**README - E-Commerce**

First, I created DATABASE eCommerce.

Then I create four tables: categories, products, customers and sales.

My main thesis is that each product belongs to one category and that each sale involves one customer and one product.

*Create Tables*

CREATE TABLE categories (

categoryID INT AUTO\_INCREMENT PRIMARY KEY,

categoryName VARCHAR(25) NOT NULL

);

CREATE TABLE products (

productID INT AUTO\_INCREMENT PRIMARY KEY,

productName VARCHAR(50) NOT NULL,

categoryID INT NOT NULL,

price DECIMAL(10, 2) NOT NULL,

stock INT NOT NULL,

FOREIGN KEY (categoryID) REFERENCES categories(categoryID)

);

CREATE TABLE customers (

customerID INT AUTO\_INCREMENT PRIMARY KEY,

firstName VARCHAR(50) NOT NULL,

lastName VARCHAR(25) NOT NULL,

eMail VARCHAR(100) UNIQUE NOT NULL

);

CREATE TABLE sales (

saleID INT AUTO\_INCREMENT PRIMARY KEY,

customerID INT NOT NULL,

productID INT NOT NULL,

quantity INT NOT NULL,

saleDate DATE,

FOREIGN KEY (customerID) REFERENCES customers(customerID),

FOREIGN KEY (productID) REFERENCES products(productID)

);

With this tables that I created, we have categories that helps us group products, products that hold details of a product, customers table with basic details about them and sales that connect customer and a product to show a transaction.

Then I added a sample data to each table, including 3 categories, 10 products, 5 customers, and 10 sales transactions.

INSERT INTO categories (categoryName)

VALUES

(“garden”),

(“toys”),

(“shoes”);

INSERT INTO products (productName, categoryID, price, stock)

VALUES

("firelighters", 1, 14.99, 20),

("ratPoison", 1, 7.99, 17),

("gloves", 1, 5.75, 54),

("lego", 2, 13.00, 15),

("paintForKids", 2, 19.13, 34),

("clogs", 3, 36.00, 4),

("skechers", 3, 39.99, 13),

("puma", 3, 19.99, 2),

("boardGame", 2, 19.49, 32),

("birdFeed", 1, 10.99, 6);

INSERT INTO customers (firstName, lastName, Email)

VALUES

("Ella", "Hubbard", "ella.hubbard@gmail.com"),

("Chris", "Spraggins", "chris.spraggins@gmail.com"),

("Tess", "Gill", "tess.gill@gmail.com"),

("Dana", "Wells", "dana.wells@gmail.com"),

("Horace", "Kinsman", "horace.kinsman@gmail.com");

INSERT INTO sales (customerID, productID, quantity, saleDate)

VALUES

(1, 10, 1, "2022-01-01"),

(2, 3, 2, "2024-12-07"),

(3, 9, 1, "2024-03-23"),

(4, 6, 3, "2023-04-01"),

(5, 4, 2, "2025-01-01"),

(1, 8, 3, "2024-02-27"),

(2, 7, 1, "2021-11-10"),

(3, 1, 1, "2021-03-21"),

(4, 2, 2, "2025-01-04"),

(5, 5, 2, "2023-04-24");

Explanation: Ella bought 1 bird feed on the 2022-01-01, Chris bought 2 gloves on the 2024-12-07 etc…

Below are SQL queries that meet the project requirements:

* *Display all products with their prices and stock quantities*

SELECT productName, price, stock

FROM products;

With this query I can show what products are available, their cost, and how many are in stock right now.

* *Identify products low in stock (stock < 5)*

SELECT productName, stock

FROM products

WHERE stock < 5;

With this query I can find items that are low on stocking.

* *Calculate total revenue generated from sales*

SELECT SUM(products.price \* sales.quantity) AS totalRevenue

FROM sales

JOIN products ON sales.productID = products.productID;

With this query we can find out how much money all sales brought in.

* *List the top best-selling products (descending order)*

SELECT p.productName, SUM(s.quantity) AS totalSold

FROM sales s

JOIN products p ON s.productID = p.productID

GROUP BY s.productID

ORDER BY totalSold DESC;

This way we can see which products are selling good.

* *Show all purchases made by a specific customer (I will use customerID 3 (Tess))*

SELECT c.firstName, c.lastName, p.productName, s.quantity, s.saleDate

FROM sales s

JOIN customers c ON s.customerID = c.customerID

JOIN products p ON s.productID = p.productID

WHERE c.customerID = 3;

Using this query, I can see what a customer bought and when.

* *Calculate the total number of products sold in each category*

SELECT c.categoryName, SUM(s.quantity) AS totalSold

FROM sales s

JOIN products p ON s.productID = p.productID

JOIN categories c ON p.categoryID = c.categoryID

GROUP BY c.categoryID;

With this query we can see how many items have sold in each category.

*Summarise:*

1. Create the Database: Run the CREATE TABLE statements in your SQL client.
2. Populate the Data: Run the INSERT statements to add sample records.
3. Run Queries: Use the SELECT statements to check data and analyze results.