(Satyendra Singh Rathore)

## Tasks:

20

-- Q-1

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

```
create database CareerHub;
         use CareerHub;
 7
2. Create tables for Companies, Jobs, Applicants and Applications
 9 ● ⊖ create table Companies(CompanyID int primary key auto increment, CompanyName varchar(30),
        Location varchar(30));
11
12 • ⊖ create table Jobs(JobID int primary key auto_increment, CompanyID int, JobTitle varchar(30),
13
        JobDescription text, JobLocation varchar(30), salary float, JobType varchar(30),
        PostedDate datetime, foreign key (CompanyID) references Companies(CompanyID));
15
16 • ⊖ create table Applicants(ApplicantID int primary key auto_increment, first_name varchar(30),
        last_name varchar(30),Email varchar(30),Phone varchar(30),Resume text);
17
```

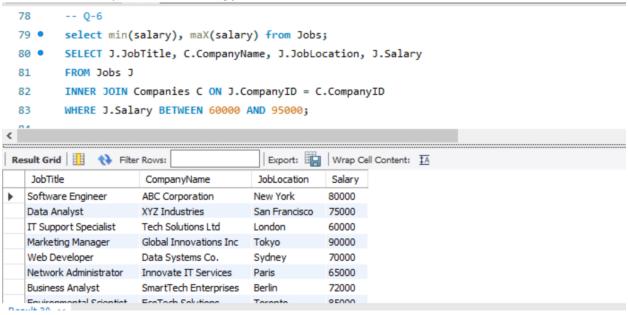
5. Write an SQL query to count the number of applications received for each job listing in the "Jobs" table. Display the job title and the corresponding application count. Ensure that it lists all jobs, even if they have no applications.

19 • ⊖ create table Applications(ApplicationID int primary key auto increment, JobID int, ApplicantID int, ApplicationDate datetime, CoverLetter text, foreign key (JobID) references Jobs (JobID),

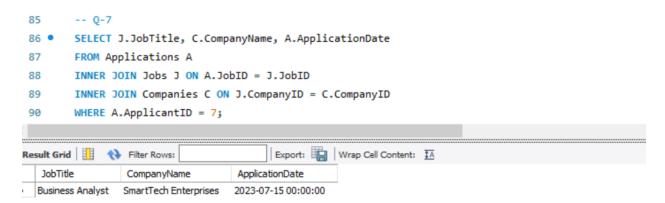
foreign key(ApplicantID) references Applicants(ApplicantID));

```
-- Q-5
 71
         SELECT J.JobID, J.JobTitle, COUNT(A.ApplicationID) AS ApplicationCount
 72 •
 73
         FROM Jobs J
 74
         LEFT JOIN Applications A ON J.JobID = A.JobID
 75
         GROUP BY J.JobID, J.JobTitle
         ORDER BY J.JobID;
 76
                                             Export: Wrap Cell Content: IA
Result Grid
              Filter Rows:
   JobID
          JobTitle
                               ApplicationCount
          Software Engineer
  2
          Data Analyst
                              1
  3
          IT Support Specialist
         Marketing Manager
  5
          Web Developer
```

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.

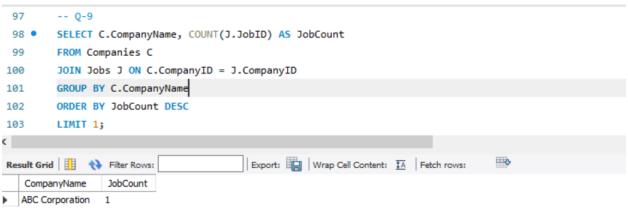


7. Write an SQL query that retrieves the job application history for a specific applicant. Allow a parameter for the ApplicantID, and return a result set with the job titles, company names, and application dates for all the jobs the applicant has applied to.

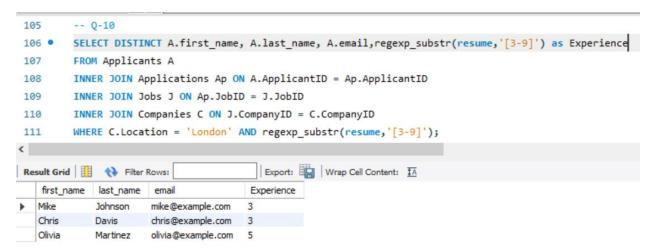


8. Create an SQL query that calculates and displays the average salary offered by all companies for job listings in the "Jobs" table. Ensure that the query filters out jobs with a salary of zero.

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.



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

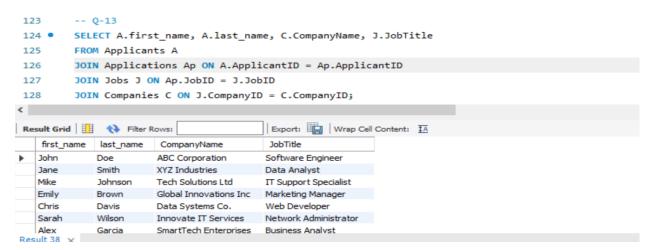


11. Retrieve a list of distinct job titles with salaries between \$60,000 and \$80,000.

```
113
          -- Q-11
114 •
         SELECT DISTINCT JobTitle
         FROM Jobs
115
116
         WHERE Salary BETWEEN 60000 AND 80000;
 117
<
Export: Wrap Cell Content: IA
    JobTitle
   Software Engineer
   Data Analyst
   IT Support Specialist
   Web Developer
   Network Administrator
   Business Analyst
   Accountant
```

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

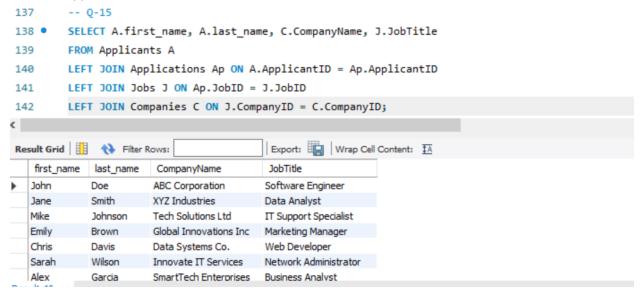
13. Retrieve a list of job applicants along with the companies they have applied to and the positions they have applied for.



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

```
130
        -- Q-14
        SELECT C.CompanyName, COUNT(J.JobID) AS JobCount
131 •
132
        FROM Companies C
        LEFT JOIN Jobs J ON C.CompanyID = J.CompanyID
133
134
        LEFT JOIN Applications A ON J.JobID = A.JobID
        GROUP BY C.CompanyName;
Export: Wrap Cell Content: IA
   CompanyName
                     JobCount
  ABC Corporation
  XYZ Industries
                  1
  Tech Solutions Ltd
  Global Innovations Inc 1
  Data Systems Co.
  Innovate IT Services 1
  SmartTech Enterprises
```

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.

```
144
145 •
        SELECT DISTINCT C.CompanyName
        FROM Companies C
146
        JOIN Jobs J ON C.CompanyID = J.CompanyID
147

⊖ WHERE J.Salary > (
148
            SELECT AVG(Salary)
149
            FROM Jobs
150
151
      ٠);
Export: Wrap Cell Content: TA
   CompanyName
  ABC Corporation
  Global Innovations Inc
  EcoTech Solutions
  Sunrise Technologies
```

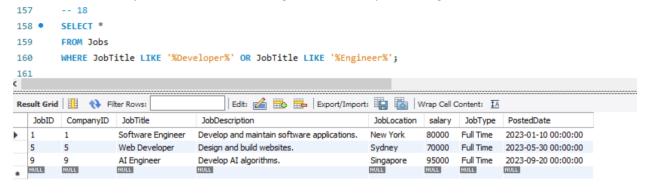
17. Display a list of applicants with their names and a concatenated string of their city and state.

```
-- 17 (Since there are no columns defined for address, city or state so there is error in running the query)

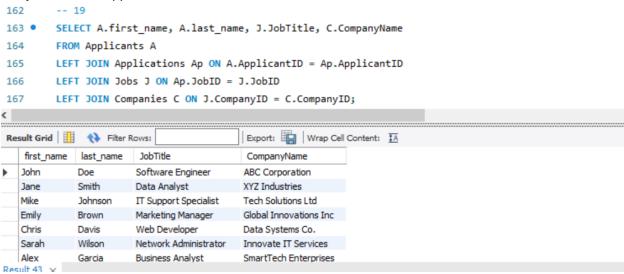
SELECT first_name, last_name, CONCAT(city, ', ', state) AS Location

FROM Applicants;
```

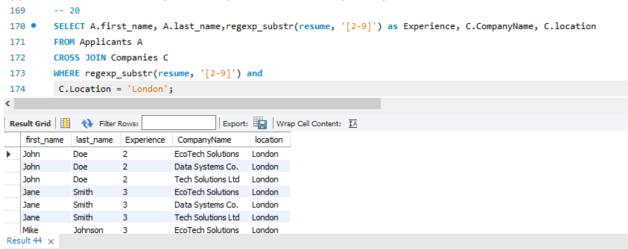
18. Retrieve a list of jobs with titles containing either 'Developer' or 'Engineer'.



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



\*\*\*\*\*