SQL

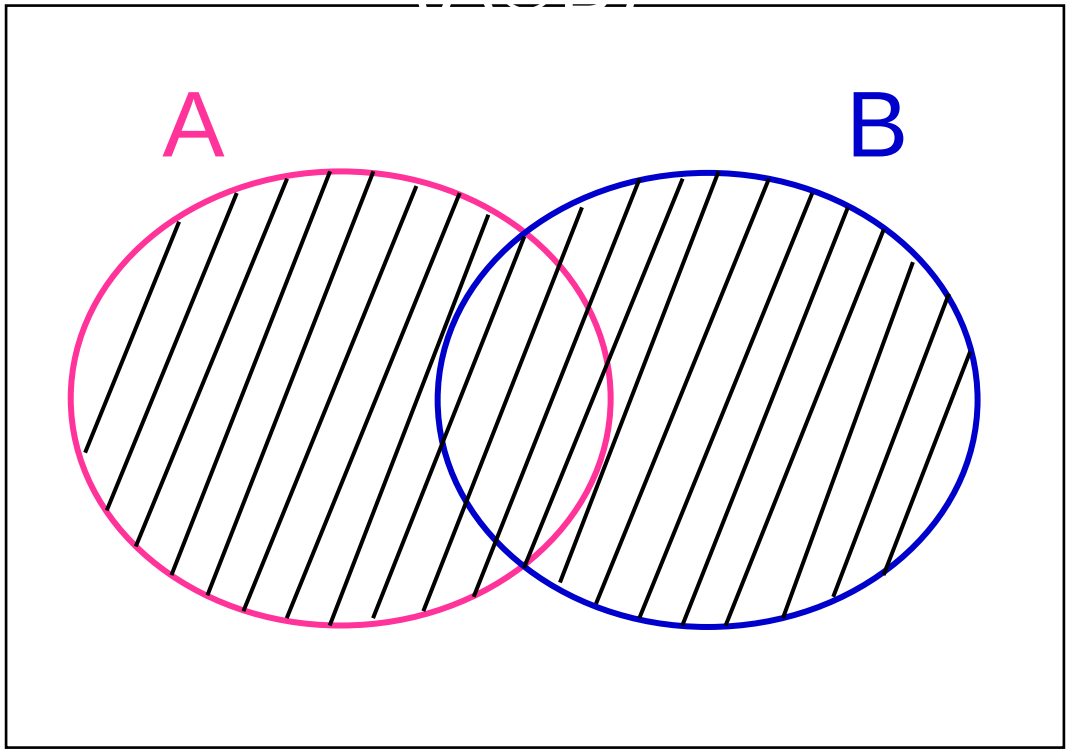
Structured Query Language

Dr. Jenila Livingston L.M.
SCSE

Set Operators

- Union
- Intersection
- Minus

union



A table containing all the rows from **A** and **B**.

UNION

```
SELECT ..... FROM ..... WHERE ..... ;
UNION
SELECT ..... FROM ..... WHERE .....;
```

eg. The two clubs want to hold a joint party. Make a list of all students. (Union)

Result

```
SELECT * FROM bridge
UNION
SELECT * FROM chess
ORDER BY class, name INTO TABLE party
```

- SELECT supplier_id FROM suppliers

UNION

SELECT supplier_id FROM orders ORDER BY supplier_id;

SELECT supplier_id, supplier_name FROM suppliers WHERE supplier_id > 2000

UNION

SELECT company_id, company_name FROM companies WHERE company_id > 1000 ORDER BY 1;

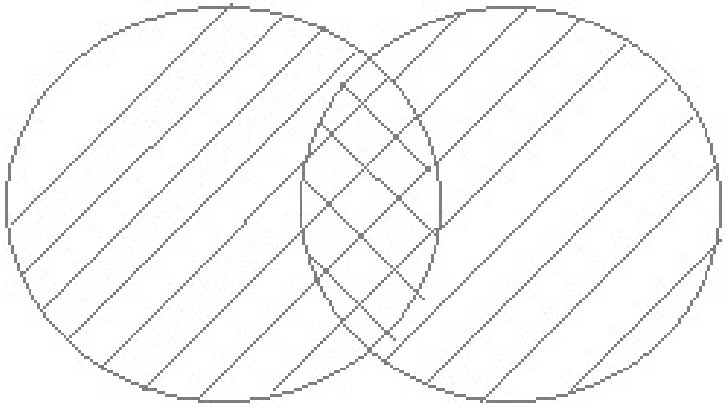
supplier_id	supplier_name
1000	Microsoft
2000	Oracle
3000	Apple
4000	Samsung

company_id	company_name
1000	Microsoft
3000	Apple
7000	Sony
8000	IBM

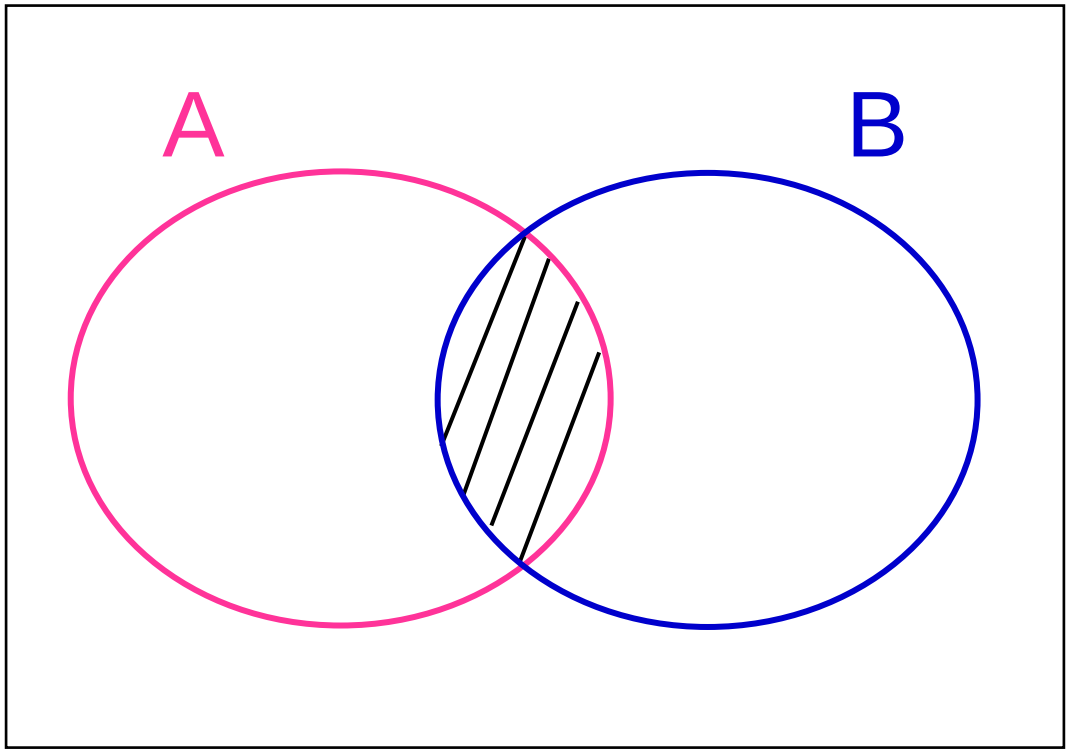
ID_Value	Name_Value
3000	Apple
4000	Samsung
7000	Sony
8000	IBM

UNION ALL

- This operation is similar to Union. But it also shows the duplicate rows.



intersection



A table containing only rows that appear in both **A** and **B**.

INTERSECTION

```
SELECT ..... FROM table1 ;
WHERE col IN  ( SELECT col FROM table2 )
```

eg. list the students who are members of both clubs. (Intersection)

Result

```
SELECT * FROM bridge
WHERE id IN ( SELECT id FROM chess ) ;
```

INTERSECT CLAUSE

- SELECT column1 [, column2] FROM tables [WHERE condition]

  INTERSECT

  SELECT column1 [, column2] FROM tables [WHERE condition]

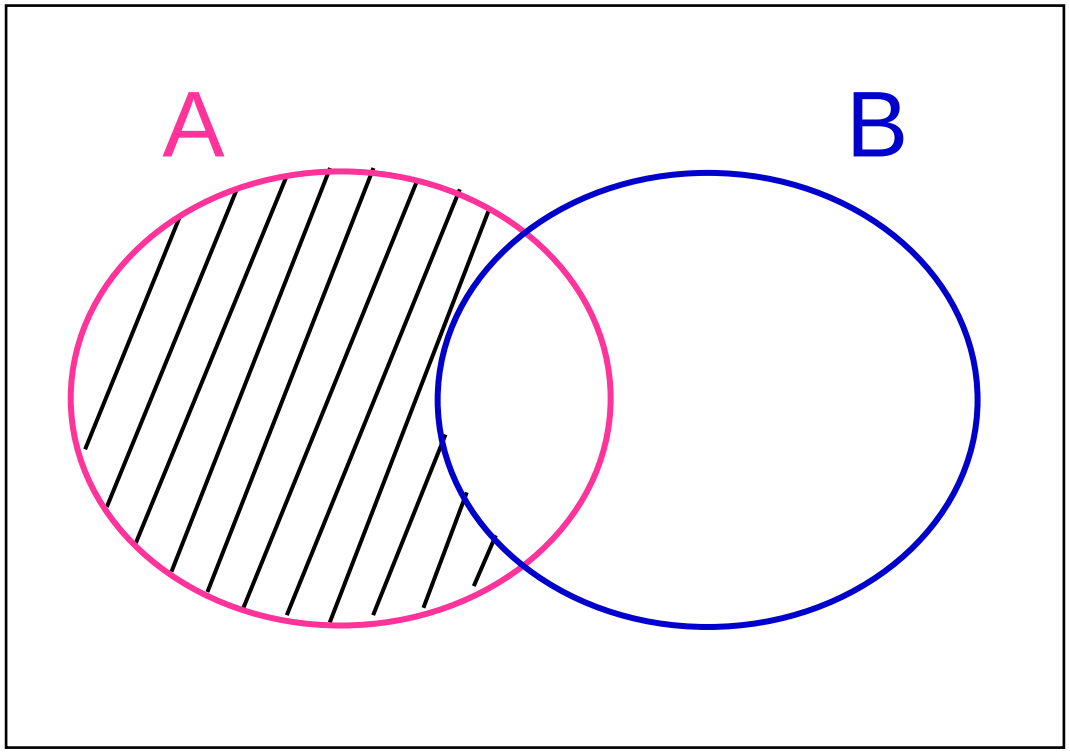
Data types of the column list must be compatible

INTERSECT - EXAMPLE

- SELECT supplier_id FROM suppliers
  INTERSECT
  SELECT supplier_id FROM orders;
- SELECT supplier_id FROM suppliers WHERE supplier_id > 78
  INTERSECT
  SELECT supplier_id FROM orders WHERE quantity <> 0;

MINUS

difference



A table containing rows that appear in **A** but not in **B**.

MINUS

SELECT ..... FROM *table1* ;
WHERE *col* **NOT IN** (SELECT *col* FROM *table2*)

eg. list the students who are members of the Bridge Club but not Chess Club. (Difference)

Result

```
SELECT * FROM bridge
WHERE id NOT IN ( SELECT id FROM chess );
```

MINUS CLAUSE

- SELECT expression1, expression2, ... expression_n FROM tables [WHERE conditions]

  MINUS

  SELECT expression1, expression2, ... expression_n FROM tables [WHERE conditions];

SELECT supplier_id FROM suppliers

MINUS

SELECT supplier_id FROM orders;

8. SQL Joins

- A SQL JOIN clause combines records from two or more tables in a database.
- It creates a set that can be saved as a table or used as is.
- A JOIN is a means for combining fields from two tables by using values common to each.

JOIN OPERATION

- A join can be specified in the FROM clause which list the **two input relations** and the WHERE clause which **lists the join condition**.

Example:

Emp

ID	State
1000	CA
1001	MA
1002	TN

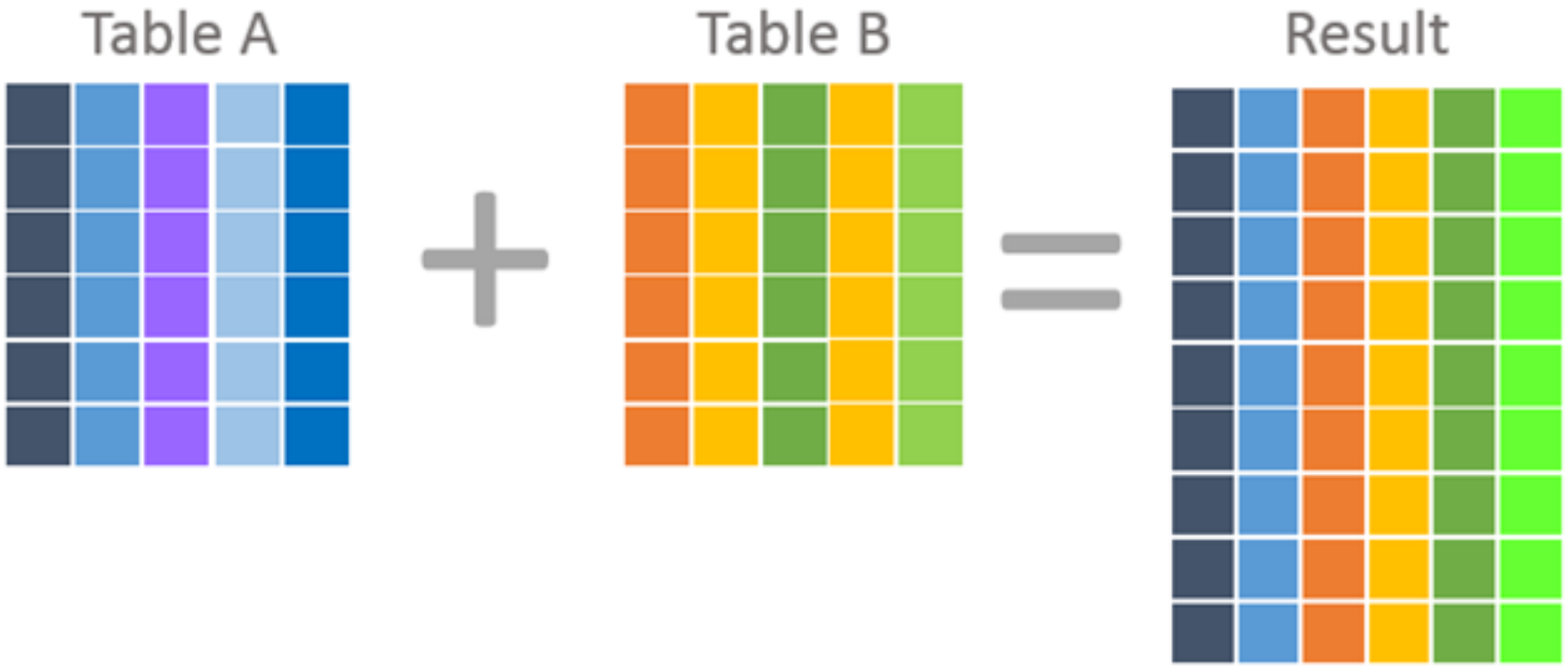
Dept

ID	Division
1001	IT
1002	Sales
1003	Biotech

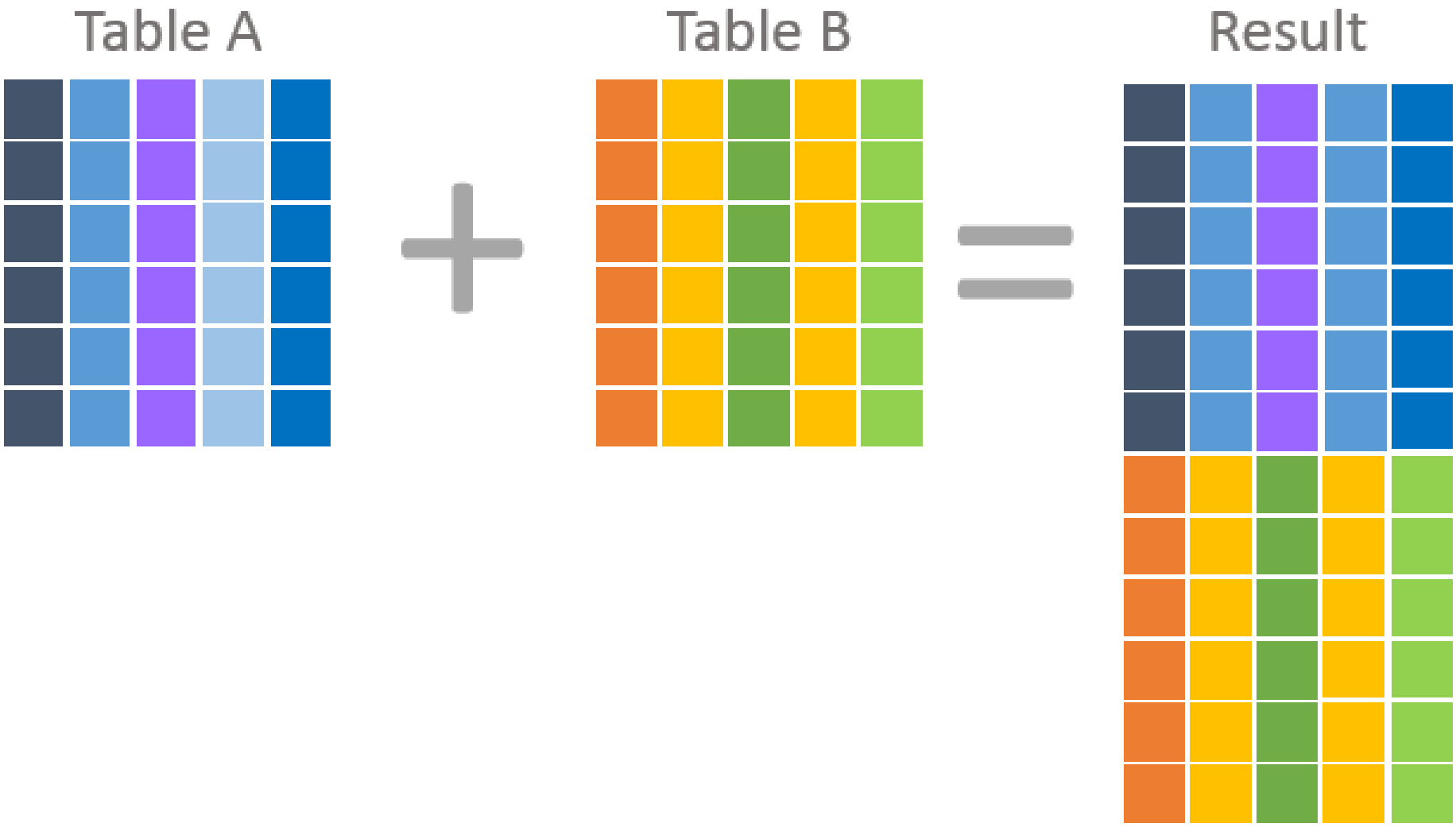
Join Vs Union

- Joins and Unions can be used to combine data from one or more tables. The difference lies in how the data is combined.
- In simple terms, **joins combine data into new columns**. If two tables are joined together, then the data from the first table is shown in one set of column alongside the second table's column in the same row.
- **Unions combine data into new rows.** If two tables are "union-ed" together, then the data from the first table is in one set of rows, and the data from the second table in another set. The rows are in the same result.

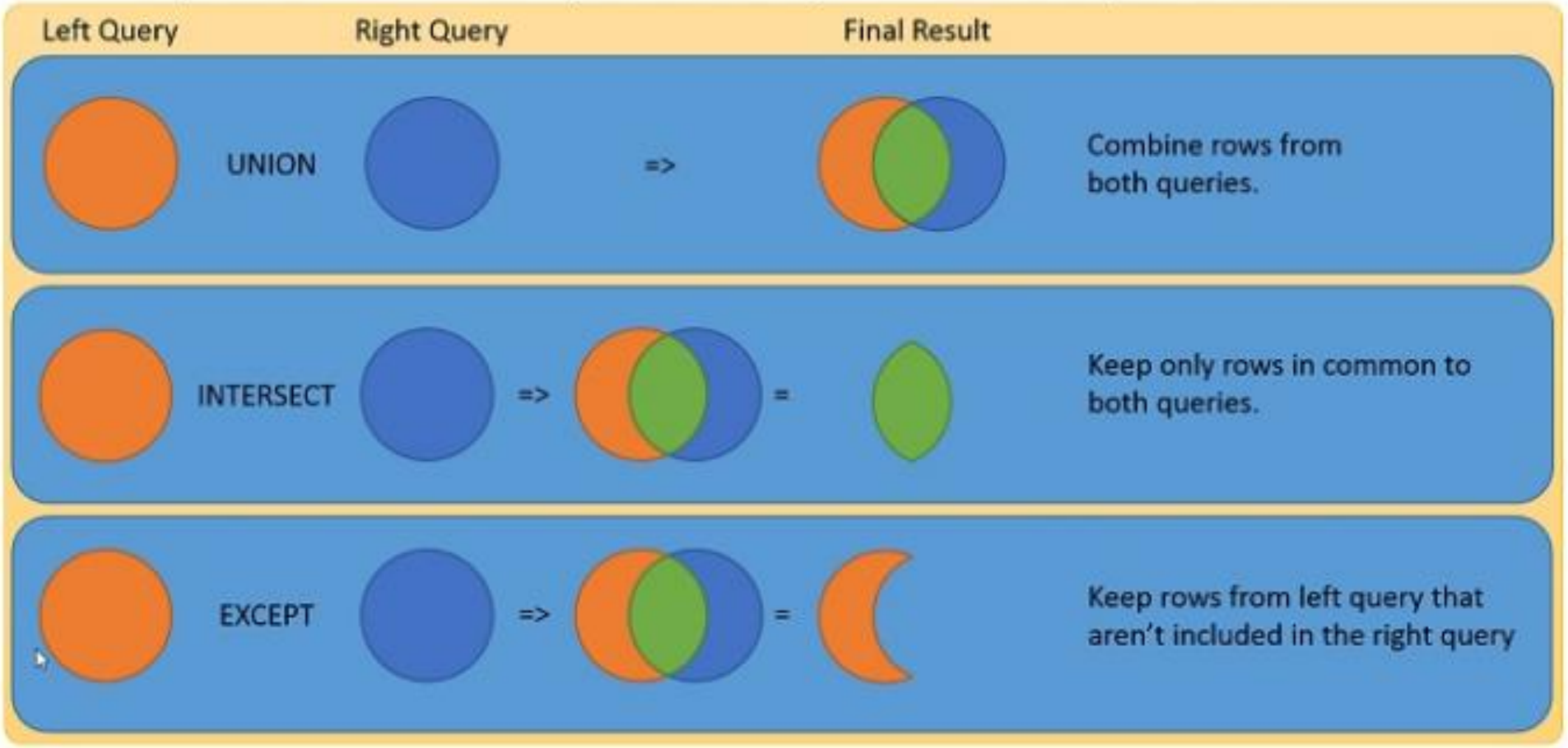
JOIN



UNION



Visual Explanation of UNION, INTERSECT, and EXCEPT operators



Types of Joins

- Equi-join
  - Inner join and Natural Join
  - Outer joins
    - Left outer join
    - Right outer joins
    - Full outer join
- Non Equi-join
- Cross join
- Self-join

EQUI JOIN Vs NON EQUI JOIN

- The SQL EQUI JOIN is a simple sql join uses the equal sign(=) as the comparison operator for the condition. It has two types - SQL Outer join and SQL Inner join

- The **SQL NON EQUI JOIN** is a join uses comparison operator other than the equal sign like >, <, >=, <= with the condition

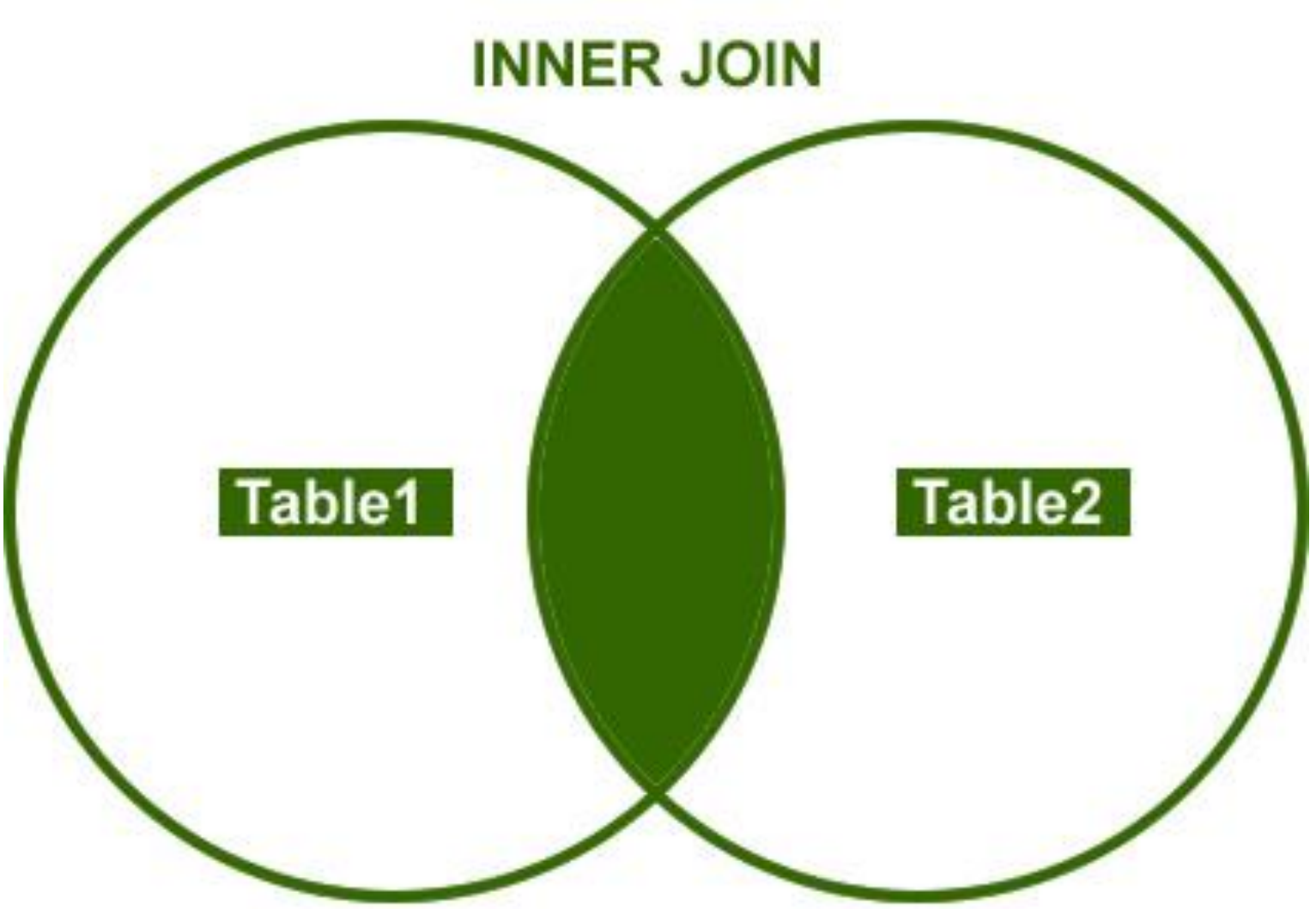
INNER JOIN Vs OUTER JOIN

SQL INNER JOIN

- This type of EQUI JOIN returns all rows from tables where the key record of one table is equal to the key records of another table.

SQL OUTER JOIN

- This type of EQUI JOIN returns all rows from one table and only those rows from the secondary table where the joined condition is satisfying i.e. the columns are equal in both tables.




```
SELECT *
FROM Table1 t1
INNER JOIN Table2 t2
        ON t1.Col1 = t2.Col1
```

JOIN DEFAULT: INNER JOIN

Inner Join

```
SELECT *
FROM table1, table2
WHERE table1.comcol = table2.comcol
```

```
SELECT a.comcol, a.col1, b.col2, expr1, expr2
FROM table1 a, table2 b
WHERE a.comcol = b.comcol
```

```
SELECT a.comcol, a.col1, b.col2, expr1, expr2
FROM table1 a INNER JOIN table2 b
ON a.comcol = b.comcol
```

INNER JOIN

- SELECT * FROM student S INNER JOIN Marks M ON S.Roll_No = M.Roll_No;

Student Table

Roll_No	Name
1	A
2	B
3	C

Marks Table

Roll_No	Marks
2	70
3	50
4	85

Roll_No	Name	Roll_No	Marks
2	B	2	70
3	C	3	50

NATURAL JOIN

- Natural Join joins two tables based on same attribute name and datatypes. The resulting table will contain all the attributes of both the table but keep only one copy of each common column.

- SELECT * FROM Student S NATURAL JOIN Marks M;

Natural Join Results

Student Table

Roll_No	Name
1	A
2	B
3	C

Marks Table

Roll_No	Marks
2	70
3	50
4	85

Roll_No	Name	Marks
2	B	70
3	C	50

select * from student s natural join marks m;

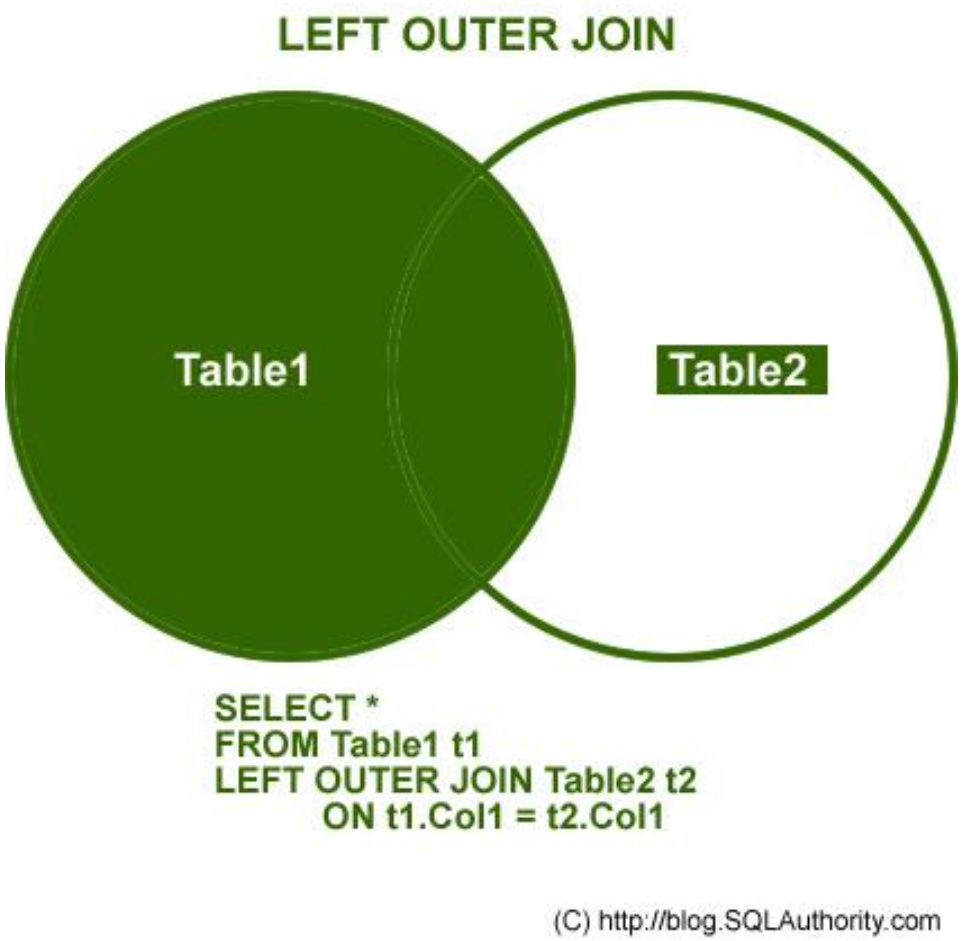
Difference between Natural JOIN and INNER JOIN in SQL

SR.NO.	NATURAL JOIN	INNER JOIN
1.	Natural Join joins two tables based on same attribute name and datatypes.	Inner Join joins two table on the basis of the column which is explicitly specified in the ON clause.
2.	In Natural Join, The resulting table will contain all the attributes of both the tables but keep only one copy of each common column	In Inner Join, The resulting table will contain all the attribute of both the tables including duplicate columns also
3.	In Natural Join, If there is no condition specifies then it returns the rows based on the common column	In Inner Join, only those records will return which exists in both the tables
4.	SYNTAX: SELECT * FROM table1 NATURAL JOIN table2;	SYNTAX: SELECT * FROM table1 INNER JOIN table2 ON table1.Column_Name = table2.Column_Name;

Outer Join

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

SQL Joins



- **left outer join** = `left join`

```
SELECT *
FROM emp left join dept
on emp.id = dept.id;
```

The **LEFT JOIN** keyword returns all records from the **left** table (Table1), and the matching records from the right table (Table2).

SELECT * FROM STUDENT S LEFT JOIN MARKS M ON S.ROLL_NO=M.ROLL_NO;

Student Table

Roll_No	Name
1	A
2	B
3	C

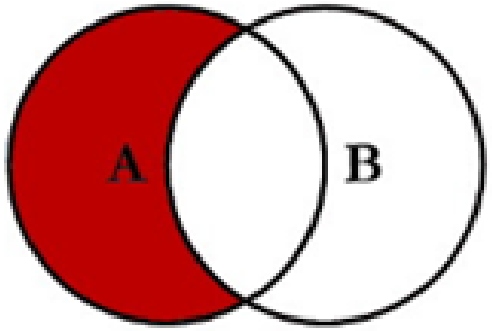
Marks Table

Roll_No	Marks
2	70
3	50
4	85

Roll_No	Name	Roll_No	Marks
2	B	2	70
3	C	3	50
1	A		

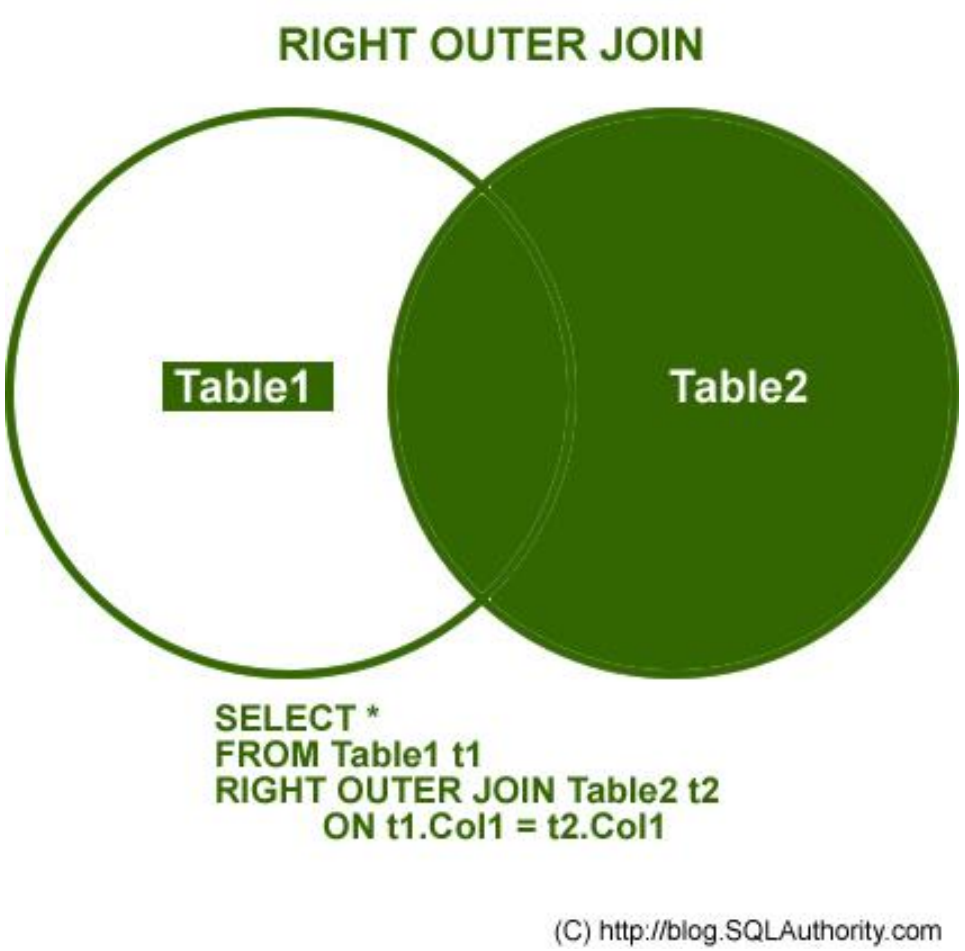
returns all rows of table on left side of join.
The rows for which there is **no matching row on right side**, result contains NULL in the right side.

SELECT * FROM STUDENT S LEFT JOIN MARKS M ON S.ROLL_NO=M.ROLL_NO
WHERE M.ROLL_NO **IS NULL**;



Roll_No	Name	Roll_No	Marks
1	A		

SQL JOINS



- **Right outer join** = `right join`

```
SELECT *
FROM emp right join dept
on emp.id = dept.id;
```

The **RIGHT JOIN** keyword returns all records from the **right** table (Table2), and the matching records from the left table (Table1).

SELECT * FROM STUDENT S RIGHT JOIN MARKS M ON S.ROLL_NO=M.ROLL_NO;

Student Table

Roll_No	Name
1	A
2	B
3	C

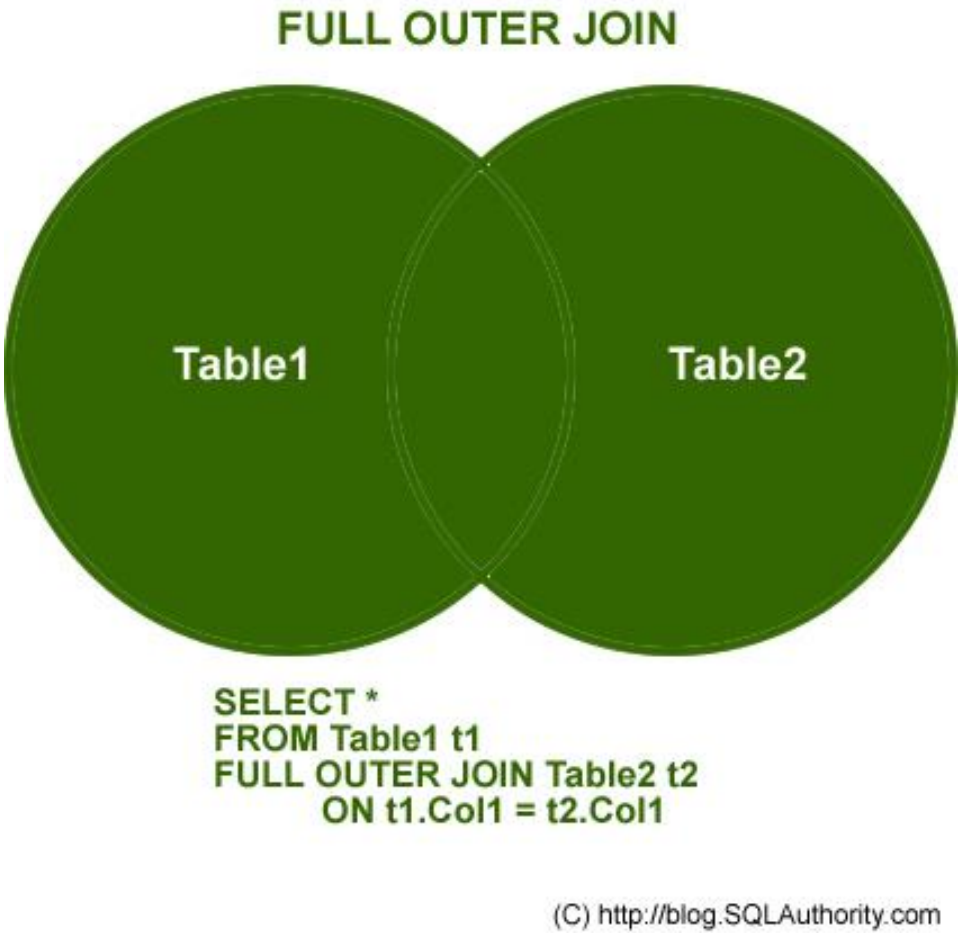
Marks Table

Roll_No	Marks
2	70
3	50
4	85

Roll_No	Name	Roll_No	Marks
2	B	2	70
3	C	3	50
		4	85

returns all rows of table on right side of join.
The rows for which there is **no matching row on left side**, result contains NULL in the left side.

SQL Joins




```
SELECT *
FROM emp Full join dept
on emp.id = dept.id;
```

SELECT * FROM STUDENT S FULL JOIN MARKS M ON S.ROLL_NO=M.ROLL_NO;

Student Table

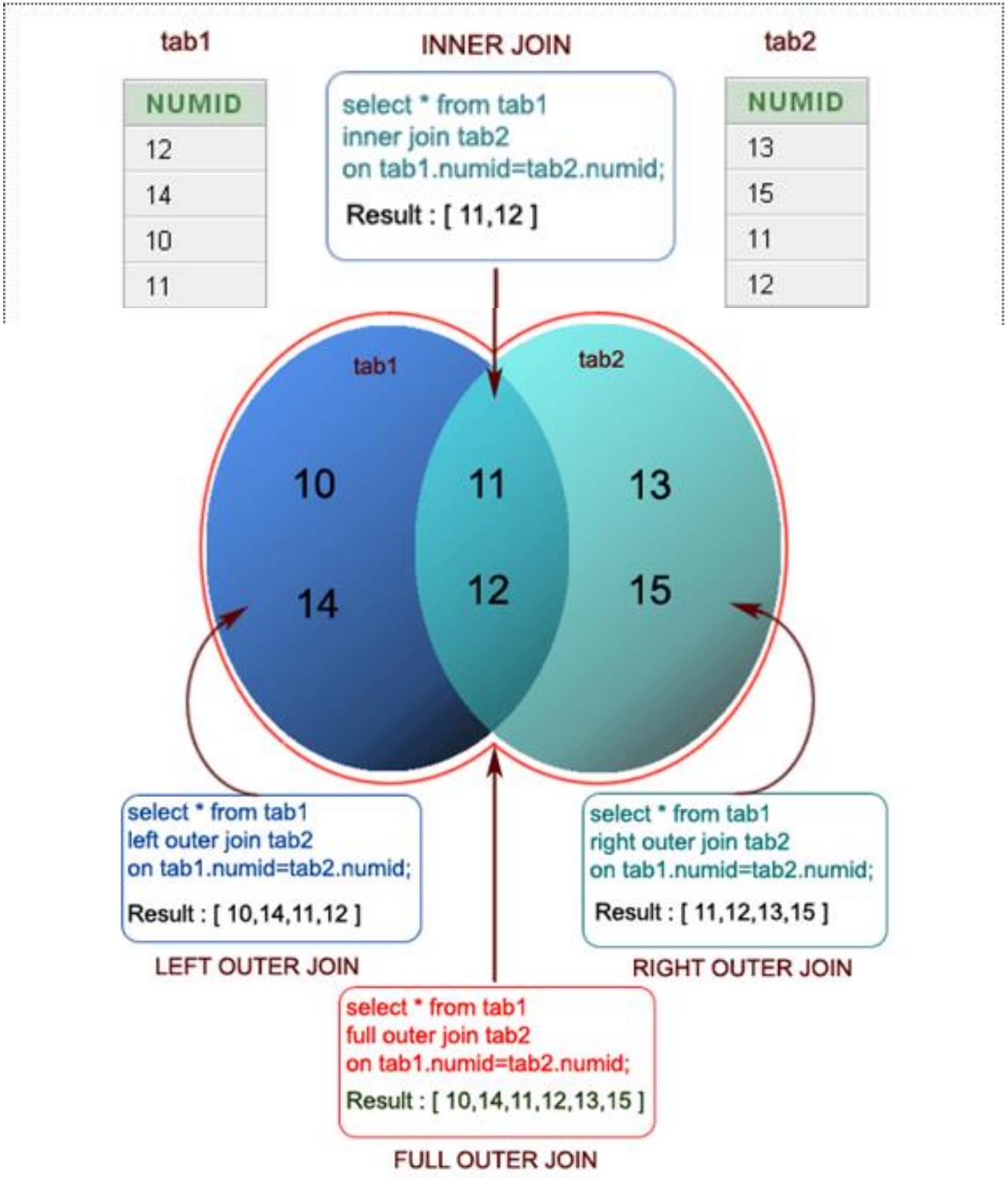
Roll_No	Name
1	A
2	B
3	C

Marks Table

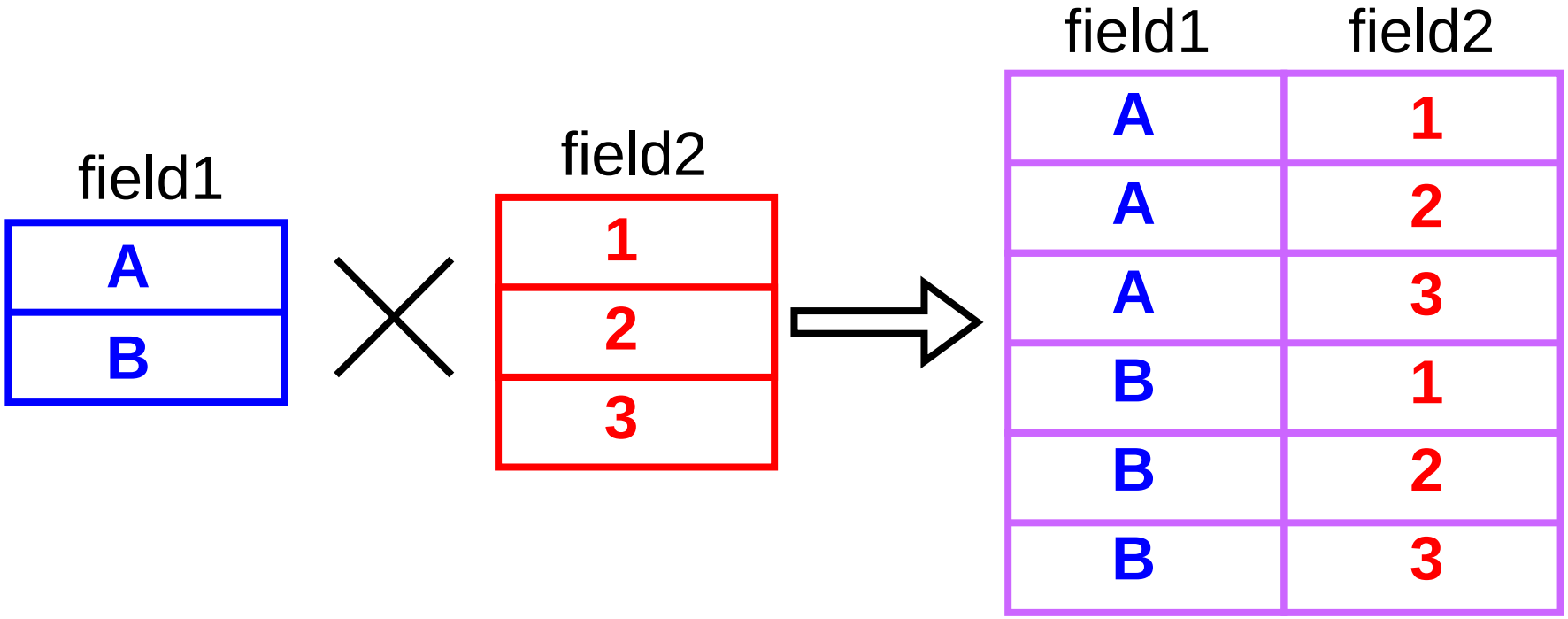
Roll_No	Marks
2	70
3	50
4	85

Roll_No	Name	Roll_No	Marks
2	B	2	70
3	C	3	50
		4	85
1	A		

Contains results of both Left and Right outer joins


tab1
NUMID
12
14
10
11
INNER JOIN
select * from tab1
inner join tab2
on tab1.numid=tab2.numid;
Result : [11,12]
tab2
NUMID
13
15
11
12
tab1
tab2
10
14
11
12
13
15
select * from tab1
left outer join tab2
on tab1.numid=tab2.numid;
Result : [10,14,11,12]
LEFT OUTER JOIN
select * from tab1
right outer join tab2
on tab1.numid=tab2.numid;
Result : [11,12,13,15]
RIGHT OUTER JOIN
select * from tab1
full outer join tab2
on tab1.numid=tab2.numid;
Result : [10,14,11,12,13,15]
FULL OUTER JOIN

Cross Join



SELECT * FROM STUDENT S CROSS JOIN MARKS M

Student Table

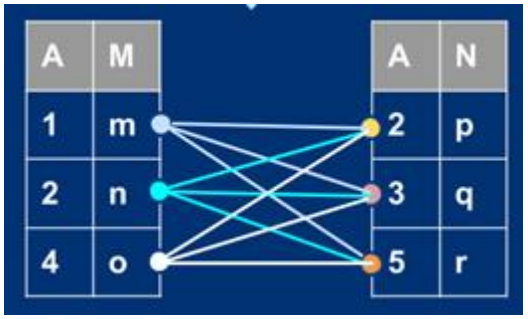
Roll_No	Name
1	A
2	B
3	C

Marks Table

Roll_No	Marks
2	70
3	50
4	85

Roll_No	Name	Roll_No	Marks
1	A	2	70
1	A	3	50
1	A	4	85
2	B	2	70
2	B	3	50
2	B	4	85
3	C	2	70
3	C	3	50
3	C	4	85

CROSS JOIN



Student

ID	Name
123	John
124	Mary
125	Mark
126	Jane

Enrolment

ID	Code
123	DBS
124	PRG
124	DBS
126	PRG

```
SELECT * FROM
   Student CROSS JOIN
   Enrolment
```

ID	Name	ID	Code
123	John	123	DBS
124	Mary	123	DBS
125	Mark	123	DBS
126	Jane	123	DBS
123	John	124	PRG
124	Mary	124	PRG
125	Mark	124	PRG
126	Jane	124	PRG
123	John	124	DBS
124	Mary	124	DBS

Self Join

- **Joining the table itself** called self join.
- Self join is used to retrieve the records having some relation or **similarity with other records in the same table**.
- Here we need to use aliases for the same table to set a self join between single table and retrieve records satisfying the condition in where clause.

Self Join - Example

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
2	Khilan	25	Delhi	1500.00
3	kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	MP	4500.00
7	Muffy	24	Indore	10000.00

- Employees getting salary less than that of an employee
- SELECT a.ID, b.NAME, a.SALARY FROM EMP a, EMP b

  WHERE **a.SALARY < b.SALARY**;

ID	NAME	SALARY
2	Ramesh	1500.00
2	kaushik	1500.00
1	Chaitali	2000.00
2	Chaitali	1500.00
3	Chaitali	2000.00
6	Chaitali	4500.00
1	Hardik	2000.00
2	Hardik	1500.00
3	Hardik	2000.00
4	Hardik	6500.00
6	Hardik	4500.00
1	Komal	2000.00
2	Komal	1500.00
3	Komal	2000.00
1	Muffy	2000.00
2	Muffy	1500.00
3	Muffy	2000.00
4	Muffy	6500.00
5	Muffy	8500.00
6	Muffy	4500.00

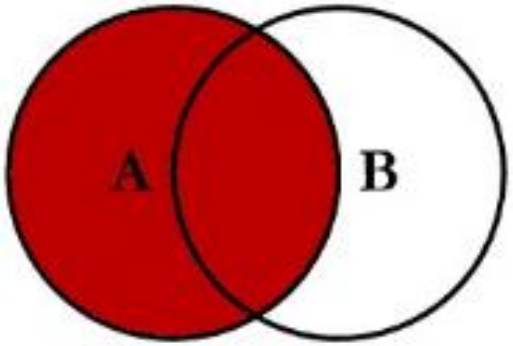
```
SELECT
A.ROLL_NO,B.ROLL_NO,A.MARKS,B.MARKS,
A.MARKS-B.MARKS AS DIFF FROM MARKS A,
MARKS B WHERE A.MARKS>B.MARKS;
```

Marks Table

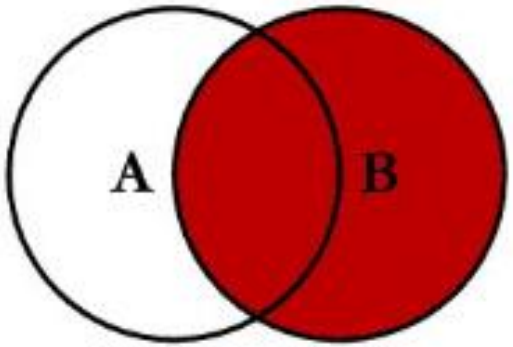
Roll_No	Marks
2	70
3	50
4	85

ROLL_NO	ROLL_NO	MARKS	MARKS	DIFF
2	3	70	50	20
4	2	85	70	15
4	3	85	50	35

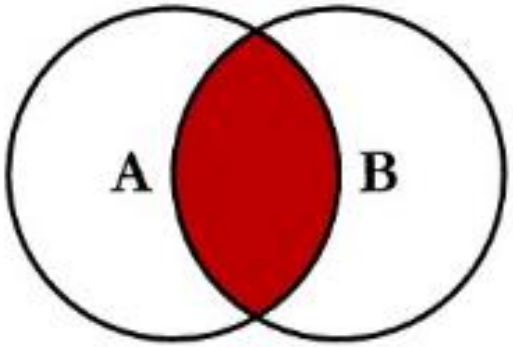
SQL JOINS



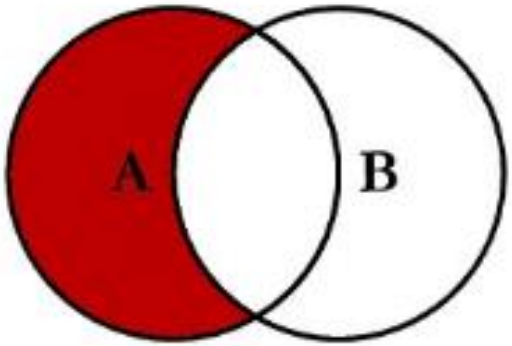

```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



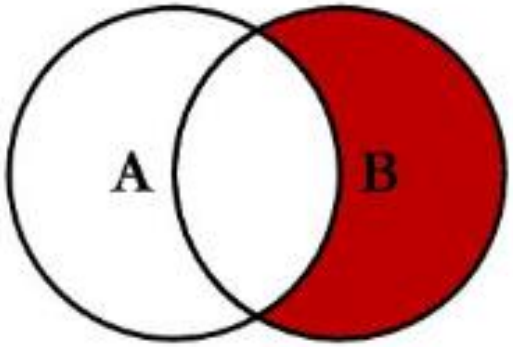

```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



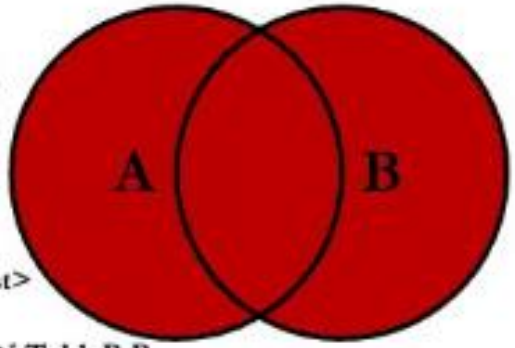

```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



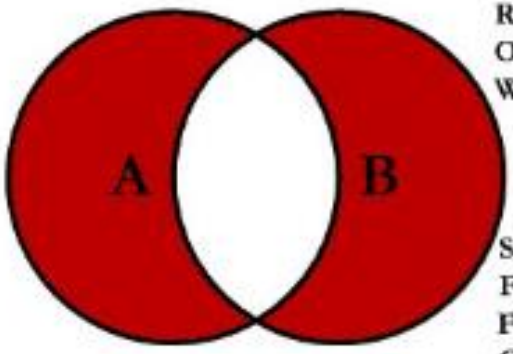

```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```




```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```




```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```




```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```

THANK YOU