**MySql\_Examination**

**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.

ANS : **Create table users (**

**user\_id INT PRIMARY KEY,**

**user\_name VARCHAR(50)**

**);**

**Create table orders (**

**order\_id INT PRIMARY KEY,**

**user\_id INT,**

**FOREIGN KEY (user\_id) REFERENCES users(user\_id)**

**);**

**SELECT u.user\_name, COUNT(o.order\_id) AS total\_orders**

**FROM users u**

**LEFT JOIN orders o ON u.user\_id = o.user\_id**

**GROUP BY u.user\_id, u.user\_name;**

**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.

Ans:

**CREATE TABLE students (**

**student\_id INT PRIMARY KEY,**

**student\_name VARCHAR(100)**

**);**

**CREATE TABLE courses (**

**course\_id INT PRIMARY KEY,**

**course\_name VARCHAR(100)**

**);**

**CREATE TABLE enrollments (**

**student\_id INT,**

**course\_id INT,**

**PRIMARY KEY (student\_id, course\_id),**

**FOREIGN KEY (student\_id) REFERENCES students(student\_id),**

**FOREIGN KEY (course\_id) REFERENCES courses(course\_id)**

**);**

**SELECT s.student\_name, c.course\_name**

**FROM students s**

**JOIN enrollments e ON s.student\_id = e.student\_id**

**JOIN courses c ON e.course\_id = c.course\_id;**

**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.

Ans : **create table products(product\_id int primary key, product\_name varchar(30),price long);**

**insert into products values (1,'mobiles',10000),(2, 'tabs',2000),(3,'laptops',4000),(4,'chargers',200);**

**create table sales**

**(sales\_id int primary key, sales int, product\_id int, foreign key(product\_id)**

**references products(product\_id));**

**insert into sales values(1,2,1),(2, 5, 2),(3,1,3),(4,7,4);**

**create table customers (customer\_id int, customer\_name varchar(30),**

**product\_id int , sales\_id int , foreign key(sales\_id) references sales(sales\_id),**

**foreign key(product\_id) references products(product\_id));**

**insert into customers values(1,'mani',1,1),(2,'durga',2,2),(3,'gopi',3,3),(4,'jeevan',4,4);**

**select product\_name, (price \* sales) from products join sales**

**on products.product\_id = sales.product\_id**

**join customers on customers.product\_id = products.product\_id;**

**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.

Ans : **select employe\_name, manager\_name from company;**

**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.

ANS : **SELECT \* FROM products**

**ORDER BY units DESC**

**LIMIT 10;**

**Task-6:**

You have tables for students, courses, and grades.

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

Ans : **CREATE TABLE student (**

**id INT PRIMARY KEY,**

**name VARCHAR(100)**

**);**

**CREATE TABLE cources (**

**id INT PRIMARY KEY,**

**course VARCHAR(100),**

**student\_id int, foreign key(student\_id) reference student(id)**

**);**

**Create table grades(id int primary key,**

**grade int, student\_id int ,**

**foreign key(student\_id) references student(id);**

**select name , avg(grade) from studen**

**join grades on**

**studen.id = grades.student\_id**

**group by name;**

**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.

**SELECT NAME FROM INSTAGRAM**

**ORDER BY followers desc;**

**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.

Ans .  **create table employ(emp\_id int primary key, employee\_name varchar(30));**

**insert into employ values(1,'vignan'),(2,'jeevan'),(3,'santhosh'),(4,'gopi'),(5,'hari');**

**select \* from employ;**

**create table departments(department\_id int primary key, department\_name varchar(30), id int, foreign key(id)**

**references employ(emp\_id));**

**insert into departments values(1,'civil',1),(2,'mec',2),(3,'cse',3),(4,'cse',1);**

**select department\_name, count(emp\_id) as total\_employees from employ**

**join departments**

**on employ.emp\_id = departments.id**

**group by departments.department\_name;**

**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).

**ANS : Select products from inventory where stock < 10;**

**Task-10:**

You have tables for customers and orders.

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

Ans . **SELECT**

**c.customer\_name,**

**AVG(o.order\_total) AS average\_order\_value**

**FROM**

**customers c**

**JOIN**

**orders o ON c.customer\_id = o.customer\_id**

**GROUP BY**

**c.customer\_name;**

**Task-11:**

In a database storing movie information,

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

Ans : **select \* from movie\_details**

**order by movie\_name**

**limit 3;**

**or**

**select \* from movie\_details**

**limit 3;**

**Task-12:**

You have tables for invoices and payments.

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

**create table invoice(id int primary key, customer\_name varchar(30));**

**insert into invoice values(1,'mani'),(2,'gopi'),(3,'jeevan'),(4,'hari');**

**create table payments(payment\_id int primary key, status varchar(20),id int,**

**foreign key(id)references invoice(id));**

**alter table payments add column amount int;**

**update payments set amount = 5000 where id = 4;**

**insert into payments values(1,'paid',1),(2,'paid',2),(3,'unpaid',3),(4,'upaid',4);**

**select customer\_name, amount from invoice join payments**

**on invoice.id = payments.id**

**where status = 'unpaid';**