Name : Suvidhi V. Pareek

Class / Div : TE / 03

Roll No : 09

| Experiment No.1 |
| --- |
| Design Star Schema for a problem statement |
| Date of Performance: 11-07-2024 |
| Date of Submission: 15-07-2024 |

**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).

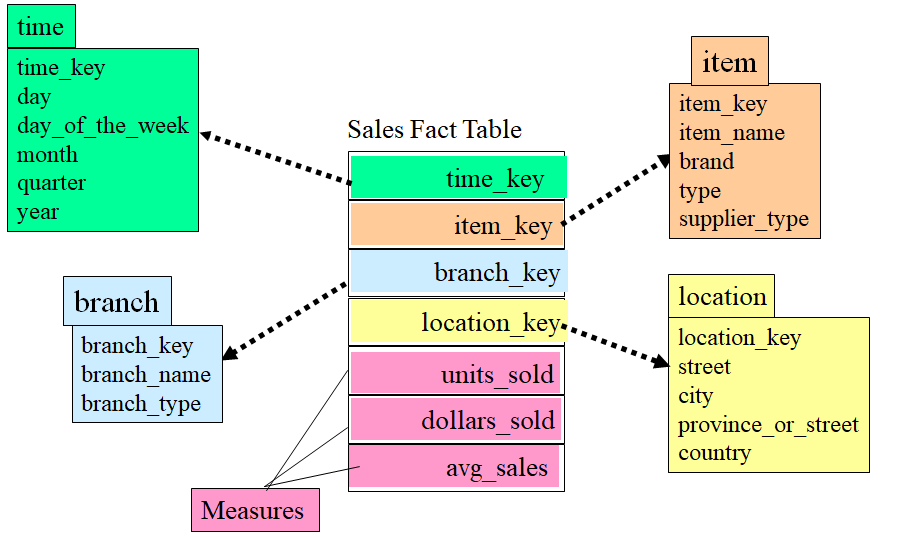


Figure 1:Star Schema

**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

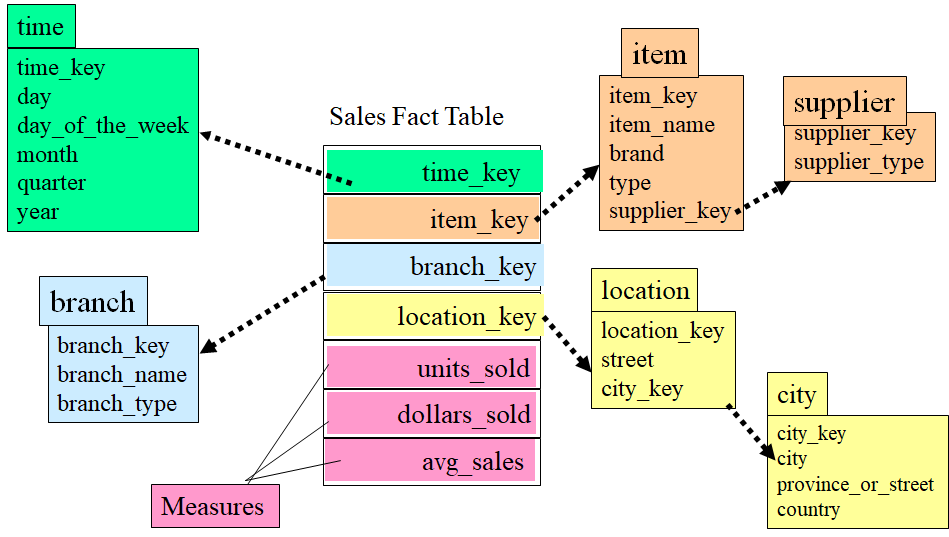


Figure 2:Snowflake Schema

**Code and output:**

-- Create Dimension Tables

CREATE TABLE time (

time\_key INT PRIMARY KEY,

day VARCHAR(45),

day\_of\_the\_week VARCHAR(45),

month VARCHAR(45),

quarter VARCHAR(45),

year VARCHAR(45)

);

CREATE TABLE branch (

Branch\_key INT PRIMARY KEY,

branch\_type VARCHAR(45),

branch VARCHAR(45)

);

CREATE TABLE item (

item\_key INT PRIMARY KEY,

item\_name VARCHAR(45),

brand VARCHAR(45),

type VARCHAR(45),

supplier\_type VARCHAR(45)

);

CREATE TABLE location (

location\_key INT PRIMARY KEY,

street VARCHAR(45),

city VARCHAR(45),

province\_or\_street VARCHAR(45),

country VARCHAR(45)

);

-- Create Measures Table

CREATE TABLE measures (

idMeasures INT PRIMARY KEY

);

-- Create Fact Table

CREATE TABLE Sales Fact Table (

time\_key INT,

item\_key VARCHAR(45),

branch\_key VARCHAR(45),

location\_key VARCHAR(45),

units\_sold VARCHAR(45),

dollars\_sold VARCHAR(45),

avg\_sales VARCHAR(45),

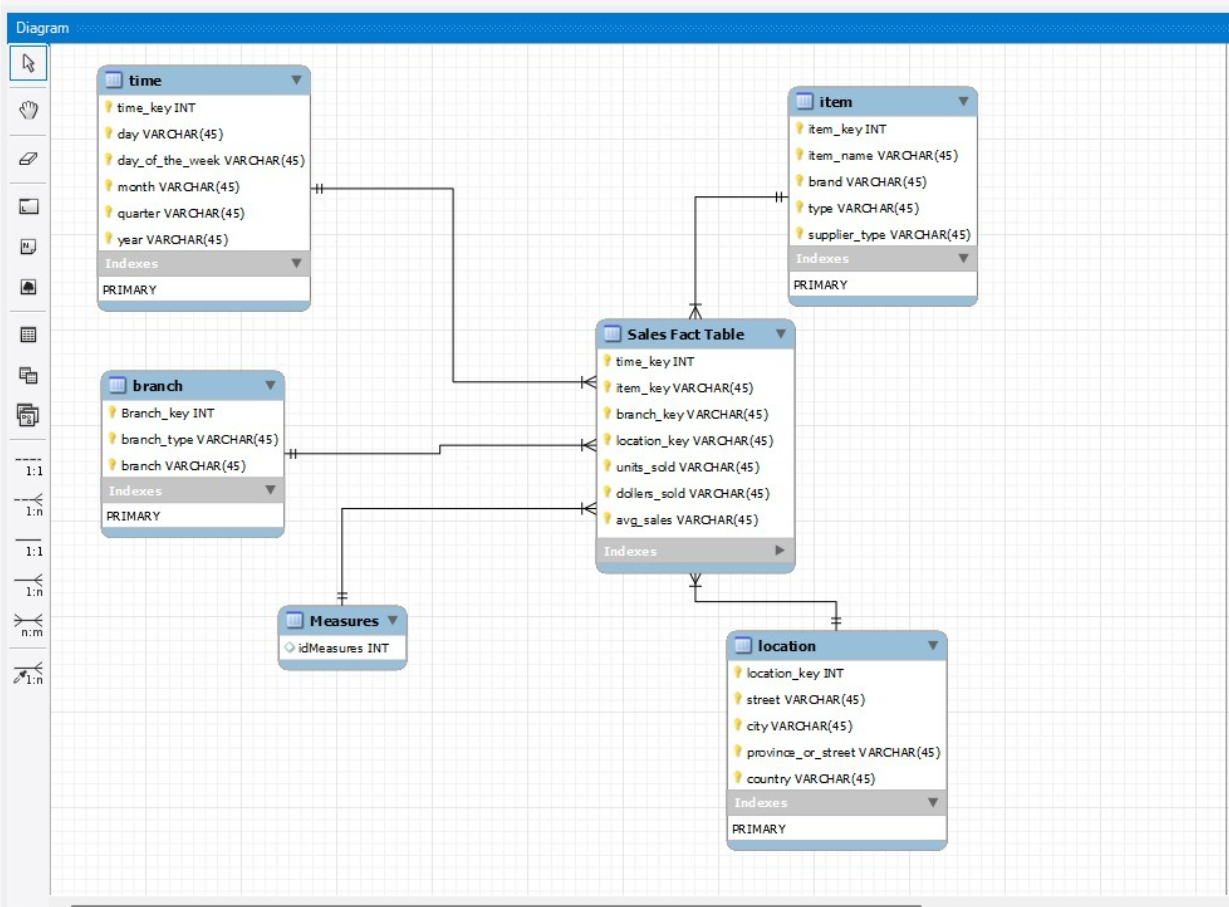
FOREIGN KEY (time\_key) REFERENCES time(time\_key),

FOREIGN KEY (item\_key) REFERENCES item(item\_key),

FOREIGN KEY (branch\_key) REFERENCES branch(branch\_key),

FOREIGN KEY (location\_key) REFERENCES location(location\_key)

);

****

-- Create Dimension Tables

CREATE TABLE time (

time\_key INT PRIMARY KEY,

day VARCHAR(45),

day\_of\_the\_week VARCHAR(45),

month VARCHAR(45),

quarter VARCHAR(45),

year VARCHAR(45)

);

CREATE TABLE branch (

branch\_key INT PRIMARY KEY,

branch\_type VARCHAR(45),

branch VARCHAR(45)

);

CREATE TABLE city (

city\_key INT PRIMARY KEY,

city VARCHAR(45),

province\_or\_street VARCHAR(45),

country VARCHAR(45)

);

CREATE TABLE location (

location\_key INT PRIMARY KEY,

street VARCHAR(45),

city\_key INT,

FOREIGN KEY (city\_key) REFERENCES city(city\_key)

);

CREATE TABLE supplier (

supplier\_key INT PRIMARY KEY,

supplier\_type VARCHAR(45)

);

CREATE TABLE item (

item\_key INT PRIMARY KEY,

item\_name VARCHAR(45),

brand VARCHAR(45),

type VARCHAR(45),

supplier\_key INT,

FOREIGN KEY (supplier\_key) REFERENCES supplier(supplier\_key)

);

-- Create Fact Table

CREATE TABLE sales (

time\_key INT,

item\_key VARCHAR(45),

branch\_key VARCHAR(45),

location\_key VARCHAR(45),

units\_sold VARCHAR(45),

dollars\_sold VARCHAR(45),

avg\_sales VARCHAR(45),

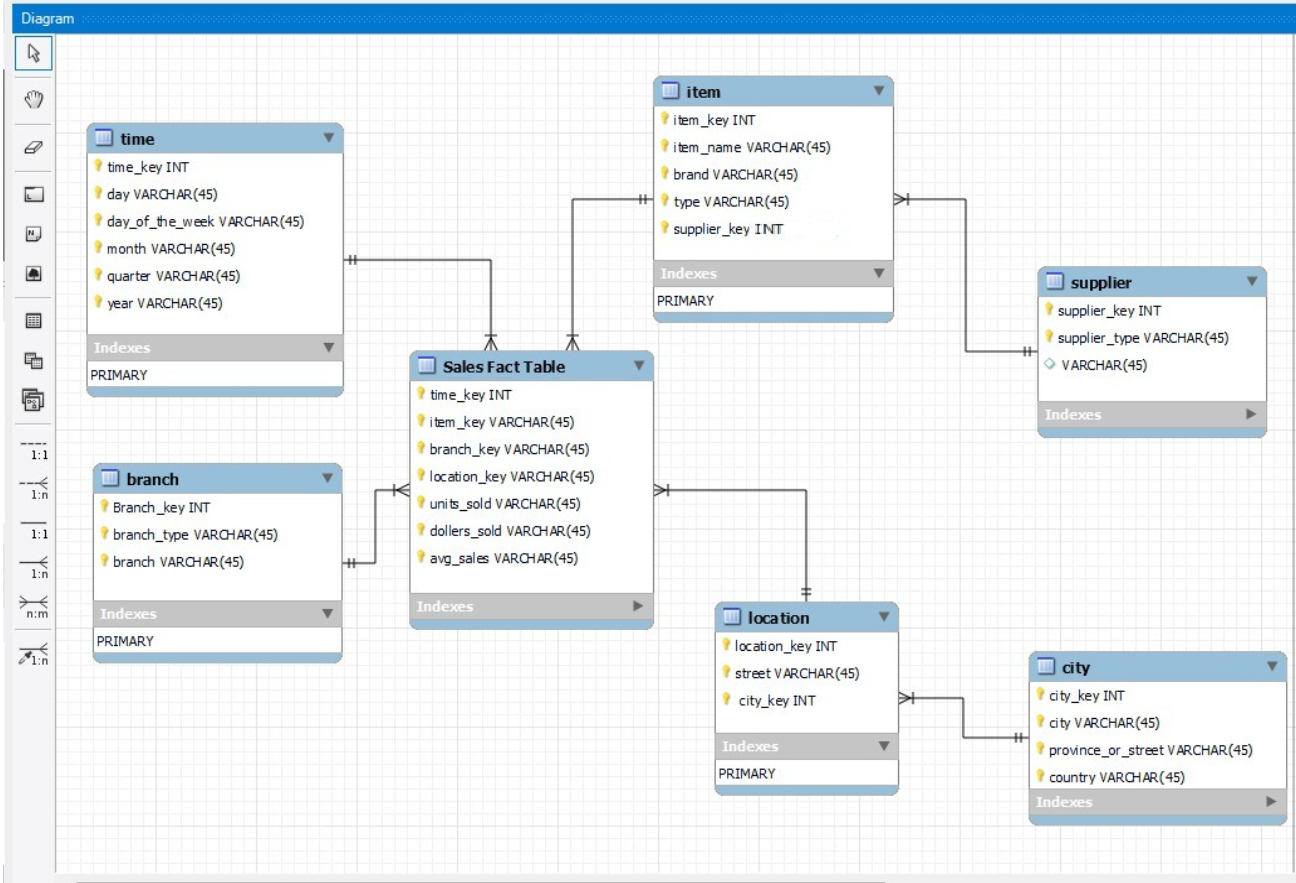
FOREIGN KEY (time\_key) REFERENCES time(time\_key),

FOREIGN KEY (item\_key) REFERENCES item(item\_key),

FOREIGN KEY (branch\_key) REFERENCES branch(branch\_key),

FOREIGN KEY (location\_key) REFERENCES location(location\_key)

);

****

**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 by linking dimension tables to a central fact table, simplifying queries and improving database performance in data warehousing.