To track the required data, we need to design an appropriate SQL database schema. Here is a step-by-step approach to designing an SQL database schema.

Define the Entities and their Attributes:

The first step is to define the entities and their attributes. Entities are the objects or concepts we want to store data at put. In this case, the entities are products, regions, time periods, customers, and inventory.

For each entity, we need to identify the attributes that define it. For example, the attributes for the product entity might include product name, description, price, and category. Similarly, the attributes for the customer entity might include name, address, age, and loyalty status.

Identify the Relationships:

Once we have defined the entities and their attributes, we need to identify the relationships between them. In this case, there are several relationships between the entities:

A customer can make multiple purchases, and a purchase can have multiple products. This is a many-to-many relationship between the customer and product entities, which can be represented by a junction table.

A product can be sold in multiple regions, and a region can have multiple products. This is a many-to-many relationship between the product and region entities, which can be represented by a function table.

A purchase Is made at a specific time, and a time period can have multiple purchases. This is a one-to-many relationship between the time period and purchase entities.

A product can have multiple inventory entries, and an inventory entry is associated with a single

Product. This is a one-to-many relationship between the product and inventory entities.

Design the SQL Database Schema:

Based on the entities and their relationships, we can design the SQL database schema. Here is an example schema that can be used to track the required data:

Product table: This table stores information about products, including product ID, name,

Description, price, and category.

Region table: This table stores information about regions, including region ID and name.

\*\*Time period table: “This table stores information about time periods, including time period ID and name.

Customer table: This table stores information about customers, including customer ID name

Address age, and loyalty status

Purchase table: This table stores information about purchases, including purchase ID, customer

ID time period ID, and total purchase amount

Purchase product table: This table represents the many-to-many relationship between purchases and products it stores the purchase ID and product ID for each product purchased in a given purchase.

\*\*Product region table: \*\*This table represents the many-to-many relationship between products and regions. It stores the product ID and region ID for each region in which the product is sold.

Inventory table: This table stores information about inventory, including inventory ID, product ID, stock level