

QUERY OUTPUT

Creation of the Database Ecommerce

Output				
Action Output				
#	Time	Action	Message	Duration / Fetch
1	10:14:23	CREATE DATABASE Ecommerce	1 row(s) affected	0.000 sec
2	10:14:23	USE Ecommerce	0 row(s) affected	0.000 sec

Creation of Table customers

```
8 CREATE TABLE customers(  
9     customer_id INT PRIMARY KEY AUTO_INCREMENT,  
10    first_name VARCHAR(50) NOT NULL,  
11    last_name VARCHAR(50) NOT NULL,  
12    email VARCHAR(100) UNIQUE,  
13    address VARCHAR(255)  
14 );  
15  
16 SELECT * FROM customers;  
17
```

customer_id	first_name	last_name	email	address
NULL	NULL	NULL	NULL	NULL

Creation of Table products

```
18 CREATE TABLE products(  
19     product_id INT PRIMARY KEY AUTO_INCREMENT,  
20     name VARCHAR(100) NOT NULL,  
21     description TEXT,  
22     price DECIMAL(10,2) NOT NULL,  
23     stockQuantity INT NOT NULL  
24 );  
25  
26 SELECT * FROM products;  
27
```

product_id	name	description	price	stockQuantity
NULL	NULL	NULL	NULL	NULL

Creation of Table cart

```
28 CREATE TABLE cart(  
29     cart_id INT PRIMARY KEY AUTO_INCREMENT,  
30     customer_id INT,  
31     product_id INT,  
32     quantity INT NOT NULL,  
33     FOREIGN KEY (customer_id) REFERENCES customers(customer_id),  
34     FOREIGN KEY (product_id) REFERENCES products(product_id)  
35 );  
36  
37 SELECT * FROM cart;
```

cart_id	customer_id	product_id	quantity
NULL	NULL	NULL	NULL

Creation of Table orders

```
39 • CREATE TABLE orders(  
40     order_id INT PRIMARY KEY AUTO_INCREMENT,  
41     customer_id INT,  
42     order_date DATE,  
43     total_price DECIMAL(10,2) NOT NULL,  
44     shipping_address VARCHAR(255),  
45     FOREIGN KEY (customer_id) REFERENCES customers(customer_id)  
46 );  
47  
48 • SELECT * FROM orders;  
49
```

Result Grid

order_id	customer_id	order_date	total_price	shipping_address
NULL	NULL	NULL	NULL	NULL

Result Grid, Form Editor

Creation of Table order_items

```
50 • CREATE TABLE order_items(  
51     order_item_id INT PRIMARY KEY AUTO_INCREMENT,  
52     order_id INT,  
53     product_id INT,  
54     quantity INT NOT NULL,  
55     itemAmount DECIMAL(10,2) NOT NULL,  
56     FOREIGN KEY (order_id) REFERENCES orders(order_id),  
57     FOREIGN KEY (product_id) REFERENCES products(product_id)  
58 );  
59  
60 • SELECT * FROM order_items;  
61
```

Result Grid

order_item_id	order_id	product_id	quantity	itemAmount
NULL	NULL	NULL	NULL	NULL

Result Grid

Insertion in Table customers

```
58 • INSERT INTO customers(customer_id,first_name,last_name,email,address) VALUES  
59     (1,'John','Doe','johndoe@example.com','123 Main St, City'),  
60     (2,'Jane','Smith','janesmith@example.com','456 Elm St, Town'),  
61     (3,'Robert','Johnson','robert@example.com','789 Oak St, Village'),  
62     (4,'Sarah','Brown','sarah@example.com','101 Pine St, Suburb'),  
63     (5,'David','Lee','david@example.com','234 Cedar St, District'),  
64     (6,'Laura','Hall','laura@example.com','567 Birch St, County'),  
65     (7,'Michael','Davis','michael@example.com','890 Maple St, State'),  
66     (8,'Emma','Wilson','emma@example.com','321 Redwood St, Country'),  
67     (9,'William','Taylor','william@example.com','432 Spruce St, Province'),  
68     (10,'Olivia','Adams','olivia@example.com','765 Fir St, Territory');  
69  
70 • SELECT * FROM customers;
```

Result Grid

customer_id	first_name	last_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	101 Pine St, Suburb
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
8	Emma	Wilson	emma@example.com	321 Redwood St, Country
9	William	Taylor	william@example.com	432 Spruce St, Province
10	Olivia	Adams	olivia@example.com	765 Fir St, Territory

Result Grid, Form Editor, Field Types, Query Suite

Insertion in Table products

```
76 • INSERT INTO products(product_id, name, description, price,stockQuantity) VALUES
77     (1,'Laptop','High-performance laptop',800.00,10),
78     (2,'Smartphone','Latest smartphone',600.00,15),
79     (3,'Tablet','Portable tablet',300.00,20),
80     (4,'Headphones','Noise-canceling',150.00,30),
81     (5,'TV','4K Smart TV',900.00,5),
82     (6,'Coffee Maker','Automatic coffee maker',50.00, 25),
83     (7,'Refrigerator','Energy-efficient',700.00,10),
84     (8,'Microwave Oven','Countertop microwave',80.00,15),
85     (9,'Blender','High-speed blender',70.00,20),
86     (10,'Vacuum Cleaner','Bagless vacuum cleaner',120.00,10);
87
88 • SELECT * FROM products;
```

Result Grid

	product_id	name	description	price	stockQuantity
▶	1	Laptop	High-performance laptop	800.00	10
	2	Smartphone	Latest smartphone	600.00	15
	3	Tablet	Portable tablet	300.00	20
	4	Headphones	Noise-canceling	150.00	30
	5	TV	4K Smart TV	900.00	5
	6	Coffee Maker	Automatic coffee maker	50.00	25
	7	Refrigerator	Energy-efficient	700.00	10
	8	Microwave Oven	Countertop microwave	80.00	15
	9	Blender	High-speed blender	70.00	20
	10	Vacuum Cleaner	Bagless vacuum cleaner	120.00	10
•	NULL	NULL	NULL	NULL	NULL

Form Editor, Field Types, Query Stats

Insertion in Table orders

```
96 • INSERT INTO orders (order_id, customer_id, order_date, total_price, shipping_address) VALUES
97     (1, 1, '2023-01-05', 1200.00, '123 Main St, City'),
98     (2, 2, '2023-02-10', 900.00, '456 Elm St, Town'),
99     (3, 3, '2023-03-15', 300.00, '789 Oak St Village'),
100     (4, 4, '2023-04-20', 150.00, '101 Pine St, Suburb'),
101     (5, 5, '2023-05-25', 1800.00, '234 Cedar St, District'),
102     (6, 6, '2023-06-30', 400.00, '567 Birch St, County'),
103     (7, 7, '2023-07-05', 700.00, '890 Maple St, State'),
104     (8, 8, '2023-08-10', 160.00, '321 Redwood St, Country'),
105     (9, 9, '2023-09-15', 140.00, '432 Spruce St, Province'),
106     (10, 10, '2023-10-20', 1400.00, '765 Fir St, Territory');
107
108 • SELECT * FROM orders;
```

Result Grid

	order_id	customer_id	order_date	total_price	shipping_address
▶	1	1	2023-01-05	1200.00	123 Main St, City
	2	2	2023-02-10	900.00	456 Elm St, Town
	3	3	2023-03-15	300.00	789 Oak St Village
	4	4	2023-04-20	150.00	101 Pine St, Suburb
	5	5	2023-05-25	1800.00	234 Cedar St, District
	6	6	2023-06-30	400.00	567 Birch St, County
	7	7	2023-07-05	700.00	890 Maple St, State
	8	8	2023-08-10	160.00	321 Redwood St, Country
	9	9	2023-09-15	140.00	432 Spruce St, Province
	10	10	2023-10-20	1400.00	765 Fir St, Territory
•	NULL	NULL	NULL	NULL	NULL

Form Editor, Field Types, Query Stats

Insertion in Table order_items

```
110 • INSERT INTO order_items (order_item_id, order_id, product_id, quantity, itemAmount) VALUES
111     (1,1,1,2,1600.00),(2,1,3,1,300.00),(3,2,2,3,1800.00),(4,3,5,2,1800.00),
112     (5,4,4,4,600.00),(6,4,6,1,50.00),(7,5,1,1,800.00),(8,5,2,2,1200.00),
113     (9,6,10,2,240.00),(10,6,9,3,210.00);
114
115 • SELECT * FROM order_items;
116
117
```

order_item_id	order_id	product_id	quantity	itemAmount
1	1	1	2	1600.00
2	1	3	1	300.00
3	2	2	3	1800.00
4	3	5	2	1800.00
5	4	4	4	600.00
6	4	6	1	50.00
7	5	1	1	800.00
8	5	2	2	1200.00
9	6	10	2	240.00
10	6	9	3	210.00
* NULL	NULL	NULL	NULL	NULL

Show tables

```
117 • show tables;
```

Tables_in_e-commerce
cart
customers
order_items
orders
products

SQL challenges outputs

```
1 /*1. Update refrigerator product price to 800*/
2
3 • UPDATE products
4 SET price = 800.00
5 WHERE product_id = 7;
6
7 • select * from products;
```

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

```

9      /*2. Remove all cart items for a specific customer*/
10
11      DELETE FROM cart WHERE customer_id=2;
12
13      select * from cart;

```

Result Grid				
Filter Rows:				
cart_id	customer_id	product_id	quantity	
1	1	1	2	
2	1	3	1	
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	
NULL	NULL	NULL	NULL	

```

15     /*3. Retrieve Products Priced Below $100*/
16
17     SELECT * FROM products WHERE price <100;

```

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

```

19     /*4. Find Products with Stock Quantity Greater Than 5*/
20
21     SELECT * FROM products WHERE stockQuantity > 5;

```

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

```

23     /*5. Retrieve Orders with Total Amount Between $500 and $1000*/
24
25     SELECT * FROM orders WHERE total_price BETWEEN 500 AND 1000;

```

Result Grid				
Filter Rows:				
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

```

27  /*6. Find Products which name end with letter 'r'*/
28
29  • SELECT * FROM products WHERE name LIKE '%r';
30

```

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

```

31  /*7. Retrieve Cart Items for Customer 5*/
32
33  • SELECT * FROM cart WHERE customer_id = 5;
34

```

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

```

35  /*8. Find Customers who Placed Orders in 2023.*/
36
37  • SELECT DISTINCT customers.*
38  FROM customers
39  JOIN orders ON customers.customer_id=orders.customer_id
40  WHERE YEAR(order_date)=2023;
41

```

customer_id	first_name	last_name	email	address
1	John	Doe	john.doe@example.com	123 Main St, City
2	Jane	Smith	jane.smith@example.com	456 Elm St, Town
3	Robert	Johnson	robert@example.com	789 Oak St, Village
4	Sarah	Brown	sarah@example.com	101 Pine St, Suburb
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
8	Emma	Wilson	emma@example.com	321 Redwood St, Country
9	William	Taylor	william@example.com	432 Spruce St, Province
10	Olivia	Adams	olivia@example.com	765 Fir St, Territory

```

42  /*9. Determine the Minimum Stock Quantity for Each Product Category*/
43
44  • SELECT * FROM products WHERE stockQuantity = (SELECT MIN(stockQuantity) FROM products);

```

product_id	name	description	price	stockQuantity
5	TV	4K Smart TV	900.00	5
•	NULL	NULL	NULL	NULL

```

46  /*10. Calculate the Total Amount Spent by Each Customer.*/
47
48  • SELECT c.customer_id,c.first_name,c.last_name,SUM(orders.total_price) AS totalAmount_spent
49  FROM customers c
50  JOIN orders ON c.customer_id = orders.customer_id
51  GROUP BY c.customer_id,c.first_name,c.last_name;
52

```

Result Grid				
Filter Rows:				
Export:				
Wrap Cell Content:				
	customer_id	first_name	last_name	totalAmount_spent
▶	1	John	Doe	1200.00
	2	Jane	Smith	900.00
	3	Robert	Johnson	300.00
	4	Sarah	Brown	150.00
	5	David	Lee	1800.00
	6	Laura	Hall	400.00
	7	Michael	Davis	700.00
	8	Emma	Wilson	160.00
	9	William	Taylor	140.00
	10	Olivia	Adams	1400.00

```

53  /*11. Find the Average Order Amount for Each Customer*/
54
55  • SELECT c.customer_id,c.first_name,c.last_name,AVG(orders.total_price) AS AvgOrder_Amount
56  FROM customers c
57  JOIN orders ON c.customer_id = orders.customer_id
58  GROUP BY c.customer_id,c.first_name,c.last_name;
59

```

Result Grid				
Filter Rows:				
Export:				
Wrap Cell Content:				
	customer_id	first_name	last_name	AvgOrder_Amount
▶	1	John	Doe	1200.000000
	2	Jane	Smith	900.000000
	3	Robert	Johnson	300.000000
	4	Sarah	Brown	150.000000
	5	David	Lee	1800.000000
	6	Laura	Hall	400.000000
	7	Michael	Davis	700.000000
	8	Emma	Wilson	160.000000
	9	William	Taylor	140.000000
	10	Olivia	Adams	1400.000000

```

60  /*12. Count the Number of Orders Placed by Each Customer*/
61
62  • SELECT c.customer_id,c.first_name,c.last_name,COUNT(orders.order_id) AS Order_Placed
63  FROM customers c
64  JOIN orders ON c.customer_id = orders.customer_id
65  GROUP BY c.customer_id,c.first_name,c.last_name;
66

```

Result Grid				
Filter Rows:				
Export:				
Wrap Cell Content:				
	customer_id	first_name	last_name	Order_Placed
▶	1	John	Doe	1
	2	Jane	Smith	1
	3	Robert	Johnson	1
	4	Sarah	Brown	1
	5	David	Lee	1
	6	Laura	Hall	1
	7	Michael	Davis	1
	8	Emma	Wilson	1
	9	William	Taylor	1
	10	Olivia	Adams	1

67 /*13. Find the Maximum Order Amount for Each Customer*/

68

```
69 • SELECT c.customer_id,c.first_name,c.last_name,MAX(orders.total_price) AS MaxOrder_Amount
70 FROM customers c
71 JOIN orders ON c.customer_id = orders.customer_id
72 GROUP BY c.customer_id,c.first_name,c.last_name;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Contents:
customer_id	first_name	last_name	MaxOrder_Amount
1	John	Doe	1200.00
2	Jane	Smith	900.00
3	Robert	Johnson	300.00
4	Sarah	Brown	150.00
5	David	Lee	1800.00
6	Laura	Hall	400.00
7	Michael	Davis	700.00
8	Emma	Wilson	160.00
9	William	Taylor	140.00
10	Olivia	Adams	1400.00

74 /*14. Get Customers Who Placed Orders Totaling Over \$1000*/

75

```
76 • SELECT c.customer_id,c.first_name,c.last_name,SUM(orders.total_price) AS total_amount
77 FROM customers c
78 JOIN orders ON c.customer_id = orders.customer_id
79 GROUP BY c.customer_id,c.first_name,c.last_name
80 HAVING total_amount > 1000;
```

81

Result Grid	Filter Rows:	Export:	Wrap Cell Contents:
customer_id	first_name	last_name	total_amount
1	John	Doe	1200.00
5	David	Lee	1800.00
10	Olivia	Adams	1400.00

82 /*15. Subquery to Find Products Not in the Cart*/

83

```
84 • SELECT * FROM products
85 WHERE product_id NOT IN (SELECT DISTINCT product_id FROM cart);
```

Result Grid	Filter Rows:	Edits:	Export/Import:	Wrap Cell Contents:
product_id	name	description	price	stockQuantity
2	Smartphone	Latest smartphone	600.00	15
8	Microwave Oven	Countertop microwave	80.00	15
*	NULL	NULL	NULL	NULL


```

87  /*16. Subquery to Find Customers Who Haven't Placed Orders*/
88
89  • SELECT * FROM customers
90  WHERE customer_id NOT IN (SELECT DISTINCT customer_id FROM orders);
91

```

Result Grid	Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:	Result Grid	Form Editor	Field Types
customer_id	first_name	last_name	email	address			
•	NULL	NULL	NULL	NULL			

```

92  /*17. Subquery to Calculate the Percentage of Total Revenue for a Product*/
93
94  • SELECT product_id,(SUM(itemAmount) / (SELECT SUM(total_price) FROM orders)) * 100 AS revenue_percentage
95  FROM order_items
96  GROUP BY product_id;
97

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:	Result Grid	Form Editor	Field Types	Query State
product_id	revenue_percentage						
▶	1	33.566434					
	2	41.958042					
	3	4.195804					
	4	8.391608					
	5	25.174825					
	6	0.699301					
	9	2.937063					
	10	3.356643					

```

98  /*18. Subquery to Find Products with Low Stock*/
99
100 • SELECT * FROM products
101 WHERE stockQuantity = (SELECT MIN(stockQuantity) FROM products);

```

Result Grid	Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:	Result Grid	Form Editor	Field Types
product_id	name	description	price	stockQuantity			
▶	5	TV	4K Smart TV	900.00	5		
•	NULL	NULL	NULL	NULL	NULL		

```

103  /*19. Subquery to Find Customers Who Placed High-Value Orders.*/
104
105  SELECT customer_id, first_name
106  FROM customers
107  WHERE customer_id IN (
108      SELECT DISTINCT customer_id
109      FROM orders
110      WHERE total_price > (SELECT AVG(total_price) FROM orders)
111  );

```

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

	customer_id	first_name
▶	1	John
	2	Jane
	5	David
	10	Olivia
•	NULL	NULL

Result Grid
Form Editor
Field