

CODING CHALLENGES: CAREERHUB, THE JOB BOARD

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 (0.01 sec)
```

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

Companies:

create table companies

(company_id int primary key auto_increment,

company_name varchar(50),

location varchar(100));

```
mysql> desc companies;  
+-----+-----+-----+-----+-----+-----+  
| Field      | Type          | Null | Key | Default | Extra          |  
+-----+-----+-----+-----+-----+-----+  
| company_id | int           | NO   | PRI | NULL    | auto_increment |  
| company_name | varchar(50)   | YES  |     | NULL    |                |  
| location    | varchar(100)  | YES  |     | NULL    |                |  
+-----+-----+-----+-----+-----+-----+  
3 rows in set (0.00 sec)
```

Jobs:

create table jobs (job_id int primary key auto_increment,

Company_id int, foreign key (company_id) references companies(company_id),

jobtitle varchar(30),

job_description text,

job_location varchar(50),

salary decimal (15,2) default 0.00,

job_type enum('full time','part time','contract'),

posted_date timestamp default current_timestamp);

```
mysql> desc jobs;
```

Field	Type	Null	Key	Default	Extra
job_id	int	NO	PRI	NULL	auto_increment
company_id	int	YES	MUL	NULL	
jobtitle	varchar(30)	YES		NULL	
job_description	text	YES		NULL	
job_location	varchar(50)	YES		NULL	
salary	decimal(15,2)	YES		0.00	
job_type	enum('full time','part time','contract')	YES		NULL	
posted_date	timestamp	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED

8 rows in set (0.00 sec)

Applicants:

create table applicants(applicant_id int primary key,

first_name varchar(20),

last_name varchar(20),

email varchar(50),

phone varchar(15),

resume text);

```
mysql> desc applicants;
```

Field	Type	Null	Key	Default	Extra
applicant_id	int	NO	PRI	NULL	
first_name	varchar(20)	YES		NULL	
last_name	varchar(20)	YES		NULL	
email	varchar(50)	YES		NULL	
phone	varchar(15)	YES		NULL	
resume	text	YES		NULL	

6 rows in set (0.00 sec)

Applications:

create table applications(application_id int primary key,

job_id int,foreign key(job_id) references jobs(job_id),

applicant_id int,foreign key(applicant_id) references applicants(applicant_id),

application_date timestamp default current_timestamp,

cover_letter text);

```
mysql> desc applications;
```

Field	Type	Null	Key	Default	Extra
application_id	int	NO	PRI	NULL	
job_id	int	YES	MUL	NULL	
applicant_id	int	YES	MUL	NULL	
application_date	timestamp	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED
cover_letter	text	YES		NULL	

```
5 rows in set (0.00 sec)
```

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

Primary keys, foreign keys and constraints are already defined.

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

I can use **if not exists** during the time of creation to check whether the database or table 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.

```
select j.jobtitle, count(a.application_id) as application_count
from jobs j
left join applications a on j.job_id = a.job_id
group by j.job_id, j.jobtitle
```

jobtitle	application_count
Software Engineer	1
Data Scientist	1
Cloud Engineer	1
IT Consultant	1
Cybersecurity Analyst	1
Frontend Developer	1
DevOps Engineer	1
AI/ML Engineer	1
Database Administrator	1
Embedded Systems 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.

```
set @min_salary = 200000;  
set @max_salary = 800000;  
select j.jobtitle, c.company_name, c.location, j.salary  
from jobs j  
join companies c on j.company_id = c.company_id  
where j.salary between @min_salary and @max_salary;
```

jobtitle	company_name	location	salary
Cloud Engineer	Microsoft	Hyderabad	500000.00
IT Consultant	Amazon	Bangalore	750000.00
Cybersecurity Analyst	TCS	Mumbai	200000.00
Frontend Developer	Infosys	Pune	650000.00
AI/ML Engineer	IBM	Bangalore	300000.00
Embedded Systems Engineer	HCL Technologies	Noida	550000.00

6 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.

```
set @applicant_id=3;  
select jobtitle,company_name,application_date  
from jobs j  
join companies c on j.company_id=c.company_id  
join applications a on j.job_id=a.job_id  
where applicant_id=@applicant_id;
```

```

+-----+-----+-----+
| jobtitle      | company_name | application_date |
+-----+-----+-----+
| Cloud Engineer | Microsoft    | 2025-03-25 14:36:53 |
+-----+-----+-----+
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.

```

Select avg(salary) as average_salary
from jobs where salary>0;

```

```

+-----+
| average_salary |
+-----+
| 582000.000000 |
+-----+
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.

```

select c.company_name, count(j.job_id) as job_count
from jobs j
join companies c on j.company_id = c.company_id
group by c.company_id
having count(j.job_id) = ( select max(job_count)
from (select count(job_id) as job_count from jobs group by company_id) as
job_counts);

```

company_name	job_count
Hexaware	1
Google	1
Microsoft	1
Amazon	1
TCS	1
Infosys	1
Wipro	1
IBM	1
Accenture	1
HCL Technologies	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.

For solving this query, I have added another column experiences to the applicants table.

I will take the location “Chennai” instead of “CityX”.

Select a.applicant_id,ap.first_name,c.company_name,ap.experience

From applications a join applicants ap on a.applicant_id=ap.applicant_id

Join jobs j on j.job_id=a.job_id

Join companies c on c.company_id=j.company_id

Where c.location='Chennai'

And ap.experience>=3;

applicant_id	first_name	company_name	experience
1	Arun	Hexaware	4

1 row in set (0.00 sec)

11. Retrieve a list of distinct job titles with salaries between \$6,00,000 and \$8,00,000.

```
select distinct jobtitle,salary
from jobs
where salary between 600000 and 800000;
```

jobtitle	salary
IT Consultant	750000.00
Frontend Developer	650000.00

2 rows in set (0.00 sec)

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

```
select j.job_id, j.jobtitle
from jobs j
left join applications a on j.job_id = a.job_id
where a.job_id is null;
```

```
mysql> select j.job_id, j.jobtitle
-> from jobs j
-> left join applications a on j.job_id = a.job_id
-> where a.job_id 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.

```
select ap.applicant_id, ap.first_name, ap.last_name, c.company_name, j.jobtitle
from applications a
join applicants ap on a.applicant_id = ap.applicant_id
join jobs j on a.job_id = j.job_id
join companies c on j.company_id = c.company_id;
```

applicant_id	first_name	last_name	company_name	jobtitle
1	Arun	Kumar	Hexaware	Software Engineer
2	Lakshmi	Narayan	Google	Data Scientist
3	Karthik	Rajan	Microsoft	Cloud Engineer
4	Revathi	Menon	Amazon	IT Consultant
5	Vignesh	Subramanian	TCS	Cybersecurity Analyst
6	Harini	Iyer	Infosys	Frontend Developer
7	Sandeep	Krishnan	Wipro	DevOps Engineer
8	Deepika	Ravi	IBM	AI/ML Engineer
9	Mohan	Ram	Accenture	Database Administrator
10	Anitha	Rajendran	HCL Technologies	Embedded Systems 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.

```
select c.company_name, count(j.job_id) as job_count
from companies c
left join jobs j on c.company_id = j.company_id
group by c.company_id, c.company_name;
```

company_name	job_count
Hexaware	1
Google	1
Microsoft	1
Amazon	1
TCS	1
Infosys	1
Wipro	1
IBM	1
Accenture	1
HCL Technologies	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.

```
select ap.applicant_id, ap.first_name, ap.last_name, c.company_name, j.jobtitle
from applicants ap
left join applications a on ap.applicant_id = a.applicant_id
left join jobs j on a.job_id = j.job_id
left join companies c on j.company_id = c.company_id;
```

applicant_id	first_name	last_name	company_name	jobtitle
1	Arun	Kumar	Hexaware	Software Engineer
2	Lakshmi	Narayan	Google	Data Scientist
3	Karthik	Rajan	Microsoft	Cloud Engineer
4	Revathi	Menon	Amazon	IT Consultant
5	Vignesh	Subramanian	TCS	Cybersecurity Analyst
6	Harini	Iyer	Infosys	Frontend Developer
7	Sandeep	Krishnan	Wipro	DevOps Engineer
8	Deepika	Ravi	IBM	AI/ML Engineer
9	Mohan	Ram	Accenture	Database Administrator
10	Anitha	Rajendran	HCL Technologies	Embedded Systems Engineer

10 rows in set (0.00 sec)

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

```
Select c.company_name,j.jobtitle,j.salary from companies c
Join jobs j on j.company_id=c.company_id
where j.salary >( select avg(salary) from jobs);
```

company_name	jobtitle	salary
Hexaware	Software Engineer	850000.00
Amazon	IT Consultant	750000.00
Infosys	Frontend Developer	650000.00
Wipro	DevOps Engineer	900000.00
Accenture	Database Administrator	1000000.00

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.

We don't have a city and state column in applicants table.

If we have those columns we can use the following query to retrieve the details.

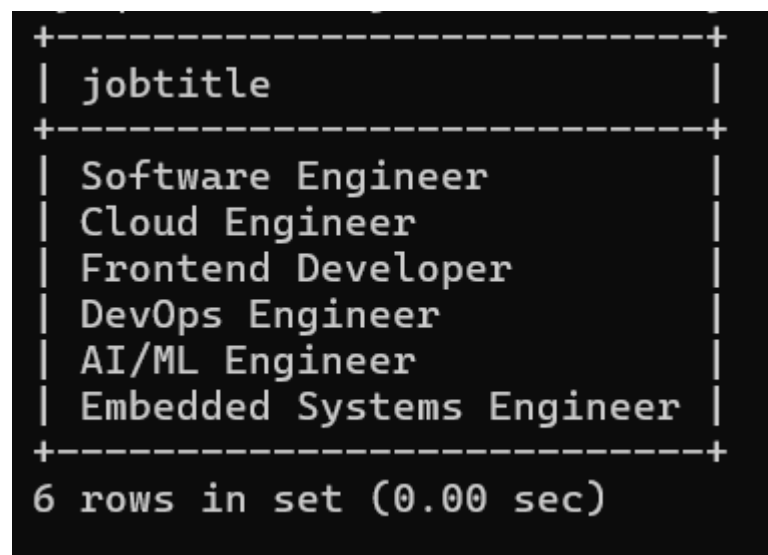
```
select applicant_id, first_name, last_name, concat (city, ', ', state) as location
from applicants;
```

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

```
select jobtitle
```

```
from jobs
```

```
where jobtitle like '%Developer%' or jobtitle like '%Engineer%';
```



A terminal window showing the output of a SQL query. The output is a table with one column, 'jobtitle', containing six rows of job titles. The table is enclosed in a box with dashed lines. Below the table, it says '6 rows in set (0.00 sec)'.

jobtitle
Software Engineer
Cloud Engineer
Frontend Developer
DevOps Engineer
AI/ML Engineer
Embedded Systems Engineer

6 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.applicant_id, a.first_name, a.last_name, j.job_id, j.jobtitle,
```

```
c.company_name
```

```
from applicants a
```

```
left join applications app on a.applicant_id = app.applicant_id
```

```

right join jobs j on app.job_id = j.job_id
join companies c on j.company_id = c.company_id
order by a.applicant_id, j.job_id;

```

applicant_id	first_name	last_name	job_id	jobtitle	company_name
1	Arun	Kumar	1	Software Engineer	Hexaware
2	Lakshmi	Narayan	2	Data Scientist	Google
3	Karthik	Rajan	3	Cloud Engineer	Microsoft
4	Revathi	Menon	4	IT Consultant	Amazon
5	Vignesh	Subramanian	5	Cybersecurity Analyst	TCS
6	Harini	Iyer	6	Frontend Developer	Infosys
7	Sandeep	Krishnan	7	DevOps Engineer	Wipro
8	Deepika	Ravi	8	AI/ML Engineer	IBM
9	Mohan	Ram	9	Database Administrator	Accenture
10	Anitha	Rajendran	10	Embedded Systems Engineer	HCL Technologies

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

```

select a.applicant_id, a.first_name, a.last_name, a.experience, c.company_id,
c.company_name, c.location
from applicants a
cross join companies c
where c.location = 'chennai'
and a.experience > 2 ;

```

applicant_id	first_name	last_name	experience	company_id	company_name	location
1	Arun	Kumar	4	1	Hexaware	Chennai
2	Lakshmi	Narayan	3	1	Hexaware	Chennai
3	Karthik	Rajan	7	1	Hexaware	Chennai
4	Revathi	Menon	5	1	Hexaware	Chennai
7	Sandeep	Krishnan	10	1	Hexaware	Chennai
8	Deepika	Ravi	6	1	Hexaware	Chennai
9	Mohan	Ram	8	1	Hexaware	Chennai

7 rows in set (0.00 sec)