## MySQL Coding Challenge - CareerHub, The Job Board

Name: Atharva Padawale

Github Repository: <a href="https://github.com/AtharvaPadawale/hexaware-Java-Batch-5">https://github.com/AtharvaPadawale/hexaware-Java-Batch-5</a>

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

```
-- CarrerHub The Job Board

CREATE DATABASE IF NOT EXISTS 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.

```
CREATE TABLE IF NOT EXISTS Companies (
CompanyID INT PRIMARY KEY,
CompanyName VARCHAR(50),
Location VARCHAR(100)
);

INSERT INTO Companies (CompanyID, CompanyName, Location) VALUES
(1, 'Google', 'Bangalore'),
(2, 'Amazon', 'Hyderabad'),
(3, 'TechMahindra', 'Pune'),
(4, 'Hexaware', 'Chennai'),
(5, 'TCS', 'Pune');
```

```
○ CREATE TABLE IF NOT EXISTS Jobs (
      JobID INT PRIMARY KEY,
      CompanyID INT,
      JobTitle VARCHAR(50),
      JobDescription TEXT,
      JobLocation VARCHAR(50),
      Salary DECIMAL(10,2),
      JobType VARCHAR(50),
      PostedDate DATETIME,
      FOREIGN KEY (CompanyID) REFERENCES Companies(CompanyID)
  );
  INSERT INTO Jobs (JobID, CompanyID, JobTitle, JobDescription, JobLocation, Salary, JobType, PostedDate) VALUES
  (11, 1, 'Software Tester', 'test and maintain software applications.', 'Bangalore', 12123000, 'Full-time', '2024-07
  (12, 2, 'Data Analyst', 'Analyze data.', 'Hyderabad', 1888800, 'Full-time', '2024-07-02 11:00:00'),
  (13, 3, 'Cloud Architect', 'Manage cloud architecture.', 'Pune', 13876000, 'Full-time', '2024-07-03 12:00:00'),
  (14, 4, 'Tableu Analyst', 'Analyzing data and creating dashboards.', 'Chennai', 1900000, 'Contract', '2024-07-04 09
  (15, 5, 'AIML Engineer', 'Develop AI-based applications.', 'Pune', 1800000, 'Full-time', '2024-07-05 08:45:00');
```

```
CREATE TABLE IF NOT EXISTS Applicants (
ApplicantID INT PRIMARY KEY,
FirstName VARCHAR(100),
LastName VARCHAR(200),
Email VARCHAR(255),
Phone VARCHAR(20),
Resume TEXT

);

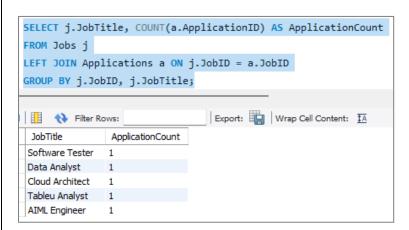
INSERT INTO Applicants (ApplicantID, FirstName, LastName, Email, Phone, Resume) VALUES (101, 'atharva', 'patil', 'patil@email.com', '9877673210', 'Resume link 1'), (102, 'Tejsinh', 'Bhosale', 'tejsinh@email.com', '9866661470', 'Resume link 2'), (103, 'patil', 'Deshpande', 'patil@email.com', '9884444698', 'Resume link 3'), (104, 'aditya', 'sarate', 'patil@email.com', '989888856', 'Resume link 4'), (105, 'utkarsh', 'chou', 'utkarsh@email.com', '9786544460', 'Resume link 5');
```

```
CREATE TABLE IF NOT EXISTS Applications (
ApplicationID INT PRIMARY KEY,
JobID INT,
ApplicantID INT,
ApplicationDate DATETIME,
CoverLetter TEXT,
FOREIGN KEY (JobID) REFERENCES Jobs(JobID),
FOREIGN KEY (ApplicantID) REFERENCES Applicants(ApplicantID)

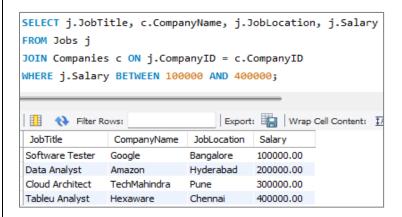
);

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 AWS cloud architect with 4 years of experience.'),
(114, 14, 104, '2024-07-09 10:00:00', 'Passionate about analyzing and predicting data with hands-on experience.'),
(115, 15, 105, '2024-07-10 11:30:00', 'Interested in AI applications and deep learning.');
```

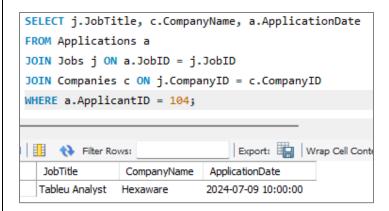
5. Write an SQL query to count the number of applications received for each job listing in the "Jobs" table. Display the job title and the corresponding application count. Ensure that it lists all jobs, even if they have no applications.



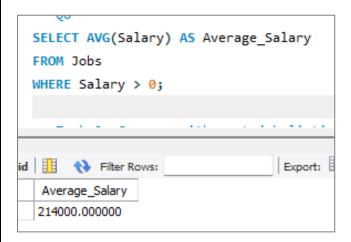
6. Develop an SQL query that retrieves job listings from the "Jobs" table within a specified salary range. Allow parameters for the minimum and maximum salary values. Display the job title, company name, location, and salary for each matching job.



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.



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.



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.CompanyID
 HAVING JobCount = (
     SELECT MAX(JobCount) FROM (
         SELECT COUNT(JobID) AS JobCount
         FROM Jobs
         GROUP BY CompanyID
     ) AS SubQuery
 );
                                 Export: Wrap Cel
d 🔢 🚷 Filter Rows:
   CompanyName JobCount
   Google
                2
   Amazon
              2
```

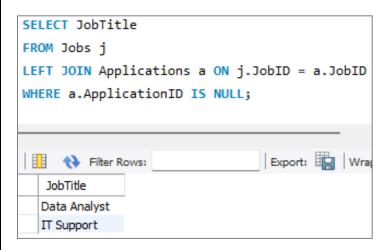
10. Find the applicants who have applied for positions in companies located in 'CityX' and have at least 3 years of experience.



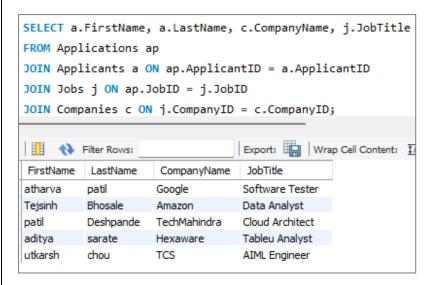
11. Retrieve a list of distinct job titles with salaries between \$60,000 and \$80,000.



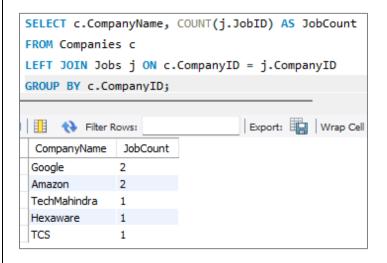
12. Find the jobs that have not received any applications.



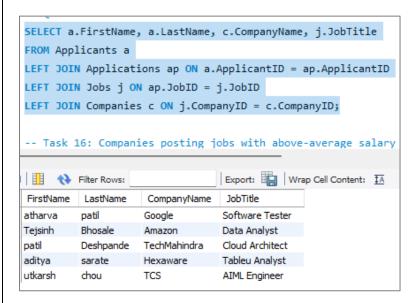
13. Retrieve a list of job applicants along with the companies they have applied to and the positions they have applied for.



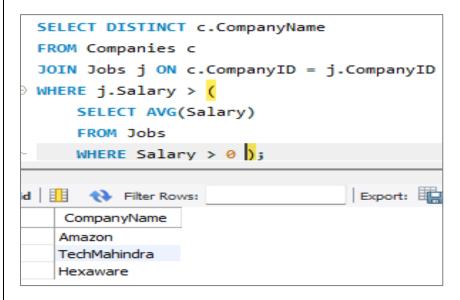
14. Retrieve a list of companies along with the count of jobs they have posted, even if they have not received any applications.



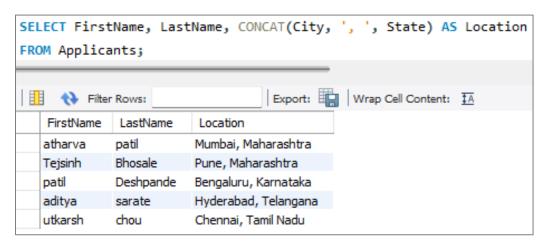
15. List all applicants along with the companies and positions they have applied for, including those who have not applied.



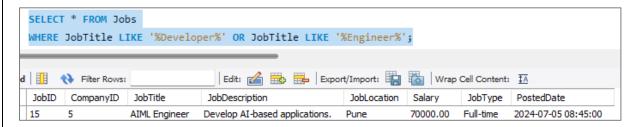
16. Find companies that have posted jobs with a salary higher than the average salary of all jobs.



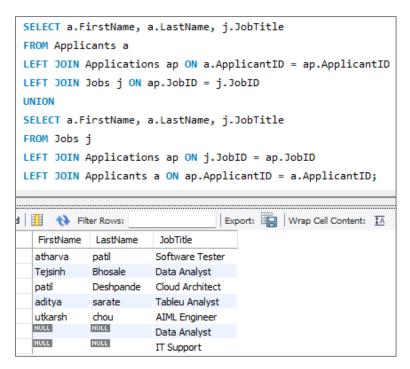
17. Display a list of applicants with their names and a concatenated string of their city and state.



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



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



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

