JOINS

INNER JOIN:

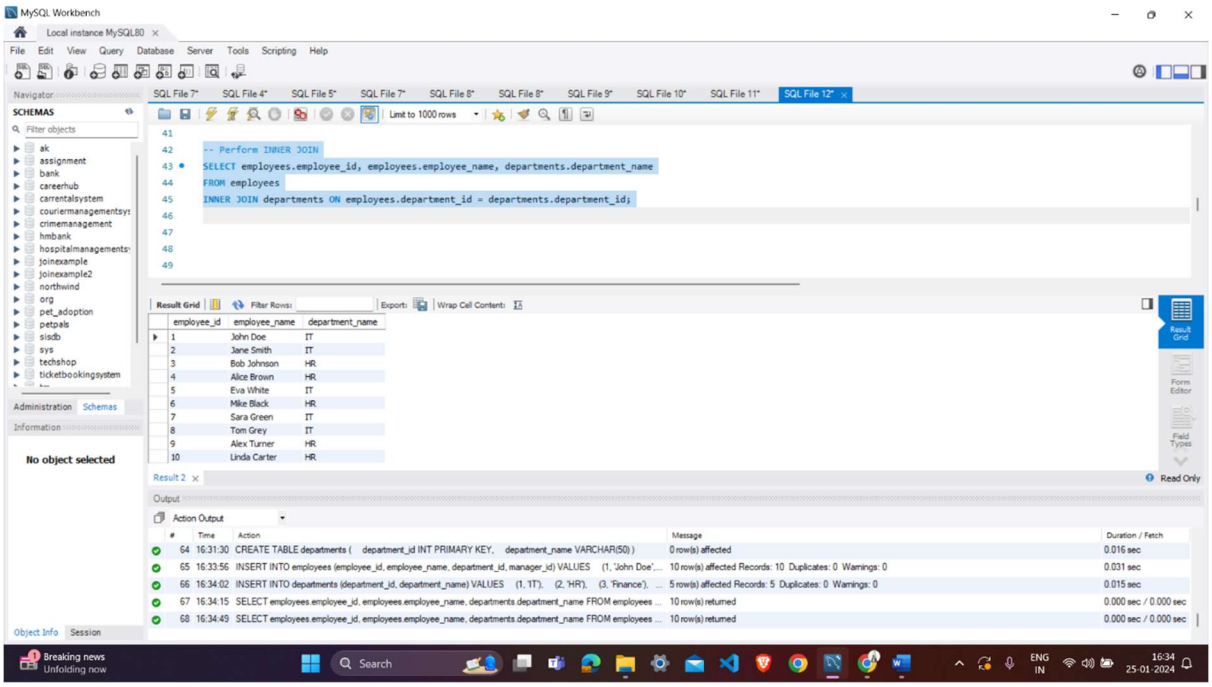
- An INNER JOIN retrieves records from both tables where there is a match based on the specified condition.
- Only rows with matching values in both tables' columns are included in the result set.
- If there is no match, the row is excluded from the result set.

-- Perform INNER JOIN

SELECT employees.employee_id, employees.employee_name, departments.department_name

FROM employees

INNER JOIN departments ON employees.department_id = departments.department_id;



LEFT JOIN (or LEFT OUTER JOIN):

- A LEFT JOIN 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.

The result set will always contain all rows from the left table, even if there are no matches in the right table.

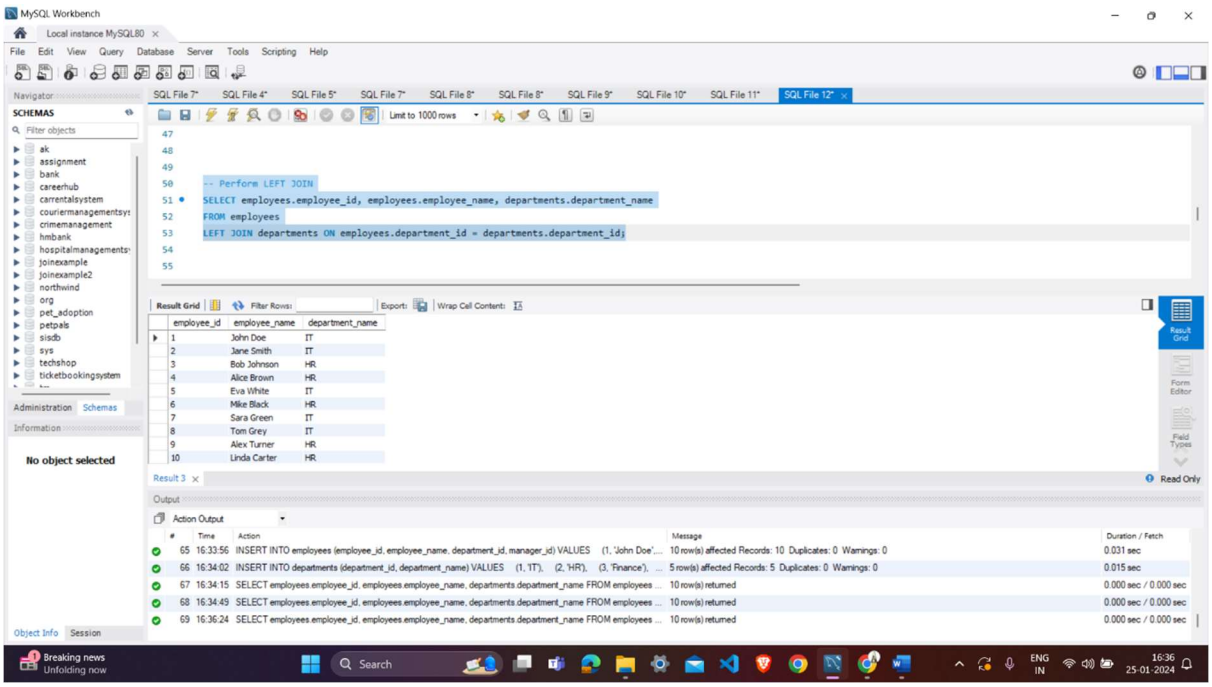
-- Perform LEFT JOIN

SELECT employees.employee_id, employees.employee_name, departments.department_name

FROM employees

LEFT JOIN departments ON employees.department_id = departments.department_id;

-



RIGHT JOIN (or RIGHT OUTER JOIN):

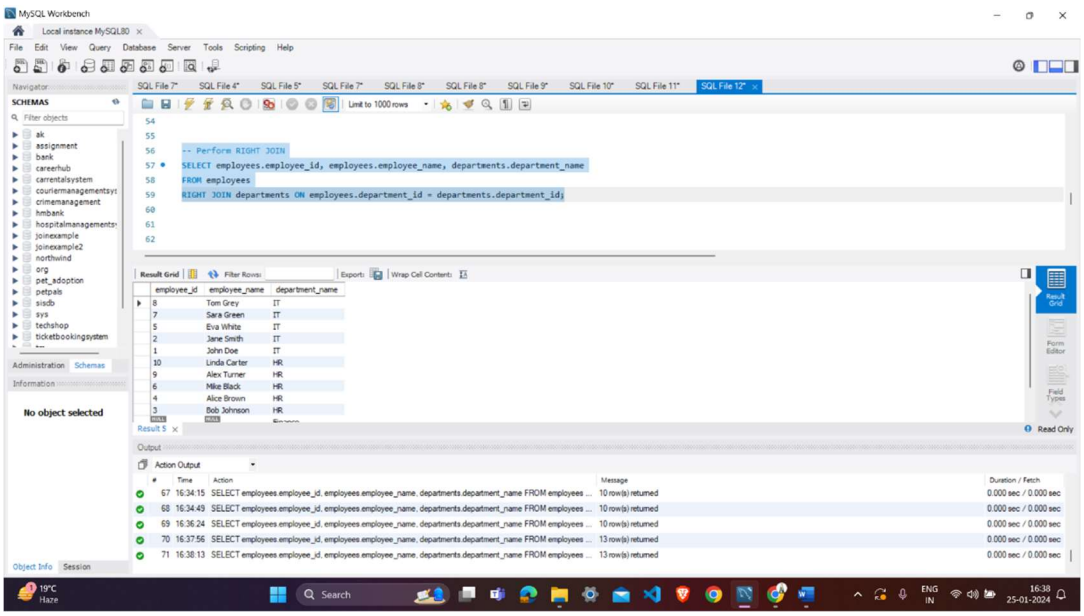
- A RIGHT JOIN is similar to a LEFT JOIN, but it returns all rows from the right table and the matching rows from the left table.
- If there is no match, NULL values are returned for columns from the left table.
- The result set will always contain all rows from the right table, even if there are no matches in the left table.

-- Perform RIGHT JOIN

SELECT employees.employee_id, employees.employee_name, departments.department_name

FROM employees

RIGHT JOIN departments ON employees.department_id = departments.department_id;



FULL JOIN (or FULL OUTER JOIN):

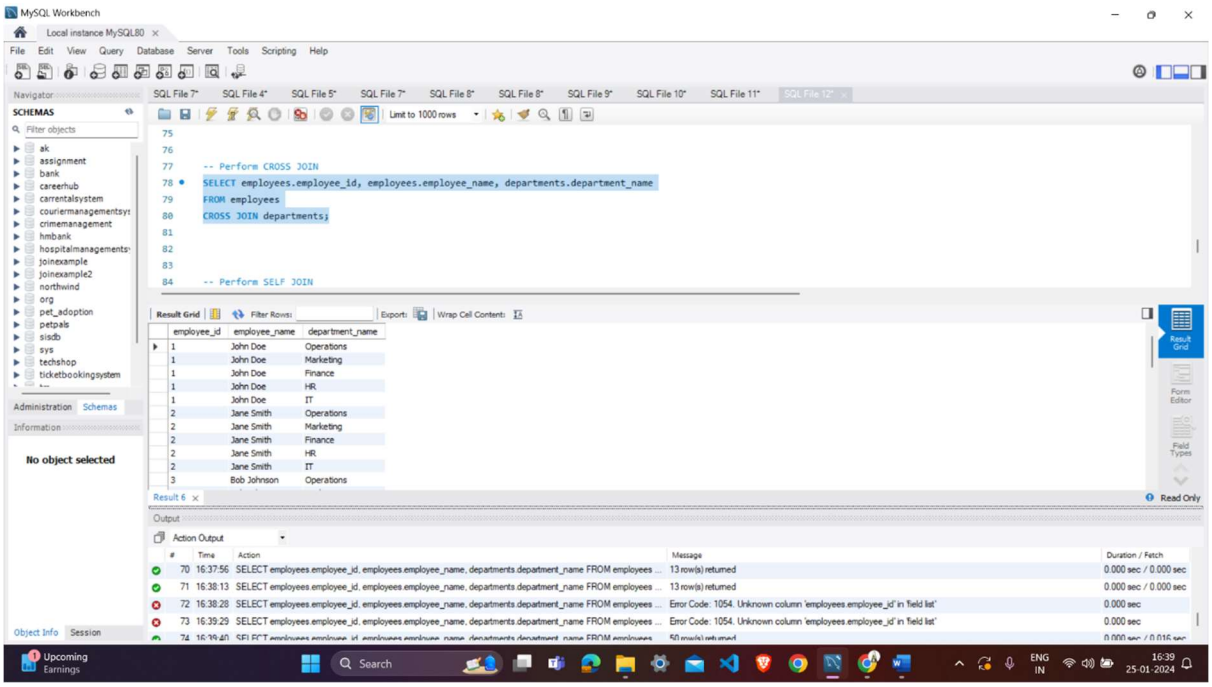
- A FULL JOIN returns all rows when there is a match in either the left or the right table.
- If there is no match, NULL values are returned for columns from the table without a match.
- The result set includes all rows from both tables, with NULL values where there are no matches.

```
-- Perform FULL JOIN
SELECT employees.employee_id, employees.employee_name, departments.department_name
FROM employees
FULL JOIN departments ON employees.department_id = departments.department_id
LIMIT 0, 1000;
```

CROSS JOIN:

- A CROSS JOIN returns the Cartesian product of the two tables, meaning it combines each row from the first table with every row from the second table.
- It does not require a matching condition.
- The result set has a number of rows equal to the product of the number of rows in the two tables being joined.

```
-- Perform CROSS JOIN
SELECT employees.employee_id, employees.employee_name, departments.department_name
FROM employees
CROSS JOIN departments;
```



SELF JOIN:

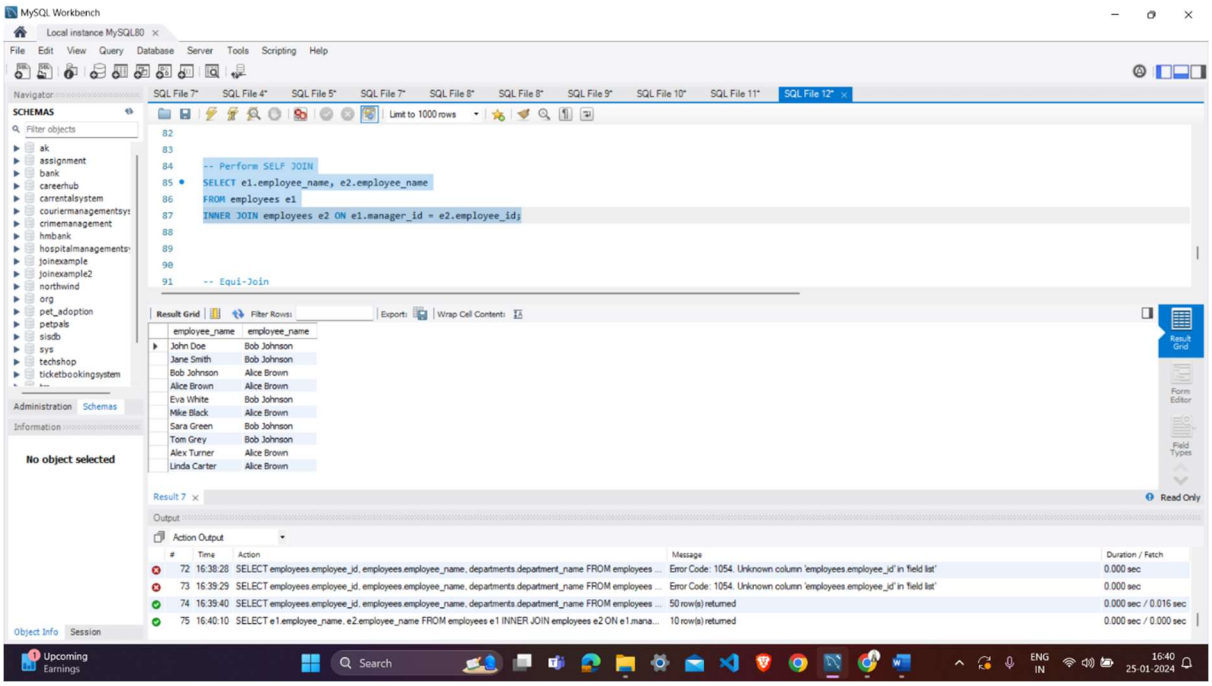
- A SELF JOIN is a regular join, but it involves joining a table with itself.
- It is used when you want to combine rows in a table based on a related column within the same table.
- Alias names are often used to distinguish between the two instances of the same table.

```
-- Perform SELF JOIN

SELECT e1.employee_name, e2.employee_name

FROM employees e1

INNER JOIN employees e2 ON e1.manager_id = e2.employee_id;
```



Equi-Join:

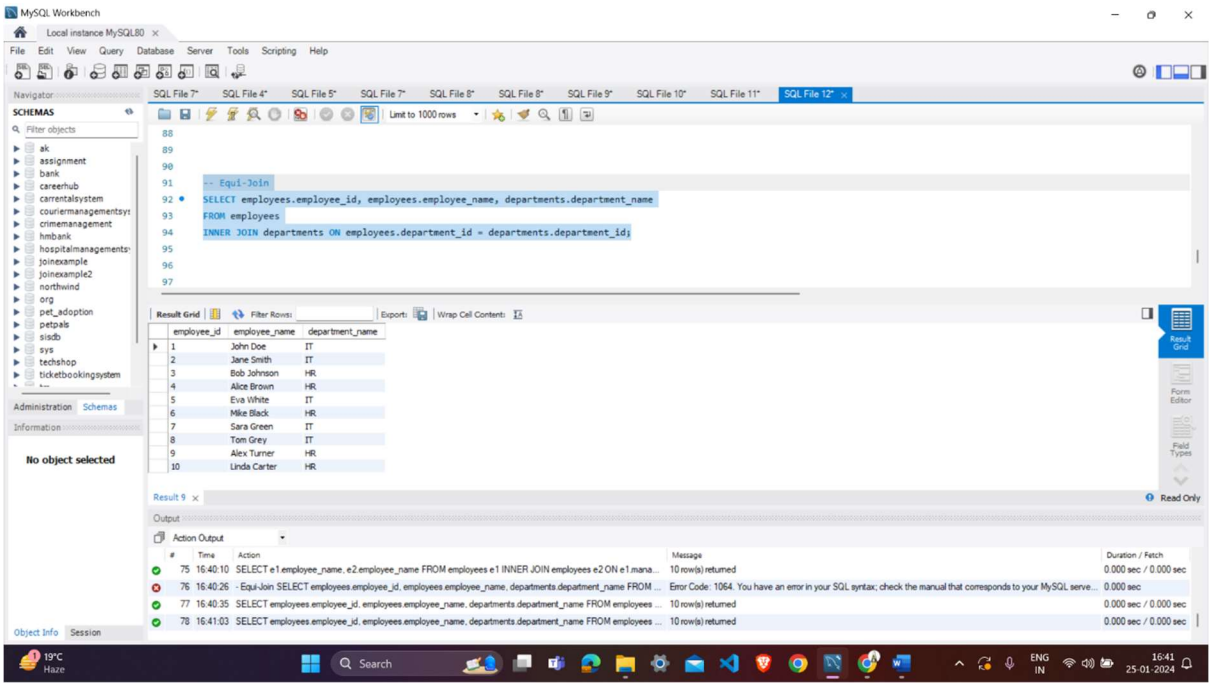
- An equi-join is a type of join where the matching condition uses the equality operator (=) to compare values in the specified columns.
- It is the most common type of join and is used to retrieve rows from both tables where the specified columns have matching values.

-- Equi-Join

SELECT employees.employee_id, employees.employee_name, departments.department_name

FROM employees

INNER JOIN departments ON employees.department_id = departments.department_id;



Non-Equi Join:

- A non-equi join is a join where the matching condition involves a comparison other than equality (using operators like <, >, <=, >=, <>).
- Non-equi joins are less common than equi-joins but can be useful in certain scenarios.
-

They are typically used when the relationship between the columns is based on a range of values rather than exact matches.

It's important to note that while equi-joins are more common, non-equi-joins provide flexibility in handling more complex relationships where exact matches are not sufficient. The choice between equi-joins and non-equi-joins depends on the specific requirements of your query and the relationships between the columns in the tables you are joining.

-- Non-Equi Join

SELECT e1.employee_name AS employee, e2.employee_name AS manager

FROM employees e1

JOIN employees e2 ON e1.manager_id = e2.employee_id;