Guitar Store (The Emporium): E-Commerce Database

Alexander Abou Khir

Old Dominion University

IT450 Database Concepts

Professor Xianrong Zheng

April 13, 2025

1. Introduction

This database shows a guitar store named the Emporium that manages inventory of guitars, customer details and purchase orders.

This exists to manage structure data for products, customers and their orders. It creates reliability of tracking stock and purchases, this code can be used for other products, just change up the names of the variables listed.

The databases were built by using MariaDB, a DBMS that's open source. SQL was the language that created the tables, records and queries for this project.

2. Table Overview

PRODUCTS table:

```
1  CREATE TABLE PRODUCTS (
2  product_id INT NOT NULL PRIMARY KEY,
3  product_name VARCHAR(45) NOT NULL,
4  product_price NUMERIC(10,2),
5  quantity_in_stock INTEGER);
6
7  INSERT INTO PRODUCTS VALUES(1001, 'Stratocaster',250.99,24);
8  INSERT INTO PRODUCTS VALUES(1002, 'Telecaster',180.99,25);
9  INSERT INTO PRODUCTS VALUES(1003, 'Acoustic Guitar',200.99,17);
10 INSERT INTO PRODUCTS VALUES(1004, 'Jagaur',650.99,6);
11 INSERT INTO PRODUCTS VALUES(1005, 'Les Paul',1599.99,3);
12 INSERT INTO PRODUCTS VALUES(1006, 'Jazzmaster',800.99,8);
13 INSERT INTO PRODUCTS VALUES(1007, 'Esquire',170.99,21);
14 INSERT INTO PRODUCTS VALUES(1008, 'Mustang',429.99,7);
15 INSERT INTO PRODUCTS VALUES(1009, 'ES-335',599.99,2);
16 INSERT INTO PRODUCTS VALUES(1010, 'SG Standard',1099.99,3);
17 SELECT * FROM PRODUCTS
18
```

PRODUCTS (10r × 4c)

#	product_id	product_name	product_price	quantity_in_stock
1	1,001	Stratocaster	250.99	24
2	1,002	Telecaster	180.99	25
3	1,003	Acoustic Guitar	200.99	17
4	1,004	Jagaur	650.99	6
5	1,005	Les Paul	1,599.99	3
6	1,006	Jazzmaster	800.99	8
7	1,007	Esquire	170.99	21
8	1,008	Mustang	429.99	7
9	1,009	ES-335	599.99	2
10	1,010	SG Standard	1,099.99	3

The PRODUCTS table consists of product_id (PK), product_name, product_price, and quantity_in_stock

~~~~~~

CUSTOMERS Table:

```
CREATE TABLE CUSTOMERS (
customer_id INTEGER NOT NULL PRIMARY KEY,
customer_name VARCHAR(60),
customer_address VARCHAR(75),
phone_number CHAR(12));

INSERT INTO CUSTOMERS VALUES(18, 'Eric Munsher', '1214 Liverstone Dr., Norfolk, VA, 19561', '757-505-5214');
INSERT INTO CUSTOMERS VALUES(19, 'Nugen Danes', '0923 Marsh Rd, Chesapeake, VA, 23436', '757-109-7683');
INSERT INTO CUSTOMERS VALUES(20, 'Harley Cramer', '1100 Guts Ln., Virginia Beach, VA, 41553', '757-100-1999');
INSERT INTO CUSTOMERS VALUES(21, 'Cristie Langley', '3478 Darth Ave., Portsmouth, VA, 39952', '757-895-1547');
INSERT INTO CUSTOMERS VALUES(22, 'Barren Yogner', '1096 Ontario Dr., Hampton, VA, 27896', '757-311-3596');
INSERT INTO CUSTOMERS VALUES(23, 'Evelyn Blue', '1984 Fighto Ln., Newport News, VA, 88751', '757-821-7653');
INSERT INTO CUSTOMERS VALUES(24, 'Philipe Gustavo', '6786 Brighton Rd, Norfolk, VA, 31589', '757-249-5050');
INSERT INTO CUSTOMERS VALUES(25, 'Hanson Dane', '9745 Mason Rd, Suffolk, VA, 78495', '757-371-7892');
INSERT INTO CUSTOMERS VALUES(26, 'Jason Porter', '8794 Yung St., Williamsburg, VA, 21894', '757-817-7100');
INSERT INTO CUSTOMERS VALUES(27, 'Wyatt Carter', '5644 Horse Rd., Norfolk, VA, 67125', '757-978-0024');

SELECT * FROM CUSTOMERS;
```

#	customer_id	customer_name	customer_address	phone_number
1	18	Eric Munsher	1214 Liverstone Dr., Norfolk, VA, 19561	757-505-5214
2	19	Nugen Danes	0923 Marsh Rd, Chesapeake, VA, 23436	757-109-7683
3	20	Harley Cramer	1100 Guts Ln., Virginia Beach, VA, 41553	757-100-1999
4	21	Cristie Langley	3478 Darth Ave., Portsmouth, VA, 39952	757-895-1547
5	22	Barren Yogner	1096 Ontario Dr., Hampton, VA, 27896	757-311-3596
6	23	Evelyn Blue	1984 Fighto Ln., Newport News, VA, 88751	757-821-7653
7	24	Philipe Gustavo	6786 Brighton Rd, Norfolk, VA, 31589	757-249-5050
8	25	Hanson Dane	9745 Mason Rd, Suffolk, VA, 78495	757-371-7892
9	26	Jason Porter	8794 Yung St., Williamsburg, VA, 21894	757-817-7100
10	27	Wyatt Carter	5644 Horse Rd., Norfolk, VA, 67125	757-978-0024

The CUSTOMERS table concists of customer_id (PK), customer_name, customer_address, phone_number

~~~~~~

ORDERS Table:

```sql
CREATE TABLE ORDERS (
  order_id INT NOT NULL PRIMARY KEY,
  customer_id INT,
  product_id INT,
  quantity INT,
  order_date DATE,
  FOREIGN KEY (customer_id) REFERENCES CUSTOMERS(customer_id),
  FOREIGN KEY (product_id) REFERENCES PRODUCTS(product_id));

INSERT INTO ORDERS VALUES (501, 18, 1001, 1, '2025-01-02');
INSERT INTO ORDERS VALUES (502, 19, 1005, 1, '2025-04-17');
INSERT INTO ORDERS VALUES (503, 20, 1003, 2, '2024-12-02');
INSERT INTO ORDERS VALUES (504, 21, 1008, 1, '2025-04-03');
INSERT INTO ORDERS VALUES (505, 22, 1009, 1, '2024-11-30');
INSERT INTO ORDERS VALUES (506, 23, 1004, 2, '2025-02-14');
INSERT INTO ORDERS VALUES (507, 24, 1002, 1, '2024-12-22');
INSERT INTO ORDERS VALUES (508, 25, 1007, 1, '2025-04-07');
INSERT INTO ORDERS VALUES (509, 26, 1006, 3, '2025-03-08');
INSERT INTO ORDERS VALUES (510, 27, 1010, 1, '2024-12-20');

SELECT * FROM ORDERS;
```

ORDERS (10r × 5c)

#	order_id	customer_id	product_id	quantity	order_date
1	501	18	1,001	1	2025-01-02
2	502	19	1,005	1	2025-04-17
3	503	20	1,003	2	2024-12-02
4	504	21	1,008	1	2025-04-03
5	505	22	1,009	1	2024-11-30
6	506	23	1,004	2	2025-02-14
7	507	24	1,002	1	2024-12-22
8	508	25	1,007	1	2025-04-07
9	509	26	1,006	3	2025-03-08
10	510	27	1,010	1	2024-12-20

The ORDERS table consists of order_id (PK), customer_id (FK), product_id (FK), quantity, order_date

~~~~~~

The designs of these tables show all three levels of normalization

1NF: All fields contain only atomic values
~~~~~~

2NF: All non-key attributes are fully functionally dependant on the entire primary key.

3NF: There are no transitive dependencies. All fields relate directly to the primary key.

This was shown in the ORDERS table uses foreign keys from customer-id and product_id.

3. **Code Demonstration of Query Screen and Table mapping**

Customer Lookup Query: Retrieves customer name, address, and phone number by using customer_id.

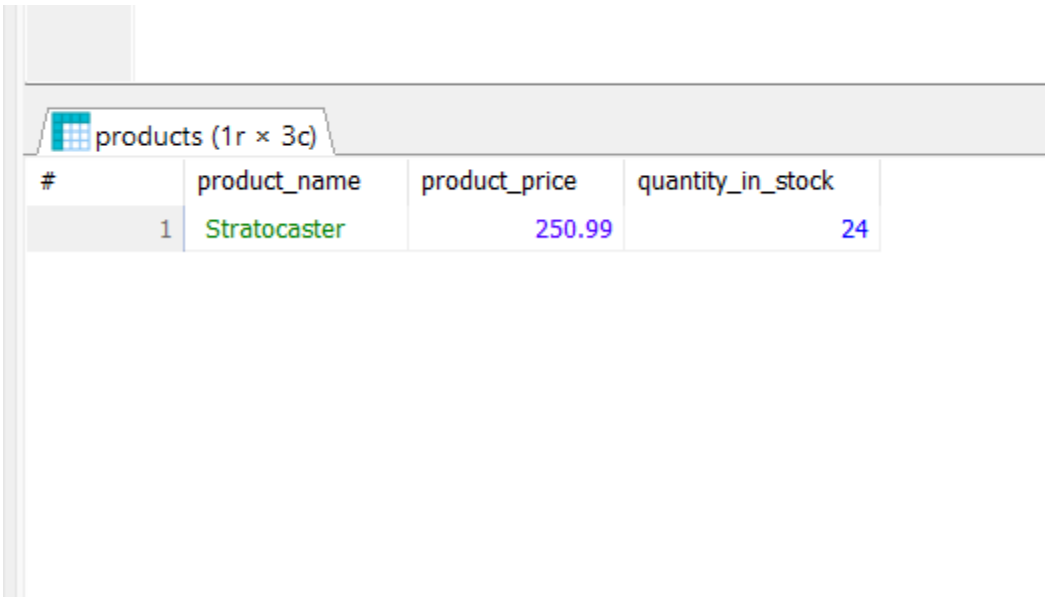
```
1  SELECT customer_name, customer_address, phone_number
2      #Table:, fields: customer_name, customer_address, phone_number
3  FROM customers
4      #Table: customers
5  WHERE customer_id =18;
6      #Table: customers, field: customer_id
```

customers (1r × 3c)

#	customer_name	customer_address	phone_number
1	Eric Munsher	1214 Liverstone Dr., Norfolk, VA, 19561	757-505-5214

Product Stock Query: Displays the current stocks for guitars and the price for certain guitars.

```sql
SELECT product_name, product_price, quantity_in_stock
    #Table:products, fields: product_name, product_price, quantity_in_stock
FROM products
    #Table: Products
WHERE product_id =1001;
    #Table: Products, field: product_id
```



products (1r × 3c)

#	product_name	product_price	quantity_in_stock
1	Stratocaster	250.99	24

Order Detail Query: Retrieves complete order details using the order_id.

```sql
SELECT order_id, customer_id, product_id, quantity, order_date
    #Table: orders, fields: order_id, customer_id, product_id, quantity, order_date
FROM orders
WHERE order_id = 501;
    #Table: orders, field: order_id
```

orders (1r × 5c)

#	order_id	customer_id	product_id	quantity	order_date
1	501	18	1,001	1	2025-01-02

4. Reflections & Final Thoughts

This project helped me have a better understanding of database normalization, relational table design, SQL query structuring, and Using MariaDB for real-world applications if I do have my own business one day. I enjoyed making it as well with the fun guitar store idea I had, thanks for making this assignment.