

SQL Server NOT Condition (Operator)

javatpoint.com/sql-server-not-operator

The SQL Server NOT condition or NOT operator is used to negate a condition in a SELECT, INSERT, UPDATE, or DELETE statement.

Syntax:

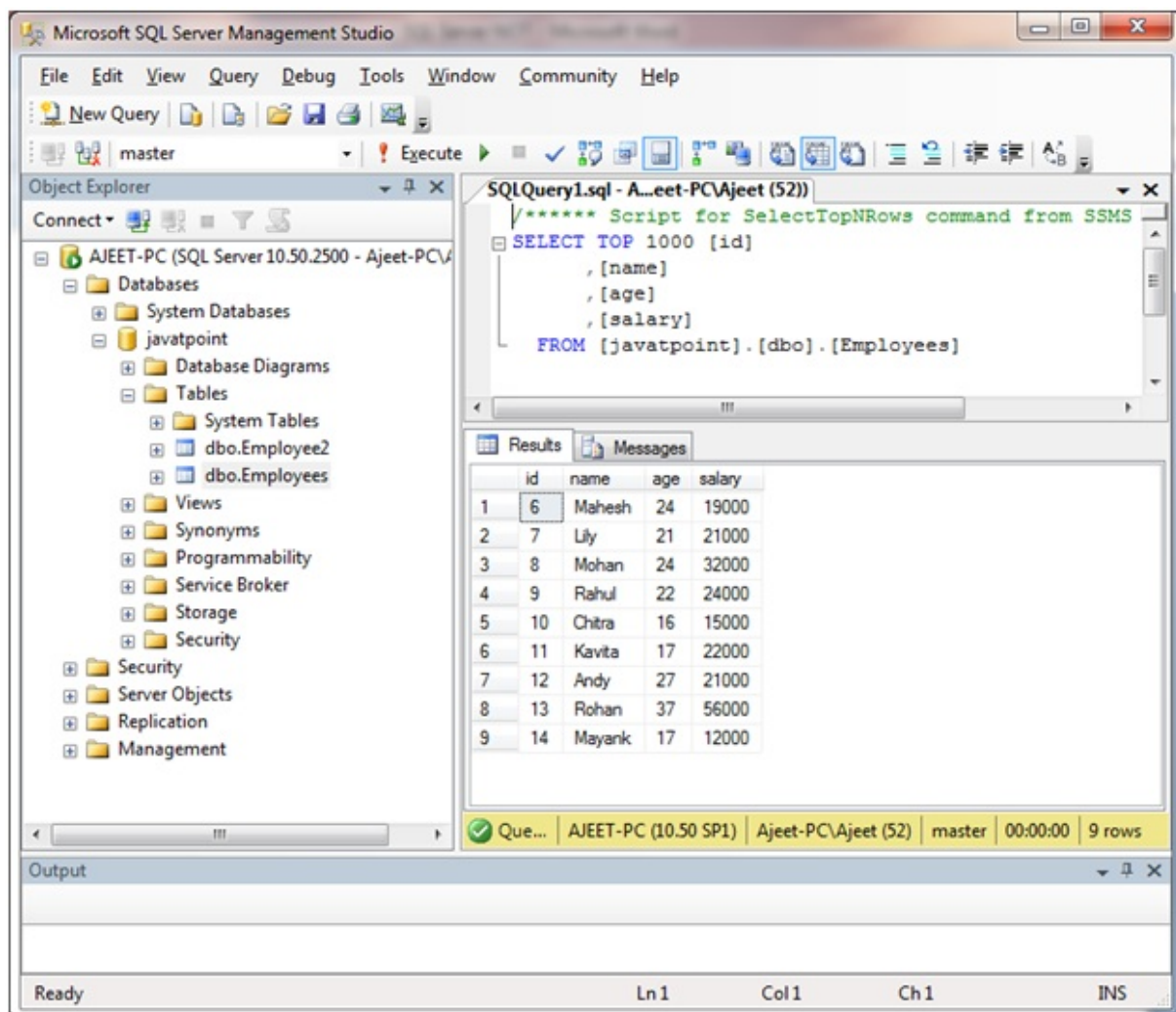
1. NOT condition

Parameter explanation

condition: It specifies the condition that you want to negate.

NOT Operator with IN condition

We have a table "Employees", having the following data:



The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'AJEET-PC (SQL Server 10.50.2500 - Ajeet-PC)'. The SQL Query window shows the following query:

```
SELECT TOP 1000 [id]
, [name]
, [age]
, [salary]
FROM [javatpoint].[dbo].[Employees]
```

The Results pane shows the following data:

	id	name	age	salary
1	6	Mahesh	24	19000
2	7	Lily	21	21000
3	8	Mohan	24	32000
4	9	Rahul	22	24000
5	10	Chitra	16	15000
6	11	Kavita	17	22000
7	12	Andy	27	21000
8	13	Rohan	37	56000
9	14	Mayank	17	12000

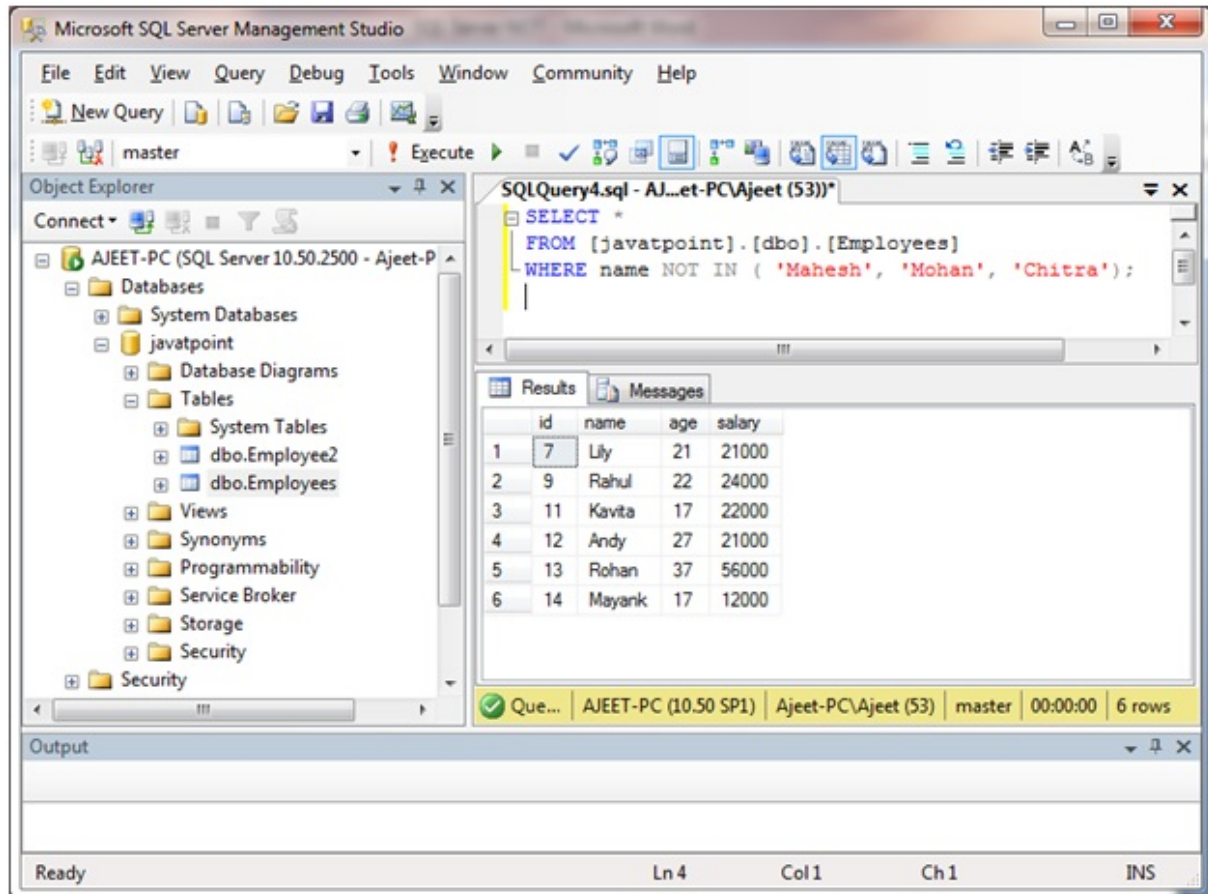
The status bar at the bottom indicates 'Ready', 'Ln 1', 'Col 1', 'Ch 1', and 'INS'.

Example

Retrieve all employees from the table where the following "NOT" condition is satisfied:

1. SELECT *
2. FROM [javatpoint].[dbo].[Employees]
3. WHERE name NOT IN ('Mahesh', 'Mohan', 'Chitra');

Output:

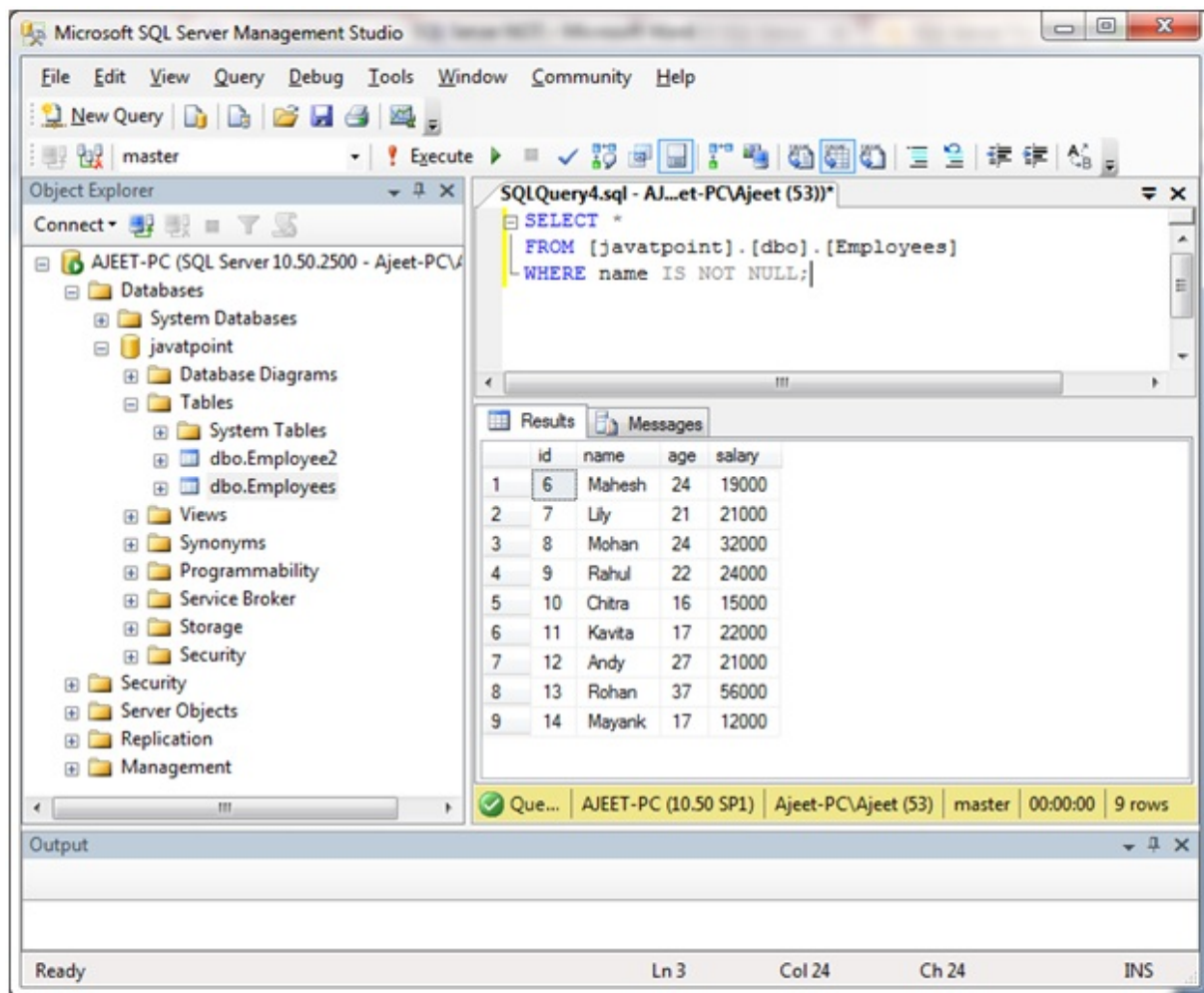


NOT Operator with IS NULL condition

Retrieve all employees from the table "Employees" which follows the IS NOT NULL condition:

1. SELECT *
2. FROM [javatpoint].[dbo].[Employees]
3. WHERE name IS NOT NULL;

Output:

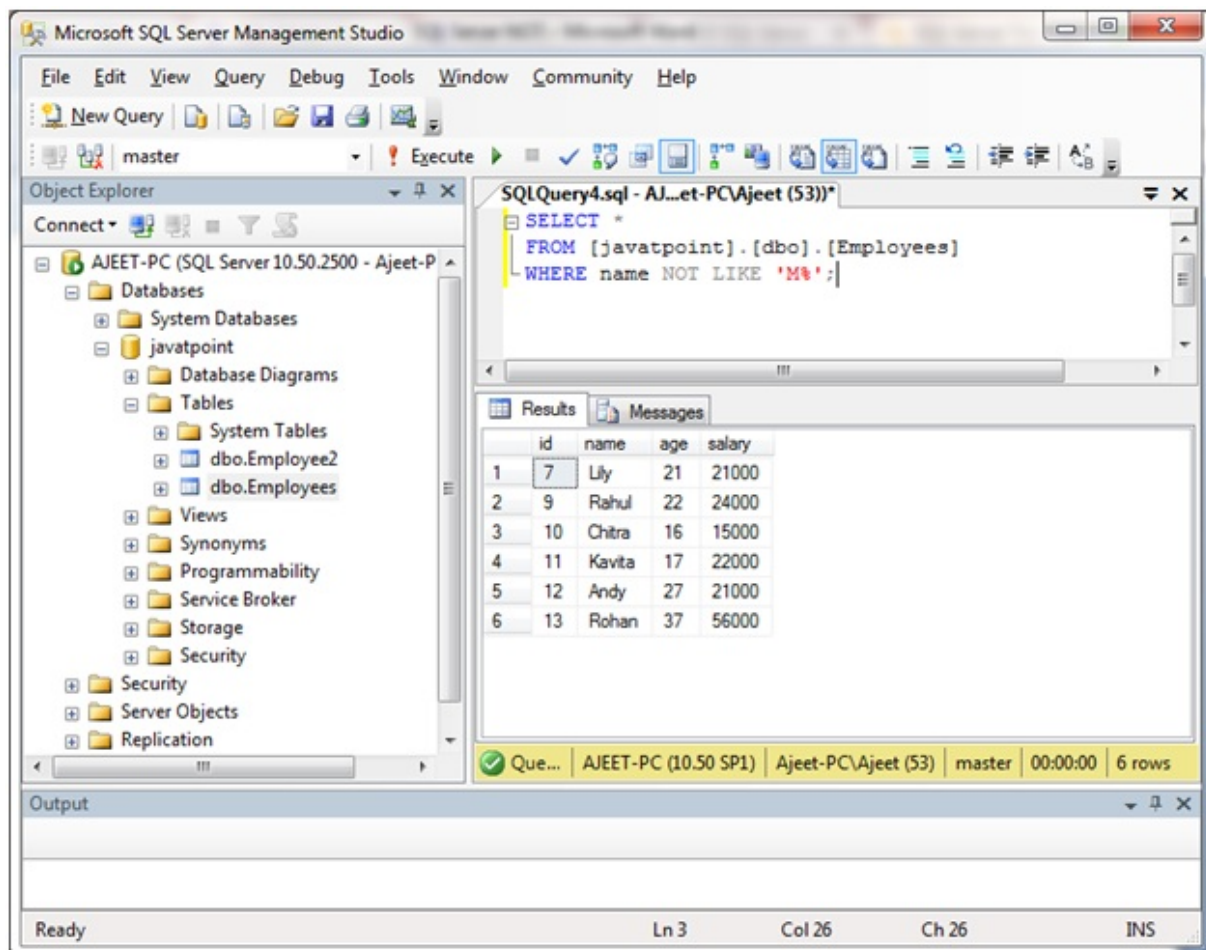


NOT Operator with LIKE condition

Retrieve all employees from the table "Employees" which follows the NOT LIKE condition. It will display only names which don't start with "M".

1. SELECT *
2. FROM [javatpoint].[dbo].[Employees]
3. WHERE name NOT LIKE 'M%';

Output:

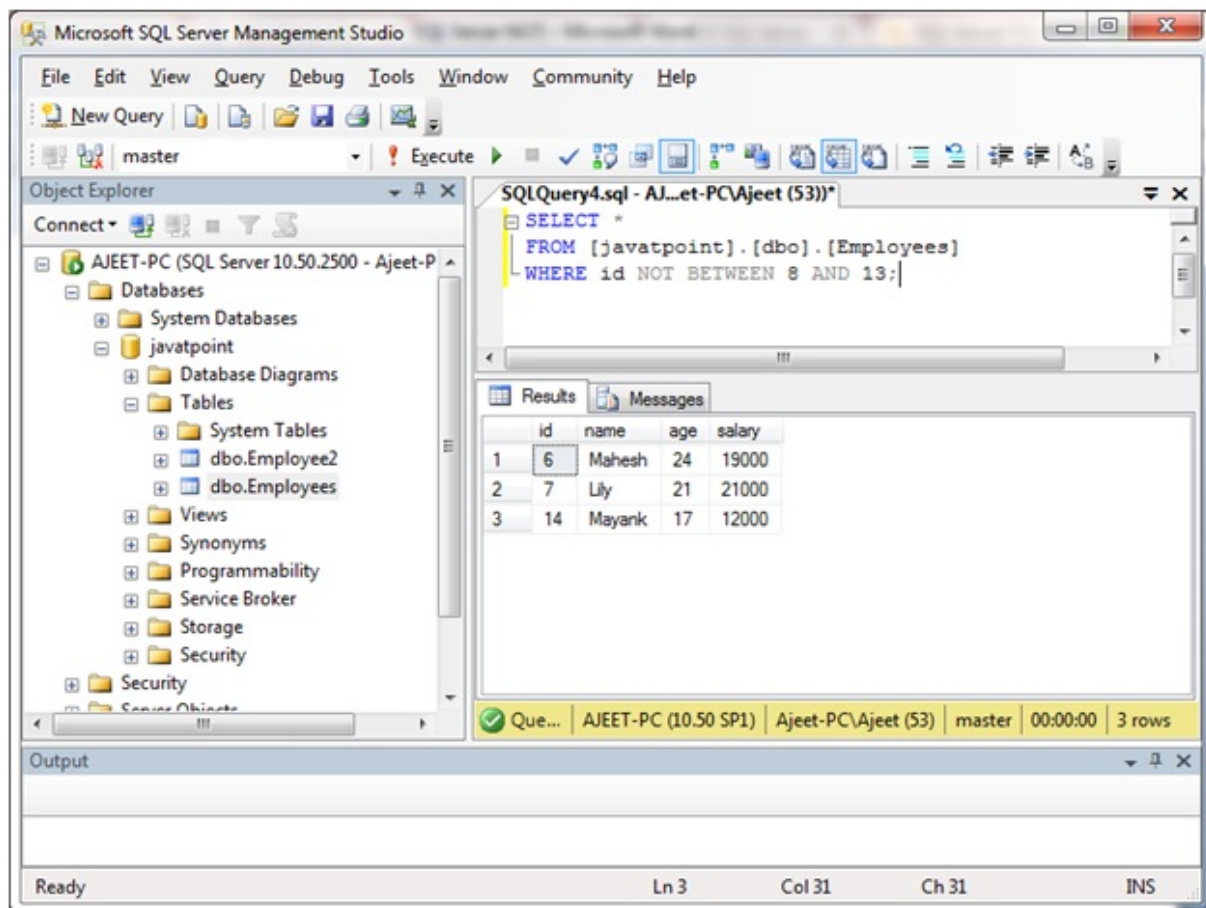


NOT Operator with BETWEEN condition

Retrieve all employees from the table "Employees", which follows the following condition. It will not show the employees between the id 8 to 13.

1. SELECT *
2. FROM [javatpoint].[dbo].[Employees]
3. WHERE id NOT BETWEEN 8 AND 13;

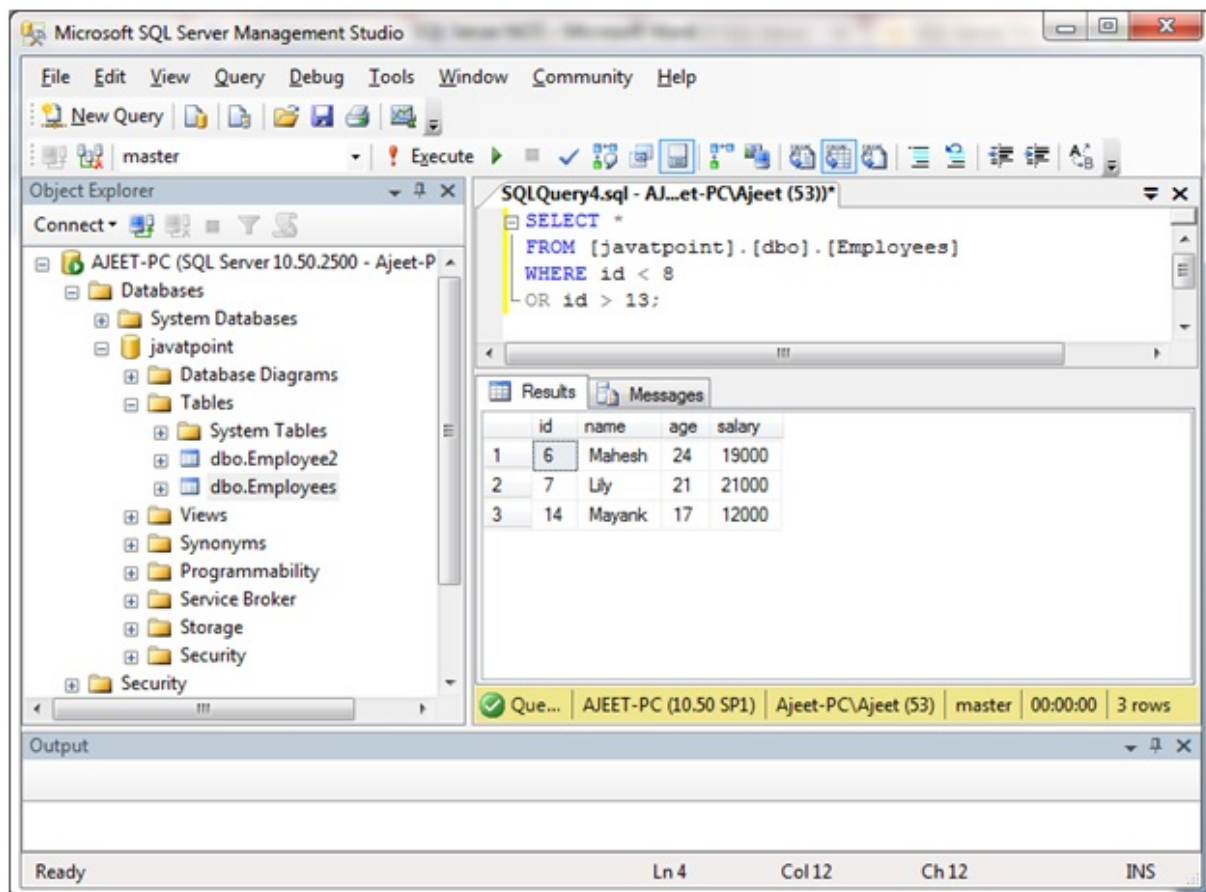
Output:



Or

1. SELECT *
2. FROM [javatpoint].[dbo].[Employees]
3. WHERE id < 8
4. OR id > 13;

Output:



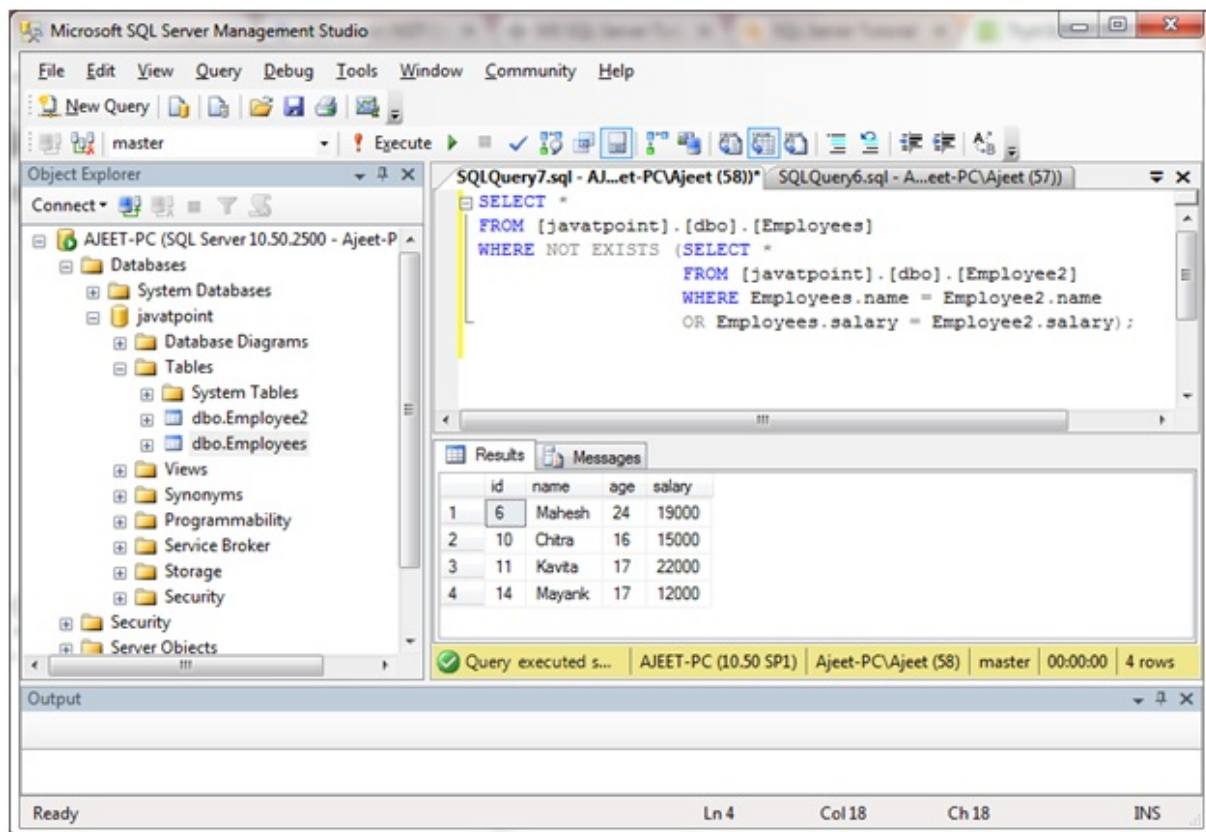
NOT Operator with EXISTS condition

You can also combine SQL Server NOT condition with the EXISTS condition.

OR condition:

1. SELECT *
2. FROM [javatpoint].[dbo].[Employees]
3. WHERE NOT EXISTS (SELECT *
4. FROM [javatpoint].[dbo].[Employee2]
5. WHERE Employees.name = Employee2.name
6. OR Employees.salary = Employee2.salary);

Output:



AND Condition:

1. SELECT *
2. FROM [javatpoint].[dbo].[Employees]
3. WHERE NOT EXISTS (SELECT *
4. FROM [javatpoint].[dbo].[Employee2]
5. WHERE Employees.name = Employee2.name
6. AND Employees.salary = Employee2.salary);

Output:

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'AJEET-PC (SQL Server 10.50.2500 - Ajeet-PC)'. The main query window shows the following SQL query:

```

SELECT *
FROM [javatpoint].[dbo].[Employees]
WHERE NOT EXISTS (SELECT *
                  FROM [javatpoint].[dbo].[Employee2]
                  WHERE Employees.name = Employee2.name
                  AND Employees.salary = Employee2.salary);

```

The Results pane shows the output of the query, which is a list of 9 rows from the Employees table. The status bar at the bottom indicates 'Query executed successfully' and '9 rows'.

	id	name	age	salary
1	6	Mahesh	24	19000
2	7	Lily	21	21000
3	8	Mohan	24	32000
4	9	Rahul	22	24000
5	10	Chitra	16	15000
6	11	Kavita	17	22000
7	12	Andy	27	21000
8	13	Rohan	37	56000
9	14	Mayank	17	12000

This SQL Server NOT example would return all records from the Employees table where there are no records in the Employee2 table for the matching name and salary.