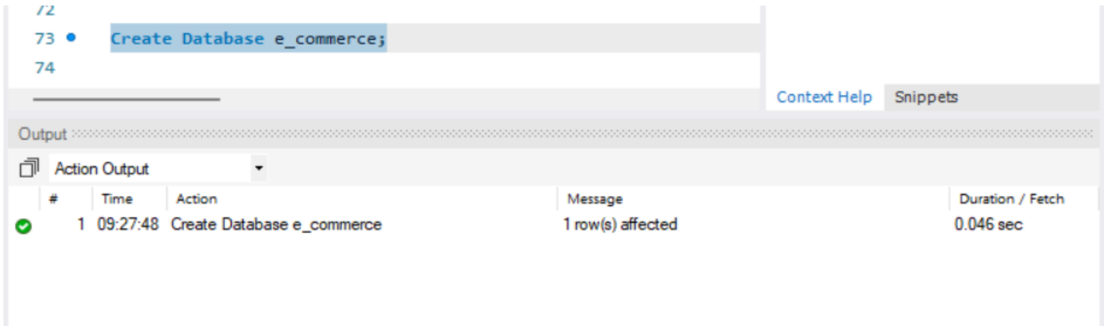
SQL Assignment

1.Create Database e_commerce

Query: Create Database e_commerce;



```
Create Database e_commerce;
```

Context Help Snippets

```
Output
Action Output
   #  Time      Action                       Message            Duration / Fetch
   1  09:27:48  Create Database e_commerce   1 row(s) affected  0.046 sec
```

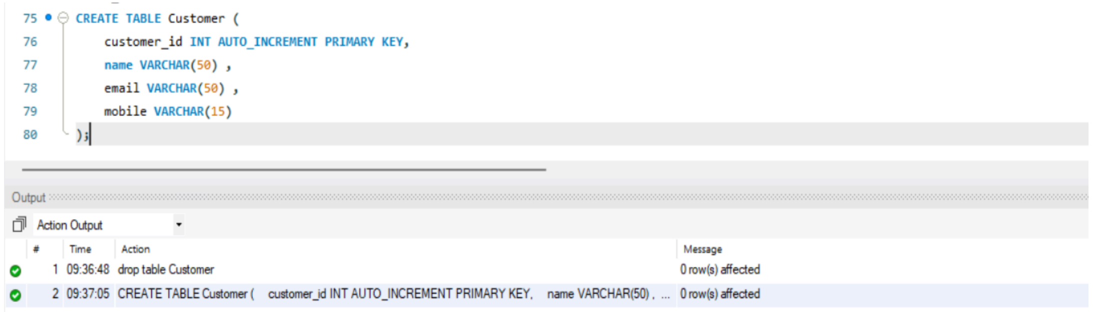
2.Create following Tables:

Customers:

a. customer_id - int auto-increment primary key
b. name - varchar(50)
c. email - varchar(50)
d. mobile - varchar(15)

Query: use e_commerce;

```
CREATE TABLE Customer (
customer_id INT AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(50) ,
email VARCHAR(50) ,
mobile VARCHAR(15)
);
```



```
CREATE TABLE Customer (
    customer_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(50) ,
    email VARCHAR(50) ,
    mobile VARCHAR(15)
);
```

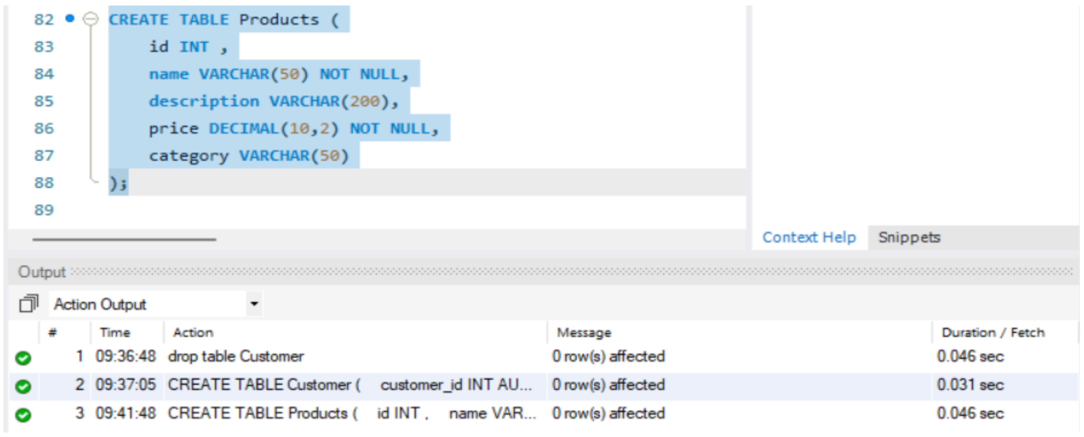
```
Output
Action Output
   #  Time      Action                                                                          Message
   1  09:36:48  drop table Customer                                                             0 row(s) affected
   2  09:37:05  CREATE TABLE Customer (   customer_id INT AUTO_INCREMENT PRIMARY KEY,   name VARCHAR(50) , ...  0 row(s) affected
```

Products:

a. id - int
b. name - varchar(50) not null
c. description - varchar(200)
d. price - decimal(10, 2) not null

e. category - varchar(50)

Query: CREATE TABLE Products (
    id INT ,
    name VARCHAR(50) NOT NULL,
    description VARCHAR(200),
    price DECIMAL(10,2) NOT NULL,
    category VARCHAR(50)
    );



```
CREATE TABLE Products (
    id INT ,
    name VARCHAR(50) NOT NULL,
    description VARCHAR(200),
    price DECIMAL(10,2) NOT NULL,
    category VARCHAR(50)
);
```

Context Help Snippets

Output

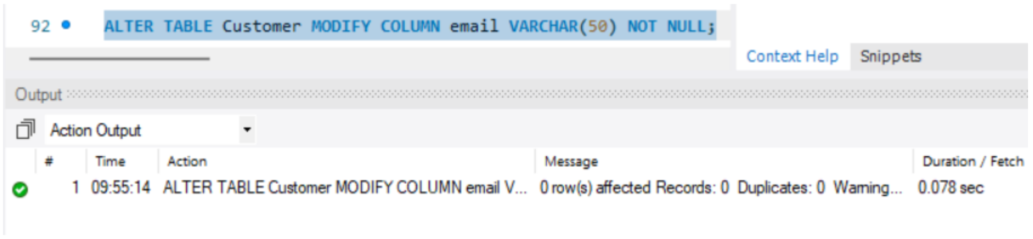
Action Output

```
#  Time      Action                                                Message             Duration / Fetch
1  09:36:48  drop table Customer                                   0 row(s) affected   0.046 sec
2  09:37:05  CREATE TABLE Customer (  customer_id INT AU...        0 row(s) affected   0.031 sec
3  09:41:48  CREATE TABLE Products (  id INT ,   name VAR...       0 row(s) affected   0.046 sec
```

3.Modify Tables(using Alter keyword):

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

Query : ALTER TABLE Customer MODIFY COLUMN name VARCHAR(50) NOT NULL;



```
ALTER TABLE Customer MODIFY COLUMN email VARCHAR(50) NOT NULL;
```

Context Help Snippets

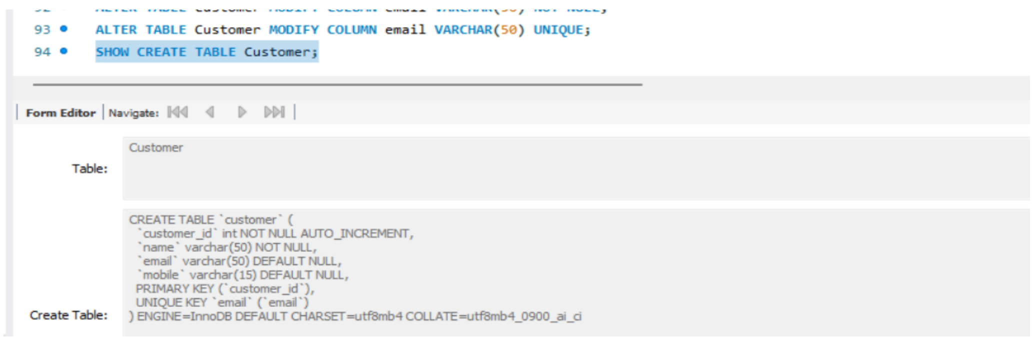
Output

Action Output

```
#  Time      Action                                         Message                                                      Duration / Fetch
1  09:55:14  ALTER TABLE Customer MODIFY COLUMN email V...  0 row(s) affected Records: 0  Duplicates: 0  Warning...      0.078 sec
```

b. Add unique key on email in the Customers table

Query: ALTER TABLE Customer MODIFY COLUMN email VARCHAR(50) UNIQUE;



```
ALTER TABLE Customer MODIFY COLUMN email VARCHAR(50) UNIQUE;
SHOW CREATE TABLE Customer;
```

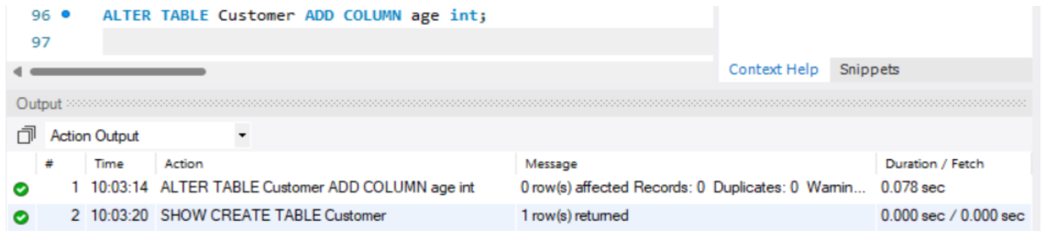
Form Editor | Navigate:

Table: Customer

Create Table:

```
CREATE TABLE `customer` (
  `customer_id` int NOT NULL AUTO_INCREMENT,
  `name` varchar(50) NOT NULL,
  `email` varchar(50) DEFAULT NULL,
  `mobile` varchar(15) DEFAULT NULL,
  PRIMARY KEY (`customer_id`),
  UNIQUE KEY `email` (`email`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

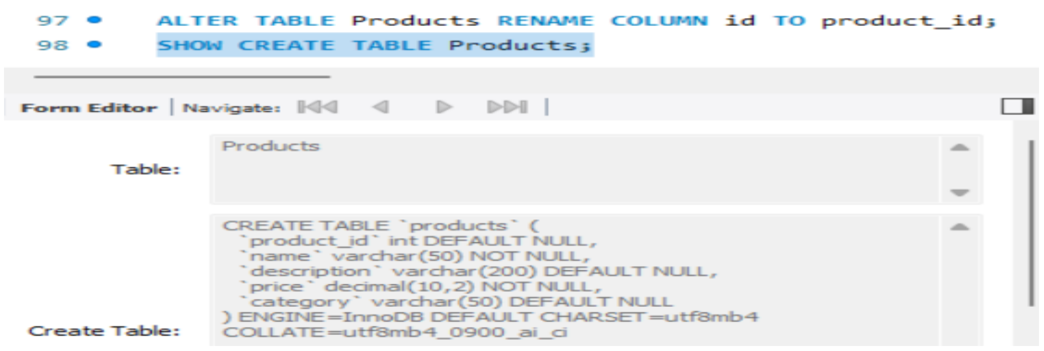
c. Add column age in the Customers table
Query: ALTER TABLE Customer ADD COLUMN age int;



```
ALTER TABLE Customer ADD COLUMN age int;
```

```
   #  Time      Action                                   Message                                                  Duration / Fetch
✓  1  10:03:14  ALTER TABLE Customer ADD COLUMN age int  0 row(s) affected Records: 0  Duplicates: 0  Warnin...  0.078 sec
✓  2  10:03:20  SHOW CREATE TABLE Customer               1 row(s) returned                                        0.000 sec / 0.000 sec
```

d. Change column name from id to product_id in the Products table;
Query : ALTER TABLE Products RENAME COLUMN id TO product_id;



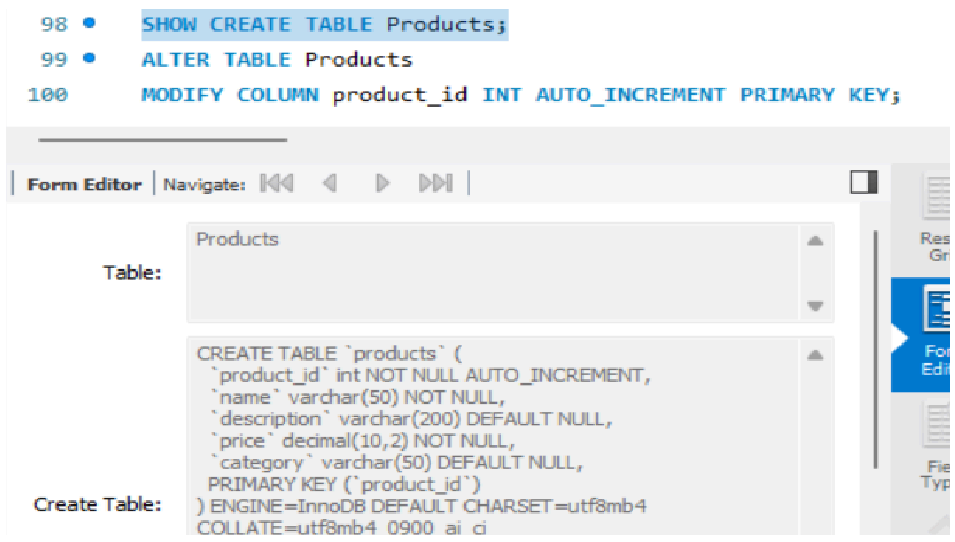
```
ALTER TABLE Products RENAME COLUMN id TO product_id;
SHOW CREATE TABLE Products;
```

Table: Products

Create Table:

```
CREATE TABLE `products` (
  `product_id` int DEFAULT NULL,
  `name` varchar(50) NOT NULL,
  `description` varchar(200) DEFAULT NULL,
  `price` decimal(10,2) NOT NULL,
  `category` varchar(50) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci
```

e. Add primary key and auto increment on product_id in the Products table
Query: ALTER TABLE Products
MODIFY COLUMN product_id INT AUTO_INCREMENT PRIMARY KEY;



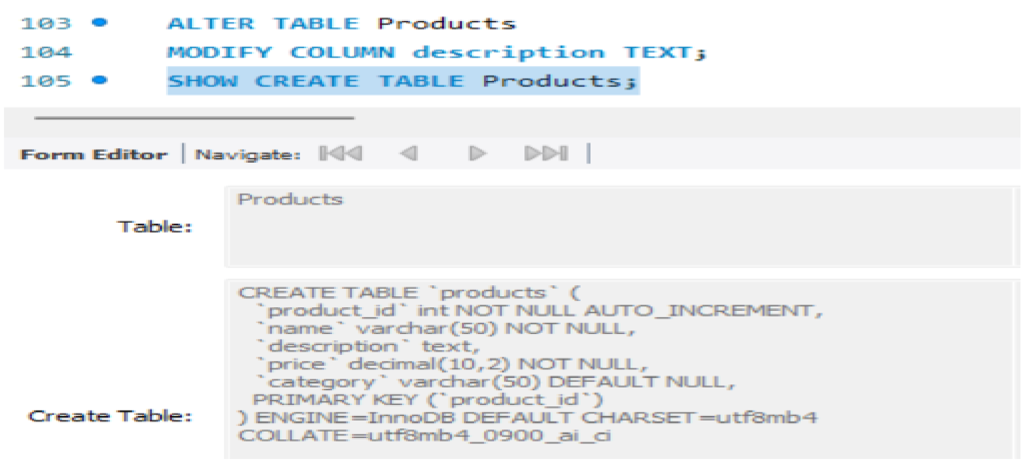
```
SHOW CREATE TABLE Products;
ALTER TABLE Products
MODIFY COLUMN product_id INT AUTO_INCREMENT PRIMARY KEY;
```

Table: Products

Create Table:

```
CREATE TABLE `products` (
  `product_id` int NOT NULL AUTO_INCREMENT,
  `name` varchar(50) NOT NULL,
  `description` varchar(200) DEFAULT NULL,
  `price` decimal(10,2) NOT NULL,
  `category` varchar(50) DEFAULT NULL,
  PRIMARY KEY (`product_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci
```

f. Change datatype of description from varchar to text in the Products table.
Query: ALTER TABLE Products
MODIFY COLUMN description TEXT;



```
ALTER TABLE Products
MODIFY COLUMN description TEXT;
SHOW CREATE TABLE Products;
```

Table: Products

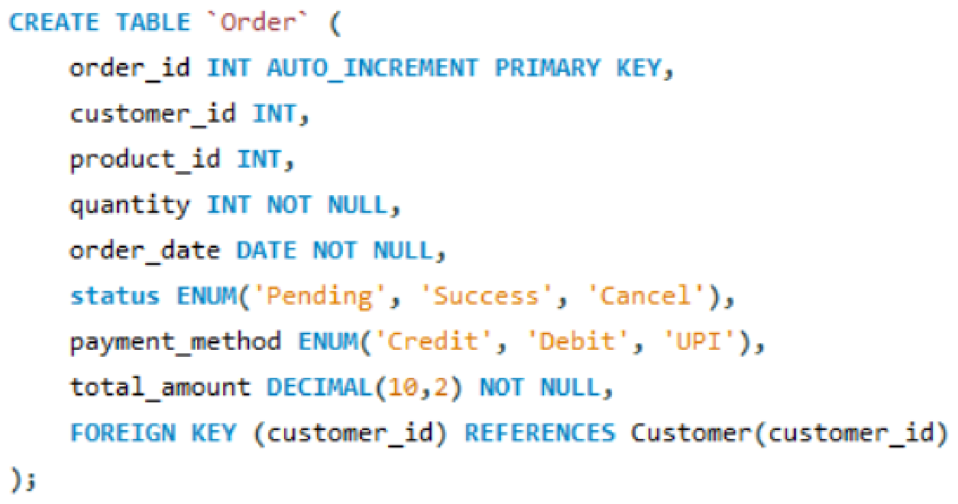
Create Table:

```
CREATE TABLE `products` (
  `product_id` int NOT NULL AUTO_INCREMENT,
  `name` varchar(50) NOT NULL,
  `description` text,
  `price` decimal(10,2) NOT NULL,
  `category` varchar(50) DEFAULT NULL,
  PRIMARY KEY (`product_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci
```

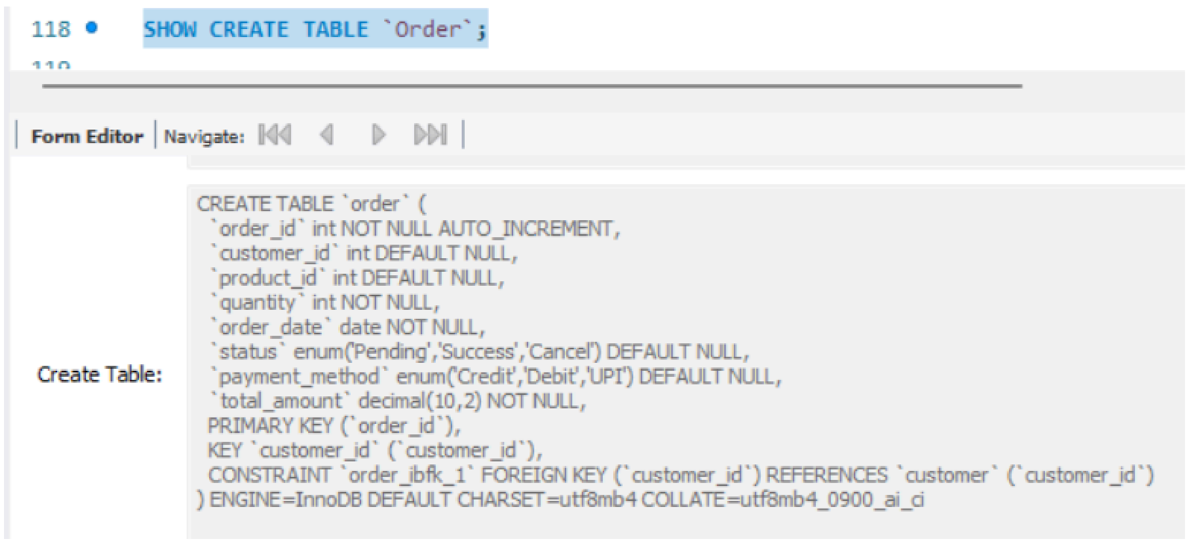
4.Create table Order:

a. order_id - int auto-increment primary key
b. customer_id - int -foreign key
c. product_id - int
d. quantity - int not null,
e. order_date - date not null,
f. status - enum(Pending, Success, Cancel),
g. payment_method - enum(Credit, Debit, UPI),
h. total_amount - decimal(10, 2) not null

-Order is reserved keyword in Sql , So to avoid conflict with reserved word backticks (`) are used to treat as it identifier



```
CREATE TABLE `Order` (
    order_id INT AUTO_INCREMENT PRIMARY KEY,
    customer_id INT,
    product_id INT,
    quantity INT NOT NULL,
    order_date DATE NOT NULL,
    status ENUM('Pending', 'Success', 'Cancel'),
    payment_method ENUM('Credit', 'Debit', 'UPI'),
    total_amount DECIMAL(10,2) NOT NULL,
    FOREIGN KEY (customer_id) REFERENCES Customer(customer_id)
);
```



```
SHOW CREATE TABLE `Order`;
```

Form Editor | Navigate:

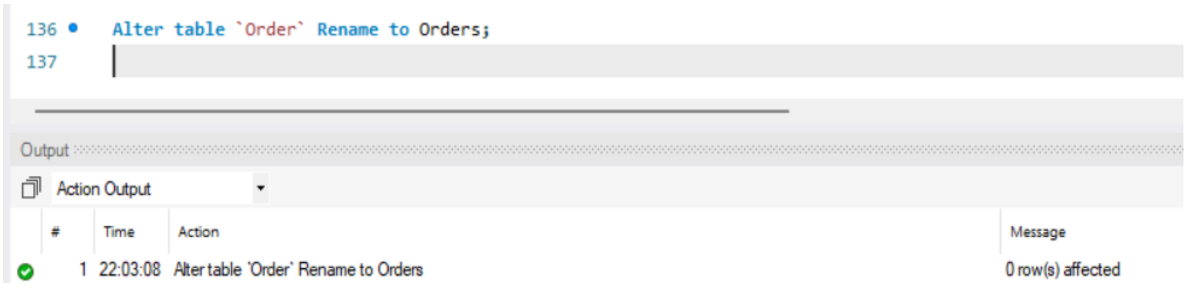
Create Table:

```
CREATE TABLE `order` (
  `order_id` int NOT NULL AUTO_INCREMENT,
  `customer_id` int DEFAULT NULL,
  `product_id` int DEFAULT NULL,
  `quantity` int NOT NULL,
  `order_date` date NOT NULL,
  `status` enum('Pending','Success','Cancel') DEFAULT NULL,
  `payment_method` enum('Credit','Debit','UPI') DEFAULT NULL,
  `total_amount` decimal(10,2) NOT NULL,
  PRIMARY KEY (`order_id`),
  KEY `customer_id` (`customer_id`),
  CONSTRAINT `order_ibfk_1` FOREIGN KEY (`customer_id`) REFERENCES `customer` (`customer_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

5.Modify Orders Table(using Alter keyword):

a. Change table name Order -> Orders

Query: Alter table `Order` Rename to Orders;



```
Alter table `Order` Rename to Orders;
```

Output

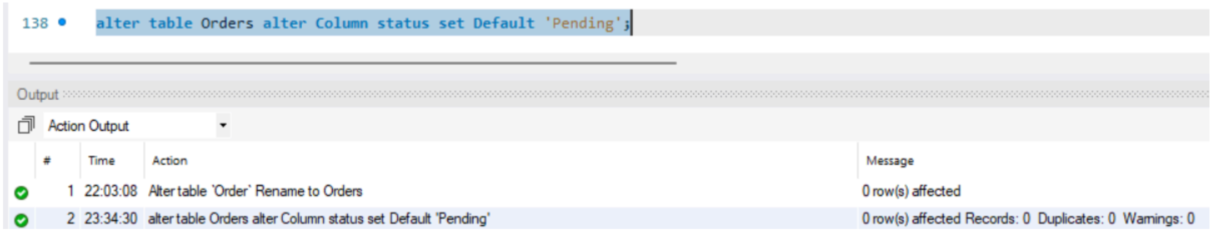
Action Output

#	Time	Action	Message
1	22:03:08	Alter table `Order` Rename to Orders	0 row(s) affected

b. Set default value pending in status.

Query: Alter table Orders alter Column status set Default 'Pending';

-By using the ALTER keyword, there is no need to redefine the data type again.



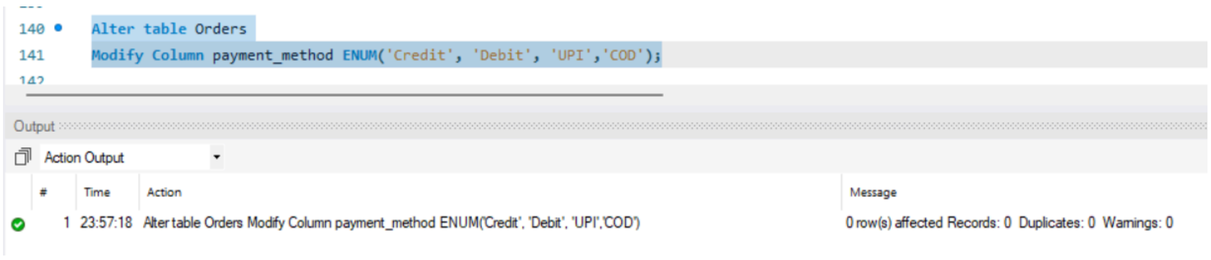
```
alter table Orders alter Column status set Default 'Pending';
```

Output

Action Output

#	Time	Action	Message
1	22:03:08	Alter table `Order` Rename to Orders	0 row(s) affected
2	23:34:30	alter table Orders alter Column status set Default 'Pending'	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

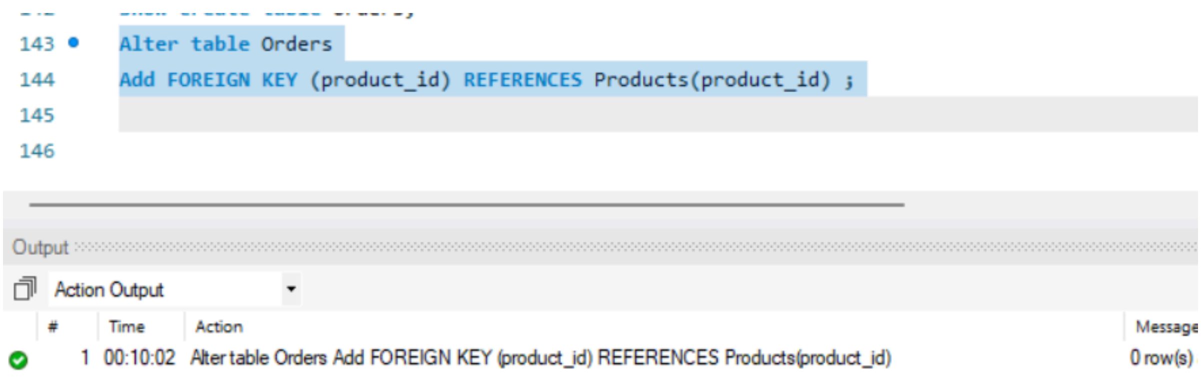
c. Modify payment_method ENUM to add one more value: 'COD'
   Query: Alter table Orders
            Modify Column payment_method ENUM('Credit', 'Debit', 'UPI','COD');



```
Alter table Orders
Modify Column payment_method ENUM('Credit', 'Debit', 'UPI','COD');
```

```
Output
Action Output
   #  Time      Action                                                                          Message
   1  23:57:18  Alter table Orders Modify Column payment_method ENUM('Credit', 'Debit', 'UPI','COD')   0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0
```

d. Make product id as foreign key
   Query:Alter table Orders
           Add FOREIGN KEY (product_id) REFERENCES Products(product_id);



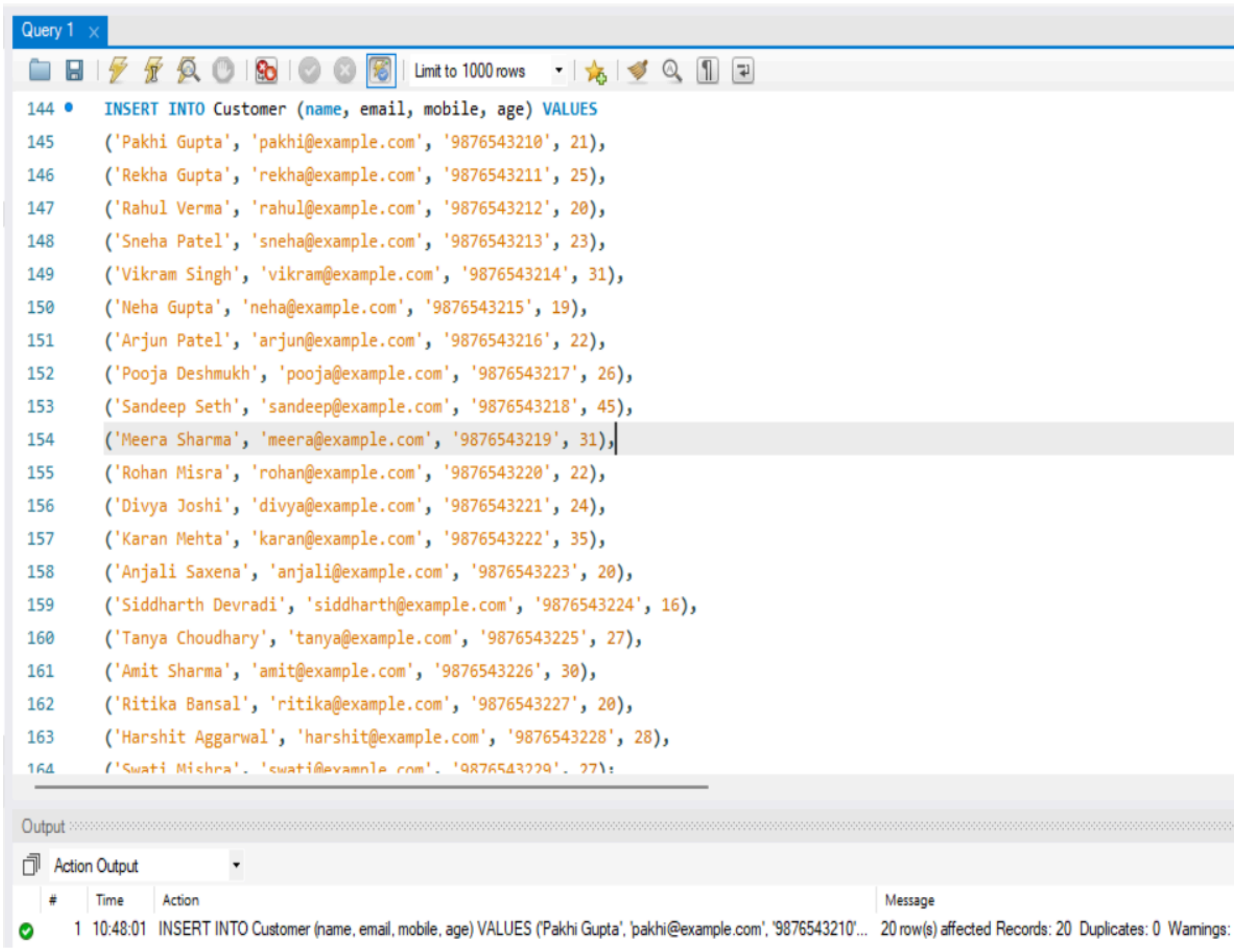
```
Alter table Orders
Add FOREIGN KEY (product_id) REFERENCES Products(product_id) ;
```

```
Output
Action Output
   #  Time      Action                                                                      Message
   1  00:10:02  Alter table Orders Add FOREIGN KEY (product_id) REFERENCES Products(product_id)   0 row(s)
```

6. Insert 20 sample records in all the tables.

Customer table:INSERT INTO Customer (name, email, mobile, age) VALUES
('Pakhi Gupta', 'pakhi@example.com', '9876543210', 21),
('Rekha Gupta', 'rekha@example.com', '9876543211', 25),
('Rahul Verma', 'rahul@example.com', '9876543212', 20),
('Sneha Patel', 'sneha@example.com', '9876543213', 23),
('Vikram Singh', 'vikram@example.com', '9876543214', 31),
('Neha Gupta', 'neha@example.com', '9876543215', 19),
('Arjun Patel', 'arjun@example.com', '9876543216', 22),
('Pooja Deshmukh', 'pooja@example.com', '9876543217', 26),
('Sandeep Seth', 'sandeep@example.com', '9876543218', 45),
('Meera Sharma', 'meera@example.com', '9876543219', 31),
('Rohan Misra', 'rohan@example.com', '9876543220', 22),
('Divya Joshi', 'divya@example.com', '9876543221', 24),
('Karan Mehta', 'karan@example.com', '9876543222', 35),
('Anjali Saxena', 'anjali@example.com', '9876543223', 20),

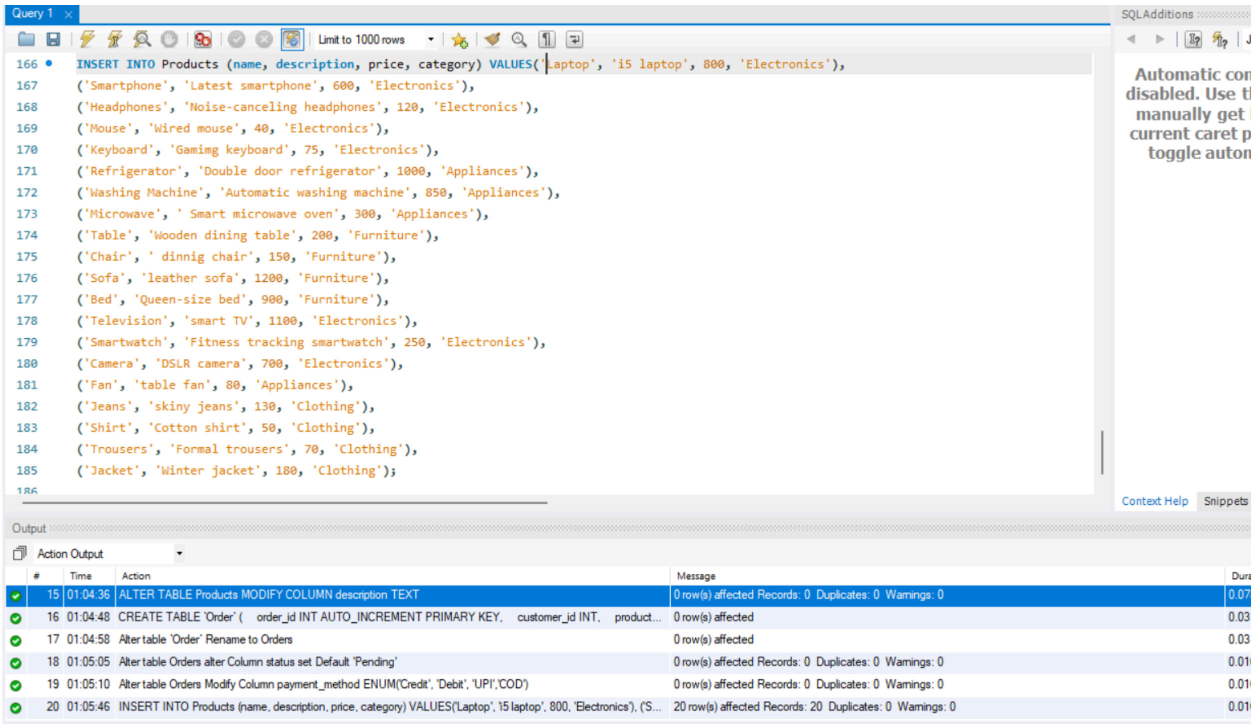
('Siddharth Devradi', 'siddharth@example.com', '9876543224', 16),
('Tanya Choudhary', 'tanya@example.com', '9876543225', 27),
('Amit Sharma', 'amit@example.com', '9876543226', 30),
('Ritika Bansal', 'ritika@example.com', '9876543227', 20),
('Harshit Aggarwal', 'harshit@example.com', '9876543228', 28),
('Swati Mishra', 'swati@example.com', '9876543229', 27);



```sql
INSERT INTO Customer (name, email, mobile, age) VALUES
('Pakhi Gupta', 'pakhi@example.com', '9876543210', 21),
('Rekha Gupta', 'rekha@example.com', '9876543211', 25),
('Rahul Verma', 'rahul@example.com', '9876543212', 20),
('Sneha Patel', 'sneha@example.com', '9876543213', 23),
('Vikram Singh', 'vikram@example.com', '9876543214', 31),
('Neha Gupta', 'neha@example.com', '9876543215', 19),
('Arjun Patel', 'arjun@example.com', '9876543216', 22),
('Pooja Deshmukh', 'pooja@example.com', '9876543217', 26),
('Sandeep Seth', 'sandeep@example.com', '9876543218', 45),
('Meera Sharma', 'meera@example.com', '9876543219', 31),
('Rohan Misra', 'rohan@example.com', '9876543220', 22),
('Divya Joshi', 'divya@example.com', '9876543221', 24),
('Karan Mehta', 'karan@example.com', '9876543222', 35),
('Anjali Saxena', 'anjali@example.com', '9876543223', 20),
('Siddharth Devradi', 'siddharth@example.com', '9876543224', 16),
('Tanya Choudhary', 'tanya@example.com', '9876543225', 27),
('Amit Sharma', 'amit@example.com', '9876543226', 30),
('Ritika Bansal', 'ritika@example.com', '9876543227', 20),
('Harshit Aggarwal', 'harshit@example.com', '9876543228', 28),
('Swati Mishra', 'swati@example.com', '9876543229', 27);
```

```
#   Time      Action                                                                               Message
1   10:48:01  INSERT INTO Customer (name, email, mobile, age) VALUES ('Pakhi Gupta', 'pakhi@example.com', '9876543210'...   20 row(s) affected Records: 20  Duplicates: 0  Warnings:
```

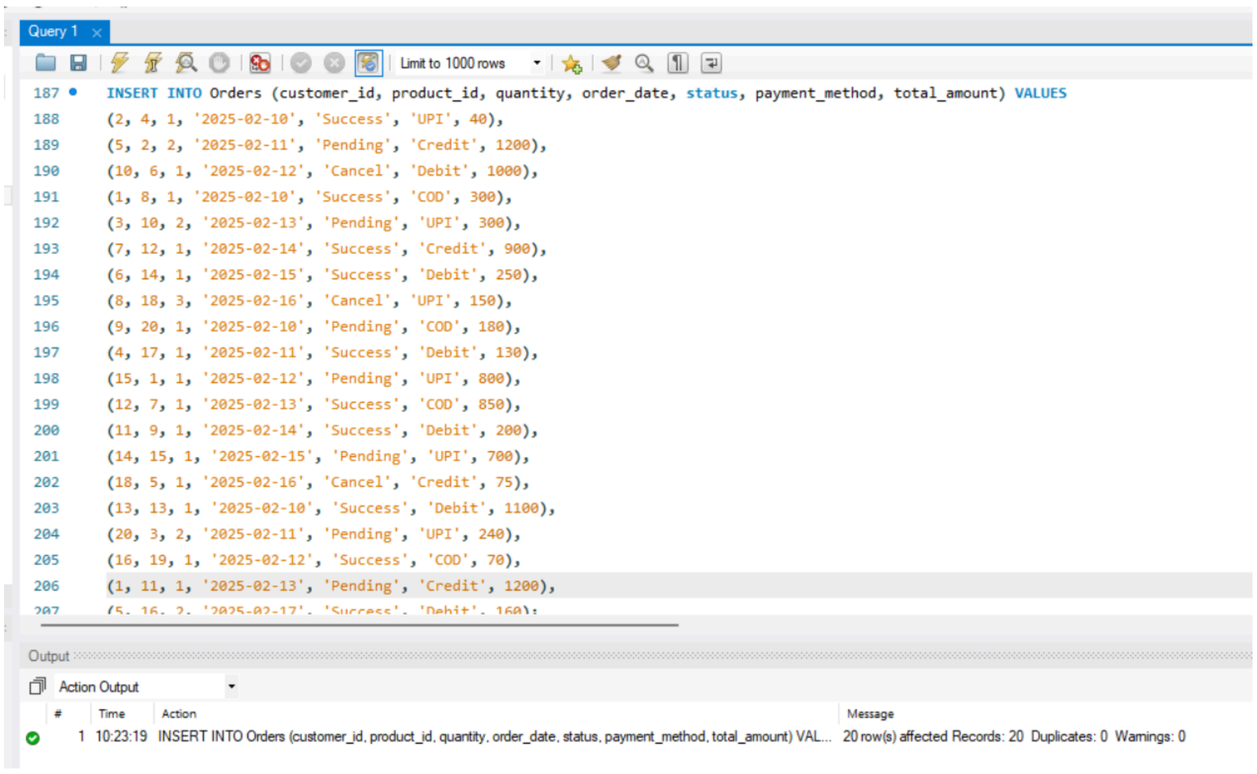
Products Table: INSERT INTO Orders (customer_id, product_id, quantity, order_date, status, payment_method, total_amount) VALUES (2, 4, 1, '2025-02-10', 'Success', 'UPI', 40), (5, 2, 2, '2025-02-11', 'Pending', 'Credit', 1200), (10, 6, 1, '2025-02-12', 'Cancel', 'Debit', 1000), (1, 8, 1, '2025-02-10', 'Success', 'COD', 300), (3, 10, 2, '2025-02-13', 'Pending', 'UPI', 300), (7, 12, 1, '2025-02-14', 'Success', 'Credit', 900), (6, 14, 1, '2025-02-15', 'Success', 'Debit', 250), (8, 18, 3, '2025-02-16', 'Cancel', 'UPI', 150), (9, 20, 1, '2025-02-10', 'Pending', 'COD', 180), (4, 17, 1, '2025-02-11', 'Success', 'Debit', 130), (15, 1, 1, '2025-02-12', 'Pending', 'UPI', 800), (12, 7, 1, '2025-02-13', 'Success', 'COD', 850), (11, 9, 1, '2025-02-14', 'Success', 'Debit', 200), (14, 15, 1, '2025-02-15', 'Pending', 'UPI', 700), (18, 5, 1, '2025-02-16', 'Cancel', 'Credit', 75), (13, 13, 1, '2025-02-10', 'Success', 'Debit', 1100), (20, 3, 2, '2025-02-11', 'Pending', 'UPI', 240), (16, 19, 1, '2025-02-12', 'Success', 'COD', 70), (17, 11, 1, '2025-02-13', 'Pending', 'Credit', 1200), (19, 16, 2, '2025-02-17', 'Success', 'Debit', 160);



```sql
INSERT INTO Products (name, description, price, category) VALUES('Laptop', 'i5 laptop', 800, 'Electronics'),
('Smartphone', 'Latest smartphone', 600, 'Electronics'),
('Headphones', 'Noise-canceling headphones', 120, 'Electronics'),
('Mouse', 'Wired mouse', 40, 'Electronics'),
('Keyboard', 'Gamimg keyboard', 75, 'Electronics'),
('Refrigerator', 'Double door refrigerator', 1000, 'Appliances'),
('Washing Machine', 'Automatic washing machine', 850, 'Appliances'),
('Microwave', ' Smart microwave oven', 300, 'Appliances'),
('Table', 'Wooden dining table', 200, 'Furniture'),
('Chair', ' dinnig chair', 150, 'Furniture'),
('Sofa', 'leather sofa', 1200, 'Furniture'),
('Bed', 'Queen-size bed', 900, 'Furniture'),
('Television', 'smart TV', 1100, 'Electronics'),
('Smartwatch', 'Fitness tracking smartwatch', 250, 'Electronics'),
('Camera', 'DSLR camera', 700, 'Electronics'),
('Fan', 'table fan', 80, 'Appliances'),
('Jeans', 'skiny jeans', 130, 'Clothing'),
('Shirt', 'Cotton shirt', 50, 'Clothing'),
('Trousers', 'Formal trousers', 70, 'Clothing'),
('Jacket', 'Winter jacket', 180, 'Clothing');
```

```
#   Time      Action                                                                                          Message                                                  Dura
15  01:04:36  ALTER TABLE Products MODIFY COLUMN description TEXT                                             0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0  0.07
16  01:04:48  CREATE TABLE `Order` (    order_id INT AUTO_INCREMENT PRIMARY KEY,    customer_id INT,    product...  0 row(s) affected                                         0.03
17  01:04:58  Alter table `Order` Rename to Orders                                                            0 row(s) affected                                         0.03
18  01:05:05  Alter table Orders alter Column status set Default 'Pending'                                    0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0  0.01
19  01:05:10  Alter table Orders Modify Column payment_method ENUM('Credit', 'Debit', 'UPI','COD')            0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0  0.01
20  01:05:46  INSERT INTO Products (name, description, price, category) VALUES('Laptop', 'i5 laptop', 800, 'Electronics'), ('S...  20 row(s) affected Records: 20  Duplicates: 0  Warnings: 0  0.01
```

Orders Table: INSERT INTO Orders (customer_id, product_id, quantity, order_date, status, payment_method, total_amount) VALUES (2, 4, 1, '2025-02-10', 'Success', 'UPI', 40), (5, 2, 2, '2025-02-11', 'Pending', 'Credit', 1200), (10, 6, 1, '2025-02-12', 'Cancel', 'Debit', 1000), (1, 8, 1, '2025-02-10', 'Success', 'COD', 300), (3, 10, 2, '2025-02-13', 'Pending', 'UPI', 300), (7, 12, 1, '2025-02-14', 'Success', 'Credit', 900), (6, 14, 1, '2025-02-15', 'Success', 'Debit', 250), (8, 18, 3, '2025-02-16', 'Cancel', 'UPI', 150), (9, 20, 1, '2025-02-10', 'Pending', 'COD', 180), (4, 17, 1, '2025-02-11', 'Success', 'Debit', 130), (15, 1, 1, '2025-02-12', 'Pending', 'UPI', 800), (12, 7, 1, '2025-02-13', 'Success', 'COD', 850), (11, 9, 1, '2025-02-14', 'Success', 'Debit', 200), (14, 15, 1, '2025-02-15', 'Pending', 'UPI', 700), (18, 5, 1, '2025-02-16', 'Cancel', 'Credit', 75), (13, 13, 1, '2025-02-10', 'Success', 'Debit', 1100), (20, 3, 2, '2025-02-11', 'Pending', 'UPI', 240), (16, 19, 1, '2025-02-12', 'Success', 'COD', 70), (1, 11, 1, '2025-02-13', 'Pending', 'Credit', 1200), (5, 16, 2, '2025-02-17', 'Success', 'Debit', 160);



Query 1

Limit to 1000 rows

```sql
INSERT INTO Orders (customer_id, product_id, quantity, order_date, status, payment_method, total_amount) VALUES
(2, 4, 1, '2025-02-10', 'Success', 'UPI', 40),
(5, 2, 2, '2025-02-11', 'Pending', 'Credit', 1200),
(10, 6, 1, '2025-02-12', 'Cancel', 'Debit', 1000),
(1, 8, 1, '2025-02-10', 'Success', 'COD', 300),
(3, 10, 2, '2025-02-13', 'Pending', 'UPI', 300),
(7, 12, 1, '2025-02-14', 'Success', 'Credit', 900),
(6, 14, 1, '2025-02-15', 'Success', 'Debit', 250),
(8, 18, 3, '2025-02-16', 'Cancel', 'UPI', 150),
(9, 20, 1, '2025-02-10', 'Pending', 'COD', 180),
(4, 17, 1, '2025-02-11', 'Success', 'Debit', 130),
(15, 1, 1, '2025-02-12', 'Pending', 'UPI', 800),
(12, 7, 1, '2025-02-13', 'Success', 'COD', 850),
(11, 9, 1, '2025-02-14', 'Success', 'Debit', 200),
(14, 15, 1, '2025-02-15', 'Pending', 'UPI', 700),
(18, 5, 1, '2025-02-16', 'Cancel', 'Credit', 75),
(13, 13, 1, '2025-02-10', 'Success', 'Debit', 1100),
(20, 3, 2, '2025-02-11', 'Pending', 'UPI', 240),
(16, 19, 1, '2025-02-12', 'Success', 'COD', 70),
(1, 11, 1, '2025-02-13', 'Pending', 'Credit', 1200),
(5, 16, 2, '2025-02-17', 'Success', 'Debit', 160);
```

Output

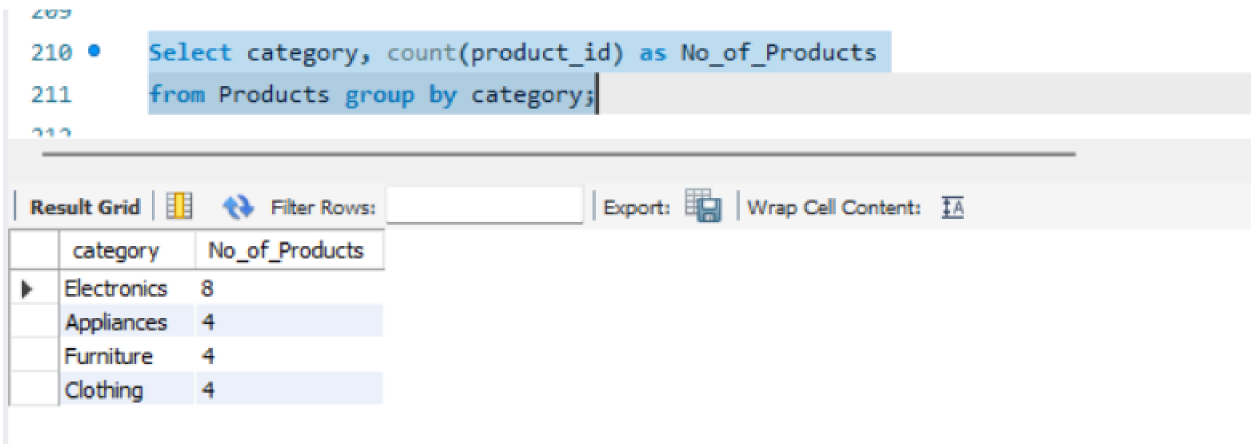
Action Output

#	Time	Action	Message
1	10:23:19	INSERT INTO Orders (customer_id, product_id, quantity, order_date, status, payment_method, total_amount) VAL...	20 row(s) affected Records: 20 Duplicates: 0 Warnings: 0

7.Perform following queries:

a. Count the number of products as product_count in each category.

Query:Select category, count(product_id) as No_of_Products
from Products group by category;



```sql
Select category, count(product_id) as No_of_Products
from Products group by category;
```

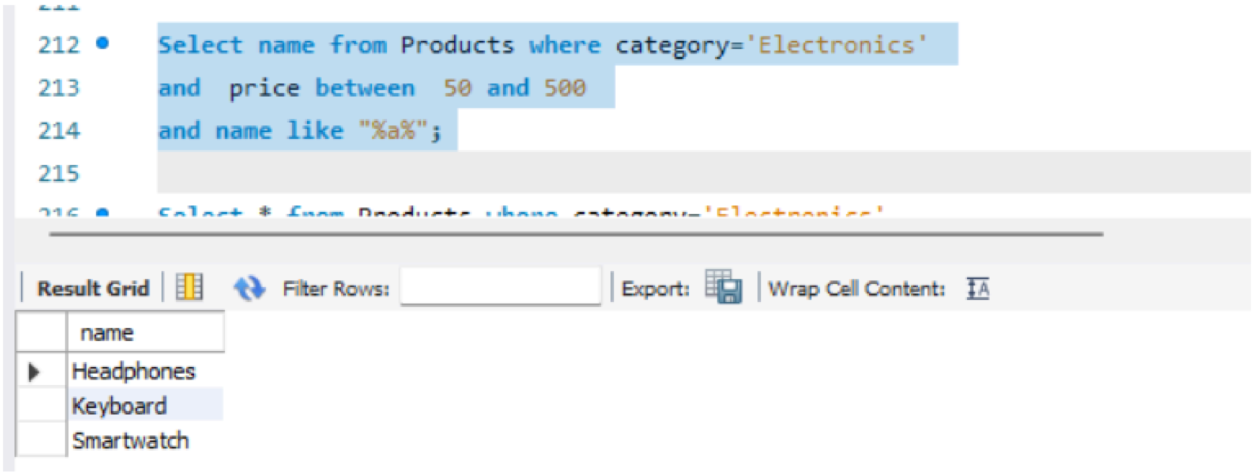
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

category	No_of_Products
Electronics	8
Appliances	4
Furniture	4
Clothing	4

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

Query:Select name from Products where category='Electronics'
and price between 50 and 500

and name like "%a%";

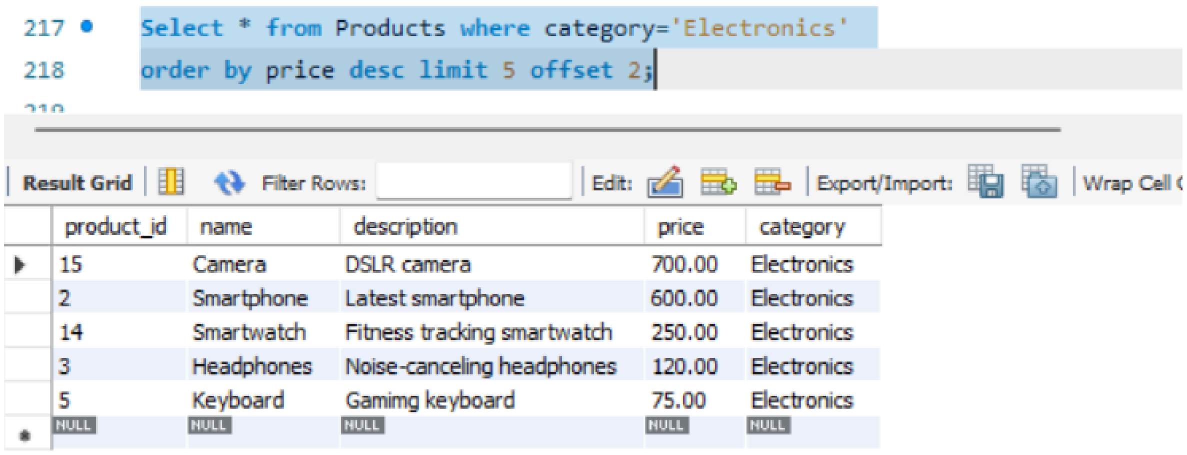


```
Select name from Products where category='Electronics'
and  price between  50 and 500
and name like "%a%";
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name
Headphones
Keyboard
Smartwatch

c. Get the top 5 most expensive products in the 'Electronics' category, skipping the first 2.
Query:Select * from Products where category='Electronics'
order by price desc limit 5 offset 2;

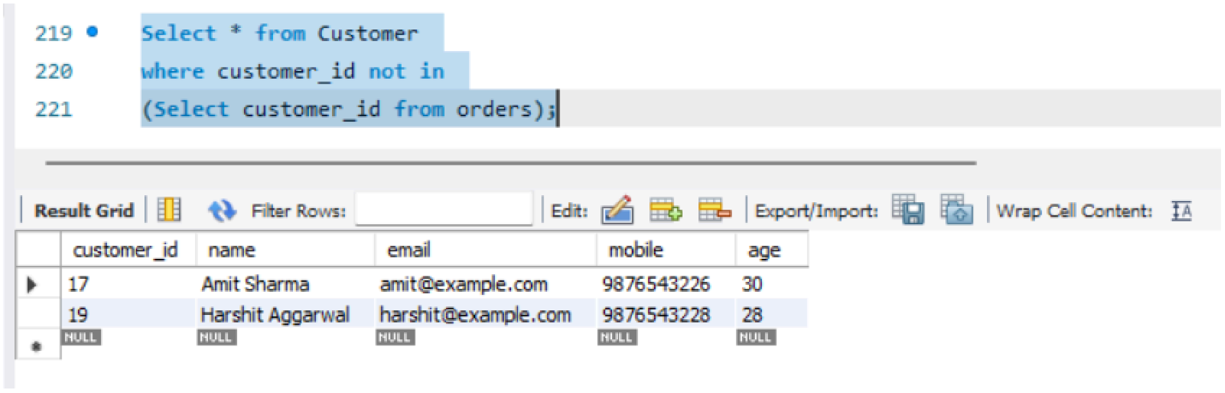


```
Select * from Products where category='Electronics'
order by price desc limit 5 offset 2;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell

product_id	name	description	price	category
15	Camera	DSLR camera	700.00	Electronics
2	Smartphone	Latest smartphone	600.00	Electronics
14	Smartwatch	Fitness tracking smartwatch	250.00	Electronics
3	Headphones	Noise-canceling headphones	120.00	Electronics
5	Keyboard	Gamimg keyboard	75.00	Electronics
NULL	NULL	NULL	NULL	NULL

d. Retrieve customers who have not placed any orders.
Query: Select * from Customer
where customer_id not in
(Select customer_id from orders);

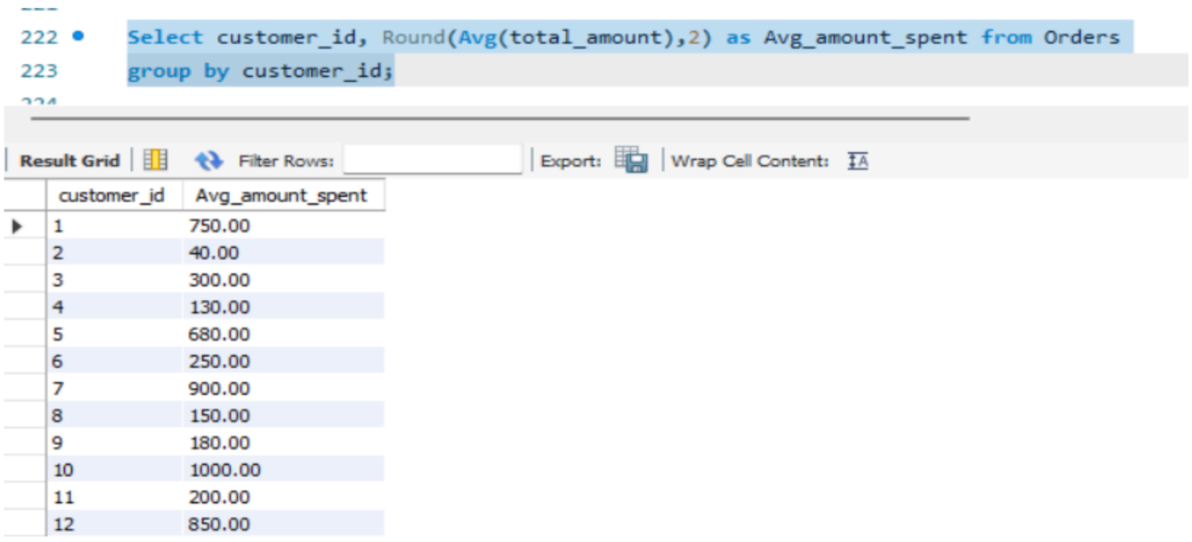


```
Select * from Customer
where customer_id not in
(Select customer_id from orders);
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

customer_id	name	email	mobile	age
17	Amit Sharma	amit@example.com	9876543226	30
19	Harshit Aggarwal	harshit@example.com	9876543228	28
NULL	NULL	NULL	NULL	NULL

e. Find the average total amount spent by each customer.
Query: Select customer_id, Round(Avg(total_amount),2) as Avg_amount_spent
from Orders group by customer_id;

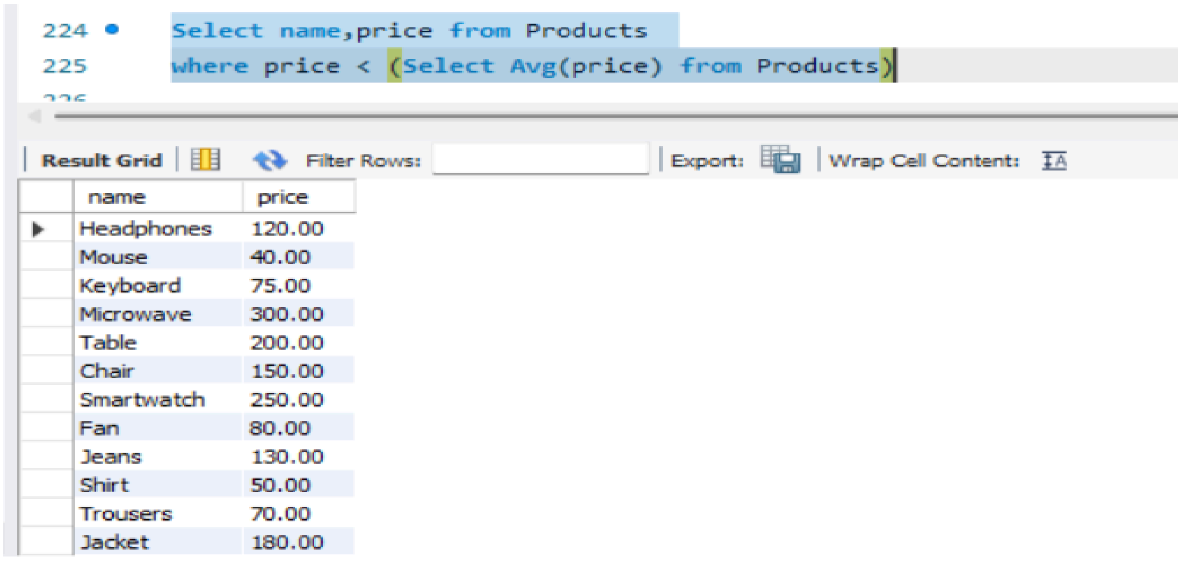


```
Select customer_id, Round(Avg(total_amount),2) as Avg_amount_spent from Orders
group by customer_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	Avg_amount_spent
1	750.00
2	40.00
3	300.00
4	130.00
5	680.00
6	250.00
7	900.00
8	150.00
9	180.00
10	1000.00
11	200.00
12	850.00

f. Get the products that have a price less than the average price of all products.
Query: Select name,price from Products
where price < (Select Avg(price) from Products);



```
Select name,price from Products
where price < (Select Avg(price) from Products)
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

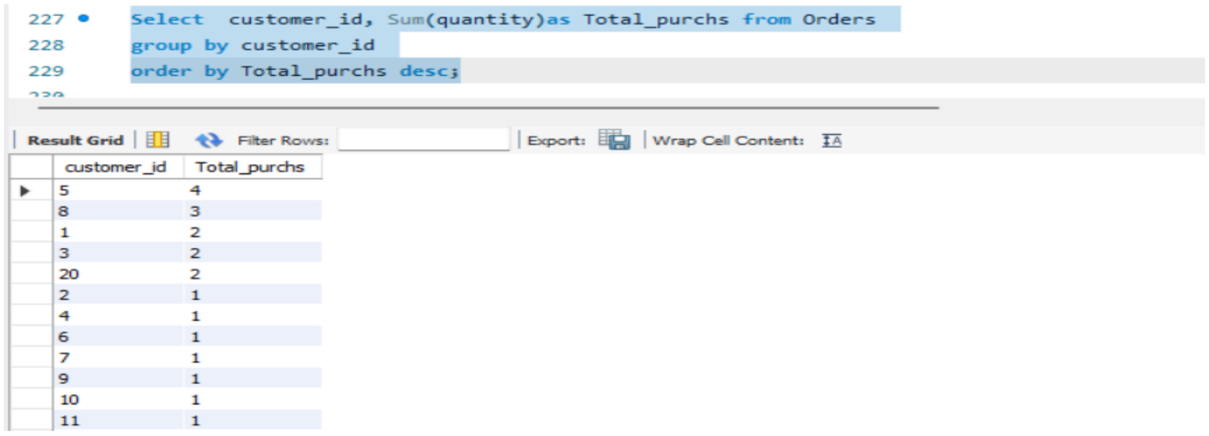
name	price
Headphones	120.00
Mouse	40.00
Keyboard	75.00
Microwave	300.00
Table	200.00
Chair	150.00
Smartwatch	250.00
Fan	80.00
Jeans	130.00
Shirt	50.00
Trousers	70.00
Jacket	180.00

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

Query: Select customer_id, Sum(quantity)as Total_purchs from Orders

group by customer_id

order by Total_purchs desc;



```
Select  customer_id, Sum(quantity)as Total_purchs from Orders
group by customer_id
order by Total_purchs desc;
```

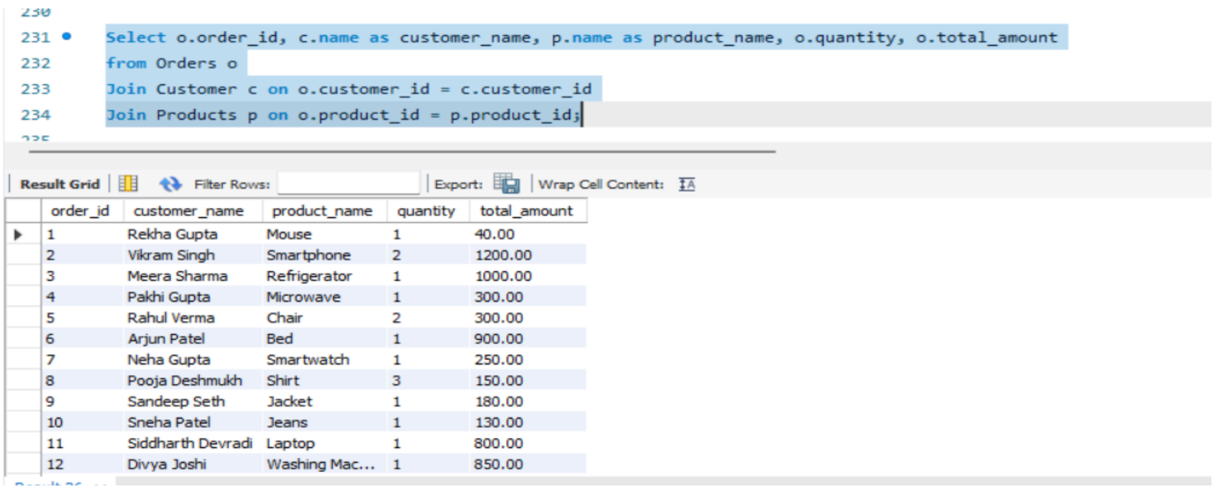
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	Total_purchs
5	4
8	3
1	2
3	2
20	2
2	1
4	1
6	1
7	1
9	1
10	1
11	1

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

Query:

Select o.order_id, c.name as customer_name, p.name as product_name,
o.quantity, o.total_amount
from Orders o
Join Customer c on o.customer_id = c.customer_id
Join Products p on o.product_id = p.product_id;



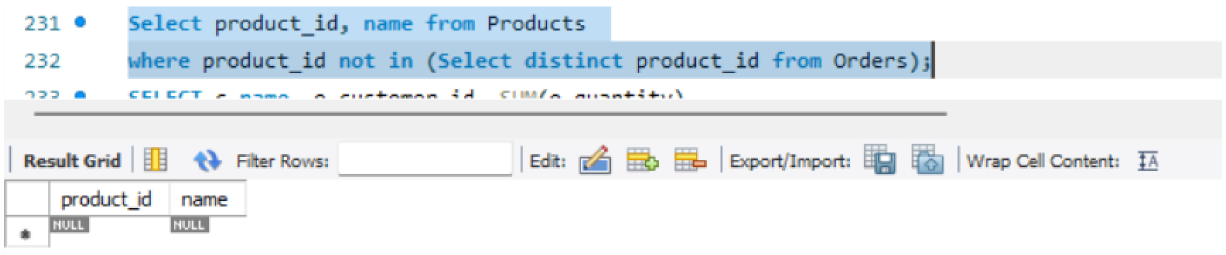
```sql
Select o.order_id, c.name as customer_name, p.name as product_name, o.quantity, o.total_amount
from Orders o
Join Customer c on o.customer_id = c.customer_id
Join Products p on o.product_id = p.product_id;
```

order_id	customer_name	product_name	quantity	total_amount
1	Rekha Gupta	Mouse	1	40.00
2	Vikram Singh	Smartphone	2	1200.00
3	Meera Sharma	Refrigerator	1	1000.00
4	Pakhi Gupta	Microwave	1	300.00
5	Rahul Verma	Chair	2	300.00
6	Arjun Patel	Bed	1	900.00
7	Neha Gupta	Smartwatch	1	250.00
8	Pooja Deshmukh	Shirt	3	150.00
9	Sandeep Seth	Jacket	1	180.00
10	Sneha Patel	Jeans	1	130.00
11	Siddharth Devradi	Laptop	1	800.00
12	Divya Joshi	Washing Mac...	1	850.00

i. Find products that have never been ordered.

Query:Select product_id, name from Products
where product_id not in (Select distinct product_id from Orders);

-Every product is purchased by a customer, so the output is displayed like this.



```sql
Select product_id, name from Products
where product_id not in (Select distinct product_id from Orders);
```

product_id	name
NULL	NULL