

**Transactions Table:**

Entities: Each row in the "transactions" table represents a specific transaction.

Attributes:

InvoiceNo: A unique identifier for each transaction (primary key).

StockCode: A code representing the stock/product associated with the transaction.

Quantity: The quantity of the product bought/sold.

InvoiceDate: The date of the transaction.

CustomerID: The identifier of the customer associated with the transaction.

Relationships:

Many-to-One: Each transaction is associated with one customer (via CustomerID).

Many-to-One: Each transaction is associated with one product (via StockCode).

**Product Table:**

Entities: Each row in the "product" table represents a specific product.

Attributes:

StockCode: A unique identifier for each product (primary key).

Description: A text description of the product.

UnitPrice: The price of one unit of the product.

Relationships:

One-to-Many: Each product can be associated with multiple transactions (via StockCode).

**Customer Table:**

Entities: Each row in the "customer" table represents a specific customer.

Attributes:

CustomerID: A unique identifier for each customer (primary key).

Country: The country associated with the customer.

Relationships:

One-to-Many: Each customer can be associated with multiple transactions (via CustomerID).

One-to-One: Each customer can be associated with one country.

Overall, the database has the following relationships:

Many-to-One relationship between "transactions" and "customer" tables (via CustomerID).

Many-to-One relationship between "transactions" and "product" tables (via StockCode).

One-to-Many relationship between "product" table and "transactions" table (via StockCode).

One-to-Many relationship between "customer" table and "transactions" table (via CustomerID).

These relationships enable the database to track and store information about transactions, products, and customers, allowing for analysis and retrieval of data based on different criteria.

**Database Constraints:**

Primary Key Constraint:

The "transactions" table has a primary key constraint on the "InvoiceNo" column, ensuring that each transaction has a unique identifier.

Not Null Constraint:

The "transactions" table enforces the "NOT NULL" constraint on the following columns: "InvoiceNo," "StockCode," and "CustomerID." These columns must have a value in every row, and null values are not allowed.

Primary Key Constraint and Uniqueness:

The "product" table has a primary key constraint on the "StockCode" column, ensuring that each product has a unique identifier.

No Specific Formatting Constraints:

There are no specific formatting constraints in the defined table. The data types used for attributes such as "StockCode," "Description," "Quantity," "InvoiceDate," "CustomerID," and "Country" allow for flexible data input without strict formatting requirements.

Views, Functions, and Procedures:

We are massaging our data through a python script to take an original .csv file and split this file into three separate tables to create a relational database. The three tables created are “transactions”, “products”, and “customers”.