

Ecommerce – SQL

1.

1) update product_table set price = 800 where product_id = 7;

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL statements:

```
55 • insert into cart values (8, 6, 10, 2);
56 • insert into cart values (9, 6, 9, 3);
57 • insert into cart values (10, 7, 7, 2);
58
59 • update product_table set price = 800 where product_id = 7;
60
61 • select * from product_table;
```

The Result Grid displays the data from the `product_table`:

| product_id | prod_name | price | description | stock_quan |
|------------|----------------|--------|-------------------------|------------|
| 1 | Laptop | 800.00 | High-performance laptop | 10 |
| 2 | Smartphone | 600.00 | Latest smartphone | 15 |
| 3 | Tablet | 300.00 | Portable tablet | 20 |
| 4 | Headphones | 150.00 | Noise-canceling | 30 |
| 5 | TV | 900.00 | 4K Smart TV | 5 |
| 6 | Coffee Maker | 50.00 | Automatic coffee maker | 25 |
| 7 | Refrigerator | 800.00 | Energy-efficient | 10 |
| 8 | Microwave Oven | 80.00 | Countertop microwave | 15 |
| 9 | Blender | 70.00 | High-speed blender | 20 |
| 10 | Vacuum Cleaner | 120.00 | Bagless vacuum cleaner | 10 |

2.

2) delete from cart where cart_id in (select cart_id from cart where customer_id = 5);

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL statements:

```
59 • update product_table set price = 800 where product_id = 7;
60
61 • select * from product_table;
62 • delete from cart where cart_id in (
63   select cart_id from cart where customer_id = 5
64 );
65 • select * from cart;
```

The Result Grid displays the data from the `cart` table:

| cart_id | customer_id | product_id | quantity |
|---------|-------------|------------|----------|
| 1 | 1 | 1 | 2 |
| 2 | 1 | 3 | 1 |
| 3 | 2 | 2 | 3 |
| 4 | 3 | 4 | 4 |
| 5 | 3 | 5 | 2 |
| 6 | 4 | 6 | 1 |
| 7 | 5 | 1 | 1 |
| 8 | 6 | 10 | 2 |
| 9 | 6 | 9 | 3 |
| 10 | 7 | 7 | 2 |

3.

3) select * from product_table where price < 100;

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
61 • select * from product_table;
62 • delete from cart where cart_id in (
63   select cart_id from cart where customer_id = 5
64 );
65 • select * from cart;
66
67 • select * from product_table where price < 100;
```

The Results Grid displays the following data:

| product_id | prod_name | price | description | stock_quan |
|------------|----------------|-------|------------------------|------------|
| 6 | Coffee Maker | 50.00 | Automatic coffee maker | 25 |
| 8 | Microwave Oven | 80.00 | Countertop microwave | 15 |
| 8 | Blender | 70.00 | High-speed blender | 20 |
| | NULL | NULL | NULL | NULL |

4.

4)select * from product_table where stock_quan > 5;

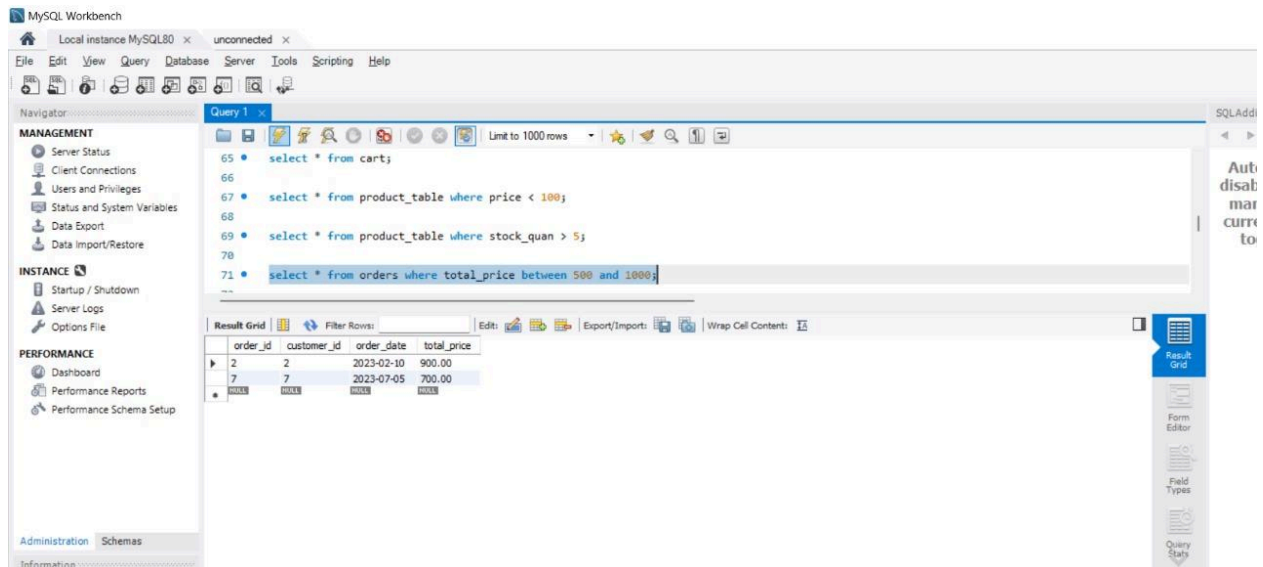
The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
63   select cart_id from cart where customer_id = 5
64 );
65 • select * from cart;
66
67 • select * from product_table where price < 100;
68
69 • select * from product_table where stock_quan > 5;
```

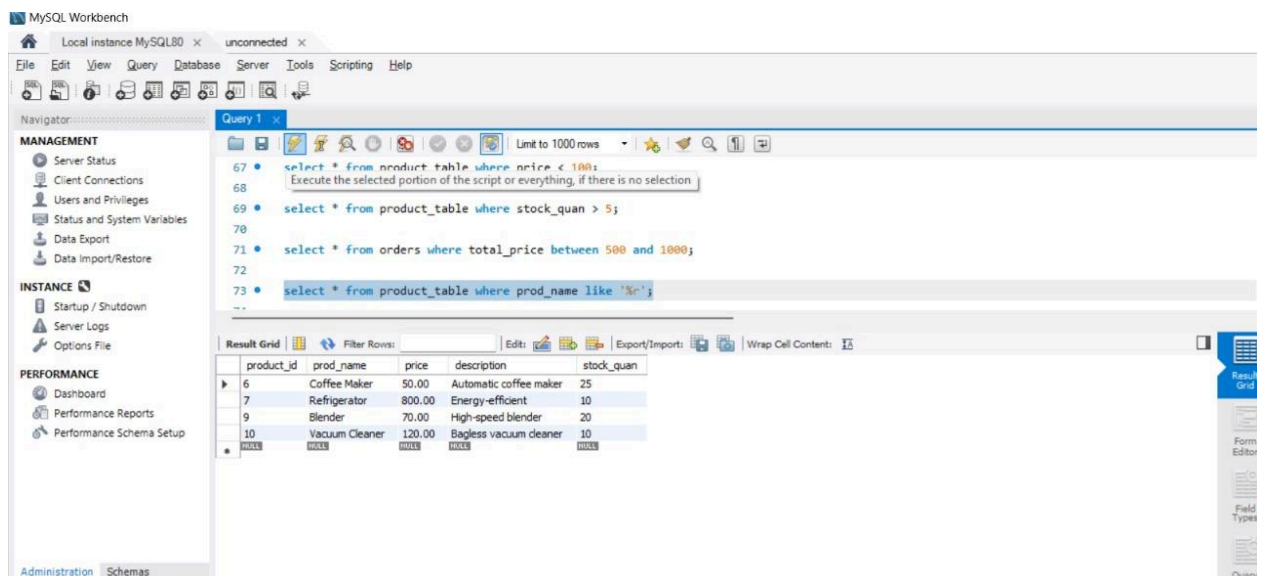
The Results Grid displays the following data:

| product_id | prod_name | price | description | stock_quan |
|------------|----------------|--------|-------------------------|------------|
| 1 | Laptop | 800.00 | High-performance laptop | 10 |
| 2 | Smartphone | 600.00 | Latest smartphone | 15 |
| 3 | Tablet | 300.00 | Portable tablet | 20 |
| 4 | Headphones | 150.00 | Noise-canceling | 30 |
| 6 | Coffee Maker | 50.00 | Automatic coffee maker | 25 |
| 7 | Refrigerator | 800.00 | Energy-efficient | 10 |
| 8 | Microwave Oven | 80.00 | Countertop microwave | 15 |
| 9 | Blender | 70.00 | High-speed blender | 20 |
| 10 | Vacuum Cleaner | 120.00 | Bagless vacuum cleaner | 10 |
| | NULL | NULL | NULL | NULL |

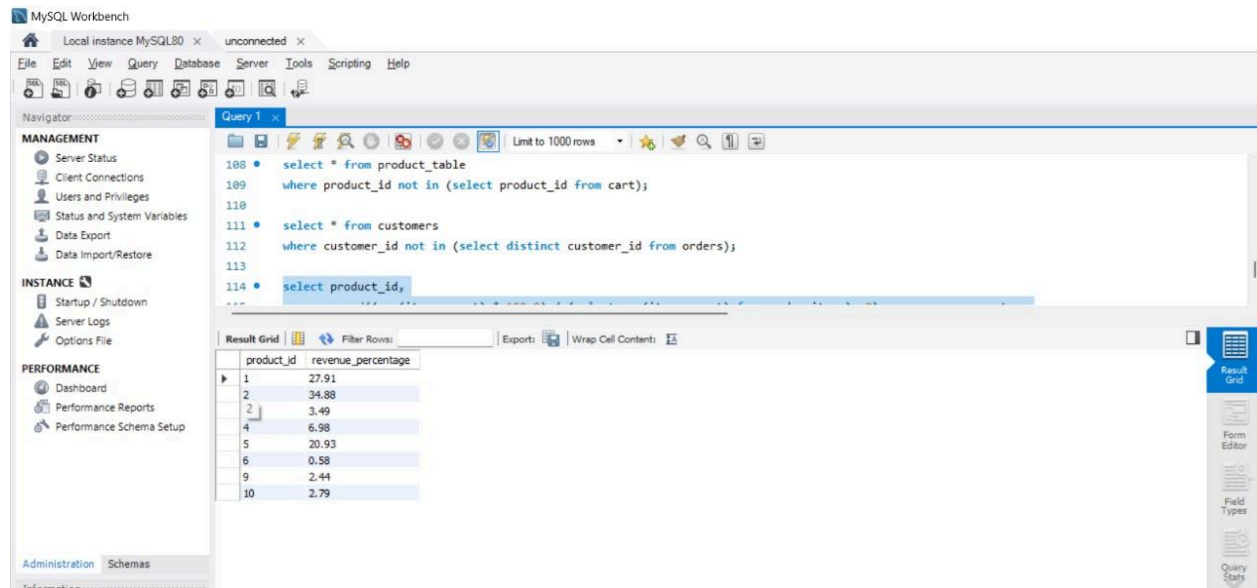
5) `select * from orders where total_price between 500 and 1000;`



6) `select * from product_table where prod_name like '%r';`



7) **select * from cart where customer_id = 5;**



The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

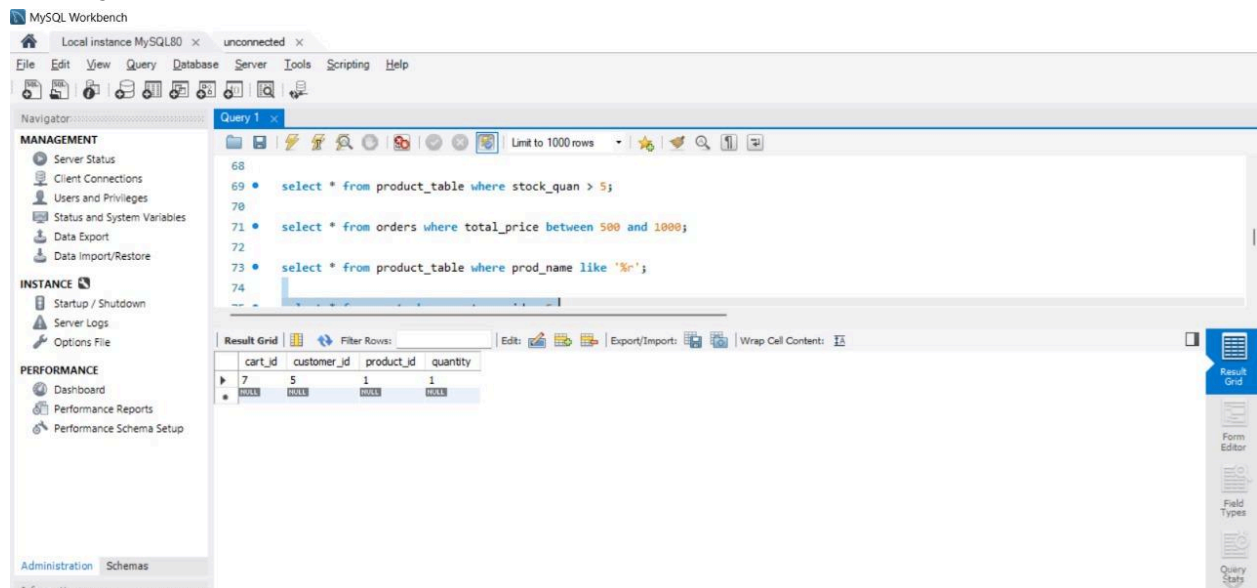
```
108 • select * from product_table
109   where product_id not in (select product_id from cart);
110
111 • select * from customers
112   where customer_id not in (select distinct customer_id from orders);
113
114 • select product_id,
```

The result grid displays the following data:

| product_id | revenue_percentage |
|------------|--------------------|
| 1 | 27.91 |
| 2 | 34.88 |
| 2 | 3.49 |
| 4 | 6.98 |
| 5 | 20.93 |
| 6 | 0.58 |
| 9 | 2.44 |
| 10 | 2.79 |

8.

8) **select distinct c.* from customers c join orders o on c.customer_id = o.customer_id where year(o.order_date) = 2023;**



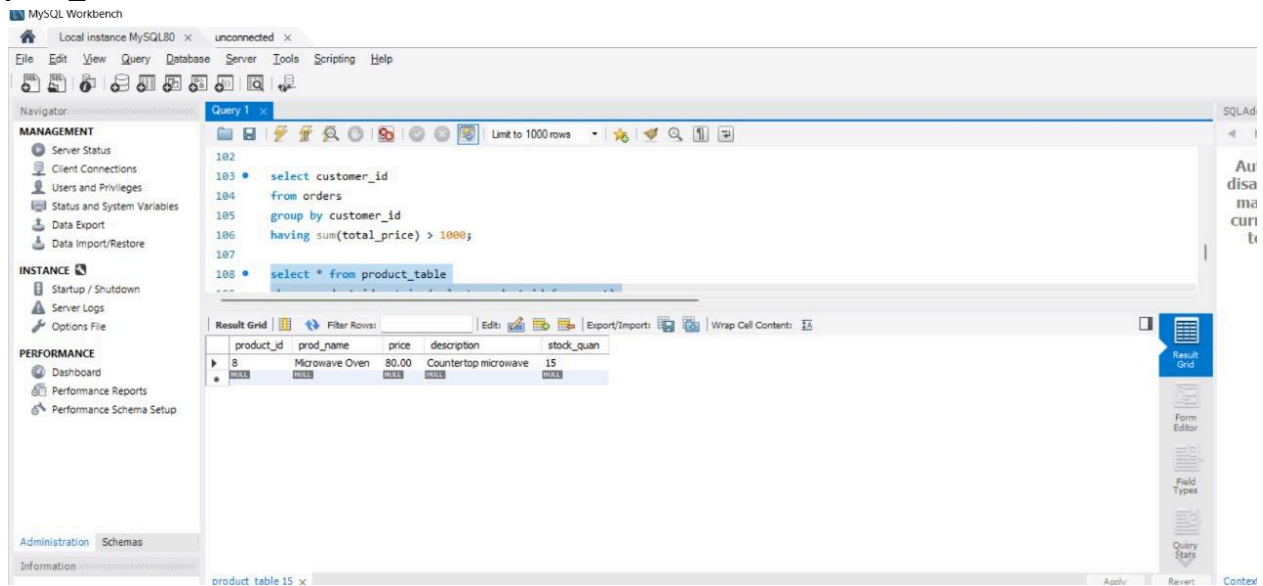
The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
68
69 • select * from product_table where stock_quan > 5;
70
71 • select * from orders where total_price between 500 and 1000;
72
73 • select * from product_table where prod_name like '%r';
74
```

The result grid displays the following data:

| cart_id | customer_id | product_id | quantity |
|---------|-------------|------------|----------|
| 7 | 5 | 1 | 1 |
| • | NULL | NULL | NULL |

9) select prod_name, min(stock_quan) as min_quantity from product_table group by prod_name;



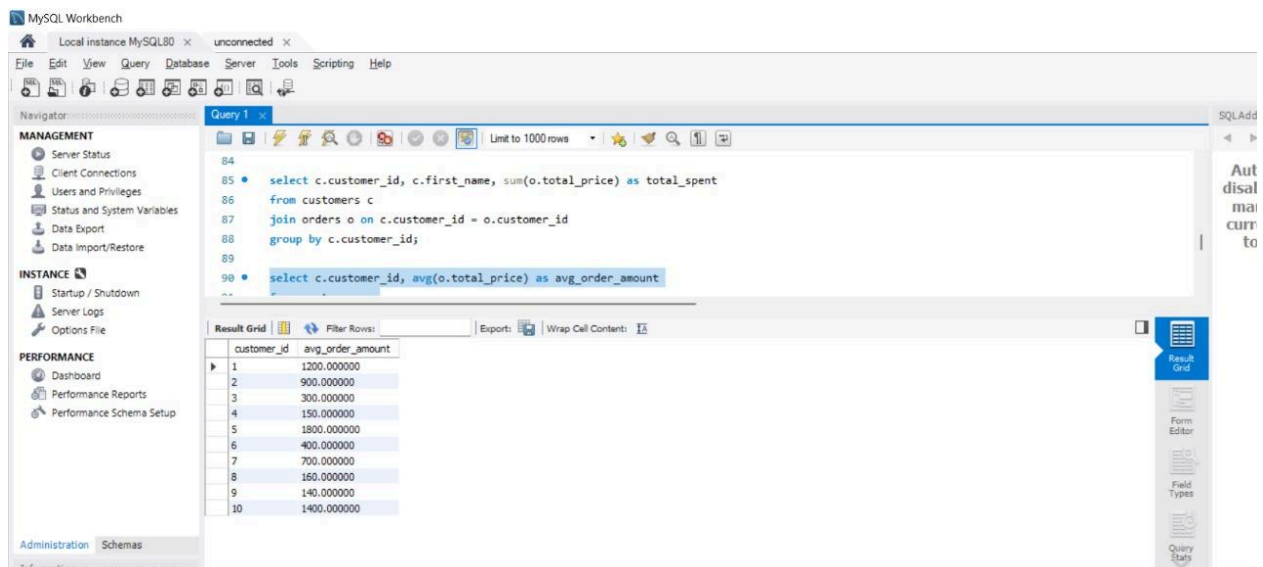
The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
102
103 • select customer_id
104   from orders
105   group by customer_id
106   having sum(total_price) > 1000;
107
108 • select * from product_table
```

The Result Grid displays the following data:

| product_id | prod_name | price | description | stock_quan |
|------------|----------------|-------|----------------------|------------|
| 8 | Microwave Oven | 80.00 | Countertop microwave | 15 |

10) select c.customer_id, c.first_name, sum(o.total_price) as total_spenfrom customers c join orders o on c.customer_id = o.customer_id group by c.customer_id;



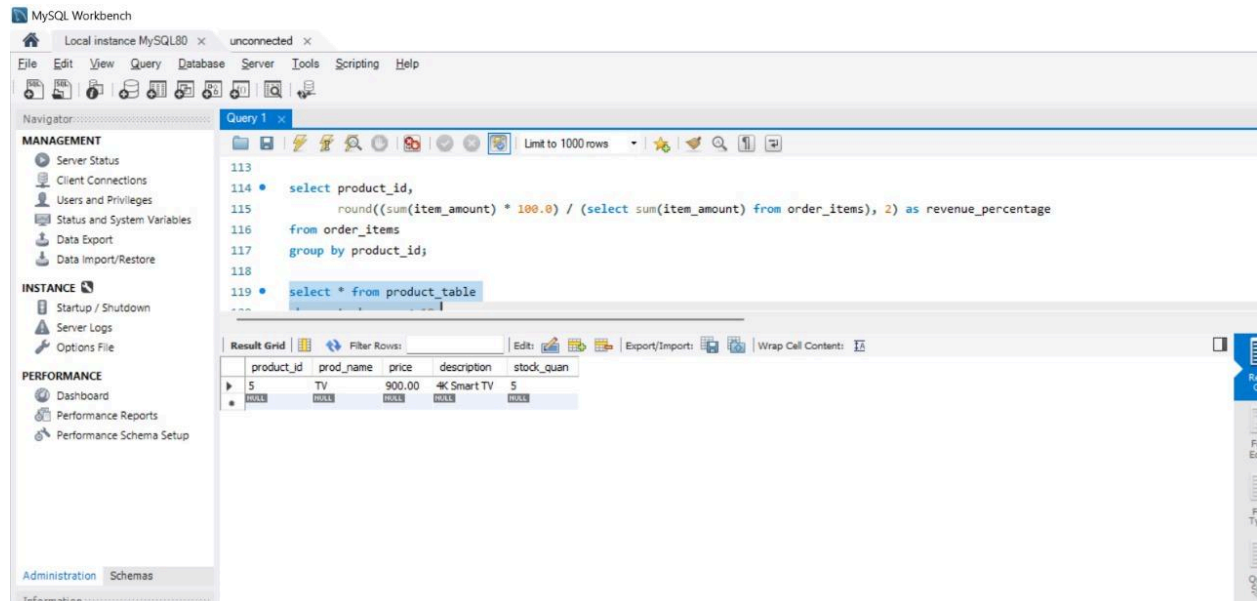
The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
84
85 • select c.customer_id, c.first_name, sum(o.total_price) as total_spent
86   from customers c
87   join orders o on c.customer_id = o.customer_id
88   group by c.customer_id;
89
90 • select c.customer_id, avg(o.total_price) as avg_order_amount
```

The Result Grid displays the following data:

| customer_id | avg_order_amount |
|-------------|------------------|
| 1 | 1200.000000 |
| 2 | 900.000000 |
| 3 | 300.000000 |
| 4 | 150.000000 |
| 5 | 1800.000000 |
| 6 | 400.000000 |
| 7 | 700.000000 |
| 8 | 160.000000 |
| 9 | 140.000000 |
| 10 | 1400.000000 |

11) select c.customer_id, avg(o.total_price) as avg_order_amount from customers c join orders o on c.customer_id = o.customer_id group by c.customer_id;



The screenshot shows the MySQL Workbench interface. The left sidebar contains the 'MANAGEMENT' and 'PERFORMANCE' sections. The main editor displays a SQL query:

```

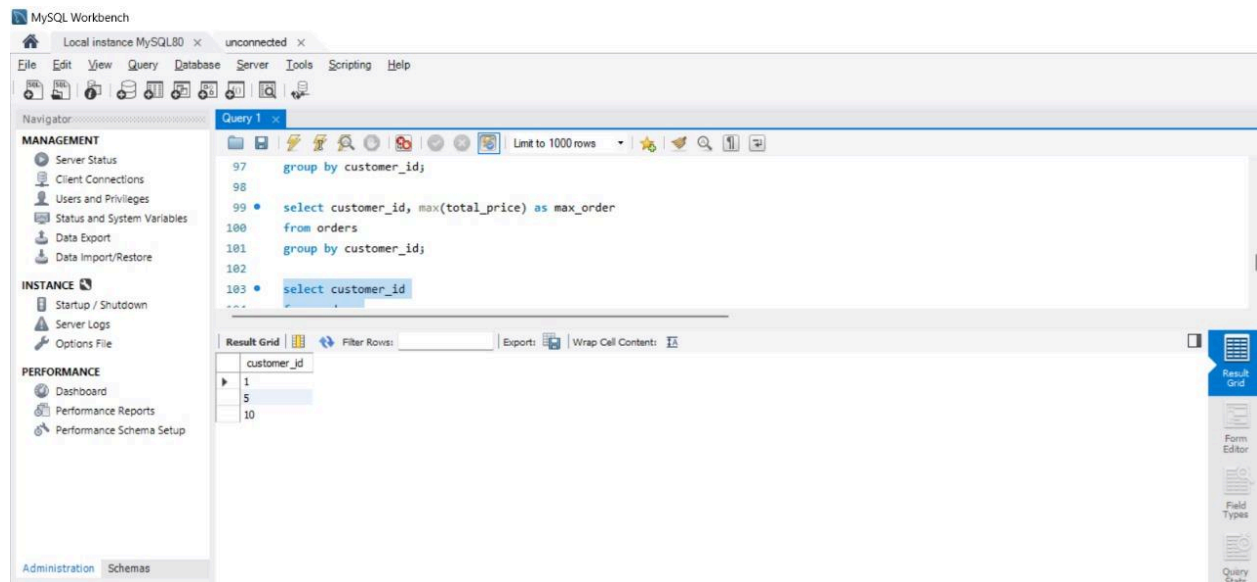
113
114 • select product_id,
115       round((sum(item_amount) * 100.0) / (select sum(item_amount) from order_items), 2) as revenue_percentage
116   from order_items
117  group by product_id;
118
119 • select * from product_table

```

The 'Result Grid' at the bottom shows the following data:

| product_id | prod_name | price | description | stock_quantity |
|------------|-----------|--------|-------------|----------------|
| 5 | TV | 900.00 | 4K Smart TV | 5 |

12) select customer_id, count(*) as order_count from orders group by customer_id;



The screenshot shows the MySQL Workbench interface. The left sidebar contains the 'MANAGEMENT' and 'PERFORMANCE' sections. The main editor displays a SQL query:

```

97 group by customer_id;
98
99 • select customer_id, max(total_price) as max_order
100   from orders
101  group by customer_id;
102
103 • select customer_id

```

The 'Result Grid' at the bottom shows the following data:

| customer_id |
|-------------|
| 1 |
| 5 |
| 10 |

13) select customer_id, max(total_price) as max_order from orders group by customer_id;

MySQL Workbench

Local instance MySQL80 x unconnected x

File Edit View Query Database Server Tools Scripting Help

Navigator Open Inspector for the selected object

MANAGEMENT Open Inspector for the selected object

Server Status

Client Connections

Users and Privileges

Status and System Variables

Data Export

Data Import/Restore

INSTANCE

Startup / Shutdown

Server Logs

Options File

PERFORMANCE

Dashboard

Performance Reports

Performance Schema Setup

Administration Schemas

Query 1

```

89
90 • select c.customer_id, avg(o.total_price) as avg_order_amount
91   from customers c
92  join orders o on c.customer_id = o.customer_id
93  group by c.customer_id;
94
95 • select customer_id, count(*) as order_count
96

```

Result Grid

| customer_id | order_count |
|-------------|-------------|
| 1 | 1 |
| 2 | 1 |
| 3 | 1 |
| 4 | 1 |
| 5 | 1 |
| 6 | 1 |
| 7 | 1 |
| 8 | 1 |
| 9 | 1 |
| 10 | 1 |

Result Grid

Form Editor

Field Types

Query Status

14.

14)select customer_id from orders group by customer_id having sum(total_price) > 1000;

MySQL Workbench

Local instance MySQL80 x unconnected x

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

Server Status

Client Connections

Users and Privileges

Status and System Variables

Data Export

Data Import/Restore

INSTANCE

Startup / Shutdown

Server Logs

Options File

PERFORMANCE

Dashboard

Performance Reports

Performance Schema Setup

Administration Schemas

Query 1

```

79 where year(o.order_date) = 2023;
80
81 • select prod_name, min(stock_quantity) as min_quantity
82   from product_table
83  group by prod_name;
84
85 • select c.customer_id, c.first_name, sum(o.total_price) as total_spent
86

```

Result Grid

| customer_id | first_name | total_spent |
|-------------|------------|-------------|
| 1 | John | 1200.00 |
| 2 | Jane | 900.00 |
| 3 | Robert | 300.00 |
| 4 | Sarah | 150.00 |
| 5 | David | 1800.00 |
| 6 | Laura | 400.00 |
| 7 | Michael | 700.00 |
| 8 | Emma | 160.00 |
| 9 | William | 140.00 |
| 10 | Olivia | 1400.00 |

Result Grid

Form Editor

Field Types

Query Status

Result 10 x

Read Only

15) select * from product_table where product_id not in (select product_id from cart);

MySQL Workbench

Local instance MySQL80 x unconnected x

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

Information

Query 1

```

75 • select * from cart where customer_id = 5;
76   Execute the selected portion of the script or everything, if there is no selection
77 • select distinct c.* from customers c
78   join orders o on c.customer_id = o.customer_id
79   where year(o.order_date) = 2023;
80
81 • select prod_name, min(stock_quantity) as min_quantity

```

Result Grid

| prod_name | min_quantity |
|----------------|--------------|
| Laptop | 10 |
| Smartphone | 15 |
| Tablet | 20 |
| Headphones | 30 |
| TV | 5 |
| Coffee Maker | 25 |
| Refrigerator | 10 |
| Microwave Oven | 15 |
| Blender | 20 |
| Vacuum Cleaner | 10 |

16) select * from customers where customer_id not in (select distinct customer_id from orders);

MySQL Workbench

Local instance MySQL80 x unconnected x

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

Information

Query 1

```

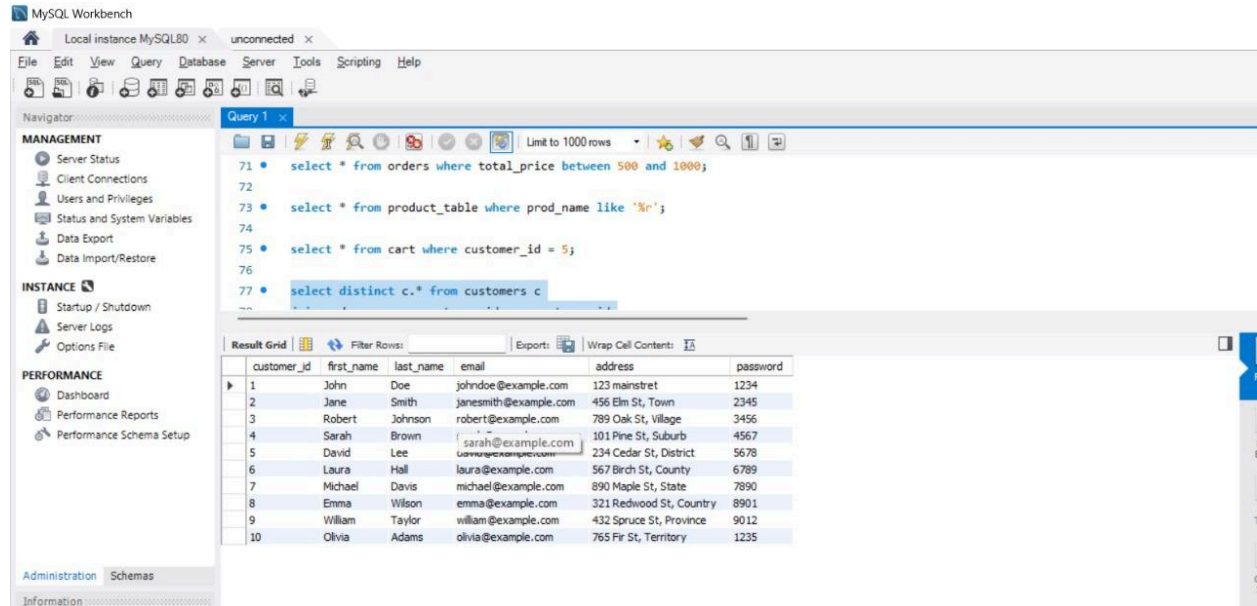
116   from order_items
117   group by product_id;
118
119 • select * from product_table
120   where stock_quantity < 10;
121
122 • select * from customers

```

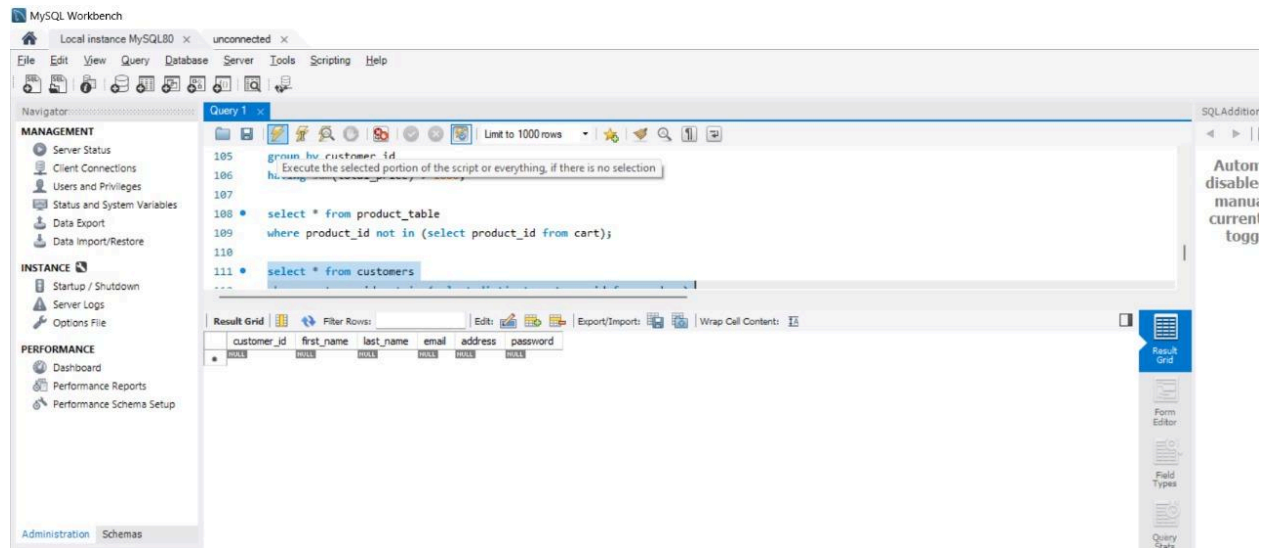
Result Grid

| customer_id | first_name | last_name | email | address | password |
|-------------|------------|-----------|----------------------|------------------------|----------|
| 1 | John | Doe | john.doe@example.com | 123 mainstreet | 1234 |
| 5 | David | Lee | david@example.com | 234 Cedar St, District | 5678 |
| 10 | Olivia | Adams | olivia@example.com | 765 Fir St, Territory | 1235 |

17) select product_id, round((sum(item_amount) * 100.0) / (select sum(item_amount) from order_items), 2) as revenue_percentage from order_items group by product_id;



18) **select * from product_table where stock_quan < 10;**



19) **select * from customers where customer_id in (select customer_id from orders where total_price > 1000);**

MySQL Workbench

Local instance MySQL80 x unconnected x

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

Query 1

```
93 group by c.customer_id;
94
95 • select customer_id, count(*) as order_count
96 from orders
97 group by customer_id;
98
99 • select customer_id, max(total_price) as max_order
---
```

Result Grid

| customer_id | max_order |
|-------------|-----------|
| 1 | 1200.00 |
| 2 | 900.00 |
| 3 | 300.00 |
| 4 | 150.00 |
| 5 | 1800.00 |
| 6 | 400.00 |
| 7 | 700.00 |
| 8 | 160.00 |
| 9 | 140.00 |
| 10 | 1400.00 |

Result Grid
Form Editor
Field Types
Query Stats