

## Coding Challenges: CareerHub, The Job Board

By-Kunal Sudhir Mishra

Create SQL Schema from the application, use the class attributes for table column names.

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

```
mysql> create database CareerHub;
Query OK, 1 row affected (0.01 sec)

mysql> use CareerHub;
Database changed
```

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

```
mysql> CREATE TABLE Companies (
    ->     CompanyID INT PRIMARY KEY,
    ->     CompanyName VARCHAR(255),
    ->     Location VARCHAR(255)
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql>
mysql> CREATE TABLE Jobs (
    ->     JobID INT PRIMARY KEY,
    ->     CompanyID INT,
    ->     JobTitle VARCHAR(255),
    ->     JobDescription TEXT,
    ->     JobLocation VARCHAR(255),
    ->     Salary DECIMAL(10, 2),
    ->     JobType VARCHAR(255),
    ->     PostedDate DATETIME,
    ->     FOREIGN KEY (CompanyID) REFERENCES Companies(CompanyID)
    -> );
Query OK, 0 rows affected (0.03 sec)
```

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

mysql>
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.

```
mysql> desc Companies;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| CompanyID | int | NO | PRI | NULL | 
| CompanyName | varchar(255) | YES | | NULL | 
| Location | varchar(255) | YES | | NULL | 
+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)

mysql> desc Jobs;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| JobID | int | NO | PRI | NULL | 
| CompanyID | int | YES | MUL | NULL | 
| JobTitle | varchar(255) | YES | | NULL | 
| JobDescription | text | YES | | NULL | 
| JobLocation | varchar(255) | YES | | NULL | 
| Salary | decimal(10,2) | YES | | NULL | 
| JobType | varchar(255) | YES | | NULL | 
| PostedDate | datetime | YES | | NULL | 
+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

```
mysql> desc Applicants;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| ApplicantID | int | NO | PRI | NULL | 
| FirstName | varchar(255) | YES | | NULL | 
| LastName | varchar(255) | YES | | NULL | 
| Email | varchar(255) | YES | | NULL | 
| Phone | varchar(255) | YES | | NULL | 
| Resume | text | YES | | NULL | 
+-----+-----+-----+-----+-----+
6 rows in set (0.00 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 | text | YES | | NULL | 
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

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

```
mysql> CREATE TABLE IF NOT EXISTS Companies (
    ->     CompanyID INT PRIMARY KEY,
    ->     CompanyName VARCHAR(255),
    ->     Location VARCHAR(255)
    -> );
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql>
mysql> CREATE TABLE IF NOT EXISTS Jobs (
    ->     JobID INT PRIMARY KEY,
    ->     CompanyID INT,
    ->     JobTitle VARCHAR(255),
    ->     JobDescription TEXT,
    ->     JobLocation VARCHAR(255),
    ->     Salary DECIMAL(10, 2),
    ->     JobType VARCHAR(255),
    ->     PostedDate DATETIME,
    ->     FOREIGN KEY (CompanyID) REFERENCES Companies(CompanyID)
    -> );
Query OK, 0 rows affected, 1 warning (0.00 sec)
```

```
mysql> CREATE TABLE IF NOT EXISTS Applicants (
    ->     ApplicantID INT PRIMARY KEY,
    ->     FirstName VARCHAR(255),
    ->     LastName VARCHAR(255),
    ->     Email VARCHAR(255),
    ->     Phone VARCHAR(255),
    ->     Resume TEXT
    -> );
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql>
mysql> 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)
    -> );
Query OK, 0 rows affected, 1 warning (0.00 sec)
```

### Sample Data Input

```
mysql> INSERT INTO Companies (CompanyID, CompanyName, Location)
-> VALUES
-> (1, 'CompanyA', 'CityX'),
-> (2, 'CompanyB', 'CityY'),
-> (3, 'CompanyC', 'CityZ'),
-> (4, 'CompanyD', 'CityX'),
-> (5, 'CompanyE', 'CityY'),
-> (6, 'CompanyF', 'CityZ'),
-> (7, 'CompanyG', 'CityX'),
-> (8, 'CompanyH', 'CityY'),
-> (9, 'CompanyI', 'CityZ'),
-> (10, 'CompanyJ', 'CityX');
Query OK, 10 rows affected (0.01 sec)
Records: 10 Duplicates: 0 Warnings: 0

mysql>
mysql> INSERT INTO Jobs (JobID, CompanyID, JobTitle, JobDescription, JobLocation, Salary, JobType, PostedDate)
-> VALUES
-> (101, 1, 'Software Developer', 'Develop software applications', 'CityX', 70000.00, 'Full-Time', '2024-01-15'),
-> (102, 2, 'Data Scientist', 'Analyzing and interpreting complex data sets', 'CityY', 80000.00, 'Full-Time', '2024-02-01'),
-> (103, 1, 'Network Engineer', 'Design and implement computer networks', 'CityX', 75000.00, 'Part-Time', '2024-02-10'),
-> (104, 3, 'Database Administrator', 'Manage and maintain databases', 'CityZ', 85000.00, 'Full-Time', '2024-02-20'),
-> (105, 4, 'UX Designer', 'Design user experiences for web applications', 'CityX', 70000.00, 'Full-Time', '2024-03-01'),
-> (106, 5, 'IT Support Specialist', 'Provide technical support to end-users', 'CityY', 60000.00, 'Part-Time', '2024-03-10'),
-> (107, 6, 'Software Engineer', 'Develop scalable software solutions', 'CityZ', 72000.00, 'Full-Time', '2024-03-20'),
-> (108, 7, 'Web Developer', 'Build responsive web applications', 'CityX', 68000.00, 'Full-Time', '2024-04-01'),
-> (109, 8, 'Network Administrator', 'Configure and maintain network systems', 'CityY', 78000.00, 'Part-Time', '2024-04-10'),
-> (110, 9, 'Business Analyst', 'Analyze business processes and requirements', 'CityZ', 76000.00, 'Full-Time', '2024-04-20');
Query OK, 10 rows affected (0.00 sec)
Records: 10 Duplicates: 0 Warnings: 0
```

```
mysql>
mysql> INSERT INTO Applicants (ApplicantID, FirstName, LastName, Email, Phone, Resume)
-> VALUES
-> (201, 'Aarav', 'Kumar', 'aarav.kumar@example.com', '1234567890', 'Aarav_Resume'),
-> (202, 'Aisha', 'Patel', 'aisha.patel@example.com', '9876543210', 'Aisha_Resume'),
-> (203, 'Arjun', 'Sharma', 'arjun.sharma@example.com', '5551234567', 'Arjun_Resume'),
-> (204, 'Ananya', 'Singh', 'ananya.singh@example.com', '1112223333', 'Ananya_Resume'),
-> (205, 'Aditi', 'Verma', 'aditi.verma@example.com', '4445556666', 'Aditi_Resume'),
-> (206, 'Advait', 'Chopra', 'advait.chopra@example.com', '7778889999', 'Advait_Resume'),
-> (207, 'Aaradhya', 'Gupta', 'aaradhya.gupta@example.com', '2223334444', 'Aaradhya_Resume'),
-> (208, 'Aryan', 'Malhotra', 'aryan.malhotra@example.com', '6667778888', 'Aryan_Resume'),
-> (209, 'Ishita', 'Verma', 'ishita.verma@example.com', '9990001111', 'Ishita_Resume'),
-> (210, 'Vivaan', 'Nair', 'vivaan.nair@example.com', '3334445555', 'Vivaan_Resume');
Query OK, 10 rows affected (0.01 sec)
Records: 10 Duplicates: 0 Warnings: 0
```

```
mysql>
mysql>
mysql> INSERT INTO Applications (ApplicationID, JobID, ApplicantID, ApplicationDate, CoverLetter)
-> VALUES
-> (301, 101, 201, '2024-01-20', 'Aarav_CoverLetter'),
-> (302, 102, 202, '2024-02-05', 'Aisha_CoverLetter'),
-> (303, 101, 203, '2024-02-15', 'Arjun_CoverLetter'),
-> (304, 103, 204, '2024-03-15', 'Ananya_CoverLetter'),
-> (305, 104, 205, '2024-03-20', 'Aditi_CoverLetter'),
-> (306, 105, 206, '2024-04-01', 'Advait_CoverLetter'),
-> (307, 106, 207, '2024-04-10', 'Aaradhya_CoverLetter'),
-> (308, 107, 208, '2024-04-20', 'Aryan_CoverLetter'),
-> (309, 108, 209, '2024-05-01', 'Ishita_CoverLetter'),
-> (310, 109, 210, '2024-05-10', 'Vivaan_CoverLetter');
Query OK, 10 rows affected (0.00 sec)
Records: 10 Duplicates: 0 Warnings: 0
```

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
    -> ORDER BY
    ->     J.JobID;
+-----+-----+-----+
| JobID | JobTitle          | ApplicationCount |
+-----+-----+-----+
| 101   | Software Developer | 2                |
| 102   | Data Scientist      | 1                |
| 103   | Network Engineer    | 1                |
| 104   | Database Administrator| 1                |
| 105   | UX Designer          | 1                |
| 106   | IT Support Specialist| 1                |
| 107   | Software Engineer    | 1                |
| 108   | Web Developer         | 1                |
| 109   | Network Administrator| 1                |
| 110   | Business Analyst     | 0                |
+-----+-----+-----+
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> DELIMITER ##
mysql> CREATE PROCEDURE GetJobListingsInSalaryRange(IN minSalary DECIMAL(10, 2), IN maxSalary DECIMAL(10, 2))
    -> BEGIN
    ->     SELECT
    ->         J.JobTitle,
    ->         C.CompanyName,
    ->         J.JobLocation,
    ->         J.Salary
    ->     FROM
    ->         Jobs J
    ->     JOIN
    ->         Companies C ON J.CompanyID = C.CompanyID
    ->     WHERE
    ->         J.Salary BETWEEN minSalary AND maxSalary;
    -> END ##
Query OK, 0 rows affected (0.02 sec)

mysql> DELIMITER ;
mysql> CALL GetJobListingsInSalaryRange(60000.00, 80000.00);
```

```

mysql> CALL GetJobListingsInSalaryRange(60000.00, 80000.00);
+-----+-----+-----+-----+
| JobTitle      | CompanyName | JobLocation | Salary   |
+-----+-----+-----+-----+
| Software Developer | CompanyA | CityX | 70000.00 |
| Data Scientist | CompanyB | CityY | 80000.00 |
| Network Engineer | CompanyA | CityX | 75000.00 |
| UX Designer | CompanyD | CityX | 70000.00 |
| IT Support Specialist | CompanyE | CityY | 60000.00 |
| Software Engineer | CompanyF | CityZ | 72000.00 |
| Web Developer | CompanyG | CityX | 68000.00 |
| Network Administrator | CompanyH | CityY | 78000.00 |
| Business Analyst | CompanyI | CityZ | 76000.00 |
+-----+-----+-----+-----+
9 rows in set (0.00 sec)

Query OK, 0 rows affected (0.01 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> DELIMITER ##
mysql> CREATE PROCEDURE GetApplicationHistory(IN applicantID INT)
    -> BEGIN
    ->     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 = applicantID;
    -> END ##
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;
mysql> CALL GetApplicationHistory(201);
+-----+-----+-----+
| JobTitle      | CompanyName | ApplicationDate |
+-----+-----+-----+
| Software Developer | CompanyA | 2024-01-20 00:00:00 |
+-----+-----+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (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 |
+-----+
| 73400.000000 |
+-----+
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> 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
->     ORDER BY
->     JobCount DESC
->     LIMIT 1;
+-----+-----+
| CompanyName | JobCount |
+-----+-----+
| CompanyA    |      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> UPDATE Applicants SET Experience = 2 WHERE ApplicantID IN (201, 202, 203, 204, 205);
Query OK, 5 rows affected (0.01 sec)
Rows matched: 5  Changed: 5  Warnings: 0

mysql> UPDATE Applicants SET Experience = 4 WHERE ApplicantID IN (206, 207, 208, 209, 210);
Query OK, 5 rows affected (0.01 sec)
Rows matched: 5  Changed: 5  Warnings: 0

mysql> SELECT
->     A.*
->     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 A.Experience >= 3;
+-----+-----+-----+-----+-----+-----+-----+
| ApplicantID | FirstName | LastName | Email           | Phone      | Resume    | Experience |
+-----+-----+-----+-----+-----+-----+-----+
|     206 | Advait   | Chopra   | advait.chopra@example.com | 7778889999 | Advait_Resume |      4 |
|     209 | Ishita   | Verma    | ishita.verma@example.com | 9990001111 | Ishita_Resume |      4 |
+-----+-----+-----+-----+-----+-----+-----+
2 rows 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          |
+-----+
| Software Developer |
| Data Scientist    |
| Network Engineer  |
| UX Designer       |
| IT Support Specialist |
| Software Engineer |
| Web Developer     |
| Network Administrator |
| Business Analyst  |
+-----+
9 rows in set (0.00 sec)
```

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

```
mysql> SELECT
    ->     J.*
    ->   FROM
    ->   Jobs J
    -> LEFT JOIN
    ->   Applications A ON J.JobID = A.JobID
    -> WHERE
    ->   A.ApplicationID IS NULL;
+-----+-----+-----+-----+-----+-----+
| JobID | CompanyID | JobTitle      | JobDescription          | JobLocation | Salary    | JobType   | PostedDate
e       |
+-----+-----+-----+-----+-----+-----+
| 110   |         9 | Business Analyst | Analyze business processes and requirements | CityZ      | 76000.00  | Full-Time | 2024-04-2
0 00:00:00 |
+-----+-----+-----+-----+-----+-----+
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.JobTitle,
    ->     AP.ApplicationDate
    ->   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;
+-----+-----+-----+-----+-----+
| FirstName | LastName | CompanyName | JobTitle      | ApplicationDate |
+-----+-----+-----+-----+-----+
| Aarav     | Kumar    | CompanyA    | Software Developer | 2024-01-20 00:00:00 |
| Aisha     | Patel    | CompanyB    | Data Scientist   | 2024-02-05 00:00:00 |
| Arjun     | Sharma   | CompanyA    | Software Developer | 2024-02-15 00:00:00 |
| Ananya   | Singh    | CompanyA    | Network Engineer | 2024-03-15 00:00:00 |
| Aditi     | Verma    | CompanyC    | Database Administrator | 2024-03-20 00:00:00 |
| Advait    | Chopra   | CompanyD    | UX Designer     | 2024-04-01 00:00:00 |
| Aaradhyaa | Gupta    | CompanyE    | IT Support Specialist | 2024-04-10 00:00:00 |
| Aryan     | Malhotra  | CompanyF    | Software Engineer | 2024-04-20 00:00:00 |
| Ishita    | Verma    | CompanyG    | Web Developer   | 2024-05-01 00:00:00 |
| Vivaan    | Nair     | CompanyH    | Network Administrator | 2024-05-10 00:00:00 |
+-----+-----+-----+-----+-----+
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.CompanyName,
    ->     COUNT(J.JobID) AS JobsPosted
    -> FROM
    ->     Companies C
    -> LEFT JOIN
    ->     Jobs J ON C.CompanyID = J.CompanyID
    -> LEFT JOIN
    ->     Applications A ON J.JobID = A.JobID
    -> GROUP BY
    ->     C.CompanyName;
+-----+-----+
| CompanyName | JobsPosted |
+-----+-----+
| CompanyA    |      3 |
| CompanyB    |      1 |
| CompanyC    |      1 |
| CompanyD    |      1 |
| CompanyE    |      1 |
| CompanyF    |      1 |
| CompanyG    |      1 |
| CompanyH    |      1 |
| CompanyI    |      1 |
| CompanyJ    |      0 |
+-----+-----+
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.FirstName, A.LastName, C.CompanyName, J.JobTitle, AP.ApplicationDate
    -> FROM Applicants A
    -> JOIN Jobs J ON TRUE
    -> LEFT JOIN Applications AP ON A.ApplicantID = AP.ApplicantID AND J.JobID = AP.JobID
    -> LEFT JOIN Companies C ON J.CompanyID = C.CompanyID;
```

FirstName	LastName	CompanyName	JobTitle	ApplicationDate
Vivaan	Nair	CompanyA	Software Developer	NULL
Ishita	Verma	CompanyA	Software Developer	NULL
Aryan	Malhotra	CompanyA	Software Developer	NULL
Aaradhya	Gupta	CompanyA	Software Developer	NULL
Advait	Chopra	CompanyA	Software Developer	NULL
Aditi	Verma	CompanyA	Software Developer	NULL
Ananya	Singh	CompanyA	Software Developer	NULL
Arjun	Sharma	CompanyA	Software Developer	2024-02-15 00:00:00
Aisha	Patel	CompanyA	Software Developer	NULL
Aarav	Kumar	CompanyA	Software Developer	2024-01-20 00:00:00
Vivaan	Nair	CompanyB	Data Scientist	NULL
Ishita	Verma	CompanyB	Data Scientist	NULL
Aryan	Malhotra	CompanyB	Data Scientist	NULL
Aaradhya	Gupta	CompanyB	Data Scientist	NULL
Advait	Chopra	CompanyB	Data Scientist	NULL
Aditi	Verma	CompanyB	Data Scientist	NULL
Ananya	Singh	CompanyB	Data Scientist	NULL
Arjun	Sharma	CompanyB	Data Scientist	NULL
Aisha	Patel	CompanyB	Data Scientist	2024-02-05 00:00:00
Aarav	Kumar	CompanyB	Data Scientist	NULL
Vivaan	Nair	CompanyA	Network Engineer	NULL
Ishita	Verma	CompanyA	Network Engineer	NULL
Aryan	Malhotra	CompanyA	Network Engineer	NULL
Aaradhya	Gupta	CompanyA	Network Engineer	NULL
Advait	Chopra	CompanyA	Network Engineer	NULL
Aditi	Verma	CompanyA	Network Engineer	NULL
Ananya	Singh	CompanyA	Network Engineer	2024-03-15 00:00:00
Arjun	Sharma	CompanyA	Network Engineer	NULL
Aisha	Patel	CompanyA	Network Engineer	NULL
Aarav	Kumar	CompanyA	Network Engineer	NULL
Vivaan	Nair	CompanyC	Database Administrator	NULL
Ishita	Verma	CompanyC	Database Administrator	NULL
Aryan	Malhotra	CompanyC	Database Administrator	NULL
Aaradhya	Gupta	CompanyC	Database Administrator	NULL
Advait	Chopra	CompanyC	Database Administrator	NULL
Aditi	Verma	CompanyC	Database Administrator	2024-03-20 00:00:00
Ananya	Singh	CompanyC	Database Administrator	NULL
Arjun	Sharma	CompanyC	Database Administrator	NULL
Aisha	Patel	CompanyC	Database Administrator	NULL
Aarav	Kumar	CompanyC	Database Administrator	NULL
Vivaan	Nair	CompanyD	UX Designer	NULL
Ishita	Verma	CompanyD	UX Designer	NULL
Aryan	Malhotra	CompanyD	UX Designer	NULL
Aaradhya	Gupta	CompanyD	UX Designer	NULL
Advait	Chopra	CompanyD	UX Designer	2024-04-01 00:00:00
Aditi	Verma	CompanyD	UX Designer	NULL
Ananya	Singh	CompanyD	UX Designer	NULL
Arjun	Sharma	CompanyD	UX Designer	NULL
Aisha	Patel	CompanyD	UX Designer	NULL
Aarav	Kumar	CompanyD	UX Designer	NULL
Vivaan	Nair	CompanyE	IT Support Specialist	NULL
Ishita	Verma	CompanyE	IT Support Specialist	NULL
Aryan	Malhotra	CompanyE	IT Support Specialist	NULL
Aaradhya	Gupta	CompanyE	IT Support Specialist	2024-04-10 00:00:00
Advait	Chopra	CompanyE	IT Support Specialist	NULL
Aditi	Verma	CompanyE	IT Support Specialist	NULL
Ananya	Singh	CompanyE	IT Support Specialist	NULL
Arjun	Sharma	CompanyE	IT Support Specialist	NULL
Aisha	Patel	CompanyE	IT Support Specialist	NULL
Aarav	Kumar	CompanyE	IT Support Specialist	NULL
Vivaan	Nair	CompanyF	Software Engineer	NULL
Ishita	Verma	CompanyF	Software Engineer	NULL

Aryan	Malhotra	CompanyF	Software Engineer	2024-04-20 00:00:00
Aaradhyा	Gupta	CompanyF	Software Engineer	NULL
Advait	Chopra	CompanyF	Software Engineer	NULL
Aditi	Verma	CompanyF	Software Engineer	NULL
Ananya	Singh	CompanyF	Software Engineer	NULL
Arjun	Sharma	CompanyF	Software Engineer	NULL
Aisha	Patel	CompanyF	Software Engineer	NULL
Aarav	Kumar	CompanyF	Software Engineer	NULL
Vivaan	Nair	CompanyG	Web Developer	NULL
Ishita	Verma	CompanyG	Web Developer	2024-05-01 00:00:00
Aryan	Malhotra	CompanyG	Web Developer	NULL
Aaradhyा	Gupta	CompanyG	Web Developer	NULL
Advait	Chopra	CompanyG	Web Developer	NULL
Aditi	Verma	CompanyG	Web Developer	NULL
Ananya	Singh	CompanyG	Web Developer	NULL
Arjun	Sharma	CompanyG	Web Developer	NULL
Aisha	Patel	CompanyG	Web Developer	NULL
Aarav	Kumar	CompanyG	Web Developer	NULL
Vivaan	Nair	CompanyH	Network Administrator	2024-05-10 00:00:00
Ishita	Verma	CompanyH	Network Administrator	NULL
Aryan	Malhotra	CompanyH	Network Administrator	NULL
Aaradhyा	Gupta	CompanyH	Network Administrator	NULL
Advait	Chopra	CompanyH	Network Administrator	NULL
Aditi	Verma	CompanyH	Network Administrator	NULL
Ananya	Singh	CompanyH	Network Administrator	NULL
Arjun	Sharma	CompanyH	Network Administrator	NULL
Aisha	Patel	CompanyH	Network Administrator	NULL
Aarav	Kumar	CompanyH	Network Administrator	NULL
Vivaan	Nair	CompanyI	Business Analyst	NULL
Ishita	Verma	CompanyI	Business Analyst	NULL
Aryan	Malhotra	CompanyI	Business Analyst	NULL
Aaradhyा	Gupta	CompanyI	Business Analyst	NULL
Advait	Chopra	CompanyI	Business Analyst	NULL
Aditi	Verma	CompanyI	Business Analyst	NULL
Ananya	Singh	CompanyI	Business Analyst	NULL
Arjun	Sharma	CompanyI	Business Analyst	NULL
Aisha	Patel	CompanyI	Business Analyst	NULL
Aarav	Kumar	CompanyI	Business Analyst	NULL

100 rows in set (0.00 sec)

mysql> |

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
    ->     JOIN
    ->     Jobs J ON C.CompanyID = J.CompanyID
    ->     WHERE
    ->     J.Salary > (SELECT AVG(Salary) FROM Jobs);
+-----+
| CompanyName |
+-----+
| CompanyB   |
| CompanyA   |
| CompanyC   |
| CompanyH   |
| CompanyI   |
+-----+
5 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> UPDATE Applicants SET City = 'Chennai', State = 'Tamil Nadu' WHERE ApplicantID IN (201, 202, 203, 204, 205);
Query OK, 5 rows affected (0.00 sec)
Rows matched: 5 Changed: 5 Warnings: 0

mysql> UPDATE Applicants SET City = 'Mumbai', State = 'Maharashtra' WHERE ApplicantID IN (206, 207, 208, 209, 210);
Query OK, 5 rows affected (0.01 sec)
Rows matched: 5 Changed: 5 Warnings: 0

mysql> SELECT
    ->     CONCAT(A.FirstName, ' ', A.LastName) AS ApplicantName,
    ->     CONCAT(A.City, ' ', A.State) AS CityAndState
    ->     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;
+-----+
| ApplicantName | CityAndState      |
+-----+
| Aarav Kumar   | Chennai, Tamil Nadu |
| Aisha Patel   | Chennai, Tamil Nadu |
| Arjun Sharma  | Chennai, Tamil Nadu |
| Ananya Singh  | Chennai, Tamil Nadu |
| Aditi Verma   | Chennai, Tamil Nadu |
| Advait Chopra | Mumbai, Maharashtra |
| Aaradhya Gupta | Mumbai, Maharashtra |
| Aryan Malhotra | Mumbai, Maharashtra |
| Ishita Verma  | Mumbai, Maharashtra |
| Vivaan Nair   | Mumbai, Maharashtra |
+-----+
10 rows in set (0.00 sec)
```

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

```
mysql> SELECT JobID, JobTitle, JobDescription, JobLocation, Salary, JobType, PostedDate FROM Jobs WHERE JobTitle LIKE '%Developer%' OR JobTitle LIKE '%Engineer%';
+-----+-----+-----+-----+-----+-----+-----+
| JobID | JobTitle      | JobDescription          | JobLocation | Salary   | JobType    | PostedDate |
+-----+-----+-----+-----+-----+-----+-----+
| 101  | Software Developer | Develop software applications | CityX       | 70000.00 | Full-Time  | 2024-01-15 00:00:00 |
| 103  | Network Engineer  | Design and implement computer networks | CityX       | 75000.00 | Part-Time  | 2024-02-10 00:00:00 |
| 107  | Software Engineer | Develop scalable software solutions | CityZ       | 72000.00 | Full-Time  | 2024-03-20 00:00:00 |
| 108  | Web Developer    | Build responsive web applications | CityX       | 68000.00 | Full-Time  | 2024-04-01 00:00:00 |
+-----+-----+-----+-----+-----+-----+-----+
4 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.FirstName,
    ->     A.LastName,
    ->     J.JobTitle,
    ->     A2.ApplicationDate
    -> FROM
    ->     Applicants A
    -> LEFT JOIN
    ->     Applications A2 ON A.ApplicantID = A2.ApplicantID
    -> LEFT JOIN
    ->     Jobs J ON A2.JobID = J.JobID;
+-----+-----+-----+-----+
| FirstName | LastName | JobTitle          | ApplicationDate |
+-----+-----+-----+-----+
| Aarav     | Kumar    | Software Developer | 2024-01-20 00:00:00 |
| Aisha     | Patel    | Data Scientist    | 2024-02-05 00:00:00 |
| Arjun     | Sharma   | Software Developer | 2024-02-15 00:00:00 |
| Ananya   | Singh    | Network Engineer  | 2024-03-15 00:00:00 |
| Aditi     | Verma    | Database Administrator | 2024-03-20 00:00:00 |
| Advait    | Chopra   | UX Designer        | 2024-04-01 00:00:00 |
| Aaradhyaa | Gupta    | IT Support Specialist | 2024-04-10 00:00:00 |
| Aryan     | Malhotra  | Software Engineer  | 2024-04-20 00:00:00 |
| Ishита   | Verma    | Web Developer      | 2024-05-01 00:00:00 |
| Vivaan    | Nair     | Network Administrator | 2024-05-10 00:00:00 |
+-----+-----+-----+-----+
10 rows in set (0.00 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,
    ->     A.City AS ApplicantCity,
    ->     A.State AS ApplicantState,
    ->     A.Experience,
    ->     C.CompanyName,
    ->     C.Location AS CompanyCity
-> 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 > 2;
+-----+-----+-----+-----+-----+-----+-----+
| FirstName | LastName | ApplicantCity | ApplicantState | Experience | CompanyName | CompanyCity |
+-----+-----+-----+-----+-----+-----+-----+
| Advait   | Chopra   | Mumbai      | Maharashtra   |        4 | CompanyD   | Chennai
| Aaradhyा | Gupta    | Mumbai      | Maharashtra   |        4 | CompanyE   | Chennai
| Aryan    | Malhotra  | Mumbai      | Maharashtra   |        4 | CompanyF   | Chennai
| Ishita   | Verma    | Mumbai      | Maharashtra   |        4 | CompanyG   | Chennai
+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```