#### Case Study: Career Hub

-- initializing database

create database if not exists careerHub;

use careerHub;

-- creating tables:

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

create table if not exists companies(

companyId int not null unique primary key,

companyName varchar(255),

location varchar(255)

);

create table if not exists jobs(

jobId int not null unique primary key,

companyId int,

jobTitle varchar(255),

jobDescription varchar(1255),

jobLocation varchar(255),

salary decimal,

jobType varchar(255),

postedDate datetime

);

create table if not exists applicants (

applicantId int not null unique primary key,

firstName varchar(255),

lastName varchar(255),

email varchar(255),

phone varchar(10),

city varchar(30),

state varchar(30), -- modified the table to add 3 more columns city state and experience to make sure some queries to work

experience int,

resume text

);

create table if not exists applications (

applicationId int not null unique primary key,

jobId int,

applicantId int,

applicationDate datetime,

coverLetter text

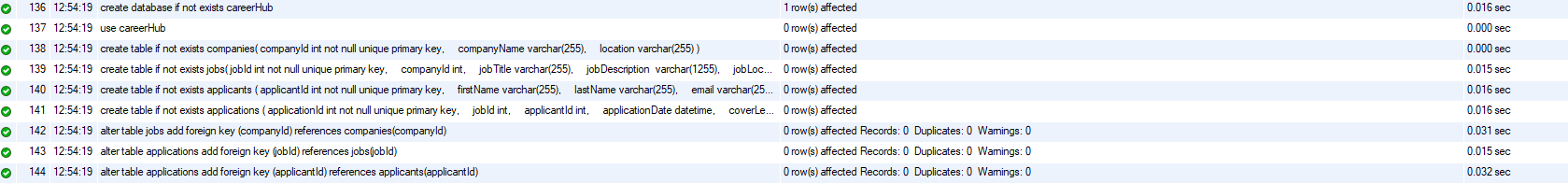
);

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

alter table jobs add foreign key (companyId) references companies(companyId);

alter table applications add foreign key (jobId) references jobs(jobId);

alter table applications add foreign key (applicantId) references applicants(applicantId);



#### Inserting data to the tables:

use careerHub;

select \* from companies;

-- Inserting data into the Companies table

INSERT INTO companies (companyId, companyName, location) VALUES

(1, 'ABC Corporation', 'Chennai'),

(2, 'XYZ Solutions', 'Coimbatore'),

(3, 'PQR Enterprises', 'Madurai'),

(4, 'LMN Technologies', 'Tiruchirappalli'),

(5, 'EFG Innovations', 'Salem'),

(6, 'RST Services', 'Tirunelveli'),

(7, 'UVW Solutions', 'Vellore'),

(8, 'IJK Industries', 'Erode'),

(9, 'OPQ Corporation', 'Thanjavur'),

(10, 'MNO Enterprises', 'Kanyakumari');

-- Inserting data into the Jobs table

INSERT INTO jobs (jobId, companyId, jobTitle, jobDescription, jobLocation, salary, jobType, postedDate) VALUES

(101, 1, 'Software Engineer', 'Developing software applications', 'Chennai', 100000, 'Full-time', '2024-04-16 09:00:00'),

(102, 2, 'Data Analyst', 'Analyzing data sets', 'Coimbatore', 80000, 'Full-time', '2024-04-16 10:00:00'),

(103, 3, 'Marketing Manager', 'Marketing strategy development', 'Madurai', 120000, 'Full-time', '2024-04-16 11:00:00'),

(104, 4, 'Web Developer', 'Building websites and web applications', 'Tiruchirappalli', 90000, 'Full-time', '2024-04-16 09:00:00'),

(105, 5, 'Data Scientist', 'Analyzing and interpreting complex data sets', 'Salem', 110000, 'Full-time', '2024-04-16 10:00:00'),

(106, 6, 'Business Analyst', 'Analyzing business processes and requirements', 'Tirunelveli', 85000, 'Full-time', '2024-04-16 11:00:00'),

(107, 7, 'UX/UI Designer', 'Designing user interfaces for digital products', 'Vellore', 95000, 'Full-time', '2024-04-16 09:00:00'),

(108, 8, 'Network Engineer', 'Designing and implementing computer networks', 'Erode', 105000, 'Full-time', '2024-04-16 10:00:00'),

(109, 9, 'Content Writer', 'Creating content for websites and marketing materials', 'Thanjavur', 75000, 'Full-time', '2024-04-16 11:00:00'),

(110, 10, 'Sales Executive', 'Promoting and selling products or services', 'Kanyakumari', 80000, 'Full-time', '2024-04-16 09:00:00'),

(111, 10, 'Python Developer', 'Entry level Python developer', 'Chennai', 80000, 'Full-time', '2024-04-16 06:00:00');

-- Inserting data into the Applicants table

INSERT INTO applicants (applicantId, firstName, lastName, email, phone, city, state, experience, resume) VALUES

(201, 'Aravind', 'Kumar', 'aravind.kumar@example.com', '9876543201', 'Chennai', 'Tamil Nadu', 3, 'Aravind\_Kumar\_Resume.pdf'),

(202, 'Priya', 'Raj', 'priya.raj@example.com', '9876543202', 'Chennai', 'Tamil Nadu', 2, 'Priya\_Raj\_Resume.pdf'),

(203, 'Manoj', 'Sharma', 'manoj.sharma@example.com', '9876543203', 'Chennai', 'Tamil Nadu',3, 'Manoj\_Sharma\_Resume.pdf'),

(204, 'Deepika', 'Nair', 'deepika.nair@example.com', '9876543204', 'Chennai', 'Tamil Nadu',3, 'Deepika\_Nair\_Resume.pdf'),

(205, 'Suresh', 'Venkat', 'suresh.venkat@example.com', '9876543205', 'Chennai', 'Tamil Nadu', 1, 'Suresh\_Venkat\_Resume.pdf'),

(206, 'Lavanya', 'Chandran', 'lavanya.chandran@example.com', '9876543206', 'Chennai', 'Tamil Nadu', 0, 'Lavanya\_Chandran\_Resume.pdf'),

(207, 'Rajesh', 'Sundar', 'rajesh.sundar@example.com', '9876543207', 'Chennai', 'Tamil Nadu',0, 'Rajesh\_Sundar\_Resume.pdf'),

(208, 'Anusha', 'Srinivasan', 'anusha.srinivasan@example.com', '9876543208', 'Chennai', 'Tamil Nadu', 2, 'Anusha\_Srinivasan\_Resume.pdf'),

(209, 'Ganesh', 'Krishnan', 'ganesh.krishnan@example.com', '9876543209', 'Chennai', 'Tamil Nadu', 2, 'Ganesh\_Krishnan\_Resume.pdf'),

(210, 'Divya', 'Ravi', 'divya.ravi@example.com', '9876543210', 'Chennai', 'Tamil Nadu', 3, 'Divya\_Ravi\_Resume.pdf');

-- Insert data into Applications table

INSERT INTO applications (applicationId, jobId, applicantId, applicationDate, coverLetter) VALUES

(301, 101, 201, '2024-04-16 12:00:00', 'Dear Hiring Manager, I am excited to apply for the Software Engineer position...'),

(302, 102, 202, '2024-04-16 13:00:00', 'Dear Hiring Manager, I am writing to express my interest in the Data Analyst role...'),

(303, 103, 203, '2024-04-16 14:00:00', 'Dear Hiring Manager, I am confident that my skills and experience make me a great fit for the Marketing Manager position...'),

(304, 104, 204, '2024-04-16 15:00:00', 'Dear Hiring Manager, I am interested in the Web Developer role and eager to contribute my expertise...'),

(305, 105, 205, '2024-04-16 16:00:00', 'Dear Hiring Manager, I am passionate about data analysis and excited to apply for the Data Scientist position...'),

(306, 106, 206, '2024-04-16 17:00:00', 'Dear Hiring Manager, I am applying for the Business Analyst role and confident in my ability to analyze and interpret data...'),

(307, 107, 207, '2024-04-16 18:00:00', 'Dear Hiring Manager, I am a creative UX/UI Designer with a strong portfolio and excited to join your team...'),

(308, 108, 208, '2024-04-16 19:00:00', 'Dear Hiring Manager, I am interested in the Network Engineer position and eager to apply my networking skills...'),

(309, 109, 209, '2024-04-16 20:00:00', 'Dear Hiring Manager, I am a skilled Content Writer with experience in creating engaging content...'),

(310, 110, 210, '2024-04-16 21:00:00', 'Dear Hiring Manager, I am enthusiastic about sales and eager to apply for the Sales Executive role...');



#### Tasks:

use careerHub;

/\*

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 jobs.jobTitle , count(applications.applicantId) as applicationCount

from jobs left join applications

on applications.jobId = jobs.jobId

group by jobs.jobTitle;



/\*

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,

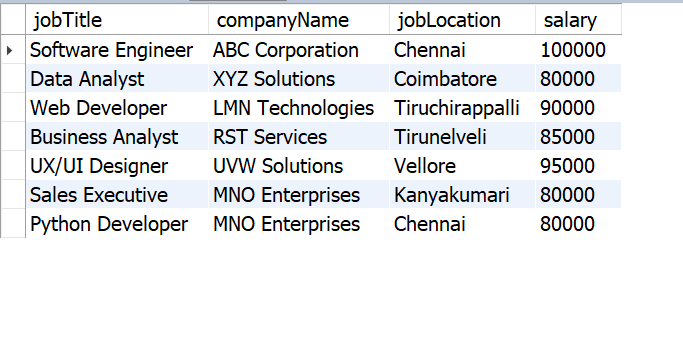
company name, location, and salary for each matching job.

\*/

select j.jobTitle, c.companyName, j.jobLocation, j.salary from jobs j

join companies c on j.companyId = c.companyId

where j.salary between 80000 and 100000;



/\*

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.

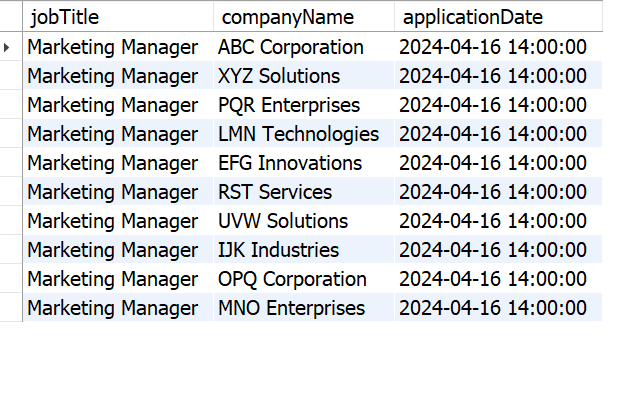
\*/

select j.jobTitle, c.companyName, a.applicationDate

from applications a join jobs j on a.jobId = j.jobId

join companies c on j.companyId = j.companyId

where a.applicantId = 203;



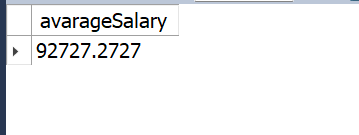
/\*

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 avarageSalary from jobs;



/\*

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.

\*/

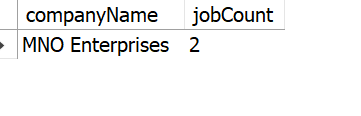
select companyName , count(\*) as jobCount from jobs j

join companies c on j.companyId = c.companyId

group by companyName

order by jobCount desc

limit 1;

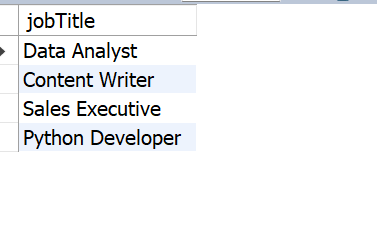


/\*

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;

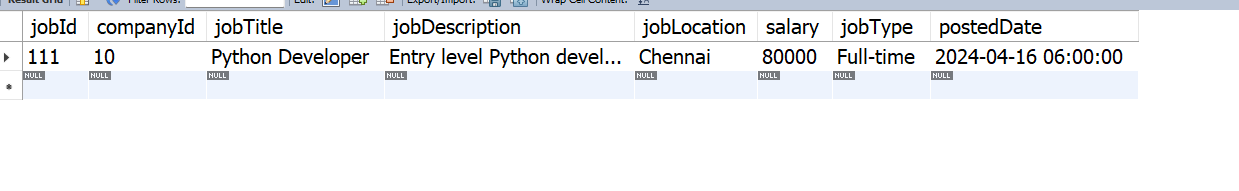


/\*

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

\*/

select \* from jobs where jobId not in (select distinct jobId from applications);



/\*

13. Retrieve a list of job applicants along with the companies they have applied to and the positions

they have applied for.

\*/

select applicants.firstName as firstName, applicants.lastName as lastName,

c.companyName as companyName, jobs.jobTitle as jobTitle

from applicants join applications on applicants.applicantId = applications.applicantId

join jobs on applications.jobId = jobs.jobId

join companies c on jobs.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.

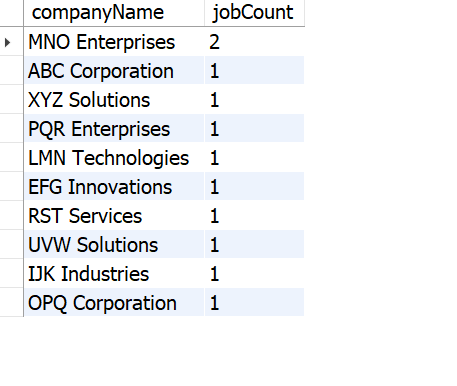
\*/

select c.companyName, count(j.jobId) as jobCount from companies c

join jobs j on c.companyId = j.companyId

group by companyName

order by jobCount desc;



/\*

15. List all applicants along with the companies and positions they have applied for, including those

who have not applied.

\*/

select a.firstName, a.lastName, companies.companyName, jobs.jobTitle from applicants a

left join applications ap on a.applicantId = ap.applicantId

left join jobs on ap.jobId = jobs.jobId

left join companies on jobs.companyId = companies.companyId;



/\*

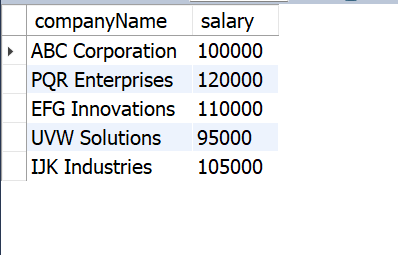
16. Find companies that have posted jobs with a salary higher than the average salary of all jobs.

\*/

select c.companyName, jobs.salary from companies c join jobs on c.companyId = jobs.companyId where jobs.salary > (

select avg(salary) from jobs

);

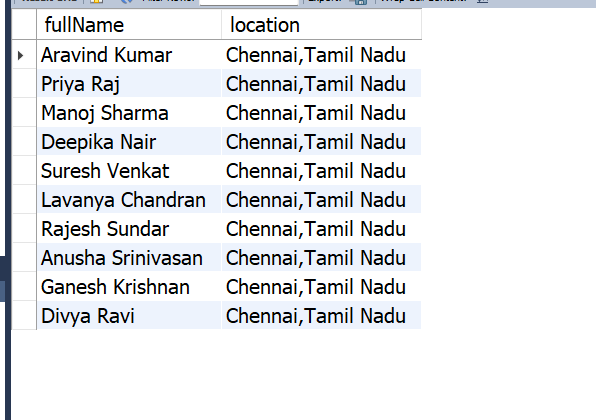


/\*

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

\*/

select concat(firstName, ' ', lastName) as fullName, concat(city, ',', state) as location from applicants;

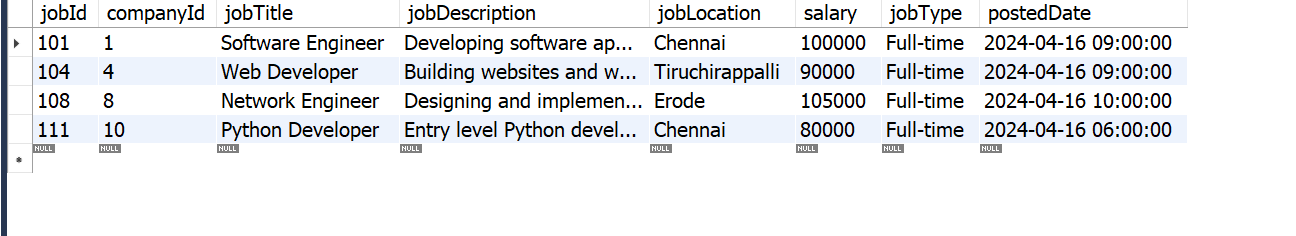


/\*

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

\*/

select \* 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

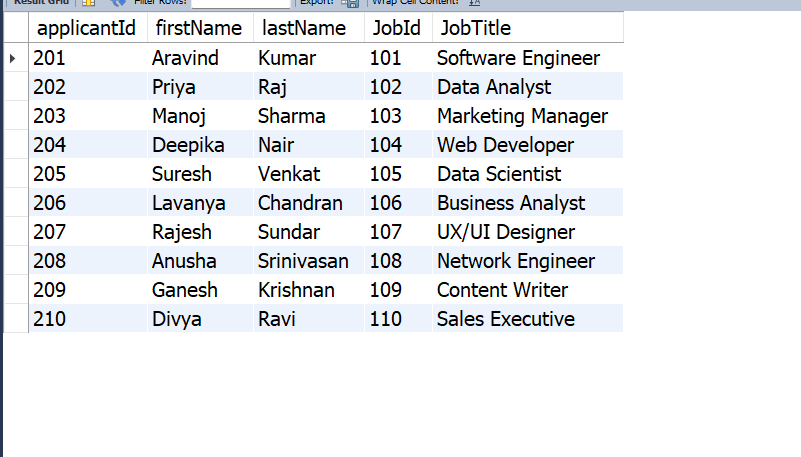
applied and jobs without applicants.

\*/

select a.applicantId, a.firstName, a.lastName, j.JobId, j.JobTitle from applicants a

left join applications ap on a.applicantId = ap.applicantId

left join jobs j on ap.jobId = j.jobId;



/\*

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.applicantId, a.firstName, a.lastName, c.companyId, c.companyName 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 = 'Madurai' and a.experience > 2;

