

Name : SHYAM S

Company: CODTECH IT SOLUTIONS

Intern ID : CT08DS6496

Domain : SQL

Duration: August to September 2024

Mentor : Muzammil Ahmed

ONLINE RETAIL STORE DATABASE

To Create a database for an online retail store, including products, customers, orders, and payments. This project involves more complex queries and database design. Design tables for products, customers, orders, and payments. Write SQL queries to handle customer orders and payment processing.

Objective :

The objective of the online retail store database is to efficiently manage customer information, product inventories, and order processing. It aims to facilitate seamless transactions and enhance user experience through structured data relationships. The design ensures data integrity, scalability, and quick retrieval of information. Ultimately, it supports the operational needs of the retail business while providing a reliable platform for customers.

Database Schema :

1) Create a database :

```
CREATE DATABASE Online_Retail_Store_Database ;
```

2) Use Database :

```
USE Online_Retail_Store_Database ;
```

3) Products Table

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'Schemas' list with 'online_retail_store_database' selected. The main editor window shows the following SQL query:

```
1 CREATE TABLE Products (  
2   product_id INT PRIMARY KEY AUTO_INCREMENT,  
3   product_name VARCHAR(100) NOT NULL,  
4   category VARCHAR(50),  
5   description TEXT,  
6   price DECIMAL(10, 2) NOT NULL,  
7   quantity_in_stock INT NOT NULL  
8 );
```

The 'Output' tab at the bottom shows the execution results:

#	Time	Action	Message	Duration / Fetch
1	13.23.18	create database Online_Retail_Store_Database	1 row(s) affected	0.031 sec
2	13.23.36	use Online_Retail_Store_Database	0 row(s) affected	0.016 sec
3	13.25.22	CREATE TABLE Products (product_id INT PRIMARY KEY AUTO_INCREMENT, product_name VARCHA...	0 row(s) affected	0.125 sec

4) Customers Table :

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'Schemas' list with 'online_retail_store_database' selected. The main editor window shows the following SQL query:

```
1 CREATE TABLE Customers (  
2   first_name VARCHAR(50) NOT NULL,  
3   last_name VARCHAR(50) NOT NULL,  
4   email VARCHAR(100) UNIQUE NOT NULL,  
5   phone VARCHAR(15),  
6   address VARCHAR(255),  
7   city VARCHAR(50),  
8   state VARCHAR(50),  
9   zip_code VARCHAR(10),  
10  country VARCHAR(50)  
11 );
```

The 'Output' tab at the bottom shows the execution results:

#	Time	Action	Message	Duration / Fetch
1	13.23.18	create database Online_Retail_Store_Database	1 row(s) affected	0.031 sec
2	13.23.36	use Online_Retail_Store_Database	0 row(s) affected	0.016 sec
3	13.25.22	CREATE TABLE Products (product_id INT PRIMARY KEY AUTO_INCREMENT, product_name VARCHA...	0 row(s) affected	0.125 sec
4	13.27.04	CREATE TABLE Customers (customer_id INT PRIMARY KEY AUTO_INCREMENT, first_name VARCHAR...	0 row(s) affected	0.047 sec

5) Orders Table :

The screenshot shows the MySQL Workbench interface. The 'Schemas' pane on the left lists various databases, with 'online_retail_store_database' selected. The 'Query' editor in the center contains the following SQL code:

```
1 CREATE TABLE Orders (  
2   order_id INT PRIMARY KEY AUTO_INCREMENT,  
3   customer_id INT,  
4   order_date DATETIME DEFAULT CURRENT_TIMESTAMP,  
5   total_price DECIMAL(10, 2),  
6   FOREIGN KEY (customer_id) REFERENCES Customers(customer_id)  
7 );
```

The 'Output' pane at the bottom displays the execution results:

#	Time	Action	Message	Duration / Fetch
1	13.23.18	create database Online_Retail_Store_Database	1 row(s) affected	0.031 sec
2	13.23.36	use Online_Retail_Store_Database	0 row(s) affected	0.016 sec
3	13.25.22	CREATE TABLE Products (product_id INT PRIMARY KEY AUTO_INCREMENT, product_name VARCHAR...	0 row(s) affected	0.125 sec
4	13.27.04	CREATE TABLE Customers (customer_id INT PRIMARY KEY AUTO_INCREMENT, first_name VARCHAR...	0 row(s) affected	0.047 sec
5	13.28.13	CREATE TABLE Customers (customer_id INT PRIMARY KEY AUTO_INCREMENT, first_name VARCHAR...	Error Code: 1050. Table 'customers' already exists	0.015 sec
6	13.28.31	CREATE TABLE Orders (order_id INT PRIMARY KEY AUTO_INCREMENT, customer_id INT, order_dat...	0 row(s) affected	0.078 sec

6) Order_Items Table :

The screenshot shows the MySQL Workbench interface. The 'Schemas' pane on the left lists various databases, with 'online_retail_store_database' selected. The 'Query' editor in the center contains the following SQL code:

```
1 CREATE TABLE Order_Items (  
2   order_id INT,  
3   product_id INT,  
4   quantity INT,  
5   price DECIMAL(10, 2),  
6   FOREIGN KEY (order_id) REFERENCES Orders(order_id),  
7   FOREIGN KEY (product_id) REFERENCES Products(product_id)  
8 );
```

The 'Output' pane at the bottom displays the execution results:

#	Time	Action	Message	Duration / Fetch
2	13.23.36	use Online_Retail_Store_Database	0 row(s) affected	0.016 sec
3	13.25.22	CREATE TABLE Products (product_id INT PRIMARY KEY AUTO_INCREMENT, product_name VARCH...	0 row(s) affected	0.125 sec
4	13.27.04	CREATE TABLE Customers (customer_id INT PRIMARY KEY AUTO_INCREMENT, first_name VARCHA...	0 row(s) affected	0.047 sec
5	13.28.13	CREATE TABLE Customers (customer_id INT PRIMARY KEY AUTO_INCREMENT, first_name VARCHA...	Error Code: 1050. Table 'customers' already exists	0.015 sec
6	13.28.31	CREATE TABLE Orders (order_id INT PRIMARY KEY AUTO_INCREMENT, customer_id INT, order_id...	0 row(s) affected	0.078 sec
7	13.29.20	CREATE TABLE Order_Items (order_item_id INT PRIMARY KEY AUTO_INCREMENT, order_id INT, pr...	0 row(s) affected	0.062 sec

7) Payments Table :

The screenshot shows the MySQL Workbench interface. The 'Schemas' pane on the left lists the 'online_retail_store_databases' schema. The 'Query' editor in the center contains the following SQL code:

```
1 CREATE TABLE Payments (  
2     payment_id INT PRIMARY KEY AUTO_INCREMENT,  
3     order_id INT,  
4     payment_date DATETIME DEFAULT CURRENT_TIMESTAMP,  
5     payment_method VARCHAR(50),  
6     amount_paid DECIMAL(10, 2),  
7     FOREIGN KEY (order_id) REFERENCES Orders(order_id)  
8 );
```

The 'Output' pane at the bottom shows the execution results:

#	Time	Action	Message	Duration / Fetch
3	13:25:22	CREATE TABLE Products (product_id INT PRIMARY KEY AUTO_INCREMENT, product_name VARCHAR(50), price DECIMAL(10, 2))	0 row(s) affected	0.125 sec
4	13:27:04	CREATE TABLE Customers (customer_id INT PRIMARY KEY AUTO_INCREMENT, first_name VARCHAR(50), last_name VARCHAR(50), email VARCHAR(100), phone VARCHAR(20), address VARCHAR(255), city VARCHAR(50), state VARCHAR(50), zip_code VARCHAR(10), country VARCHAR(50))	0 row(s) affected	0.047 sec
5	13:28:13	CREATE TABLE Customers (customer_id INT PRIMARY KEY AUTO_INCREMENT, first_name VARCHAR(50), last_name VARCHAR(50), email VARCHAR(100), phone VARCHAR(20), address VARCHAR(255), city VARCHAR(50), state VARCHAR(50), zip_code VARCHAR(10), country VARCHAR(50))	Error Code: 1050 Table 'customers' already exists	0.015 sec
6	13:28:31	CREATE TABLE Orders (order_id INT PRIMARY KEY AUTO_INCREMENT, customer_id INT, order_date DATETIME, order_status VARCHAR(20), order_items VARCHAR(255))	0 row(s) affected	0.078 sec

7) Inserting Customer's Data :

The screenshot shows the MySQL Workbench interface. The 'Query' editor in the center contains the following SQL code:

```
1 INSERT INTO Customers (first_name, last_name, email, phone, address, city, state, zip_code, country) VALUES  
2 ('John', 'Doe', 'john.doe@example.com', '123-456-7890', '123 Elm St', 'Springfield', 'IL', '62701', 'USA'),  
3 ('Jane', 'Smith', 'jane.smith@example.com', '234-567-8901', '456 Oak St', 'Springfield', 'IL', '62701', 'USA'),  
4 ('Alice', 'Johnson', 'alice.johnson@example.com', '345-678-9012', '789 Pine St', 'Springfield', 'IL', '62701', 'USA'),  
5 ('Bob', 'Brown', 'bob.brown@example.com', '456-789-0123', '321 Maple St', 'Springfield', 'IL', '62701', 'USA'),  
6 ('Charlie', 'Davis', 'charlie.davis@example.com', '567-890-1234', '654 Cedar St', 'Springfield', 'IL', '62701', 'USA'),  
7 ('Diana', 'Wilson', 'diana.wilson@example.com', '678-901-2345', '987 Birch St', 'Springfield', 'IL', '62701', 'USA'),  
8 ('Ethan', 'Martinez', 'ethan.martinez@example.com', '789-012-3456', '159 Spruce St', 'Springfield', 'IL', '62701', 'USA'),  
9 ('Fiona', 'Garcia', 'fiona.garcia@example.com', '890-123-4567', '753 Willow St', 'Springfield', 'IL', '62701', 'USA'),  
10 ('George', 'Hernandez', 'george.hernandez@example.com', '901-234-5678', '852 Fir St', 'Springfield', 'IL', '62701', 'USA'),  
11 ('Hannah', 'Lopez', 'hannah.lopez@example.com', '012-345-6789', '456 Ash St', 'Springfield', 'IL', '62701', 'USA');
```

The 'Output' pane at the bottom shows the execution results:

#	Time	Action	Message	Duration / Fetch
4	13:27:04	CREATE TABLE Customers (customer_id INT PRIMARY KEY AUTO_INCREMENT, first_name VARCHAR(50), last_name VARCHAR(50), email VARCHAR(100), phone VARCHAR(20), address VARCHAR(255), city VARCHAR(50), state VARCHAR(50), zip_code VARCHAR(10), country VARCHAR(50))	0 row(s) affected	0.047 sec
5	13:28:13	CREATE TABLE Customers (customer_id INT PRIMARY KEY AUTO_INCREMENT, first_name VARCHAR(50), last_name VARCHAR(50), email VARCHAR(100), phone VARCHAR(20), address VARCHAR(255), city VARCHAR(50), state VARCHAR(50), zip_code VARCHAR(10), country VARCHAR(50))	Error Code: 1050 Table 'customers' already exists	0.015 sec
6	13:28:31	CREATE TABLE Orders (order_id INT PRIMARY KEY AUTO_INCREMENT, customer_id INT, order_date DATETIME, order_status VARCHAR(20), order_items VARCHAR(255))	0 row(s) affected	0.078 sec
7	13:29:20	CREATE TABLE Order_Items (order_item_id INT PRIMARY KEY AUTO_INCREMENT, order_id INT, product_id INT, quantity INT, price DECIMAL(10, 2))	0 row(s) affected	0.052 sec
8	13:30:24	CREATE TABLE Payments (payment_id INT PRIMARY KEY AUTO_INCREMENT, order_id INT, payment_date DATETIME, payment_method VARCHAR(50), amount_paid DECIMAL(10, 2))	0 row(s) affected	0.063 sec
9	13:31:45	INSERT INTO Customers (first_name, last_name, email, phone, address, city, state, zip_code, country) VALUES ('John', 'Doe', 'john.doe@example.com', '123-456-7890', '123 Elm St', 'Springfield', 'IL', '62701', 'USA'), ('Jane', 'Smith', 'jane.smith@example.com', '234-567-8901', '456 Oak St', 'Springfield', 'IL', '62701', 'USA'), ('Alice', 'Johnson', 'alice.johnson@example.com', '345-678-9012', '789 Pine St', 'Springfield', 'IL', '62701', 'USA'), ('Bob', 'Brown', 'bob.brown@example.com', '456-789-0123', '321 Maple St', 'Springfield', 'IL', '62701', 'USA'), ('Charlie', 'Davis', 'charlie.davis@example.com', '567-890-1234', '654 Cedar St', 'Springfield', 'IL', '62701', 'USA'), ('Diana', 'Wilson', 'diana.wilson@example.com', '678-901-2345', '987 Birch St', 'Springfield', 'IL', '62701', 'USA'), ('Ethan', 'Martinez', 'ethan.martinez@example.com', '789-012-3456', '159 Spruce St', 'Springfield', 'IL', '62701', 'USA'), ('Fiona', 'Garcia', 'fiona.garcia@example.com', '890-123-4567', '753 Willow St', 'Springfield', 'IL', '62701', 'USA'), ('George', 'Hernandez', 'george.hernandez@example.com', '901-234-5678', '852 Fir St', 'Springfield', 'IL', '62701', 'USA'), ('Hannah', 'Lopez', 'hannah.lopez@example.com', '012-345-6789', '456 Ash St', 'Springfield', 'IL', '62701', 'USA');	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.016 sec

8) Inserting Products Data's:

The screenshot shows the MySQL Workbench interface with the 'online_retail_store_database' schema selected. The 'Query' tab is active, displaying an SQL script to insert product data into the 'Products' table. The script consists of 11 lines, each inserting a new product record with details like product name, category, description, price, and quantity in stock.

```
1 INSERT INTO Products (product_name, category, description, price, quantity_in_stock) VALUES
2 ('Laptop', 'Electronics', '15-inch laptop with 16GB RAM', 999.99, 50),
3 ('Smartphone', 'Electronics', 'Latest model smartphone with 128GB storage', 799.99, 100),
4 ('Headphones', 'Electronics', 'Noise-cancelling over-ear headphones', 199.99, 75),
5 ('Coffee Maker', 'Home Appliances', 'Automatic coffee maker with grinder', 89.99, 30),
6 ('Blender', 'Home Appliances', 'High-speed blender for smoothies', 49.99, 40),
7 ('Desk Chair', 'Furniture', 'Ergonomic office chair', 129.99, 20),
8 ('Gaming Console', 'Electronics', 'Latest gaming console', 499.99, 25),
9 ('Wireless Mouse', 'Electronics', 'Ergonomic wireless mouse', 29.99, 60),
10 ('LED Monitor', 'Electronics', '27-inch LED monitor', 299.99, 35),
11 ('Smartwatch', 'Electronics', 'Fitness smartwatch with heart rate monitor', 199.99, 80);
```

The 'Output' tab at the bottom shows the execution results of the queries. It includes a table with columns for Time, Action, Message, and Duration / Fetch. The messages indicate that the tables 'Customers', 'Orders', 'Order_Items', and 'Payments' were created successfully, and the 'Products' table was populated with 10 rows.

#	Time	Action	Message	Duration / Fetch
5	13:28:13	CREATE TABLE Customers (customer_id INT PRIMARY KEY AUTO_INCREMENT, first_name VARCHAR(50), last_name VARCHAR(50), email VARCHAR(100), phone VARCHAR(20), address VARCHAR(100), city VARCHAR(20), state VARCHAR(20), zip_code VARCHAR(10), country VARCHAR(20))	Error Code: 1050 Table 'customers' already exists	0.015 sec
6	13:28:31	CREATE TABLE Orders (order_id INT PRIMARY KEY AUTO_INCREMENT, customer_id INT, order_date DATE, order_status VARCHAR(20), order_items TEXT)	0 row(s) affected	0.078 sec
7	13:29:20	CREATE TABLE Order_Items (order_item_id INT PRIMARY KEY AUTO_INCREMENT, order_id INT, product_id INT, quantity INT, price DECIMAL(10,2))	0 row(s) affected	0.062 sec
8	13:30:24	CREATE TABLE Payments (payment_id INT PRIMARY KEY AUTO_INCREMENT, order_id INT, payment_date DATE, amount DECIMAL(10,2))	0 row(s) affected	0.063 sec

9) Table view of customer's data:

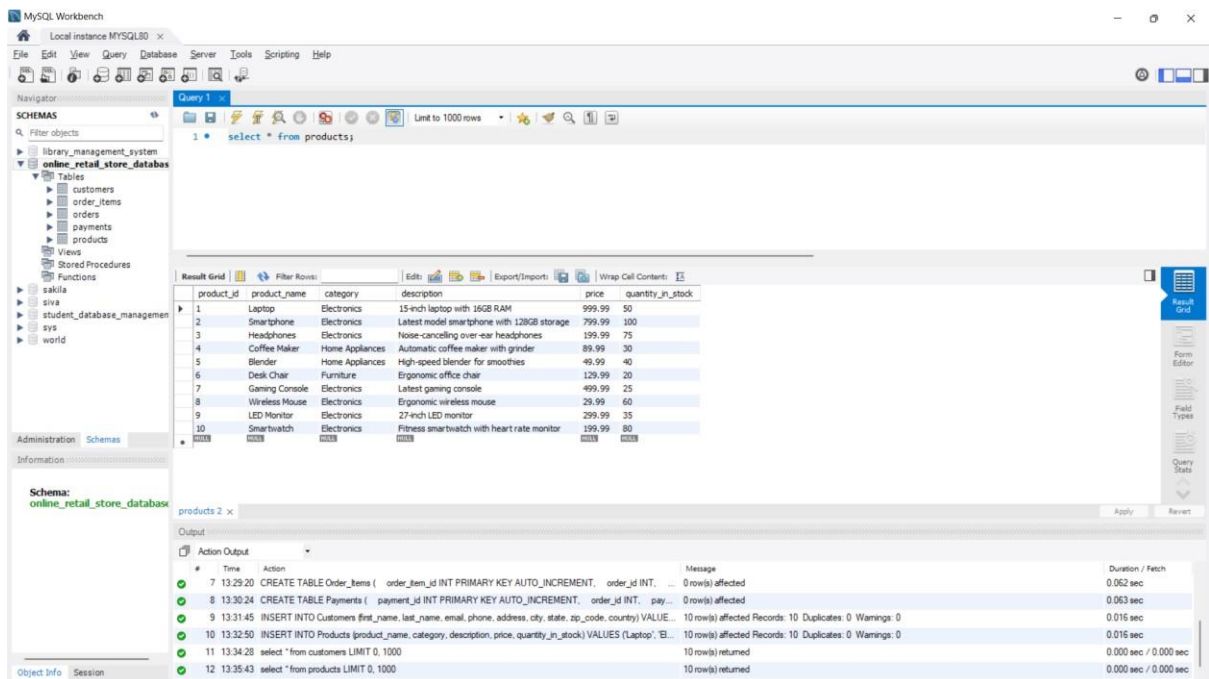
The screenshot shows the MySQL Workbench interface with the 'online_retail_store_database' schema selected. The 'Query' tab is active, displaying the SQL query 'select * from customers;'. The 'Result Grid' tab is also active, showing the data for the 'customers' table.

customer_id	first_name	last_name	email	phone	address	city	state	zip_code	country
1	John	Doe	john.doe@example.com	123-456-7890	123 Elm St	Springfield	IL	62701	USA
2	Jane	Smith	jane.smith@example.com	234-567-8901	456 Oak St	Springfield	IL	62701	USA
3	Alice	Johnson	alice.johnson@example.com	345-678-9012	789 Pine St	Springfield	IL	62701	USA
4	Bob	Brown	bob.brown@example.com	456-789-0123	321 Maple St	Springfield	IL	62701	USA
5	Charlie	Davis	charlie.davis@example.com	567-890-1234	654 Cedar St	Springfield	IL	62701	USA
6	Diana	Wilson	diana.wilson@example.com	678-901-2345	987 Birch St	Springfield	IL	62701	USA
7	Ethan	Martinez	ethan.martinez@example.com	789-012-3456	159 Spruce St	Springfield	IL	62701	USA
8	Fiona	Garcia	fiona.garcia@example.com	890-123-4567	753 Hawthorn St	Springfield	IL	62701	USA
9	George	Hernandez	george.hernandez@example.com	901-234-5678	852 Fir St	Springfield	IL	62701	USA
10	Hannah	Lopez	hannah.lopez@example.com	012-345-6789	456 Ash St	Springfield	IL	62701	USA

The 'Output' tab at the bottom shows the execution results of the queries. It includes a table with columns for Time, Action, Message, and Duration / Fetch. The messages indicate that the tables 'Orders', 'Order_Items', and 'Payments' were created successfully, and the 'Products' table was populated with 10 rows. The final query 'select * from customers;' returned 10 rows.

#	Time	Action	Message	Duration / Fetch
6	13:28:31	CREATE TABLE Orders (order_id INT PRIMARY KEY AUTO_INCREMENT, customer_id INT, order_date DATE, order_status VARCHAR(20), order_items TEXT)	0 row(s) affected	0.078 sec
7	13:29:20	CREATE TABLE Order_Items (order_item_id INT PRIMARY KEY AUTO_INCREMENT, order_id INT, product_id INT, quantity INT, price DECIMAL(10,2))	0 row(s) affected	0.062 sec
8	13:30:24	CREATE TABLE Payments (payment_id INT PRIMARY KEY AUTO_INCREMENT, order_id INT, payment_date DATE, amount DECIMAL(10,2))	0 row(s) affected	0.063 sec
9	13:31:45	INSERT INTO Customers (first_name, last_name, email, phone, address, city, state, zip_code, country) VALUES ('John', 'Doe', 'john.doe@example.com', '123-456-7890', '123 Elm St', 'Springfield', 'IL', '62701', 'USA')	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.015 sec
10	13:32:50	INSERT INTO Products (product_name, category, description, price, quantity_in_stock) VALUES ('Laptop', 'Electronics', '15-inch laptop with 16GB RAM', 999.99, 50)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.015 sec
11	13:34:28	select * from customers LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec

10) Table view of Product data:



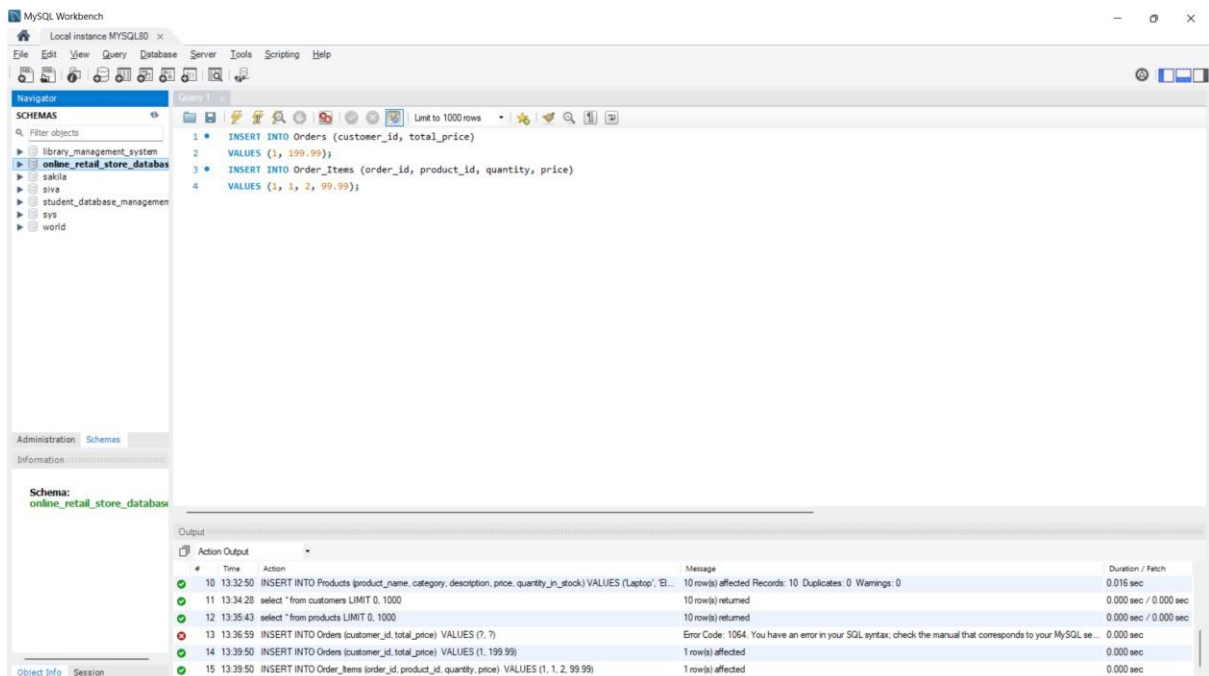
The screenshot displays the MySQL Workbench interface. On the left, the 'Schemas' pane shows the 'online_retail_store_database' selected. The main window shows the 'products' table with the following data:

product_id	product_name	category	description	price	quantity_in_stock
1	Laptop	Electronics	15-inch laptop with 16GB RAM	999.99	50
2	Smartphone	Electronics	Latest model smartphone with 128GB storage	799.99	100
3	Headphones	Electronics	Noise-canceling over-ear headphones	199.99	75
4	Coffee Maker	Home Appliances	Automatic coffee maker with grinder	89.99	30
5	Blender	Home Appliances	High-speed blender for smoothies	49.99	40
6	Desk Chair	Furniture	Ergonomic office chair	129.99	20
7	Gaming Console	Electronics	Latest gaming console	499.99	25
8	Wireless Mouse	Electronics	Ergonomic wireless mouse	29.99	60
9	LED Monitor	Electronics	27-inch LED monitor	299.99	35
10	Smartwatch	Electronics	Fitness smartwatch with heart rate monitor	199.99	80

The bottom pane shows the 'Action Output' for the query, indicating that 10 rows were returned.

SQL Queries for Customer Orders and Payment Processing :

1) Placing an Order :



The screenshot displays the MySQL Workbench interface. The 'Schemas' pane shows the 'online_retail_store_database' selected. The main window shows a SQL query with the following statements:

```
1. INSERT INTO Orders (customer_id, total_price)
2. VALUES (1, 199.99);
3. INSERT INTO Order_Items (order_id, product_id, quantity, price)
4. VALUES (1, 1, 2, 99.99);
```

The bottom pane shows the 'Action Output' for the query. The output indicates that 1 row was affected for the first two statements, but an error occurred for the third statement (Error Code: 1064). The error message is: "You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'VALUES (1, 1, 2, 99.99)' at line 4".

2) Processing Payment :

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' list with 'online_retail_store_database' selected. The main query editor contains the following SQL code:

```
1 INSERT INTO Payments (order_id, payment_method, amount_paid)
2 VALUES (1, 'Credit Card', 199.99);
```

The 'Output' tab at the bottom shows the execution results:

#	Time	Action	Message	Duration / Fetch
11	13:34:28	select * from customers LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
12	13:35:43	select * from products LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
13	13:36:59	INSERT INTO Orders (customer_id, total_price) VALUES (1, 199.99)	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL se...	0.000 sec
14	13:39:50	INSERT INTO Orders (customer_id, total_price) VALUES (1, 199.99)	1 row(s) affected	0.000 sec
15	13:39:50	INSERT INTO Order_Items (order_id, product_id, quantity, price) VALUES (1, 1, 2, 99.99)	1 row(s) affected	0.000 sec
16	13:41:04	INSERT INTO Payments (order_id, payment_method, amount_paid) VALUES (1, 'Credit Card', 199.99)	1 row(s) affected	0.000 sec

3) Viewing Customer Order History :

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' list with 'online_retail_store_database' selected. The main query editor contains the following SQL code:

```
1 SELECT o.order_id, o.order_date, oi.product_id, p.product_name, oi.quantity, oi.price
2 FROM Orders o
3 JOIN Order_Items oi ON o.order_id = oi.order_id
4 JOIN Products p ON oi.product_id = p.product_id
5 WHERE o.customer_id = 1; -- Replace with the actual customer_id
```

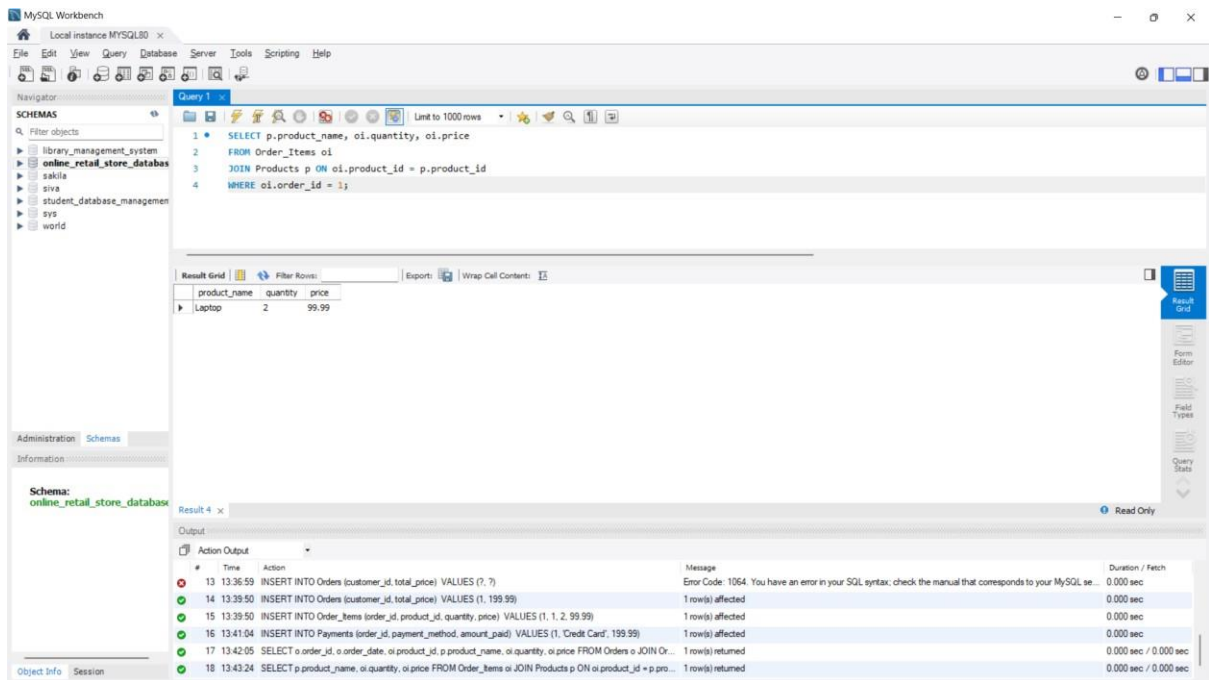
The 'Result Grid' tab at the bottom shows the execution results:

order_id	order_date	product_id	product_name	quantity	price
1	2024-08-11 13:39:50	1	Laptop	2	99.99

The 'Output' tab at the bottom shows the execution results:

#	Time	Action	Message	Duration / Fetch
12	13:35:43	select * from products LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
13	13:36:59	INSERT INTO Orders (customer_id, total_price) VALUES (1, 199.99)	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL se...	0.000 sec
14	13:39:50	INSERT INTO Orders (customer_id, total_price) VALUES (1, 199.99)	1 row(s) affected	0.000 sec
15	13:39:50	INSERT INTO Order_Items (order_id, product_id, quantity, price) VALUES (1, 1, 2, 99.99)	1 row(s) affected	0.000 sec
16	13:41:04	INSERT INTO Payments (order_id, payment_method, amount_paid) VALUES (1, 'Credit Card', 199.99)	1 row(s) affected	0.000 sec
17	13:42:05	SELECT o.order_id, o.order_date, oi.product_id, p.product_name, oi.quantity, oi.price FROM Orders o JOIN Or...	1 row(s) returned	0.000 sec / 0.000 sec

4) Retrieve All Products in an Order :



The screenshot shows the MySQL Workbench interface. The 'Query' tab is active, displaying a SQL query that selects product names, quantities, and prices from the 'Order_Items' table, joined with the 'Products' table. The query is as follows:

```
1 SELECT p.product_name, oi.quantity, oi.price
2 FROM Order_Items oi
3 JOIN Products p ON oi.product_id = p.product_id
4 WHERE oi.order_id = 1;
```

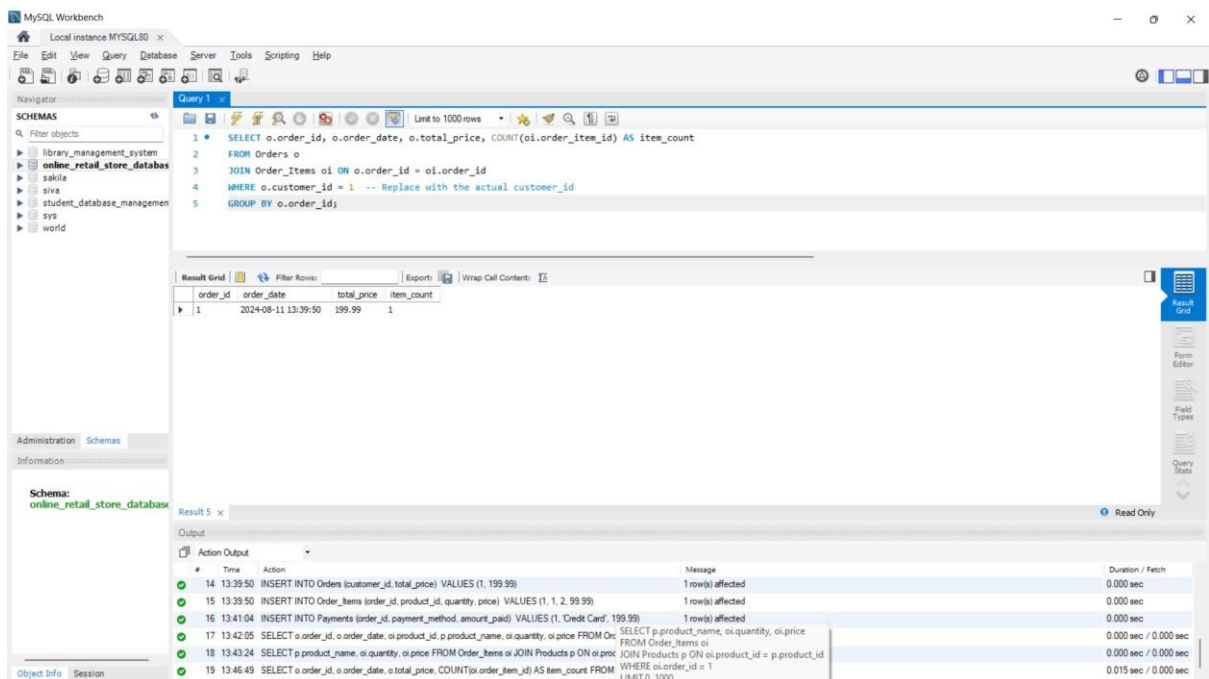
The 'Result Grid' shows the output of the query, which is a single row for the product 'Laptop' with a quantity of 2 and a price of 99.99.

product_name	quantity	price
Laptop	2	99.99

The 'Output' tab shows the execution log, including the time taken for each step and the number of rows affected or returned.

#	Time	Action	Message	Duration / Fetch
13	13:36:59	INSERT INTO Orders (customer_id, total_price) VALUES (1, 7)	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL se...	0.000 sec
14	13:39:50	INSERT INTO Orders (customer_id, total_price) VALUES (1, 199.99)	1 row(s) affected	0.000 sec
15	13:39:50	INSERT INTO Order_Items (order_id, product_id, quantity, price) VALUES (1, 1, 2, 99.99)	1 row(s) affected	0.000 sec
16	13:41:04	INSERT INTO Payments (order_id, payment_method, amount_paid) VALUES (1, 'Credit Card', 199.99)	1 row(s) affected	0.000 sec
17	13:42:05	SELECT o.order_id, o.order_date, oi.product_id, p.product_name, oi.quantity, oi.price FROM Orders o JOIN Or...	1 row(s) returned	0.000 sec / 0.000 sec
18	13:43:24	SELECT p.product_name, oi.quantity, oi.price FROM Order_Items oi JOIN Products p ON oi.product_id = p.pro...	1 row(s) returned	0.000 sec / 0.000 sec

5) Retrieve All Orders for a Specific Customer :



The screenshot shows the MySQL Workbench interface. The 'Query' tab is active, displaying a SQL query that selects order IDs, order dates, total prices, and item counts for a specific customer (customer_id = 1). The query is as follows:

```
1 SELECT o.order_id, o.order_date, o.total_price, COUNT(oi.order_item_id) AS item_count
2 FROM Orders o
3 JOIN Order_Items oi ON o.order_id = oi.order_id
4 WHERE o.customer_id = 1 -- Replace with the actual customer_id
5 GROUP BY o.order_id;
```

The 'Result Grid' shows the output of the query, which is a single row for the order with ID 1, dated 2024-08-11 13:39:50, with a total price of 199.99 and an item count of 1.

order_id	order_date	total_price	item_count
1	2024-08-11 13:39:50	199.99	1

The 'Output' tab shows the execution log, including the time taken for each step and the number of rows affected or returned.

#	Time	Action	Message	Duration / Fetch
14	13:39:50	INSERT INTO Orders (customer_id, total_price) VALUES (1, 199.99)	1 row(s) affected	0.000 sec
15	13:39:50	INSERT INTO Order_Items (order_id, product_id, quantity, price) VALUES (1, 1, 2, 99.99)	1 row(s) affected	0.000 sec
16	13:41:04	INSERT INTO Payments (order_id, payment_method, amount_paid) VALUES (1, 'Credit Card', 199.99)	1 row(s) affected	0.000 sec
17	13:42:05	SELECT o.order_id, o.order_date, oi.product_id, p.product_name, oi.quantity, oi.price FROM Orders o...	1 row(s) returned	0.000 sec / 0.000 sec
18	13:43:24	SELECT p.product_name, oi.quantity, oi.price FROM Order_Items oi JOIN Products p ON oi.product_id = p.pro...	1 row(s) returned	0.000 sec / 0.000 sec
19	13:46:49	SELECT o.order_id, o.order_date, o.total_price, COUNT(oi.order_item_id) AS item_count FROM ...	1 row(s) returned	0.015 sec / 0.000 sec

Conclusion:

Thus, the SQL queries offer a robust set of tools for efficiently managing customer orders and payment processing in an online retail store database. By implementing these queries, retailers can streamline their operations, improve customer satisfaction, and gain valuable insights into their business performance.