03 Star Schema vs Snowflake Schema

Star Schema vs Snowflake Schema

Introduction

In dimensional modeling for data warehousing and business intelligence, **Star Schema** and **Snowflake Schema** are two fundamental approaches. Both are designed to support analytical queries efficiently but differ in structure, normalization, and performance characteristics. Understanding these differences is essential for designing scalable and maintainable data warehouses.

Star Schema

Definition

A **Star Schema** is a type of data warehouse schema where a central **fact table** is directly connected to multiple **dimension tables**, forming a star-like structure. All dimension tables are denormalized, containing all the descriptive attributes needed for analysis in a single table.

Structure

- **Fact Table**: Stores quantitative data (measures) such as revenue, sales, quantity, etc.
- **Dimension Tables**: Contain descriptive, textual, or categorical attributes such as product name, customer region, and sales date.

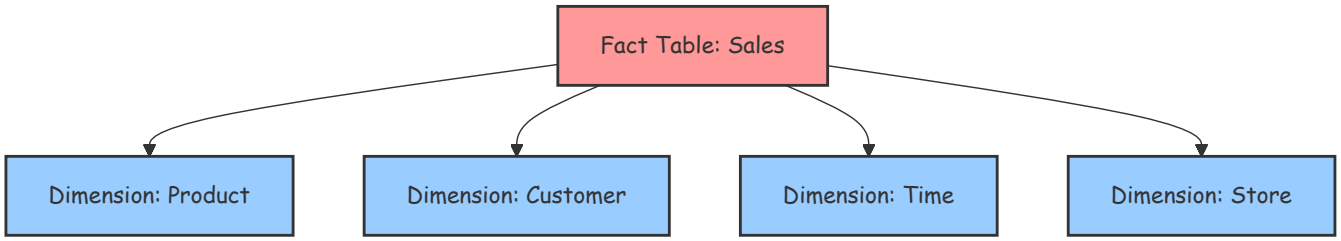
Characteristics

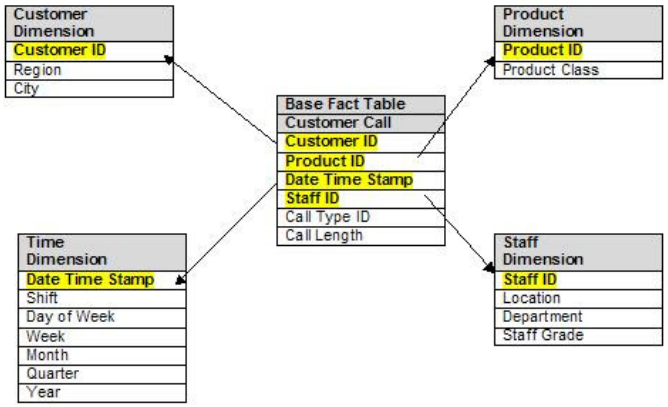
Feature	Description
Normalization	Denormalized
Query Simplicity	Easy to write and interpret
Storage	Redundant data; uses more space
Performance	Optimized for fast read queries
Maintenance	Simpler structure, easier to manage
Use Case Suitability	Best for environments with frequent querying and simpler data hierarchies

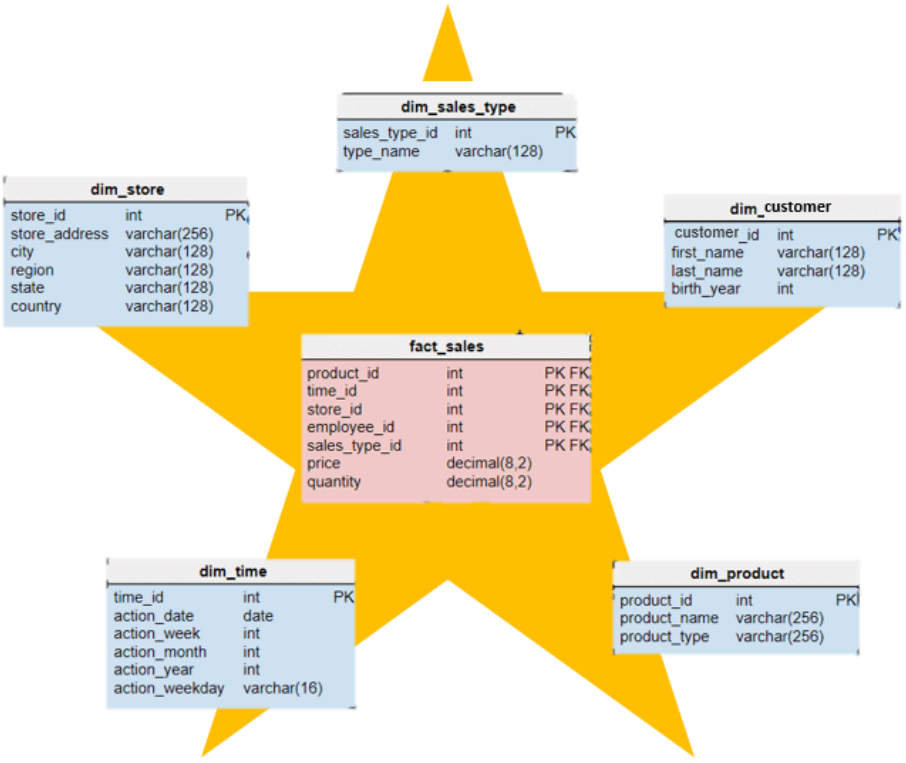
Real-World Examples

1. **Retail Sales Analytics**
   - **Fact Table**: Sales
   - **Dimensions**: Product, Customer, Time, Store
   - Used for analyzing sales trends by product and region.
2. **Hospital Visit Records**
   - **Fact Table**: Patient Visits
   - **Dimensions**: Patient, Doctor, Department, Diagnosis, Visit Date
   - Used for tracking medical service usage across departments.

Star Schema Diagram







Snowflake Schema

Definition

A **Snowflake Schema** is a more complex version of the Star Schema where dimension tables are **normalized** into multiple related tables. This reduces data redundancy but increases the number of joins required for queries.

Structure

- **Fact Table**: Same as in star schema, contains numerical measures.

- **Normalized Dimensions**: Each dimension is broken into related tables (sub-dimensions) organized hierarchically.

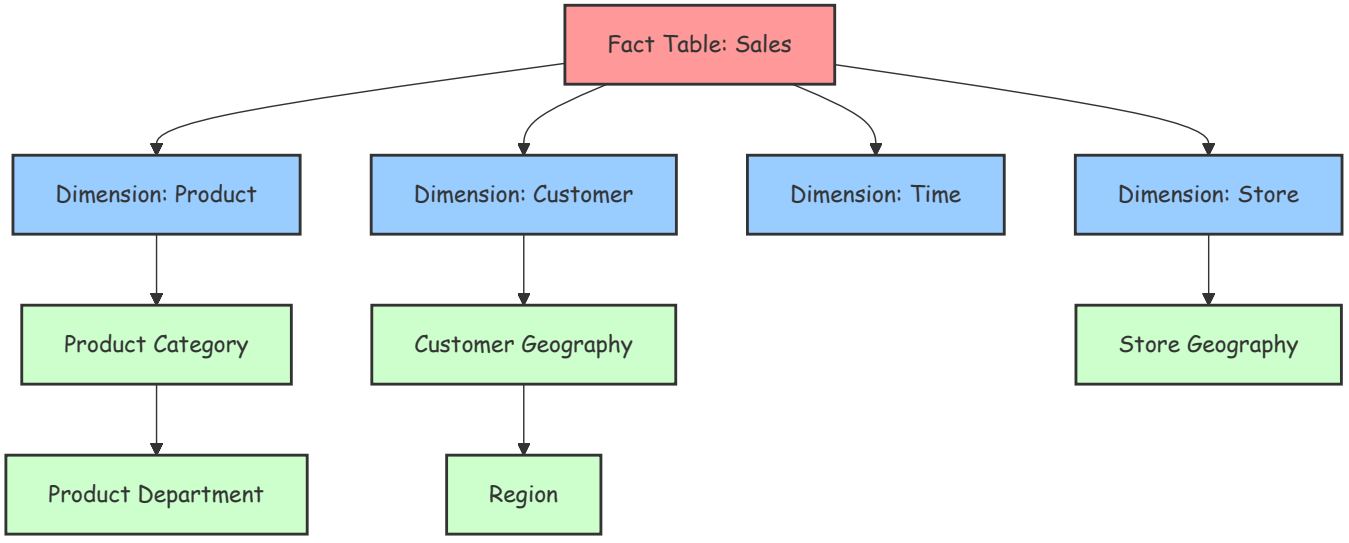
Characteristics

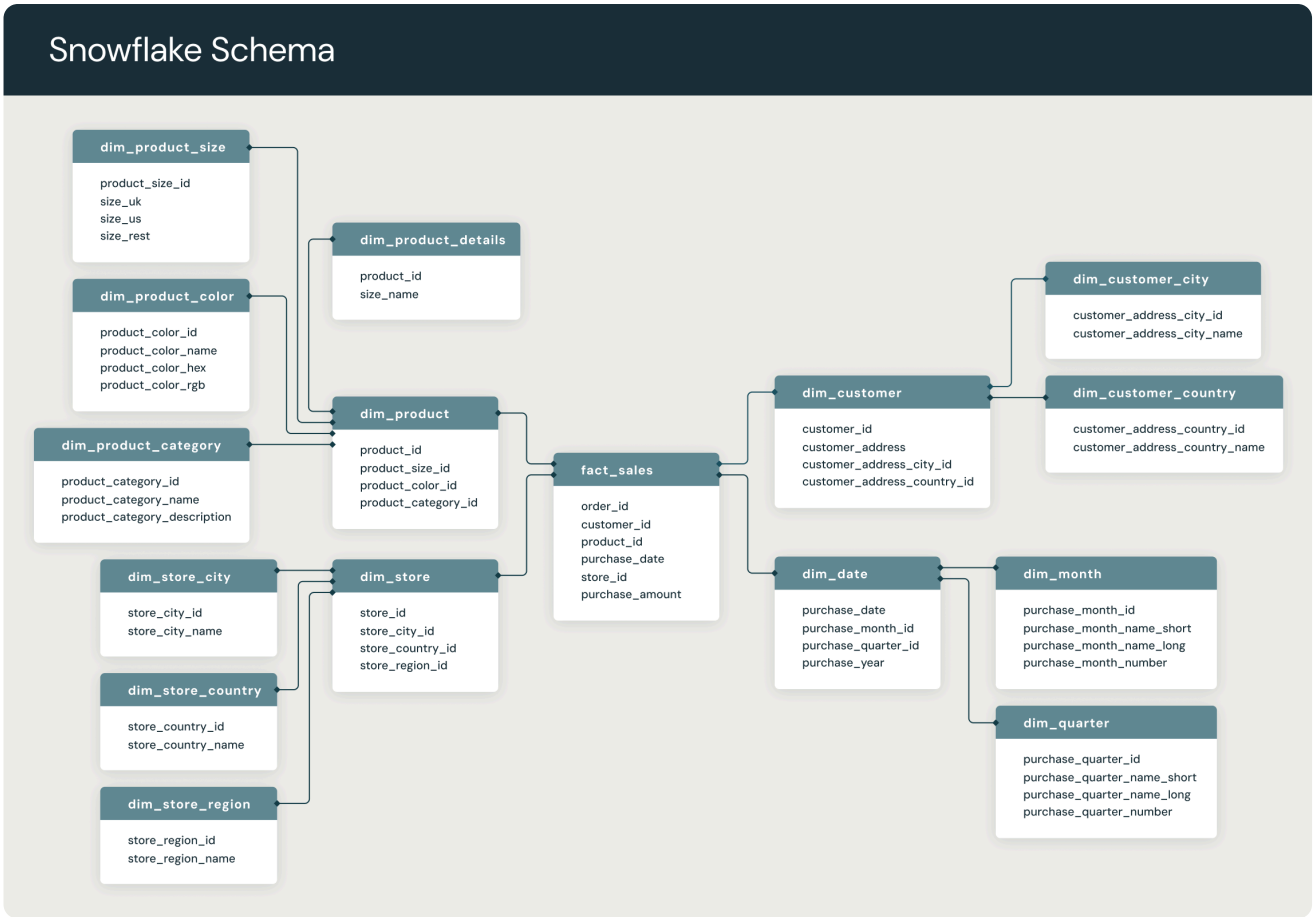
Feature	Description
Normalization	Normalized dimension tables
Query Complexity	Requires more complex joins
Storage	Reduced redundancy; saves space
Performance	Slightly slower due to joins
Maintenance	More complex to manage
Use Case Suitability	Suitable for large datasets with hierarchical dimensions and a focus on storage efficiency

Real-World Examples

1. **E-commerce Analytics**
   - **Fact Table**: Orders
   - **Dimensions**:
     - Product → Category → Supplier
     - Customer → Geography → Region
   - Used to track customer behavior and product performance across categories and suppliers.
2. **University Reporting System**
   - **Fact Table**: Course Enrollments
   - **Dimensions**:
     - Student → Program → School
     - Course → Department → Faculty
   - Used to analyze academic performance and departmental statistics.

Snowflake Schema Diagram





Comparison: Star Schema vs Snowflake Schema

Feature	Star Schema	Snowflake Schema
Normalization	Denormalized	Normalized (multi-level dimension tables)
Query Simplicity	Fewer joins, simpler queries	More joins, more complex queries
Performance	Faster query performance	Slightly slower due to join overhead
Storage Requirements	Higher, due to data duplication	Lower, due to normalization
Design Complexity	Simple and easy to understand	More complex and harder to maintain
Data Integrity	Moderate (some redundancy)	Higher (minimal redundancy)
Best Use Case	Interactive dashboards, OLAP reporting	Enterprise-level reporting, normalized structures

Conclusion

- **Star Schema** is typically preferred when **performance and simplicity** are more important than storage efficiency.
- **Snowflake Schema** is better when **data consistency**, **storage optimization**, and support for complex **hierarchical relationships** are priorities.
- In practice, many data warehouses use a **hybrid approach**, employing denormalization where performance matters and normalization where data consistency is critical.