Tasks:

1. Provide a SQL script that initializes the database for the job Board scenario “CareerHub”
   mysql> CREATE DATABASE CareerHub;

```
mysql> CREATE DATABASE CareerHub;
Query OK, 1 row affected (0.01 sec)
```

2. Create tables for Companies, Jobs, Applicants, and Applications.
   Creating table for **Companies**
   mysql> CREATE TABLE Companies (
   -> CompanyID INT PRIMARY KEY,
   -> CompanyName VARCHAR(255),
   -> Location VARCHAR(255)
   ->);

```
mysql> USE CareerHub;
Database changed
mysql> CREATE TABLE Companies (
    ->     CompanyID INT PRIMARY KEY,
    ->     CompanyName VARCHAR(255),
    ->     Location VARCHAR(255)
    -> );
Query OK, 0 rows affected (0.04 sec)
```

   Creating table for **Jobs**
   CREATE TABLE Jobs (
   -> JobID INT PRIMARY KEY,
   -> CompanyID INT,
   -> JobTitle VARCHAR(255),
   -> JobDescription TEXT,
   -> JobLocation VARCHAR(255),
   -> Salary DECIMAL,
   -> JobType VARCHAR(255),
   -> PostedDate DATETIME,
   -> FOREIGN KEY (CompanyID) REFERENCES Companies(CompanyID)
   ->);

```
mysql> CREATE TABLE Jobs (
    ->     JobID INT PRIMARY KEY,
    ->     CompanyID INT,
    ->     JobTitle VARCHAR(255),
    ->     JobDescription TEXT,
    ->     JobLocation VARCHAR(255),
    ->     Salary DECIMAL,
    ->     JobType VARCHAR(255),
    ->     PostedDate DATETIME,
    ->     FOREIGN KEY (CompanyID) REFERENCES Companies(CompanyID)
    -> );
Query OK, 0 rows affected (0.04 sec)
```

Creating table **Applicants:**

```
mysql> CREATE TABLE Applicants (
    ->     ApplicantID INT PRIMARY KEY,
    ->     FirstName VARCHAR(255),
    ->     LastName VARCHAR(255),
    ->     Email VARCHAR(255),
    ->     Phone VARCHAR(20),
    ->     Resume TEXT
    -> );
Query OK, 0 rows affected (0.02 sec)
```

Creating table **Applications**

```
mysql> 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)
    -> );
Query OK, 0 rows affected (0.03 sec)
```

3. Define appropriate primary keys, foreign keys, and constraints
Please refer to the previous tasks Snippets as this task has already been completed.

4. Ensure the script handles potential errors, such as if the database or tables already exist.
The following task has been performed successfully.

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.JobID,
    ->     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;
+-------+------------------------+------------------+
| JobID | JobTitle               | ApplicationCount |
+-------+------------------------+------------------+
|     1 | Software Engineer      |                1 |
|     2 | Marketing Specialist   |                1 |
|     3 | Data Analyst           |                1 |
|     4 | Graphic Designer       |                1 |
|     5 | IT Support Specialist  |                1 |
|     6 | Project Manager        |                1 |
|     7 | Sales Representative   |                1 |
|     8 | Financial Analyst      |                1 |
|     9 | Software Tester        |                1 |
|    10 | Environmental Engineer |                1 |
+-------+------------------------+------------------+
10 rows in set (0.00 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,
    ->     C.Location AS CompanyLocation,
    ->     J.JobLocation,
    ->     J.Salary
    -> FROM
    ->     Jobs J
    -> JOIN
    ->     Companies C ON J.CompanyID = C.CompanyID
    -> WHERE
    ->     J.Salary BETWEEN 50000 AND 80000;
+----------------------+---------------------+-----------------+-------------+----------+
| JobTitle             | CompanyName         | CompanyLocation | JobLocation | Salary   |
+----------------------+---------------------+-----------------+-------------+----------+
| Marketing Specialist | XYZ Industries      | Los Angeles     | Los Angeles | 75000.50 |
| Graphic Designer     | Global Innovations  | London          | London      | 65000.25 |
| IT Support Specialist| Metro Systems Inc   | Tokyo           | Tokyo       | 70000.00 |
| Software Tester      | Infinite Innovations| Singapore       | Singapore   | 80000.00 |
+----------------------+---------------------+-----------------+-------------+----------+
4 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 = 7;
+---------------------+---------------------+---------------------+
| JobTitle            | CompanyName         | ApplicationDate     |
+---------------------+---------------------+---------------------+
| Sales Representative | Dynamic Technologies | 2023-02-07 20:00:00 |
+---------------------+---------------------+---------------------+
1 row in set (0.00 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(Salary) AS AverageSalary
    -> FROM
    ->     Jobs
    -> WHERE
    ->     Salary > 0;
+---------------+
| AverageSalary |
+---------------+
|  82500.350000 |
+---------------+
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> SELECT
    ->     C.CompanyName,
    ->     COUNT(J.JobID) AS JobCount
    -> FROM
    ->     Companies C
    -> JOIN
    ->     Jobs J ON C.CompanyID = J.CompanyID
    -> GROUP BY
    ->     C.CompanyID, C.CompanyName
    -> HAVING
    ->     COUNT(J.JobID) = (
    ->         SELECT
    ->             MAX(JobCount)
    ->         FROM
    ->             (
    ->                 SELECT
    ->                     COUNT(JobID) AS JobCount
    ->                 FROM
    ->                     Jobs
    ->                 GROUP BY
    ->                     CompanyID
    ->             ) AS MaxJobCount
    ->     );
+---------------------+----------+
| CompanyName         | JobCount |
+---------------------+----------+
| ABC Corporation     |        1 |
| XYZ Industries      |        1 |
| Tech Solutions Ltd  |        1 |
| Global Innovations  |        1 |
| Metro Systems Inc   |        1 |
| Data Services Co.   |        1 |
| Dynamic Technologies |        1 |
| Smart Ventures      |        1 |
| Infinite Innovations |        1 |
| Eco Enterprises     |        1 |
+---------------------+----------+
10 rows in set (0.00 sec)
```

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

```
mysql> SELECT DISTINCT
    ->     A.ApplicantID,
    ->     A.FirstName,
    ->     A.LastName,
    ->     A.Email,
    ->     A.Phone,
    ->     A.Resume
    -> 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 = 'CityX'
    ->     AND J.JobDescription LIKE '%3 years%';
Empty set (0.00 sec)
```

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

```
mysql> SELECT DISTINCT
    ->     JobTitle
    -> FROM
    ->     Jobs
    -> WHERE
    ->     Salary BETWEEN 60000 AND 80000;
+----------------------+
| JobTitle             |
+----------------------+
| Marketing Specialist |
| Graphic Designer     |
| IT Support Specialist |
| Software Tester      |
+----------------------+
4 rows in set (0.01 sec)
```

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

```
mysql> SELECT
    ->     J.JobID,
    ->     J.JobTitle
    -> FROM
    ->     Jobs J
    -> LEFT JOIN
    ->     Applications A ON J.JobID = A.JobID
    -> WHERE
    ->     A.ApplicationID IS NULL;
Empty 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.ApplicantID,
    ->     A.FirstName,
    ->     A.LastName,
    ->     A.Email,
    ->     A.Phone,
    ->     A.Resume,
    ->     C.CompanyName,
    ->     J.JobTitle
    -> 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;
+-------------+-----------+----------+---------------------------+--------------+---------------------------+---------------------+-------------------------+
| ApplicantID | FirstName | LastName | Email                     | Phone        | Resume                    | CompanyName         | JobTitle                |
|
+-------------+-----------+----------+---------------------------+--------------+---------------------------+---------------------+-------------------------+
|           1 | John      | Doe      | john.doe@example.com      | 123-456-7890 | John_Doe_Resume.pdf       | ABC Corporation     | Software Engineer       |
|           2 | Jane      | Smith    | jane.smith@example.com    | 987-654-3210 | Jane_Smith_Resume.docx    | XYZ Industries      | Marketing Specialist    |
|           3 | Michael   | Johnson  | michael.johnson@example.com | 456-789-0123 | Michael_Johnson_Resume.txt | Tech Solutions Ltd  | Data Analyst            |
|           4 | Emily     | Williams | emily.williams@example.com | 789-012-3456 | Emily_Williams_Resume.pdf | Global Innovations  | Graphic Designer        |
|           5 | Daniel    | Brown    | daniel.brown@example.com  | 012-345-6789 | Daniel_Brown_Resume.docx  | Metro Systems Inc   | IT Support Specialist   |
|           6 | Olivia    | Jones    | olivia.jones@example.com  | 345-678-9012 | Olivia_Jones_Resume.txt   | Data Services Co.   | Project Manager         |
|           7 | William   | Miller   | william.miller@example.com | 678-901-2345 | William_Miller_Resume.pdf | Dynamic Technologies | Sales Representative    |
|           8 | Sophia    | Davis    | sophia.davis@example.com  | 901-234-5678 | Sophia_Davis_Resume.docx  | Smart Ventures      | Financial Analyst       |
|           9 | Liam      | Anderson | liam.anderson@example.com | 234-567-8901 | Liam_Anderson_Resume.txt  | Infinite Innovations | Software Tester         |
|          10 | Ava       | Taylor   | ava.taylor@example.com    | 567-890-1234 | Ava_Taylor_Resume.pdf     | Eco Enterprises     | Environmental Engineer  |
+-------------+-----------+----------+---------------------------+--------------+---------------------------+---------------------+-------------------------+
10 rows in set (0.00 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.CompanyID,
    ->     C.CompanyName,
    ->     COUNT(J.JobID) AS JobCount
    -> FROM
    ->     Companies C
    -> LEFT JOIN
    ->     Jobs J ON C.CompanyID = J.CompanyID
    -> LEFT JOIN
    ->     Applications A ON J.JobID = A.JobID
    -> GROUP BY
    ->     C.CompanyID, C.CompanyName;
+------------+----------------------+----------+
| CompanyID | CompanyName          | JobCount |
+------------+----------------------+----------+
|          1 | ABC Corporation      |        1 |
|          2 | XYZ Industries       |        1 |
|          3 | Tech Solutions Ltd   |        1 |
|          4 | Global Innovations   |        1 |
|          5 | Metro Systems Inc    |        1 |
|          6 | Data Services Co.    |        1 |
|          7 | Dynamic Technologies |        1 |
|          8 | Smart Ventures       |        1 |
|          9 | Infinite Innovations |        1 |
|         10 | Eco Enterprises      |        1 |
+------------+----------------------+----------+
10 rows in set (0.00 sec)
```

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

```
mysql> SELECT
    ->     A.ApplicantID,
    ->     A.FirstName,
    ->     A.LastName,
    ->     A.Email,
    ->     A.Phone,
    ->     A.Resume,
    ->     C.CompanyName,
    ->     J.JobTitle
    -> FROM
    ->     Applicants A
    -> CROSS JOIN
    ->     Companies C
    -> CROSS JOIN
    ->     Jobs J
    -> LEFT JOIN
    ->     Applications Ap ON A.ApplicantID = Ap.ApplicantID AND J.JobID = Ap.JobID;
+-------------+-----------+----------+-----------------------------+--------------+-----------------------------+-----------------+-----------------------+
| ApplicantID | FirstName | LastName | Email                       | Phone        | Resume                      | CompanyName     | JobTitle              |
+-------------+-----------+----------+-----------------------------+--------------+-----------------------------+-----------------+-----------------------+
|           1 | John      | Doe      | john.doe@example.com        | 123-456-7890 | John_Doe_Resume.pdf         | Eco Enterprises | Software Engineer     |
|           2 | Jane      | Smith    | jane.smith@example.com      | 987-654-3210 | Jane_Smith_Resume.docx      | Eco Enterprises | Software Engineer     |
|           3 | Michael   | Johnson  | michael.johnson@example.com | 456-789-0123 | Michael_Johnson_Resume.txt  | Eco Enterprises | Software Engineer     |
|           4 | Emily     | Williams | emily.williams@example.com  | 789-012-3456 | Emily_Williams_Resume.pdf   | Eco Enterprises | Software Engineer     |
|           5 | Daniel    | Brown    | daniel.brown@example.com    | 012-345-6789 | Daniel_Brown_Resume.docx    | Eco Enterprises | Software Engineer     |
|           6 | Olivia    | Jones    | olivia.jones@example.com    | 345-678-9012 | Olivia_Jones_Resume.txt     | Eco Enterprises | Software Engineer     |
|           7 | William   | Miller   | william.miller@example.com  | 678-901-2345 | William_Miller_Resume.pdf   | Eco Enterprises | Software Engineer     |
|           8 | Sophia    | Davis    | sophia.davis@example.com    | 901-234-5678 | Sophia_Davis_Resume.docx    | Eco Enterprises | Software Engineer     |
|           9 | Liam      | Anderson | liam.anderson@example.com   | 234-567-8901 | Liam_Anderson_Resume.txt    | Eco Enterprises | Software Engineer     |
```

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

```
mysql> SELECT DISTINCT
    ->     C.CompanyID,
    ->     C.CompanyName
    -> FROM
    ->     Companies C
    -> JOIN
    ->     Jobs J ON C.CompanyID = J.CompanyID
    -> WHERE
    ->     J.Salary > (SELECT AVG(Salary) FROM Jobs);
+------------+-----------------------+
| CompanyID | CompanyName           |
+------------+-----------------------+
|          1 | ABC Corporation       |
|          6 | Data Services Co.     |
|          7 | Dynamic Technologies  |
|          8 | Smart Ventures        |
|         10 | Eco Enterprises       |
+------------+-----------------------+
5 rows in set (0.00 sec)
```

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

```
mysql> SELECT
    ->     *
    -> FROM
    ->     Jobs
    -> WHERE
    ->     JobTitle LIKE '%Developer%' OR JobTitle LIKE '%Engineer%';
+-------+-----------+----------------------+-----------------------------------------------+-------------+---------+-----------+---------------------+
| JobID | CompanyID | JobTitle             | JobDescription                                | JobLocation | Salary  | JobType   | PostedDate          |
+-------+-----------+----------------------+-----------------------------------------------+-------------+---------+-----------+---------------------+
|     1 |         1 | Software Engineer    | Developing and maintaining software applications. | New York    | 90000.00 | Full-time | 2023-01-15 08:30:00 |
|    10 |        10 | Environmental Engineer | Designing and implementing environmental solutions. | Paris       | 95000.50 | Full-time | 2023-01-24 02:45:00 |
+-------+-----------+----------------------+-----------------------------------------------+-------------+---------+-----------+---------------------+
2 rows in set (0.00 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.ApplicantID,
    ->     CONCAT(A.FirstName, ' ', A.LastName) AS FullName,
    ->     J.JobID,
    ->     J.JobTitle,
    ->     COALESCE(Ap.ApplicationDate, 'Not Applied') AS ApplicationDate
    -> FROM
    ->     Applicants A
    -> CROSS JOIN
    ->     Jobs J
    -> LEFT JOIN
    ->     Applications Ap ON A.ApplicantID = Ap.ApplicantID AND J.JobID = Ap.JobID;
+-------------+-----------------+-------+-----------------------+---------------------+
| ApplicantID | FullName        | JobID | JobTitle              | ApplicationDate     |
+-------------+-----------------+-------+-----------------------+---------------------+
|          10 | Ava Taylor      |     1 | Software Engineer     | Not Applied         |
|           9 | Liam Anderson   |     1 | Software Engineer     | Not Applied         |
|           8 | Sophia Davis    |     1 | Software Engineer     | Not Applied         |
|           7 | William Miller  |     1 | Software Engineer     | Not Applied         |
|           6 | Olivia Jones    |     1 | Software Engineer     | Not Applied         |
|           5 | Daniel Brown    |     1 | Software Engineer     | Not Applied         |
|           4 | Emily Williams  |     1 | Software Engineer     | Not Applied         |
|           3 | Michael Johnson |     1 | Software Engineer     | Not Applied         |
|           2 | Jane Smith      |     1 | Software Engineer     | Not Applied         |
|           1 | John Doe        |     1 | Software Engineer     | 2023-02-01 09:30:00 |
```