-- 1. Initialize the database

DROP DATABASE IF EXISTS CareerHub;



CREATE DATABASE CareerHub;



USE CareerHub;



-- 2,3.Create Tables with Primary Keys, Foreign Keys, and Constraints

CREATE TABLE Companies(

CompanyId INT PRIMARY KEY AUTO\_INCREMENT,

CompanyName VARCHAR(100),

Location VARCHAR(100),

State VARCHAR(100));

CREATE TABLE Jobs(

JobId INT PRIMARY KEY AUTO\_INCREMENT,

CompanyId INT,

JobTitle vARCHAR(100),

JobDescription TEXT,

JobLocation VARCHAR(100),

Salary DECIMAL(10,2),

JobType VARCHAR(50),

PostedDate DATETIME DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (CompanyId) REFERENCES Companies(CompanyId));

CREATE TABLE Applicants(

ApplicantId INT PRIMARY KEY AUTO\_INCREMENT,

FirstName VARCHAR(50),

LastName VARCHAR(50),

Email VARCHAR(100) UNIQUE,

Phone VARCHAR(20),

Resume TEXT,

ExperienceYears INT);

CREATE TABLE Applications(

ApplicationID INT PRIMARY KEY AUTO\_INCREMENT,

JobId INT,

ApplicantId INT,

ApplicationDate DATETIME DEFAULT CURRENT\_TIMESTAMP,

CoverLetter TEXT,

FOREIGN KEY (JobId) REFERENCES Jobs(JobId),

FOREIGN KEY (ApplicantId) REFERENCES Applicants(ApplicantId));

INSERT INTO Companies (CompanyName, Location, State)VALUES

('TechCorp', 'San Francisco', 'California'),

('CloudSolutions', 'New York', 'New York'),

('DataVerse', 'Chicago', 'Illinois'),

('CodeCrafters', 'Austin', 'Texas'),

('NextGen Software', 'Seattle', 'Washington'),

('AI Innovations', 'Boston', 'Massachusetts'),

('WebWorks Studio', 'Denver', 'Colorado'),

('FinTech Flow', 'Atlanta', 'Georgia'),

('GreenByte', 'Portland', 'Oregon'),

('CyberNetics', 'Miami', 'Florida');

INSERT INTO Jobs (CompanyID, JobTitle, JobDescription, JobLocation, Salary, JobType, PostedDate)VALUES

(1, 'Frontend Developer', 'Develop UI using React.', 'San Francisco', 90000, 'Full-time', NOW()),

(2, 'Backend Developer', 'Handle APIs and databases.', 'New York', 95000, 'Full-time', NOW()),

(3, 'Data Analyst', 'Analyze company data.', 'Chicago', 85000, 'Contract', NOW()),

(4, 'DevOps Engineer', 'Manage cloud infrastructure.', 'Austin', 98000, 'Full-time', NOW()),

(5, 'Full Stack Developer', 'Frontend & backend dev.', 'Seattle', 105000, 'Full-time', NOW()),

(6, 'ML Engineer', 'Build machine learning models.', 'Boston', 110000, 'Full-time', NOW()),

(7, 'Web Designer', 'Design modern websites.', 'Denver', 70000, 'Part-time', NOW()),

(8, 'Finance Analyst', 'Analyze market trends.', 'Atlanta', 78000, 'Full-time', NOW()),

(9, 'IT Support', 'Support internal systems.', 'Portland', 60000, 'Full-time', NOW()),

(10, 'Cybersecurity Analyst', 'Monitor threats.', 'Miami', 99000, 'Full-time', NOW());

INSERT INTO Applicants (FirstName, LastName, Email, Phone, Resume, ExperienceYears)VALUES

('John', 'Doe', 'john@example.com', '1234567890', 'Resume 1', 2),

('Jane', 'Smith', 'jane@example.com', '2234567890', 'Resume 2', 4),

('Alice', 'Johnson', 'alice@example.com', '3234567890', 'Resume 3', 3),

('Bob', 'Williams', 'bob@example.com', '4234567890', 'Resume 4', 1),

('Charlie', 'Brown', 'charlie@example.com', '5234567890', 'Resume 5', 5),

('David', 'Miller', 'david@example.com', '6234567890', 'Resume 6', 0),

('Eva', 'Davis', 'eva@example.com', '7234567890', 'Resume 7', 6),

('Frank', 'Wilson', 'frank@example.com', '8234567890', 'Resume 8', 3),

('Grace', 'Moore', 'grace@example.com', '9234567890', 'Resume 9', 2),

('Henry', 'Taylor', 'henry@example.com', '1034567890', 'Resume 10', 4);

INSERT INTO Applications (JobID, ApplicantID, ApplicationDate, CoverLetter)VALUES

(1, 1, NOW(), 'Cover letter from John'),

(2, 2, NOW(), 'Cover letter from Jane'),

(3, 3, NOW(), 'Cover letter from Alice'),

(4, 4, NOW(), 'Cover letter from Bob'),

(5, 5, NOW(), 'Cover letter from Charlie'),

(6, 6, NOW(), 'Cover letter from David'),

(7, 7, NOW(), 'Cover letter from Eva'),

(8, 8, NOW(), 'Cover letter from Frank'),

(9, 9, NOW(), 'Cover letter from Grace'),

(10, 10, NOW(), 'Cover letter from Henry');

-- 4.Error Handling for Existing Tables

USE CareerHub;

DROP TABLE IF EXISTS Applications;

DROP TABLE IF EXISTS Applicants;

DROP TABLE IF EXISTS Jobs;

DROP TABLE IF EXISTS Companies;

CREATE TABLE IF NOT EXISTS Companies (

CompanyID INT PRIMARY KEY AUTO\_INCREMENT,

CompanyName VARCHAR(100),

Location VARCHAR(100),

State VARCHAR(100));

CREATE TABLE IF NOT EXISTS Jobs (

JobID INT PRIMARY KEY AUTO\_INCREMENT,

CompanyID INT,

JobTitle VARCHAR(100),

JobDescription TEXT,

JobLocation VARCHAR(100),

Salary DECIMAL(10, 2),

JobType VARCHAR(50),

PostedDate DATETIME DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (CompanyID) REFERENCES Companies(CompanyID));

CREATE TABLE IF NOT EXISTS Applicants (

ApplicantID INT PRIMARY KEY AUTO\_INCREMENT,

FirstName VARCHAR(50),

LastName VARCHAR(50),

Email VARCHAR(100) UNIQUE,

Phone VARCHAR(20),

Resume TEXT,

ExperienceYears INT);

CREATE TABLE IF NOT EXISTS Applications (

ApplicationID INT PRIMARY KEY AUTO\_INCREMENT,

JobID INT,

ApplicantID INT,

ApplicationDate DATETIME DEFAULT CURRENT\_TIMESTAMP,

CoverLetter TEXT,

FOREIGN KEY (JobID) REFERENCES Jobs(JobID),

FOREIGN KEY (ApplicantID) REFERENCES Applicants(ApplicantID));

INSERT INTO Companies (CompanyName, Location, State)VALUES

('TechCorp', 'San Francisco', 'California'),

('CloudSolutions', 'New York', 'New York'),

('DataVerse', 'Chicago', 'Illinois'),

('CodeCrafters', 'Austin', 'Texas'),

('NextGen Software', 'Seattle', 'Washington'),

('AI Innovations', 'Boston', 'Massachusetts'),

('WebWorks Studio', 'Denver', 'Colorado'),

('FinTech Flow', 'Atlanta', 'Georgia'),

('GreenByte', 'Portland', 'Oregon'),

('CyberNetics', 'Miami', 'Florida');

INSERT INTO Jobs (CompanyID, JobTitle, JobDescription, JobLocation, Salary, JobType, PostedDate)VALUES

(1, 'Frontend Developer', 'Develop UI using React.', 'San Francisco', 90000, 'Full-time', NOW()),

(2, 'Backend Developer', 'Handle APIs and databases.', 'New York', 95000, 'Full-time', NOW()),

(3, 'Data Analyst', 'Analyze company data.', 'Chicago', 85000, 'Contract', NOW()),

(4, 'DevOps Engineer', 'Manage cloud infrastructure.', 'Austin', 98000, 'Full-time', NOW()),

(5, 'Full Stack Developer', 'Frontend & backend dev.', 'Seattle', 105000, 'Full-time', NOW()),

(6, 'ML Engineer', 'Build machine learning models.', 'Boston', 110000, 'Full-time', NOW()),

(7, 'Web Designer', 'Design modern websites.', 'Denver', 70000, 'Part-time', NOW()),

(8, 'Finance Analyst', 'Analyze market trends.', 'Atlanta', 78000, 'Full-time', NOW()),

(9, 'IT Support', 'Support internal systems.', 'Portland', 60000, 'Full-time', NOW()),

(10, 'Cybersecurity Analyst', 'Monitor threats.', 'Miami', 99000, 'Full-time', NOW());

INSERT INTO Applicants (FirstName, LastName, Email, Phone, Resume, ExperienceYears)VALUES

('John', 'Doe', 'john@example.com', '1234567890', 'Resume 1', 2),

('Jane', 'Smith', 'jane@example.com', '2234567890', 'Resume 2', 4),

('Alice', 'Johnson', 'alice@example.com', '3234567890', 'Resume 3', 3),

('Bob', 'Williams', 'bob@example.com', '4234567890', 'Resume 4', 1),

('Charlie', 'Brown', 'charlie@example.com', '5234567890', 'Resume 5', 5),

('David', 'Miller', 'david@example.com', '6234567890', 'Resume 6', 0),

('Eva', 'Davis', 'eva@example.com', '7234567890', 'Resume 7', 6),

('Frank', 'Wilson', 'frank@example.com', '8234567890', 'Resume 8', 3),

('Grace', 'Moore', 'grace@example.com', '9234567890', 'Resume 9', 2),

('Henry', 'Taylor', 'henry@example.com', '1034567890', 'Resume 10', 4);

INSERT INTO Applications (JobID, ApplicantID, ApplicationDate, CoverLetter)VALUES

(1, 1, NOW(), 'Cover letter from John'),

(2, 2, NOW(), 'Cover letter from Jane'),

(3, 3, NOW(), 'Cover letter from Alice'),

(4, 4, NOW(), 'Cover letter from Bob'),

(5, 5, NOW(), 'Cover letter from Charlie'),

(6, 6, NOW(), 'Cover letter from David'),

(7, 7, NOW(), 'Cover letter from Eva'),

(8, 8, NOW(), 'Cover letter from Frank'),

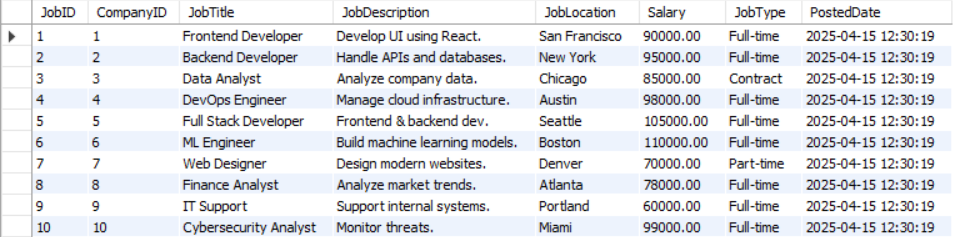
(9, 9, NOW(), 'Cover letter from Grace'),

(10, 10, NOW(), 'Cover letter from Henry');

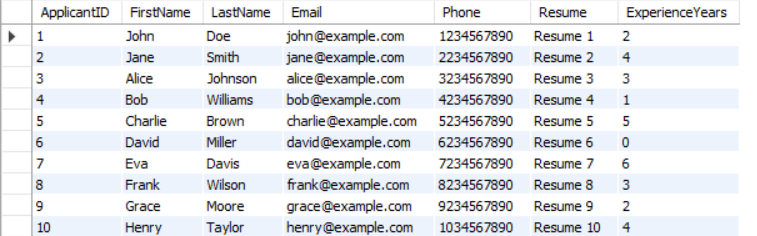
SELECT \* FROM Companies;



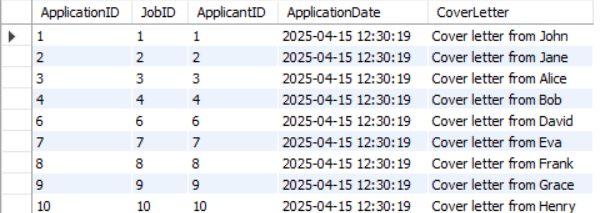
SELECT \* FROM Jobs;



SELECT \* FROM Applicants;



SELECT \* FROM Applications;



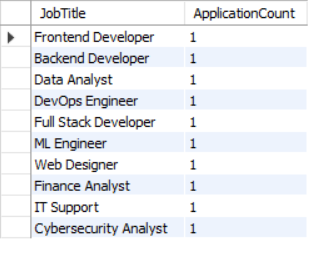
-- 5.Count Applications per Job (Include Jobs with 0 Applications)

SELECT jo.JobTitle,Count(app.ApplicationId) AS ApplicationCount

FROM Jobs jo

LEFT JOIN Applications app ON jo.JobId=app.JobId

GROUP BY jo.JobId,jo.JobTitle;



-- 6.Job Listings within a Salary Range (With Parameters)

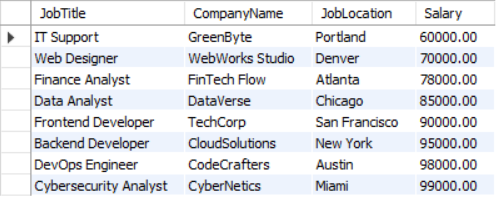
SELECT j.JobTitle,c.CompanyName,j.JobLocation,j.Salary

FROM Jobs j

INNER JOIN Companies c ON j.CompanyId=c.CompanyId

WHERE j.Salary BETWEEN 50000 AND 100000

ORDER BY j.Salary;



-- 7.Job Application History for a Specific Applicant

SELECT j.JobTitle,c.CompanyName,app.ApplicationDate

FROM Applications app

INNER JOIN Jobs j ON app.JobID = j.JobID

INNER JOIN Companies c ON j.CompanyID = c.CompanyID

WHERE app.ApplicantID = 5;



-- 8.Average Salary Offered by All Companies (Excluding Salary 0)

SELECT AVG(Salary) AS AverageSalary

FROM Jobs

WHERE Salary > 0;



-- 9.Company with the Most Job Listings (Handle Ties)

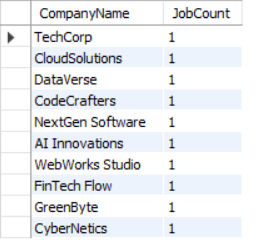
SELECT c.CompanyName,COUNT(j.JobID) AS JobCount

FROM Companies c

LEFT JOIN Jobs j ON c.CompanyID = j.CompanyID

GROUP BY c.CompanyID, c.CompanyName

HAVING COUNT(j.JobID) = (SELECT MAX(JobCount) FROM (SELECT COUNT(JobID) AS JobCount FROM Jobs GROUP BY CompanyID) AS JobCounts);



-- 10.Applicants Who Have Applied for Jobs in Companies Located in 'CityX' with 3+ Years of Experience

UPDATE Companies SET Location = 'CityX' WHERE CompanyID = 3;

SELECT a.FirstName, a.LastName

FROM Applicants a

INNER JOIN Applications app ON a.ApplicantID = app.ApplicantID

INNER JOIN Jobs j ON app.JobID = j.JobID

INNER JOIN Companies c ON j.CompanyID = c.CompanyID

WHERE c.Location = 'CityX' AND a.ExperienceYears >= 3;

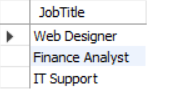


-- 11.Distinct Job Titles with Salaries Between $60,000 and $80,000

SELECT DISTINCT JobTitle

FROM Jobs

WHERE Salary BETWEEN 60000 AND 80000;



-- 12.Jobs with No Applications

DELETE FROM Applications WHERE JobID = 5;

SELECT j.JobTitle,c.CompanyName

FROM Jobs j

LEFT JOIN Applications a ON j.JobID = a.JobID

INNER JOIN Companies c ON j.CompanyID = c.CompanyID

WHERE a.ApplicationID IS NULL;



-- 13.Job Applicants and Their Applied Jobs

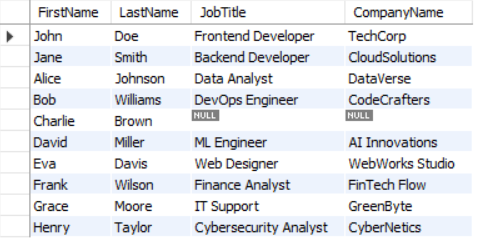
SELECT a.FirstName,a.LastName,j.JobTitle,c.CompanyName

FROM Applicants a

LEFT JOIN Applications app ON a.ApplicantID = app.ApplicantID

LEFT JOIN Jobs j ON app.JobID = j.JobID

LEFT JOIN Companies c ON j.CompanyID = c.CompanyID;



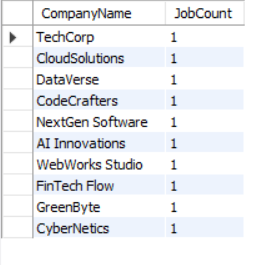
-- 14.Companies and the Count of Jobs They Have Posted (Including Those with No Jobs)

SELECT 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;



-- 15.Applicants and Companies/Positions They Have Applied For (Including Those Who Have Not Applied)

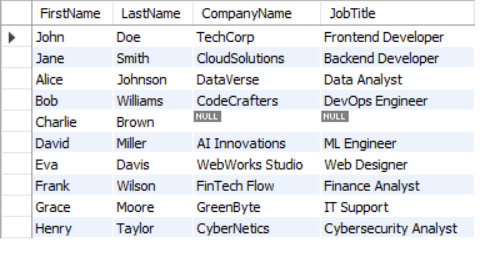
SELECT a.FirstName,a.LastName,c.CompanyName,j.JobTitle

FROM Applicants a

LEFT JOIN Applications app ON a.ApplicantID = app.ApplicantID

LEFT JOIN Jobs j ON app.JobID = j.JobID

LEFT JOIN Companies c ON j.CompanyID = c.CompanyID;



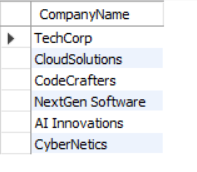
-- 16.Companies with Jobs Offering Salary Higher Than the Average Salary

SELECT c.CompanyName

FROM Companies c

INNER JOIN Jobs j ON c.CompanyID = j.CompanyID

WHERE j.Salary > (SELECT AVG(Salary) FROM Jobs WHERE Salary > 0);



-- 17.Applicants with Their Names and Concatenated

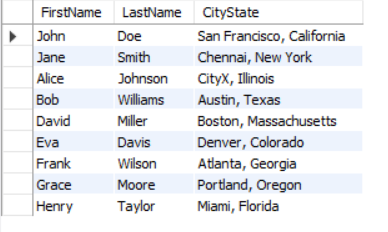
SELECT a.FirstName,a.LastName,CONCAT(c.Location, ', ', c.State) AS CityState

FROM Applicants a

INNER JOIN Applications app ON a.ApplicantID = app.ApplicantID

INNER JOIN Jobs j ON app.JobID = j.JobID

INNER JOIN Companies c ON j.CompanyID = c.CompanyID;



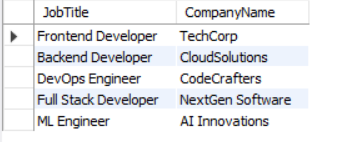
-- 18.Jobs with Titles Containing 'Developer' or 'Engineer'

SELECT j.JobTitle,c.CompanyName

FROM Jobs j

INNER JOIN Companies c ON j.CompanyID = c.CompanyID

WHERE j.JobTitle LIKE '%Developer%' OR j.JobTitle LIKE '%Engineer%';



-- 19.Applicants and Jobs They Have Applied For (Including Those Who Have Not Applied and Jobs Without Applicants)

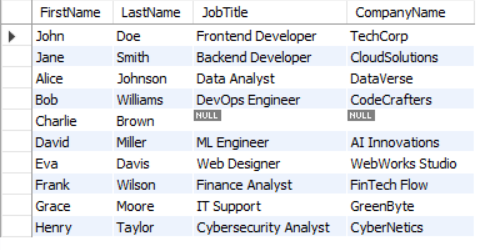
SELECT a.FirstName,a.LastName,j.JobTitle,c.CompanyName

FROM Applicants a

LEFT JOIN Applications app ON a.ApplicantID = app.ApplicantID

LEFT JOIN Jobs j ON app.JobID = j.JobID

LEFT JOIN Companies c ON j.CompanyID = c.CompanyID;



-- 20.Applicants and Companies Where Company is in Specific City and Applicant Has More Than 2 Years of Experience

UPDATE Companies SET Location = 'Chennai' WHERE CompanyID = 2;

SELECT a.FirstName,a.LastName,c.CompanyName

FROM Applicants a

INNER JOIN Applications app ON a.ApplicantID = app.ApplicantID

INNER JOIN Jobs j ON app.JobID = j.JobID

INNER JOIN Companies c ON j.CompanyID = c.CompanyID

WHERE c.Location = 'Chennai' AND a.ExperienceYears > 2;

