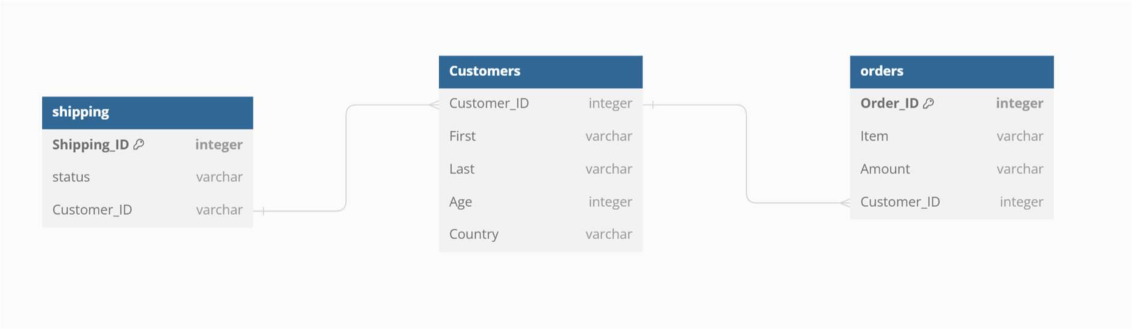
Data Modelling

We will start the process of data modelling using the original three datasets I have been provided. The goal is to create a comprehensive data model that includes the necessary tables, their relationships, and constraints. This will lead to the development of an ER (Entity-Relationship) diagram, which will visually represent the data model. Afterward, I will create a Data Flow Diagram (DFD) to show how data moves through the system.

Step 1: Understanding the Original Datasets

Let's start with the structure of my data. The three datasets I am working with are:

1. **Customers**: Contains information about the customers.
2. **Orders**: Contains details of customer orders.
3. **Shipping**: Contains shipping information related to customer orders.



shipping	
Shipping_ID	integer
status	varchar
Customer_ID	varchar

Customers	
Customer_ID	integer
First	varchar
Last	varchar
Age	integer
Country	varchar

orders	
Order_ID	integer
Item	varchar
Amount	varchar
Customer_ID	integer

Step 2: Identify Entities and Attributes

In this step, we identify the main entities (tables) and their attributes (columns) from the datasets.

- **Customer Entity**:
  - Columns/Attribute: Customer_ID, First Name, Last Name, Email, Age, Country
- **Order Entity**:
  - Columns/Attribute: Order_ID, Customer_ID, Product_ID, Quantity, Amount, Order Date
- **Shipping Entity**:
  - Columns/Attribute: Shipping_ID, Order_ID, Shipping Date, Shipping Status, Customer_ID
- **Product Entity** (Derived from Orders data):
  - Columns/Attribute: Product_ID, Product Name

Step 3: Establish Relationships Between Entities

Next, we determine the relationships between these entities. The relationships are based on the common keys that link the tables together.

- **Customer ↔ Order**:
    - Relationship: A customer can place multiple orders, but each order is associated with one customer.
    - Type: One-to-Many
    - Key: Customer_ID
- **Customer ↔ Shipping**:
    - Relationship: An order can have one shipping record, but each shipping record is associated with one order.
    - Type: One-to-One
    - Key: Customer_ID
- **Order ↔ Product**:
    - Relationship: An order can include multiple products, and a product can appear in multiple orders.
    - Type: Many-to-Many (resolved using a join table)
    - Key: Product_ID in Order and Product

Step 4: Normalize the Data

Normalization is the process of organizing the data to minimize redundancy and dependency. We will ensure that each table is in at least the third normal form (3NF).

- **First Normal Form (1NF)**: Ensure that all attributes contain only atomic (indivisible) values.
- **Second Normal Form (2NF)**: Ensure that all non-key attributes are fully functionally dependent on the primary key.
- **Third Normal Form (3NF)**: Ensure that there are no transitive dependencies between non-key attributes.

After normalizing, our data model should look something like this:

- **Customer Table**:
    - Customer_ID (Primary Key)
    - First Name
    - Last Name
    - Age
    - Country_ID (Foreign Key)
- **OrderFact Table** (For resolving Many-to-Many relationship between Order and Product):
    - Order_ID (Primary Key)
    - Product_ID (Foreign Key)
    - Amount
    - Customer_ID (Foreign Key)

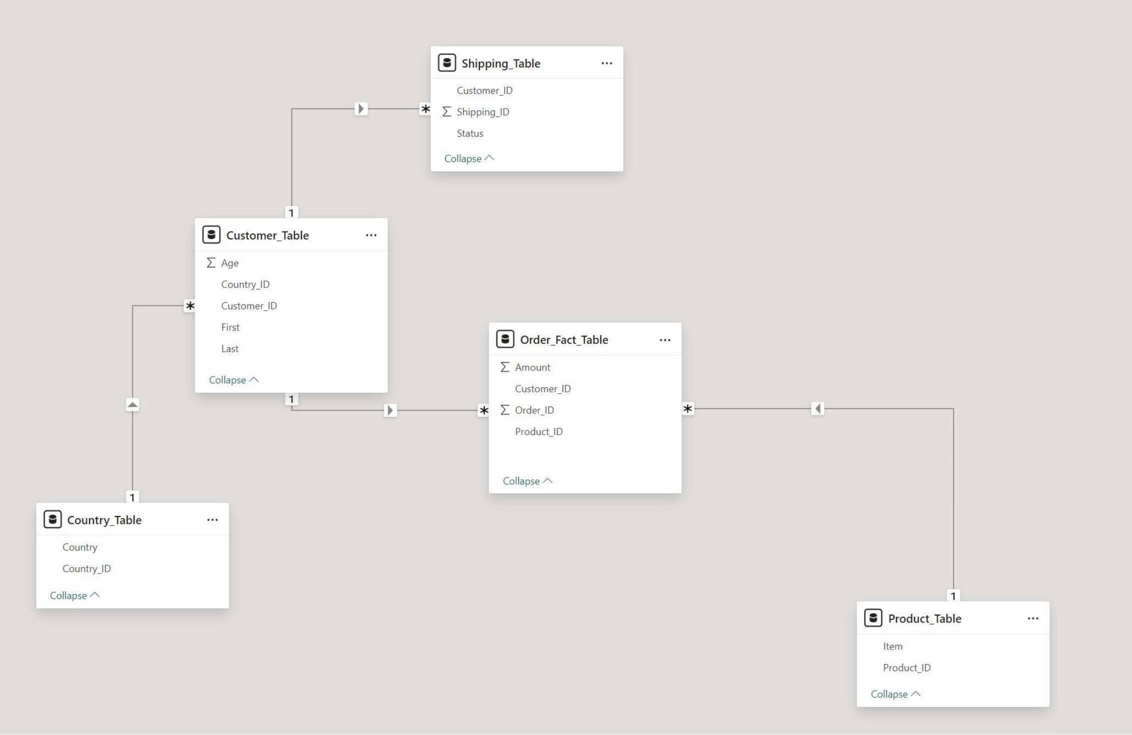
- **Shipping Table**:
    - Shipping_ID (Primary Key)
    - Customer_ID (Foreign Key)
    - Shipping Status
- **Product Table**:
    - Product_ID (Primary Key)
    - Item
- **Country Table** (For normalization):
    - Country_ID (Primary Key)
    - Country Name

Step 5: Define Primary and Foreign Keys

In each table, we identify the primary and foreign keys to establish the relationships between tables:

- **Primary Keys**:
    - Customer Table: Customer_ID
    - OrderFact Table: Order_ID
    - Shipping Table: Shipping_ID
    - Product Table: Product_ID
    - Country Table: Country_ID
- **Foreign Keys**:
    - OrderFact Table: Customer_ID (References Customer_ID in Customer Table), Product_ID (References Product_ID in Product Table)
    - Shipping Table: Customer_ID (References Customer_ID in Customer Table)
    - Customer Table: Country_ID (References Country_ID in Country Table)

ER Diagram



Entities:

- Customer
- Order_Fact
- Shipping
- Product
- Country

Relationships:

- Customer ↔ OrderFact (One-to-Many)
- Product ↔ OrderFact (One-to-Many)
- Customer ↔ Shipping (One-to-Many)
- Customer ↔ Country (Many-to-One)

Data Flow Diagram

Step 1: Identify the Key Processes

1. **Customer Management**:
   - This process handles all operations related to managing customer information, such as adding new customers, updating existing customer records, and retrieving customer details.
2. **Order Processing**:
   - This process involves capturing and managing customer orders, including recording order details, calculating totals, and updating inventory.
3. **Shipping Management**:
   - This process is responsible for managing the shipping of orders, including updating shipping statuses and tracking shipments.

Step 2: Identify the Data Stores

1. **Customer Database**:
   - Stores all customer-related information such as Customer_ID, Name, Email, Age, and Country.
2. **Order Database**:
   - Stores information related to customer orders, including Order_ID, Customer_ID, Product_ID, Quantity, Amount, and Order Date.
3. **Product Database**:
   - Contains details about products, such as Product_ID, Product Name, Category, and Price.
4. **Shipping Records**:
   - Stores all information related to the shipping of orders, including Shipping_ID, Order_ID, Customer_ID, Shipping Date, and Shipping Status.

Step 3: Identify External Entities

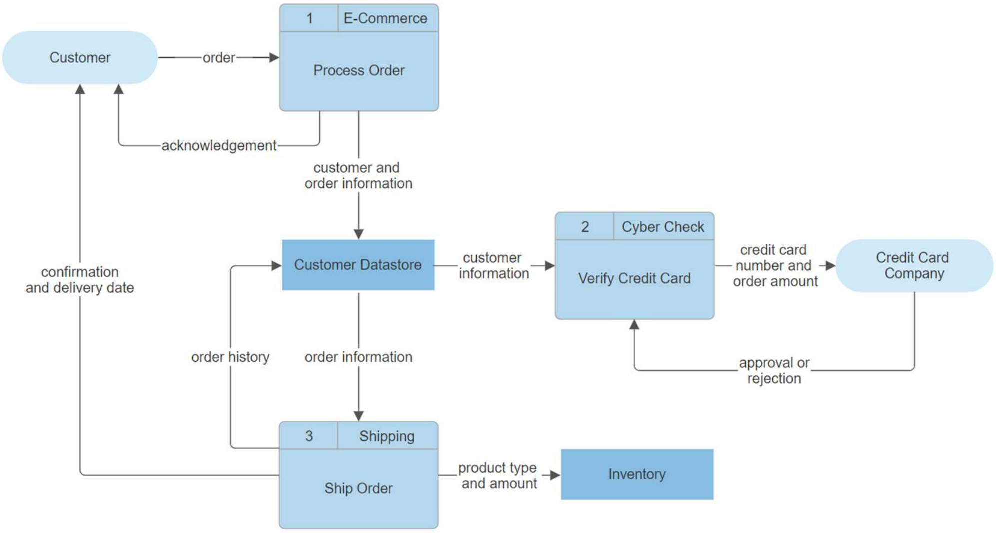
1. **Customers**:
   - External entity representing individuals who place orders.
2. **Suppliers**:
   - External entity representing suppliers that provide the products.
3. **Shipping Companies**:
   - External entity representing companies responsible for shipping the products to customers.

Step 4: Define Data Flows

1. **Customers to Customer Management**:
   - Data flow where customers provide their information (e.g., during sign-up) to be stored in the Customer Database.
2. **Customer Management to Customer Database**:
   - Customer data is stored in the Customer Database after processing.
3. **Customers to Order Processing**:

   - Data flow where customers place orders, which are processed and recorded in the Order Database.

4. **Order Processing to Order Database**:
   - Order details are stored in the Order Database after processing.
5. **Order Processing to Shipping Management**:
   - Order details are passed to the shipping process for further action.
6. **Shipping Management to Shipping Records**:
   - Shipping details are stored in the Shipping Records after processing.
7. **Shipping Management to Shipping Companies**:
   - Data flow where shipping details are sent to the shipping companies for execution.
8. **Suppliers to Product Database**:
   - Data flow where product information from suppliers is stored in the Product Database.



This is a Level 1 DFD, and we can define it further as per Level 2 DFD.