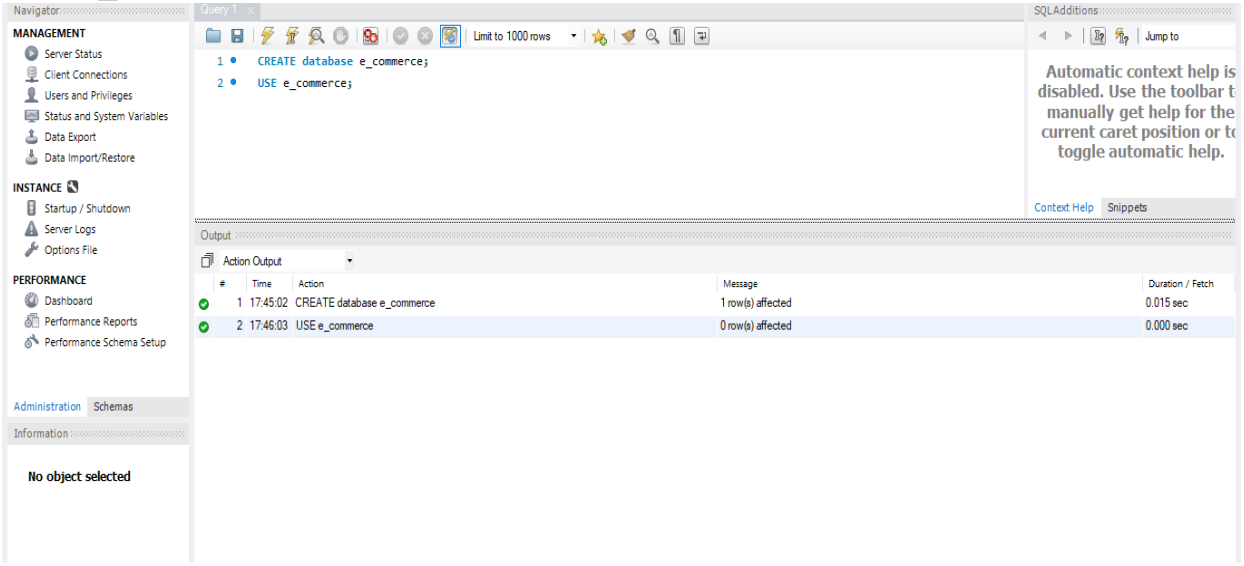
SQL Assignment – 1

1. **Create database e_commerce**

   CREATE database e_commerce;

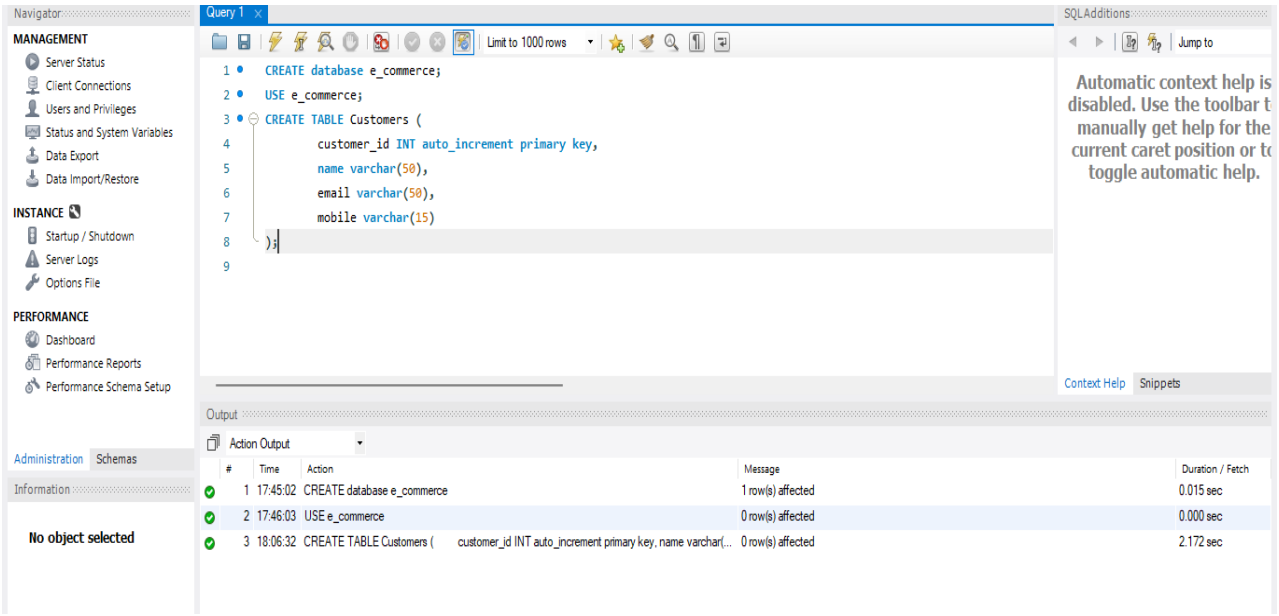
   USE e_commerce;



2. **Create following Tables:**

   Customers:

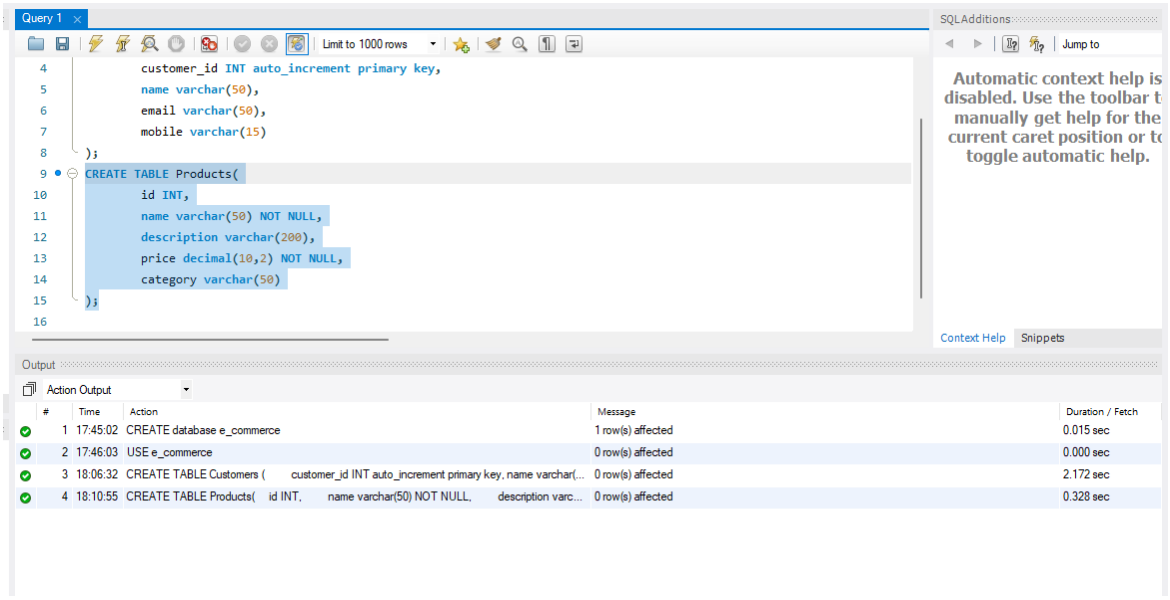
   a. customer_id - int auto-increment primary key
   b. name - varchar(50)
   c. email - varchar(50)
   d. mobile - varchar(15)



   Products:

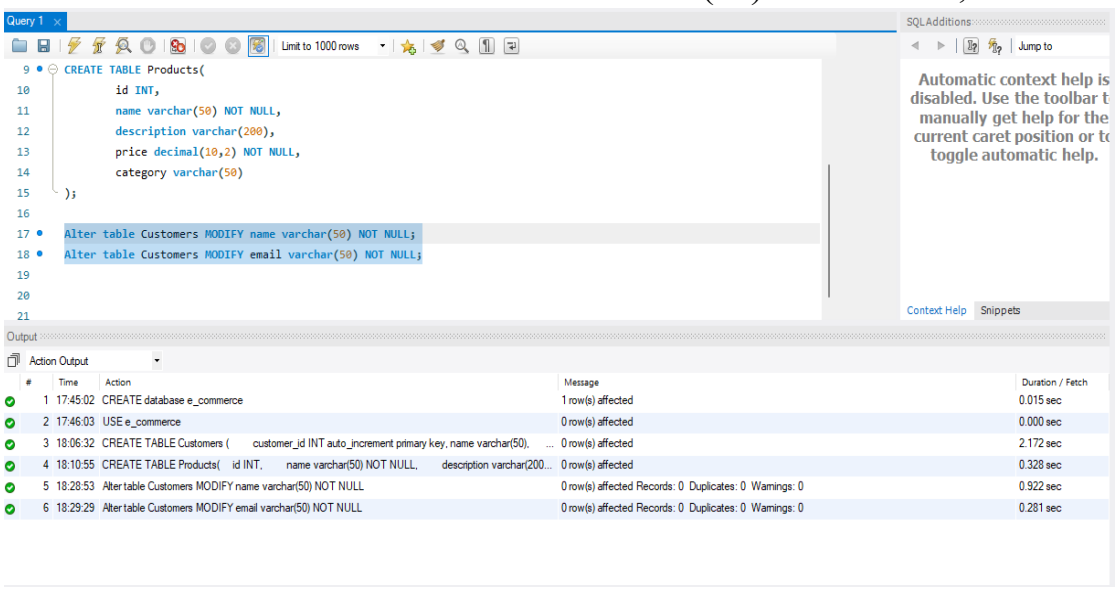
   a. id – int
   b. name - varchar(50) not null

c. description - varchar(200)
d. price - decimal(10, 2) not null
e. category - varchar(50)

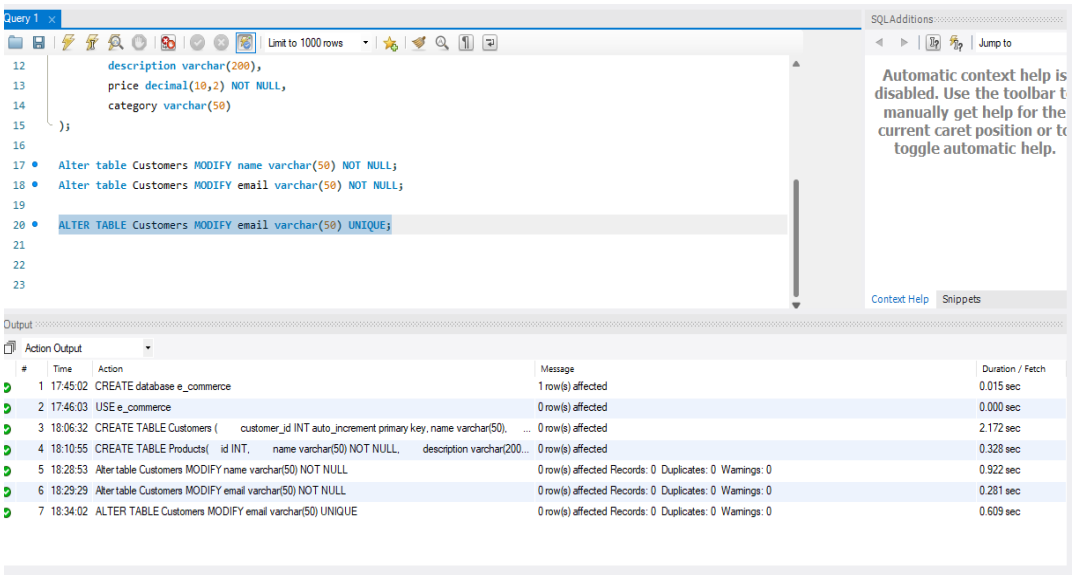


3. Modify Tables(using Alter keyword):

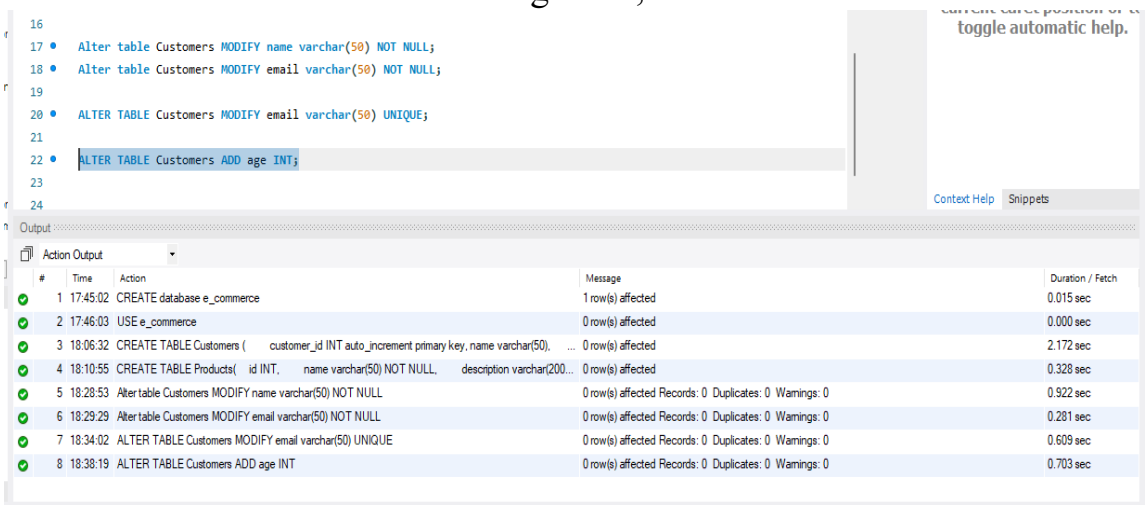
a. Add not null on name and email in the Customers table.
Alter table Customers MODIFY name varchar(50) NOT NULL;
Alter table Customers MODIFY email varchar(50) NOT NULL;



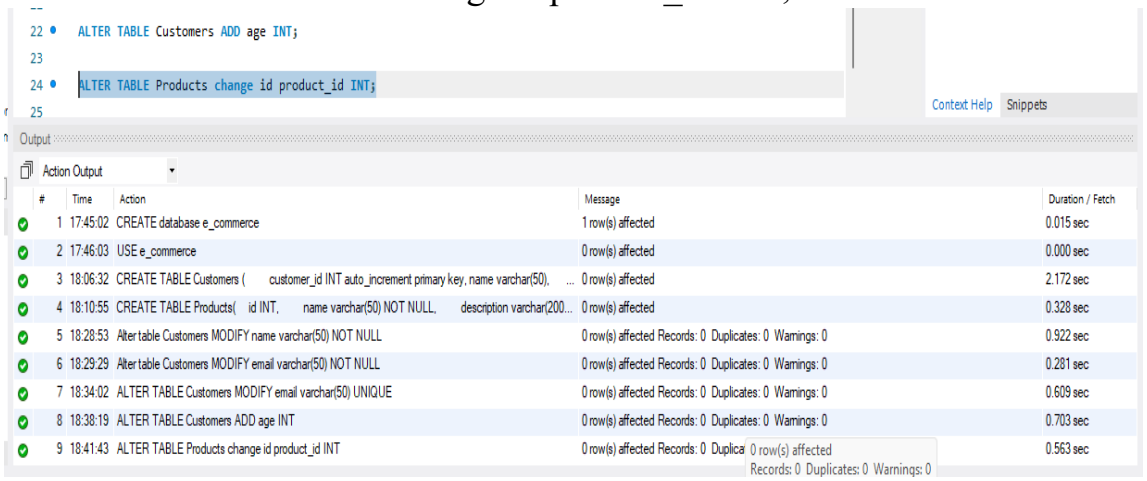
b. Add unique key on email in the Customers table
ALTER TABLE Customers MODIFY email varchar(50) UNIQUE;



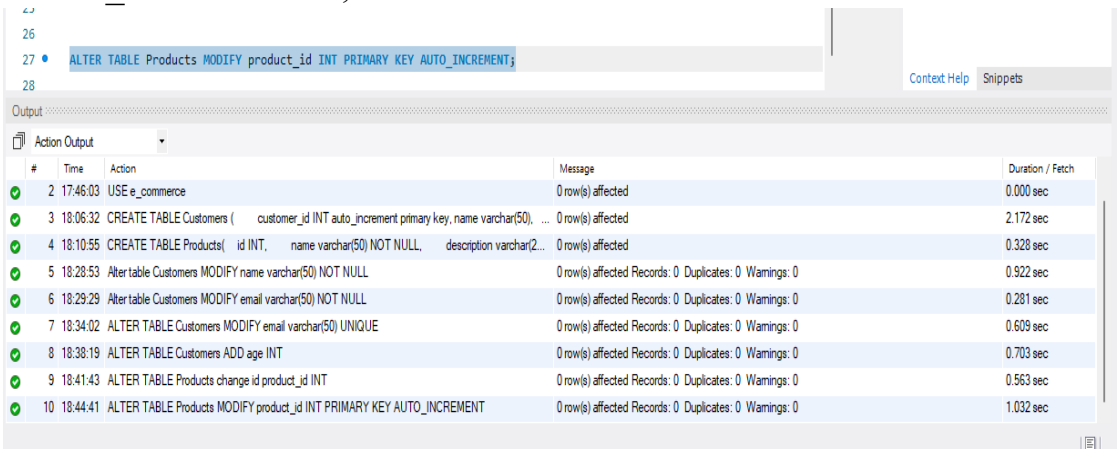
c. Add column age in the Customers table
ALTER TABLE Customers ADD age INT;



d. Change column name from id to product_id in the Products table;
ALTER TABLE Products change id product_id INT;

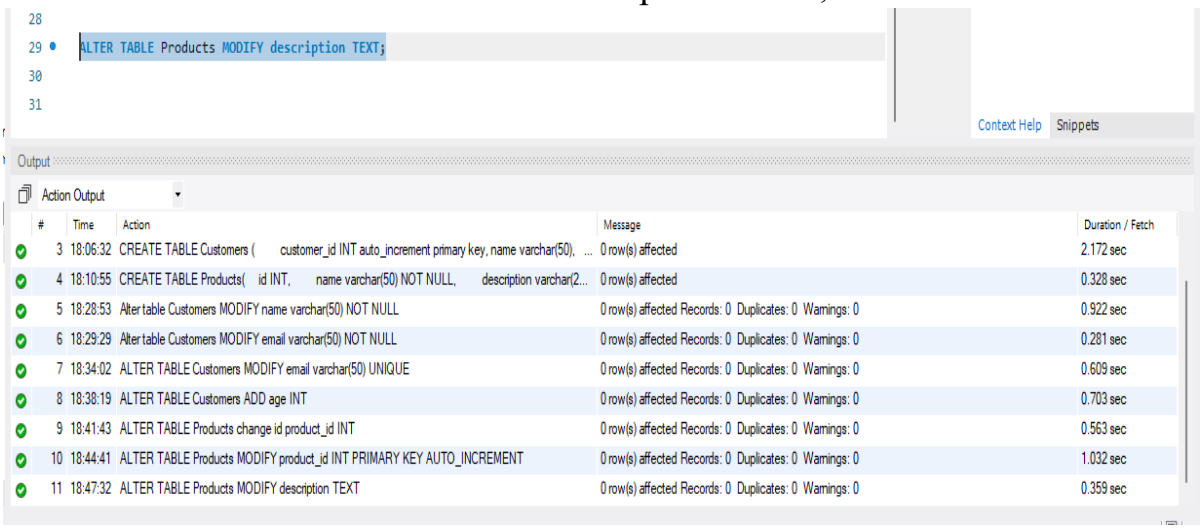


e. Add primary key and auto increment on product_id in the Products table.
ALTER TABLE Products MODIFY product_id INT PRIMARY KEY AUTO_INCREMENT;



#	Time	Action	Message	Duration / Fetch
2	17:46:03	USE e_commerce	0 row(s) affected	0.000 sec
3	18:06:32	CREATE TABLE Customers (customer_id INT auto_increment primary key, name varchar(50), ...	0 row(s) affected	2.172 sec
4	18:10:55	CREATE TABLE Products(id INT, name varchar(50) NOT NULL, description varchar(2...	0 row(s) affected	0.328 sec
5	18:28:53	Alter table Customers MODIFY name varchar(50) NOT NULL	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.922 sec
6	18:29:29	Alter table Customers MODIFY email varchar(50) NOT NULL	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.281 sec
7	18:34:02	ALTER TABLE Customers MODIFY email varchar(50) UNIQUE	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.609 sec
8	18:38:19	ALTER TABLE Customers ADD age INT	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.703 sec
9	18:41:43	ALTER TABLE Products change id product_id INT	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.563 sec
10	18:44:41	ALTER TABLE Products MODIFY product_id INT PRIMARY KEY AUTO_INCREMENT	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	1.032 sec

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



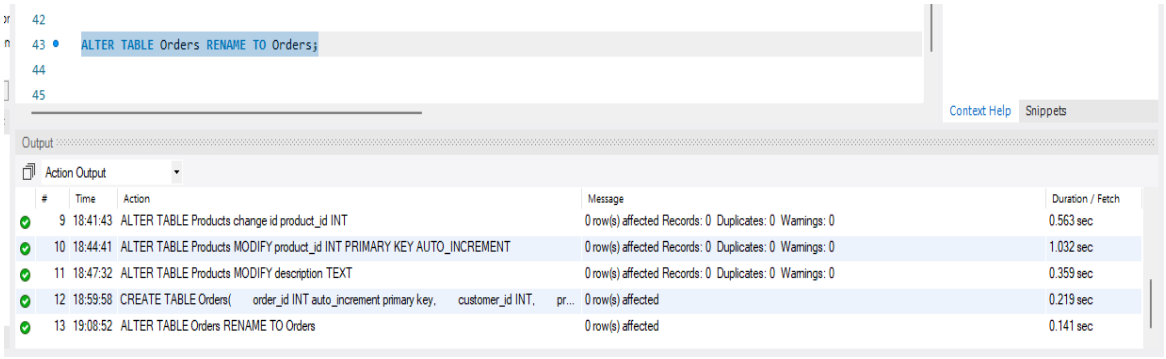
#	Time	Action	Message	Duration / Fetch
3	18:06:32	CREATE TABLE Customers (customer_id INT auto_increment primary key, name varchar(50), ...	0 row(s) affected	2.172 sec
4	18:10:55	CREATE TABLE Products(id INT, name varchar(50) NOT NULL, description varchar(2...	0 row(s) affected	0.328 sec
5	18:28:53	Alter table Customers MODIFY name varchar(50) NOT NULL	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.922 sec
6	18:29:29	Alter table Customers MODIFY email varchar(50) NOT NULL	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.281 sec
7	18:34:02	ALTER TABLE Customers MODIFY email varchar(50) UNIQUE	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.609 sec
8	18:38:19	ALTER TABLE Customers ADD age INT	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.703 sec
9	18:41:43	ALTER TABLE Products change id product_id INT	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.563 sec
10	18:44:41	ALTER TABLE Products MODIFY product_id INT PRIMARY KEY AUTO_INCREMENT	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	1.032 sec
11	18:47:32	ALTER TABLE Products MODIFY description TEXT	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.359 sec

4. Create table Order:

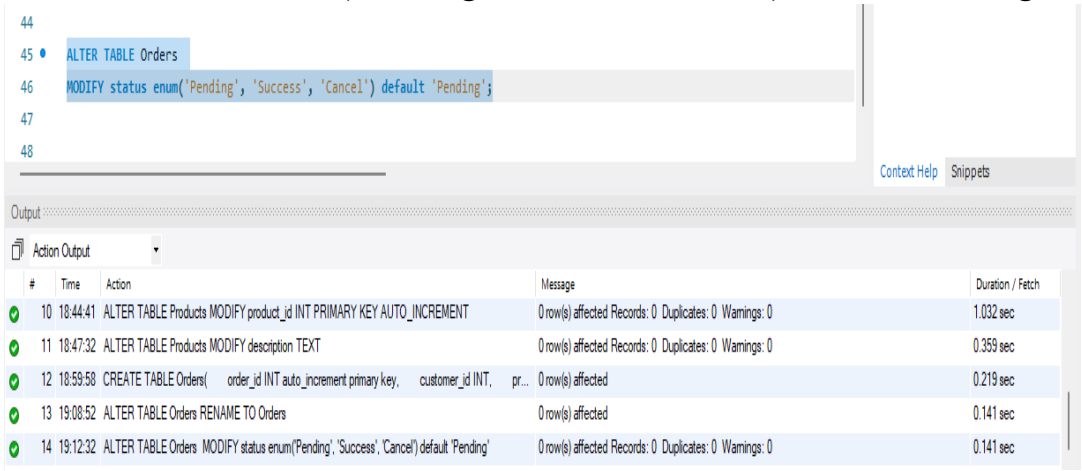
#	Time	Action	Message	Duration / Fetch
4	18:10:55	CREATE TABLE Products(id INT, name varchar(50) NOT NULL, description varchar(2...	0 row(s) affected	0.328 sec
5	18:28:53	Alter table Customers MODIFY name varchar(50) NOT NULL	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.922 sec
6	18:29:29	Alter table Customers MODIFY email varchar(50) NOT NULL	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.281 sec
7	18:34:02	ALTER TABLE Customers MODIFY email varchar(50) UNIQUE	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.609 sec
8	18:38:19	ALTER TABLE Customers ADD age INT	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.703 sec
9	18:41:43	ALTER TABLE Products change id product_id INT	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.563 sec
10	18:44:41	ALTER TABLE Products MODIFY product_id INT PRIMARY KEY AUTO_INCREMENT	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	1.032 sec
11	18:47:32	ALTER TABLE Products MODIFY description TEXT	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.359 sec
12	18:59:58	CREATE TABLE Orders(order_id INT auto_increment primary key, customer_id INT, pr...	0 row(s) affected	0.219 sec

5. Modify Orders Table(using Alter keyword):

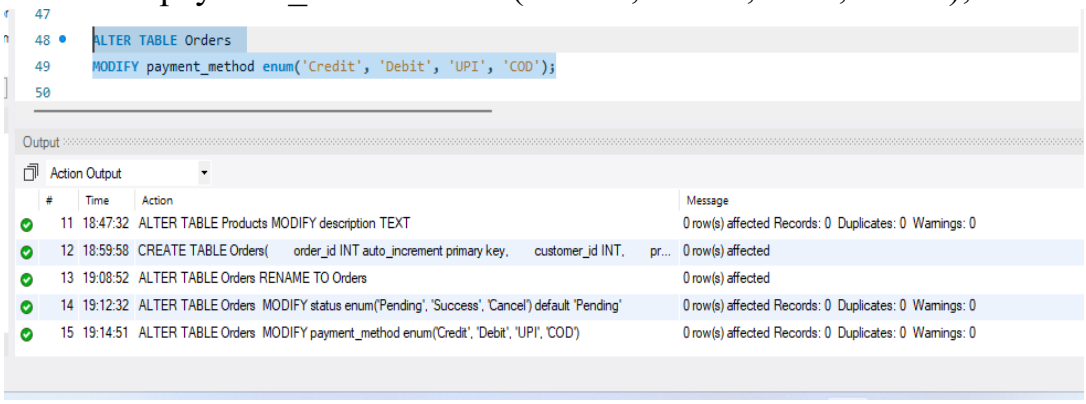
a. Change table name Order -> Orders
ALTER TABLE Orders RENAME TO Orders;



b. Set default value pending in status.
ALTER TABLE Orders
MODIFY status enum('Pending', 'Success', 'Cancel') default 'Pending';



c. Modify payment_method ENUM to add one more value: 'COD'
ALTER TABLE Orders
MODIFY payment_method enum('Credit', 'Debit', 'UPI', 'COD');



d. Make product id as foreign key
ALTER TABLE Orders ADD CONSTRAINT fk_product FOREIGN KEY(product_id) REFERENCES Products(product_id);

```
50
51 •  ALTER TABLE Orders
52    ADD CONSTRAINT fk_product
53    FOREIGN KEY(product_id) REFERENCES
54    Products(product_id);
55
```

Context H

Output

Action Output

#	Time	Action	Message
12	18:59:58	CREATE TABLE Orders(order_id INT auto_increment primary key, customer_id INT, pr...	0 row(s) affected
13	19:08:52	ALTER TABLE Orders RENAME TO Orders	0 row(s) affected
14	19:12:32	ALTER TABLE Orders MODIFY status enum('Pending', 'Success', 'Cancel') default 'Pending'	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
15	19:14:51	ALTER TABLE Orders MODIFY payment_method enum('Credit', 'Debit', 'UPI', 'COD')	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
16	19:24:48	ALTER TABLE Orders ADD CONSTRAINT fk_product FOREIGN KEY(product_id) REFERENCES ...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

6. Insert 20 sample records in all the tables.

Customers:

SQL Assignment -1

Limit to 1000 rows

```
55
56 •  INSERT INTO Customers (name, email, mobile, age) VALUES
57    ('John Doe', 'john@example.com', '9876543210', 30),
58    ('Jane Smith', 'jane@example.com', '9876543211', 28),
59    ('Alice Johnson', 'alice@example.com', '9876543212', 35),
```

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

customer_id	name	email	mobile	age
1	John Doe	john@example.com	9876543210	30
2	Jane Smith	jane@example.com	9876543211	28
3	Alice Johnson	alice@example.com	9876543212	35
4	Bob Brown	bob@example.com	9876543213	40
5	Charlie Adams	charlie@example.com	9876543214	25
6	David Wilson	david@example.com	9876543215	32
7	Emma Davis	emma@example.com	9876543216	27
8	Frank Thomas	frank@example.com	9876543217	29
9	Grace White	grace@example.com	9876543218	31
10	Henry Harris	henry@example.com	9876543219	33
11	Isabella Lewis	isabella@example.com	9876543220	26
12	Jack Robinson	jack@example.com	9876543221	34
13	Kelly Martinez	kelly@example.com	9876543222	28

Customers 1 | Apply

Output

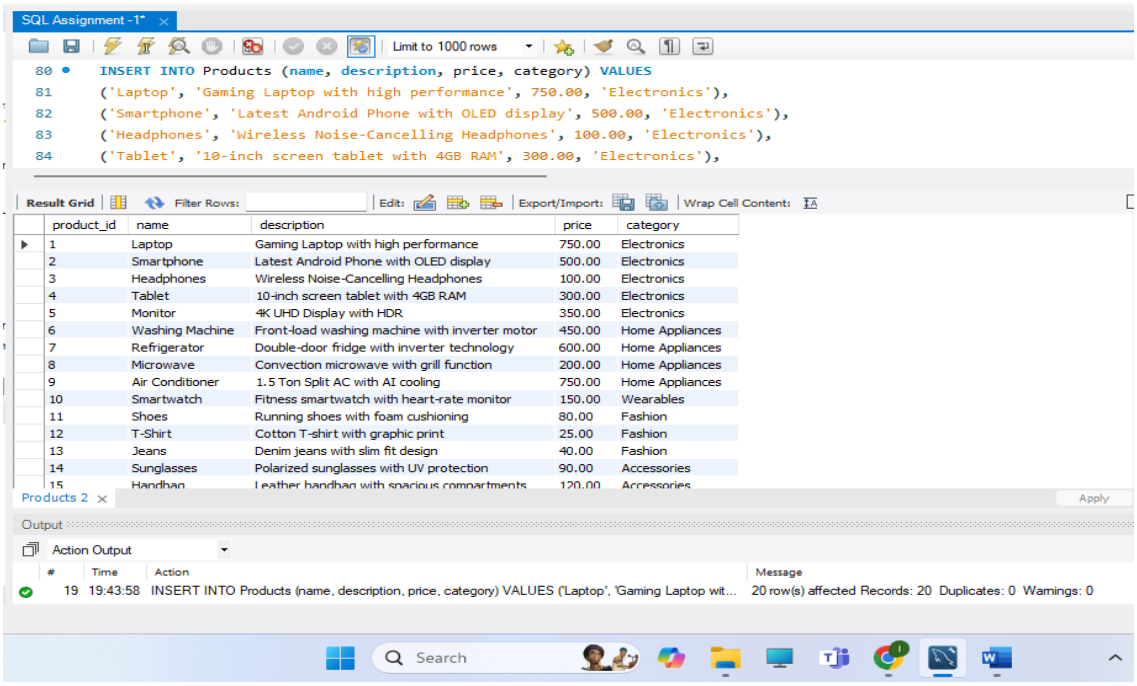
Action Output

#	Time	Action	Message
17	19:33:58	INSERT INTO Customers (name, email, mobile, age) VALUES ('John Doe', 'john@example.com', '98...	20 row(s) affected Records: 20 Duplicates: 0 Warnings: 0
18	19:34:29	select * from Customers LIMIT 0, 1000	20 row(s) returned

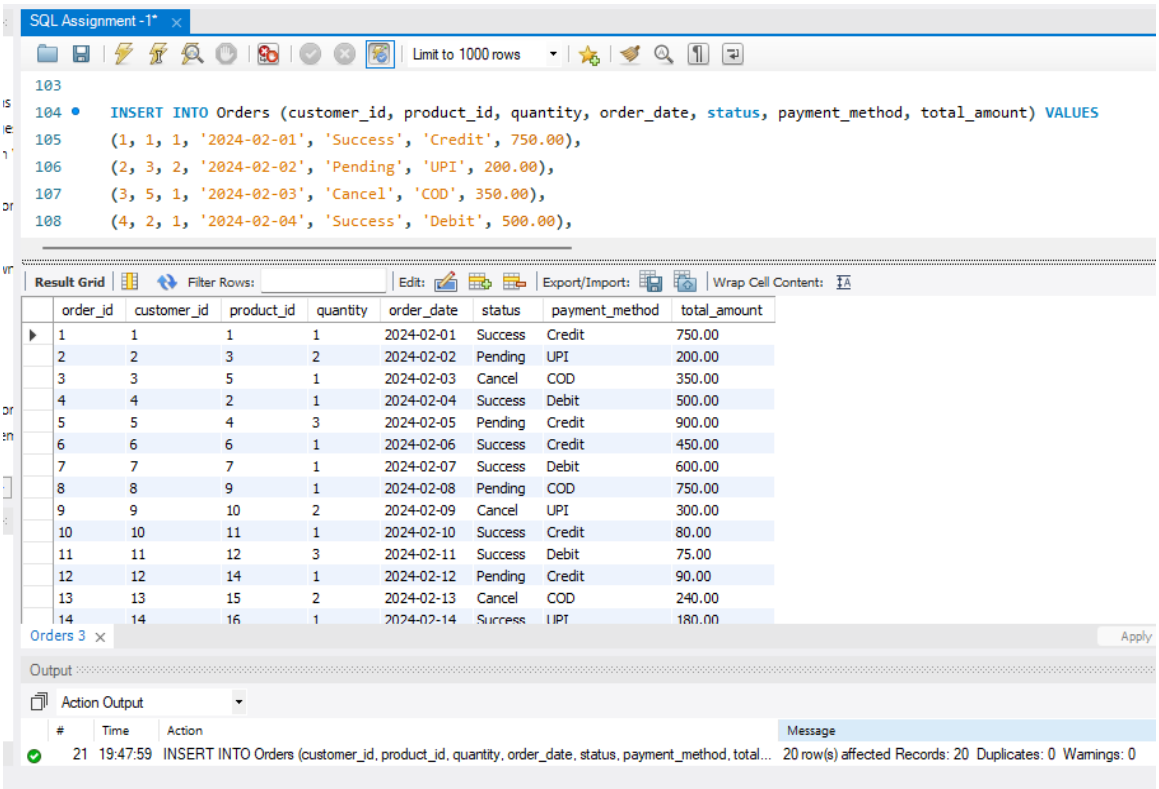
s\Admin\Documents\SQL Assignment -1.sql'

Search

Products:



Orders:



7. Perform following queries:

a. Count the number of products as product_count in each category. SELECT category, COUNT(*) AS product_count from Products group by category;

```
126
127 •  SELECT category, COUNT(*) AS product_count from Products group by category;
128
129
```

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

category	product_count
Electronics	8
Home Appliances	5
Wearables	1
Fashion	3
Accessories	2
Furniture	1

Result 4 ×

Output

b. Retrieve all products that belong to the 'Electronics' category, have a price between $50 and $500, and whose name contains the letter 'a'.
SELECT * FROM Products where category='Electronics' AND price BETWEEN 50 AND 500 AND name LIKE '%a%';

```
131 •  SELECT * FROM Products where category='Electronics' AND price BETWEEN 50 AND 500 AND name LIKE '%a%';
132
133
```

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

product_id	name	description	price	category
2	Smartphone	Latest Android Phone with OLED display	500.00	Electronics
3	Headphones	Wireless Noise-Cancelling Headphones	100.00	Electronics
4	Tablet	10-inch screen tablet with 4GB RAM	300.00	Electronics
17	Keyboard	Mechanical keyboard with RGB lighting	75.00	Electronics
NULL	NULL	NULL	NULL	NULL

c. Get the top 5 most expensive products in the 'Electronics' category, skipping the first 2.
SELECT * FROM Products where category=’Electronics’ order by price desc LIMIT 5 OFFSET 2;

```
132
133 •  SELECT * FROM Products where category='Electronics' order by price desc LIMIT 5 OFFSET 2;
134
```

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

product_id	name	description	price	category
2	Smartphone	Latest Android Phone with OLED display	500.00	Electronics
5	Monitor	4K UHD Display with HDR	350.00	Electronics
4	Tablet	10-inch screen tablet with 4GB RAM	300.00	Electronics
3	Headphones	Wireless Noise-Cancelling Headphones	100.00	Electronics
17	Keyboard	Mechanical keyboard with RGB lighting	75.00	Electronics
NULL	NULL	NULL	NULL	NULL

d. Retrieve customers who have not placed any orders.
SELECT * FROM Customers where customer_id NOT IN(SELECT Distinct customer_id from Orders);

```
SELECT * FROM Customers where customer_id NOT IN(SELECT Distinct customer_id from Orders);
```

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

customer_id	name	email	mobile	age
NULL	NULL	NULL	NULL	NULL

e. Find the average total amount spent by each customer.
SELECT customer_id, avg(total_amount) from Orders group by customer_id;

```
SELECT customer_id, avg(total_amount) from Orders group by customer_id;
```

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

customer_id	avg(total_amount)
1	750.000000
2	200.000000
3	350.000000
4	500.000000
5	900.000000
6	450.000000
7	600.000000
8	750.000000
9	300.000000
10	80.000000
11	75.000000
12	90.000000
13	240.000000
14	180.000000
15	150.000000
16	50.000000
17	900.000000
18	250.000000

Result 8

f. Get the products that have a price less than the average price of all products.
SELECT * FROM Products where price< (SELECT AVG(price) FROM Products);

```
SELECT * FROM Products where price< (SELECT AVG(price) FROM Products);
```

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

product_id	name	description	price	category
3	Headphones	Wireless Noise-Cancelling Headphones	100.00	Electronics
8	Microwave	Convection microwave with grill function	200.00	Home Appliances
10	Smartwatch	Fitness smartwatch with heart-rate monitor	150.00	Wearables
11	Shoes	Running shoes with foam cushioning	80.00	Fashion
12	T-Shirt	Cotton T-shirt with graphic print	25.00	Fashion
13	Jeans	Denim jeans with slim fit design	40.00	Fashion
14	Sunglasses	Polarized sunglasses with UV protection	90.00	Accessories
15	Handbag	Leather handbag with spacious compartments	120.00	Accessories
16	Coffee Maker	Automatic coffee machine with milk frother	180.00	Home Appliances
17	Keyboard	Mechanical keyboard with RGB lighting	75.00	Electronics
18	Mouse	Wireless gaming mouse with high DPI	50.00	Electronics
20	Desk Chair	Ergonomic chair with lumbar support	250.00	Furniture
NULL	NULL	NULL	NULL	NULL

g. Calculate the total quantity of products ordered by each customer.
SELECT customer_id, SUM(quantity) FROM Orders group by customer_id;

```
139
140 •  SELECT customer_id, SUM(quantity) FROM Orders group by customer_id;
141
142
```

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

customer_id	SUM(quantity)
1	1
2	2
3	1
4	1
5	3
6	1
7	1
8	1
9	2
10	1
11	3
12	1
13	2
14	1
15	2
16	1
17	1
18	1

Result 11

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

SELECT O.order_id, C.name AS customer_name, P.name AS product_name, O.quantity, O.total_amount from Orders AS O
JOIN
Customers AS C ON O.customer_id= C.customer_id
JOIN
Products AS P ON O.product_id= P.product_id;

SQL Assignment -1*

Limit to 1000 rows

```
142 •  SELECT O.order_id, C.name AS customer_name, P.name AS product_name, O.quantity, O.total_amount from Orders AS O
143    JOIN
144    Customers AS C ON O.customer_id= C.customer_id
145    JOIN
146    Products AS P ON O.product_id= P.product_id;
```

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

order_id	customer_name	product_name	quantity	total_amount
1	John Doe	Laptop	1	750.00
2	Jane Smith	Headphones	2	200.00
3	Alice Johnson	Monitor	1	350.00
4	Bob Brown	Smartphone	1	500.00
5	Charlie Adams	Tablet	3	900.00
6	David Wilson	Washing Machine	1	450.00
7	Emma Davis	Refrigerator	1	600.00
8	Frank Thomas	Air Conditioner	1	750.00
9	Grace White	Smartwatch	2	300.00
10	Henry Harris	Shoes	1	80.00
11	Isabella Lewis	T-Shirt	3	75.00
12	Jack Robinson	Sunglasses	1	90.00
13	Kelly Martinez	Handbag	2	240.00
14	Liam Clark	Coffee Maker	1	180.00
15	Mia Rodriguez	Keyboard	2	150.00
16	Noah Walker	Mouse	1	50.00
17	Olivia Allen	Smart TV	1	900.00
18	Paul Young	Desk Chair	1	250.00

Result 12

i. Find products that have never been ordered.
SELECT * FROM Products where product_id NOT IN (SELECT DISTINCT product_id FROM Orders);