Retail Company SQL Project (With Questions & Solutions)

```
Q1. Create Database retail_company
CREATE DATABASE retail_company;
USE retail_company;

Q2. Create customers table with constraints
CREATE TABLE customers(
customer_id INT PRIMARY KEY,
customer_name VARCHAR(15) NOT NULL,
email VARCHAR(15) NOT NULL UNIQUE,
city VARCHAR(10) NOT NULL,
created_at DATE DEFAULT CURRENT_TIMESTAMP
);

Q3. Create orders table with constraints
CREATE TABLE orders (
order_id INT PRIMARY KEY,
customer_id INT NOT NULL,
order_date DATE NOT NULL,
total_amount DECIMAL(10,2) NOT NULL CHECK (total_amount > 0),
status ENUM('Pending','Shipped','Delivered','Cancelled'),
FOREIGN KEY (customer_id) REFERENCES customers(customer_id)
);

Q4. Insert Customers
INSERT INTO customers VALUES
(101,'Omkar Kulkarni','omk@gmail.com','kolhapur',DEFAULT),
(102,'Manoj Kulkarni','maonj@gmail.com','Kerur','2026-01-31'),
(103,'ninad Kulkarni','ninad@gmail.com','pune','2025-12-30'),
(104,'Pranav Patil','pranav@gmail.com','Mumbai','2025-10-10'),
(105,'Sahil Fepade','sahil@gmail.com','ratnagiri','2025-08-24');

Q5. Insert Orders
INSERT INTO orders VALUES
(1,101,'2026-02-14',7777,'Shipped'),
(2,103,'2026-01-23',312324,'Delivered'),
(3,104,'2026-02-14',325412,'Pending'),
(4,105,'2025-12-23',5453132,'Cancelled'),
(5,102,'2026-02-10',5452454,'Shipped'),
(6,101,'2026-02-14',17777,'Pending');

Q6. Show total order amount per customer
SELECT c.customer_id, c.customer_name,
SUM(o.total_amount) AS total_order_amount
FROM customers c
JOIN orders o ON c.customer_id = o.customer_id
GROUP BY c.customer_name;

Q7. Show number of orders per customer
SELECT c.customer_name,
COUNT(o.order_id) AS total_orders
FROM customers c
JOIN orders o ON c.customer_id = o.customer_id
GROUP BY c.customer_name;

Q8. Show customers whose total order amount > 10000
SELECT c.customer_name,
SUM(o.total_amount) AS total_amt
FROM customers c
JOIN orders o ON c.customer_id = o.customer_id
GROUP BY c.customer_name
HAVING SUM(o.total_amount) > 10000;
```

Q9. Show highest order amount

```
SELECT customer_id, total_amount
FROM orders
WHERE total_amount = (SELECT MAX(total_amount) FROM orders);
```

Q10. Show second highest order amount

```
SELECT customer_id, total_amount
FROM orders
WHERE total_amount = (
SELECT MAX(total_amount)
FROM orders
WHERE total_amount < (SELECT MAX(total_amount) FROM orders)
);
```

Q11. Show customers whose total order amount is above average

```
SELECT customer_id,
SUM(total_amount) AS total_amt
FROM orders
GROUP BY customer_id
HAVING SUM(total_amount) > (
SELECT AVG(customer_total)
FROM (
SELECT SUM(total_amount) AS customer_total
FROM orders
GROUP BY customer_id
) AS temp
);
```