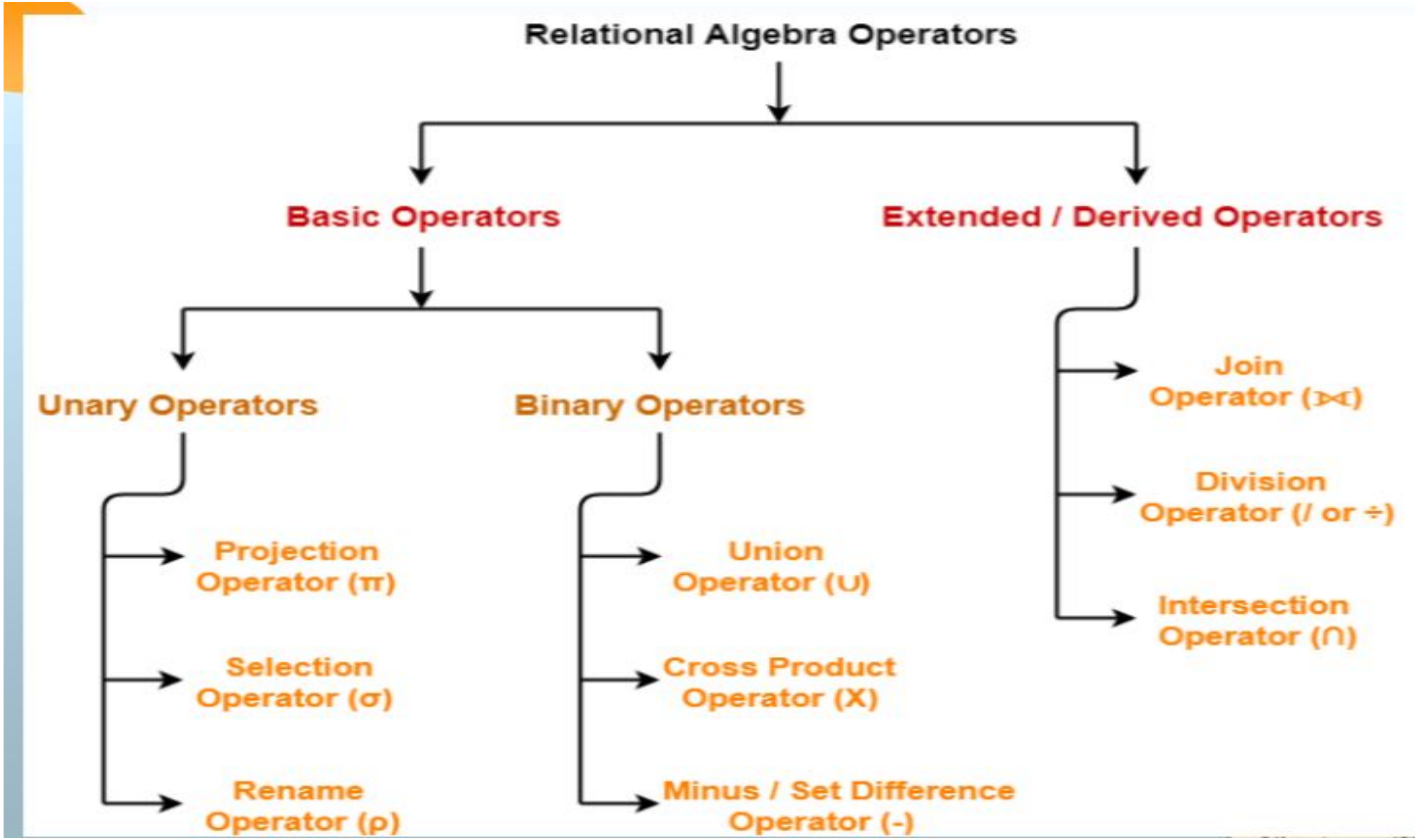
Unit 3

Relational Algebra

Index

- Procedural language
- Six basic operators
  - select
  - project
  - union
  - set difference
  - Cartesian product
  - rename
- The operators take one or more relations as inputs and give a new relation as a result.


Relational Algebra Operators
Basic Operators
Extended / Derived Operators
Unary Operators
Binary Operators
Join Operator (⋈)
Division Operator (/ or ÷)
Intersection Operator (∩)
Projection Operator (π)
Selection Operator (σ)
Rename Operator (ρ)
Union Operator (∪)
Cross Product Operator (X)
Minus / Set Difference Operator (-)

- Unary Relational Operations
- SELECT (symbol: σ)
- PROJECT (symbol: π)
- RENAME (symbol: ρ)

- Relational Algebra Operations From Set Theory
- UNION (υ)
- INTERSECTION ( ),
- DIFFERENCE (-)
- CARTESIAN PRODUCT (x)

- Binary Relational Operations
- JOIN
- DIVISION

- Notation: $\sigma_p(r)$
- p is called the **selection predicate**
- Defined as:

$$\sigma_p(\boldsymbol{r}) = \{t \mid t \in r \textbf{ and } p(t)\}$$

Where p is a formula in propositional calculus consisting of **terms** connected by : $\wedge$ (**and**), $\vee$ (**or**), $\neg$ (**not**)
Each **term** is one of:

<attribute>*op* <attribute> or <constant>

where *op* is one of: =, $\neq$, >, $\geq$. <. $\leq$

- Example of selection:

$$\sigma_{\textit{dept\_name=“Physics”}}(\textit{instructor})$$

SELECTION (σ)

- The SELECT operation is used for selecting a subset of the tuples according to a given selection condition. Sigma(σ)Symbol denotes it.
- ***Ex:-*** find all employees born after 1st Jan 1950:

$$\sigma_{dob\ '01/JAN/1950'}(employee)$$

- σp(r)
- σ is the predicate
- r stands for relation which is the name of the table
- p is prepositional logic

Examples

- Selects tuples from Tutorials where topic = 'Database’
- Selects tuples from Tutorials where the topic is 'Database' and 'author' is guru99.
- Selects tuples from Customers where sales is greater than 50000
- Selects tuples from books where subject is 'database’.
- Selects tuples from books where subject is 'database' and 'price' is 450.
- Selects tuples from books where subject is 'database' and 'price' is 450 or those books published after 2010.

- 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

PROJECTION(∏)Pi

- ∏ (pi) symbol used to choose attributes from a relation.
- This helps to extract the values of specified attributes to eliminates duplicate values. (pi) symbol is used to choose attributes from a relation.

Notation: ∏ *attribute set* (Relation)

SELECTION & PROJECTION Example

Person

Id	Name	Address	Hobby
1123	John	123 Main	stamps
1123	John	123 Main	coins
5556	Mary	7 Lake Dr	hiking
9876	Bart	5 Pine St	stamps

σ Hobby='stamps'(Person)

Id	Name	Address	Hobby
1123	John	123 Main	stamps
9876	Bart	5 Pine St	stamps

Π Name, Hobby(Person)

Name	Hobby
John	stamps
John	coins
Mary	Hiking
Bart	stamps

Project Operation – Example

- Relation *r*:

A	B	C
α	10	1
α	20	1
β	30	1
β	40	2

- $\Pi_{A,C}(r)$

A	C
α	1
α	1
β	1
β	2

=

A	C
α	1
β	1
β	2

Examples

- projection of CustomerName and status

- 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

Examples on selection and projection

- Display Roll No,Name, Branch of “ME” students
- Display Roll No, SPI of “CE” students whose SPI is greater than 6

Student

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

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

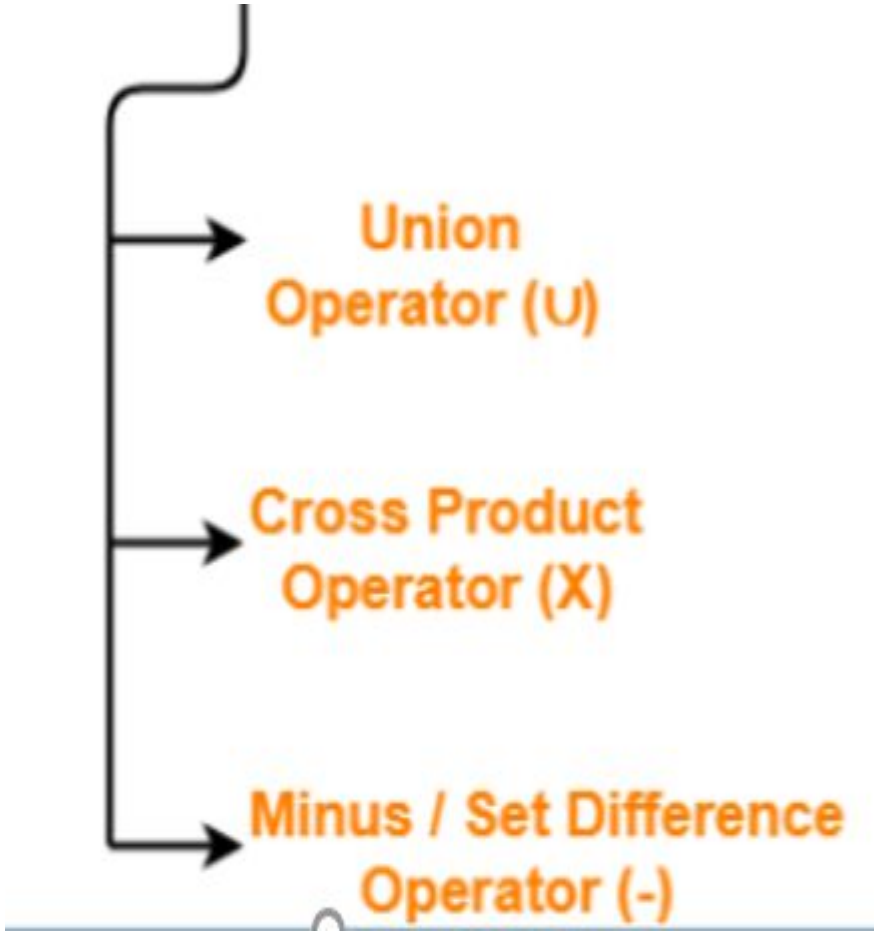
RollNo	Name	Branch	SPI
101	Raj	CE	6
102	Meet	ME	8
103	Harsh	EE	7
104	Punit	CE	9

RENAME (ρ) Operation in Relational Algebra

Notation:

```
ρ x (R)
```

Binary Operations



UNION

- UNION is symbolized by ∪ symbol. It includes all tuples that are in tables A or in B. It also eliminates duplicate tuples. So, set A UNION set B would be expressed as:

- The result <- A ∪ B

- For a union operation to be valid, the following conditions must hold -

- R and S must be the same number of attributes.
- Attribute domains need to be compatible.
- Duplicate tuples should be automatically removed.

Union Operation – Example

- Relations *r*, *s*:

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

$r \cup s$:

A	B
α	1
α	2
β	1
β	3

Set Difference(-)

- - Symbol denotes it. The result of A - B, is a relation which includes all tuples that are in A but not in B.
- The attribute name of A has to match with the attribute name in B.

Set Difference Operation – Example

- Relations *r*, *s*:

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

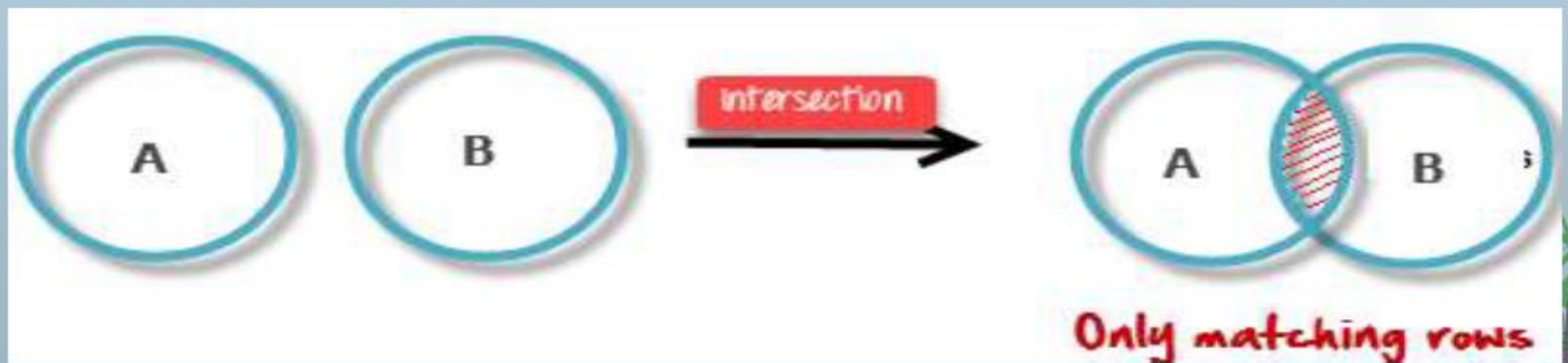
s

r – *s*:

A	B
α	1
β	1

What about Intersection ?

- An intersection is defined by the symbol ∩
- A ∩ B
- Defines a relation consisting of a set of all tuple that are in both A and B. However, A and B must be union-compatible.



Cartesian-Product Operation

- artesian Product in DBMS is an operation used to merge columns from two relations.
- Notation − r X s
- Where r and s are relations and their output will be defined as −
- σauthor = 'tutorialspoint'(Books X Articles)
- Output − Yields a relation, which shows all the books and articles written by tutorialspoint.

Relations *r*, *s*:

A	B
α	1
β	2

r

C	D	E
α	10	a
β	10	a
β	20	b
γ	10	b

s

r x *s*:

A	B	C	D	E
α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b

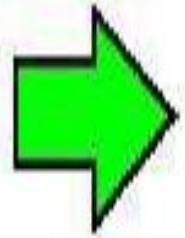
A1(Name, Roll No)

Name	Roll No
Anoop	1
Anurag	2

X

A2(Name, Roll No)

Name	Roll No
Anoop	1
Anurag	2
Ganesh	3



Query Output(A1 X A2)

Name	Roll No	Name	Roll No
Anoop	1	Anoop	1
Anoop	1	Anurag	2
Anoop	1	Ganesh	3
Anurag	2	Anoop	1
Anurag	2	Anurag	2
Anurag	2	Ganesh	3

Examples

- Display the student rollno,name and branch and result of student rollno,spi and of all the students.

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

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

Example

- Display the students and result whose branchname is CE and they are belongs to 3rd sem ,SPI is more than 7 and BL is less than 1

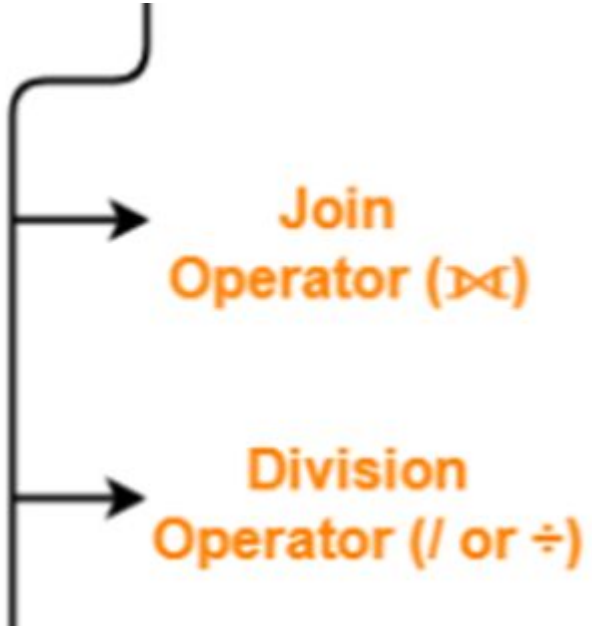
Student

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

Derived operations

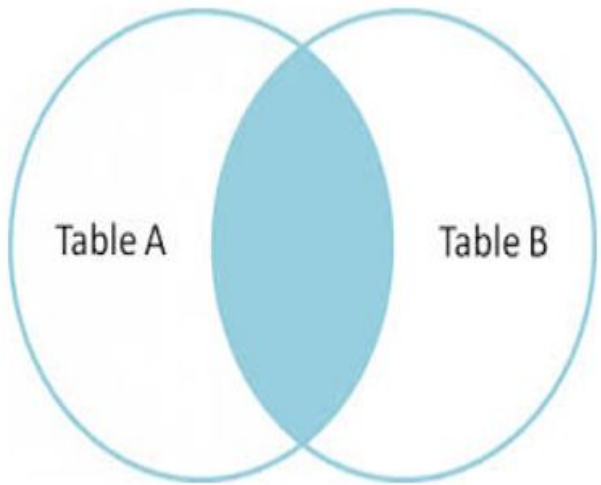


Natural Join or Inner Join

- 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



Example

- Natural join on student and result
- Natural join on Branch and Faculty

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

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 and SPI of all student belong to “CE” department.

Display the name of students with their project name whose guide is “A. J. Shah”.

- 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".

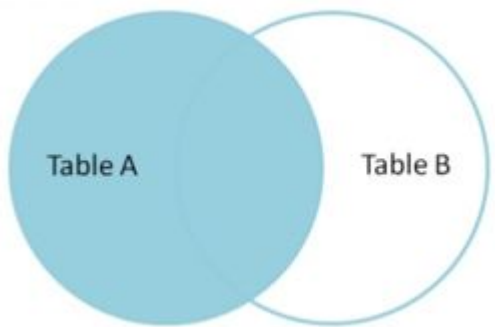
Outer Join

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.

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.

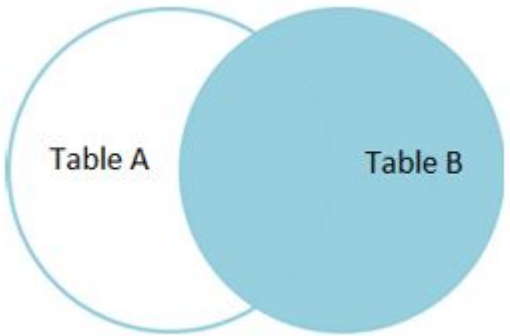


Examples

- Perform left outer join between student and result
- Display name and spi using left outer join between student and result

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.

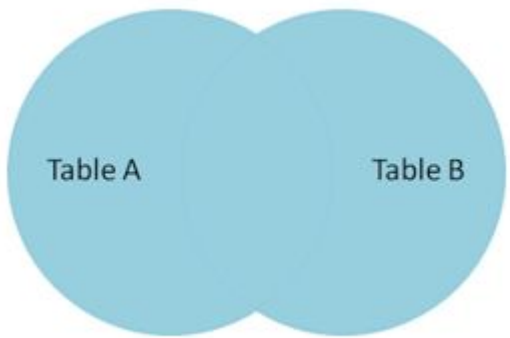


Example

- Perform Right outer join between student and result.
- Display name and spi using right outer join between student and result.

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.



Examples

- Perform full outer join between student and result.
- Display name and spi using full outer join between student and result

Set operators
(Union, Intersection, Difference)

Customer

ID	Name	Balance
1	Raju	10000
2	Suresh	20000

Employee

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

Display Name of person who are either employee or customer.

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

Display Name of person who are employee but not customer.

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.

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

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

B2

PNo
P2
P4

B3

PNo
P1
P2
P4

B4

PNo
P2
P5

Algebra (A) ÷ (B1)

Output

SNo
S1
S2
S3
S4

Algebra (A) ÷ (B2)

Output

SNo
S1
S4

Algebra (A) ÷ (B3)

Output

SNo
S1

Algebra (A) ÷ (B4)

Output

SNo

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

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

Output

Name
Meet

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

Aggregation Function

- 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.)

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

Find out sum of CPI of all students.

Find out maximum & minimum CPI.

Count the number of students.

Find out average of CPI of all students.

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

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

Open source vs Commercial DBMS

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

GTU Questions

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.

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)

   I. List all pairs of employee names and the project numbers they work on.
   II. List out department number, department name and department budget.
   III. List all projects that Raj Yadav works on by project name.
   IV. List the names of employees who supervise themselves.

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)

   I. Find the name of all instructors in the physics department.
   II. Find all the courses taught in the fall 2009 semester but not in Spring semester.
   III. Find the names of all instructors in the Comp. Sci. department together with the course titles of all the courses that the instructors teach.
   IV. Find the average salary in each department.

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

I. Find the Employees working in the department number10, 20, 30 only.
II. Find Employees whose names start with letter A or letter a.
III. Find Employees along with their department name.
IV. Find the Employees who are working in Smith's department
V. Find the Employees who get salary more than Allen's salary.
VI. Display employees who are getting maximum salary in each department.
VII. Find list of employees whose hire date is on or before 1-April-18.

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)

I. Find the names of all employees in this database who live in the same city as the company for which they work.
II. Find the names, street address, and cities of residence of all employees who work for HCL and earn more than $10,000 per annum.

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.