Name:	**James Lewis**
Roll No:	**32**
Class/Sem:	TE/V
Experiment No.:	1
Title:	Data Warehouse Construction – Star schema and Snowflake schema
Date of Performance:	
Date of Submission:	
Marks:	
Sign of Faculty:	

Aim: To Build a Data Warehouse – Star Schema, Snowflake Schema and Fact Constellation Schema

Objective: A data warehouse is a large store of data collected from multiple sources within a business. The objective of a data warehouse system is to provide consolidated, flexible, meaningful data storage to the end user for reporting and analysis.

Theory:

In general, the warehouse design process consists of the following steps:

1. Choose a business process to model (e.g., orders, invoices, shipments, inventory, account administration, sales, or the general ledger). If the business process is organizational and involves multiple complex object collections, a data warehouse model should be followed. However, if the process is departmental and focuses on the analysis of one kind of business process, a data mart model should be chosen.
2. Choose the business process grain, which is the fundamental, atomic level of data to be represented in the fact table for this process (e.g., individual transactions, individual daily snapshots, and so on).
3. Choose the dimensions that will apply to each fact table record. Typical dimensions are time, item, customer, supplier, warehouse, transaction type, and status.
4. Choose the measures that will populate each fact table record. Typical measures are numeric additive quantities like dollars sold and units sold.

Steps to Draw Star, Snowflake, and Fact Constellation Schemas

1. Star Schema:

- Step 1: Identify the central fact table, which contains quantitative data (e.g., sales, revenue).

- Step 2: Determine the dimension tables related to the fact table, such as time, product, customer, etc.

- Step 3: Define the relationships between the fact table and each dimension table, usually a one-to-many relationship.

- Step 4: Draw the fact table at the center and connect it to each dimension table using lines, creating a star-like structure.

2. Snowflake Schema:

- Step 1: Start with the fact table as in the Star Schema.

- Step 2: Identify the dimension tables and further normalize them by breaking them into multiple related tables (e.g., split "Location" into "Country" and "City").

- Step 3: Establish relationships between these normalized dimension tables and the fact table.

- Step 4: Draw the fact table at the center, then connect it to the dimension tables, which in turn connect to their sub-tables, forming a snowflake-like structure.

3. Fact Constellation Schema:

- Step 1: Identify multiple fact tables representing different processes or subjects (e.g., sales and inventory).

- Step 2: Identify the shared dimension tables that will connect to these fact tables.

- Step 3: Define relationships between each fact table and the shared dimension tables.

- Step 4: Draw all fact tables and connect them to the shared dimension tables, creating a constellation of facts and dimensions.

Problem Statement:

Design and construct a data warehouse for a chosen business process by developing the following schemas:

1. Star Schema:
   Create a schema with a central fact table containing quantitative business data connected directly to denormalized dimension tables representing descriptive attributes.
2. Snowflake Schema:
   Extend the star schema by normalizing one or more dimension tables into multiple related tables to eliminate redundancy, forming a snowflake-like structure.
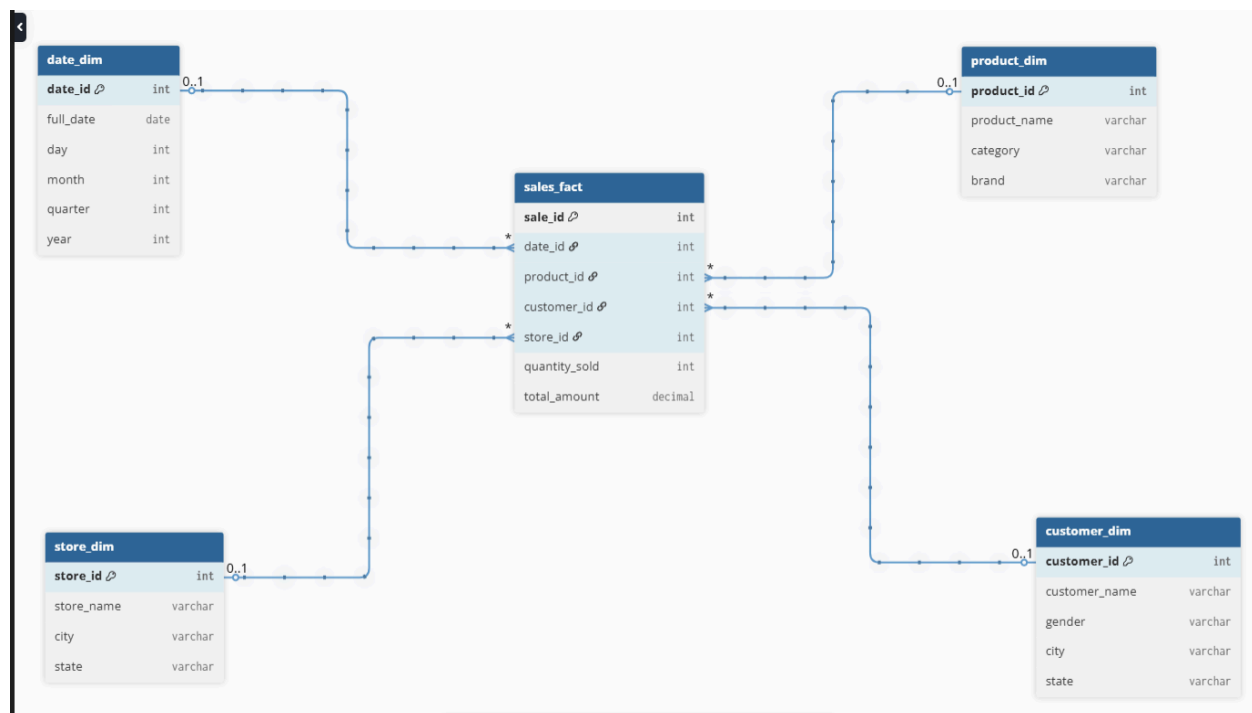3. Fact Constellation Schema:

Design a schema involving two or more fact tables representing related but distinct business processes, sharing common dimension tables to form a constellation of facts and dimensions.

Tasks:

- Select a business process such as Sales, Orders, or Inventory.
- Identify the fact(s), dimension(s), and measures relevant to the chosen business process.
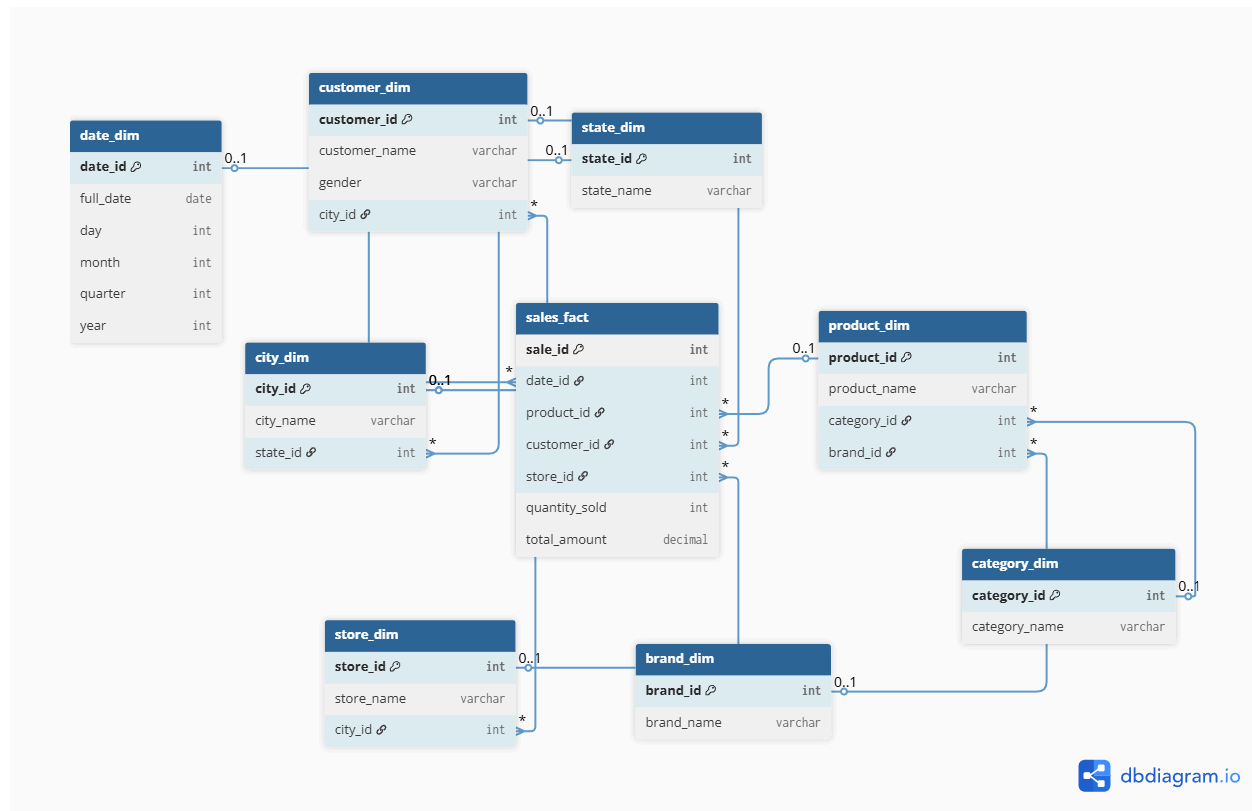- Design and draw diagrams for all three schema types.

Construction of Star schema, Snowflake schema and Fact Constellation Schema:
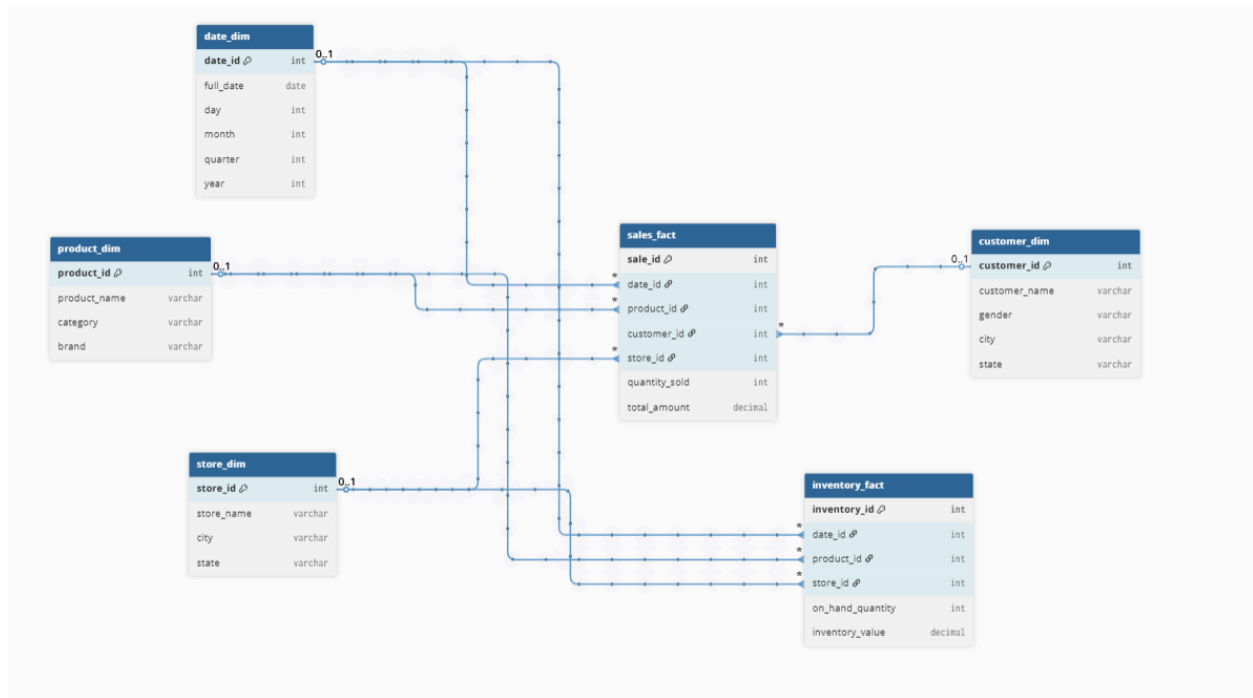
Star Schema

Snowflake Schema (Sales, Normalized Dimensions)

Fact Constellation Schema (Sales + Inventory Analysis)

Conclusion:

1. How does the Snowflake Schema compare to the Star Schema in terms of ease of maintenance and scalability?

→

The Snowflake Schema and Star Schema are two common data warehouse modeling techniques, and they differ significantly in terms of ease of maintenance and scalability:

Ease of Maintenance

- Star Schema:
    - Dimension tables are denormalized (all descriptive data stored together).
    - This simplicity makes the schema easy to understand, maintain, and modify. If you need to add attributes or update descriptions, the changes are straightforward.
    - Drawback: More data redundancy—updating a single attribute (like a city name) requires updates in multiple rows if it's repeated.
- Snowflake Schema:
    - Dimension tables are normalized (broken into related sub-tables—e.g., splitting a "Location" dimension into separate City and State tables).
    - Less redundancy and better data integrity: if you need to update a city name, you update it in one place.

- Drawback: More tables and relationships make the schema more complex to maintain and modify. Changes often require navigating through additional relations.

Scalability
- Star Schema:
  - Easier horizontal scalability—adding new fields or dimension tables is generally simpler because of fewer relationships.
  - Query performance is often better since there are fewer joins.
  - As the data volume grows, the denormalized nature may lead to increased storage usage.
- Snowflake Schema:
  - Scales well for very large, complex datasets. Normalization saves disk space and keeps updates efficient.
  - However, as tables and relations increase, queries may require more joins, which can slow performance on very complex queries.
  - Schema design can accommodate very large datasets without as much risk of redundancy or inconsistency.

Aspect	Star Schema	Snowflake Schema
Structure	Denormalized dimensions	Normalized dimensions
Maintenance	Easier, less complex, some redundancy	More complex, less redundancy
Update Effort	May require updating multiple rows	Usually update once in one table
Query Performance	Faster (fewer joins needed)	Potentially slower (more joins)
Scalability	Simple to add fields/dimensions	More scalable for huge datasets

CSL503: Data warehousing and Mining Lab

2. How do you manage the complexity of ETL (Extract, Transform, Load) processes in a Fact Constellation Schema?

→

To manage ETL complexity in a Fact Constellation Schema, use a modular design to break down workflows, automate repetitive tasks, and maintain clear documentation. Implement data quality checks, utilize version control for scripts, and monitor performance for optimization. These strategies simplify and enhance the effectiveness of ETL processes.