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

Demo 2 – Composing T-SQL

Composing T-SQL

Problem Statement: A survey is conducted by the Lions club company for all the employees. They want all employees' details so that they can plan expansion and further hiring process.

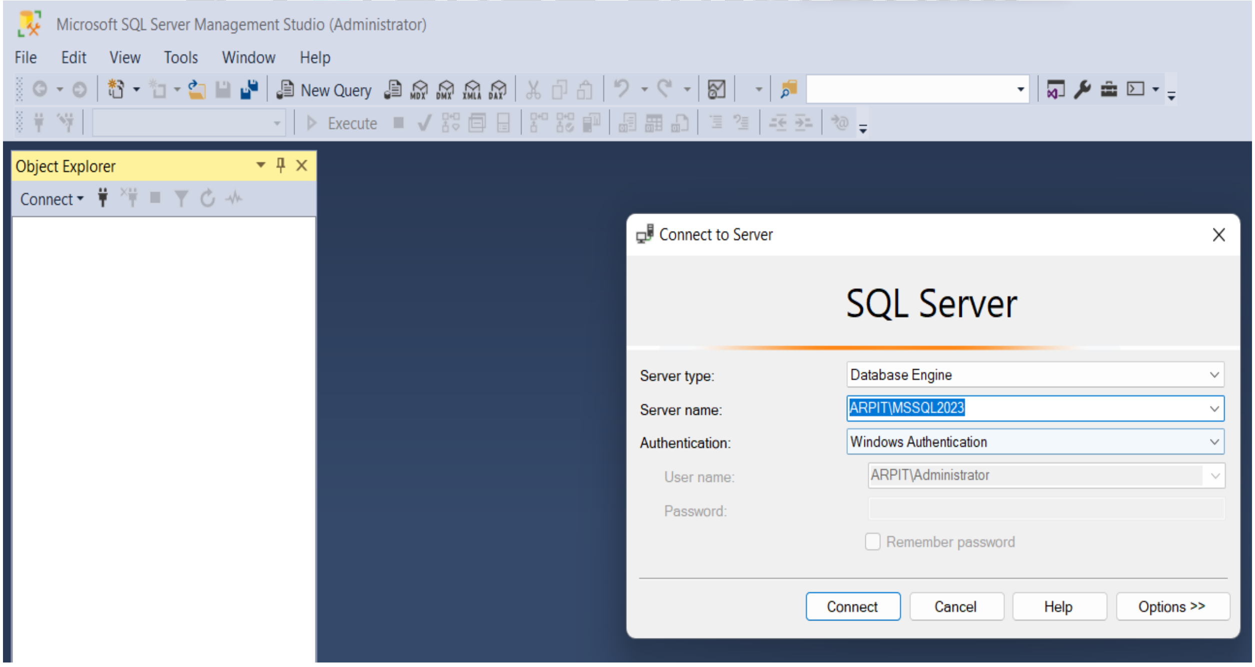
Create a database name **Employees**, in which the employee table holds the data of all employees.

Write a SQL server Query to find out the following:

- Create an employee table and insert the data in same table.
- Calculate the total number of employees listed in the given table.
- Find the total number of employees whose salary is less than 50000 and working hours greater than 9.
- Find the lowest salary of an employee stored in the employee table.
- Find the highest salary of an employee stored in the employee table.
- Create another table named "emp_address" that stores the address of each employee. It should contain name, cellphone, address, city values.
- Calculate the total number of employee and their addresses from two different tables.

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.



Server type: Database Engine

Server name: ARPIT\MSSQL2023

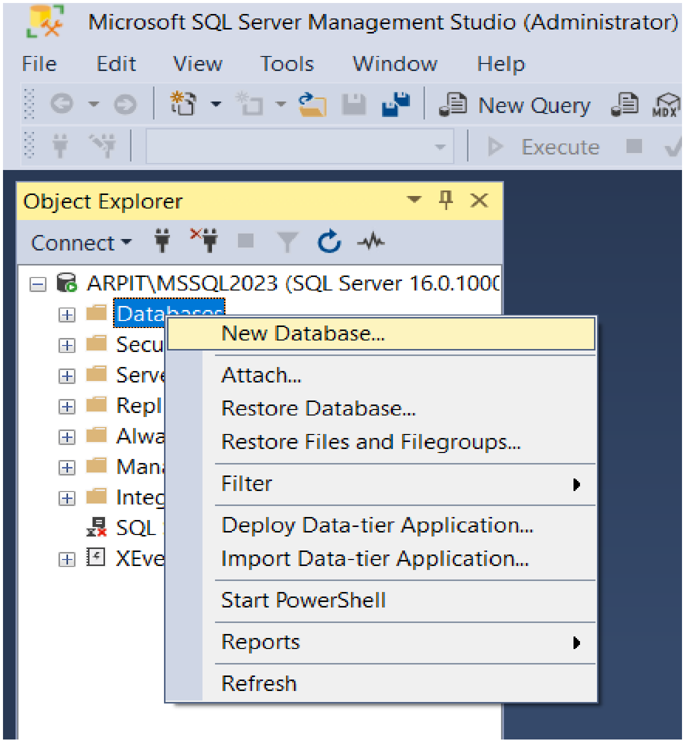
Authentication: Windows Authentication

User name: ARPIT\Administrator

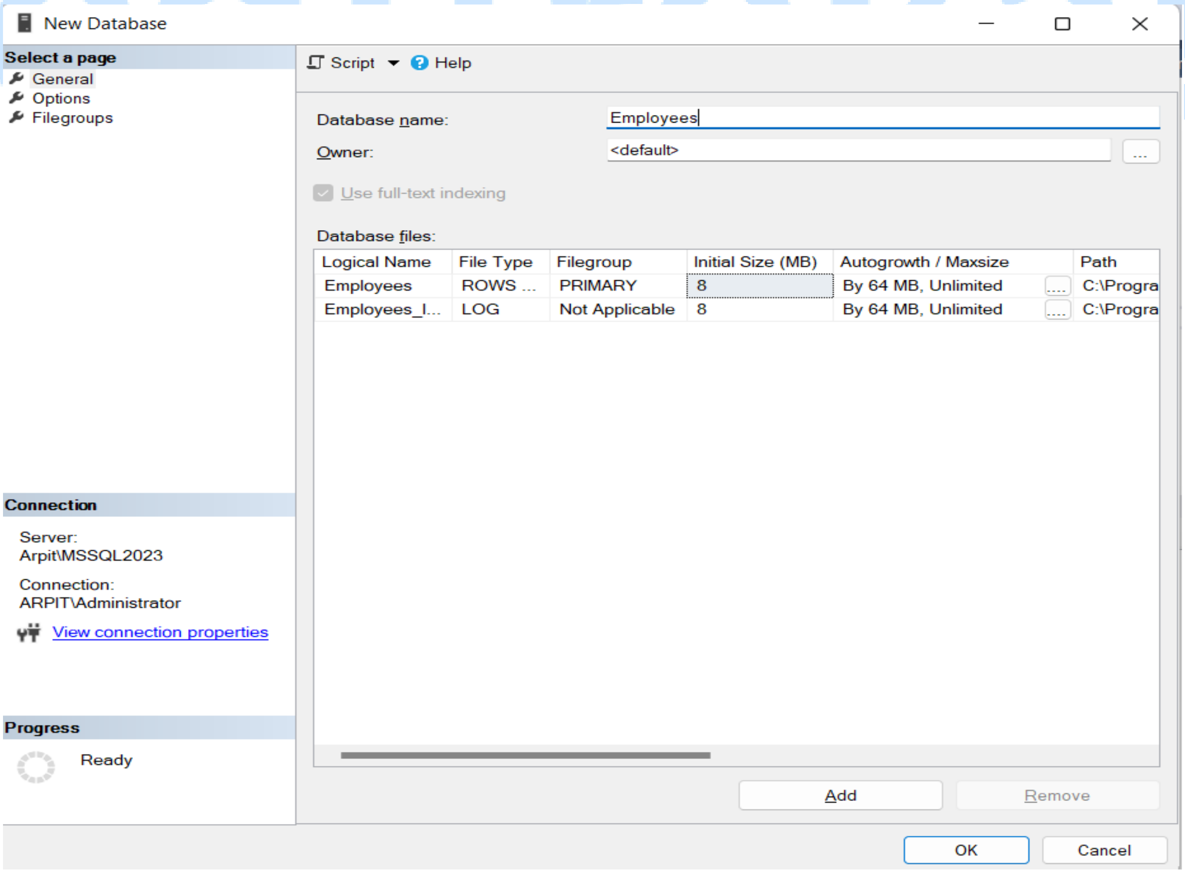
Password:

□ Remember password

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


Microsoft SQL Server Management Studio (Administrator)
File Edit View Tools Window Help
New Query
Execute
Object Explorer
Connect
ARPIT\MSSQL2023 (SQL Server 16.0.100(
Databases
Secu
Serv
Repl
Alwa
Man
Integ
SQL
XEve
New Database...
Attach...
Restore Database...
Restore Files and Filegroups...
Filter
Deploy Data-tier Application...
Import Data-tier Application...
Start PowerShell
Reports
Refresh

Step 3: Give a database name an **Employees** and check the connection and server name than click on the OK.



Database name: Employees

Owner: <default>

☑ Use full-text indexing

Database files:

Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth / Maxsize	Path
Employees	ROWS ...	PRIMARY	8	By 64 MB, Unlimited	C:\Progra
Employees_l...	LOG	Not Applicable	8	By 64 MB, Unlimited	C:\Progra

Connection

Server:
Arpit\MSSQL2023

Connection:
ARPIT\Administrator

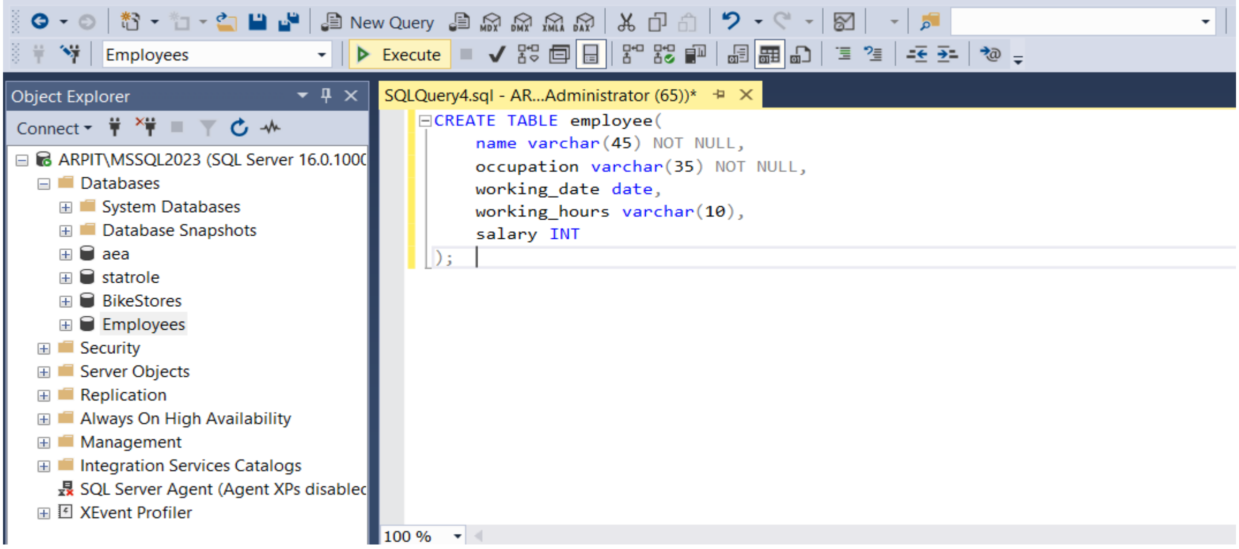
View connection properties

Progress

Ready

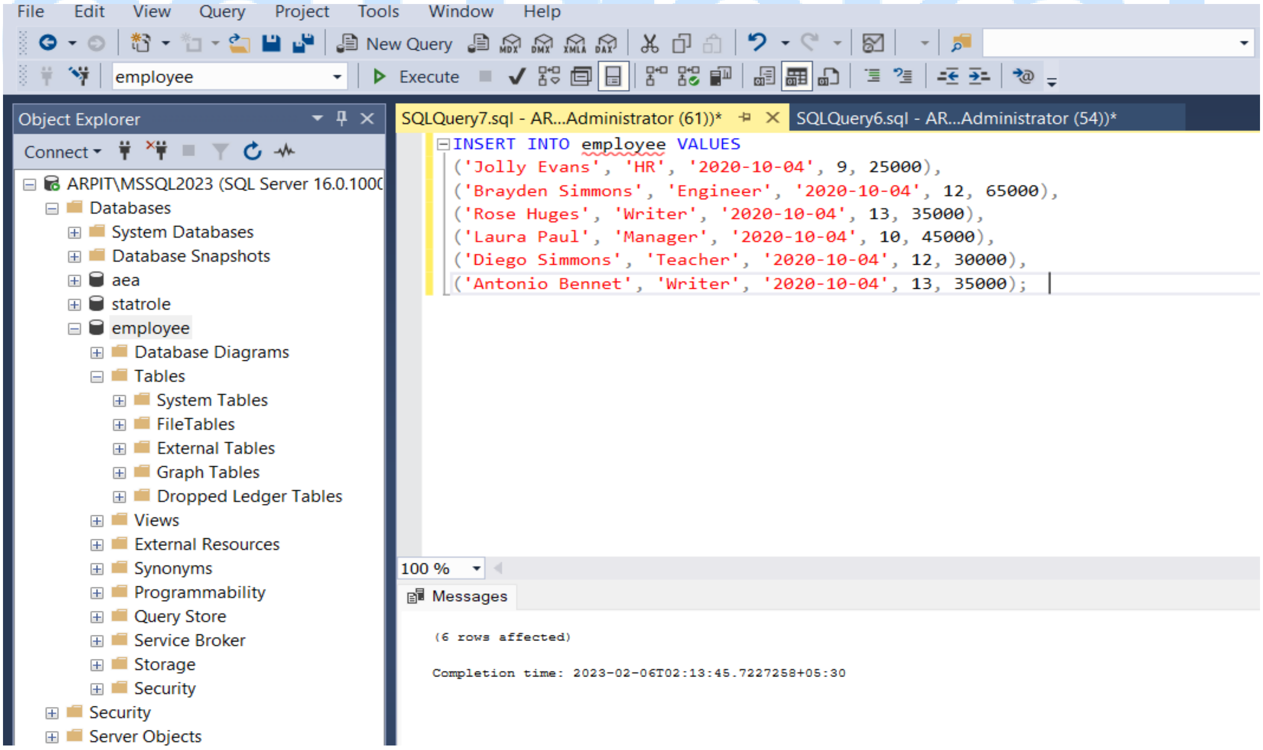
Step 4: Create a table named **employee** and specify the datatypes.

```
Run the query CREATE TABLE employee(
name varchar(45) NOT NULL,
occupation varchar(35) NOT NULL,
working_date date,
working_hours varchar(10),
salary INT  );
```



```
CREATE TABLE employee(
    name varchar(45) NOT NULL,
    occupation varchar(35) NOT NULL,
    working_date date,
    working_hours varchar(10),
    salary INT
);
```

Now, we will insert some data into the table as following.



```
INSERT INTO employee VALUES
('Jolly Evans', 'HR', '2020-10-04', 9, 25000),
('Brayden Simmons', 'Engineer', '2020-10-04', 12, 65000),
('Rose Huges', 'Writer', '2020-10-04', 13, 35000),
('Laura Paul', 'Manager', '2020-10-04', 10, 45000),
('Diego Simmons', 'Teacher', '2020-10-04', 12, 30000),
('Antonio Bennet', 'Writer', '2020-10-04', 13, 35000);
```

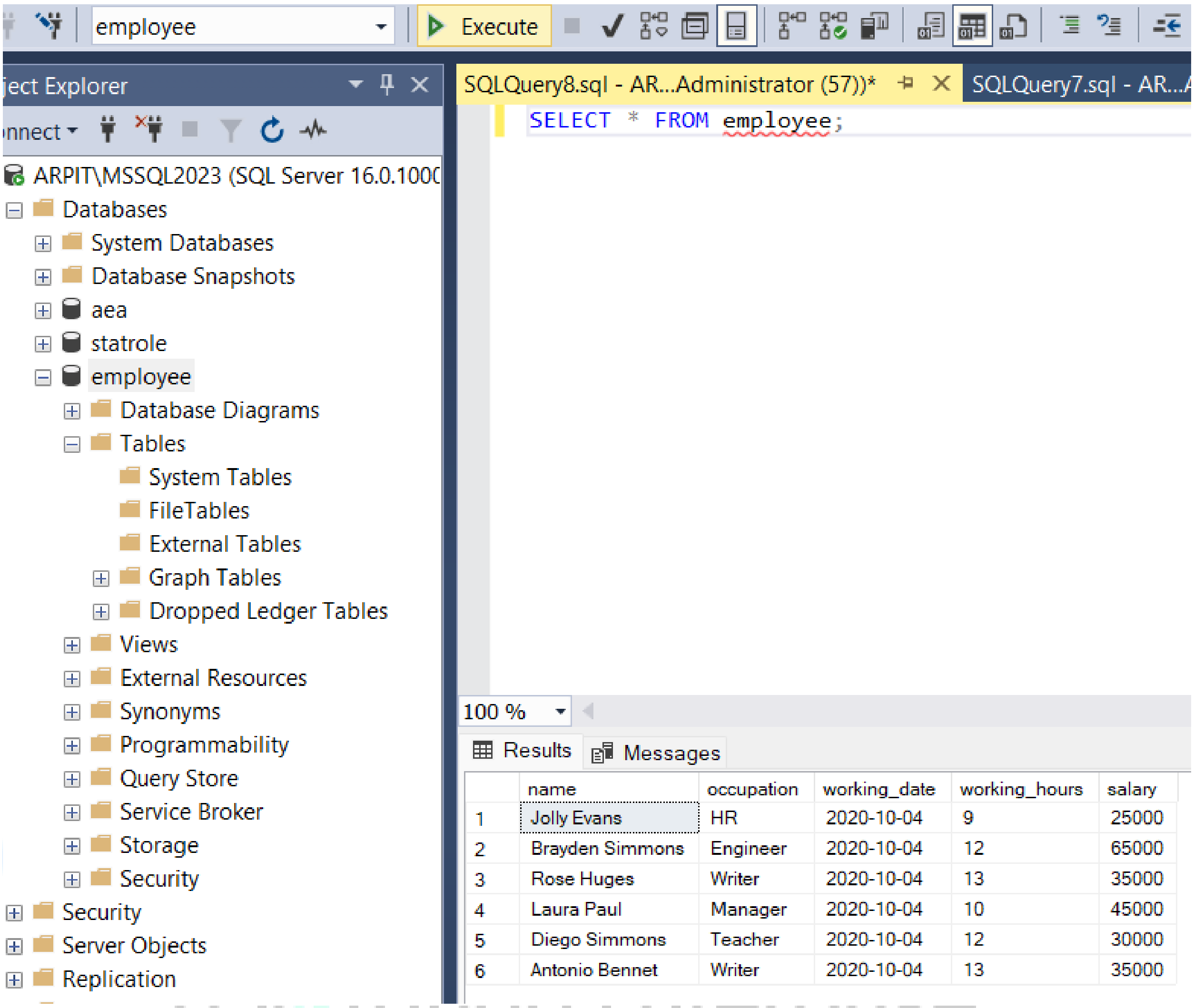
```
(6 rows affected)

Completion time: 2023-02-06T02:13:45.7227258+05:30
```

Step 5: Now, we will check all the inserted data in the employee table.

Run the query:

```
Select * FROM employee;
```

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



```sql
SELECT * FROM employee;
```

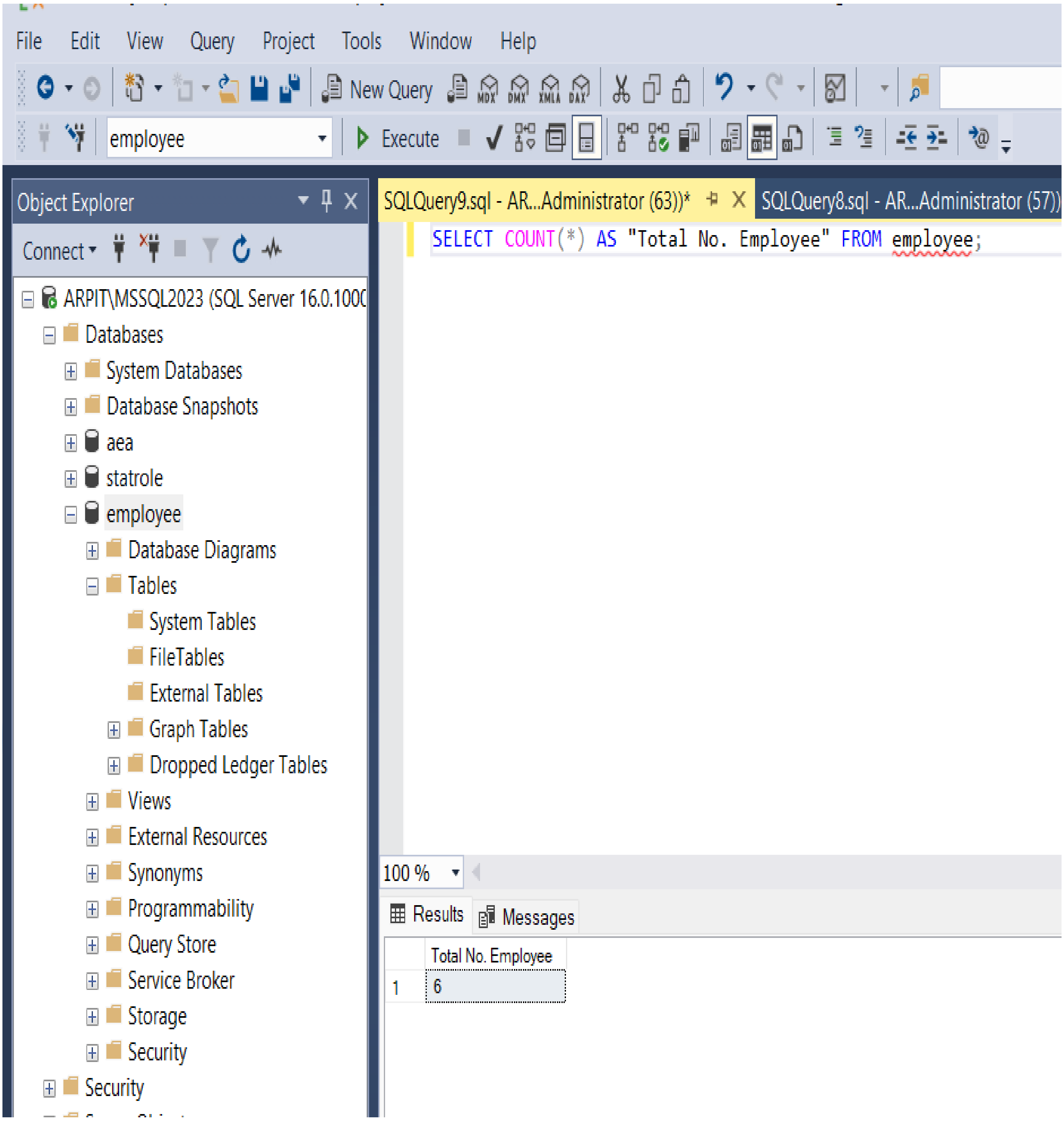
	name	occupation	working_date	working_hours	salary
1	Jolly Evans	HR	2020-10-04	9	25000
2	Brayden Simmons	Engineer	2020-10-04	12	65000
3	Rose Huges	Writer	2020-10-04	13	35000
4	Laura Paul	Manager	2020-10-04	10	45000
5	Diego Simmons	Teacher	2020-10-04	12	30000
6	Antonio Bennet	Writer	2020-10-04	13	35000

Step 6: We want to calculate the total number of employees listed in the given table.

Run the query:

```sql
SELECT COUNT(*) AS Total No.Employee FROM employee;
```

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



```
SELECT COUNT(*) AS "Total No. Employee" FROM employee;
```

	Total No. Employee
1	6

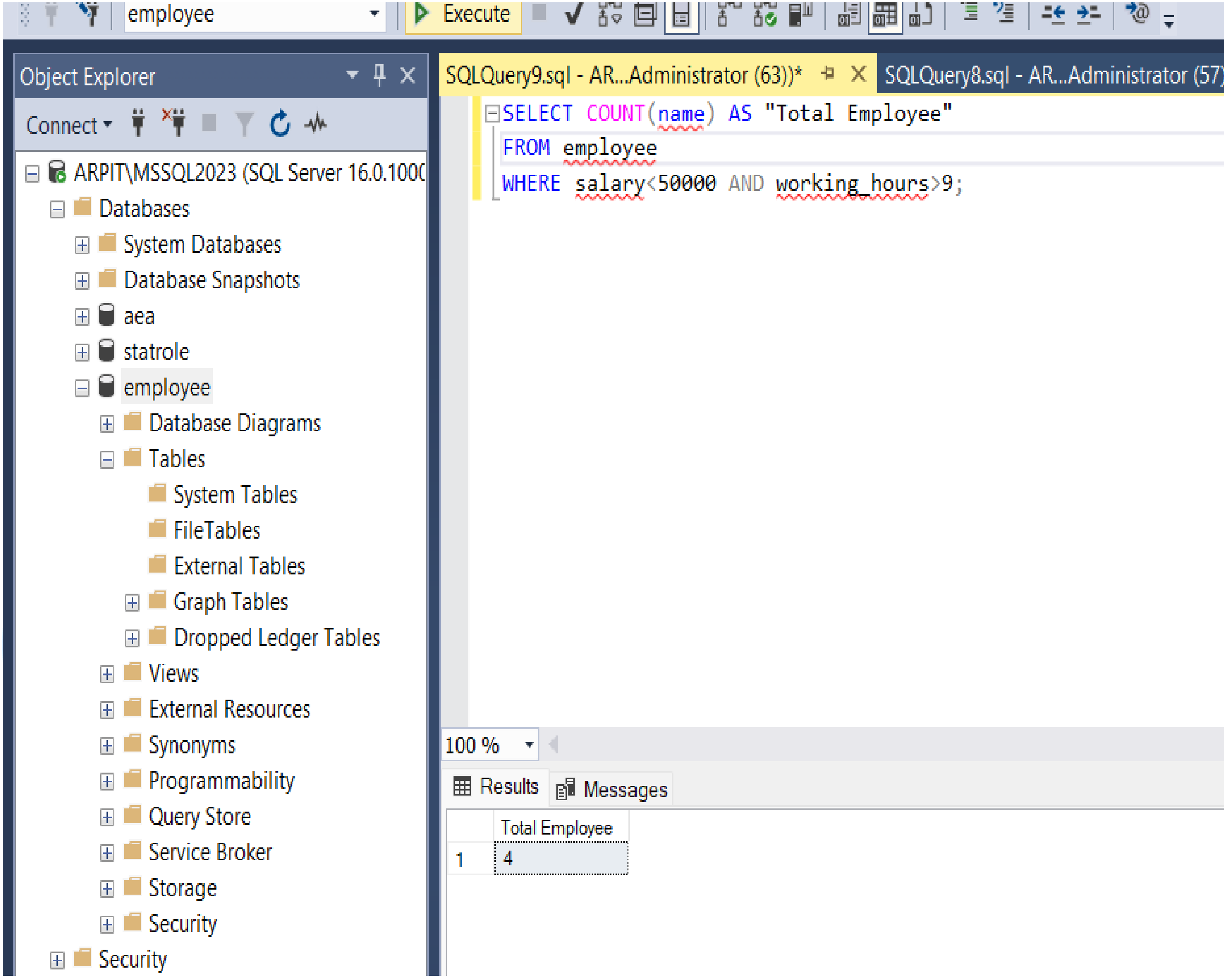
Step 7: Using the Count () function to return, finding the total number of employees whose salary is less than 50000 and working hours greater than 9.

Run the query:

```
SELECT COUNT(name) AS "Total Employee"
FROM employee
```

```
WHERE salary<50000 AND working_hours>9;
```

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



```
SELECT COUNT(name) AS "Total Employee"
FROM employee
WHERE salary<50000 AND working_hours>9;
```

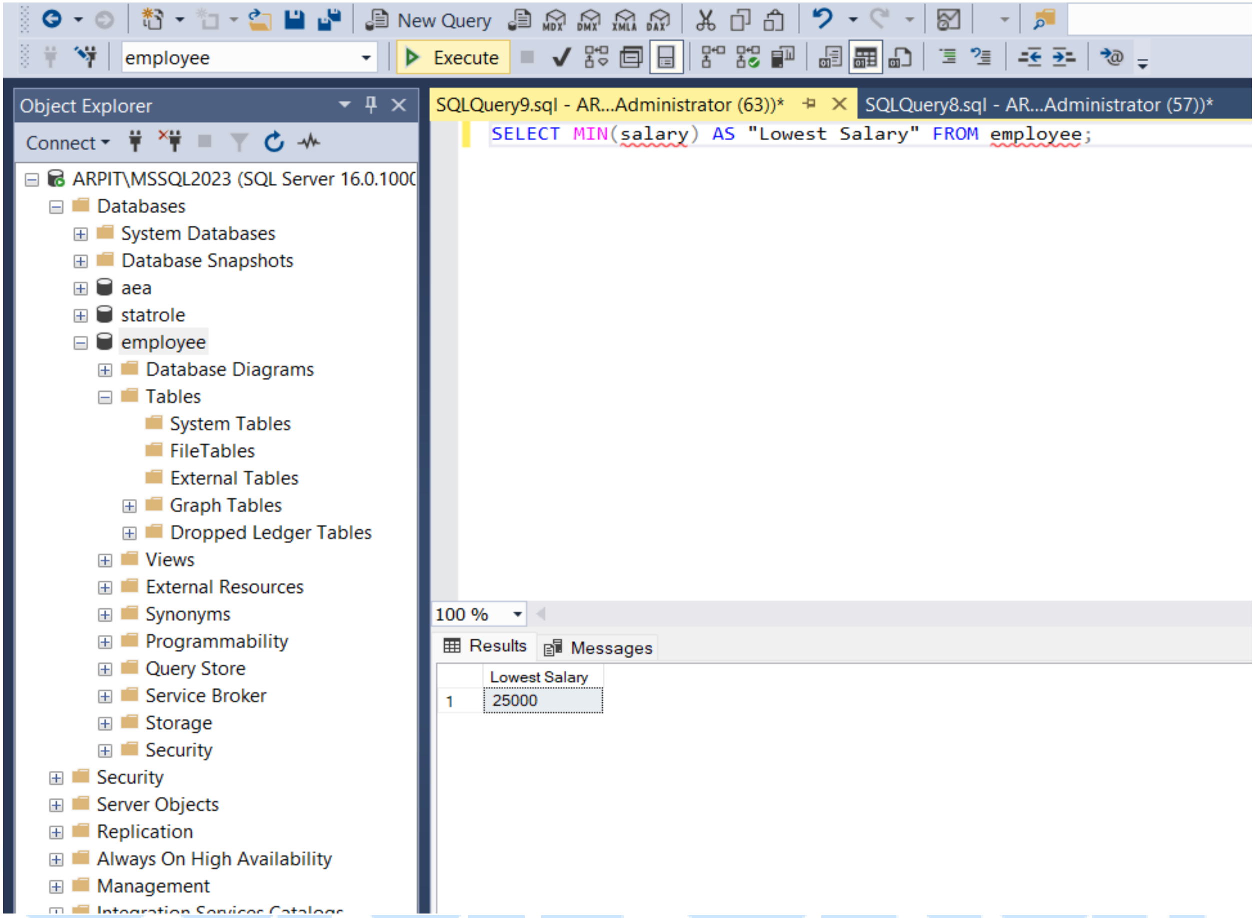
	Total Employee
1	4

Step 8: Now we will find the lowest salary of an employee stored in the employee table.

Run the Query:

```
SELECT MIN(salary) AS "Lowest Salary" FROM employee;
```

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



```
SELECT MIN(salary) AS "Lowest Salary" FROM employee;
```

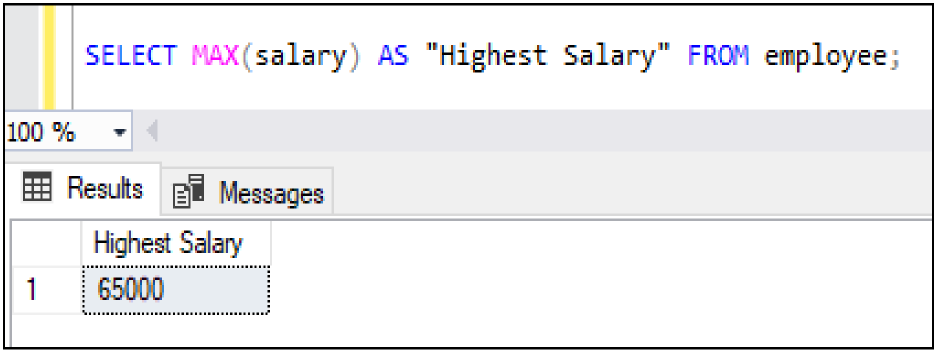
	Lowest Salary
1	25000

Step 9: Now we will find the highest salary of an employee stored in the employee table.

Run the Query:

```
SELECT MAX (salary) AS "Highest Salary" FROM employee;
```

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



```
SELECT MAX(salary) AS "Highest Salary" FROM employee;
```

	Highest Salary
1	65000

Step 10: Here, we will create another table named "**emp_address**" that stores the address of each employee.

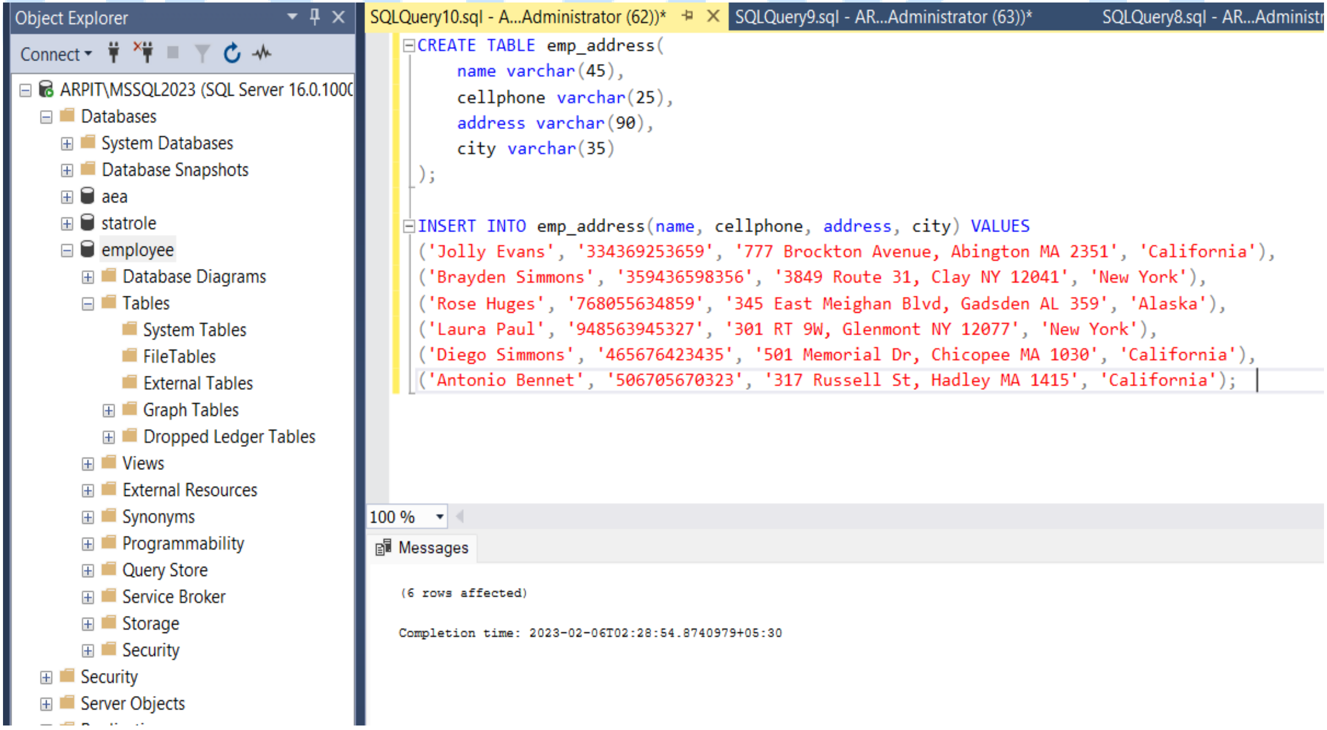
Here is the query to create a table:

```
CREATE TABLE emp_address(
    name varchar(45),
    cellphone varchar(25),
    address varchar(90),
    city varchar(35)  );
```

Insert the value into the new table

```
INSERT INTO emp_address(name, cellphone, address, city)
VALUES
('Jolly Evans', '334369253659', '777 Brockton Avenue,
Abington MA 2351', 'California'),
('Brayden Simmons', '359436598356', '3849 Route 31, Clay
NY 12041', 'New York'),
('Rose Huges', '768055634859', '345 East Meighan Blvd,
Gadsden AL 359', 'Alaska'),
('Laura Paul', '948563945327', '301 RT 9W, Glenmont NY
12077', 'New York'),
('Diego Simmons', '465676423435', '501 Memorial Dr,
Chicopee MA 1030', 'California'),
('Antonio Bennet', '506705670323', '317 Russell St,
Hadley MA 1415', 'California');
```

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



```
CREATE TABLE emp_address(
    name varchar(45),
    cellphone varchar(25),
    address varchar(90),
    city varchar(35)
);

INSERT INTO emp_address(name, cellphone, address, city) VALUES
('Jolly Evans', '334369253659', '777 Brockton Avenue, Abington MA 2351', 'California'),
('Brayden Simmons', '359436598356', '3849 Route 31, Clay NY 12041', 'New York'),
('Rose Huges', '768055634859', '345 East Meighan Blvd, Gadsden AL 359', 'Alaska'),
('Laura Paul', '948563945327', '301 RT 9W, Glenmont NY 12077', 'New York'),
('Diego Simmons', '465676423435', '501 Memorial Dr, Chicopee MA 1030', 'California'),
('Antonio Bennet', '506705670323', '317 Russell St, Hadley MA 1415', 'California');
```

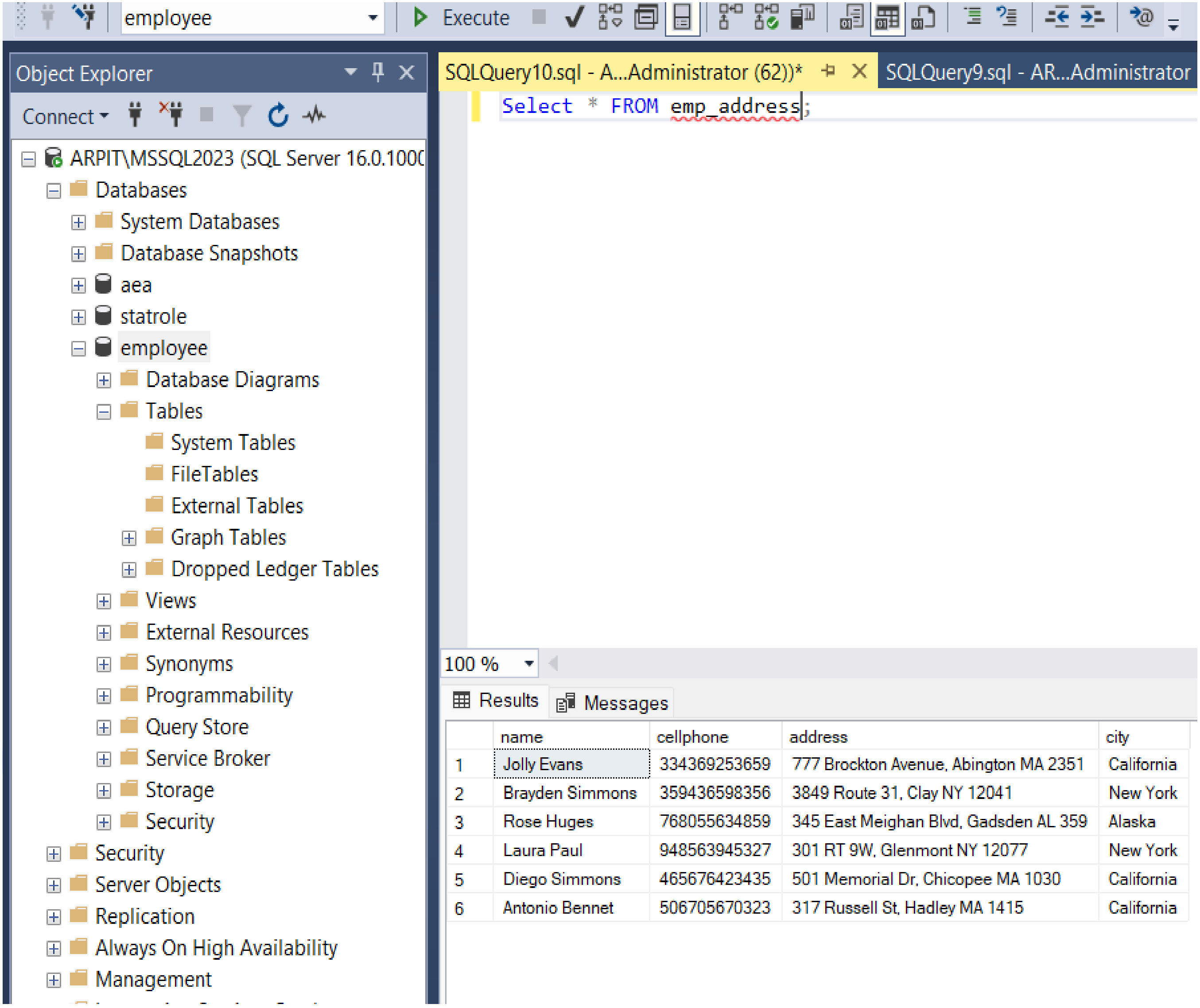
```
(6 rows affected)

Completion time: 2023-02-06T02:28:54.8740979+05:30
```

Step 11: Now, we will check all the inserted data into the employee table.

Run the query:

```
Select * FROM emp_address;
```

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



```
Select * FROM emp_address;
```

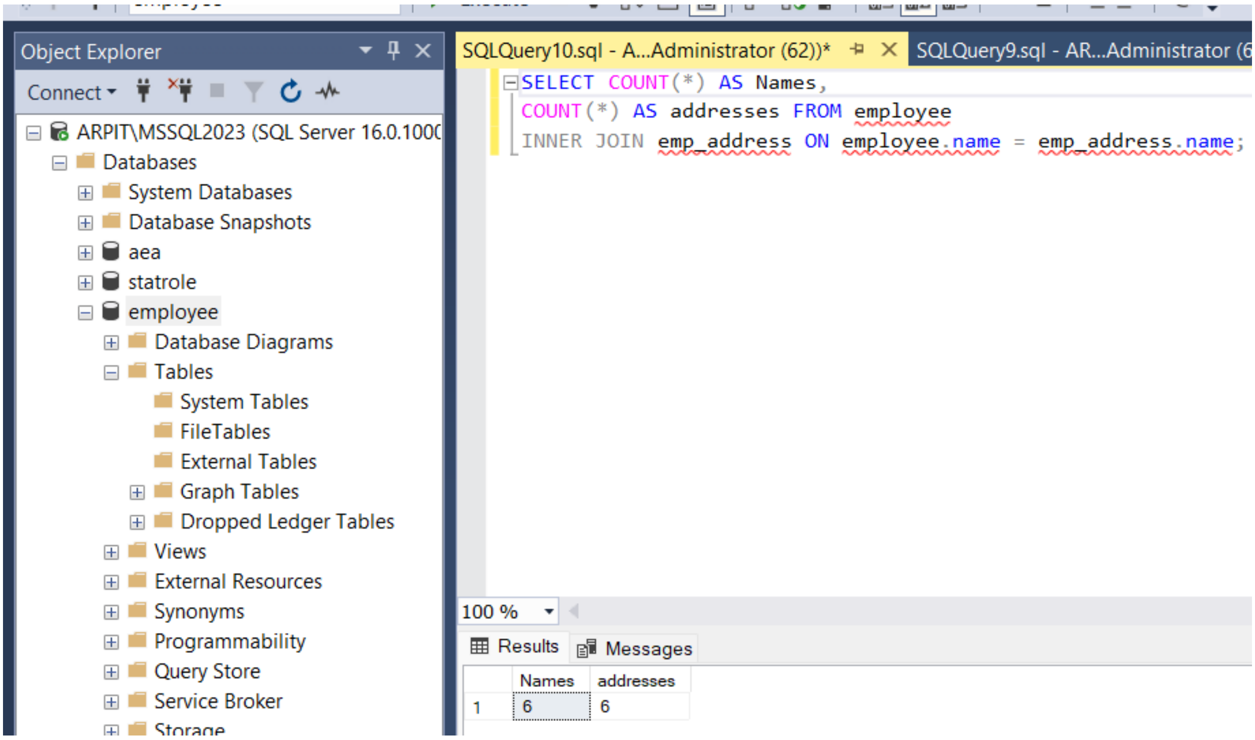
	name	cellphone	address	city
1	Jolly Evans	334369253659	777 Brockton Avenue, Abington MA 2351	California
2	Brayden Simmons	359436598356	3849 Route 31, Clay NY 12041	New York
3	Rose Huges	768055634859	345 East Meighan Blvd, Gadsden AL 359	Alaska
4	Laura Paul	948563945327	301 RT 9W, Glenmont NY 12077	New York
5	Diego Simmons	465676423435	501 Memorial Dr, Chicopee MA 1030	California
6	Antonio Bennet	506705670323	317 Russell St, Hadley MA 1415	California

Step 12: Now we will calculate the total number of employee and their addresses from two different tables.

Run the query:

```
SELECT COUNT(*) AS Names,
COUNT(*) AS addresses FROM employee
INNER JOIN emp_address ON employee.name =
emp_address.name;
```

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



```sql
SELECT COUNT(*) AS Names,
COUNT(*) AS addresses FROM employee
INNER JOIN emp_address ON employee.name = emp_address.name;
```

	Names	addresses
1	6	6