DBMS LAB – III

CONCEPTS : WHERE, GROUP BY, ORDER BY, AGGREGATE FUNCTIONS

1. CASE STUDY – I

Table schema : products

Column Name	Data Type
product_id	integer
name	varchar
price	decimal
category	varchar
rating	integer

products table:

```
product_id | name             | price  | category    | rating
----------------------------------------------------------------
1          | Apple iPhone 12   | 799.00 | Electronics | 4
2          | Samsung Galaxy S21| 899.00 | Electronics | 5
3          | Sony 55" TV       | 1299.00| Electronics | 4
4          | Sony Playstation 5| 499.00 | Electronics | 5
5          | Dell XPS 13       | 1199.00| Computers   | 4
6          | MacBook Pro 13    | 1299.00| Computers   | 5
7          | Nike Air Zoom     | 129.00 | Footwear    | 4
8          | Adidas Ultraboost | 149.00 | Footwear    | 5
9          | Calvin Klein T-Shirt| 29.99 | Clothing   | 3
10          | Levi's Jeans      | 59.99  | Clothing   | 4
```

1. Retrieve the names and prices of all products with a rating of 5, ordered by price from lowest to highest.

2. Find the average price of all products in each category, ordered by category name alphabetically.

3. Retrieve the names and categories of all products with a price greater than 1000, ordered by category alphabetically.

4. Find the total number of products in each category with a rating greater than or equal to 4.

5. Retrieve the names and ratings of all products with names containing the word 'red', ordered by rating from highest to lowest.

2. CASE STUDY - 2

Table schema : orders

Column Name	Data Type
order_id	integer
customer_id	integer
order_date	date
total_price	decimal

orders table:

```
order_id | customer_id | order_date  | total_price
--------------------------------------------------
1        | 101         | 2022-03-01 | 25.00
2        | 102         | 2021-12-05 | 50.00
3        | 103         | 2022-02-14 | 100.00
4        | 104         | 2021-11-28 | 75.00
5        | 101         | 2022-01-10 | 40.00
6        | 105         | 2021-10-15 | 80.00
7        | 106         | 2022-04-02 | 60.00
8        | 102         | 2022-02-21 | 35.00
9        | 101         | 2022-03-15 | 20.00
10       | 105         | 2021-12-10 | 90.00
```

1. Find the total price of all orders placed on or after January 1st, 2022.

2. Retrieve the customer IDs and the total number of orders each customer has placed, ordered by the number of orders from highest to lowest.

3. Find the total price of all orders placed by customer with ID 123.

4. Retrieve the order IDs and total prices of all orders placed in 2021, ordered by total price from highest to lowest.

5. Find the number of orders placed on each day, ordered by date from oldest to newest.

3. CASE STUDY -3

Table Schema: customers

Column Name	**Data Type**
customer_id	integer
name	varchar
email	varchar

customers table:

```
customer_id | name         | email
--------------------------------------
1           | John Smith   | john.smith@gmail.com
2           | Jane Doe     | jane.doe@yahoo.com
3           | James Brown  | james.brown@outlook.com
4           | Jennifer Lee | jennifer.lee@stanford.edu
5           | Jason Chen   | jason.chen@berkeley.edu
6           | Jessica Kim  | jessica.kim@gmail.com
7           | Jack Lee     | jack.lee@hotmail.com
8           | Joyce Chen   | joyce.chen@nyu.edu
9           | Jasmine Liu  | jasmine.liu@mit.edu
10          | Justin Wong  | justin.wong@gmail.com
```

1. Retrieve the names and email addresses of customers who have "gmail.com" domain email addresses.

2. Retrieve the customer IDs and the number of customers with the same name, ordered by the number of customers from highest to lowest.

3. Find the names and email addresses of all customers whose email addresses end with ".edu".

4. Retrieve the customer IDs and names of all customers whose names start with the letter "J", ordered by name from A to Z.

5. Find the number of customers who have email addresses from each domain, ordered by the number of customers from highest to lowest.

----

4. CASE STUDY - 4

Table Schema: order_items

Column Name	Data Type
order_id	integer
product_id	integer
quantity	integer

order_items table:

```
order_id | product_id | quantity
---------------------------------
1        | 1          | 2
1        | 3          | 1
2        | 2          | 1
2        | 4          | 1
2        | 6          | 2
3        | 5          | 1
3        | 7          | 1
3        | 10          | 3
4        | 8          | 2
4        | 9          | 2
```

1. Retrieve the total number of products ordered for a specific product_id.

2. Retrieve the order_id and total number of items ordered for each order, sorted by the order_id in ascending order.

3. Find the product_id of the product(s) that has/have been ordered the most across all orders.

4. Retrieve the order_id and product_id for each order where the quantity ordered is greater than 1.

5. Retrieve the total quantity of each product ordered across all orders, sorted in descending order of total quantity.

----

5. CASE STUDY -5

Table Schema: employees

Column Name	Data Type
employee_id	integer
name	varchar
hire_date	date
salary	decimal

employees table:

```
employee_id | name        | hire_date  | salary
-----------------------------------------------
1           | John Smith  | 2018-06-01 | 55000.00
2           | Jane Doe    | 2019-01-15 | 65000.00
3           | Bob Johnson | 2020-05-20 | 75000.00
4           | Mary Lee    | 2021-02-28 | 80000.00
5           | Mike Smith  | 2022-01-01 | 90000.00
6           | Sarah Chen  | 2022-04-15 | 70000.00
```

1. Retrieve the names of all employees who were hired after a specific date.

2. Retrieve the average salary for all employees who were hired in a specific year.

3. Retrieve the total number of employees hired in each year, sorted in ascending order by year.

4. Retrieve the employee(s) with the highest salary, along with their salary.

5. Retrieve the total salary paid to employees for each year, sorted in descending order of total salary.