Introduction to Business Intelligence (BI) and Enterprise data management.

Hari Sankar Pandu Ranga Sri Venkatesh Jandhyala

Questions of interest to sales & marketing managers:

- What was the mix of branch, ATM, Credit Card and Debit card transactions during peak and off-peak seasons in different regions in the last three years?
- What were the monthly sales (in no of units and value) of the company's products in the 4 metro cities during the last 5 years?
- What were the region-wise weekly sales of refrigerators in the 4 summer months and 4 winter months in the last five years?

Questions of interest to HR managers:

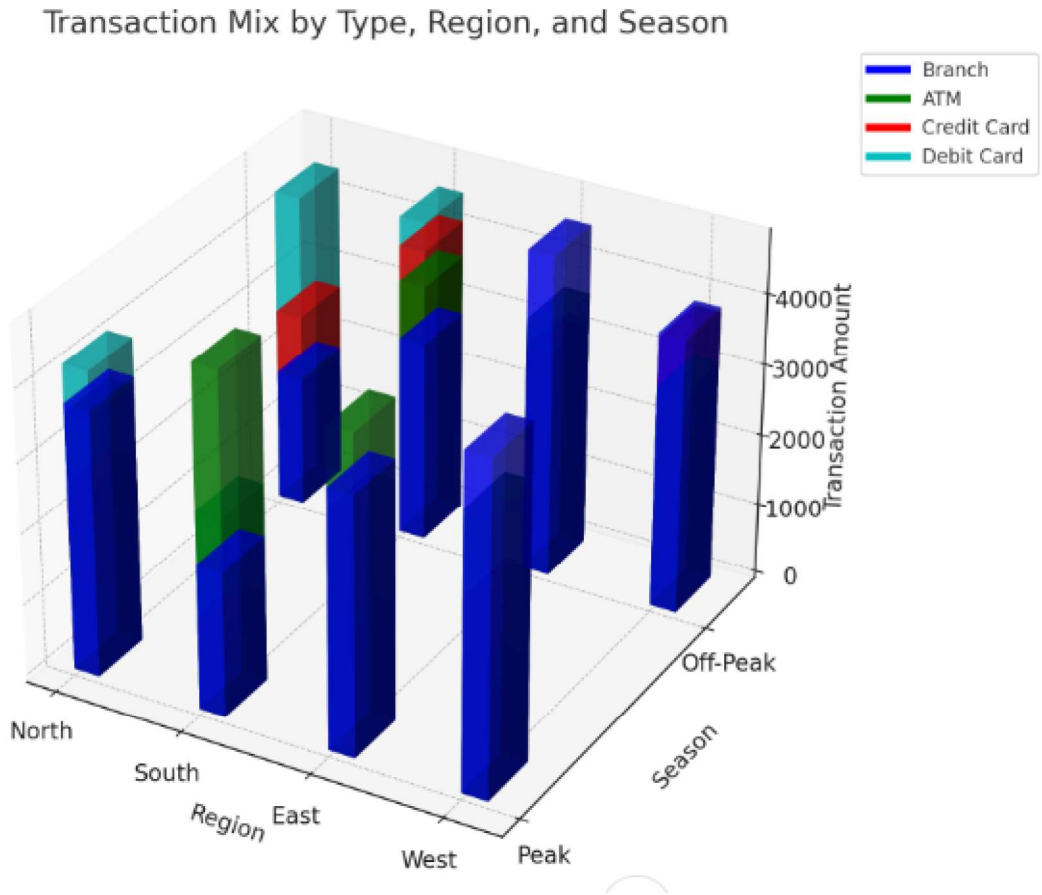
- What was the correlation between the variable pay package introduced last year and performance of sales reps?

- What is the impact of the new recruitment process on the performance of programmers on off-shore projects?

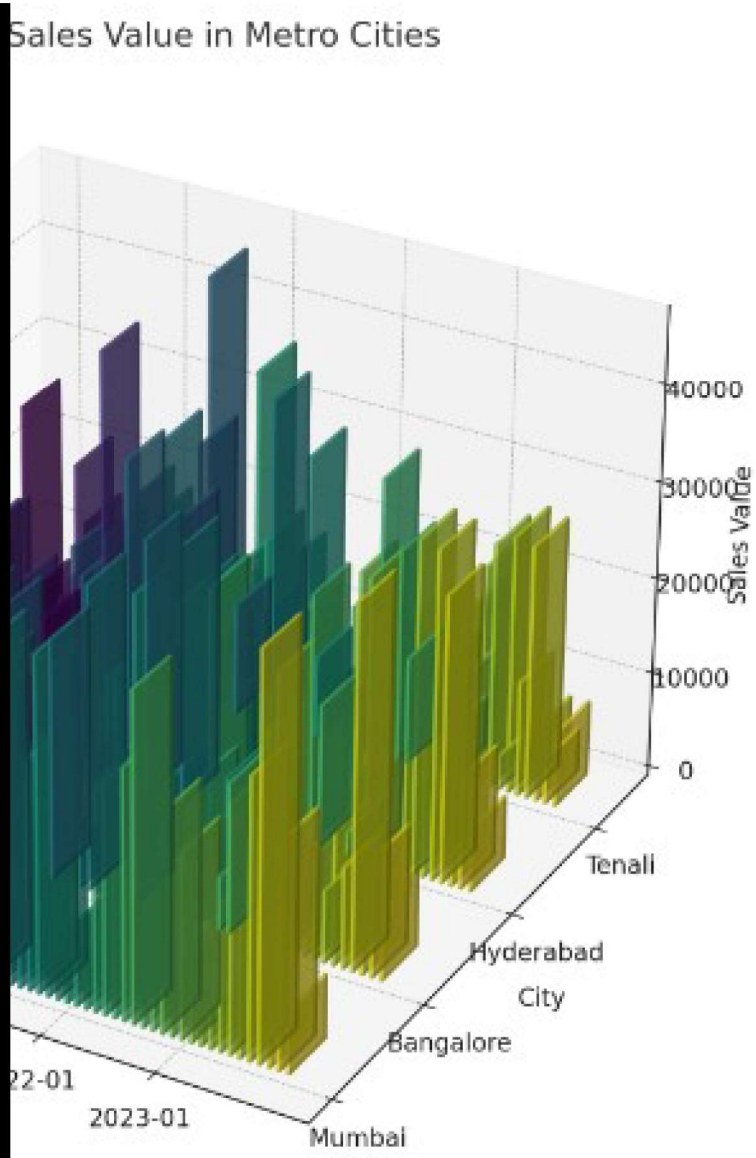
What was the mix of branch, ATM, Credit Card and Debit card transactions during peak and off-peak seasons in different regions in the last three years?

- **Data Needed:**
- Transaction data segmented by branch, ATM, credit card, and debit card.
- Seasonal categorization (peak and off-peak) for the last three years.
- Regional segmentation.

- **Data Collection Analysis**
- **Data Segmentation Visualization**

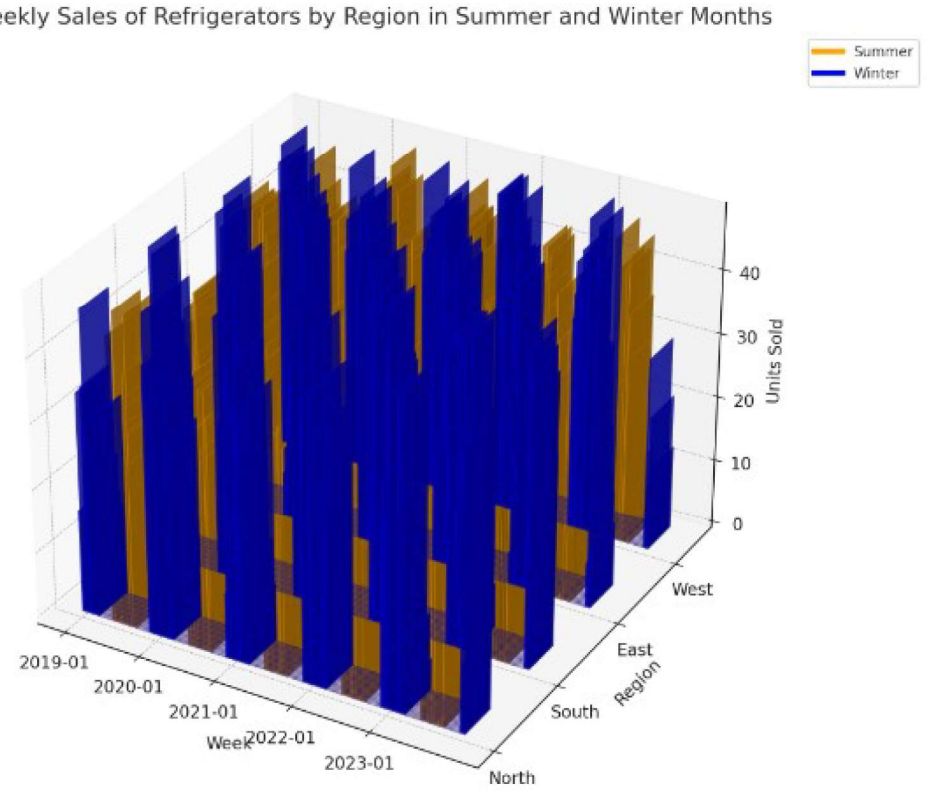


What were the monthly sales (in no of units and value) of the company's products in the 4 metro cities during the last 5 years?



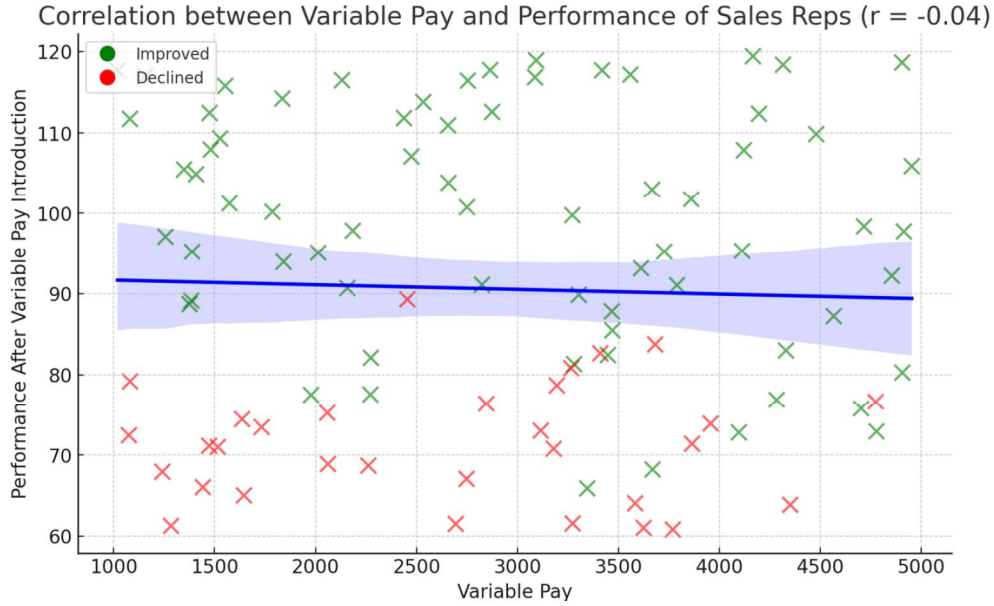
What were the region-wise weekly sales of refrigerators in the 4 summer months and 4 winter months in the last five years?

Weekly Sales Of Refrigerators By Region In Summer And Winter...



What was the correlation between the variable pay package introduced last year and performance of sales reps?

- **Blue Line**: Represents the regression line showing the trend between variable pay and performance after the variable pay introduction.
- **Green Dots**: Indicate sales reps whose performance improved.
- **Red Dots**: Indicate sales reps whose performance declined.




Performance Distribution Before and After Recruitment Process
Performance
95
90
85
80
75
70
65
60
55
Before
After

Some real-world comments

- “We have mountains of data in this company, but we can’t access it.”
- We need to slice and dice the data every which way.”
- “You’ve got to make it easy for business people to get at the data directly.”
- “Just show me what is important.”
- “It drives me crazy to have two people present the same business metrics at a meeting, but with different numbers.”
- “We want people to use information to support more fact-based decision making.

They Need Business Intelligence !

Business Intelligence (BI)

- Business intelligence (BI) refers to the technologies, processes, and practices that collect, integrate, analyze, and present business data to help organizations make better decisions.

Business Intelligence (BI) and Database Types

- Understanding OLTP and OLAP Databases
- Online Transaction Processing (OLTP)
- Online Analytical Processing (OLAP).

Data Warehousing

- A data warehouse is a centralized system that stores and analyzes structured and semi-structured data from multiple sources.

- Databases can be classified in two ways: the one that is used for Transactional & Analytical.

- OLTP vs. OLAP

  We can divide IT systems into transactional (OLTP) and analytical (OLAP). In general we can assume that OLTP systems provide source data to data warehouses, whereas OLAP systems help to analyze it.

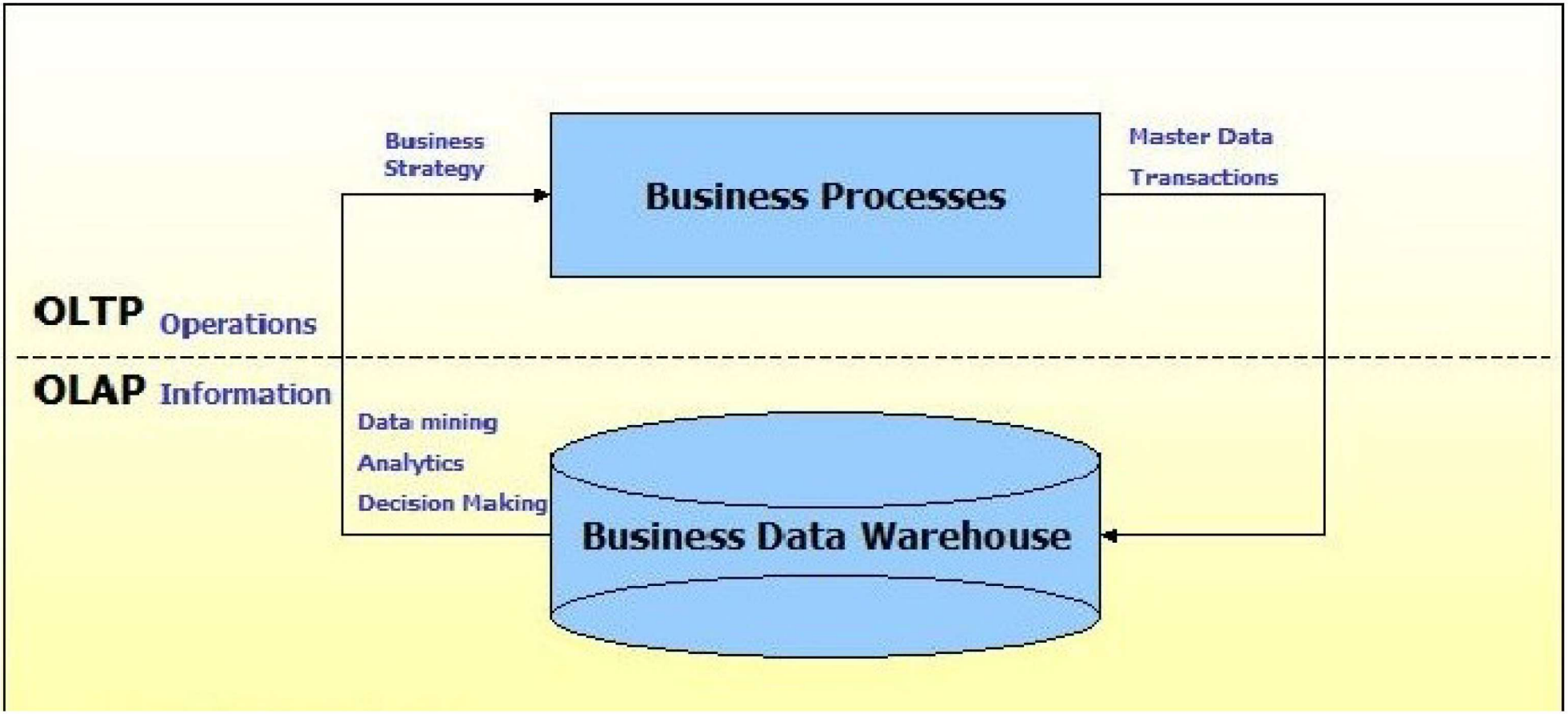
OLTP (Online Transaction Processing) Databases

- Used for many different purposes in business
- Optimized for quickly inserting and updating records
- Relational database structure
- Supports daily operations and transaction processing
- These databases are normalized to reduce the redundancy of the data and increase performance while inserting the data

OLAP (Online Analytical Processing) Databases

- Used to analyze data and support decision-making
- Optimized for quickly retrieving data
- Multi-dimensional database structure
- Created from information in OLTP databases
- Often referred to as Decision Support Systems (DSS)

	OLTP System Online Transaction Processing (Operational System)	OLAP System Online Analytical Processing (Data Warehouse)
Source of data	Operational data; OLTPs are the original source of the data.	Consolidation data; OLAP data comes from the various OLTP Databases
Purpose of data	To control and run fundamental business tasks	To help with planning, problem solving, and decision support
What the data	Reveals a snapshot of ongoing business processes	Multi-dimensional views of various kinds of business activities
Inserts and Updates	Short and fast inserts and updates initiated by end users	Periodic long-running batch jobs refresh the data
Queries	Relatively standardized and simple queries Returning relatively few records	Often complex queries involving aggregations
Processing Speed	Typically very fast	Depends on the amount of data involved; batch data refreshes and complex queries may take many hours; query speed can be improved by creating indexes
Space Requirements	Can be relatively small if historical data is archived	Larger due to the existence of aggregation structures and history data; requires more indexes than OLTP
Database Design	Highly normalized with many tables	Typically de-normalized with fewer tables; use of star and/or snowflake schemas
Backup and Recovery	Backup religiously; operational data is critical to run the business, data loss is likely to entail significant monetary loss and legal liability	Instead of regular backups, some environments may consider simply reloading the OLTP data as a recovery method


Business
Strategy
Business Processes
Master Data
Transactions
OLTP Operations
OLAP Information
Data mining
Analytics
Decision Making
Business Data Warehouse

Decision Support Systems (DSS)

- Also known as Business Intelligence (BI)
- Synthesizing useful knowledge from large data sets
- Integration, summarization, and abstraction of data
- Involves ratios, trends, and allocations
- Comparing data-based generalizations with model-based assumptions

Dimensional Data Model:

- Dimensional data modeling is an analytical approach used in databases and data warehouses for organizing and categorizing facts into dimension tables.

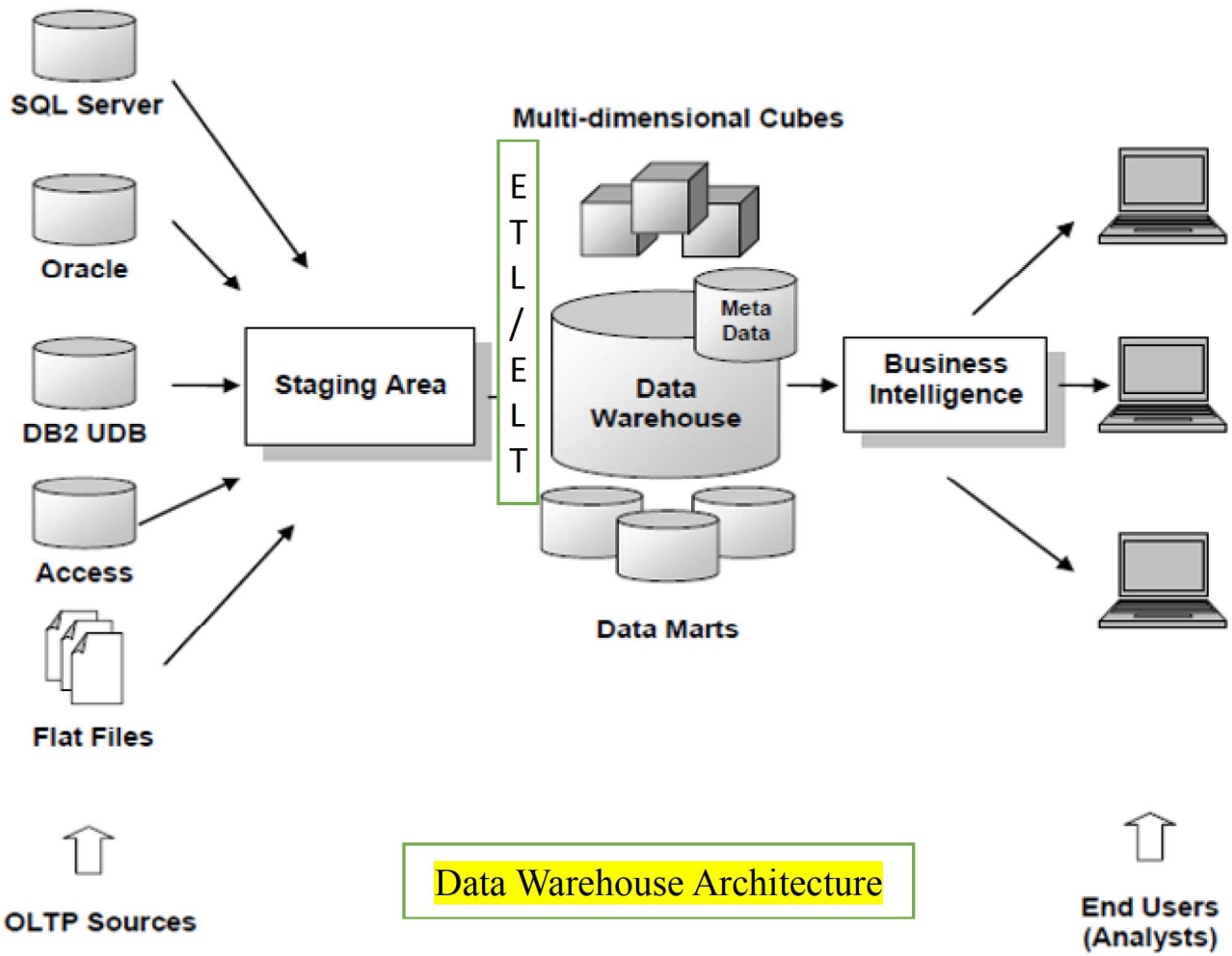
- Dimensional data model is commonly used in data warehousing systems. This section describes this modeling technique, and the two common schema types, star schema and snowflake schema.

Metadata

- Metadata is simply defined as data about data.
- The data that are used to represent other data is known as metadata.
- For example, the index of a book serves as a metadata for the contents in the book.
- In other words, we can say that metadata is the summarized data that leads us to the detailed data.

In terms of data warehouse, we can define metadata as following −

- Metadata is a road-map to data warehouse.
- Metadata in data warehouse defines the warehouse objects.
- Metadata acts as a directory. This directory helps the decision support system to locate the contents of a data warehouse.



Data Warehouse Architecture

DATA WAREHOUSE LIFE CYCLE

Data Cleansing

- Data Cleansing is the process of cleansing or validating the data brought from multiple sources.
- The sources data may be invalid for reasons more than one. The data might also have become invalid because of improper manual feeding done at the OLTP level.
- Organizational policies change with the time and hence their business logic.
- Data from the OLTP source becomes invalid if it no longer meets the new business logic and policies.

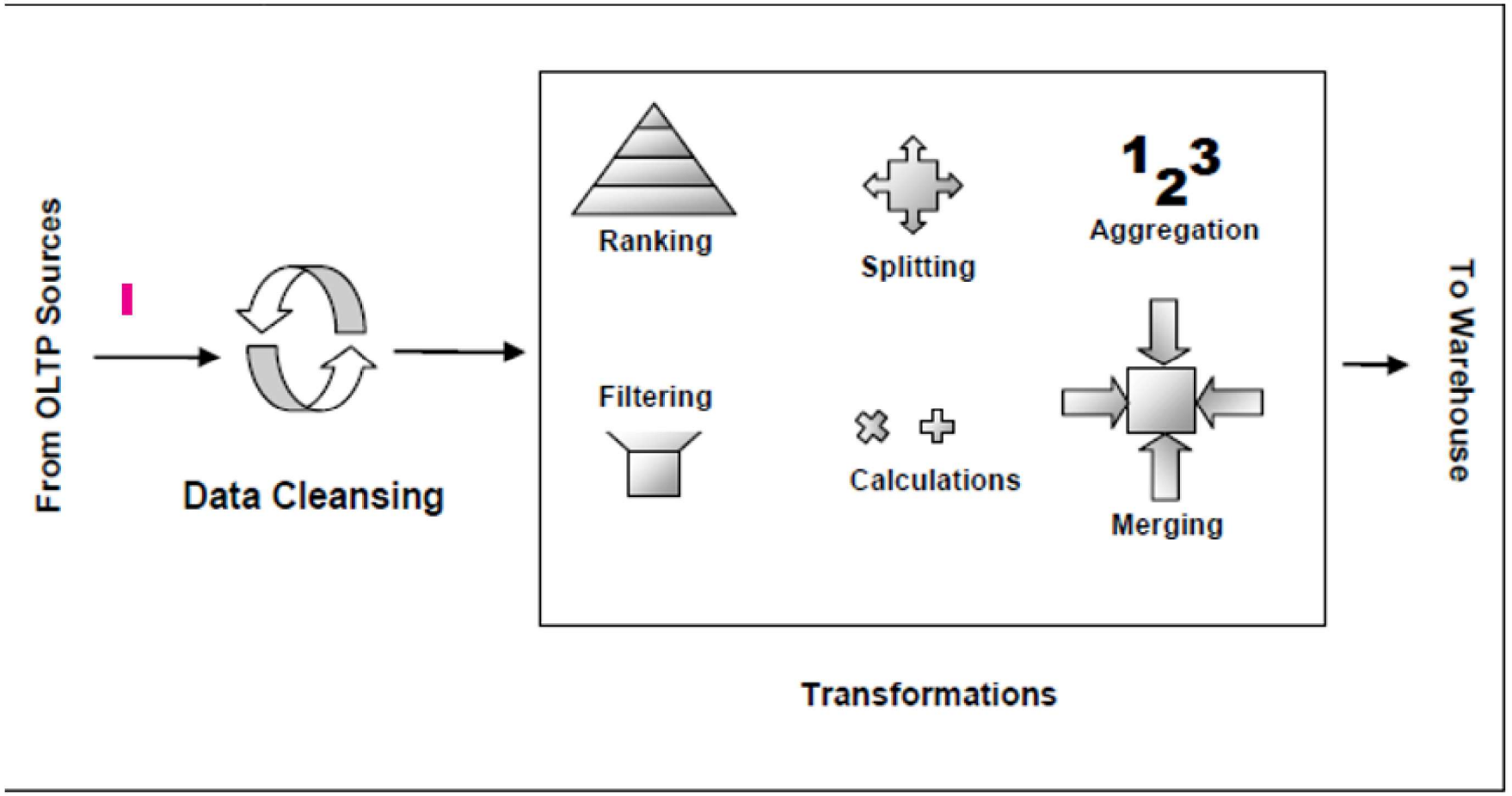
Extracting, Transforming and Loading (ETL)

- Extracting, Transforming and Loading (ETL) is the process of reading (extracting) data from heterogeneous sources and transform them so that the discrete data from different sources gets integrated and then loading into the target Data Warehouse.

- During this process, data from multiple tables may be merged in to one and/or data from single table may be routed into multiple tables and/or can be sorted, grouped, filtered and so on…

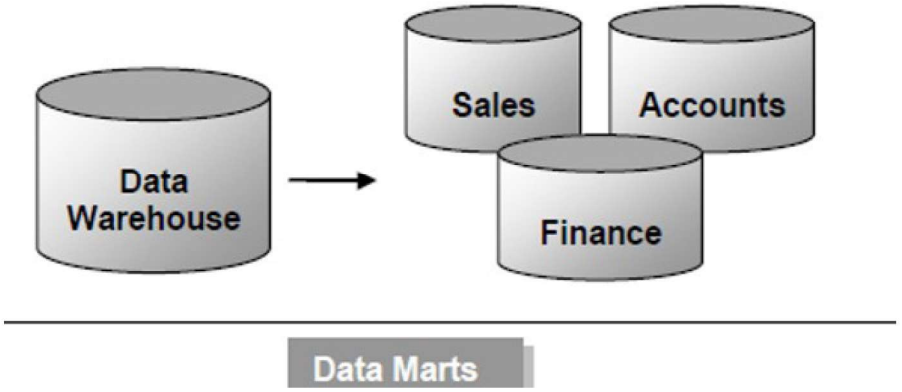
- These operations are done at a special dedicated area wherein all the data from all the sources are first dumped off, well known as **Staging Area.**

- Software **like Informatica Power Center / IBM Data Stage, Oracle Data Integrator, SAS Data Management, AWS Glue, Azure Data Factory** etc., are used for these operations.


From OLTP Sources
Data Cleansing
Ranking
Splitting
Aggregation
Filtering
Calculations
Merging
To Warehouse
Transformations

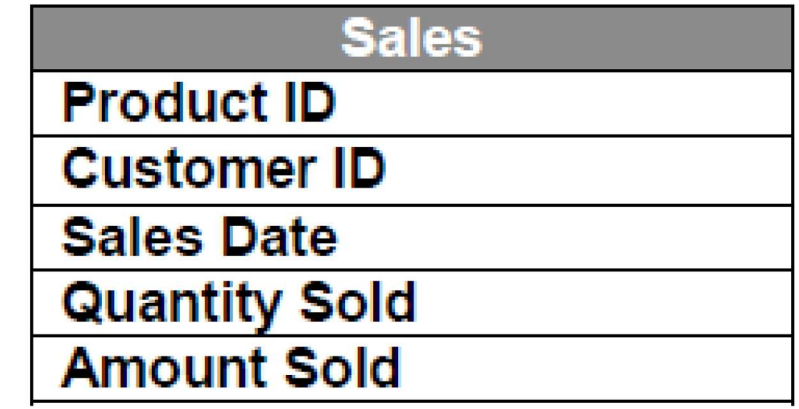
Data Marts:

- Data Marts contain the summarized data of the ware houses and are referred as
- High Performance Query Structures. They consist of Materialized Views and Special Indexes.
- In some businesses these data marts may be maintained within the ware houses whereas, in some other scenarios they may be maintained apart from the data warehouses.



Facts or Fact Tables

- A fact table is the primary table in a dimensional model where the numerical performance measurements of the business are stored.
- A row in a fact table corresponds to a measurement.
- All the measurements in a fact table must be at the same grain. () the same grain means that each row in the fact table represents a consistent level of detail.
- Fact tables generally have less number of columns and more number of rows.
- This table contains foreign keys to link to dimension tables.



Fact Table

Dimension tables

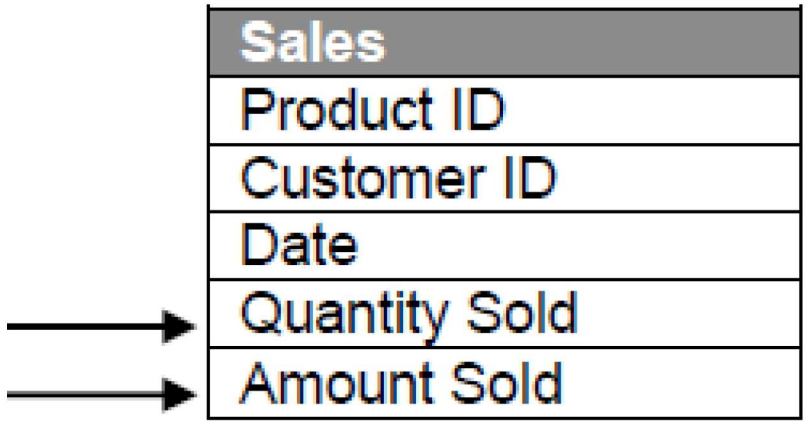
- Dimension tables contain textual descriptors of business. Dimension table are integral companions to a fact table.
- It is not uncommon for a dimension table to have 100 to 1000 attributes.
- Dimension tables generally have more number of columns and less number of rows.
- Each dimension table is linked to the fact table through a foreign key.

Products
Product ID
Product Name
Product Description
Category
Category Description
Sub Category
Sub Category Description
List Price
Minimum Price
Model Number
Unit of Measure
Supplier Name

Dimension Table

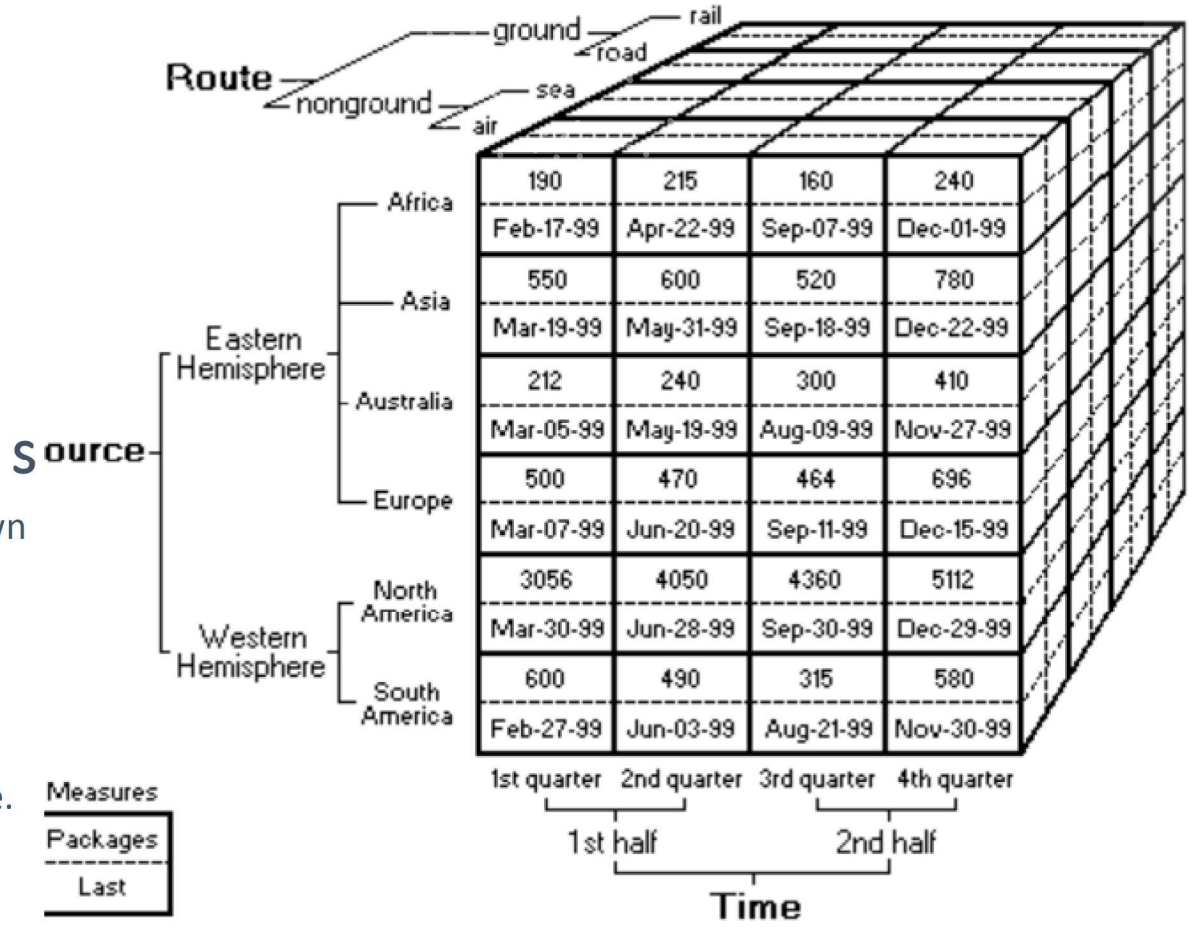
Measures

- Measures are the numbers on which you make your comparisons.
- It includes members such as: cost, profit, or taxes.
- Measures are the measurable quantities upon which the business is measured.
- They are always numeric quantities.
- They are the ones on which we perform aggregations.



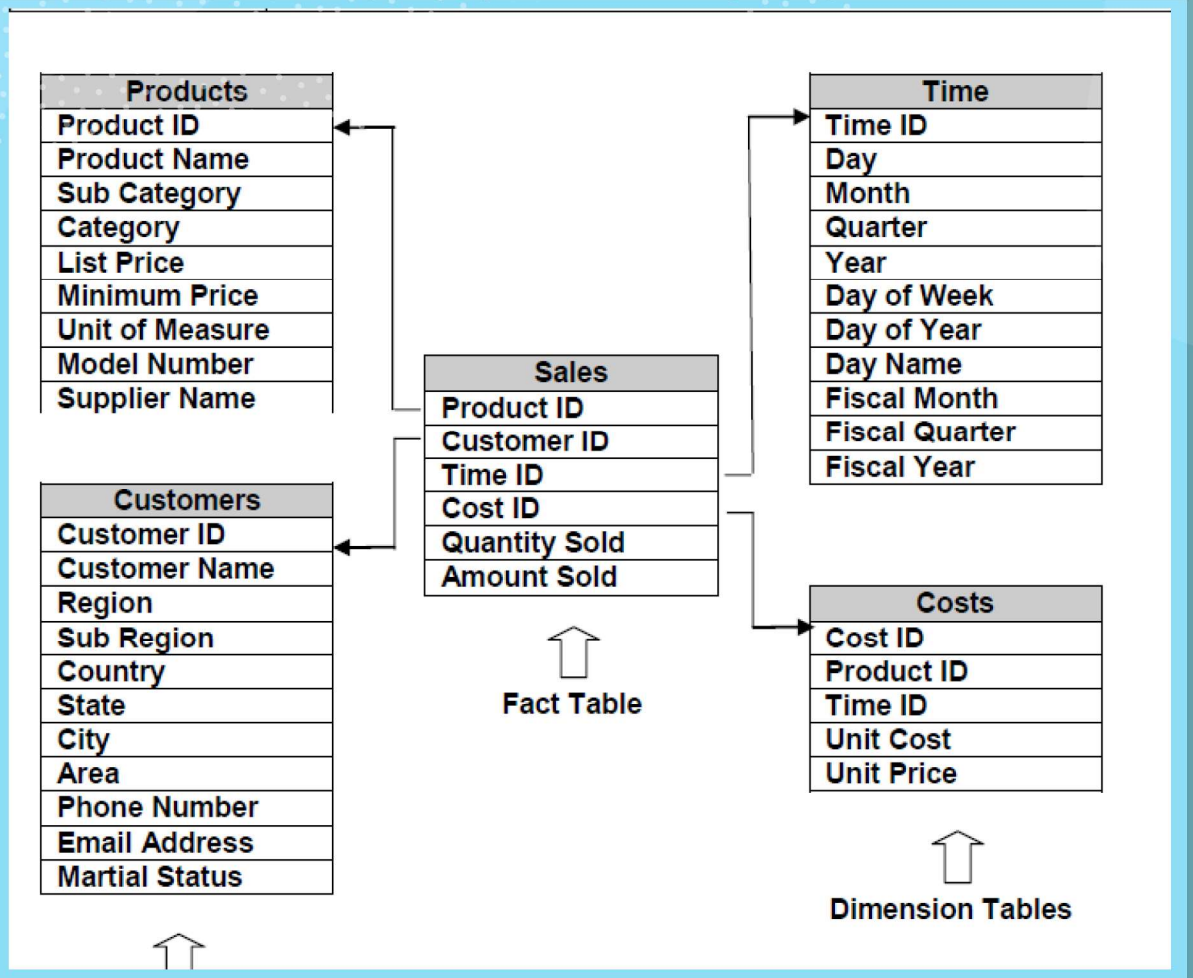
Multi-dimensional DM or cube

- A multidimensional cube, also known as an OLAP cube, is a logical structure that stores pre-calculated data for analytical purposes.
- It's a multidimensional representation of data that's designed to make it easy to analyze and retrieve.



Star Schema

- A **Star Schema** is a type of database schema optimized for data warehousing and business intelligence, where a central fact table is linked to multiple dimension tables, forming a star-like structure.



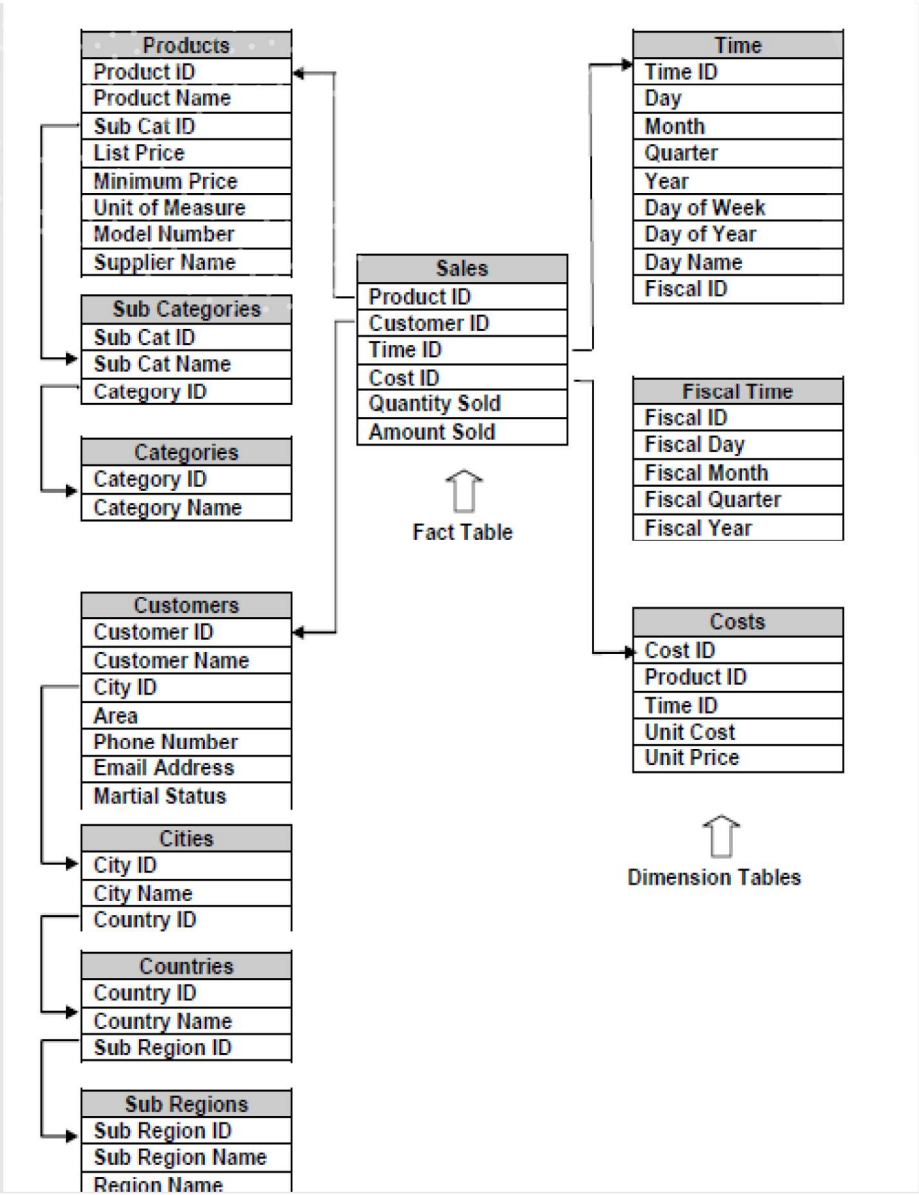
- The fact table in a star schema contains the measures or metrics that are of interest to the user or organization.
- The dimension tables in a star schema contain the descriptive attributes of the measures in the fact table.
- In a star schema, each dimension table is joined to the fact table through a foreign key relationship.
- The star schema is a popular data modeling technique in data warehousing because it is easy to understand and query.

- A star schema is denormalized, Star schema is designed to make queries simple and fast.

- Star schema is designed for fast query performance.

- The star schema is easy to understand and interpret, even for non-technical users.

A **Snowflake Schema** is a type of database schema used in data warehousing, where the dimension tables are normalized into multiple related tables, forming a snowflake-like structure.



- A snowflake schema is a multi-dimensional data model that is an extension of a star schema, where dimension tables are broken down into subdimensions.

- Snowflake schema is preferred when a data warehouse is most of the time used as a source for one more rather high-end data warehouse than for direct analysis.

- A snowflake schema consists of a single fact table and multiple dimension tables.

- Dimension tables in star schema are de-normalized, while those in a snowflake schema are normalized.

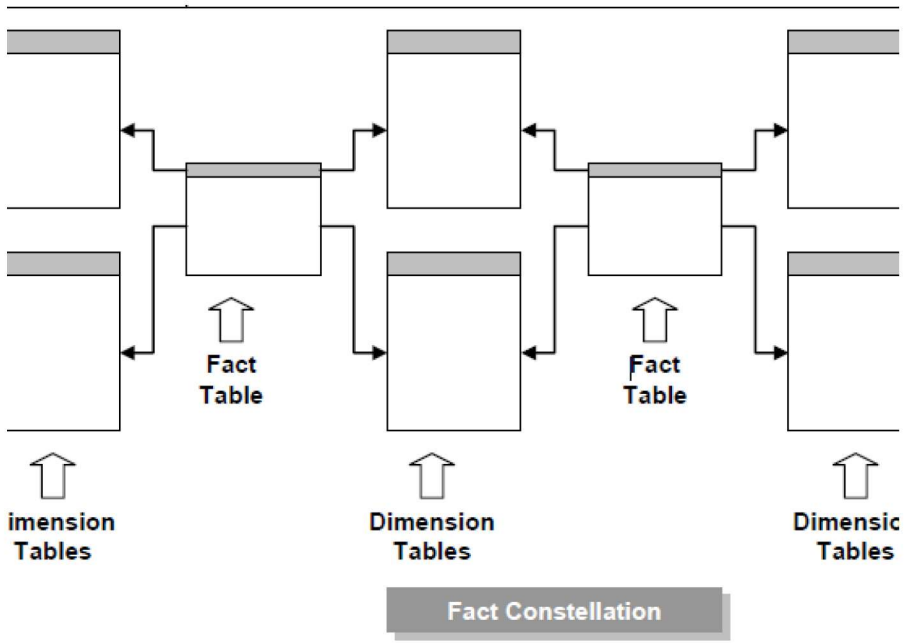
- The major difference between a Star schema and a Snowflake schema is that Star schema highly de-normalized, whereas, Snowflake schema is partially normalized.

Star Schema and Snowflake Schema

Parameters	Star Schema	Snowflake Schema
Definition and Meaning	A star schema contains both dimension tables and fact tables in it.	A snowflake schema contains all three- dimension tables, fact tables, and sub-dimension tables.
Type of Model	It is a top-down model type.	It is a bottom-up model type.
Space Occupied	It makes use of more allotted space.	It makes use of less allotted space.
Time Taken for Queries	With the Star Schema, the process of execution of queries takes less time.	With the Snowflake Schema, the process of execution of queries takes more time.
Use of Normalization	The Star Schema does not make use of normalization.	The Snowflake Schema makes use of both Denormalization as well as Normalization.
Complexity of Design	The design of a Star Schema is very simple.	The designing of a Snowflake Schema is very complex.
Query Complexity	It is very low in the case of a Star Schema.	It is comparatively much higher in the case of a Snowflake Schema.
Complexity of Understanding	It is very easy to understand a Star Schema.	It is comparatively more difficult to understand a Snowflake Schema.
Total Number of Foreign Keys	The total number of foreign keys is less in the case of a Star Schema.	The total number of foreign keys is more in the case of a Snowflake Schema.
Data Redundancy	Data redundancy is comparatively higher in Star Schema.	Data redundancy is comparatively lower in Snowflake Schema.

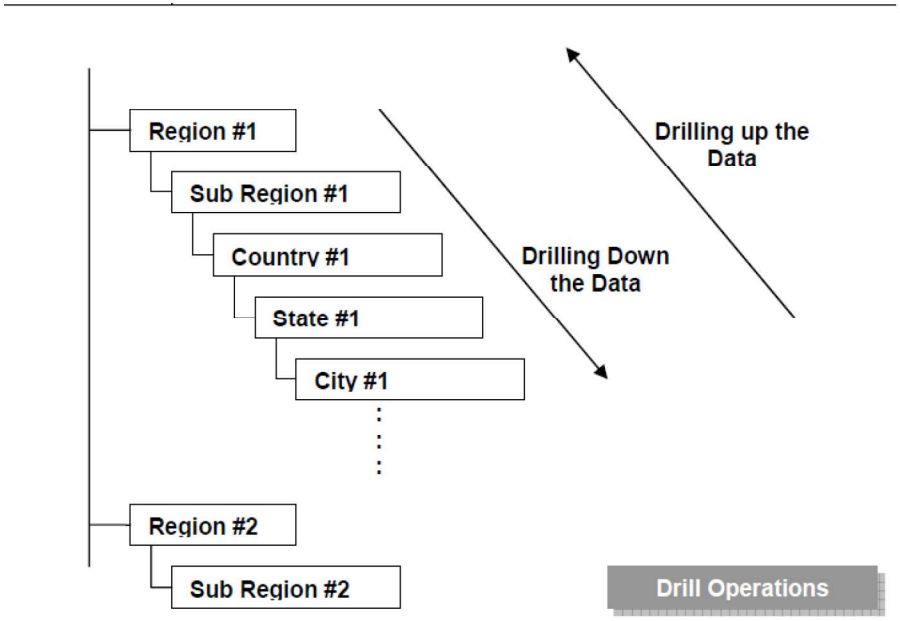
Fact Constellation / Multi-Star schema

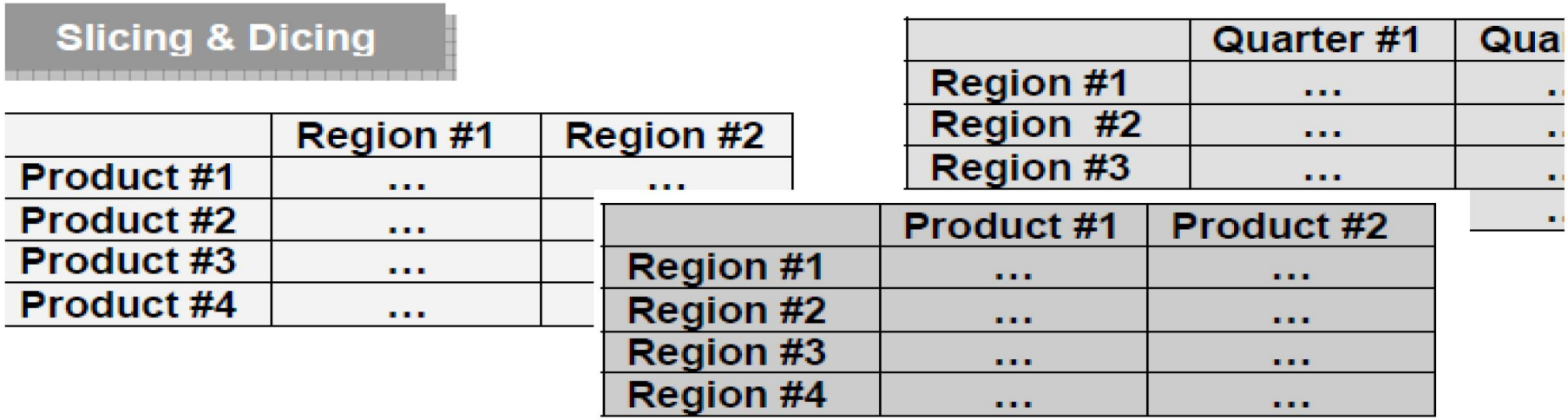
- Most often, there may be a need to have more than one Fact Table, and these are called Fact Constellations.
- A Fact Constellation is a kind of schema where we have more than one Fact Table sharing among them some Dimension Tables.
- It is also called a **Galaxy schema**. If such a schema is highly de-normalized it is also called as **Multi-Star schema**.



OLAP operations:
Drill up & Drill Down

- Drilling is the term used to navigate through the warehouse data through a given dimension. We can say we drill the sales data by region.
- In that case, viewing region wise sales report and then sub region wise sales report for a given region and then moving forward to country wise sales and then to state wise sales data is what is referred to as Drilling Down the data. Drilling up is navigating back.



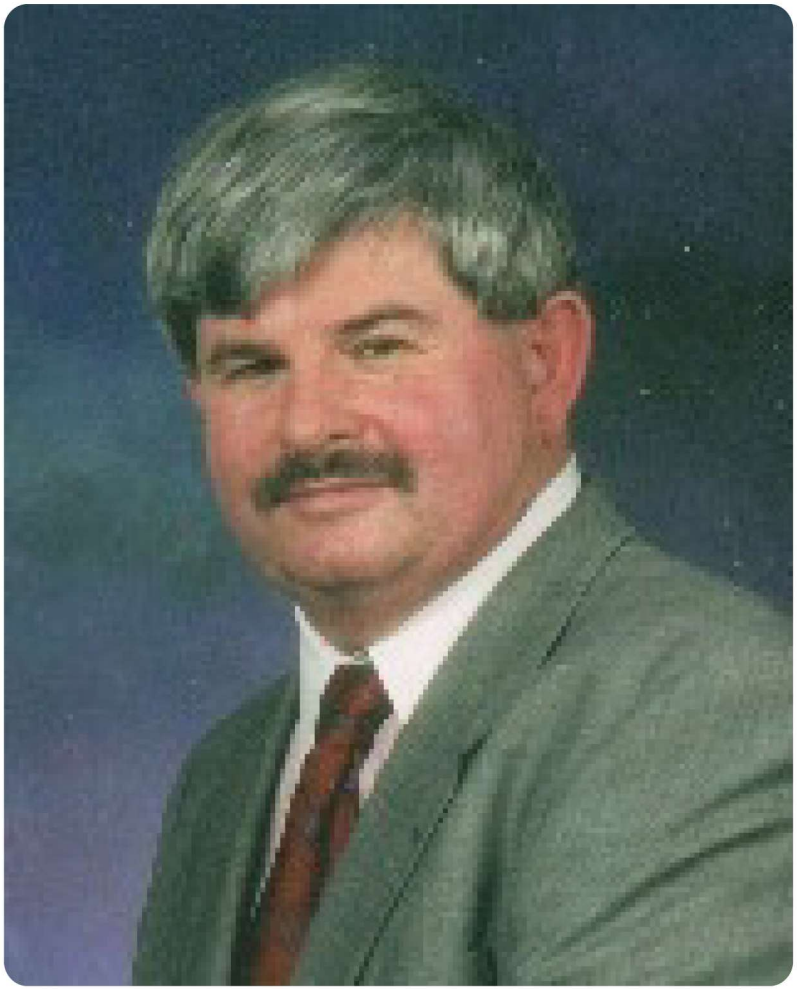


OLAP operations: Slicing & Dicing

- Slicing & dicing the data means analyzing the same data in different fashions and groups.
- Let us consider a sales report, which measures my sales by the number of items sold region wise-time wise-product category wise.
- Changing the order and way we view the data within these given dimensions is what is known as slicing & dicing.

Ralph Kimball

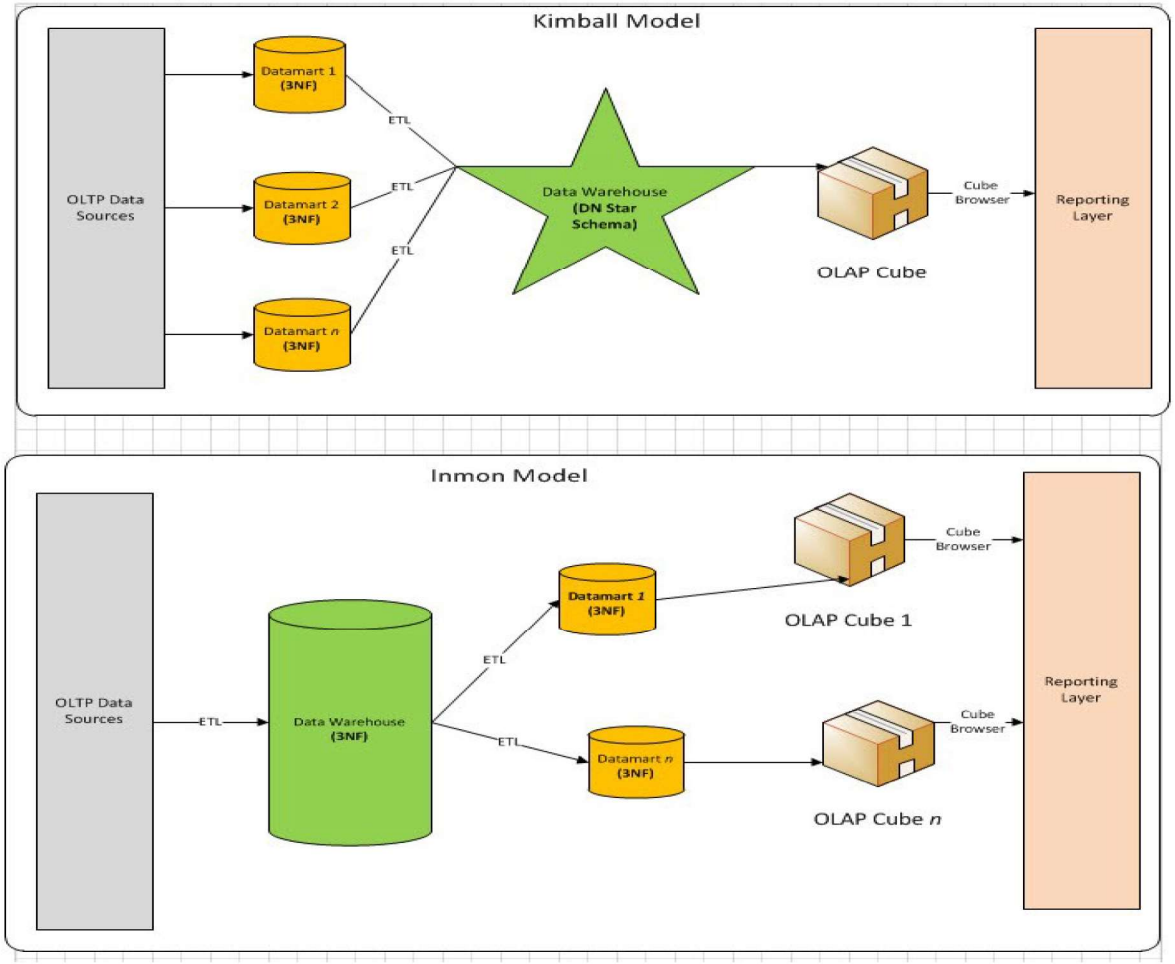
- Contrast to Bill Inmon approach, Ralph Kimball recommends building the data warehouse that follows the bottom-up approach. In Kimball's philosophy, it first starts with mission-critical data marts that serve the analytic needs of departments. Then it is integrating these data marts for data consistency through a so-called information bus. Kimball makes uses of the dimensional model to address the needs of departments in various areas within the enterprise.



Bill Inmon

- Bill Inmon recommends building a data warehouse that follows the top-down approach.
- In Inmon's philosophy, it is starting with building a big centralized enterprise data warehouse where all available data from transaction systems are consolidated into a subject-oriented, integrated, time-variant, and non-volatile collection of data that supports decision making. then data marts are built for the analytic needs of departments.

Parameters	Kimball	Inmon
Introduced by	Introduced by Ralph Kimball.	Introduced by Bill Inmon.
Approach	It has a Bottom-Up Approach for implementation.	It has Top-Down Approach for implementation.
Data Integration	It focuses on Individual business areas.	It focuses on Enterprise-wide areas.
Building Time	It is efficient and takes less time.	It is complex and consumes a lot of time.
Cost	It has iterative steps and is cost-effective.	Initial cost is huge and the development cost is low.
Skills Required	It does not need such skills but a generic team will do the job.	It needs specialized skills to make work.
Maintenance	Here maintenance is difficult.	Here maintenance is easy.
Data Model	It prefers data to be in the De-normalized model.	It prefers data to be in a normalized model.
Data Store Systems	In this, source systems are highly stable.	In this, source systems have a high rate of change.


Kimball Model
OLTP Data Sources
Datamart 1 (3NF)
Datamart 2 (3NF)
Datamart n (3NF)
ETL
ETL
ETL
Data Warehouse (DN Star Schema)
OLAP Cube
Cube Browser
Reporting Layer
Inmon Model
OLTP Data Sources
ETL
Data Warehouse (3NF)
ETL
ETL
Datamart 1 (3NF)
Datamart n (3NF)
OLAP Cube 1
OLAP Cube n
Cube Browser
Cube Browser
Reporting Layer

Data Modelling

- Data modelling is the process of creating a visual representation of a complex data system.

- This model illustrates how data is structured, related, and manipulated within a database or information system.

Types of Data Models

- Conceptual Data Model
- Logical Data Model
- Physical Data Model

- Conceptual Data Model: High-level view of organizational data, focusing on business entities and relationships.

- Logical Data Model: More detailed, including attributes and keys, without considering how they will be physically. implemented.

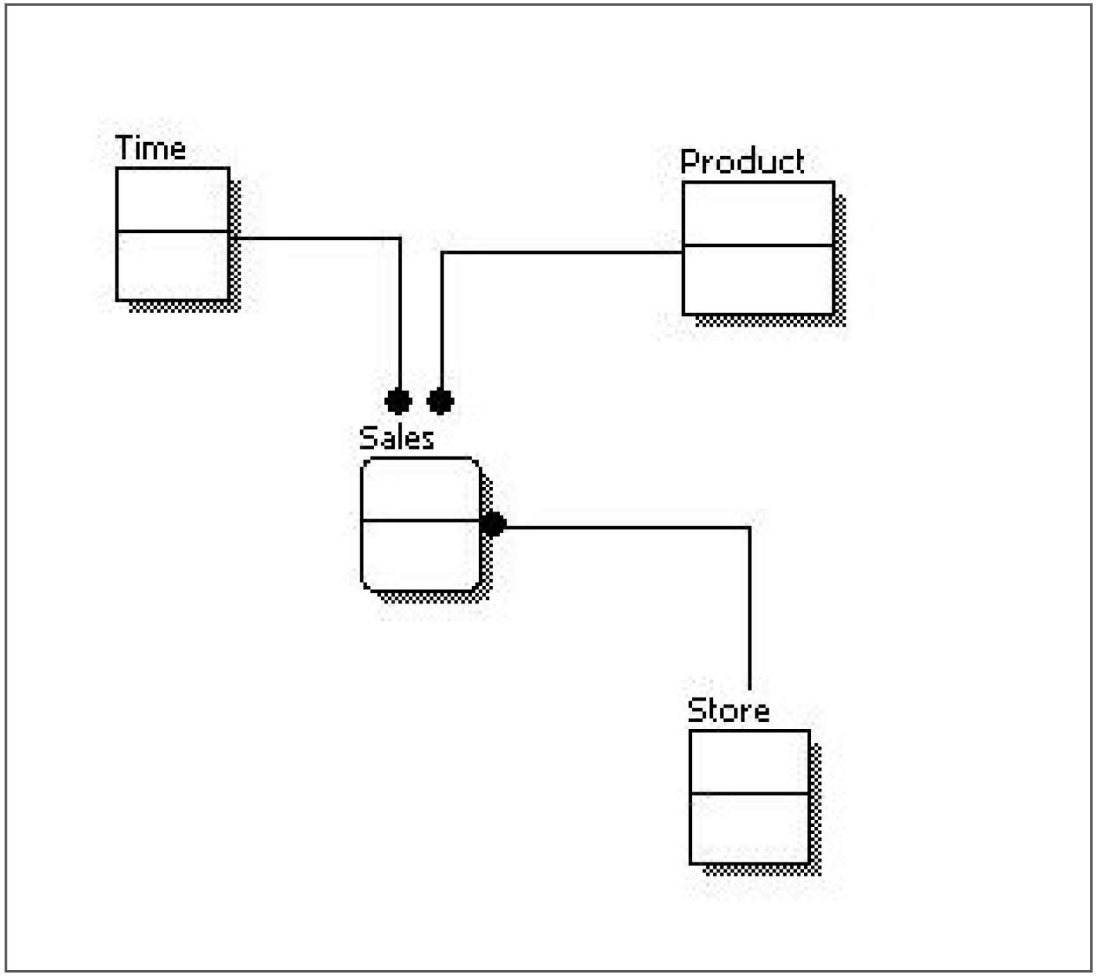
- Physical Data Model: Detailed representation of data, including how it will be stored in the database.

Conceptual Data Model

- A conceptual data model identifies the highest-level relationships between the different entities. Features of conceptual data model include:
- Includes the important entities and the relationships among them.
- No attribute is specified.
- No primary key is specified.
- The figure below is an example of a conceptual data model.

Conceptual Data Model

- Conceptual data model is the entities that describe the data and the relationships between those entities.

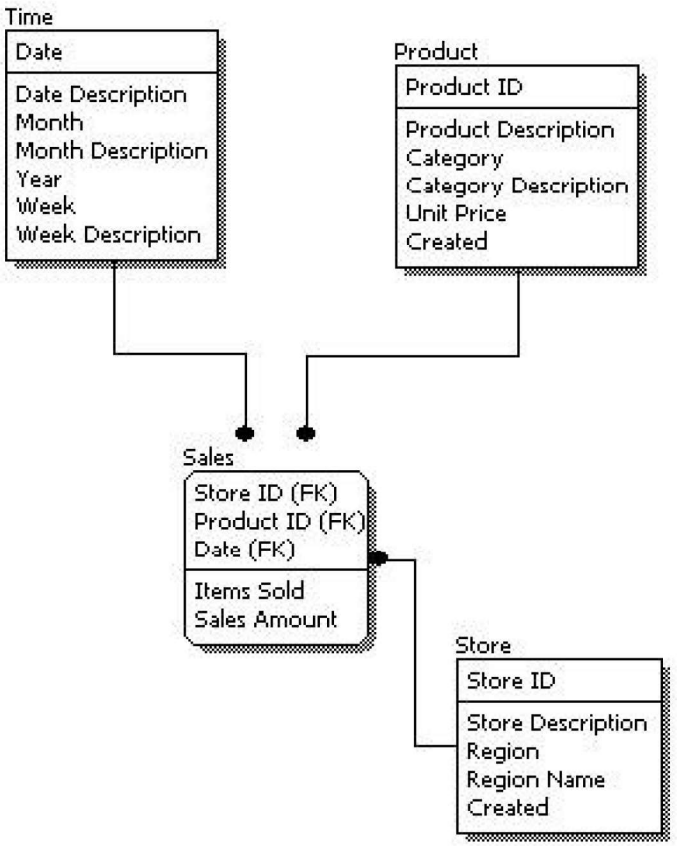


Logical Data Model

- A logical data model describes the data in as much detail as possible, without regard to how they will be physical implemented in the database.
- Includes all entities and relationships among them.
- All attributes for each entity are specified.
- The primary key for each entity is specified.
- Foreign keys (keys identifying the relationship between different entities) are specified.
- Normalization occurs at this level.

The steps for designing the logical data model are as follows:

1. Specify primary keys for all entities.
2. Find the relationships between different entities.
3. Find all attributes for each entity.
4. Resolve many-to-many relationships.
5. Normalization.



Comparing the logical data model shown above with the conceptual data model diagram, we see the main differences between the two:

In a logical data model, primary keys are present, whereas in a conceptual data model, no primary key is present.

In a logical data model, all attributes are specified within an entity. No attributes are specified in a conceptual data model.

Relationships between entities are specified using primary keys and foreign keys in a logical data model.

In a conceptual data model, the relationships are simply stated, not specified, so we simply know that two entities are related, but we do not specify what attributes are used for this relationship.

Physical Data Model

- Physical data model represents how the model will be built in the database.

- A physical database model shows all table structures, including column name, column data type, column constraints, primary key, foreign key, and relationships between tables.

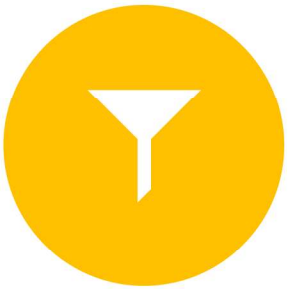
Features of a physical data model include:

- Specification all tables and columns.
- Foreign keys are used to identify relationships between tables.
- Denormalization may occur based on user requirements.
- Physical considerations may cause the physical data model to be quite different from the logical data model.
- Physical data model will be different for different RDBMS. For example, data type for a column may be different between MySQL and SQL Server.

The steps for physical data model design are as follows:

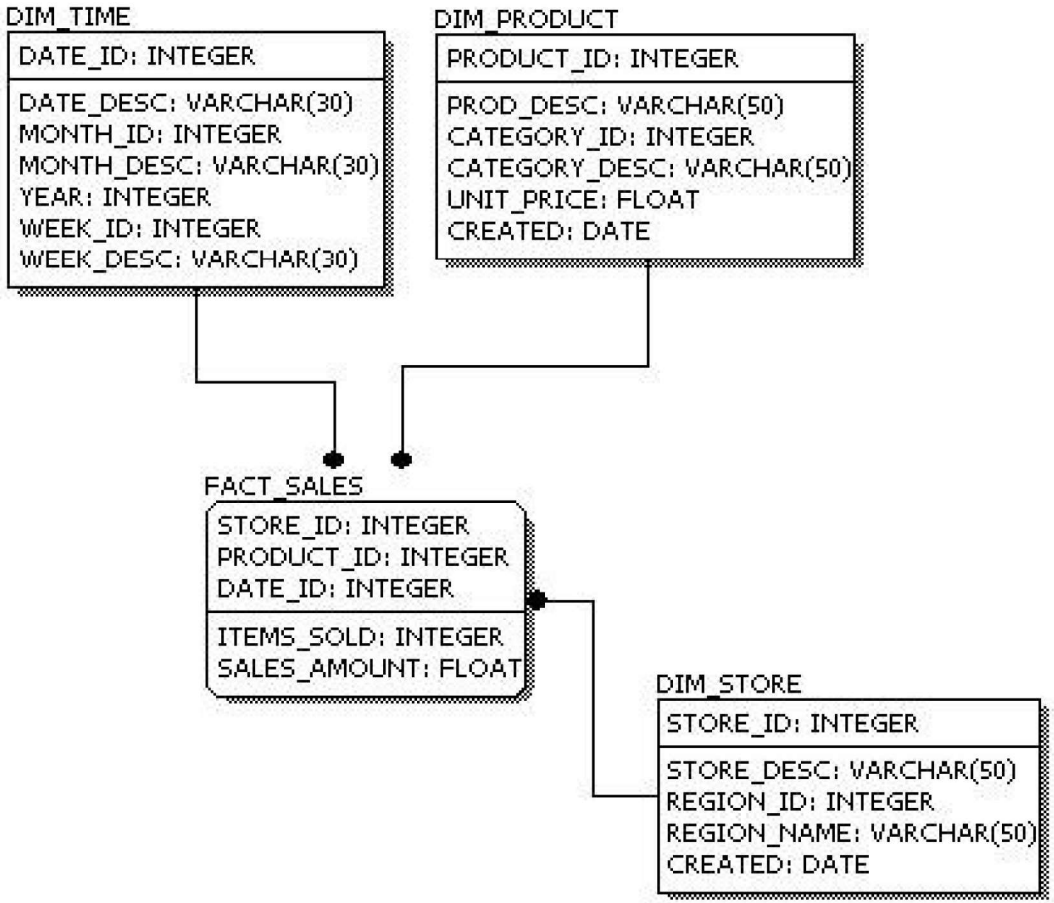
CONVERT ENTITIES INTO TABLES.

CONVERT RELATIONSHIPS INTO FOREIGN KEYS.

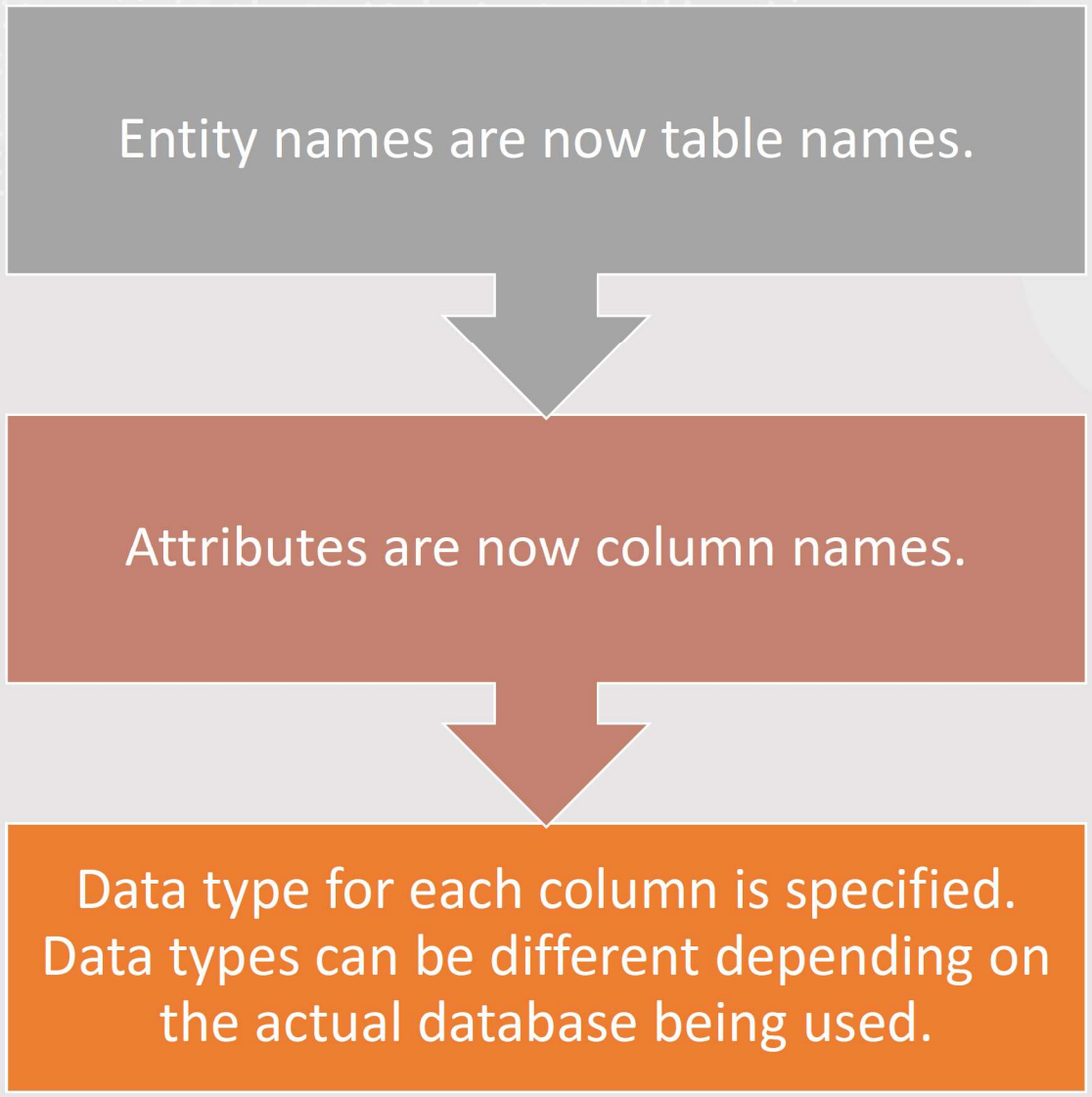


CONVERT ATTRIBUTES INTO COLUMNS.

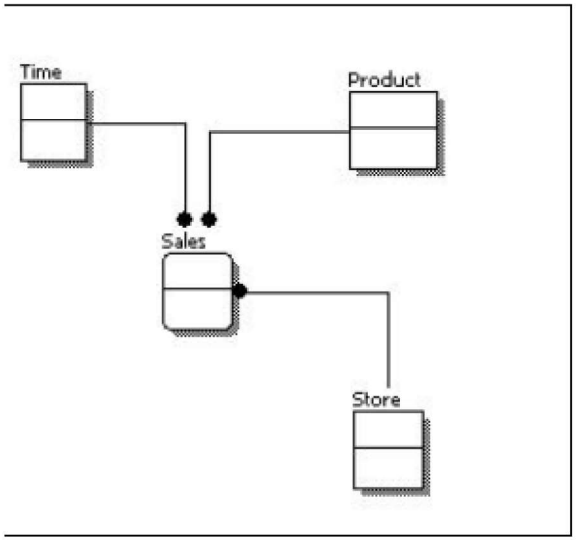
MODIFY THE PHYSICAL DATA MODEL BASED ON PHYSICAL CONSTRAINTS / REQUIREMENTS.


DIM_TIME
DATE_ID: INTEGER
DATE_DESC: VARCHAR(30)
MONTH_ID: INTEGER
MONTH_DESC: VARCHAR(30)
YEAR: INTEGER
WEEK_ID: INTEGER
WEEK_DESC: VARCHAR(30)
DIM_PRODUCT
PRODUCT_ID: INTEGER
PROD_DESC: VARCHAR(50)
CATEGORY_ID: INTEGER
CATEGORY_DESC: VARCHAR(50)
UNIT_PRICE: FLOAT
CREATED: DATE
FACT_SALES
STORE_ID: INTEGER
PRODUCT_ID: INTEGER
DATE_ID: INTEGER
ITEMS_SOLD: INTEGER
SALES_AMOUNT: FLOAT
DIM_STORE
STORE_ID: INTEGER
STORE_DESC: VARCHAR(50)
REGION_ID: INTEGER
REGION_NAME: VARCHAR(50)
CREATED: DATE

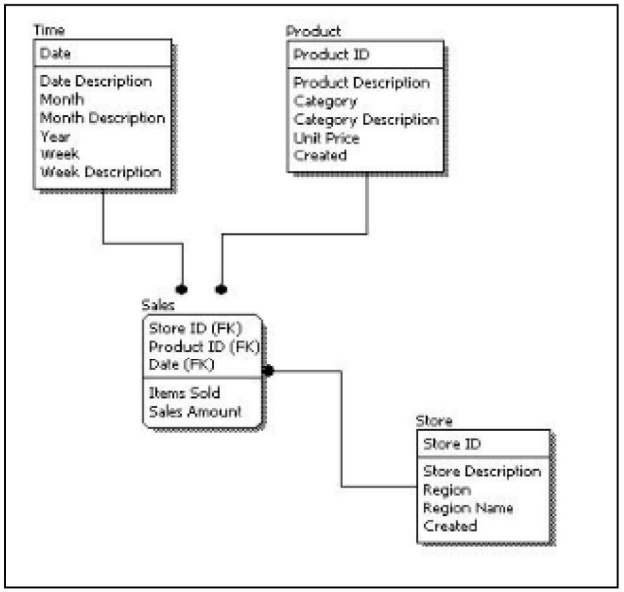
Comparing the physical data model shown above with the logical data model diagram, we see the main differences between the two:



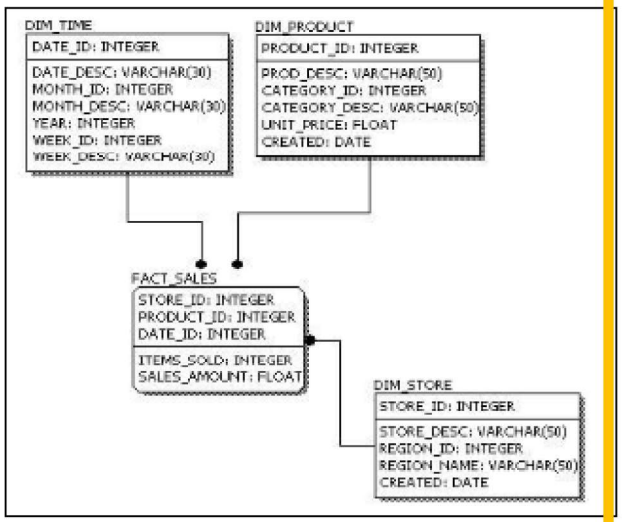
Conceptual Model Design



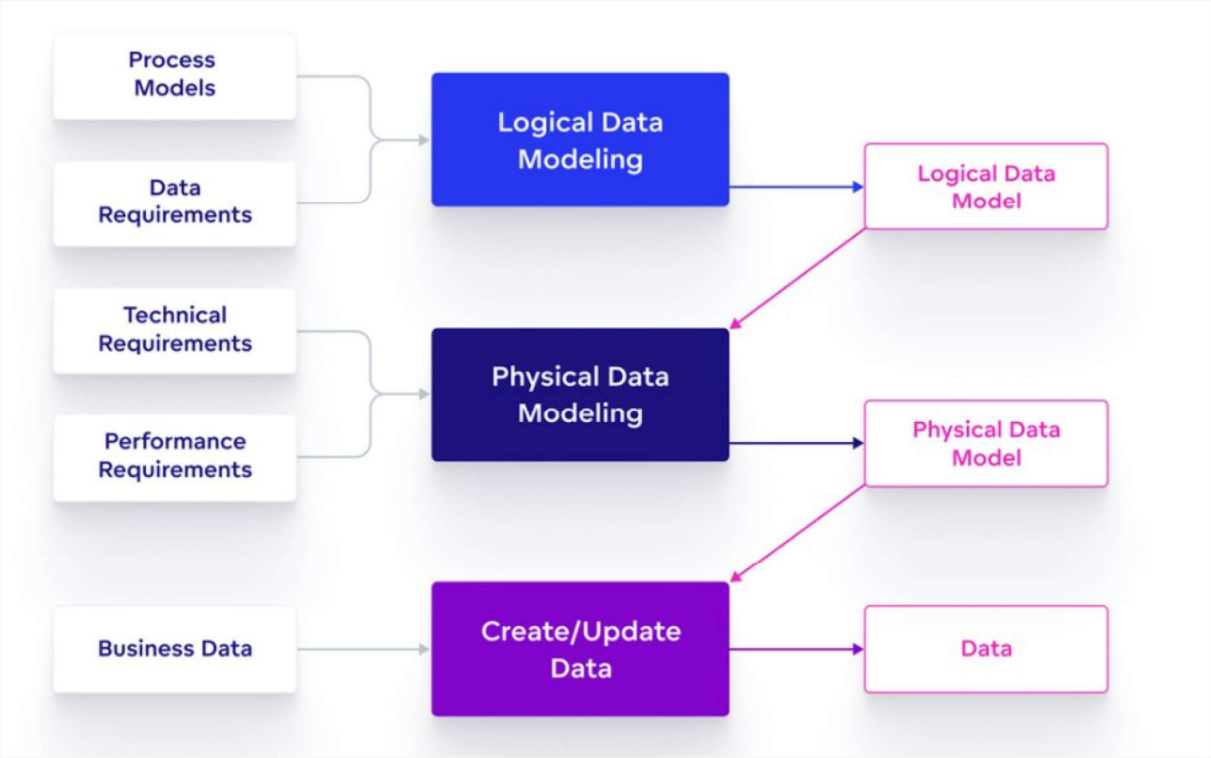
Logical Model Design



Physical Model Design



Feature	Conceptual	Logical	Physical
Entity Names	✓	✓	
Entity Relationships	✓	✓	
Attributes		✓	
Primary Keys		✓	✓
Foreign Keys		✓	✓
Table Names			✓
Column Names			✓
Column Data Types			✓



- The end result is a physical data model that is ready to store business data and generate a multiple queries and reports.

Surrogate Key

- A surrogate key is a unique identifier for each record in a data warehouse that is not derived from application data.
- It is typically a sequential number (e.g., an auto-incremented integer) used as the primary key for a dimension table.
- Surrogate keys are used to simplify joins and improve query performance, and they remain consistent even if the source data changes.

Critical Columns

- The huge collection of data within a warehouse is useful for analysis only if the data in it is consistent. However, there exists some information which when updated in a warehouse makes the data in a warehouse inconsistent. The columns that contain such vital information are identified to be as Critical Columns.
- Critical columns exist only at the OLTP source.
- In a data warehouse, a **critical column** refers to an essential column that plays a pivotal role in the accuracy, integrity, and usability of the data.
- These columns are crucial for business intelligence, reporting, and analytical tasks. Critical columns typically include primary keys, foreign keys, and significant business metrics or dimensions.

Examples of Critical Columns

- Primary Keys: Unique identifiers for records in a dimension or fact table.

- Example: CustomerID in a Customer Dimension table.
- Foreign Keys: Columns used to join fact tables with dimension tables.

- Example: ProductID in a Sales Fact table linking to the Product Dimension table.
- Business Metrics: Important measures used for analysis.

Example Table with Critical Columns

- **Customer Dimension Table**:

CustomerID (Primary Key)	CustomerName	DateOfBirth	Region
1	John Doe	1985-04-23	North America
2	Jane Smith	1990-07-15	Europe

- **Sales Fact Table**:

SalesID	CustomerID (Foreign Key)	ProductID	SaleDate	SalesAmount
1001	1	PROD001	2024-01-10	150.00
1002	2	PROD002	2024-01-11	200.00

Surrogate Keys

- Having surrogate key is the solution for the critical column.
- A surrogate key is a unique identifier for each record in a data warehouse that is not derived from application data. It is typically a sequential number (e.g., an auto-incremented integer) used as the primary key for a dimension table.

Example

- Consider a **Customer Dimension** table in a data warehouse:

SurrogateKey	CustomerID	CustomerName	DateOfBirth	Address
1	CUST001	John Doe	1985-04-23	123 Elm St, Cityville
2	CUST002	Jane Smith	1990-07-15	456 Oak St, Townsville

- Surrogate Key is the surrogate key and is unique for each row.
- CustomerID is a natural key from the source system.
- Surrogate Key remains stable and unique even if CustomerID or other attributes change.

Benefits of Surrogate Keys

1. **Consistency**: Surrogate keys remain consistent even if the natural keys change.
2. **Performance**: Joins based on integer surrogate keys are faster than joins on text-based natural keys.
3. **Simplicity**: Simplifies handling of slowly changing dimensions by providing a stable unique identifier.
4. **Uniqueness**: Ensures uniqueness in the data warehouse independently of the source system's key.

Usage in Fact Tables

- In a fact table, surrogate keys from dimension tables are used as foreign keys:
- Fact Table: SalesFact

SalesID	SurrogateKey_Customer	ProductID	SaleDate	SalesAmount
1001	1	PROD001	2024-01-10	150.00
1002	2	PROD002	2024-01-11	200.00

In this example, SurrogateKey_Customer refers to the surrogate key in the Customer Dimension table. This ensures that even if customer details change, the reference in the fact table remains consistent.

Introduction to Enterprise Data Management (EDM)

- Enterprise Data Management (EDM) is the process of managing data as a valuable resource to ensure its quality, accessibility, and security across an enterprise.

- Key Components of Enterprise Data Management
  - Data Governance
    - Establishing policies and procedures to manage data assets
    - Ensuring compliance with regulations and standards
  - Data Integration
    - Consolidating data from different sources
    - Ensuring data consistency and accessibility

- Data Quality Management
  - Monitoring and improving data accuracy, completeness, and reliability
- Master Data Management
  - Creating a single, authoritative source for critical business data
  - Ensuring data consistency across the organization
- Metadata Management
  - Managing data about data
  - Providing context and meaning to data assets

Master Data Management

- Master Data Management (MDM) refers to the process of creating and managing data that an organization must have as a single master copy, called the master data.
- Usually, master data can include customers, vendors, employees, and products, but can differ by different industries and even different companies within the same industry.
- MDM is the practice of defining and managing the critical data of an organization to provide, with data integration, a single point of reference.
- The main purpose is to ensure consistency, accuracy, and accountability of the organization's shared master data.

Key Components :MDM

- Data Governance: Establishes policies, procedures, and standards for managing data.
- Data Integration: Consolidates data from various sources to create a unified view.
- Data Quality: Ensures the accuracy, completeness, and reliability of data.
- Data Stewardship: Assigns roles and responsibilities for managing data.

Types of Master Data:

- Customer Data: Information about customers and their interactions.
- Product Data: Details about products and services.
- Location Data: Geographic data related to locations.
- Employee Data: Information about employees and organizational structure.

Benefits: MDM

- Improved Data Quality: Reduces errors and inconsistencies in data.
- Enhanced Decision-Making: Provides a reliable basis for strategic decisions.
- Operational Efficiency: Streamlines processes by ensuring data consistency across systems.
- Regulatory Compliance: Helps in meeting data-related regulatory requirements.

Challenges: MDM

- Data Silos: Integrating data from disparate sources.
- Change Management: Managing the cultural change associated with new data governance practices.
- Data Security: Ensuring the protection of sensitive data.
- Scalability: Managing growing volumes of data.

Best Practices :MDM

- Establish Clear Governance: Define roles, responsibilities, and policies.
- Engage Stakeholders: Involve all relevant departments and users in the MDM process.
- Implement Robust Data Quality Measures: Regularly monitor and clean data.
- Leverage Technology: Use MDM tools and platforms to automate and streamline processes.
- Continuous Improvement: Regularly review and refine MDM practices.

MDM Solutions and Tools:

- Informatica MDM: Offers comprehensive data management, governance, and integration capabilities.
- SAP Master Data Governance (MDG): Integrates with SAP ERP systems to manage master data across the enterprise.
- IBM InfoSphere MDM: Provides a robust platform for managing and governing master data across various domains.
- Oracle MDM: Includes multiple products for managing customer, product, supplier, and other types of master data.

- Talend MDM: An open-source platform that integrates MDM with data integration and data quality capabilities.
- Microsoft Master Data Services (MDS): Part of SQL Server, providing a platform for managing and storing master data.
- TIBCO EBX: Offers a comprehensive, multi-domain MDM solution with data governance and data quality features.
- SAS Data Management: Provides a comprehensive suite of tools for data integration, data quality, and MDM.

Benefits of Enterprise Data Management

- Improved Data Quality
- Enhanced Decision Making
- Regulatory Compliance
- Operational Efficiency

Challenges in Implementing EDM

- Data Silos (A data silo is a collection of data that's isolated from the rest of an organization and only accessible to one department.)
- Data Integration Complexity
- Ensuring Data Quality
- User Adoption

Future Trends in EDM

- Increased Focus on Data Governance
- Integration with AI and ML
- Real-Time Data Processing
- Enhanced Data Security

Benefits of Business Intelligence for Competitive Advantage

- Improved Decision Making
  - Data-Driven Insights: BI enables organizations to base their decisions on accurate and timely data.
  - Real-Time Analysis: Access to real-time data allows for quick adjustments and responses to market changes.
- Enhanced Operational Efficiency
  - Process Optimization: BI helps identify inefficiencies and bottlenecks in business processes.
  - Resource Management: Better insights into resource utilization help in optimizing inventory, reducing costs, and improving overall efficiency.

- Customer Insights and Personalization
  - Customer Behavior Analysis: BI tools analyze customer data to understand preferences, buying patterns, and behavior.
  - Personalized Marketing: Insights from BI enable personalized marketing campaigns, improving customer engagement and loyalty.
- Market Intelligence
  - Competitive Analysis: BI provides insights into competitors' strategies, strengths, and weaknesses.
  - Market Trends: Identifying and analyzing market trends helps in anticipating changes and staying ahead of competitors.
- Innovation and New Opportunities
  - Identifying Gaps: BI helps in identifying gaps in the market and potential areas for new products or services.
  - Predictive Analytics: Using historical data to predict future trends and customer needs, allowing for proactive innovation.

- Financial Performance Management
  - Budgeting and Forecasting: BI tools enhance financial planning, budgeting, and forecasting accuracy.
  - Cost Reduction: Identifying cost-saving opportunities and optimizing expenditure.

- Regulatory Compliance
  - Data Governance: Ensuring that data is accurate, consistent, and compliant with regulations.
  - Risk Management: Identifying and mitigating risks through comprehensive data analysis.

Use Cases of Business Intelligence for Competitive Advantage

- Retail Industry
  - Customer Segmentation: Analyzing customer data to segment the market and tailor marketing strategies.
  - Inventory Management: Optimizing inventory levels based on sales trends and forecasts.

- Healthcare Industry
  - Patient Care Optimization: Using BI to improve patient care by analyzing treatment outcomes and optimizing workflows.
  - Operational Efficiency: Streamlining hospital operations and reducing costs through data-driven insights.

- Financial Services
  - Fraud Detection: Analyzing transaction data to detect and prevent fraudulent activities.
  - Customer Retention: Identifying at-risk customers and developing retention strategies.
- Manufacturing Industry
  - Production Optimization: Analyzing production data to improve efficiency and reduce waste.
  - Supply Chain Management: Enhancing supply chain visibility and optimizing logistics.

Implementing Business Intelligence for Competitive Advantage

- Strategic Planning
  - Define Objectives: Clearly define business objectives and how BI will help achieve them.
  - Align with Business Goals: Ensure that BI initiatives align with overall business goals and strategies.
- Data Management
  - Data Quality: Invest in data quality management to ensure accurate and reliable data.
  - Data Integration: Integrate data from various sources for a holistic view.

- Technology and Tools
  - Choose the Right Tools: Select BI tools that meet the specific needs of the organization.
  - Leverage Advanced Analytics: Use advanced analytics such as machine learning and AI to gain deeper insights.
- Skill Development
  - Training and Development: Train employees on BI tools and data analysis techniques.
  - Hire Experts: Consider hiring data scientists and BI experts to maximize the value of BI.

- Change Management
  - Stakeholder Engagement: Engage stakeholders early and often to ensure buy-in and support.
  - Communication: Clearly communicate the benefits and changes associated with BI initiatives.
- Continuous Improvement
  - Monitor and Evaluate: Regularly monitor BI performance and make necessary adjustments.
  - Stay Updated: Keep up with the latest BI trends and technologies to continuously improve BI capabilities.

Key Takeaways:

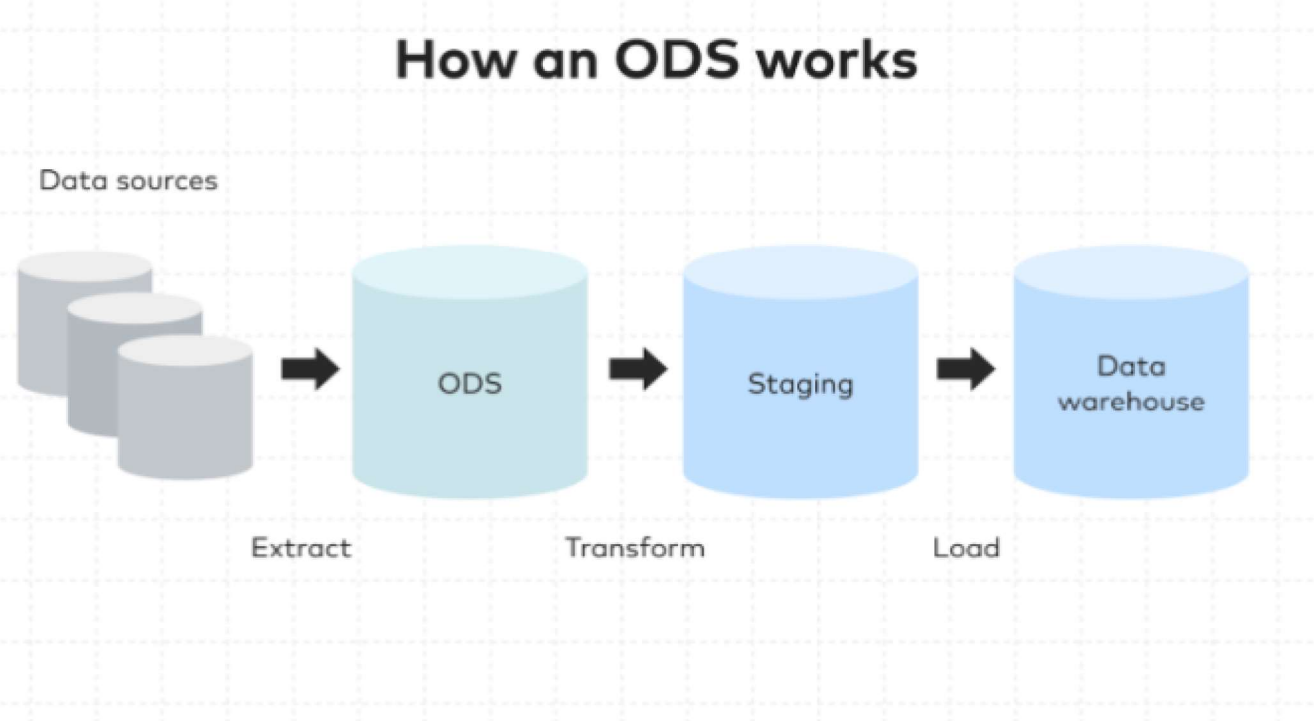
- Clear Planning: Detailed project planning and stakeholder alignment are crucial for success.
- Robust Data Governance: Effective data governance ensures consistent and reliable data.
- Comprehensive Testing: Rigorous testing phases minimize errors and ensure system readiness.
- User Training and Adoption: Proper training ensures users can effectively leverage the data warehouse.
- Continuous Improvement: Ongoing maintenance and enhancements keep the data warehouse relevant and efficient.

Operational Data Store

An **Operational Data Store (ODS)** is a database designed to integrate and store data from multiple sources for operational reporting and decision-making. Unlike a data warehouse, which is optimized for historical and analytical queries, an ODS is typically updated in real-time or near real-time and is used to support day-to-day operations.

An ODS is a centralized database that integrates data from various sources in near real-time for operational reporting and decision-making.

Operational Data Store (ODS)



Purpose:

Designed for day-to-day operations and short-term decision-making.

Provides a real-time, integrated view of data.

Data Refresh:

Continuously updated with real-time data.

Supports frequent, real-time updates.

Data Storage Duration:

Typically stores data for short-term use.

Data is often purged or archived after a short period.

Use Cases:
Operational reporting.
Real-time customer service support.
Real-time inventory and supply chain management.

Data Type:
•Stores current, transactional data.
•Data is detailed and unaggregated.

Users:
•Primarily used by operational staff and managers.
•Supports day-to-day business processes.

Feature	Operational Data Store (ODS)	Data Warehouse (DW)
Purpose	Short-term, operational decision-making	Long-term, strategic decision-making
Data Refresh	Real-time or near real-time	Periodic (batch processing)
Data Storage Duration	Short-term	Long-term
Use Cases	Operational reporting, real-time analysis	BI, analytics, historical trend analysis
Data Type	Current, transactional data	Historical, aggregated data
Users	Operational staff, managers	Analysts, data scientists, senior management
Data Integration	Real-time integration from multiple sources	Periodic integration from multiple sources
Storage Complexity	Less complex, focuses on current data	More complex, handles large volumes of historical data

Normalization

Normalization

- Normalization is the process of organizing data in a database.
- It helps in reducing data redundancy (repeating data) and improving data integrity (accuracy and consistency).

1. Decompose tables: Break down a single table into multiple tables.
2. Apply rules: Use rules to divide larger tables into smaller ones and link them with relationships.
3. Repeat: Continue decomposing tables until the Single Responsibility Principle (SRP) is achieved, which states that each table should have only one role.

Why is Normalization Important?

- Reduces Redundancy:
  - Prevents storing the same piece of data in multiple places.
- Improves Data Integrity:
  - Ensures the data is accurate and consistent.
- Easier Maintenance:
  - Simplifies updating data.

Normalization Levels (Forms)

1. First Normal Form (1NF):

   Eliminate repeating groups; make sure each column contains atomic (indivisible) values.

2. Second Normal Form (2NF):

   Remove subsets of data that apply to multiple rows; ensure each column depends on the whole primary key.

3. Third Normal Form (3NF):

   Eliminate columns not dependent on the primary key

Normal Form	Description
1NF	A relation is in 1NF if it contains an atomic value.
2NF	A relation will be in 2NF if it is in 1NF and all non-key attributes are fully functional dependent on the primary key.
3NF	A relation will be in 3NF if it is in 2NF and no transition dependency exists.

Unnormalized Table

- Imagine you have a table of students and their favorite subjects:

StudentID	StudentName	Subjects
1	Alice	Math, Science
2	Bob	History, Math
3	Charlie	Science, History

First Normal Form (1NF)

- Separate subjects into different rows:

Student ID	Student Name	Subject
1	Alice	Math
1	Alice	Science
2	Bob	History
2	Bob	Math
3	Charlie	Science
3	Charlie	History

Second Normal Form (2NF)

- Remove partial dependencies; create separate tables:
- **Students Table:**

StudentID	Student Name
1	Alice
2	Bob
3	Charlie

Subjects Table:

SubjectID	Subject
1	Math
2	Science
3	History

Third Normal Form (3NF)

- These tables are structured as follows:

- Students Table: Contains unique Student IDs and their corresponding names.
- Subjects Table: Contains unique Subject IDs and their corresponding subject names.
- Student Subjects Table: Acts as a relationship table that links students to their subjects using Student IDs and Subject IDs.

- **Students Table:**

StudentID	Student Name
1	Alice
2	Bob
3	Charlie

Subjects Table:

Subject ID	Subject
1	Math
2	Science
3	History

Student Subjects Table:

StudentID	SubjectID
1	1
1	2
2	3
2	1
3	2
3	3

Slowly Changing Dimension

Slowly Changing Dimension

- The "Slowly Changing Dimension" problem is a common one particular to data warehousing. In a nutshell, this applies to cases where the attribute for a record varies over time. We give an example below:
- Murali Krishna Jandhyala became a customer with JGC Inc. in 2021. He first lived in Wilmington, DE.

CustomerID	Name	Address
1	MuraliKrishna Jandhyala	4 Murphy Road, Wilmington, DE 19803

Type 0: Fixed Dimension

- **Description:**
- In a Type 0 SCD, the data in the dimension is never changed or updated once it's entered.
- This type is useful for data that should remain historically accurate without any changes.
- **Example:** A table of product launch dates where historical accuracy is crucial.

ProductID	ProductName	LaunchDate
1	Product A	1947-15-08

SCD : Type 1 Overwrite

- **Type 1 Slowly Changing Dimension** In Type 1 Slowly Changing Dimension, the new information simply overwrites the original information. In other words, no history is kept.
- This method does not maintain any history of changes.
- **Example:** Imagine a customer table where a customer's address can change over time.

Initial Data:

CustomerID	Name	Address
1	MuraliKrishna Jandhyala	4 Murphy Road, Wilmington, DE 19803

- **Update:** MK Jandhyala moves to a new address, 456 Oak St.

CustomerID	Name	Address
1	MuraliKrishna Jandhyala	456 Oak St, Wilmington,DE 19803

The address for Murali Krishna is simply overwritten with the new address.

- Advantages:
- - This is the easiest way to handle the Slowly Changing Dimension problem, since there is no need to keep track of the old information.
- Disadvantages:
- - All history is lost. By applying this methodology, it is not possible to trace back in history. For example, in this case, the company would not be able to know that MK Jandhyala lived in DE before.
- Usage:
- About 50% of the time.

Type 2: Add New Row

- In a Type 2 SCD, changes in the source data result in a new row being added to the dimension table.
- This method maintains a full history of changes.
- Typically includes additional columns like "StartDate" and "EndDate" to indicate the validity period of each row.
- **Example:** Using the same customer example, let’s implement a Type 2 SCD

- Initial Data:

CustomerID	Name	Address	StartDate	EndDate
1	MuraliKrishna Jandhyala	4 Murphy Road, Wilmington, DE 19803	2022-01-01	9999-12-31

Update:

MK Jandhyala moves to a new address, 456 Oak St, on 2023-01-01.

CustomerID	Name	Address	StartDate	EndDate	Status Y/N
1	MuraliKrishna Jandhyala	4 Murphy Road, Wilmington, DE 19803	2022-01-01	2023-01-01	
2	MuraliKrishna Jandhyala	68 Hamilton Edge Cary, NC 27519	2023-01-01	9999-12-31	

A new row is added with the new address, and the old row's "EndDate" is updated to reflect the change.

- Advantages:
- This allows us to accurately keep all historical information.
- Disadvantages:
- - This will cause the size of the table to grow fast. In cases where the number of rows for the table is very high to start with, storage and performance can become a concern.
- - This complicates the ETL process.

Type 3: Add New Column

- In a Type 3 SCD, changes in the source data are stored in additional columns in the dimension table.
- This method maintains a limited history of changes, typically the current and one previous value.
- Initial Data

CustomerID	Name	Current Address	Previous Address
1	MuraliKrishna Jandhyala	4 Murphy Road, Wilmington, DE 19803	

Initial Data

CustomerID	Name	Current Address	Previous Address
1	MuraliKrishna Jandhyala	4 Murphy Road, Wilmington, DE 19803	

- Updated Data

2023-01-01	Name	Current Address	Previous Address
1	MuraliKrishna Jandhyala	68 Hamiltoin Edge, Cary NC 27519	4 Murphy Road, Wilmington, DE 19803

2023-01-01	Name	Current Address	Previous Address
1	MuraliKrishna Jandhyala	60 Groton, CT	68 Hamiltoin Edge, Cary NC 27519

The new address is stored in the "CurrentAddress" column, and the old address is moved to the "PreviousAddress" column.

- Advantages:
- - This does not increase the size of the table, since new information is updated.
- - This allows us to keep some part of history.
- Disadvantages:
- - Type 3 will not be able to keep all history where an attribute is changed more than once.

Type 4: Add Historical Table

- Type 4 Slowly Changing Dimension, the dimension table always holds the latest data. At the same time, there is a history table that tracks the change.
- The history table will have some of the same columns as the dimension table, but there will be some other columns, such as update, that are used to track the changes.
- In a Type 4 SCD, a separate historical table is used to keep track of old records.
- The main table only holds the current data.

Example:

- A customer table with a separate historical table for old addresses.

Main Table

CustomerID	Name	Address
1	MuraliKrishna Jandhyala	68 Hamiltoin Edge, Cary AR 27519

Historical Table:

CustomerID	Name	Old Address	StartDate	EndDate
1	MuraliKrishna Jandhyala	4 Murphy Road, Wilmington, DE 19803	2022-01-01	9999-12-31
		NC		
		NY		

Type	Name	Description	Example
Type 0	Retain Original	No changes are made to the existing records. The original data remains unchanged.	Customer's original address remains unchanged even if they move.
Type 1	Overwrite	Updates the existing dimension record with the new data, thus overwriting the old data.	Changing a customer's address directly in the database without keeping a history.
Type 2	Add New Row	Adds a new record to the dimension table with a new version of the data. Keeps historical records of all changes by adding rows with new version keys.	Inserting a new row for each change in a customer's address along with effective date and expiration date.
Type 3	Add New Attribute	Adds a new column to the dimension table to store the old value of the attribute. Limited historical data.	Adding a column to store the previous address of a customer.
Type 4	Add Historical Table	Uses a separate historical table to track changes, while the main dimension table stores only the current data.	Maintaining a history of changes in a separate table linked to the main customer table.
Type 5	Combination of Type 1, Type 2, and Mini-Dimension	Combines aspects of Type 1 and Type 2 with the use of a mini-dimension for rapidly changing attributes.	Using a mini-dimension to store customer preferences that change frequently, while other attributes use Type 1 or Type 2.
Type 6	Hybrid	Combines techniques from Type 1, Type 2, and Type 3. Keeps current values, historical values, and additional metadata.	Updating the current record, adding a new historical record, and keeping previous values in additional columns.

Power BI Components

- **Power BI Desktop:**
- Primary development tool for creating reports.
- Importing and transforming data using Power Query.
- Building data models and creating visuals.
- **Power BI Service:**
- Online SaaS (Software as a Service) platform for sharing and collaboration.
- Publishing reports from Power BI Desktop to the service.
- Creating and managing dashboards.

- **Power BI Mobile:**
- Accessing and interacting with reports and dashboards on mobile devices.
- Real-time notifications and data alerts.

Data Connectivity and Transformation

- **Data Sources:**
- Connecting to various data sources: databases (SQL Server, Oracle, MySQL), cloud services (Azure, AWS), Excel, CSV, web data, and more.
- Using Power Query Editor to clean and transform data.

- **Data Modeling:**
- Creating relationships between different data tables.
- Understanding star schema and snowflake schema.

Creating Visualizations

- Visualization Types:

- Different types of visuals: bar charts, line charts, pie charts, maps, tables, matrix, and custom visuals.
- Best practices for choosing the right visual for your data.

- Interactivity:
- Adding filters, slicers, and drill-through actions.
- Creating interactive and dynamic reports with tooltips and bookmarks.

Best Practices and Performance Optimization

Report Design Best Practices:

- Keeping reports simple and focused.
- Ensuring readability and accessibility.
- Using themes and templates for consistency.

Performance Optimization:

- Tips for improving report performance: reducing data load, optimizing DAX calculations, using aggregations.
- Monitoring and analyzing performance using the Performance Analyzer tool in Power BI Desktop.

Case Studies and Practical Examples

- Real-World Use Cases
  - Presenting examples of how Power BI is used in different industries.
  - Showcasing dashboards and reports that solve specific business problems.

  Hands-On Demonstrations:
  - Live demonstration of creating a Power BI report from scratch.
  - Interactive session where participants can follow along and create their own reports.


THANK YOU!