BHARATIYA VIDYA BHAVAN'S

SARDAR PATEL INSTITUTE OF TECHNOLOGY

Munshi nagar, Andheri (W) ,Mumbai - 400058

DEPARTMENT OF MASTER OF COMPUTER APPLICATION

CLASS: F.Y. MCA **SEM: I**

COURSE CODE: MC504 **SUBJECT NAME: DATABASE MANAGEMENT SYSTEM LAB**

ROLL NO. : 2022510034 **BATCH: B**

NAME: Rishikesh Prakash Lingayat

EXPERIMENT NO: 05

EXPERIMENT TITLE: To study SQL Joins

SQL Joins:

- Inner JOIN
- Full JOIN
- Outer JOIN
  - Left Outer JOIN
  - Right Outer JOIN
- Cross JOIN

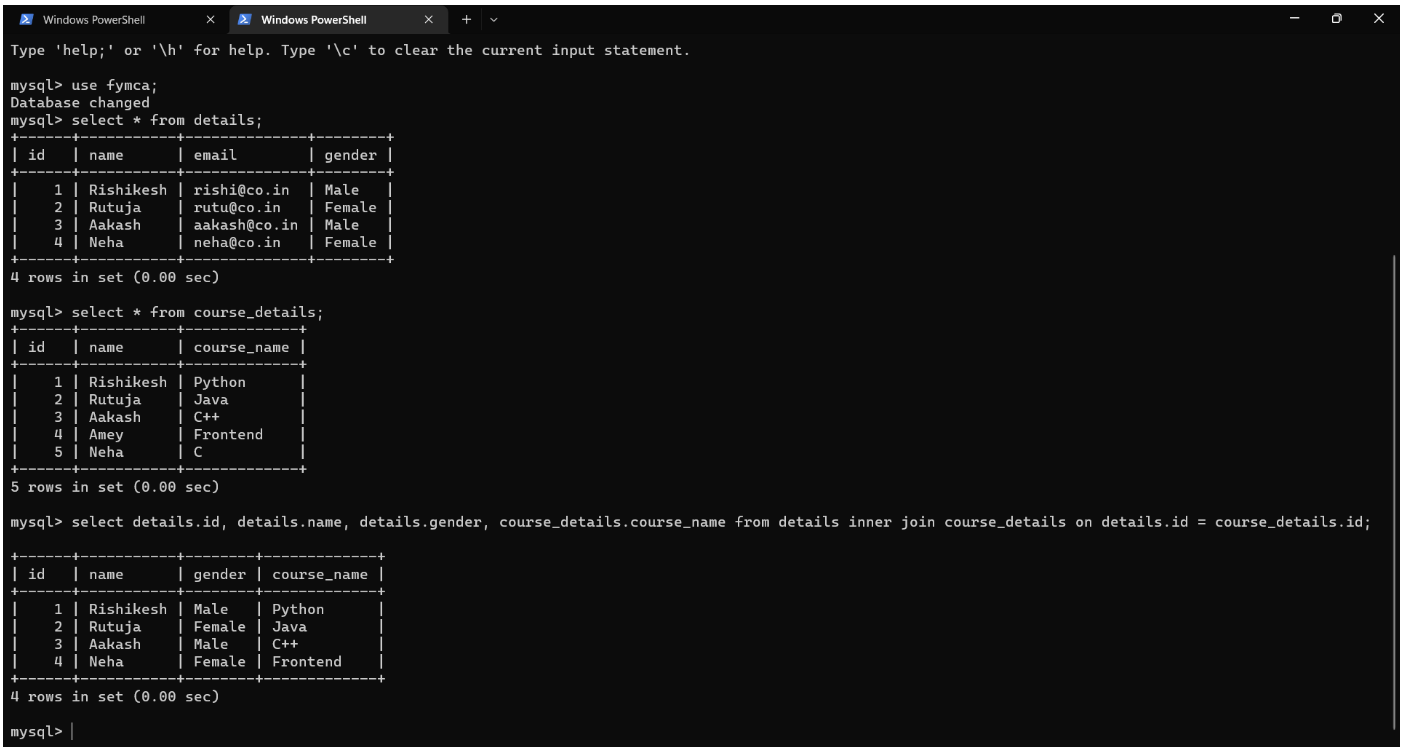
Inner Join:

The MySQL INNER JOIN is used to return all rows from multiple tables where the join condition is satisfied. It is the most common type of join.

Syntax:

SELECT columns
FROM table1
INNER JOIN table2
ON table1.column = table2.column;

Output:-



```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use fymca;
Database changed
mysql> select * from details;
+------+-----------+--------------+--------+
| id   | name      | email        | gender |
+------+-----------+--------------+--------+
|    1 | Rishikesh | rishi@co.in  | Male   |
|    2 | Rutuja    | rutu@co.in   | Female |
|    3 | Aakash    | aakash@co.in | Male   |
|    4 | Neha      | neha@co.in   | Female |
+------+-----------+--------------+--------+
4 rows in set (0.00 sec)

mysql> select * from course_details;
+------+-----------+-------------+
| id   | name      | course_name |
+------+-----------+-------------+
|    1 | Rishikesh | Python      |
|    2 | Rutuja    | Java        |
|    3 | Aakash    | C++         |
|    4 | Amey      | Frontend    |
|    5 | Neha      | C           |
+------+-----------+-------------+
5 rows in set (0.00 sec)

mysql> select details.id, details.name, details.gender, course_details.course_name from details inner join course_details on details.id = course_details.id;

+------+-----------+--------+-------------+
| id   | name      | gender | course_name |
+------+-----------+--------+-------------+
|    1 | Rishikesh | Male   | Python      |
|    2 | Rutuja    | Female | Java        |
|    3 | Aakash    | Male   | C++         |
|    4 | Neha      | Female | Frontend    |
+------+-----------+--------+-------------+
4 rows in set (0.00 sec)

mysql> |
```

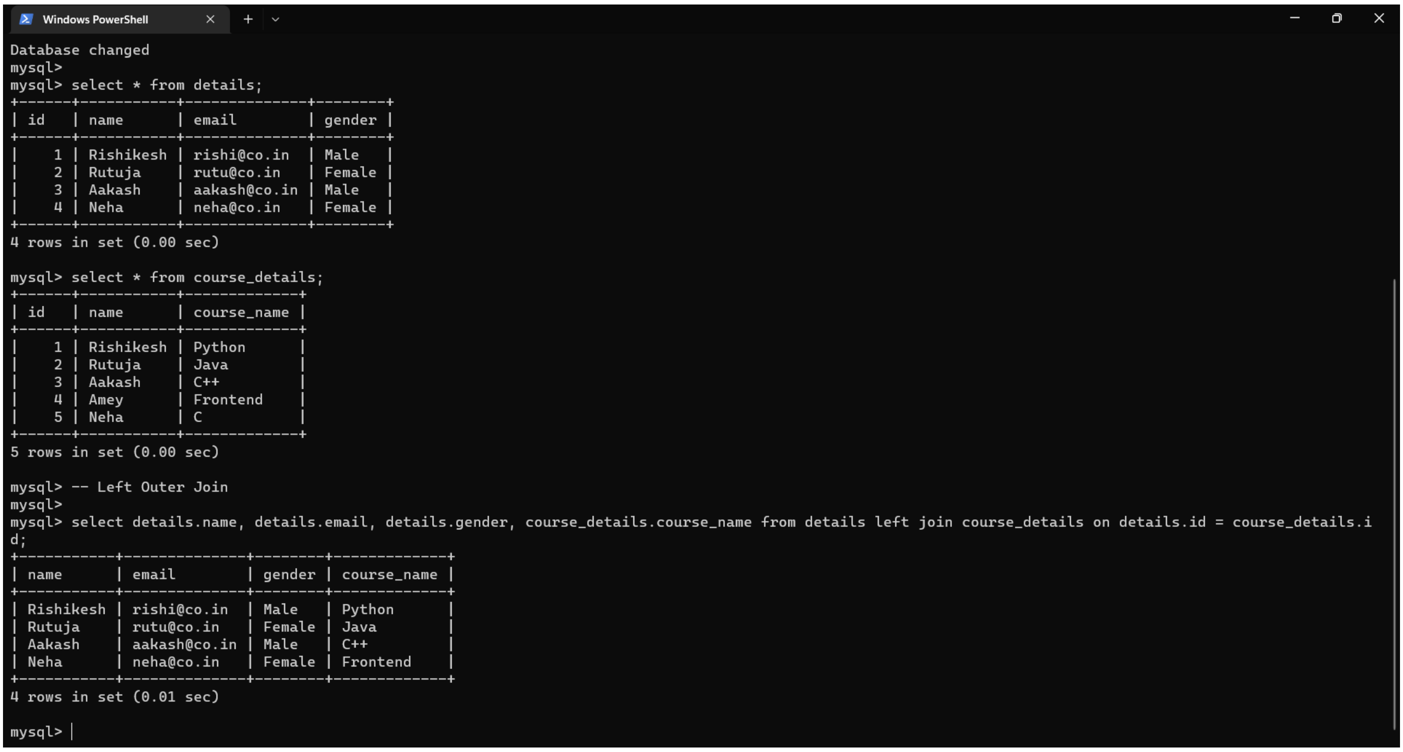
Left Outer Join

The LEFT OUTER JOIN returns all rows from the left hand table specified in the ON condition and only those rows from the other table where the join condition is fulfilled.

Syntax:

SELECT columns
FROM table1
LEFT [OUTER] JOIN table2
ON table1.column = table2.column;

Output:-



```
Database changed
mysql>
mysql> select * from details;
+------+-----------+--------------+--------+
| id   | name      | email        | gender |
+------+-----------+--------------+--------+
|    1 | Rishikesh | rishi@co.in  | Male   |
|    2 | Rutuja    | rutu@co.in   | Female |
|    3 | Aakash    | aakash@co.in | Male   |
|    4 | Neha      | neha@co.in   | Female |
+------+-----------+--------------+--------+
4 rows in set (0.00 sec)

mysql> select * from course_details;
+------+-----------+-------------+
| id   | name      | course_name |
+------+-----------+-------------+
|    1 | Rishikesh | Python      |
|    2 | Rutuja    | Java        |
|    3 | Aakash    | C++         |
|    4 | Amey      | Frontend    |
|    5 | Neha      | C           |
+------+-----------+-------------+
5 rows in set (0.00 sec)

mysql> -- Left Outer Join
mysql>
mysql> select details.name, details.email, details.gender, course_details.course_name from details left join course_details on details.id = course_details.i
d;
+-----------+--------------+--------+-------------+
| name      | email        | gender | course_name |
+-----------+--------------+--------+-------------+
| Rishikesh | rishi@co.in  | Male   | Python      |
| Rutuja    | rutu@co.in   | Female | Java        |
| Aakash    | aakash@co.in | Male   | C++         |
| Neha      | neha@co.in   | Female | Frontend    |
+-----------+--------------+--------+-------------+
4 rows in set (0.01 sec)

mysql> |
```

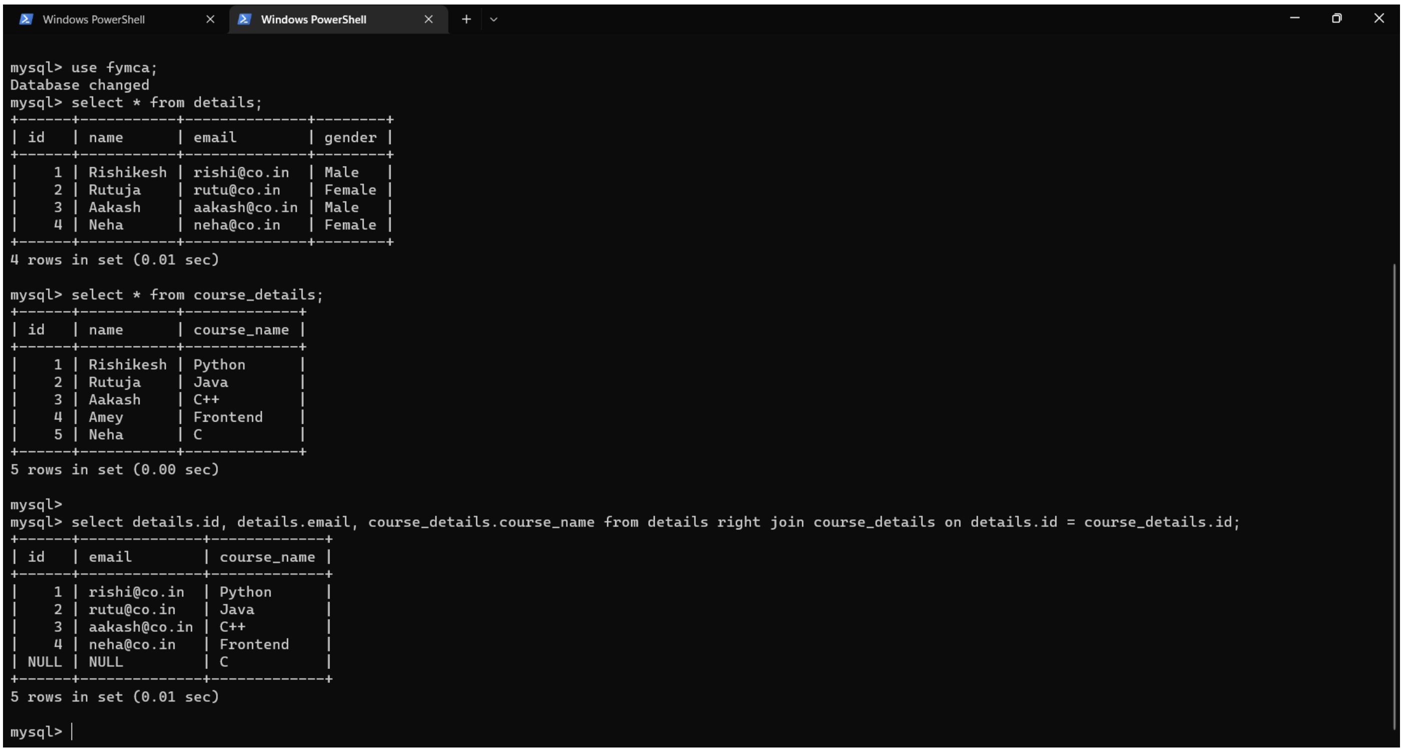
Right Outer Join

The MySQL Right Outer Join returns all rows from the RIGHT-hand table specified in the ON condition and only those rows from the other table where the join condition is fulfilled.

Syntax:

SELECT columns
FROM table1
RIGHT [OUTER] JOIN table2
ON table1.column = table2.column;

Output :-



```
mysql> use fymca;
Database changed
mysql> select * from details;
+------+-----------+--------------+--------+
| id   | name      | email        | gender |
+------+-----------+--------------+--------+
|    1 | Rishikesh | rishi@co.in  | Male   |
|    2 | Rutuja    | rutu@co.in   | Female |
|    3 | Aakash    | aakash@co.in | Male   |
|    4 | Neha      | neha@co.in   | Female |
+------+-----------+--------------+--------+
4 rows in set (0.01 sec)

mysql> select * from course_details;
+------+-----------+-------------+
| id   | name      | course_name |
+------+-----------+-------------+
|    1 | Rishikesh | Python      |
|    2 | Rutuja    | Java        |
|    3 | Aakash    | C++         |
|    4 | Amey      | Frontend    |
|    5 | Neha      | C           |
+------+-----------+-------------+
5 rows in set (0.00 sec)

mysql>
mysql> select details.id, details.email, course_details.course_name from details right join course_details on details.id = course_details.id;
+------+--------------+-------------+
| id   | email        | course_name |
+------+--------------+-------------+
|    1 | rishi@co.in  | Python      |
|    2 | rutu@co.in   | Java        |
|    3 | aakash@co.in | C++         |
|    4 | neha@co.in   | Frontend    |
| NULL | NULL         | C           |
+------+--------------+-------------+
5 rows in set (0.01 sec)

mysql>
```

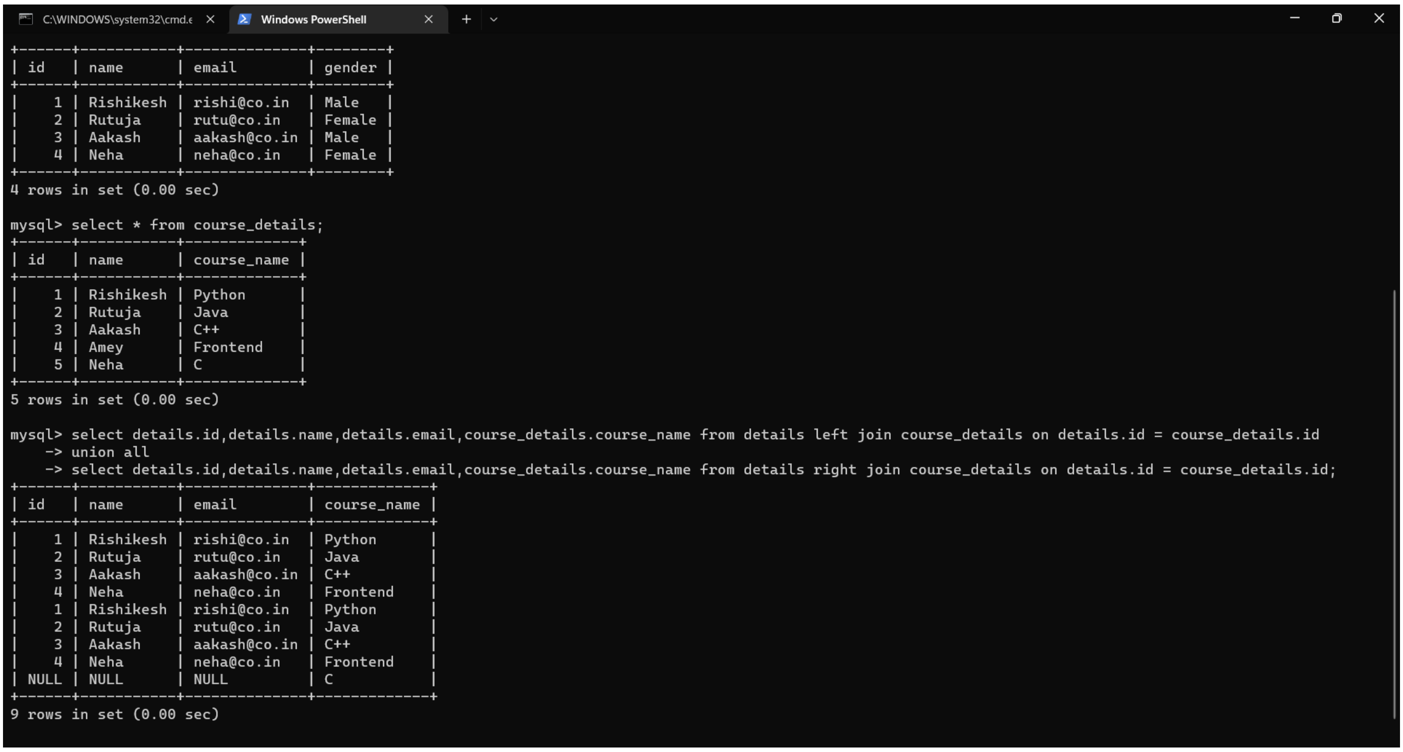
Full Join

1. The SQL full join is the result of combination of both left and right outer join and the join tables have all the records from both tables. It puts NULL on the place of matches not found.
2. Since MySQL doesn't have a **Full Join** command we will combine both **Left** and **Right** joins with the help of **Union** command to achieve **Full Join**.
3. **MySQL Union** clause allows us to combine two or more relations using multiple SELECT queries into a single result set.

Syntax:

SELECT col1,col2,... from table1 left join table2 on table1.column = table2.column
UNION ALL
SELECT col1,col2,... from table1 right join table2 on table1.column = table2.column;

Output:-



```
+------+-----------+--------------+--------+
| id   | name      | email        | gender |
+------+-----------+--------------+--------+
|    1 | Rishikesh | rishi@co.in  | Male   |
|    2 | Rutuja    | rutu@co.in   | Female |
|    3 | Aakash    | aakash@co.in | Male   |
|    4 | Neha      | neha@co.in   | Female |
+------+-----------+--------------+--------+
4 rows in set (0.00 sec)

mysql> select * from course_details;
+------+-----------+-------------+
| id   | name      | course_name |
+------+-----------+-------------+
|    1 | Rishikesh | Python      |
|    2 | Rutuja    | Java        |
|    3 | Aakash    | C++         |
|    4 | Amey      | Frontend    |
|    5 | Neha      | C           |
+------+-----------+-------------+
5 rows in set (0.00 sec)

mysql> select details.id,details.name,details.email,course_details.course_name from details left join course_details on details.id = course_details.id
    -> union all
    -> select details.id,details.name,details.email,course_details.course_name from details right join course_details on details.id = course_details.id;
+------+-----------+--------------+-------------+
| id   | name      | email        | course_name |
+------+-----------+--------------+-------------+
|    1 | Rishikesh | rishi@co.in  | Python      |
|    2 | Rutuja    | rutu@co.in   | Java        |
|    3 | Aakash    | aakash@co.in | C++         |
|    4 | Neha      | neha@co.in   | Frontend    |
|    1 | Rishikesh | rishi@co.in  | Python      |
|    2 | Rutuja    | rutu@co.in   | Java        |
|    3 | Aakash    | aakash@co.in | C++         |
|    4 | Neha      | neha@co.in   | Frontend    |
| NULL | NULL      | NULL         | C           |
+------+-----------+--------------+-------------+
9 rows in set (0.00 sec)
```

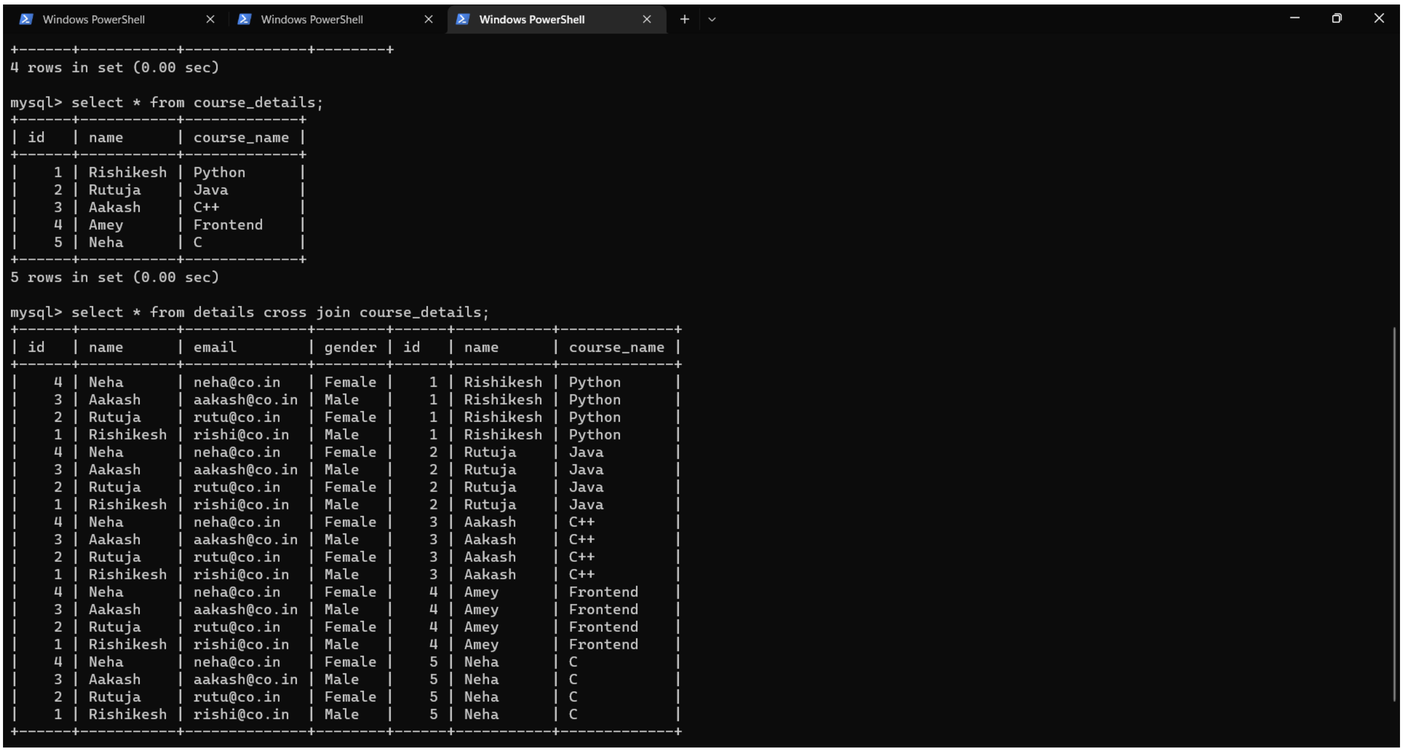
Cross Join

MySQL CROSS JOIN is used to combine all possibilities of the two or more tables and returns the result that contains every row from all contributing tables. The CROSS JOIN is also known as CARTESIAN JOIN, which provides the Cartesian product of all associated tables.

Syntax:

SELECT column-lists
FROM table1
CROSS JOIN table2;

Output:-



```
+------+-----------+--------------+--------+
4 rows in set (0.00 sec)

mysql> select * from course_details;
+------+-----------+-------------+
| id   | name      | course_name |
+------+-----------+-------------+
|    1 | Rishikesh | Python      |
|    2 | Rutuja    | Java        |
|    3 | Aakash    | C++         |
|    4 | Amey      | Frontend    |
|    5 | Neha      | C           |
+------+-----------+-------------+
5 rows in set (0.00 sec)

mysql> select * from details cross join course_details;
+------+-----------+--------------+--------+------+-----------+-------------+
| id   | name      | email        | gender | id   | name      | course_name |
+------+-----------+--------------+--------+------+-----------+-------------+
|    4 | Neha      | neha@co.in   | Female |    1 | Rishikesh | Python      |
|    3 | Aakash    | aakash@co.in | Male   |    1 | Rishikesh | Python      |
|    2 | Rutuja    | rutu@co.in   | Female |    1 | Rishikesh | Python      |
|    1 | Rishikesh | rishi@co.in  | Male   |    1 | Rishikesh | Python      |
|    4 | Neha      | neha@co.in   | Female |    2 | Rutuja    | Java        |
|    3 | Aakash    | aakash@co.in | Male   |    2 | Rutuja    | Java        |
|    2 | Rutuja    | rutu@co.in   | Female |    2 | Rutuja    | Java        |
|    1 | Rishikesh | rishi@co.in  | Male   |    2 | Rutuja    | Java        |
|    4 | Neha      | neha@co.in   | Female |    3 | Aakash    | C++         |
|    3 | Aakash    | aakash@co.in | Male   |    3 | Aakash    | C++         |
|    2 | Rutuja    | rutu@co.in   | Female |    3 | Aakash    | C++         |
|    1 | Rishikesh | rishi@co.in  | Male   |    3 | Aakash    | C++         |
|    4 | Neha      | neha@co.in   | Female |    4 | Amey      | Frontend    |
|    3 | Aakash    | aakash@co.in | Male   |    4 | Amey      | Frontend    |
|    2 | Rutuja    | rutu@co.in   | Female |    4 | Amey      | Frontend    |
|    1 | Rishikesh | rishi@co.in  | Male   |    4 | Amey      | Frontend    |
|    4 | Neha      | neha@co.in   | Female |    5 | Neha      | C           |
|    3 | Aakash    | aakash@co.in | Male   |    5 | Neha      | C           |
|    2 | Rutuja    | rutu@co.in   | Female |    5 | Neha      | C           |
|    1 | Rishikesh | rishi@co.in  | Male   |    5 | Neha      | C           |
+------+-----------+--------------+--------+------+-----------+-------------+
```