EXPERIMENT. NO: 02

Aim: One case study on building Data warehouse/Data Mart.

Theory:

Difference between the two terms – DBMS & Data Warehouse

A Database Management System (DBMS) stores data in the form of tables, uses ER model and the goal is ACID properties. For example, a DBMS of college has tables for students, faculty, etc.

A Data Warehouse is separate from DBMS, it stores a huge amount of data, which is typically collected from multiple heterogeneous sources like files, DBMS, etc. The goal is to produce statistical results that may help in decision makings. For example, a college might want to see quick different results, like how the placement of CS students has improved over the last 10 years, in terms of salaries, counts, etc.

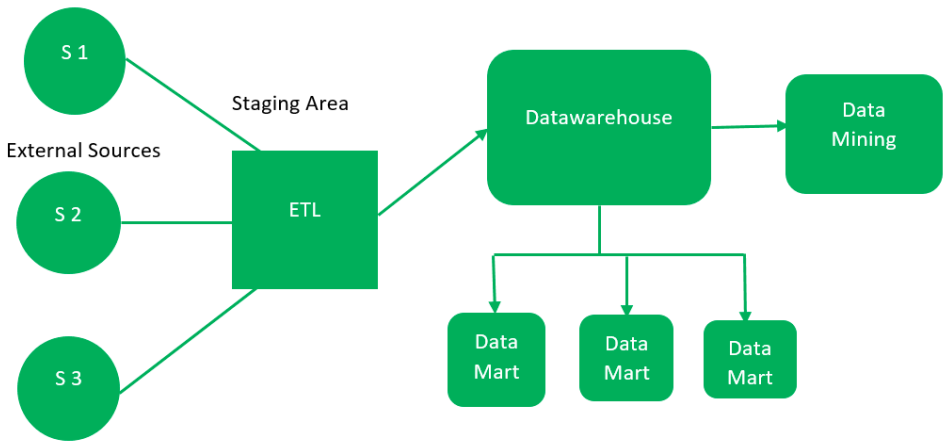
Benefits of Data Warehouse

1. Better business analytics: Data warehouse plays an important role in every business to store and analysis of all the past data and records of the company. which can further increase the understanding or analysis of data to the company.
2. Faster Queries: Data warehouse is designed to handle large queries that's why it runs queries faster than the database.
3. Improved data Quality: In the data warehouse the data you gathered from different sources is being stored and analyzed it does not interfere with or add data by itself so your quality of data is maintained and if you get any issue regarding data quality then the data warehouse team will solve this.
4. Historical Insight: The warehouse stores all your historical data which contains details about the business so that one can analyze it at any time and extract insights from it

Construction of a Data Warehouse

There are 2 approaches for constructing data-warehouse: Top-down approach and Bottom-up approach are explained as below.

A. Top-Down Approach:



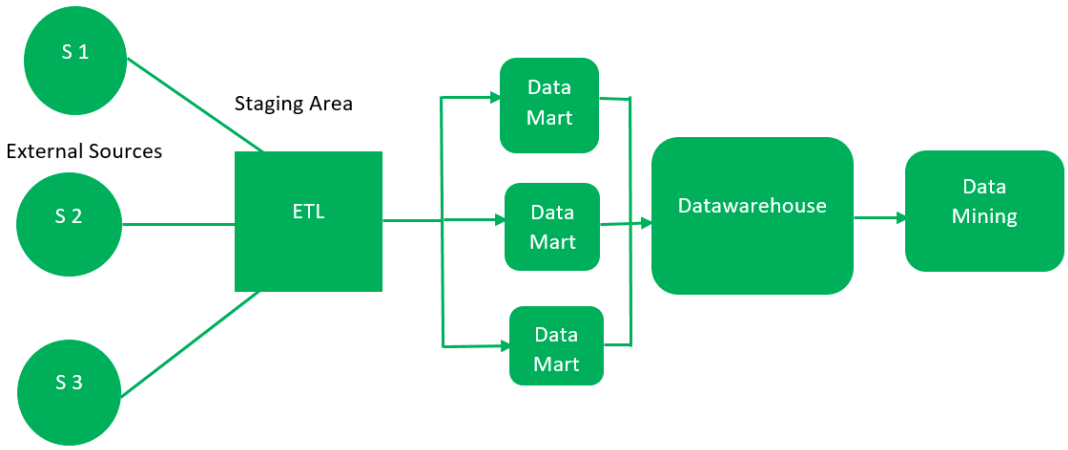
The essential components are discussed below:

1. External Sources –
   External source is a source from where data is collected irrespective of the type of data. Data can be structured, semi structured and unstructured as well.
2. Stage Area –
   Since the data, extracted from the external sources does not follow a particular format, so there is a need to validate this data to load into data warehouse. For this purpose, it is recommended to use ETL tool.
3. E(Extracted): Data is extracted from External data source.
4. T(Transform): Data is transformed into the standard format.
5. L(Load): Data is loaded into data warehouse after transforming it into the standard format.
6. Data-warehouse –
   After cleansing of data, it is stored in the data warehouse as central repository. It actually stores the meta data and the actual data gets stored in the data marts. Note that data warehouse stores the data in its purest form in this top-down approach.
7. Data Marts –
   Data mart is also a part of storage component. It stores the information of a particular function of an organisation which is handled by single authority. There can be as many number of data marts in an organisation

depending upon the functions. We can also say that data mart contains subset of the data stored in data warehouse.

8. Data Mining –
   The practice of analysing the big data present in data warehouse is data mining. It is used to find the hidden patterns that are present in the database or in data warehouse with the help of algorithm of data mining. This approach is defined by Inmon as – data warehouse as a central repository for the complete organisation and data marts are created from it after the complete data warehouse has been created.

B. Bottom-Up Approach:



1. First, the data is extracted from external sources (same as happens in top-down approach).
2. Then, the data go through the staging area (as explained above) and loaded into data marts instead of data warehouse. The data marts are created first and provide reporting capability. It addresses a single business area.
3. These data marts are then integrated into data warehouse.

This approach is given by Kinball as – data marts are created first and provides a thin view for analyses and data warehouse is created after complete data marts have been created.

Fact Table

- In a data warehouse, a fact table is a table that stores the measurements, metrics, or facts related to a business operation.
- It is located at the centre of a star or snowflake schema and is surrounded by dimension tables.
- When multiple fact tables are used, they can be organized using a "fact constellation schema."
- A fact table has two types of columns: those that contain the facts and those that serve as foreign keys linking to dimension tables.
- The primary key of a fact table is often a composite key made up of all of the foreign keys in the table.
- Fact tables can hold various types of measurements, such as additive, non-additive, and partly additive measures, and store important information in the data warehouse.
- They are useful for evaluating dimensional attributes because they provide additive values that can act as independent variables.

Dimension Table

- Dimension tables contain descriptions of the objects in a fact table and provide information about dimensions such as values, characteristics, and keys.
- These tables are usually small, with a number of rows ranging from a few hundred to a few thousand.
- The term "dimension table" refers to a set of data related to any quantifiable event and is the foundation for dimensional modelling.
- Dimension tables have a column that serves as a primary key, allowing each dimension row or record to be uniquely identified. This key is used to link the dimension table to the fact tables. A surrogate key, which is a system-generated key, is often used to uniquely identify the rows in the dimension table.

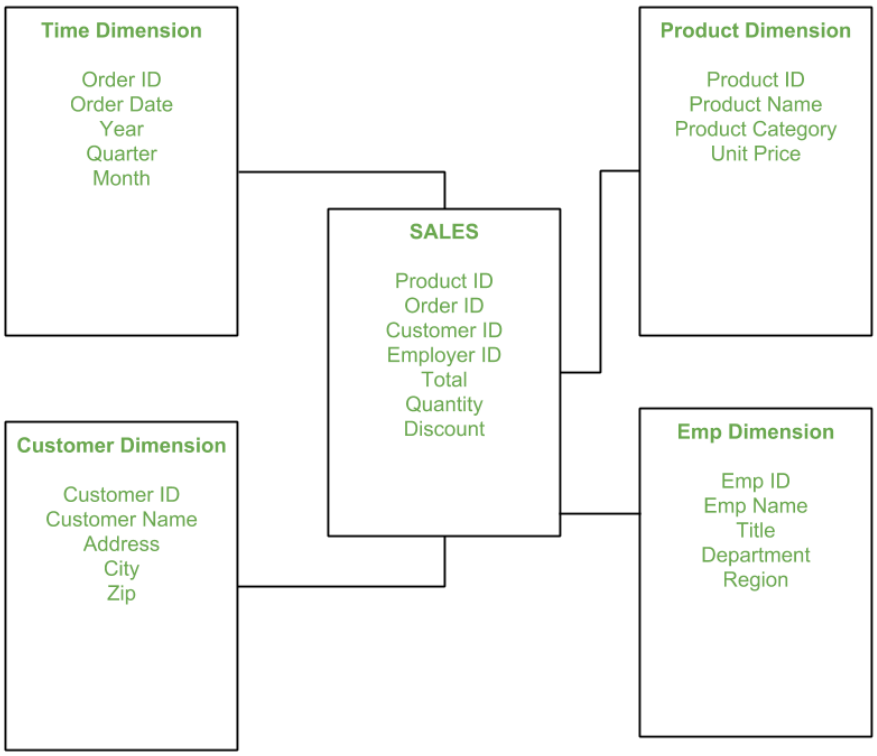
Star Schema

A star schema is a type of data modelling technique used in data warehousing to represent data in a structured and intuitive way. In a star schema, data is organized into a central fact table that contains the measures of interest, surrounded by dimension tables that describe the attributes of the measures.

The fact table in a star schema contains the measures or metrics that are of interest to the user or organization. For example, in a sales data warehouse, the fact table might contain sales revenue, units sold, and profit margins. Each record in the fact table represents a specific event or transaction, such as a sale or order.

The dimension tables in a star schema contain the descriptive attributes of the measures in the fact table. These attributes are used to slice and dice the data in the fact table, allowing users to analyse the data from different perspectives. For example, in a sales data warehouse, the dimension tables might include product, customer, time, and location.

In a star schema, each dimension table is joined to the fact table through a foreign key relationship. This allows users to query the data in the fact table using attributes from the dimension tables.



Advantages of Star Schema

1. Simpler Queries – Join logic of star schema is quite cinch in comparison to other join logic which are needed to fetch data from a transactional schema that is highly normalized.
2. Simplified Business Reporting Logic – In comparison to a transactional schema that is highly normalized, the star schema makes simpler common business reporting logic, such as of reporting and period-over-period.
3. Feeding Cubes – Star schema is widely used by all OLAP systems to design OLAP cubes efficiently. In fact, major OLAP systems deliver a ROLAP mode of operation which can use a star schema as a source without designing a cube structure.

Disadvantages of Star Schema

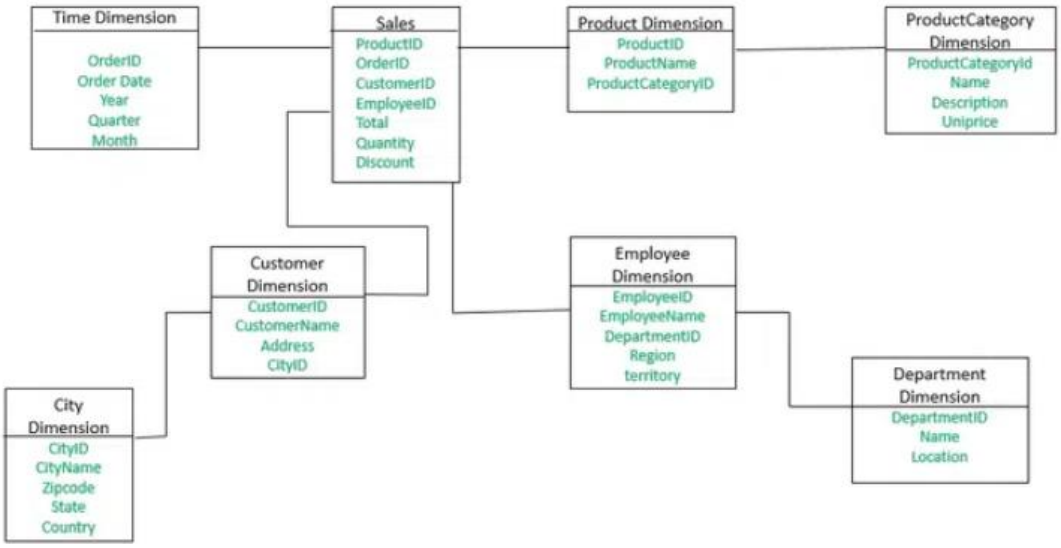
1. Data integrity is not enforced well since in a highly de-normalized schema state.
2. Not flexible in terms if analytical needs as a normalized data model.
3. Star schemas don't reinforce many-to-many relationships within business entities – at least not frequently.

Snowflake Schema

The snowflake schema is a variant of the star schema. Here, the centralized fact table is connected to multiple dimensions. In the snowflake schema, dimensions are present in a normalized form in multiple related tables. The snowflake structure materialized when the dimensions of a star schema are detailed and highly structured, having several levels of relationship, and the child tables have multiple parent tables. The snowflake effect affects only the dimension tables and does not affect the fact tables.

In a snowflake schema, the dimension tables are normalized into multiple related tables, creating a hierarchical or “snowflake” structure.

In a snowflake schema, the fact table is still located at the centre of the schema, surrounded by the dimension tables. However, each dimension table is further broken down into multiple related tables, creating a hierarchical structure that resembles a snowflake.



The Employee dimension table now contains the attributes: EmployeeID, EmployeeName, DepartmentID, Region, and Territory. The DepartmentID attribute links with the Employee table with the Department dimension table. The Department dimension is used to provide detail about each department, such as the Name and Location of the department.

The Customer dimension table now contains the attributes: CustomerID, CustomerName, Address, and CityID. The CityID attributes link the Customer dimension table with the City dimension table. The City dimension table has details about each city such as city name, Zipcode, State, and Country.

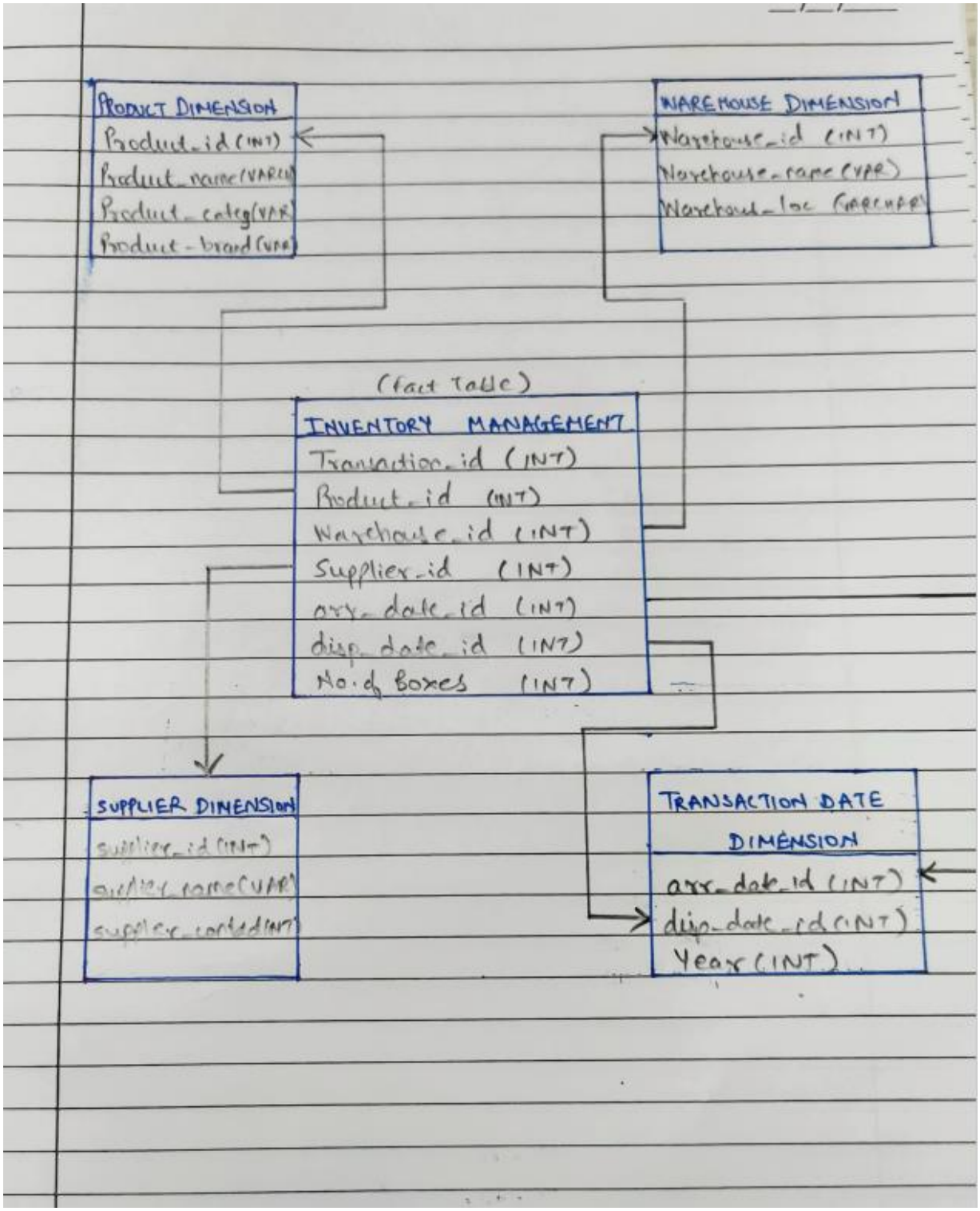
Advantages of Snowflake Schema

1. It provides structured data which reduces the problem of data integrity.
2. It uses small disk space because data are highly structured.

Disadvantages of Snowflake Schema

1. Snowflaking reduces space consumed by dimension tables but compared with the entire data warehouse the saving is usually insignificant.
2. Avoid snowflaking or normalization of a dimension table, unless required and appropriate.
3. Do not snowflake hierarchies of dimension table into separate tables. Hierarchies should belong to the dimension table only and should never be snowflakes.
4. Multiple hierarchies that can belong to the same dimension have been designed at the lowest possible detail.

Star Schema


PRODUCT DIMENSION
Product_id (INT)
Product_name (VARCHAR)
Product_categ (VARCHAR)
Product_brand (VARCHAR)
WAREHOUSE DIMENSION
Warehouse_id (INT)
Warehouse_name (VARCHAR)
Warehouse_loc (VARCHAR)
(Fact Table)
INVENTORY MANAGEMENT
Transaction_id (INT)
Product_id (INT)
Warehouse_id (INT)
Supplier_id (INT)
arr_date_id (INT)
disp_date_id (INT)
No. of Boxes (INT)
SUPPLIER DIMENSION
Supplier_id (INT)
Supplier_name (VARCHAR)
Supplier_contact (INT)
TRANSACTION DATE DIMENSION
arr_date_id (INT)
disp_date_id (INT)
Year (INT)

Snowflake Schema