MySQL Report

SQL Commands for Database Tables:

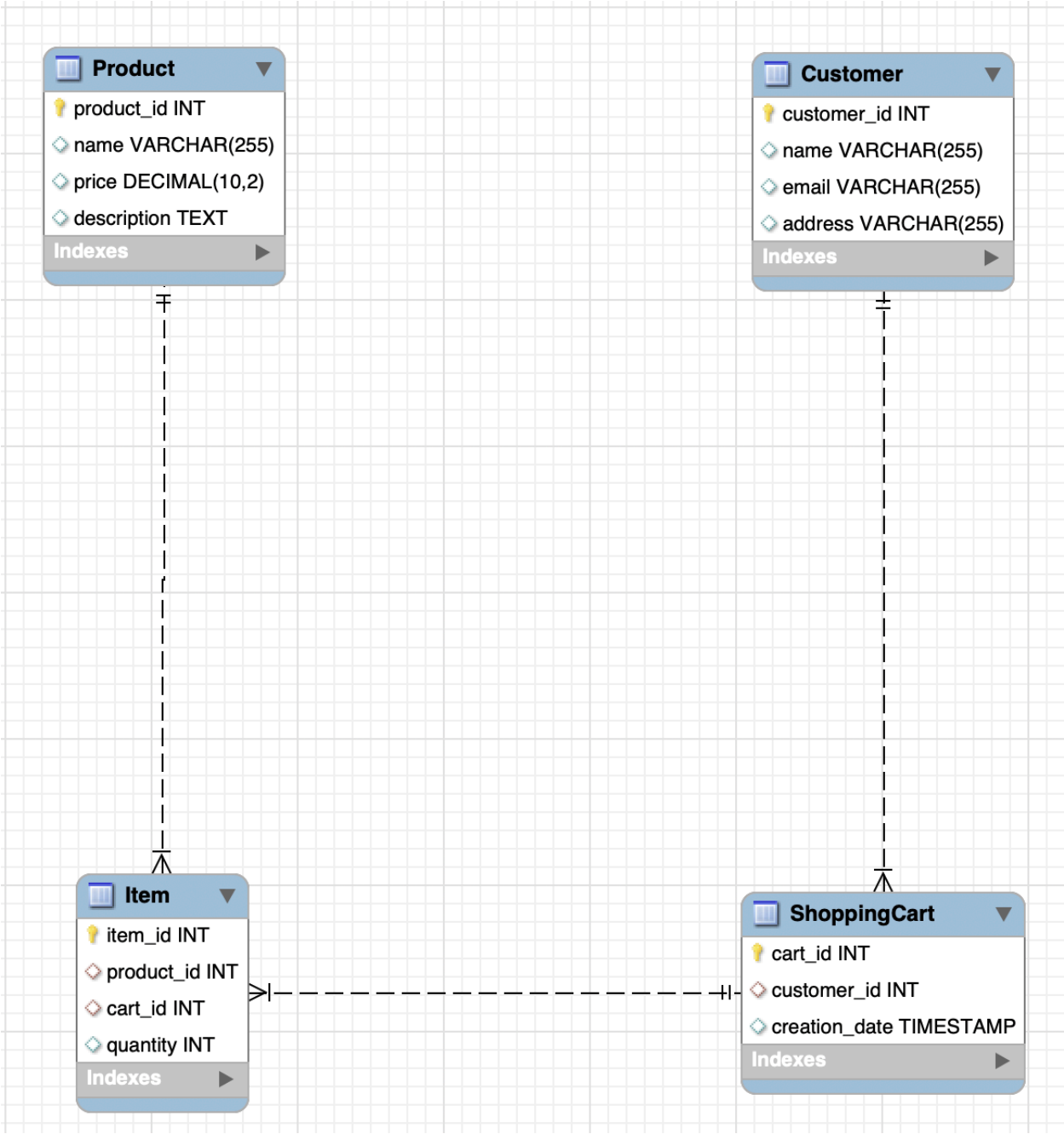
```
CREATE TABLE Customer
(
        customer_id INT PRIMARY KEY,
    name VARCHAR(255),
    email VARCHAR(255),
    address VARCHAR(255)
);


CREATE TABLE Product
(
        product_id INT PRIMARY KEY,
    name VARCHAR(255),
    price DECIMAL(10,2),
    description TEXT
);


CREATE TABLE ShoppingCart
(
        cart_id INT PRIMARY KEY,
    customer_id INT,
    creation_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (customer_id) REFERENCES Customer(customer_id)
);


CREATE TABLE Item
(
        item_id INT PRIMARY KEY,
    product_id INT,
    cart_id INT,
    quantity INT,
    FOREIGN KEY (product_id) REFERENCES Product(product_id),
    FOREIGN KEY (cart_id) REFERENCES ShoppingCart(cart_id)
);
```

Schema Diagram:

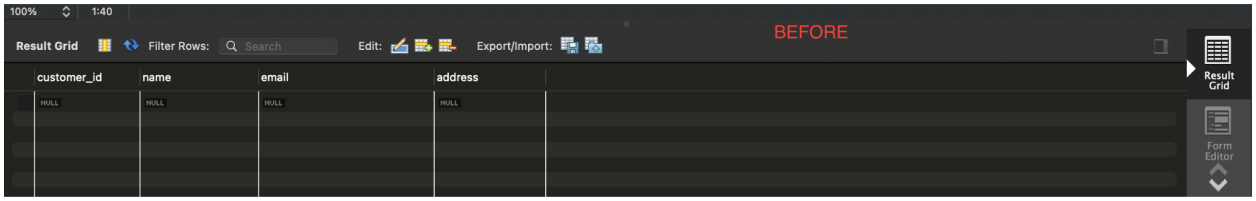


Insert Tuples Into Tables (With Before & After Commands)

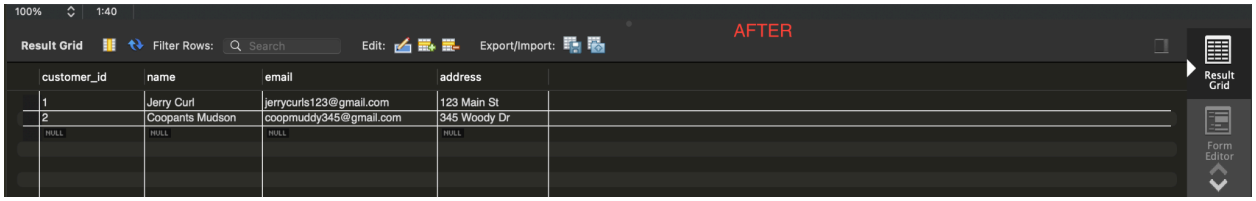
SQL Command:
INSERT INTO Customer (customer_id, name, email, address)
VALUES
(1, 'Jerry Curl', 'jerrycurls123@gmail.com', '123 Main St'),

(2, 'Coopants Mudson', 'coopmuddy345@gmail.com', '345 Woody Dr');

Before Command:



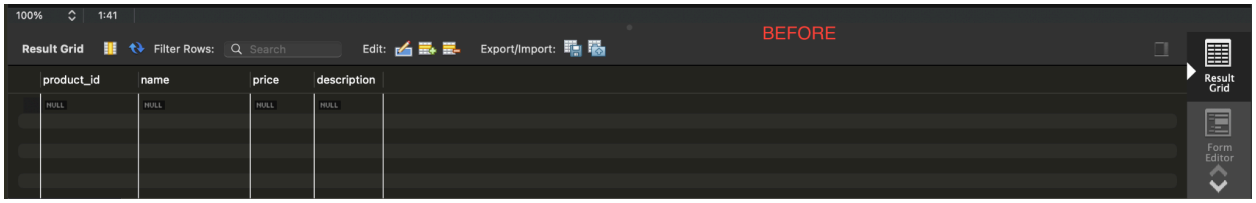
After Command:



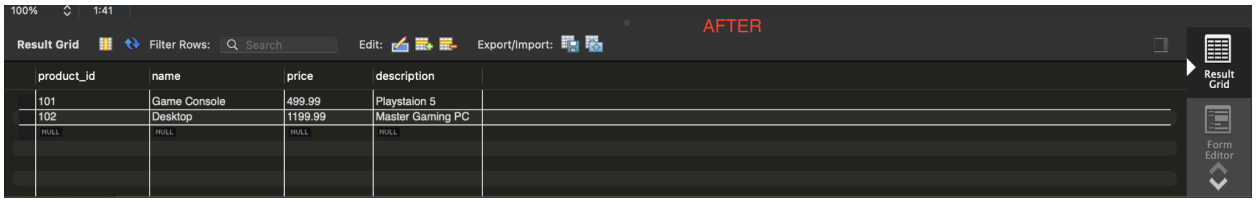
SQL Command:

```
INSERT INTO Product (product_id, name, price, description)
VALUES
        (101, 'Game Console', 499.99, 'Playstaion 5'),
        (102, 'Desktop', 1199.99, 'Master Gaming PC');
```

Before Command:



After Command:

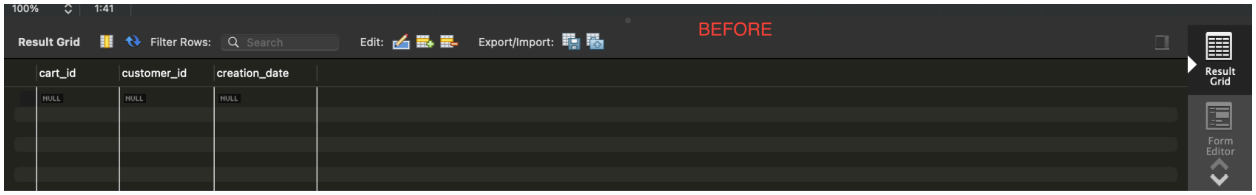


SQL: Command

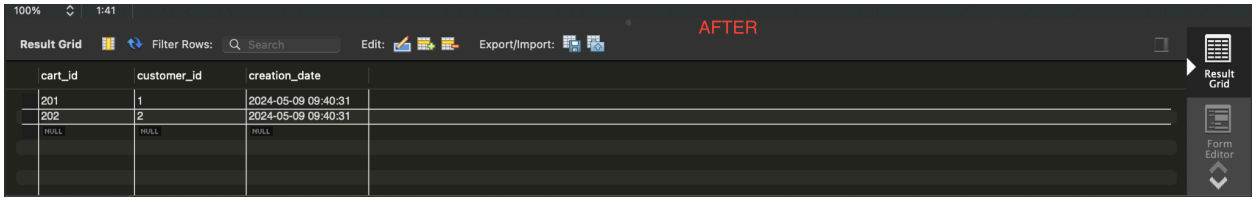
```
INSERT INTO ShoppingCart (cart_id, customer_id)
VALUES
        (201, 1),
```

(202, 2);

Before Command:



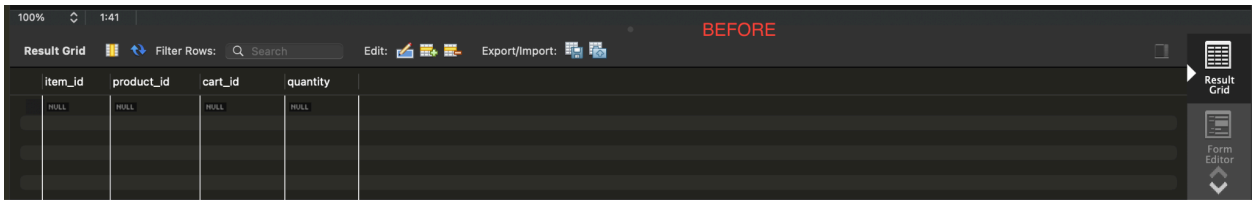
After Command:



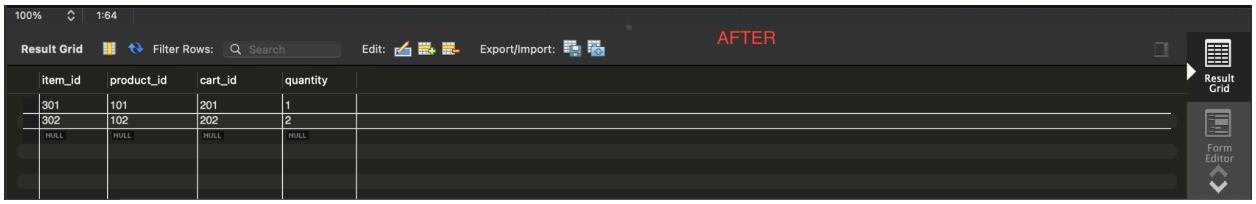
SQL Command:

```
INSERT INTO Item (item_id, product_id, cart_id, quantity)
VALUES
        (301, 101, 201, 1),
    (302, 102, 202, 2);
```

Before Command:



After Command:



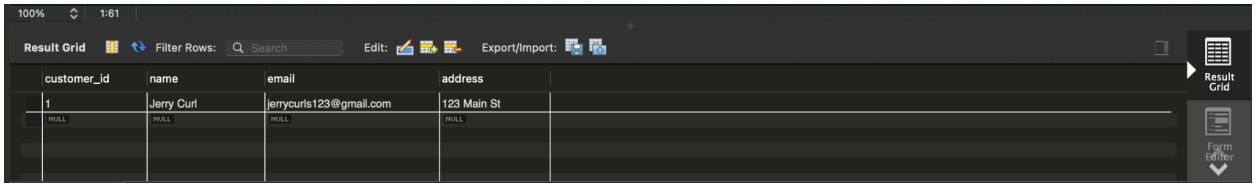
Execute at least 2 SQL Commands using WHERE, more than one table in FROM, SET operation, aggregate function and/or GROUP BY, SUBQUERY, EXISTS or UNIQUE, and WITH:

Use WHERE:

Query in Plain English: Retrieve the details of the customer with the email ‘jerrycurls123@gmail.com’

SQL Command: SELECT * FROM Customer WHERE email = 'jerrycurls123@gmail.com';

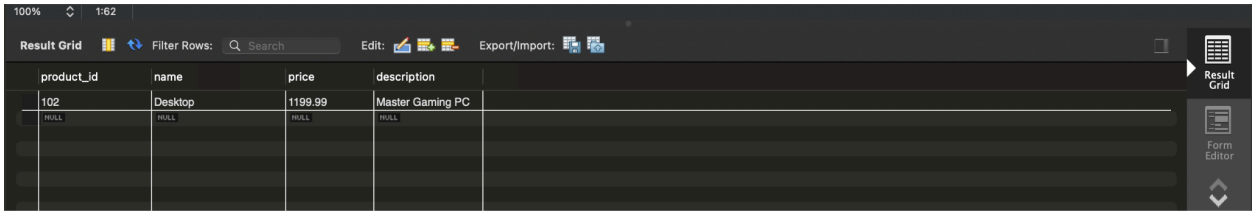
Result:



Query in Plain English: Retrieve the details of the products with a price greater than 1000

SQL Command: SELECT * FROM Product WHERE price > 1000;

Result:



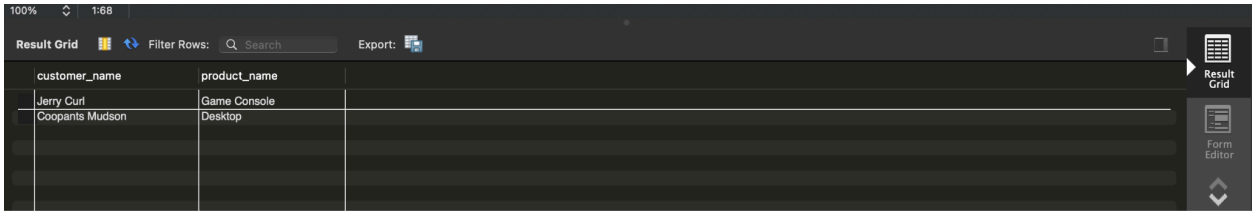
Use more than one table in FROM:

Query in English: Retrieves customers' names along with the products they have in their shopping carts.

SQL Command:

```
SELECT c.name AS customer_name, p.name AS product_name
FROM ShoppingCart sc
JOIN Customer c ON sc.customer_id = c.customer_id
JOIN Item i ON sc.cart_id = i.cart_id
JOIN Product p ON i.product_id = p.product_id;
```

Result:



Query in English: Retrieves the names, prices, and quantities of products associated with a specific shopping cart

SQL Command:
SELECT p.name AS product_name, p.price, i.quantity
FROM Item i
JOIN Product p ON i.product_id = p.product_id
WHERE i.cart_id = 202;

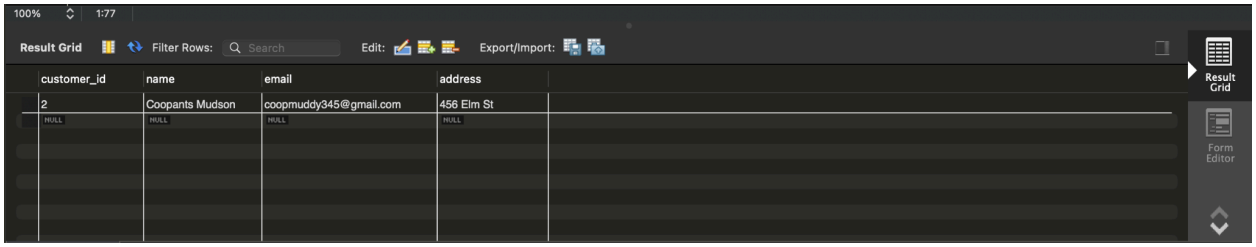
Result:

Use SET operation:

Query in English: Updates the address of the customer with id 2 to ‘456 Elm St’

SQL Command: UPDATE Customer SET address = '456 Elm St' WHERE customer_id = 2;

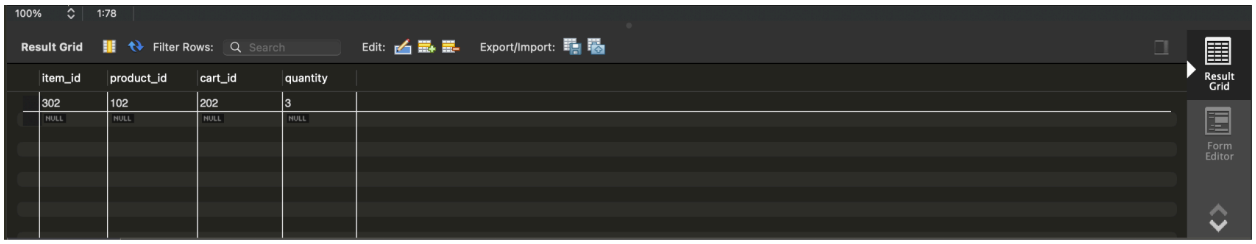
Result:



Query in English: Updates the quantity of a specific item in the Item table to a new value

SQL Command: UPDATE Item SET quantity = 3 WHERE item_id = 302;

Results:



Use aggregate function and/or GROUP BY:

Query in English: Calculates the total price of all items in a specific shopping cart.

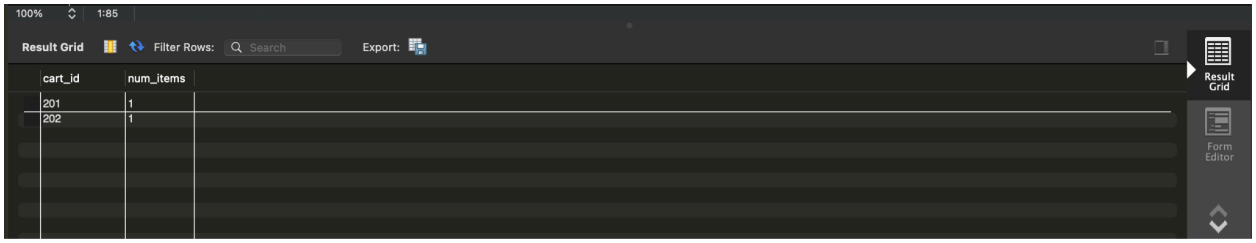
SQL Command:
SELECT SUM(p.price * i.quantity) AS total_price
FROM Item i
JOIN Product p ON i.product_id = p.product_id
WHERE i.cart_id = 202;

Result:

Query in English: Retrieves the number of items in each shopping cart

SQL Command:
SELECT cart_id, COUNT(*) AS num_items
FROM Item
GROUP BY cart_id;

Result:



Use SUBQUERY:

Query in English: Retrieves the names of customers who have items in their shopping cart with the cart ID equal to 202

SQL Command:
SELECT name
FROM Customer
WHERE customer_id IN (SELECT customer_id FROM ShoppingCart WHERE cart_id = 202);

Result:

Query in English: Retrieves the names and prices of products that are present in the Item table associated with a specific shopping cart

SQL Command:
SELECT name, price
FROM Product
WHERE product_id IN (SELECT product_id FROM Item WHERE cart_id = 201);

Result:

Use EXISTS OR UNIQUE

Query in English: Checks whether there are any items associated with a specific shopping cart

SQL Command: SELECT CASE WHEN EXISTS (SELECT * FROM Item WHERE cart_id = 203) THEN 'Yes' ELSE 'No' END AS items_exist;

Result:

Query in English: Provides a summary status of the customers and their shopping carts

SQL Command:
SELECT CASE WHEN NOT EXISTS (SELECT * FROM ShoppingCart) THEN 'No customers'
WHEN EXISTS (SELECT * FROM Customer WHERE customer_id NOT IN (SELECT customer_id FROM ShoppingCart)) THEN 'Customers without items'
ELSE 'All customers have items' END AS status;

Result:

Use WITH

Query in English: Retrieves the names, emails, and the number of items in the shopping cart for each customer

SQL Command:

```
WITH CartItems AS (
    SELECT customer_id, COUNT(*) AS num_items
    FROM ShoppingCart sc
    JOIN Item i ON sc.cart_id = i.cart_id
    GROUP BY customer_id
)
SELECT c.name, c.email, ci.num_items
FROM Customer c
JOIN CartItems ci ON c.customer_id = ci.customer_id;
```

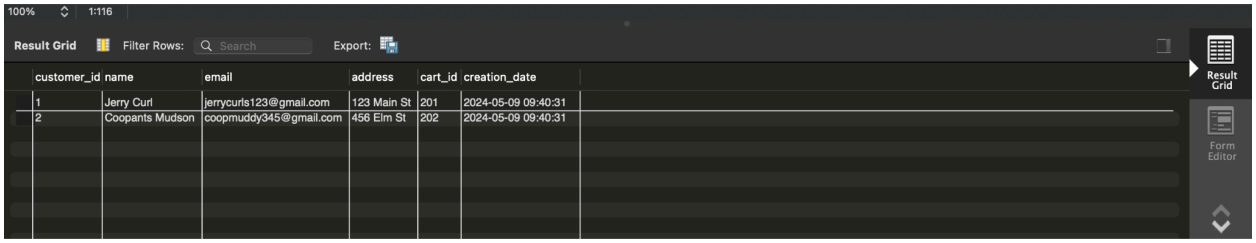
Result:

Query in English: Retrieves information about customers and their corresponding shopping carts

SQL Command:

```
 WITH CustomerCarts AS (
    SELECT c.*, sc.cart_id, sc.creation_date
    FROM Customer c
    LEFT JOIN ShoppingCart sc ON c.customer_id = sc.customer_id
)
SELECT * FROM CustomerCarts;
```

Result:

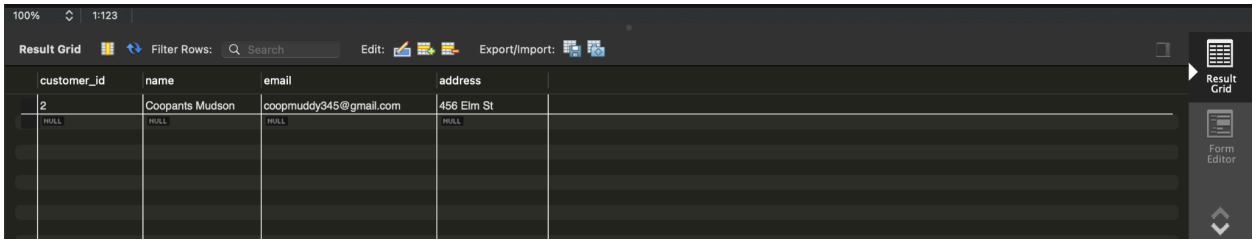


Execute at least 2 SQL commands that change the value(s) of some attributes using some conditions:

Query in Plain English: Updates the address of a specific customer in the customer table, setting the address to ‘456 Elm St’ for customer whose customer_id equals 2

SQL Command:
```
UPDATE Customer
SET address = '456 Elm St'
WHERE customer_id = 2;
```

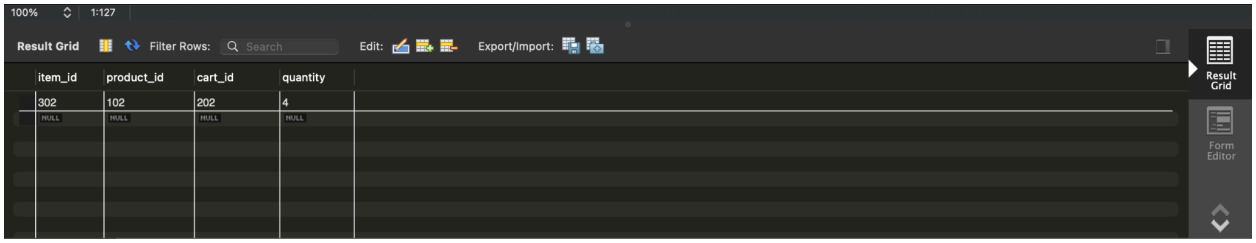
Result:



Query in Plain English: Increase the quantity of item with ID 302 by 1

SQL Command:
```
UPDATE Item
SET quantity = quantity + 1
WHERE item_id = 302;
```

Result:

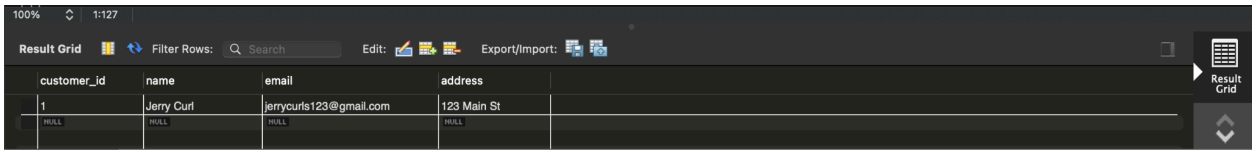


Execute at least 5 SQL commands that you think are important or necessary for your application:

Query in Plain English: Retrieves all columns from Customer table where email address is ‘jerrycurls123@gmail.com’

SQL Command: SELECT * FROM Customer WHERE email = 'jerrycurls123@gmail.com';

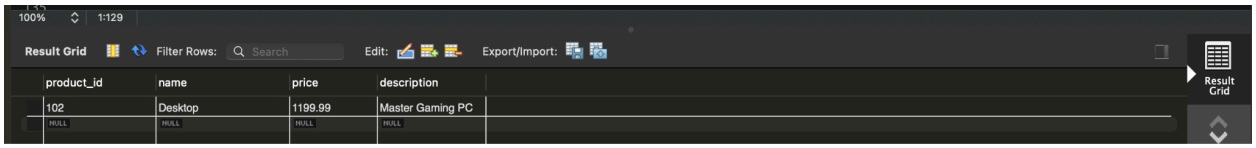
Result:



Query in Plain English: Selects products with prices greater than 1000

SQL Command: SELECT * FROM Product WHERE price > 1000;

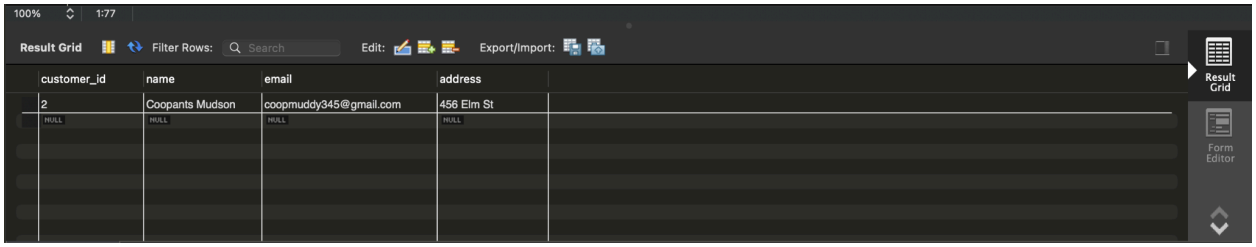
Result:



Query in Plain English: Updates the address of the customer with id 2 to ‘456 Elm St’

SQL Command: UPDATE Customer SET address = '456 Elm St' WHERE customer_id = 2;

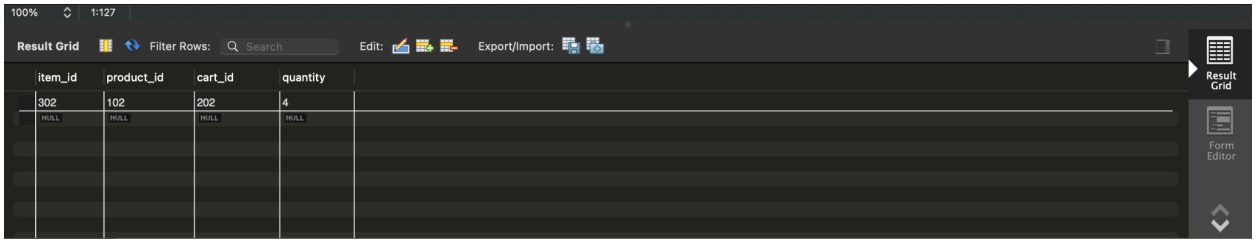
Result:



Query in English: Increments the quantity of item with ID 302 by 1

SQL Command: UPDATE Item SET quantity = quantity + 1 WHERE item_id = 302;

Result:



Query in English: Calculates the total price of all items in shopping cart with ID 202 by multiplying each item's price with its quantity, then summing them up

SQL Command:
SELECT SUM(p.price * i.quantity) AS total_price
FROM Item i
JOIN Product p ON i.product_id = p.product_id
WHERE i.cart_id = 202;

Result: