

CODING CHALLENGE

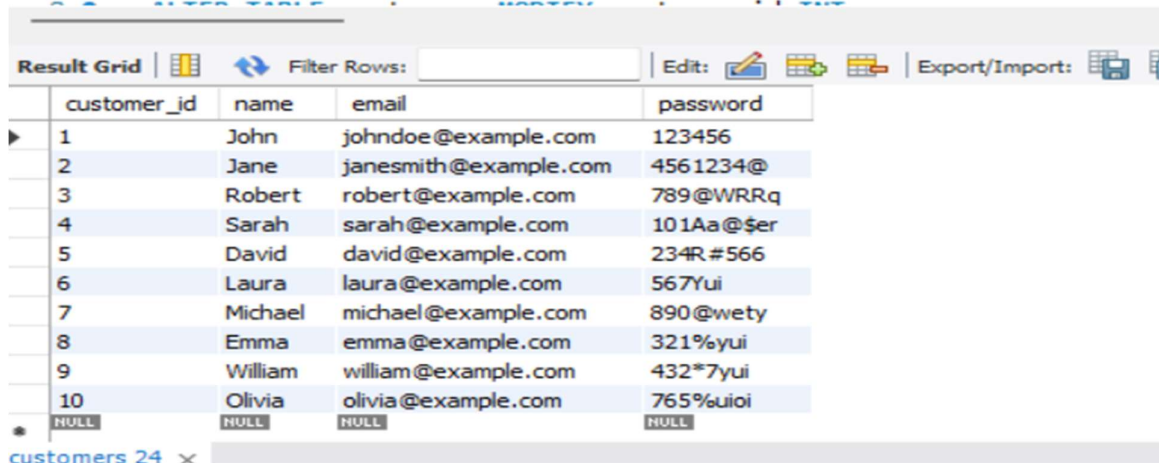
Submitted by: Fahimunnisha A

Topic: E-Commerce

Tables

1) Customer table:

```
2 • CREATE TABLE customers (  
3     customer_id INT PRIMARY KEY,  
4     name VARCHAR(100) NOT NULL,  
5     email VARCHAR(100) UNIQUE NOT NULL,  
6     password VARCHAR(255) NOT NULL  
7 );
```

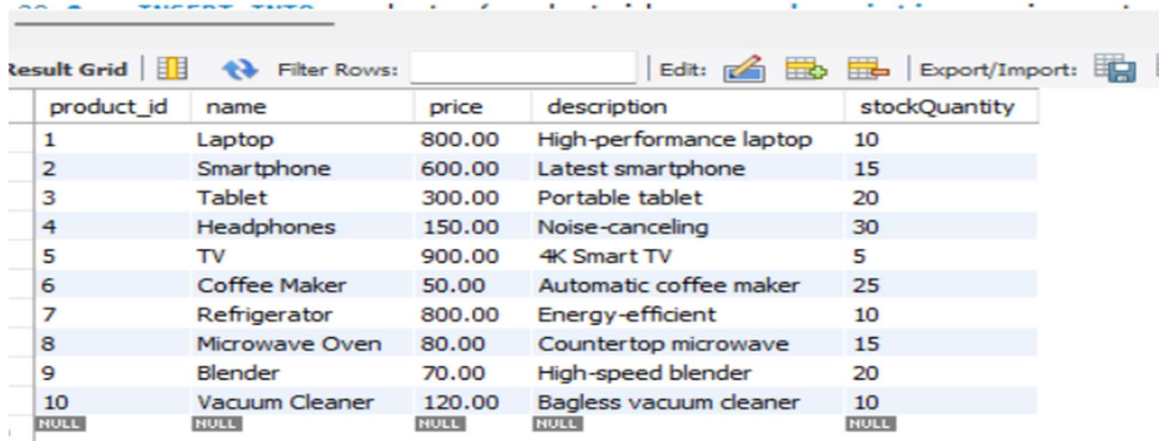


customer_id	name	email	password
1	John	johndoe@example.com	123456
2	Jane	janesmith@example.com	4561234@
3	Robert	robert@example.com	789@WRRq
4	Sarah	sarah@example.com	101Aa@\$er
5	David	david@example.com	234R#566
6	Laura	laura@example.com	567Yui
7	Michael	michael@example.com	890@wety
8	Emma	emma@example.com	321%yui
9	William	william@example.com	432*7yui
10	Olivia	olivia@example.com	765%uioi
NULL	NULL	NULL	NULL

customers 24 ×

2) Products table:

```
22 • CREATE TABLE products (  
23     product_id INT PRIMARY KEY,  
24     name VARCHAR(100) NOT NULL,  
25     price DECIMAL(10, 2) NOT NULL,  
26     description TEXT,  
27     stockQuantity INT NOT NULL  
28 );
```



product_id	name	price	description	stockQuantity
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
NULL	NULL	NULL	NULL	NULL

3)Cart table:

```
82 • CREATE TABLE cart (  
83     cart_id INT PRIMARY KEY,  
84     customer_id INT NOT NULL,  
85     product_id INT NOT NULL,  
86     quantity INT NOT NULL DEFAULT 1,  
87     FOREIGN KEY (customer_id) REFERENCES customers(customer_id),  
88     FOREIGN KEY (product_id) REFERENCES products(product_id)  
89 )
```

Result Grid				
Filter Rows:				
	cart_id	customer_id	product_id	quantity
1	1	1	1	2
2	1	3	3	1
3	2	2	2	3
4	3	4	4	4
5	3	5	5	2
6	4	6	6	1
8	6	10	10	2
9	6	9	9	3
10	7	7	7	2
*	NULL	NULL	NULL	NULL

4)Orders table:

```
43 • CREATE TABLE orders (  
44     order_id INT PRIMARY KEY,  
45     customer_id INT,  
46     order_date DATE,  
47     total_price DOUBLE,  
48     shipping_address VARCHAR(255),  
49     FOREIGN KEY (customer_id) REFERENCES customers(customer_id)  
50 )
```

Result Grid

Filter Rows:

Edit:

Export/Import:

order_id	customer_id	order_date	total_price	shipping_address
1	1	2023-01-05	1200	123 Main St, City
2	2	2023-02-10	900	456 Elm St, Town
3	3	2023-03-15	300	789 Oak St, Village
4	4	2023-04-20	150	101 Pine St, Suburb
5	5	2023-05-25	1800	234 Cedar St, District
6	6	2023-06-30	400	567 Birch St, County
7	7	2023-07-05	700	890 Maple St, State
8	8	2023-08-10	160	321 Redwood St, Country
9	9	2023-09-15	140	432 Spruce St, Province
10	10	2023-10-20	1400	765 Fir St, Territory
NULL	NULL	NULL	NULL	NULL

5)Order_items table:

```
63 • CREATE TABLE order_items (order_item_id INT PRIMARY KEY,  
64     order_id INT,  
65     product_id INT,  
66     quantity INT,  
67     FOREIGN KEY (order_id) REFERENCES orders(order_id),  
68     FOREIGN KEY (product_id) REFERENCES products(product_id)
```

Result Grid	Filter Rows:	Edit:	Export/Import:
order_item_id	order_id	product_id	quantity
1	1	1	2
2	1	3	1
3	2	2	3
4	3	5	2
5	4	4	4
6	4	6	1
7	5	1	1
8	5	2	2
9	6	10	2
10	6	9	3
NULL	NULL	NULL	NULL

Challenges

1)Update refrigerator product price to 800.

```
118 • UPDATE products  
119     SET price = 800  
120     WHERE name = 'Refrigerator';  
121 • select * from products;
```

Result Grid

Filter Rows:

Edit:

Export/Import

	product_id	name	price	description	stockQuantity
▶	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
✱	NULL	NULL	NULL	NULL	NULL

2) Remove all cart items for a specific customer.

```
124 • DELETE FROM cart
125     WHERE customer_id = 5;
126 • select *from cart;
```

Result Grid | Filter Rows: | Edit: | Export

	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
	8	6	10	2
	9	6	9	3
	10	7	7	2
*	NULL	NULL	NULL	NULL

3) Retrieve Products Priced below \$100.

```
128 • SELECT * FROM products
129     WHERE price < 100;
```

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

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

4)Find products with stock quantity greater than 5

131 • SELECT * FROM products

132 WHERE stockQuantity > 5;

133

134 • SELECT * FROM

Result Grid

Filter Rows:

Edit:

Export/Import:

	product_id	name	price	description	stockQuantity
▶	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	NULL

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

```
134 • SELECT * FROM orders
135 WHERE total_price BETWEEN 500 AND 1000;
136
```

Result Grid

Filter Rows:

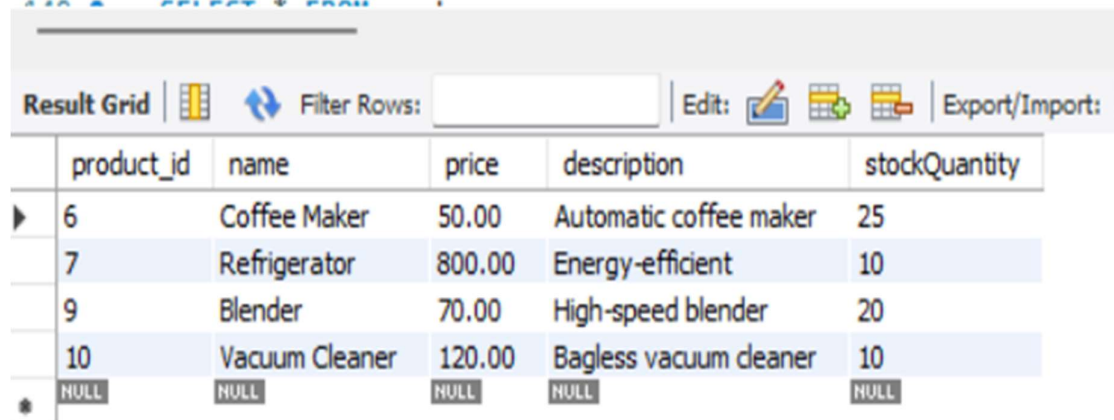
Edit:

Export/Import:

	order_id	customer_id	order_date	total_price	shipping_address
▶	2	2	2023-02-10	900	456 Elm St, Town
	7	7	2023-07-05	700	890 Maple St, State
*	NULL	NULL	NULL	NULL	NULL

6) Find Products which name end with letter 'r'.

```
137 • SELECT * FROM products
138     WHERE name LIKE '%r';
139
```

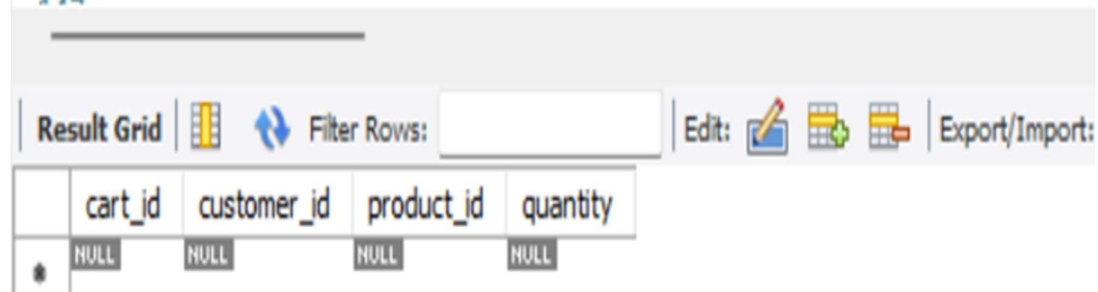


The screenshot shows a database query result grid. The toolbar includes 'Result Grid', 'Filter Rows', 'Edit', and 'Export/Import'. The query results are displayed in a table with the following columns: product_id, name, price, description, and stockQuantity. The results show four products: Coffee Maker, Refrigerator, Blender, and Vacuum Cleaner. The last row shows NULL values for all columns.

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

7) Retrieve Cart Items for Customer 5.

```
145 • SELECT * FROM cart
146     WHERE customer_id = 5;
```



The screenshot shows a database query result grid. The toolbar includes 'Result Grid', 'Filter Rows', 'Edit', and 'Export/Import'. The query results are displayed in a table with the following columns: cart_id, customer_id, product_id, and quantity. The results show a single row with NULL values for all columns.

cart_id	customer_id	product_id	quantity
NULL	NULL	NULL	NULL

8) Find Customers Who Placed Orders in 2023.

```
149 • SELECT DISTINCT c.*
150 FROM customers c
151 JOIN orders o ON c.customer_id = o.customer_id
152 WHERE YEAR(o.order_date) = 2023;
```

Result Grid				
		Filter Rows:	Export:	
		Wrap Cell Content:		
	customer_id	name	email	password
▶	1	John	johndoe@example.com	123456
	2	Jane	janesmith@example.com	4561234@
	3	Robert	robert@example.com	789@WRRq
	4	Sarah	sarah@example.com	101Aa@\$er
	5	David	david@example.com	234R#566
	6	Laura	laura@example.com	567Yui
	7	Michael	michael@example.com	890@wety
	8	Emma	emma@example.com	321%yui
	9	William	william@example.com	432*7yui
	10	Olivia	olivia@example.com	765%uioi



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

```
153 • SELECT name, MIN(stockQuantity) AS min_stock
154 FROM products
155 GROUP BY name;
```

Result Grid		
		Filter Rows:
		Export:
		Wrap
	name	min_stock
▶	Laptop	10
	Smartphone	15
	Tablet	20
	Headphones	30
	TV	5
	Coffee Maker	25
	Refrigerator	10
	Microwave Oven	15
	Blender	20
	Vacuum Cleaner	10



10) Calculate the Total Amount Spent by Each Customer.

```
157 • SELECT customer_id, SUM(total_price) AS total_spent
158 FROM orders
159 GROUP BY customer_id;
```

Result Grid  Filter Rows: <input type="text"/> Export:  Wrap Cell Content: <input type="checkbox"/>		
	customer_id	total_spent
▶	1	1200
	2	900
	3	300
	4	150
	5	1800
	6	400
	7	700
	8	160
	9	140
	10	1400

11) Find the Average Order Amount for Each Customer.

```
161 • SELECT customer_id, AVG(total_price) AS average_order
162 FROM orders
163 GROUP BY customer_id;
```

Result Grid  Filter Rows: <input type="text"/> Export:  Wrap Cell Content: <input type="checkbox"/>		
	customer_id	average_order
▶	1	1200
	2	900
	3	300
	4	150
	5	1800
	6	400
	7	700
	8	160
	9	140
	10	1400

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

```
167 • SELECT customer_id, COUNT(*) AS order_count
168 FROM orders
169 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
	8	1
	9	1
	10	1

13) Find the Maximum Order Amount for Each Customer.

```
171 • SELECT customer_id, MAX(total_price) AS max_order
172 FROM orders
173 GROUP BY customer_id;
```

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

	customer_id	max_order
▶	1	1200
	2	900
	3	300
	4	150
	5	1800
	6	400
	7	700
	8	160
	9	140
	10	1400

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

```
176 • SELECT customer_id, SUM(total_price) AS total_spent
177 FROM orders
178 GROUP BY customer_id
179 HAVING total_spent > 1000;
```

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

	customer_id	total_spent
▶	1	1200
	5	1800
	10	1400

15)Subquery to Find Products Not in the Cart.

```
181 • SELECT * FROM products
182 WHERE product_id NOT IN (
183     SELECT product_id FROM cart
184 );
```

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

product_id	name	price	description	stockQuantity
8	Microwave Oven	80.00	Countertop microwave	15
NULL	NULL	NULL	NULL	NULL

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

```
187 • SELECT * FROM customers
188 WHERE customer_id NOT IN (
189     SELECT customer_id FROM orders
190 );
191
```

Result Grid	Filter Rows:	Edit:	Export/Import:
customer_id	name	email	password
* NULL	NULL	NULL	NULL

17) Subquery to Calculate the Percentage of Total Revenue for a Product.

```
192 • SELECT order_items.product_id,
193     (SUM(products.price * order_items.quantity) /
194     (SELECT SUM(total_price) FROM orders)) * 100 AS revenue_percentage
195 FROM order_items
196 JOIN products ON order_items.product_id = products.product_id
197 GROUP BY order_items.product_id;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
product_id	revenue_percentage		
1	33.56643356643357		
3	4.195804195804196		
2	41.95804195804196		
5	25.174825174825177		
4	8.391608391608392		
6	0.6993006993006993		
10	3.3566433566433567		
9	2.937062937062937		

