Design Context:

Analyze and design an e-commerce data warehouse for a nationwide chain of superstores of Bangladesh. There are many suppliers for the chain shop. The suppliers have sup-id, name, type of products to supply (clothes, machineries, food etc.), address (street, city, district). The chain shops serve the customers physically from the superstores and each superstore has its own system for all kinds of transactions (sale, procurement from the suppliers etc.). There are 10 million registered customers and each customer has customer id, name, NID, address (House no., street, thana, city, district, division and age group. A customer can purchase many items in a single transaction with transaction id, transaction type (cash or card), timestamp id, time of the day, day of the week, date, week, month, year, quantity, unit price and total price. Each item has an item id, name, type, country of manufacture. You have to design and implement a warehouse for this chain of superstores to support policy decision process and knowledge discovery. Perform the following tasks for the warehouse.

Solution:

In this context, we will design a star schema. First, we have to understand the design of our warehouse system. We have two types of tables. One type is Dimension table and the fact table according to the theory of data warehousing. Dimension tables contain the information of that particular tables. Fact table connects all the dimension tables by using a primary key. In our case, we have five dimension-tables-

1. Transaction table (Trans_dim)
2. Item table (Item_dim)
3. Store table (Store_dim)
4. Time table (Time_dim)
5. Customer table (Customer_dim)

And our fact table is –

❖ Fact table (Fact_table)

The details descriptions of each table are given below in a table which contains primary key and the attributes of each table-

Table Name	Attributes
Transaction	Payment_key(pk), transaction_type, bank_name
Item	Item_key (pk), item_name, description, unit_price, manufacturing_country, supplier, unit
Store	Store_key(pk), division, district, upazila
Time	Time_key(pk), date, hour, day, week, month, quarter, year
Customer	Coustomer_key(pk), name, contact_no, NID

Fact	Payment_key(pk&fk), Item_key (pk&fk), Store_key(pk&fk), Time_key(pk&fk), Coustomer_key(pk&fk), quantity, unit, unit_price, total_price

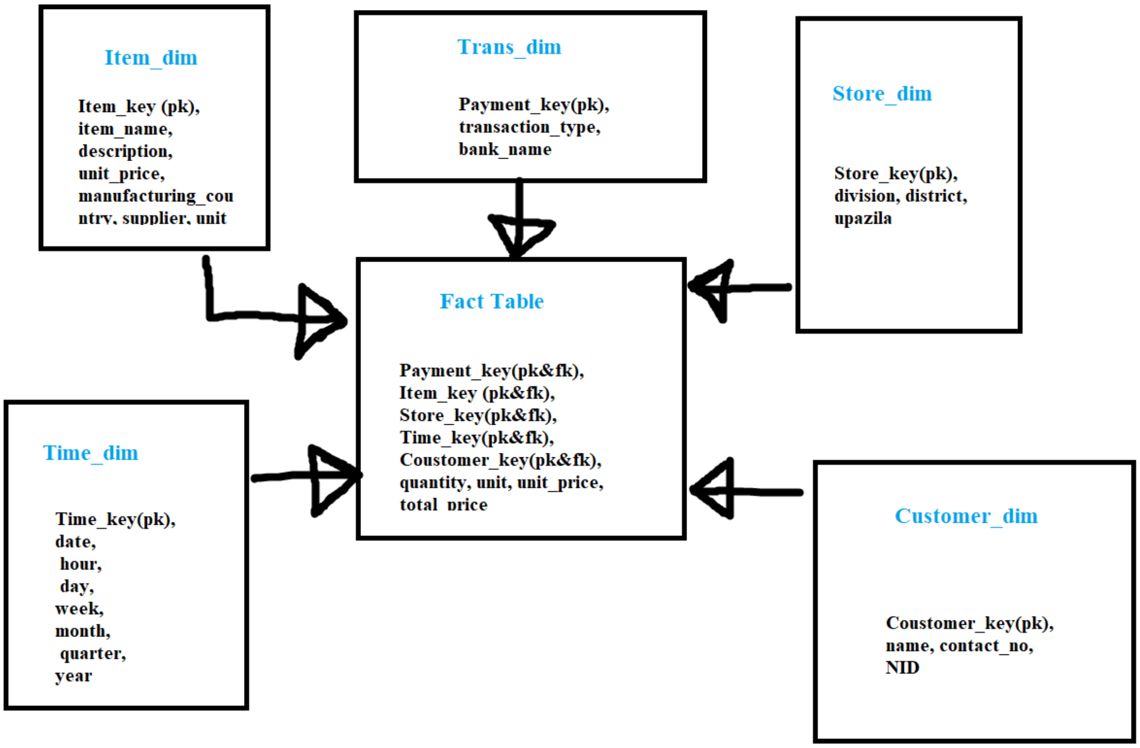

Figure1: Star schema diagram design

Programming Guide:

Required software: **PostgreSQL** and **SqlDBM** online (can be other software as well)

Step1: Go to the SqlDBM and create a design of star schema by drag and drop

Step2: Go to the forward engineering for generating the SQL code for the schema

Step3: Open PostgreSQL and create a Schema

Step4: Go to the SQL tool and then paste the Code.

It will generate a star schema.

Here is my code for this project:

Analysis:

Sources of data: In this case, data can be collected from many sources for example, if we consider the Item table, then there are a lot of items in real world. So, for each item there will be a new entity. This data can be collected by sorting out the item that we want to sell. Then we can also collect some data about which product demand is good in market. We can collect data from super-shops, grocery, online shopping site etc.

Preprocessing: Data preprocessing includes cleaning, integration, transformation, reduction, discretization, normalization and so on. Before inserting data to warehouse, it is a must that first we must do some preprocessing and then we can store our data. Unprocessed data can result in a very inefficient warehouse.

Noise Reduction: Noise reduction is one kind of data preprocessing that need to be done before storing the data. Noisy data can be meaningless to our program. Noise can be generated due to suing incorrect way for collecting data or due to wrong input of data. Noise can be removes in different ways-

- Binding Method: This is method of smoothing. The entire dataset is divided into equal size of segment. User can replace all data of a segment by it's mean value.
- Regression: We can also smooth the noisy part of our data by doing regression. For example, in the simplest case, we can use linear regression for smoothing the noisy part. We can also use Neural Network for solving this issue.
- Clustering: If we cluster our data based on similarity then all the same types of data will be in the same cluster and the noise / outlier will be in a separate place. Then it will be really easy to detect that part

Data Transformation: Data transformation is a good approach for further analysis of data. It allows us to use the data without considering the domain or constrains. Data transformation can be done by several ways-

- Normalization: Normalization is rescaling al the values of an attribute in a specific range. For example, 0 to 1 or -1 to 1
- Attribute selection: Attributes are generated from the existing set of attributes for the further processing of data
- Discretization: The raw values of an attribute is replaced by interval levels. This is applied for only numeric attributes

Uploading: Data is uploaded into multiple tables in the data warehouse. For that integrity is required. For example, here we have created a star schema for connecting all our tables. Meaning there is a connection between each and every table in our dataset. Before uploading we have to make sure that each dimension table has connection with fact table.

Collecting process of data for the superstore (source driven /destination driven)

In our scenario, we are dealing a e-commerce store. In real life scenario collecting data manually is very difficult for this type of warehouse. Eventually manually collecting data is time consuming and requires more manpower. For that reason, we have to design our pipeline in such a way so that we can collect our data automatically from any site or data store. For that we need to design an API as we can't access the database of any company or organization directly. We need to send API request to the data store for data. Now here comes a challenge. Should we deign our pipeline in source driven or destination driven way.

Source-driven architecture: In source driven architecture the data transfer in initiated by each data source.

Advantages & disadvantages:

a. Data can be transferred to warehouse once it is available in the source. But in the destination-driven model, the warehouse need to make request frequently for data to the data source which result in high overhead.
b. Once the data is sent to the destination source doesn't need to keep the history of the data. This is comfortable and required less complexity for the data source. But in the destination-driven model each history need to store for a duration of time.

Destination-driven architecture: In this architecture the data is collected according to the demand of the warehouse.

Advantages & disadvantages:

a. In source-driven architecture source need to be active until the data is transferred to destination. Source also need to handle if any error occurs during the transfer. On the other hand, destination-driven architecture, source is required to provide only a basic functionality of executing queries.
b. The data warehouse has more control on collecting the data. As the request is sent to the source by the warehouse and data source provides data according to the request.

Decision: In our case we need specific attributes of data.