**Star and Snowflake Schema**

**1. Schema Designs**

Star Schema Design (Denormalized)

**Fact Table:**

* Sales\_Fact:
  + sale\_id (PK)
  + product\_id (FK)
  + customer\_id (FK)
  + time\_id (FK)
  + store\_id (FK)
  + quantity
  + total\_amount

**Dimension Tables (denormalized):**

* Product\_Dim: product\_id, product\_name, category\_name, brand
* Customer\_Dim: customer\_id, customer\_name, age, gender, city, state, country
* Time\_Dim: time\_id, date, day, month, quarter, year
* Store\_Dim: store\_id, store\_name, city, state, region

**Diagram:**

Product\_Dim

|

Customer\_Dim — Sales\_Fact — Time\_Dim

|

Store\_Dim

**Snowflake Schema Design (Normalized)**

**Fact Table:**

* Sales\_Fact:
  + sale\_id (PK)
  + product\_id (FK)
  + customer\_id (FK)
  + time\_id (FK)
  + store\_id (FK)
  + quantity
  + total\_amount

**Normalized Dimension Tables:**

* Product\_Dim: product\_id, product\_name, category\_id
* Product\_Category\_Dim: category\_id, category\_name, department
* Customer\_Dim: customer\_id, customer\_name, age, gender, location\_id
* Customer\_Location\_Dim: location\_id, city, state, country
* Time\_Dim: time\_id, date, day, month, quarter, year
* Store\_Dim: store\_id, store\_name, location\_id
* Store\_Location\_Dim: location\_id, city, state, region

**Diagram:**

Product\_Category\_Dim

|

Product\_Dim

|

Customer\_Location\_Dim — Customer\_Dim Store\_Location\_Dim —Store\_Dim

\ | | /

Sales\_Fact

|

Time\_Dim

**2. Dimensional Modelling Differences**

| **Aspect** | **Star Schema** | **Snowflake Schema** |
| --- | --- | --- |
| **Design** | Flat, denormalized | Hierarchical, normalized |
| **Joins** | Few joins needed | Multiple joins across dimension tables |
| **Dimension Table Example** | Customer includes location fields | Customer and Location in separate tables |
| **Storage** | Repetitive data in dimensions | Data organized efficiently |

**3. Performance and Storage Analysis**

| **Characteristic** | **Star Schema** | **Snowflake Schema** |
| --- | --- | --- |
| **Query Performance** | High (fewer joins, better for OLAP) | Moderate to low (more joins) |
| **Storage Efficiency** | Low (data redundancy) | High (normalized, no redundancy) |
| **Join Complexity** | Simple (flat structure) | Complex (multi-level joins) |
| **Maintenance** | Easier to understand, modify | Harder to manage due to normalization |
| **ETL Speed** | Faster loads due to fewer tables | Slower loads, more transformations required |

**4. Practical Considerations & Use Cases**

| **Consideration** | **Star Schema** | **Snowflake Schema** |
| --- | --- | --- |
| **Use Case** | Fast analytics dashboards, OLAP systems | Complex, scalable systems with strict data governance |
| **Best For** | BI tools like Power BI, Tableau | Data warehouses focused on efficiency (e.g., Snowflake, BigQuery) |
| **Trade-Offs** | Redundancy vs simplicity | Performance vs maintainability |
| **Real-World Example** | E-commerce sales report generation for managers | Scalable e-commerce data warehouse with multiple apps accessing |

**Summary Table**

| **Characteristic** | **Star Schema** | **Snowflake Schema** |
| --- | --- | --- |
| **Number of Tables** | 4-5 | 7-8 |
| **Normalization** | Denormalized | Highly Normalized |
| **Query Performance** | Faster | Potentially Slower |
| **Storage Efficiency** | Less Efficient | More Efficient |
| **Complexity** | Simpler | More Complex |
|  |  |  |