MySQL Coding Challenge – CareerHub, The Job Board

Name: Atharva Padawale

Github Repository: https://github.com/AtharvaPadawale/hexaware-Java-Batch-5

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');
```

```sql
CREATE TABLE IF NOT EXISTS Applicants (
    ApplicantID INT PRIMARY KEY,
    FirstName VARCHAR(100),
    LastName VARCHAR(100),
    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', '9854444698', 'Resume link 3'),
(104, 'aditya', 'sarate', 'patil@email.com', '989888856', 'Resume link 4'),
(105, 'utkarsh', 'chou', 'utkarsh@email.com', '9786544460', 'Resume link 5');
```

```sql
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.

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

Filter Rows: Export: Wrap Cell Content:

JobTitle	ApplicationCount
Software Tester	1
Data Analyst	1
Cloud Architect	1
Tableu Analyst	1
AIML 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.

```sql
SELECT j.JobTitle, c.CompanyName, j.JobLocation, j.Salary
FROM Jobs j
JOIN Companies c ON j.CompanyID = c.CompanyID
WHERE j.Salary BETWEEN 100000 AND 400000;
```

Filter Rows: | Export: | Wrap Cell Content:

JobTitle	CompanyName	JobLocation	Salary
Software Tester	Google	Bangalore	100000.00
Data Analyst	Amazon	Hyderabad	200000.00
Cloud Architect	TechMahindra	Pune	300000.00
Tableu Analyst	Hexaware	Chennai	400000.00

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
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 = 104;
```

Filter Rows: | Export: | Wrap Cell Cont

JobTitle	CompanyName	ApplicationDate
Tableu Analyst	Hexaware	2024-07-09 10:00:00

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.

```sql
SELECT AVG(Salary) AS Average_Salary
FROM Jobs
WHERE Salary > 0;
```

id | Filter Rows: | Export:

Average_Salary
214000.000000

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.CompanyID
HAVING JobCount = (
    SELECT MAX(JobCount) FROM (
        SELECT COUNT(JobID) AS JobCount
        FROM Jobs
        GROUP BY CompanyID
    ) AS SubQuery
);
```

Filter Rows: | Export: | Wrap Cel

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.

```
SELECT a.firstname, a.lastname, j.jobTitle, c.location, a.experience
FROM Applicants a
JOIN Applications ap ON a.ApplicantID = ap.ApplicantID
JOIN Jobs j ON ap.JobID = j.JobID
JOIN Companies c ON j.CompanyID = c.CompanyID
WHERE c.Location = 'chennai' AND a.experience >= 3;
```

Filter Rows: | Export: | Wrap Cell Content:

firstname	lastname	jobTitle	location	experience
aditya	sarate	Tableu Analyst	Chennai	4

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;
```

Filter Rows: | Export:

JobTitle
AIML Engineer
IT Support

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

```
SELECT JobTitle
FROM Jobs j
LEFT JOIN Applications a ON j.JobID = a.JobID
WHERE a.ApplicationID IS NULL;
```

Filter Rows: Export: Wra

JobTitle
Data Analyst
IT Support

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 ap
JOIN Applicants a ON ap.ApplicantID = a.ApplicantID
JOIN Jobs j ON ap.JobID = j.JobID
JOIN Companies c ON j.CompanyID = c.CompanyID;
```

Filter Rows: Export: Wrap Cell Content:

FirstName	LastName	CompanyName	JobTitle
atharva	patil	Google	Software Tester
Tejsinh	Bhosale	Amazon	Data Analyst
patil	Deshpande	TechMahindra	Cloud Architect
aditya	sarate	Hexaware	Tableu Analyst
utkarsh	chou	TCS	AIML 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.CompanyID;
```

Filter Rows: Export: Wrap Cell

CompanyName	JobCount
Google	2
Amazon	2
TechMahindra	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 ap ON a.ApplicantID = ap.ApplicantID
LEFT JOIN Jobs j ON ap.JobID = j.JobID
LEFT JOIN Companies c ON j.CompanyID = c.CompanyID;

-- Task 16: Companies posting jobs with above-average salary
```

Filter Rows: | Export: | Wrap Cell Content:

FirstName	LastName	CompanyName	JobTitle
atharva	patil	Google	Software Tester
Tejsinh	Bhosale	Amazon	Data Analyst
patil	Deshpande	TechMahindra	Cloud Architect
aditya	sarate	Hexaware	Tableu Analyst
utkarsh	chou	TCS	AIML 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 );
```

Filter Rows: | Export:

CompanyName
Amazon
TechMahindra
Hexaware

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;
```

Filter Rows: | Export: | Wrap Cell Content:

FirstName	LastName	Location
atharva	patil	Mumbai, Maharashtra
Tejsinh	Bhosale	Pune, Maharashtra
patil	Deshpande	Bengaluru, Karnataka
aditya	sarate	Hyderabad, Telangana
utkarsh	chou	Chennai, Tamil Nadu

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

```
SELECT * FROM Jobs
WHERE JobTitle LIKE '%Developer%' OR JobTitle LIKE '%Engineer%';
```

Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

JobID	CompanyID	JobTitle	JobDescription	JobLocation	Salary	JobType	PostedDate
15	5	AIML Engineer	Develop AI-based applications.	Pune	70000.00	Full-time	2024-07-05 08:45:00

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
LEFT JOIN Applications ap ON a.ApplicantID = ap.ApplicantID
LEFT JOIN Jobs j ON ap.JobID = j.JobID
UNION
SELECT a.FirstName, a.LastName, j.JobTitle
FROM Jobs j
LEFT JOIN Applications ap ON j.JobID = ap.JobID
LEFT JOIN Applicants a ON ap.ApplicantID = a.ApplicantID;
```

Filter Rows: | Export: | Wrap Cell Content:

FirstName	LastName	JobTitle
atharva	patil	Software Tester
Tejsinh	Bhosale	Data Analyst
patil	Deshpande	Cloud Architect
aditya	sarate	Tableu Analyst
utkarsh	chou	AIML Engineer
NULL	NULL	Data Analyst
NULL	NULL	IT Support

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, c.Location
FROM Applicants a
CROSS JOIN Companies c
WHERE c.Location = 'Chennai'
  AND a.Experience > 2;
```

Filter Rows: | Export: | Wrap Cell Content:

FirstName	LastName	CompanyName	Location
atharva	patil	Hexaware	Chennai
Tejsinh	Bhosale	Hexaware	Chennai
aditya	sarate	Hexaware	Chennai