**Batch -** T7

**Assignment No. –** 7

**Title -** Build the data warehouse for X-Mart

**Student Name -** Sharaneshwar Bharat Punjal

**Student PRN -** 23520011

**Problem Statement:**

X-Mart is having different malls in city, where daily sales take place for various products. Higher management is facing an issue while decision making due to non-availability of integrated data they can't do study on their data as per their requirement. So objective is to design a system which can help them quickly in decision making and provide Return on Investment (ROI).

**Activity:**

**Identify and Collect Requirements**

We need to interview the key decision makers to know, what factors define the success in the business? How does management want to analyze their data? What are the most important business questions, which need to be satisfied by this new system?

We also need to work with persons in different departments to know the data. and their common relations if any, document their entire requirement which need to be satisfied by this system.

Let us first identify the requirement from management about their requirements.

* Need to see daily, weekly, monthly, quarterly profit of each store.
* Comparison of sales and profit on various time periods.
* Comparison of sales in various time bands of the day.
* Need to know which product has more demand on which location?
* Need to study trend of sales by time period of the day over the week, month, and year?
* On what day sales is higher?
* On every Sunday of this month, what is sales and what is profit?
* What is trend of sales on weekday and weekend?
* Need to compare weekly, monthly and yearly sales to know growth and KPI

**Design the Dimensional Model**

We need to design Dimensional Model to suit requirements of users which must address business needs and contains information which can be easily accessible. Design of model should be easily extensible according to future needs. This model design must supports OLAP cubes to provide "instantaneous" query results for analysts.

Let us take a quick look at a few new terms and then we will identify/derive it for our requirement.

Dimension - The dimension is a master table composed of individual, non-overlapping data elements. The primary functions of dimensions are to provide filtering, grouping and labelling on your data. Dimension tables contain textual descriptions about the subjects of the business.

Let me give you a glimpse on different types of dimensions available like confirmed dimension, Role Playing dimension, Degenerated dimension, Junk Dimension.

Slowly changing dimension (SCD) specifies the way using which you are storing values of your dimension which is changing over a time and preserver the history. Different methods /types are available to store history of this change E.g. SCD1, SCD2, and SCD3 you can use as per your requirement.

Let us identify dimensions related to the above case study.

Product, Customer, Store, Date, Time, Sales person Measure

A measure represents a column that contains quantifiable data, usually numeric, that can be aggregated. A measure is generally mapped to a column in a fact table. For your information, various types of measures are there. E.g. Additive, semi. additive and Non additive.

Let us define what will be the Measures in our case. Actual Cost, Total Sales, Quantity, Fact table record count Fact Table

Data in fact table are called measures (or dependent attributes), Fact table provides statistics for sales broken down by customer, salesperson, product, period and store dimensions. Fact table usually contains historical transactional entries of your live system, it is mainly made up of foreign key column which references to various dimension and numeric measure values on which aggregation will be performed. Fact tables are of different types, E.g. Transactional, Cumulative and Snapshot

Let us identify what attributes should be there in our Fact Sales Table.

Foreign Key Column

Sales Date key, Sales Time key, Invoice Number, Sales Person ID, Store ID, Customer ID

Measures

Actual Cost, Total Sales, Quantity, Fact table record count

Design the Relational Database

We have done some basic workout to identify dimensions and measures, now we have to use appropriate schema to relate this dimension and Fact tables.

Few popular schemas used to develop dimensional model are as follows:

E.g. Star Schema, Snow Flake Schema, Star Flake Schema, Distributed Star

Schema, etc. In a different article, we will discuss all these schemas, dimension

types, measure types, etc., in detail.

First try to use Star schema due to hierarchical attribute model it provides for analysis and speedy performance in querying the data.

Star schema the diagram resembles a star, with points radiating from a center. The center of the star consists of fact table and the points of the star are the dimension tables.

A diagram of a product

AI-generated content may be incorrect.

**1. Introduction**

**Objective:**  
X-Mart operates multiple malls across the city where daily sales transactions occur for various products. The higher management has faced difficulties in decision making due to fragmented data sources. This project aims to develop an integrated data warehouse that quickly aggregates sales data and computes key performance indicators such as daily, weekly, monthly, and quarterly profits. The system will facilitate rapid decision making and improve Return on Investment (ROI) by enabling analysis over multiple dimensions.

**Scope:**

* **Integration:** Combine data from various departmental systems (sales, inventory, customer, and store data) into one centralized data warehouse.
* **Analysis:** Enable business users to run OLAP operations like slicing, dicing, drill-down, and roll-up to analyze sales performance across different time bands and store locations.
* **Extensibility:** The design will support future dimensions and measures, and accommodate evolving business needs.
* **Performance:** Utilize a star schema for high-performance querying and straightforward aggregation of measures.

**2. Business Requirements Analysis**

**Requirement Gathering**

* **Interviews and Workshops:**
  + **Key Decision Makers:** Interviewed to determine success factors such as profit margins, sales trends, and product demand by location.
  + **Departmental Collaboration:** Coordinated with sales, inventory, and marketing teams to understand data relationships and existing reporting pain points.

**Identified Business Questions**

* What are the daily, weekly, monthly, and quarterly profits of each store?
* How do sales and profit figures compare over various time periods?
* Which time bands during the day yield the highest sales?
* Which products are in high demand in different locations?
* What are the trends in sales over the week, month, and year by time of day?
* Which day records the highest overall sales?
* What are the sales and profits on every Sunday of a given month?
* How do weekday and weekend sales trends compare?
* How can weekly, monthly, and yearly sales be compared to determine growth and key performance indicators (KPIs)?

**Summary of Requirements**

* **Dimensions:**
  + **Product:** Describes product details.
  + **Customer:** Captures customer demographics and behavior.
  + **Store:** Information on mall/store location and contact details.
  + **Date:** Calendar-based dimension for day, week, month, quarter, and year analysis.
  + **Time:** Time bands (morning, afternoon, evening, etc.) to analyze sales during specific periods of the day.
  + **Salesperson:** (if applicable) to monitor performance and commissions.
* **Measures:**
  + **Actual Cost:** The cost incurred for the products sold.
  + **Total Sales:** The revenue from sales transactions.
  + **Quantity:** Number of units sold.
  + **Record Count:** Count of transactions (useful for transaction volume analysis).

**3. Dimensional Model Design**

**Identifying Dimensions and Measures**

* **Dimensions:**
  + **Dim\_Product:** Product\_ID, Product\_Name, Category, Description, and attributes such as Size, Color, etc.
  + **Dim\_Customer:** Customer\_ID, Name, Demographics, and Loyalty information.  
    *Considerations for Slowly Changing Dimensions (SCD) can be applied if customer information changes over time (e.g., SCD Type 2).*
  + **Dim\_Store:** Store\_ID, Store\_Name, Location (Address, City, State), and Contact information.
  + **Dim\_Date:** Date\_Key, Date, Day, Month, Quarter, Year, Day of Week, etc.
  + **Dim\_Time:** Time\_Key, Hour, Minute, Time Band (morning, afternoon, evening), etc.
  + **Dim\_Salesperson:** Salesperson\_ID, Name, Territory, and other relevant attributes.
* **Fact Table – Fact\_Sales:**
  + **Foreign Keys:**
    - Date\_Key (from Dim\_Date)
    - Time\_Key (from Dim\_Time)
    - Invoice\_Number (could serve as a unique transaction identifier)
    - Salesperson\_ID (from Dim\_Salesperson)
    - Store\_ID (from Dim\_Store)
    - Customer\_ID (from Dim\_Customer)
    - Product\_ID (from Dim\_Product)
  + **Measures:**
    - Actual\_Cost
    - Total\_Sales
    - Quantity
    - Record\_Count (could be derived as 1 per transaction entry)

**Designing the Star Schema**

The star schema places the fact table at the center with surrounding dimension tables. This layout supports fast aggregation and simple, intuitive queries for OLAP operations.

* **Benefits:**
  + Simplifies query writing and improves performance.
  + Supports rapid roll-up/drill-down operations for time-based analysis and store comparisons.

**4. Data Warehouse Implementation**

**ETL Process**

* **Extract:** Retrieve data from source systems such as POS systems, inventory databases, and customer relationship management (CRM) systems.
* **Transform:**
  + Clean and standardize data formats (e.g., date and time formats).
  + Apply business rules (e.g., classification of time bands or customer segments).
  + Manage slowly changing dimensions to preserve history (e.g., using SCD Type 2 for customer changes).
* **Load:** Insert the transformed data into the fact and dimension tables in the warehouse using MySQL scripts or ETL tools.

**Example SQL Code Snippets**

**Creating the Fact Table:**

CREATE TABLE FactProductSales (

TransactionID INT AUTO\_INCREMENT PRIMARY KEY,

SalesInvoiceNumber VARCHAR(50),

SalesDateKey INT,

SalesTimeKey INT,

StoreID INT,

CustomerID INT,

ProductKey INT,

SalesPersonID INT,

Quantity INT,

TotalAmount DECIMAL(10,2),

DateKey INT,

TimeKey INT,

FOREIGN KEY (SalesDateKey) REFERENCES DimDate(DateKey),

FOREIGN KEY (SalesTimeKey) REFERENCES DimTime(TimeKey),

FOREIGN KEY (StoreID) REFERENCES DimStores(StoreID),

FOREIGN KEY (CustomerID) REFERENCES DimCustomer(CustomerID),

FOREIGN KEY (ProductKey) REFERENCES DimProduct(ProductKey),

FOREIGN KEY (SalesPersonID) REFERENCES DimSalesPerson(SalesPersonID)

);

**Creating Dimension Tables:**

CREATE TABLE DimTime (

TimeKey INT PRIMARY KEY,

TimeAltKey INT,

Time30 INT,

Hour30 INT,

MinuteNumber INT,

SecondNumber INT,

TimeInSecond INT,

HourlyBucket VARCHAR(50),

DayTimeBucketGroupKey VARCHAR(50),

DayTimeBucket VARCHAR(50)

);

CREATE TABLE DimDate (

DateKey INT PRIMARY KEY,

FullDate DATE,

Day INT,

Month INT,

Quarter INT,

Year INT,

DayName VARCHAR(10)

);

CREATE TABLE DimSalesPerson (

SalesPersonID INT PRIMARY KEY,

SalesPersonAltID VARCHAR(50),

SalesPersonName VARCHAR(255),

StoreID INT,

City VARCHAR(100),

State VARCHAR(100),

Country VARCHAR(100),

FOREIGN KEY (StoreID) REFERENCES DimStores(StoreID)

);

CREATE TABLE DimStores (

StoreID INT PRIMARY KEY,

StoreAltID VARCHAR(50),

StoreName VARCHAR(255),

StoreLocation VARCHAR(255),

City VARCHAR(100),

State VARCHAR(100),

Country VARCHAR(100)

);

CREATE TABLE DimCustomer (

CustomerID INT PRIMARY KEY,

CustomerAltID VARCHAR(50),

CustomerName VARCHAR(255),

Gender VARCHAR(10)

);

CREATE TABLE DimProduct (

ProductKey INT PRIMARY KEY,

ProductAltKey VARCHAR(50),

ProductName VARCHAR(255),

ProductCost DECIMAL(10,2)

);

**5. Data Cube Implementation and OLAP Operations**

**Data Cube Implementation**

**Aggregation Example:**

Create a materialized view or summary table to pre-aggregate sales by store and date to speed up report generation.

CREATE TABLE Sales\_Cube AS

SELECT s.Store\_ID, d.Year, d.Month,

SUM(f.Total\_Sales) AS Total\_Sales,

SUM(f.Actual\_Cost) AS Total\_Cost,

SUM(f.Quantity) AS Total\_Quantity

FROM Fact\_Sales f

JOIN Dim\_Store s ON f.Store\_ID = s.Store\_ID

JOIN Dim\_Date d ON f.Date\_Key = d.Date\_Key

GROUP BY s.Store\_ID, d.Year, d.Month;

**OLAP Operations**

* **Roll-Up:**  
  Aggregate daily sales to weekly or monthly sales.

SELECT d.Year, d.Month, SUM(f.Total\_Sales) AS Monthly\_Sales

FROM Fact\_Sales f

JOIN Dim\_Date d ON f.Date\_Key = d.Date\_Key

GROUP BY d.Year, d.Month;

* **Drill-Down:**  
  Drill down from monthly sales to daily sales.

SELECT d.Full\_Date, f.Total\_Sales

FROM Fact\_Sales f

JOIN Dim\_Date d ON f.Date\_Key = d.Date\_Key

WHERE d.Year = 2025 AND d.Month = 2;

* **Slice:**  
  Filter data to view sales for a specific store or product.

SELECT \*

FROM Sales\_Cube

WHERE Store\_ID = 101;

* **Dice:**  
  Analyze data across two or more dimensions (e.g., sales on weekends for a specific product).

SELECT d.Day\_Name, SUM(f.Total\_Sales) AS Sales

FROM Fact\_Sales f

JOIN Dim\_Date d ON f.Date\_Key = d.Date\_Key

JOIN Dim\_Product p ON f.Product\_ID = p.Product\_ID

WHERE d.Day\_Name IN ('Saturday', 'Sunday') AND p.Product\_Name = 'ProductX'

GROUP BY d.Day\_Name;

**6. Observations**

**a. OLAP Report Generation**

* **Interactive Panels:**  
  End-users can generate reports by selecting various dimensions such as store, date, or product category. Dashboards can be created for quick visual analysis (e.g., profit trends, peak sales hours).
* **Real-Time Analysis:**  
  Pre-aggregated data cubes help management get instantaneous query results, aiding in rapid decision making.

**b. Data Verification**

* **Cross-Checking:**  
  OLAP reports are regularly verified against the source systems. Reconciliation queries compare aggregated totals from the data warehouse with those computed from the operational databases.
* **Automation:**  
  Scheduled ETL jobs and trigger-based updates ensure that the data warehouse remains synchronized with real-time data from the sales channels.

**7. Conclusion**

This project presents a complete design for an integrated data warehousing solution tailored for X-Mart’s multi-mall sales environment. By identifying critical business requirements and applying dimensional modelling with a star schema, the solution enables rapid, multi-dimensional analysis across various time frames and locations. The use of OLAP operations and data cubes significantly improves query performance, allowing management to quickly assess profitability, sales trends, and product demand. Overall, this approach not only enhances decision making but also drives the company toward improved ROI and strategic growth.