $R \cap S$, add a tuple *t* to the result, where
    - *t* has the same value as t_r on *r*
    - *t* has the same value as t_s on *s*

- Example:

  $R = (A, B, C, D)$

  $S = (E, B, D)$
  - Result schema = (A, B, C, D, E)
  - $r \bowtie s$ is defined as:

    $$\Pi_{r.A,\ r.B,\ r.C,\ r.D,\ s.E} (\sigma_{r.B = s.B \wedge r.D = s.D} (r \times s))$$

Relations r, s:

A	B	C	D
α	1	α	a
β	2	γ	a
γ	4	β	b
α	1	γ	a
δ	2	β	b

r

B	D	E
1	a	α
3	a	β
1	a	γ
2	b	δ
3	b	$\in$

s

r $\bowtie$ s

A	B	C	D	E
α	1	α	a	α
α	1	α	a	γ
α	1	γ	a	α
α	1	γ	a	γ
δ	2	β	b	δ

Natural Join / Inner Join Example

Example Perform Natural Join between Student and Result.

Answer (Student) ⋈ (Result)

Student

RNo	Name	Branch
101	Raju	CE
102	Mitesh	ME

Result

RNo	SPI
101	8
103	9

Output

RNo	Name	Branch	SPI
101	Raju	CE	8

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

Steps performed in Natural Join

Step:1 Perform Cross Product

Student.RNo	Name	Branch	Result.RNo	SPI
101	Raju	CE	101	8
101	Raju	CE	103	9
102	Mitesh	ME	101	8
102	Mitesh	ME	103	9

Step:2 Removes inconsistent tuples

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

Step:3 Removes an attribute from duplicate

RNo	Name	Branch	SPI
101	Raju	CE	8

Natural Join / Inner Join Example

Example Perform Natural Join between Branch and Faculty.

Answer (Branch) ⋈ (Faculty)

Branch

BID	BName	HOD
1	CE	Shah
2	ME	Patel

Faculty

FID	FName	BID
101	Raj	1
103	Meet	2

Output

BID	Bname	HOD	FID	FName
1	CE	Shah	101	Raj
2	ME	Patel	102	Meet

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

Write down relational algebra for the following tables/relations

- Relations
  - Student (Rno, Sname, Address, City, Mobile)
  - Department (Did, Dname)
  - Academic (Rno, Did, SPI, CPI, Backlog)
  - Guide (Rno, PName, Fid)
  - Faculty (Fid, Fname, Subject, Did, Salary)

Example List the name of students with their department name and SPI of all student belong to "CE" department.

Answer $\prod_{Sname,\ Dname,\ SPI} (\sigma_{Dname='CE'} (Student \bowtie (Department \bowtie Academic)))$

Example Display the name of students with their project name whose guide is "A. J. Shah".

Answer $\prod_{Sname,\ Pname} (\sigma_{Fname='A.J.Shah'} (Student \bowtie (Guide \bowtie Faculty)))$

Exercise: Write down relational algebra for the following tables/relations

- Relations
  - Student (Rno, Sname, Address, City, Mobile)
  - Department (Did, Dname)
  - Academic (Rno, Did, SPI, CPI, Backlog)
  - Guide (Rno, PName, Fid)
  - Faculty (Fid, Fname, Subject, Did, Salary)
- List the name of students with their department name having backlog 0.
- List the name of faculties with their department name and salary having salary more than 25000 and belongs to “CE” department.
- List the name of all faculties of “CE” and “ME” department whose salary is more than 50000.
- Display the students name with their project name of all “CE” department’s students whose guide is “Z.Z. Patel”.
- Display the name of faculties with their department name who belongs to “CE” department and tough “CPU” subject having salary more than 25000.
- List the name of students with their department name doing project “Hackathon” under guide “I. I. Shah”.

4(a)Theta Join Operation

- The **join** operation allows us to combine a select operation and a Cartesian-Product operation into a single operation.
- Consider relations *r* (*R*) and *s* (*S*)
- Let **"theta"** be a predicate on attributes in the schema R "union" S. The join operation r $\bowtie_\theta$ s is defined as follows:

$$r \bowtie_\theta s = \sigma_\theta (r \times s)$$

- Thus

$$\sigma_{instructor.id = teaches.id} (instructor \times teaches))$$

- Can equivalently be written as

$$instructor \bowtie_{Instructor.id = teaches.id} teaches.$$

5. 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.

5(a) 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 though 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.

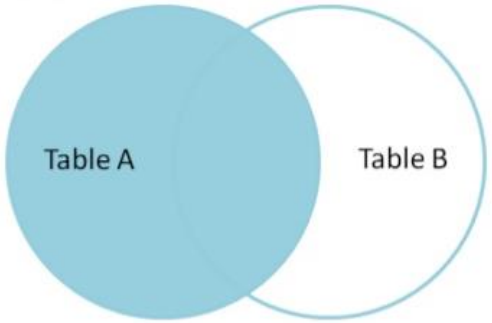
Answer (Student) ⟕ (Result)

Student

RollNo	Name	Branch
101	Raju	CE
102	Meet	ME

Result

RollNo	SPI
101	8
103	9



Output

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

Exercise What is the output of (Result) ⟕ (Student).

Left Outer Join Example

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

Student

RollNo	Name	Branch
101	Raju	CE
102	Meet	ME

Result

RollNo	SPI	BL
101	8	1
103	9	0

Answer $\Pi_{RollNo, Name, SPI}$ ((Student) $\overrightarrow{\bowtie}$ (Result))

Output

RollNo	Name	SPI
101	Raju	8
102	Meet	**NULL**

5(b) 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.

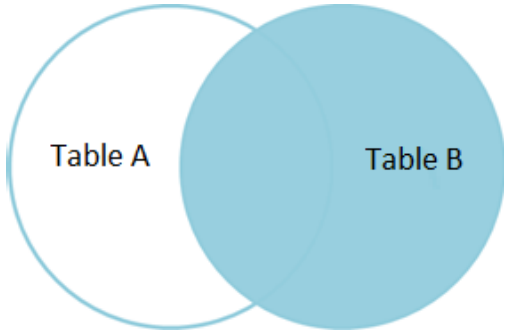
Answer (Student) ⟖ (Result)

Student

RollNo	Name	Branch
101	Raju	CE
102	Meet	ME

Result

RollNo	SPI
101	8
103	9



Output

RollNo	Name	Branch	SPI
101	Raju	CE	8
103	**NULL**	**NULL**	9

Exercise What is the output of (Result) ⟖ (Student).

Right Outer Join Example

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

Student

RollNo	Name	Branch
101	Raju	CE
102	Meet	ME

Result

RollNo	SPI	BL
101	8	1
103	9	0

Answer $\Pi_{RollNo, Name, SPI}$ ((Student) $\overset{\subset}{\bowtie}$ (Result))

Output

RollNo	Name	SPI
101	Raju	8
103	**NULL**	9

5(c) 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.

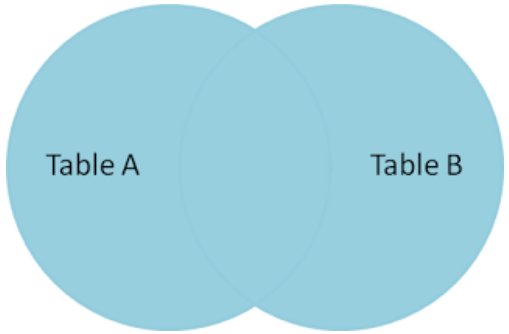
Answer (Student) ⟗ (Result)

Student

RollNo	Name	Branch
101	Raju	CE
102	Meet	ME

Result

RollNo	SPI
101	8
103	9



Output

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

Exercise What is the output of (Result) ⟗ (Student).

Full Outer Join Example

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

Student

RollNo	Name	Branch
101	Raju	CE
102	Meet	ME

Result

RollNo	SPI	BL
101	8	1
103	9	0

Answer $\Pi_{RollNo, Name, SPI}$ ((Student) ⟗ (Result))

Output

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

6. Set Operators

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

Three types of Set Operators		
Sr.	**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 relation should have **same (equal) number of columns**
- Corresponding **attributes should have the same data type** or **domain**

Conditions to perform Set Operators

Conditions-1 Both relation 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	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

Conditions-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

Set Operators [Exercise]

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))

X (Not compatible) Both tables have 3 attributes but **third attributes datatype is different**.

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

✓ (Compatible) Both tables have 3 attributes and of same data type.

- Person (PersonID, Name, Address, Hobby)
- Professor (ProfessorID, Name, OfficeAddress, Salary)
- (Not compatible) Both tables have 4 attributes but **forth attributes datatype is different**.

$\prod_{Name,\ Address}$ (Person) & $\prod_{Name,\ OfficeAddress}$ (Professor)

- (Compatible) Both tables have 2 attributes and of same data type.

6(a) 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

Exercise Is there any difference in the output if we swap the tables in Union operator. (Employee) **U** (Customer).

6(b) 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.

Customer

Name
Raju
Suresh
Meet

Employee

Name
Meet
Suresh
Manoj

Answer (Customer) ∩ (Employee)

Output

Name
Meet
Suresh

Exercise Is there any difference in the output if we swap the tables in Intersection. (Employee) ∩ (Customer).

6(c) 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.

Customer

Name
Raju
Suresh
Meet

Employee

Name
Meet
Suresh
Manoj

Answer (Customer) – (Employee)

Output

Name
Raju

Exercise Is there any difference in the output if we swap the tables in Set difference. (Employee) – (Customer).

Union Operators Example

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

Customer

ID	Name	Dept	Balance
1	Raju	CE	10000
2	Suresh	CE	20000

Employee

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

Answer $\Pi_{Name}(\text{Customer}) \cup \Pi_{Name}(\text{Employee})$

Output

Name
Manoj
Raju
Suresh

Intersect/ Intersection Operators Example

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

Customer

ID	Name	Dept	Balance
1	Raju	CE	10000
2	Suresh	CE	20000

Employee

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

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

Output

Name
Suresh

Minus/ Set difference Operators Example

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

Customer

ID	Name	Dept	Balance
1	Raju	CE	10000
2	Suresh	CE	20000

Employee

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

Answer $\prod_{Name}(\text{Employee}) - \prod_{Name}(\text{Customer})$

Output

Name
Manoj

Minus/ Set difference Operators Example

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

Customer

ID	Name	Dept	Balance
1	Raju	CE	10000
2	Suresh	CE	20000

Employee

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

Answer $\prod_{Name}(\text{Customer}) - \prod_{Name}(\text{Employee})$

Output

Name
Raju

Set Operators [Exercise]

Exercise What is the output of following relational algebra for the below mentioned tables:

Customer

ID	Name	Dept	Balance
1	Raju	CE	10000
2	Suresh	CE	20000

Employee

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

Algebra-1 $\prod_{ID, Name}$ (Customer) **U** $\prod_{ID, Name}$ (Employee)

Algebra-2 $\prod_{ID, Name, Balance}$ (Customer) **U** $\prod_{ID, Name, Salary}$ (Employee)

Algebra-3 $\prod_{ID, Name}$ (Customer) $\cap$ $\prod_{ID, Name}$ (Employee)

Algebra-4 $\prod_{ID, Name, Balance}$ (Customer) $\cap$ $\prod_{ID, Name, Salary}$ (Employee)

Set Operators [Exercise]

Exercise What is the output of following relational algebra for the below mentioned tables:

Customer

ID	Name	Dept	Balance
1	Raju	CE	10000
2	Suresh	CE	20000

Employee

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

Algebra-1 $\prod_{ID, Name}(Customer) - \prod_{ID, Name}(Employee)$

Algebra-2 $\prod_{ID, Name, Balance}(Customer) - \prod_{ID, Name, Salary}(Employee)$

Algebra-3 $\prod_{ID, Name}(Employee) - \prod_{ID, Name}(Customer)$

Algebra-4 $\prod_{ID, Name, salary}(Employee) - \prod_{ID, Name, balance}(Customer)$

6(d) 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.

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

Answer (Student) ÷ (Subject)

Output

Name
Rohit
Suresh

Division Operator Example

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

Division Operator Example

Example List the name of students doing a project in all technologies.

Student

RNo	Name	Technology
101	Raj	.NET
101	Raj	PHP
102	Meet	.NET
102	Meet	PHP
102	Meet	iPhone
102	Meet	Android
103	Rohit	Android
104	Suresh	.NET
104	Suresh	iPhone
104	Suresh	Android

Project

TID	Technology
1	.NET
2	PHP
3	Android
4	iPhone

Answer $\prod_{Name,\ Technology}(Student) \div \prod_{Technology}(Project)$

Output

Name
Meet

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.
  - $\rho_{A1, A2. ...,An}(E)$
    Returns a relation E with the attributes renamed to A1, A2, …., An.
  - $\rho_{x(A1, A2. ...,An)}(E)$
    Returns a relation E under a new name X with the attributes renamed to A1, A2, …., An.

Rename Operator Example

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

Rename Operator Example

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

Rename Operator Example

Example Find out maximum CPI from student table.

Student

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

Step-2 $\sigma_{A.CPI<B.CPI}(\rho_A(Student) \times \rho_B(Student))$

Output-2

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

Step-1 $\rho_A(Student) \times \rho_B(Student)$

Output-1

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

Rename Operator Example

Example Find out maximum CPI from student table.

Student

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

Step-2 $\sigma_{A.CPI<B.CPI}(\rho_A(Student) \times \rho_B(Student))$

Output-2

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

Step-3 $\prod_{A.CPI}(\sigma_{A.CPI<B.CPI}(\rho_A(Student) \times \rho_B(Student)))$

Output-3

A.CPI
8
7

Rename Operator Example

Example Find out maximum CPI from student table.

Student

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

Step-3 $\prod_{A.CPI} (\sigma_{A.CPI<B.CPI} (\rho_A (Student) \ X \ \rho_B (Student)))$

Output-3

A.CPI
8
7

Step-4 $\prod_{CPI} (Student) - \prod_{A.CPI} (\sigma_{A.CPI<B.CPI} (\rho_A (Student) \ X \ \rho_B (Student)))$

Student

CPI
8
9
7

−

Output-3

A.CPI
8
7

=

Output

CPI
9

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.)

Aggregate Functions Example

Student

Rno	Name	Branch	Semester	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

Example Find out sum of CPI of all students.

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

Output

sum
73

Example Find out maximum & minimum CPI.

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

Output

max	min
9	7

Example Count the number of students.

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

Output

count
9

Example Find out average of CPI of all students.

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

Output

avg
8.11

Assignment Operator

- It is convenient at times to write a relational-algebra expression by assigning parts of it to temporary relation variables.
- The assignment operation is denoted by $\leftarrow$ and works like assignment in a programming language.
- Example: Find all instructor in the “Physics” and Music department.

$$Physics \leftarrow \sigma_{dept\_name=\text{“}Physics\text{”}}(instructor)$$
$$Music \leftarrow \sigma_{dept\_name=\text{“}Music\text{”}}(instructor)$$
$$Physics \cup Music$$

- With the assignment operation, a query can be written as a sequential program consisting of a series of assignments followed by an expression whose value is displayed as the result of the query.

Equivalent Queries

There is more than one way to write a query in relational algebra.

Example: Find information about courses taught by instructors in the Physics department with salary greater than 90,000

- Query 1

$$\sigma_{dept\_name=\text{“Physics”} \wedge salary > 90{,}000}(instructor)$$

- Query 2

$$\sigma_{dept\_name=\text{“Physics”}}(\sigma_{salary > 90.000}(instructor))$$

The two queries are not identical; they are, however, equivalent -- they give the same result on any database.

Example: Find information about courses taught by instructors in the Physics department

- Query 1

  $\sigma_{dept\_name=\text{“Physics”}}(instructor \bowtie_{instructor.ID = teaches.ID} teaches)$

- Query 2

  $(\sigma_{dept\_name=\text{“Physics”}}(instructor)) \bowtie_{instructor.ID = teaches.ID} teaches$

- The two queries are not identical; they are, however, equivalent -- they give the same result on any database.

Exercise 1

employee (*name*, *street*, *city*)

works (*name*, *cname*, *salary*)

company (*cname*, *city*)

1. Find the names of all employees who live in city “Miami”.

$$\Pi_{name}\ (\sigma_{city = \text{“Miami”}}\ (employee))$$

2. Find the names of all employees whose salary is greater than $100,000.

$$\Pi_{name}\ (\sigma_{salary > 100000}\ (works))$$

3. Find the names of all employees who live in “Miami” and whose salary is greater than $100,000.

$$\Pi_{name}\ (\sigma_{city = \text{“Miami”} \wedge salary > 100000}\ (employee \bowtie works))$$

$$\Pi_{name}\ ((\sigma_{city = \text{“Miami”}}\ (employee)) \bowtie (\sigma_{salary > 100000}\ (works))$$

Exercise 2

employee (*person-name*, *street*, *city*)
works (*person-name*, *company-name*, *salary*)
company (*company-name*, *city*)
manages (*person-name*, *manager-name*)

a. Find the names of all employees who work for First Bank Corporation.
b. Find the names and cities of residence of all employees who work for First Bank Corporation.
c. 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.
d. Find the names of all employees in this database who live in the same city as the company for which they work.

a. $\Pi_{person\text{-}name}\,(\sigma_{company\text{-}name\,=\,\text{“First Bank Corporation”}}\,(works))$

b. $\Pi_{person\text{-}name,\,city}\,(employee \bowtie (\sigma_{company\text{-}name\,=\,\text{“First Bank Corporation”}}\,(works)))$

c. $\Pi_{person\text{-}name,\,street,\,city}\,(\sigma_{(company\text{-}name\,=\,\text{“First Bank Corporation”}\,\wedge\,salary\,>\,10000)}\,works \bowtie employee)$

d. $\Pi_{person\text{-}name}\,(employee \bowtie works \bowtie company)$

employee (*person-name*, *street*, *city*)
works (*person-name*, *company-name*, *salary*)
company (*company-name*, *city*)
manages (*person-name*, *manager-name*)

g. Find the names of all employees who earn more than every employee of Small Bank Corporation.

h. Assume the companies may be located in several cities. Find all companies located in every city in which Small Bank Corporation is located.

g. $\Pi_{person\text{-}name}(works) - (\Pi_{works.person\text{-}name}(works \bowtie_{(works.salary \leq works2.salary \wedge works2.company\text{-}name = \text{“Small Bank Corporation”})} \rho_{works2}(works)))$

h. Note: Small Bank Corporation will be included in each answer.

$$\Pi_{company\text{-}name}(company \div (\Pi_{city}(\sigma_{company\text{-}name = \text{“Small Bank Corporation”}}(company))))$$

Example 3

branch (branch-name, branch-city, assets)
customer (customer-name, customer-street, customer-only)
account (account-number, branch-name, balance)
loan (loan-number, branch-name, amount)
depositor (customer-name, account-number)
borrower (customer-name, loan-number)

- Find the names of all branches located in “Chicago”.

$$\Pi_{branch\_name} (\sigma_{branch\_city = \text{“Chicago”}} (branch))$$

- Find the names of all borrowers who have a loan in branch “Downtown”.

$$\Pi_{customer\_name} (\sigma_{branch\_name = \text{“Downtown”}} (borrower \bowtie loan))$$

branch (branch-name, branch-city, assets)
customer (customer-name, customer-street, customer-only)
account (account-number, branch-name, balance)
loan (loan-number, branch-name, amount)
depositor (customer-name, account-number)
borrower (customer-name, loan-number)

- Find all loans of over $1200 $\sigma_{amount > 1200}(loan)$
- Find the loan number for each loan of an amount greater than $1200 $\Pi_{loan\text{-}number}(\sigma_{amount > 1200}(loan))$
- Find the names of all customers who have a loan, an account, or both, from the bank

$$\Pi_{customer\text{-}name}(borrower) \cup \Pi_{customer\text{-}name}(depositor)$$

branch (branch-name, branch-city, assets)
customer (customer-name, customer-street, customer-only)
account (account-number, branch-name, balance)
loan (loan-number, branch-name, amount)
depositor (customer-name, account-number)
borrower (customer-name, loan-number)

- Find the names of all customers who have a loan and an account at bank.

$$\Pi_{customer\text{-}name}\ (borrower) \cap \Pi_{customer\text{-}name}\ (depositor)$$

- Find the names of all customers who have a loan at the Perryridge branch.

$$\Pi_{customer\text{-}name}\ (\sigma_{branch\text{-}name=\text{“}Perryridge\text{”}}\ (\sigma_{borrower.loan\text{-}number\ =\ loan.loan\text{-}number}(borrower\ x\ loan)))$$

- Find the largest account balance(Rename *account* relation as *d*)

$$\Pi_{balance}(account) - \Pi_{account.balance}(\sigma_{account.balance < d.balance}(account \times \rho_{d}(account)))$$

account

1	A	500
2	B	300
3	C	700

d(account)

1	A	500
2	B	300
3	C	700

account x d

1	A	500	1	A	500
1	A	500	2	B	300
1	A	500	3	C	700
2	B	300	1	A	500
2	B	300	2	B	300
2	B	300	3	C	700
3	C	700	1	A	500
3	C	700	2	B	300
3	C	700	3	C	700

Example 4

Suppliers(*sid, sname, address*)
Parts(*pid, pname, color*)
Catalog(*sid, pid, cost*)

1. *"Find the names of suppliers who supply some red part."*

$$\pi_{\mathsf{sname}}((\sigma_{\mathsf{colour}='\mathtt{red}'}(\mathsf{Part}) \bowtie \mathsf{Catalog}) \bowtie \mathsf{Supplier}))$$

2. *"Find the IDs of suppliers who supply some red or green part."*

$$\pi_{\mathsf{sid}}(\sigma_{\mathsf{colour}='\mathtt{red}' \vee \mathsf{colour}='\mathtt{green}'}(\mathsf{Part}) \bowtie \mathsf{Catalog})$$

3. *"Find the IDs of suppliers who supply some red part or are based at 21 George Street."*

$$\pi_{\mathsf{sid}}(\sigma_{\mathsf{colour}='\mathtt{red}'}(\mathsf{Part}) \bowtie \mathsf{Catalog}) \cup \pi_{\mathsf{sid}}(\sigma_{\mathsf{address}='21\mathtt{G.S.}'}(\mathsf{Supplier}))$$

Suppliers(*sid, sname, address*)
Parts(*pid, pname, color*)
Catalog(*sid, pid, cost*)

4. *"Find the names of suppliers who supply some red part or are based at 21 George Street."*

$$\pi_{\text{sname}}(\sigma_{\text{colour}='\texttt{red}'}(\text{Part}) \bowtie \text{Catalog} \bowtie \text{Supplier}) \cup \pi_{\text{sname}}(\sigma_{\text{address}='21\text{G.S.}'}(\text{Supplier}))$$

5. *"Find the IDs of suppliers who supply some red part and some green part."*

$$\pi_{\text{sid}}(\sigma_{\text{colour}='\texttt{red}'}(\text{Part}) \bowtie \text{Catalog}) \cap \pi_{\text{sid}}(\sigma_{\text{colour}='\texttt{green}'}(\text{Part}) \bowtie \text{Catalog})$$

6. *"Find pairs of* sids *such that the supplier with the first* sid *charges more for some part than the supplier with the second* sid*."*

$$\text{Cat1} \leftarrow \text{Catalog}$$
$$\text{Cat2} \leftarrow \text{Catalog}$$
$$\pi_{\text{Cat1.sid,Cat2.sid}}(\text{Cat1} \bowtie_{\text{Cat1.pid}=\text{Cat2.pid} \wedge \text{Cat1.cost}>\text{Cat2.cost}} \text{Cat2})$$

Suppliers(*sid, sname, address*)

Parts(*pid, pname, color*)

Catalog(*sid, pid, cost*)

7. *"Find the IDs of suppliers who supply only red parts."*

$$\pi_{\text{sid}}(\text{Supplier}) - \pi_{\text{sid}}(\text{Catalog} \bowtie \sigma_{\text{colour} \neq \text{'red'}}(\text{Part}))$$

8. *"Find the IDs of suppliers who supply every part."*

$\pi_{\text{sid,pid}}(\text{Catalog})$

$\pi_{\text{sid,pid}}(\text{Catalog}) / \pi_{\text{pid}}(\text{Part})$

$\pi_{\text{sid}}(\text{Catalog}) \times \pi_{\text{pid}}(\text{Part})$

$\pi_{\text{sid}}(\text{Catalog}) \times \pi_{\text{pid}}(\text{Part}) \quad \pi_{\text{sid,pid}}(\text{Catalog})$

$\pi_{\text{sid}}(\text{Catalog}) \quad \pi_{\text{sid}}(\text{Temp1})$.

"Find the names of suppliers supplying some red part for less than 100 Quid."

$$\pi_{\text{sname}}(\sigma_{\text{colour}=\text{'red'}}(\text{Part}) \bowtie \sigma_{\text{cost}<100}(\text{Catalog}) \bowtie \text{Supplier})$$

Relational Algebra [Exercise-1]

- Write down relational algebras for the following table:
  - Employee (person-name, street, city)
  - Works (person-name, company-name, salary)
  - Company (company-name, city)
  - Managers (person-name, manager-name)

1. Find the names of all employees who work for “TCS”.
2. Find the names and cities of residence of all employees who work for “Infosys”.
3. Find the names, street and city of residence of all employees who work for “ITC” and earn more than $10,000 per annum.
4. Find the names of all employees in this database who live in the same city as the company for which they work.
5. Find the names of all employees working in “TCS” who earn more than 25000 and less than 40000.
6. Find the name of employee whose manager is “Ajay Patel” and salary is more than 50000.
7. Display the name of employee with street, city, company name, salary and manager name staying in “Rajkot” and working in “Ahmedabad”.
8. Find maximum, minimum and average salary of all employee.
9. Find out the total number of employee.

Relational Algebra [Exercise-2]

Reserves(sid, bid, day)

Sailors(sid, sname, rating, age)

Boats(bid, bname, color)

1. Find names of sailors who have reserved boat #103
2. Find names of sailors who have reserved a red boat
3. Find the colors of boats reserved by Lubber
4. Find the names of sailors who have reserved at least one boat
5. Find the names of sailors who have reserved a red or a green boat
6. Find the names of sailors who have reserved a red and a green boat
7. Find the sids of sailors with age over 20 who have not reserved a red boat

Relational Algebra [Exercise-3]

Reserves(sid, bid, day)

Sailors(sid, sname, rating, age)

Boats(bid, bname, color)

1. Find the colors of boats reserved by Albert.
2. Find all sailor id's of sailors who have a rating of at least 8 or reserved boat 103.
3. Find the names of sailors who have not reserved a red boat.
4. Find the sailor id's of sailors with age over 20 who have not reserved a red boat.
5. Find the names of sailors who have reserved at least two boats.
6. Find the names of sailors who have reserved all boats.
7. Find the names of sailors who have reserved all boats called BigBoat.
8. Find the sailor id's of sailors whose rating is better than some sailor called Bob.
9. Find the sailor id's of sailors whose rating is better than every sailor called Bob.
10. Find the sailor id's of sailors with the highest rating.
11. Find the name and age of the oldest sailor.
12. Find the names of sailors who have reserved every boat reserved by those with a lower rating.

Exercise

1. Consider the relations $r_1(P, Q, R)$ and $r_2(R, S, T)$ with primary keys P and R respectively. The relation r_1 contains 2000 tuples and r_2 contains 2500 tuples. The maximum size of the join $r_1 \bowtie r_2$ is.

(A) 2000
(B) 2500
(C) 4500
(D) 5000

2. Consider a database table T containing two columns X and Y each of type integer. After the creation of the table, one record (X=1, Y=1) is inserted in the table.

Let MX and MY denote the respective maximum values of X and Y among all records in the table at any point in time. Using MX and MY, new records are inserted in the table 128 times with X and Y values being MX+1, 2*MY+1 respectively. It may be noted that each time after the insertion, values of MX and MY change. What will be the output of the following SQL query after the steps mentioned above are carried out?

```
SELECT Y FROM T WHERE X=7;
```