

Unit – 4

Relational Algebra & Implementation using SQL



Topics to be covered

- Keys of relations
- Set Operators
- Joins
- Sub queries

Definitions

Arity = No. of
Columns = 3

- Attribute = Column Name
- 2 attribute in same relation (table) can't have same name

Total No. of
Tuples =
Cardinality
of relation

Row =
Record =
Tuple

STAFF_ID	STAFF_NAME	STAFF_MNO
1	CEANS	8200106799
2	CECNK	1234567890
3	CENJR	123456789

Set of
permitted
values for that
attribute =
Domain

- Predefined row/column format for storing information.
- Relation = Table

Keys

▪ Super Key

- A super key is a set of one or more attributes whose values uniquely identifies each record within a relation.

Super Key (EnrollNo)		Super Key (RollNo, Branch, Semester)				
EnrollNo	RollNo	Name	Branch	Semester	SPI	BackLog
166620307001	101	Pritesh	CE	3	8	0
166620307002	102	Vaibha	CE	3	7	0
166620306001	101	Mihir	CI	3	8	0
166620306002	102	Darshan	CI	3	6	1
156620307001	101	Priya	CE	5	8	0
156620307002	102	Shrey	CE	5	7	1
156620306001	101	Niddhi	CI	5	7	1

Keys

- **Candidate Key**

- A candidate is a subset of a super key.
- A candidate key is the least combination of attribute that uniquely identifies each record in the table.
- The least combination of fields distinguishes a candidate key from a super key.

Keys

- Candidate Key

What is the difference between super key and candidate key??



Candidate Key
(RollNo, Branch, Semester)

EnrollNo	RollNo	Name	Branch	Semester	SPI	BackLog
166620307001	101	Pritesh	CE	3	8	0
166620307002	102	Vaibha	CE	3	7	0
166620306001	101	Mihir	CI	3	8	0
166620306002	102	Darshan	CI	3	6	1
156620307001	101	Priya	CE	5	8	0
156620307002	102	Shrey	CE	5	7	1
156620306001	101	Niddhi	CI	5	7	1

Keys

- Super Key V/S Candidate Key

Super Key
(EnrollNo, Branch)

EnrollNo alone works as a
super key??

Answer is **Yes**

So (EnrollNo, Branch) is **super
key but not candidate key.**

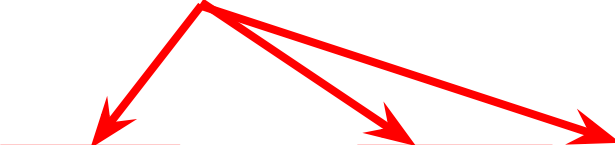
EnrollNo	RollNo	Name	Branch	Semester	SPI	BackLog
166620307001	101	Pritesh	CE	3	8	0
166620307002	102	Vaibha	CE	3	7	0
166620306001	101	Mihir	CI	3	8	0
166620306002	102	Darshan	CI	3	6	1
156620307001	101	Priya	CE	5	8	0
156620307002	102	Shrey	CE	5	7	1
156620306001	101	Niddhi	CI	5	7	1

Keys

- Super Key V/S Candidate Key

Super Key
(RollNo, Branch, Semester)

RollNo and Branch alone works
as a super key??
Answer is **No**




EnrollNo	RollNo	Name	Branch	Semester	SPI	BackLog
166620307001	101	Pritesh	CE	3	8	0
166620307002	102	Vaibha	CE	3	7	0
166620306001	101	Mihir	CI	3	8	0
166620306002	102	Darshan	CI	3	6	1
156620307001	101	Priya	CE	5	8	0
156620307002	102	Shrey	CE	5	7	1
156620306001	101	Niddhi	CI	5	7	1

Keys

- Super Key V/S Candidate Key

Super Key
(RollNo, ~~Branch~~, Semester)

RollNo and Semester alone
works as a super key??
Answer is **No**




EnrollNo	RollNo	Name	Branch	Semester	SPI	BackLog
166620307001	101	Pritesh	CE	3	8	0
166620307002	102	Vaibha	CE	3	7	0
166620306001	101	Mihir	CI	3	8	0
166620306002	102	Darshan	CI	3	6	1
156620307001	101	Priya	CE	5	8	0
156620307002	102	Shrey	CE	5	7	1
156620306001	101	Niddhi	CI	5	7	1

Keys

- Super Key V/S Candidate Key

Super Key
(~~RollNo~~, Branch, Semester)

Branch and Semester alone
works as a super key??
Answer is **No**




EnrollNo	RollNo	Name	Branch	Semester	SPI	BackLog
166620307001	101	Pritesh	CE	3	8	0
166620307002	102	Vaibha	CE	3	7	0
166620306001	101	Mihir	CI	3	8	0
166620306002	102	Darshan	CI	3	6	1
156620307001	101	Priya	CE	5	8	0
156620307002	102	Shrey	CE	5	7	1
156620306001	101	Niddhi	CI	5	7	1

Keys

- Super Key V/S Candidate Key

Super Key
(RollNo, Branch, Semester)

(RollNo, Branch, Semester) is
**super key as well as candidate
key.**



EnrollNo	RollNo	Name	Branch	Semester	SPI	BackLog
166620307001	101	Pritesh	CE	3	8	0
166620307002	102	Vaibha	CE	3	7	0
166620306001	101	Mihir	CI	3	8	0
166620306002	102	Darshan	CI	3	6	1
156620307001	101	Priya	CE	5	8	0
156620307002	102	Shrey	CE	5	7	1
156620306001	101	Niddhi	CI	5	7	1

Keys

- **Primary key**

- A Primary key is chosen by database designer to identify tuples uniquely in a relation.

- **Alternate key**

- If any table have more than one candidate key, then after choosing primary key from those candidate key, rest of candidate keys are known as an alternate key of that table.

Keys

- Primary Key V/S Alternate Key

Primary Key
EnrollNo

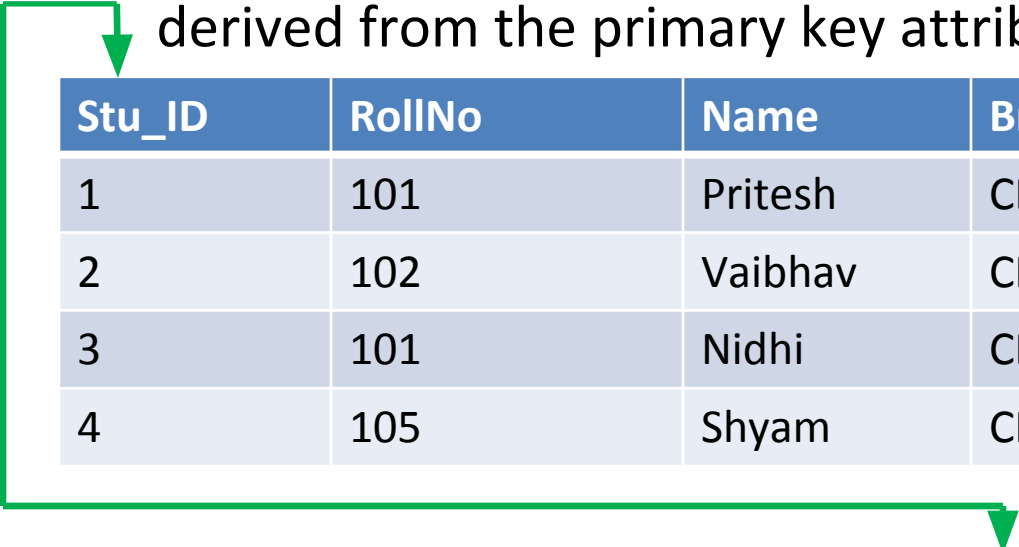
Alternate Key
(RollNo, Branch, Semester)

EnrollNo	RollNo	Name	Branch	Semester	SPI	BackLog
166620307001	101	Pritesh	CE	3	8	0
166620307002	102	Vaibha	CE	3	7	0
166620306001	101	Mihir	CI	3	8	0
166620306002	102	Darshan	CI	3	6	1
156620307001	101	Priya	CE	5	8	0
156620307002	102	Shrey	CE	5	7	1
156620306001	101	Niddhi	CI	5	7	1

Keys

Foreign key

- A foreign key is a set of one or more attributes whose values are derived from the primary key attribute of another relation.

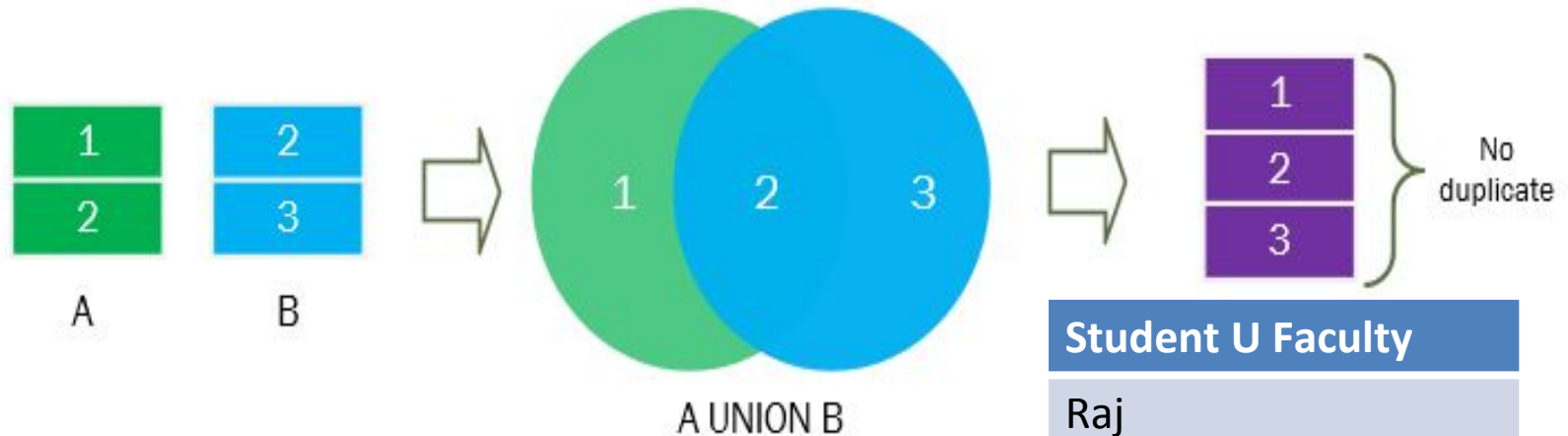


Stu_ID	RollNo	Name	Branch	Semester	SPI	BackLog
1	101	Pritesh	CE	3	9	0
2	102	Vaibhav	CI	3	8	0
3	101	Nidhi	CE	5	7	1
4	105	Shyam	CI	5	8	2

Fac_ID	Stu_ID	Fac_SN	Fac_Name	Branch
1	1	501	Akash	CE
2	2	502	Mohit	CI
3	4	504	Nitin	CE
4	1	515	Harnish	CI

Union operator

- **Symbol:** \cup
- **Notation:** $\text{Relation1} \cup \text{Relation2}$
- **Operation:** Combine the records from two or more tables (sets) into a single table (set), without duplicates.



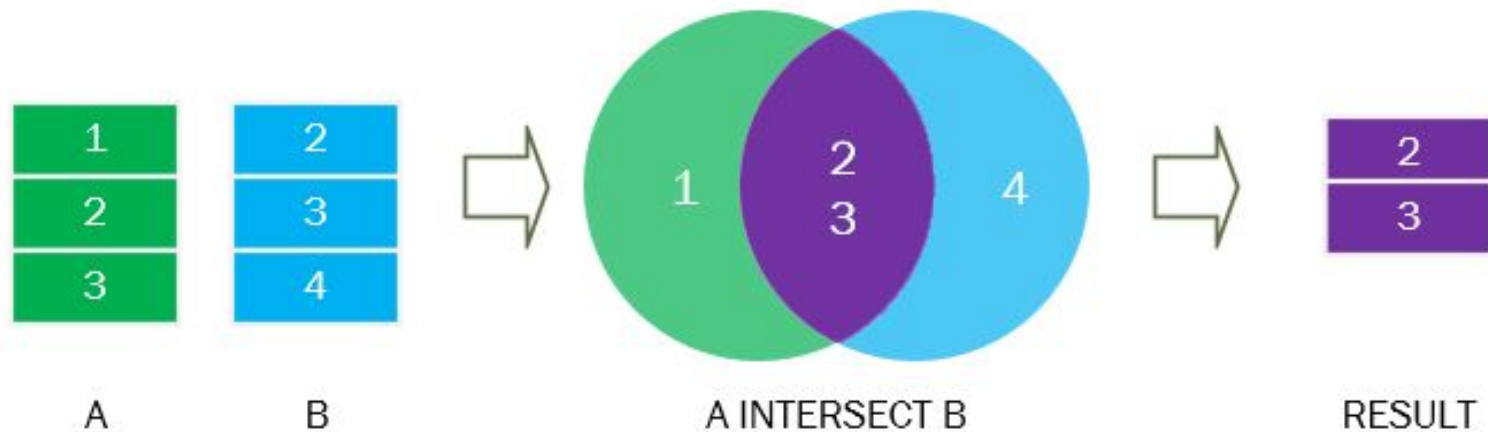
Student
Raj
Suresh
Meet

Faculty
Nitin
Raj
Akash

Student U Faculty
Raj
Suresh
Meet
Nitin
Akash

Intersect operator

- **Symbol:** \cap (Intersection)
- **Notation:** $\text{Relation1} \cap \text{Relation2}$
- **Operation:** Returns the records which are common from both relations.



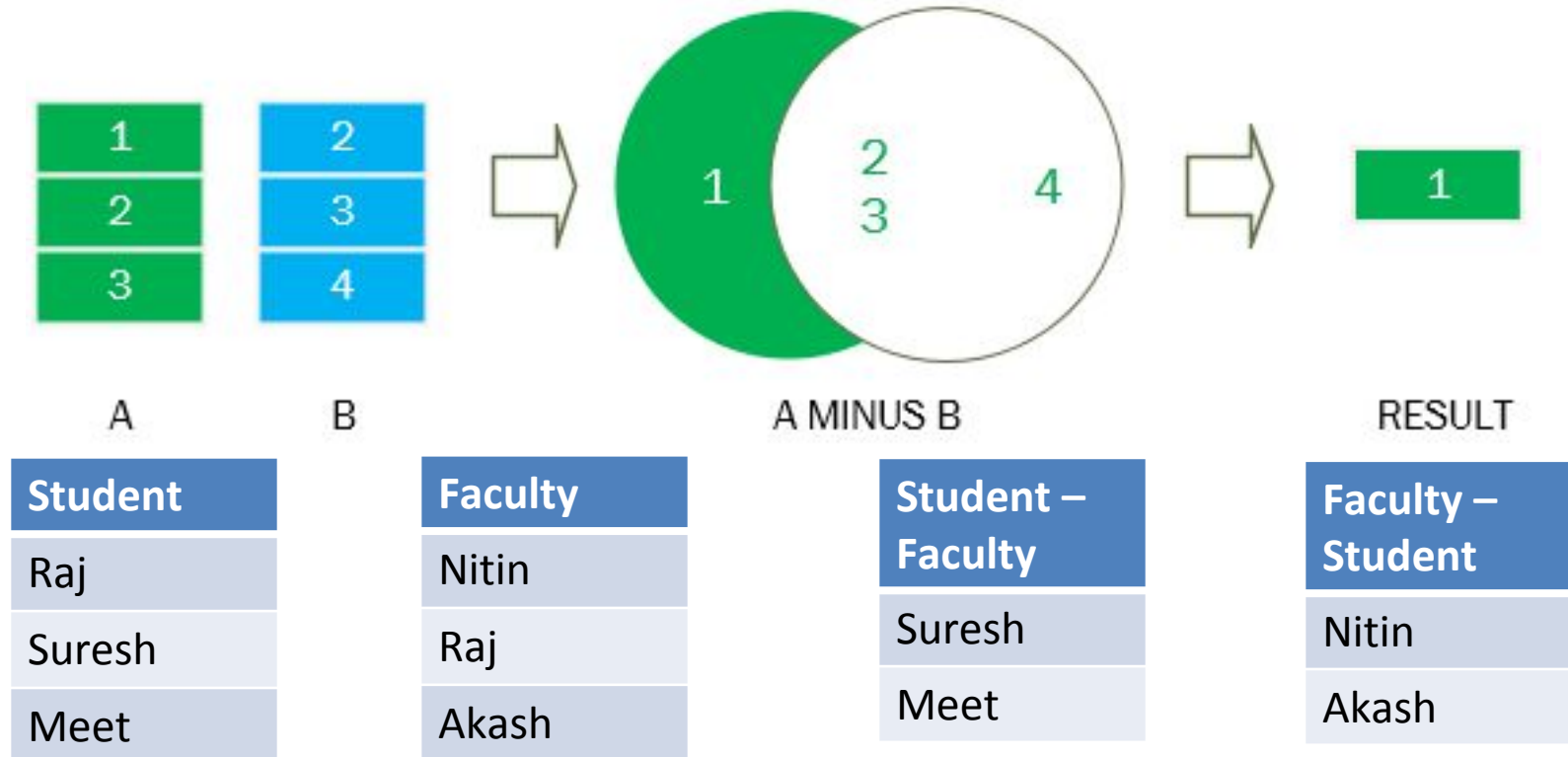
Student
Raj
Suresh
Meet

Faculty
Nitin
Raj
Akash

Student \cap Faculty
Raj

Minus/Difference Operator

- **Symbol:** $-$
- **Notation:** $\text{Relation1} - \text{Relation2}$
- **Operation:** Returns all the records from first (left) relation that are not contained in the second relation.



Create table

- Table: Department

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>DEPARTMENT</u>	<u>DEP_ID</u>	NUMBER	22	-	-	-	-	-	-
	<u>DEP_NAME</u>	VARCHAR2	20	-	-	-	✓	-	-
	<u>DEP_INITIAL</u>	VARCHAR2	5	-	-	-	✓	-	-
									1 - 3

DEP_ID	DEP_NAME	DEP_INITIAL
1	Civil	CI
2	Computer	CE
3	Electrical	EE
4	Mechanical	ME
5	Electronics	EC
6	Information Tech.	IT
7	Automobile	AU
8	Instrument	IC
9	Plastic Eng	PE
10	Production Eng	PE

Create table

- Table: Staff

STAFF_ID	STAFF_NAME	STAFF_INITIAL	STAFF_MNO	STAFF_ADDRESS	STAFF_SALARY	STAFF_HIREDATE	STAFF_TYPE	DEP_ID	STAFF_HEAD
101	Akash Siddhpura	CEANS	9429794513	Jamnagar	12345	01/08/2012	Lecturer	2	102
103	Chintan Kanani	CECNK	121456789	Rajkot	12378	01/06/2010	Lecturer	1	102
105	Ravi Shrimali	CERGS	4567891230	Ahmedabad	1234	01/03/2013	Lecturer	6	102
102	Nitin Rola	CENJR	1234567890	Junagadh	100000	01/06/2009	Head	2	-
104	Vishal Makwana	CEVKM	1237894560	Jetpur	123401	01/08/2012	Lecturer	4	102
106	Payal Boda	CEPMB	1230456078	Rajkot	-	01/06/2015	Lecturer	3	103
107	Shubham Kansara	CESDK	4561237809	Rajkot	1234	07/01/2017	Lecturer	5	101
108	Vikram Mevasa	CEVDM	1230004567	Rajkot	1235	07/01/2017	Lecturer	5	102

Create table

- Table: Student

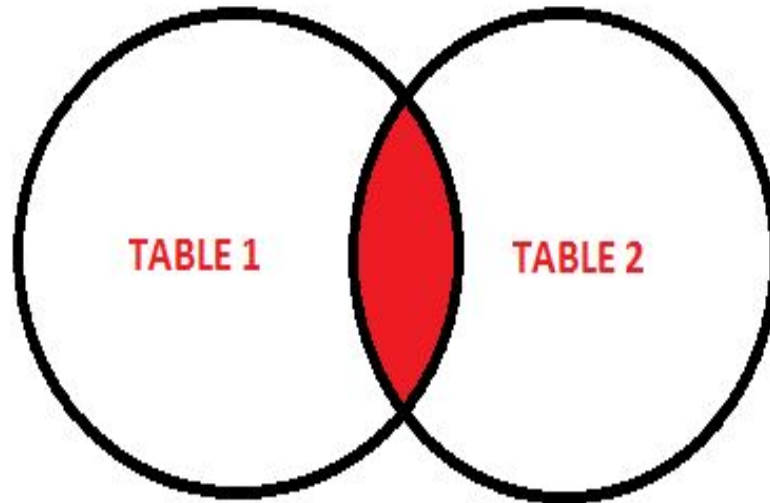
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>STUDENT</u>	<u>STU_ID</u>	NUMBER	-	5	0	-	✓	-	-
	<u>STU_NAME</u>	VARCHAR2	20	-	-	-	✓	-	-
	<u>STU_LASTNAME</u>	VARCHAR2	20	-	-	-	✓	-	-
	<u>STU_DEPARMENT</u>	NUMBER	-	5	0	-	✓	-	-
	<u>STU_MOBILE</u>	NUMBER	-	10	0	-	✓	-	-
	<u>STU_CITY</u>	VARCHAR2	10	-	-	-	✓	-	-

STU_ID	STU_NAME	STU_LASTNAME	STU_DEPARMENT	STU_MOBILE	STU_CITY
151	Pritesh	Ramani	2	8866221144	Rajkot
152	Vaibhav	Sorthiya	6	8877445566	Rajkot
153	Reena	Parmar	4	1237894560	Junagadh
154	Tina	Jadeja	5	8200145566	Jamnagar
155	Gopal	Verma	4	1122334455	Ahmedabad
156	Raj	Sharma	3	4455661122	Baroda
156	Krunal	Zala	2	8822665544	Junagadh
157	Mohammad	Khan	2	7878989822	Jamnagar

JOIN

- Purpose of join is to combine the data across tables.

Equi join



- A join which contains an equal operator (=) to join the tables.
- While performing joining if column names are different then also it will execute.

Equi join

DEP_ID	DEP_NAME	DEP_INITIAL
1	Civil	CI
2	Computer	CE
3	Electrical	EE
4	Mechanical	ME
5	Electronics	EC
6	Information Tech.	IT
7	Automobile	AU
8	Instrument	IC
9	Plastic Eng	PE
10	Production Eng	PE

u

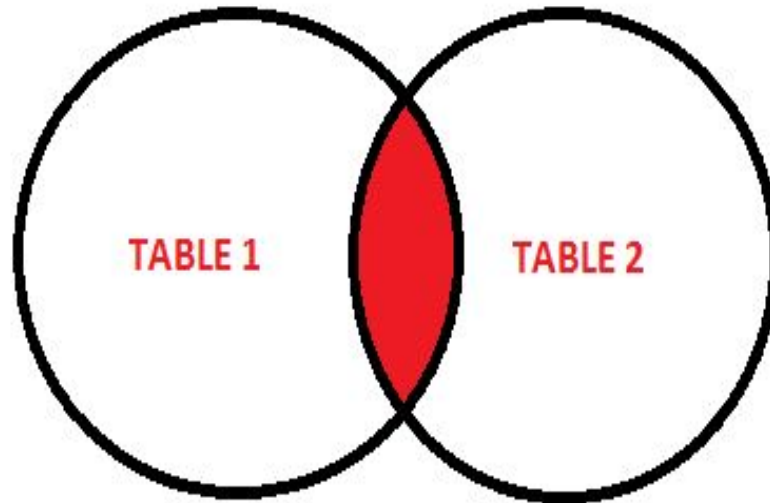
n

STU_ID	STU_NAME	STU_LASTNAME	STU_DEPARMENT	STU_MOBILE	STU_CITY
151	Pritesh	Ramani	2	8866221144	Rajkot
152	Vaibhav	Sorthiya	6	8877445566	Rajkot
153	Reena	Parmar	4	1237894560	Junagadh
154	Tina	Jadeja	5	8200145566	Jamnagar
155	Gopal	Verma	4	1122334455	Ahmedabad
156	Raj	Sharma	3	4455661122	Baroda
156	Krunal	Zala	2	8822665544	Junagadh
157	Mohammad	Khan	2	7878989822	Jamnagar

■ Exa

STU_NAME	STU_LASTNAME	DEP_NAME	DEP_INITIAL
Pritesh	Ramani	Computer	CE
Vaibhav	Sorthiya	Information Tech.	IT
Reena	Parmar	Mechanical	ME
Tina	Jadeja	Electronics	EC
Gopal	Verma	Mechanical	ME
Raj	Sharma	Electrical	EE
Krunal	Zala	Computer	CE
Mohammad	Khan	Computer	CE

Non-equi join



- A join which contains all operators ($\neq, \geq, \leq, <, >, <>$) except ($=$) to join the tables.
- While performing joining if column names are different then also it will execute.

Non-equi join

DEP_ID	DEP_NAME	DEP_INITIAL
1	Civil	CI
2	Computer	CE
3	Electrical	EE
4	Mechanical	ME
5	Electronics	EC
6	Information Tech.	IT
7	Automobile	AU
8	Instrument	IC
9	Plastic Eng	PE
10	Production Eng	PE

STU_ID	STU_NAME	STU_LASTNAME	STU_DEPARMENT	STU_MOBILE	STU_CITY
151	Pritesh	Ramani	2	8866221144	Rajkot
152	Vaibhav	Sorthiya	6	8877445566	Rajkot
153	Reena	Parmar	4	1237894560	Junagadh
154	Tina	Jadeja	5	8200145566	Jamnagar
155	Gopal	Verma	4	1122334455	Ahmedabad
156	Raj	Sharma	3	4455661122	Baroda
156	Krunal	Zala	2	8822665544	Junagadh
157	Mohammad	Khan	2	7878989822	Jamnagar

■ Exan

S

F

W

STU_NAME	STU_LASTNAME	DEP_NAME	DEP_INITIAL
Vaibhav	Sorthiya	Computer	CE
Reena	Parmar	Computer	CE
Tina	Jadeja	Computer	CE
Gopal	Verma	Computer	CE
Raj	Sharma	Computer	CE
Pritesh	Ramani	Electronics	EC
Vaibhav	Sorthiya	Electronics	EC
Reena	Parmar	Electronics	EC
Gopal	Verma	Electronics	EC
Raj	Sharma	Electronics	EC
More than 10 rows available. Increase rows selector to view more rows.			

Self join



- Self join is a join in which a table is joined with itself.
- Means that each row of the table is combined with itself and with every row of the table.
- After that it returns rows of the table that satisfy the join condition.
- Table name appears two times after from clause with 2 different aliases

Self join

- **Syntax:**

select a.column
from table a, ta
where a.column

STAFF_ID	STAFF_NAME	STAFF_NAME
101	Shubham Kansara	Akash Siddhpura
103	Payal Boda	Chintan Kanani
102	Vikram Mevasa	Nitin Rola
102	Vishal Makwana	Nitin Rola
102	Ravi Shrimali	Nitin Rola
102	Chintan Kanani	Nitin Rola
102	Akash Siddhpura	Nitin Rola

column

- **Example:**

```
SELECT Emp.staff_id, Super.staff_name, Emp.staff_name  
FROM Staff Super, Staff Emp  
WHERE Super.staff_head = Emp.staff_id;
```

Natural Join

- **Symbol:** \bowtie
- **Notation:** $Relation1 \bowtie Relation2$
- **Operation:** Natural join will retrieve information (consistent data) from multiple relations.
- *Here Must have common column name in both tables.*

Steps

1. It performs Cartesian product
2. Then it finds consistent tuples and inconsistent tuples are deleted
3. Then it deletes duplicate attributes

Natural Join

- **Operation:**

- Step 1 : It performs Cartesian product

Student

RollNo	Name	Branch
101	Raj	CE
102	Meet	ME

RollNo	SPI
101	8
103	9

Result

Step 1

Student.RollNo	Name	Branch	Result.RollNo	SPI
101	Raj	CE	101	8
101	Raj	CE	103	9
102	Meet	ME	101	8
102	Meet	ME	103	9

Natural Join

■ Operation:

- Step 2 : Finds consistent tuples and inconsistent tuples are deleted

<i>Student</i>	RollNo	Name	Branch		RollNo	SPI	<i>Result</i>
	101	Raj	CE		101	8	
	102	Meet	ME		103	9	

Step 2	Student.RollNo	Name	Branch	Result.RollNo	SPI	Inconsistent Records
	101	Raj	CE	101	8	
	101	Raj	CE	103	9	
	102	Meet	ME	101	8	
	102	Meet	ME	103	9	

Natural Join

- **Operation:**

- Step 2 : Finds consistent tuples and inconsistent tuples are deleted

<i>Student</i>	RollNo	Name	Branch			<i>Result</i>
	101	Raj	CE	101	8	
	102	Meet	ME	103	9	

Step 2	Student			Result	
	RollNo	Name	Branch	RollNo	SPI
	101	Raj	CE	101	8

Natural Join

- **Operation:**
 - Step 3 : Delete duplicate attributes

Student

RollNo	Name	Branch
101	Raj	CE
102	Meet	ME

RollNo	SPI
101	8
103	9

Result

Step 3

RollNo	Name	Branch	SPI
101	Raj	CE	8

Exercise of Natural Join

1. List out the staffname with department name
 2. List out the staffname whose department is “Computer”
 3. List out the staffname with their department name and salary having salary more than 25000 and belongs to “Computer” department.
- **ANS1** : SELECT staff_name,dep_name FROM Staff JOIN Department USING (Dep_ID);
 - **ANS2** : SELECT staff_name,dep_name FROM Staff JOIN Department USING (Dep_ID) where Dep_ID=2;
 - **ANS3** : SELECT staff_name,dep_name,staff_salary FROM Staff JOIN Department USING (Dep_ID) where Dep_ID=2 and staff_salary>25000;

Outer Join

- **Operation:** In natural join some records are missing if we want that missing records then we have to use outer join.
- **Types:** Three types of Outer Join
 1. Left Outer Join
 2. Right Outer Join
 3. Full Outer Join

Left Outer Join

- **Left Outer Join:** Display all the tuples of the left relation even through there is no matching tuple in the right relation.
- For such kind of tuples having no matching, the attributes of right relation will be padded with null in resultant relation.
- **Example:**

Input

Student

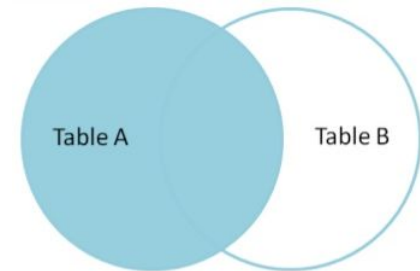
RollNo	Name	Branch
101	Raj	CE
102	Meet	ME

Result

RollNo	SPI
101	8
103	9

Student \bowtie Result

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



Left Outer Join

- **Syntax:**

```
SELECT column1, column2,  
FROM table1 LEFT OUTER  
ON table1.column_name =
```

STAFF_NAME	STAFF_INITIAL	DEP_NAME
Akash Siddhpura	CEANS	Computer
Chintan Kanani	CECNK	Civil
Ravi Shrimali	CERGS	Information Tech.
Nitin Rola	CENJR	Computer
Vishal Makwana	CEVKM	Mechanical
Payal Boda	CEPMB	Electrical
Shubham Kansara	CESDK	Electronics
Vikram Mevasa	CEVDM	Electronics
-	-	Instrument
-	-	Production Eng
-	-	Automobile
-	-	Plastic Eng

- **Example:**

```
select staff_name,staff_initial, dep_name  
from department left outer join staff  
on department.dep_id = staff.dep_id;
```

Right Outer Join

- **Right Outer Join:** Right outer join returns 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:**

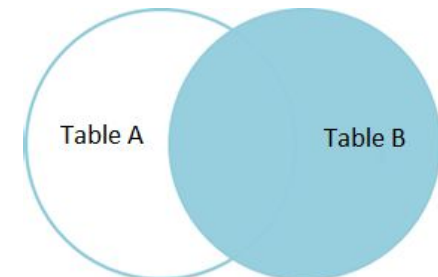
Input

RollNo	Name	Branch
101	Raj	CE
102	Meet	ME

RollNo	SPI
101	8
103	9

Student ⋈ Result

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



Right Outer Join

- **Syntax:**

```
SELECT column1, column2  
FROM table1 RIGHT OUTER JOIN table2  
ON table1.column_name = table2.column_name
```

STU_NAME	STU_LASTNAME	DEP_NAME
Pritesh	Ramani	Computer
Vaibhav	Sorthiya	Information Tech.
Reena	Parmar	Mechanical
Tina	Jadeja	Electronics
Gopal	Verma	Mechanical
Raj	Sharma	Electrical
Krunal	Zala	Computer
Mohammad	Khan	Computer
-	-	Instrument
-	-	Civil
-	-	Production Eng
-	-	Automobile
-	-	Plastic Eng

- **Example:**

```
select staff_name, staff_initial, dep_name  
from department right outer join student  
on department.dep_id = student.dep_id;
```

Full Outer Join

- **Full Outer Join:** The full outer join returns all the tuples of both of the relations. It also pads null values whenever required.
- For such kind of tuples having no matching, it will be padded with null in resultant relation.
- **Example:**

Input

Student

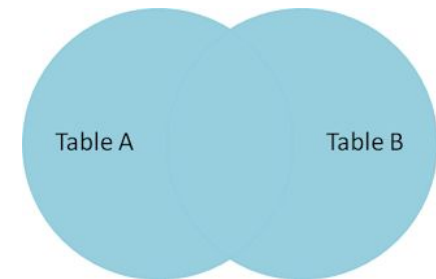
RollNo	Name	Branch
101	Raj	CE
102	Meet	ME

Result

RollNo	SPI
101	8
103	9

Student \bowtie Result

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



Full Outer Join

- **Syntax:**

```
SELECT column1, column2  
FROM table1 FULL OUTER JOIN table2  
ON table1.column_name = table2.column_name
```

STU_NAME	STU_LASTNAME	DEP_NAME
Mohammad	Khan	Computer
Krunal	Zala	Computer
Pritesh	Ramani	Computer
Tina	Jadeja	Electronics
-	-	Automobile
-	-	Instrument
-	-	Plastic Eng
-	-	Production Eng
-	-	Civil
Raj	Sharma	Electrical
Gopal	Verma	Mechanical
Reena	Parmar	Mechanical
Vaibhav	Sorthiya	Information Tech.

- **Example:**

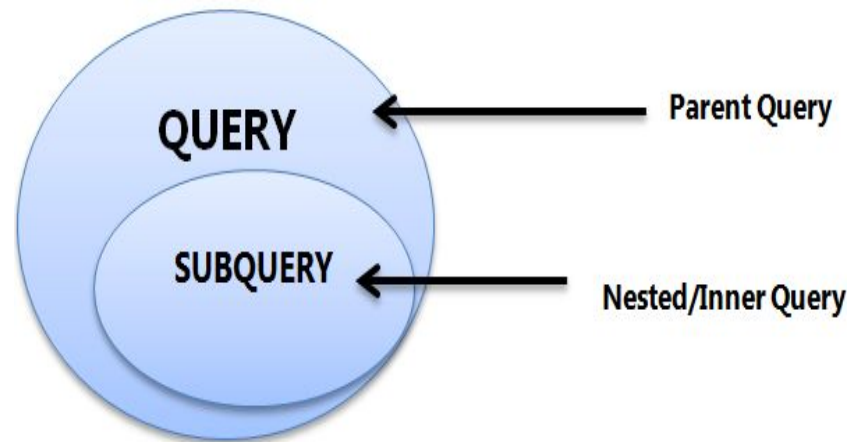
```
select staff_name, staff_initial, dep_name  
from department full outer join staff  
on department.dep_id = staff.dep_id;
```


Sub Query

- Query in a query.
- Sub query is usually added in a where clause.
- We will use sub query when we want to search some data using select query but we don't know the exact value from data.

- Types of sub query,

1. Single row sub query
2. Multiple row sub query
3. Correlated sub query



Single row sub query

- Returns 0 or 1 row
- Can be used with <,>,<=,>= etc operators.
- **Example:** Find out the name of staff whose salary is maximum.

```
select staff_name from  
staff
```

```
where staff_salary=(select max(staff_salary) from staff);
```



Single row sub query

Multiple row sub query

- Returns one or more rows
- Can be used with IN, NOT IN, ANY, ALL etc operators.
- **Example:** Find out the name of staff who are from “Computer” department using sub query.

SELECT * FROM

staff

WHERE dep_id IN

(SELECT dep_id FROM department where dep_id='2');




Multiple row sub query

Correlated sub query

- If a sub query references columns in the parent query.
- This makes it impossible to evaluate the sub query before evaluating the parent query.
- **Example:** Find out the name of staff who earn less salary then average salary.

```
SELECT staff_name FROM  
staff  
WHERE staff_salary <  
(select avg(staff_salary) from staff);
```



Correlated sub query

Important Questions

Sr. No.	Questions	Count
1	List and explain types of sub queries with example	4
2	Explain following operations of relational algebra (1) Intersect (2) Union (3) Minus	4
3	Explain Outer join in detail	4
4	List different types of Joins. Explain Equi-join, Simple Join & Natural with example	4

Marks Covered – 16*

ANY
QUESTIONS?

