Coding Challenge : CareerHub

Vivek Chauhan
Java Batch 1

Tasks:

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

```
mysql> CREATE DATABASE IF NOT EXISTS CareerHub;
Query OK, 1 row affected (0.00 sec)

mysql> USE CareerHub;
Database changed
```

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

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

```
mysql> CREATE TABLE Companies (
    ->     CompanyID INT AUTO_INCREMENT PRIMARY KEY,
    ->     CompanyName VARCHAR(255) NOT NULL,
    ->     Location VARCHAR(255) NOT NULL
    -> );
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> CREATE TABLE Jobs (
    ->     JobID INT AUTO_INCREMENT PRIMARY KEY,
    ->     CompanyID INT,
    ->     JobTitle VARCHAR(255) NOT NULL,
    ->     JobDescription TEXT,
    ->     JobLocation VARCHAR(255),
    ->     Salary DECIMAL(10, 2) CHECK (Salary >= 0),
    ->     JobType VARCHAR(50),
    ->     PostedDate DATETIME DEFAULT CURRENT_TIMESTAMP,
    ->     FOREIGN KEY (CompanyID) REFERENCES Companies(Company
ID) ON DELETE CASCADE
    -> );
```

```
mysql> CREATE TABLE Applicants (
    ->     ApplicantID INT AUTO_INCREMENT PRIMARY KEY,
    ->     FirstName VARCHAR(100) NOT NULL,
    ->     LastName VARCHAR(100) NOT NULL,
    ->     Email VARCHAR(255) NOT NULL UNIQUE,
    ->     Phone VARCHAR(20),
    ->     Resume TEXT
    -> );
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> CREATE TABLE Applications (
    ->     ApplicationID INT AUTO_INCREMENT PRIMARY KEY,
    ->     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(Appl
icantID) ON DELETE CASCADE
    -> );
Query OK, 0 rows affected (0.01 sec)
```

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

```
mysql> DROP TABLE IF EXISTS Applications;
Query OK, 0 rows affected, 1 warning (0.01 sec)

mysql> DROP TABLE IF EXISTS Jobs;
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> DROP TABLE IF EXISTS Applicants;
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> DROP TABLE IF EXISTS Companies;
Query OK, 0 rows affected, 1 warning (0.00 sec)
```

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.

```
mysql> 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 Analyst       |                1 |
| Project Manager    |                1 |
| QA Engineer        |                1 |
| Research Scientist |                0 |
+--------------------+------------------+
5 rows in set (0.01 sec)
```

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.

```
mysql> select j.JobTitle, c.CompanyName, j.JobLocation,j.Salary

    -> from Jobs j
    -> join Companies c
    -> on j.CompanyID = c.CompanyID
    -> where j.salary between 65000 and 95000;
+--------------------+-------------+---------------+----------+
| JobTitle           | CompanyName | JobLocation   | Salary   |
+--------------------+-------------+---------------+----------+
| Software Developer | TechCorp    | New York      | 75000.00 |
| Data Analyst       | HealthPlus  | San Francisco | 65000.00 |
| Project Manager    | EduWorks    | Los Angeles   | 85000.00 |
| QA Engineer        | InnoTech    | Seattle       | 70000.00 |
| Research Scientist | BioMed      | Chicago       | 95000.00 |
+--------------------+-------------+---------------+----------+
5 rows in set (0.00 sec)
```

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.

```
mysql> 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 = 3;
+-----------------+-------------+---------------------+
| JobTitle        | CompanyName | ApplicationDate     |
+-----------------+-------------+---------------------+
| Project Manager | EduWorks    | 2024-03-15 14:30:00 |
+-----------------+-------------+---------------------+
1 row in set (0.01 sec)
```

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.

```
mysql> select avg(j.Salary) as AverageSalary
    -> from Jobs j
    -> where j.salary >0;
+---------------+
| AverageSalary |
+---------------+
|  78000.000000 |
+---------------+
1 row in set (0.00 sec)
```

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.

```
mysql> WITH CompanyJobCounts AS (
    ->     SELECT c.CompanyName, COUNT(j.JobID) AS JobCount
    ->     FROM Companies c
    ->     JOIN Jobs j ON c.CompanyID = j.CompanyID
    ->     GROUP BY c.CompanyName
    -> )
    -> SELECT CompanyName, JobCount
    -> FROM CompanyJobCounts
    -> WHERE JobCount = (SELECT MAX(JobCount) FROM CompanyJobCo
unts);
+-------------+----------+
| CompanyName | JobCount |
+-------------+----------+
| TechCorp    |        1 |
| HealthPlus  |        1 |
| EduWorks    |        1 |
| InnoTech    |        1 |
| BioMed      |        1 |
+-------------+----------+
5 rows in set (0.01 sec)
```

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

```
mysql> ALTER TABLE Applicants
    -> ADD Experience INT;
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> SELECT a.FirstName, a.LastName, j.JobTitle, c.CompanyNam
e
    -> 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
    -> where c.Location = 'Chicago'
    -> and a.Experience>=3;
Empty set (0.00 sec)
```

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

```
mysql> select distinct j.JobTitle
    -> from Jobs j
    -> where j.Salary between 60000 and 80000;
+--------------------+
| JobTitle           |
+--------------------+
| Software Developer |
| Data Analyst       |
| QA Engineer        |
+--------------------+
3 rows in set (0.00 sec)
```

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

```
mysql> select j.JobTitle
    -> from Jobs j
    -> left join Applications a
    -> on j.JobID = a.JobID
    -> where a.ApplicationID is null;
+--------------------+
| JobTitle           |
+--------------------+
| Research Scientist |
+--------------------+
1 row in set (0.00 sec)
```

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

```
mysql> select a.FirstName, a.LastName, c.CompanyName, j.JobTitl
e
    -> 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           |
+-----------+----------+-------------+--------------------+
| John      | Doe      | TechCorp    | Software Developer |
| Jane      | Smith    | HealthPlus  | Data Analyst       |
| Michael   | Johnson  | EduWorks    | Project Manager    |
| Emily     | Davis    | InnoTech    | QA Engineer        |
| David     | Brown    | TechCorp    | Software Developer |
+-----------+----------+-------------+--------------------+
5 rows in set (0.04 sec)
```

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

```
mysql> select c.CompanyName,count(j.JobID) as JobCount from Com
panies c left join Jobs j on c.CompanyID = j.CompanyID group by
 c.CompanyName;
+-------------+----------+
| CompanyName | JobCount |
+-------------+----------+
| TechCorp    |        1 |
| HealthPlus  |        1 |
| EduWorks    |        1 |
| InnoTech    |        1 |
| BioMed      |        1 |
+-------------+----------+
5 rows in set (0.04 sec)
```

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

```
mysql> select a.FirstName, a.LastName, c.CompanyName, j.JobTitl
e
    -> 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           |
+-----------+----------+-------------+--------------------+
| John      | Doe      | TechCorp    | Software Developer |
| Jane      | Smith    | HealthPlus  | Data Analyst       |
| Michael   | Johnson  | EduWorks    | Project Manager    |
| Emily     | Davis    | InnoTech    | QA Engineer        |
| David     | Brown    | TechCorp    | Software Developer |
+-----------+----------+-------------+--------------------+
5 rows in set (0.00 sec)
```

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

```
mysql> select c.CompanyName, j.JobTitle, j.Salary
    -> from Jobs j
    -> Join companies c on j.CompanyID = c.CompanyID
    -> where j.salary > (Select avg(Salary) from Jobs);
+-------------+--------------------+----------+
| CompanyName | JobTitle           | Salary   |
+-------------+--------------------+----------+
| EduWorks    | Project Manager    | 85000.00 |
| BioMed      | Research Scientist | 95000.00 |
+-------------+--------------------+----------+
2 rows in set (0.00 sec)
```

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

```
mysql> ALTER TABLE Applicants
    -> ADD City VARCHAR(100),
    -> ADD State VARCHAR(100);
```

```
mysql> SELECT a.FirstName, a.LastName, CONCAT(a.City, ', ', a.S
tate) AS Location
    -> FROM Applicants a;
+-----------+----------+-----------------+
| FirstName | LastName | Location        |
+-----------+----------+-----------------+
| John      | Doe      | New York, NY    |
| Jane      | Smith    | Los Angeles, CA |
| Michael   | Johnson  | Chicago, IL     |
| Emily     | Davis    | Houston, TX     |
| David     | Brown    | Miami, FL       |
+-----------+----------+-----------------+
5 rows in set (0.01 sec)
```

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

```
mysql> SELECT j.JobTitle
    -> FROM Jobs j
    -> WHERE j.JobTitle LIKE '%Developer%' OR j.JobTitle LIKE '
%Engineer%';
+--------------------+
| JobTitle           |
+--------------------+
| Software Developer |
| QA Engineer        |
+--------------------+
2 rows in set (0.04 sec)
```

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

```
mysql> SELECT a.FirstName, a.LastName, j.JobTitle
    -> FROM Applicants a
    -> LEFT JOIN Applications app ON a.ApplicantID = app.Applic
antID
    -> LEFT JOIN Jobs j ON app.JobID = j.JobID;
+-----------+----------+--------------------+
| FirstName | LastName | JobTitle           |
+-----------+----------+--------------------+
| John      | Doe      | Software Developer |
| Jane      | Smith    | Data Analyst       |
| Michael   | Johnson  | Project Manager    |
| Emily     | Davis    | QA Engineer        |
| David     | Brown    | Software Developer |
+-----------+----------+--------------------+
5 rows in set (0.01 sec)
```

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

```
mysql> SELECT a.FirstName, a.LastName, c.CompanyName
    -> 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 = 'New York' AND a.Experience > 2;
+-----------+----------+-------------+
| FirstName | LastName | CompanyName |
+-----------+----------+-------------+
| John      | Doe      | TechCorp    |
| David     | Brown    | TechCorp    |
+-----------+----------+-------------+
2 rows in set (0.00 sec)
```