Coding Challenge - Ecommerce– SQL

(Satyendra Singh Rathore)

Tasks:

1. Update refrigerator product price to 800.

```
106     -- 1
107 •  UPDATE products
108     SET price = 800.00
109     WHERE name = 'Refrigerator';
110 •  select * from products;
```

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

product_id	name	price	description	stockQuantity	category
6	Coffee Maker	50.00	Automatic coffee maker	25	House Hold
7	Refrigerator	800.00	Energy-efficient	10	House Hold
8	Microwave Oven	80.00	Countertop microwave	15	House Hold
9	Blender	70.00	High-speed blender	20	House Hold
10	Vacuum Cleaner	120.00	Bagless vacuum cleaner	10	House Hold
NULL	NULL	NULL	NULL	NULL	NULL

2. Remove all cart items for a specific customer.

```
112     -- 2
113 •  DELETE FROM cart
114     WHERE customer_id = 4;
115
```

3. Retrieve Products Priced Below $100.

```
116     -- 3
117 •  SELECT *
118     FROM products
119     WHERE price < 100.00;
```

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

product_id	name	price	description	stockQuantity	category
6	Coffee Maker	50.00	Automatic coffee maker	25	House Hold
8	Microwave Oven	80.00	Countertop microwave	15	House Hold
9	Blender	70.00	High-speed blender	20	House Hold
NULL	NULL	NULL	NULL	NULL	NULL

4. Find Products with Stock Quantity Greater Than 5.

```
121     -- 4
122 •   SELECT *
123     FROM products
124     WHERE stockQuantity > 5;
```

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

product_id	name	price	description	stockQuantity	category
1	Laptop	800.00	High-performance laptop	10	Tech Gadgets
2	Smartphone	600.00	Latest smartphone	15	Tech Gadgets
3	Tablet	300.00	Portable tablet	20	Tech Gadgets
4	Headphones	150.00	Noise-canceling	30	Tech Gadgets
6	Coffee Maker	50.00	Automatic coffee maker	25	House Hold
7	Refrigerator	800.00	Energy-efficient	10	House Hold
8	Microwave Oven	80.00	Countertop microwave	15	House Hold

products 22

5. Retrieve Orders with Total Amount Between $500 and $1000.

```
126     -- 5
127 •   SELECT *
128     FROM orders
129     WHERE total_price BETWEEN 500.00 AND 1000.00;
```

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

order_id	customer_id	order_date	total_price	shipping_address
2	2	2023-02-10	900.00	456 Elm St, Town
7	7	2023-07-05	700.00	890 Maple St, State
NULL	NULL	NULL	NULL	NULL

6. Find Products which name end with letter ‘r’.

```
131     -- 6
132 •   SELECT *
133     FROM products
134     WHERE name LIKE '%r';
```

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

product_id	name	price	description	stockQuantity	category
6	Coffee Maker	50.00	Automatic coffee maker	25	House Hold
7	Refrigerator	800.00	Energy-efficient	10	House Hold
9	Blender	70.00	High-speed blender	20	House Hold
10	Vacuum Cleaner	120.00	Bagless vacuum cleaner	10	House Hold
NULL	NULL	NULL	NULL	NULL	NULL

7. Retrieve Cart Items for Customer 5.

```
136    -- 7
137    SELECT *
138    FROM cart
139    WHERE customer_id = 5;
```

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

cart_id	customer_id	product_id	quantity
7	5	1	1
NULL	NULL	NULL	NULL

8. Find Customers Who Placed Orders in 2023.

```
141    -- 8
142    SELECT DISTINCT c.*
143    FROM customers c
144    JOIN orders o ON c.customer_id = o.customer_id
145    WHERE YEAR(o.order_date) = 2023;
```

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

customer_id	name	email	address
1	John Doe	johndoe@example.com	123 Main St, City
2	Jane Smith	janesmith@example.com	456 Elm St, Town
3	Robert Johnson	robert@example.com	789 Oak St, Village
4	Sarah Brown	sarah@example.com	789 Oak St, Village
5	David Lee	david@example.com	234 Cedar St, District
6	Laura Hall	laura@example.com	567 Birch St, County
7	Michael Davis	michael@example.com	890 Maple St, State

9. Determine the Minimum Stock Quantity for Each Product Category.

```
147    -- 9
148    SELECT category, MIN(stockQuantity) AS min_stock_quantity
149    FROM products
150    GROUP BY category;
```

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

category	min_stock_quantity
Tech Gadgets	5
House Hold	10

10. Calculate the Total Amount Spent by Each Customer.

```
152     -- 10
153 •   SELECT customer_id, SUM(total_price) AS total_amount_spent
154     FROM orders
155     GROUP BY customer_id;
```

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

customer_id	total_amount_spent
1	1200.00
2	900.00
3	300.00
4	150.00
5	1800.00
6	400.00
7	700.00

Result 28

11. Find the Average Order Amount for Each Customer.

```
157     -- 11
158 •   SELECT customer_id, AVG(total_price) AS average_order_amount
159     FROM orders
160     GROUP BY customer_id;
161
```

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

customer_id	average_order_amount
1	1200.000000
2	900.000000
3	300.000000
4	150.000000
5	1800.000000
6	400.000000
7	700.000000

Result 29

12. Count the Number of Orders Placed by Each Customer.

```
162     -- 12
163 •   SELECT customer_id, COUNT(order_id) AS order_count
164     FROM orders
165     GROUP BY customer_id;
```

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

customer_id	order_count
1	1
2	1
3	1
4	1
5	1
6	1
7	1

13. Find the Maximum Order Amount for Each Customer.

```
167     -- 13
168 •   SELECT customer_id, MAX(total_price) AS max_order_amount
169     FROM orders
170     GROUP BY customer_id;
```

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

customer_id	max_order_amount
1	1200.00
2	900.00
3	300.00
4	150.00
5	1800.00
6	400.00
7	700.00

14. Get Customers Who Placed Orders Totaling Over $1000.

```
172     -- 14
173 •   SELECT customer_id, SUM(total_price) AS total_amount_spent
174     FROM orders
175     GROUP BY customer_id
176     HAVING SUM(total_price) > 1000;
```

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

customer_id	total_amount_spent
1	1200.00
5	1800.00
10	1400.00

15. Subquery to Find Products Not in the Cart.

```
178    -- 15
179    SELECT *
180    FROM products
181    WHERE product_id NOT IN (SELECT DISTINCT product_id FROM cart);
182
```

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

product_id	name	price	description	stockQuantity	category
6	Coffee Maker	50.00	Automatic coffee maker	25	House Hold
8	Microwave Oven	80.00	Countertop microwave	15	House Hold
NULL	NULL	NULL	NULL	NULL	NULL

16. Subquery to Find Customers Who Haven't Placed Orders.

```
183    -- 16
184    SELECT *
185    FROM customers
186    WHERE customer_id NOT IN (SELECT DISTINCT customer_id FROM orders);
187
```

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

customer_id	name	email	address
NULL	NULL	NULL	NULL

18. Subquery to Find Products with Low Stock.

```
202    -- 18
203    SELECT *
204    FROM products
205    WHERE stockQuantity < (
206        SELECT AVG(stockQuantity)
207        FROM products
208    );
```

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

product_id	name	price	description	stockQuantity	category
1	Laptop	800.00	High-performance laptop	10	Tech Gadgets
2	Smartphone	600.00	Latest smartphone	15	Tech Gadgets
5	TV	900.00	4K Smart TV	5	Tech Gadgets
7	Refrigerator	800.00	Energy-efficient	10	House Hold
8	Microwave Oven	80.00	Countertop microwave	15	House Hold
10	Vacuum Cleaner	120.00	Bagless vacuum cleaner	10	House Hold
NULL	NULL	NULL	NULL	NULL	NULL

19. Subquery to Find Customers Who Placed High-Value Orders.

```
210    -- 19
211 •  SELECT *
212    FROM customers
213    WHERE customer_id IN (
214        SELECT customer_id
215        FROM orders
216        GROUP BY customer_id
217        HAVING SUM(total_price) > 1000
218    );
219
```

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

customer_id	name	email	address
1	John Doe	johndoe@example.com	123 Main St, City
5	David Lee	david@example.com	234 Cedar St, District
10	Olivia Adams	olivia@example.com	765 Fir St, Territory
NULL	NULL	NULL	NULL

*****