Task 1 :

1. Provide a SQL script that initializes the database for the Job Board scenario “CareerHub”

Query:

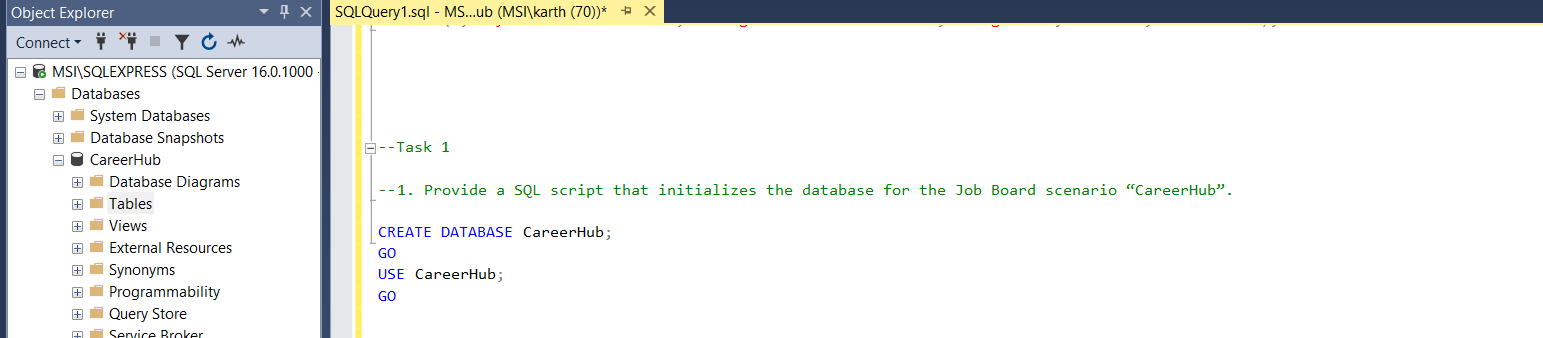
CREATE DATABASE CareerHub;

GO

USE CareerHub;

GO

Output:



Task 2:

2. Create tables for Companies, Jobs, Applicants and Applications.

Query:

CREATE TABLE Companies (

CompanyID INT PRIMARY KEY IDENTITY(1,1),

CompanyName VARCHAR(255) NOT NULL,

Location VARCHAR(255) NOT NULL

);

CREATE TABLE Jobs (

JobID INT PRIMARY KEY IDENTITY(1,1),

CompanyID INT NOT NULL,

JobTitle VARCHAR(255) NOT NULL,

JobDescription TEXT NOT NULL,

JobLocation VARCHAR(255) NOT NULL,

Salary DECIMAL(10,2) NOT NULL,

JobType VARCHAR(50) NOT NULL,

PostedDate DATETIME DEFAULT GETDATE(),

FOREIGN KEY (CompanyID) REFERENCES Companies(CompanyID) ON DELETE CASCADE

);

CREATE TABLE Applicants (

ApplicantID INT PRIMARY KEY IDENTITY(1,1),

FirstName VARCHAR(100) NOT NULL,

LastName VARCHAR(100) NOT NULL,

Email VARCHAR(255) UNIQUE NOT NULL,

Phone VARCHAR(20) UNIQUE NOT NULL,

Resume TEXT NOT NULL

);

ALTER TABLE Applicants

ADD City VARCHAR(100),

State VARCHAR(100);

UPDATE Applicants

SET City = 'chennai', State = 'Tamilnadu'

WHERE ApplicantID = 1;

UPDATE Applicants

SET City = 'mumbai', State = 'Maharastra'

WHERE ApplicantID = 2;

UPDATE Applicants

SET City = 'delhi', State = 'Newdelhi'

WHERE ApplicantID = 3;

CREATE TABLE Applications (

ApplicationID INT PRIMARY KEY IDENTITY(1,1),

JobID INT NOT NULL,

ApplicantID INT NOT NULL,

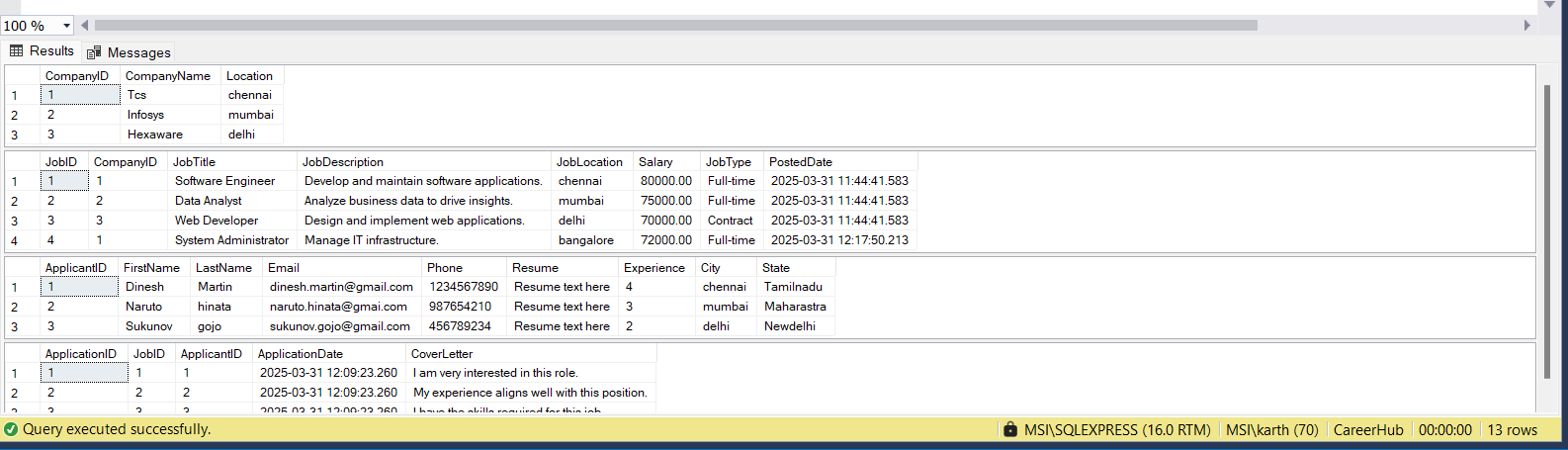
ApplicationDate DATETIME DEFAULT GETDATE(),

CoverLetter TEXT NOT NULL,

FOREIGN KEY (JobID) REFERENCES Jobs(JobID) ON DELETE CASCADE,

FOREIGN KEY (ApplicantID) REFERENCES Applicants(ApplicantID) ON DELETE CASCADE

);



Task 3:

3.Define appropriate primary keys, foreign keys, and constraints.

Query: Assinging Primary keys and foreign keys

CREATE TABLE Jobs (

JobID INT PRIMARY KEY IDENTITY(1,1),

CompanyID INT NOT NULL,

JobTitle VARCHAR(255) NOT NULL,

JobDescription TEXT NOT NULL,

JobLocation VARCHAR(255) NOT NULL,

Salary DECIMAL(10,2) NOT NULL,

JobType VARCHAR(50) NOT NULL,

PostedDate DATETIME DEFAULT GETDATE(),

FOREIGN KEY (CompanyID) REFERENCES Companies(CompanyID) ON DELETE CASCADE

);

Task 4 :

1. Ensure the script handles potential errors, such as if the database or tables already exist.

Query: Ensuring the query if not exists or not

IF NOT EXISTS (SELECT \* FROM sys.databases WHERE name = 'CareerHub')

BEGIN

CREATE DATABASE CareerHub;

END

GO

USE CareerHub;

GO

Task 5:

5. Write an SQL query to count the number of applications received for each job listing in the

"Jobs" table. Display the job title and the corresponding application count. Ensure that it lists all jobs, even if they have no applications.

Query:

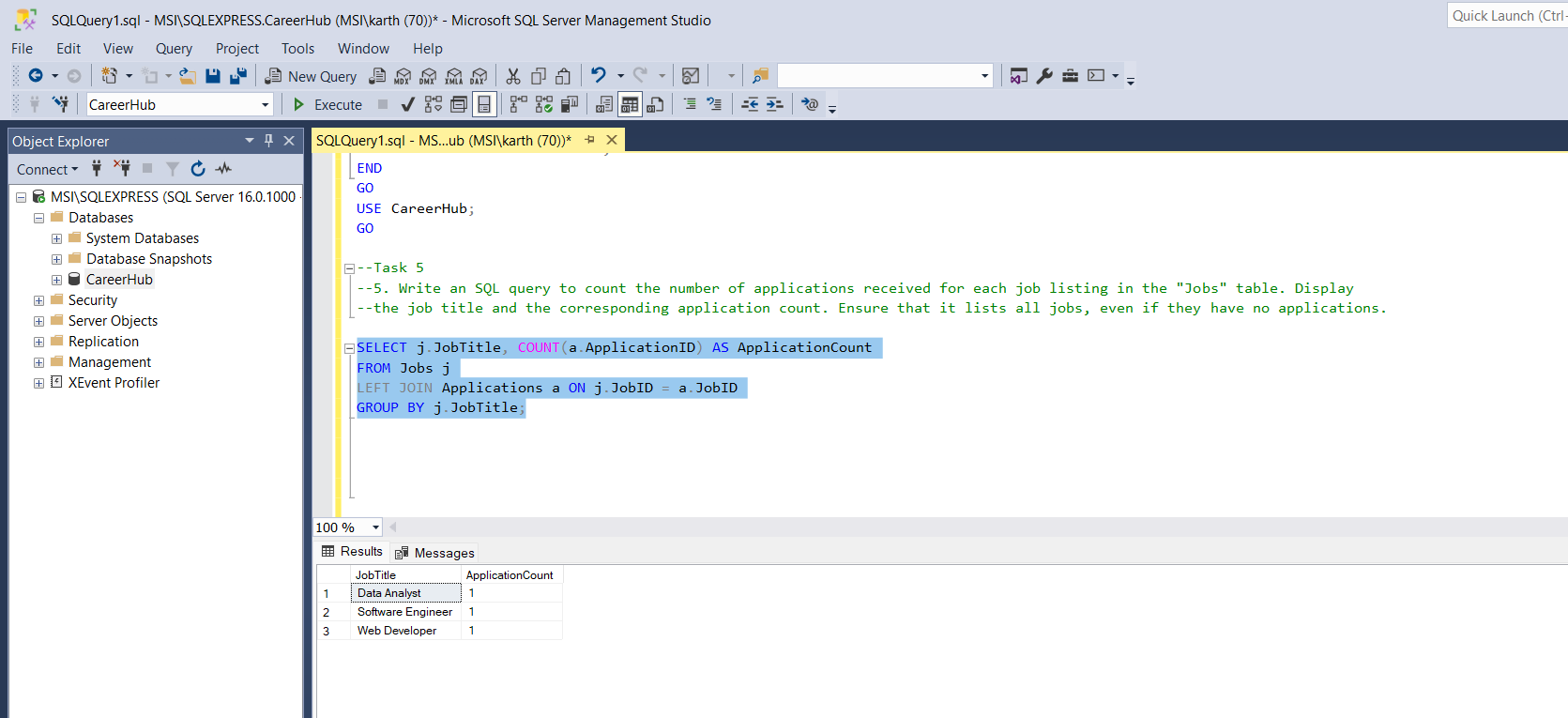
SELECT j.JobTitle, COUNT(a.ApplicationID) AS ApplicationCount

FROM Jobs j

LEFT JOIN Applications a ON j.JobID = a.JobID

GROUP BY j.JobTitle;

Output:



Task 6:

6.Develop an SQL query that retrieves job listings from the "Jobs" table within a specified salary range. Allow parameters for the minimum and maximum salary values. Display the job title,company name, location, and salary for each matching job.

Query:

DECLARE @MinSalary DECIMAL(10,2) = 70000;

DECLARE @MaxSalary DECIMAL(10,2) = 90000;

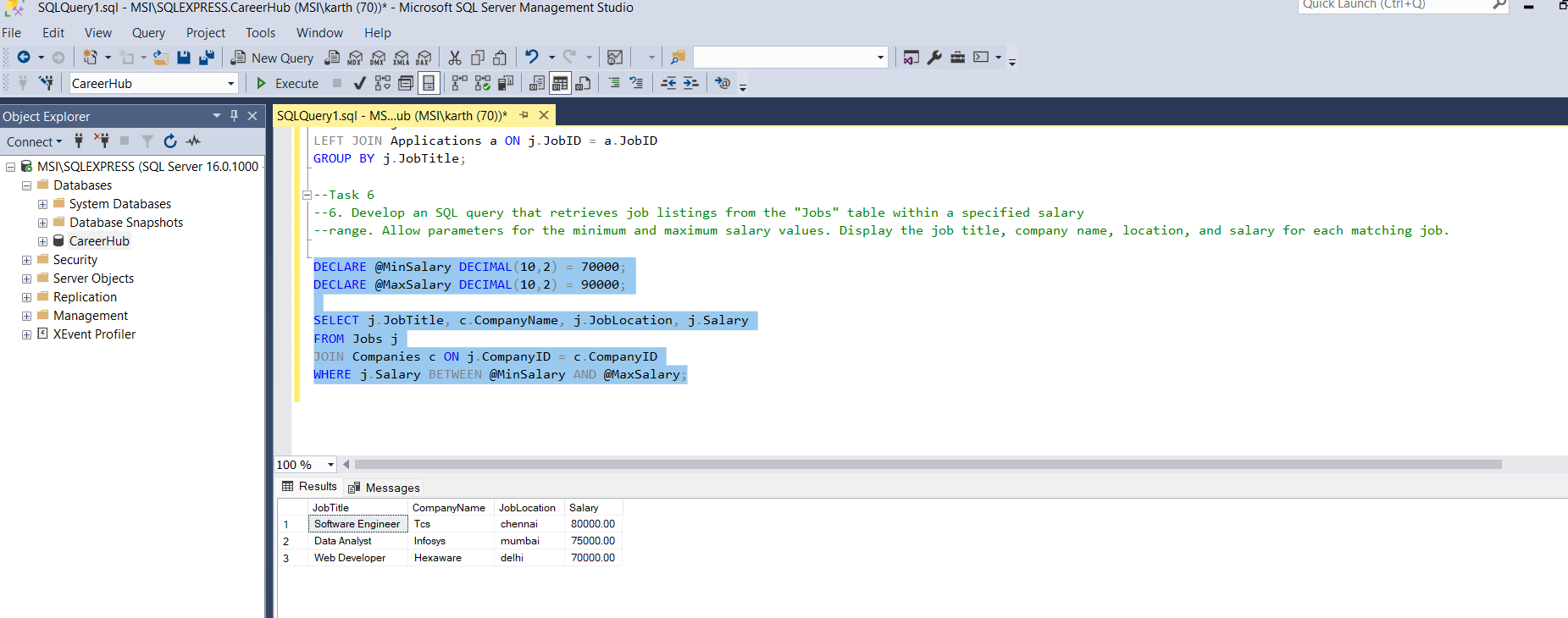
SELECT j.JobTitle, c.CompanyName, j.JobLocation, j.Salary

FROM Jobs j

JOIN Companies c ON j.CompanyID = c.CompanyID

WHERE j.Salary BETWEEN @MinSalary AND @MaxSalary;

Output:



Task 7:

1. Write an SQL query that retrieves the job application history for a specific applicant. Allow a parameter for the ApplicantID, and return a result set with the job titles, company names, and application dates for all the jobs the applicant has applied to.

Query

DECLARE @ApplicantID INT = 1; -- Set the ApplicantID dynamically

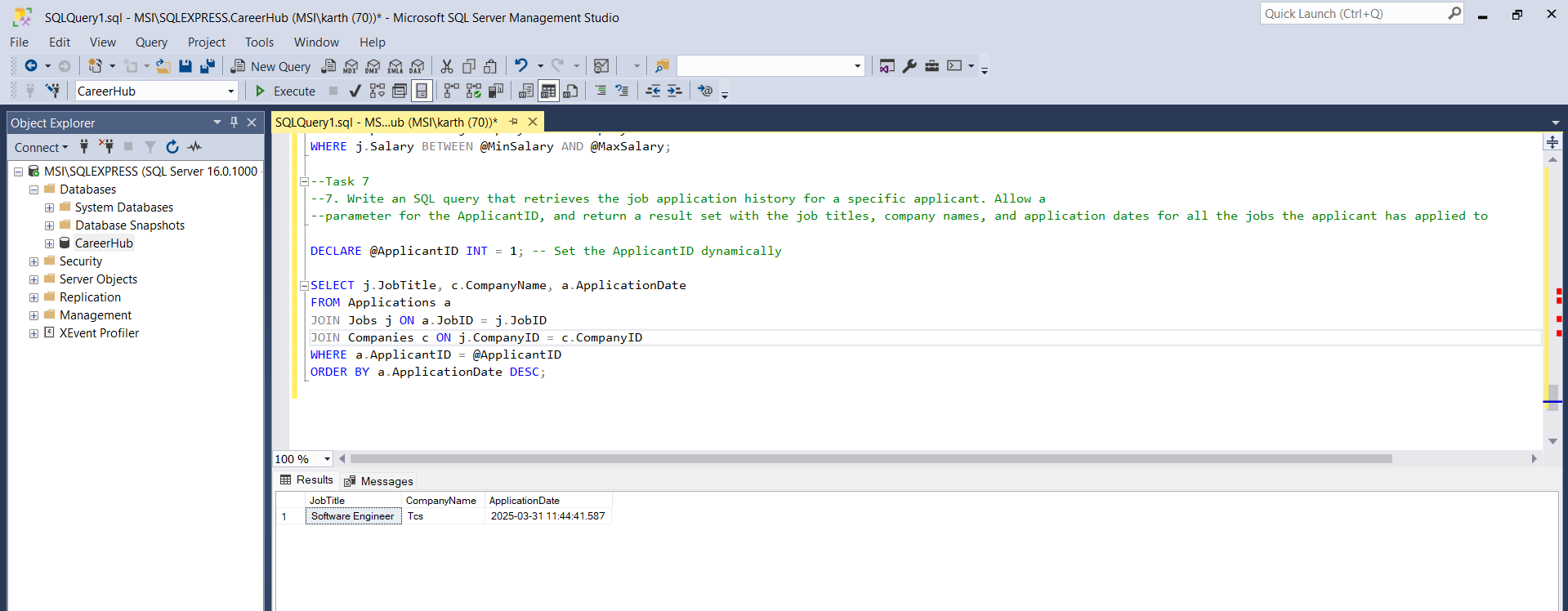
SELECT j.JobTitle, c.CompanyName, a.ApplicationDate

FROM Applications a

JOIN Jobs j ON a.JobID = j.JobID

JOIN Companies c ON j.CompanyID = c.CompanyID

WHERE a.ApplicantID = @ApplicantID ORDER BY a.ApplicationDate DESC;



Task 8:

8. Create an SQL query that calculates and displays the average salary offered by all companies for

job listings in the "Jobs" table. Ensure that the query filters out jobs with a salary of zero.

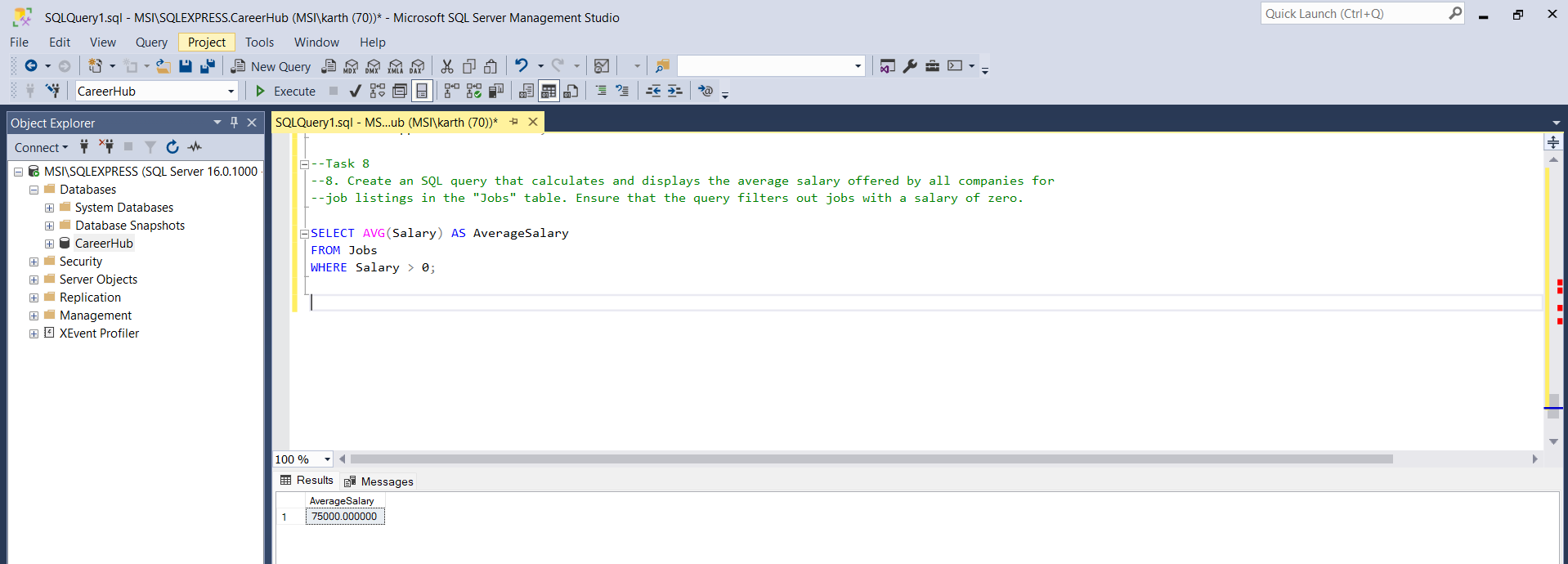
Query:

SELECT AVG(Salary) AS AverageSalary

FROM Jobs

WHERE Salary > 0;

Output:



Task 9:

9. Write an SQL query to identify the company that has posted the most job listings. Display the company name along with the count of job listings they have posted. Handle ties if multiple companies have the same maximum count.

Query:

WITH JobCounts AS (

SELECT CompanyID, COUNT(JobID) AS JobCount

FROM Jobs

GROUP BY CompanyID

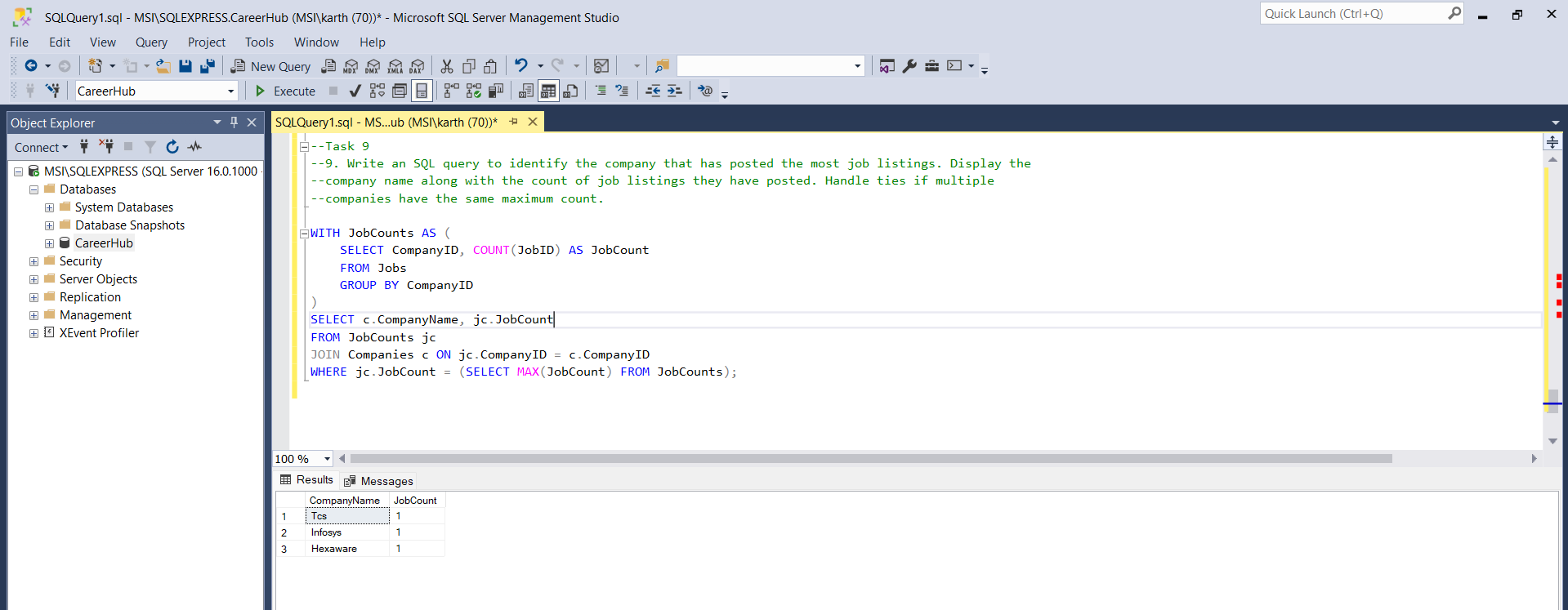
)

SELECT c.CompanyName, jc.JobCount

FROM JobCounts jc

JOIN Companies c ON jc.CompanyID = c.CompanyID

WHERE jc.JobCount = (SELECT MAX(JobCount) FROM JobCounts);



Task 10:

1. Find the applicants who have applied for positions in companies located in 'CityX' and have at least 3 years of experience.

Query:

SELECT DISTINCT A.ApplicantID, A.FirstName, A.LastName, A.Experience, C.CompanyName, C.Location

FROM Applications AP

JOIN Applicants A ON AP.ApplicantID = A.ApplicantID

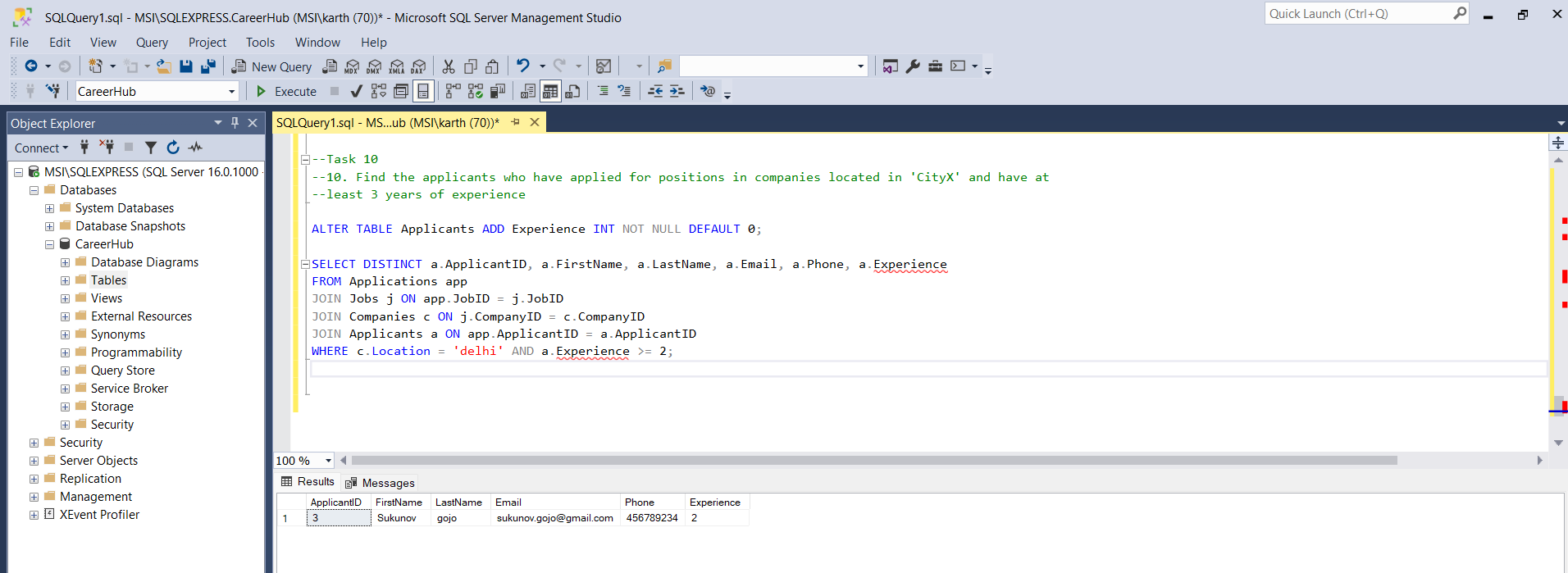
JOIN Jobs J ON AP.JobID = J.JobID

JOIN Companies C ON J.CompanyID = C.CompanyID

WHERE C.Location = 'mumbai'

AND A.Experience >= 3;

Output:



Task 11:

11. Retrieve a list of distinct job titles with salaries between $60,000 and $80,000.

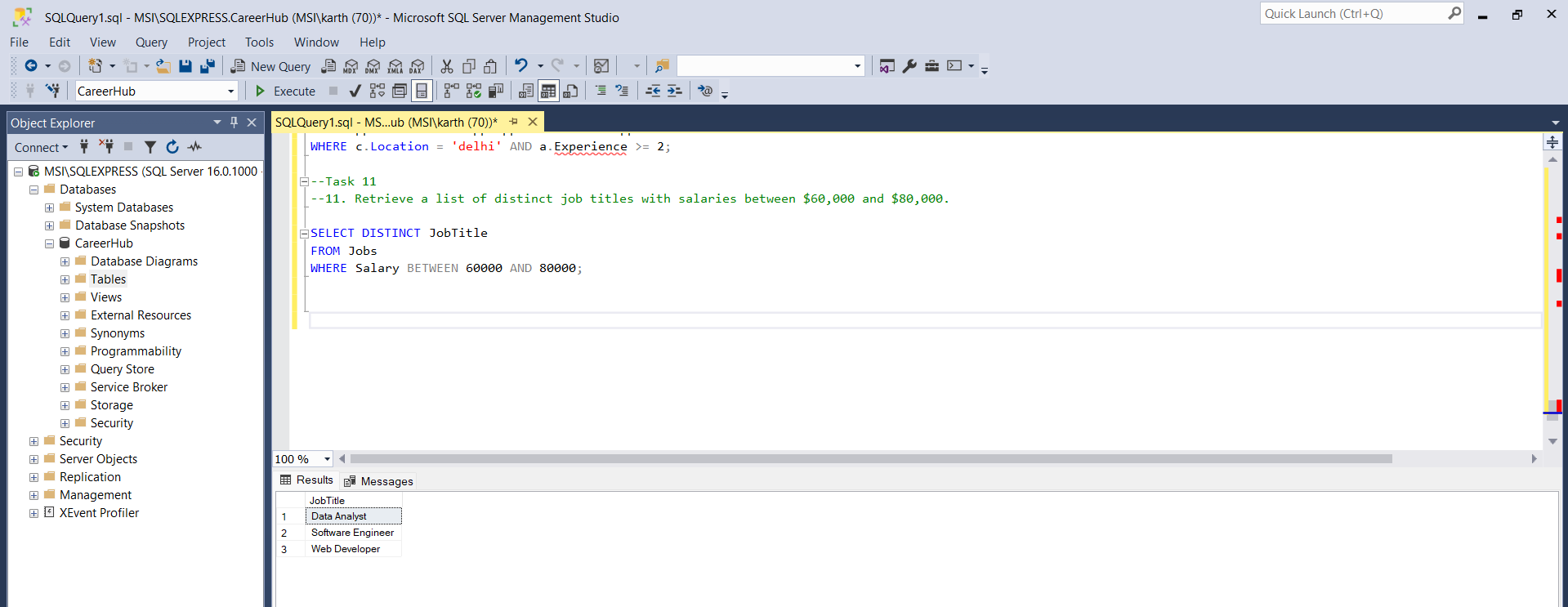
Query:

SELECT DISTINCT JobTitle

FROM Jobs

WHERE Salary BETWEEN 60000 AND 80000;

Output:



Task 12:

12. Find the jobs that have not received any applications.

Query:

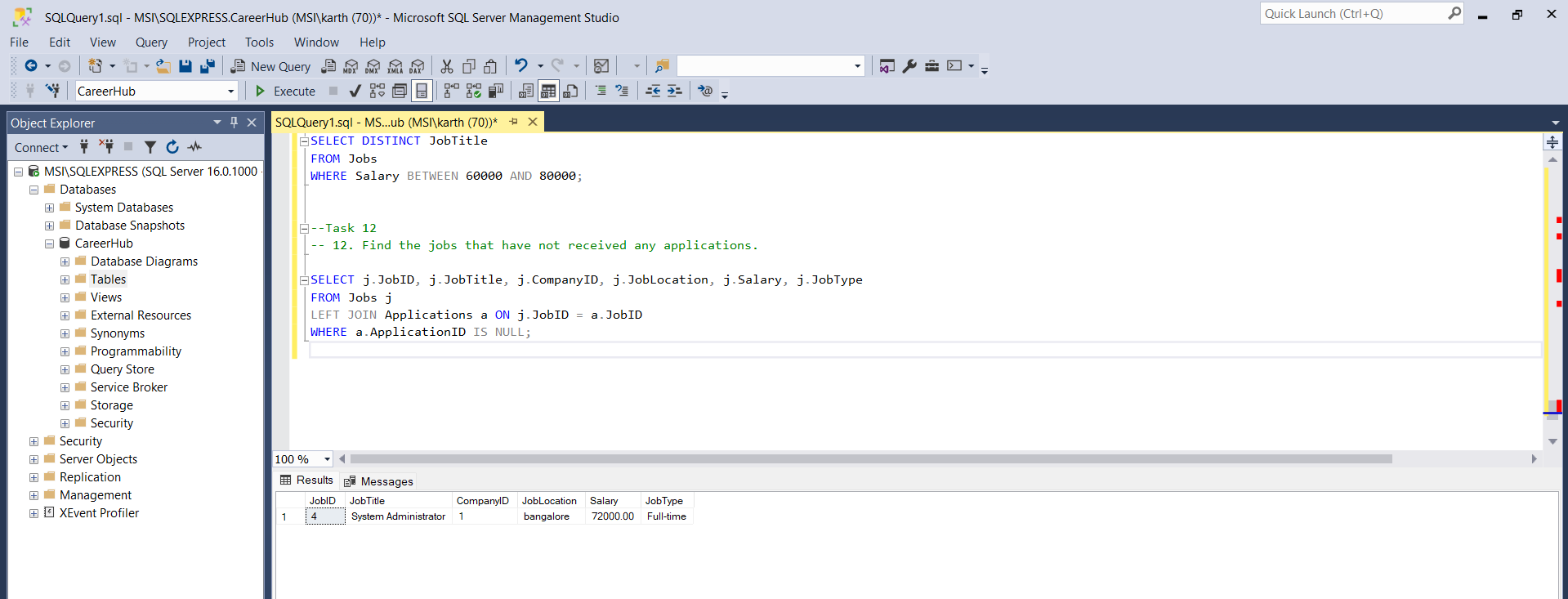
SELECT j.JobID, j.JobTitle, j.CompanyID, j.JobLocation, j.Salary, j.JobType

FROM Jobs j

LEFT JOIN Applications a ON j.JobID = a.JobID

WHERE a.ApplicationID IS NULL;

Output:



Task 13:

1. Retrieve a list of job applicants along with the companies they have applied to and the positions they have applied for.

Query:

SELECT

a.ApplicantID,

a.FirstName,

a.LastName,

a.Email,

c.CompanyName,

j.JobTitle,

ap.ApplicationDate

FROM Applications ap

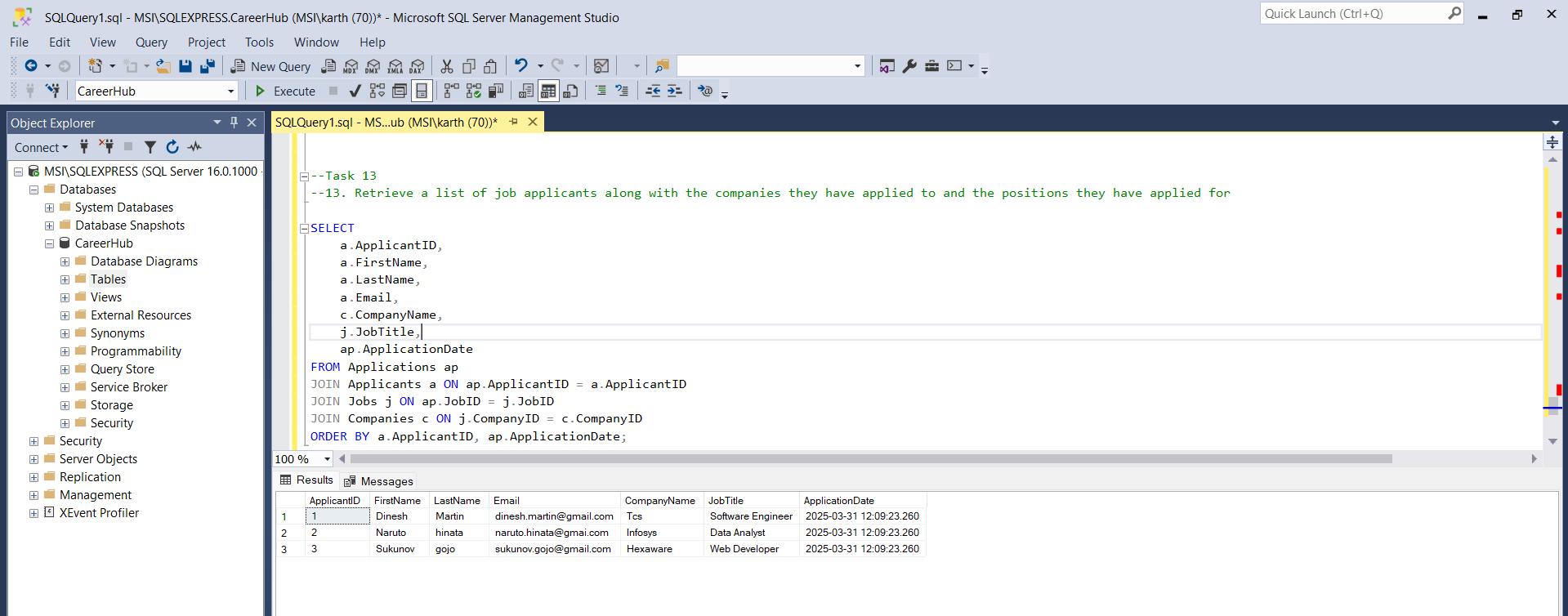
JOIN Applicants a ON ap.ApplicantID = a.ApplicantID

JOIN Jobs j ON ap.JobID = j.JobID

JOIN Companies c ON j.CompanyID = c.CompanyID

ORDER BY a.ApplicantID, ap.ApplicationDate;

Output:



Task 14:

1. Retrieve a list of companies along with the count of jobs they have posted, even if they have not received any applications.

Query

SELECT

c.CompanyID,

c.CompanyName,

COUNT(j.JobID) AS JobCount

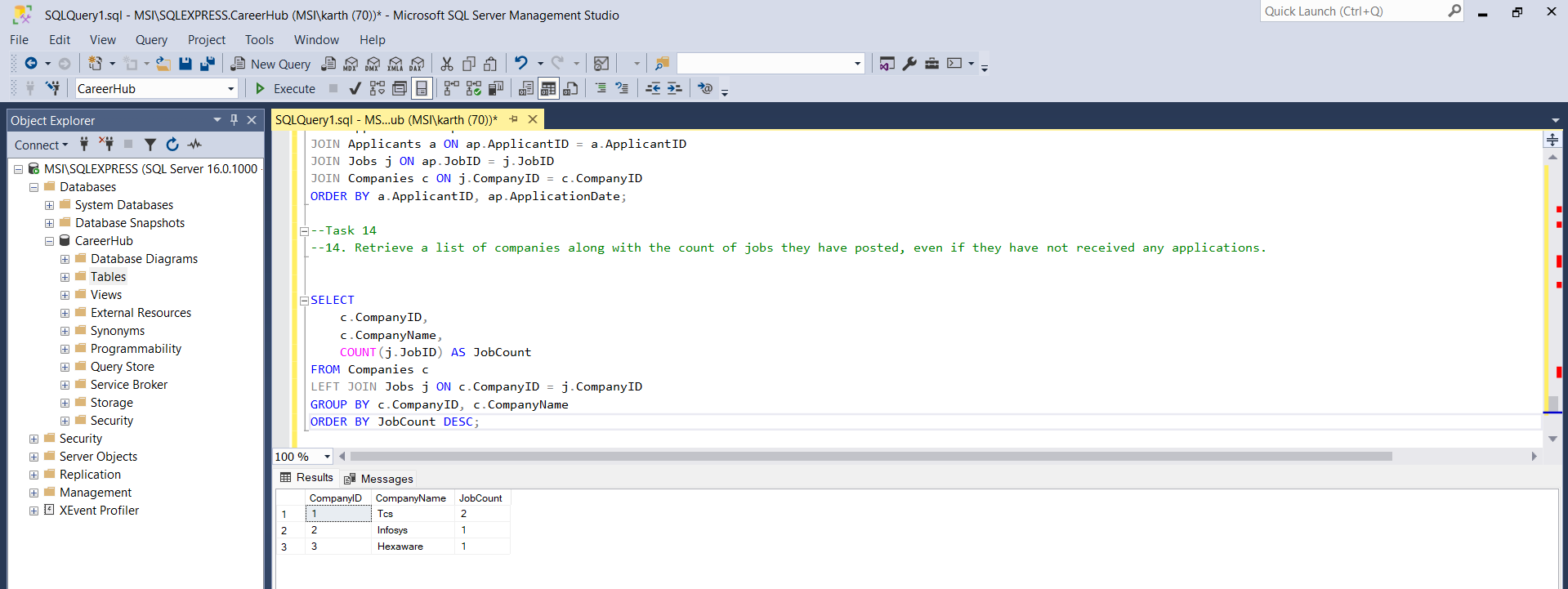
FROM Companies c

LEFT JOIN Jobs j ON c.CompanyID = j.CompanyID

GROUP BY c.CompanyID, c.CompanyName

ORDER BY JobCount DESC;

Ouput:



Task 15:

1. List all applicants along with the companies and positions they have applied for, including those who have not applied.

Query:

SELECT

a.ApplicantID,

a.FirstName,

a.LastName,

a.Email,

COALESCE(c.CompanyName, 'No Application') AS CompanyName,

COALESCE(j.JobTitle, 'No Application') AS JobTitle

FROM Applicants a

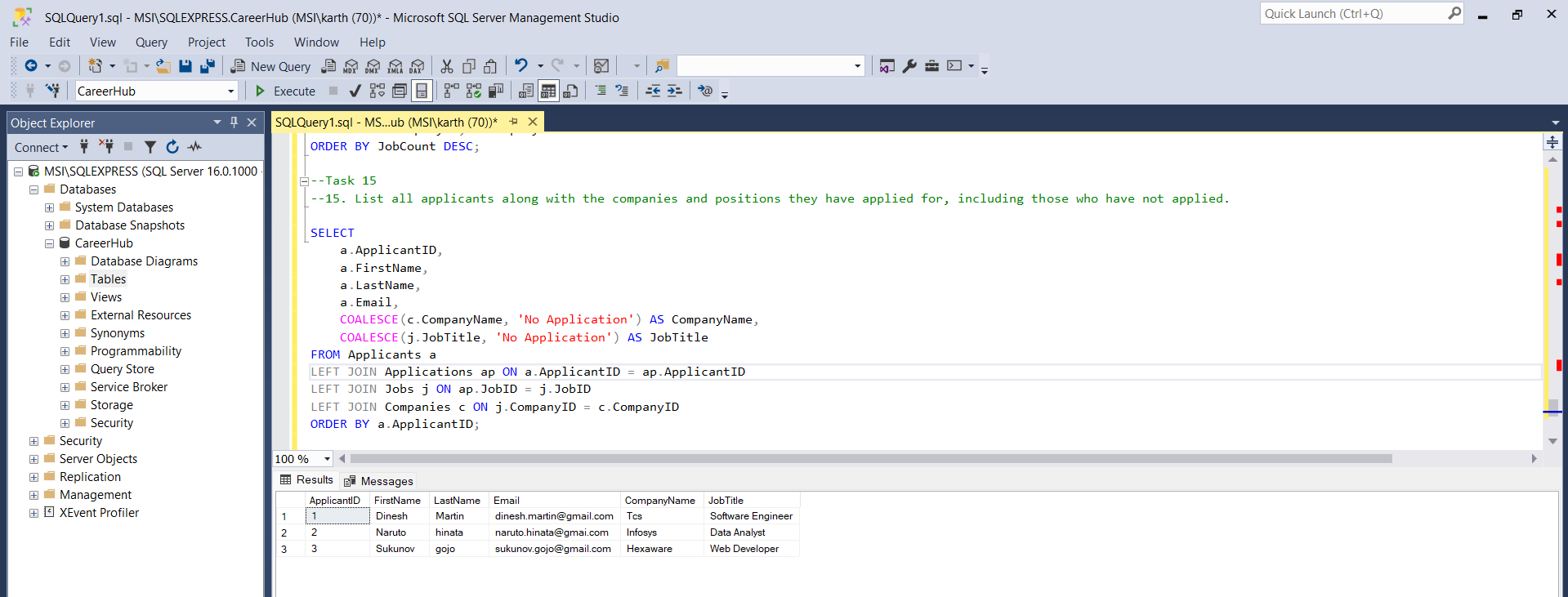
LEFT JOIN Applications ap ON a.ApplicantID = ap.ApplicantID

LEFT JOIN Jobs j ON ap.JobID = j.JobID

LEFT JOIN Companies c ON j.CompanyID = c.CompanyID

ORDER BY a.ApplicantID;

Output:



Task 16:

16. Find companies that have posted jobs with a salary higher than the average salary of all jobs.

Query:

SELECT DISTINCT

c.CompanyID,

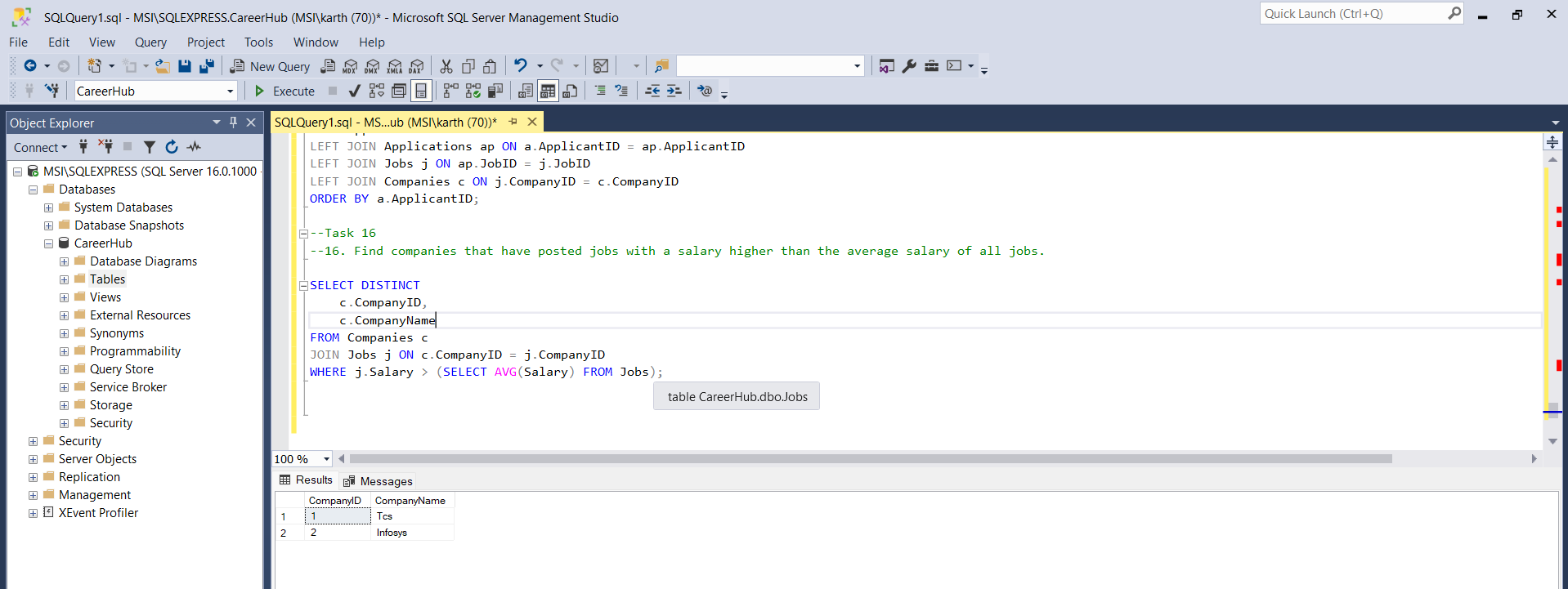
c.CompanyName

FROM Companies c

JOIN Jobs j ON c.CompanyID = j.CompanyID

WHERE j.Salary > (SELECT AVG(Salary) FROM Jobs);

Output:



Task 17:

17. Display a list of applicants with their names and a concatenated string of their city and state.

Query:

SELECT

ApplicantID,

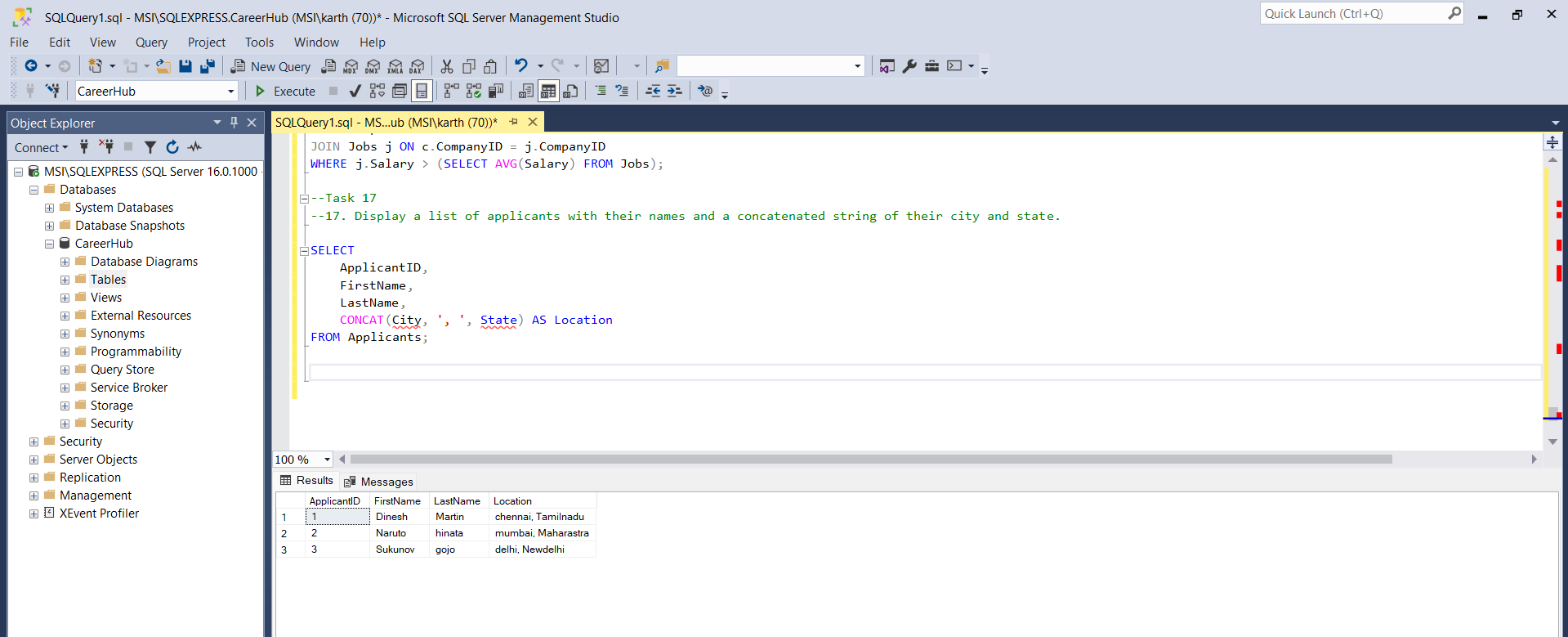
FirstName,

LastName,

CONCAT(City, ', ', State) AS Location

FROM Applicants;

Output:



Task 18:

18. Retrieve a list of jobs with titles containing either 'Developer' or 'Engineer'.

Query:

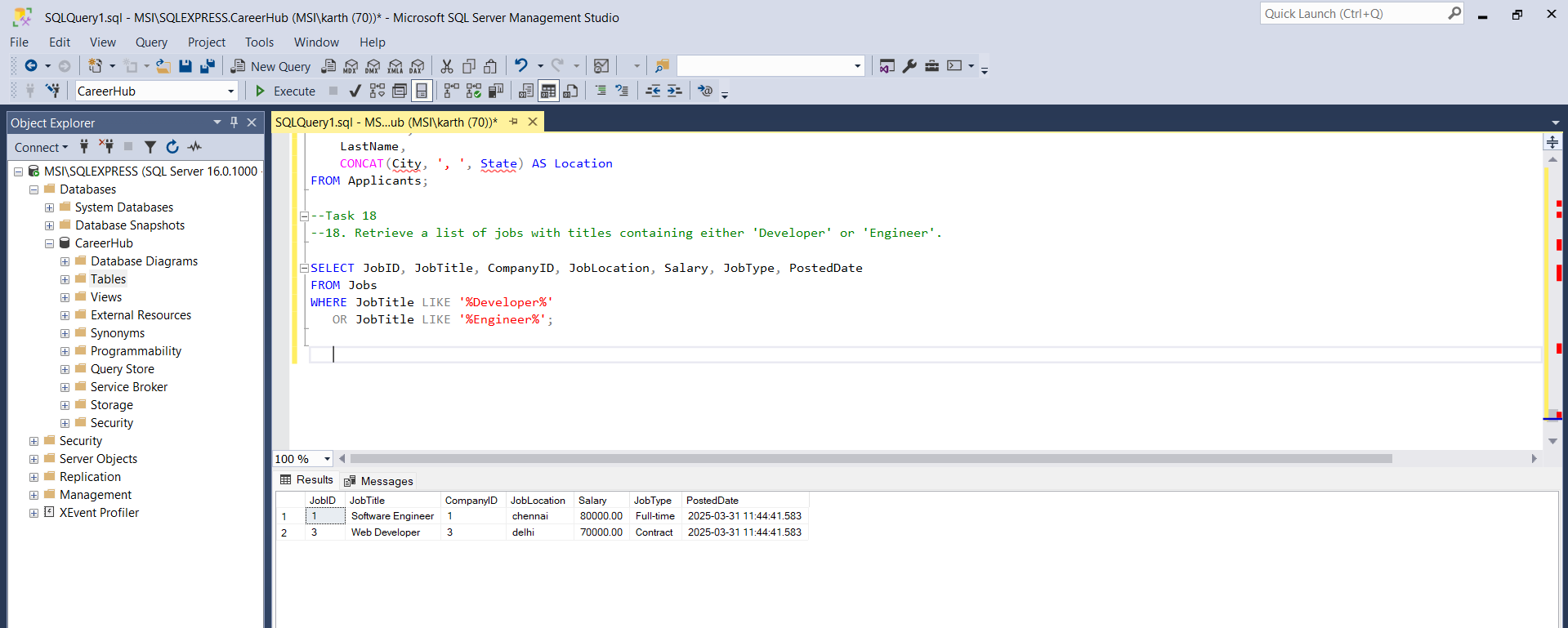
SELECT JobID, JobTitle, CompanyID, JobLocation, Salary, JobType, PostedDate

FROM Jobs

WHERE JobTitle LIKE '%Developer%'

OR JobTitle LIKE '%Engineer%';

Output:



Task 19:

1. Retrieve a list of applicants and the jobs they have applied for, including those who have not applied and jobs without applicants.

Query:

SELECT

A.ApplicantID,

A.FirstName,

A.LastName,

J.JobID,

J.JobTitle,

C.CompanyName

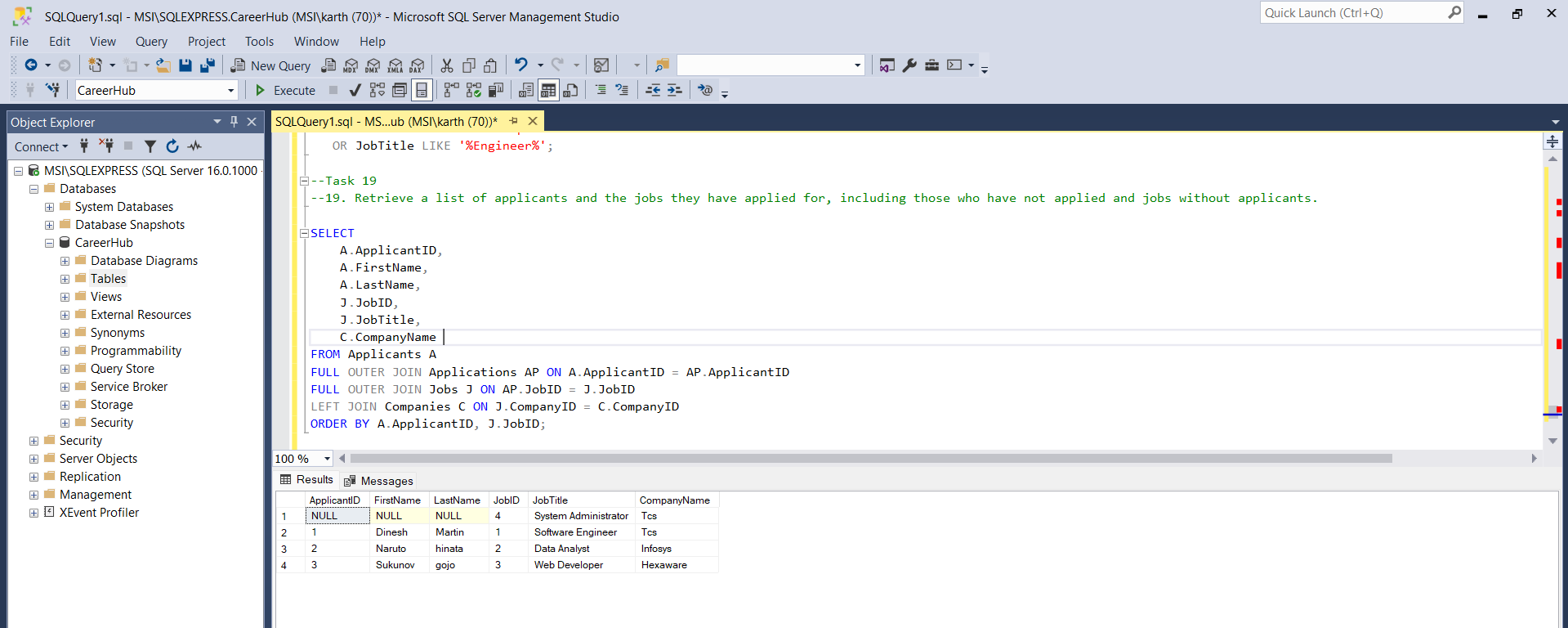
FROM Applicants A

FULL OUTER JOIN Applications AP ON A.ApplicantID = AP.ApplicantID

FULL OUTER JOIN Jobs J ON AP.JobID = J.JobID

LEFT JOIN Companies C ON J.CompanyID = C.CompanyID

ORDER BY A.ApplicantID, J.JobID;

Output:

Task 20:

1. List all combinations of applicants and companies where the company is in a specific city and the applicant has more than 2 years of experience. For example: city=Chennai

Query:

SELECT

A.ApplicantID,

A.FirstName,

A.LastName,

A.Experience,

C.CompanyID,

C.CompanyName,

C.Location

FROM Applicants A

CROSS JOIN Companies C

WHERE C.Location = 'chennai'

AND A.Experience > 2;

Output:

