**Coding Challenges: CareerHub, The Job Board**

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

mysql> create database careerhub;

query ok, 1 row affected (0.18 sec)

mysql> use careerhub;

database changed

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

mysql> create table companies (

companyid int primary key auto\_increment,

companyname varchar(255) not null,

location varchar(255) not null

);

mysql> create table jobs (

jobid int primary key auto\_increment,

companyid int not null,

jobtitle varchar(255) not null,

jobdescription text not null,

joblocation varchar(255) not null,

salary decimal(10,2) not null,

jobtype enum('full-time', 'part-time', 'contract') not null,

posteddate datetime default current\_timestamp,

foreign key (companyid) references companies(companyid) on delete cascade

);

mysql> create table applicants (

applicantid int primary key auto\_increment,

firstname varchar(100) not null,

lastname varchar(100) not null,

email varchar(255) unique not null,

phone varchar(20) not null,

resume text not null

);

mysql> create table applications (

applicationid int primary key auto\_increment,

jobid int not null,

applicantid int not null,

applicationdate datetime default current\_timestamp,

coverletter text not null,

foreign key (jobid) references jobs(jobid) on delete cascade,

foreign key (applicantid) references applicants(applicantid) on delete cascade

);

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

mysql> create table companies (

companyid int primary key auto\_increment,

companyname varchar(255) not null,

location varchar(255) not null

);

mysql> create table jobs (

jobid int primary key auto\_increment,

companyid int not null,

jobtitle varchar(255) not null,

jobdescription text not null,

joblocation varchar(255) not null,

salary decimal(10,2) not null,

jobtype enum('full-time', 'part-time', 'contract') not null,

posteddate datetime default current\_timestamp,

foreign key (companyid) references companies(companyid) on delete cascade

);

mysql> create table applicants (

applicantid int primary key auto\_increment,

firstname varchar(100) not null,

lastname varchar(100) not null,

email varchar(255) unique not null,

phone varchar(20) not null,

resume text not null

);

mysql> create table applications (

applicationid int primary key auto\_increment,

jobid int not null,

applicantid int not null,

applicationdate datetime default current\_timestamp,

coverletter text not null,

foreign key (jobid) references jobs(jobid) on delete cascade,

foreign key (applicantid) references applicants(applicantid) on delete cascade

);

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

**To handle database:**

drop database if exists careerhub;

create database careerhub;

use careerhub;

**To handle tables:**

create table if not exists companies(...)

create table if not exists jobs(..)

create table if not exists applicants(...)

create table if not exists applications(...)

**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.jobtitle,

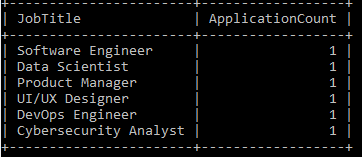
count(a.applicationid) as applicationcount

From Jobs j

left join applications a on j.jobid = a.jobid

group by j.jobid, j.jobtitle

order by applicationcount;

****

**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> set @minsalary = 60000;

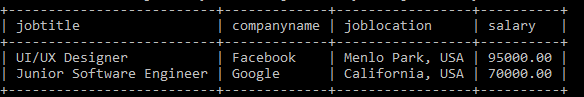
mysql> set @maxsalary = 100000;

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 @minsalary and @maxsalary;

****

**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.**

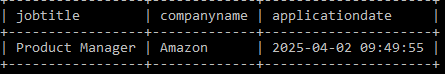
mysql > select j.jobtitle, c.companyname, app.applicationdate

from applications app

join jobs j on app.jobid = j.jobid

join companies c on j.companyid = c.companyid

where app.applicantid = 3;

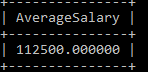


**8. Create an SQL query that calculates and displays the average salary offered by all companies for job listings in the "Jobs" table. Ensure that the query filters out jobs with a salary of zero.**

mysql> select avg(salary) as averagesalary

from jobs

where salary > 0;

****

**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 jobs j

join companies c on j.companyid = c.companyid

group by c.companyname

having count(j.jobid) = (

select max(jobcount)

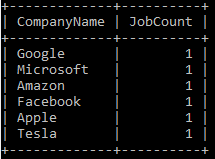
from ( select count(jobid) as jobcount

from jobs

group by companyid

) as jobcounts

);

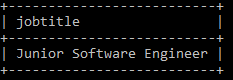


**11. Retrieve a list of distinct job titles with salaries between $60,000 and $80,000.**

mysql> select distinct jobtitle

from jobs

where salary between 60000 and 80000;

****

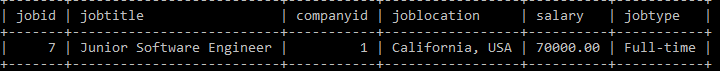
**12. Find the jobs that have not received any applications.**

mysql> select j.jobid, j.jobtitle, j.companyid, j.joblocation, j.salary, j.jobtype

from jobs j

left join applications a on j.jobid = a.jobid

where a.applicationid is null;

****

**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, a.email, c.companyname, j.jobtitle

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;



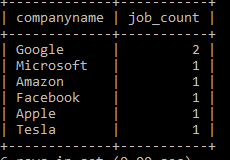
**14. Retrieve a list of companies along with the count of jobs they have posted, even if they have not received any applications.**

mysql > select c.companyname, count(j.jobid) as job\_count

from companies c

left join jobs j on c.companyid = j.companyid

group by c.companyid;



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

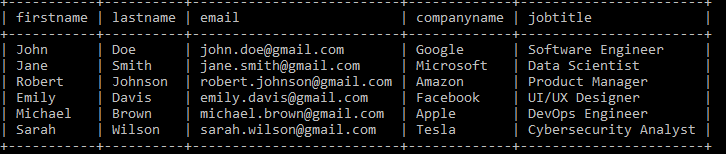
mysql > select a.firstname, a.lastname, a.email, c.companyname, j.jobtitle

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;



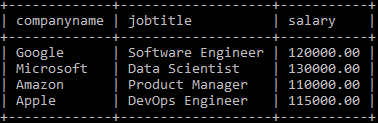
**16. Find companies that have posted jobs with a salary higher than the average salary of all jobs.**

mysql > select c.companyname, j.jobtitle, j.salary

from companies c

join jobs j on c.companyid = j.companyid

where j.salary > (select avg(salary) from jobs);

****

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

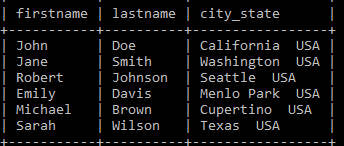
mysql > select a.firstname, a.lastname,

concat(substring\_index(j.joblocation, ',', 1), ' ', substring\_index(j.joblocation, ',', -1)) as city\_state

from applicants a

join applications app on a.applicantid = app.applicantid

join jobs j on app.jobid = j.jobid;

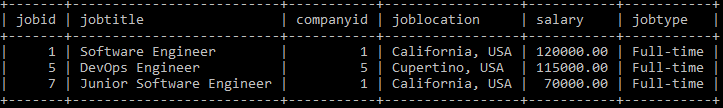
****

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

mysql> select jobid, jobtitle, companyid, joblocation, salary, jobtype

from jobs

where jobtitle like '%developer%' or jobtitle like '%engineer%';

****

**19. Retrieve a list of applicants and the jobs they have applied for, including those who have not applied and jobs without applicants.**

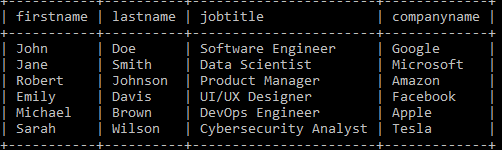
mysql> select a.firstname, a.lastname, j.jobtitle, c.companyname

from applicants a

left join applications app on a.applicantid = app.applicantid

left join jobs j on app.jobid = j.jobid

left join companies c on j.companyid = c.companyid;

****