

SQL CODING CHALLENGE

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

create database CareerHub;

```
mysql> create database CareerHub;
Query OK, 1 row affected (0.10 sec)

mysql> use CareerHub;
Database changed
mysql> show tables;
Empty set (0.05 sec)
```

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

*create table Companies(
-> CompanyID INT Primary Key,
-> CompanyName Varchar(30),
-> Location Varchar(30));*

```
mysql> create table Companies(  
-> CompanyID INT Primary Key,  
-> CompanyName Varchar(30),  
-> Location Varchar(30));  
Query OK, 0 rows affected (0.09 sec)
```

*create table Jobs (
-> JobID INT primary key,
-> CompanyID INT,
-> JobTitle Varchar(30),
-> JobDescription Text,
-> JobLocation Varchar(30),
-> Salary Decimal(6, 2),
-> JobType Varchar(30),
-> PostedDate datetime);*

```
mysql> create table Jobs (  
-> JobID INT primary key,  
-> CompanyID INT,  
-> JobTitle Varchar(30),  
-> JobDescription Text,  
-> JobLocation Varchar(30),  
-> Salary Decimal(6, 2),  
-> JobType Varchar(30),  
-> PostedDate datetime);  
Query OK, 0 rows affected (0.04 sec)
```

```
create table Applicants(  
-> ApplicantID INT primary key,  
-> FirstName Varchar(30),  
-> LastName Varchar(30),  
-> Email Varchar(50),  
-> Phone Varchar(12),  
-> Resume Text);
```

```
mysql> create table Applicants(  
-> ApplicantID INT primary key,  
-> FirstName Varchar(30),  
-> LastName Varchar(30),  
-> Email Varchar(50),  
-> Phone Varchar(12),  
-> Resume Text);  
Query OK, 0 rows affected (0.02 sec)
```

```
create table Applications(  
-> ApplicationID INT primary key,  
-> JobID INT,  
-> ApplicantID INT,  
-> ApplicationDate datetime,  
-> CoverLetter Text);
```

```
mysql> create table Applications(  
    -> ApplicationID INT primary key,  
    -> JobID INT,  
    -> ApplicantID INT,  
    -> ApplicationDate datetime,  
    -> CoverLetter Text);  
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> show tables;  
+-----+  
| Tables_in_careerhub |  
+-----+  
| applicants          |  
| applications        |  
| companies           |  
| jobs                |  
+-----+  
4 rows in set (0.03 sec)
```

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

alter table Jobs

-> add foreign key (CompanyId) references Companies(CompanyId);

```
mysql> alter table Jobs  
    -> add foreign key (CompanyId) references Companies(CompanyId);  
Query OK, 0 rows affected (0.12 sec)  
Records: 0  Duplicates: 0  Warnings: 0
```

alter table Applications

-> add foreign key (JobID) references Jobs(JobID);

alter table Applications

-> add foreign key (ApplicantID) references Applicants(ApplicantID);

```
mysql> alter table Applications
      -> add foreign key (JobID) references Jobs(JobID);
Query OK, 0 rows affected (0.07 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> alter table Applications
      -> add foreign key (ApplicantID) references Applicants(ApplicantID);
Query OK, 0 rows affected (0.08 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

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

Create database CareerHub;

```
mysql> create database CareerHub;
ERROR 1007 (HY000): Can't create database 'careerhub'; database exists
```

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.*

```
select J.JobID, J.JobTitle, COUNT(ApplicationID)
      -> from Jobs J
      -> left join Applications A on J.JobID = A.JobID
      -> group by JobID;
```

```
mysql> select J.JobID, J.JobTitle, COUNT(ApplicationID)
      -> from Jobs J
      -> left join Applications A on J.JobID = A.JobID
      -> group by JobID;
```

JobID	JobTitle	COUNT(ApplicationID)
1	Software Developer	3
2	Data Analyst	2
3	Web Developer	2
4	Customer Support	0
5	Graphic Designer	0
6	Content Writer	0
7	Administrative Assistant	0

```
7 rows in set (0.01 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.

```
select J.jobtitle, C.companyname, J.joblocation, j.salary
-> from job J
-> join companies C on J.companyid = C.companyid
-> where J.salary between 60000 and 90000;
```

```
mysql> select J.jobtitle, C.companyname, J.joblocation, j.salary
-> from jobs J
-> join companies C on J.companyid = C.companyid
-> where J.salary between 60000 and 90000;
+-----+-----+-----+-----+
| jobtitle      | companyname      | joblocation | salary  |
+-----+-----+-----+-----+
| Software Developer | Hexaversity Technologies | Chennai    | 80000.00 |
| Data Analyst      | Hexaversity Technologies | Chennai    | 70000.00 |
| Web Developer     | SouthCode Solutions   | Bangalore   | 85000.00 |
+-----+-----+-----+-----+
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.*

```
select J.JobTitle, C.companyname, A.applicationdate
-> from Jobs J
-> join Companies C on J.CompanyID = C.CompanyID
-> join Applications A on J.JobID = A.JobID
-> where ApplicantID = '6';
```

```
mysql> select J.JobTitle, C.companyname, A.applicationdate
-> from Jobs J
-> join Companies C on J.CompanyID = C.CompanyID
-> join Applications A on J.JobID = A.JobID
-> where ApplicantID = '6';
+-----+-----+-----+
| JobTitle      | companyname      | applicationdate |
+-----+-----+-----+
| Web Developer | SouthCode Solutions | 2024-04-15 12:00:00 |
+-----+-----+-----+
1 row in set (0.01 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.*

```
select AVG(salary)
-> from Jobs
-> where salary > 0;
```

```
mysql> select AVG(salary)
-> from Jobs
-> where salary > 0;
+-----+
| AVG(salary) |
+-----+
| 66285.714286 |
+-----+
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.

```
select C.CompanyID, C.CompanyName, COUNT(*) as val
-> from jobs J
-> join Companies C on J.CompanyID = C.CompanyID
-> group by C.CompanyID, C.CompanyName
-> Order by val DESC
-> LIMIT 1;
```

```
mysql> select C.CompanyID, C.CompanyName, COUNT(*) as val
-> from jobs J
-> join Companies C on J.CompanyID = C.CompanyID
-> group by C.CompanyID, C.CompanyName
-> Order by val DESC
-> LIMIT 1;
+-----+-----+-----+
| CompanyID | CompanyName | val |
+-----+-----+-----+
| 1 | Hexavarsity Technologies | 3 |
+-----+-----+-----+
1 row in set (0.01 sec)
```

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

```

select A.ApplicantID, A.FirstName, A.LastName, C.CompanyName
-> from Applicants A
-> join Applications App on A.ApplicantID = App.ApplicantID
-> join Jobs J on App.JobID = J.JobID
-> join Companies C on J.CompanyID = C.CompanyID
-> where A.City = 'Chennai' AND year(App.ApplicationDate) < 2020;

```

```

mysql> select A.ApplicantID, A.FirstName, A.LastName, C.CompanyName
-> from Applicants A
-> join Applications App on A.ApplicantID = App.ApplicantID
-> join Jobs J on App.JobID = J.JobID
-> join Companies C on J.CompanyID = C.CompanyID
-> where A.City = 'Chennai' AND year(App.ApplicationDate) < 2020;
+-----+-----+-----+-----+
| ApplicantID | FirstName | LastName | CompanyName |
+-----+-----+-----+-----+
|          2 | Priya    | Sundaram | SouthCode Solutions |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

```

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;

```

```

mysql> select distinct JobTitle
-> from Jobs
-> where salary between 60000 and 80000;
+-----+
| JobTitle |
+-----+
| Software Developer |
| Data Analyst |
+-----+
2 rows in set (0.00 sec)

```

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

```

select JobTitle
-> from(
-> select J.JobTitle, COUNT(A.JobID) as ApplicationCount
-> from Jobs J
-> left join Applications A on J.JobID = A.JobID
-> group by J.JobID)

```

-> as K
-> where ApplicationCount = '0';

```
mysql> select JobTitle
      -> from(
      -> select J.JobTitle, COUNT(A.JobID) as ApplicationCount
      -> from Jobs J
      -> left join Applications A on J.JobID = A.JobID
      -> group by J.JobID)
      -> as K
      -> where ApplicationCount = '0';
+-----+
| JobTitle |
+-----+
| Customer Support |
| Graphic Designer |
| Content Writer |
| Administrative Assistant |
+-----+
4 rows 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.

```
select ApplicantID, FirstName, LastName, CompanyName, JobTitle
      -> from(
      -> select A1.ApplicantID, A1.FirstName, A1.LastName, C.CompanyName,
      J.JobTitle
      -> from Applicants A1
      -> join Applications A on A1.ApplicantID = A.ApplicantID
      -> join Jobs J on A.JobID = J.JobID
      -> join Companies C on J.CompanyID = C.CompanyID)
      -> as K;
```



```
mysql> select ApplicantID, FirstName, LastName, CompanyName, JobTitle
-> from(
-> select A1.ApplicantID, A1.FirstName, A1.LastName, C.CompanyName, J.JobTitle
-> from Applicants A1
-> join Applications A on A1.ApplicantID = A.ApplicantID
-> join Jobs J on A.JobID = J.JobID
-> join Companies C on J.CompanyID = C.CompanyID)
-> as K;
```

ApplicantID	FirstName	LastName	CompanyName	JobTitle
2	Priya	Sundaram	Hexavarsity Technologies	Software Developer
4	Deepika	Menon	Hexavarsity Technologies	Software Developer
7	Vijay	Sharma	Hexavarsity Technologies	Software Developer
3	Manoj	Rao	Hexavarsity Technologies	Data Analyst
5	Karthik	Nair	Hexavarsity Technologies	Data Analyst
1	Arjun	Kumar	SouthCode Solutions	Web Developer
6	Sneha	Iyer	SouthCode Solutions	Web Developer

7 rows in set (0.01 sec)

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.CompanyID, C.CompanyName, count(J.JobID) as number
-> from jobs J join companies C on C.CompanyID = J.CompanyID
-> group by C.CompanyID, C.CompanyName;
```

```
mysql> select C.CompanyID, C.CompanyName, count(J.JobID) as number
-> from jobs J join companies C on C.CompanyID = J.CompanyID
-> group by C.CompanyID, C.CompanyName;
```

CompanyID	CompanyName	number
1	Hexavarsity Technologies	3
2	SouthCode Solutions	2
3	TechTalent Innovations	2

3 rows in set (0.01 sec)

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

```
select A.ApplicantID, 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;
```

```
mysql> select A.ApplicantID, 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;
```

ApplicantID	FirstName	LastName	CompanyName	JobTitle
1	Arjun	Kumar	SouthCode Solutions	Web Developer
2	Priya	Sundaram	Hexavarsity Technologies	Software Developer
3	Manoj	Rao	Hexavarsity Technologies	Data Analyst
4	Deepika	Menon	Hexavarsity Technologies	Software Developer
5	Karthik	Nair	Hexavarsity Technologies	Data Analyst
6	Sneha	Iyer	SouthCode Solutions	Web Developer
7	Vijay	Sharma	Hexavarsity Technologies	Software Developer

```
7 rows in set (0.01 sec)
```

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
-> );
```

```
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
Hexavarsity Technologies
SouthCode Solutions

```
2 rows in set (0.02 sec)
```

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

```
select ApplicantID, FirstName, CONCAT(City, ' ', State)
-> from Applicants;
```

```
mysql> select ApplicantID, FirstName, CONCAT(City, ' ', State)
-> from Applicants;
```

ApplicantID	FirstName	CONCAT(City, ' ', State)
1	Arjun	Tadepalligudem AP
2	Priya	Chennai Tamil Nadu
3	Manoj	Bengaluru Karnataka
4	Deepika	Pune Maharashtra
5	Karthik	Bengaluru Karnataka
6	Sneha	Chennai Tamil Nadu
7	Vijay	Tadepalligudem AP

```
7 rows in set (0.01 sec)
```

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

```
select *
-> from Jobs
-> where JobTitle LIKE '%Developer%' OR JobTitle LIKE '%Engineer%';
```

```
mysql> select *
-> from Jobs
-> where JobTitle LIKE '%Developer%' OR JobTitle LIKE '%Engineer%';
```

JobID	CompanyID	JobTitle	JobDescription	JobLocation	salary	JobType	PostedDate
1	1	Software Developer	Develop software applications.	Chennai	80000.00	Full-time	2024-04-10 08:00:00
3	2	Web Developer	Design and develop websites.	Bangalore	85000.00	Full-time	2024-04-12 12:00:00

```
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.

```
select A.ApplicantID, A.FirstName, A.LastName, 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;
```

```
mysql> select A.ApplicantID, A.FirstName, A.LastName, 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;
```

ApplicantID	FirstName	LastName	JobTitle
1	Arjun	Kumar	Web Developer
2	Priya	Sundaram	Software Developer
3	Manoj	Rao	Data Analyst
4	Deepika	Menon	Software Developer
5	Karthik	Nair	Data Analyst
6	Sneha	Iyer	Web Developer
7	Vijay	Sharma	Software Developer

```
7 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

```
select A.FirstName, A.LastName, C.CompanyName, C.Location
-> from Applicants A
-> join Applications App on A.ApplicantID = App.ApplicantID
-> join Jobs J on App.JobID = J.JobID
-> join Companies C on J.CompanyID = C.CompanyID
-> where A.City = 'Chennai' AND DATEDIFF(CURRENT_DATE(),
App.ApplicationDate) >= 730;
```

```
mysql> select A.FirstName, A.LastName, C.CompanyName, C.Location
-> from Applicants A
-> join Applications App on A.ApplicantID = App.ApplicantID
-> join Jobs J on App.JobID = J.JobID
-> join Companies C on J.CompanyID = C.CompanyID
-> where A.City = 'Chennai' AND DATEDIFF(CURRENT_DATE(), App.ApplicationDate) >= 730;
```

FirstName	LastName	CompanyName	Location
Priya	Sundaram	SouthCode Solutions	Bangalore

```
1 row in set (0.01 sec)
```
