CareerHub, The Job Board

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

```
1    CREATE DATABASE CareerHub;
2    USE CareerHub;
```

```
2  02:30:59  CREATE DATABASE CareerHub
3  02:30:59  USE CareerHub
```

.

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

```
CREATE TABLE Companies(
    CompanyID INT PRIMARY KEY,
    CompanyName VARCHAR(255),
    Location VARCHAR(255)
);

CREATE TABLE Jobs(
    JobID INT PRIMARY KEY,
    CompanyID INT,
    JobTitle VARCHAR(255),
    JobDescription VARCHAR(255),
    JobLocation VARCHAR(255),
    Salary DECIMAL,
    JobType VARCHAR(255),
    PostedDate DATETIME,
    FOREIGN KEY (CompanyID) REFERENCES Companies(CompanyID)
);

CREATE TABLE Applicants (
    ApplicantID INT PRIMARY KEY,
    FirstName VARCHAR(255),
    LastName VARCHAR(255),
    Email VARCHAR(255),
    Phone VARCHAR(255),
    Resume TEXT
);
```

```sql
CREATE TABLE 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)
);
```

✓	4	02:42:32	CREATE TABLE Companies(CompanyID INT PRIMARY KEY, CompanyName VARCHAR(255), Location V...	0 row(s) affected
✓	5	02:42:33	CREATE TABLE Jobs(JobID INT PRIMARY KEY, CompanyID INT, JobTitle VARCHAR(255), JobDescri...	0 row(s) affected
✓	6	02:42:33	CREATE TABLE Applicants (ApplicantID INT PRIMARY KEY, FirstName VARCHAR(255), LastName VARCHA...	0 row(s) affected
✓	7	02:42:33	CREATE TABLE Applications (ApplicationID INT PRIMARY KEY, JobID INT, ApplicantID INT, Applicati...	0 row(s) affected

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

   Did it in 2nd Question

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

   ——————————

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 Jobs.JobID, COALESCE(COUNT(*), 0) as "No of Applications"
FROM Jobs LEFT JOIN Applications ON Jobs.JobID = Applications.JobID
GROUP BY Jobs.JobID;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

JobID	No of Applications
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	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 75000 AND 100000;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

JobTitle	CompanyName	JobLocation	Salary
Software Engineer	Tech Innovators Inc.	Silicon Valley	90000
Data Scientist	Global Solutions Co.	New York	85000
UX/UI Designer	Digital Ventures LLC	Austin	80000
Network Engineer	Innovate Tech Solutions	Seattle	95000
Business Analyst	Dynamic Enterprises	Chicago	75000
Environmental Scientist	EcoTech Solutions	San Francisco	90000
Cybersecurity Analyst	Cyber Systems International	Boston	100000
Project Manager	Swift Tech Services	Denver	85000
Marketing Specialist	NexGen Innovations	Atlanta	80000

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;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Cont

JobTitle	CompanyName	ApplicationDate
Software Engineer	Tech Innovators Inc.	2023-01-12 00:00:00

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;
```

Result Grid | Filter Rows: | Export:

Average Salary
85000.0000

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.

```sql
SELECT C.CompanyName, COUNT(J.JobID) AS JobCount
FROM Companies C JOIN Jobs J ON C.CompanyID = J.CompanyID
GROUP BY C.CompanyName
ORDER BY JobCount DESC, C.CompanyName ASC LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

CompanyName	JobCount
Creative Minds Ltd.	1

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

```sql
SELECT AP.ApplicantID, AP.FirstName, AP.LastName
FROM Jobs J JOIN Applications A ON J.JobID =  A.JobID
JOIN Applicants AP ON A.ApplicantID = AP.ApplicantID
WHERE J.JobLocation = "Austin" AND AP.Experience >= 3;
```

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

```
1 •  SELECT JobTitle FROM Jobs
2    WHERE Salary BETWEEN 60000 AND 80000;
3
```

Result Grid | Filter Rows: | Export:

JobTitle
Graphic Designer
UX/UI Designer
Business Analyst
Marketing Specialist

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

```
1 •  SELECT J.JobID, J.JobTitle, C.CompanyName
2    FROM Jobs J
3    JOIN Companies C ON J.CompanyID = C.CompanyID
4    WHERE J.JobID NOT IN (SELECT DISTINCT JobID FROM Applications);
5
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

JobID	JobTitle	CompanyName

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

```sql
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;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

FirstName	LastName	CompanyName	JobTitle
John	Doe	Tech Innovators Inc.	Software Engineer
Jane	Smith	Global Solutions Co.	Data Scientist
Robert	Johnson	Creative Minds Ltd.	Graphic Designer
Emily	White	Digital Ventures LLC	UX/UI Designer
Michael	Brown	Innovate Tech Solutions	Network Engineer
Amanda	Jones	Dynamic Enterprises	Business Analyst
Christopher	Miller	EcoTech Solutions	Environmental Scientist
Jessica	Clark	Cyber Systems International	Cybersecurity Analyst
Daniel	Anderson	Swift Tech Services	Project Manager
Olivia	Wilson	NexGen Innovations	Marketing Specialist

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

```sql
SELECT C.CompanyID, C.CompanyName, COUNT(J.JobID) AS PostedJobsCount
FROM Companies C
LEFT JOIN Jobs J ON C.CompanyID = J.CompanyID
GROUP BY C.CompanyID;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

CompanyID	CompanyName	PostedJobsCount
1	Tech Innovators Inc.	1
2	Global Solutions Co.	1
3	Creative Minds Ltd.	1
4	Digital Ventures LLC	1
5	Innovate Tech Solutions	1
6	Dynamic Enterprises	1
7	EcoTech Solutions	1
8	Cyber Systems International	1
9	Swift Tech Services	1
10	NexGen Innovations	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;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

FirstName	LastName	CompanyName	JobTitle
John	Doe	Tech Innovators Inc.	Software Engineer
Jane	Smith	Global Solutions Co.	Data Scientist
Robert	Johnson	Creative Minds Ltd.	Graphic Designer
Emily	White	Digital Ventures LLC	UX/UI Designer
Michael	Brown	Innovate Tech Solutions	Network Engineer
Amanda	Jones	Dynamic Enterprises	Business Analyst
Christopher	Miller	EcoTech Solutions	Environmental Scientist
Jessica	Clark	Cyber Systems International	Cybersecurity Analyst
Daniel	Anderson	Swift Tech Services	Project Manager
Olivia	Wilson	NexGen Innovations	Marketing Specialist

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

```
SELECT C.CompanyName, COUNT(J.JobID) AS PostedJobsCount
FROM Companies C
LEFT JOIN Jobs J ON C.CompanyID = J.CompanyID AND J.Salary > (SELECT AVG(Salary) FROM Jobs)
GROUP BY C.CompanyName;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

CompanyName	PostedJobsCount
Tech Innovators Inc.	1
Global Solutions Co.	0
Creative Minds Ltd.	0
Digital Ventures LLC	0
Innovate Tech Solutions	1
Dynamic Enterprises	0
EcoTech Solutions	1
Cyber Systems International	1
Swift Tech Services	0
NexGen Innovations	0

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

```
SELECT A.FirstName, A.LastName, CONCAT(A.City, ', ', A.State) AS CityState
FROM Applicants A;
```

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

```sql
SELECT JobTitle, CompanyName
FROM Jobs J
JOIN Companies C ON J.CompanyID = C.CompanyID
WHERE J.JobTitle LIKE '%Developer%' OR J.JobTitle LIKE '%Engineer%';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

JobTitle	CompanyName
Software Engineer	Tech Innovators Inc.
Network Engineer	Innovate Tech Solutions

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

    (To Do)

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

```sql
SELECT * FROM Applicants
CROSS JOIN Companies
WHERE Applicants.experience > 3 AND Companies.Location = "Chennai";
```