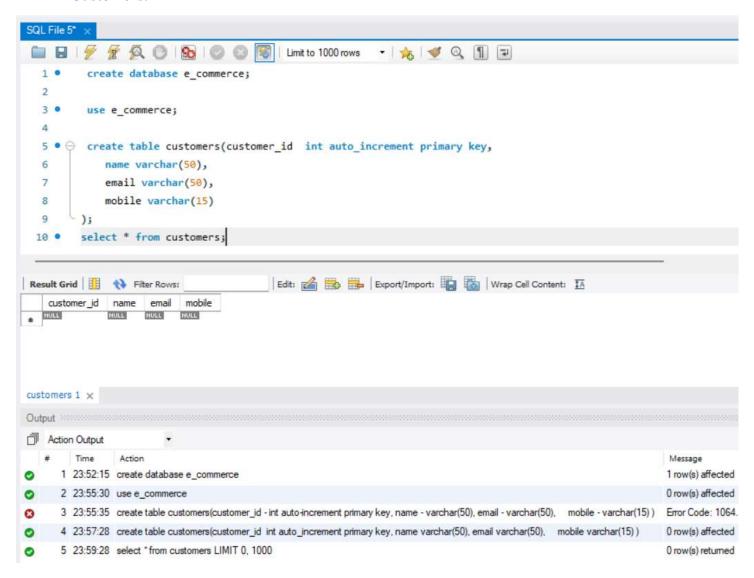
# **SQL Assignment-1**

- 1. Create Database e commerce
- 2. Create following Tables:

#### **Customers:**



#### **Products:**

```
12 • @ create table products(id int,
              name varchar(50) not null,
  13
              description varchar(200),
  14
              price decimal(10, 2) not null,
  15
              category varchar(50)
  16
  17
           select * from products;
  18 •
                                                   Export: Wrap Cell Content: IA
Result Grid
                Filter Rows:
    id
                   description
                               price
                                       category
products 2 ×
Output
Action Output
          Time
                   Action
       1 23:52:15 create database e_commerce
       2 23:55:30 use e_commerce
       3 23:55:35 create table customers(customer_id - int auto-increment primary key, name - varchar(50), email - varchar(50), mobile - varchar(15)
0
       4 23:57:28 create table customers(customer_id_int_auto_increment primary key, name varchar(50), email varchar(50), mobile varchar(15))
0
       5 23:59:28 select *from customers LIMIT 0, 1000
       6 00:04:59 create table products@d int, name varchar(50) not null, description varchar(200), price decimal(10, 2) not null, category varch.
       7 00:05:18 select *from products LIMIT 0, 1000
```

### 3. Modify Tables(using Alter keyword):

a.Add not null on name and email in the Customers table

```
20 ALTER TABLE customers
21 MODIFY COLUMN name VARCHAR(50) NOT NULL,
22 MODIFY COLUMN email VARCHAR(50) NOT NULL;

Output

* Time Action

1 16:47:11 ALTER TABLE customers MODIFY COLUMN name VARCHAR(50) NOT NULL, MODIFY COLUMN email VARCHAR(50)
```

```
25
        -- b. Add UNIQUE constraint on email in Customers table
        ALTER TABLE Customers
        ADD CONSTRAINT unique email UNIQUE (email);
27
28
        -- c. Add column age in Customers table
29
30 •
        ALTER TABLE Customers
        ADD COLUMN age INT;
31
32
        -- d. Change column name from id to product id in Products table
33
        ALTER TABLE Products
        CHANGE COLUMN id product id INT;
35
36
        -- e. Add primary key and auto increment on product id in Products table
37
        ALTER TABLE Products
39
        MODIFY COLUMN product_id INT AUTO_INCREMENT PRIMARY KEY;
40
        -- f. Change datatype of description from VARCHAR to TEXT in Products table
41
        ALTER TABLE Products
42 0
        MODIFY COLUMN description TEXT;
43
Action Output
               Action
       Time
     1 16:47:11 ALTER TABLE customers MODIFY COLUMN name VARCHAR(50) NOT NULL, MODIFY COLUMN email VARCHAR(50) NC
     2 16:49:48 ALTER TABLE Customers ADD CONSTRAINT unique email UNIQUE (email)
    3 16:49:48 ALTER TABLE Customers ADD COLUMN age INT
    4 16:49:48 ALTER TABLE Products CHANGE COLUMN id product_id INT
     5 16:49:48 ALTER TABLE Products MODIFY COLUMN product_id INT AUTO_INCREMENT PRIMARY KEY
     6 16:49:48 ALTER TABLE Products MODIFY COLUMN description TEXT
```

## After modification tables:





#### 4. Create table Order:

```
48 • ⊖ CREATE TABLE Orders (
 49
           order id INT AUTO INCREMENT PRIMARY KEY,
           customer id INT,
 50
           product id INT,
 51
           quantity INT NOT NULL,
 52
           order date DATE NOT NULL,
 53
 54
           status ENUM('Pending', 'Success', 'Cancel'),
           payment method ENUM('Credit', 'Debit', 'UPI'),
 55
           total_amount DECIMAL(10, 2) NOT NULL,
 56
 57
           FOREIGN KEY (customer_id) REFERENCES Customers(customer_id)
 58
      );
Output
Action Output
       Time
    1 17:15:12 CREATE TABLE Orders ( order_id INT AUTO_INCREMENT PRIMARY KEY, cu
```

### 5. Modify Orders Table(using Alter keyword):

```
-- a. Rename table Order to Orders
         ALTER TABLE Orders
 61
         RENAME TO Orders;
 62
         -- b. Set default value 'Pending' in status column
 63
 64 •
         ALTER TABLE Orders
         MODIFY COLUMN status ENUM('Pending', 'Success', 'Cancel') DEFAULT 'Pending';
 65
 66
         -- c. Modify payment method ENUM to add 'COD'
 67
        ALTER TABLE Orders
 68 •
         MODIFY COLUMN payment_method ENUM('Credit', 'Debit', 'UPI', 'COD');
 69
 70
 71
         -- d. Make product_id a foreign key
        ALTER TABLE Orders
 72 •
         ADD CONSTRAINT fk_product FOREIGN KEY (product_id) REFERENCES Products(product_id);
 73
 74
Output
Action Output
       Time
                Action
    1 17:17:47 ALTER TABLE Orders RENAME TO Orders

    2 17:17:47 ALTER TABLE Orders MODIFY COLUMN status ENUM("Pending", "Success", "Cancel") DEFAULT "Pending"

      3 17:17:47 ALTER TABLE Orders MODIFY COLUMN payment_method ENUM('Credit', 'Debit', 'UPI', 'COD')
     4 17:17:47 ALTER TABLE Orders ADD CONSTRAINT fk_product FOREIGN KEY (product_id) REFERENCES Products(product_id)
```

### 6. Insert 20 sample records in all the tables.

#### **Customers:**

```
INSERT INTO Customers (name, email, mobile, age) VALUES
 76
         ('John ', 'john@example.com', '98765', 30),
         ('Alice', 'alice@example.com', '99887', 28),
 77
         ('Bob', 'bob@example.com', '98712', 35),
 78
         ('Emma ', 'emma@example.com', '91234', 26),
 79
         ('Chris', 'chris@example.com', '87654', 32),
 80
 81
         ('Davis', 'olivia@example.com', '90987', 27),
         ('Michael', 'michael@example.com', '92123', 40),
 82
 83
         ('Sophia', 'sophia@example.com', '93456', 24),
         ('Daniel', 'daniel@example.com', '90126', 29),
 85
         ('Emily', 'emily@example.com', '91239', 31),
         ('David', 'david@example.com', '9456123789', 33),
 86
         ('Ella ', 'ella@example.com', '9654321098', 23),
 87
         ('Henry ', 'henry@example.com', '9784561230', 38),
 88
         ('Tsahella '. 'isahella@examnle.com'. '9876541230'. 25).
 89
Output
Action Output
      1 17:23:28 INSERT INTO Customers (name, email, mobile, age) VALUES ('John ', 'john'
```

customer_id name		email	mobile	age	
1	John	john@example.com	98765	30	
2	Alice	alice@example.com	99887	28	
3	Bob	bob@example.com	98712	35	
4	Emma	emma@example.com	91234	26	
5	Chris	chris@example.com	87654	32	
6	Davis	olivia@example.com	90987	27	
7	Michael	michael@example.com	92123	40	
8	Sophia	sophia@example.com	93456	24	
9	Daniel	daniel@example.com	90126	29	
10	Emily	emily@example.com	91239	31	
11	David	david@example.com	9456	33	
12	Ella	ella@example.com	9654	23	
13	Henry	henry@example.com	9784	38	
14	Isabella	isabella@example.com	9876	25	
15	James	james@example.com	9123	37	
16	Grace	grace@example.com 9998.		29	
17	Lucas	lucas@example.com 976		34	
18	Young	charlotte@example.com	9018	27	
19	King	mason@example.com	9554	30	
20	Green	amelia@example.com	9654	22	
NULL	NULL	NULL	HULL	NULL	

### **Products:**

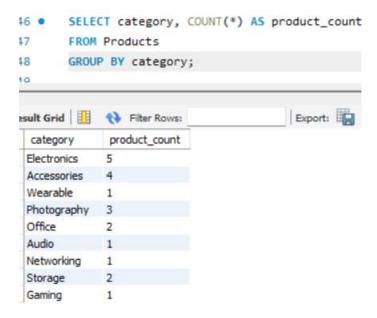


### **Orders:**

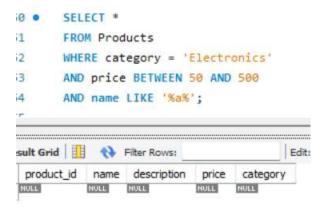
order_id	customer_id	product_id	quantity	order_date	status	payment_method	total_amount
21	1	3	2	2024-01-10	Success	Credit	16000.00
22	2	1	1	2024-01-15	Pending	Debit	75000.00
23	3	5	1	2024-02-05	Cancel	UPI	30000.00
24	4	2	1	2024-02-10	Success	COD	45000.00
25	5	7	3	2024-02-20	Success	Credit	21000.00
26	6	10	1	2024-03-01	Pending	Debit	25000.00
27	7	12	2	2024-03-15	Cancel	UPI	30000.00
28	8	8	1	2024-03-20	Success	COD	2000.00
29	9	6	1	2024-04-01	Pending	Credit	50000.00
30	10	4	1	2024-04-10	Success	Debit	12000.00
31	11	9	2	2024-04-15	Success	UPI	10000.00
32	12	14	1	2024-05-01	Cancel	COD	18000.00
33	13	16	1	2024-05-10	Pending	Credit	70000.00
34	14	11	1	2024-05-15	Success	Debit	85000.00
35	15	13	1	2024-06-01	Pending	UPI	6000.00
36	16	15	2	2024-06-10	Success	COD	8000.00
37	17	18	1	2024-06-20	Cancel	Credit	10000.00
38	18	19	1	2024-07-01	Success	Debit	20000.00
39	19	17	1	2024-07-05	Pending	UPI	3500.00
40	20 NULL	20 NGE	1	2024-07-10	Success	COD	12000.00

## 7. Perform following queries:

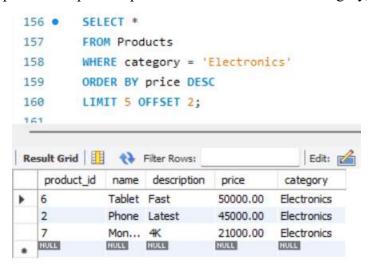
a. Count the number of products as product count in each category.



b. Retrieve all products that belong to the 'Electronics' category, have a price between \$50 and \$500, and whose name contains the letter 'a'.



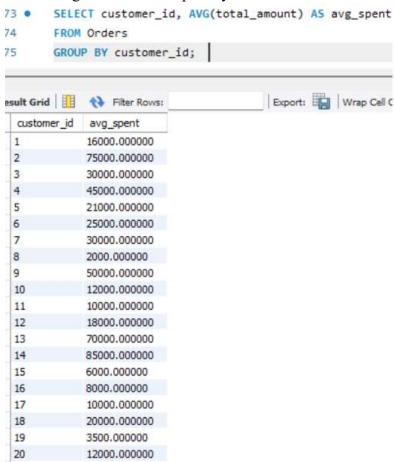
c. Get the top 5 most expensive products in the 'Electronics' category, skipping the first 2.



d.Retrieve customers who have not placed any orders.



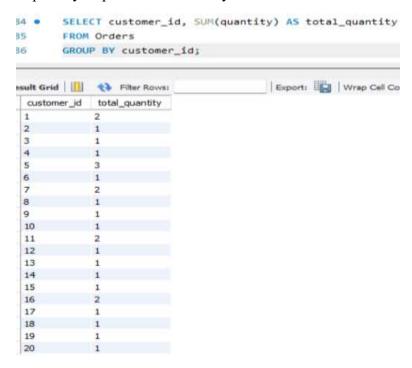
e. Find the average total amount spent by each customer.



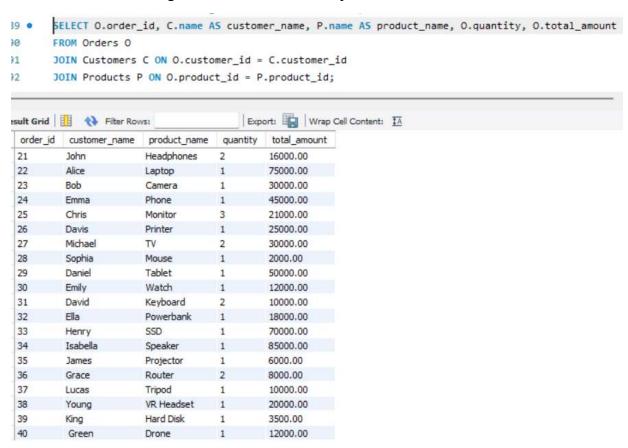
f. Get the products that have a price less than the average price of all products.



g. Calculate the total quantity of products ordered by each customer:



h. List all orders along with customer name and product name.



i. Find products that have never been ordered.

