1. Write an SQL statement to create the tables for the Employee Database, including the necessary referential-integrity constraints.

CREATE DATABASE Employee;

USE Employee;

CREATE TABLE employee (employee\_name VARCHAR(100) PRIMARY KEY, street VARCHAR(100), city VARCHAR(100));

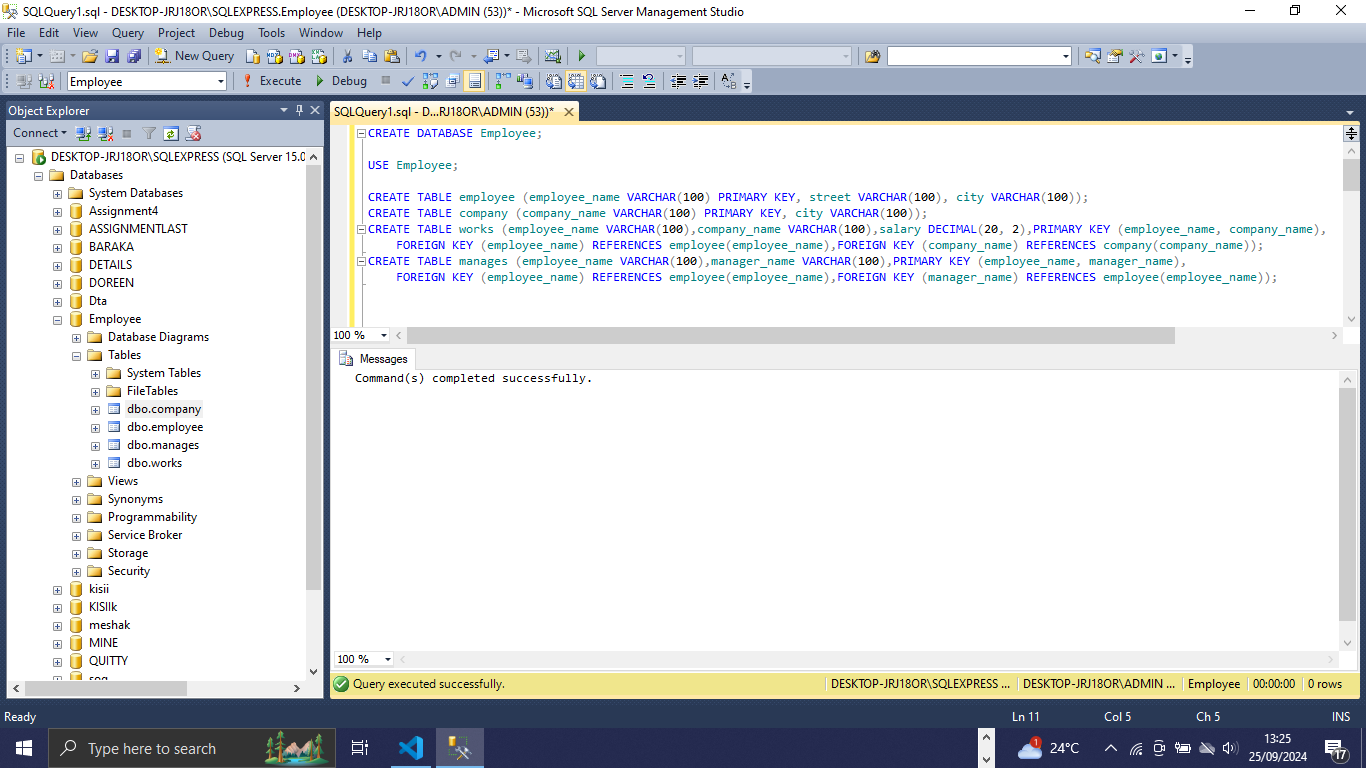
CREATE TABLE company (company\_name VARCHAR(100) PRIMARY KEY, city VARCHAR(100));

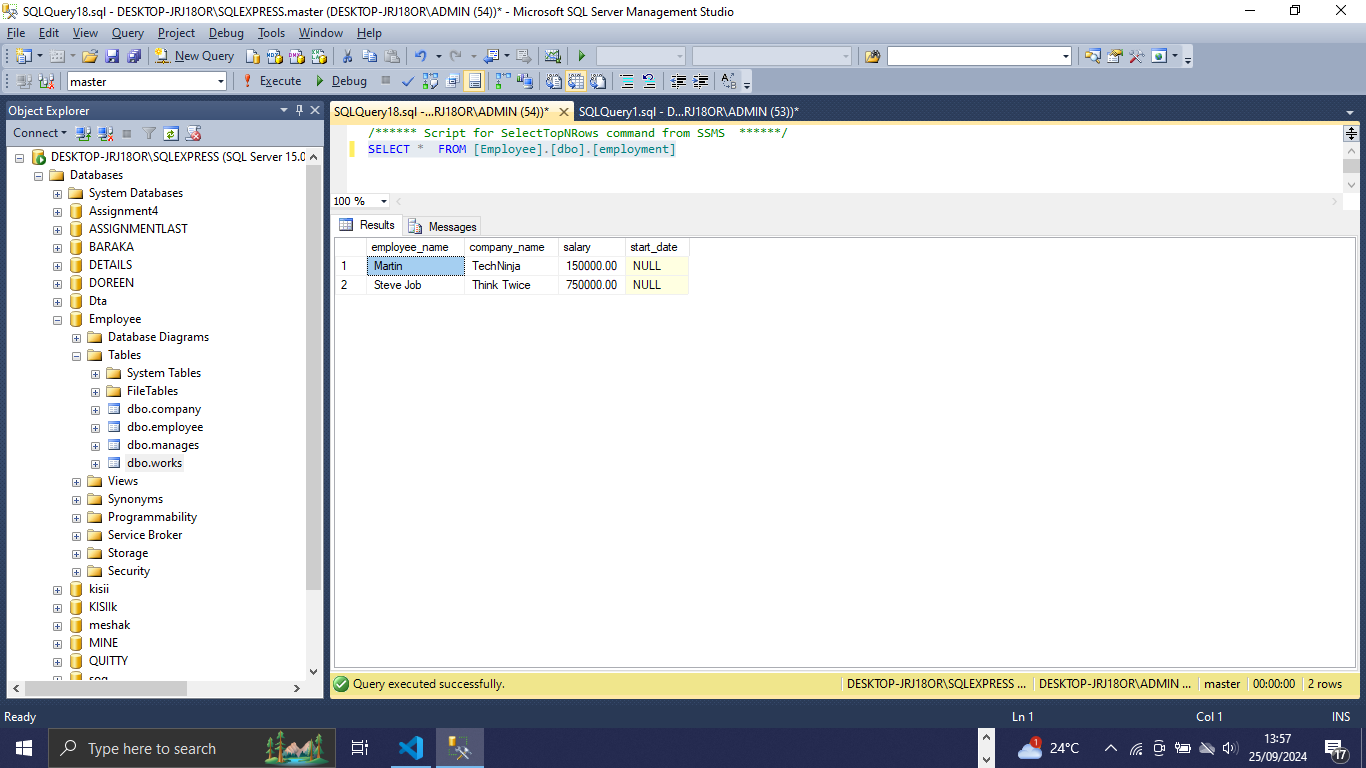
CREATE TABLE works (employee\_name VARCHAR(100),company\_name VARCHAR(100),salary DECIMAL(20, 2),PRIMARY KEY (employee\_name, company\_name),

FOREIGN KEY (employee\_name) REFERENCES employee(employee\_name),FOREIGN KEY (company\_name) REFERENCES company(company\_name));

CREATE TABLE manages (employee\_name VARCHAR(100),manager\_name VARCHAR(100),PRIMARY KEY (employee\_name, manager\_name),

FOREIGN KEY (employee\_name) REFERENCES employee(employee\_name),FOREIGN KEY (manager\_name) REFERENCES employee(employee\_name));





1. Insert at least 4 employees into the "employee" table with their respective details (employee\_name, street, city). Insert at least 3 tuples into each table.

INSERT INTO employee (employee\_name, street, city) VALUES ('Martin', 'Zion Road', 'Washington DC'),('Dehan', 'Maple Road', 'Chicago'),

('Steve Job', 'Alpine 704', 'Florida'),('Zeceal', 'Westside', 'Juan Park'),

('Bill Gates', 'Zion Road', 'Washington DC'),('Bezos', 'Maple Road', 'Chicago'),

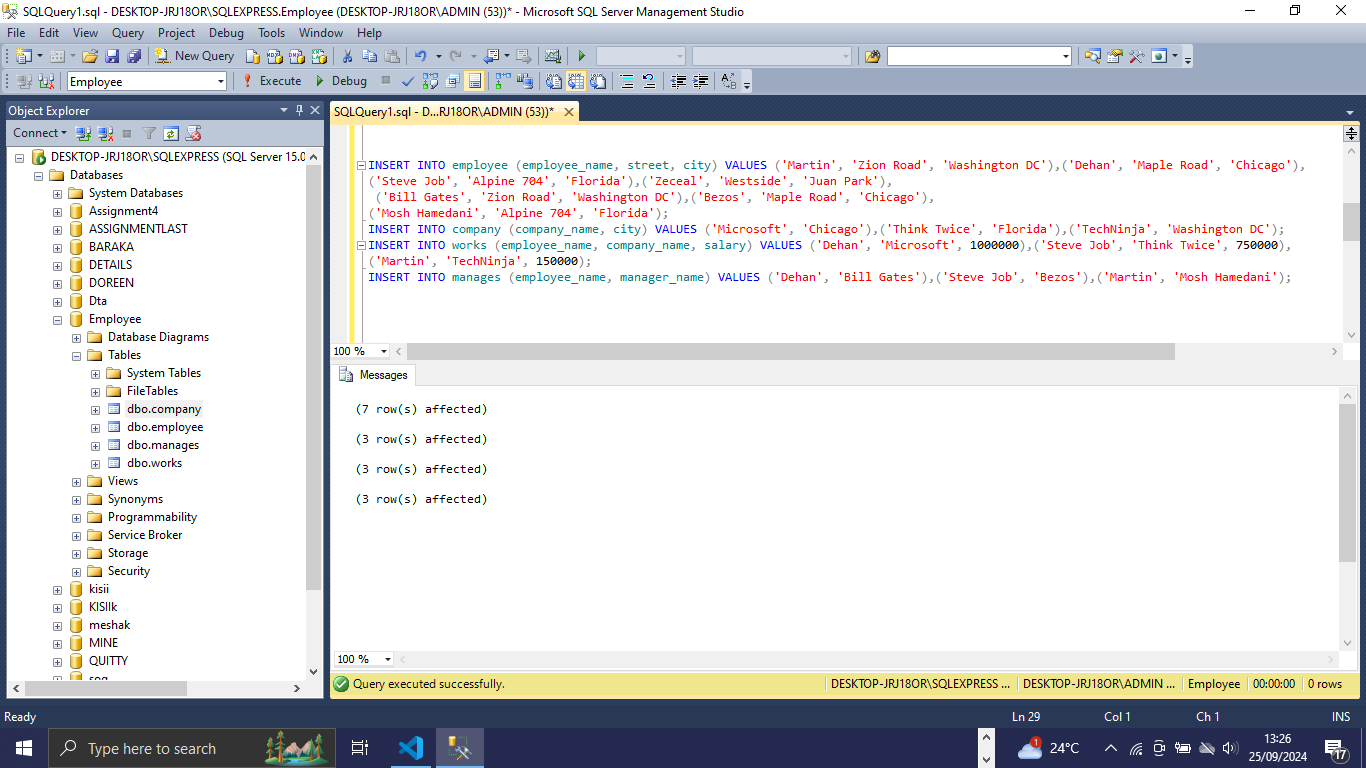
('Mosh Hamedani', 'Alpine 704', 'Florida');

INSERT INTO company (company\_name, city) VALUES ('Microsoft', 'Chicago'),('Think Twice', 'Florida'),('TechNinja', 'Washington DC');

INSERT INTO works (employee\_name, company\_name, salary) VALUES ('Dehan', 'Microsoft', 1000000),('Steve Job', 'Think Twice', 750000),

('Martin', 'TechNinja', 150000);

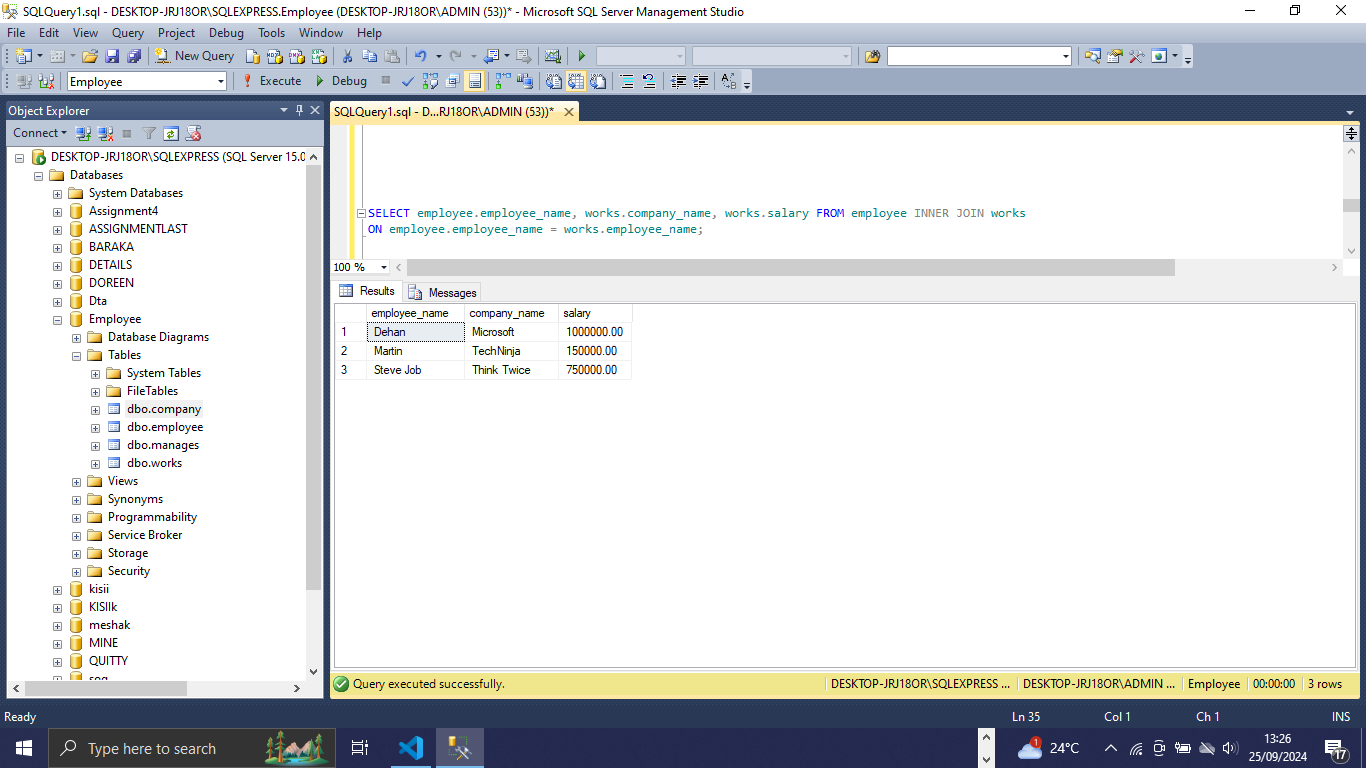
INSERT INTO manages (employee\_name, manager\_name) VALUES ('Dehan', 'Bill Gates'),('Steve Job', 'Bezos'),('Martin', 'Mosh Hamedani');



1. Write an SQL query to retrieve the names of all employees and their associated company names and salaries using an inner join operation.

SELECT employee.employee\_name, works.company\_name, works.salary FROM employee INNER JOIN works

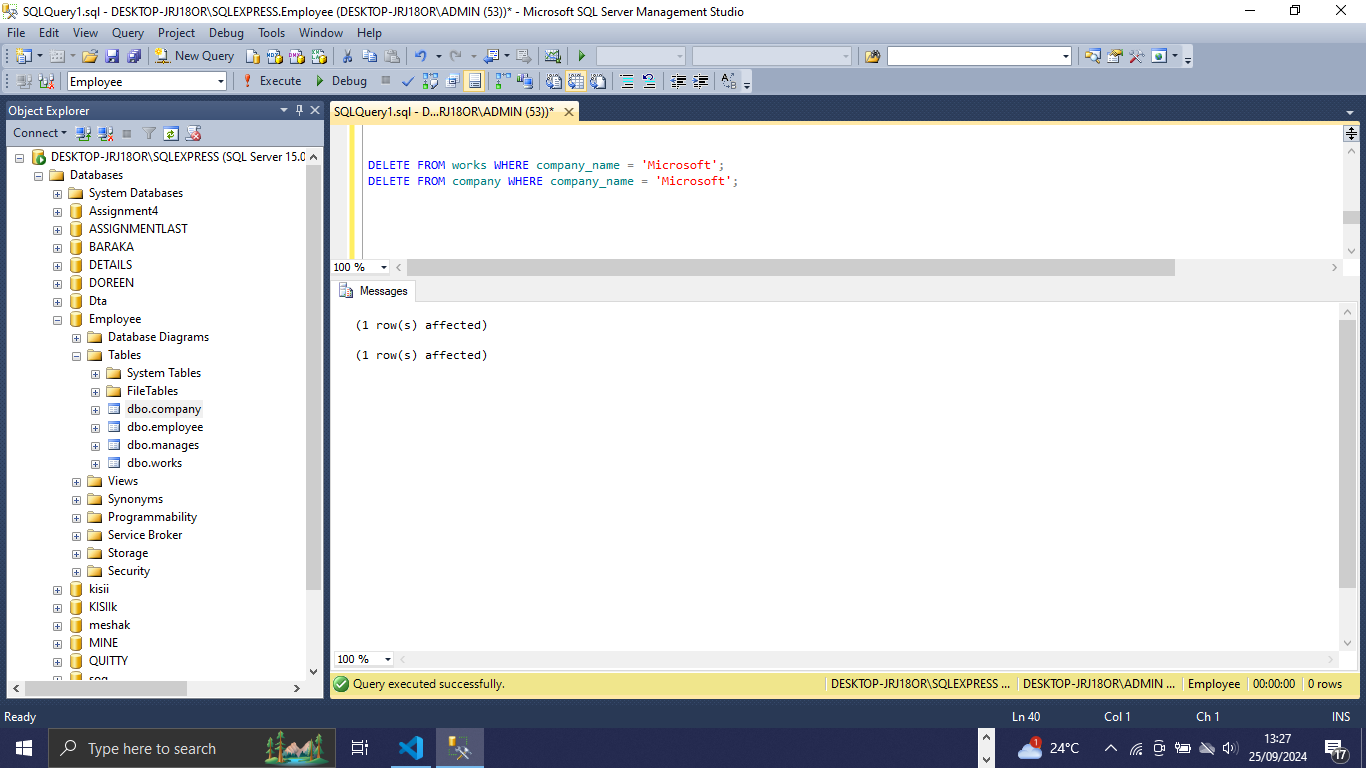
ON employee.employee\_name = works.employee\_name;



1. Write an SQL query to delete a company from the "company" table along with its associated data in the "works" table.

DELETE FROM works WHERE company\_name = 'Microsoft';

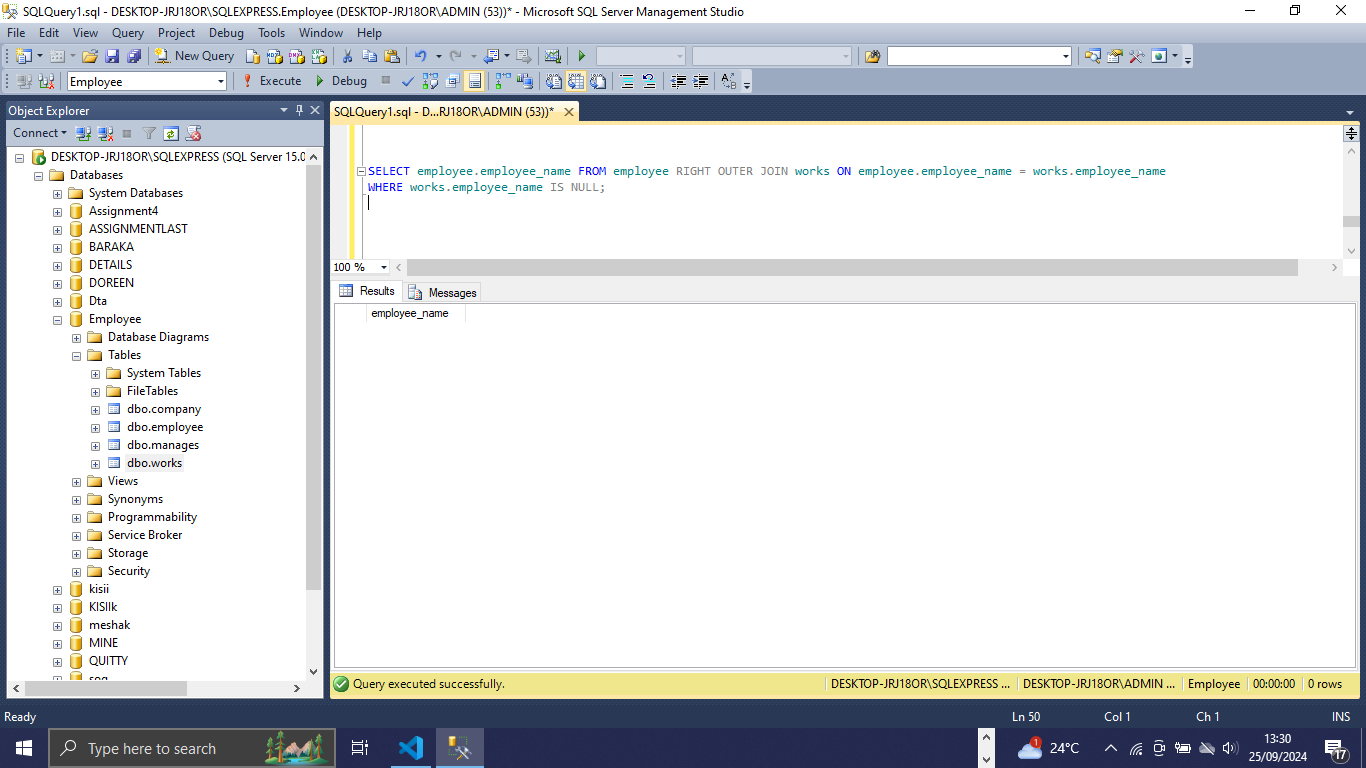
DELETE FROM company WHERE company\_name = 'Microsoft';



1. Write an SQL query to retrieve the names of employees who do not have a company listed in the "works" table using a right outer join operation.

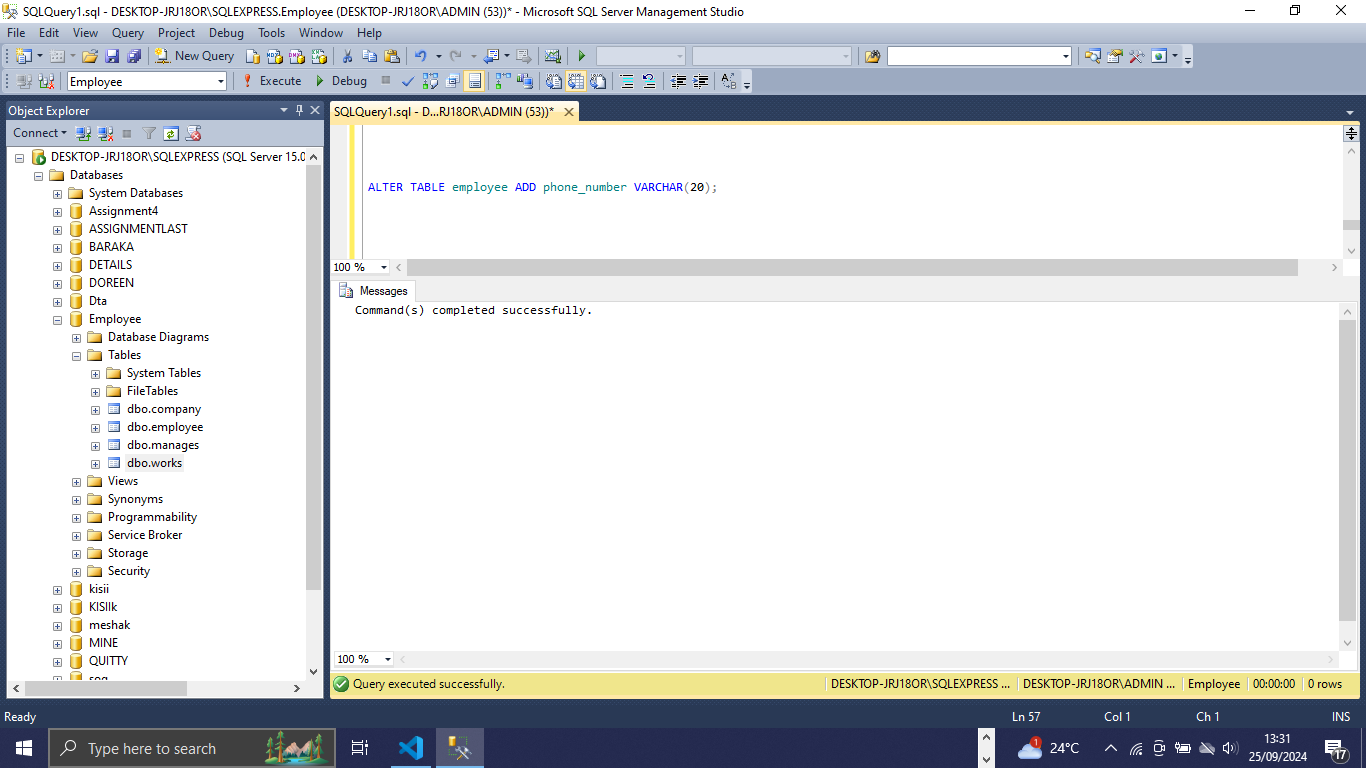
SELECT employee.employee\_name FROM employee RIGHT OUTER JOIN works ON employee.employee\_name = works.employee\_name

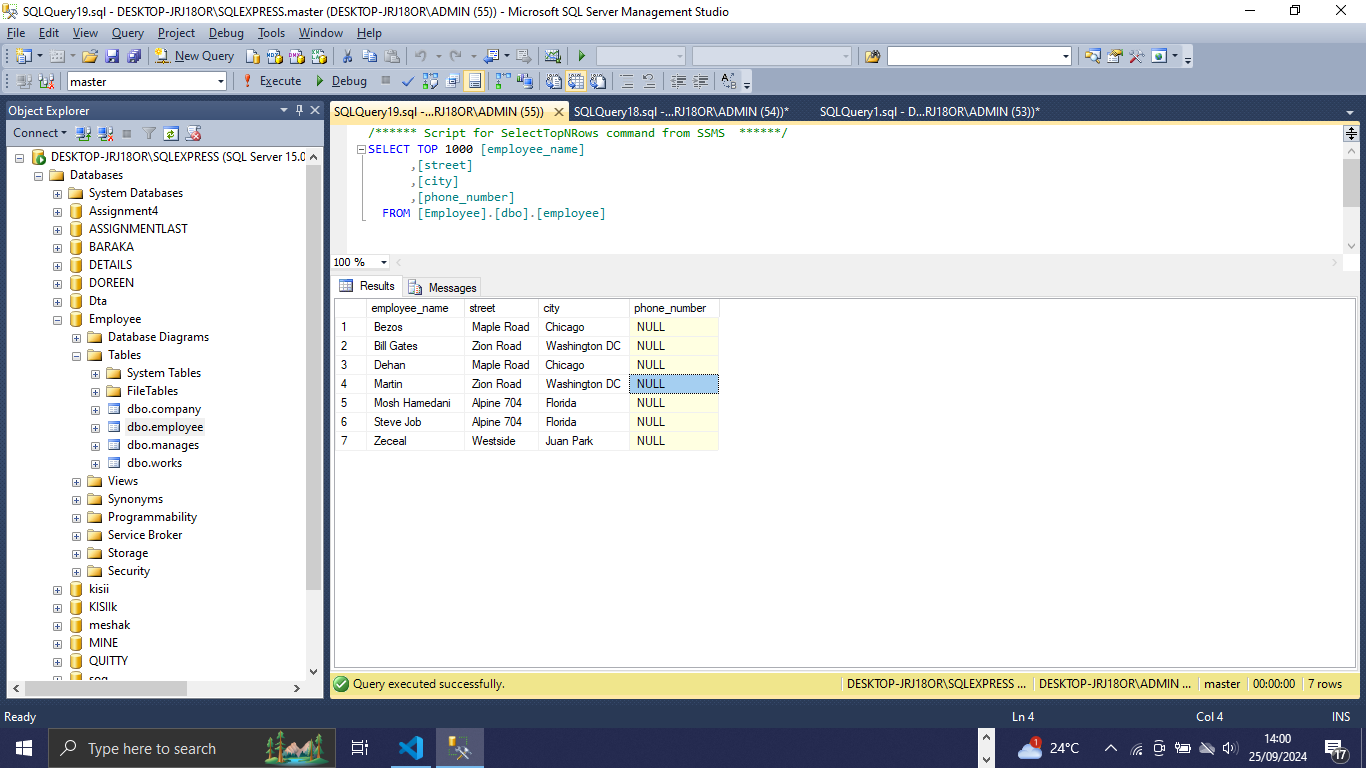
WHERE works.employee\_name IS NULL;



1. Alter the structure of the "employee" table to add a new column called "phone\_number" of type VARCHAR(20).

ALTER TABLE employee ADD phone\_number VARCHAR(20);

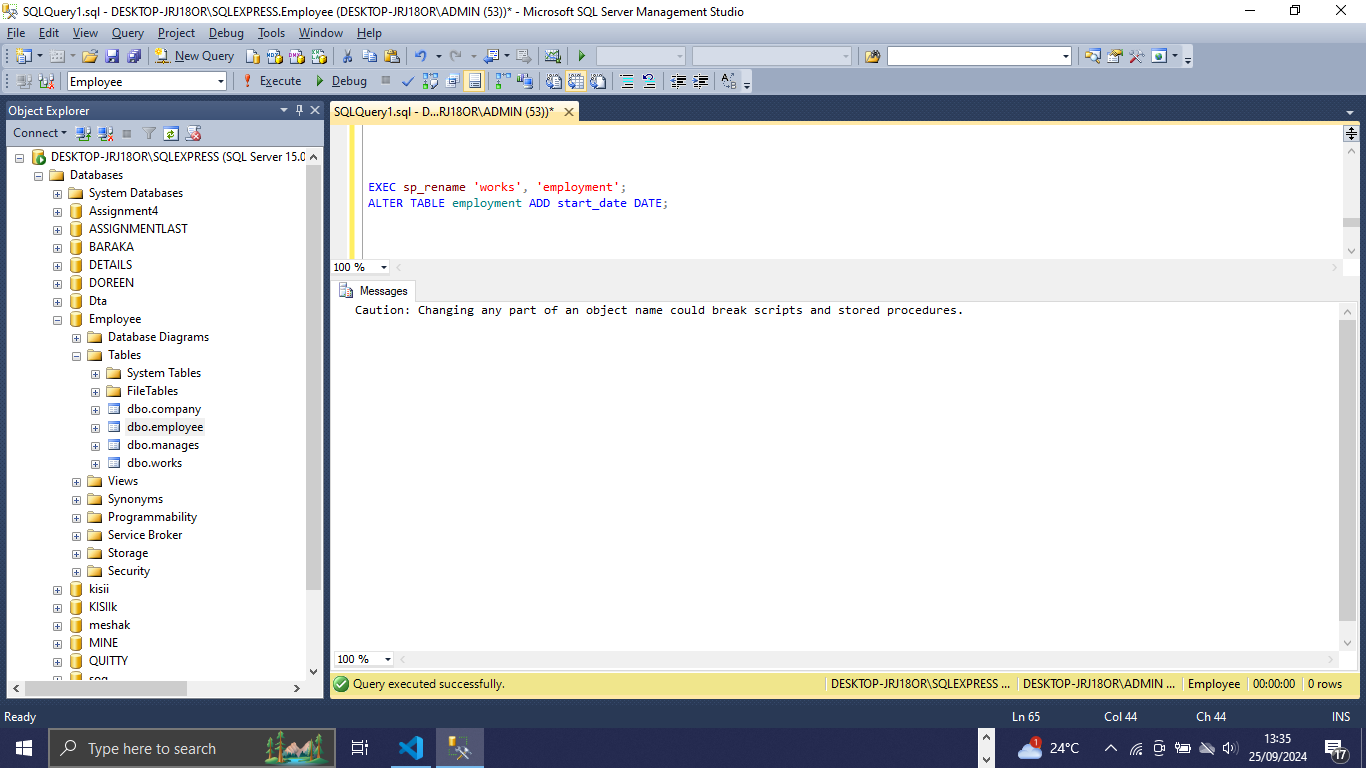


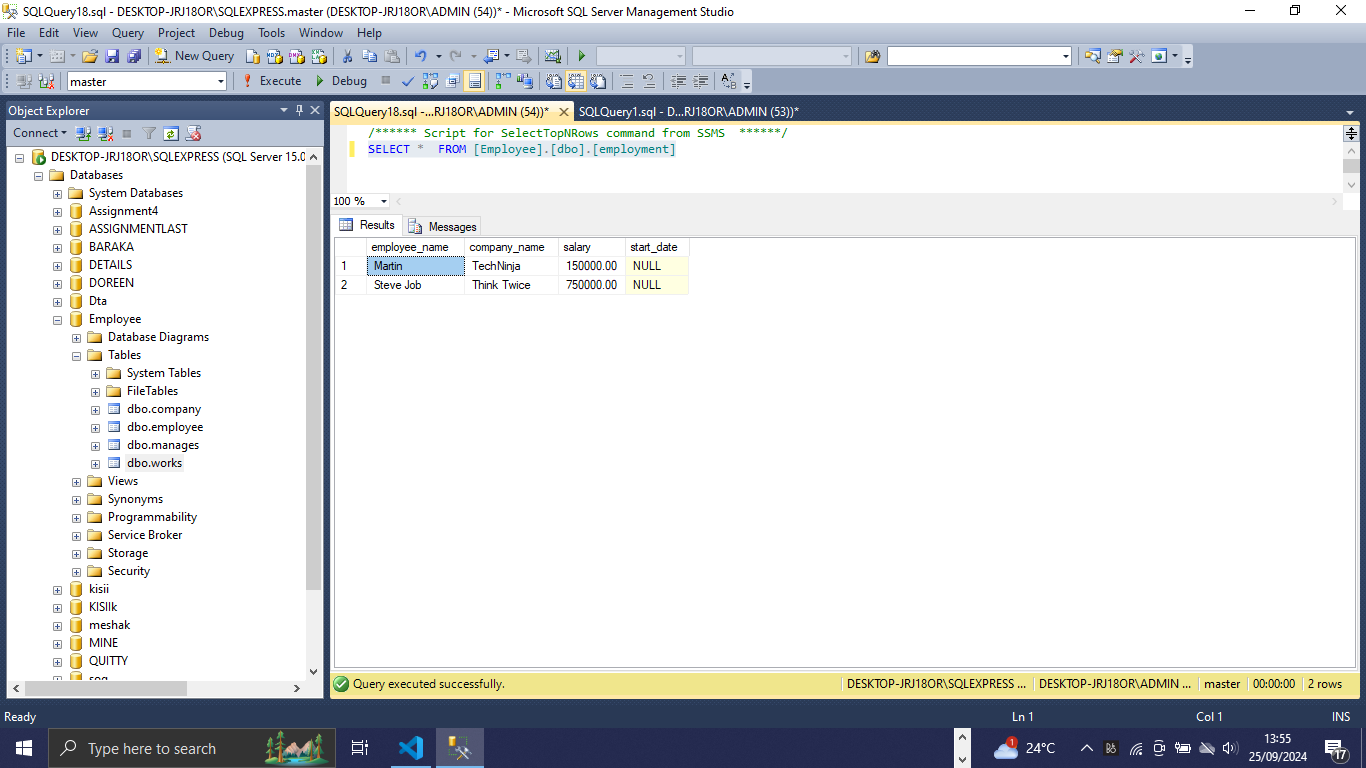


1. Rename the "works" table to "employment" and modify its structure to include an additional column called "start\_date" of type DATE.

EXEC sp\_rename 'works', 'employment';

ALTER TABLE employment ADD start\_date DATE;





1. Write an SQL query to count the number of employees in each city using a group by clause and join operations.

SELECT employee.city, COUNT(employee.employee\_name) AS employee\_count FROM employee GROUP BY employee.city;

