## **SQL Code Challenge**

## **Career Hub**

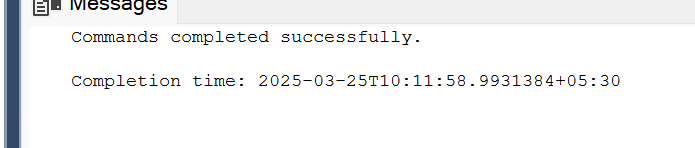
**Name: Gokul T M**

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

use master

create database careerhub;

use careerhub;

****

1. **Create tables for Companies, Jobs, Applicants and Applications.**
2. **Define appropriate primary keys, foreign keys, and constraints**
3. **Ensure the script handles potential errors, such as if the database or tables already exist.**

**Answer for (2,3,4) together:**

IF OBJECT\_ID('dbo.tblCompanies', 'U') IS NULL

begin

create table tblCompanies (

companyId int primary key,

companyName varchar(255),

companyLocation varchar(255)

)

end

IF OBJECT\_ID('dbo.tblJobs', 'U') IS NULL

begin

create table tblJobs (

jobId int primary key,

companyId int,

jobTitle varchar(255),

jobDescription text,

jobLocation varchar(255),

jobSalary decimal(10,2),

jobType varchar(50),

postedDate datetime,

foreign key (companyId) references tblCompanies(companyId)

);

end

IF OBJECT\_ID('dbo.tblApplicants', 'U') IS NULL

begin

create table tblApplicants (

applicantId int primary key,

firstName varchar(255),

lastName varchar(255),

email varchar(255) unique,

phone varchar(15) unique,

applicantResume text

);

end

IF OBJECT\_ID('dbo.tblApplications', 'U') IS NULL

begin

create table tblApplications (

applicationId int primary key,

jobId int,

applicantId int,

applicationDate datetime,

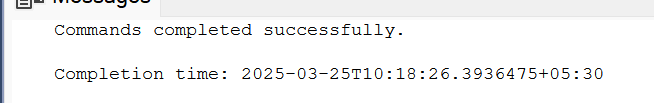
coverLetter text,

foreign key (jobId) references tblJobs(jobId),

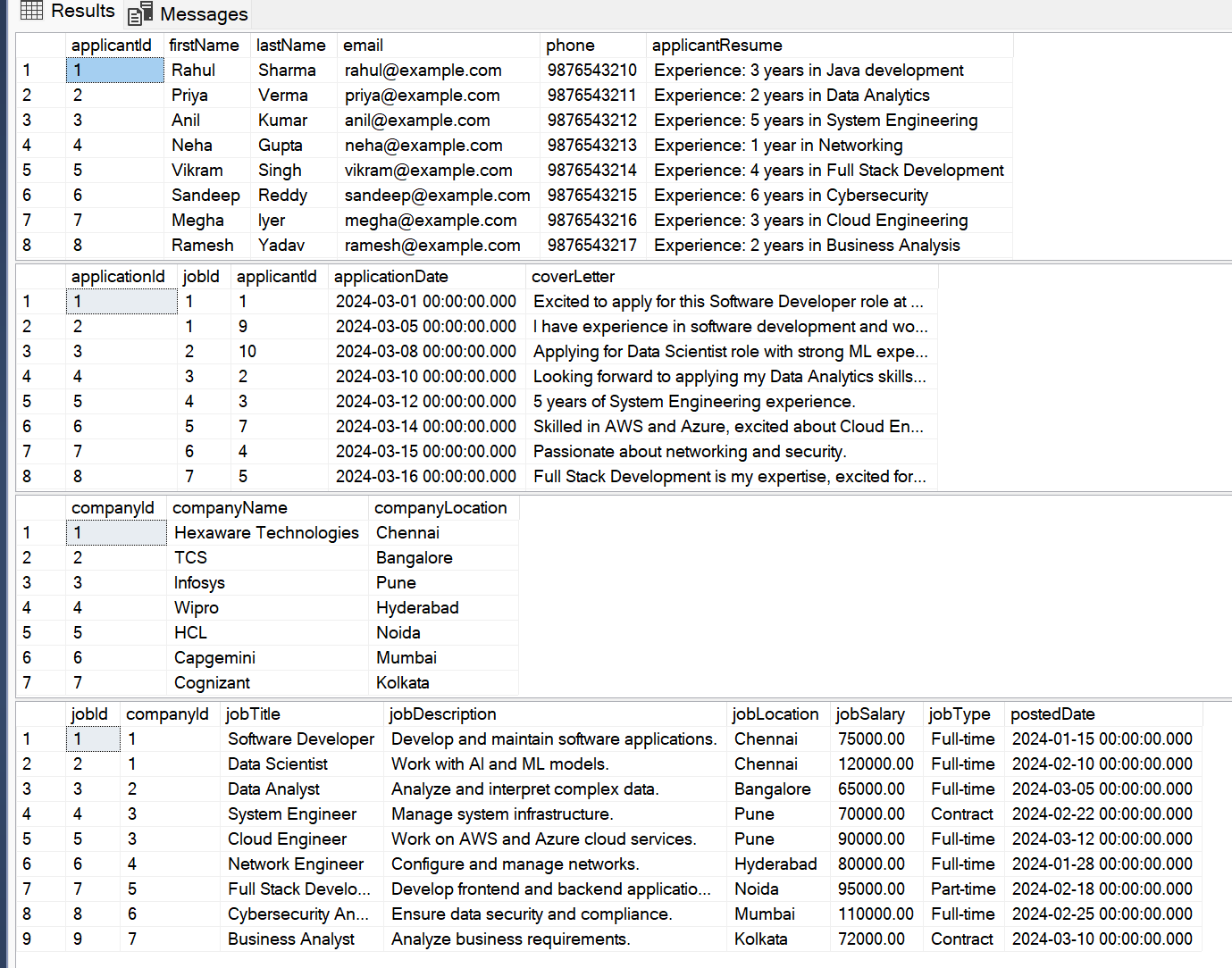
foreign key (applicantId) references tblApplicants(applicantId)

);

end

****

**Values inserted in created tables:**

****

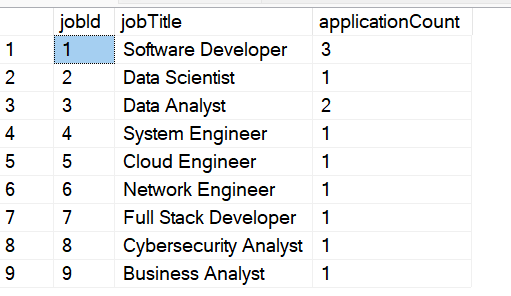
1. **Write an SQL query to count the number of applications received for each job listing in the "Jobs" table. Display the job title and the corresponding application count. Ensure that it lists all jobs, even if they have no applications.**

select j.jobId,j.jobTitle, count(a.applicationId) as applicationCount

from tblJobs j

left join tblApplications a on j.jobId = a.jobId

group by j.jobId, j.jobTitle;

****

1. **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.**

declare @minSalary decimal(10,2) = 60000;

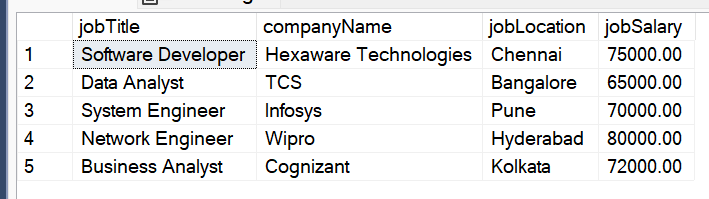
declare @maxSalary decimal(10,2) = 80000;

select j.jobTitle, c.companyName, j.jobLocation, j.jobSalary

from tblJobs j

join tblCompanies c on j.companyId = c.companyId

where j.jobSalary between @minSalary and @maxSalary;

****

1. **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.**

declare @applicantId int = 1;

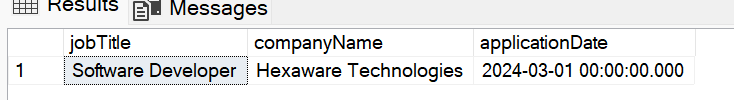
select j.jobTitle, c.companyName, a.applicationDate

from tblApplications a

join tblJobs j on a.jobId = j.jobId

join tblCompanies c on j.companyId = c.companyId

where a.applicantId = @applicantId;

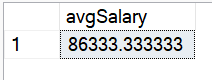
****

1. **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(j.jobSalary) as avgSalary

from tblJobs j

where j.jobSalary > 0;

****

1. **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.**

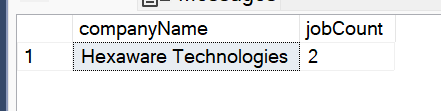
select top 1 c.companyName, count(j.jobId) as jobCount

from tblCompanies c

join tblJobs j on c.companyId = j.companyId

group by c.companyName

order by jobCount desc;

****

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

--assuming cityX = chennai

select distinct a.applicantId, a.firstName, a.lastName

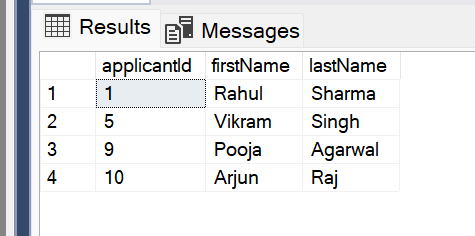
from tblApplicants a

join tblApplications app on a.applicantId = app.applicantId

join tblJobs j on app.jobId = j.jobId

join tblCompanies c on j.companyId = c.companyId

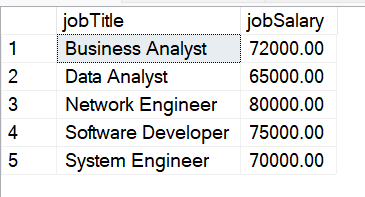
where c.companyLocation = 'Chennai' and a.applicantResume not like '%[0-2] years%'



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

select distinct jobTitle,jobSalary from tbljobs j

where jobSalary between 60000 and 80000

****

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

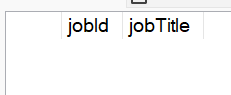
select j.jobId,j.jobTitle from tblJobs j

where not exists (

select 1 from tblApplications t

where j.jobId=t.jobId

)

****

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

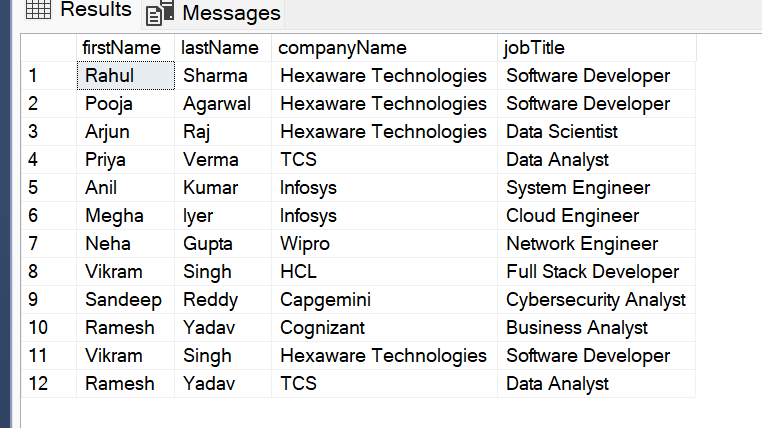
select a.firstName, a.lastName, c.companyName, j.jobTitle

from tblApplications app

join tblApplicants a on app.applicantId = a.applicantId

join tblJobs j on app.jobId = j.jobId

join tblCompanies c on j.companyId = c.companyId;

****

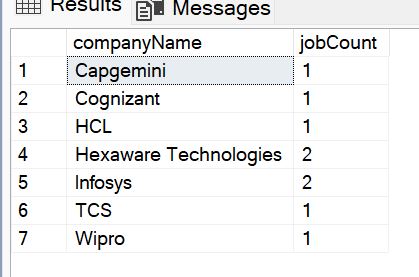
1. **Retrieve a list of companies along with the count of jobs they have posted, even if they have not received any applications.**

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

from tblCompanies c

left join tblJobs j on c.companyId = j.companyId

group by c.companyName;

****

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

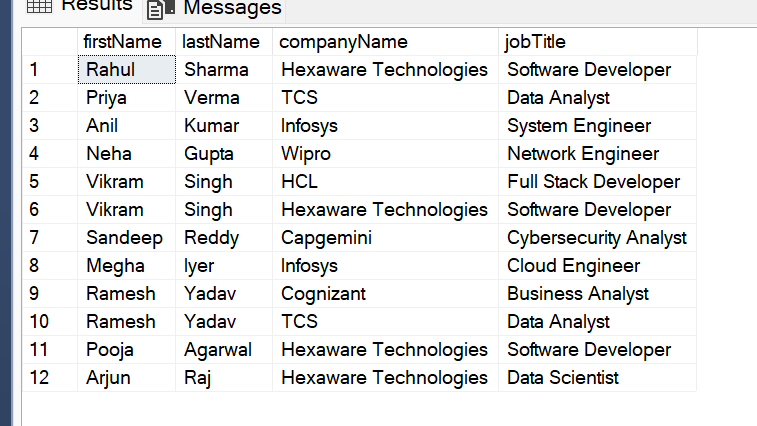
select a.firstName, a.lastName, c.companyName, j.jobTitle

from tblApplicants a

left join tblApplications app on a.applicantId = app.applicantId

left join tblJobs j on app.jobId = j.jobId

left join tblCompanies c on j.companyId = c.companyId;

****

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

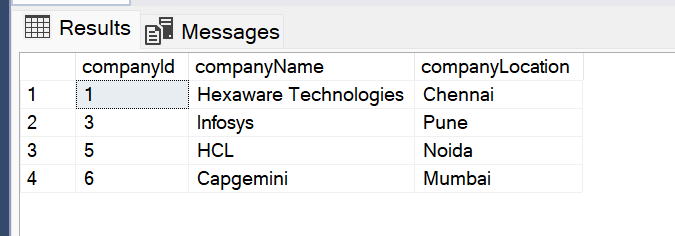
select \* from tblCompanies

where companyId in (

select companyId from tblJobs

where jobSalary > (select avg(jobSalary) from tblJobs)

)

****

1. **Display a list of applicants with their names and a concatenated string of their city and state.**
2. select distinct a.firstName, a.lastName, c.companyLocation + ', state not given' as location
3. from tblApplicants a
4. join tblApplications app on a.applicantId = app.applicantId
5. join tblJobs j on app.jobId = j.jobId
6. join tblCompanies c on j.companyId = c.companyId;

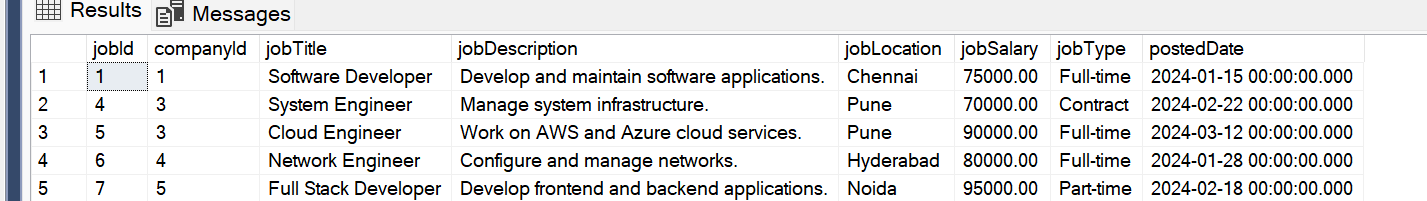
****

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

select \* from tblJobs

where jobTitle like '%Developer%'

or jobTitle like '%Engineer%';

****

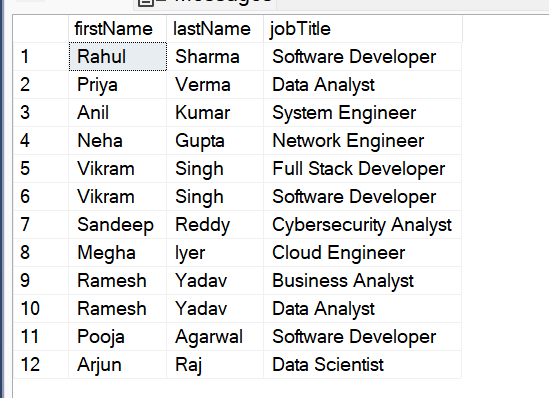
**19. Retrieve a list of applicants and the jobs they have applied for, including those who have not applied and jobs without applicants.**

select a.firstName, a.lastName, j.jobTitle

from tblApplicants a

left join tblApplications app on a.applicantId = app.applicantId

left join tblJobs j on app.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.firstName, a.lastName, c.companyName

from tblApplicants a

join tblApplications app on a.applicantId = app.applicantId

join tblJobs j on app.jobId = j.jobId

join tblCompanies c on j.companyId = c.companyId

where c.companyLocation = 'Chennai' and a.applicantResume not like '%[0-2] years%';

