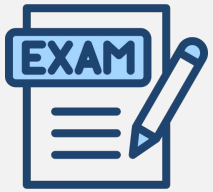
DATA WAREHOUSE

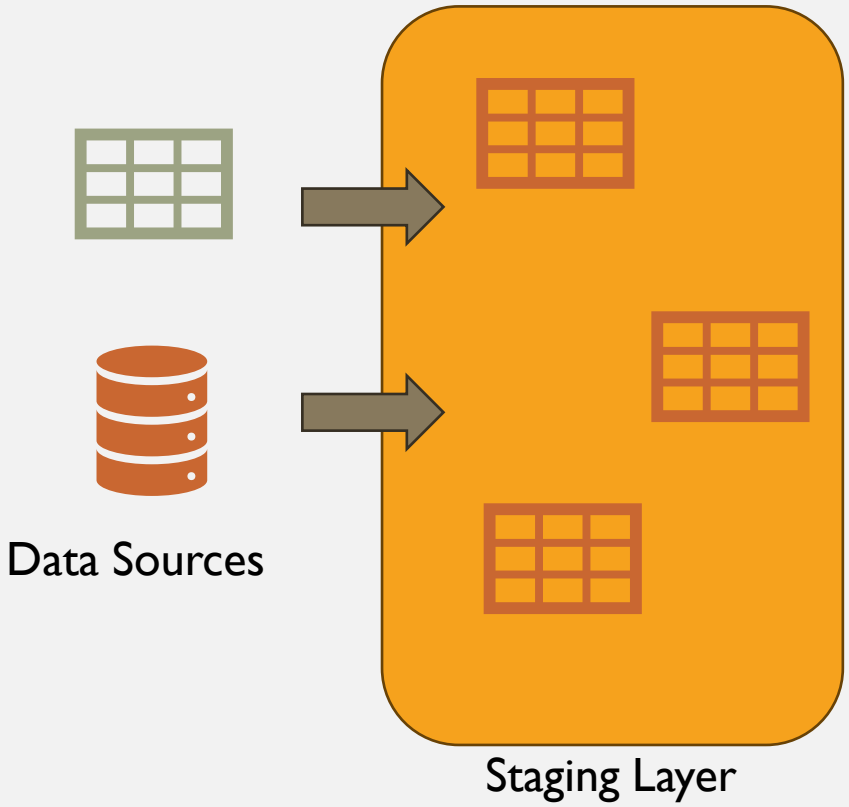
Chapter II – Data Warehouse Architecture

AIMS
(PART 1)

- Data Warehouse Layer Architecture
- Data Mart
- Stagin-Layer

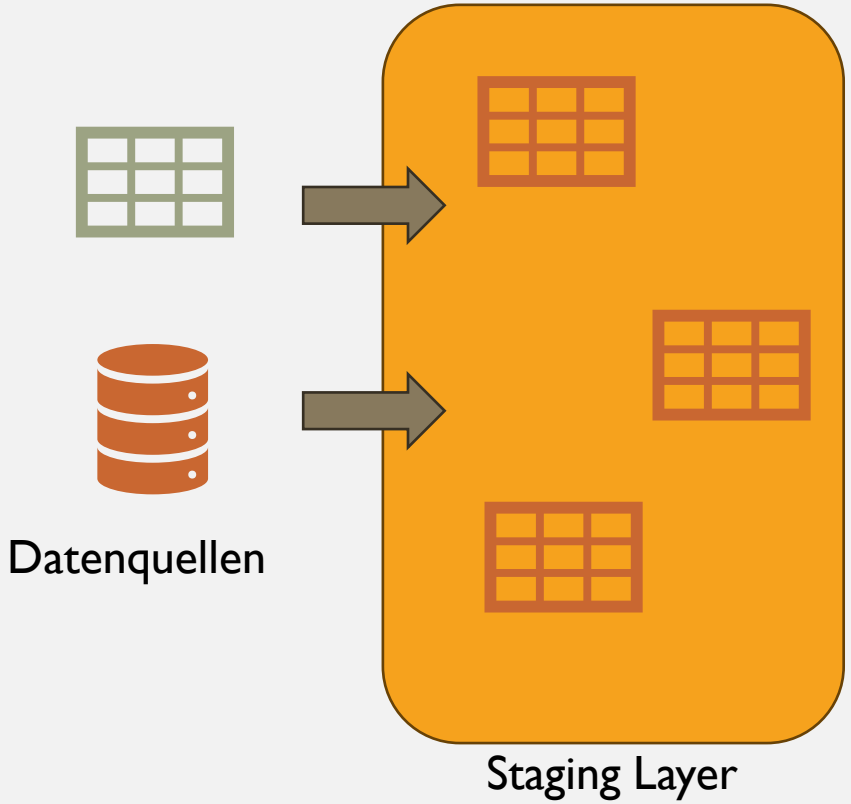


DATA WAREHOUSE LAYER ARCHITECTURE



- Architecture of a Data Warehouse:
  - Multiple layers
  - Data sources and ETL process:
  - Extraction, Transformation, and Loading of data from the data sources
- ETL process:
  - Utilization of an ETL tool for extracting various data formats
- Staging Layer:
  - First layer of the Data Warehouse
  - Used for extracting data into tabular form

DATA WAREHOUSE LAYER ARCHITECTURE



Department B

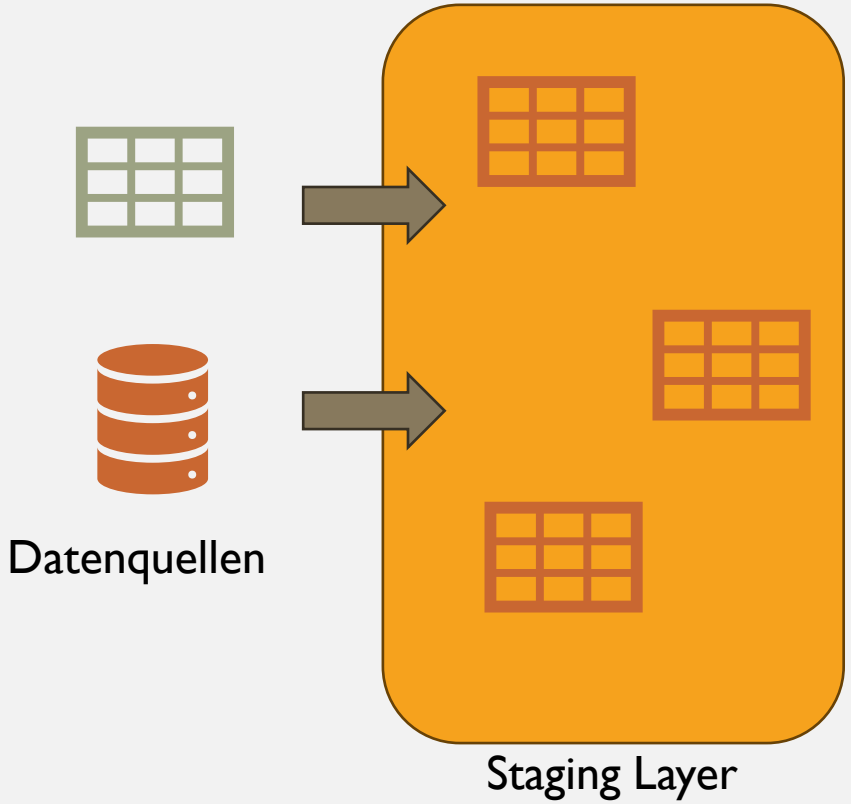
employee_id	entry_date	position_level
1	1/2/2022	HR
2	1/6/2022	IT
3	1/4/2022	IT
4	1/7/2022	UM
5	1/6/2022	PM

Department A

employee_id	entry_date	position_level
6	1/5/2022	HR
7	1/6/2022	HR
8	1/8/2022	PM
9	1/7/2022	UM
10	1/6/2022	PM

- Data in tables:
  - No data transformation performed
  - Goal: Keep data as unchanged as possible
- Example: Employee data in different departments
  - Different formats such as CSV files and databases
  - Extraction into the staging area:
- Data extracted from various sources into tabular form
- Possibility of multiple tables in the staging area.

DATA WAREHOUSE LAYER ARCHITECTURE



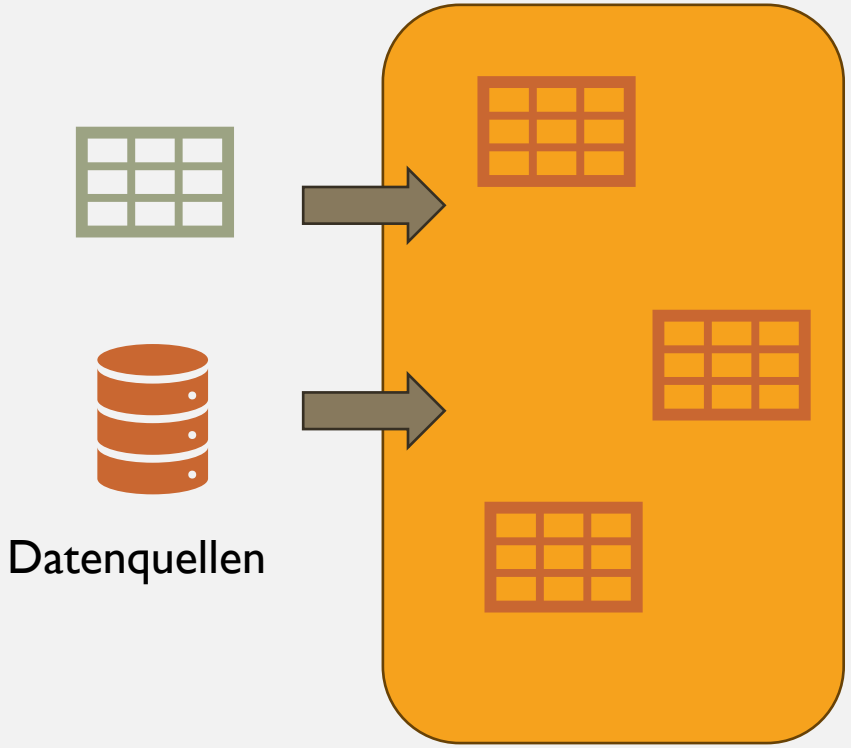
Departments

employee_id	entry_date	position_level
1	1/2/2022	HR
2	1/6/2022	IT
3	1/4/2022	IT
4	1/7/2022	UM
5	1/6/2022	PM
6	1/5/2022	HR
7	1/6/2022	HR
8	1/8/2022	PM
9	1/7/2022	UM
10	1/6/2022	PM

- Possibility of summarization if tables have the same structure.

Last session

DATA WAREHOUSE LAYER ARCHITECTURE



Department B

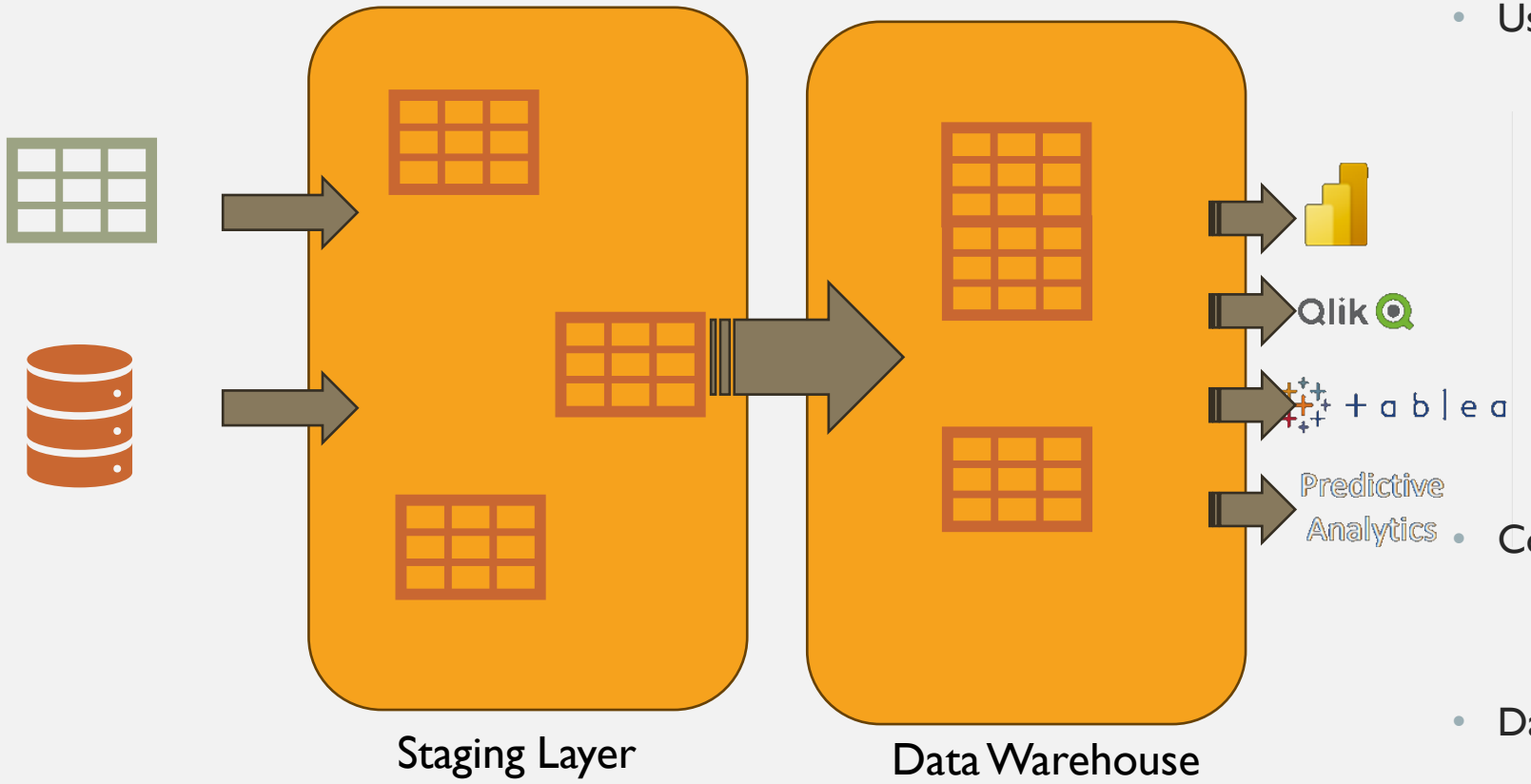
employee_id	entry_date	position_level
1	1/2/2022	HR
2	1/6/2022	IT
3	1/4/2022	IT
4	1/7/2022	UM
5	1/6/2022	PM

Department A

employee_id	entry_date	position
1	1/2/2022	Human Ressources
2	1/6/2022	Information Technologies
3	1/4/2022	Information Technologies
4	1/7/2022	Upper Management
5	1/6/2022	Project Manager

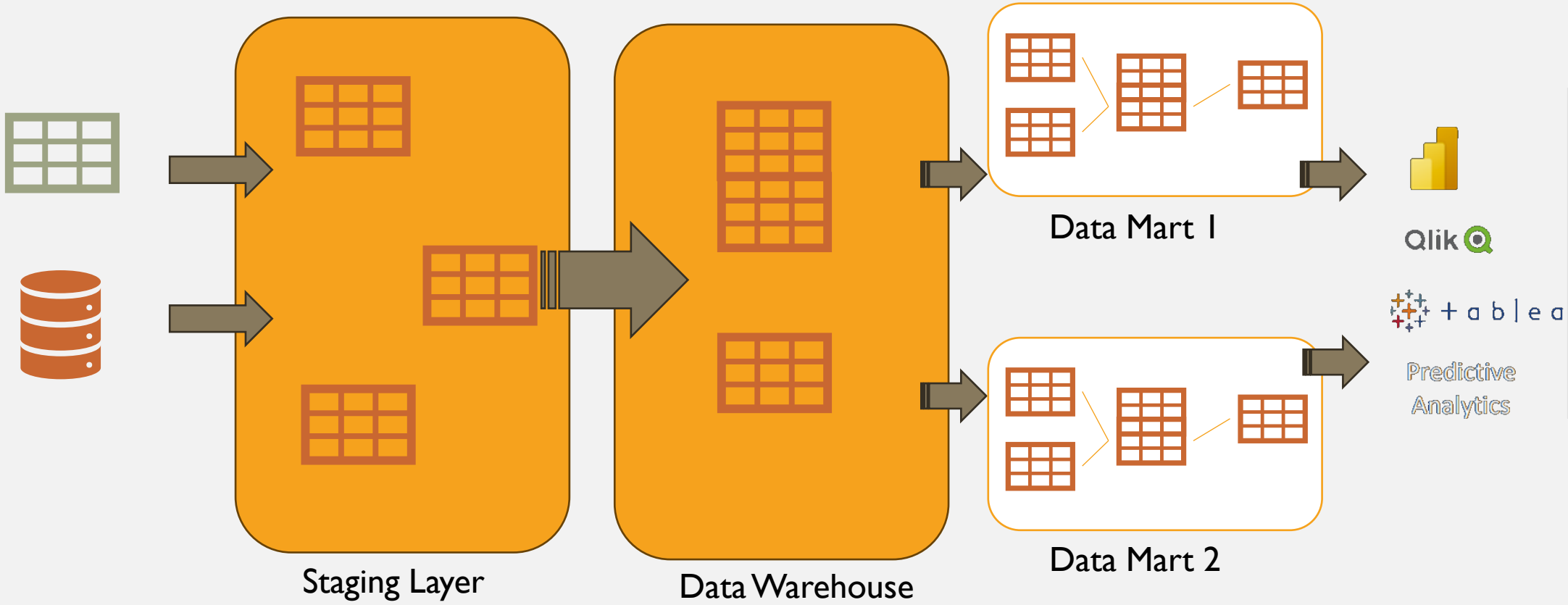
- Differences in the tables:
  - Different column names such as "position" and "position_level"
  - Different data formatting and abbreviations
  - Repetition of IDs in various departments
- Data transformation:
  - Steps to integrate the data and resolve differences
- Data modeling:
  - Restructuring the data according to desired requirements

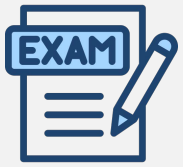
DATA WAREHOUSE LAYER ARCHITECTURE



- Usage of the ETL tool:
  - Performing data copying and transformations
  - Copying the data from the staging to the core layer:
  - The core layer is sometimes considered as the actual data warehouse
  - Transformations occur during the copying process
- Core layer as user interface:
  - Main access point for end users and applications
- Data access and utilization:
  - Creation of reports, data analysis, and predictive analytics

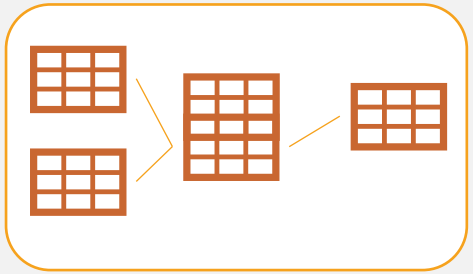
DATA WAREHOUSE LAYER ARCHITECTURE



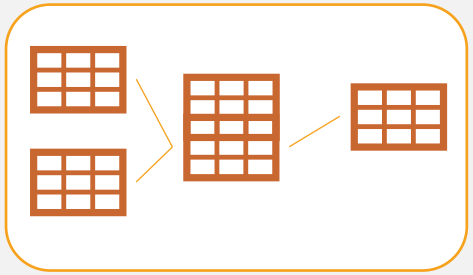


TASK
WHAT IS „DATA MART“

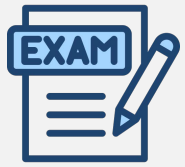
- Please search for the term on the internet.
- Explain what you understand by it.
- Time: 5 minutes



Data Mart 1

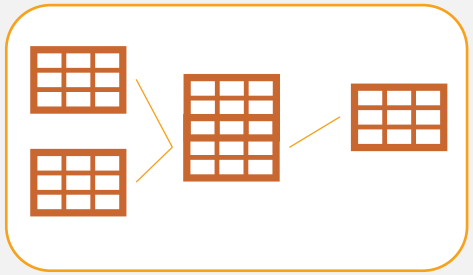


Data Mart 2

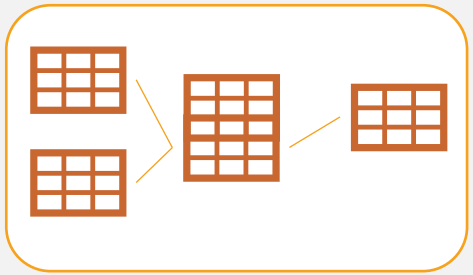


DATA MART

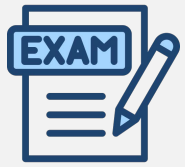
- Building Data Marts:
  - Addition to the core layer in large data warehouses with many tables and use cases
- Purpose:
  - Selecting relevant tables for specific use cases
  - Improving user-friendliness and query performance
- Advantages:
  - Reducing complexity by selecting relevant data
  - Relieving data warehouse performance for specific user groups



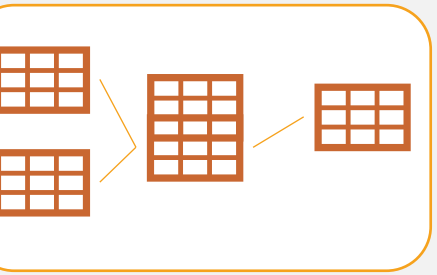
Data Mart 1



Data Mart 2

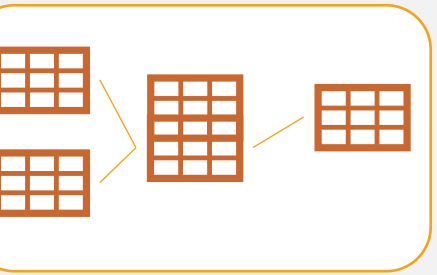


DATA MART



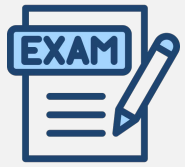
Data Mart 1

- Utilization of specialized databases:
  - In-memory databases or cubes to enhance performance
- Goal: Improving performance through specialization and targeted data provisioning.

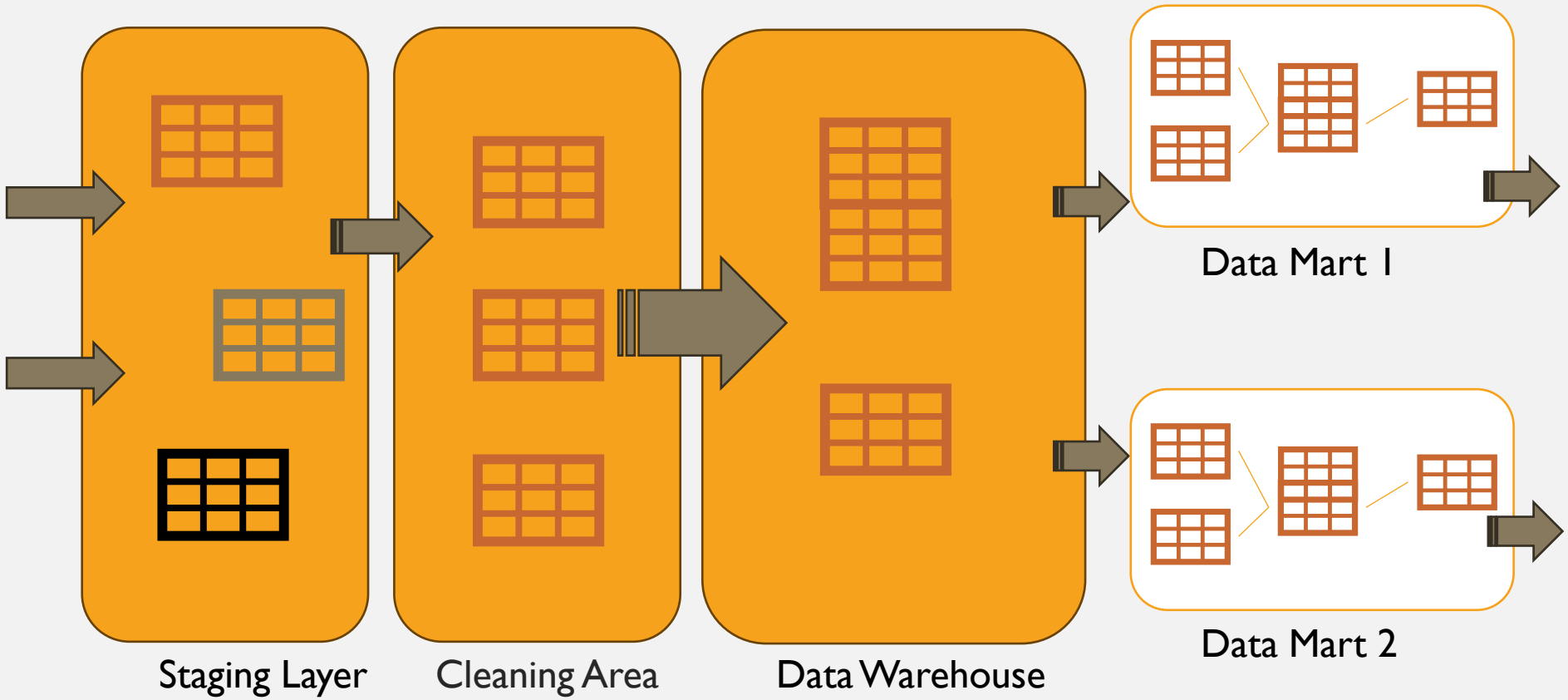


Data Mart 2

- Use case for scenarios with very large data volumes
- Data Mart not always necessary
  - Additional layer, depending on the requirements

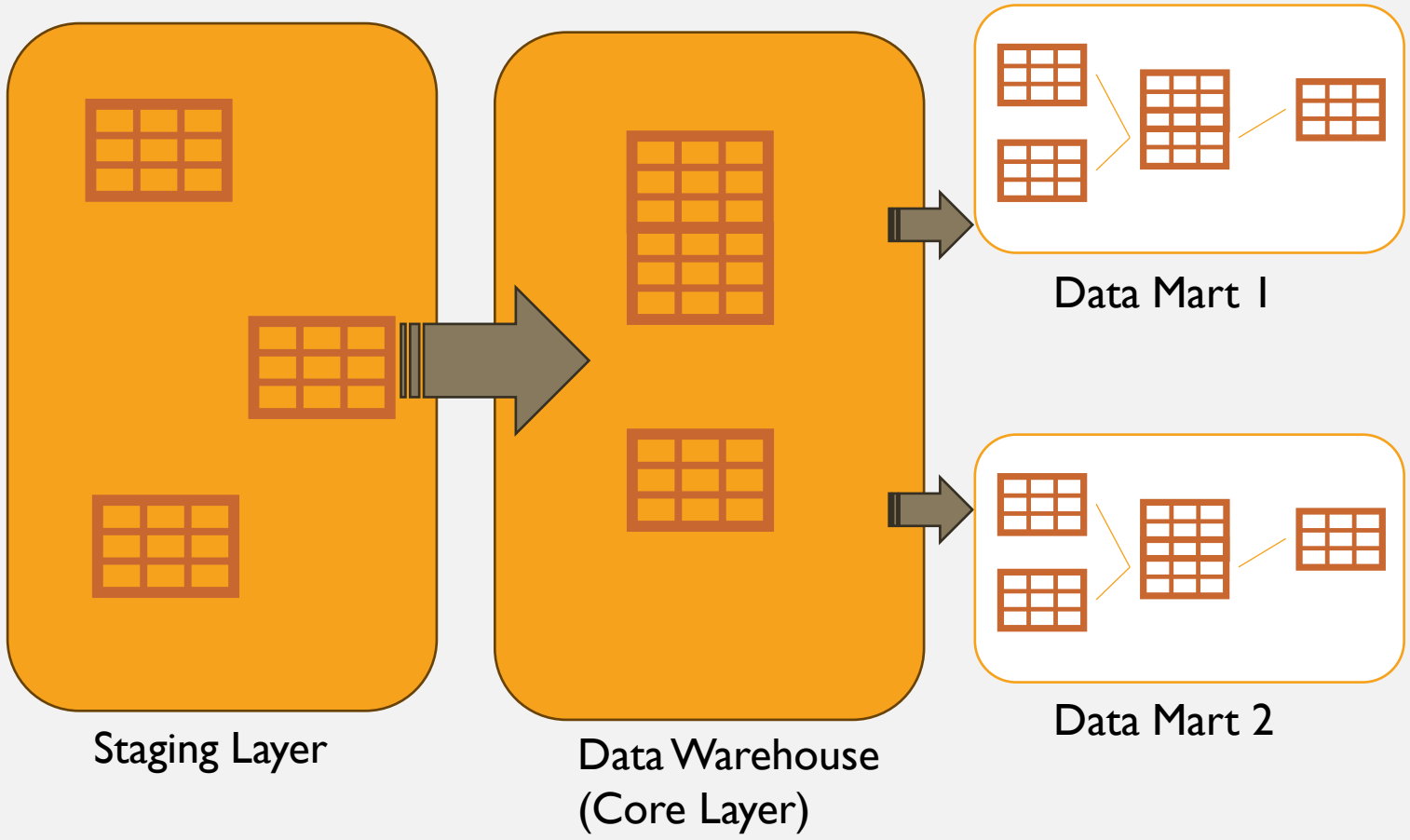


DATA WAREHOUSE LAYER ARCHITECTURE



- Cleaning Area is optional, depending on the rawness of the data
- Special area for data cleaning, if necessary

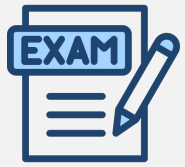
DATA WAREHOUSE
THE TERM IS DIFFICULT TO CLASSIFY



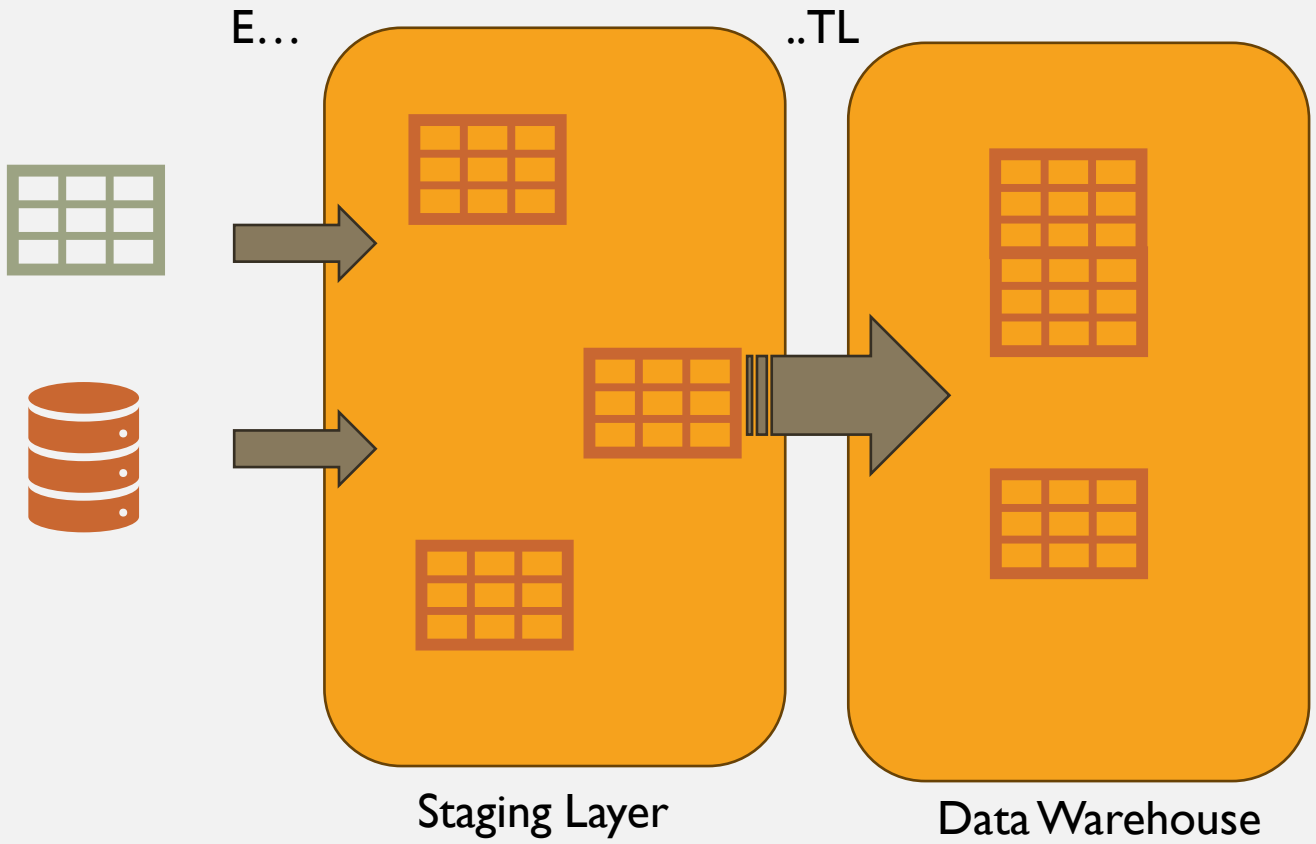
- Confusion about the term "Data Warehouse":
  - Different perceptions of the various layers
- Core Layer:
  - Often considered as the Data Warehouse by end users
  - Single Point of Truth
- Data warehouse encompasses all layers:
  - Core, Data Marts, and Cleansing Area
- Significance of the Staging Area:
  - Important component of the process
- Goal: In-depth examination of the Staging Area in the next lecture.

CHAPTER II – II – STAGING LAYER
TASK: WHAT IS STAGING LAYER

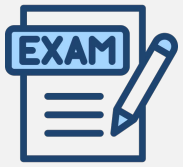
- Search for the term on the internet.
- Explain what you understand by it.
- Time: 5 minutes.



ROLE OF STAGIN-LAYER



- Role of the Staging Environment in the ETL Process:
  - Data extraction from sources
  - Data access without significant time investment
  - Extraction and storage of data in the staging environment
- Connection to the core layer:
  - Definition of transformations with ETL tool
  - Loading into the core layer
- Role of the core layer:
  - Access layer for end users and applications
  - Perception as a data repository (Data Warehouse)



WHY WE NEED A STAGIN-LAYER?

- Why do we need the Staging Layer?
  - Avoidance of complications
  - Prevention of redundant data
  - Weighing the pros and cons
- Reasons for using a Staging Layer:
  - Risk avoidance for operational systems
    - Risk of slowdown due to data access
    - Minimization of system resource usage
  - Need for quick data access
  - Structuring data in relational database tables
- Benefits of data structuring:
  - Application of transformations to relational data
  - Utilization of CSV and JSON files for data structuring
- Implementation in the staging environment:
  - Import of data into relational database tables
  - Definition and application of transformations with ETL tool
- Significance of the Staging Layer in the Data Warehouse:
  - Secure data access and processing
  - Basis for further analysis and queries

PRACTICAL EXAMPLE FOR ILLUSTRATION

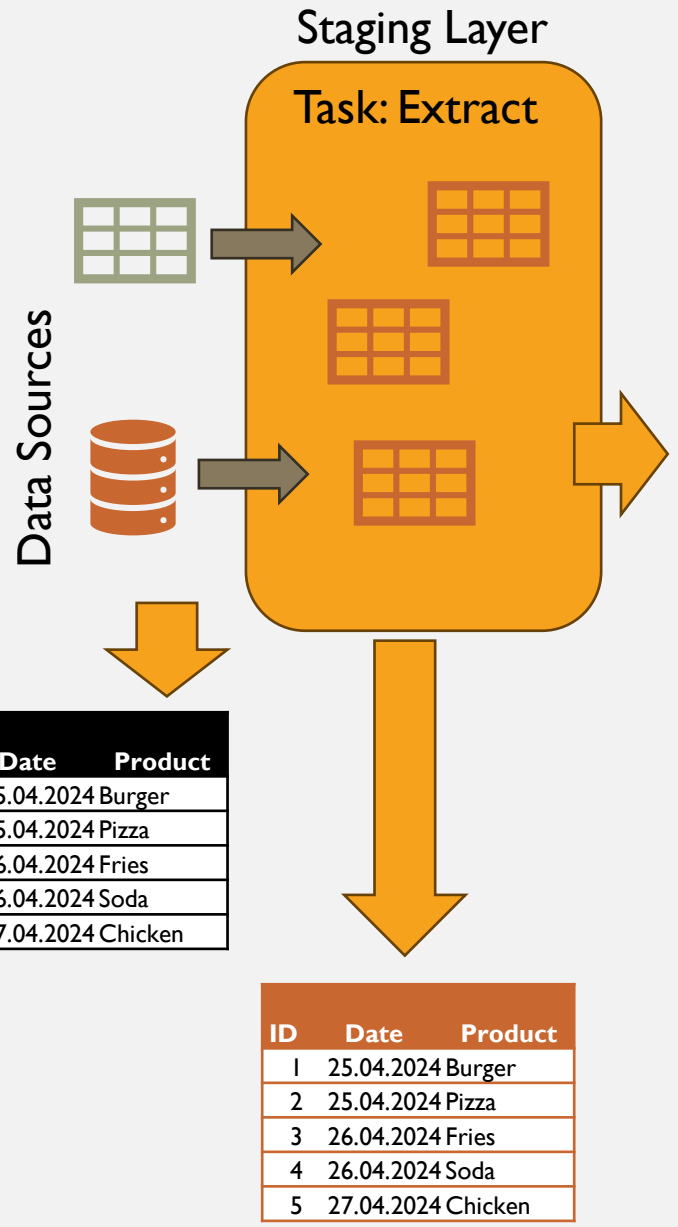
- E-Commerce Plattform:
  - Data sources: Online-Shops, payment processors, inventory management systems
  - Role of the Staging Layer:
    - Extraction, transformation, and loading (ETL) of raw data
    - Integration of data from various sources
    - Preparation of data for the main data warehouse
- Customer Data Management:
  - Data sources: CRM systems, social media, customer surveys
  - Role of the Staging Layer:
    - Integration and preparation of customer information
    - Data cleaning and standardization
    - Preparation of data for analysis and reports

PRACTICAL EXAMPLE FOR ILLUSTRATION

- Financial Reporting:
  - Data sources: Trading systems, payment processors, bank accounts
  - Role of the Staging Layer:
    - Collecting and processing transaction data
    - Cleaning and standardizing financial data
    - Generating financial reports and risk analyses
- Logistics and Supply Chain:
  - Data sources: Warehouse management systems, transportation management systems, GPS trackers
  - Role of the Staging Layer:
    - Integration and cleansing of logistics data
    - Structuring and standardizing supply chain information
    - Analysis of efficiency, inventory management, and route optimization

HOW DOES THE STAGING LAYER WORK?

1. Extraction of data from the source systems:
   - Rapid reading of data from the source systems
   - Extraction of data into the staging environment
2. Application of transformations:
   - Merging the data with additional tables
   - Adding additional columns
   - Performing minor transformations
3. Loading the transformed data into the Data Warehouse:
   - Using the staging environment as an intermediary step
   - Integration of the transformed data into the Data Warehouse



ID	Date	Product	Customer ID	Store ID
1	25.04.2024	Burger	102	5
2	25.04.2024	Pizza	203	3
3	26.04.2024	Fries	305	7
4	26.04.2024	Soda	104	2
5	27.04.2024	Chicken	206	4

ID	Date	Product
1	25.04.2024	Burger
2	25.04.2024	Pizza
3	26.04.2024	Fries
4	26.04.2024	Soda
5	27.04.2024	Chicken

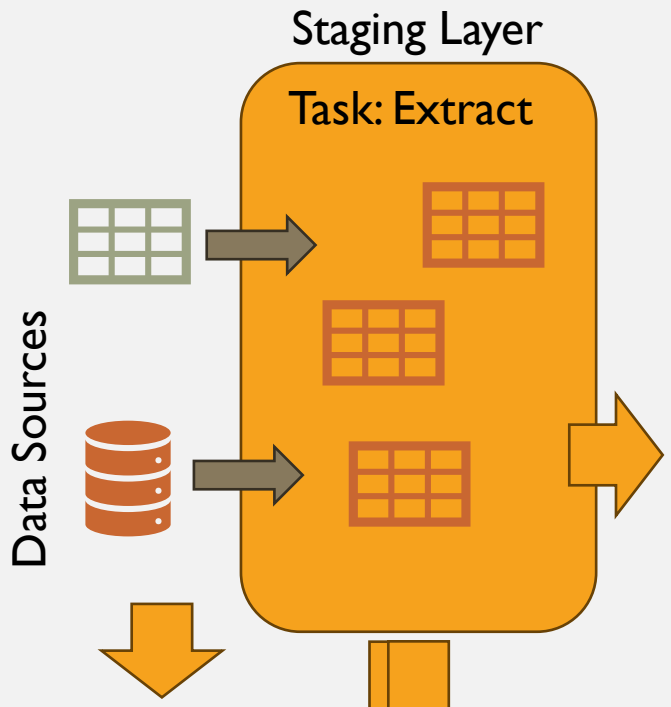
ID	Date	Product
1	25.04.2024	Burger
2	25.04.2024	Pizza
3	26.04.2024	Fries
4	26.04.2024	Soda
5	27.04.2024	Chicken

HOW DOES THE STAGING LAYER WORK?

After a few hours, new data is available

4. Cleansing of the Staging Layer:
   - Emptying the staging environment
   - Temporary contents of the staging layer
   - Truncating the staging environment after each ETL cycle

5. Identification of new data:
   - Implementation of delta logic
   - Using a delta column to identify new data
     - Example: ID column or date column as a delta column
     - Ensuring that the ID column contains strictly increasing numbers
     - Preference for date column in practice



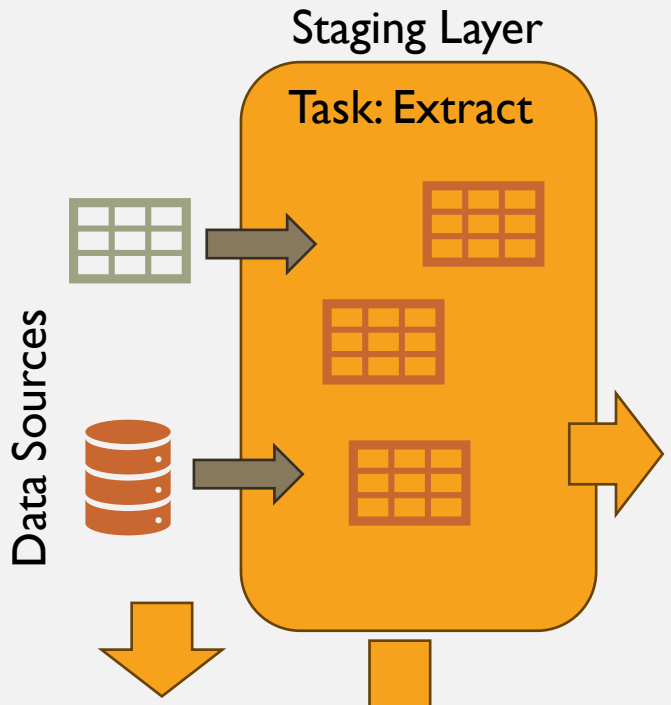
ID	Date	Product	Customer ID	Store ID
1	25.04.2024	Burger	102	5
2	25.04.2024	Pizza	203	3
3	26.04.2024	Fries	305	7
4	26.04.2024	Soda	104	2
5	27.04.2024	Chicken	206	4

ID	Date	Product
1	25.04.2024	Burger
2	25.04.2024	Pizza
3	26.04.2024	Fries
4	26.04.2024	Soda
5	27.04.2024	Chicken
6	27.04.2024	Taco
7	28.04.2024	Hot Dog
8	28.04.2024	Ice Cream
9	29.04.2024	Nachos
10	29.04.2024	Smoothie

ID	Date	Product
6	27.04.2024	Taco
7	28.04.2024	Hot Dog
8	28.04.2024	Ice Cream
9	29.04.2024	Nachos
10	29.04.2024	Smoothie

HOW DOES THE STAGING LAYER WORK?

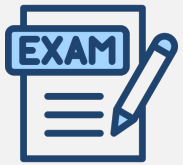
- Loading new data into the Data Warehouse:
  - Selection and loading of new records based on delta logic
  - Example: Loading all records created after a certain date
  - Application of transformations to the new records
  - Appending the transformed data to the Data Warehouse



ID	Date	Product
1	25.04.2024	Burger
2	25.04.2024	Pizza
3	26.04.2024	Fries
4	26.04.2024	Soda
5	27.04.2024	Chicken
6	27.04.2024	Taco
7	28.04.2024	Hot Dog
8	28.04.2024	Ice Cream
9	29.04.2024	Nachos
10	29.04.2024	Smoothie

ID	Date	Product
6	27.04.2024	Taco
7	28.04.2024	Hot Dog
8	28.04.2024	Ice Cream
9	29.04.2024	Nachos
10	29.04.2024	Smoothie

ID	Date	Product	Customer ID	Store ID
1	25.04.2024	Burger	102	5
2	25.04.2024	Pizza	203	3
3	26.04.2024	Fries	305	7
4	26.04.2024	Soda	104	2
5	27.04.2024	Chicken	206	4
6	27.04.2024	Taco	108	1
7	28.04.2024	Hot Dog	210	6
8	28.04.2024	Ice Cream	112	8
9	29.04.2024	Nachos	314	5
10	29.04.2024	Smoothie	116	3

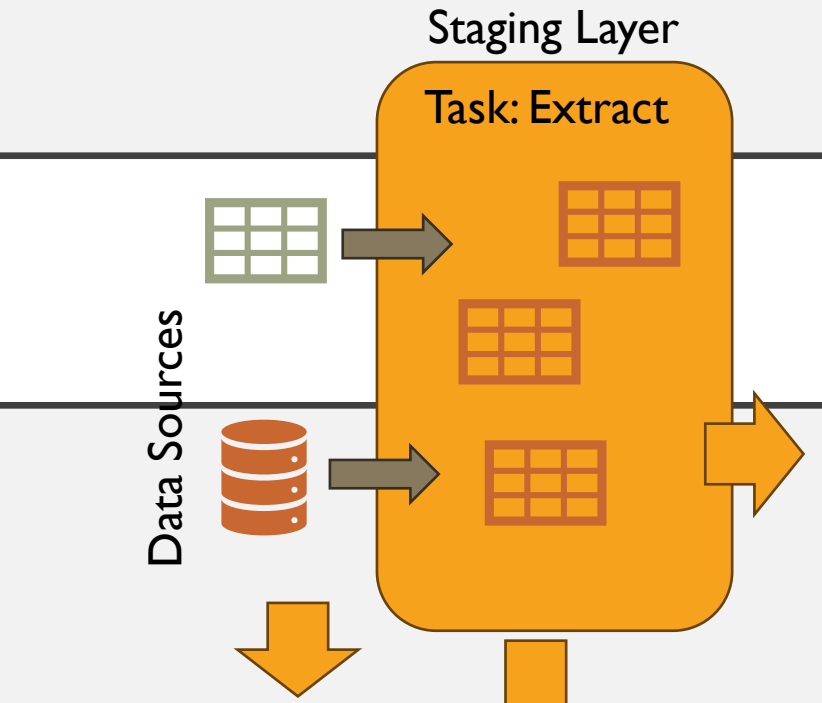


CHALLENGES

- Problem with transformations:
  - Possible errors and issues with transformations
  - Data changes may occur
- Need for rollback:
  - Reversal of erroneous transformations
  - Reverting to previous records
  - Possible restart of the process from previous days

SOLUTION

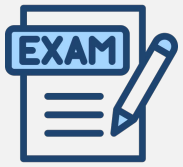
- Persisting Layer
  - Staging Layer will not trunked
  - Rare use of method



ID	Date	Product
1	25.04.2024	Burger
2	25.04.2024	Pizza
3	26.04.2024	Fries
4	26.04.2024	Soda
5	27.04.2024	Chicken
6	27.04.2024	Taco
7	28.04.2024	Hot Dog
8	28.04.2024	Ice Cream
9	29.04.2024	Nachos
10	29.04.2024	Smoothie

ID	Date	Product	Customer ID	Store ID
1	25.04.2024	Burger	102	5
2	25.04.2024	Pizza	203	3
3	26.04.2024	Fries	305	7
4	26.04.2024	Soda	104	2
5	27.04.2024	Chicken	206	4
6	27.04.2024	Taco	108	1
7	28.04.2024	Hot Dog	210	6
8	28.04.2024	Ice Cream	112	8
9	29.04.2024	Nachos	314	5
10	29.04.2024	Smoothie	116	3

ID	Date	Product	Customer ID	Store ID
1	25.04.2024	Burger	102	5
2	25.04.2024	Pizza	203	3
3	26.04.2024	Fries	305	7
4	26.04.2024	Soda	104	2
5	27.04.2024	Chicken	206	4
6	27.04.2024	Taco	108	1
7	28.04.2024	Hot Dog	210	6
8	28.04.2024	Ice Cream	112	8
9	29.04.2024	Nachos	314	5
10	29.04.2024	Smoothie	116	3

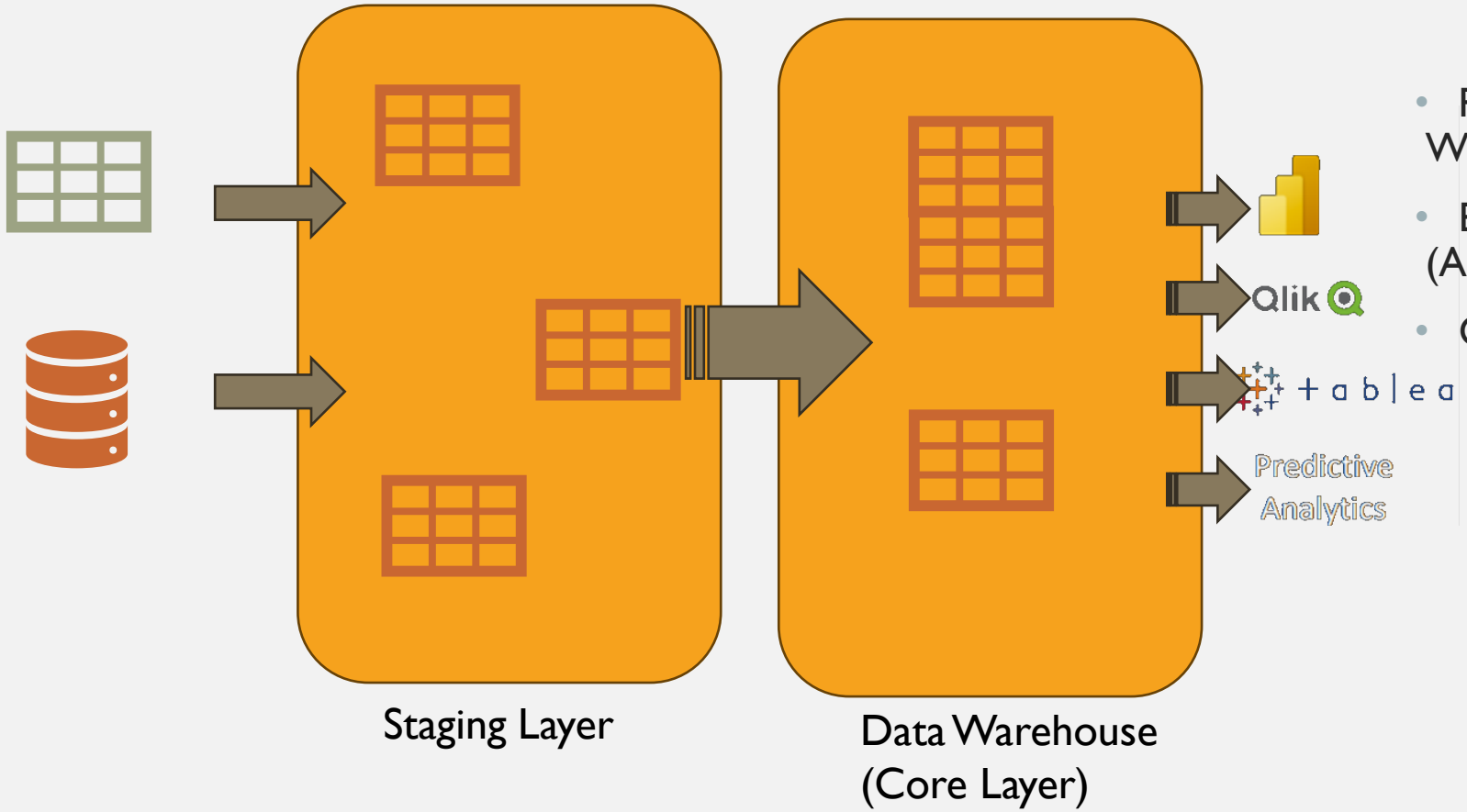


SUMMARY

- Definition of the Staging Layer:
  - Landing zone in the Data Warehouse for extracted data from the data sources
  - Objective: Extraction of data from various files and formats into a relational and separate database
  - Minimization of changes in the Staging Layer to avoid burdening the source systems

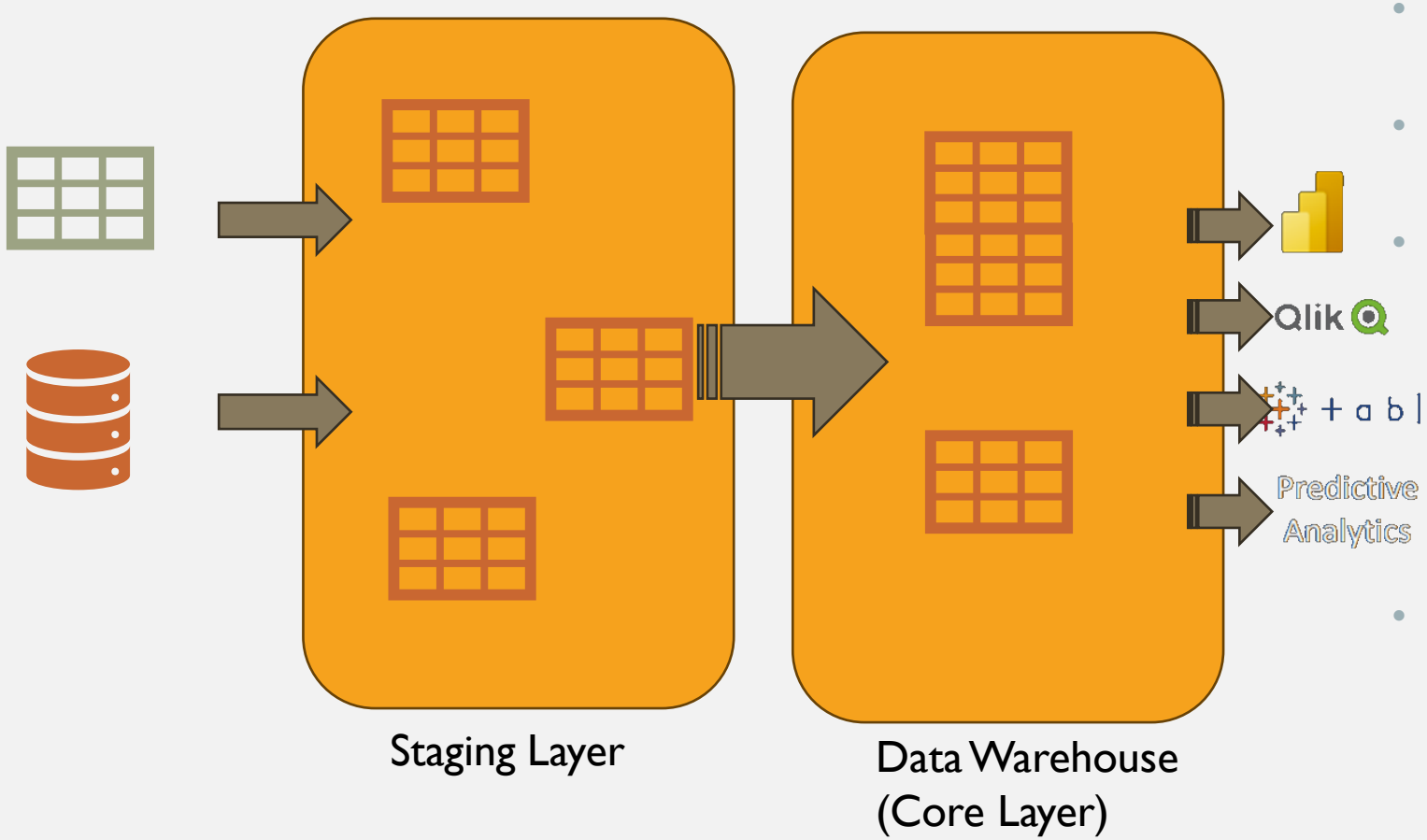
- Types of Staging Layers:
  - Temporary Staging Layer
    - Truncation after each ETL cycle
  - Persistent Staging Layer
    - No truncation after each ETL cycle
    - Retention of source data at this level

CHAPTER II – III
DATA MARTS



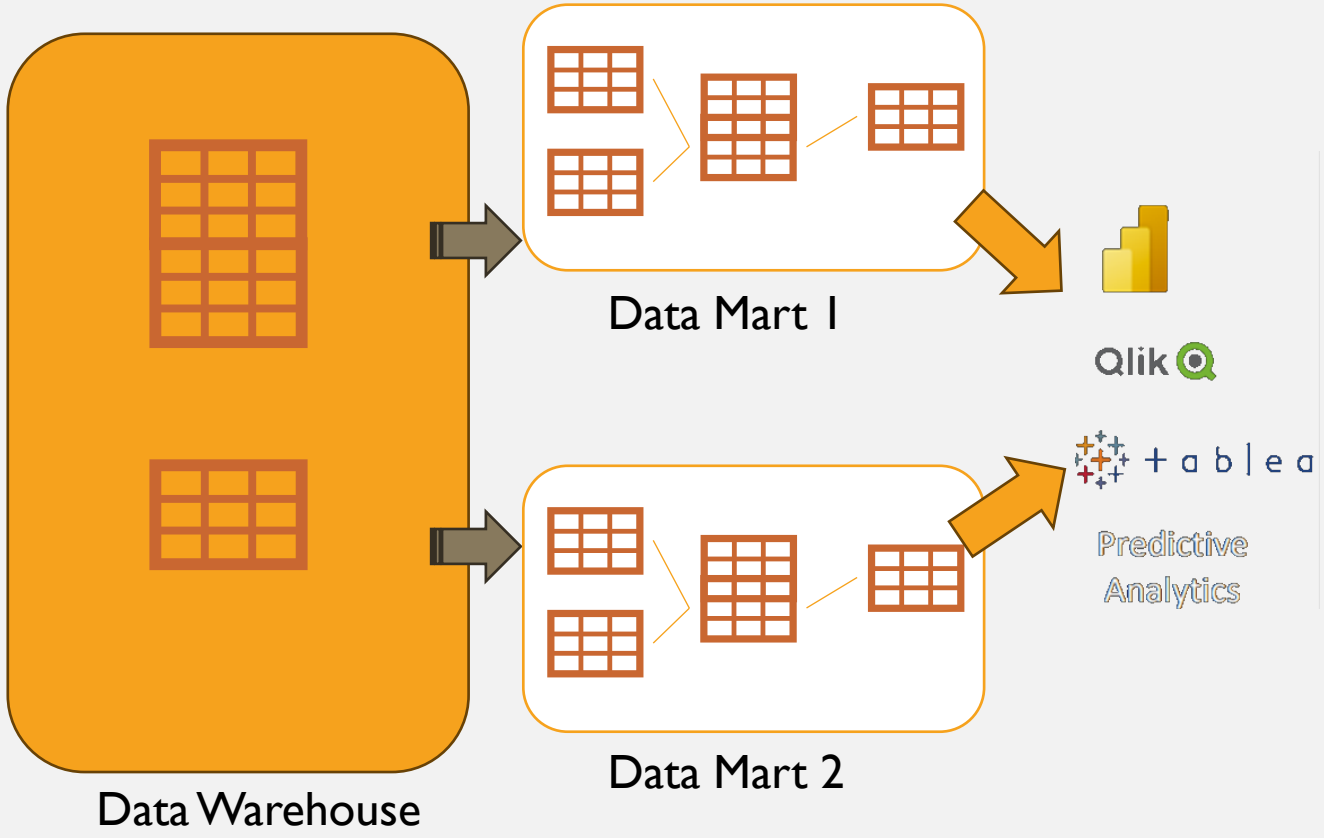
- Placement within the layers of the Data Warehouse
- Existing layers: Staging Layer and Core Layer (Access Layer)
- Challenges in large enterprises
  - Complexity with many different use cases
  - Utilization of various tools in the Data Warehouse
  - All user groups, departments, and regions use the same Data Warehouse

CHAPTER II – III
DATA MARTS



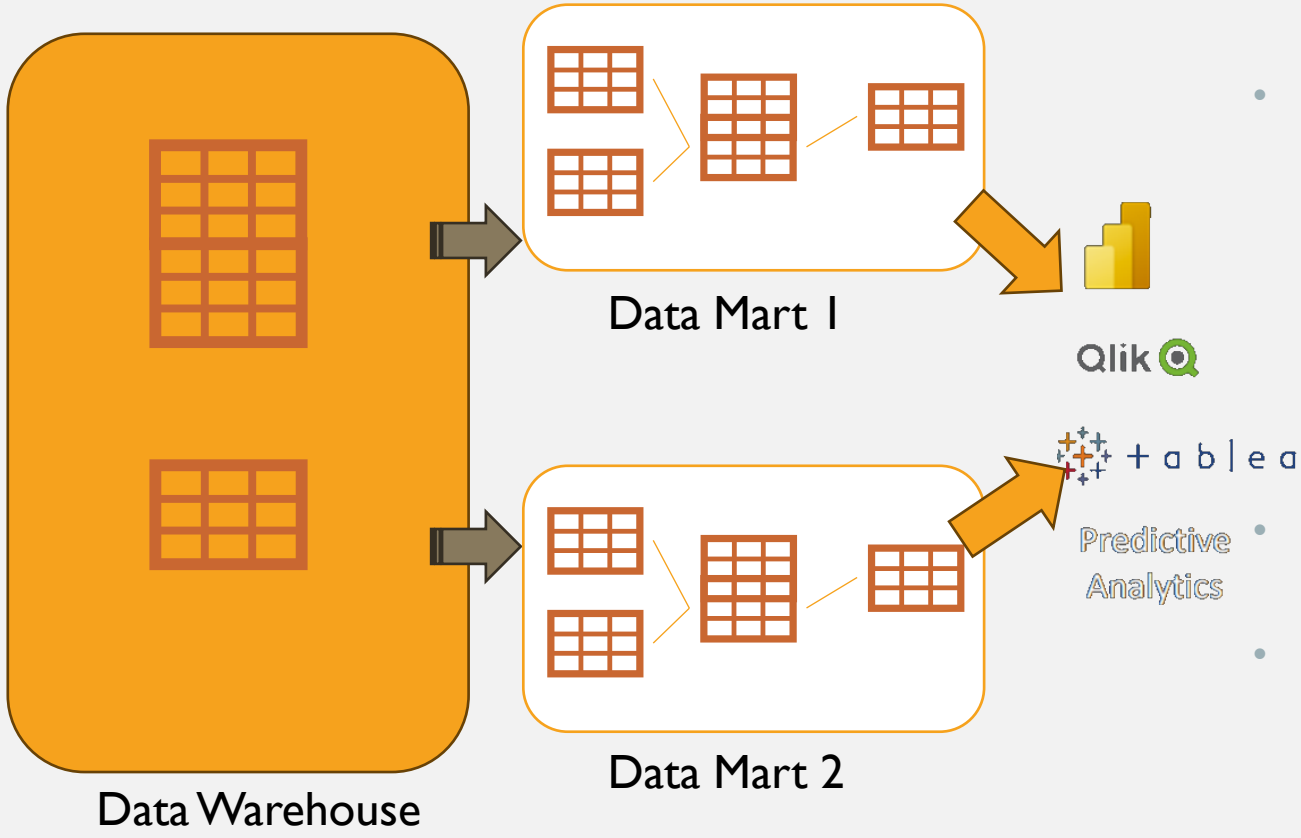
- Placement within the layers of the Data Warehouse
- Existing layers: Staging Layer and Core Layer (Access Layer)
- Challenges in large enterprises
  - Complexity with many different use cases
  - Utilization of various tools in the Data Warehouse
  - All user groups, departments, and regions use the same Data Warehouse
- Additional layer: Data Marts
  - Purpose: Complement to the Core Layer to reduce complexity
  - Data Marts are a subset of the Data Warehouse, specifically the Core Layer

CHAPTER II – III
DATA MARTS



- Additional layer:
  - Data Marts Purpose:
    - Supplement to the Core Layer to reduce complexity Data Marts are a subset of the Data Warehouse, specifically the Core Layer.

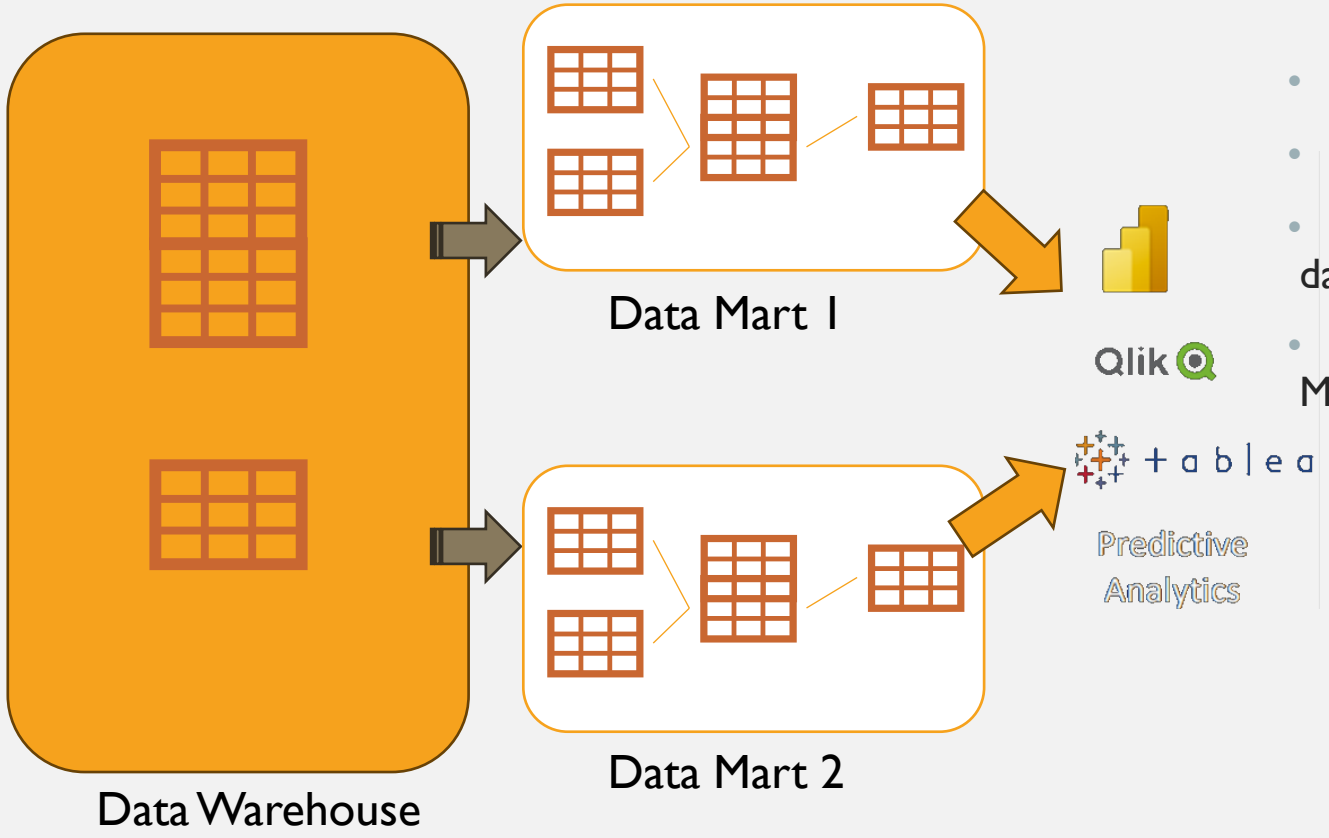
CHAPTER II – III
DATA MARTS



- Data modeling in Data Marts
  - Dimensional modeling
    - Fact tables in the middle
    - Dimension tables around these fact tables
  - This modeling approach will be discussed later.

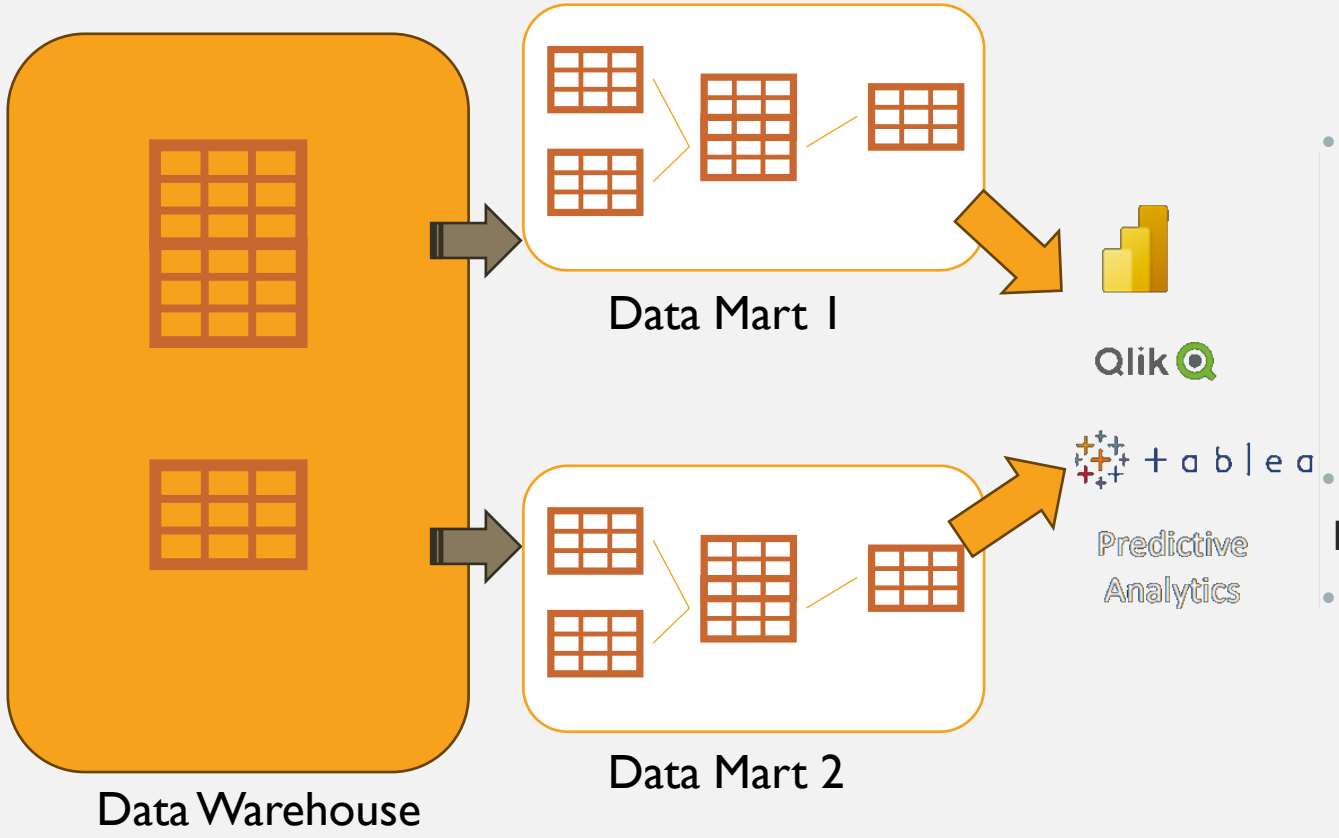
- Modeling in the core layer
  - Possibility of dimension and fact modeling
- Specificity of Data Marts
  - Structure for specific use cases
  - Ability to further aggregate data according to the use case

CHAPTER II – III
DATA MARTS - REASONS



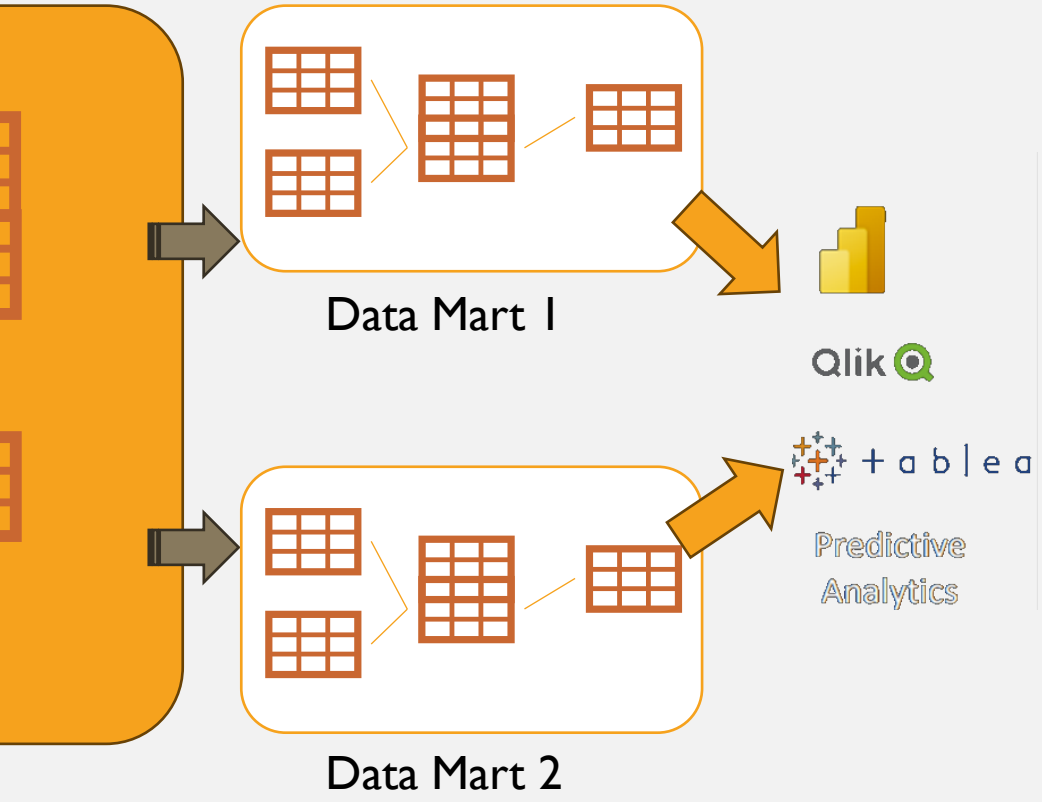
- Increase in user-friendliness
- Facilitation of focusing on relevant data
- Avoidance of overload from working with many tables in the database
- Important for the acceptance of the Data Warehouse or Data Marts, especially for non-technical users

CHAPTER II – III
DATA MARTS - REASONS



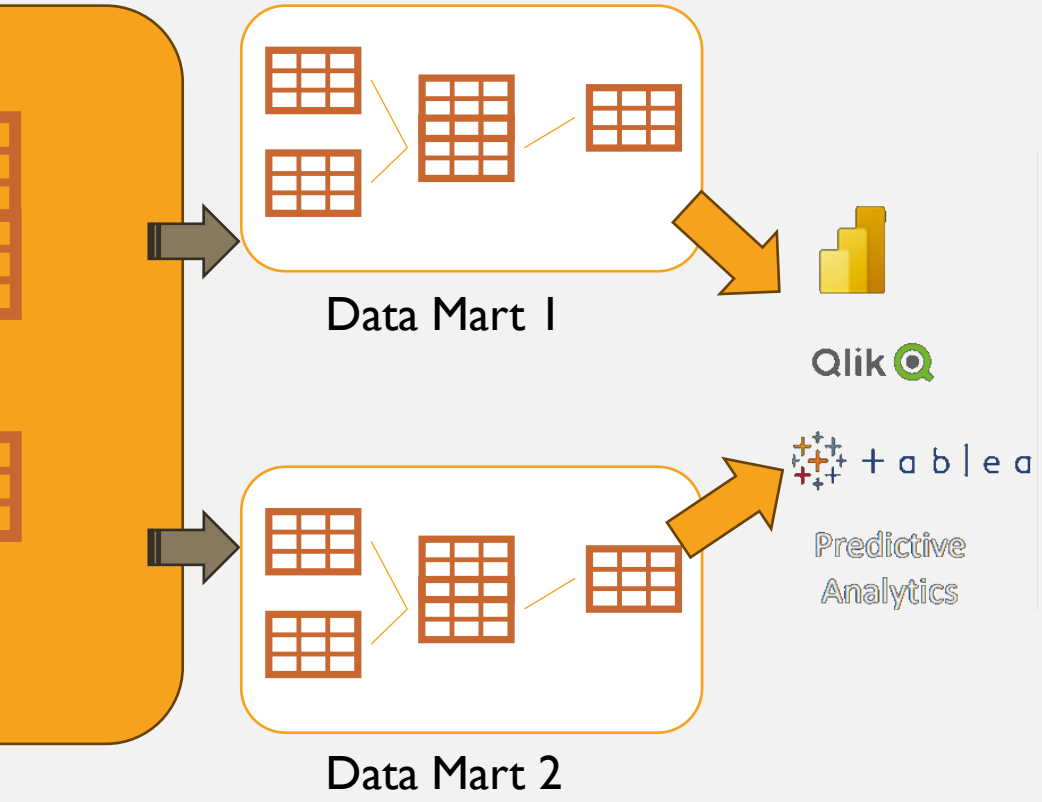
- Improvement of performance
  - Dimensional data modeling enables specific technologies
  - Utilization of in-memory databases with fast query performance
  - Construction of so-called cubes for better performance
- Better user-friendliness and acceptance through increased performance
- Main reason for the use of Data Marts

CHAPTER II – III
DATA MARTS – USE CASES



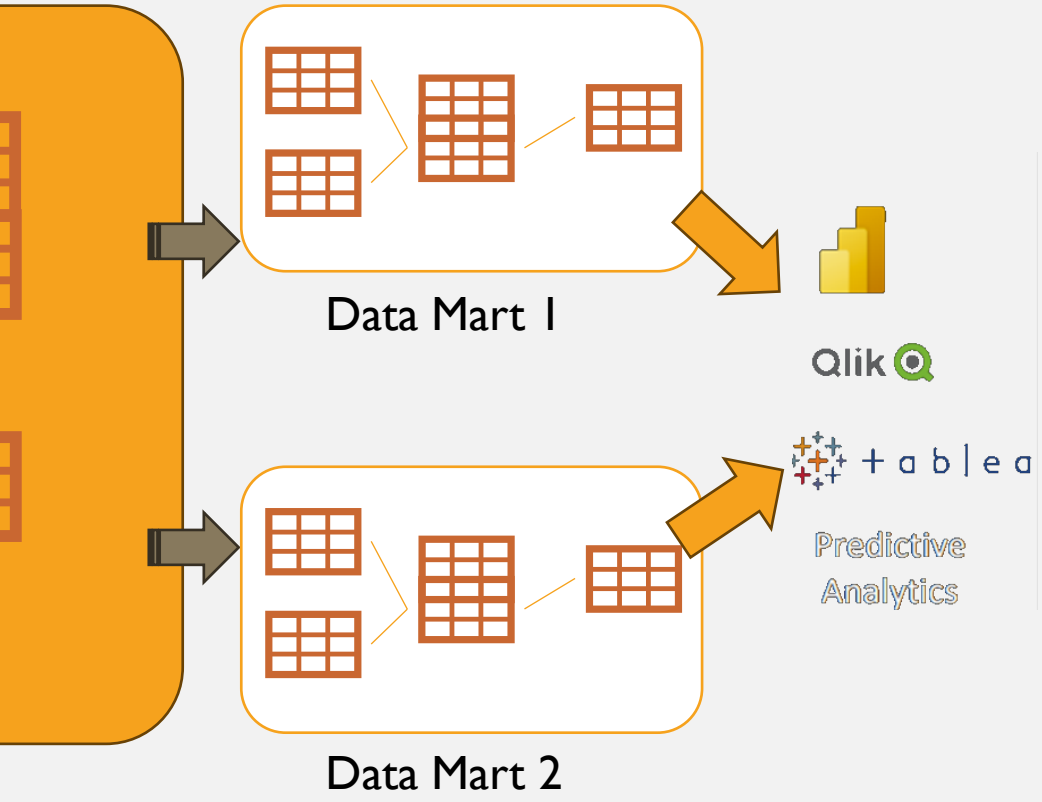
- Usage of different tools with the Data Warehouse
  - Data visualization with Power BI
    - Utilization of in-memory databases for good performance
- Other tools, such as predictive analytics, do not necessarily require in-memory databases

CHAPTER II – III
DATA MARTS – USE CASES

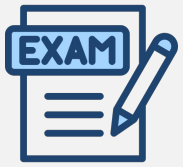


- Need for different Data Marts for various use cases
  - Different departments with different use cases
    - Sales team, finance team, marketing team
  - Storage of all data in the core, but not all relevant for every use case and department
- Setting up Data Marts for different regions is possible
- In summary, Data Marts are built for various use cases, whether for different tools, departments, or regions.

CHAPTER II – III
DATA MARTS – BENEFITS



1. Enhanced usability: Focuses on relevant data, simplifying usage.
2. Reduces complexity: Users avoid dealing with numerous tables in the database.
3. Improved acceptance: Especially beneficial for non-technical business users.
4. Performance boost: Data is modeled dimensionally, enabling faster query processing.
5. Leveraging specific technology: In-memory databases and cubes enhance query speed.
6. Better usability and acceptance: Result from increased performance.



CHAPTER II – IV
RELATIONAL DATABASES

ID	Date	Product
1	25.04.2024	Burger
2	25.04.2024	Pizza
3	26.04.2024	Fries
4	26.04.2024	Soda
5	27.04.2024	Chicken

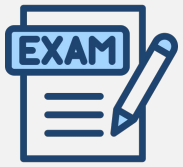
Keys

Tables (relations)

Relational databases

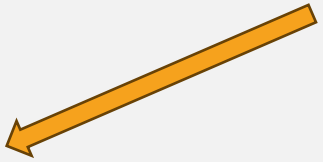
SQL

- Introduction to Data Warehouse Technology
  - Focus on Relational Databases
    - Primary housing for data warehouse
  - Role of Relational Databases
    - Storage of data in tables
    - Tables also termed as relations
    - Data structured into columns and rows
  - Utilization of SQL
    - Query language for accessing data
    - Natural and relatively simple to learn
    - Example: SELECT statement for data retrieval
  - Use of Keys
    - Establishing relations between tables



CHAPTER II – IV
RELATIONAL DATABASES

id	date	product	customer_id	name
1	2024-05-01	Laptop	101	Alice
2	2024-05-02	Tablet	102	Bob
3	2024-05-03	Phone	103	Charlie
4	2024-05-04	Speaker	104	David
5	2024-05-05	Monitor	105	Emily

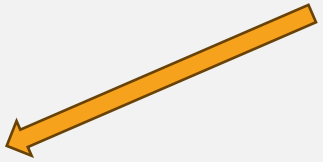


id	name	city
101	Alice	New York
102	Bob	Los Angeles
103	Charlie	Chicago
104	David	Houston
105	Emily	San Francisco

- Specifics of Relational Databases
  - Utilization of Keys and Table Relations
- Primary Key
  - Purpose: Uniquely identifies each row
  - Column designated as primary key
  - Requires unique and non-null values
- Foreign Keys
  - Purpose: Referencing another table
  - Contains values from primary key of another table
  - Facilitates reference to specific rows in other tables
- Querying and Joining Tables
  - Joins combine results of multiple tables
  - Enables combining columns from different tables in queries

CHAPTER II – IV
RELATIONAL DATABASES

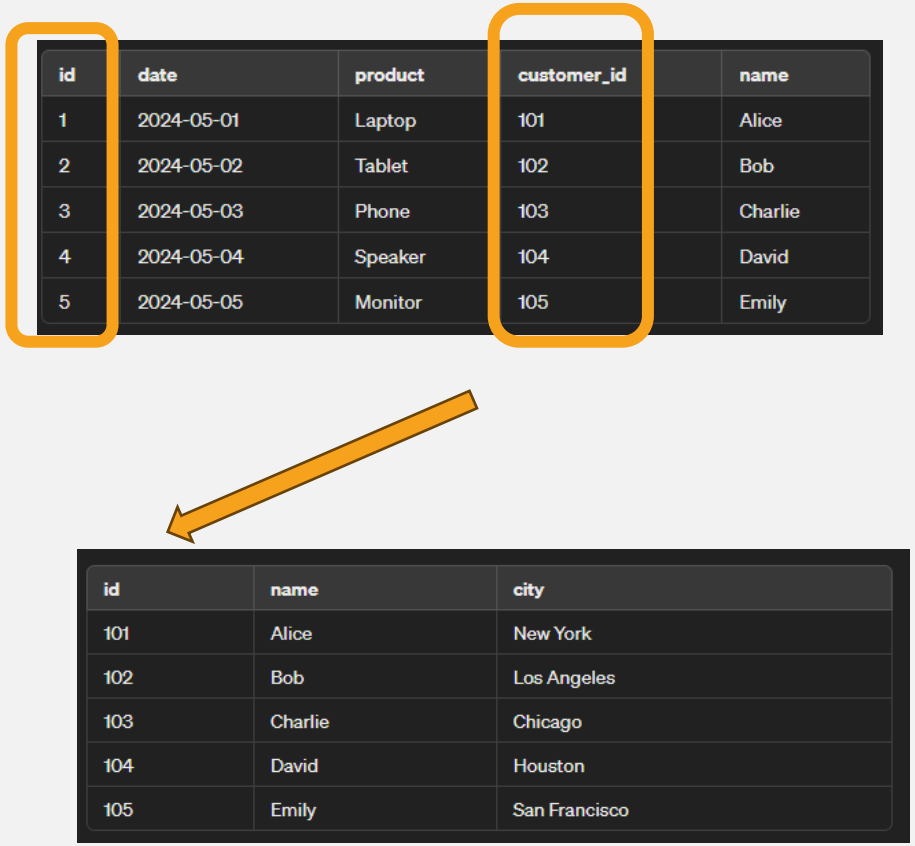
id	date	product	customer_id	name
1	2024-05-01	Laptop	101	Alice
2	2024-05-02	Tablet	102	Bob
3	2024-05-03	Phone	103	Charlie
4	2024-05-04	Speaker	104	David
5	2024-05-05	Monitor	105	Emily



id	name	city
101	Alice	New York
102	Bob	Los Angeles
103	Charlie	Chicago
104	David	Houston
105	Emily	San Francisco

- Significance of Relational Databases
  - Game changer for data analysis
  - Development of logic and algorithms for query performance
  - Transition from querying single tables to analyzing multiple tables
- Advancement of OLAP
  - Enhancement of analysis capabilities
- Connection to Data Warehouses
  - Organization of data in multiple tables
  - Facilitation of data analysis through table relations
- Modeling of Tables
  - Utilization of star schemas for table modeling

CHAPTER II – IV
RELATIONAL DATABASES



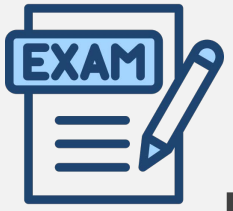
id	date	product	customer_id	name
1	2024-05-01	Laptop	101	Alice
2	2024-05-02	Tablet	102	Bob
3	2024-05-03	Phone	103	Charlie
4	2024-05-04	Speaker	104	David
5	2024-05-05	Monitor	105	Emily

id	name	city
101	Alice	New York
102	Bob	Los Angeles
103	Charlie	Chicago
104	David	Houston
105	Emily	San Francisco

- Overview of Relational Database Products
  - Relational Database Management Systems (RDBMS)
    - Examples: PostgreSQL, Oracle, Microsoft SQL Server
    - Commonly used in enterprises for database management
    - Open source alternatives: PostgreSQL, MySQL
    - Cloud services: Amazon Relational Databases, Azure SQL databases
- Introduction to In-Memory Databases
  - Growing importance in modern data management
  - Next topic of discussion in upcoming lecture

CHAPTER II – V
IN-MEMORY DATABASES - INTRO

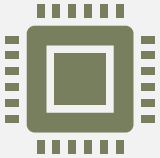
- Highly optimized for query performance
- Common Use Cases
  - Analytical purposes
  - High query volume scenarios
- Application in Data Marts
  - Access layer for users or applications
  - High query performance essential for user experience
- Technology Independence
  - Suitable for both relational and non-relational data structures



CHAPTER II – V
IN-MEMORY DATABASES – TRADITIONAL DATABASES VS. IN-MEMORY DATABASES

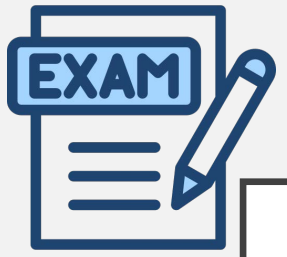
Databases

Hard drive



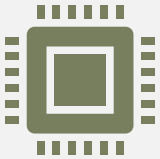
In-memory

- Storage Mechanism in Traditional Databases
  - Data stored on hard drives or solid-state disks
  - Data loaded into memory when queried, resulting in response time
- Limitations of Traditional Approach
  - Response time from disk to memory contributes to query delay
  - Suboptimal for high query performance requirements
- Advantages of In-Memory Databases
  - Elimination of disk-based storage
  - Entire data stored in memory, reducing response time
  - Significant improvement in query performance



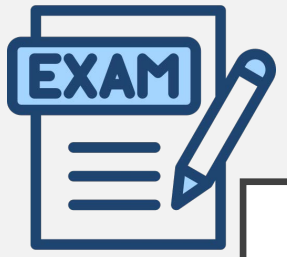
CHAPTER II – V
IN-MEMORY DATABASES – TRADITIONAL DATABASES VS. IN-MEMORY DATABASES

Databases



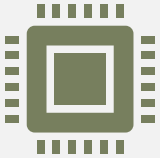
In-memory

- Advantages of In-Memory Databases
  - Elimination of disk-based storage
  - Entire data stored in memory, reducing response time
  - Significant improvement in query performance
- Technology and Methods in In-Memory Databases
  - Different algorithms and methods utilized
  - Example: Columnar storage, scanning data by columns
  - Potential use of parallel query plans for faster processing
- Simplified Explanation
  - In-memory databases eliminate response time from disk loading
  - Result: Enhanced query performance without technical intricacies



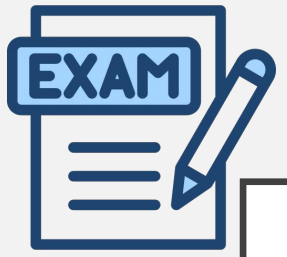
CHAPTER II – V
IN-MEMORY DATABASES – CHALLENGES OF IN-MEMORY DATABASES

Databases



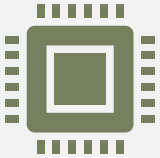
In-memory

- Durability of Data
  - Essential requirement for databases
  - In-memory storage susceptible to data loss during power disconnection or reset
  - Solutions for Durability
    - Creation of snapshots or images representing database state
    - Storage of data on disks before updates or restarts
    - Ensures availability of data even after disruptions
- Cost Considerations
  - Expensive technology despite hardware advancements
  - Increasing data volume outpaces reduction in hardware costs
  - Consideration of cost-effectiveness in implementing in-memory databases



CHAPTER II – V
IN-MEMORY DATABASES – CHALLENGES OF IN-MEMORY DATABASES

Databases

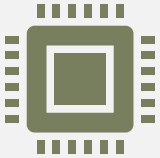


In-memory

- Advancements in Traditional Databases
  - Optimization efforts to reduce disk usage
  - Improvements in query performance
- Strategic Use of In-Memory Databases in Data Marts
  - Load only relevant data for specific use cases
  - Cost-effective approach to leverage in-memory databases

CHAPTER II – V
IN-MEMORY DATABASES – TECHNOLOGIES IN-MEMORY DATABASES

Databases

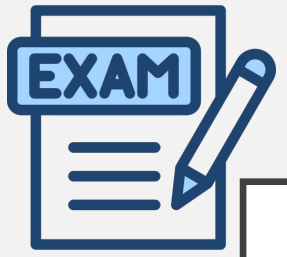


In-memory

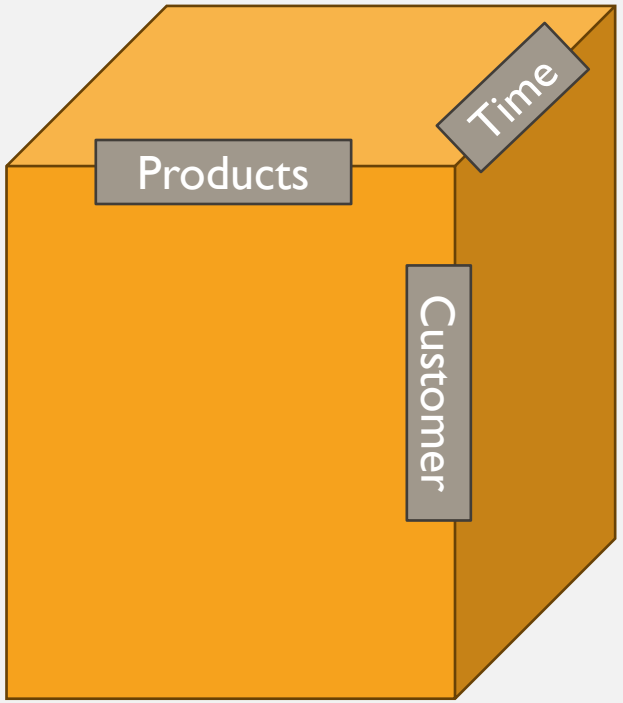
- Examples of In-Memory Database Technologies
  - SAP Hana, Microsoft Secure In-memory Tables, Oracle
  - Cloud services: Amazon Memory DB

CHAPTER II – VI –
OLAP CUBES –
TRADITIONAL DATA WAREHOUSE VS. CUBES

- Data Storage in Relational Databases
  - Organized into tables with relations
- Cube Structure
  - Non-relational organization into dimensions
  - Absence of table relations
- Multidimensional Dataset
  - Referred to as MOLAP (Multidimensional Online Analytical Processing)
  - Data stored in arrays instead of tables
- Importance of Cubes
  - Analytical Purposes
  - Fast query performance
  - Exclusive use in data marts
- Utilization in BI Solutions
  - Integration with various software
  - Example: Excel integration for analysis

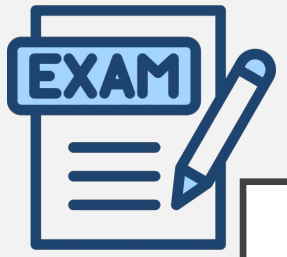


CHAPTER II – VI – OLAP CUBES – THE CUBE VIEW

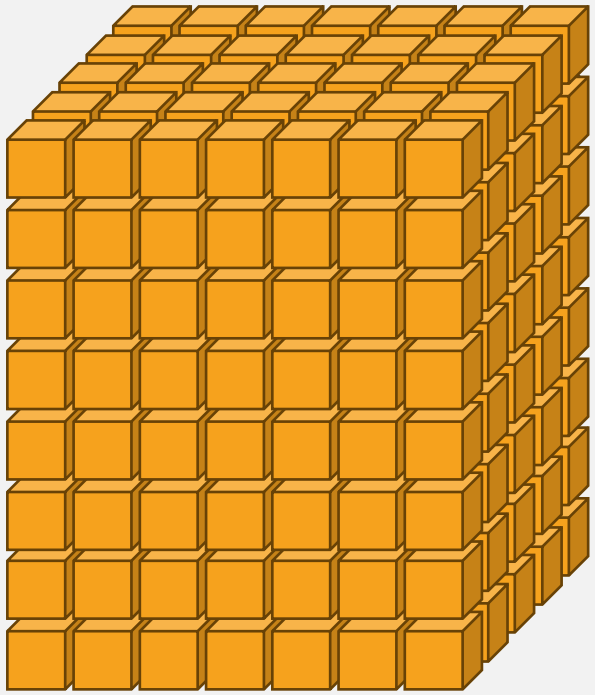


id	date	product	customer_id	name
1	2024-05-01	Laptop	101	Alice
2	2024-05-02	Tablet	102	Bob
3	2024-05-03	Phone	103	Charlie
4	2024-05-04	Speaker	104	David
5	2024-05-05	Monitor	105	Emily

- Visualization of Data Organization
  - Cubes organize data into multiple dimensions
  - Typically represented with three dimensions for simplicity
- Example Scenario: Sales Analysis
  - Dimensions: Products, Time, Customers
  - Measurement: Sales

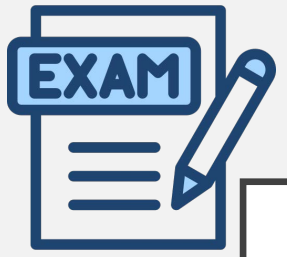


CHAPTER II – VI – OLAP CUBES – THE CUBE VIEW

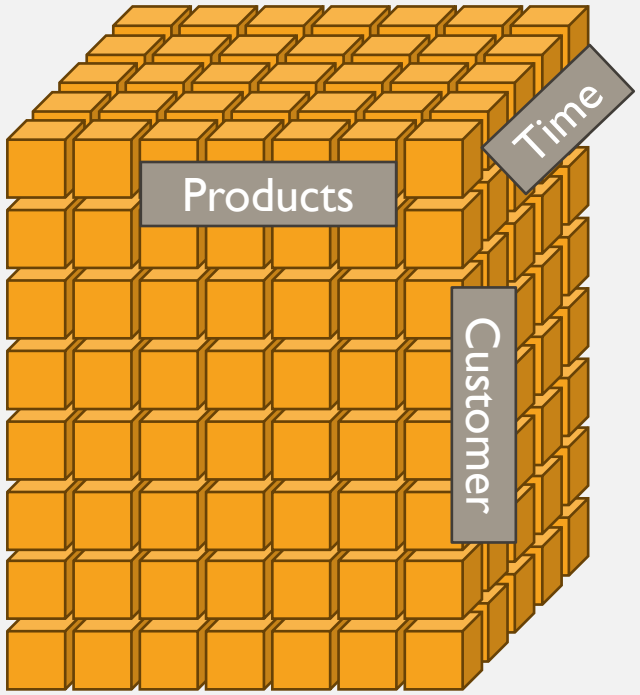


id	date	product	customer_id	name
1	2024-05-01	Laptop	101	Alice
2	2024-05-02	Tablet	102	Bob
3	2024-05-03	Phone	103	Charlie
4	2024-05-04	Speaker	104	David
5	2024-05-05	Monitor	105	Emily

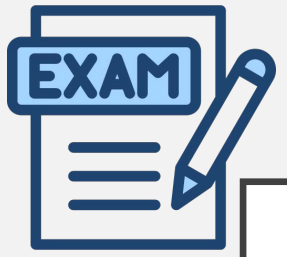
- Slicing and Dicing of Data
  - Using arrays and cells to manipulate data
  - Example: Analyzing sales for specific customers in certain months
- Pre-calculated Data
  - Values in cells are pre-calculated
  - Enables instant access and visualization of data
- Benefits of Pre-calculated Values
  - Facilitates quick data retrieval and visualization
  - Aggregated in a way suitable for reports and applications



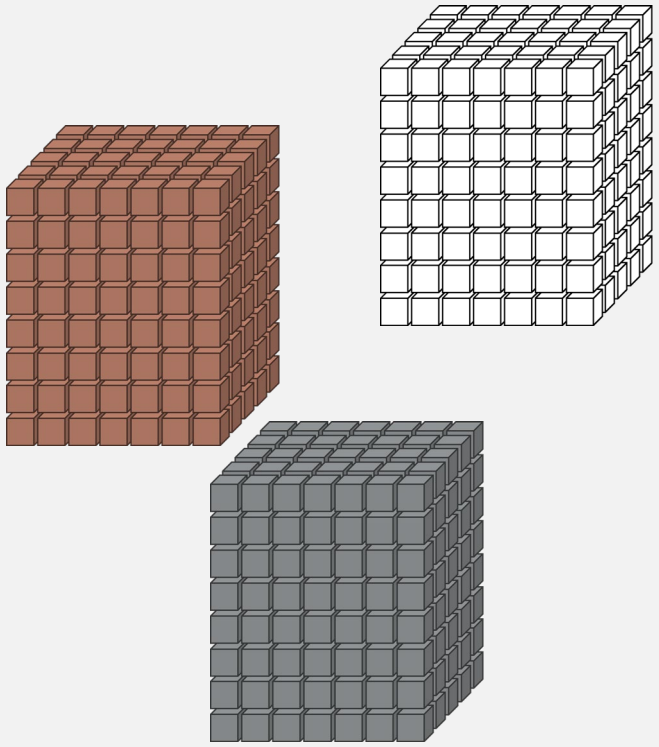
CHAPTER II – VI – OLAP CUBES – TECHNOLGIES



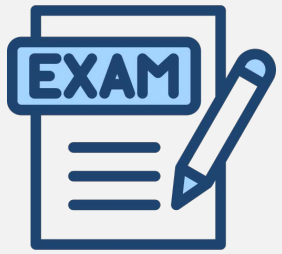
- MDX Language in Cube Technology
  - Alternative to SQL for querying cube data
  - MDX: Multidimensional Expression
    - Developed by Microsoft
    - Most commonly used query language for cubes
- Purpose of Cubes
  - High performance due to pre-calculated values
  - Main benefit in interactive tools with hierarchies
  - Enables efficient drilling, slicing, and dicing of data
- Storage of Data
  - Multidimensional databases instead of relational databases
  - Utilizes different hardware for storage and processing



CHAPTER II – VI – OLAP CUBES – ALTERNATIVE TECHNOLOGIES



- In-Memory Databases
  - Advancements leading to better performance
  - Decreasing significance of cubes
- Tabular Models
  - Utilized, particularly by Microsoft
  - Features columnar storage, parallel processing
  - Provides alternatives to traditional cube usage
- Evolving Technologies
  - Continuous improvements making cubes less vital
  - Better hardware and storage methods contributing to alternatives
- Relational Databases
  - Improved performance in modern setups
  - Viable option without resorting to cubes
- Considerations for Data Marts
  - Selection of methods based on query performance requirements



QUESTIONS CHAPTER II

WHAT IS THE MAIN REASON FOR IMPROVED QUERY PERFORMANCE IN CUBES?

- Less latency due to multi-dimensional approach
- Better query optimizer
- Precalculated (aggregated) values

WHAT IS THE KEY IDEA FOR IMPROVING PERFORMANCE WITH IN-MEMORY DATABASES?

- Defining multiple dimension and precalculating the relevant values.</p>e
- Eliminating response time from disc by processing all data directly in memory.
- Optimizing queries with better query optimizers

QUESTIONS CHAPTER II – V – VI

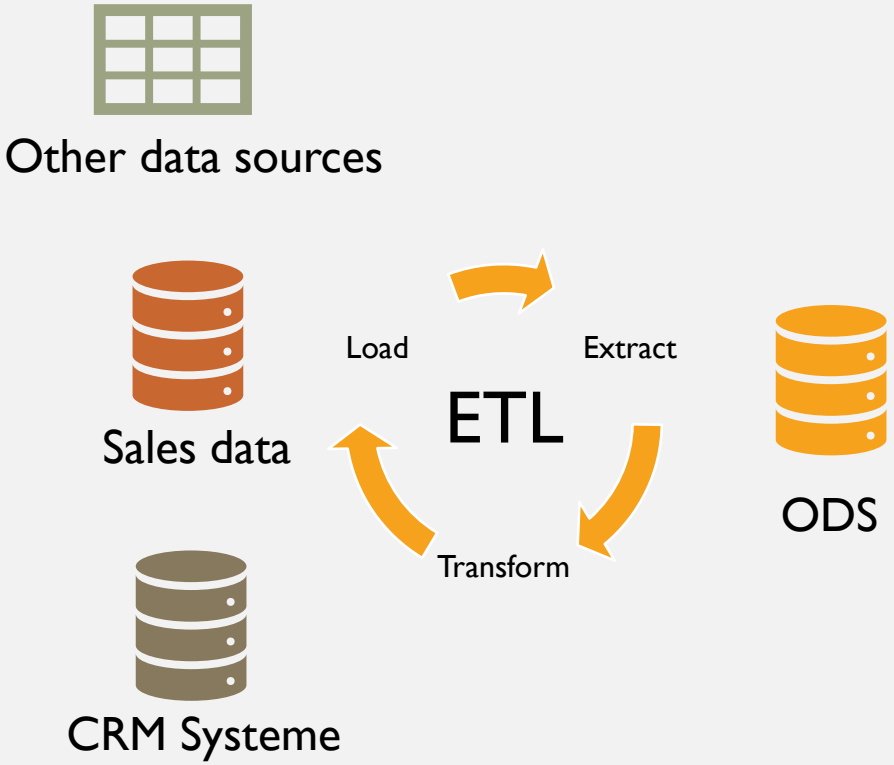
WHAT IS THE MAIN REASON FOR IMPROVED QUERY PERFORMANCE IN CUBES?

- Less latency due to multi-dimensional approach
- Better query optimizer
- **Precalculated (aggregated) values**

WHAT IS THE KEY IDEA FOR IMPROVING PERFORMANCE WITH IN-MEMORY DATABASES?

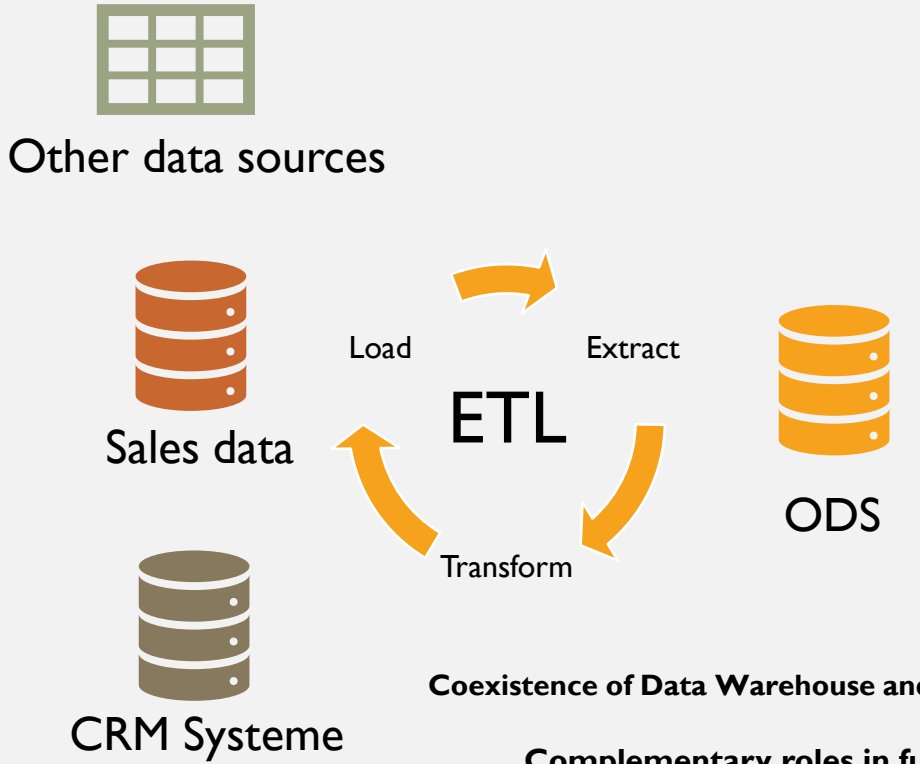
- Defining multiple dimension and precalculating the relevant values.</p>e
- **Eliminating response time from disc by processing all data directly in memory.**
- Optimizing queries with better query optimizers

CHAPTER II – VII – ODS (OPERATIONAL DATA STORAGE) – OVERVIEW



- Similarity to Data Warehouse
  - Integration of data from various operational systems
- Purpose
  - Consolidation of important data from operational systems
- Integration Process
  - Utilization of ETL (Extract, Transform, Load)
  - Data integrated into a single database
- Difference from Data Warehouse
  - Usage for operational decision making
  - Contrasting requirements and processes
- Operational Decision Making
  - Focus on quick, tactical decisions
  - Not primarily for analytical or strategic purposes

CHAPTER II – VII – ODS (OPERATIONAL DATA STORAGE) – REQUIREMENTS

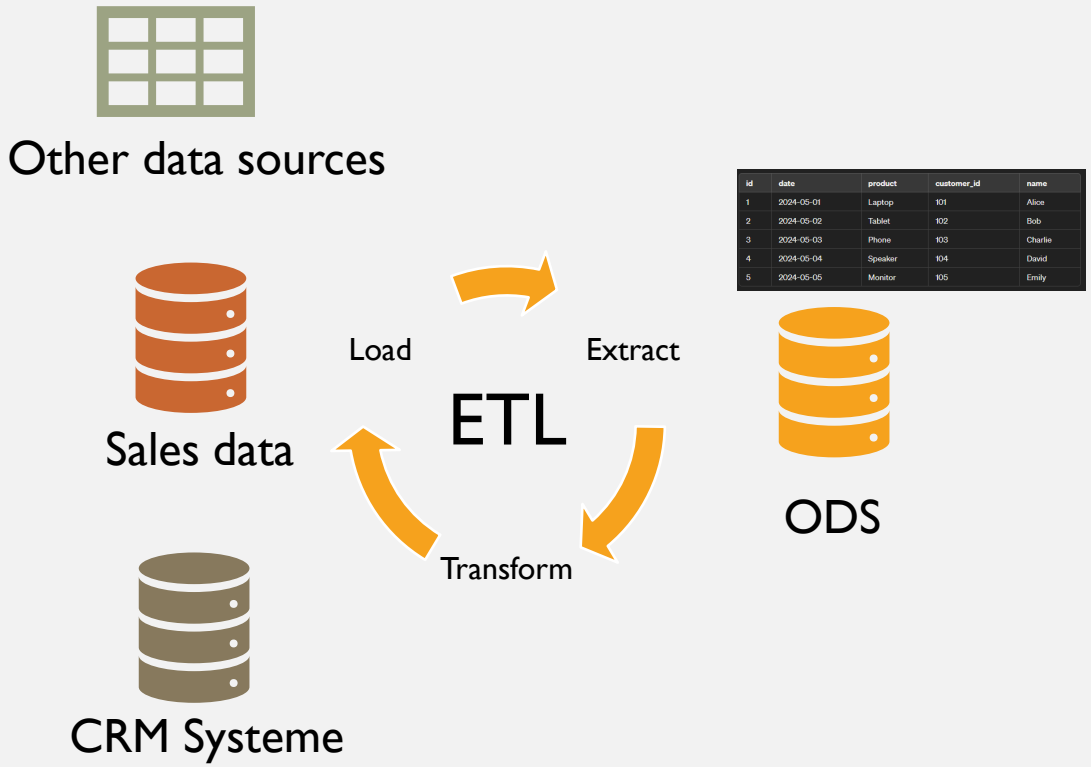


Coexistence of Data Warehouse and ODS

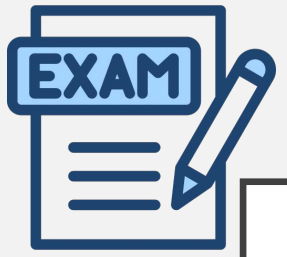
Complementary roles in fulfilling different decision-making needs in an organization

- Focus on Operational Decisions
  - No requirement for extensive historical data
  - Emphasis on current state of data
- Timeliness of Data
  - Need for immediate reflection of data from source systems
  - Importance of near-real-time data updates
- Importance of Data Accuracy
  - Avoidance of operational decisions based on outdated or incorrect data
- Contrast with Data Warehouse
  - Differentiated by data update frequency and historical data retention
  - Data warehouse for strategic decisions, ODS for operational decisions

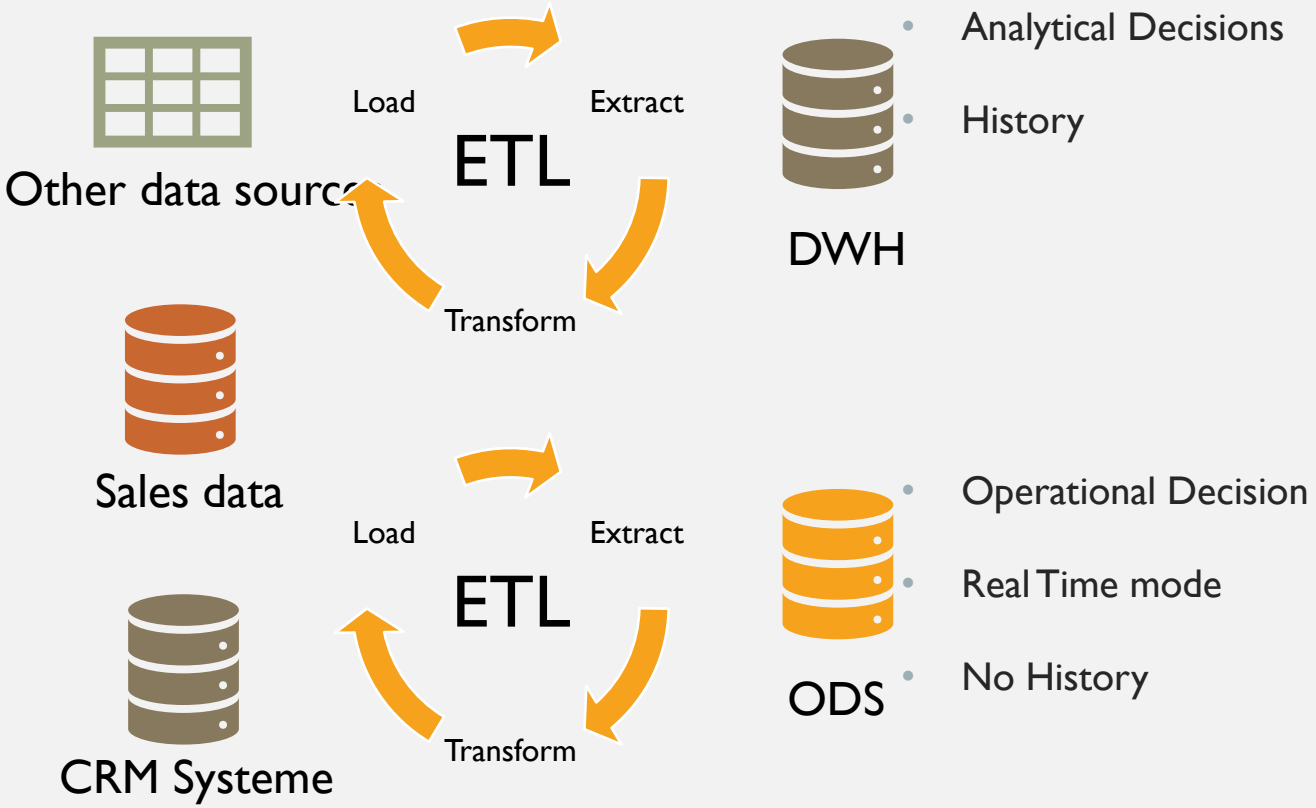
CHAPTER II – VII – ODS (OPERATIONAL DATA STORAGE) – EXAMPLE (USE CASE)

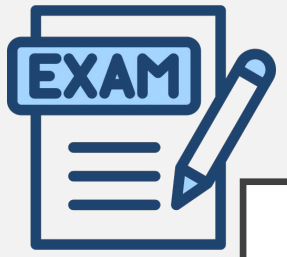


- Scenario: Financial Service Company
  - Customers engage in various activities: ETFs, stocks, cryptocurrencies, account balance
- Integration of Data from Different Systems
  - Multiple systems for crypto, stock trading, etc.
  - Need to consolidate overall customer balance across systems
- Importance of Timely Data Reflection
  - Immediate availability of combined customer balance for operational decisions
  - Decision-making for customer credit assessments, for instance
- Focus on Operational Decision Making
  - Requirement for accurate, current data from operational systems
  - Not for strategic analysis but for immediate operational decisions
- Data Update Approach
  - ETL or real-time data feed for near-real-time updates
  - Data replacement or update rather than appending for history

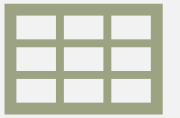


CHAPTER II – VII – ODS (OPERATIONAL DATA STORAGE) – ODS AND/OR DWH





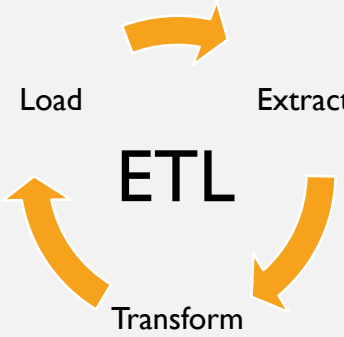
CHAPTER II – VII – ODS (OPERATIONAL DATA STORAGE) – SEQUENTUAL ODS



Other data sources

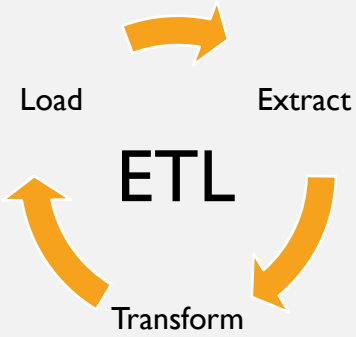
Sales data

CRM Systeme



ODS

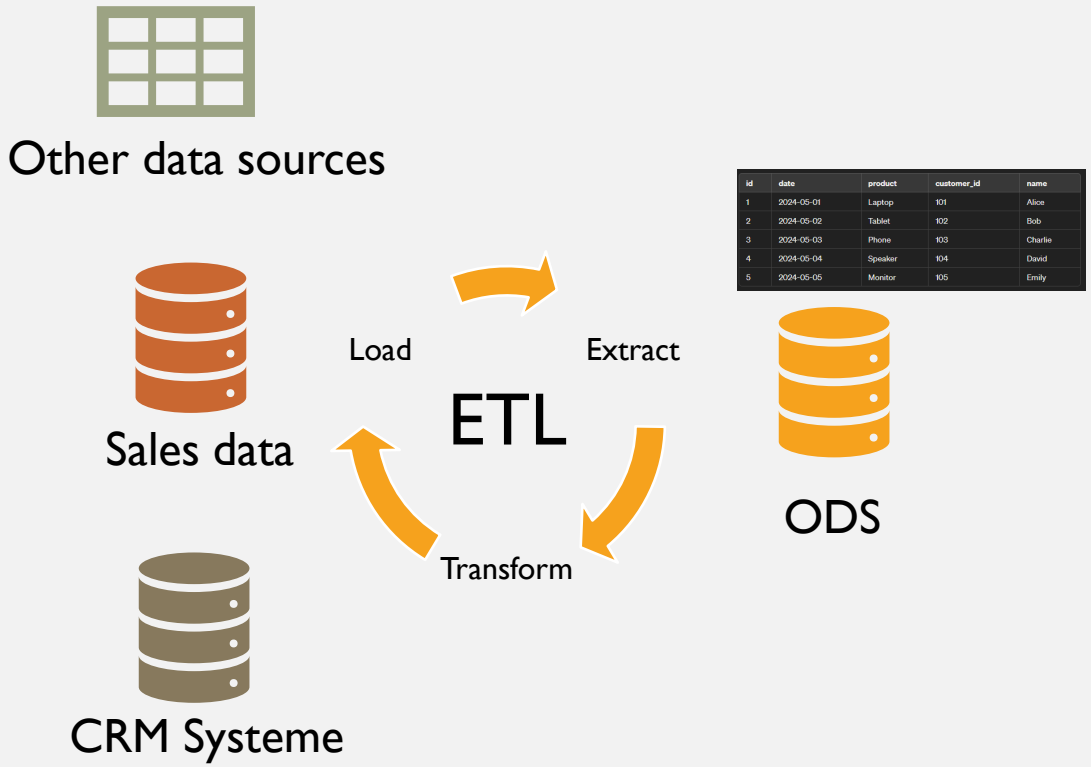
- Operational Decision
- Real Time mode
- No History



DWH

- Analytical Decisions
- History

CHAPTER II – VII – ODS (OPERATIONAL DATA STORAGE) – EXAMPLE (USE CASE)



id	date	product	customer_id	name
1	2024-05-01	Laptop	101	Alice
2	2024-05-02	Tablet	102	Bob
3	2024-05-03	Phone	103	Charlie
4	2024-05-04	Speaker	104	David
5	2024-05-05	Monitor	105	Emily

- Decreasing Relevance of ODS
  - Improved Hardware Performance
    - Faster data loading reduces need for ODS
  - Emergence of Alternative Technologies
    - Other solutions offer real-time or rapidly updated data
    - Diminishing need for traditional ODS
- Uncommon Implementation
  - Increasing rarity of ODS in companies
  - Shift towards alternative data management approaches
- Pragmatic Approach to Terminology
  - Focus on practical usage rather than terminology
  - Utilize existing ODS if available in the company
    - Potential integration into ETL processes
  - Avoid fixation on definitions, prioritize functionality and utility
- Summary of Architectural Concepts
  - Recap of discussed data warehouse architectures
  - Consolidation of key learnings in upcoming lecture