

# SQL Coding Challenge

## CareerHUB The Job Board

- J708 JITESH MANIKANDAN

**-- Tasks:**

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

```
create database CareerHub_The_Job_Board;  
use careerhub_the_job_board;
```

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

```
create table Companies (  
    Company_ID int primary key,  
    Company_Name varchar(255) not null,  
    Location varchar(255) not null  
);
```

```
create table Jobs (  
    Job_ID int primary key,  
    Company_ID int,  
    Job_Title varchar(255) not null,  
    Job_Description text not null,  
    Job_Location varchar(255) not null,  
    Salary decimal(10,2) not null,  
    Job_Type varchar(50) not null,
```

```
    Posteddate datetime not null,  
    foreign key (Company_ID) references Companies(Company_ID)  
);
```

```
create table Applicants (  
    Applicant_ID int primary key,  
    FirstName varchar(100) not null,  
    LastName varchar(100) not null,  
    Email varchar(255) not null unique,  
    Phone varchar(20) not null,  
    Resume text not null  
);
```

```
create table Applications (  
    Application_ID int primary key,  
    Job_ID int not null,  
    Applicant_ID int not null,  
    Application_Date datetime not null,  
    CoverLetter text not null,  
    foreign key (Job_ID) references Jobs(Job_ID),  
    foreign key (Applicant_ID) references Applicants(Applicant_ID)  
);
```

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

```
alter table jobs
```

```
add constraint fk_jobs_company foreign key (company_id)
references companies(company_id) on delete cascade;
```

```
alter table jobs
add constraint chk_jobs_salary check (salary >= 0);
```

```
alter table jobs
add constraint chk_jobs_type check (job_type in ('full-time', 'part-time', 'contract'));
```

```
alter table applicants
add constraint uq_applicants_email unique (email);
```

```
alter table applicants
add constraint uq_applicants_phone unique (phone);
```

```
alter table applications
add constraint fk_applications_job foreign key (job_id)
references jobs(job_id) on delete cascade;
```

```
alter table applications
add constraint fk_applications_applicant foreign key (applicant_id)
references applicants(applicant_id) on delete cascade on update cascade;
```

```
alter table applications
add constraint uq_applications unique (job_id, applicant_id);
```

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

-- // Yes It Handles All the Potential Errors

-- INSERTION OF RECORDS

-- Insert records into Companies

Insert Into companies (company\_id, company\_name, location) Values

(1, 'Tech Solutions', 'New York'),

(2, 'Innovatech', 'San Francisco'),

(3, 'Data Corp', 'Chicago'),

(4, 'Cybernetix', 'Los Angeles'),

(5, 'Finserve', 'Boston'),

(6, 'MediTech', 'Houston'),

(7, 'EduSoft', 'Seattle'),

(8, 'Green Energy', 'Denver'),

(9, 'AutoTech', 'Detroit'),

(10, 'RetailHub', 'Miami');

-- Insert records into Jobs

Insert Into jobs (job\_id, company\_id, job\_title, job\_description, job\_location, salary, job\_type, posteddate) Values

(101, 1, 'Software Engineer', 'Develop and maintain software applications.', 'New York', 85000.00, 'Full-Time', '2025-03-01 10:00:00'),

(102, 2, 'Data Analyst', 'Analyze and interpret complex data sets.', 'San Francisco', 75000.00, 'Full-Time', '2025-03-05 09:30:00'),  
(103, 3, 'Cyber Security Expert', 'Ensure network and data security.', 'Chicago', 95000.00, 'Contract', '2025-03-07 14:15:00'),  
(104, 4, 'Cloud Architect', 'Design and implement cloud solutions.', 'Los Angeles', 110000.00, 'Full-Time', '2025-03-10 16:45:00'),  
(105, 5, 'Financial Analyst', 'Analyze financial statements and trends.', 'Boston', 78000.00, 'Part-Time', '2025-03-12 11:00:00'),  
(106, 6, 'Medical Researcher', 'Conduct research on medical advancements.', 'Houston', 92000.00, 'Full-Time', '2025-03-14 08:50:00'),  
(107, 7, 'Education Consultant', 'Advise schools on curriculum development.', 'Seattle', 67000.00, 'Contract', '2025-03-15 09:15:00'),  
(108, 8, 'Environmental Scientist', 'Study environmental impacts and solutions.', 'Denver', 89000.00, 'Full-Time', '2025-03-16 10:30:00'),  
(109, 9, 'Automobile Engineer', 'Develop and test vehicle designs.', 'Detroit', 95000.00, 'Full-Time', '2025-03-17 13:40:00'),  
(110, 10, 'Retail Manager', 'Oversee retail store operations and sales.', 'Miami', 72000.00, 'Part-Time', '2025-03-18 15:00:00');

-- Insert records into Applicants

Insert Into applicants (applicant\_id, firstname, lastname, email, phone, resume) Values

(201, 'Alice', 'Johnson', 'alice.johnson@email.com', '1234567890', 'Experienced software engineer with expertise in Java and React.'),  
(202, 'Bob', 'Smith', 'bob.smith@email.com', '0987654321', 'Data analyst proficient in SQL and Python, strong analytical skills.'),  
(203, 'Charlie', 'Brown', 'charlie.brown@email.com', '1122334455', 'Cyber security expert with experience in penetration testing and risk analysis.'),  
(204, 'Diana', 'Lee', 'diana.lee@email.com', '5566778899', 'Cloud architect specializing in AWS, Azure, and GCP solutions.'),  
(205, 'Edward', 'Martin', 'edward.martin@email.com', '6677889900', 'Financial analyst with a strong background in investment banking.'),

(206, 'Fiona', 'White', 'fiona.white@email.com', '7788991122', 'Medical researcher with experience in genetic studies and clinical trials.'),

(207, 'George', 'Harris', 'george.harris@email.com', '8899001122', 'Education consultant with expertise in e-learning and curriculum development.'),

(208, 'Helen', 'Young', 'helen.young@email.com', '9900112233', 'Environmental scientist focused on sustainability and climate change research.'),

(209, 'Isaac', 'Wright', 'isaac.wright@email.com', '1100223344', 'Automobile engineer specializing in electric vehicle technology.'),

(210, 'Julia', 'Adams', 'julia.adams@email.com', '2200334455', 'Retail manager with extensive experience in sales and team leadership.');

-- Insert records into Applications

Insert Into applications (application\_id, job\_id, applicant\_id, application\_date, coverletter)  
Values

(301, 101, 201, '2025-03-15 08:30:00', 'I am excited to apply for the software engineer role at Tech Solutions.'),

(302, 102, 202, '2025-03-16 09:00:00', 'My experience in data analysis makes me a strong candidate for this role.'),

(303, 103, 203, '2025-03-17 10:45:00', 'I have a passion for cyber security and a proven track record in risk assessment.'),

(304, 104, 204, '2025-03-18 11:20:00', 'I am interested in designing and implementing scalable cloud solutions.'),

(305, 105, 205, '2025-03-19 13:15:00', 'I bring extensive experience in financial modeling and investment analysis.'),

(306, 106, 206, '2025-03-20 14:00:00', 'I have a strong background in medical research and innovative healthcare solutions.'),

(307, 107, 207, '2025-03-21 15:45:00', 'I am eager to contribute my expertise in education consultancy to your organization.'),

(308, 108, 208, '2025-03-22 16:30:00', 'I am passionate about environmental sustainability and would love to join your team.'),

(309, 109, 209, '2025-03-23 17:00:00', 'I am an experienced automobile engineer with a focus on electric vehicle innovation.');

(310, 110, 210, '2025-03-24 18:20:00', 'I have a proven track record in retail management and increasing sales performance.');

**-- 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_Title , count(a.job_id) as Application_Count from jobs j
left join applications a on j.job_id = a.job_id
group by j.job_title order by count(a.job_id) desc;
```

**-- 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,j.job_location,j.salary from jobs j join companies c on
j.company_id = c.company_id
where salary between 40000 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.job_title, c.company_name, a.application_date  from applications a
join jobs j on a.job_id = j.job_id
join companies c on j.company_id = c.company_id
```

```
where a.applicant_id = 202  
order by a.application_date desc;
```

**-- 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;
```

**-- 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_name  
having count(j.job_id) = ( select max(count(j.job_id)) from (  
select count(job_id) from jobs  
group by company_id ) as job_counts  
);
```

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

```
alter table applicants  
add experience int;
```



```
update applicants set experience = case applicant_id
    when 201 then 5
    when 202 then 2
    when 203 then 4
    when 204 then 1
    when 205 then 3
    when 206 then 6
    when 207 then 8
    when 208 then 0
    when 209 then 7
    when 210 then 3
end
where applicant_id between 301 and 310;
```

```
update applicants
set experience = 3
where applicant_id = 201;
```

```
select distinct a.applicant_id, a.firstname, a.lastname, a.email, a.phone from applications b
join jobs j on b.job_id = j.job_id
join companies c on j.company_id = c.company_id
join applicants a on b.applicant_id = a.applicant_id
where c.location = 'New york'
and a.experience >= 3;
```

```
select * from applicants;
```

**-- 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 60000 and 80000;
```

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

```
select job_title from jobs
where job_id not in (select job_id 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 concat(a.firstname," ",a.lastname)as Name , c.company_name, j.job_title as Position
from applications a
join applicants b on b.applicant_id = a.applicant_id
join jobs j on b.job_id = j.job_id
join companies c on j.company_id = c.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_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_name;
```

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

```
select concat(a.firstname," ",a.lastname)as Name, c.company_name, j.job_title from
applicants a
```

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

```
left join jobs j on b.job_id = j.job_id
```

```
left join companies c on j.company_id = c.company_id;
```

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

```
select distinct c.company_name from companies c
```

```
join jobs j on c.company_id = j.company_id
```

```
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.**

```
select firstname, lastname, concat(city, ', ', state) as location
```

```
from applicants;
```

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

```
select 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.firstname, a.lastname, j.job_title
```

```
from applicants a
left join applications b on a.applicant_id = b.applicant_id
left join jobs j on b.job_id = j.job_id
```

union

```
select a.firstname, a.lastname, j.job_title
from jobs j
left join applications b on j.job_id = b.job_id
left join applicants a on b.applicant_id = a.applicant_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.firstname, a.lastname, c.company_name, c.location from applicants a
join companies c where c.location = 'New York' and a.experience > 2;
```