# CareerHUB


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

**Instructions**

- Coding Challenges submissions should be done through the partcipants' Github repository, and the link should be shared with trainers and Hexavarsity.

**Problem Statement:**

**Create SQL Schema from the application, use the class attributes for table column names.**

**SQL Schema:**

**Table: Companies**
**Attributes:**

- CompanyID (Primary Key, int): Unique identifier for each company.
- CompanyName (string): The name of the hiring company.
- Location (string): The location of the company.

**Table: Jobs**
**Attributes:**

- JobID (Primary Key, int): Unique identifier for each job listing.
- CompanyID (Foreign Key, int): References the CompanyID of the hiring company.
- JobTitle (string): The title of the job.
- JobDescription (text): A detailed description of the job.
- JobLocation (string): The location where the job is based.
- Salary (decimal): The salary offered for the job.
- JobType (string): Type of job (e.g., Full-time, Part-time, Contract).
- PostedDate (datetime): Date and time when the job was posted.

**Table: Applicants**
**Attributes:**
ApplicantID (Primary Key, int): Unique identifier for each applicant.

- FirstName (string): The first name of the applicant.
- LastName (string): The last name of the applicant.
- Email (string): The email address of the applicant.
- Phone (string): The phone number of the applicant.
- Resume (text): The applicant's resume or CV (text or file reference).

**Table: Applications**
**Attributes:**

- ApplicationID (Primary Key, int): Unique identifier for each job application.
- JobID (Foreign Key, int): References the JobID of the job listing.
- ApplicantID (Foreign Key, int): References the ApplicantID of the applicant.
- ApplicationDate (datetime): Date and time when the application was submitted.
- CoverLetter (text): The applicant's cover letter for the specific job.

**Tasks:**

1. Provide a SQL script that initializes the database for the Job Board scenario "CareerHub".
2. Create tables for Companies, Jobs, Applicants and Applications.
3. Define appropriate primary keys, foreign keys, and constraints.
4. Ensure the script handles potential errors, such as if the database or tables already exist.
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.
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.
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.
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.
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.
10. Find the applicants who have applied for positions in companies located in 'CityX' and have at least 3 years of experience.
11. Retrieve a list of distinct job titles with salaries between $60,000 and $80,000.
12. Find the jobs that have not received any applications.
13. Retrieve a list of job applicants along with the companies they have applied to and the positions they have applied for.
14. Retrieve a list of companies along with the count of jobs they have posted, even if they have not received any applications.
15. List all applicants along with the companies and positions they have applied for, including those who have not applied.
16. Find companies that have posted jobs with a salary higher than the average salary of all jobs.
17. Display a list of applicants with their names and a concatenated string of their city and state.
18. Retrieve a list of jobs with titles containing either 'Developer' or 'Engineer'.
19. Retrieve a list of applicants and the jobs they have applied for, including those who have not applied and jobs without applicants.
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

Coding challenge Answers

1. create database career_hub;
2.

| Table Name | Creation |
|---|---|
| companies | create table companies(companyID int primary key, companyName varchar(50), companyLocation varchar(50)); |
| jobs | create table jobs(jobID int primary key, companyID int, jobTitle varchar(50), jobDescription varchar(255), jobLocation varchar(30), salary decimal(10,2), jobType varchar(70), postedDate date, foreign key(companyID) references companies(companyID)); |
| applicants | create table applicants(applicantID int primary key, firstName varchar(70),lastName varchar(70), email_id varchar(80),phone varchar(20), resume text); |
| applications | create table applications(applicationID int primary key, jobID int, applicantID int, applicationDate datetime, coverLetter text, foreign key(jobID) references jobs(jobID), foreign key(applicantID) references applicants(applicantID) ); |

3.

| Constraints | columns |
|---|---|
| Primary key | companyID, jobID, applicantID, applicationID |
| Foreign key | companyID, jobID, applicantID |
| constraints | Unique, not null |

4. create database career_hub;
   create table companies(companyID int);
   Error Code: 1007. Can't create database 'career_hub'; database exists
   Error Code: 1050. Table 'companies' already exists
5. select j.jobTitle,count(a.applicationID) as applications_count from jobs j left join applications a on j.jobID=a.jobID group by j.jobID,j.jobTitle;

| jobTitle | applications_count |
|---|---|
| Software Engineer | 1 |
| Data Analyst | 1 |
| Project Manager | 1 |
| Software Developer | 1 |
| Sales Associate | 1 |
| Network Administrator | 1 |
| UI/UX Designer | 1 |
| Quality Assurance Analyst | 1 |
| Business Development Manager | 0 |
| Data Scientist | 0 |

10 rows in set (0.01 sec)

6. select j.jobTitle,c.companyName,j.jobLocation,j.salary from jobs j join companies c on j.companyID=c.companyID where j.salary between 50000 and 90000;

| jobTitle | companyName | jobLocation | salary |
|---|---|---|---|
| Software Engineer | Hexaware | Chennai | 80000.00 |
| Data Analyst | Infosys | Hyderabad | 75000.00 |
| Software Developer | TCS | Delhi | 85000.00 |
| Sales Associate | Walmart | Andhra Pradesh | 60000.00 |
| Network Administrator | Tech Mahindra | Mumbai | 90000.00 |
| UI/UX Designer | Zoho | Chennai | 85000.00 |
| Quality Assurance Analyst | Hexaware | Chennai | 70000.00 |

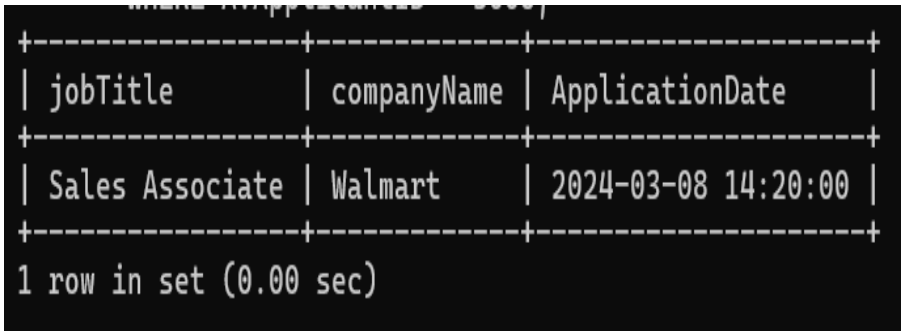
7 rows in set (0.00 sec)

select j.jobTitle,c.companyName,j.jobLocation,j.salary from jobs j join companies c on j.companyID=c.companyID where j.salary between (select min(salary) from jobs) and (select max(salary) from jobs) ;

| jobTitle | companyName | jobLocation | salary |
|---|---|---|---|
| Software Engineer | Hexaware | Chennai | 80000.00 |
| Data Analyst | Infosys | Hyderabad | 75000.00 |
| Project Manager | Accenture | Bangalore | 100000.00 |
| Software Developer | TCS | Delhi | 85000.00 |
| Sales Associate | Walmart | Andhra Pradesh | 60000.00 |
| Network Administrator | Tech Mahindra | Mumbai | 90000.00 |
| UI/UX Designer | Zoho | Chennai | 85000.00 |
| Quality Assurance Analyst | Hexaware | Chennai | 70000.00 |
| Business Development Manager | Infosys | Hyderabad | 95000.00 |
| Data Scientist | Accenture | Bangalore | 110000.00 |

10 rows in set (0.01 sec)

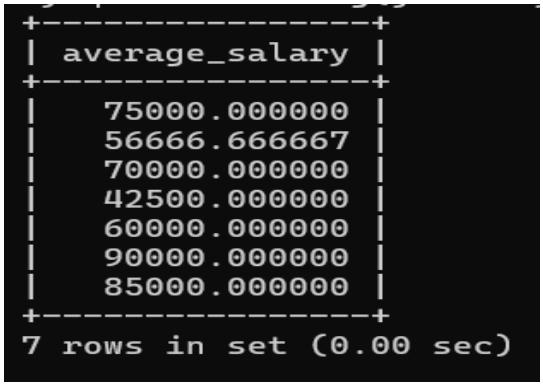
7. select j.jobTitle, c.companyName, a.ApplicationDate from applications a join jobs j on a.jobID = j.jobID join companies c on j.companyID = c.companyID where A.ApplicantID = 3005;



| jobTitle | companyName | ApplicationDate |
|---|---|---|
| Sales Associate | Walmart | 2024-03-08 14:20:00 |

1 row in set (0.00 sec)

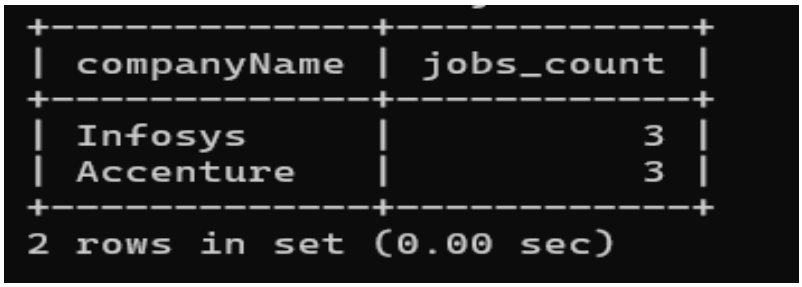
8. select avg(j.salary) as average_salary from jobs j join companies c on j.companyID=c.companyID group by c.companyID;



| average_salary |
|---|
| 75000.000000 |
| 56666.666667 |
| 70000.000000 |
| 42500.000000 |
| 60000.000000 |
| 90000.000000 |
| 85000.000000 |

7 rows in set (0.00 sec)

9. select c.companyName, count(*) AS jobs_count from jobs j join companies c on j.companyID = c.companyID group by j.companyID having count(*) = (select count(*) from jobs group by companyID order by count(*) desc limit 1);



| companyName | jobs_count |
|---|---|
| Infosys | 3 |
| Accenture | 3 |

2 rows in set (0.00 sec)

10. select distinct a.* from applications ap join jobs j on ap.jobID = j.jobID join companies c on j.companyID = c.companyID join applicants a on ap.applicantID = a.applicantID where c.companylocation = 'banglore' and A.experience >= 3;

| applicantID | firstName | lastName | email_id | phone | resume | experience |
|---|---|---|---|---|---|---|
| 3003 | Michael | Johnson | michael.johnson@example.com | 456-789-0123 | Michael Johnson - Resume.txt | 3 |

1 row in set (0.00 sec)

11. select distinct jobTitle from jobs where salary between 60000 and 80000;

| jobTitle |
|---|
| Software Engineer |
| Data Analyst |
| Sales Associate |
| Quality Assurance Analyst |

4 rows in set (0.00 sec)

12. select jobID,jobTitle,jobLocation,jobType,postedDate from jobs where jobID not in (select jobID from applications);

| jobID | jobTitle | jobLocation | jobType | postedDate |
|---|---|---|---|---|
| 2009 | Business Development Manager | Hyderabad | Full-time | 2024-03-04 |
| 2010 | Data Scientist | Bangalore | Part-time | 2024-03-03 |
| 2011 | Intern | Hyderabad | Part-time | 2024-03-13 |
| 2012 | Volunteer | Bangalore | Volunteer | 2024-03-13 |
| 2013 | Trainee | Delhi | Full-time | 2024-03-13 |

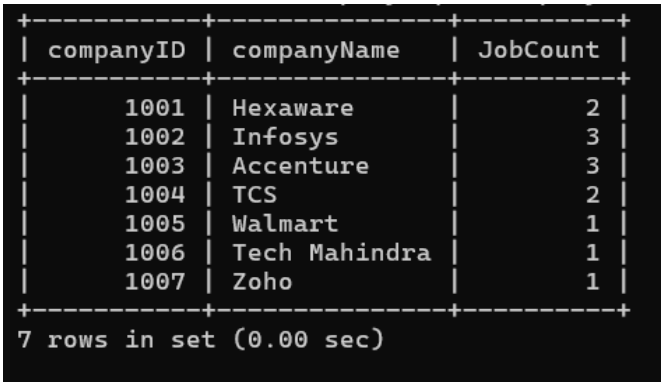
5 rows in set (0.00 sec)

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

| firstName | lastName | companyName | jobTitle |
|---|---|---|---|
| John | Doe | Hexaware | Software Engineer |
| Jane | Smith | Infosys | Data Analyst |
| Michael | Johnson | Accenture | Project Manager |
| Emily | Davis | TCS | Software Developer |
| David | Brown | Walmart | Sales Associate |
| Sarah | Wilson | Tech Mahindra | Network Administrator |
| Matthew | Martinez | Zoho | UI/UX Designer |
| Olivia | Taylor | Hexaware | Quality Assurance Analyst |

8 rows in set (0.00 sec)

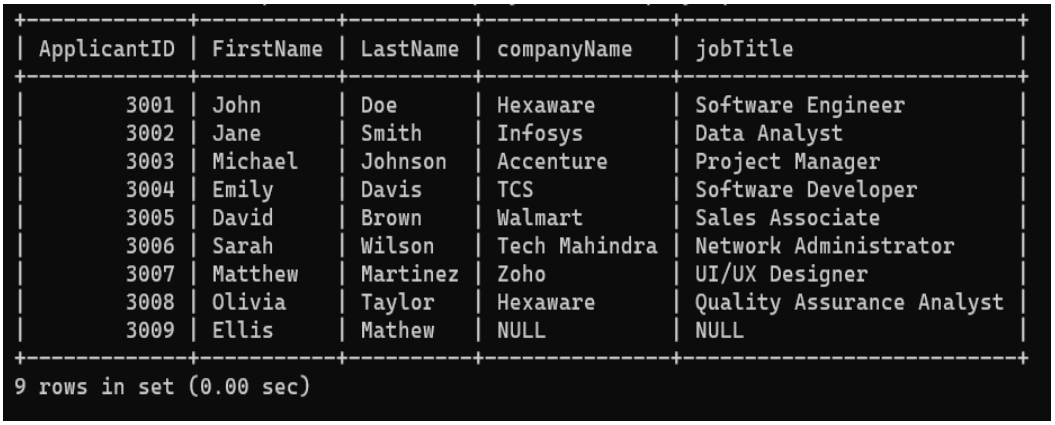
14. select c.companyID, c.companyName, count(j.jobID) as jobCount from companies c left join jobs j on c.companyID = j.companyID group by c.companyID, c.companyName;



| companyID | companyName | JobCount |
|---|---|---|
| 1001 | Hexaware | 2 |
| 1002 | Infosys | 3 |
| 1003 | Accenture | 3 |
| 1004 | TCS | 2 |
| 1005 | Walmart | 1 |
| 1006 | Tech Mahindra | 1 |
| 1007 | Zoho | 1 |

7 rows in set (0.00 sec)

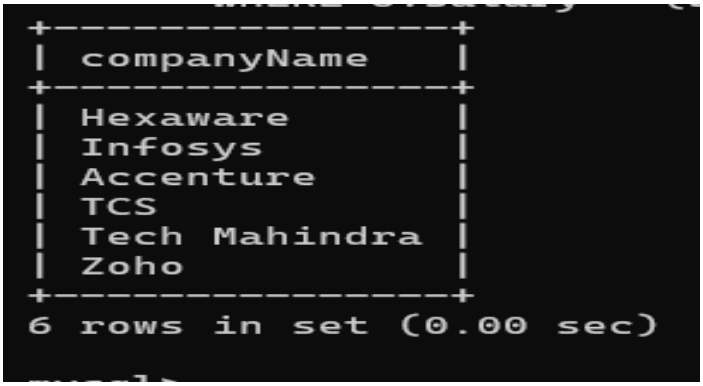
15. select a.applicantID, a.firstName, a.lastName, c.companyName, j.jobTitle from applicants a left join applications ap on a.applicantID = ap.applicantID left join jobs j on ap.jobID = j.jobID left join companies c on j.companyID = c.companyID;



| ApplicantID | FirstName | LastName | companyName | jobTitle |
|---|---|---|---|---|
| 3001 | John | Doe | Hexaware | Software Engineer |
| 3002 | Jane | Smith | Infosys | Data Analyst |
| 3003 | Michael | Johnson | Accenture | Project Manager |
| 3004 | Emily | Davis | TCS | Software Developer |
| 3005 | David | Brown | Walmart | Sales Associate |
| 3006 | Sarah | Wilson | Tech Mahindra | Network Administrator |
| 3007 | Matthew | Martinez | Zoho | UI/UX Designer |
| 3008 | Olivia | Taylor | Hexaware | Quality Assurance Analyst |
| 3009 | Ellis | Mathew | NULL | NULL |

9 rows in set (0.00 sec)

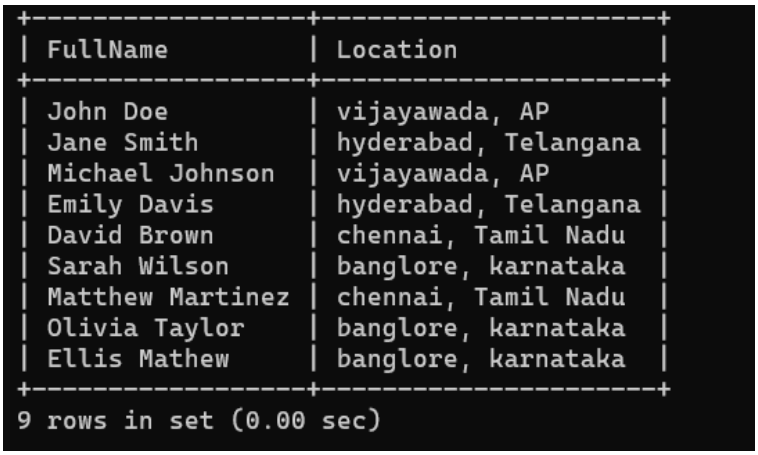
16. Select distinct c.companyName from companies c join jobs j on c.companyID = j.companyID where j.salary > (select avg(salary) from jobs);



| companyName |
|---|
| Hexaware |
| Infosys |
| Accenture |
| TCS |
| Tech Mahindra |
| Zoho |

6 rows in set (0.00 sec)

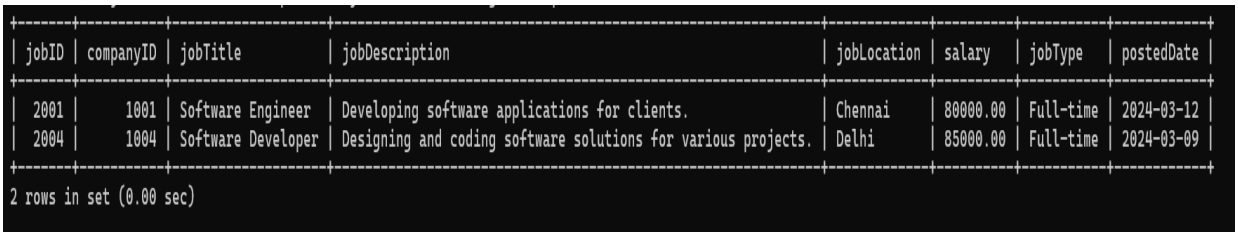
17. Select concat(firstName, ' ', lastName) as FullName, concat(city, ', ', state) as Location from applicants;



| FullName | Location |
|---|---|
| John Doe | vijayawada, AP |
| Jane Smith | hyderabad, Telangana |
| Michael Johnson | vijayawada, AP |
| Emily Davis | hyderabad, Telangana |
| David Brown | chennai, Tamil Nadu |
| Sarah Wilson | banglore, karnataka |
| Matthew Martinez | chennai, Tamil Nadu |
| Olivia Taylor | banglore, karnataka |
| Ellis Mathew | banglore, karnataka |

9 rows in set (0.00 sec)

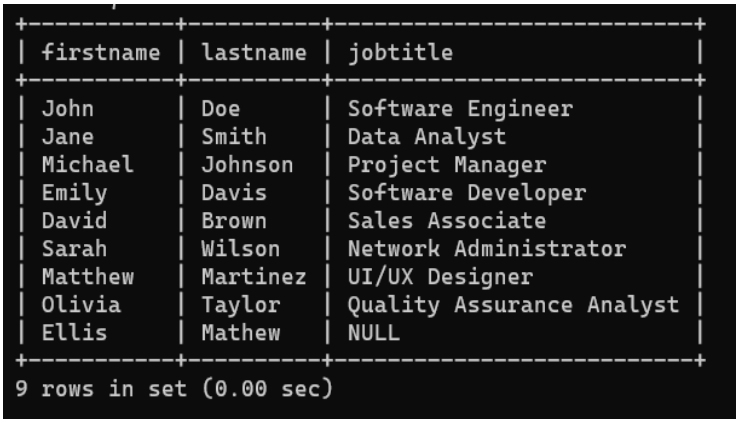
18. select * from jobs where jobTitle like '%Developer%' or jobTitle like '%Engineer%';



| jobID | companyID | jobTitle | jobDescription | jobLocation | salary | jobType | postedDate |
|---|---|---|---|---|---|---|---|
| 2001 | 1001 | Software Engineer | Developing software applications for clients. | Chennai | 80000.00 | Full-time | 2024-03-12 |
| 2004 | 1004 | Software Developer | Designing and coding software solutions for various projects. | Delhi | 85000.00 | Full-time | 2024-03-09 |

2 rows in set (0.00 sec)

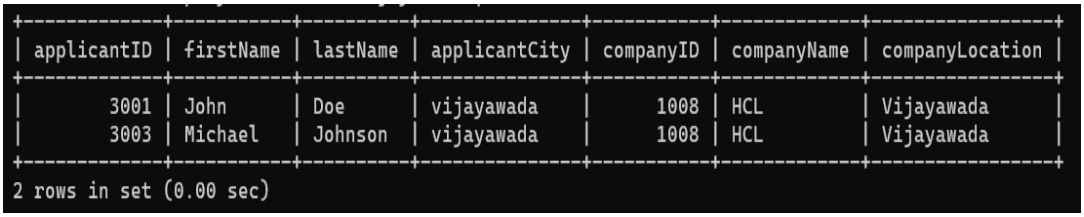
19. Select a.firstname,a.lastname,j.jobtitle from applicants a left join applications ap on a.applicantid=ap.applicantid left join jobs j on ap.jobid=j.jobid;



| firstname | lastname | jobtitle |
|---|---|---|
| John | Doe | Software Engineer |
| Jane | Smith | Data Analyst |
| Michael | Johnson | Project Manager |
| Emily | Davis | Software Developer |
| David | Brown | Sales Associate |
| Sarah | Wilson | Network Administrator |
| Matthew | Martinez | UI/UX Designer |
| Olivia | Taylor | Quality Assurance Analyst |
| Ellis | Mathew | NULL |

9 rows in set (0.00 sec)

20. Select a.applicantID, a.firstName, a.lastName, a.city as applicantCity, c.companyID, c.companyName, c.companyLocation from applicants a join companies c on a.city = c.companyLocation where a.experience > 2 and c.companyLocation = 'vijayawada';



| applicantID | firstName | lastName | applicantCity | companyID | companyName | companyLocation |
|---|---|---|---|---|---|---|
| 3001 | John | Doe | vijayawada | 1008 | HCL | Vijayawada |
| 3003 | Michael | Johnson | vijayawada | 1008 | HCL | Vijayawada |

2 rows in set (0.00 sec)