MySql Tasks

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. create table users (user_id int primary key auto_increment, name varchar(50) not null); insert into users (name) value ('nishanth'), ('uday'), ('venu'), ('sai teja'), ('rathod'); create table orders (order_id int primary key auto_increment, user_id int, item varchar (50), order_date date, foreign key (user_id) references users(user_id));

```
INSERT INTO orders (user_id, item, order_date) VALUES
(1, 'Laptop', '2025-02-01'),
(1, 'Mouse', '2025-02-05'),
(1, 'Keyboard', '2025-02-10'),
(2, 'zebrononics Monitor', '2025-02-15'),
(3, 'Headset', '2025-02-20'),
(3, 'ipad', '2025-02-25'),
(4, 'phone', '2025-03-01'),
(5, 'hp Printer', '2025-03-04'),
(5, 'hp Scanner', '2025-02-13');
select * from orders
select users.name, count(orders.order_id)
from users
right join orders on users.user_id = orders.user_id
group by users.user_id, users.name;
```

	name	count(orders.order_id)			
•	nishanth	3			
	uday	1			
	venu	2			
	sai teja	1			
	rathod	2			

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.

```
create database college;
use college;
create table students (
student_ID INT PRIMARY KEY auto_increment,
name varchar (30) not null unique
);
create table courses (
course_id int primary key auto_increment,
course_name varchar(60)
);
create table enrollment (
enroll_id int primary key auto_increment,
student_id int,
course_id int,
foreign key (student_id) references students(student_id),
foreign key (course_id) references courses (course_id)
on update cascade
on delete cascade
```

```
insert into students (name)
values
('nishanth'),
('uday'),
('venu'),
('sai teja'),
('rathod');
insert into courses (course_name)
values
('computer science engg'),
('information technology'),
('civil engineering'),
('automobile engineering'),
('cyber security'),
('ai & machine learning'),
('Electrical engineering'),
('machenical engineering');
insert into enrollment (student_id , course_id)
values
```

- (1,1),
- (1,2),
- (5,5),
- (5,6),
- (4,3),
- (2,4),
- (2,7),
- (3,8);

 ${\tt SELECT\ students.name\ ,\ courses.course_name}$

FROM students

JOIN enrollment ON students.student_id = enrollment.student_id

JOIN courses ON enrollment.course_id = courses.course_id

ORDER BY students.name;

	name	course_name		
١	nishanth	computer science engg		
	nishanth	information technology		
	rathod	cyber security		
	rathod	ai & machine learning		
	sai teja	civil engineering		
	uday	automobile engineering		
	uday	Electrical engineering		
	venu	machenical engineering		

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.

```
CREATE TABLE products (
  product_id INT PRIMARY KEY AUTO_INCREMENT,
  product_name VARCHAR(100),
  price DECIMAL(10,2)
);
CREATE TABLE sales (
  sale_id INT PRIMARY KEY AUTO_INCREMENT,
  product_id INT,
  quantity INT,
  FOREIGN KEY (product_id) REFERENCES products(product_id)
);
INSERT INTO products (product_name, price) VALUES
('Laptop', 10000.00),
('Smartphone', 8000.00),
('Headphones', 1000.00),
('gaming Chair', 5000.00),
('Table', 3000.00);
INSERT INTO sales (product_id, quantity) VALUES
```

```
(1, 5),
(2, 10),
(3, 15),
(4, 20),
```

(5, 8);

select * from sales;

SELECT products.product_name, SUM(sales.quantity * products.price) AS total_amount

FROM products

JOIN sales ON products.product_id = sales.product_id

GROUP BY products.product_name;



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.

use assignment;

```
CREATE TABLE departments (

department_id INTEGER PRIMARY KEY,

department_name VARCHAR(30),

location_id int
);
```

INSERT INTO departments

```
VALUES
```

- (1, 'Human resource', 101),
- (2, 'business analyst', 102),
- (3, 'IT', 103),
- (4, 'Marketing team', 104),
- (5, 'Sales team', 105),
- (6, 'Customer Support', 106),
- (7, 'Operations', 107),
- (8, 'Legal team', 108),
- (9, 'Research & Development', 109),
- (10, 'Quality Assurance and testing', 110),
- (11, 'Security', 111),
- (12, 'deployment team', 112),
- (13, 'Product Management', 113),
- (14, 'soc operations', 114),
- (15, 'Training & Development', 115),
- (16, 'Administration', 116);

CREATE TABLE employees

```
( employee_id INTEGER
```

- , first_name VARCHAR(20)
- , last_name VARCHAR(25)
- , email VARCHAR(25)

```
, phone_number VARCHAR(20)
 , hire date DATE
 , job_id VARCHAR(10)
 , salary INTEGER
 , commission_pct INTEGER
 , manager_id INTEGER
 , department_id INTEGER
 , constraint pk_emp primary key (employee_id)
 , constraint fk_deptno foreign key (department_id) references departments(department_id)
 );
 INSERT INTO employees (employee_id, first_name, last_name, email, phone_number, hire_date,
job_id, salary, commission_pct, manager_id, department_id) VALUES
(1, 'Amit', 'Sharma', 'amit.sharma@gmail.com', '9876543210', '2015-03-12', 'HR_MGR', 75000, 5, NULL,
1),
(2, 'nishanth', 'Varma', 'nishanth.varma@gmail.com', '9876543211', '2017-06-25', 'FIN ANAL', 65000, 3,
1, 2),
(3, 'Rajesh', 'Gupta', 'rajesh.gupta@gmail.com', '9876543212', '2019-09-15', 'IT_ENG', 80000, 7, 2, 3),
(4, 'Sonal', 'Patel', 'sonal.patel@gmail.com', '9876543213', '2021-01-10', 'MKT_EXEC', 55000, 4, 3, 4),
(5, 'Vikram', 'Singh', 'vikram.singh@gmail.com', '9876543214', '2018-07-19', 'SALES MGR', 72000, 6, 4,
5),
(6, 'Neha', 'Chopra', 'neha.chopra@gmail.com', '9876543215', '2020-05-30', 'CUST_SUPP', 50000, 2, 5, 6),
(7, 'Ravi', 'Kumar', 'ravi.kumar@gmail.com', '9876543216', '2016-11-23', 'OPS MGR', 78000, 8, 6, 7),
(8, 'Anjali', 'Mishra', 'anjali.mishra@gmail.com', '9876543217', '2019-08-14', 'LEGAL_ADV', 90000, 5,
NULL, 8),
(9, 'Deepak', 'Yadav', 'deepak.yadav@gmail.com', '9876543218', '2022-02-18', 'RND SCI', 85000, 6, 7, 9),
(10, 'Sneha', 'Joshi', 'sneha.joshi@gmail.com', '9876543219', '2023-04-07', 'QA_ENG', 60000, 3, 8, 10),
```

```
(11, 'Pankaj', 'Bansal', 'pankaj.bansal@gmail.com', '9876543220', '2021-06-21', 'SEC_OFFI', 65000, 2, 9, 11),
```

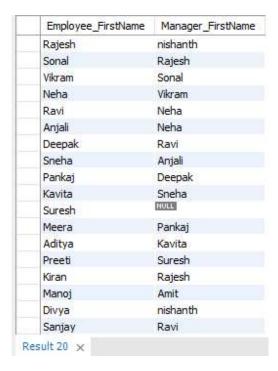
- (12, 'Kavita', 'Rao', 'kavita.rao@gmail.com', '9876543221', '2017-09-12', 'PROC_MGR', 70000, 4, 10, 12),
- (13, 'Suresh', 'Nair', 'suresh.nair@gmail.com', '9876543222', '2020-10-05', 'PROD_MGR', 80000, 6, NULL, 13),
- (14, 'Meera', 'lyer', 'meera.iyer@gmail.com', '9876543223', '2015-12-31', 'PR_EXEC', 55000, 4, 11, 14),
- (15, 'Aditya', 'Goyal', 'aditya.goyal@gmail.com', '9876543224', '2018-03-17', 'TRAIN_DEV', 67000, 5, 12, 15),
- (16, 'Preeti', 'Das', 'preeti.das@gmail.com', '9876543225', '2019-07-25', 'ADMIN', 60000, 3, 13, 16),
- (17, 'Kiran', 'Jadhav', 'kiran.jadhav@gmail.com', '9876543226', '2022-01-11', 'IT_SUPPORT', 58000, 2, 3, 3),
- (18, 'Manoj', 'Deshmukh', 'manoj.deshmukh@gmail.com', '9876543227', '2016-04-30', 'HR_EXEC', 62000, 3, 1, 1),
- (19, 'Divya', 'Kapoor', 'divya.kapoor@gmail.com', '9876543228', '2017-12-20', 'FIN_OFF', 69000, 4, 2, 2),
- (20, 'Sanjay', 'Reddy', 'sanjay.reddy@gmail.com', '9876543229', '2023-05-01', 'DATA_ANAL', 71000, 6, 7, 3);

SELECT

```
e.first_name AS Employee_FirstName,
m.first_name AS Manager_FirstName
```

FROM employees e

LEFT JOIN employees m ON e.manager_id = m.employee_id;



Task-5:

You are managing a database for an online store.

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

SELECT

products.product_name,SUM(sales.quantity) as units_sold

FROM products

JOIN sales ON products.product_id = sales.product_id

GROUP BY products.product_name

ORDER BY units_sold DESC

LIMIT 5;

	product_name	units_sold
•	gaming Chair	20
	Headphones	15
	Smartphone	10
	Table	8
	Laptop	5

Task-6:

You have tables for students, courses, and grades.

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

```
CREATE TABLE student_grades (

student_id INT PRIMARY KEY AUTO_INCREMENT,

student_name VARCHAR(50),

course_name VARCHAR(50),

grade DECIMAL(5,2)
);
```

INSERT INTO student_grades (student_name, course_name, grade)

VALUES

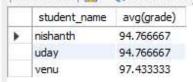
```
('nishanth', 'maths', 92.6),
('nishanth', 'science', 97.9),
('nishanth', 'social', 93.8),
('uday', 'maths', 97.6),
('uday', 'science', 94.9),
('uday', 'social', 91.8),
('venu', 'maths', 98.6),
```

```
('venu', 'science', 94.9),

('venu', 'social', 98.8);

select * from student_grades;

select student_name , avg(grade) from student_grades
group by student_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.

create table facebook_friends(

facebook_id int primary key auto_increment,

facebook_name varchar (50) unique not null,

no_of_friends int

);

 $INSERT\ INTO\ facebook_friends\ (facebook_name,\ no_of_friends)\ VALUES$

('wonder girl', 1000),

('virat kohli', 15000),

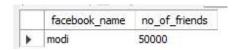
('dhoni', 20000),

('mom little girl', 900),

('modi', 50000),

```
('KCR', 18000),
('chandrababu naidu', 17000),
('yogi ji', 45000),
('rohit sharma', 25000),
('ronaldo', 30000);
```

select facebook_name,no_of_friends from facebook_friends order by no_of_friends desc limit 1;



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.

SELECT

departments.department_name,

COUNT(employees.employee_id)

FROM departments

JOIN employees ON departments.department_id = employees.department_id

GROUP BY departments.department_name;

	department_name	COUNT(employees.employee_id)		
•	Human resource	2		
	business analyst	2		
	П	3		
	Marketing team	1		
	Sales team	1		
	Customer Support	1		
	Operations	1		
	Legal team	1		
	Research & Development	1		
	Quality Assurance and testing	1		
	Security	1		
	deployment team	1		
	Product Management	1		
	soc operations	1		
	Training & Development	1		
	Administration	1		

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).

```
CREATE TABLE categories (

Category_ID INT AUTO_INCREMENT PRIMARY KEY,

Category_Name VARCHAR(70)
);
```

INSERT INTO categories (Category_Name)

VALUES

('drinks'),

('chips'),

('tea'),

('Spreads'),

('milk Products'),

```
('Grains'),
('Coffee'),
('Chocolates');
select * from categories;
CREATE TABLE suppliers (
  Supplier_ID INT AUTO_INCREMENT PRIMARY KEY,
  Supplier_Name VARCHAR(50),
  City VARCHAR(100)
);
INSERT INTO suppliers (Supplier_Name, City) VALUES
('laxmi Distributors', 'nizamabd'),
('nishanth Imports Ltd.', 'hyderabad'),
('dhanalaxmi Supplies', 'waranagl'),
('uday Wholesale Traders', 'karimnagar'),
('venu Beverages', 'nalgonda'),
('sai teja Delights', 'hyderabad'),
('bhavesh Exports', 'vizag'),
('tanvi spices', 'guntur');
select * from suppliers
```

```
CREATE TABLE products (
  ProductID INT AUTO_INCREMENT,
  ProductName VARCHAR(50) NOT NULL,
  SupplierID INT,
  CategoryID INT,
  QuantityPerUnit VARCHAR(255),
  UnitPrice DECIMAL(10,2),
  UnitsInStock INT,
  UnitsOnOrder INT,
  ReorderLevel INT,
  Discontinued BOOLEAN,
  PRIMARY KEY (ProductID),
  UNIQUE (ProductName),
  FOREIGN KEY (CategoryID) REFERENCES categories(Category_ID),
  FOREIGN KEY (SupplierID) REFERENCES suppliers(Supplier_ID)
  on update cascade
  on delete cascade
);
INSERT INTO products (ProductName, SupplierID, CategoryID, QuantityPerUnit, UnitPrice, UnitsInStock,
UnitsOnOrder, ReorderLevel, Discontinued)
VALUES
('mazza,appy_fizz,zeera_soda', 1, 1, '1 Litre Bottle', 50.00, 100, 10, 20, 0),
('lays,kurkure,unclechips', 2, 2, '200g Pack', 30.00, 200, 20, 50, 0),
('dabur_Tea,redlabel', 4, 3, '250g Pack', 150.00, 50, 5, 15, 0),
('ghee,Butter', 5, 4, '500g pack', 380.00, 80, 8, 15, 0),
```

('Milk,curd', 6, 5, '1 Litre Pack', 75.00, 0, 0, 5, 1), -- Discontinued due to no orders

('Wheatfloor, maizefloor', 3, 6, '5kg Bag', 35.00, 500, 50, 100, 0),

('bru_Coffee', 4, 7, '1kg Pack', 700.00, 20, 2, 5, 0),

('dairy_milk_Chocolate', 7, 8, '200g Bar', 80.00, 150, 15, 30, 0);

	ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	UnitsInStock	UnitsOnOrder	ReorderLevel	Discontinued
>	5	Milk,curd	6	5	1 Litre Pack	75.00	0	0	5	1
	8	dairy_milk_Chocolate	7	8	200g Bar	80.00	9	15	30	0
	NULL	NULL	NULL	NULL	RULE	NUUT	NULL	EUR	HULL	RULE

select * from products where UnitsInStock <= 10;</pre>

Task-10:

You have tables for customers and orders.

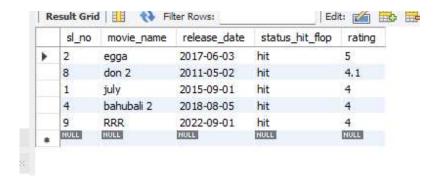
Write a query to show the average order value for each customer.

```
CREATE TABLE customer_orders (
    customer_id INT AUTO_INCREMENT PRIMARY KEY,
    customer_name VARCHAR(100) NOT NULL,
    order_total DECIMAL(10,2)
);
INSERT INTO customer_orders (customer_name, order_total) VALUES
('nishanth', 2000.00),
('uday', 1000.00),
('venu', 1500.00),
('sai teja', 900.00),
('nishanth', 1200.00),
```

```
('sai teja', 800.00),
('uday', 850.00);
SELECT customer_name, avg(order_total) from customer_orders group by customer_name;
Task-11:
In a database storing movie information,
Write a query to show the top 5 highest-rated movies by users.
create table movies(
  sl_no int primary key,
  movie_name varchar(50),
  release_date date,
  status_hit_flop varchar (50)
  );
        insert into movies
  values
  (1, 'july','2015-09-01','hit'),
  (2, 'egga','2017-06-03','hit'),
  (3, 'bahubali 1','2015-07-03','hit'),
  (4, 'bahubali 2','2018-08-05','hit'),
  (5, 'don','2005-04-06','flop'),
  (6, 'transformers','2019-09-01','hit'),
  (7, 'falaknama das','2021-02-02','flop'),
```

```
(8, 'don 2','2011-05-02','hit'),
(9, 'RRR','2022-09-01','hit'),
(10, 'shiva', '2023-09-01', 'flop');
select * from movies;
alter table movies add column rating float;
update movies set rating= 4 where sl_no = 1;
     update movies set rating= 5 where sl_no = 2;
update movies set rating= 3.5 where sl_no = 3;
update movies set rating= 4 where sl_no = 4;
update movies set rating= 1.5 where sl_no = 5;
update movies set rating= 3.9 where sl_no = 6;
update movies set rating= 1.5 where sl_no = 7;
update movies set rating= 4.1 where sl_no = 8;
update movies set rating= 4 where sl_no = 9;
update movies set rating= 1.2 where sl no = 10;
```

select * from movies order by rating desc limit 5;



Task-12:

You have tables for invoices and payments.

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

```
invoice_id INT PRIMARY KEY AUTO_INCREMENT,

customer_name VARCHAR(100) NOT NULL,

invoice_amount DECIMAL(10,2),

amount_paid DECIMAL(10,2) DEFAULT 0

);
```

INSERT INTO invoices (customer_name, invoice_amount, amount_paid) VALUES ('nishanth', 5000.00, 2000.00), ('kiran', 3000.00, 1500.00), ('uday', 7000.00, 7000.00), ('sai teja', 2500.00, 0.00), ('venu', 4500.00, 1000.00);

select invoice_id,customer_name ,invoice_amount-amount_paid as due_amount from invoices

where invoice_amount > amount_paid;

	invoice_id	customer_name	due_amount
>	1	nishanth	3000.00
	2	kiran	1500.00
	4	sai teja	2500.00
	5	venu	3500.00