**Task- 1:**

Create two tables: users and orders.

Each user can have multiple orders.

Write a SQL query to fetch the names of users along with the total number of orders they have placed.

Code:

/\* create the database \*/

create database task1;

/\* use the database \*/

use task1;

/\* create the tables for users and orders \*/

/\* users table \*/

create table users(

user\_id int primary key,

user\_name varchar(50)

);

select \* from users;

drop table orders;

/\* orders table \*/

create table orders(

order\_id int primary key,

user\_id int,

order\_date date,

foreign key(user\_id) references users(user\_id)

);

select \* from orders;

/\* insert the values into the users \*/

insert into users values

(1,"santosh"),

(2,"manibabu"),

(3,"gopi nadh"),

(4,"hari krishna"),

(5,"jeevan"),

(6,"vignan");

select \* from users;

/\* insert the values into the orders \*/

set foreign\_key\_checks=0;

insert into orders (order\_id,user\_id,order\_date) values

(1,1,"2024-01-01"),

(2,1,"2024-01-05"),

(3,2,"2024-01-10"),

(4,2,"2024-01-15"),

(5,3,"2024-01-20"),

(6,3,"2024-01-25");

select \* from orders;

select \* from orders;

select \* from users;

/\* fetch the names with orders \*/

/\* for join the both tables on left \*/

select \* from users left join orders on

users.user\_id = orders.user\_id;

/\* count the orders \*/

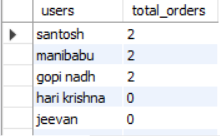
select users.user\_name as users, count(orders.order\_id) as total\_orders from users

left join orders on users.user\_id=orders.user\_id group by

users.user\_name, users.user\_id;

select \* from orders;

Output:



**Task-2:**

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

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

Code:

create database task2;

use task2;

create table students(

std\_id int primary key,

std\_name varchar(45)

);

insert into students values

(1,"santosh"),

(2,"mani babu"),

(3,"jeevan"),

(4,"gopi"),

(5,"hari"),

(6,"vignan");

select \* from students;

create table courses(

crs\_id int primary key,

crs\_title varchar(45)

);

insert into courses values

(1,"Full stack java"),

(2,"Mern stack"),

(3,"devops"),

(4,"full stack python"),

(5,"full stack dot net"),

(6,"full stack java");

select \* from courses;

create table enrollments(

std\_id int,

crs\_id int,

foreign key(std\_id) references students(std\_id),

foreign key(crs\_id) references courses(crs\_id)

);

insert into enrollments values

(1,2),

(1,3),

(2,3),

(2,4),

(3,4),

(3,5);

select \* from enrollments;

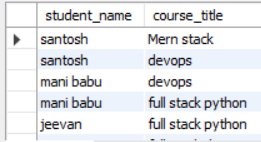
select students.std\_name as student\_name,

courses.crs\_title as course\_title from enrollments join

students on enrollments.std\_id = students.std\_id join

courses on enrollments.crs\_id=courses.crs\_id;

Output:



**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.

Code:

create database task3;

use task3;

create table products(

pro\_id int primary key,

pro\_name varchar(45),

pro\_category varchar(45),

pro\_price decimal(10,2)

);

insert into products values

(1,"laptops","electronics",250.00),

(2,"Head phones","electronics",100.00),

(3,"Chairs","furnitures",50.00),

(4,"cars","automobiles",2500.00),

(5,"mobiles","electronics",500.00);

select \* from products;

create table sales (

sale\_id int primary key,

pro\_id int,

qty int,

foreign key(pro\_id) references products(pro\_id)

);

insert into sales values

(1,1,4),

(2,2,8),

(3,3,12),

(4,4,16),

(5,5,20);

select \* from sales;

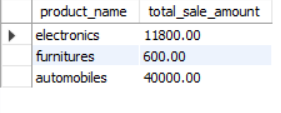
select products.pro\_category as product\_name,

sum(products.pro\_price \* sales.qty) as total\_sale\_amount from products join

sales on products.pro\_id = sales.pro\_id group by

product\_name;

Output:



**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.

Code:

create database task4;

use task4;

create table manager(

mgr\_id int primary key,

mgr\_role varchar(50)

);

create table employees(

emp\_id int primary key,

emp\_name varchar(45),

mgr\_id int,

foreign key(mgr\_id) references manager(mgr\_id)

);

insert into manager values

(1,"mahesh"),

(2,"suresh"),

(3,"prakash"),

(4,"ganesh"),

(5,"bhaskar");

update manager set mgr\_role="bhaskar" where mgr\_id=5;

select \* from manager;

insert into employees values

(1,"santosh",1),

(2,"mani babu",2),

(3,"gopinadh",3),

(4,"jeevan",4),

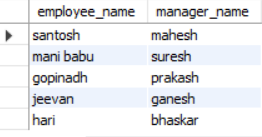
(5,"hari",5);

select \* from employees;

select employees.emp\_name as employee\_name,

manager.mgr\_role as manager\_role from employees left join manager on employees.mgr\_id = manager.mgr\_id;

Output:



**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.

Code:

create database task5;

use task5;

create table products(

pdt\_id int primary key,

pdt\_name varchar(50)

);

insert into products values

(1,"mobiles"),

(2,"laptops"),

(3,"air conditioners"),

(4,"televisions"),

(5,"cloths"),

(6,"shoes"),

(7,"gadgets"),

(8,"books"),

(9,"gifts"),

(10,"choclates");

create table orders(

odr\_id int primary key,

pdt\_id int,

pdt\_qty int,

foreign key(pdt\_id) references products(pdt\_id)

);

insert into orders values

(1,1,2),

(2,2,3),

(3,3,5),

(4,4,8),

(5,5,3),

(6,6,4),

(7,7,10),

(8,8,2),

(9,9,12),

(10,10,7);

use task5;

select products.pdt\_name as product\_name,

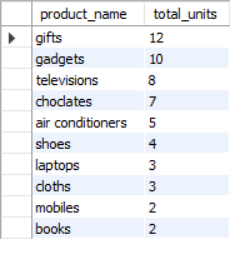
sum(orders.pdt\_qty) as total\_units from products join

orders on products.pdt\_id=orders.pdt\_id group by

products.pdt\_id,products.pdt\_name order by total\_units desc

limit 10;

Output:



**Task-6:**

You have tables for students, courses, and grades.

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

Code:

drop database task6;

create database task6;

use task6;

create table students(

std\_id int primary key,

std\_name varchar(50)

);

insert into students values

(1,"santosh"),

(2,"manibabu"),

(3,"gopi"),

(4,"jeevan"),

(5,"hari"),

(6,"vignan"),

(7,"santosh"),

(8,"gopi"),

(9,"hari"),

(10,"santosh");

select \* from students;

create table courses(

crs\_id int primary key,

crs\_name varchar(50)

);

insert into courses values

(1,"python"),

(2,"java"),

(3,"javascript"),

(4,"mern stack"),

(5,"full stack"),

(6,"dot Net"),

(7,"java"),

(8,"python"),

(9,"javascript"),

(10,"php");

select \*from courses;

create table grades(

grades\_id int,

std\_id int,

crs\_id int,

grade int,

foreign key(std\_id) references students(std\_id),

foreign key(crs\_id) references courses(crs\_id)

);

insert into grades values

(1,1,1,10),

(2,2,2,9),

(3,3,3,6),

(4,4,4,8),

(5,5,5,5),

(6,6,6,10),

(7,7,7,5),

(8,8,8,6),

(9,9,9,10),

(10,10,10,8);

select \* from grades;

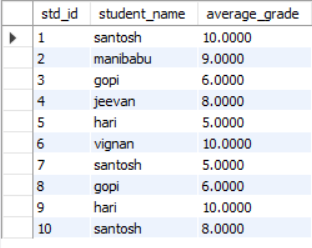
select students.std\_id, students.std\_name as student\_name,

avg(grades.grade) as average\_grade from students join

grades on students.std\_id = grades.grades\_id group by

students.std\_id,students.std\_name order by students.std\_id;

Output:



**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.

Code:

create database task7;

use task7;

create table users (

user\_id int primary key,

user\_name varchar(50),

friends\_count int

);

insert into users values

(1,"santosh",10),

(2,"mani babu",8),

(3,"jeevan",15),

(4,"haribabu",11),

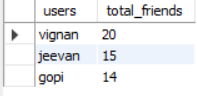
(5,"gopi",14),

(6,"vignan",20);

select \* from users;

select user\_id, user\_name, friends\_count as total\_friends from users order by total\_friends desc limit 3;

Output:



**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.

Code:

create database task8;

use task8;

create table departments (

dept\_id int primary key,

dept\_name varchar(50)

);

insert into departments values

(1,"police"),

(2,"railways"),

(3,"aviation"),

(4,"busses");

select \* from departments;

create table employees(

emp\_id int primary key,

emp\_name varchar(50),

dept\_id int,

foreign key(dept\_id) references departments(dept\_id)

);

insert into employees values

(1,"santosh",1),

(2,"gopi",2),

(3,"jeevan",3),

(4,"vignan",4);

select \* from employees;

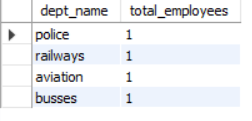
select departments.dept\_name, count(employees.emp\_id) as

total\_employees from departments left join employees on

departments.dept\_id = employees.dept\_id group by

departments.dept\_name order by total\_employees desc;

Output:



**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).

Code:

create database task9;

use task9;

create table products(

pdt\_id int,

pdt\_name varchar(50),

pdt\_stock int

);

insert into products values

(1,"laptops",9),

(2,"cameras",12),

(3,"phones",5),

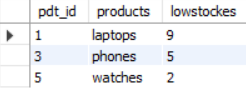
(4,"tablets",15),

(5,"watches",2);

select products.pdt\_id, products.pdt\_name as products,products.pdt\_stock as

lowstockes from products where products.pdt\_stock < 10;

Output:



**Task-10:**

You have tables for customers and orders.

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

Code:

create database task10;

use task10;

create table customers (

cms\_id int primary key,

cms\_name varchar(50)

);

create table orders(

odr\_id int primary key,

cms\_id int,

odr\_orders int,

foreign key(cms\_id) references customers(cms\_id)

);

insert into customers values

(1,"santosh"),

(2,"gopi"),

(3,"hari"),

(4,"vignan"),

(5,"jeevan"),

(6,"santosh"),

(7,"hari");

select \* from customers;

insert into orders values

(1,1,16),

(2,2,13),

(3,3,14),

(4,4,20),

(5,5,30),

(6,6,20),

(7,7,25);

select \* from orders;

select

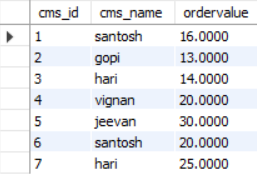
customers.cms\_id, customers.cms\_name, avg(orders.odr\_orders) AS ordervalue

from customers left join orders on

customers.cms\_id = orders.cms\_id group by

customers.cms\_id, customers.cms\_name;

Output:



**Task-11:**

In a database storing movie information,

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

Code:

create database task11;

use task11;

create table movie\_info(

mve\_id int primary key auto\_increment,

mve\_name varchar(50),

mve\_rating int

);

insert into movie\_info(mve\_name,mve\_rating) values

("bahubali",9),

("stranger things",7),

("dark",4),

("agnathavasi",2),

("okkadu",6),

("jalsa",8),

("Pokiri",7),

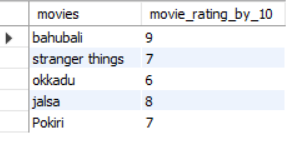
("salar",9);

select movie\_info.mve\_name as movies, movie\_info.mve\_rating as

movie\_rating\_by\_10 from movie\_info where

movie\_info.mve\_rating > 5 limit 5;

Output:



**Task-12:**

You have tables for invoices and payments.

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

Code:

create database task12;

use task12;

create table invoices(

ivs\_id int primary key auto\_increment,

ivs\_name varchar(50)

);

select \* from invoices;

create table payments(

pmt\_id int primary key auto\_increment,

ivs\_id int,

pmt\_status varchar(50),

pmt\_amount decimal(5,2),

foreign key(ivs\_id) references invoices(ivs\_id)

);

insert into invoices(ivs\_name) values

("furnitures"),

("grocessories"),

("houses"),

("rents"),

("cars");

insert into payments values

(1,"paid",100.00),

(2,"not paid",150.00),

(3,"paid",100.00),

(4,"paid",220.00),

(5,"not paid",200.00);

update payments set ivs\_id=5 where pmt\_id=5;

select \* from payments;

select \* from invoices;

select invoices.ivs\_name as invoices,

payments.pmt\_status as payment\_status,payments.pmt\_amount as

total\_amount from invoices join payments on

invoices.ivs\_id = payments.ivs\_id

where payments.pmt\_status = "not paid";

Output:

