Coding Challenges: CareerHub, The Job Board

Instructions

- Coding Challenges submissions should be done through the partcipants' Github repository, and the link should be shared with trainers and Hexavarsity.

Problem Statement:

Create SQL Schema from the application, use the class attributes for table column names.

SQL Schema:

Table: Companies
Attributes:
- CompanyID (Primary Key, int): Unique identifier for each company.
- CompanyName (string): The name of the hiring company.
- Location (string): The location of the company.

Table: Jobs
Attributes:
- JobID (Primary Key, int): Unique identifier for each job listing.
- CompanyID (Foreign Key, int): References the CompanyID of the hiring company.
- JobTitle (string): The title of the job.
- JobDescription (text): A detailed description of the job.
- JobLocation (string): The location where the job is based.
- Salary (decimal): The salary offered for the job.
- JobType (string): Type of job (e.g., Full-time, Part-time, Contract).
- PostedDate (datetime): Date and time when the job was posted.

Table: Applicants
Attributes:
ApplicantID (Primary Key, int): Unique identifier for each applicant.
- FirstName (string): The first name of the applicant.
- LastName (string): The last name of the applicant.
- Email (string): The email address of the applicant.
- Phone (string): The phone number of the applicant.
- Resume (text): The applicant's resume or CV (text or file reference).

Table: Applications
Attributes:
- ApplicationID (Primary Key, int): Unique identifier for each job application.
- JobID (Foreign Key, int): References the JobID of the job listing.
- ApplicantID (Foreign Key, int): References the ApplicantID of the applicant.
- ApplicationDate (datetime): Date and time when the application was submitted.
- CoverLetter (text): The applicant's cover letter for the specific job.

Tasks:

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

```
Create database careerhub;
```

2. Create tables for Companies, Jobs, Applicants and Applications.
3. Define appropriate primary keys, foreign keys, and constraints.
4. Ensure the script handles potential errors, such as if the database or tables already exist.

Queries

```
CREATE TABLE Companies (
   CompanyID INT PRIMARY KEY ,
   CompanyName VARCHAR(255) NOT NULL,
   Location VARCHAR(255) NOT NULL
);


CREATE TABLE Jobs (
   JobID INT PRIMARY KEY ,
   CompanyID INT,
   JobTitle VARCHAR(255) NOT NULL,
   JobDescription TEXT NOT NULL,
   JobLocation VARCHAR(255) NOT NULL,
   Salary DECIMAL(10,2),
   JobType VARCHAR(50) NOT NULL CHECK (JobType IN ('Full-time', 'Part-time', 'Contract')),
   PostedDate DATETIME DEFAULT CURRENT_TIMESTAMP,
   FOREIGN KEY (CompanyID) REFERENCES Companies(CompanyID)
);
drop table applicants
CREATE TABLE Applicants (
   ApplicantID INT PRIMARY KEY ,
   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
ALTER COLUMN Phone VARCHAR(20);


CREATE TABLE Applications (
   ApplicationID INT PRIMARY KEY ,
   JobID INT,
   ApplicantID INT,
   ApplicationDate DATETIME DEFAULT CURRENT_TIMESTAMP,
   CoverLetter TEXT NOT NULL,
   FOREIGN KEY (JobID) REFERENCES Jobs(JobID) ,
   FOREIGN KEY (ApplicantID) REFERENCES Applicants(ApplicantID)
);
```

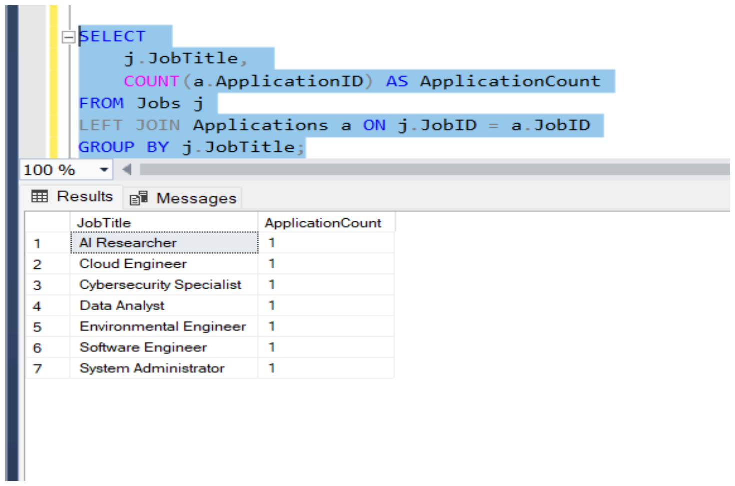
Insertion of records:

```
INSERT INTO Companies (CompanyID, CompanyName, Location) VALUES
(101, 'Tech Solutions', 'New York'),
(102, 'InnovateX', 'San Francisco'),
(103, 'DataVision', 'Chicago'),
(104, 'CloudNet', 'Seattle'),
(105, 'CyberCore', 'Austin'),
(106, 'NextGen AI', 'Boston'),
(107, 'GreenTech', 'Denver');
select * from Companies

INSERT INTO Jobs (JobID, CompanyID, JobTitle, JobDescription, JobLocation, Salary, JobType, PostedDate)
VALUES
(1, 101, 'Software Engineer', 'Develop and maintain software applications.', 'New York', 90000.00, 'Full-time',
CURRENT_TIMESTAMP),
(2, 102, 'Data Analyst', 'Analyze large datasets to provide insights.', 'San Francisco', 80000.00, 'Full-time',
CURRENT_TIMESTAMP),
(3, 103, 'System Administrator', 'Manage IT infrastructure and networks.', 'Chicago', 75000.00, 'Full-time',
CURRENT_TIMESTAMP),
(4, 104, 'Cloud Engineer', 'Design cloud-based solutions.', 'Seattle', 95000.00, 'Full-time',
CURRENT_TIMESTAMP),
(5, 105, 'Cybersecurity Specialist', 'Ensure system security and prevent threats.', 'Austin', 100000.00,
'Full-time', CURRENT_TIMESTAMP),
(6, 106, 'AI Researcher', 'Work on AI and machine learning models.', 'Boston', 110000.00, 'Full-time',
CURRENT_TIMESTAMP),
(7, 107, 'Environmental Engineer', 'Develop sustainable solutions.', 'Denver', 85000.00, 'Full-time',
CURRENT_TIMESTAMP);
Select * form jobs

INSERT INTO Applicants (ApplicantID, FirstName, LastName, Email, Phone, Resume) VALUES
(201, 'John', 'Doe', 'johndoe@example.com', '123-456-7890', 'Experienced software developer skilled in Java
and Python.'),
(202, 'Jane', 'Smith', 'janesmith@example.com', '234-567-8901', 'Data analyst with expertise in SQL, Python,
and Tableau.'),
(203, 'Michael', 'Brown', 'michaelbrown@example.com', '345-678-9012', 'System administrator with
hands-on experience in network security.'),
(204, 'Emily', 'Johnson', 'emilyjohnson@example.com', '456-789-0123', 'Cloud engineer with a passion for
AWS and DevOps practices.'),
(205, 'David', 'Williams', 'davidwilliams@example.com', '567-890-1234', 'Cybersecurity expert with
knowledge of ethical hacking and risk assessment.'),
(206, 'Sarah', 'Miller', 'sarahmiller@example.com', '678-901-2345', 'AI researcher focusing on deep learning
and NLP solutions.'),
(207, 'Robert', 'Davis', 'robertdavis@example.com', '789-012-3456', 'Environmental engineer working on
sustainable energy solutions.');

select * from Applicants

INSERT INTO Applications (ApplicationID, JobID, ApplicantID, ApplicationDate, CoverLetter) VALUES
(301, 1, 201, CURRENT_TIMESTAMP, 'I am excited to apply for the Software Engineer role. My experience in
Java and Python makes me a great fit.'),
(302, 2, 202, CURRENT_TIMESTAMP, 'I am passionate about data analytics and eager to contribute to your
team as a Data Analyst.'),
(303, 3, 203, CURRENT_TIMESTAMP, 'With years of experience in system administration, I am confident in my
ability to manage IT infrastructure effectively.'),
(304, 4, 204, CURRENT_TIMESTAMP, 'I have a strong background in cloud computing and DevOps, making me
the perfect candidate for this Cloud Engineer position.'),
(305, 5, 205, CURRENT_TIMESTAMP, 'Cybersecurity is my passion, and I am eager to help secure your systems
from potential threats.'),
```

(306, 6, 206, CURRENT_TIMESTAMP, 'As an AI researcher, I am thrilled to apply my deep learning expertise to your innovative projects.'),
(307, 7, 207, CURRENT_TIMESTAMP, 'I am enthusiastic about sustainable engineering and would love the opportunity to work on environmental solutions.');
Select * applicantions

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.

```
SELECT
   j.JobTitle,
   COUNT(a.ApplicationID) AS ApplicationCount
FROM Jobs j
LEFT JOIN Applications a ON j.JobID = a.JobID
GROUP BY j.JobTitle;
```

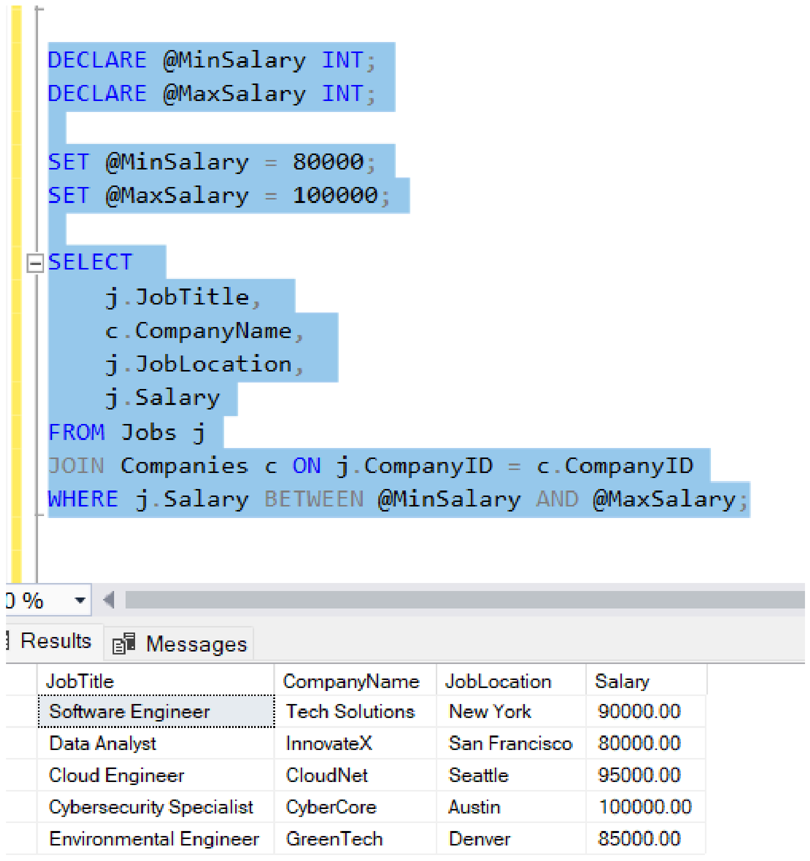


```
SELECT
    j.JobTitle,
    COUNT(a.ApplicationID) AS ApplicationCount
FROM Jobs j
LEFT JOIN Applications a ON j.JobID = a.JobID
GROUP BY j.JobTitle;
```

	JobTitle	ApplicationCount
1	AI Researcher	1
2	Cloud Engineer	1
3	Cybersecurity Specialist	1
4	Data Analyst	1
5	Environmental Engineer	1
6	Software Engineer	1
7	System Administrator	1

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.

```
DECLARE @MinSalary INT;
DECLARE @MaxSalary INT;

SET @MinSalary = 80000;
SET @MaxSalary = 100000;

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;
```

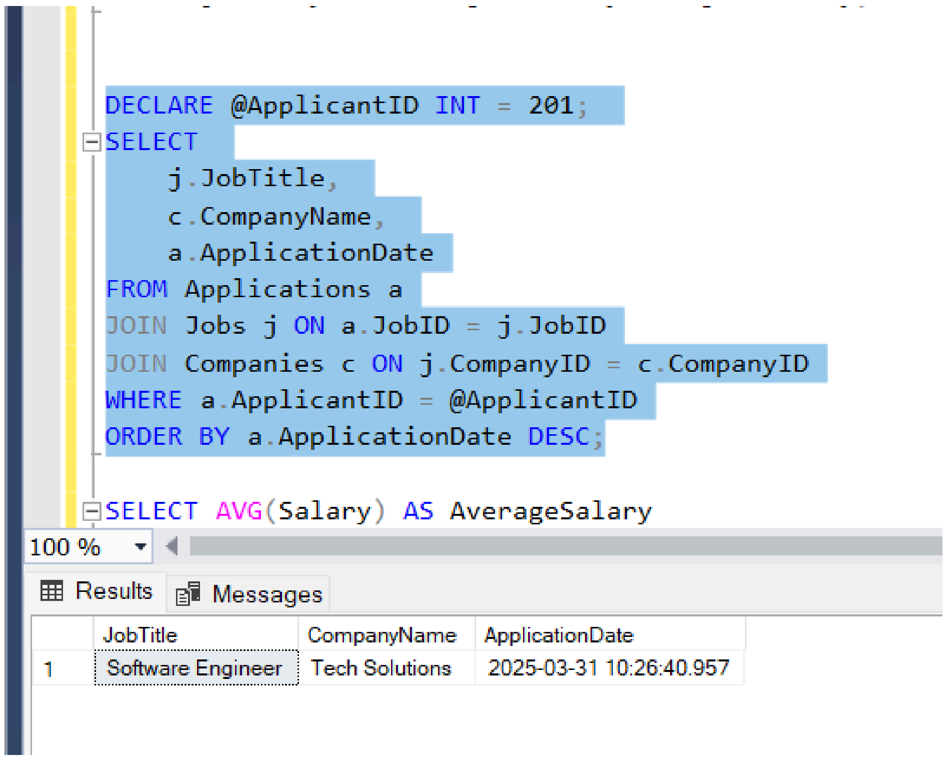


```sql
DECLARE @MinSalary INT;
DECLARE @MaxSalary INT;

SET @MinSalary = 80000;
SET @MaxSalary = 100000;

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;
```

JobTitle	CompanyName	JobLocation	Salary
Software Engineer	Tech Solutions	New York	90000.00
Data Analyst	InnovateX	San Francisco	80000.00
Cloud Engineer	CloudNet	Seattle	95000.00
Cybersecurity Specialist	CyberCore	Austin	100000.00
Environmental Engineer	GreenTech	Denver	85000.00

7. 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.

```sql
DECLARE @ApplicantID INT = 201;
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;
```

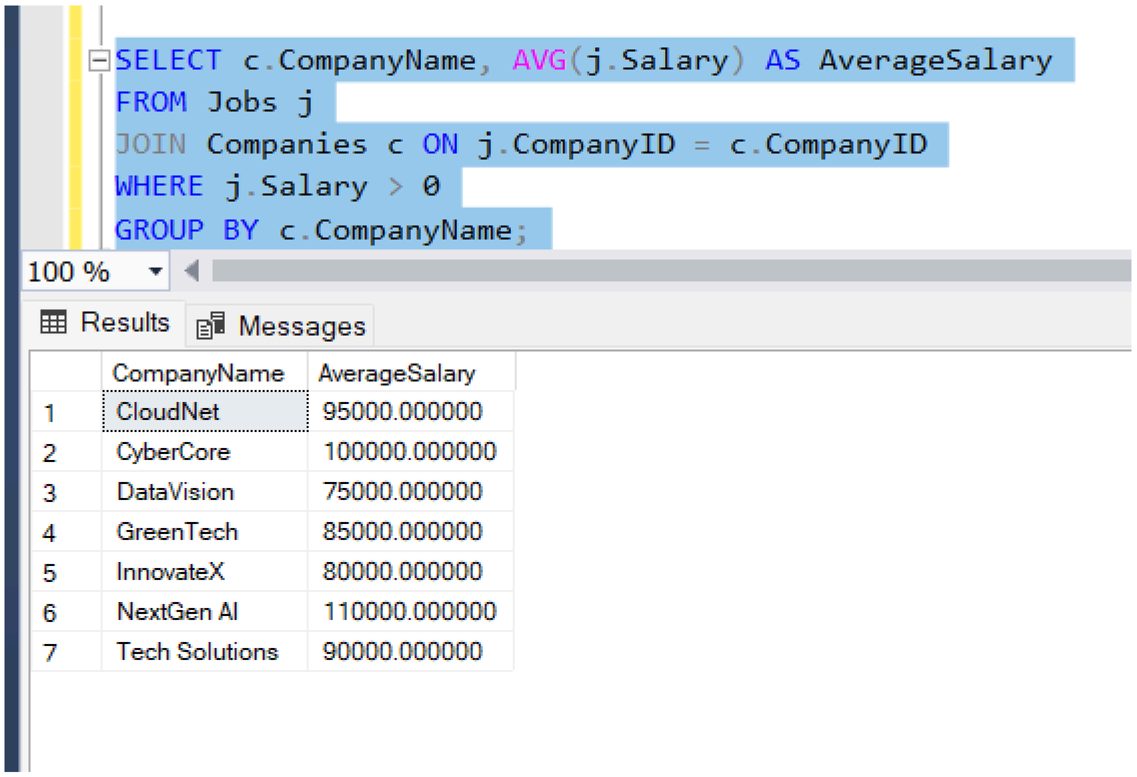


```sql
DECLARE @ApplicantID INT = 201;
SELECT
    j.JobTitle,
    c.CompanyName,
    a.ApplicationDate
FROM Applications a
JOIN Jobs j ON a.JobID = j.JobID
JOIN Companies c ON j.CompanyID = c.CompanyID
WHERE a.ApplicantID = @ApplicantID
ORDER BY a.ApplicationDate DESC;

SELECT AVG(Salary) AS AverageSalary
```

	JobTitle	CompanyName	ApplicationDate
1	Software Engineer	Tech Solutions	2025-03-31 10:26:40.957

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.

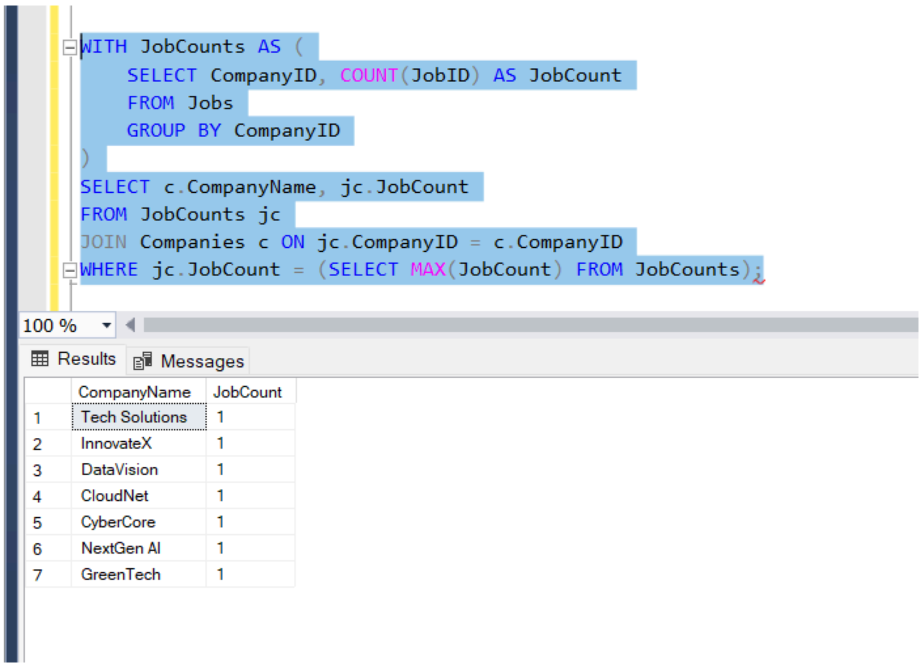
SELECT c.CompanyName, AVG(j.Salary) AS AverageSalary
FROM Jobs j
JOIN Companies c ON j.CompanyID = c.CompanyID
WHERE j.Salary > 0
GROUP BY c.CompanyName;



```sql
SELECT c.CompanyName, AVG(j.Salary) AS AverageSalary
FROM Jobs j
JOIN Companies c ON j.CompanyID = c.CompanyID
WHERE j.Salary > 0
GROUP BY c.CompanyName;
```

	CompanyName	AverageSalary
1	CloudNet	95000.000000
2	CyberCore	100000.000000
3	DataVision	75000.000000
4	GreenTech	85000.000000
5	InnovateX	80000.000000
6	NextGen AI	110000.000000
7	Tech Solutions	90000.000000

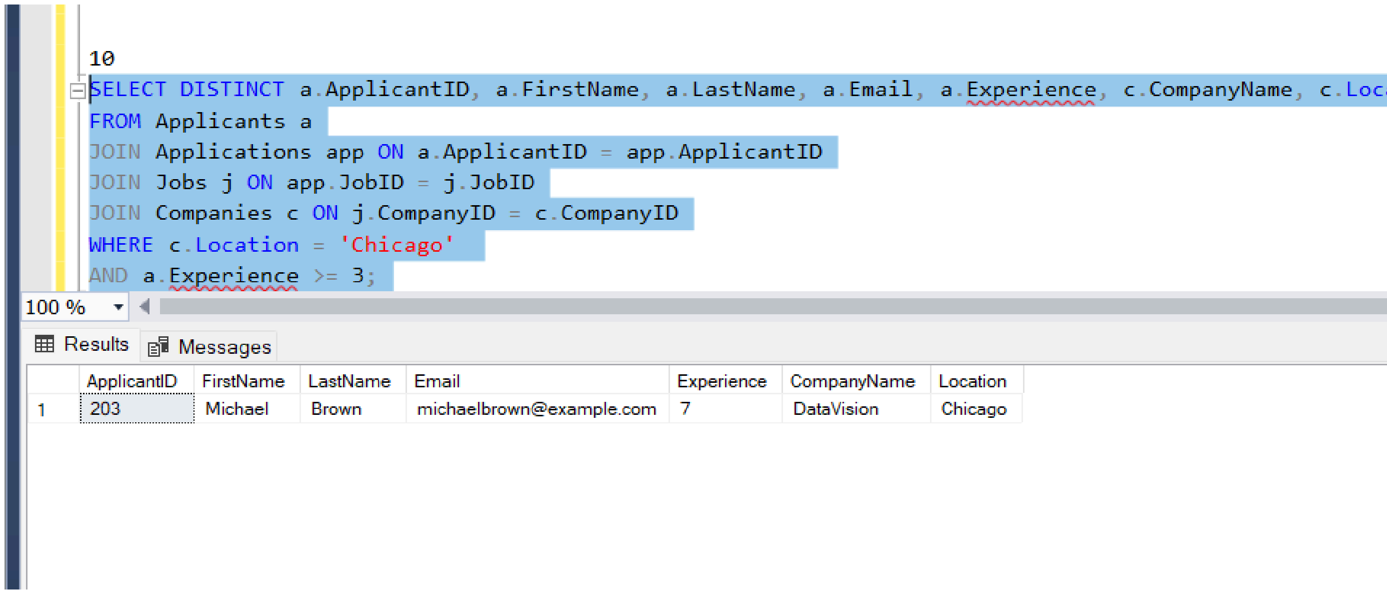
9. Write an SQL query to identify the company that has posted the most job listings. Display the company name along with the count of job listings they have posted. Handle ties if multiple


```sql
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);
```

```
   CompanyName     JobCount
1  Tech Solutions  1
2  InnovateX       1
3  DataVision      1
4  CloudNet        1
5  CyberCore       1
6  NextGen AI      1
7  GreenTech       1
```

companies have the same maximum count.

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

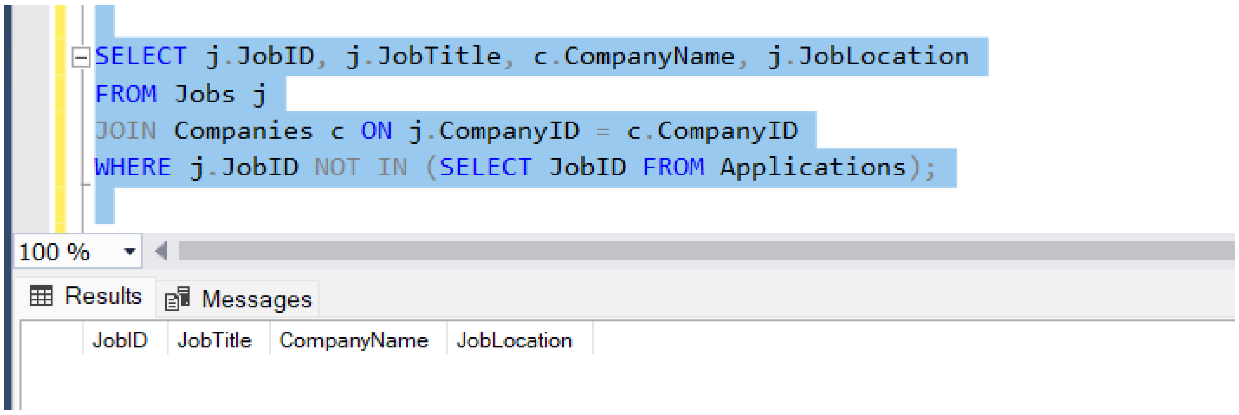

```sql
10
SELECT DISTINCT a.ApplicantID, a.FirstName, a.LastName, a.Email, a.Experience, c.CompanyName, c.Loc
FROM Applicants a
JOIN Applications app ON a.ApplicantID = app.ApplicantID
JOIN Jobs j ON app.JobID = j.JobID
JOIN Companies c ON j.CompanyID = c.CompanyID
WHERE c.Location = 'Chicago'
AND a.Experience >= 3;
```

```
   ApplicantID  FirstName  LastName  Email                      Experience  CompanyName  Location
1  203          Michael    Brown     michaelbrown@example.com   7           DataVision   Chicago
```

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

select distinct jobtitle from jobs
where salary between 60000 and 80000;

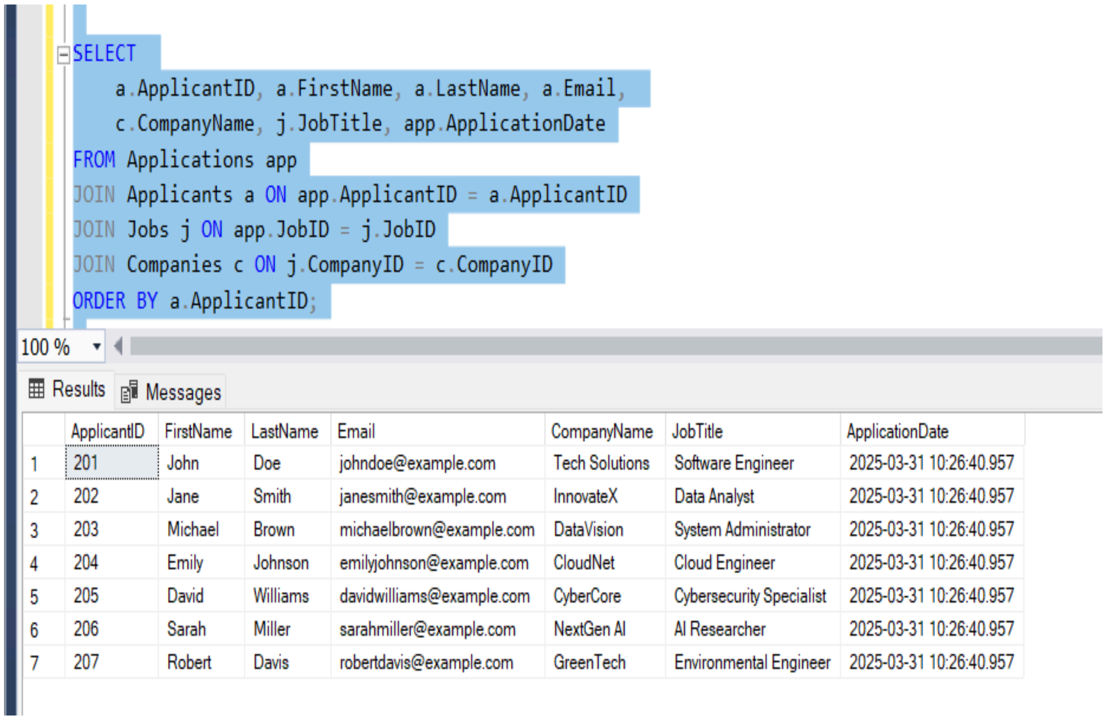
12. Find the jobs that have not received any applications.



```
SELECT j.JobID, j.JobTitle, c.CompanyName, j.JobLocation
FROM Jobs j
JOIN Companies c ON j.CompanyID = c.CompanyID
WHERE j.JobID NOT IN (SELECT JobID FROM Applications);
```

JobID	JobTitle	CompanyName	JobLocation

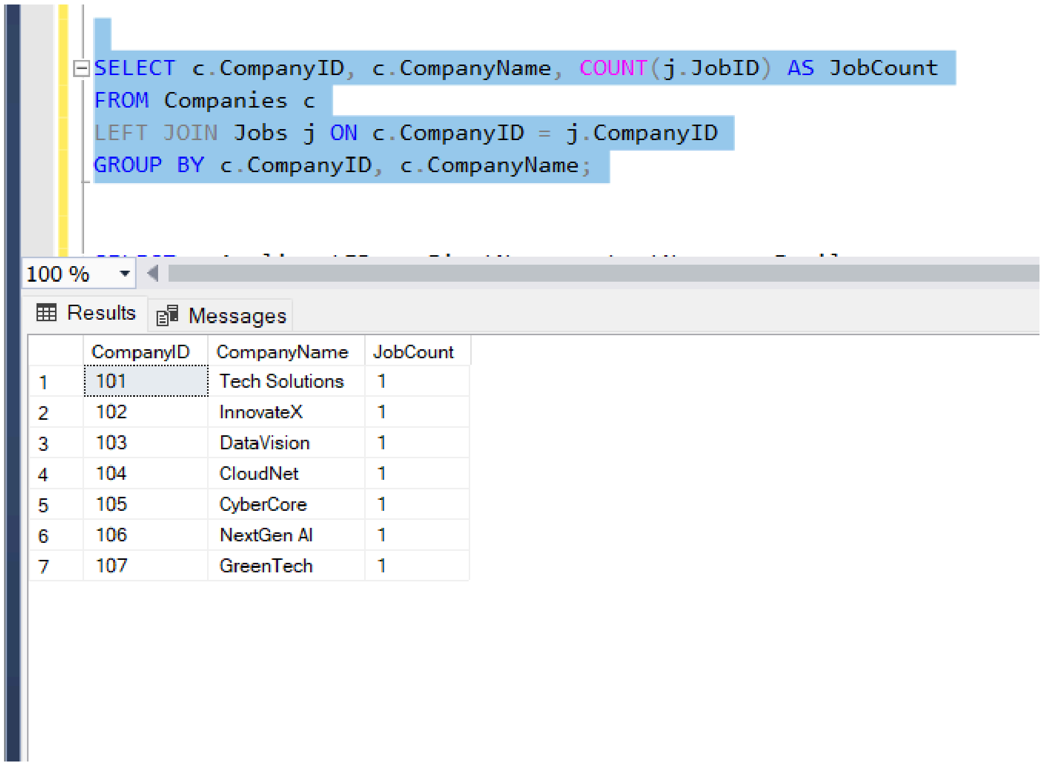
13. Retrieve a list of job applicants along with the companies they have applied to and the positions they have applied for.



```
SELECT
    a.ApplicantID, a.FirstName, a.LastName, a.Email,
    c.CompanyName, j.JobTitle, app.ApplicationDate
FROM Applications app
JOIN Applicants a ON app.ApplicantID = a.ApplicantID
JOIN Jobs j ON app.JobID = j.JobID
JOIN Companies c ON j.CompanyID = c.CompanyID
ORDER BY a.ApplicantID;
```

	ApplicantID	FirstName	LastName	Email	CompanyName	JobTitle	ApplicationDate
1	201	John	Doe	johndoe@example.com	Tech Solutions	Software Engineer	2025-03-31 10:26:40.957
2	202	Jane	Smith	janesmith@example.com	InnovateX	Data Analyst	2025-03-31 10:26:40.957
3	203	Michael	Brown	michaelbrown@example.com	DataVision	System Administrator	2025-03-31 10:26:40.957
4	204	Emily	Johnson	emilyjohnson@example.com	CloudNet	Cloud Engineer	2025-03-31 10:26:40.957
5	205	David	Williams	davidwilliams@example.com	CyberCore	Cybersecurity Specialist	2025-03-31 10:26:40.957
6	206	Sarah	Miller	sarahmiller@example.com	NextGen AI	AI Researcher	2025-03-31 10:26:40.957
7	207	Robert	Davis	robertdavis@example.com	GreenTech	Environmental Engineer	2025-03-31 10:26:40.957

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



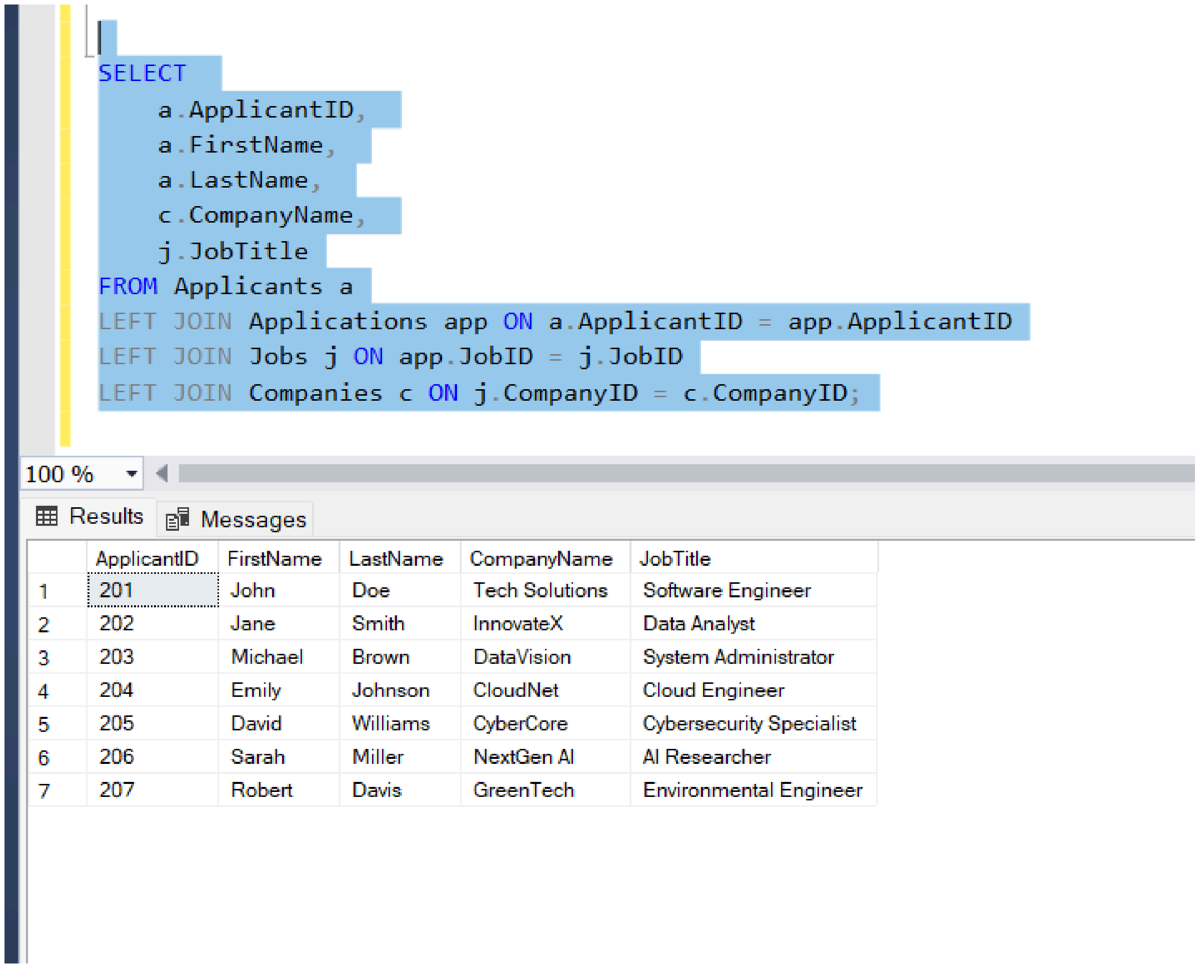
```
SELECT c.CompanyID, c.CompanyName, COUNT(j.JobID) AS JobCount
FROM Companies c
LEFT JOIN Jobs j ON c.CompanyID = j.CompanyID
GROUP BY c.CompanyID, c.CompanyName;
```

100 %

Results | Messages

	CompanyID	CompanyName	JobCount
1	101	Tech Solutions	1
2	102	InnovateX	1
3	103	DataVision	1
4	104	CloudNet	1
5	105	CyberCore	1
6	106	NextGen AI	1
7	107	GreenTech	1

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



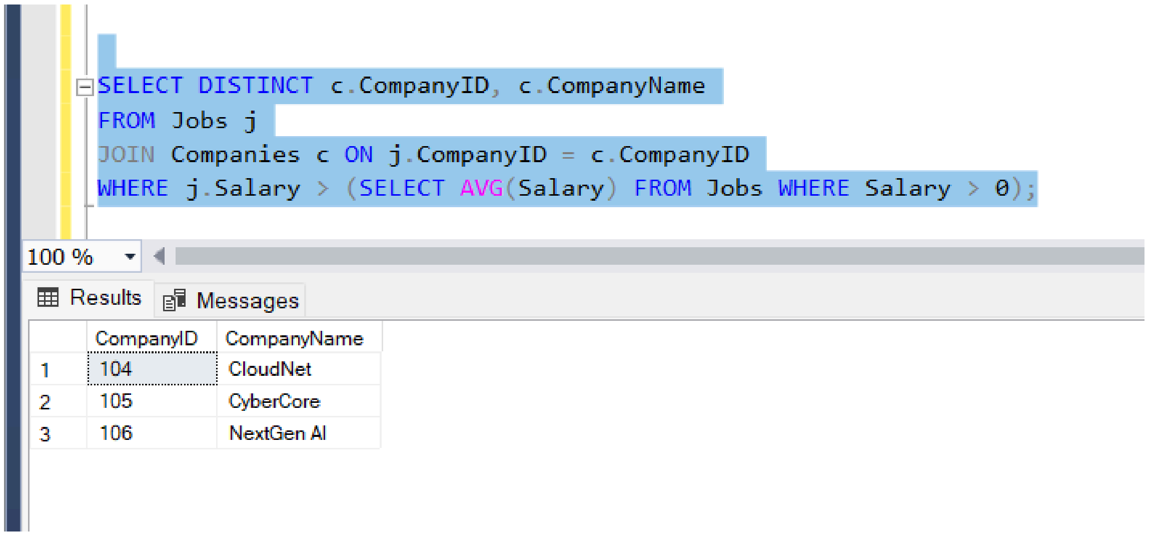
```
SELECT
    a.ApplicantID,
    a.FirstName,
    a.LastName,
    c.CompanyName,
    j.JobTitle
FROM Applicants a
LEFT JOIN Applications app ON a.ApplicantID = app.ApplicantID
LEFT JOIN Jobs j ON app.JobID = j.JobID
LEFT JOIN Companies c ON j.CompanyID = c.CompanyID;
```

100 %

Results | Messages

	ApplicantID	FirstName	LastName	CompanyName	JobTitle
1	201	John	Doe	Tech Solutions	Software Engineer
2	202	Jane	Smith	InnovateX	Data Analyst
3	203	Michael	Brown	DataVision	System Administrator
4	204	Emily	Johnson	CloudNet	Cloud Engineer
5	205	David	Williams	CyberCore	Cybersecurity Specialist
6	206	Sarah	Miller	NextGen AI	AI Researcher
7	207	Robert	Davis	GreenTech	Environmental Engineer

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



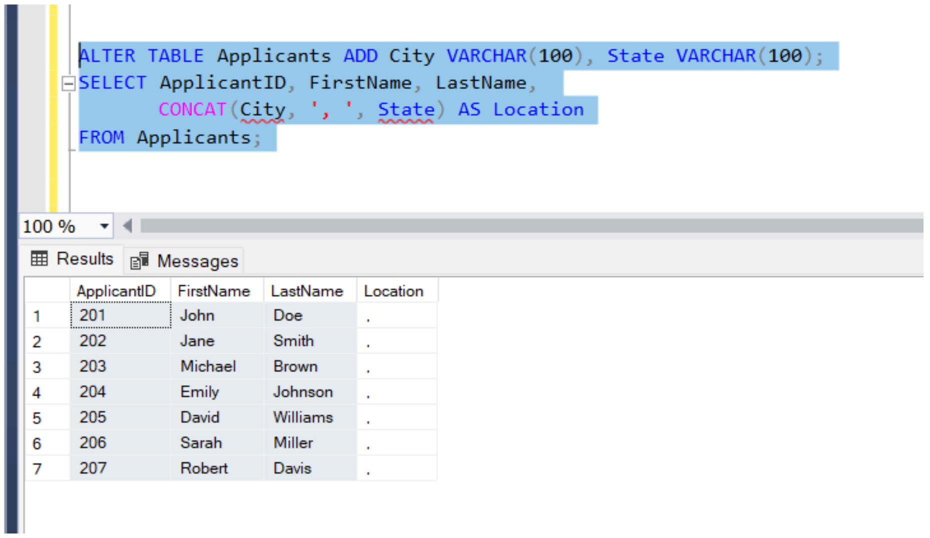
```sql
SELECT DISTINCT c.CompanyID, c.CompanyName
FROM Jobs j
JOIN Companies c ON j.CompanyID = c.CompanyID
WHERE j.Salary > (SELECT AVG(Salary) FROM Jobs WHERE Salary > 0);
```

100 %

Results | Messages

	CompanyID	CompanyName
1	104	CloudNet
2	105	CyberCore
3	106	NextGen AI

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



```sql
ALTER TABLE Applicants ADD City VARCHAR(100), State VARCHAR(100);
SELECT ApplicantID, FirstName, LastName,
       CONCAT(City, ', ', State) AS Location
FROM Applicants;
```

100 %

Results | Messages

	ApplicantID	FirstName	LastName	Location
1	201	John	Doe	,
2	202	Jane	Smith	,
3	203	Michael	Brown	,
4	204	Emily	Johnson	,
5	205	David	Williams	,
6	206	Sarah	Miller	,
7	207	Robert	Davis	,

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

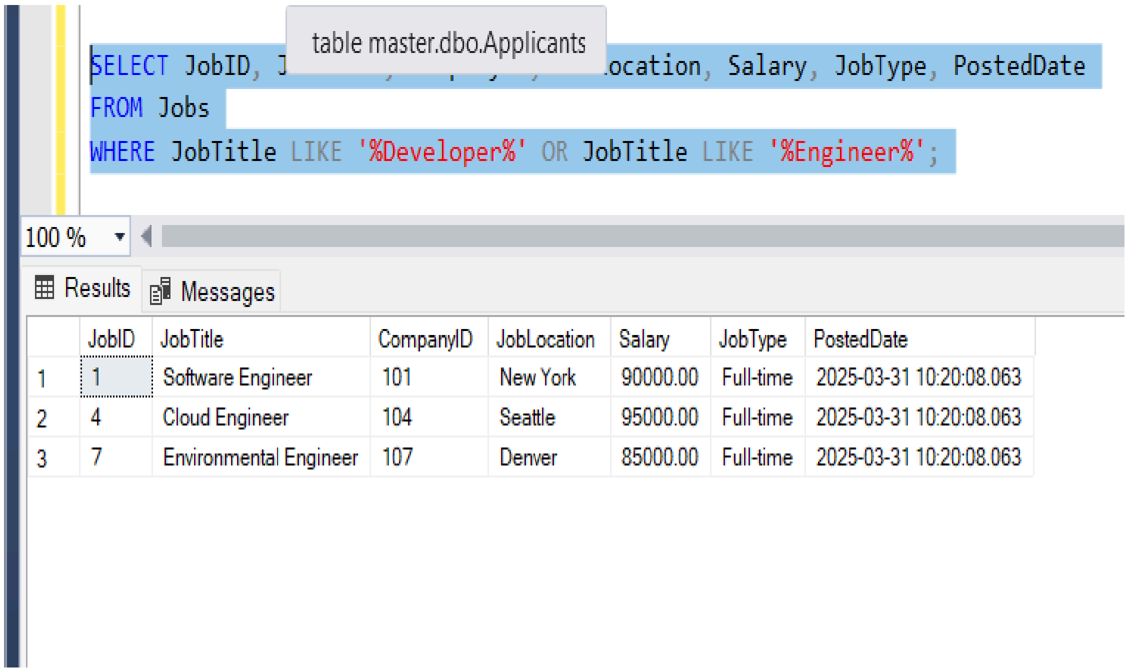


table master.dbo.Applicants

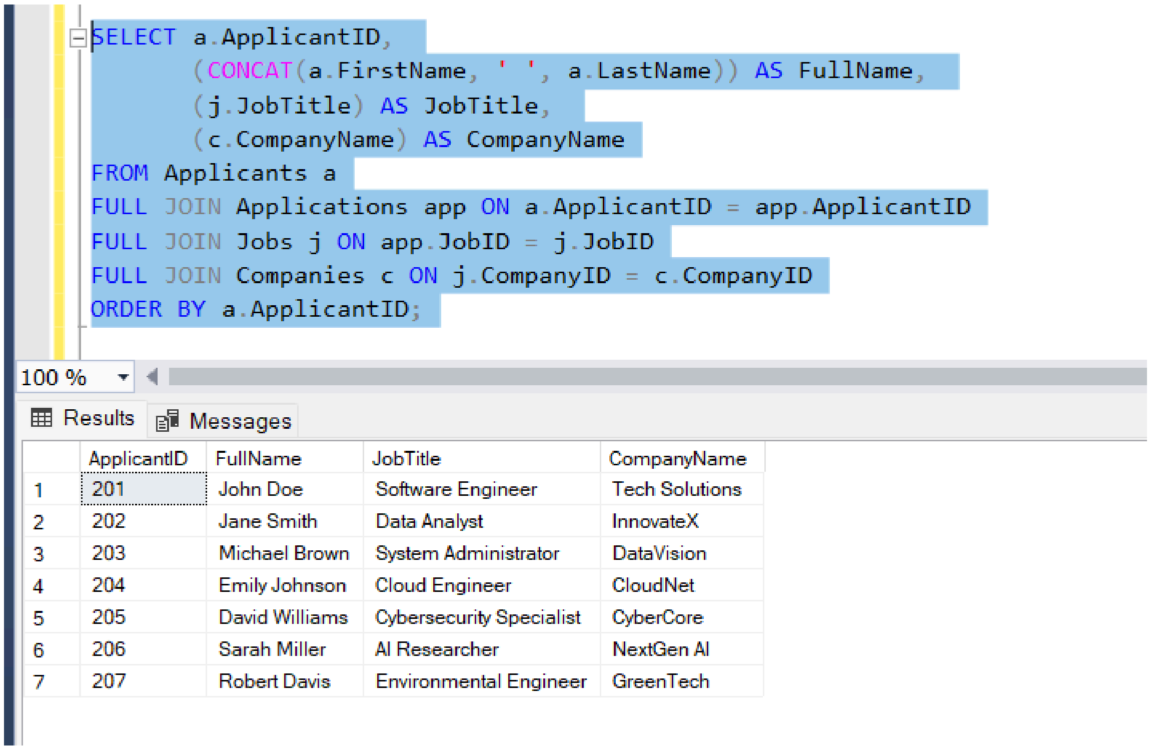
```sql
SELECT JobID, J                      ocation, Salary, JobType, PostedDate
FROM Jobs
WHERE JobTitle LIKE '%Developer%' OR JobTitle LIKE '%Engineer%';
```

100 %

Results | Messages

	JobID	JobTitle	CompanyID	JobLocation	Salary	JobType	PostedDate
1	1	Software Engineer	101	New York	90000.00	Full-time	2025-03-31 10:20:08.063
2	4	Cloud Engineer	104	Seattle	95000.00	Full-time	2025-03-31 10:20:08.063
3	7	Environmental Engineer	107	Denver	85000.00	Full-time	2025-03-31 10:20:08.063

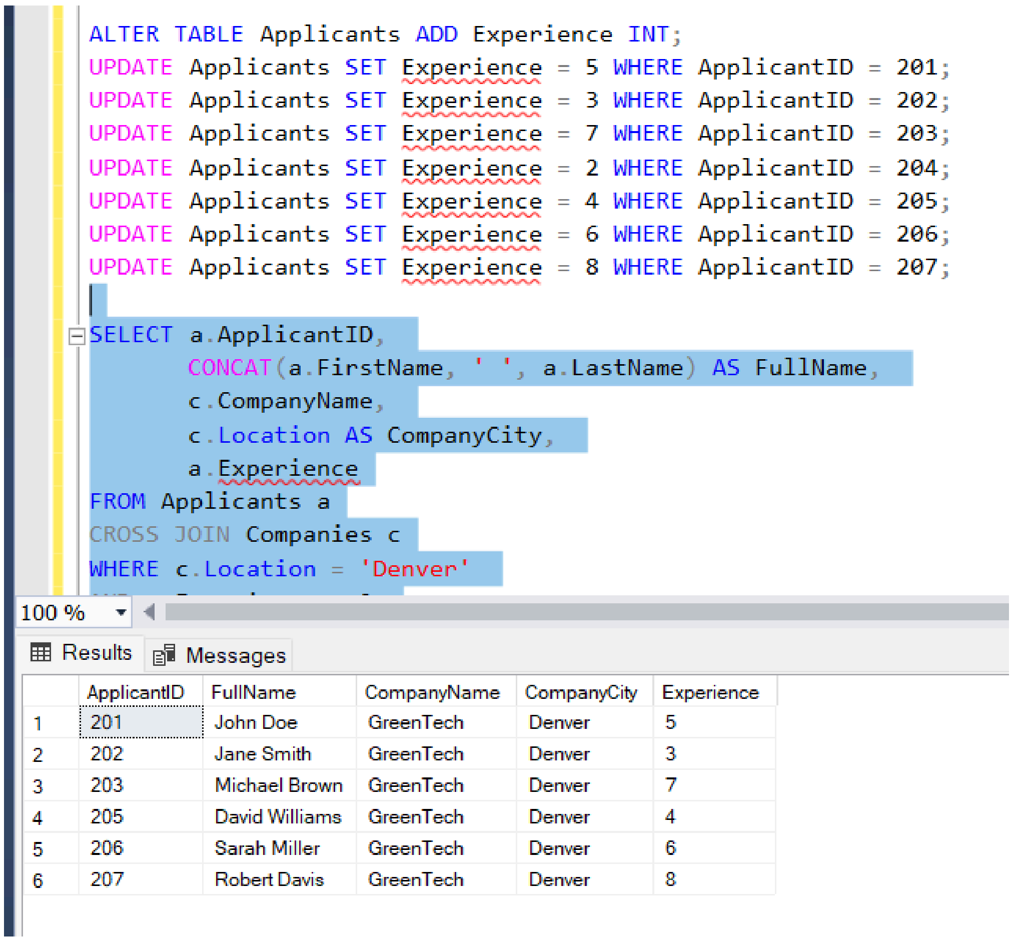
19. Retrieve a list of applicants and the jobs they have applied for, including those who have not applied and jobs without applicants.



```sql
SELECT a.ApplicantID,
       (CONCAT(a.FirstName, ' ', a.LastName)) AS FullName,
       (j.JobTitle) AS JobTitle,
       (c.CompanyName) AS CompanyName
FROM Applicants a
FULL JOIN Applications app ON a.ApplicantID = app.ApplicantID
FULL JOIN Jobs j ON app.JobID = j.JobID
FULL JOIN Companies c ON j.CompanyID = c.CompanyID
ORDER BY a.ApplicantID;
```

	ApplicantID	FullName	JobTitle	CompanyName
1	201	John Doe	Software Engineer	Tech Solutions
2	202	Jane Smith	Data Analyst	InnovateX
3	203	Michael Brown	System Administrator	DataVision
4	204	Emily Johnson	Cloud Engineer	CloudNet
5	205	David Williams	Cybersecurity Specialist	CyberCore
6	206	Sarah Miller	AI Researcher	NextGen AI
7	207	Robert Davis	Environmental Engineer	GreenTech

20. 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=denver



```sql
ALTER TABLE Applicants ADD Experience INT;
UPDATE Applicants SET Experience = 5 WHERE ApplicantID = 201;
UPDATE Applicants SET Experience = 3 WHERE ApplicantID = 202;
UPDATE Applicants SET Experience = 7 WHERE ApplicantID = 203;
UPDATE Applicants SET Experience = 2 WHERE ApplicantID = 204;
UPDATE Applicants SET Experience = 4 WHERE ApplicantID = 205;
UPDATE Applicants SET Experience = 6 WHERE ApplicantID = 206;
UPDATE Applicants SET Experience = 8 WHERE ApplicantID = 207;

SELECT a.ApplicantID,
       CONCAT(a.FirstName, ' ', a.LastName) AS FullName,
       c.CompanyName,
       c.Location AS CompanyCity,
       a.Experience
FROM Applicants a
CROSS JOIN Companies c
WHERE c.Location = 'Denver'
```

	ApplicantID	FullName	CompanyName	CompanyCity	Experience
1	201	John Doe	GreenTech	Denver	5
2	202	Jane Smith	GreenTech	Denver	3
3	203	Michael Brown	GreenTech	Denver	7
4	205	David Williams	GreenTech	Denver	4
5	206	Sarah Miller	GreenTech	Denver	6
6	207	Robert Davis	GreenTech	Denver	8