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

**Tasks:**

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

create database CareerHub

use CareerHub

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

create table Companies (

company\_id int primary key,

company\_name varchar(255),

location varchar(255))

create table Jobs(

job\_id int primary key,

company\_id int,

job\_title varchar(255),

job\_description text,

job\_location varchar(255),

salary decimal(18,2),

job\_type varchar(255),

posted\_date datetime,

foreign key (company\_id) references companies(company\_id))

create table Applicants(

applicant\_id int primary key,

firstname varchar(100),

lastname varchar(100),

email varchar(255),

phone nvarchar(100),

resume text)

create table Applications(

application\_id int primary key,

job\_id int,

applicant\_id int,

application\_date datetime,

coverletter text,

foreign key(job\_id) references jobs(job\_id),

foreign key(application\_id) references applicants(applicant\_id))

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

create table Companies (

company\_id int primary key,

company\_name varchar(255),

location varchar(255))

create table Jobs(

job\_id int primary key,

company\_id int,

job\_title varchar(255),

job\_description text,

job\_location varchar(255),

salary decimal(18,2),

job\_type varchar(255),

posted\_date datetime,

foreign key (company\_id) references companies(company\_id))

create table Applicants(

applicant\_id int primary key,

firstname varchar(100),

lastname varchar(100),

email varchar(255),

phone nvarchar(100),

resume text)

create table Applications(

application\_id int primary key,

job\_id int,

applicant\_id int,

application\_date datetime,

coverletter text,

foreign key(job\_id) references jobs(job\_id),

foreign key(application\_id) references applicants(applicant\_id))

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

if DB\_ID ('CareerHub') is not null

begin

print 'Database exists'

end

else

print 'database doesnt exist'

end--4

if OBJECT\_ID('Companies') is not null

begin

print 'table exist'

end

else

begin

print 'table doesnt exist'

end

if OBJECT\_ID('Jobs') is not null

begin

print 'table exist'

end

else

begin

print 'table doesnt exist'

end

if OBJECT\_ID('Applicants') is not null

begin

print 'table exist'

end

else

begin

print 'table doesnt exist'

end

if OBJECT\_ID('Applications') is not null

begin

print 'table exist'

end

else

begin

print 'table doesnt exist'

end

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.job\_id,j.job\_title,count(a.application\_id)

from applications a right join jobs j

on j.job\_id=a.job\_id

group by j.job\_id,j.job\_title

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.job\_title,c.company\_name,c.location,j.salary

from jobs j join Companies c

on c.company\_id=j.company\_id where j.salary between 600000 and 800000

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.job\_title,c.company\_name,a.application\_date

from Companies c join jobs j on j.company\_id=c.company\_id

join Applications a on j.job\_id=a.job\_id

where applicant\_id>201--7

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 job\_id , avg(salary) as average\_salary from jobs where salary !=0 group by job\_id

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 count\_joblistings from Companies c join jobs j

on c.company\_id=j.company\_id

group by c.company\_name

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

select a.applicant\_id,a.firstname,a.experience ,c.company\_name,j.job\_title from Applicants a

join applications ap on a.applicant\_id=ap.applicant\_id

join jobs j on ap.job\_id=j.job\_id join Companies c on j.company\_id=c.company\_id

where c.location='Kolkata' and a. experience>=3; --10 this query has no experience in the table

11. Retrieve a list of distinct job titles with salaries between $60,000 and $80,000. select distinct job\_title from jobs where salary between 600000 and 800000

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

select j.job\_id,j.job\_title from jobs j left join Applications a

on a.job\_id=j.job\_id where a.application\_id is null

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

select a.applicant\_id,a.firstname as name, c.company\_id,c.company\_name,j.job\_id,j.job\_title

from applicants a join applications ap on a.applicant\_id=ap.applicant\_id

join jobs j on j.job\_id=ap.job\_id join Companies c on

c.company\_id=j.company\_id

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\_id,c.company\_name, count(j.job\_id) from Companies c

left join jobs j on c.company\_id=j.company\_id

group by c.company\_id,c.company\_name

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

select a.applicant\_id,a.firstname as name, c.company\_id,c.company\_name,j.job\_id,j.job\_title

from applicants a left join applications ap on a.applicant\_id=ap.applicant\_id

left join jobs j on j.job\_id=ap.job\_id left join Companies c on

c.company\_id=j.company\_id

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

select c.company\_id,c.company\_name from companies c join jobs j on c.company\_id=j.company\_id

where 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 applicant\_id,firstname+' '+lastname as name,city+','+state from applicants--17 this query has the city and state missing in the table

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

select job\_id,job\_title from jobs where job\_title like '%developer%' or job\_title 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.applicant\_id,a.firstname as name,j.job\_id,j.job\_title from Applicants a

full outer join Applications ap

on a.applicant\_id= ap.applicant\_id

full outer join jobs j on j.job\_id=ap.job\_id

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.firstname as name,c.company\_id,c.company\_name

from applicants a cross join Companies c

where c.location='chennai' and a.experience>2--20 this query has experience column missinig in the table