

# Querying Data using Built-in Functions and T-SQL-II

---

Demo 1 – Manage Data using SQL Operators



**edureka!**

© Brain4ce Education Solutions Pvt. Ltd.

## Manage Data using SQL Operators

**Problem Statement:** A startup company called Electro is planning to open a new branch in New Delhi. They need to do a cost analysis for this. They require employee details from other branches for this analysis.

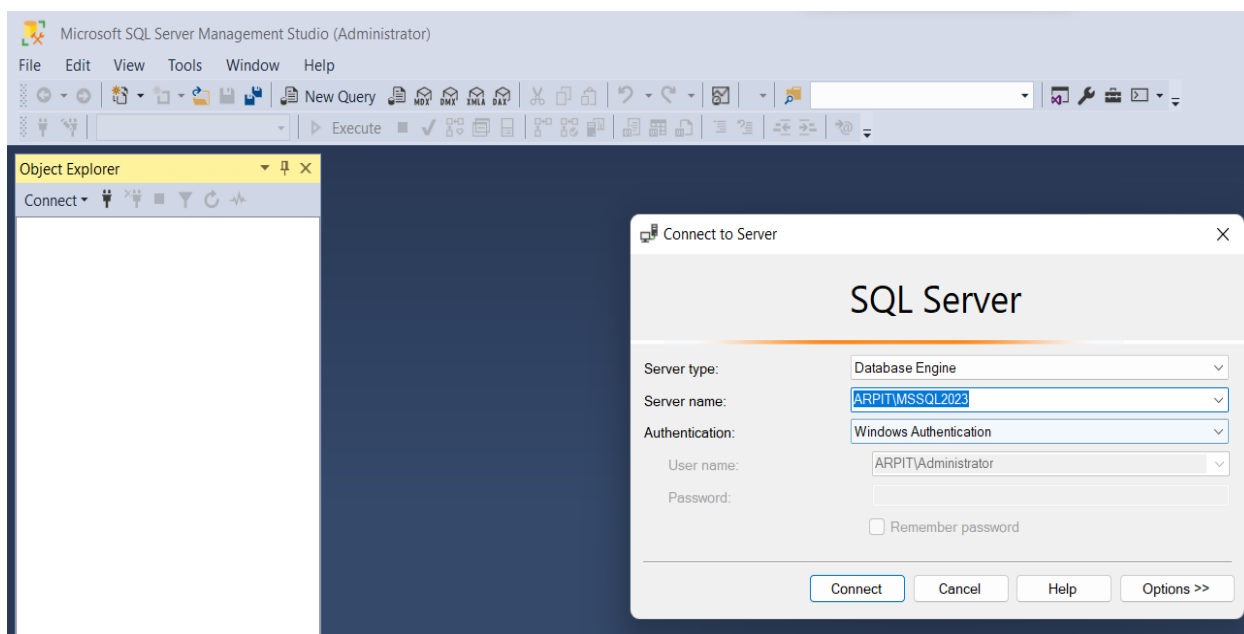
Create a database name **EmployeeDetails**, in which the employee table holds the data of all employees.

Write a SQL server Query to find out the following:

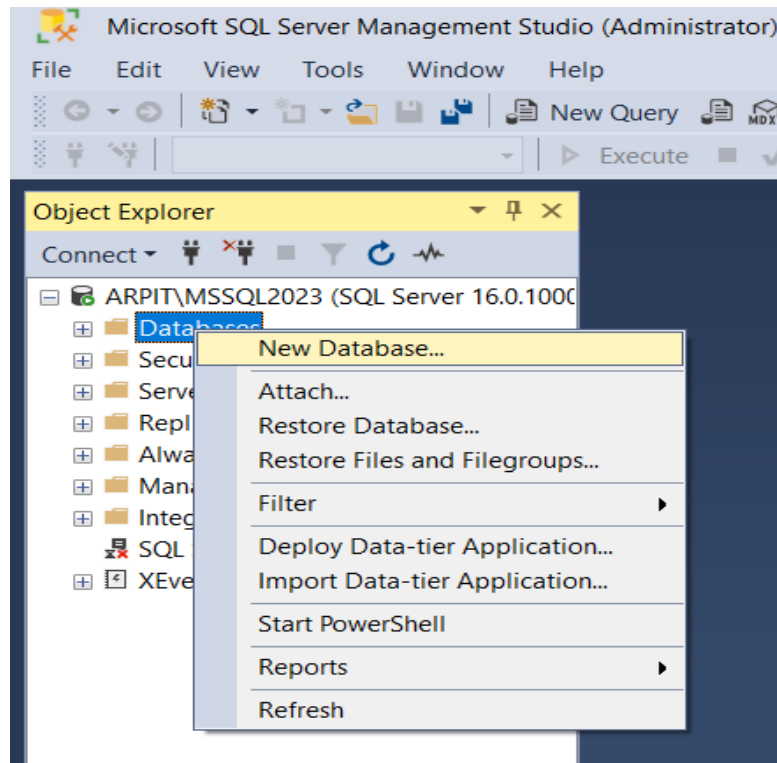
- Create a EmployeeDetails table and insert the data in same table.
- Find the employee from location "guntur" and their salary should be less than 40000.
- Find the employee from location "chennai" and their Designation should be AEO.
- Find the employee either from location "guntur" or their salary should be less than 40000.
- Find the employee either from location "CA" or their salary should be less than 20000.
- Find the employees whose **location** starts with character 'c' and followed by any string of characters.
- Find the employee who is not living in the bangalore, chennai, guntur.

### Working on the Demo

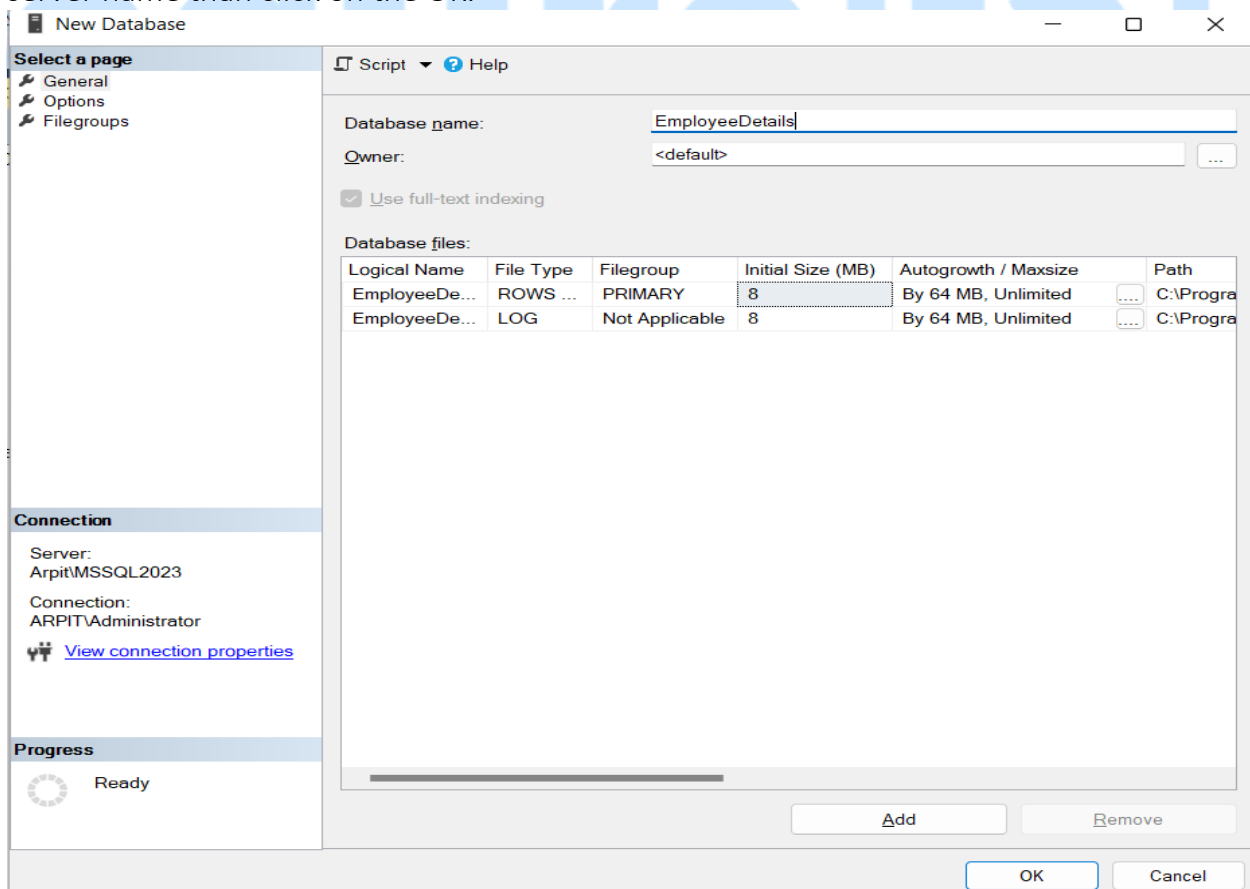
**Step 1:** Connect to the SQL Server using the Windows Authentication credential and selecting the server's name and server type.



**Step 2:** Right click on the Database and select the New Databases.



**Step 3:** Give a database name as **EmployeeDetails** and check the connection and server name then click on the OK.



**Step 4:** Now we will create a **EmployeeDetails** table and insert the data in same table by using the following query

```
create table EmployeeDetails(empid int,  
    empname varchar(50),designation varchar(50),salary int,  
    Location varchar(50))  
  
insert into EmployeeDetails  
values(1,'suresh','software engineer',25000,'chennai'),  
(2,'rohini','AEO',15000,'chennai'),  
(3,'madhavsai','business analyst',50000,'nagpur'),  
(4,'mahendra','CA',75000,'guntur'),  
(5,'sateesh','Doctor',65000,'guntur')  
  
select * from EmployeeDetails
```

When we run the above SQL script, the “EmployeeDetails” table will create, and the result is like below.

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the Object Explorer displays the database structure for 'ARPIT\MSSQL2023 (SQL Server 16.0.1000)'. The 'EmployeeDetails' table is visible under the 'Databases' folder. The main window shows the execution of the SQL script. The script includes a 'create table' statement, an 'insert into' statement with five rows of data, and a 'select \* from EmployeeDetails' statement. The results pane at the bottom displays the data inserted into the table.

	empid	empname	designation	salary	Location
1	1	suresh	software engineer	25000	chennai
2	2	rohini	AEO	15000	chennai
3	3	madhavsai	business analyst	50000	nagpur
4	4	mahendra	CA	75000	guntur
5	5	sateesh	Doctor	65000	guntur

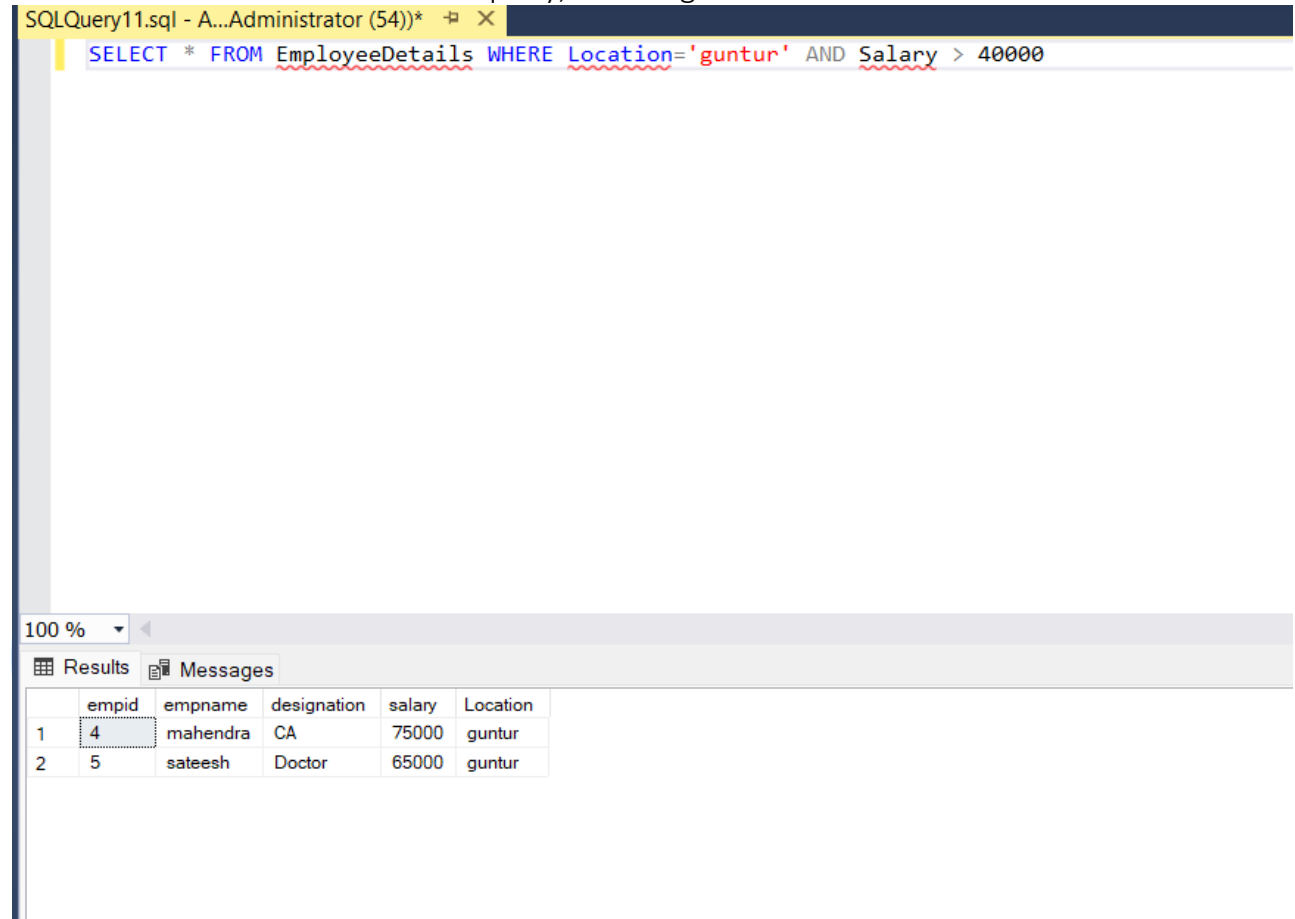
**Step 5:** We check multiple conditions **Location, Salary** with **AND** operator. It will return records that satisfy both conditions.

In order to find the employee from location “Guntur” and their salary should be less

than 40000. We will run the logical operator in the query.

```
SELECT * FROM EmployeeDetails WHERE Location='guntur' AND  
Salary > 40000
```

When we execute the above SQL query, we will get the result below:



The screenshot shows a SQL query editor window titled "SQLQuery11.sql - A...Administrator (54))\*". The query entered is: `SELECT * FROM EmployeeDetails WHERE Location='guntur' AND Salary > 40000`. Below the query editor, the "Results" tab is active, displaying a table with 2 rows and 6 columns: empid, empname, designation, salary, and Location. The first row shows empid 4, empname mahendra, designation CA, salary 75000, and Location guntur. The second row shows empid 5, empname sateesh, designation Doctor, salary 65000, and Location guntur.

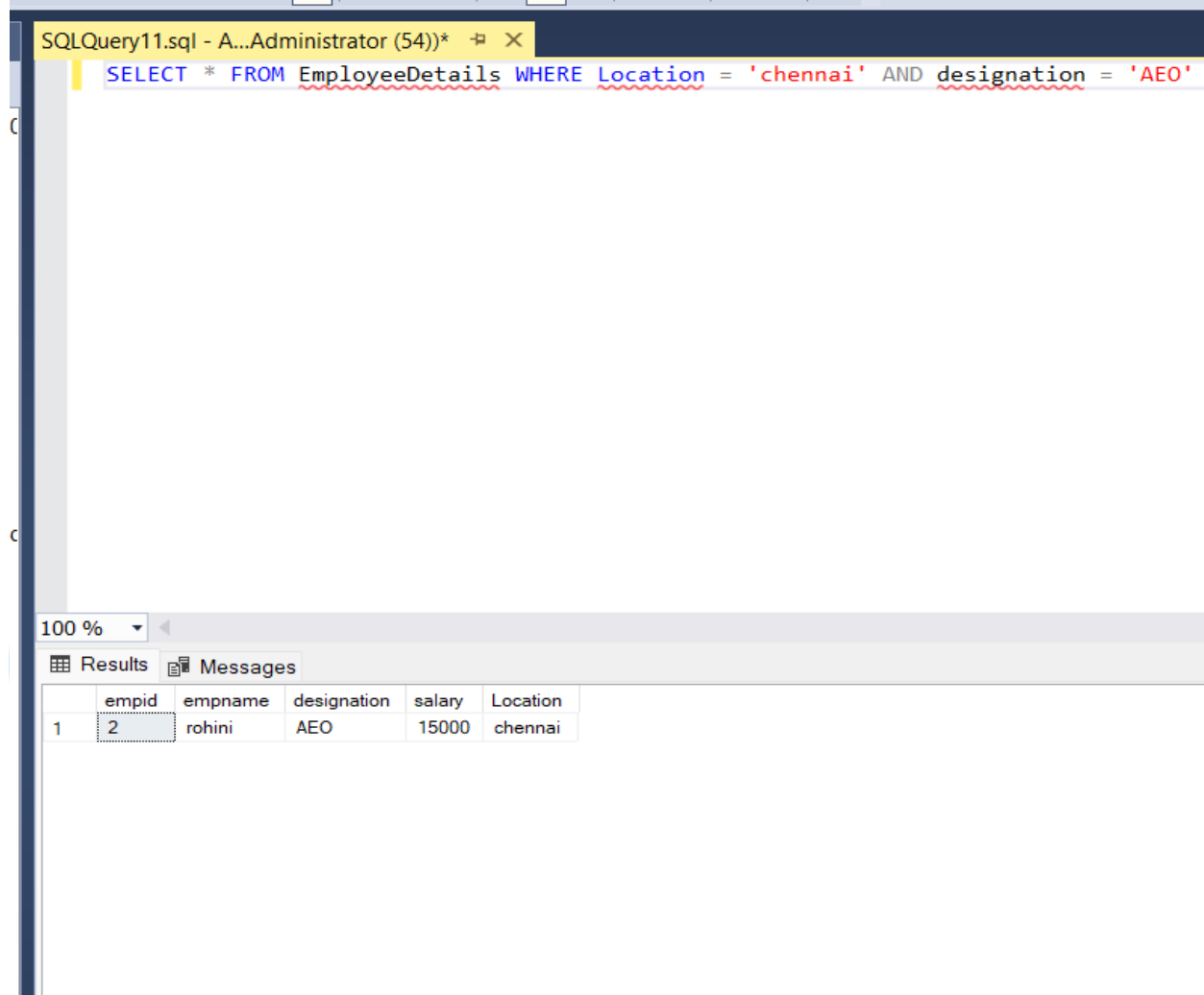
	empid	empname	designation	salary	Location
1	4	mahendra	CA	75000	guntur
2	5	sateesh	Doctor	65000	guntur

**Step 6:** We check multiple conditions **Location**, **Designation** with **AND** operator. It will return records that satisfy both conditions.

In order to find the employee from location "chennai" and their Designation should be AEO. We will run the logical operator in the query.

```
SELECT * FROM EmployeeDetails WHERE Location  
= 'chennai' AND designation = 'AEO'
```

When we execute the above SQL query, we will get the result below:



The screenshot shows a SQL query editor window titled "SQLQuery11.sql - A...Administrator (54)\*". The query entered is: `SELECT * FROM EmployeeDetails WHERE Location = 'chennai' AND designation = 'AEO'`. Below the query editor, the "Results" tab is active, displaying a single row of data. The results are as follows:

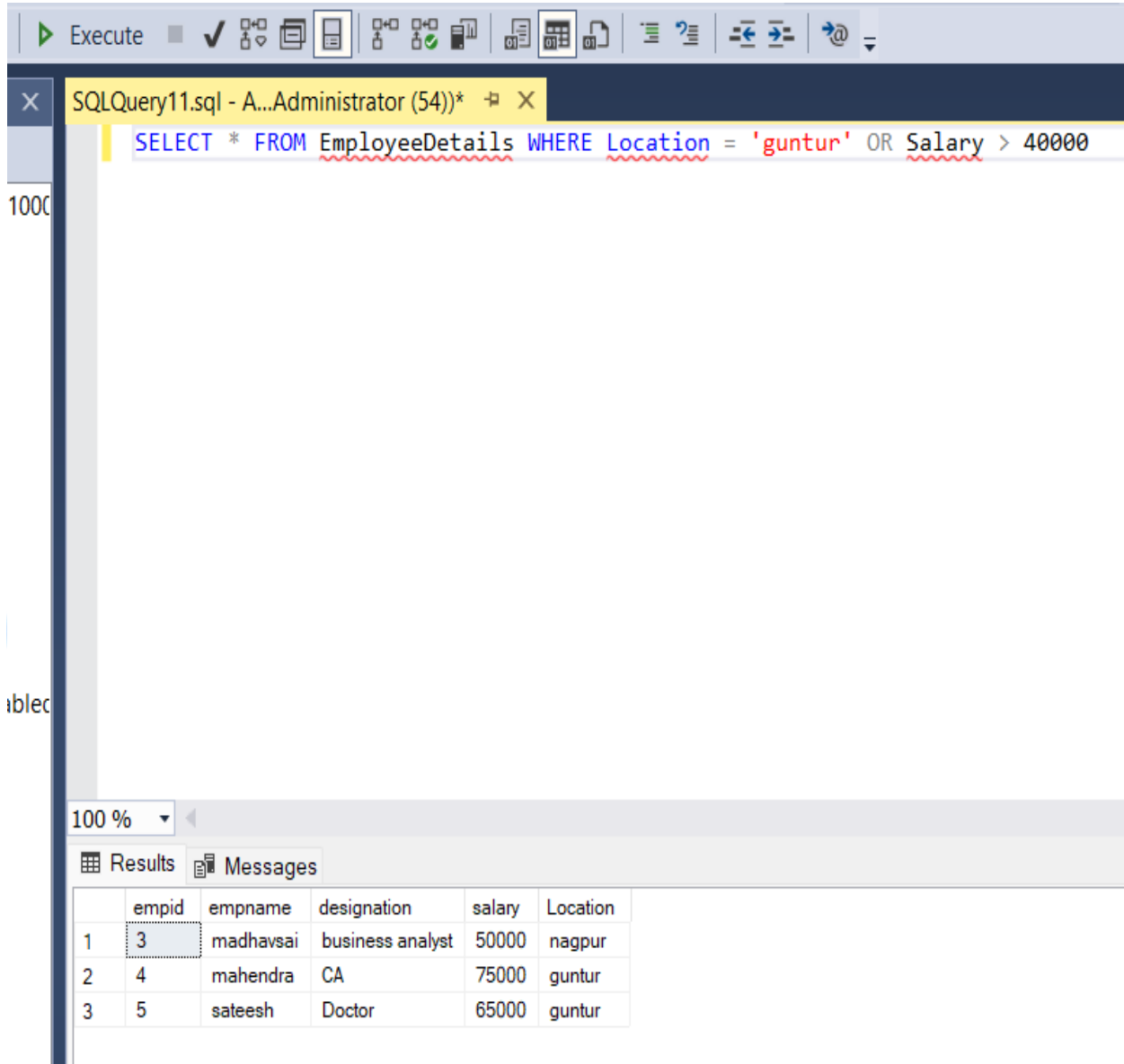
	empid	empname	designation	salary	Location
1	2	rohini	AEO	15000	chennai

**Step 7:** We check multiple conditions **Location, Salary** with **OR** operator. It will return records that satisfy both conditions.

In order to find the employee either from location "Guntur" or their salary should be less than 40000. We will run the logical operator in the query.

```
SELECT * FROM EmployeeDetails WHERE Location='guntur' OR  
Salary > 40000
```

When we execute the above SQL query, we will get the result below:



The screenshot shows a SQL query editor with the query: `SELECT * FROM EmployeeDetails WHERE Location = 'guntur' OR Salary > 40000`. The query is executed, and the results are displayed in a table. The table has 6 columns: empid, empname, designation, salary, and Location. The results are as follows:

	empid	empname	designation	salary	Location
1	3	madhavsai	business analyst	50000	nagpur
2	4	mahendra	CA	75000	guntur
3	5	sateesh	Doctor	65000	guntur

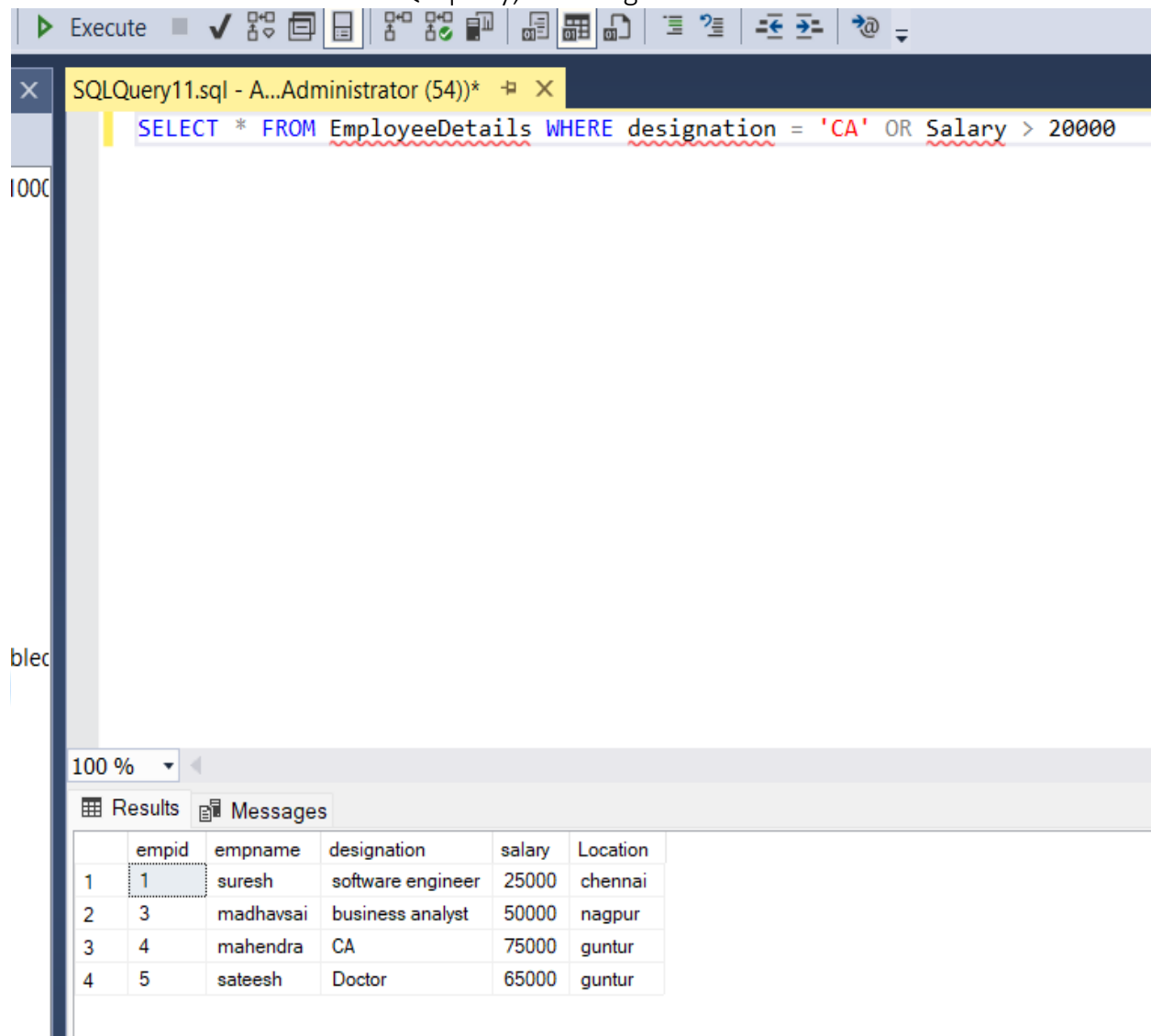
**Step 8:** We check multiple conditions **Designation, Salary** with **OR** operator. It will return records that satisfy both conditions.

In order to find the employee either from location “CA” or their salary should be less than 20000. We will run the logical operator in the query.

```
SELECT * FROM EmployeeDetails WHERE Designation='CA' OR S
```

```
alary > 20000
```

When we execute the above SQL query, we will get the result below:



The screenshot shows a SQL query window with the following query:

```
SELECT * FROM EmployeeDetails WHERE designation = 'CA' OR Salary > 20000
```

The Results pane displays the following data:

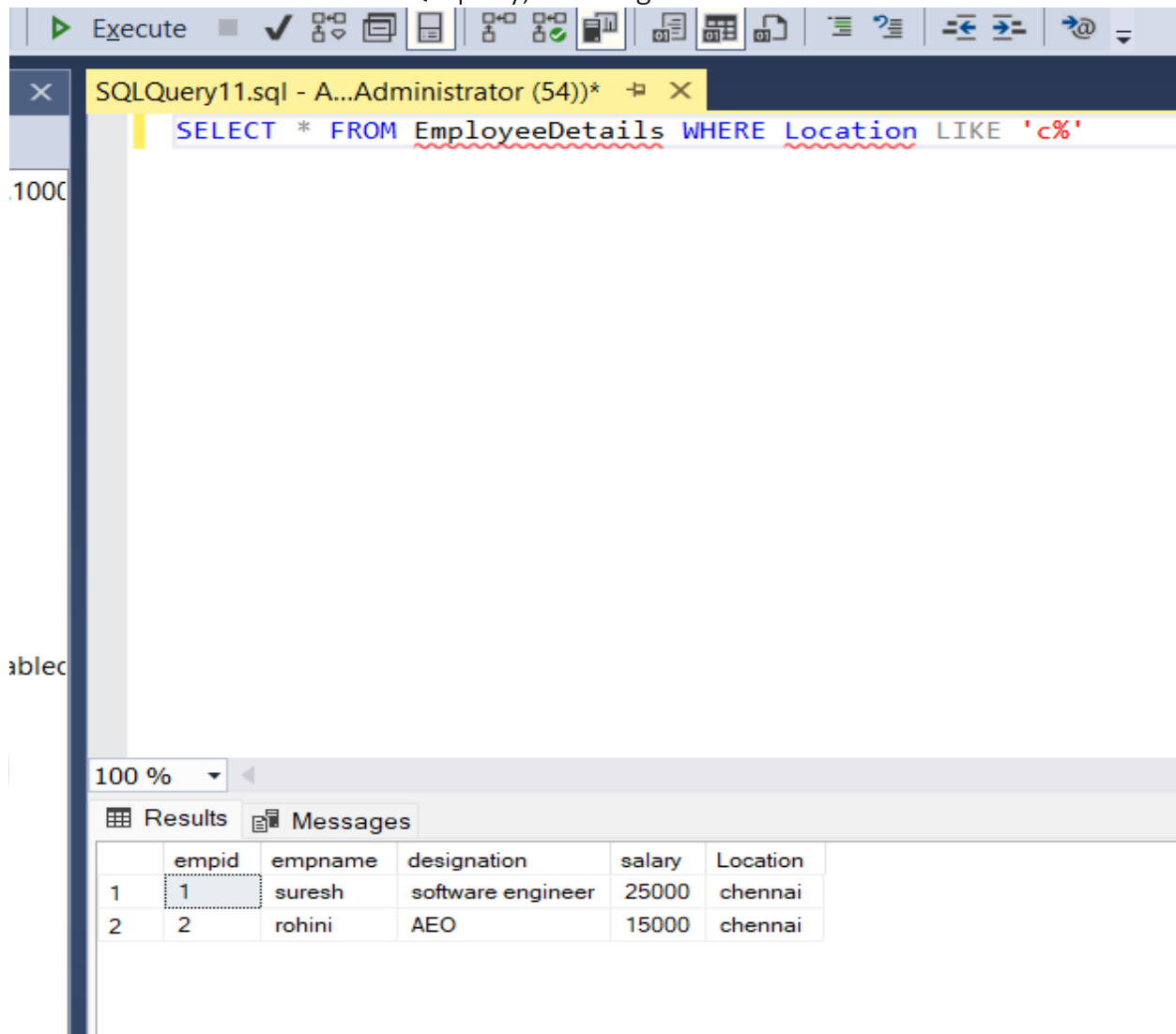
	empid	empname	designation	salary	Location
1	1	suresh	software engineer	25000	chennai
2	3	madhavsai	business analyst	50000	nagpur
3	4	mahendra	CA	75000	guntur
4	5	sateesh	Doctor	65000	guntur

**Step 9:** The following SQL like query will return all employees whose **location** starts with character 'c' and followed by any string of characters because we mentioned a pattern like 'c%'.

```
SELECT * FROM EmployeeDetails WHERE Location LIKE 'c%'
```



When we execute the above SQL query, we will get the result below:



The screenshot shows the SQL Developer interface. The query editor at the top contains the following SQL query:

```
SELECT * FROM EmployeeDetails WHERE Location LIKE 'c%'
```

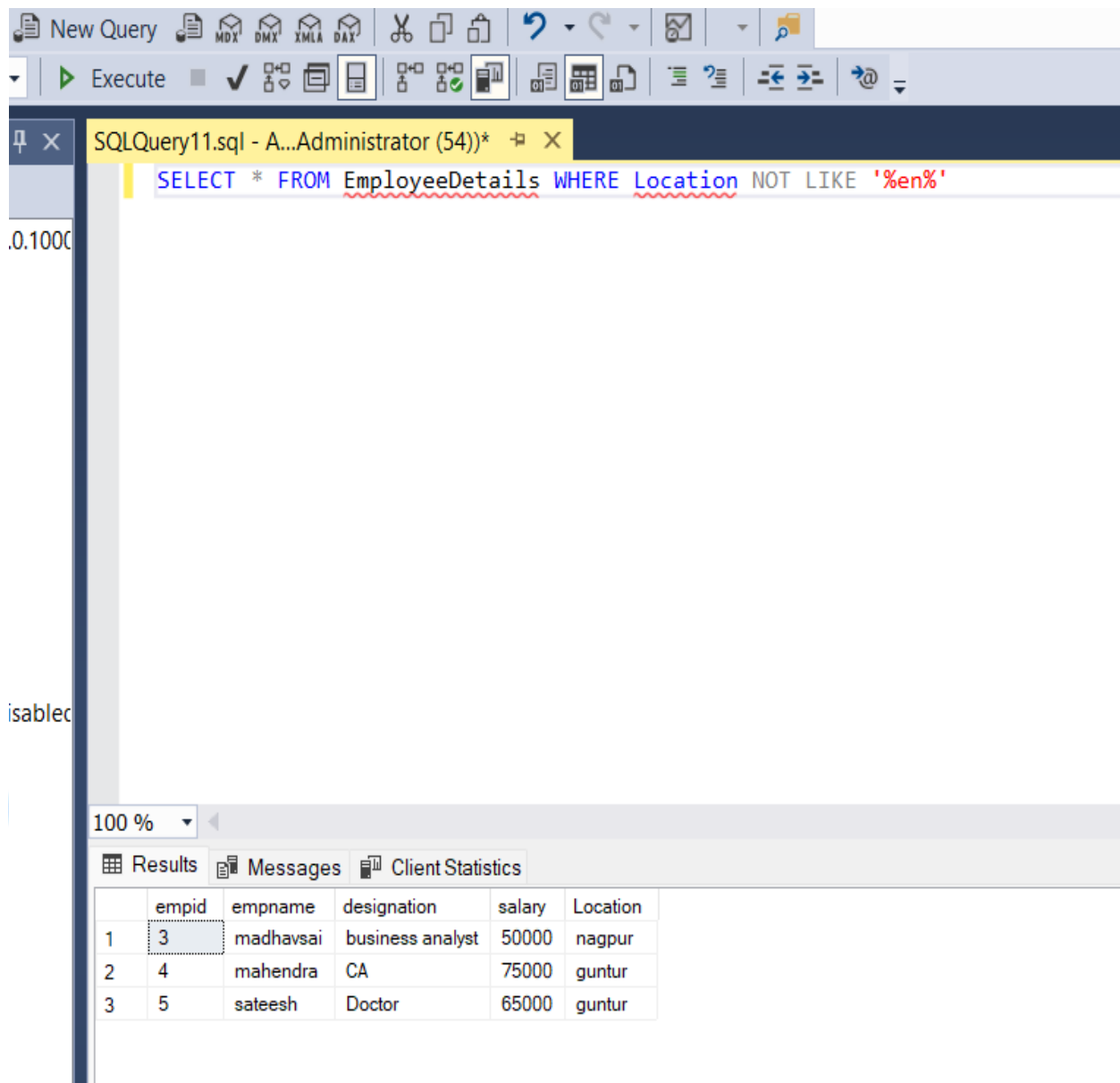
Below the query editor, the 'Results' tab is active, displaying the following data:

	empid	empname	designation	salary	Location
1	1	suresh	software engineer	25000	chennai
2	2	rohini	AEO	15000	chennai

**Step 10:** The following SQL statement will return all the employees whose **location** does not contain a word called 'en', anywhere within the **location** column because we used a **NOT** keyword with **LIKE** operator and mentioned a pattern like '%en%'.

```
SELECT * FROM EmployeeDetails WHERE Location NOT LIKE '%en%'
```

When we execute the above SQL query, we will get the result below:



The screenshot shows the SQL Server Enterprise Manager interface. The query editor displays the following SQL statement:

```
SELECT * FROM EmployeeDetails WHERE Location NOT LIKE '%en%'
```

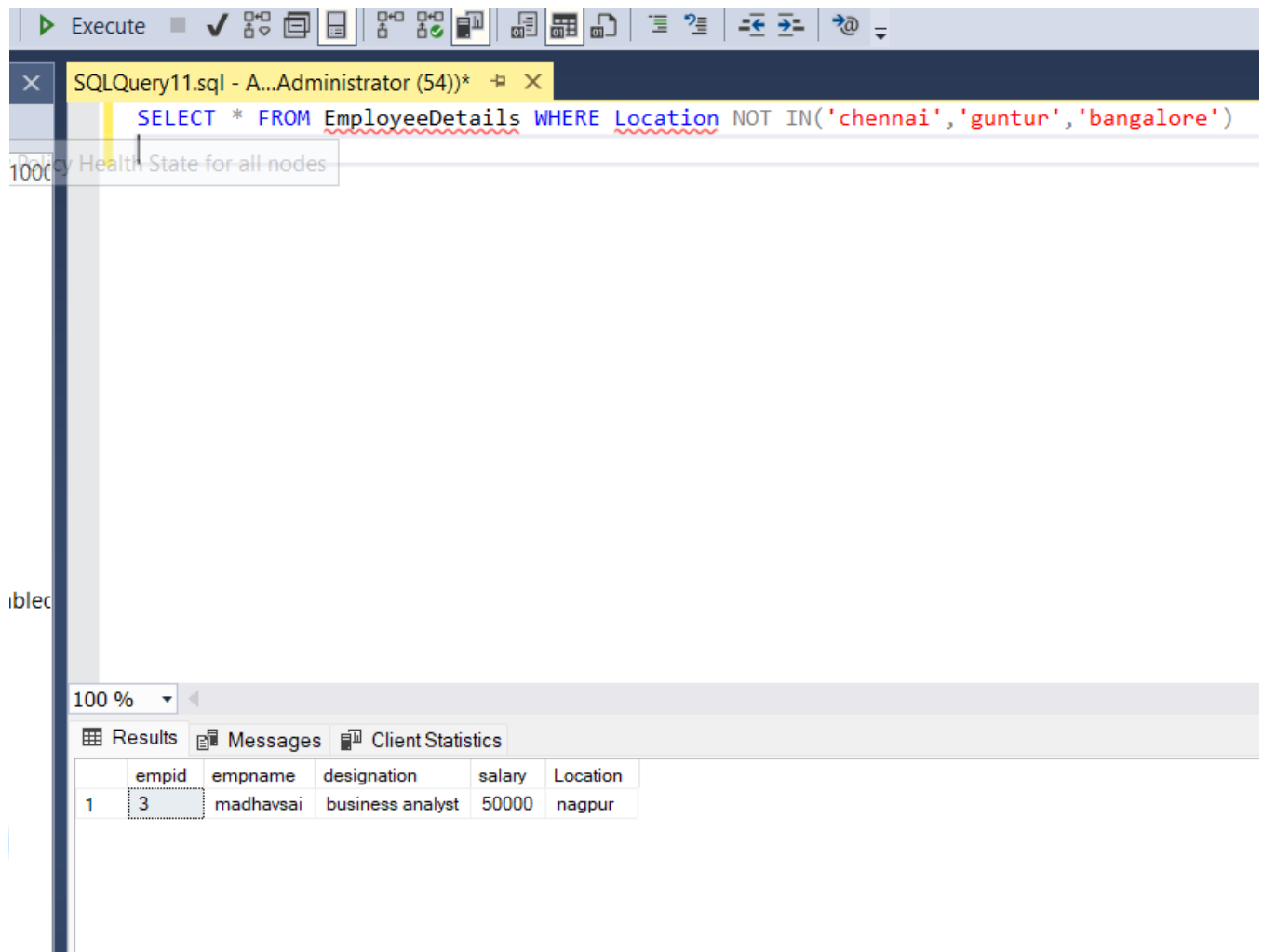
The query is executed, and the results are displayed in the Results pane. The results are as follows:

	empid	empname	designation	salary	Location
1	3	madhavsai	business analyst	50000	nagpur
2	4	mahendra	CA	75000	guntur
3	5	sateesh	Doctor	65000	guntur

**Step 11:** The following SQL statement will return all the employees whose **location** not mentioned in the values.

```
SELECT * FROM EmployeeDetails WHERE Location NOT IN  
('chennai', 'guntur', 'bangalore')
```

When we execute the above SQL query, we will get the result below:



The screenshot shows a SQL query execution window. The query is: `SELECT * FROM EmployeeDetails WHERE Location NOT IN('chennai', 'guntur', 'bangalore')`. The results are displayed in a table with the following data:

	empid	empname	designation	salary	Location
1	3	madhavsai	business analyst	50000	nagpur

*a Veranda Enterprise*