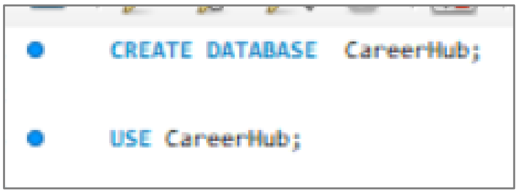
MySQL Coding Challenge – CareerHub, The Job Board

Name: Tejsinh J. Harale-Bhosale

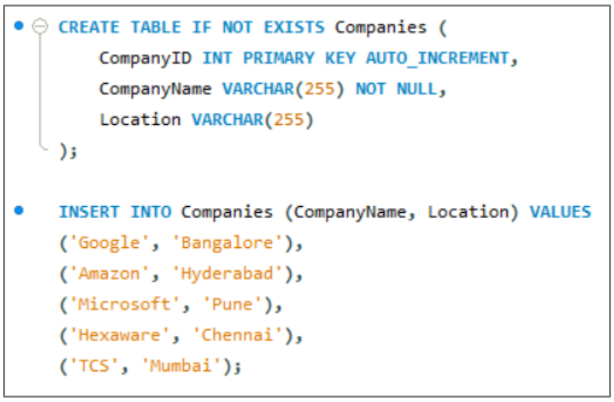
Github Repository: https://github.com/tejsinh3600/DA_foundation

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

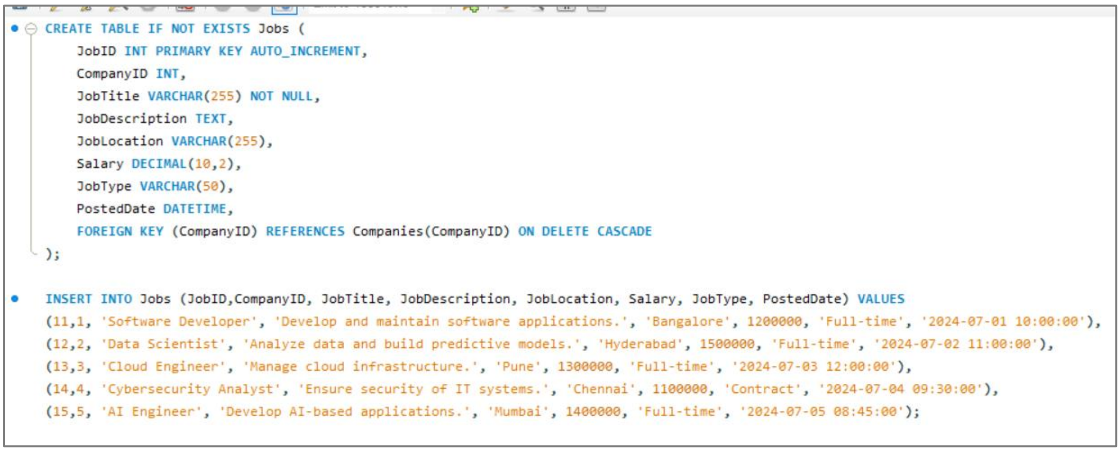


```sql
CREATE DATABASE CareerHub;

USE CareerHub;
```

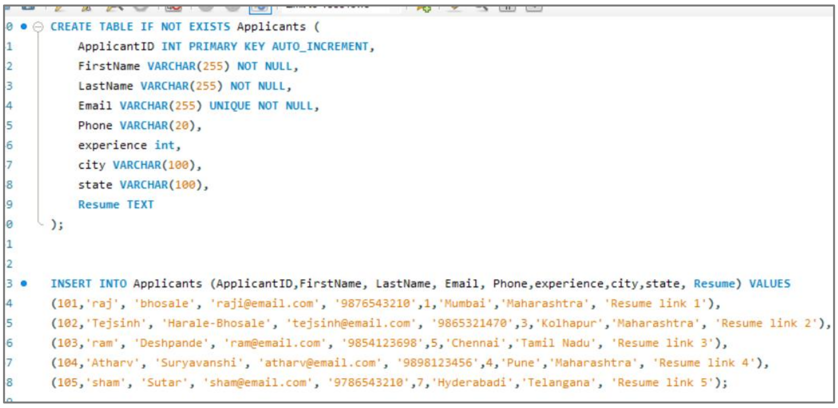
2. Create tables for Companies, Jobs, Applicants and Applications.
3. Define appropriate primary keys, foreign keys, and constraints.
4. Ensure the script handles potential errors, such as if the database or tables already exist.



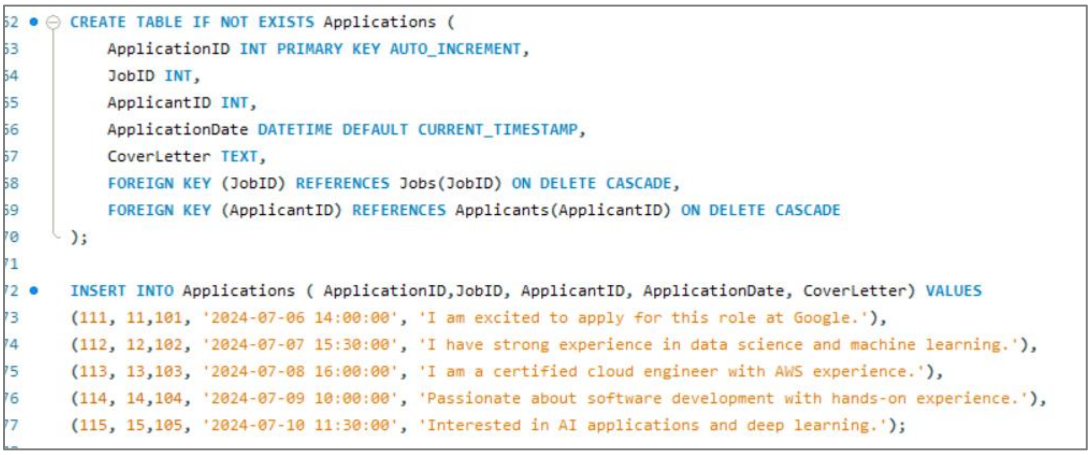
```sql
CREATE TABLE IF NOT EXISTS Companies (
    CompanyID INT PRIMARY KEY AUTO_INCREMENT,
    CompanyName VARCHAR(255) NOT NULL,
    Location VARCHAR(255)
);

INSERT INTO Companies (CompanyName, Location) VALUES
('Google', 'Bangalore'),
('Amazon', 'Hyderabad'),
('Microsoft', 'Pune'),
('Hexaware', 'Chennai'),
('TCS', 'Mumbai');
```



```sql
CREATE TABLE IF NOT EXISTS Jobs (
    JobID INT PRIMARY KEY AUTO_INCREMENT,
    CompanyID INT,
    JobTitle VARCHAR(255) NOT NULL,
    JobDescription TEXT,
    JobLocation VARCHAR(255),
    Salary DECIMAL(10,2),
    JobType VARCHAR(50),
    PostedDate DATETIME,
    FOREIGN KEY (CompanyID) REFERENCES Companies(CompanyID) ON DELETE CASCADE
);

INSERT INTO Jobs (JobID,CompanyID, JobTitle, JobDescription, JobLocation, Salary, JobType, PostedDate) VALUES
(11,1, 'Software Developer', 'Develop and maintain software applications.', 'Bangalore', 1200000, 'Full-time', '2024-07-01 10:00:00'),
(12,2, 'Data Scientist', 'Analyze data and build predictive models.', 'Hyderabad', 1500000, 'Full-time', '2024-07-02 11:00:00'),
(13,3, 'Cloud Engineer', 'Manage cloud infrastructure.', 'Pune', 1300000, 'Full-time', '2024-07-03 12:00:00'),
(14,4, 'Cybersecurity Analyst', 'Ensure security of IT systems.', 'Chennai', 1100000, 'Contract', '2024-07-04 09:30:00'),
(15,5, 'AI Engineer', 'Develop AI-based applications.', 'Mumbai', 1400000, 'Full-time', '2024-07-05 08:45:00');
```



```
CREATE TABLE IF NOT EXISTS Applicants (
    ApplicantID INT PRIMARY KEY AUTO_INCREMENT,
    FirstName VARCHAR(255) NOT NULL,
    LastName VARCHAR(255) NOT NULL,
    Email VARCHAR(255) UNIQUE NOT NULL,
    Phone VARCHAR(20),
    experience int,
    city VARCHAR(100),
    state VARCHAR(100),
    Resume TEXT
);


INSERT INTO Applicants (ApplicantID,FirstName, LastName, Email, Phone,experience,city,state, Resume) VALUES
(101,'raj', 'bhosale', 'raji@email.com', '9876543210',1,'Mumbai','Maharashtra', 'Resume link 1'),
(102,'Tejsinh', 'Harale-Bhosale', 'tejsinh@email.com', '9865321470',3,'Kolhapur','Maharashtra', 'Resume link 2'),
(103,'ram', 'Deshpande', 'ram@email.com', '9854123698',5,'Chennai','Tamil Nadu', 'Resume link 3'),
(104,'Atharv', 'Suryavanshi', 'atharv@email.com', '9898123456',4,'Pune','Maharashtra', 'Resume link 4'),
(105,'sham', 'Sutar', 'sham@email.com', '9786543210',7,'Hyderabadi','Telangana', 'Resume link 5');
```



```
CREATE TABLE IF NOT EXISTS Applications (
    ApplicationID INT PRIMARY KEY AUTO_INCREMENT,
    JobID INT,
    ApplicantID INT,
    ApplicationDate DATETIME DEFAULT CURRENT_TIMESTAMP,
    CoverLetter TEXT,
    FOREIGN KEY (JobID) REFERENCES Jobs(JobID) ON DELETE CASCADE,
    FOREIGN KEY (ApplicantID) REFERENCES Applicants(ApplicantID) ON DELETE CASCADE
);

INSERT INTO Applications ( ApplicationID,JobID, ApplicantID, ApplicationDate, CoverLetter) VALUES
(111, 11,101, '2024-07-06 14:00:00', 'I am excited to apply for this role at Google.'),
(112, 12,102, '2024-07-07 15:30:00', 'I have strong experience in data science and machine learning.'),
(113, 13,103, '2024-07-08 16:00:00', 'I am a certified cloud engineer with AWS experience.'),
(114, 14,104, '2024-07-09 10:00:00', 'Passionate about software development with hands-on experience.'),
(115, 15,105, '2024-07-10 11:30:00', 'Interested in AI applications and deep learning.');
```

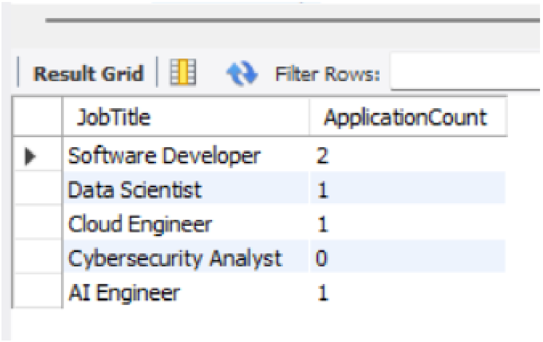
5. Write an SQL query to count the number of applications received for each job listing in the "Jobs" table. Display the job title and the corresponding application count. Ensure that it lists all jobs, even if they have no applications.

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

FROM Jobs j

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

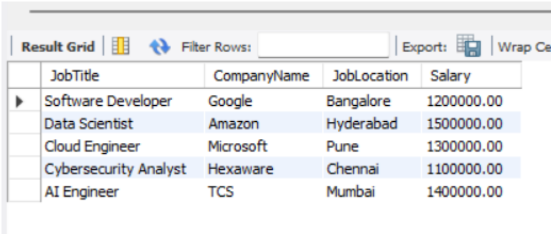
GROUP BY j.JobTitle;



JobTitle	ApplicationCount
Software Developer	2
Data Scientist	1
Cloud Engineer	1
Cybersecurity Analyst	0
AI Engineer	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.

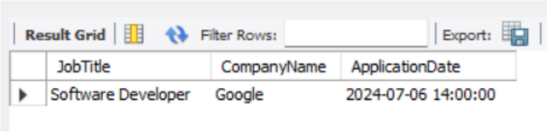
```
SELECT j.JobTitle, c.CompanyName, j.JobLocation,
j.Salary

FROM Jobs j

JOIN Companies c ON j.CompanyID = c.CompanyID

WHERE j.Salary BETWEEN 600000 AND 1500000;
```



JobTitle	CompanyName	JobLocation	Salary
Software Developer	Google	Bangalore	1200000.00
Data Scientist	Amazon	Hyderabad	1500000.00
Cloud Engineer	Microsoft	Pune	1300000.00
Cybersecurity Analyst	Hexaware	Chennai	1100000.00
AI Engineer	TCS	Mumbai	1400000.00

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.

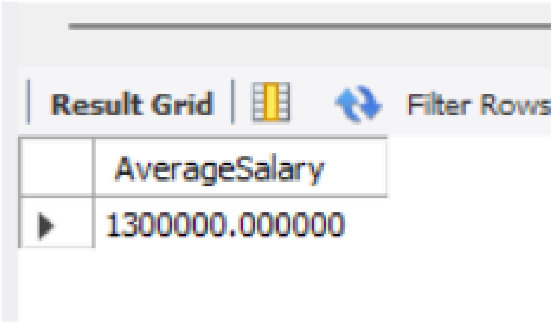
```
SELECT j.JobTitle, c.CompanyName, a.ApplicationDate

FROM Applications a

JOIN Jobs j ON a.JobID = j.JobID

JOIN Companies c ON j.CompanyID = c.CompanyID

WHERE a.ApplicantID = 1;
```



JobTitle	CompanyName	ApplicationDate
Software Developer	Google	2024-07-06 14:00:00

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.

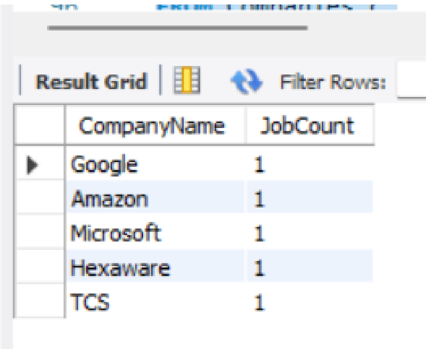
```
SELECT AVG(Salary) AS AverageSalary

FROM Jobs

WHERE Salary > 0;
```



	AverageSalary
▶	1300000.000000

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 companies have the same maximum count.

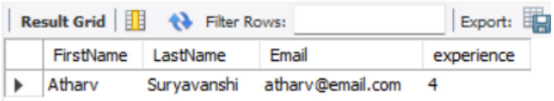
```
SELECT c.CompanyName, COUNT(j.JobID) AS JobCount

FROM Companies c

JOIN Jobs j ON c.CompanyID = j.CompanyID

GROUP BY c.CompanyName

HAVING JobCount = (

  SELECT MAX(JobCount)

  FROM (

    SELECT COUNT(JobID) AS JobCount

    FROM Jobs

    GROUP BY CompanyID

  ) AS Sub

);
```



	CompanyName	JobCount
▶	Google	1
	Amazon	1
	Microsoft	1
	Hexaware	1
	TCS	1

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

```
SELECT DISTINCT ap.FirstName, ap.LastName, ap.Email,
ap.experience

FROM Applications a

JOIN Jobs j ON a.JobID = j.JobID

JOIN Companies c ON j.CompanyID = c.CompanyID

JOIN Applicants ap ON ap.ApplicantID = a.ApplicantID

WHERE c.Location = 'Chennai' AND ap.experience >= 3;
```

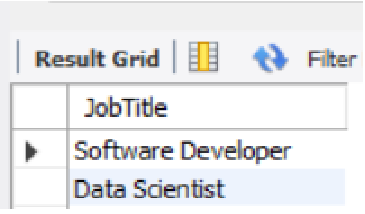


FirstName	LastName	Email	experience
Atharv	Suryavanshi	atharv@email.com	4

11.

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

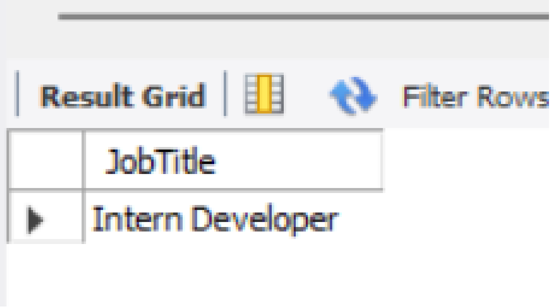
```
SELECT DISTINCT JobTitle

FROM Jobs

WHERE Salary BETWEEN 60000 AND 80000;
```



JobTitle
Software Developer
Data Scientist

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

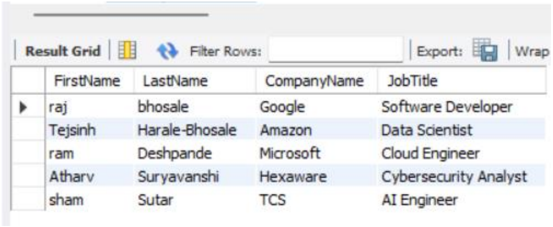
```
SELECT JobTitle

FROM Jobs

WHERE JobId NOT IN (SELECT DISTINCT JobID FROM
Applications);
```



JobTitle
Intern Developer

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

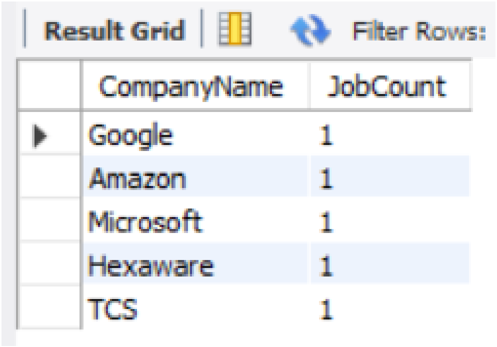
```
SELECT a.FirstName, a.LastName, c.CompanyName,
j.JobTitle

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;
```



FirstName	LastName	CompanyName	JobTitle
raj	bhosale	Google	Software Developer
Tejsinh	Harale-Bhosale	Amazon	Data Scientist
ram	Deshpande	Microsoft	Cloud Engineer
Atharv	Suryavanshi	Hexaware	Cybersecurity Analyst
sham	Sutar	TCS	AI Engineer

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

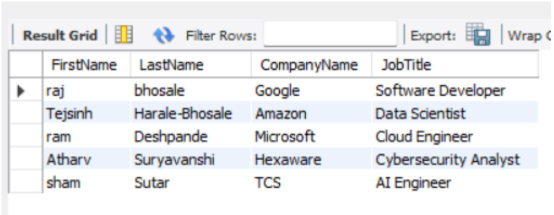
```
SELECT c.CompanyName, COUNT(j.JobID) AS JobCount

FROM Companies c

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

GROUP BY c.CompanyName;
```



CompanyName	JobCount
Google	1
Amazon	1
Microsoft	1
Hexaware	1
TCS	1

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

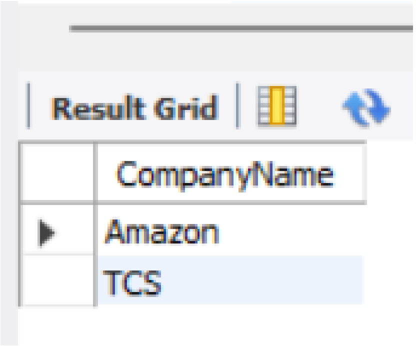
```
SELECT a.FirstName, a.LastName, c.CompanyName,
j.JobTitle

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;
```



FirstName	LastName	CompanyName	JobTitle
raj	bhosale	Google	Software Developer
Tejsinh	Harale-Bhosale	Amazon	Data Scientist
ram	Deshpande	Microsoft	Cloud Engineer
Atharv	Suryavanshi	Hexaware	Cybersecurity Analyst
sham	Sutar	TCS	AI Engineer

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

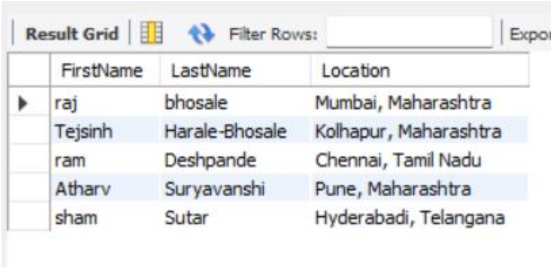
```
SELECT DISTINCT c.CompanyName

FROM Companies c

JOIN Jobs j ON c.CompanyID = j.CompanyID

WHERE j.Salary > (SELECT AVG(Salary) FROM Jobs
WHERE Salary > 0);
```



CompanyName
Amazon
TCS

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

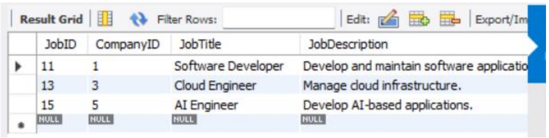
```
SELECT FirstName, LastName, CONCAT(City, ', ', State) AS Location

FROM Applicants;
```



FirstName	LastName	Location
raj	bhosale	Mumbai, Maharashtra
Tejsinh	Harale-Bhosale	Kolhapur, Maharashtra
ram	Deshpande	Chennai, Tamil Nadu
Atharv	Suryavanshi	Pune, Maharashtra
sham	Sutar	Hyderabadi, Telangana

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

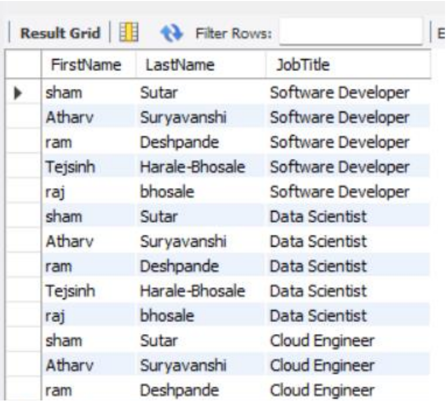
```
SELECT *FROM Jobs

WHERE JobTitle LIKE '%Developer%' OR JobTitle LIKE '%Engineer%';
```



JobID	CompanyID	JobTitle	JobDescription
11	1	Software Developer	Develop and maintain software applicatio
13	3	Cloud Engineer	Manage cloud infrastructure.
15	5	AI Engineer	Develop AI-based applications.
NULL	NULL	NULL	NULL

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

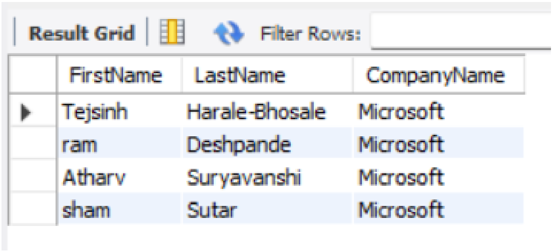
```
SELECT a.FirstName, a.LastName, j.JobTitle

FROM Applicants a

CROSS JOIN Jobs j

LEFT JOIN Applications app

  ON a.ApplicantID = app.ApplicantID AND j.JobID = app.JobID;
```



FirstName	LastName	JobTitle
sham	Sutar	Software Developer
Atharv	Suryavanshi	Software Developer
ram	Deshpande	Software Developer
Tejsinh	Harale-Bhosale	Software Developer
raj	bhosale	Software Developer
sham	Sutar	Data Scientist
Atharv	Suryavanshi	Data Scientist
ram	Deshpande	Data Scientist
Tejsinh	Harale-Bhosale	Data Scientist
raj	bhosale	Data Scientist
sham	Sutar	Cloud Engineer
Atharv	Suryavanshi	Cloud Engineer
ram	Deshpande	Cloud Engineer

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=Chennai

```
SELECT a.FirstName, a.LastName, c.CompanyName

FROM Applicants a

JOIN Companies c ON c.Location = 'Pune'

WHERE a.Experience > 2;
```



FirstName	LastName	CompanyName
Tejsinh	Harale-Bhosale	Microsoft
ram	Deshpande	Microsoft
Atharv	Suryavanshi	Microsoft
sham	Sutar	Microsoft