Makara Ramoabi

20240045

Database

1. Scenario Analysis
An e-commerce platform needs a well-structured database schema to manage its operations efficiently. The schema must store information about customers, products, orders, payments, and inventory. For example, a **customer's** table contains user details such as name, email, and address, while a **Products** table stores product names, prices, and stock levels. An **Orders** table links customers to purchased products using foreign keys, ensuring clear relationships. The **Order Details** table captures specific product quantities within each order. A **Payments** table records transaction, connecting them back to the corresponding orders. Designing the schema this way avoids redundancy, improves data consistency, and makes querying more efficient. With proper indexing, searching for customer orders or tracking inventory updates becomes fast and reliable. Additionally, the schema supports scalability, allowing the business to add new modules such as promotions, reviews, or shipping details without disrupting existing data relationships.

---

2. Concept Research: Normalization
Normalization is the process of organizing a database to reduce redundancy and improve data integrity. It involves dividing large, complex tables into smaller, related ones and defining clear relationships between them. By applying rules known as **normal forms** (1NF, 2NF, 3NF, etc.), normalization ensures that data is stored logically, with minimal duplication. This reduces anomalies during updates, insertions, or deletions. For example, storing customer addresses in a separate table avoids repeating the same information across multiple orders. Normalization improves efficiency, consistency, and scalability, making databases easier to maintain and query while ensuring reliable, accurate results.

3. Tool Practice: Map ER Diagram to Schema + Reflection
Mapping an ER diagram to a schema involves converting entities, attributes, and relationships into structured tables. For example, in an e-commerce ER diagram:

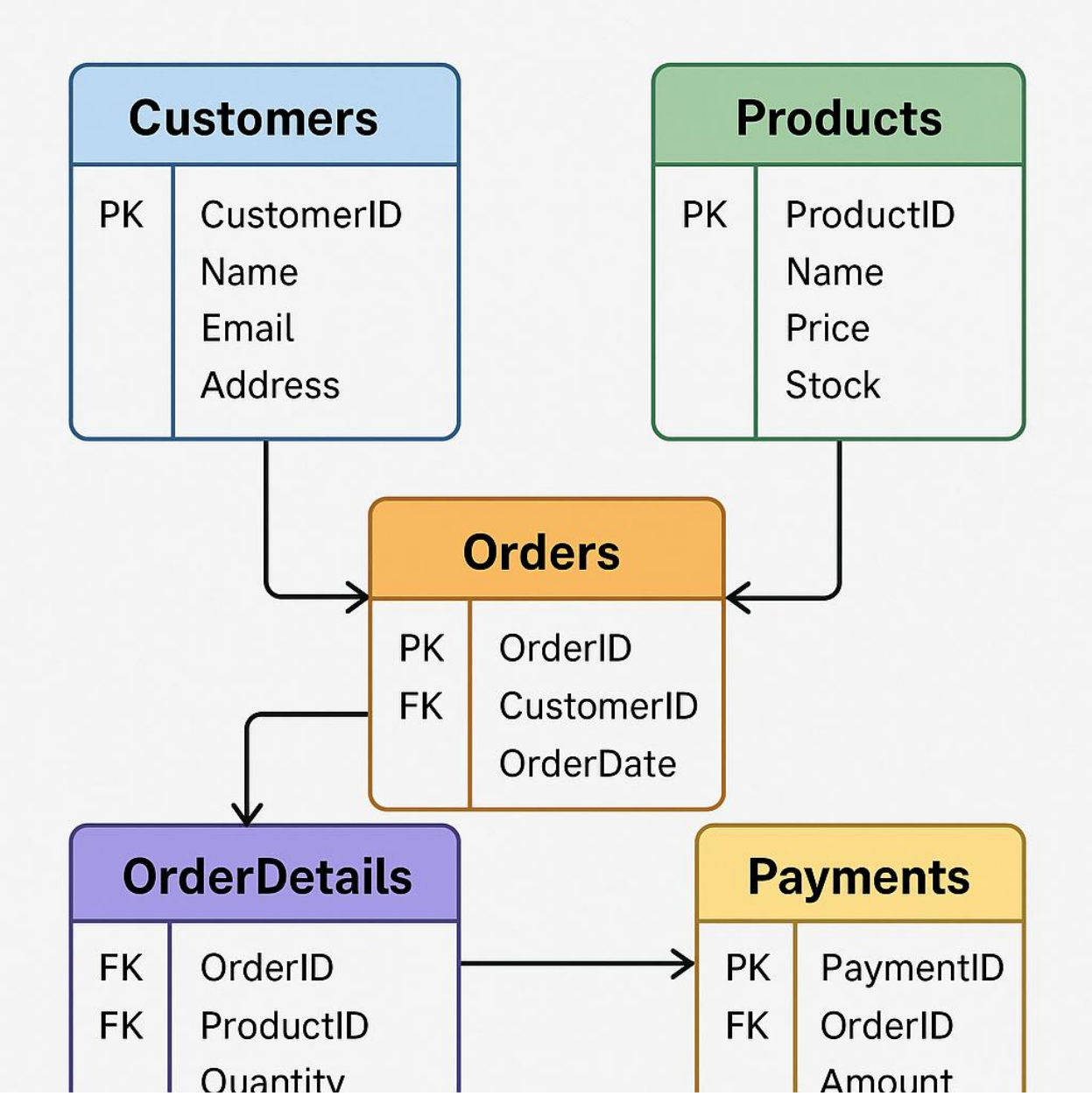
- **Customer** → Customers (CustomerID, Name, Email, Address)
- **Product** → Products (ProductID, Name, Price, Stock)
- **Order** → Orders (OrderID, CustomerID, Order Date)
- **Order Details** → Order Details (OrderID, ProductID, Quantity)
- **Payment** → Payments (Payment, OrderID, Amount, Method)

Reflection:
This process helped me understand how abstract models become practical database designs. Relationships like one-to-many (Customer–Orders) and many-to-many (Order–Products) are clarified through primary and foreign keys. It showed me how schemas prevent redundancy while enabling efficient queries.

---

4. Diagram Design: Schema Diagram in Canva + Explanation

- Place **Customers**, **Products**, **Orders**, **Order Details**, and **Payments** as labeled boxes.
- Inside each box, list key attributes (e.g., Customers: CustomerID [PK], Name, Email).
- Draw arrows/lines to represent relationships:
    - Customers → Orders (1-to-Many).
    - Orders → Order Details (1-to-Many).
    - Products → Order Details (1-to-Many).
    - Orders → Payments (1-to-1 or 1-to-Many).
      This diagram visually explains how data is structured and connected, making it easier to understand schema logic.


Customers
PK
CustomerID
Name
Email
Address
Products
PK
ProductID
Name
Price
Stock
Orders
PK
OrderID
FK
CustomerID
OrderDate
OrderDetails
FK
OrderID
FK
ProductID
Quantity
Payments
PK
PaymentID
FK
OrderID
Amount