**MySql\_Examination**

**Task- 1:**

Create two tables: users and orders.

Each user can have multiple orders.

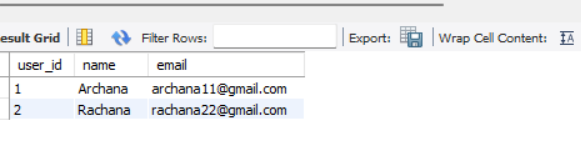
Sol:

use training;

create table users(user\_id int(10),name varchar(50),email varchar(100));

insert into users values('1','Archana','archana11@gmail.com'),('2','Rachana','rachana22@gmail.com');

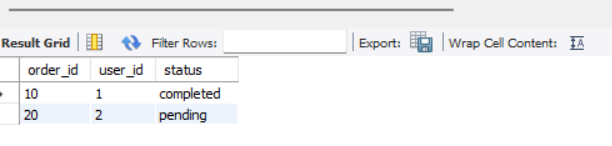
select \* from users;



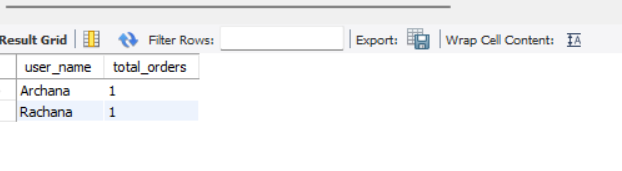
Create table orders(order\_id int(20),user\_id int(30),status varchar(35));

insert into orders values ('10','1','completed'),('20','2','pending');

select \* from orders;



select u.name AS user\_name,(select count(\*)from orders o where o.user\_id=u.user\_id) AS total\_orders from users u;



**Task-2:**

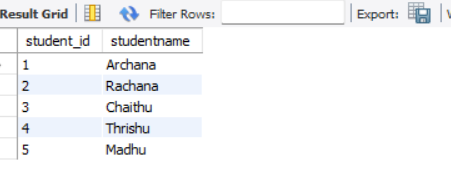
You are working with a database that stores information about students and their courses. There are three tables: students, courses, and enrollments.

Write a SQL query to display the names of students along with the courses they have enrolled in.

**Sol: create table students(student\_id int(20),studentname varchar(45));**

**insert into students values ('1','Archana'),('2','Rachana'),('3','Chaithu'),('4','Thrishu'),('5','Madhu');**

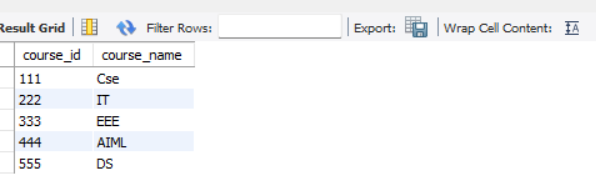
**select \* from students;**

****

**create table courses(course\_id int(30),course\_name varchar(30));**

**insert into courses values('111','Cse'),('222','IT'),('333','EEE'),('444','AIML'),('555','DS');**

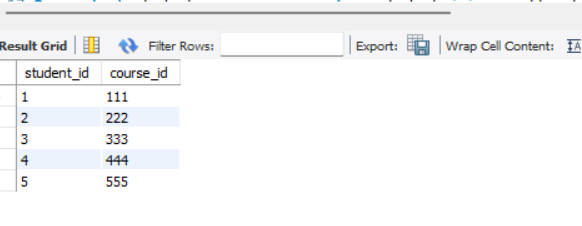
**select \* from courses;**

****

**create table enrollments(student\_id int(30),course\_id int(45));**

**insert into enrollments values('1','111'),('2','222'),('3','333'),('4','444'),('5','555');**

**select \* from enrollments;**

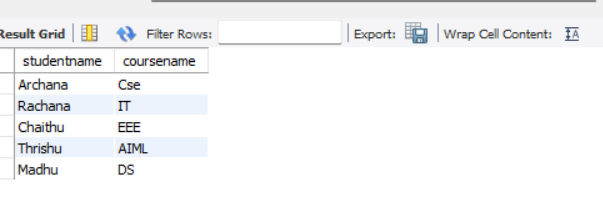
****

**create table enrollment(studentid int(30),courseid int(45));**

**insert into enrollment values('1','111'),('2','222'),('3','333'),('4','444'),('5','555');**

**select \* from enrollment;**

**select studentname,coursename from student join enrollment on student.studentid=enrollment.studentid join course on enrollment.courseid=course.courseid;**

****

**Task-3:**

You need to retrieve data from a database that tracks product sales. There are tables for products, sales, and customers

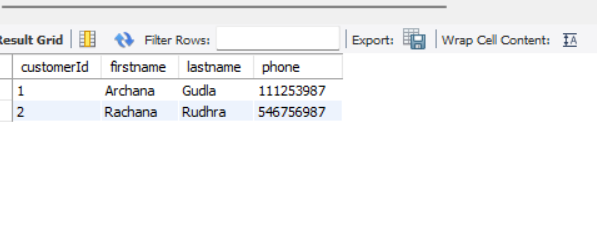
Write a SQL query to show the total sales amount for each product category.

Sol:

create table customer(customerId int(20),firstname varchar(50),lastname varchar(50),phone int(100));

insert into customer values('1','Archana','Gudla','111253987'),('2','Rachana','Rudhra','546756987');

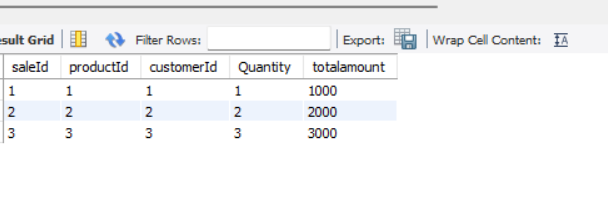
select \* from customer;



create table sales(saleId int(20),productId int(30),customerId int(40),Quantity int(40), totalamount int(100));

insert into sales values('1','1','1','1','1000'),('2','2','2','2','2000'),('3','3','3','3','3000');

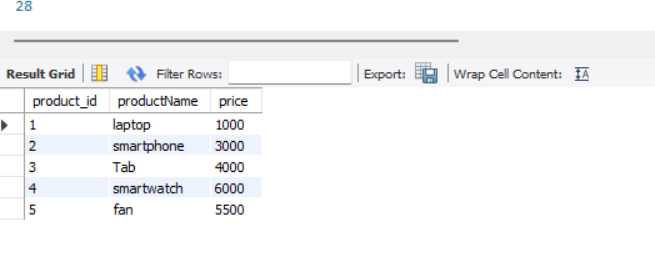
select \* from sales;



create table product(product\_id int(40),productName varchar(45), price int(45));

insert into product values('1','laptop','1000'),('2','smartphone','3000'),('3','Tab','4000'),('4','smartwatch','6000'),('5','fan','5500');

select \* from product;



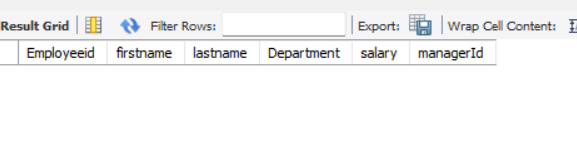
**Task-4:**

You have a database containing information about employees in a company.

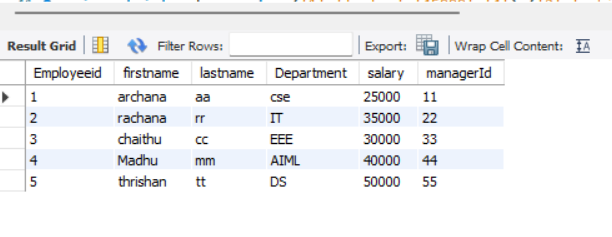
Write a SQL query to list the names of employees along with their respective managers' names.

Sol:

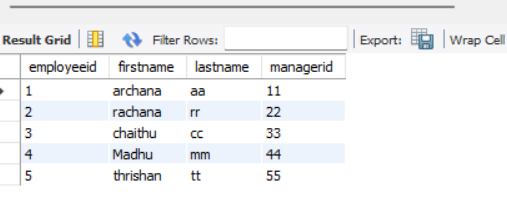
create table store(productID int(20),productName varchar(50),price int(50),selling int(50));



insert into employees values('1','archana','aa','cse','25000','11'),('2','rachana','rr','IT','35000','22'),('3','chaithu','cc','EEE','30000','33'),('4','Madhu','mm','AIML','40000','44'),('5','thrishan','tt','DS','50000','55');



select employeeid,firstname,lastname,managerid from employees;



**Task-5:**

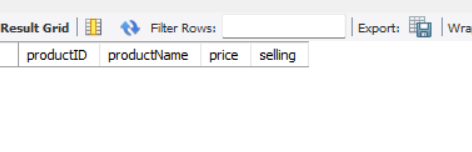
You are managing a database for an online store.

Write a query to retrieve the top 10 bestselling products based on the total number of units sold.

Sol:

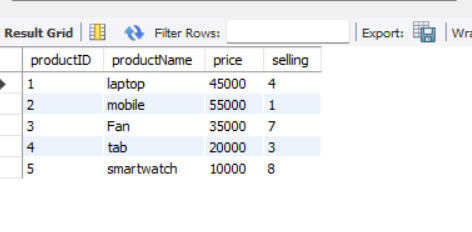
create table store(productID int(20),productName varchar(50),price int(50),selling int(50));

select \* from store;



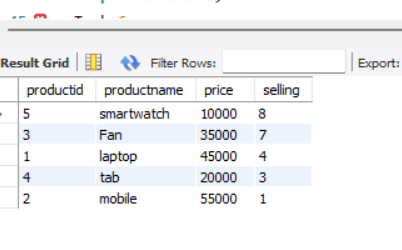
insert into store values('1','laptop','45000','4'),('2','mobile','55000','1'),('3','Fan','35000','7'),('4','tab','20000','3'),('5','smartwatch','10000','8');

select \* from store;



select productid,productname, price,selling from store order by selling Desc limit 5;

select \* from store;



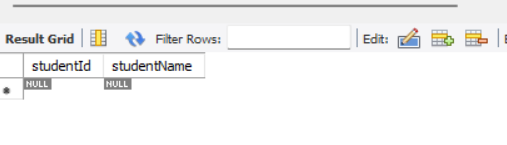
**Task-6:**

You have tables for students, courses, and grades.

Write a SQL query to display the average grade for each student.

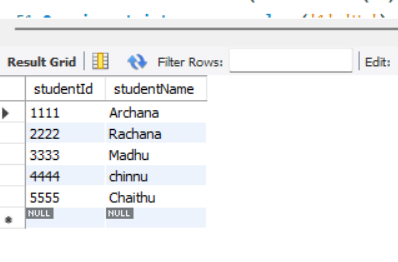
Sol: create table student(studentId int(40) primary key,studentName varchar(30));

select \* from student;



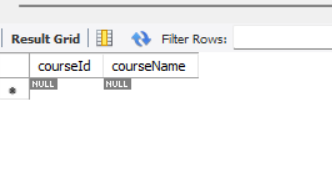
insert into student values(1111,'Archana'),(2222,'Rachana'),(3333,'Madhu'),(4444,'chinnu'),(5555,'Chaithu');

select \* from student;



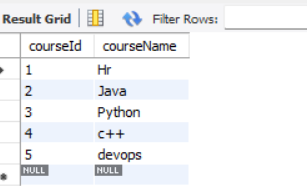
create table course(courseId int(30) primary key,courseName varchar(40));

select \* from course;

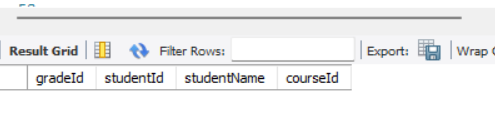


insert into course values('1','Hr'),('2','Java'),('3','Python'),('4','c++'),('5','devops');

select \* from course;



create table grades(gradeId int(30),studentId int(40),studentName varchar(50),courseId int(30),foreign key (studentId) references student(studentId),foreign key (courseId) references course(courseId));



insert into grades values('11','1111','Archana','1');

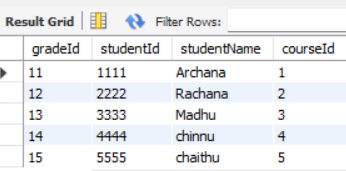
insert into grades values('12','2222','Rachana','2');

insert into grades values('13','3333','Madhu','3');

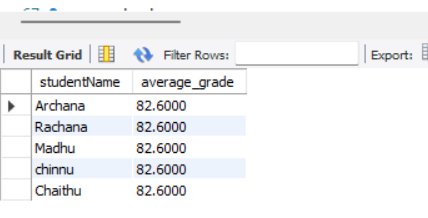
insert into grades values('14','4444','chinnu','4');

insert into grades values('15','5555','chaithu','5');

select \* from grades;



select studentName,(select avg(grade) from grades where grades.studentid=studentId) as average\_grade from student;



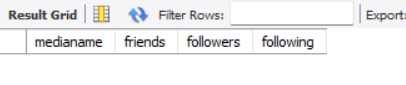
**Task-7:**

You are working with a database for a social media platform.

Write a query to show the users who have the most friends.

Sol:

Create table socialmedia(medianame varchar(50),friends int(40),followers int(50),following int(55));



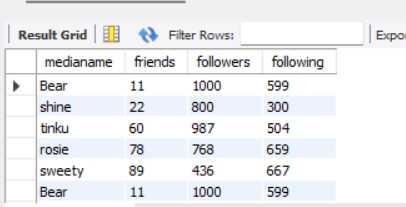
insert into socialmedia values('Bear','11','1000','599');

insert into socialmedia values('shine','22','800','300');

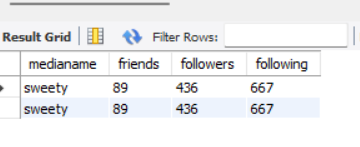
insert into socialmedia values('tinku','60','987','504');

insert into socialmedia values('rosie','78','768','659');

insert into socialmedia values('sweety','89','436','667');



SELECT \* FROM socialmedia WHERE friends = (SELECT MAX(friends) FROM socialmedia);



**Task-8:**

You have tables for employees and departments.

Write a query to display the department names along with the total number of employees in each department.

Sol: create table employes(Employeeid int(35),firstname varchar(50),lastname varchar(50),DepartmentId int(40),salary int(50),managerId int(40));

insert into employes values('1','archana','aa','100','25000','50');

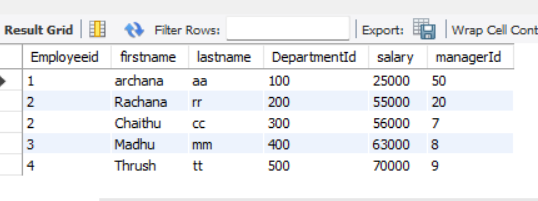
insert into employes values('2','Rachana','rr','200','55000','20');

insert into employes values('2','Chaithu','cc','300','56000','7');

insert into employes values('3','Madhu','mm','400','63000','8');

insert into employes values('4','Thrush','tt','500',70000,'9');

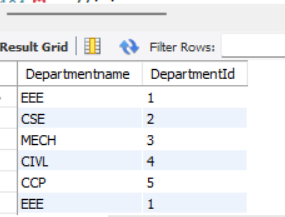
select \* from employes;



create table Department(Departmentname varchar(50),DepartmentId int(45));

insert into Department values('EEE','1'),('CSE','2'),('MECH','3'),('CIVL','4'),('CCP','5');

select \* from department;



select count(employeeid) as total,departmentname from department join employe on employe.departmentId=departments.departmentid group by department.departmentname;

**Task-9:**

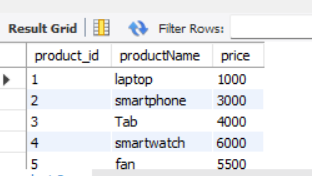
You need to retrieve data from a database tracking product inventory.

Write a query to display products with low stock (less than 10 units).

Sol:

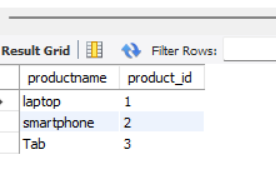
create table product(product\_id int(40),productName varchar(45), price int(45));

insert into product values('1','laptop','1000'),('2','smartphone','3000'),('3','Tab','4000'),('4','smartwatch','6000'),('5','fan','5500');



select \* from product;

select productname,product\_id from product where quamtity<10;



**Task-10:**

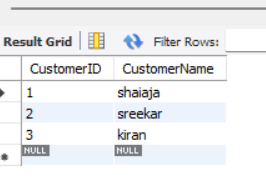
You have tables for customers and orders.

Write a query to show the average order value for each customer.

CREATE TABLE customer (CustomerID INT(10) PRIMARY KEY, CustomerName VARCHAR(100));

INSERT INTO customer VALUES (1,'shaiaja'),(2,'sreekar'),(3,'kiran');

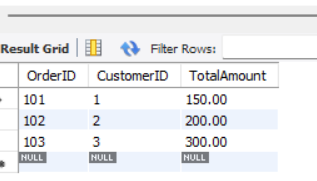
select \* from customer;



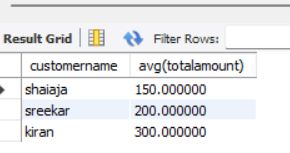
CREATE TABLE orders (OrderID INT(10) PRIMARY KEY,CustomerID INT(10), TotalAmount DECIMAL(10, 2),FOREIGN KEY (CustomerID) REFERENCES customers(CustomerID) );

INSERT INTO orders VALUES (1, 101,150.00), (1,102, 200.00), (2,103, 300.00), (3,104, 400.00), (3,105, 500.00);

select \* from orders;



select customername,avg(totalamount) from customer join orders on customer.customerid=orders.customerid group by customername;



**Task-11:**

In a database storing movie information,

Write a query to show the top 5 highest-rated movies by users.

Sol:

CREATE TABLE Movies (MovieID INT PRIMARY KEY,Title VARCHAR(255) NOT NULL,ReleaseYear INT,Genre VARCHAR(100));

INSERT INTO Movies (MovieID, Title, ReleaseYear, Genre) VALUES

(1, 'Inception', 2010, 'Sci-Fi'),

(2, 'The Godfather', 1972, 'Crime'),

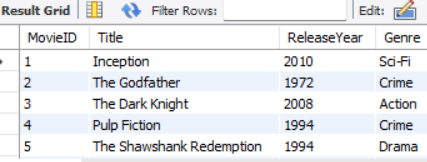
(3, 'The Dark Knight', 2008, 'Action'),

(4, 'Pulp Fiction', 1994, 'Crime'),

(5, 'The Shawshank Redemption', 1994, 'Drama'),

(6, 'Forrest Gump', 1994, 'Drama');

select \* from movies;



**CREATE TABLE Ratings (RatingID INT PRIMARY KEY,MovieID INT NOT NULL,UserID INT NOT NULL,Rating DECIMAL(3, 2) CHECK (Rating >= 0 AND Rating <= 10),FOREIGN KEY (MovieID) REFERENCES Movies(MovieID)**

**);**

**INSERT INTO Ratings (RatingID, MovieID, UserID, Rating) VALUES**

**(1, 1, 101, 9.5),**

**(2, 1, 102, 9.0),**

**(3, 2, 103, 9.8),**

**(4, 3, 104, 9.7),**

**(5, 3, 105, 9.6),**

**(6, 4, 106, 9.4),**

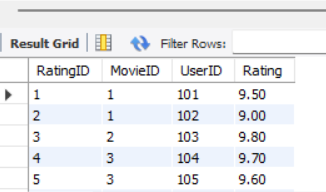
**(7, 4, 107, 9.2),**

**(8, 5, 108, 9.9),**

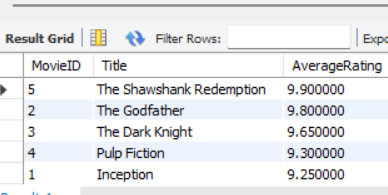
**(9, 6, 109, 9.1),**

**(10, 6, 110, 8.9);**

**select \* from Ratings;**

****

**SELECT M.MovieID,M.Title,AVG(R.Rating) AS AverageRatingFROM Movies MJOIN Ratings RON M.MovieID = R.MovieIDGROUP BY M.MovieID, M.TitleORDER BY AverageRating DESCLIMIT 5;**

****

**Task-12:**

You have tables for invoices and payments.

Write a query to show the unpaid invoices and their total amount.

Sol:

CREATE TABLE Invoices (InvoiceID INT PRIMARY KEY,CustomerID INT NOT NULL,InvoiceDate DATE NOT NULL,TotalAmount DECIMAL(10, 2) NOT NULL);

INSERT INTO Invoices (InvoiceID, CustomerID, InvoiceDate, TotalAmount) VALUES

(1, 101, '2024-01-01', 500.00),

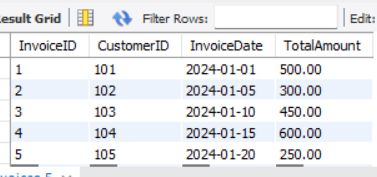
(2, 102, '2024-01-05', 300.00),

(3, 103, '2024-01-10', 450.00),

(4, 104, '2024-01-15', 600.00),

(5, 105, '2024-01-20', 250.00);

select \* from Invoices ;



CREATE TABLE Payments (PaymentID INT PRIMARY KEY,InvoiceID INT NOT NULL,PaymentDate DATE NOT NULL,AmountPaid DECIMAL(10, 2) NOT NULL,FOREIGN KEY (InvoiceID) REFERENCES Invoices(InvoiceID)

);

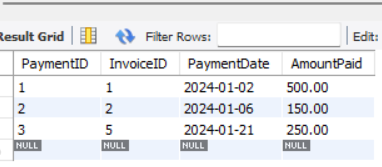
INSERT INTO Payments (PaymentID, InvoiceID, PaymentDate, AmountPaid) VALUES

(1, 1, '2024-01-02', 500.00),

(2, 2, '2024-01-06', 150.00),

(3, 5, '2024-01-21', 250.00);

select \* from payments;



SELECT

I.InvoiceID,I.CustomerID,I.InvoiceDate,I.TotalAmount,COALESCE(SUM(P.AmountPaid), 0) AS TotalPaid,(I.TotalAmount - COALESCE(SUM(P.AmountPaid), 0)) AS UnpaidAmount

FROM

Invoices I LEFT JOIN Payments P ON I.InvoiceID = P.InvoiceID GROUP BY I.InvoiceID, I.CustomerID, I.InvoiceDate, I.TotalAmount HAVING (I.TotalAmount - COALESCE(SUM(P.AmountPaid), 0)) > 0

ORDER BY

I.InvoiceDate;

