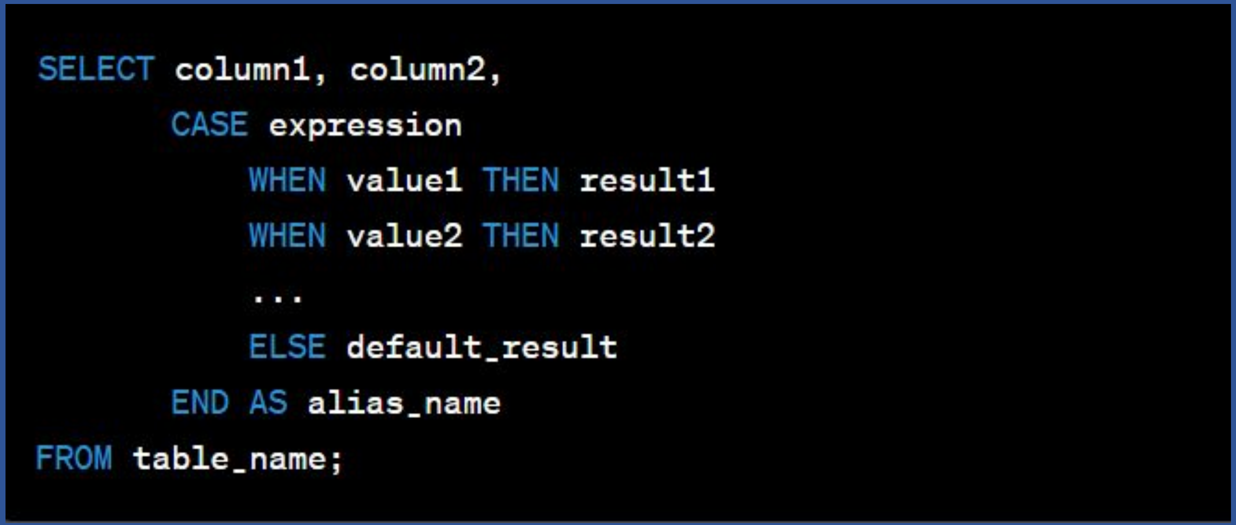
AGENDA

- CASE STATEMENT
- JOINS IN SQL
- SUBQUERIES
- SQL QUERY ORDER EXECUTION

CASE STATEMENT

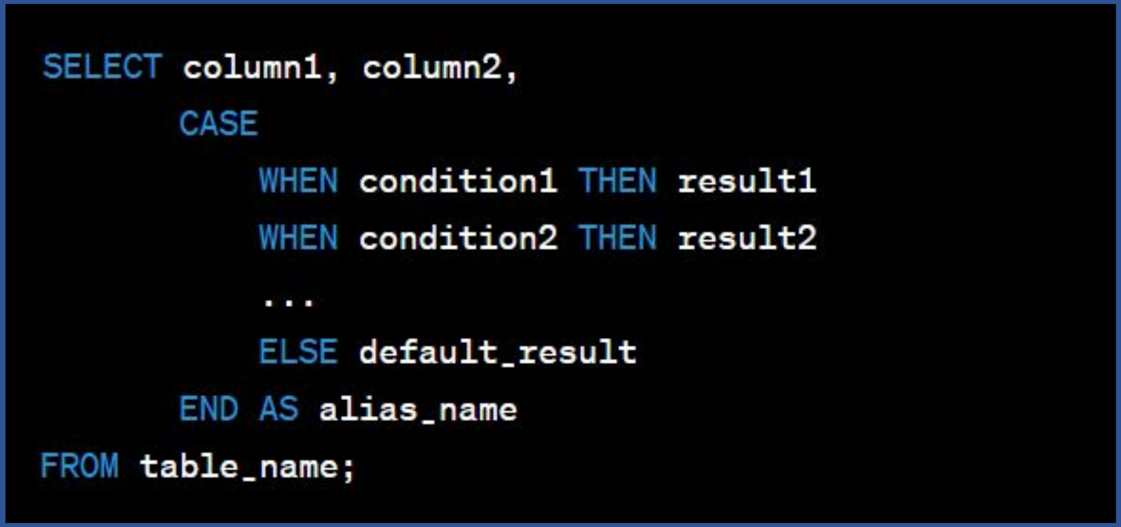
The CASE statement is used for conditional logic within a query. It allows you to perform different actions based on different conditions. The CASE statement has two main forms: the simple CASE expression and the searched CASE expressio

SIMPLE CASE EXPRESSION



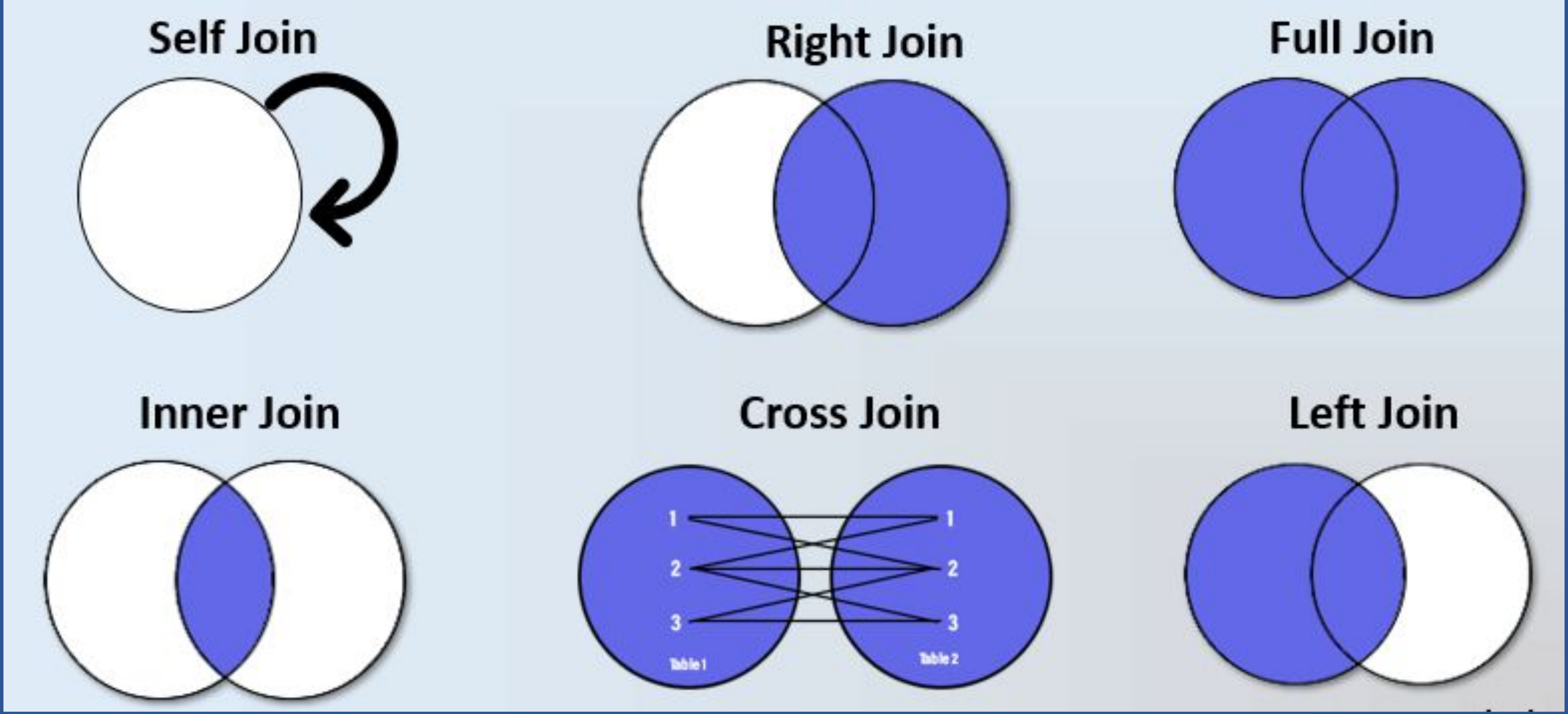
```
SELECT column1, column2,
       CASE expression
           WHEN value1 THEN result1
           WHEN value2 THEN result2
           ...
           ELSE default_result
       END AS alias_name
FROM table_name;
```

SEARCHED CASE EXPRESSION

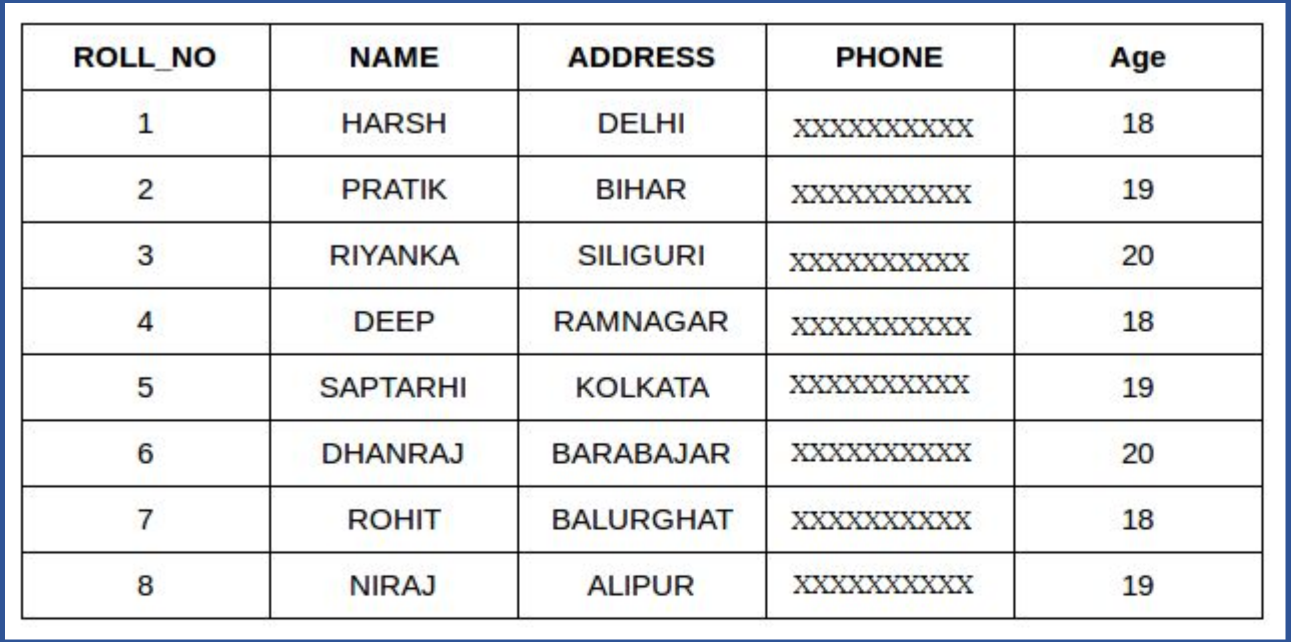


```
SELECT column1, column2,
       CASE
           WHEN condition1 THEN result1
           WHEN condition2 THEN result2
           ...
           ELSE default_result
       END AS alias_name
FROM table_name;
```

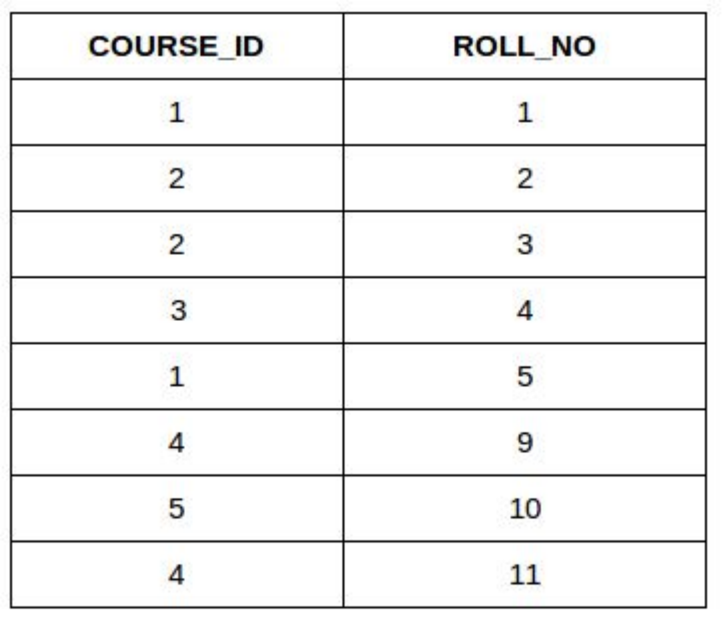
JOINS IN MYSQL



JOINS IN MYSQL

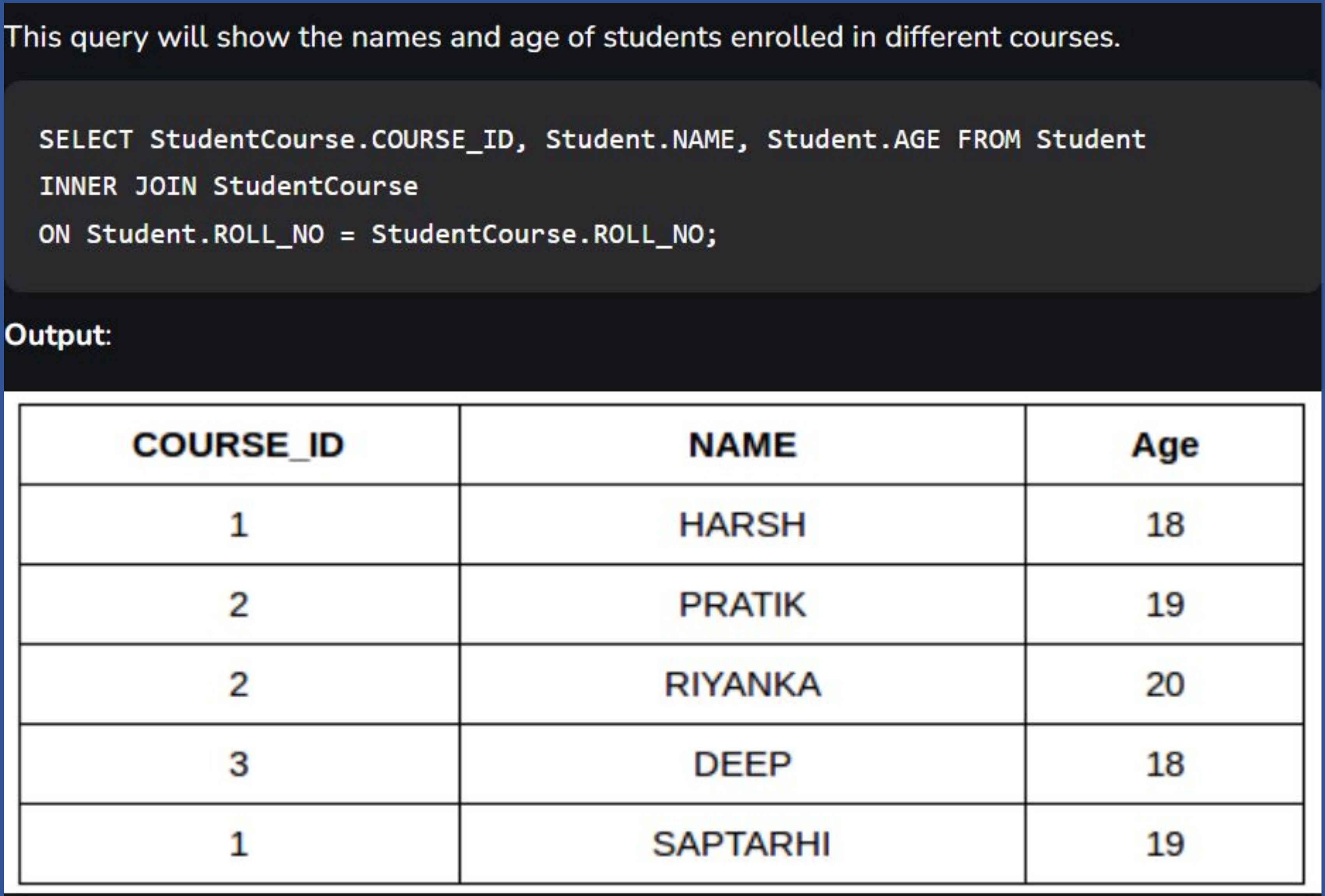


ROLL_NO	NAME	ADDRESS	PHONE	Age
1	HARSH	DELHI	XXXXXXXXXX	18
2	PRATIK	BIHAR	XXXXXXXXXX	19
3	RIYANKA	SILIGURI	XXXXXXXXXX	20
4	DEEP	RAMNAGAR	XXXXXXXXXX	18
5	SAPTARHI	KOLKATA	XXXXXXXXXX	19
6	DHANRAJ	BARABAJAR	XXXXXXXXXX	20
7	ROHIT	BALURGHAT	XXXXXXXXXX	18
8	NIRAJ	ALIPUR	XXXXXXXXXX	19



COURSE_ID	ROLL_NO
1	1
2	2
2	3
3	4
1	5
4	9
5	10
4	11

INNER JOIN



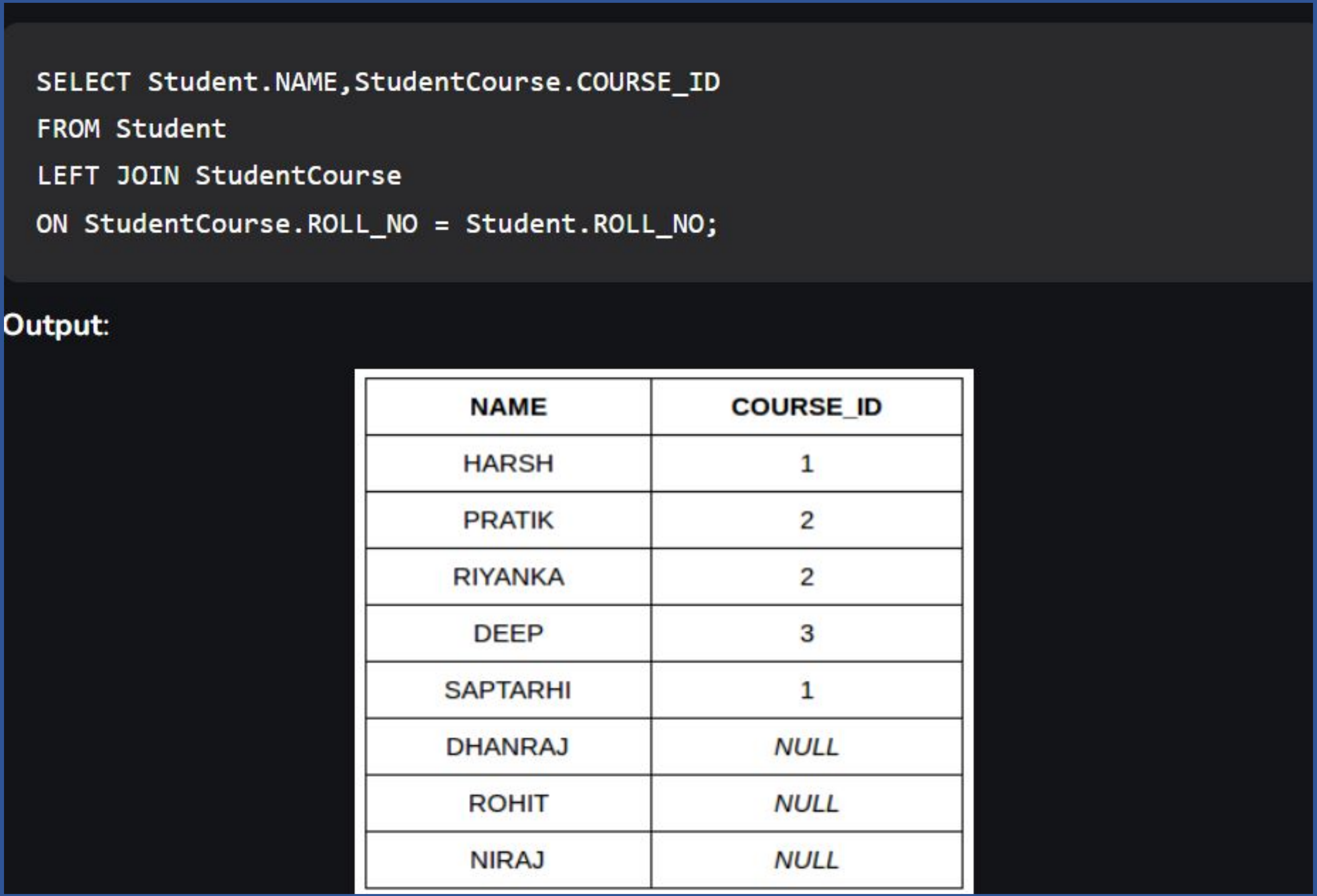
This query will show the names and age of students enrolled in different courses.

```
SELECT StudentCourse.COURSE_ID, Student.NAME, Student.AGE FROM Student
INNER JOIN StudentCourse
ON Student.ROLL_NO = StudentCourse.ROLL_NO;
```

Output:

COURSE_ID	NAME	Age
1	HARSH	18
2	PRATIK	19
2	RIYANKA	20
3	DEEP	18
1	SAPTARHI	19

LEFT JOIN

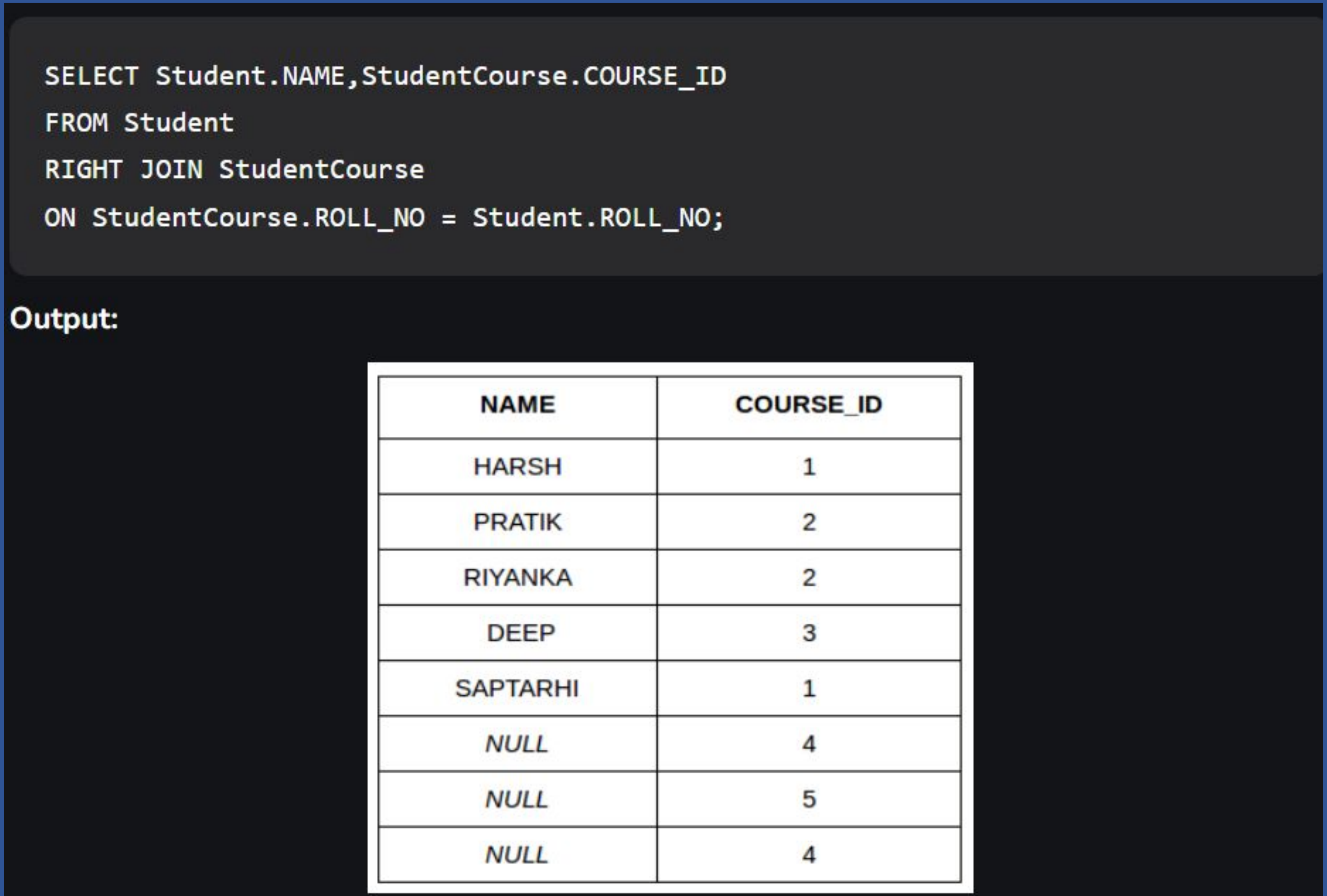


```
SELECT Student.NAME,StudentCourse.COURSE_ID
FROM Student
LEFT JOIN StudentCourse
ON StudentCourse.ROLL_NO = Student.ROLL_NO;
```

Output:

NAME	COURSE_ID
HARSH	1
PRATIK	2
RIYANKA	2
DEEP	3
SAPTARHI	1
DHANRAJ	*NULL*
ROHIT	*NULL*
NIRAJ	*NULL*

RIGHT JOIN

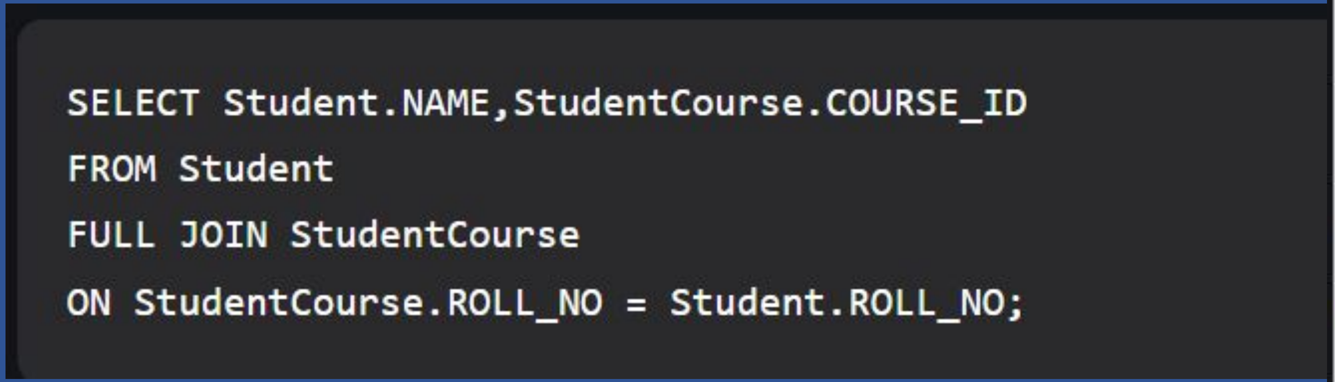


```
SELECT Student.NAME,StudentCourse.COURSE_ID
FROM Student
RIGHT JOIN StudentCourse
ON StudentCourse.ROLL_NO = Student.ROLL_NO;
```

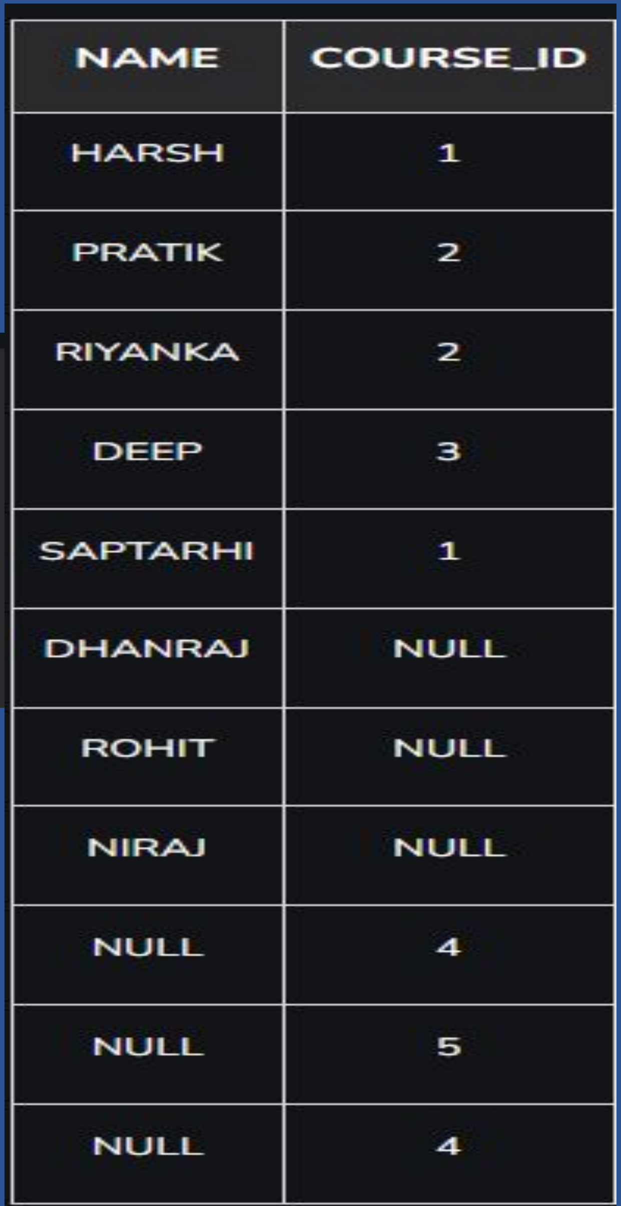
Output:

NAME	COURSE_ID
HARSH	1
PRATIK	2
RIYANKA	2
DEEP	3
SAPTARHI	1
NULL	4
NULL	5
NULL	4

FULL JOIN

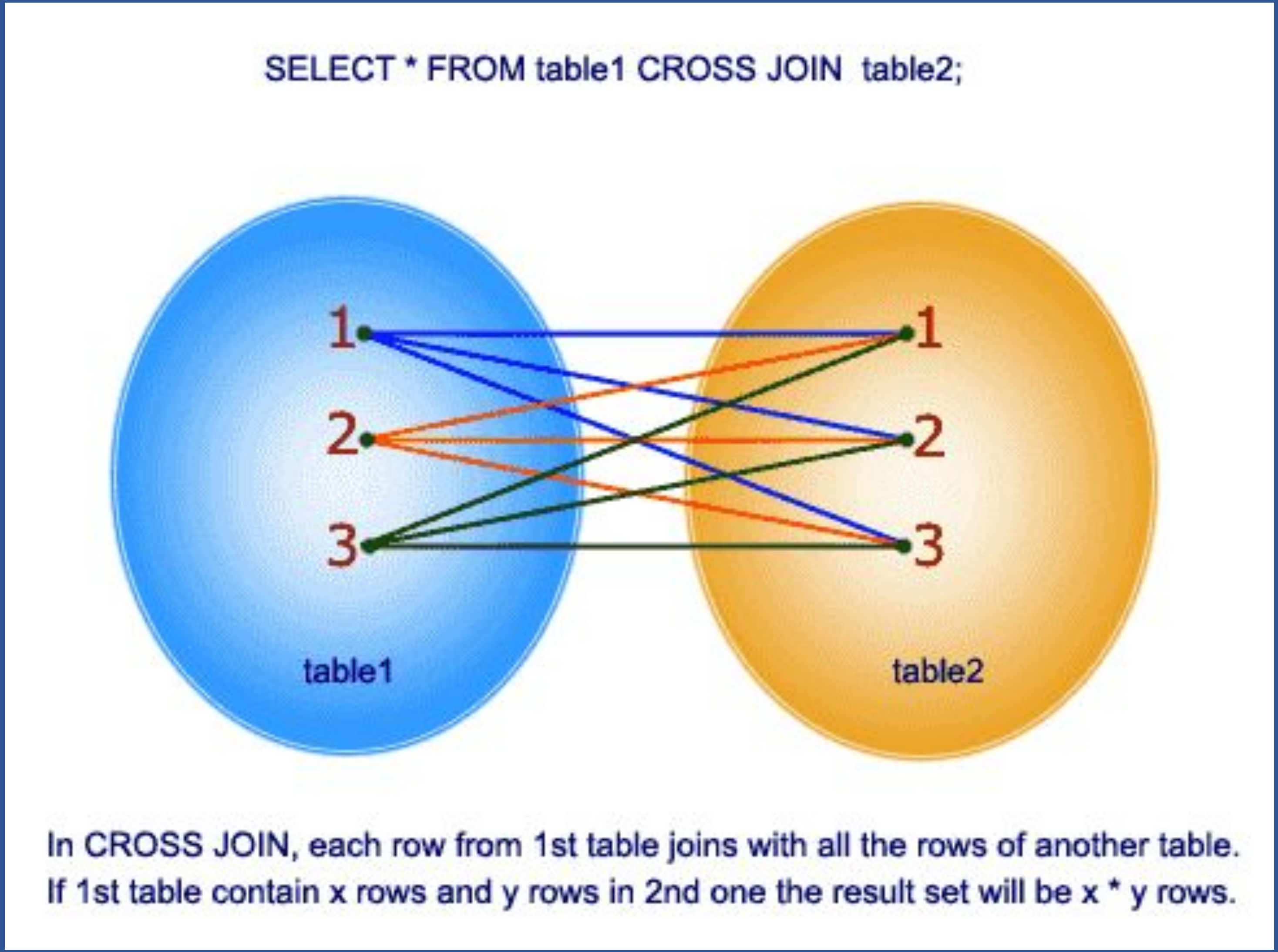


```
SELECT Student.NAME,StudentCourse.COURSE_ID
FROM Student
FULL JOIN StudentCourse
ON StudentCourse.ROLL_NO = Student.ROLL_NO;
```



NAME	COURSE_ID
HARSH	1
PRATIK	2
RIYANKA	2
DEEP	3
SAPTARHI	1
DHANRAJ	NULL
ROHIT	NULL
NIRAJ	NULL
NULL	4
NULL	5
NULL	4

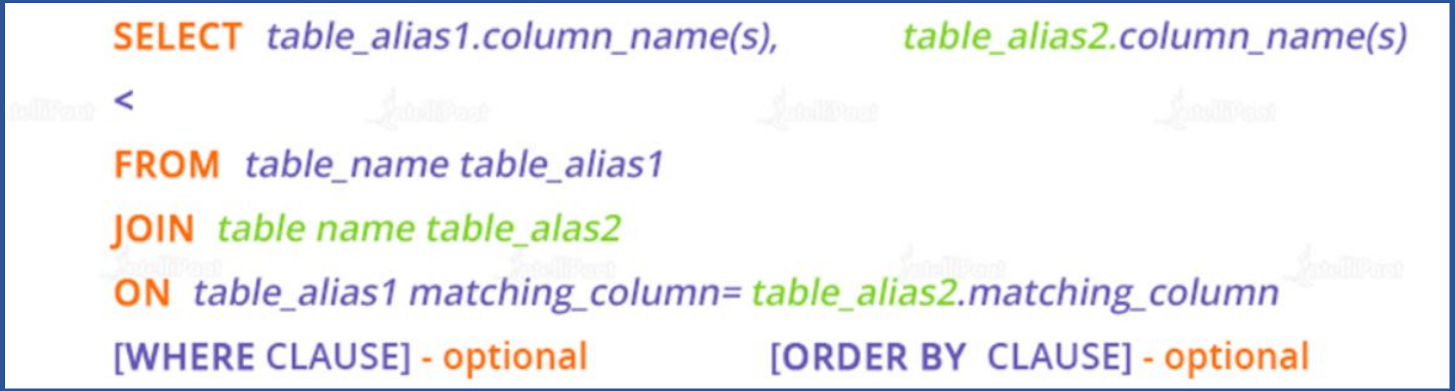
CROSS JOIN



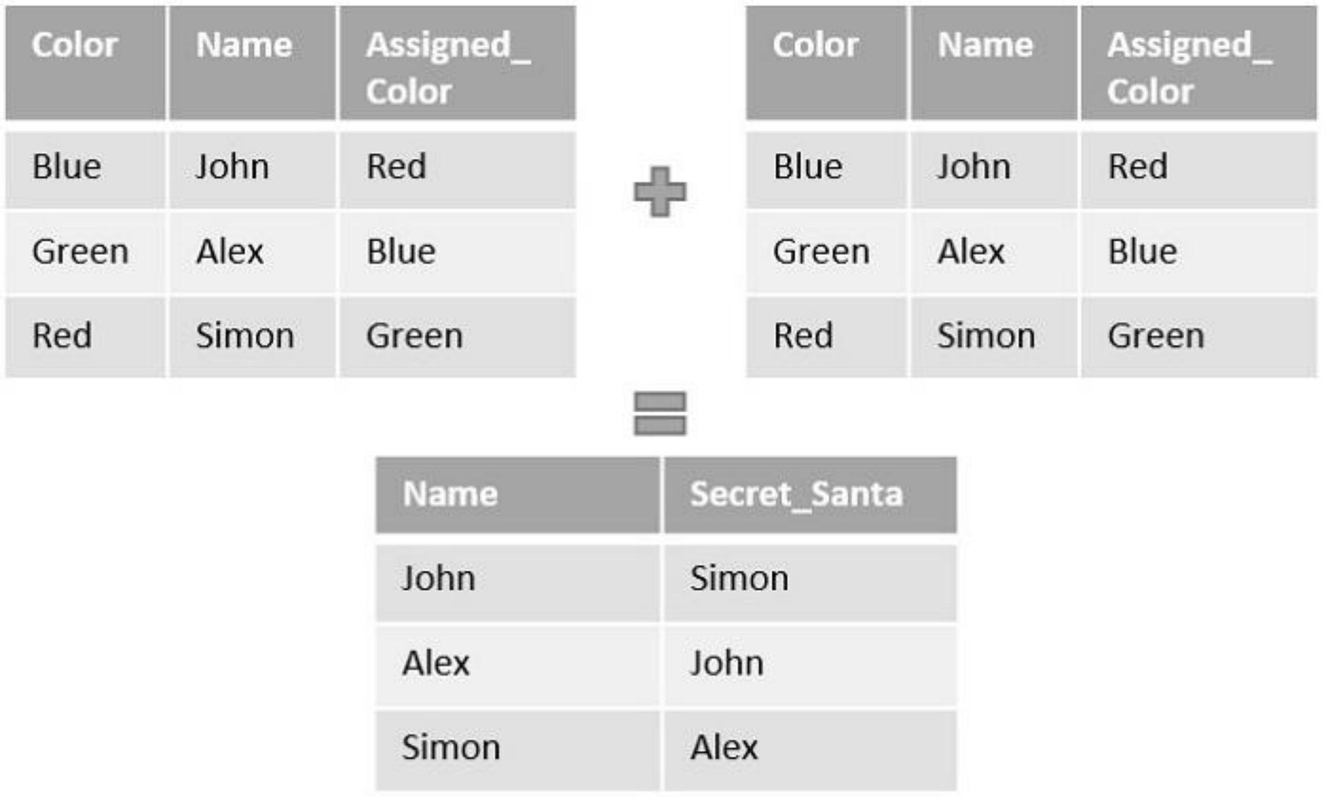
```
SELECT * FROM table1 CROSS JOIN  table2;
```

In CROSS JOIN, each row from 1st table joins with all the rows of another table. If 1st table contain x rows and y rows in 2nd one the result set will be x * y rows.

SELF JOIN



```
SELECT  table_alias1.column_name(s),        table_alias2.column_name(s)
<
FROM  table_name table_alias1
JOIN  table name table_alas2
ON  table_alias1 matching_column= table_alias2.matching_column
[WHERE CLAUSE] - optional            [ORDER BY  CLAUSE] - optional
```



Color	Name	Assigned_ Color
Blue	John	Red
Green	Alex	Blue
Red	Simon	Green

+

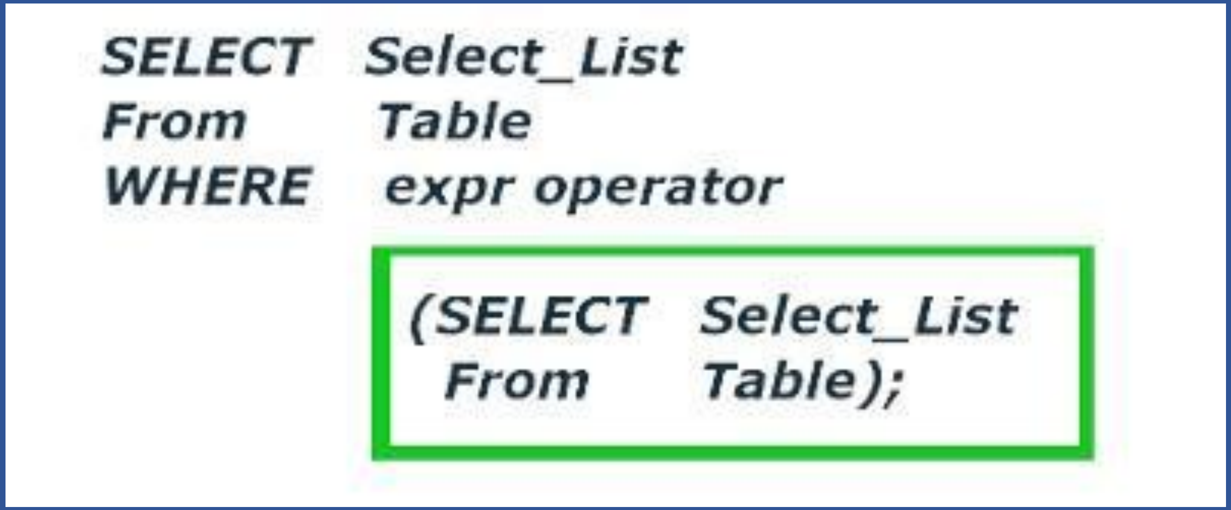
Color	Name	Assigned_ Color
Blue	John	Red
Green	Alex	Blue
Red	Simon	Green

=

Name	Secret_Santa
John	Simon
Alex	John
Simon	Alex

SUBQUERY IN MYSQL

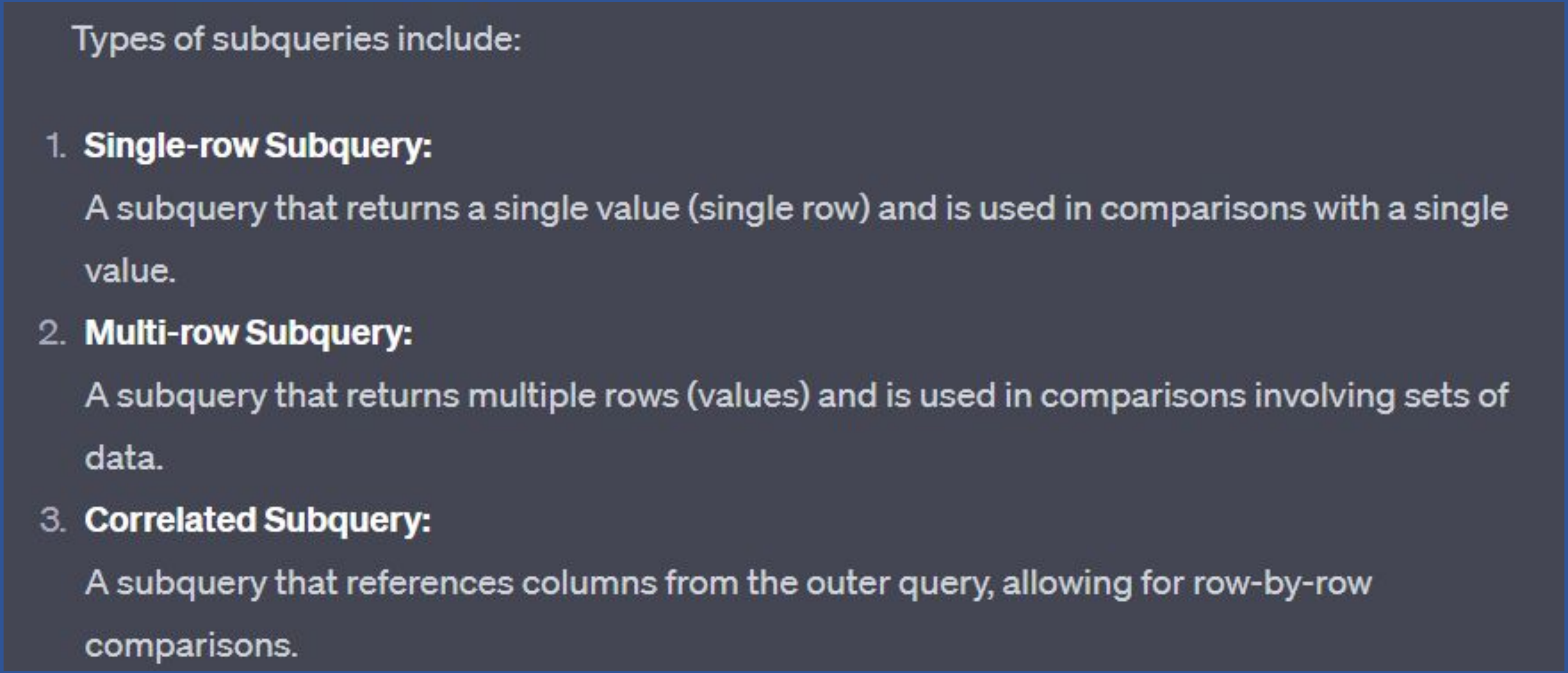
A subquery, also known as a nested query or inner query, is a query that is embedded within another query. Subqueries are used to retrieve data that will be used in the main query's criteria, calculations, or conditions. They are a powerful tool in SQL for performing complex operations and fetching specific data based on conditions that involve data from multiple tables.



```
SELECT   Select_List
From      Table
WHERE     expr operator

           (SELECT   Select_List
             From      Table);
```

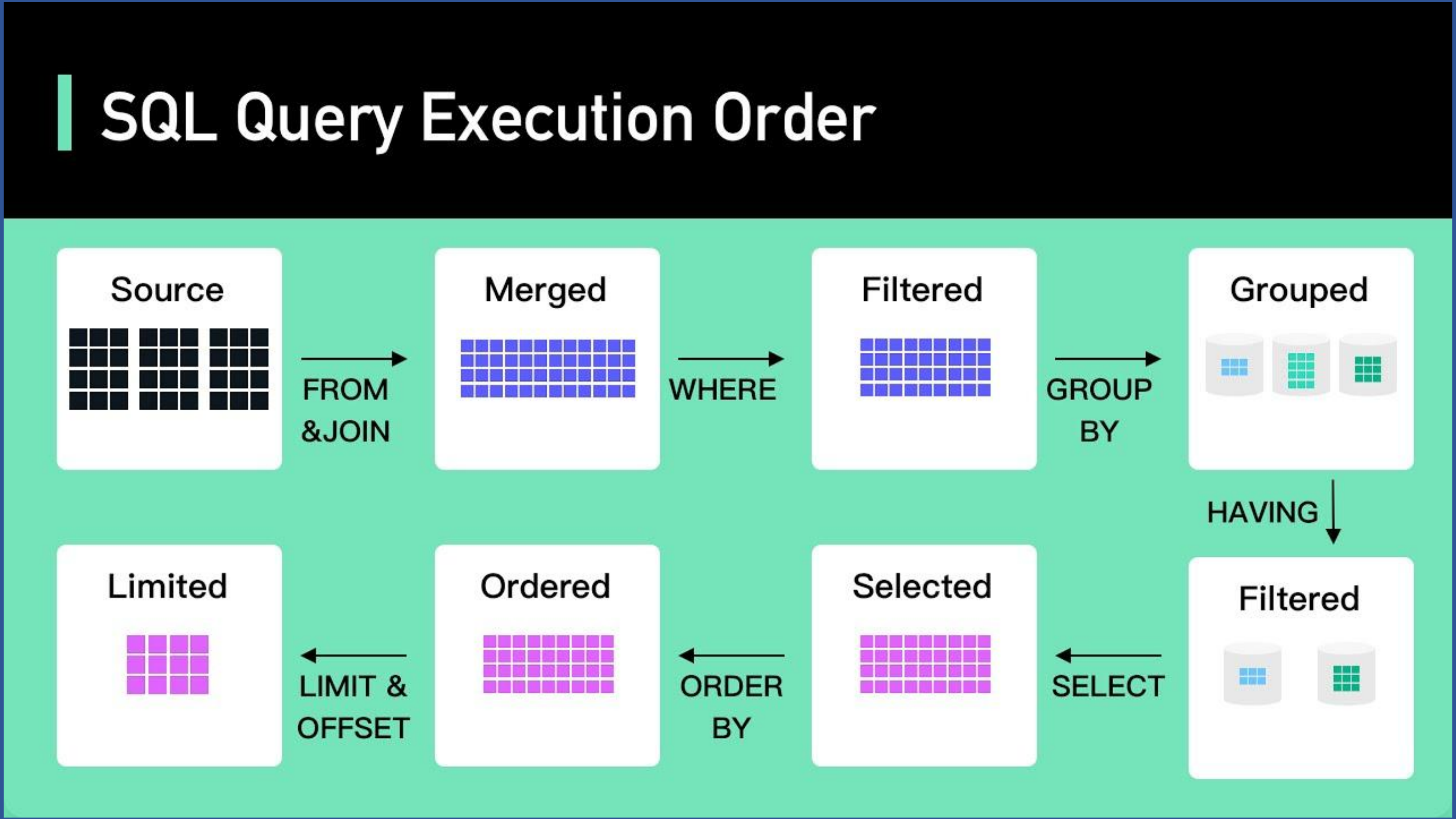
SUBQUERY IN MYSQL



Types of subqueries include:

1. **Single-row Subquery:**
   A subquery that returns a single value (single row) and is used in comparisons with a single value.
2. **Multi-row Subquery:**
   A subquery that returns multiple rows (values) and is used in comparisons involving sets of data.
3. **Correlated Subquery:**
   A subquery that references columns from the outer query, allowing for row-by-row comparisons.

SQL QUERY ORDER EXECUTION



SQL Query Execution Order