1. Table companies

create table companies(CompanyId integer(10),

-> CompanyName varchar(30),

-> Location varchar(30),

-> constraint companies\_CompanyId\_pk primary key (CompanyId));

A black screen with white text

Description automatically generated

2. Table Jobs

create table Jobs(JobId integer(10),

-> CompanyId integer(10),

-> JobTitle Varchar(20),

-> JobDescription Text(50),

-> JobLocation varchar(30),

-> Salary Decimal(10),

-> JobType varchar(20),

-> PostedDate DateTime ,

-> constraint Jobs\_JobId\_pk primary key(JobId),

-> constraint Jobs\_CompanyId\_fk foreign key(CompanyId) references compan

ies(CompanyId));

A screen shot of a computer program

Description automatically generated

3 . Table Applicants

create table Applicants(ApplicatId integer(10),

-> FirstName varchar(20),

-> LastName varchar(20),

-> Email varchar(20),

-> Phone varchar(20),

-> Resume Text(50),

-> constraint Applicants\_ApplicatId\_pk primary key(ApplicatId));

A screen shot of a computer

Description automatically generated

4. Table Applications

create table Applications(ApplicationID integer(10),

-> JobID integer(10),

-> ApplicantId integer(10),

-> ApplicationDate DateTime,

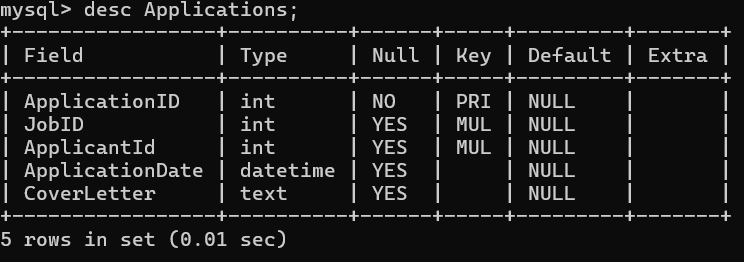
-> CoverLetter Text,

-> constraint Applications\_ApplicationID\_pk primary key (ApplicationID),

-> constraint Applications\_JobID\_fk foreign key(JobID) references jobs(j

obId),

-> constraint Applications\_ApplicantId\_fk foreign key(ApplicantId) references Applicants(ApplicatId));



Inserting values :

->insert into companies values(101,'hexaware tech','chennai');

-> insert into companies values(102,'vipro','bangalore');

->insert into companies values(103,'infosys','hyderabad');

->insert into companies values(104,'westerdigital','bangalore');

->insert into companies values(105,'bosch','delhi');

A black screen with white text

Description automatically generated

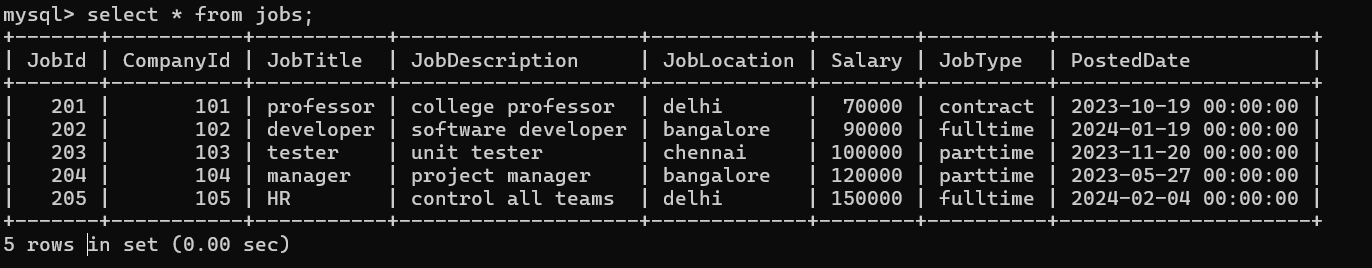
->insert into jobs values(201,101,'professor','college professor','delhi',70000,'contract','2023-10-19');

->insert into jobs values(202,102,'developer','software developer','bangalore',90000,'fulltime','2024-01-19');

->insert into jobs values(203,103,'tester','unit tester','chennai',100000,'parttime','2023-11-20');

-> insert into jobs values(204,104,'manager','project manager','bangalore',120000,'parttime','2023-05-27');

-> insert into jobs values(205,105,'HR','control all teams','delhi',150000,'fulltime','2024-02-04');



->insert into applicants values(1,'Ganga','yadahalli','ganga@gmail.com',9087657897,'fresher');

->insert into applicants values(2,'Gouri','sukali','gouri@gmail.com',9980765481,'working as an employee');

->insert into applicants values(3,'Aishwarya','TS','aishwarya@gmail.com',7789690765,'working as an developer');

->insert into applicants values(4,'siya','bhatt','siya@gmail.com',9898767897,'skilled in java');

-> insert into applicants values(5,'bhumika','malaghan','bhumika@gmail.com',9878754318,'skilled in c++');

A screen shot of a computer

Description automatically generated

-> insert into applications values(1001,201,1,'2023-09-11','deatiled cover letter');

->insert into applications values(1002,202,2,'2023-12-20','cover letter description');

->insert into applications values(1003,203,3,'2023-08-28','cover letter details provided');

->insert into applications values(1004,204,4,'2023-02-18','cover letter details');

-> insert into applications values(1005,205,5,'2024-01-15','description of cover letter ');

A screen shot of a computer

Description automatically generated

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.jobtitle , count(a.applicationId) as count from jobs j

-> left join applications a on j.jobId=a.jobId

-> group by j.jobtitle;

A black screen with white text

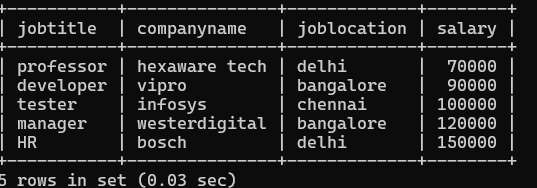
Description automatically generated

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.jobtitle , c.companyname , j.joblocation, j.salary from jobs as j

->join companies c ON j.CompanyId = c.CompanyId

->where j.salary between 10000 and 150000;



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.jobtitle ,c.companyname ,a.applicationdate from applications as a

-> join jobs as j on a.jobId=j.jobid

-> join companies as c on c.companyId=j.companyid

-> where a.applicantId =3;

A black screen with white text

Description automatically generated

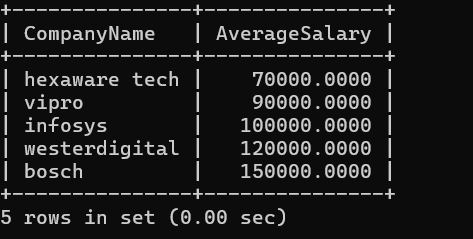
8.=> Create an SQL query that calculates and displays the average salary offered by all companies for job listings in the "Jobs" table. Ensure that the query filters out jobs with a salary of zero.

SELECT c.CompanyName, AVG(j.Salary) AS AverageSalary

-> FROM companies c INNER JOIN jobs j ON c.CompanyId = j.CompanyId

-> WHERE j.Salary > 0

-> GROUP BY c.CompanyName;



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

=> alter table jobs add (experience integer(20));

-> update jobs set experience=3 where jobId=201;

->update jobs set experience=4 where jobId=202;

->update jobs set experience=1 where jobId=203;

-> update jobs set experience=2 where jobId=204;

A black screen with white text

Description automatically generated

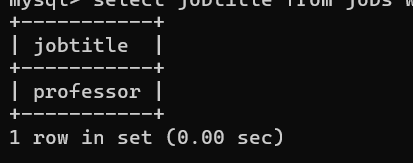
->select jobtitle,joblocation from jobs where experience >= 3 and jobtitle =’delhi’;

A black screen with white text

Description automatically generated

11.=> Retrieve a list of distinct job titles with salaries between $60,000 and $80,000

select distinct(jobtitle) from jobs where salary between 60000 and 80000;

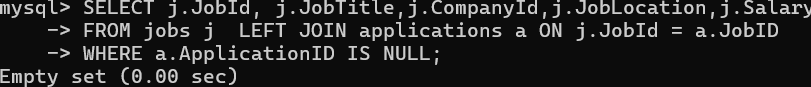


12 .=> Find the jobs that have not received any applications.

SELECT j.JobId, j.JobTitle,j.CompanyId,j.JobLocation,j.Salary

-> FROM jobs j LEFT JOIN applications a ON j.JobId = a.JobID

-> WHERE a.ApplicationID 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.FirstName, a.LastName, c.CompanyName, j.JobTitle

->FROM applicants a INNER JOIN applications app ON a.ApplicatId = app.ApplicantId -> INNER JOIN jobs j ON app.JobID = j.JobId

->INNER JOIN companies c ON j.CompanyId = c.CompanyId;

A black screen with white text

Description automatically generated

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.CompanyName, COUNT(j.JobId) AS JobCount

-> FROM companies c LEFT JOIN jobs j ON c.CompanyId = j.CompanyId

-> group by c.CompanyName;

A black screen with white text

Description automatically generated

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

SELECT a.ApplicatId, a.FirstName, a.LastName, a.Email, a.Phone, c.CompanyName, j.JobTitle

->FROM Applicants a LEFT JOIN Applications ap ON a.ApplicatId = ap.ApplicantId

-> LEFT JOIN Jobs j ON ap.JobID = j.JobId

->LEFT JOIN Companies c ON j.CompanyId = c.CompanyId ORDER BY a.ApplicatId;

A screen shot of a computer

Description automatically generated

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

SELECT c.CompanyName,j.JobTitle,j.salary from

-> companies c INNER JOIN jobs j ON c.CompanyId = j.CompanyId

-> WHERE j.Salary > (SELECT AVG(Salary) FROM jobs);

A black screen with white text

Description automatically generated

17.=> Display a list of applicants with their names and a concatenated string of their city and state.

-> alter table applicants add(state varchar(20));

mysql> update applicants set state='tamilnadu' where applicatId=1;

mysql> update applicants set state='karnataka' where applicatId=2;

mysql> update applicants set state='andrapradesh' where applicatId=3;

mysql> update applicants set state='karanatka' where applicatId=4;

mysql> update applicants set state='up' where applicatId=5;

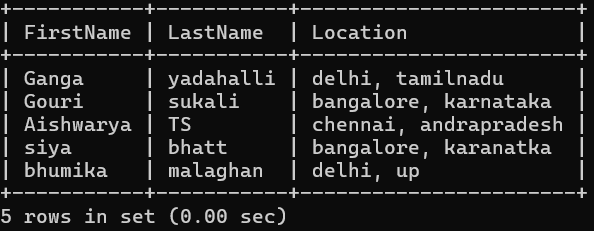
A screen shot of a computer

Description automatically generated

SELECT a.FirstName,a.LastName,CONCAT(j.JobLocation, ', ', a.State) AS Location

-> FROM applicants a INNER JOIN applications app ON a.ApplicatId = app.ApplicantId

-> INNER JOIN jobs j ON app.JobID = j.JobId;



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

->select jobtitle ,jobId from jobs where jobTitle='developer' or jobtitle='engineer';

A black screen with white text

Description automatically generated

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, j.JobLocation FROM applicants a LEFT JOIN applications app ON a.ApplicatId = app.ApplicantId LEFT JOIN jobs j ON app.JobID = j.JobId

-> UNION

-> SELECT a.FirstName, a.LastName, j.JobTitle, j.JobLocation FROM applicants a RIGHT JOIN applications app ON a.ApplicatId = app.ApplicantId RIGHT JOIN jobs j ON app.JobID = j.JobId where applicatid=null;

A black screen with white text

Description automatically generated

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 applicants a

-> INNER JOIN applications app ON a.ApplicatId = 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 j.Experience > 2;

A black screen with white text

Description automatically generated