**Part B:**

1. The dataset follows a simplified **Star schema**. Here's how the schema could be structured:

Fact Table: The fact table in this dataset would likely be the "Transaction" table, with columns such as TransactionNo, Date, ProductNo, Price, Quantity, CustomerNo, and Country. Each row in the fact table represents a unique sales transaction, with information about the specific product, customer, and transaction details.

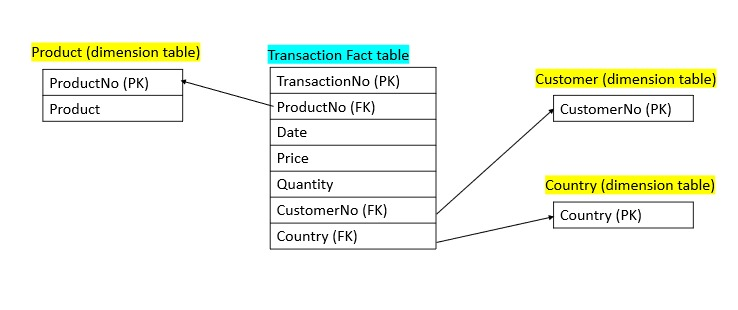
Dimension Tables: The dataset doesn't explicitly mention dimension tables, but based on the provided information, possible dimension tables could include:

"Product" dimension table: This table could contain information about each product, including the ProductNo and Product name.

"Customer" dimension table: This table could contain information about each customer, including the CustomerNo and possibly additional details such as customer demographics or segmentation.

"Country" dimension table: This table could provide additional details about each country, such as the country name and any relevant attributes or demographics.

In this schema, the fact table (Transaction) is at the center, representing the sales transactions, while the dimension tables (Product, Customer, and Country) surround it, providing additional descriptive information about the entities involved in the transactions.

1. ERD:
2. Use Case: Sales Analysis for an E-commerce Business

Scenario: An e-commerce business wants to analyze sales data to gain insights into customer behavior, product performance, and geographical trends.

Efficient Data Analysis: The star schema and ERD design offer efficiency in data analysis for this use case. The fact table "Transaction" contains essential information about each sales transaction, such as the transaction number, date, product details, customer information, and country. This consolidated and denormalized structure enables quick and straightforward queries on sales data without the need for complex joins across multiple tables.

In addition, the simplified structure of the star schema makes it easier for the database query optimizer to generate efficient query plans. With fewer tables and straightforward relationships, the optimizer can better utilize indexing, caching, and other optimization techniques to speed up data retrieval.

**Part C:**

1. ETL process in our E-commerce business transaction dataset:  
     
   a. Extract – this phase involves retrieving the data from its source, which in our case would be the online shop's sales transaction data. The dataset contains 500K rows and 8 columns, likely serves as the source for the extraction process. The data may be obtained from a database, flat files, APIs, or other sources where the sales transaction data is stored.  
     
   b. Transform - this phase may involve various data transformations, cleaning operations, aggregations, calculations, filtering, or formatting. For example, handling negative quantities related to cancelled transactions, converting data types, ensuring data consistency, or enriching the data with additional attributes.  
     
   C. Load - during the load process, the transformed data is inserted into the appropriate tables of the target schema, such as the fact table (e.g., "Transaction") and dimension tables (e.g., "Product," "Customer," "Country"). The data loading process may also involve indexing, partitioning, or other optimizations to ensure efficient storage and retrieval.

**In this ETL process, we would like to answer the question: What is the average transaction price for each customer? (not including cancelled transactions).  
In order to answer this question, we will create a new fact table, including the columns below: TransactionNo, TotalPrice, CustomerNo.   
Total price is the outcome of multiplication operation between Price and Quantity columns.**

*\*\* in order to run the code effectively, we minimize the dataset to 20K rows (instead of ~500K).*