Quick Revision Tour:

OLAP (Online Analytical Processing) VS OLTP (Online Transactional Processing):

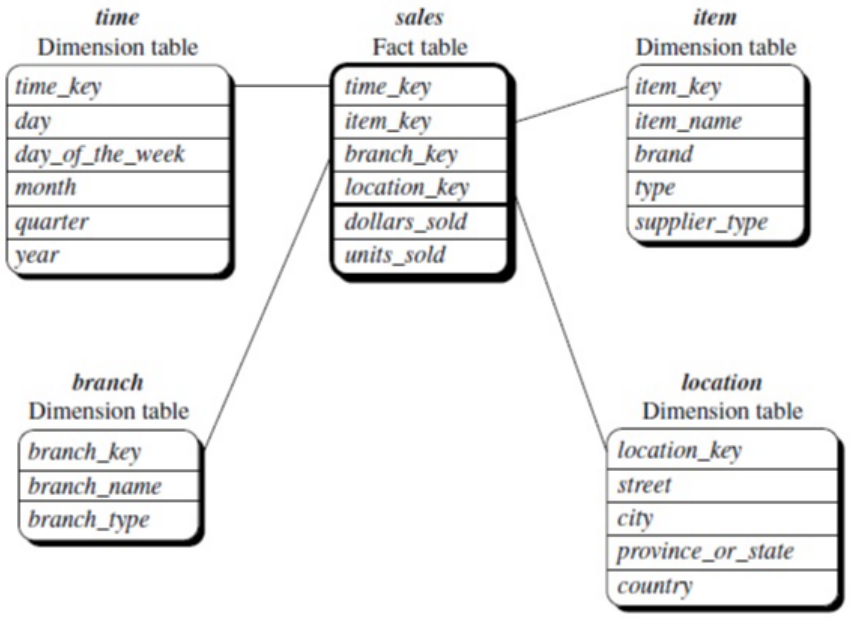
Aspect	**OLAP**	**OLTP**
Purpose	Analytical processing for decision-making	Transactional processing for day-to-day operations
Focus	Historical data analysis	Real-time data manipulation
Database Design	Star or snowflake schema	Normalized schema
Queries	Complex queries for reporting and analysis	Simple queries for CRUD operations
Data	Aggregated and summarized	Detailed and granular
Usage	Read-intensive	Read and write-intensive
Performance	Optimized for complex queries	Optimized for fast transaction processing
Examples	Data warehousing systems	Banking systems, e-commerce platforms

Data Warehouse VS Data Lake

Aspect	**Data Warehouse**	**Data Lake**
Data Structure	Structured data (tables, rows, columns)	Structured, semi-structured, and unstructured data
Data Storage	Centralized storage system	Distributed storage system
Data Processing	ETL (Extract, Transform, Load) processes	ELT (Extract, Load, Transform) processes
Schema	Schema-on-write (data is structured before storage)	Schema-on-read (data is structured upon retrieval)
Data Variety	Primarily structured data	Supports structured, semi-structured, and unstructured data
Use Cases	Standardized reporting, business intelligence	Advanced analytics, machine learning, exploratory analysis
Agility	Less flexible in handling diverse data types	Highly flexible, supports diverse data types and formats
Scalability	Limited scalability due to predefined schemas	Highly scalable, can handle large volumes of data and diverse data types
Cost	Typically higher upfront costs	May offer cost savings due to use of commodity hardware and open-source software

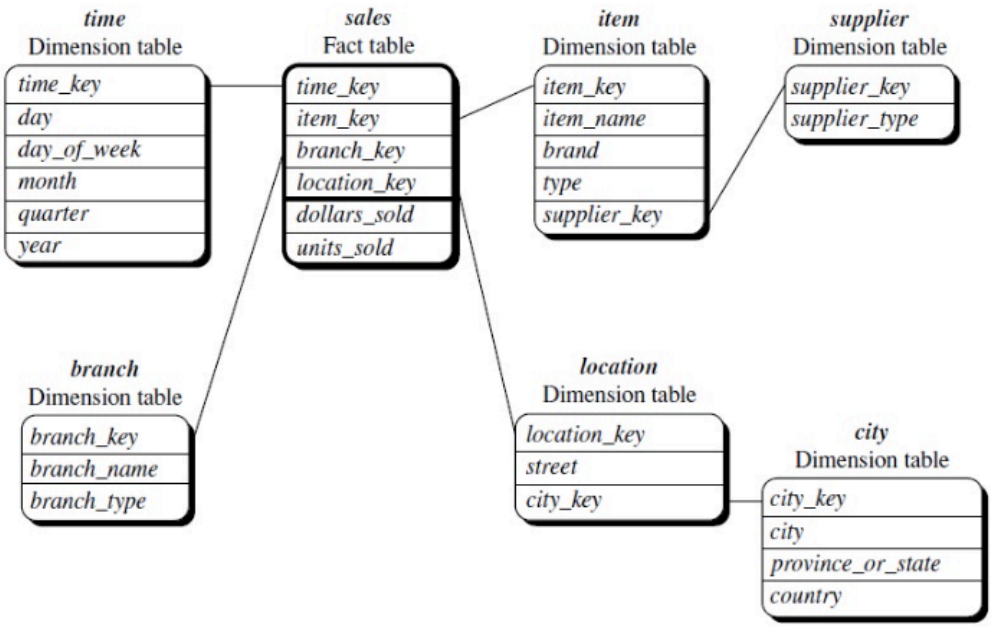
Feature	Star Schema	Snowflake Schema	Galaxy Schema	Fact Constellations
Structure	Fact table at center with denormalized dimensions	Fact table at center with normalized dimensions	Hybrid approach with some normalized and denormalized dimensions	Multiple fact tables sharing dimension tables
Dimension Tables	Denormalized	Normalized	Combination of normalized and denormalized	Shared across multiple fact tables
Fact Tables	Single central fact table	Single central fact table	Single central fact table	Multiple fact tables
Performance	Generally good performance due to denormalization	May experience slower performance due to normalization	Balanced performance between normalization and denormalization	Performance depends on individual fact tables
Use Cases	Suitable for performance-critical applications	Suitable for storage-efficient applications	Offers flexibility in balancing storage and performance	Suitable for scenarios with multiple related facts

Example



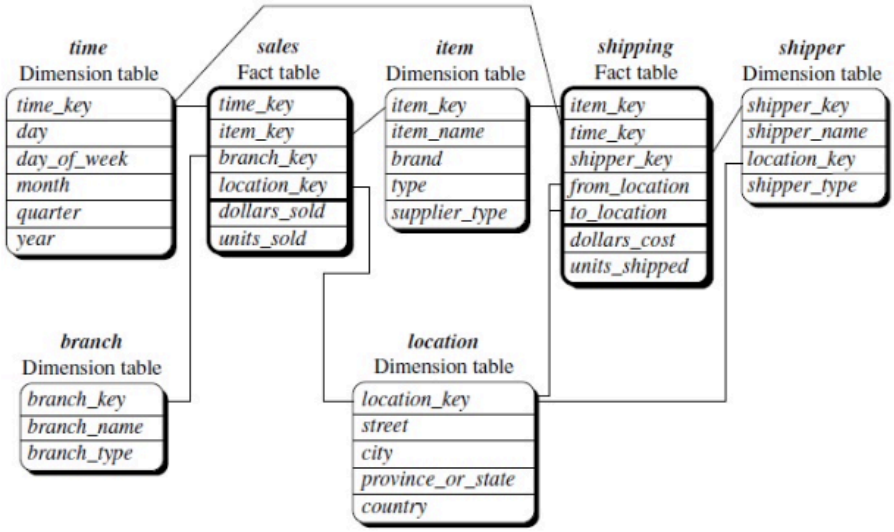
Star schema of *sales* data warehouse.

Example



Snowflake schema of a *sales* data warehouse.

Example



Fact constellation schema of a sales and shipping data warehouse.

Techniques for Ingesting Unstructured Data into Databases

Ingesting unstructured data into databases requires specific techniques to transform it into a format suitable for storage and analysis. Here are some common methods:

1. Extract, Transform, Load (ETL):

- **Extract:** Data is retrieved from its source, which can be diverse like social media feeds, log files, emails, etc. APIs or custom scripts are often used for this step.
- **Transform:** The raw data is cleaned, filtered, and enriched with additional information to make it meaningful. This may involve techniques like:
    - **Normalization:** Standardizing data formats and removing inconsistencies.
    - **Entity recognition:** Identifying entities like people, locations, and organizations within text.
    - **Sentiment analysis:** Understanding the emotional tone of text data.
    - **Image recognition:** Extracting features and tags from images.
- **Load:** The transformed data is loaded into the target database, which could be a traditional relational database, a NoSQL database, or a data lake.

Example: Extracting tweets containing product reviews, identifying sentiment (positive, negative, neutral), and loading them into a database for sentiment analysis.

2. Schema-on-Read:

- Unlike ETL, data is loaded into the database in its raw format without upfront schema definition.
- Schema is applied only when querying the data, allowing for flexible analysis of diverse data types.
- Suitable for data exploration and initial analysis before defining a fixed structure.

Example: Loading social media posts directly into a NoSQL database like MongoDB and applying schema later for specific analysis tasks.

3. Semi-structured Storage:

- Data is stored in formats like JSON or XML, which offer some structure but are not as rigid as relational tables.
- Enables efficient storage and retrieval of complex data with nested elements.

- Often used for storing documents, logs, and configuration files.

Example: Storing product information with details like name, price, and reviews in a JSON format within the database.

4. Specialized Databases:

- Specific database solutions are designed for handling particular types of unstructured data, such as:
    - **Document databases:** Efficiently store and query large collections of documents like text files and emails (e.g., MongoDB, Elasticsearch).
    - **Graph databases:** Represent relationships between entities and are ideal for social network analysis and recommendation systems (e.g., Neo4j, Amazon Neptune).

Choosing the right technique depends on factors like:

- **Data volume and velocity:** Batch or real-time processing requirements.
- **Data complexity:** Level of transformation needed before storage.
- **Analysis needs:** Flexibility for querying and exploring diverse data.

By combining these techniques with appropriate tools and technologies, organizations can effectively ingest and leverage the valuable insights hidden within unstructured data.

Extracting structured information from unstructured data involves a number of steps and faces several challenges. Here's a breakdown:

Steps:

1. **Data identification:** This involves understanding the type of information you want to extract from the unstructured data source. This could be anything from names and addresses to product reviews or financial data.
2. **Data preparation:** This step involves cleaning and pre-processing the data to make it suitable for further processing. This may involve removing irrelevant information, correcting errors, and standardizing formats.

3. **Information extraction:** This is the core step where relevant information is identified and extracted from the unstructured data. This can be done using various techniques, such as:
    - **Rule-based methods:** These methods rely on predefined rules and patterns to identify and extract specific information.
    - **Machine learning:** Machine learning algorithms can be trained on labeled data to automatically learn patterns and extract information from unseen data.
    - **Natural language processing (NLP):** NLP techniques are specifically designed to process and understand human language, making them valuable for extracting information from text data.
4. **Data validation:** Once information is extracted, it's crucial to verify its accuracy and consistency. This may involve manual review or using additional validation techniques.
5. **Data storage:** The extracted structured information is then stored in a suitable format, such as a relational database or data warehouse, for further analysis and use.

Challenges:

- **Data variability:** Unstructured data can be highly variable in terms of format, language, and content, making it difficult to define consistent extraction rules.
- **Data complexity:** Unstructured data can be complex and nuanced, requiring advanced techniques like NLP to accurately extract meaningful information.
- **Data quality:** The quality of the extracted information directly depends on the quality of the raw data and the effectiveness of the extraction process.
- **Scalability:** Processing large volumes of unstructured data can be computationally expensive and require scalable infrastructure and algorithms.

Choosing the right techniques and tools for extracting structured information from unstructured data depends on various factors, such as the specific type of data, the desired level of accuracy, and the available resources.