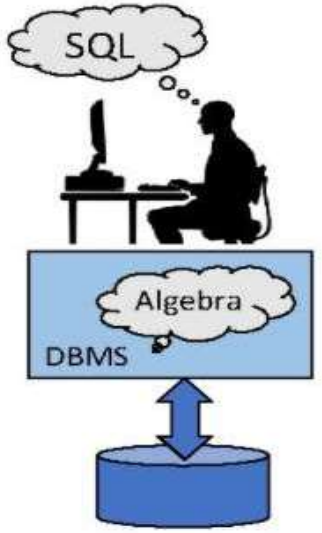
C. V. Raman Global University
ODISHA BHUBANESWAR INDIA

RELATIONAL ALGEBRA

Adyasha Rath
Assistant Professor, CSE
C.V Raman Global University, Bhubaneswar

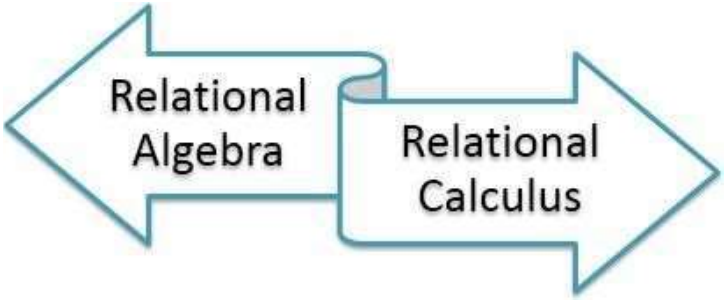
Relational Query Languages

- *Query languages:* Allow manipulation and retrieval of data from a database.
- Relational model supports simple, powerful QLs
- Query Languages != programming languages!



Formal Relational Query Languages

Two mathematical Query Languages:



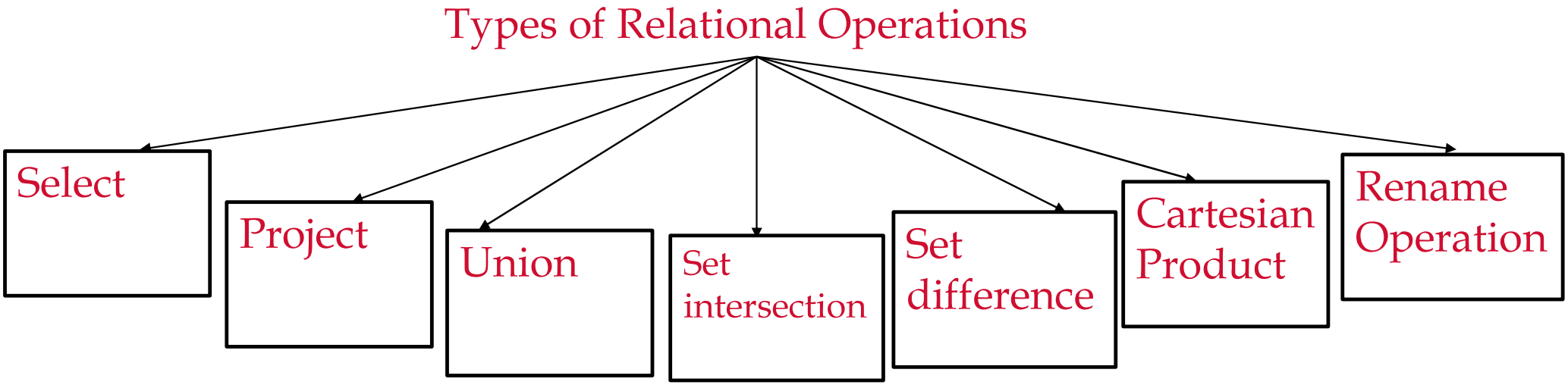
Relational Algebra: **More operational, very useful for representing execution plans. (Procedural)**

Relational Calculus: **Lets users describe what they want, rather than how to compute it. (Non-procedural)**

Relational Algebra

- The basic set of operations for relational model is Relational Algebra
- Relational algebra is a procedural query language. It gives a step by step process to obtain the result of the query.
- It uses operators to perform queries.

Relational Algebra : Basic Operations



Relational Algebra : Basic Operations

- *Selection* (σ) Selects a subset of ***rows*** from relation (horizontal).
- *Projection* (π) Retains only wanted ***columns*** from relation (vertical).
- *Cross-product* (x) Allows us to combine two relations.
- *Set-difference* (–) Tuples in r1, but not in r2.
- *Union* ($\cup$) Tuples in r1 and/or in r2.

Relational Algebra : Basic Operations

- *Intersection* ($\cap$) Common tuples in r1 and r2.
- *Division* ($\div$) Projects attributes present in r1 but not present in r2.
- *Rename* (ρ) Returns an existing relation with a new name.
- *Join* Combines two relations r1 and r2 to form a single relation r.

Relational Algebra : Basic Operations

Select Operation (σ)

Used for selection predictions
r is used for relation
P is used as a propositional logic formula which may use connectors like AND, OR and NOT.
Relational Operators like: =, ≠, ≥, ≤, >, <

Syntax

σ (Relation Name)

<select condition>

Relational Algebra : Basic Operations

Example

LOAN Relation

BRANCH NAME	LOAN_NO	AMOUNT
Cuttack	L-1	1000
BBSR	L-2	2000
Rourkela	L-3	3000
Balasore	L-4	4000
BBSR	L-5	5000

σ (LOAN)

BRANCH_NAME = “BBSR”

Output

BRANCH NAME	LOAN_NO	AMOUNT
BBSR	L-2	2000
BBSR	L-5	5000

Relational Algebra : Basic Operations

Q. Retrieve the student information who have secured more than 70% marks?

$$\sigma_{\text{Per\_Marks}>70}(\text{STUDENT})$$

Q. Select employee tuples who are getting salary less than 80,000

$$\sigma_{\text{Salary}<80,000}(\text{EMPLOYEE})$$

Relational Algebra : Basic Operations

Q. Retrieve the employee taken (tuples) who are getting more than 20,000 salary and they are staying at BBSR

σ (EMPOYEE)

Salary>20,000 AND City='BBSR'

Q. Selects tuples from Tutorials where the topic is "database" and "author" is NAVATHE

σ (TUTORIALS)

Topic=''Database "AND Author= "NAVATHE"

Example Instances

R1

sid	bid	day
22	101	10/10/96
58	103	11/12/96

S1

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

S2

sid	sname	rating	age
28	yuppy	9	35.0
31	lubber	8	55.5
44	guppy	5	35.0
58	rusty	10	35.0

bid	bname	color
101	Interlake	blue
102	Interlake	red
103	Clipper	green
104	Marine	red

Boats

Selection

- Selects rows that satisfy *selection condition*.
- Example:

sid	sname	rating	age
28	yuppy	9	35.0
31	lubber	8	55.5
44	guppy	5	35.0
58	rusty	10	35.0

$$\sigma_{rating>8}(S2)$$

sname	rating
yuppy	9
rusty	10

$$\pi_{sname,rating}(\sigma_{rating>8}(S2))$$

Relational Algebra : Basic Operations

Example

PLAYER Relation

Player ID	Team ID	Country	Age	Runs	Wickets
1001	101	India	25	10000	300
1004	101	India	28	20000	200
1006	101	India	22	15000	150
1005	101	India	21	12000	400
1008	101	India	22	15000	150
1009	103	England	24	6000	90
1010	104	Australia	35	1300	0
1011	104	Australia	29	3530	10
1012	105	Pakistan	28	1421	166
1014	105	Pakistan	21	3599	205

Relational Algebra : Basic Operations

Q. Find all tuples from Player relation for which country is India

Player ID	Team ID	Country	Age	Runs	Wickets
1001	101	India	25	10000	300
1004	101	India	28	20000	200
1006	101	India	22	15000	150
1005	101	India	21	12000	400
1008	101	India	22	15000	150

σ (PLAYER)

"Country"=India

Q. Select all the tuples for which runs are greater than or equal to 15000

σ (PLAYER)

"runs">=15000

Player ID	Team ID	Country	Age	Runs	Wickets
1004	101	India	28	20000	200
1006	101	India	22	15000	150
1008	101	India	22	15000	150

Relational Algebra : Basic Operations

Q. Select all the players whose runs are greater than or equal to 6000 and age is less than 25

σ (PLAYER)

"runs>=6000 ^ Age<25

Player ID	Team ID	Country	Age	Runs	Wickets
1006	101	India	22	15000	150
1005	101	India	21	12000	400
1008	101	India	22	15000	150
1009	103	England	24	6000	90

Relational Algebra : Basic Operations

EXAMPLE

CUSTOMER

Customer ID	Customer Name	Customer City
C10100	Ram	Cuttack
C10111	Shyam	Cuttack
C10115	Arun	BBSR
C10117	Vimal	Delhi
C10118	Neha	Delhi

Customer ID	Customer Name	Customer City
C10100	Ram	Cuttack
C10111	Shyam	Cuttack

σ (CUSTOMER)

Customer City="cuttack"

Projection

- Examples: $\pi_{age}(S2)$; $\pi_{sname,rating}(S2)$

- *Schema* of result: Retains only attributes that are in the "*projection list*".

- Projection operator has to *eliminate duplicates*

Projection

sid	sname	rating	age
28	yuppy	9	35.0
31	lubber	8	55.5
44	guppy	5	35.0
58	rusty	10	35.0

S2

sname	rating
yuppy	9
lubber	8
guppy	5
rusty	10

$\pi_{sname,rating}(S2)$

age
35.0
55.5

$\pi_{age}(S2)$

Relational Algebra : Basic Operations

Projection Operation (π)

Select the certain columns from the table and discard the other column

$$\pi_{\text{Attribute list}}(\text{Relation Name})$$

Retrieve the student name and registration number of the students

$$\pi_{\text{Name, Regd No}}(\text{STUDENT})$$

Relational Algebra : Basic Operations

Retrieve the employee ID, employee name and address of all employees

$$\pi_{\text{Employee ID, name, address}}(\text{EMPLOYEE})$$

Relational Algebra : Basic Operations

EXAMPLE

CUSTOMER

Customer ID	Customer Name	Customer City
C10100	Ram	Cuttack
C10111	Shyam	Cuttack
C10115	Arun	BBSR
C10117	Vimal	Delhi
C10118	Neha	Delhi

QUERY

π (CUSTOMER)

Customer_Name, Customer_city

Customer Name	Customer City
Ram	Cuttack
Shyam	Cuttack
Arun	BBSR
Vimal	Delhi
Neha	Delhi

Union and Set-Difference

- All of these operations take two input relations, which must be ***union-compatible***:
  - Same number of fields.
  - `Corresponding' fields have the same type.

Union

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

S1

sid	sname	rating	age
28	yuppy	9	35.0
31	lubber	8	55.5
44	guppy	5	35.0
58	rusty	10	35.0

S2

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0
44	guppy	5	35.0
28	yuppy	9	35.0

$S1 \cup S2$

Relational Algebra : Basic Operations

UNION

EMPLOYEE

Name	Emp No	Sal
aaa	10	10,000
bbb	11	20,000
ccc	12	30,000
aaa	13	25,000
ddd	14	40,000

S1

MANAGER

Name	Emp No	Salary
aaa	10	10,000
ccc	12	30,000
fff	19	45000

S2

S1 ∪ S2

Name	Emp No	Sal
aaa	10	10,000
bbb	11	20,000
ccc	12	30,000
ddd	14	40,000
fff	19	45,000
aaa	13	25000

The result is denoted by S1 ∪ S2 which is a relation that includes all the tuples that are either in relation 1 or in relation 2 or in both but without duplicate tuples

Relational Algebra : Basic Operations

ID	Name	Subject
100	Ankit	English
200	Pooja	Maths
300	Komal	Science

Relation R

ID	Name	Subject
100	Ankit	English
400	Kajol	French

Relation S

ID	Name	Subject
100	Ankit	English
200	Pooja	Maths
300	Komal	Science
400	Kajol	French

Relation R ∪ S

Relational Algebra : Basic Operations

The union operation can be performed only if:

1.Both the tables have the **same number of attributes**. Therefore the degree of both the tables should be the same.

2.The domain of the corresponding attribute in both the tables must be the same and compatible.

3.Suppose the domain of attribute A in table R1 is "integer" while the domain of attribute B in table R2 is "varchar"(set of characters).

4.If we try to perform the union operation on both the tables i.e. (R1 U R2), it would not be possible to do so.

5.However, if both the attributes have "integer" or "varchar" as their domain, we can successfully perform union operation.

6.The important point here is that the corresponding attribute should have the same domain.

Relational Algebra : Basic Operations

Essential points about union operation:

1.It can be performed only if the number of attributes is the same in both the tables.

2.The domain of the corresponding attribute must be the same.

3.Union operation removes the duplicate tuples, if any.

Relational Algebra : Basic Operations

SET DIFFERENCE (-)

EMPLOYEE

Name	Emp No	Sal
aaa	10	10,000
bbb	11	20,000
ccc	12	30,000
ddd	14	40,000

MANAGER

Emp No	Name	Salary
10	aaa	10,000
12	ccc	30,000
19	fff	45000

S1

S2

S1 – S2

Tuples present in S1 but not in S2

Name	Emp No	Sal
bbb	11	20,000
ddd	14	40,000

Difference

Course_1

C_id	C_name
11	Foundation C
21	C++
31	JAVA

Course_2

C_id	C_name
12	Python
21	C++

Course_1 - Course_2

C_id	C_name
11	Foundation C
31	JAVA

Relational Algebra : Basic Operations

EMPLOYEE

Name	Emp No	Sal
aaa	10	10,000
bbb	11	20,000
ccc	12	30,000
ddd	14	40,000

S1

MANAGER

Emp No	Name	Salary
10	aaa	10,000
12	ccc	30,000
19	fff	45000

S2

S2 – S1

Tuples present in S2 but not in S1

Name	Emp No	Salary
fff	19	45000

Set Difference

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

S1

sid	sname	rating	age
28	yuppy	9	35.0
31	lubber	8	55.5
44	guppy	5	35.0
58	rusty	10	35.0

S2

sid	sname	rating	age
22	dustin	7	45.0

$S1 - S2$

sid	sname	rating	age
28	yuppy	9	35.0
44	guppy	5	35.0

$S2 - S1$

Intersection

- Intersection takes two input relations, which must be ***union-compatible***.

- Q: How to express it using basic operators?

$$R \cap S = R - (R - S)$$

Intersection

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

S1

sid	sname	rating	age
28	yuppy	9	35.0
31	lubber	8	55.5
44	guppy	5	35.0
58	rusty	10	35.0

S2

sid	sname	rating	age
31	lubber	8	55.5
58	rusty	10	35.0

$$S1 \cap S2$$

Relational Algebra : Basic Operations

INTERSECTION (∩)

The result is represented as S1 ∩ S2 which will include only those tuples which are common to R1 and R2

EMPLOYEE

Name	Emp No	Sal
aaa	10	10,000
bbb	11	20,000
ccc	12	30,000
ddd	14	40,000

S1

MANAGER

Name	Emp No	Salary
aaa	10	10,000
ccc	12	30,000
fff	19	45000

S2

S1 ∩ S2

Name	Emp No	Sal
aaa	10	10,000
ccc	12	30,000

Relational Algebra : Basic Operations

Cartesian Product/Cross Product (×)

DEPARTMENT

Dept.No	Dloc
10	B1
20	B2
30	B3

EMPLOYEE

Emp No	Emp Name
101	a
102	b

DEPARTMENT × EMPLOYEE

Dept.No	Dloc	Emp No	Emp Name
10	B1	101	a
10	B1	102	b
20	B2	101	a
20	B2	102	b
30	B3	101	a
30	B3	102	b

Cross-Product

- S1 x R1: Each row of S1 paired with each row of R1.
- Also termed as, Cartesian Product or Cross Join.
- Q: How many rows in the result?

  Known by the cardinality

Cross-Product

Student

S_id	Name	Class	Age
1	Andrew	5	25
2	Angel	10	30
3	Anamika	8	35

Course

C_id	C_name
11	Foundation C
21	C++

Student X Course

S_id	Name	Class	Age	C_id	C_name
1	Andrew	5	25	11	Foundation C
1	Andrew	5	25	21	C++
2	Angel	10	30	11	Foundation C
2	Angel	10	30	21	C++
3	Anamika	8	35	11	Foundation C
3	Anamika	8	35	21	C++

Cross-Product

STUDENT

SNO	FNAME	LNAME
1	Albert	Singh
2	Nora	Fatehi

DETAIL

ROLLNO	AGE
5	18
9	21

STUDENT × DETAILS

SNO	FNAME	LNAME	ROLLNO	AGE
1	Albert	Singh	5	18
1	Albert	Singh	9	21
2	Nora	Fatehi	5	18
2	Nora	Fatehi	9	21

Cross-Product

A

Name	Age	Sex
Ram	14	M
Sona	15	F
Kim	20	M

B

ID	Course
1	DS
2	DBMS

A× B

Name	Age	Sex	ID	Course
Ram	14	M	1	DS
Ram	14	M	2	DBMS
Sona	15	F	1	DS
Sona	15	F	2	DBMS
Kim	20	M	1	DS
Kim	20	M	2	DBMS

Cross Product Example

sid	bid	day
22	101	10/10/96
58	103	11/12/96

R1

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

S1

S1 X R1 =

(sid)	sn a me	rating	age	(sid)	bid	day
22	dustin	7	45.0	22	101	10 / 1 0 /9
22	dustin	7	45.0	58	103	11 / 1 2 /9
31	lubber	8	55.5	22	101	10 / 1 0 /9
31	lubber	8	55.5	58	103	11 / 1 2 /9
58	rusty	10	35.0	22	101	10 / 1 0 /9
58	rusty	10	35.0	58	103	11 / 1 2 /9

Relational Algebra : Basic Operations

RENAME OPERATION ($\rho - rho$)

Rename operation is used to rename either a relation or the attributes in a relation

Syntax:

1. $\rho_{S}^{(R)}$

2. $\rho_{(B1, B2,B3\ldots\ldots Bn)}^{(R)}$

3. $\rho_{S(B1, B2,B3\ldots\ldots Bn)}^{(R)}$

S – is the new name given to the existing relation

R- is the existing relation

B1, B2,B3…..Bn – are the new name given to the attributes

Relational Algebra : Basic Operations

Example

Q. Rename the relation, Department to CGU Department

$$\rho_{\text{CGU Department}}(\text{Department})$$

Q. Rename the attributes name and id of the table student

$$\rho_{\text{Fname, Reg No}}(\text{name, id of the student})$$

Relational Algebra : Basic Operations

Q. Rename the attributes Name, age of the Table Department to (A,B)

ρ (Department)

(A,B)

Q. Rename the table name Project to Pro and its attributes to P, Q, R

ρ (Project)

Pro (P, Q, R)

Relational Algebra : Basic Operations

Renaming can be done:

1. Renaming a relation
2. Renaming an attribute
3. Renaming both

1. Renaming a Relation

Relation named Students and we want to change it to Final Year Students, the rename operation work as follows:

ρ (Students)

Final Year Students

Relational Algebra : Basic Operations

2. Rename an attribute

Relation named Students and we want to change its attributes, Student ID, Student Name to SID and SName, the rename operation works as follows:

$$\rho_{(SID,\ SName)}(Students)$$

3. Renaming both

Both the relation name and attributes of the students class:

$$\rho_{Final\ Year\ Students\ (SID,\ SName)}(Students)$$

Relational Algebra : Basic Operations

DIVISION OPERATION

Division Operator is used for queries which involves the 'all'
R1/R2= Tuples of R1 associated with all tuples of R2

Q. Retrieve the name of the subject that is taught in all courses

Courses

Name	Course
System	B. Tech
Database	M. Tech
Database	B. Tech
Algebra	B. Tech

÷

Course
B. Tech
M. Tech

=

Name
Database

Relational Algebra : Basic Operations

Q. Retrieve names of employees who work on all the projects that John, Smith works on

EMPLOYEE TABLE

Name	ENo	PNo
John	123	P1
Smith	123	P2
A	121	P3

÷

Works on the following

ENo	Pno	PName
123	P1	Market
123	P2	Sales

=

Result

ENo
123

Relational Algebra : Basic Operations

A

sno	pno
s1	p1
s1	p2
s1	p3
s1	p4
s2	p1
s2	p2
s3	p2
s4	p2
s4	p4

pno
p2

B1

pno
p2
p4

B2

pno
p1
p2
p4

B3

sno
s1
s2
s3
s4

A/B1 *(associated with B1)*

sno
s1
s4

A/B2 *(associated with 2 tuples of B2)*

sno
s1

A/B3 *(associated with 3 tuples of B3)*

Relational Algebra : Basic Operations

R1

A	B
a1	b1
a1	b2
a2	b2
a1	b1
a3	b2
a2	b1
a2	b3
a1	b1
a3	b1
a2	b3
a4	b1

B
b1

R2

B
b1
b2

R3

B
b1
b2
b3

R4

A
a1
a2
a3
a4

R1/R2

A
a1
a2
a3

R1/R3

A
a2

R1/R4

Relational Algebra : Basic Operations

Completed

Student	Task
Fred	Database1
Fred	Database2
Fred	Compiler1
Eugene	Database1
Eugene	Compiler1
Sarah	Database1
Sarah	Database2

DBProject

Task
Database1
Database2

Completed
÷
DBProject

Student
Fred
Sarah

Relational Algebra : Basic Operations

- Relations *r*, *s*:

A	B	C	D	E
α	a	α	a	1
α	a	γ	a	1
α	a	γ	b	1
β	a	γ	a	1
β	a	γ	b	3
γ	a	γ	a	1
γ	a	γ	b	1
γ	a	β	b	1

r

D	E
a	1
b	1

s

- $r \div s$:

A	B	C
α	a	γ
γ	a	γ

Relational Algebra : Basic Operations

R

A	B
α	1
α	2
α	3
β	1
γ	1
δ	1
δ	3
δ	4
ε	6
ε	1
β	2

S

B
1
2

R/S

A

A
α
β

Relational Algebra : Basic Operations

R

Col A	Col B
F	1
F	2
F	3
E	1
E	3
S	1
S	2

S

Col B
1
2

R/S

A

F
S

Join

- Joins are compound operators involving cross product, selection, and (sometimes) projection.

- Most common type of join is a "*natural join*" (often just called "join"). R⋈S conceptually is:
  - Compute R X S
  - Select rows where attributes that appear in both relations have equal values
  - Project all unique atttributes and one copy of each of the common ones.

Join ⋈

- Join is a binary operation which allows you to combine join product and selection in one single statement.
- The goal of creating a join condition is that it helps you to combine the data from two or more DBMS tables.
- There are mainly two types of joins in DBMS:

1. **Inner Joins: Theta, Natural, EQUI**
2. **Outer Join: Left, Right, Full**

Theta Join

- Theta Join is used to join two tables based on some conditions.
- The condition can be on any attributes of the tables performing Theta join. Any comparison operator can be used in the condition.

- $A \bowtie_{\theta} B$ where θ is the condition for join.

Example of Theta Join

Theta Join: Example

Product

Pname	Price
Laptop	1500
Car	20000

Component

PName	CName	Cost
Laptop	CPU	500
Laptop	hdd	300
Laptop	case	700
Car	wheels	1000

ProductInfo := Product $\bowtie_{\text{Product.pname=Component.PName}}$ Component

Pname	Price	Cname	Cost
Laptop	1500	CPU	500
Laptop	1500	hdd	300
Laptop	1500	case	700
Car	20000	wheels	1000

Example of Theta Join

THETA Join

Example:

Car

CarModel	CarPrice
CarA	20'000
CarB	30'000
CarC	50'000

Boat

BoatModel	BoatPrice
Boat1	10'000
Boat2	40'000
Boat3	60'000

$Car \bowtie Boat$

$CarPrice > BoatPrice$

CarModel	CarPrice	BoatModel	BoatPrice
CarA	20'000	Boat1	10'000
CarB	30'000	Boat1	10'000
CarC	50'000	Boat1	10'000
CarC	50'000	Boat2	40'000

- RESULT=Car $\bowtie_{\{CarPrice>BoatPrice\}}$ Boat;
- Result=R1$\bowtie_{\{Condition\}}$ R2; Condition: $\{<, >, =, \leq, \geq, \neq\}$;
- EquiJoin when "=".
- **SELECT** * **FROM** Car, Boat
  WHERE CarPrice>BoatPrice;

Example of Theta Join

Customer

C id	Cname	age
101	Ajay	20
102	Vijay	19
103	Sita	21

Order

O id	O name
101	Pizza
101	Noodles

Customer $\bowtie$ Customer.cid > Order.oid Order

C id	Cname	age	Oid	Oname
102	Vijay	19	101	Pizza
102	Vijay	19	101	Noodles
103	Sita	21	101	Pizza
103	Sita	21	101	Noodles

Example of Theta Join

Student_Detail

Roll No	Name	Address
1	Anoop	Delhi
2	Anurag	Noida

Student_Result

RollNo	Subject	Marks
1	DBMS	10
1	C++	20
2	Java	30
2	C++	40

Student_Detail $\bowtie$ Student_Detail.RollNo = Student_Result.RollNo (Student_Result)

Roll	Name	Address	Subject	Marks
1	Anoop	Delhi	DBMS	10
1	Anoop	Delhi	C++	20
2	Anurag	Noida	Java	30
2	Anurag	Noida	C++	40

Example of Theta Join

Student

SID	Name	Standard
101	Alex	10
102	Marie	11

Subject

Class	Subject
10	Maths
10	English
11	Music
11	Sports

Student $\bowtie$ Student.standard = Subject.class (Subject)

SID	Name	Standard	Class	Sub
101	Alex	10	10	Maths
101	Alex	10	10	English
102	Marie	11	11	Music
102	Marie	11	11	Sports

Natural Join

- Natural join can join tables based on the common columns in the tables being joined.
- A natural join returns all rows by matching values in common columns having same name and data type of columns and that column should be present in both tables.
- Both table must have at least one common column with same column name and same data type.
- Relation will get join with respect to common attributes
- Here we don't have any conditions
- **Natural join is a type of inner join in which we not need of any comparison operators. In natural join columns should have the same name and domain. There should be at least one common attribute between two tables.**

Example of Natural Join

S

SID	S Name	Dept ID
101	Raju	1
102	Ravi	2
103	Ramu	1
104	Hari	3
105	Suresh	5

D

Dept ID	DName
1	CSE
2	ECE
3	EEE
4	MECH

S ⋈ D

SID	S Name	Dept ID	DName
101	Raju	1	CSE
102	Ravi	2	ECE
103	Ramu	1	CSE
104	Hari	3	EEE

Example of Natural Join

COURSES

C ID	Course	Dept
CS01	DBE	CSE
ME01	Mechanics	ME
EE01	Electronics	EE

HOD

Dept	Head
CSE	Alex
ME	Maya
EE	Mira

COURSES ⋈ HOD

CID	Courses	Dept	Head
CS01	DBE	CSE	Alex
ME01	Mechanics	ME	Maya
EE01	Electronics	EE	Mira

Natural Join Example

Table A

Number	Square
2	4
3	9

Table B

Number	Cube
2	8
3	27

$A \bowtie B$

Number	Square	Cube
2	4	8
3	9	27

Natural Join Example

sid	bid	day
22	101	10/10/96
58	103	11/12/96

R1

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

S1

S1 ⋈ R1 =

sid	sname	rating	age	bid	day
22	dustin	7	45.0	101	10/10/96
58	rusty	10	35.0	103	11/12/96

EXAMPLE


NATURAL JOIN(INNER JOIN)
STU(S)
ROLL.NO NAME COURSE
101 PIYUSH M-01
102 SUMIT M-02
103 SHEENA M-04
104 RITU M-03
COURSE(C)
C_ID C_NAME C_DUR
M-01 MBA(HR) 2
M-02 MSC(PHY) 2
M-03 MSC(IT) 2
M-04 MCA 3
NATURAL JOIN ON S AND C
ROLL.NO NAME COURSE C_NAME C_DUR
101 PIYUSH M-01 MBA(HR) 2
102 SUMIT M-02 MSC(PHY) 2
103 SHEENA M-04 MCA 3
104 RITU M-03 MSC(IT) 2

Equi Join

When a theta join uses only equivalence condition, it becomes a equi join.

Student		
SID	**Name**	**Std**
101	Alex	10
102	Maria	11

Subjects	
Class	**Subject**
10	Math
10	English
11	Music
11	Sports

Student_Detail –

$$STUDENT \bowtie_{Student.Std = Subject.Class} SUBJECT$$

Student_detail				
SID	**Name**	**Std**	**Class**	**Subject**
101	Alex	10	10	Math
101	Alex	10	10	English
102	Maria	11	11	Music
102	Maria	11	11	Sports

Equi Join

EquiJoin

- In the case where the predicate θ contains only equality (=), the term **Equijoin** is used instead.

Customer

Cid	Cname	C_country
11	David	UK
12	Mary	US
13	John	China

Agent

Aid	Aname	A_country
101	Sara	China
102	Mike	Italy
103	Jim	UK

Customer $\bowtie_{\text{Customer.C\_country=Agent.A\_country}}$ **Agent**

Cid	Cname	C_country	Aid	Aname	A_country
11	David	UK	103	Jim	UK
13	John	China	101	Sara	China

Equi Join

Example

Sells(

bar,	beer,	price
Joe's	Bud	2.50
Joe's	Miller	2.75
Sue's	Bud	2.50
Sue's	Coors	3.00

)

Bars(

name,	addr
Joe's	Maple St.
Sue's	River Rd.

)

BarInfo := Sells JOIN $_{\text{Sells.bar = Bars.name}}$ Bars

BarInfo(

bar,	beer,	price,	name,	addr
Joe's	Bud	2.50	Joe's	Maple St.
Joe's	Miller	2.75	Joe's	Maple St.
Sue's	Bud	2.50	Sue's	River Rd.
Sue's	Coors	3.00	Sue's	River Rd.

)

Equi Join

State

State_ID	State_Name
1	Uttar Pradesh
2	Uttarakhand
3	Madhya Pradesh

City

City_ID	City_Name
1	Lucknow
1	Gorakhpur
1	Noida
2	Dehradun
2	Rishikesh
3	Gwalior

$$State \bowtie_{State.State\_Id=City.City\_Id} City$$

Equi Join

	State_ID	State_Name	city_ID	city_Name
▶	1	Uttar Pradesh	1	Lucknow
	1	Uttar Pradesh	1	Gorakhpur
	1	Uttar Pradesh	1	Noida
	2	Uttarakhand	2	Dehradun
	2	Uttarakhand	2	Rishikesh
	3	Madhya Pradesh	3	Gwalior

Equi Join

Faculty

FacID	Name	Department	Designation
1001	Ram	CSE	Asst. Prof
1002	Varun	Finance	Professor
1003	Rahul	Maths	Professor
1004	Aditya	Mech	Lecturer

Course

Course ID	Title	FID
CS502	DBMS	1001
CS511	OOPS	1005
CS430	CO	1003

Equi Join

$$Faculty \bowtie_{Faculty.FacID=Course.FID} Course$$

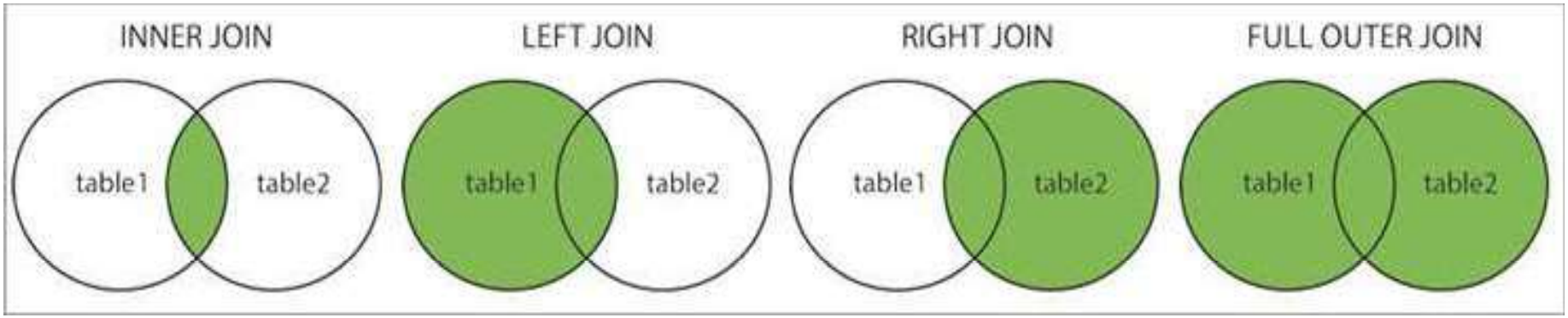
FacID	Name	Department	Designation	Course ID	Title	FID
1001	Ram	CSE	Asst. Professor	CS502	DBMS	1001
1003	Rahul	Maths	Professor	CS430	CO	1003

Outer Join

Outer join is a type of join that retrieve matching as well as non-matching records from related tables.

There three types of outer join

- Left outer join
- Right outer join
- Full outer join



Left outer Join (⟕)

- It is also called left join.
- This type of outer join retrieve all records from left table and retrieve matching record from right table.

Table A

Number	Square
2	4
3	9
4	16

Table B

Number	Cube
2	8
3	27
5	75

Left outer Join (⟕)

- This type of outer join retrieve all records from left table and retrieve matching record from right table.

A ⟕ B

Result:

Number	Square	Cube
2	4	8
3	9	27
4	16	–

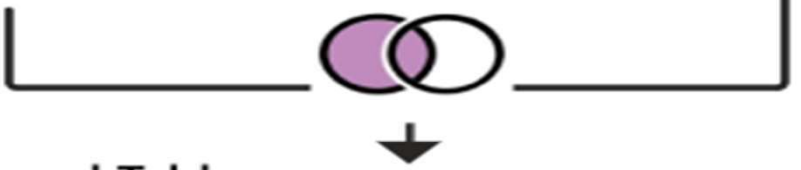
Left outer Join (⟕)

Left Table

Date	CountryID	Units
1/1/2020	1	40
1/2/2020	1	25
1/3/2020	3	30
1/4/2020	4	35

Right Table

ID	Country
1	USA
2	Canada
3	Panama



Merged Table

Date	CountryID	Units	Country
1/1/2020	1	40	USA
1/2/2020	1	25	USA
1/3/2020	3	30	Panama
1/4/2020	4	35	*null*

Left outer Join (⟕)

TABLE 1

Products	Price
Kiwis	$6
Onions	$3
Tomatoes	$7

TABLE 2

Products	Quantity
Kiwis	10
Onions	6
Broccoli	5

LEFT JOIN

Products	Price	Quantity
Kiwis	$6	10
Onions	$3	6
Tomatoes	$7	Null

Left outer Join (⟕)

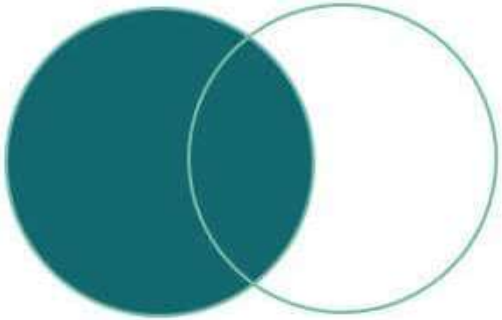
LEFT OUTER JOIN in SQL

Left Table

Food	Price
Pizza	$8
Burger	$6
French Fries	$4

Right Table

Food	Quantity
Pizza	3
Burger	5
Cake	2



Food	Price	Quantity
Pizza	$8	3
Burger	$6	5
French Fries	$4	Null

Left outer Join (⟕)

Left Table

id	name
1	Sayak
2	Alex
3	Sameer
4	Rick

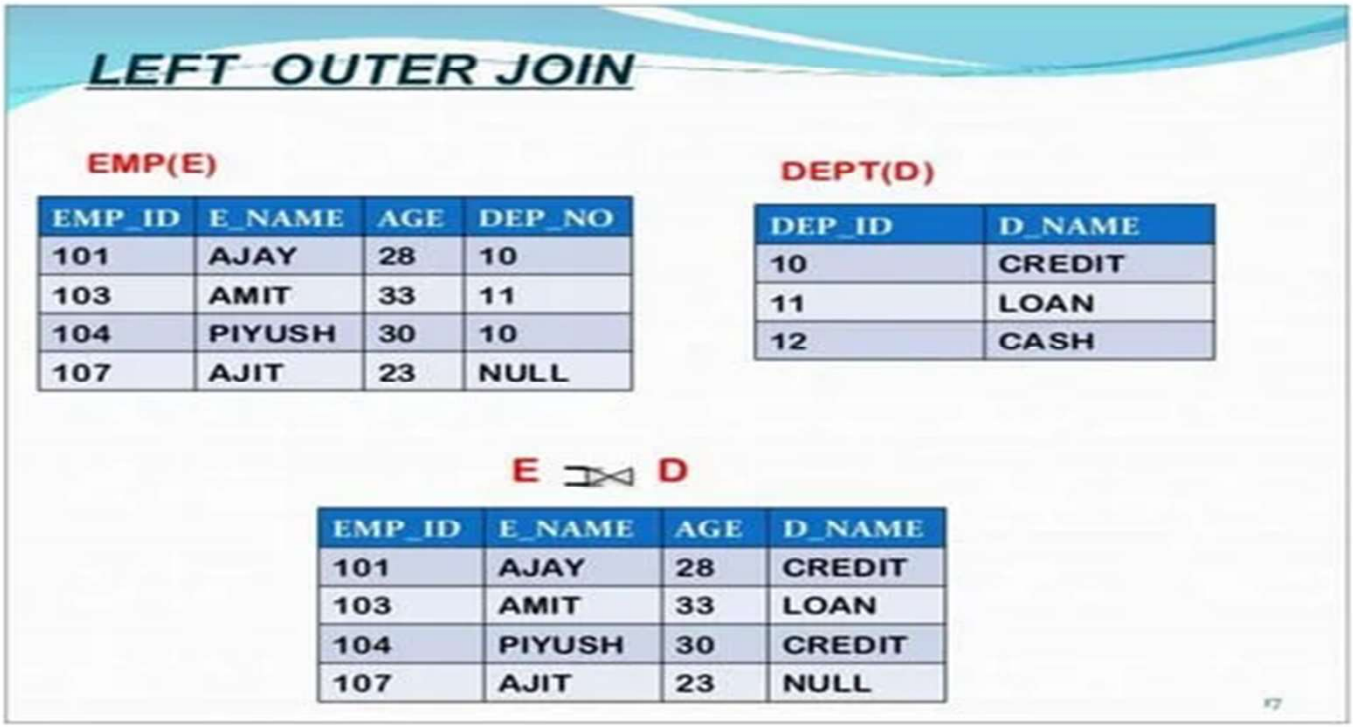
Right Table

id	stream
1	CS
1	IT
2	ECE
9	ECE

Left Join

id	name	stream
1	Sayak	CS
1	Sayak	IT
2	Alex	ECE
3	Sameer	
4	Rick	

Left outer Join (⟕)



Right outer Join (⟖)

- It is also called right join.
- This type of outer join retrieve all records from right table and retrieve matching record from right table.

Table A

Number	Square
2	4
3	9
4	16

Table B

Number	Cube
2	8
3	27
5	75

Right outer Join (⟖)

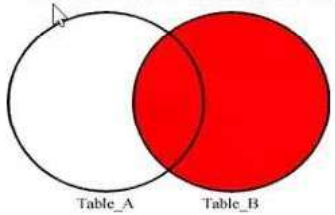
$A \rightouterjoin B$

Number	Square	Cube
2	4	8
3	9	27
5	–	75

Right outer Join (⋈)

RIGHT JOIN(⋈)

- RIGHT JOIN is similar to LEFT JOIN.
- This join returns all the rows of the table on the right side of the join and matching rows for the table on the left side of join.
- The rows for which there is no matching row on left side, the result-set will contain *null*.
- RIGHT JOIN is also known as RIGHT OUTER JOIN.



A

Id	Name
10	Jay
20	Veer
30	John

B

Name	Marks
Rohan	20
Veer	18
John	14
Sam	13

A ⋈ B

id	Name	Marks
Null	Rohan	20
20	Veer	18
30	John	14
Null	Sam	13

Right outer Join (⋈)

Left Table

Date	CountryID	Units
1/1/2020	1	40
1/2/2020	1	25
1/3/2020	3	30
1/4/2020	4	35

Right Table

ID	Country
3	Panama

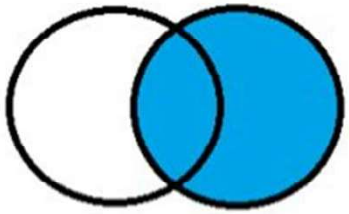
Merged Table

Date	CountryID	Units	Country
1/3/2020	3	30	Panama

Right outer Join (⟖)

Student ID	Name
1001	A
1002	B
1003	C
1004	D

Student ID	Department
1004	Mathematics
1005	Mathematics
1006	History
1007	Physics
1008	Computer Science



Student ID	Name	Department
1004	D	Mathematics
1005	NULL	Mathematics
1006	NULL	History
1007	NULL	Physics
1008	NULL	Computer Science

Full outer Join (⟗)

In full outer join all the rows from both table are inserted in result table

Table A

Number	Square
2	4
3	9
4	16

Table B

Number	Cube
2	8
3	27
5	75

Full outer Join (⟗)

$A \text{ ⟗ } B$

Number	Square	Cube
2	4	8
3	9	27
4	16	–
5	–	75

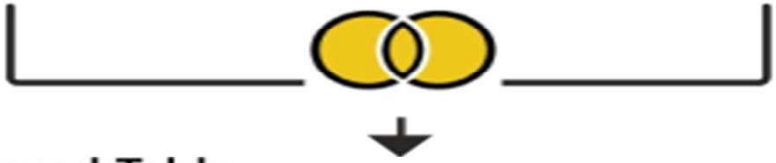
Full outer Join (⟗)

Left Table

Date	CountryID	Units
1/1/2020	1	40
1/2/2020	1	25
1/3/2020	3	30
1/4/2020	2	35

Right Table

ID	Country
1	USA
2	Canada
3	Panama
4	Spain



Merged Table

Date	CountryID	Units	Country
1/1/2020	1	40	USA
1/2/2020	1	25	USA
1/4/2020	2	35	Canada
1/3/2020	3	30	Panama
null	*null*	*null*	Spain

FULL OUTER JOIN

EMP(E)

EMP_ID	E_NAME	AGE	DEP_NO
101	AJAY	28	10
103	AMIT	33	11
104	PIYUSH	30	10
107	AJIT	23	NULL

DEPT(D)

DEP_ID	D_NAME
10	CREDIT
11	LOAN
12	CASH

E ⟗ D

EMP_ID	E_NAME	AGE	D_NAME
101	AJAY	28	CREDIT
103	AMIT	33	LOAN
104	PIYUSH	30	CREDIT
107	AJIT	23	NULL
NULL	NULL	NULL	CASH

Full outer Join (⟗)

Student

S_id	Name	Class	Age	C_type
1	Andrew	5	25	A
2	Angel	10	30	A
3	Anamika	8	35	C

Course

C_type	C_name
A	Foundation C
B	C++

Student ⟗ Course

S_id	Name	Class	Age	C_type	C_name
1	Andrew	5	25	A	Foundation C
2	Angel	10	30	A	Foundation C
3	Anamika	8	35	C	-
-	-	-	-	B	C++

Examples

ID	X1
1	a1
2	a2

ID	X2
2	b1
3	b2

inner_join

ID	X1	X2
2	a2	b1

left_join

ID	X1	X2
1	a1	NA
2	a2	b1

right_join

ID	X1	X2
2	a2	b1
3	NA	b2

full_join

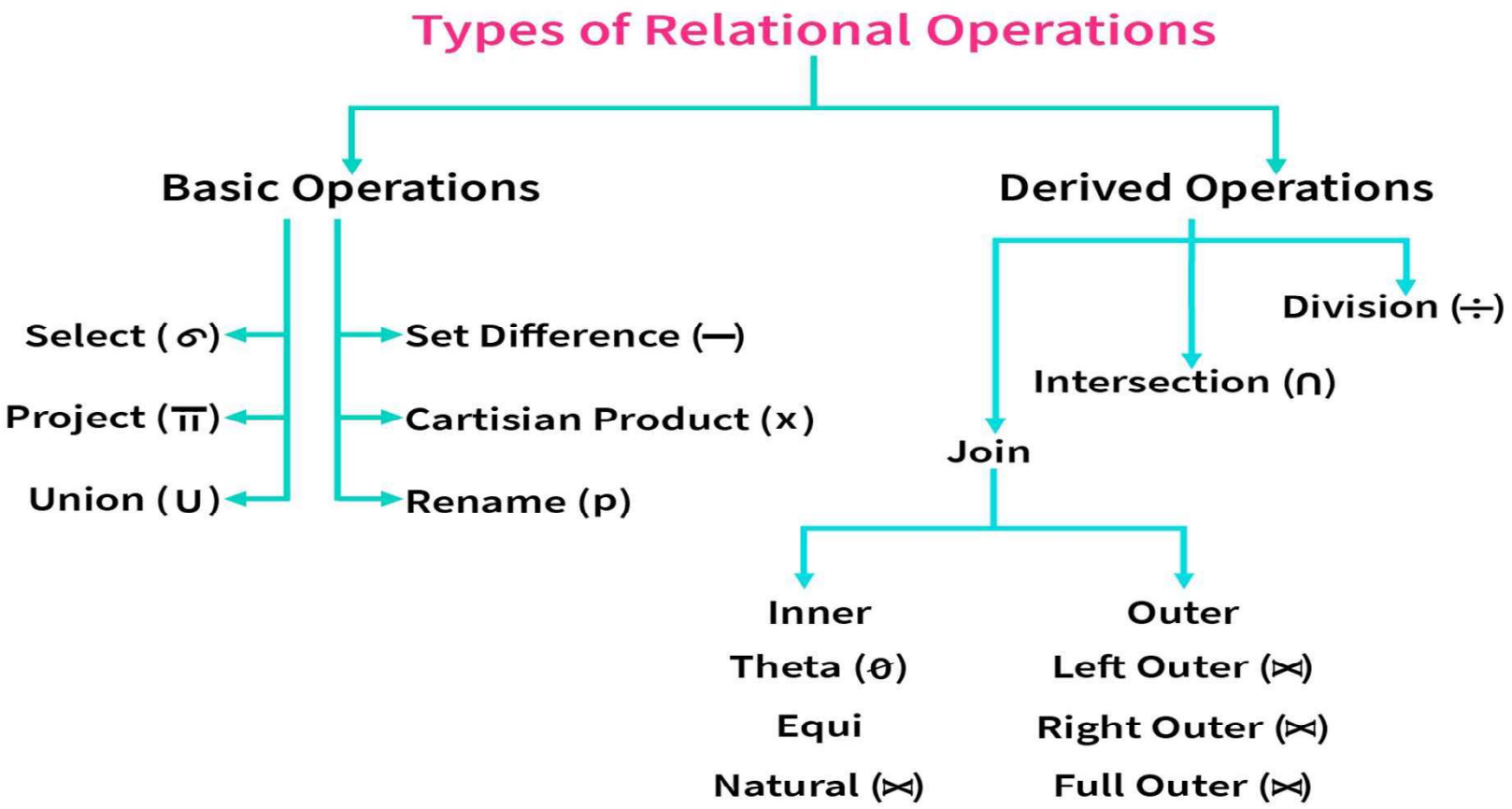
ID	X1	X2
1	a1	NA
2	a2	b1
3	NA	b2

semi_join

ID	X1
2	a2

anti_join

ID	X1
1	a1


Types of Relational Operations
Basic Operations
Derived Operations
Select (σ)
Project (π)
Union (U)
Set Difference (—)
Cartisian Product (x)
Rename (p)
Division (÷)
Intersection (∩)
Join
Inner
Theta (θ)
Equi
Natural (⋈)
Outer
Left Outer (⟕)
Right Outer (⟖)
Full Outer (⟗)