Experiment No.1
Design Star Schema for a problem statement
Name: Sumit Metkari-32
Class: TE/2
Date of Performance: 07-07-25
Date of Submission: 23-07-25

Aim: To implement Star Schema /Snowflake Schema

Objective: Develop a program to implement Star Schema /Snowflake Schema

Theory:-

The entity-relationship data model is commonly used in the design of relational databases, where a database schema consists of a set of entities and the relationships between them. Such a data model is appropriate for on-line transaction processing. A data warehouse, however, requires a concise, subject-oriented schema that facilitates on-line data analysis. The most popular data model for a data warehouse is a multidimensional model. Such a model can exist in the form of a star schema, a snowflake schema, or a fact constellation schema.

A star schema for AllElectronicssales is shown in Figure 1. Sales are considered along four dimensions, namely,time, item, branch, and location. The schema contains a central fact table for sales that contains keys to each of the four dimensions, along with two measures: dollars sold and units sold. To minimize the size of the fact table, dimension identifiers (such as time key and item key) are system-generated identifiers. Notice that in the star schema, each dimension is represented by only one table, and each table contains a set of attributes. For example, the location dimension table contains the attribute set {location key, street, city, province or state, country}. This constraint may introduce some redundancy. For example, "Vancouver" and "Victoria" are both cities in the Canadian province of British Columbia. Entries for such cities in the location dimension table will create redundancy among the attributes province or state and country, that is, (..., Vancouver, British Columbia, Canada) and (..., Victoria, British Columbia, Canada). Moreover, the attributes within a dimension table may form either a hierarchy (total order) or a lattice (partial order).
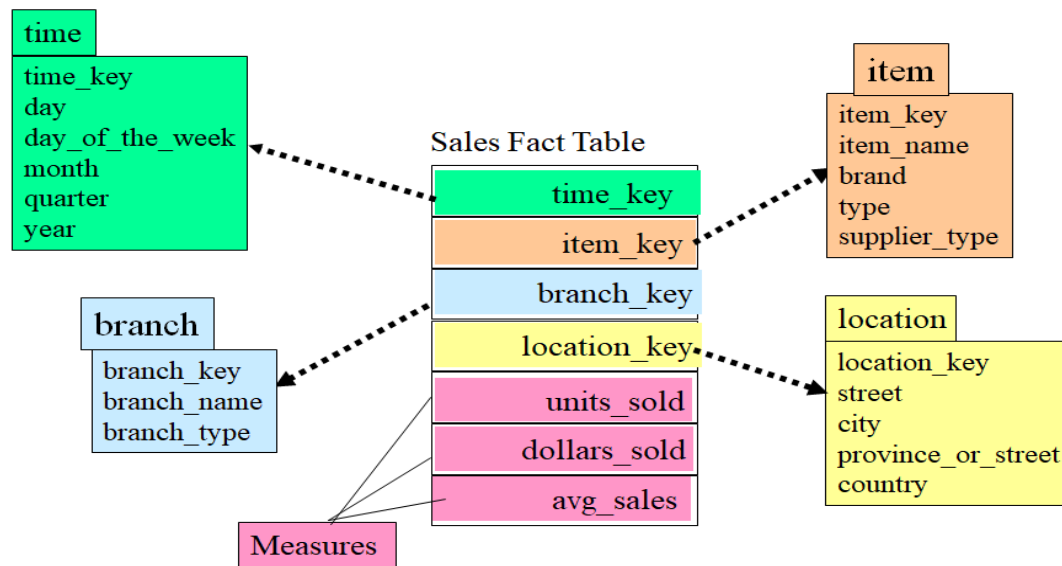
Figure 1:Star Schema

Snowflake Schema Architecture: The snowflake schema is a variant of the star schema model, where some dimension tables are normalized, thereby further splitting the data into additional tables. The resulting schema graph forms a shape similar to a snowflake.

A snowflake schema for AllElectronics sales is given in Figure 2. Here, the sales fact table is identical to that of the star schema in Figure 1. The main difference between the two schemas is in the definition of dimension tables. The single dimension table for item in the star schema is normalized in the snowflake schema, resulting in new item and supplier tables. For example, the item dimension table now contains the attributes item key, item name, brand, type, and supplier key, where supplier key is linked to the supplier dimension table, containing supplier key and supplier type information. Similarly, the single dimension table for location in the star schema can be normalized into two new tables: location and city. The city key in the new location table links to the city dimension. Notice that further normalization can be performed on province or state and country in the snowflake schema shown in Figure2, when desirable
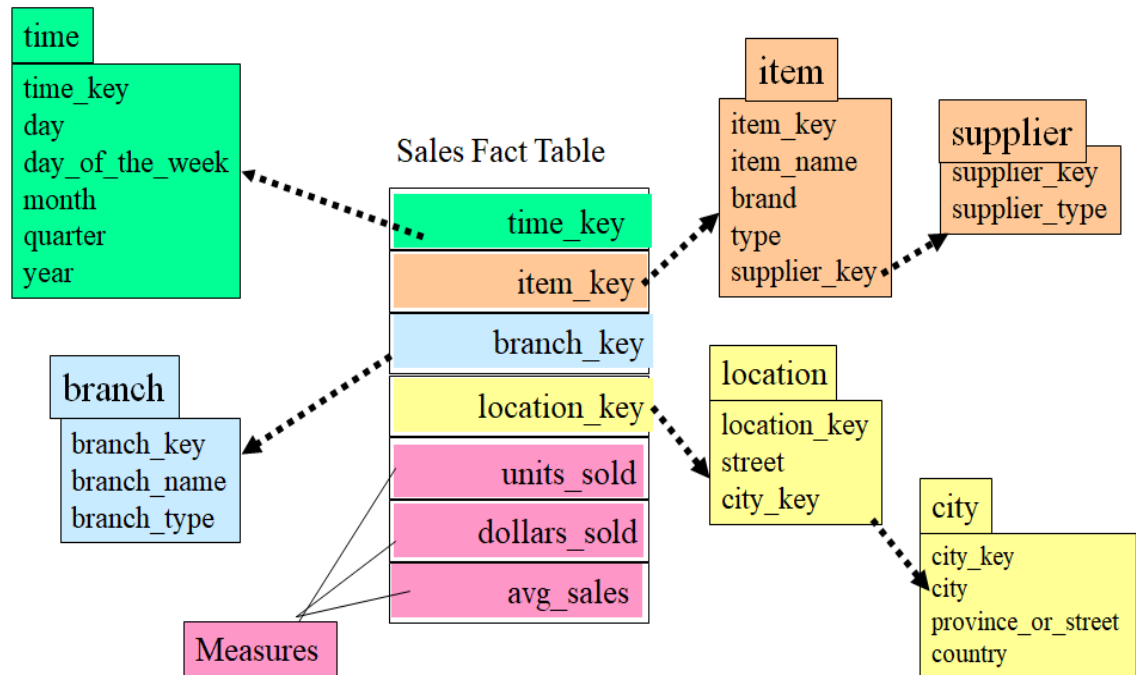
Figure 2:Snowflake Schema
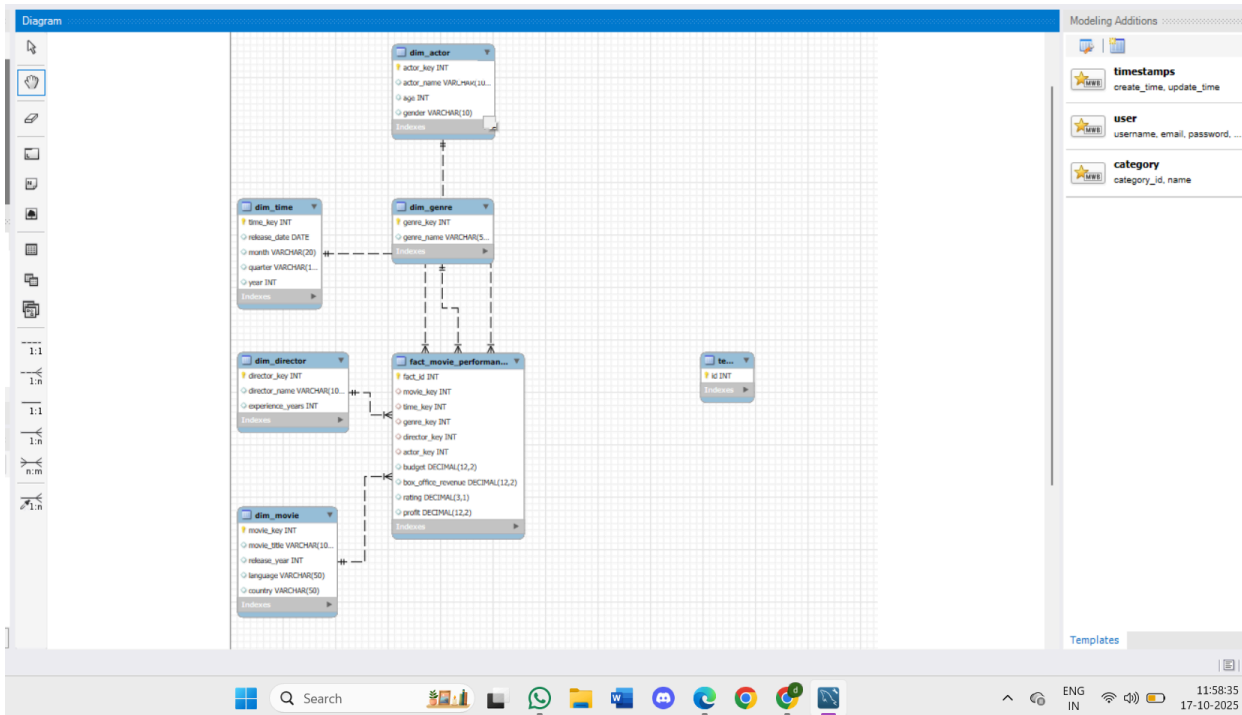
Problem statement:

The movie industry generates large volumes of data on movies, actors, directors, genres, and box office performance. Analyzing this data efficiently requires a structured approach to consolidate measures and dimensions. A star schema can organize this data into a central fact table with surrounding dimension tables. This schema enables quick retrieval of insights such as profit by genre, ratings by director, and yearly box office trends.

Star Schema:

dim_actor
- 🔑 actor_key INT
- ◇ actor_name VARCHAR(10...
- ◇ age INT
- ◇ gender VARCHAR(10)
- Indexes

dim_time
- 🔑 time_key INT
- ◇ release_date DATE
- ◇ month VARCHAR(20)
- ◇ quarter VARCHAR(1...
- ◇ year INT
- Indexes

dim_genre
- 🔑 genre_key INT
- ◇ genre_name VARCHAR(5...
- Indexes

dim_director
- 🔑 director_key INT
- ◇ director_name VARCHAR(10...
- ◇ experience_years INT
- Indexes

fact_movie_performan...
- 🔑 fact_id INT
- ◇ movie_key INT
- ◇ time_key INT
- ◇ genre_key INT
- ◇ director_key INT
- ◇ actor_key INT
- ◇ budget DECIMAL(12,2)
- ◇ box_office_revenue DECIMAL(12,2)
- ◇ rating DECIMAL(3,1)
- ◇ profit DECIMAL(12,2)
- Indexes

te...
- 🔑 id INT
- Indexes

dim_movie
- 🔑 movie_key INT
- ◇ movie_title VARCHAR(10...
- ◇ release_year INT
- ◇ language VARCHAR(50)
- ◇ country VARCHAR(50)
- Indexes
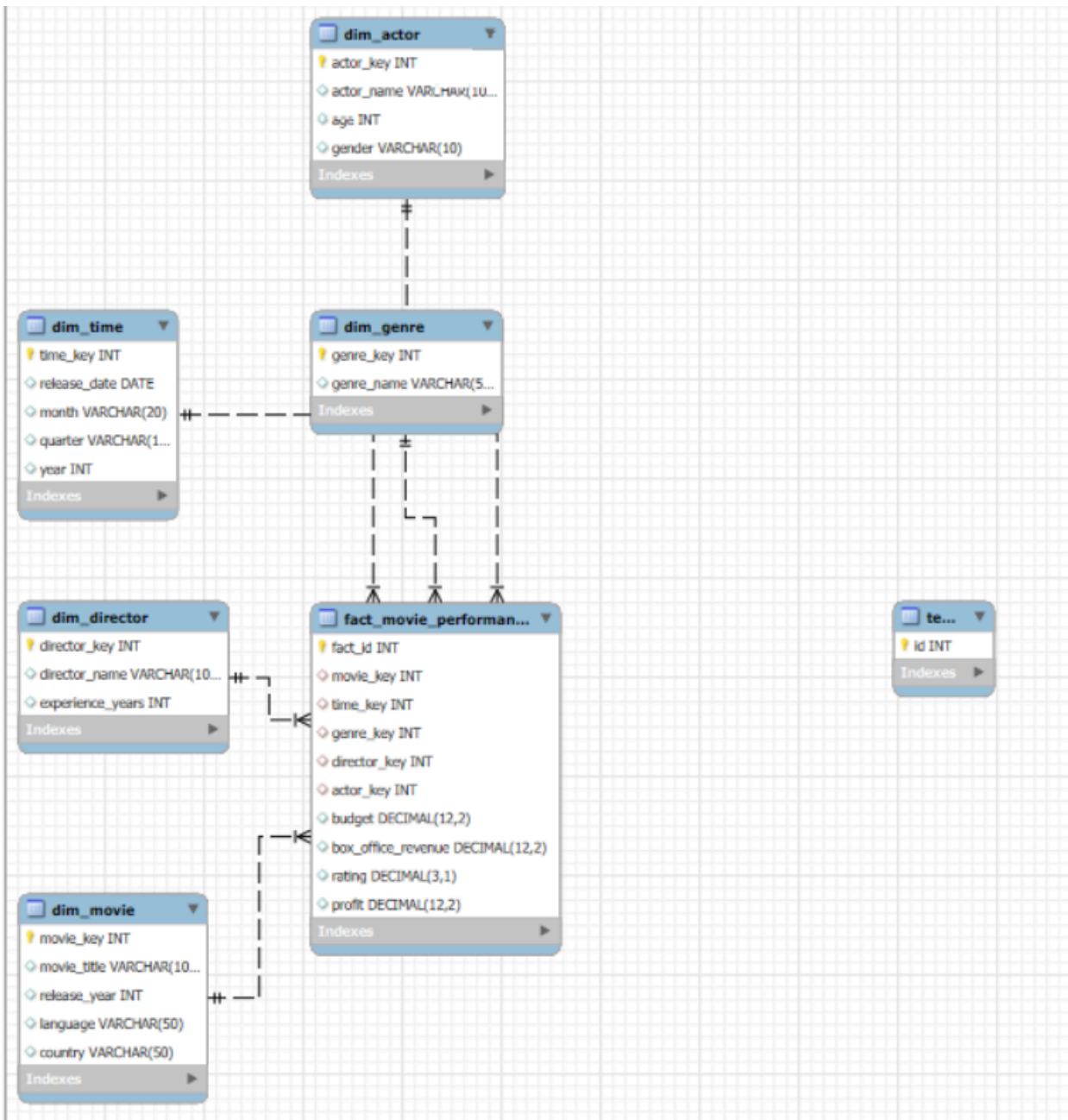
Conclusion: Comment on how star schema depicts the relationship between dimension and fact tables and its use in creating the tables.

The star schema organizes data so that a central fact table holds measurable metrics, while surrounding dimension tables store descriptive attributes linked through primary–foreign

key relationships. This structure simplifies query design and improves performance when creating and analyzing tables in data warehouses.