1. What are joins in SQL? State its types.

In SQL, **joins** are used to combine rows from two or more tables based on a related column between them. Joins allow you to retrieve data that is spread across multiple tables, enabling more complex queries and reporting.

Types of Joins in SQL:

1. **Inner Join**:
   - **Definition**: Returns only the rows that have matching values in both tables.
   - **Usage**: SELECT * FROM A INNER JOIN B ON A.PK = B.PK;
2. **Left Join (Left Outer Join)**:
   - **Definition**: Returns all rows from the left table and the matching rows from the right table. If there is no match, NULL values are returned for columns from the right table.
   - **Usage**: SELECT * FROM A Left Outer Join B ON A.PK = B.PK;
3. **Right Join (Right Outer Join)**:
   - **Definition**: Returns all rows from the right table and the matching rows from the left table. If there is no match, NULL values are returned for columns from the left table.
   - **Usage**: SELECT * FROM A Right Outer Join B ON A.PK = B.PK;
4. **Full Join (Full Outer Join)**:
   - **Definition**: Returns all rows when there is a match in either the left or right table. If there is no match, NULL values are returned for columns from the table that lacks a matching row.
   - **Usage**: Select * from A Full Join B on A.PK = B.Pk;

     Or (SELECT * FROM A Left Outer Join B ON A.PK = B.PK) Union (SELECT * FROM A Right Outer Join B ON A.PK = B.PK);

5. **Cross Join**:
    - **Definition**: Returns the Cartesian product of the two tables, meaning it returns all possible combinations of rows from both tables.
    - **Usage**: SELECT * FROM A CROSS JOIN B;
6. **Natural Join**:
    - **Definition**: Automatically joins tables based on columns with the same name and data type in both tables. This join doesn't require an explicit ON condition.
    - **Usage**: SELECT * FROM A Natural Join B;

Example:-

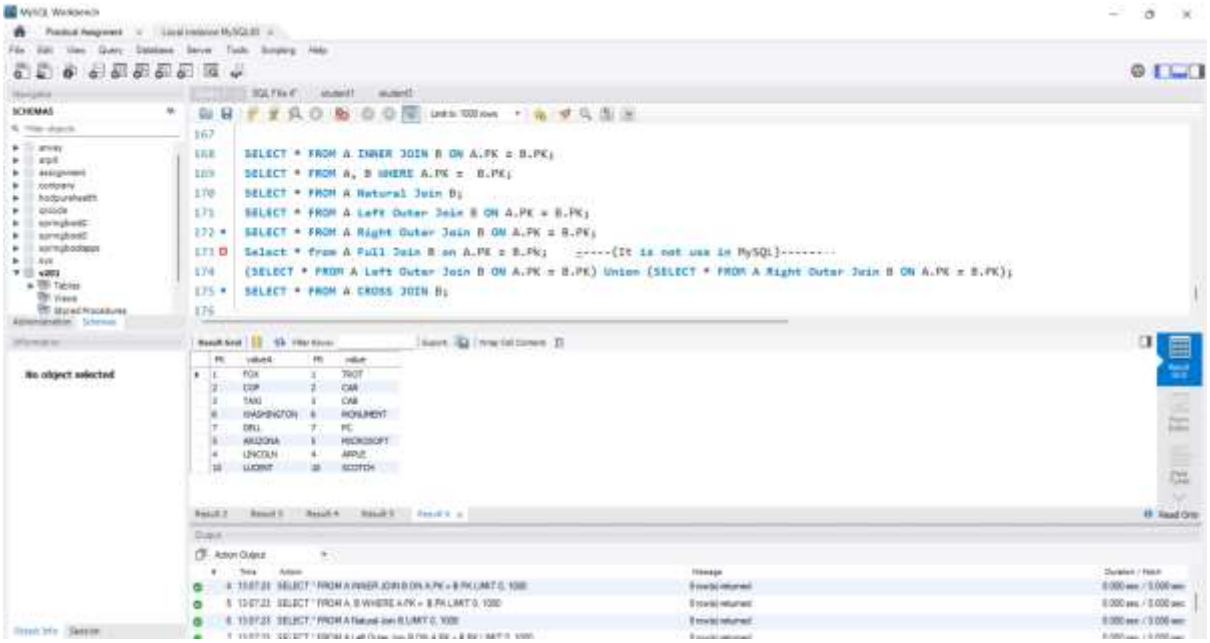
```
create table A (PK int primary key, value varchar(50));

insert into A values

(1, 'FOX'),

 (2, 'COP'),

 (3, 'TAXI'),

 (6, 'WASHINGTON'),

 (7, 'DELL'),

 (5, 'ARIZONA'),

 (4, 'LINCOLN'),

 (10, 'LUCENT');
```

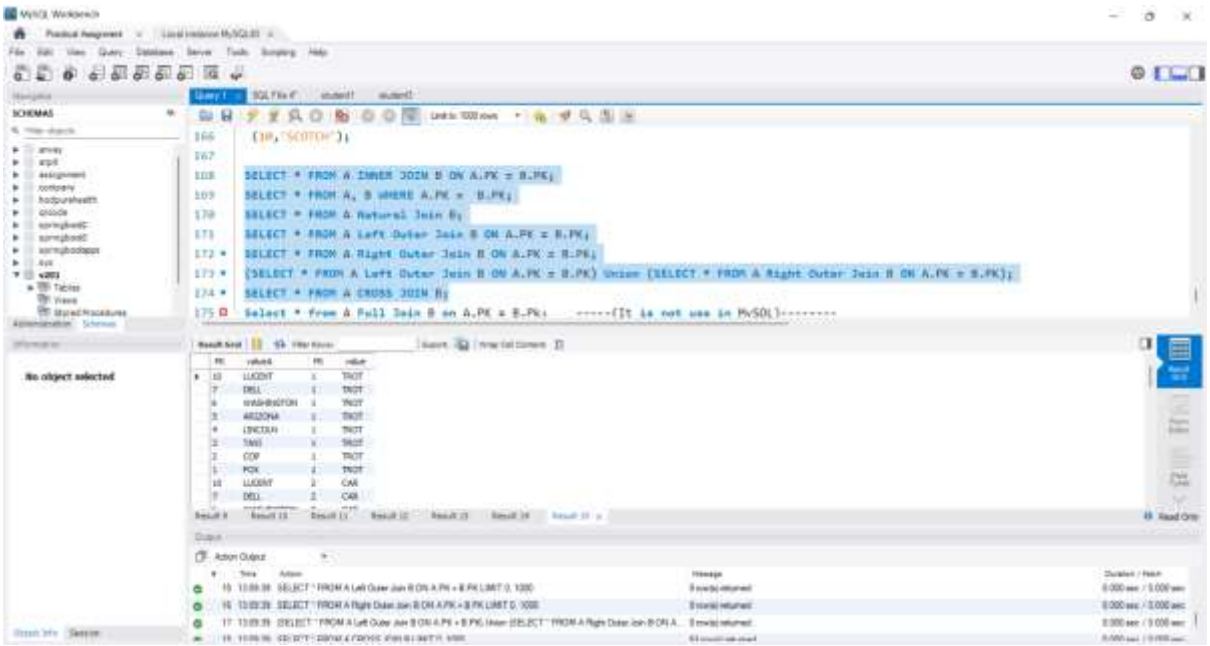
```
 create table B (PK int, value varchar(50),FOREIGN KEY (PK) REFERENCES A(PK));

 insert into B values

 (1, 'TROT'),

 (2, 'CAR'),

 (3, 'CAB'),

 (6, 'MONUMENT'),

 (7, 'PC'),

 (5, 'MICROSOFT'),

 (4, 'APPLE'),

 (10,'SCOTCH');


SELECT * FROM A INNER JOIN B ON A.PK = B.PK;

SELECT * FROM A, B WHERE A.PK =  B.PK;

SELECT * FROM A Natural Join B;

SELECT * FROM A Left Outer Join B ON A.PK = B.PK;

SELECT * FROM A Right Outer Join B ON A.PK = B.PK;
```

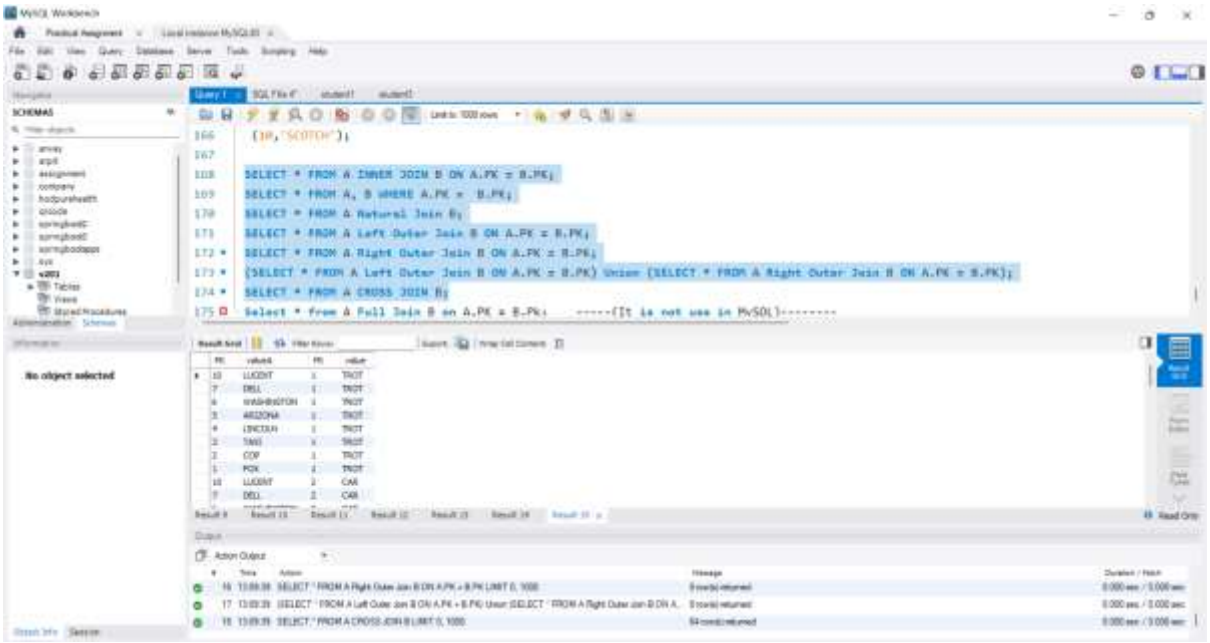
Select * from A Full Join B on A.PK = B.Pk; -----(It is not use in MySQL)--------

(SELECT * FROM A Left Outer Join B ON A.PK = B.PK) Union (SELECT * FROM A Right Outer Join B ON A.PK = B.PK);

SELECT * FROM A CROSS JOIN B;



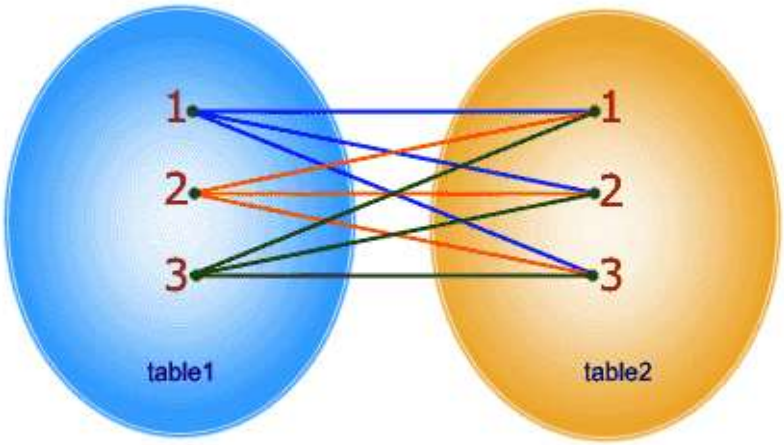




2. Define Cross join.

- If we use the cross join to combine two different tables, then we will get the Cartesian product of the sets of rows from the joined table.
- When each row of the first table is combined with each row from the second table, it is known as Cartesian join or cross join.
- After performing the cross join operation, the total number of rows present in the final table will be equal to the product of the number of rows present in table 1 and the number of rows present in table 2.
- **For example:**
  If there are two records in table 1 and three records in table 2, then after performing cross join operation, we will get six records in the final table.

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.

Example:-

create table A (PK int primary key, value varchar(50));

insert into A values

(1, 'FOX'),

(2, 'COP'),

(3, 'TAXI'),

(6, 'WASHINGTON'),

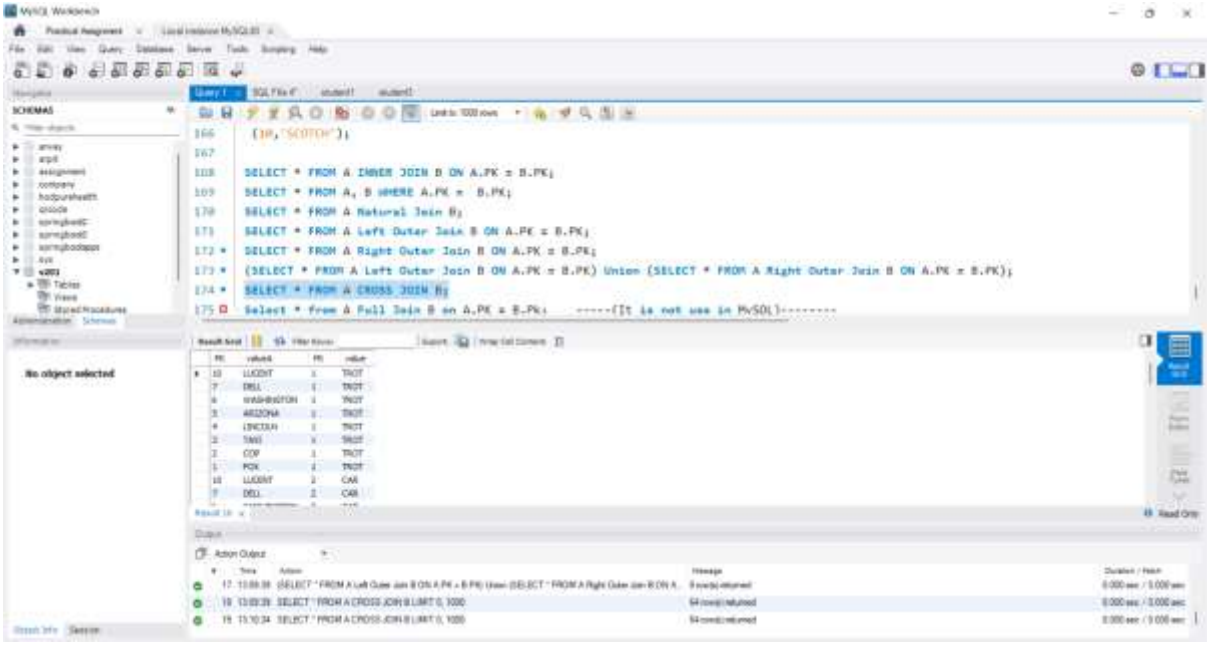
(7, 'DELL'),

(5, 'ARIZONA'),

(4, 'LINCOLN'),

(10, 'LUCENT');

```
 create table B (PK int, value varchar(50),FOREIGN KEY (PK) REFERENCES A(PK));

 insert into B values

 (1, 'TROT'),

 (2, 'CAR'),

 (3, 'CAB'),

 (6, 'MONUMENT'),

 (7, 'PC'),

 (5, 'MICROSOFT'),

 (4, 'APPLE'),

 (10,'SCOTCH');

SELECT * FROM A CROSS JOIN B;
```


MySQL Workbench
SCHEMAS
No object selected
(10,'SCOTCH');
SELECT * FROM A INNER JOIN B ON A.PK = B.PK;
SELECT * FROM A, B WHERE A.PK = B.PK;
SELECT * FROM A Natural Join B;
SELECT * FROM A Left Outer Join B ON A.PK = B.PK;
SELECT * FROM A Right Outer Join B ON A.PK = B.PK;
(SELECT * FROM A Left Outer Join B ON A.PK = B.PK) Union (SELECT * FROM A Right Outer Join B ON A.PK = B.PK);
SELECT * FROM A CROSS JOIN B;
Select * from A Full Join B on A.PK = B.Pk; -----(It is not use in MySQL)--------
Action Output