Module 2

DATA WAREHOUSE PLANNING AND ARCHITECTURE

Planning a Data Warehouse and Key Issues

1

Planning and Project Management are crucial in constructing a data warehouse.

2

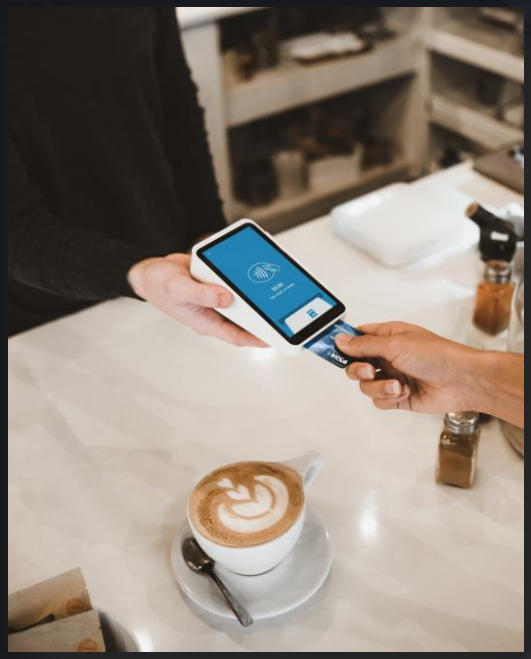
The project team must have a clear understanding of the organization's business objectives and data requirements.

3

Key issues that need to be addressed during planning include identifying the data sources, selecting appropriate technology platforms

4

Defining the scope and goals of the data warehouse project, and establishing a project plan that includes timelines, budgets, and resource allocation.



Stages of Data Warehouse

- Requirements Gathering
- Data Modeling
- Data Extraction, Transformation, and Loading (ETL)
- Data Storage
- Data Integration and Consolidation
- Data Quality Assurance
- Metadata Management
- Query and Analysis
- Data Visualization and Reporting
- Performance Tuning and Optimization
- Ongoing Maintenance and Support
- Data Governance and Security

Enterprise Data Warehouse (EDW): An enterprise data warehouse is a centralized repository that integrates data from various sources across an entire organization. It serves as a comprehensive and unified view of the organization's data, supporting enterprise-wide reporting, analytics, and decision-making. An EDW is designed to store large volumes of historical data and provide a consistent and reliable source of information.

Enterprise Data Warehouse

Operational Data Store (ODS): An operational data store is a database that stores real-time or near-real-time data from operational systems. It acts as a staging area for integrating and consolidating data from multiple sources before it is further processed and loaded into a data warehouse or used for operational reporting. An ODS focuses on operational and transactional data, providing quick access to the most current data for operational decision-making.

Operational Data Store

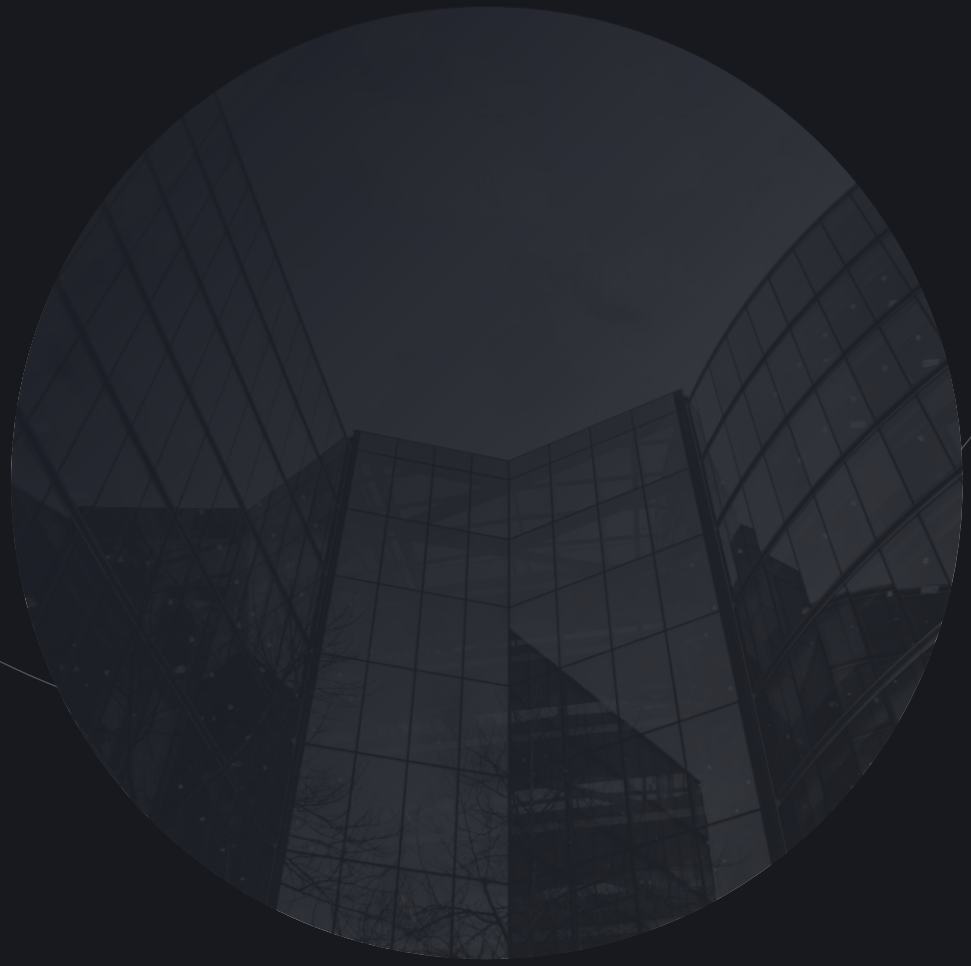
Data Mart: A data mart is a subset of a data warehouse that is focused on a specific business function, department, or user group within an organization. It contains a subset of data relevant to the specific needs of that particular group. Data marts are designed to support specific reporting and analysis requirements, providing targeted and simplified views of data for end-users. They are typically created to address the needs of specific business units or departments, such as sales, marketing, finance, or human resources.

Data Mart

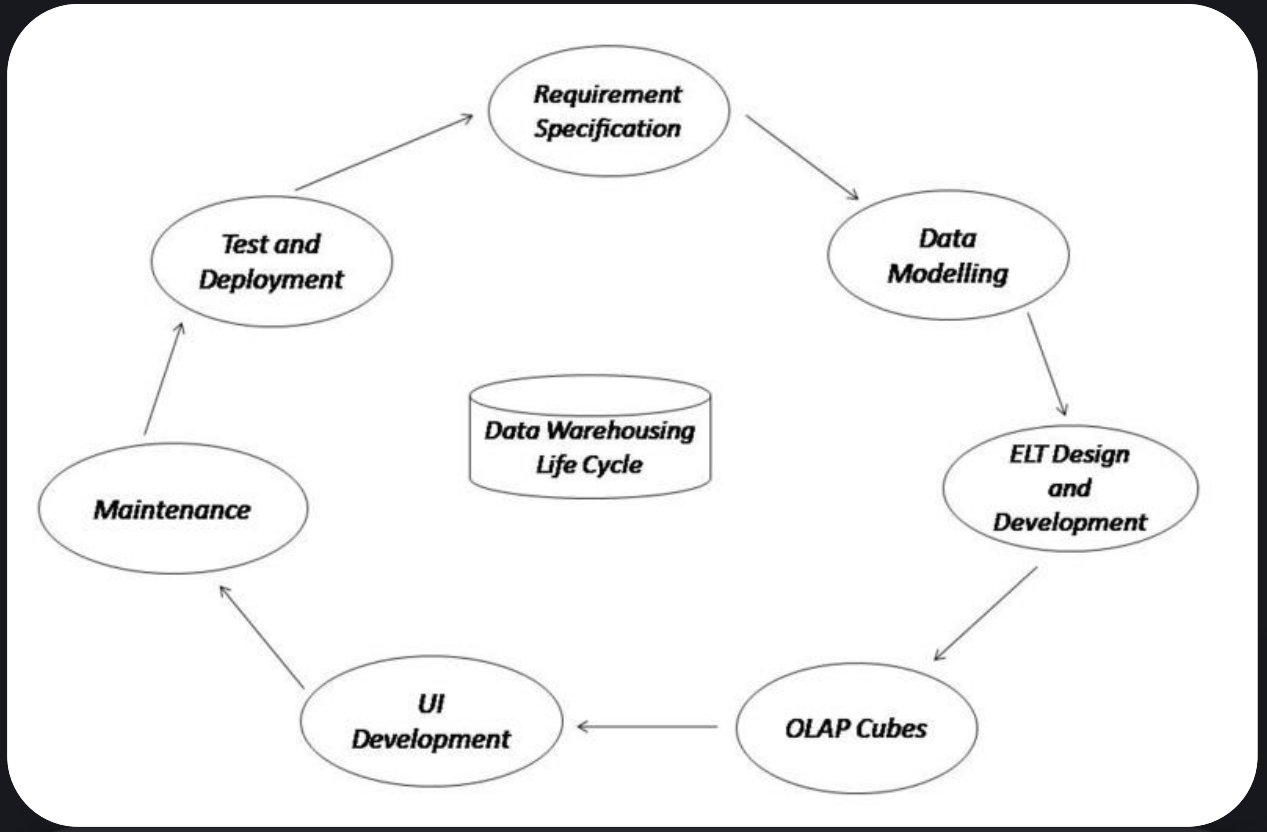
Data Warehouse Development Life Cycle

The Data Warehouse Development Life Cycle is a comprehensive framework outlining the phases of building a data warehouse. These phases typically include

- *Requirements gathering,*
- *Data modeling,*
- *Data extraction, transformation, and loading (ETL),*
- *Testing,*
- *Deployment, and*
- *Ongoing maintenance.*



- *Requirement Specification*: Gathering business requirements for the Data Warehouse.
- *Data Modelling*: Designing the structure and relationships of the data.
- *ELT Design and Development*: Extracting, transforming, and loading data into the Data Warehouse.
- *OLAP Cubes*: Creating multidimensional data structures for efficient analysis.
- *UI Development*: Creating a user-friendly interface for data exploration and reporting.
- *Maintenance*: Updating and managing the Data Warehouse schema and data.
- *Test and Deployment*: Testing the Data Warehouse and making it available to users.



Kimball Lifecycle Diagram

The Kimball Lifecycle Diagram is a visual representation of the Kimball methodology for building a data warehouse. It consists of four main phases: *requirements gathering, dimensional modeling, ETL design and development, and business intelligence (BI) application development.*

- The Inmon Lifecycle, also known as the Corporate Information Factory (CIF) methodology, is an approach to data warehousing introduced by Bill Inmon. It focuses on building a centralized data warehouse as the core component of an organization's information architecture.
- *The Inmon Lifecycle aims to build a scalable and robust data infrastructure that supports enterprise-wide reporting and analysis. It focuses on centralized data integration and the creation of a comprehensive data warehouse, providing a foundation for consistent and reliable information delivery across the organization.*

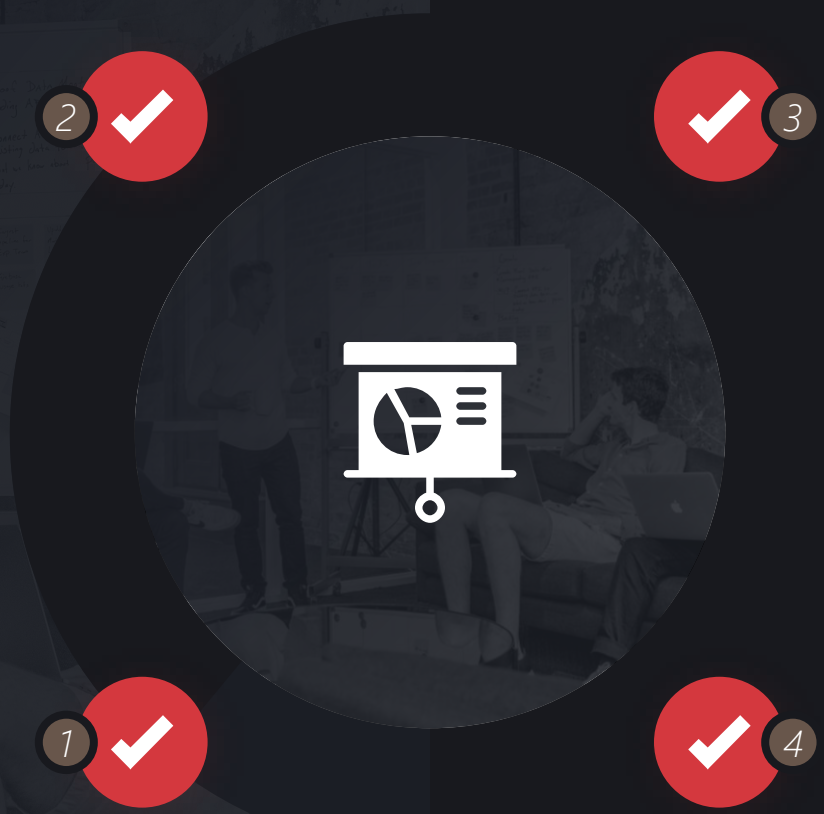
INMON V/S KIMBALL

Aspect	Inmon Methodology	Kimball Methodology
Approach	Top-down approach	Bottom-up approach
Focus	Enterprise-wide data integration and consistency	Business-specific data marts and quick deliverables
Data Warehouse Design	Data warehouse as a central repository for all data	Data marts designed for specific business areas
Data Integration	Emphasizes extensive data integration and transformation	Prioritizes data integration within individual data marts
Data Normalization	Highly normalized data structures	Dimensional data models (star/snowflake schemas)
Data Granularity	Fine-grained data granularity	Coarser data granularity
ETL Processes	Complex and time-consuming ETL processes	Simplified ETL processes, focused on business needs
Scalability	Suited for large enterprises with complex data needs	Suited for smaller or mid-sized organizations
Development Time	Longer development time due to complexity	Shorter development time with iterative approach
Flexibility	Less flexibility for agile changes and new requirements	More flexibility for adapting to changing needs
Business User Involvement	Less direct involvement of business users during development	Strong business user involvement and iterative feedback
Data Governance	Strong emphasis on centralized data governance	Balanced approach with both centralized and decentralized governance

Requirements Gathering Approaches

Requirements gathering is the process of *identifying the data requirements of the organization.*

The two main approaches to requirements gathering are top-down and bottom-up.



In a top-down approach, the project team *starts with the organization's overall business objectives and then drills down to specific data requirements*.

In contrast, a bottom-up approach *starts with specific data requirements and builds up* to the organization's overall business objectives.

Team Organization, Roles, and Responsibilities:

Team organization, roles, and responsibilities are critical to the success of a data warehouse project. The project team should have a mix of technical and business expertise, including data architects, *ETL developers, database administrators, business analysts, and project managers*. Clear roles and responsibilities should be defined for each team member, ensuring that everyone understands their roles and expectations.

- Data Architect:
  - Designs the overall data architecture of the warehouse.
  - Defines data models, schema, and data integration strategies.
  - Collaborates with business analysts to understand data requirements.
- ETL Developers:
  - Design and develop Extract, Transform, Load (ETL) processes.
  - Extract data from various source systems and transform it for loading into the warehouse.
  - Ensure data quality and integrity during the ETL process.

Team Organization, Roles, and Responsibilities:

- Database Administrators (DBAs):
  - Responsible for the performance, security, and availability of the data warehouse database.
  - Optimize database performance and manage backups and recovery processes.
  - Monitor and troubleshoot database issues.
- Business Analysts:
  - Gather business requirements and translate them into data requirements.
  - Collaborate with data architects to define the structure and content of the warehouse.
  - Ensure the data warehouse aligns with business needs and supports reporting and analysis.
- Project Manager:
  - Oversees the entire data warehouse project.
  - Manages project scope, timeline, and resources.
  - Coordinates communication among team members and stakeholders.
- Data Quality Analysts:
  - Ensure data accuracy, completeness, and consistency in the warehouse.
  - Develop and execute data quality assurance processes.
  - Identify and resolve data quality issues.

Team Organization, Roles, and Responsibilities:

- Business Intelligence (BI) Developers:
  - Develop reports, dashboards, and visualizations for data analysis.
  - Implement data access and reporting tools.
  - Collaborate with business analysts to understand reporting requirements.
- System Administrators:
  - Manage the hardware and software infrastructure supporting the data warehouse.
  - Configure and optimize servers, networks, and storage systems.
  - Ensure the availability and scalability of the data warehouse environment.
- Data Stewards:
  - Define data governance policies and standards.
  - Ensure compliance with data privacy and security regulations.
  - Establish data ownership and stewardship responsibilities.

It's important to note that these roles can vary depending on the size and complexity of the project. In some cases, individuals may take on multiple roles or there may be additional roles specific to the organization's needs.

Data Warehouse Architecture

Data warehouse architecture refers to the overall design of the data warehouse, including the technology platforms, data storage, and processing. There are three main types of data warehouse architecture: MOLAP, ROLAP, and HOLAP.

MOLAP

MOLAP architecture stores data in a multidimensional cube format

ROLAP

ROLAP architecture uses a relational database to store data.

HOLAP

HOLAP architecture combines both MOLAP and ROLAP, allowing for both multidimensional and relational data storage.

DATA MARTS

A data mart is a subset of a data warehouse that is focused on a specific business function or department within an organization.

MOLAP V/S ROLAP V/S HOLAP

Aspect	MOLAP	ROLAP	HOLAP
Data Storage	Multidimensional cubes	Relational tables	Multidimensional & Relational cubes & relational tables
Performance	Fast	Moderate	Fast in MOLAP, Moderate in ROLAP
Scalability	Limited	High	High
Query Complexity	Simple	Complex	Simple in MOLAP, Complex in ROLAP
Storage Space	High	Moderate to High	Moderate to High
Data Refresh	May require reload of entire cube	Real-time or scheduled	Mix of real-time and scheduled
Development	User-friendly tools and interfaces	SQL-based development environment	Mix of user-friendly tools and SQL-based development
Example Tools	Microsoft Analysis Services (SSAS)	Oracle OLAP, IBM Cognos, MicroStrategy	SAP BW, IBM Cognos

Dimensional Modelling

Dimensional modeling is a design technique used in data warehousing to structure and organize data in a way that supports efficient querying and analysis. It involves the identification and definition of dimensions and their attributes, which provide the context for analyzing business data.

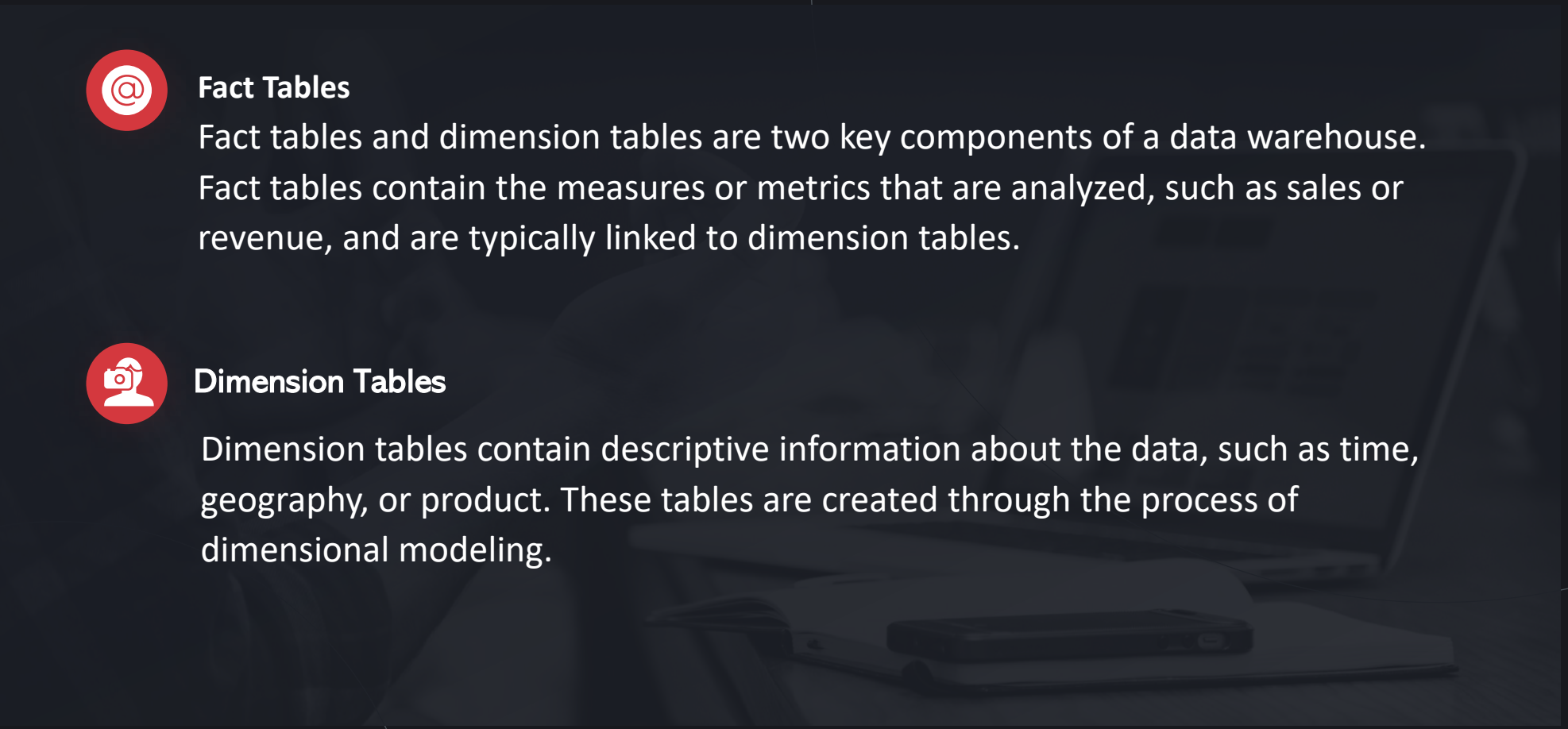
- Dimensional modeling is a technique used to design data warehouses for efficient querying and analysis.
- Dimensions represent different perspectives or characteristics of the data, such as time, geography, product, and customer.
- Attributes provide detailed information about dimensions and enable granular analysis.
- Hierarchies define relationships and levels of granularity within dimensions, enabling drill-down and roll-up analysis.
- Slowly Changing Dimensions (SCD) handle changes in dimension attributes over time, ensuring historical accuracy.
- Dimension keys are unique identifiers used to establish relationships between fact and dimension tables.
- Star schema is a widely used dimensional modeling technique with a central fact table surrounded by dimension tables.
- Snowflake schema extends the star schema by normalizing dimension tables for space optimization.
- Dimension modeling is crucial for organizing data and supporting efficient analysis in a data warehouse.

Multidimensional Data Model

Multidimensional data modeling organizes data for efficient analysis using multiple dimensions.

- Dimensions represent different attributes for analysis, such as time, geography, and product.
- Measures are the numerical values or metrics being analyzed, like sales revenue or profit.
- Cubes provide a multidimensional structure, storing data in cells based on dimension combinations.
- Hierarchies define levels of detail within dimensions, enabling drill-down and roll-up analysis.
- Aggregation summarizes data at higher levels, improving query performance.
- OLAP tools enable interactive analysis by slicing, dicing, drilling down, and rolling up data.
- Multidimensional data modeling allows for intuitive exploration and insights from various perspectives.

Creation of Fact Tables and Dimension Tables in Data Warehousing



Fact Tables

Fact tables and dimension tables are two key components of a data warehouse. Fact tables contain the measures or metrics that are analyzed, such as sales or revenue, and are typically linked to dimension tables.

Dimension Tables

Dimension tables contain descriptive information about the data, such as time, geography, or product. These tables are created through the process of dimensional modeling.

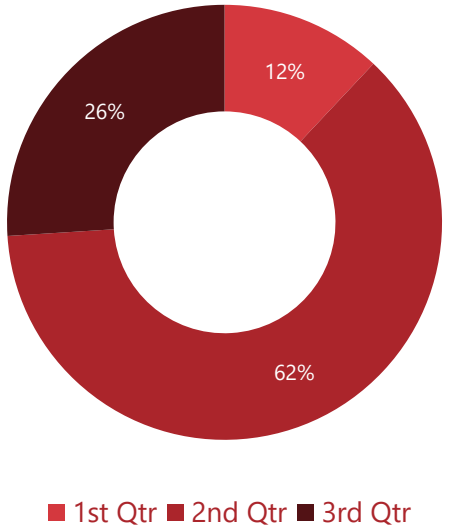
Types of Facts in a Dimension Table

Fact Type	Description
Additive	Numerical values that can be aggregated across all dimensions. They can be summed up or aggregated using mathematical operations like addition.
Semi-Additive	Numerical values that can be aggregated across some dimensions but not all. They cannot be summed up across all dimensions but can be aggregated using other operations.
Non-Additive	Numerical values that cannot be aggregated at all. They do not have meaningful aggregations across any dimension.
Factless	Tables that capture events or occurrences without associated numerical values. They contain only foreign keys to the dimension tables.

Basic Querying and Reporting on an OLAP Database

OLAP

OLAP (Online Analytical Processing) is a technology that enables users to query and analyze large data sets quickly.



OLAP databases typically use multidimensional data structures, such as cubes, to organize and store data

Basic querying and reporting on an OLAP database involves creating queries to retrieve data and generating reports that provide insights into the data.

Data Warehouse Schemas

Definition

Data warehouse schemas define the structure of the data warehouse, including how data is organized, stored, and accessed

Types of Schema

The two main types of data warehouse schemas are star schema and snowflake schema.

Star Schema

Star schema is a simple and denormalized schema that consists of a central fact table linked to several dimension tables

Snowflake Schema

Snowflake schema is a normalized schema that includes additional tables to represent complex hierarchies or relationships between dimension tables.

Use Case

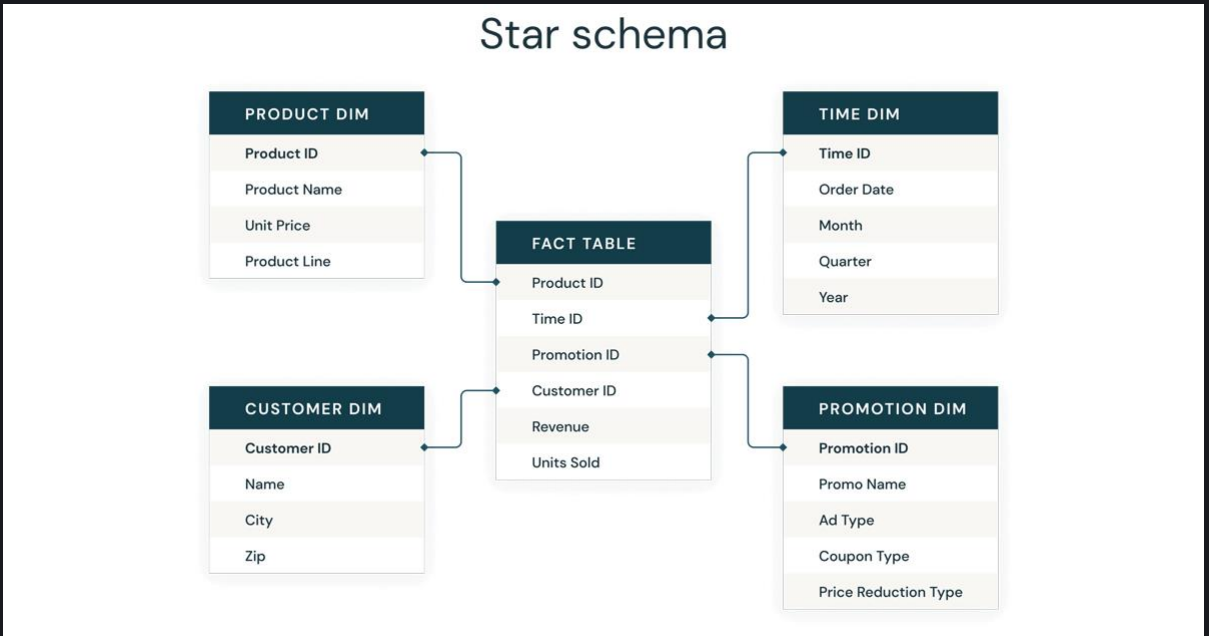
Used to logically organize and segregate data within a database, to categorize and group related database objects, Schemas help with data isolation, access control, and security permissions

Star Schema

The star schema is a widely used dimensional modeling technique in data warehousing. *It consists of a single fact table connected to multiple dimension tables in a star-like structure.*

In a star schema, *the fact table sits at the center of the schema, representing the primary focus of analysis or measurement.*

The dimension tables are connected to the fact table through these foreign keys. Each dimension table represents a specific attribute or perspective related to the measures in the fact table

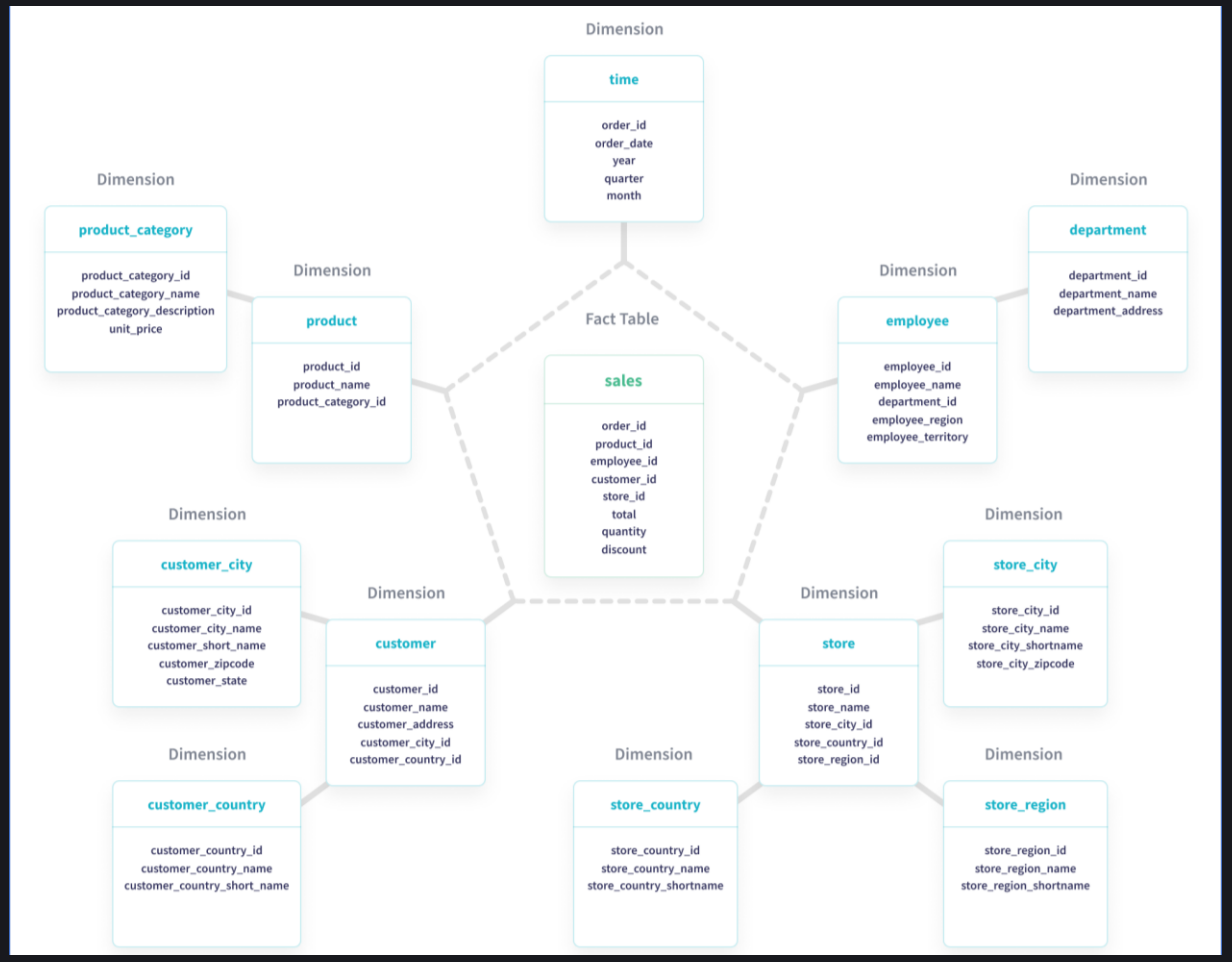


Draw this diagram to fill pages!

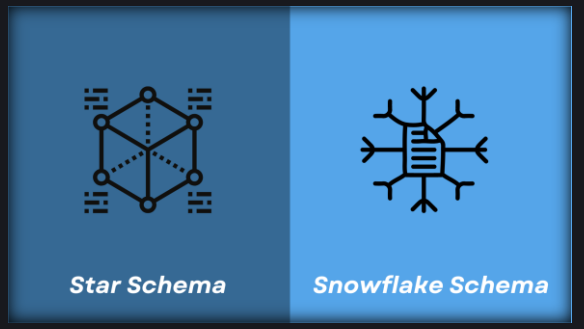
Snowflake Schema

It is an extension of the star schema and offers a more normalized structure. In a snowflake schema, *dimension tables are further normalized by splitting them into multiple related tables, resulting in a shape that resembles a snowflake*

In a snowflake schema, the fact table is connected to multiple-dimension tables, just like in a star schema. However, the dimension tables in a snowflake schema are normalized by breaking down their attributes into separate tables. This normalization *reduces data redundancy and improves data integrity*



Star Schema V/S Snowflake



Aspect	Star Schema	Snowflake Schema
Structure	Denormalized structure	Normalized structure with additional tables
Normalization	Less normalized	More normalized
Complexity	Simple and easy to understand	More complex and harder to maintain
Query Performance	Faster due to fewer joins	Slightly slower due to more joins
Storage Space	May require more storage space due to redundancy	May require less storage space due to normalization
Flexibility	Better flexibility for ad-hoc querying and reporting	Slightly lower flexibility for ad-hoc querying and reporting

OLTP V/S DATA WAREHOUSE

Aspect	OLTP	Data Warehouse
Purpose	Supports day-to-day operational transactions and processes	Supports analytical reporting and decision-making
Data Structure	Normalized data structure	Denormalized or dimensional data structure
Database Design	Emphasizes transactional integrity	Emphasizes query performance and analysis capabilities
Data Volume	Handles smaller data volumes	Handles large data volumes
Data Updates	Frequent data updates	Periodic or batch data updates
Response Time	Requires quick response times	Accepts longer query response times
Schema Design	Relational database schema	Star schema or snowflake schema
Query Types	Simple and transactional queries	Complex analytical queries
Data Granularity	Detailed, granular data	Aggregated, summarized data
User Concurrent Access	Supports high concurrent user access	Supports concurrent user access for reporting and analysis
Data Consistency	Strict transactional consistency	Historical data consistency
Performance Optimization	Optimized for transaction processing	Optimized for query performance and analysis
Example Systems and Tools	Oracle Database, MySQL, SAP ERP	Oracle Exadata, Amazon Redshift, Microsoft Azure Synapse

ETL IN DATA WAREHOUSE

Attribution Standardization and Cleansing: Standardize and clean data attributes to ensure consistency and accuracy, including resolving naming discrepancies, formatting data values, correcting misspellings, and eliminating duplicate or inconsistent data. Establish proper linking between related data entities to maintain referential integrity.

Consolidate Data Using Matching and Merge/Purge Logic: Identify and merge duplicate or related records from multiple sources. Utilize matching techniques to identify similar records and establish links between them. Apply merge/purge logic to eliminate duplicates and maintain a consolidated view of the data. Implement history tracking to capture changes and track data lineage over time.

Business Rules and Consolidation: Apply business rules to transform and consolidate data based on predefined criteria. Perform aggregations, calculations, and data manipulations while ensuring proper linking between related data elements. Capture and track changes to maintain historical data for audit and analysis purposes.

By incorporating *proper linking and history tracking* into the ETL process, you can establish data integrity, enable comprehensive analysis of historical trends, and ensure the accuracy and reliability of data within the data warehouse.