Coding Challenges-CareerHub,The Job Board

1.Create database “CareerHub”

```
mysql> create database CareerHub;
Query OK, 1 row affected (0.03 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| careerhub          |
| d1                 |
| hexaware           |
| hmbank             |
| information_schema |
| mysql              |
| performance_schema |
| petpals            |
| sakila             |
| sisdb              |
| sys                |
| techshop           |
| world              |
+--------------------+
13 rows in set (0.05 sec)
```

2.Create tables

Companies

```
mysql> create table companies(CompanyId int primary key,CompanyName text,Location text);
Query OK, 0 rows affected (0.14 sec)

mysql> desc companies;
+-------------+------+------+-----+---------+-------+
| Field       | Type | Null | Key | Default | Extra |
+-------------+------+------+-----+---------+-------+
| CompanyId   | int  | NO   | PRI | NULL    |       |
| CompanyName | text | YES  |     | NULL    |       |
| Location    | text | YES  |     | NULL    |       |
+-------------+------+------+-----+---------+-------+
3 rows in set (0.05 sec)
```

Jobs

```
mysql> create table Jobs(JobId int primary key,CompanyId int,foreign key(CompanyId) references companies(CompanyId),JobTitle text,JobDescription t
ext,JobLocation text, Salary decimal(10,2));
Query OK, 0 rows affected (0.10 sec)

mysql> desc jobs;
+----------------+---------------+------+-----+---------+-------+
| Field          | Type          | Null | Key | Default | Extra |
+----------------+---------------+------+-----+---------+-------+
| JobId          | int           | NO   | PRI | NULL    |       |
| CompanyId      | int           | YES  | MUL | NULL    |       |
| JobTitle       | text          | YES  |     | NULL    |       |
| JobDescription | text          | YES  |     | NULL    |       |
| JobLocation    | text          | YES  |     | NULL    |       |
| Salary         | decimal(10,2) | YES  |     | NULL    |       |
+----------------+---------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

```
mysql> alter table jobs add JobType text;
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> alter table jobs add PostedDate datetime;
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc jobs;
+-----------------+---------------+------+-----+---------+-------+
| Field           | Type          | Null | Key | Default | Extra |
+-----------------+---------------+------+-----+---------+-------+
| JobId           | int           | NO   | PRI | NULL    |       |
| CompanyId       | int           | YES  | MUL | NULL    |       |
| JobTitle        | text          | YES  |     | NULL    |       |
| JobDescription  | text          | YES  |     | NULL    |       |
| JobLocation     | text          | YES  |     | NULL    |       |
| Salary          | decimal(10,2) | YES  |     | NULL    |       |
| JobType         | text          | YES  |     | NULL    |       |
| PostedDate      | datetime      | YES  |     | NULL    |       |
+-----------------+---------------+------+-----+---------+-------+
8 rows in set (0.01 sec)
```

Applicants

```
mysql> create table Applicants(ApplicantId int primary key,FirstName text,LastName text,Email varchar(100),Phone bigint,Resume blob);
Query OK, 0 rows affected (0.04 sec)

mysql> desc Applicants;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| ApplicantId | int          | NO   | PRI | NULL    |       |
| FirstName   | text         | YES  |     | NULL    |       |
| LastName    | text         | YES  |     | NULL    |       |
| Email       | varchar(100) | YES  |     | NULL    |       |
| Phone       | bigint       | YES  |     | NULL    |       |
| Resume      | blob         | YES  |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
6 rows in set (0.04 sec)
```

Applications

```
mysql> create table Applications(ApplicationId int primary key,JobId int,foreign key(JobId) references Jobs(JobId),ApplicantId int,foreign key(App
licantId) references Applicants(ApplicantId),ApplicationDate datetime,CoverLetter blob);
Query OK, 0 rows affected (0.09 sec)

mysql> desc Applications;
+-----------------+----------+------+-----+---------+-------+
| Field           | Type     | Null | Key | Default | Extra |
+-----------------+----------+------+-----+---------+-------+
| ApplicationId   | int      | NO   | PRI | NULL    |       |
| JobId           | int      | YES  | MUL | NULL    |       |
| ApplicantId     | int      | YES  | MUL | NULL    |       |
| ApplicationDate | datetime | YES  |     | NULL    |       |
| CoverLetter     | blob     | YES  |     | NULL    |       |
+-----------------+----------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

3.define primary key,foreign key

Already done

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

```
mysql> CREATE DATABASE IF NOT EXISTS CareerHub;
Query OK, 1 row affected, 1 warning (0.01 sec)

mysql> show databases;
+----------------------------+
| Database                   |
+----------------------------+
| careerhub                  |
| courier_management_system  |
| d1                         |
| hexaware                   |
| hmbank                     |
| information_schema         |
| mysql                      |
| performance_schema         |
| pet                        |
| petpals                    |
| sakila                     |
| sisdb                      |
| subquery                   |
| sys                        |
| techshop                   |
| ticketbookingsystem        |
| world                      |
+----------------------------+
17 rows in set (0.02 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 Jobs.JobTitle, COUNT(Applications.ApplicationID) AS ApplicationCount
    -> FROM Jobs
    -> LEFT JOIN Applications ON Jobs.JobID = Applications.JobID
    -> GROUP BY Jobs.JobID, Jobs.JobTitle
    -> ORDER BY Jobs.JobID;
+---------------------+------------------+
| JobTitle            | ApplicationCount |
+---------------------+------------------+
| Software Engineer   |                1 |
| Marketing Manager   |                1 |
| Data Analyst        |                1 |
| UX/UI Designer      |                1 |
| Sales Representative |                1 |
+---------------------+------------------+
5 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 Jobs.JobTitle, Companies.CompanyName, Jobs.JobLocation, Jobs.Salary
    -> FROM Jobs
    -> INNER JOIN Companies ON Jobs.CompanyID = Companies.CompanyID
    -> WHERE Jobs.Salary BETWEEN 70000 AND 80000;
+---------------------+------------------+-------------+----------+
| JobTitle            | CompanyName      | JobLocation | Salary   |
+---------------------+------------------+-------------+----------+
| Marketing Manager   | XYZ Company      | New York    | 80000.00 |
| Data Analyst        | Alpha Industries | Chicago     | 75000.00 |
| Sales Representative | Beta Enterprises | Los Angeles | 70000.00 |
+---------------------+------------------+-------------+----------+
3 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 Jobs.JobTitle, Companies.CompanyName, Applications.ApplicationDate
    -> FROM Applications
    -> INNER JOIN Jobs ON Applications.JobID = Jobs.JobID
    -> INNER JOIN Companies ON Jobs.CompanyID = Companies.CompanyID
    -> WHERE Applications.ApplicantID = 1;
+------------------+-------------+---------------------+
| JobTitle         | CompanyName | ApplicationDate     |
+------------------+-------------+---------------------+
| Software Engineer | ABC Corp    | 2023-01-15 08:00:00 |
+------------------+-------------+---------------------+
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 |
+---------------+
|  82000.000000 |
+---------------+
1 row in set (0.01 sec)

mysql>
```

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 CompanyName, COUNT(*) AS JobCount
    -> FROM Companies
    -> INNER JOIN Jobs ON Companies.CompanyID = Jobs.CompanyID
    -> GROUP BY Companies.CompanyID, Companies.CompanyName
    -> HAVING COUNT(*) = (
    ->     SELECT COUNT(*) AS MaxJobCount
    ->     FROM Jobs
    ->     GROUP BY CompanyID
    ->     ORDER BY MaxJobCount DESC
    ->     LIMIT 1
    -> );
+-------------+----------+
| CompanyName | JobCount |
+-------------+----------+
| ABC Corp    |        2 |
+-------------+----------+
1 row 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.FirstName, A.LastName, A.Email, A.Phone
    -> FROM Applicants A
    -> INNER JOIN Applications Ap ON A.ApplicantID = Ap.ApplicantID
    -> INNER JOIN Jobs J ON Ap.JobID = J.JobID
    -> INNER JOIN Companies C ON J.CompanyID = C.CompanyID
    -> WHERE C.Location = 'New York'
    ->   AND A.ExperienceYear >= 3;
+-----------+----------+------------------+------------+
| FirstName | LastName | Email            | Phone      |
+-----------+----------+------------------+------------+
| John      | Doe      | johndoe@email.com | 1234567890 |
+-----------+----------+------------------+------------+
1 row in 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 Manager    |
| Data Analyst         |
| Sales Representative |
+----------------------+
3 rows in set (0.00 sec)
```

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

```
mysql> SELECT Jobs.JobID, Jobs.JobTitle
    -> FROM Jobs
    -> LEFT JOIN Applications ON Jobs.JobID = Applications.JobID
    -> WHERE Applications.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.FirstName, A.LastName, C.CompanyName, J.JobTitle
    -> FROM Applicants A
    -> INNER JOIN Applications Ap ON A.ApplicantID = Ap.ApplicantID
    -> INNER JOIN Jobs J ON Ap.JobID = J.JobID
    -> INNER JOIN Companies C ON J.CompanyID = C.CompanyID;
+-----------+----------+-----------------+----------------------+
| FirstName | LastName | CompanyName     | JobTitle             |
+-----------+----------+-----------------+----------------------+
| John      | Doe      | ABC Corp        | Software Engineer    |
| Alice     | Smith    | XYZ Company     | Marketing Manager    |
| Bob       | Johnson  | Alpha Industries | Data Analyst         |
| Emily     | Brown    | ABC Corp        | UX/UI Designer       |
| Sarah     | Garcia   | Beta Enterprises | Sales Representative |
+-----------+----------+-----------------+----------------------+
5 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 Companies.CompanyName, COUNT(Jobs.JobID) AS PostedJobsCount
   -> FROM Companies
   -> LEFT JOIN Jobs ON Companies.CompanyID = Jobs.CompanyID
   -> GROUP BY Companies.CompanyID, Companies.CompanyName;
+-----------------+-----------------+
| CompanyName     | PostedJobsCount |
+-----------------+-----------------+
| ABC Corp        |               2 |
| XYZ Company     |               1 |
| Alpha Industries |               1 |
| Beta Enterprises |               1 |
| Gamma Co.       |               0 |
+-----------------+-----------------+
5 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.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;
+-----------+----------+-----------------+----------------------+
| FirstName | LastName | CompanyName     | JobTitle             |
+-----------+----------+-----------------+----------------------+
| John      | Doe      | ABC Corp        | Software Engineer    |
| Alice     | Smith    | XYZ Company     | Marketing Manager    |
| Bob       | Johnson  | Alpha Industries | Data Analyst         |
| Emily     | Brown    | ABC Corp        | UX/UI Designer       |
| Sarah     | Garcia   | Beta Enterprises | Sales Representative |
+-----------+----------+-----------------+----------------------+
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 DISTINCT C.CompanyName
    -> FROM Companies C
    -> INNER JOIN Jobs J ON C.CompanyID = J.CompanyID
    -> WHERE J.Salary > (
    ->     SELECT AVG(Salary) AS AvgSalary
    ->     FROM Jobs
    -> );
+-------------+
| CompanyName |
+-------------+
| ABC Corp    |
+-------------+
1 row 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 state text;
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> update applicants set city='Gwalior',state='mp' where applicantsid=1;
ERROR 1054 (42S22): Unknown column 'applicantsid' in 'where clause'
mysql> update applicants set city='Gwalior',state='mp' where applicantid=1;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> update applicants set city='Chennai',state='tn' where applicantid=1;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> update applicants set city='Gwalior',state='mp' where applicantid=2;
Query OK, 1 row affected (0.04 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> update applicants set city='Lucknow',state='up' where applicantid=3;
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> update applicants set city='Agra',state='up' where applicantid=4;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> update applicants set city='Bangalore',state='karnataka' where applicantid=5;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select firstname,lastname, concat(city,' ',state) as location from applicants;
+-----------+----------+----------------------+
| firstname | lastname | location             |
+-----------+----------+----------------------+
| John      | Doe      | Chennai tn           |
| Alice     | Smith    | Gwalior mp           |
| Bob       | Johnson  | Lucknow up           |
| Emily     | Brown    | Agra up              |
| Sarah     | Garcia   | Bangalore karnataka  |
+-----------+----------+----------------------+
5 rows in set (0.01 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 software applications | San Francisco | 95000.00 | Full-time | 2023-12-01 00:00:00 |
+-------+-----------+------------------+--------------------------------+--------------+----------+-----------+---------------------+
1 row 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.FirstName, A.LastName, J.JobTitle, C.CompanyName
    -> 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
    ->
    -> UNION
    ->
    -> SELECT A.FirstName, A.LastName, NULL AS JobTitle, NULL AS CompanyName
    -> FROM Applicants A
    -> WHERE NOT EXISTS (
    ->     SELECT 1
    ->     FROM Applications Ap
    ->     WHERE A.ApplicantID = Ap.ApplicantID
    -> )
    -> UNION
    ->
    -> SELECT NULL AS FirstName, NULL AS LastName, J.JobTitle, C.CompanyName
    -> FROM Jobs J
    -> LEFT JOIN Applications Ap ON J.JobID = Ap.JobID
    -> LEFT JOIN Companies C ON J.CompanyID = C.CompanyID
    -> WHERE NOT EXISTS (
    ->     SELECT 1
    ->     FROM Applications Ap
    ->     WHERE J.JobID = Ap.JobID
    -> )
    -> ;
+-----------+----------+----------------------+------------------+
| FirstName | LastName | JobTitle             | CompanyName      |
+-----------+----------+----------------------+------------------+
| John      | Doe      | Software Engineer    | ABC Corp         |
| Alice     | Smith    | Marketing Manager    | XYZ Company      |
| Bob       | Johnson  | Data Analyst         | Alpha Industries |
| Emily     | Brown    | UX/UI Designer       | ABC Corp         |
| Sarah     | Garcia   | Sales Representative | Beta Enterprises |
+-----------+----------+----------------------+------------------+
```

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
    -> CROSS JOIN Companies C
    -> WHERE C.Location = 'Chennai'
    ->   AND A.ExperienceYear > 2;
Empty set (0.00 sec)
```