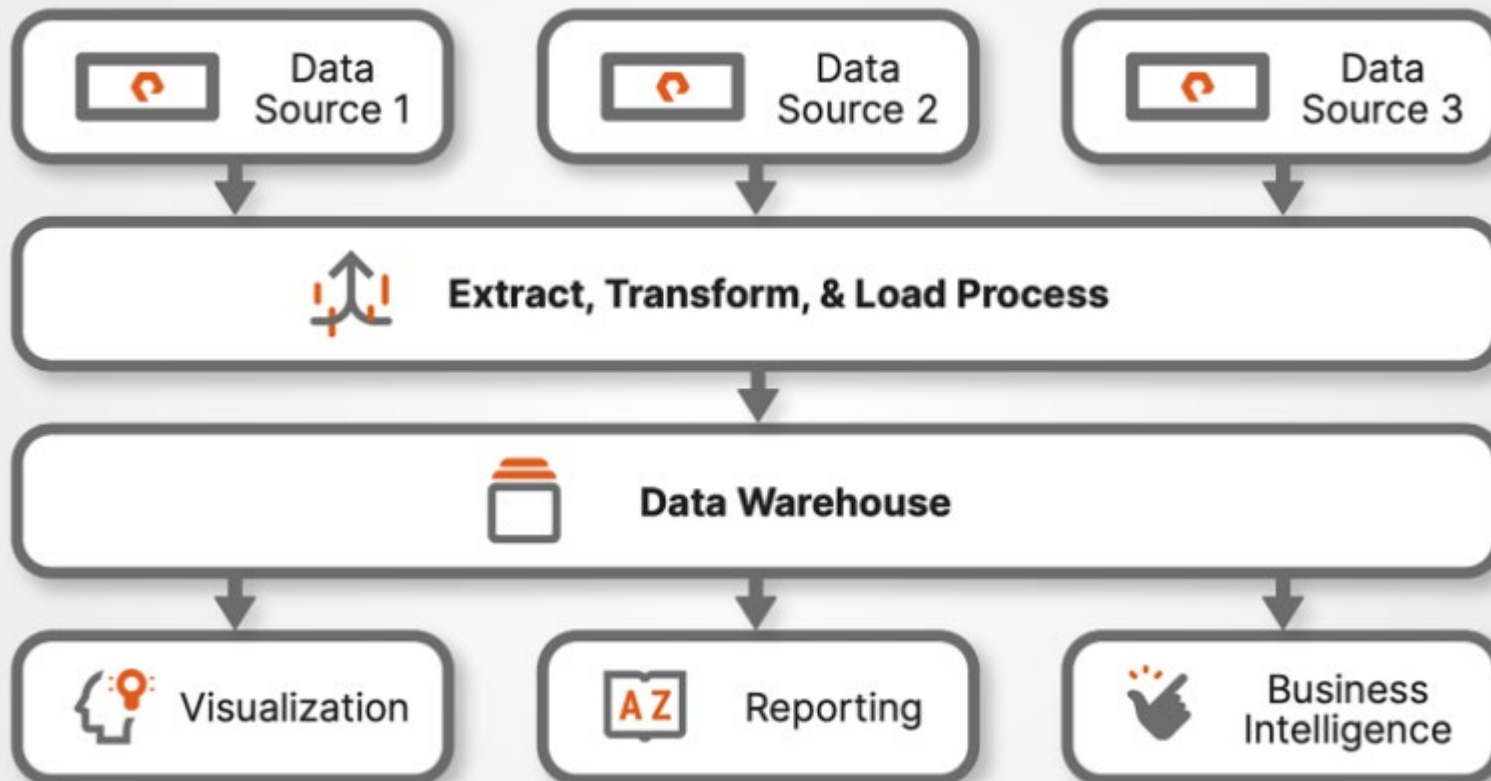
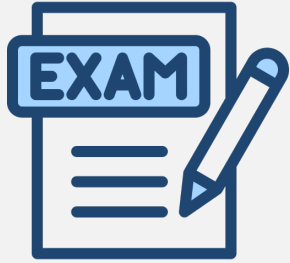


# DATA WAREHOUSE ETL PROCESS & TOOLS

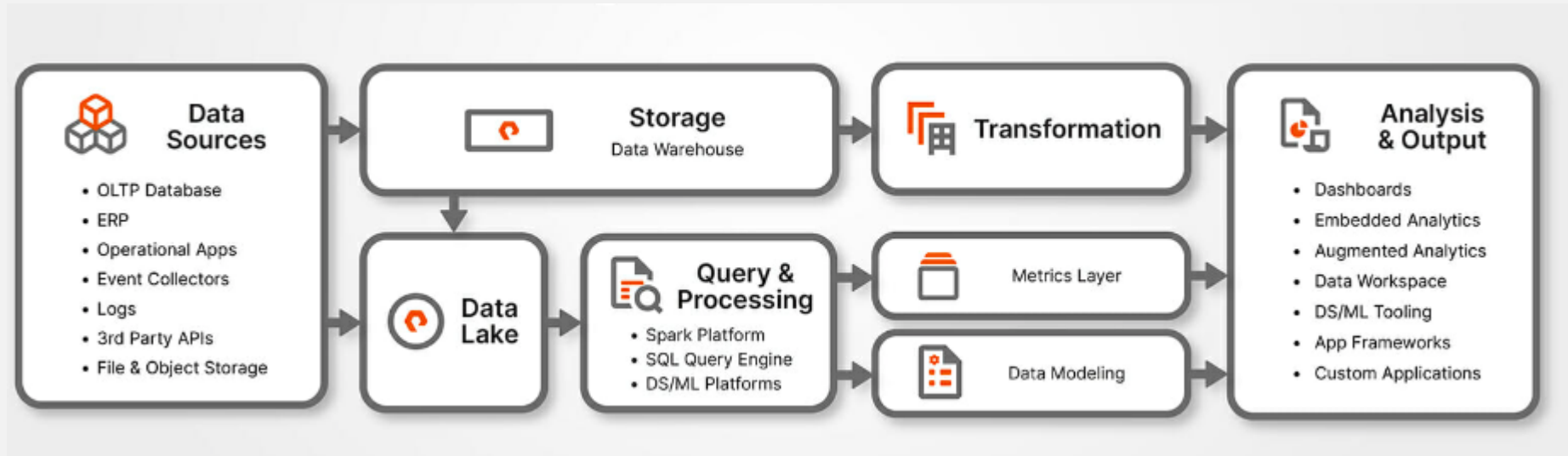
## Lecture 6

# ETL OVERVIEW





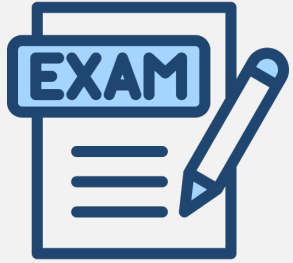
# DATA WAREHOUSE EXAMPLE



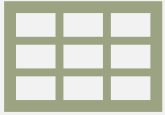
## Motivation

---

- Is ETL is Interesting area?  
70 to 80% BI(DI or DW) projects is reliable **ETL process**
- Let's have a look on the DW & DI market size
  - In 2003, DI was **USD 9.3 billion market**
  - In 2008, DI was **USD 13 billion market**
  - By 2015, yearly grow estimated to USD 20 Billion
- The **more systems in the world, the more work in Data Integration!**



# ETL INTRODUCTION (RECAP)



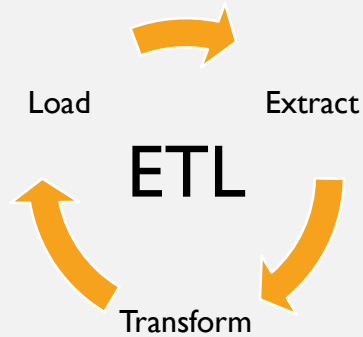
Other data sources



Sales data

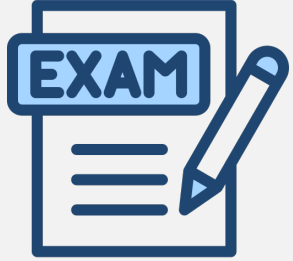


CRM Systeme



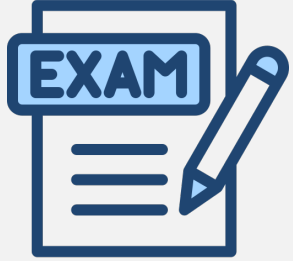
Data Warehouse

- ETL Process:Data Sources:
  - Various sources for the data warehouse.
- ETL Steps:
  - Extraction:
    - Extract data from sources.
  - Transformation:
    - Clean data.
    - Transform data.
  - Loading:
    - Integrate data into the data warehouse.



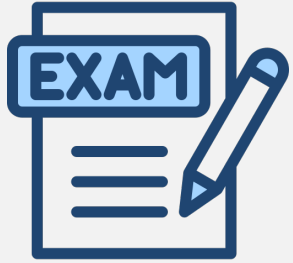
## OLAP/DSS/DWH

- Objective of OLAP database to process data as Quickly as possible with less complexity
- It is use for decision making purpose
- Used by Management People



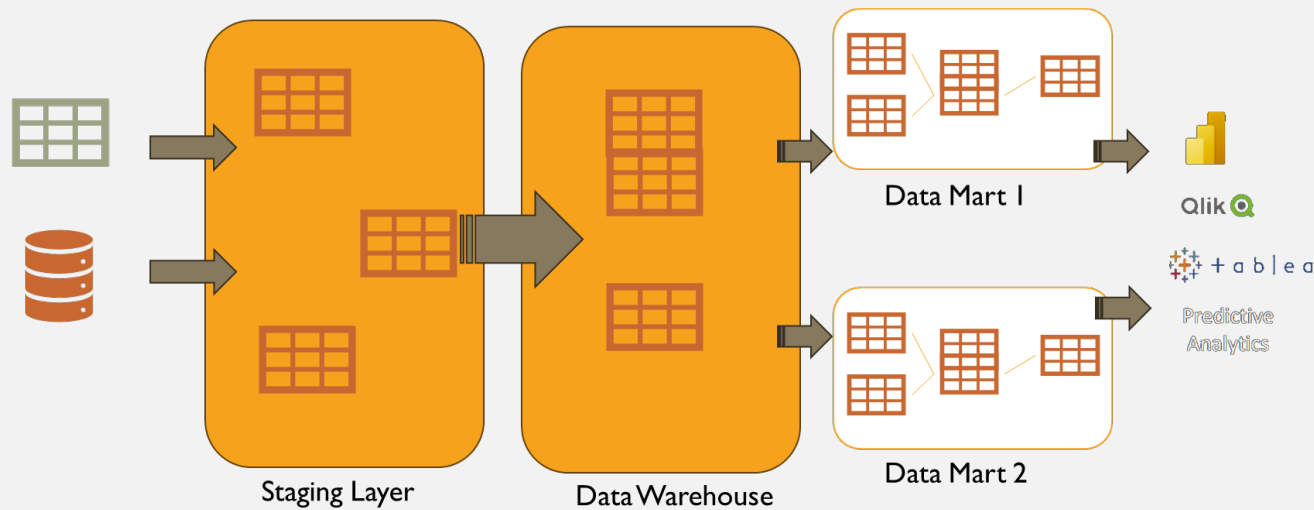
## OLTP

- Objective of OLTP is to process data as quickly as possible
- Support client Server technology
- Support Large Amount of Data
- Data is secure



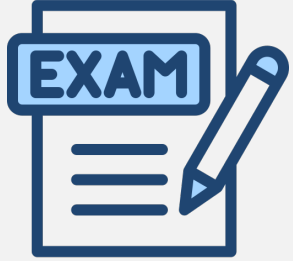
# ETL IN DATA WAREHOUSE QUESTION?

## Extract, Transform, Load



- Data Warehouse Layers
  - Staging Layer
    - Initial extraction of data into tables.
    - Minor cleaning and extraction of relevant information.
  - Core Layer
    - Transformation of staging data.
    - Dimensional modeling takes place.
    - Transformed data loaded into the core.
- Data Marts
  - Tailored to specific use cases.
  - Improves data value, ease of use, and performance.





# WHY IS ETL (SYSTEM) IMPORTANT?

- Adds **value to data**
  - Removes mistakes and corrects data
  - Documented measures of confidence in data
  - Captures the flow of transactional data
  - Adjusts data from multiple sources to be used together (conforming)
  - Structures data to be usable by BI tools
  - Enables subsequent business / analytical data processing

## ETL TOOL: TRUE DATA INTEGRATION



True Data Integration is agnostic of source or target application

**ETL is a bridge** for bi-directional flow

# ETL TOOLS ENVIRONMENT

## ELT / ETL



## Streaming / CDC



## Open Source ETL / ETL



## Open Source Streaming / CDC



## Automation / iPaaS



## End-To-End / Analytics

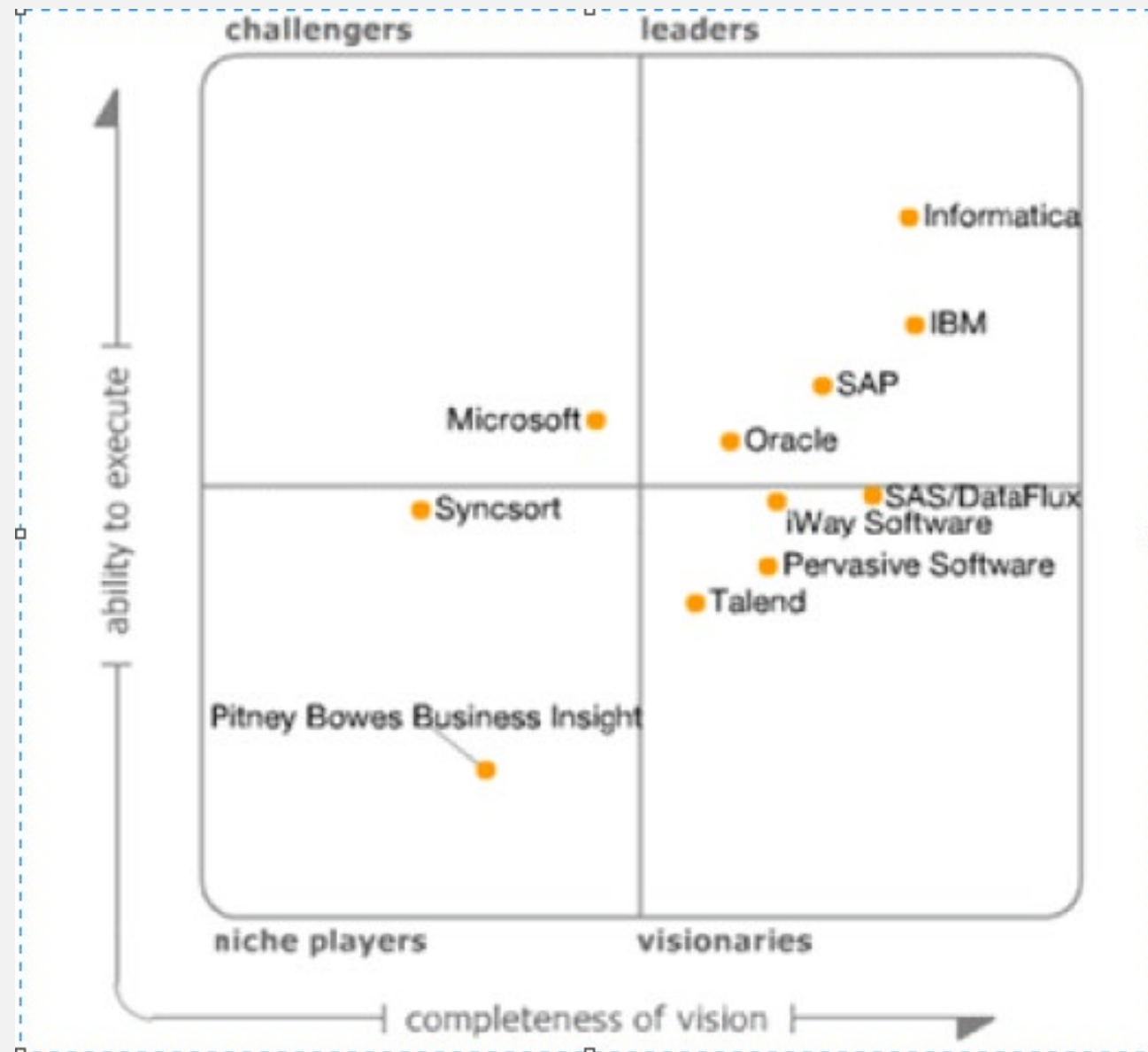


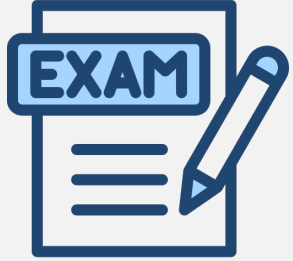
## Vertical Specific Integration Solutions



## Data Collection / Creation







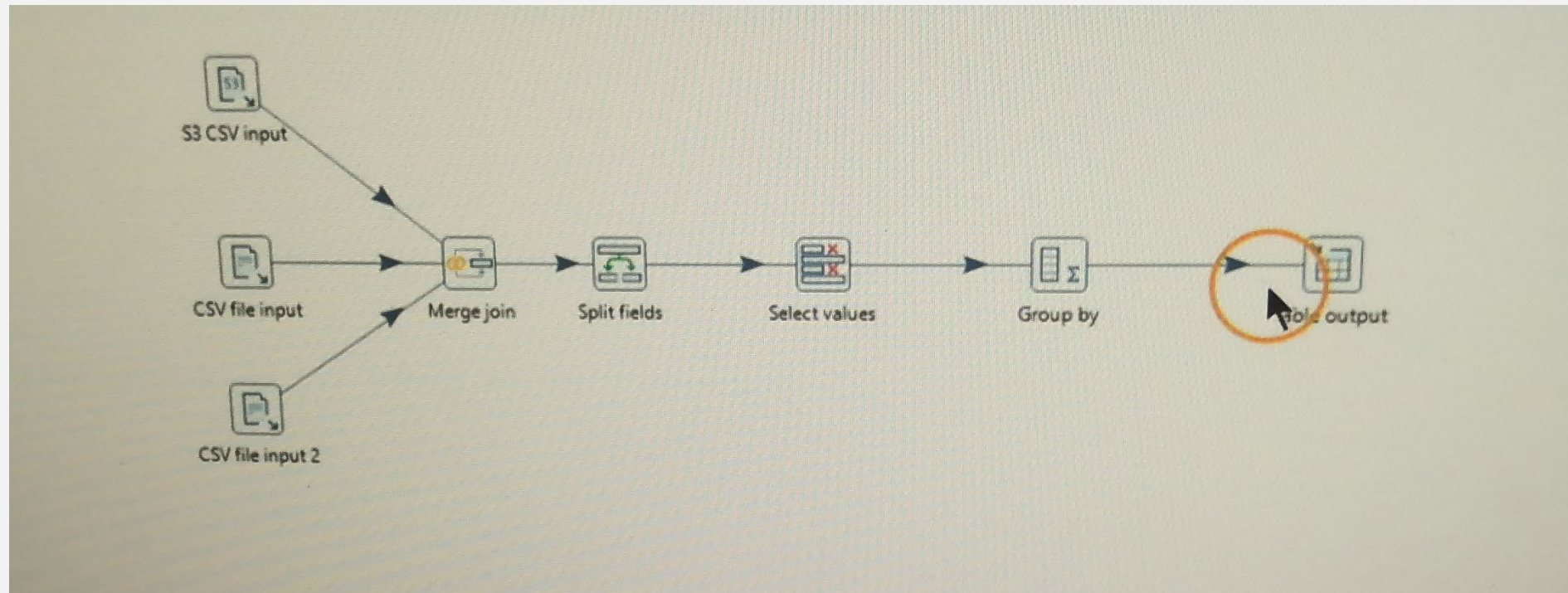
## IMPLEMENTATION IN PRACTICE ETL TOOL

- ETL Tools:
  - Variety of ETL tools available.
  - Discussion of important tools will follow later.
- ETL Tool Capabilities:
  - Connect to Different Data Sources:
    - Extract data from various sources.
  - Transform Data:
    - Change data types.
    - Add additional columns.
    - Clean data.
    - Model and restructure data.
- Load Data:
  - Write data back in different formats.
  - Primarily into databases (data warehouse).
  - Many other options available.
- ETL Tool Features:
  - Thousands of different tools and possibilities.
  - Typically use only a small subset.
- Objective:
  - Load data into a data warehouse.
- Essential for building the data warehouse.

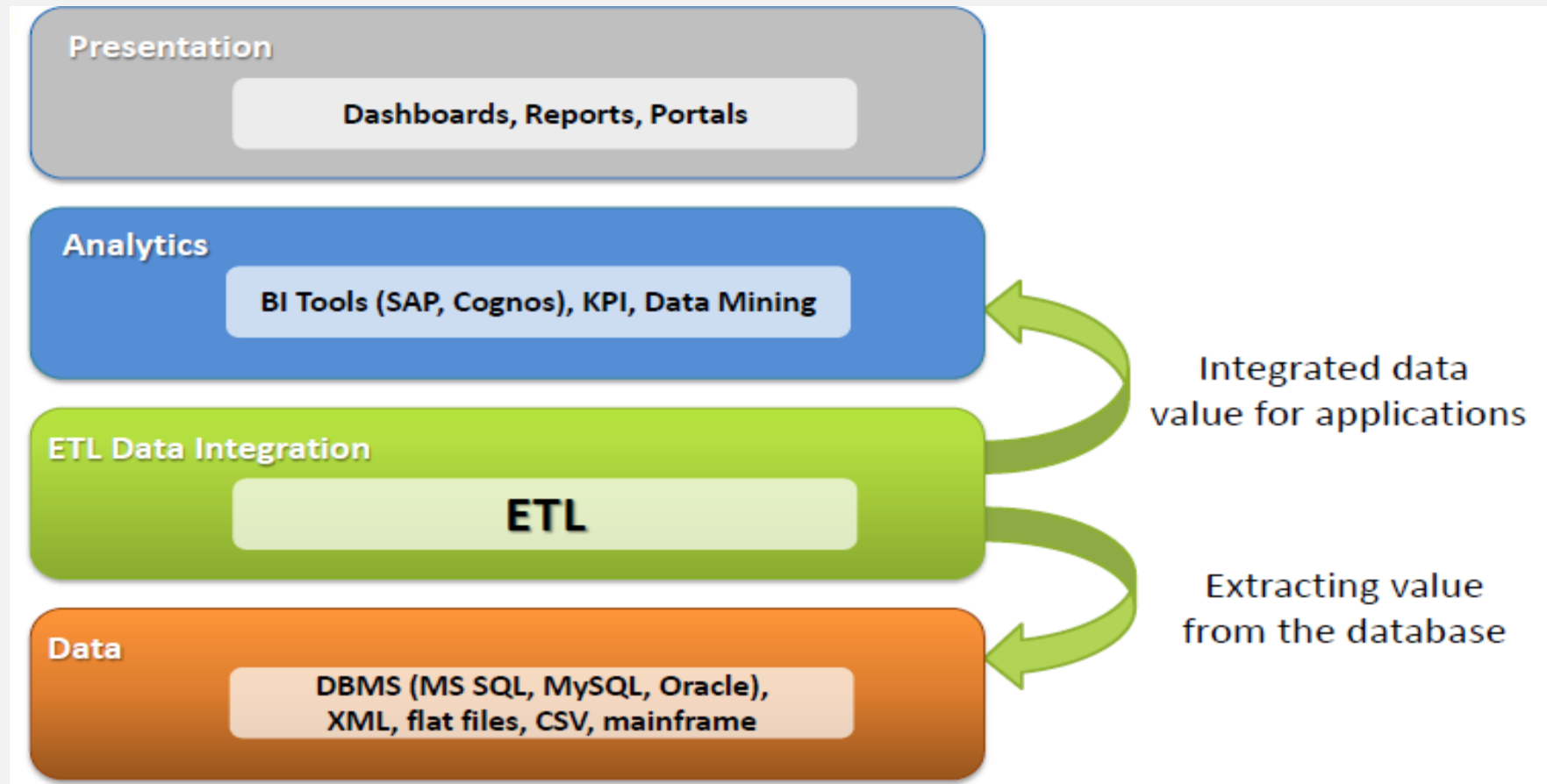
# ETL PROCESS

- Staging Layer Workflow:
  - Separate staging schema in the database.
  - Possible to have separate databases for each layer, but usually handled via different schemas.
- Other Workflows:
  - Core layer workflow.
  - Data mart workflow.
- Common Default Strategy:
  - Separate workflows for staging, core, and data mart layers.
- Workflow Scheduling:
  - Workflows are scheduled using jobs.
  - Jobs run workflows based on defined rules (specific times and frequencies).

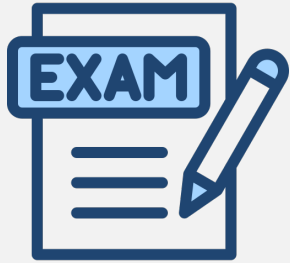
# WORKFLOW EXAMPLE



# ETLPROCESS







## ETL DATA INTEGRATION SOLUTIONS



### Data Migration

Process of transferring data between storage types or formats. An *automated migration* frees up human resources from tedious tasks. Design, extraction, cleansing, load and verification are done for moderate to high complexity jobs.



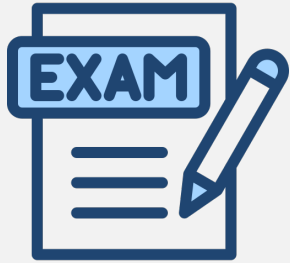
### Data Consolidation

Usually associated with moving data from remote locations to a central location or combining data due to an acquisition or merger.



### Data Integration

Process of combining data residing at different sources and providing a unified view. Emerges in both commercial and scientific fields and is focus of extensive theoretical work. Also referred to as *Enterprise Information Integration*.



### Master Data Management

Processes and tools to define and manage non-transactional data. Provides for collecting, aggregating, matching, consolidating, quality-assuring, persisting and distributing data to an organization to ensure consistency and control.



### Data Warehouse

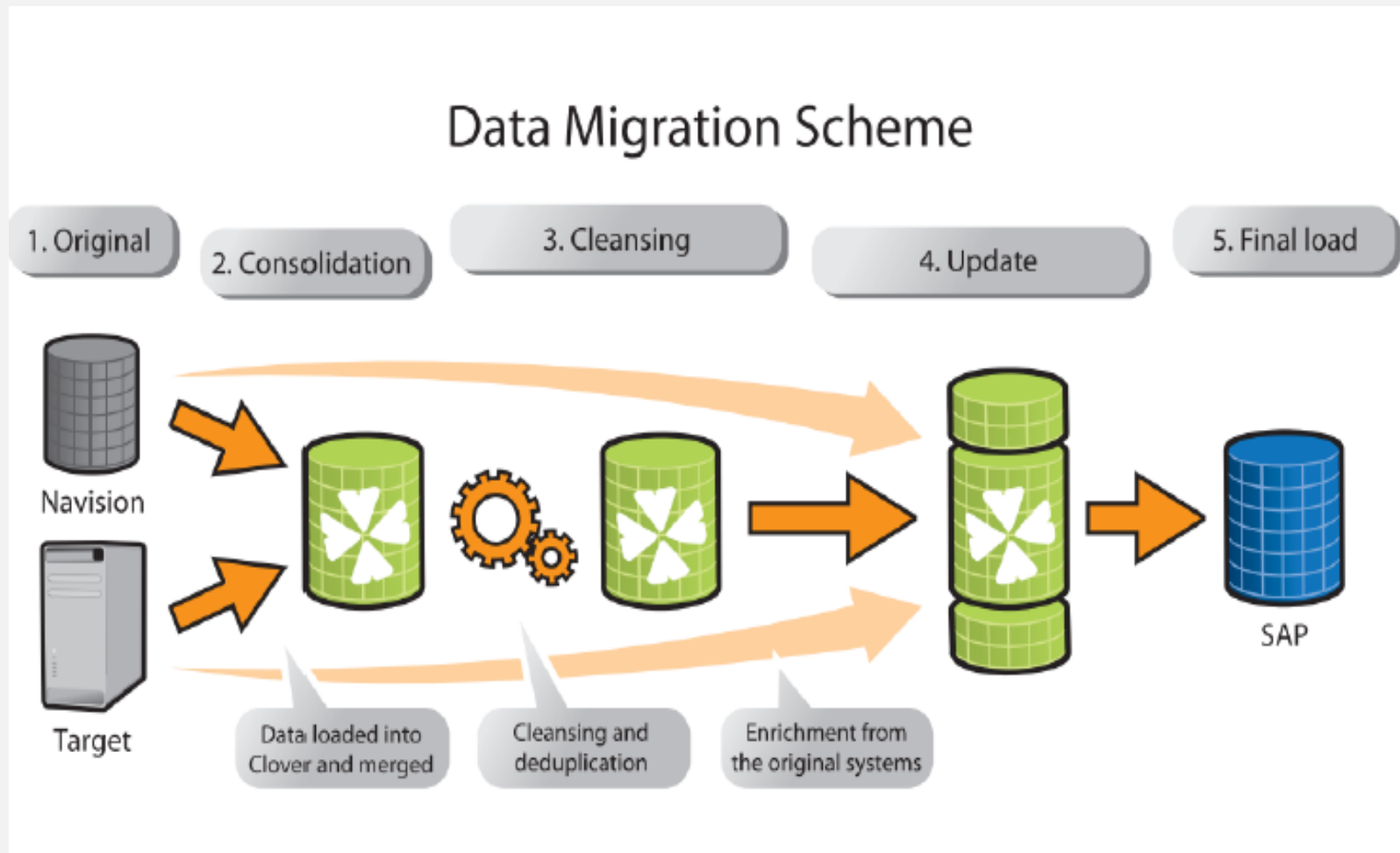
Repository of electronically stored data. ETL facilitates populating, reporting and analysis. Includes business intelligence as well as metadata retrieval and management tools.



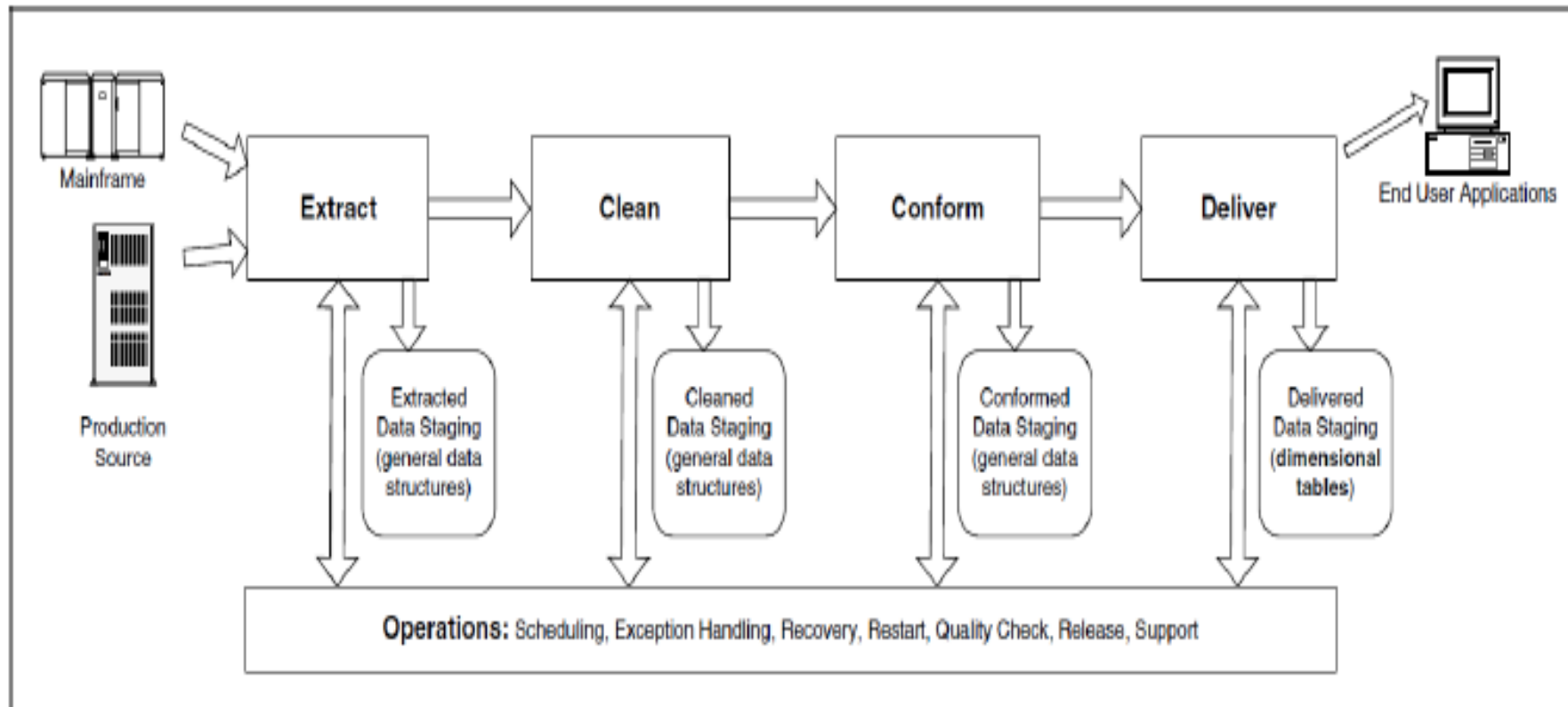
### Data Synchronization

Process of making sure two or more locations contain the same up-to-date files. Add, change, or delete a file from one location, synchronization will mirror the action at the new location.

Where is ETL used?



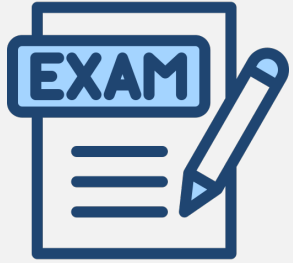
# HOW TO IMPLEMENT ETL SYSTEM



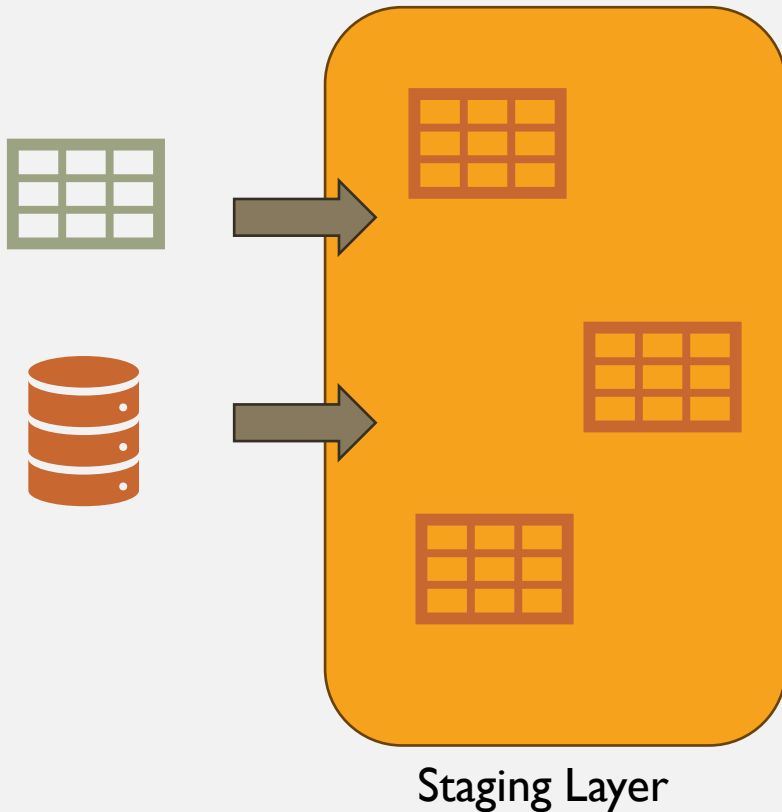
**Figure 1.2 The Four Staging Steps of a Data Warehouse**



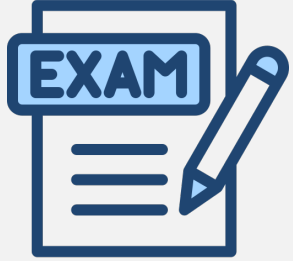
# EXTRACTING



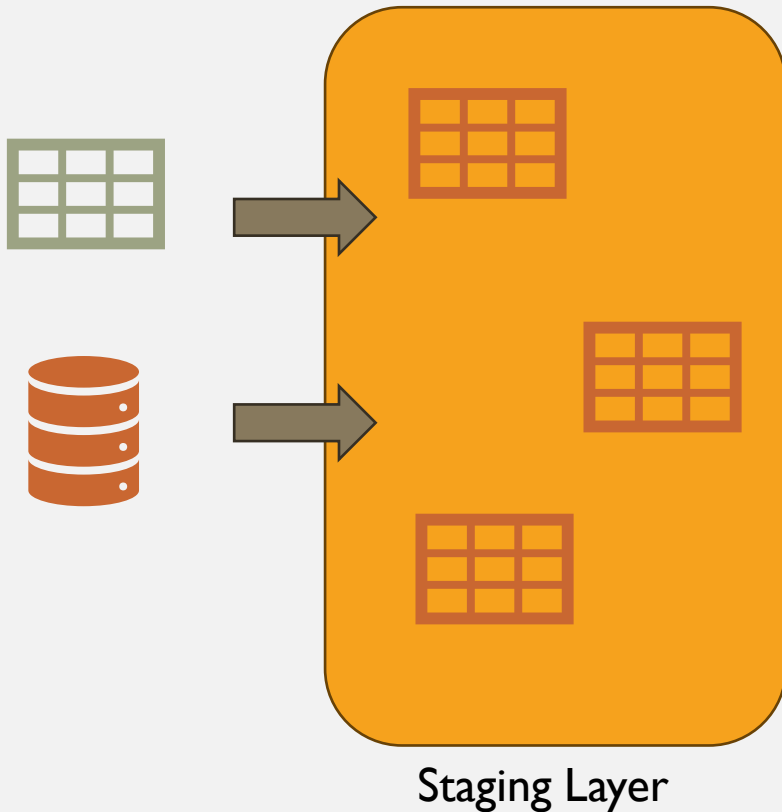
# DEEP DIVE INTO DATA EXTRACTION



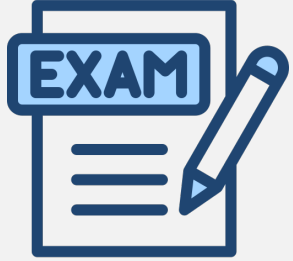
- Extract data from sources into the staging layer.
- Purpose: Prevent unnecessary load on source systems.
  - Source systems are productive and cannot be slowed down.
- Staging Environment
  - Allows data to be available in SQL tables.
  - Enables work on data, understanding it, and planning transformations.



## DEEP DIVE INTO DATA EXTRACTION STAGING LAYER



- Transient Type
  - Most common type.
  - Data is deleted or truncated after being copied with transformations.
  - Staging layer is empty until the next run when new data is loaded.
  - New data is then copied to the core layer.
- Permanent Type
  - Exists in some cases.
  - Focus is on the transient type due to its common usage.



# EXTRACTING TYPES

## Types of Data Loads

### **Initial Load**

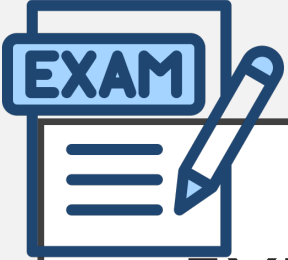
First (real) run  
All data

### **Delta Load**

Subsequent runs  
Only additional data

- Initial Load
  - First real run of ETL.
  - Loads all relevant data.
  - Preceded by testing and small extractions.
- Delta Load
  - Subsequent loads.
  - Only loads new data from the source system.
- Deep Dive
  - Understanding the workings of initial load and delta load.



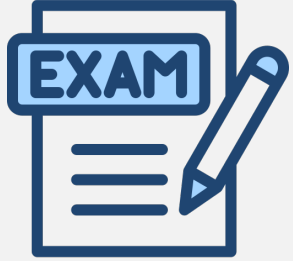


## INITIAL LOAD

EXTRACT ALL INITIAL DATA IS CALLED INITIAL LOAD.

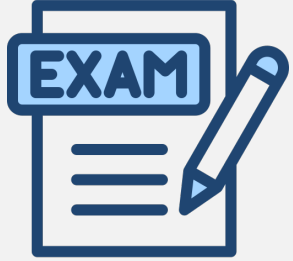
- Business case
  - Engage with business users (report users) and IT responsables (administrators of source systems/databases).
  - Understand data structure and needs of business users.
- Timing
  - Determine a good time for the initial load to minimize impact on productive systems.
  - Prefer non-operational hours (e.g., night, weekends).
  - Discuss timing with responsible parties to avoid unnecessary slowdowns.
- Testing
  - Conduct smaller extractions to estimate time required.
  - Provide estimates (e.g., three hours) and agree on suitable time for full load.
- Importance: Ensures minimal disruption to productive systems.
- Critical for successful initial data extraction.

DISCUSSION  
WHEN (DATETIME) INITIAL LOAD MAKE  
SENSE?



# INITIAL LOAD

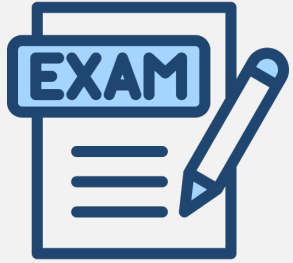
- Initial Load Process
  - Staging to Core Layer
    - Once data is in the staging layer, perform initial load to the core layer.
    - Apply all planned and tested transformation steps.
    - Copy all data from staging to core layer.
  - Steps:
    - Design, plan, and test transformation steps in ETL tool.
    - Execute transformation and data transfer from staging to core.



## DELTA LOAD

Sales Date	Name	Amount
2022-06-06	Sunglasses TR-7	\$25
2022-06-06	Chocolate bar 70% cacao	\$3
2022-06-07	Oat meal biscuits	\$4
2022-06-07	Chocolate bar 70% cacao	\$3
2022-06-08	Oat meal biscuits	\$4

- Transition to Delta Load:
  - Understanding the initial load sets the stage for the delta load.
  - Delta load involves incrementally loading new data on a regular basis.
- Frequency:
  - Set a frequency for delta loads (e.g., once per day, every night).
- Process:
  - Load new data from source system not previously loaded.
  - Steps:
    - First, bring new data into the staging layer.
    - Then, transfer it to the core layer.

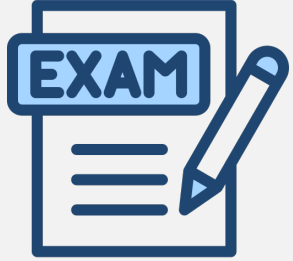


## DELTA LOAD AND DEKTA COLUMN

Sales Date	Name	Amount
2022-06-06	Sunglasses TR-7	\$25
2022-06-06	Chocolate bar 70% cacao	\$3
2022-06-07	Oat meal biscuits	\$4
2022-06-07	Chocolate bar 70% cacao	\$3
2022-06-08	Oat meal biscuits	\$4

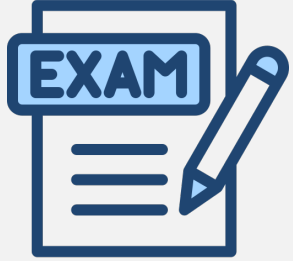
Essential for the delta load process.

- Required for every table in the data warehouse.
- Typically a timestamp or create date.
- Used to identify new data not yet loaded by the ETL process.
- Usually available in source systems.



## DELTA LOAD

- Workflow Consistency
  - No changes in the workflow structure.
  - Same transformations are applied.
  - ETL process retains identical structure and components.
- Periodic Execution
  - ETL process runs periodically for delta loads.
  - Filtering based on delta columns ensures only new data is processed.



## DELTA LOAD

Sales Date	Name	Amount
2022-06-06	Sunglasses TR-7	\$25
2022-06-06	Chocolate bar 70% cacao	\$3
2022-06-07	Oat meal biscuits	\$4
2022-06-07	Chocolate bar 70% cacao	\$3
2022-06-08	Oat meal biscuits	\$4

- Delta Columns
  - Crucial for identifying new data.
  - Typically timestamps or create dates.
- Alternative identification methods:
  - Consider other columns like primary keys.
  - Ensure incremental values for primary keys to accurately identify new rows.
- Challenges with natural keys (random sets of numbers) for primary key use.

# EXAMPLE

Sales Date	Name	Amount
2022-06-06	Sunglasses TR-7	\$25
2022-06-06	Chocolate bar 70% cacao	\$3
2022-06-07	Oat meal biscuits	\$4
2022-06-07	Chocolate bar 70% cacao	\$3
2022-06-08	Oat meal biscuits	\$4

**Remember MAX(Sales\_Key)**

MAX(Sales\_Key) -> Variable X

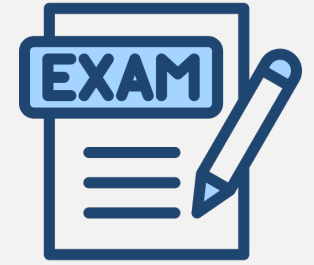
Next run: Sales\_Key > X

- Last ETL Run:
  - Identify highest value in sales key or timestamp (Delta column).
  - Example: Maximum sales key value is four.
- Next ETL Run:
  - Store maximum value (e.g., sales key = 4) in a variable (e.g., X).
  - Read data from source systems in the next run.
  - Apply filter on sales key: Load data where sales key is greater than X (4).
  - Result: Only load data where sales key is 5 or higher in the next run.

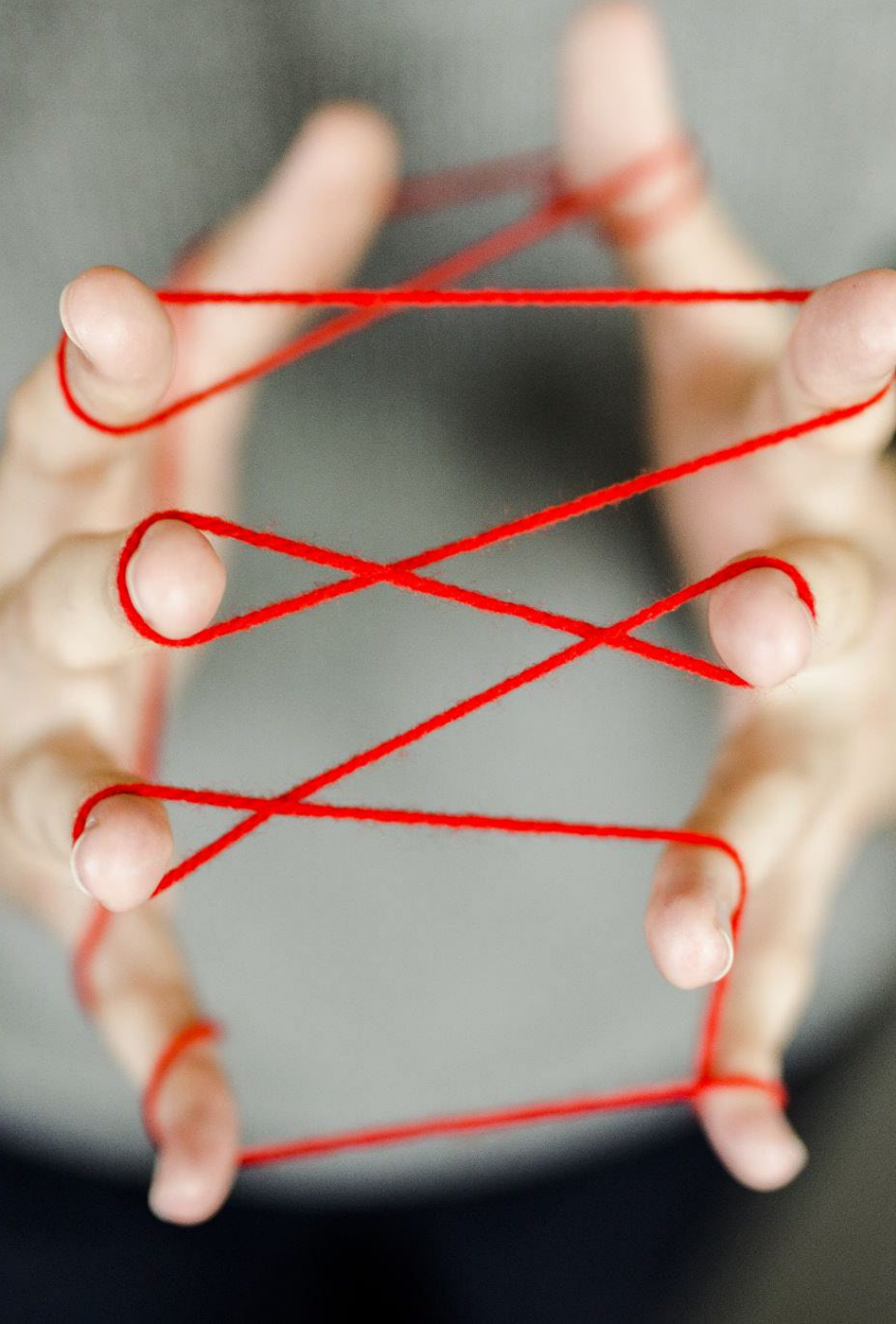


# DELTA LOAD

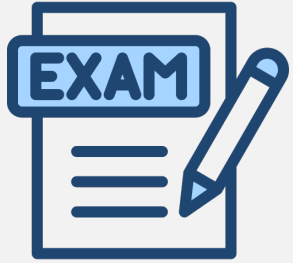
## USUALLY A TIMESTAMP AVAILABLE FOR TRACKING CHANGES, BUT ALTERNATIVES



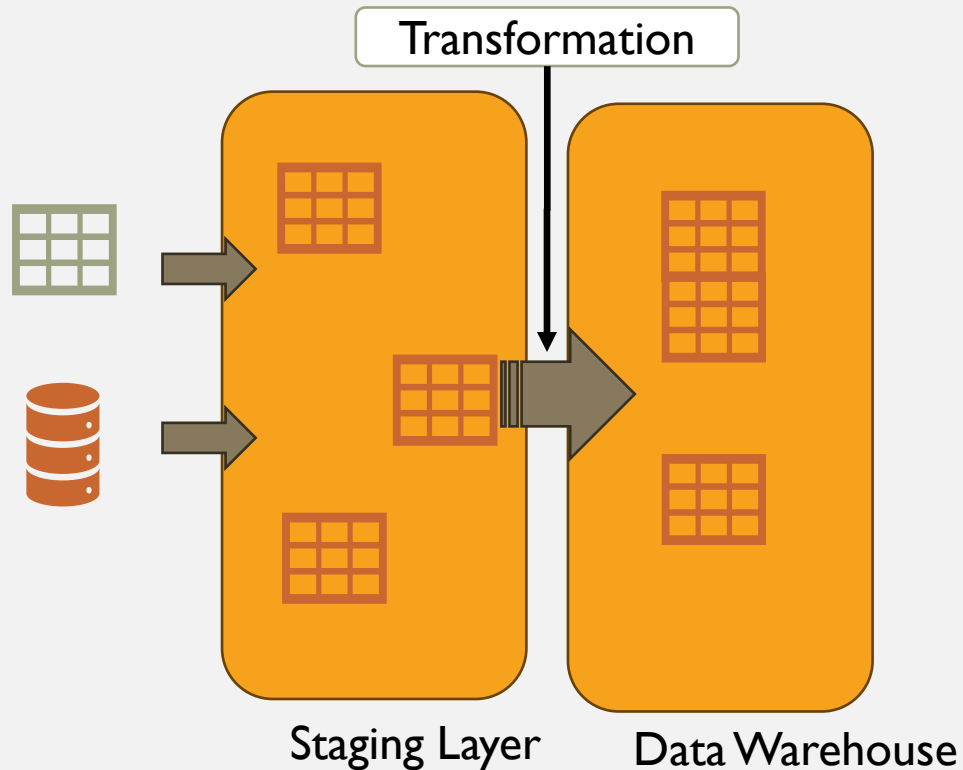
- Automated tools capturing metadata changes.
  - Automatically identify and extract new or changed data.
- Full Load Strategy:
  - For tables lacking a time-based identifier (e.g., dimension tables).
- Load all data and compare with existing records.
- Check for changes or additional columns.
- Considerations:
  - Performance impact on source systems.
  - Timing of full load (e.g., during off-peak hours like night).
  - Dimension tables are generally smaller, mitigating performance concerns.
- Performance and Frequency:
  - Load duration increases with data volume.
  - Balancing high frequency ETL requests with longer load times.
  - Example: ETL process updates every 30 minutes but may take 40 minutes due to occasional full loads.
- Decision Making:
  - Evaluate table size and performance implications before opting for full load strategy.



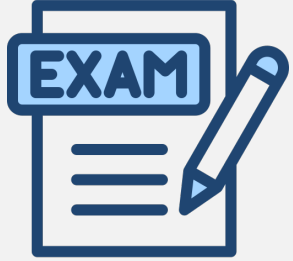
# TRANSFORMATION



# INTRODUCTION TRANSFORMATION

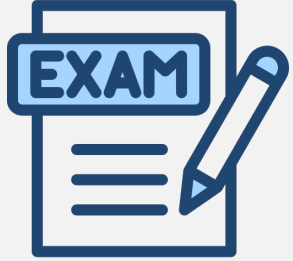


- Purpose of transforming data:
  - Ensure data consistency.
  - Extract more value from the data.
- From the staging layer, specific transformations are defined
- Transformed data is loaded into the core layer.
  - This is typically done via insert or update operations.



## GOALS

- Create a consolidated view of all data for analysis purposes
  1. Consolidate (from multiple sosystems)
  2. Reshape (for Analysis purposes)



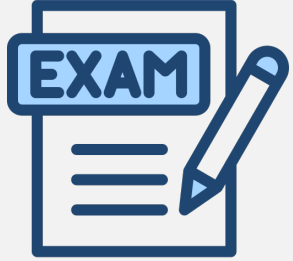
## TWO MAIN GOALS OF DATA TRANSFORMATION

### CONSOLIDATE (FROM MULTIPLE SYSTEMS)

Transaction ID	Amount	Date
A1	€5030	10/1/2024
A2	€5053	11/1/2024
A3	€654	10/1/2024

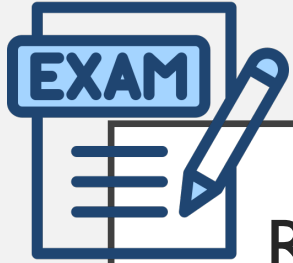
Transaction ID	Amount in thousand	Transaction_date
A14	€5.030	10-1-2024
A15	€7.654	11-1-2024
A16	€8.426	10-1-2024

- Create a consolidated view:
  - Integrate data from multiple systems.
  - Perform data type conversions.
  - Standardize column names and other formats.
- Additional reshaping for analytical or reporting needs:
  - Add additional information.
  - Reshape data for better analysis.



# CONSOLIDATION

- Different systems may have varied data formats, types, and column names.
- Example:
  - One column might show amounts in thousands with different data types (decimal vs. integer).
  - Date formats might differ across systems.
- Aim: Standardize and consolidate data to load into a single table.
- Importance: Ensures compatibility and consistency in the data warehouse.



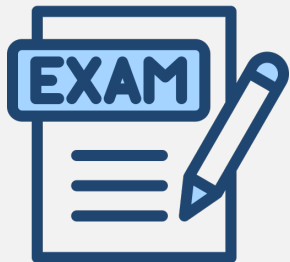
## RESHAPE

- Initial table may need restructuring.
- Goal: Use data in a dimensional way in the data warehouse.
- Example:
  - Include foreign keys.
  - These are simple reshaping and restructurings of the data.

Transaction ID	Amount	Date
A1	€5030	10/1/2024
A2	€5053	11/1/2024
A3	€654	10/1/2024



Transaction ID	Amount	Date
1	€5030	20240110
2	€5053	20240111
3	€654	20240110



## OTHER EXAMPLE RESHAPE ACCORDING TO BUSINESS REQUIREMENTS

Month	Januar-2022	February-2022	March-2022	Total
Amount	\$5030	\$6053	\$2455	\$13548



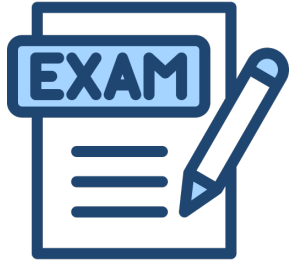
Month	Amount
Januar-2022	\$5030
February-2022	\$6053
March-2022	\$2455
Total	\$13548



Month	Amount
Januar-2022	\$5030
February-2022	\$6053
March-2022	\$2455

- Example: Data in Excel tables not suitable for analysis. Data needs to be stored in columns and rows.
- Pivoting data might be required.
  - Pivoting is less common but a good example of significant reshaping.
- Removing unnecessary columns (e.g., a total column).





Basic	Advanced
Deduplication	Joining
Filtering (rows & columns)	Splitting
Cleaning & Mapping (Integration)	Aggregating
Value Standardization (Integration)	Deriving new values
Key Generation	-

## OVERVIEW OF TRANSFORMATIONS

product_id	name	category
P521	Almonds 150g	Nuts
P252	Garlic	Fruits & Vegetables
P533	Banana	Fruits & Vegetables
P684	Chocolate Vcookies	Sweets & Snacks
P755	Spice Chips	Sweets & Snacks

product_id	name	category
P521	Almonds 150g	Nuts
P672	Orange Juice	Drinks
P423	Green Apples	Fruits & Vegetables
P564	Chocolate Vcookies	Sweets & Snacks
P756	Spice Chips	Sweets & Snacks

# BASIC TRANSFORMATION KEY ADDITION HANDLE DUPLICATES

product_id	name	category
P521	Almonds 150g	Nuts
P672	Orange Juice	Drinks
P423	Green Apples	Fruits & Vegetables
P564	Chocolate Vcookies	Sweets & Snacks
P756	Spice Chips	Sweets & Snacks
P521	Almonds 150g	Nuts
P252	Garlic	Fruits & Vegetables
P533	Banana	Fruits & Vegetables
P684	Chocolate Vcookies	Sweets & Snacks
P755	Spice Chips	Sweets & Snacks

- **Ensure No Duplicates**

- Important when dealing with data from multiple systems.
- Example: Data from two different stores.

- **Unified Dimension Creation**

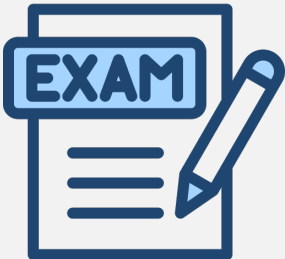
- Combine data to create one product dimension.
- Append data from multiple sources.

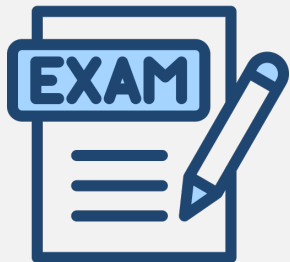
- **Handling Duplicates**

- Products may be available in both stores, leading to duplicates.
- Remove duplicate values to avoid redundancy in dimension tables.

- **Deduplication Process**

- Take distinct values after combining data into one table.
- Ensure only unique entries remain in the unified dimension.





## BASIC TRANSFORMATION KEY ADDITION FILTERING ROWS AND COLUMNS

Sales Date	Name	Amount	Type
2022-06-06	Sun lases TR-7	\$25	Sale
2022-06-06	Chocolate bar 70% cacao	\$3	Refund
2022-06-07	Oat meal biscuits	\$4	Sale
2022-06-07	Chocolate bar 70% cacao	\$3	Sale
2022-06-08	Oat meal biscuits	\$4	Sale

Sales Date	Name	Amount	Type
2022-06-06	Sun lases TR-7	\$25	Sale
2022-06-07	Oat meal biscuits	\$4	Sale
2022-06-07	Chocolate bar 70% cacao	\$3	Sale
2022-06-08	Oat meal biscuits	\$4	Sale

Sales Date	Name	Amount
2022-06-06	Sun lases TR-7	\$25
2022-06-07	Oat meal biscuits	\$4
2022-06-07	Chocolate bar 70% cacao	\$3
2022-06-08	Oat meal biscuits	\$4

### Filtering Rows

Similar to deduplication by removing irrelevant data.

Example: Creating a sales fact table.

Source data may include irrelevant data, such as refund transactions.

### Filter Conditions:

Remove data where the amount is negative.

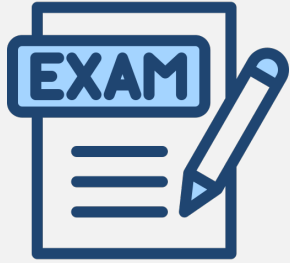
Remove rows where the data type is "Refund".

### Filtering Columns

Remove columns with redundant or unnecessary data.

Example: If every row in a column is of the type "Sale", the column can be removed.

Focus on relevant data only.



## BASIC TRANSFORMATION

### KEY ADDITION

### DATA CLEANING

Name	Gender
Taylor	M
Isabella	F
Sofia	F

Name	Gender
Lydia	Female
Naomi	Female
Leon	Male



Male == M  
Female == F

Name	Gender
Taylor	M
Isabella	FE
Sofia	F

- **Consistent Data Formatting**

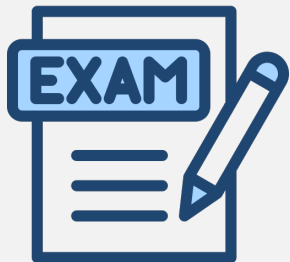
- Combine tables from different source systems into a consistent format.
- Example: Standardize customer gender information.
  - One system uses abbreviations (M for Male, F for Female).
  - Another system writes it out (Male, Female).
  - Map abbreviations to full terms for consistency (M to Male, F to Female).

- **Value Replacement**

- Replace specific values to standardize data.
- Example: Replace abbreviations with full terms.

- **Data Cleaning**

- Remove or replace unwanted characters.
- Example: Remove extraneous characters such as stray letters.
- Ensure data is clean and ready for integration.



## BASIC TRANSFORMATION

### KEY ADDITION NULL HANDLING AND VALUE STANDARDIZATION

Date	Sales
Monday	\$3500
Tuesday	\$760
Wednesday	null



Date	Sales
Monday	\$3500
Tuesday	\$760
Wednesday	0

Month	Sales
01 January 2022	\$500
01 February 2022	\$760
01 March 2022	\$245

Month	Sales in thousand
01 January 2022	\$1.5
01 February 2022	\$4.555
01 March 2022	\$3.321



Month	Sales
01 January 2022	\$1500
01 February 2022	\$4555
01 March 2022	\$3321

#### •Handling Null Values

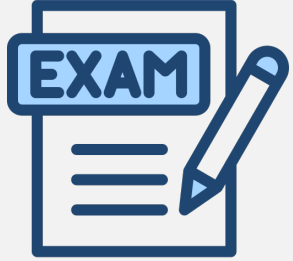
- Replace nulls with specific values based on context.
- Example: Replace null sales values with \$0 if no sales occurred on Wednesdays.

#### •Value Standardization

- Ensure uniformity in data types and units for integration.
- Example: Standardize sales figures:
  - Sales reported in thousands (e.g., 1.5 means 1,500).
  - Multiply values by 1,000 to convert to whole numbers.
  - Change data type to reflect the standardized values.

#### •Consolidating Data

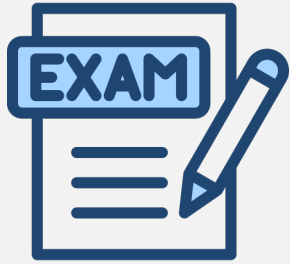
- Apply standardization and null handling to integrate data from different sources.
- Combine data into a single, consistent table for a consolidated view.



## BASIC TRANSFORMATION KEY ADDITION

P-id-pk	product_id	name	category
1	P521	Almonds 150g	Nuts
2	P672	Orange Juice	Drinks
3	P423	Green Apples	Fruits & Vegetables
4	P564	Chocolate Vcookies	Sweets & Snacks
5	P756	Spice Chips	Sweets & Snacks
6	P252	Garlic	Fruits & Vegetables
7	P533	Banana	Fruits & Vegetables
8	P684	Chocolate Vcookies	Sweets & Snacks
9	P755	Spice Chips	Sweets & Snacks

- **Adding a Key**
  - Auto-generate a key in the database management system or ETL tool.
  - Use a surrogate key as a replacement for the natural key.
  - Recommended practice for ensuring unique identifiers in the data.



# ADVANCED TRANSFORMATION

Product PK	Product id	name	category
1	P521	Almonds 150	Nuts
2	P252	Garlic	Fruits & Vegetable
3	P533	Banana	Fruits & Vegetable
4	P684	Chocolate	Sweets & Snacks
5	P755	Spicy Chips	Sweets & Snacks

Sales PK	Product id	Date
3	P533	01.01.2022
4	P252	01.01.2022
5	P755	02.01.2022
6	P648	02.01.2022
7	P755	02.01.2022



Sales PK	Product id	Product FK	Date
3	P533	3	01.01.2022
4	P252	2	01.01.2022
5	P755	5	02.01.2022
6	P648	4	02.01.2022
7	P755	5	02.01.2022

- **Joining Multiple Tables**

- Necessary to incorporate foreign keys into fact tables.
- Fact tables may initially contain only natural keys.
- Surrogate keys are added to dimension tables.

- **Referencing Keys**

- Join data to reference surrogate keys as foreign keys in fact tables.
- Use the common column (e.g., product ID) to perform joins.

- **Example Process**

- Identify the natural key in the fact table.
- Join with the dimension table to bring in the surrogate key.
- Reference surrogate keys in the fact table as foreign keys.

# ADVANCED TRANSFORMATION

## JOINING MULTIPLE TABLES BY REFERENCING KEYS

Product PK	Product id	name	category
1	P521	Almonds 150	Nuts
2	P252	Garlic	Fruits & Vegetable
3	P533	Banana	Fruits & Vegetable
4	P684	Chocolate	Sweets & Snacks
5	P755	Spicy Chips	Sweets & Snacks

Sales PK	Product id	Date
3	P533	01.01.2022
4	P252	01.01.2022
5	P755	02.01.2022
6	P648	02.01.2022
7	P755	02.01.2022

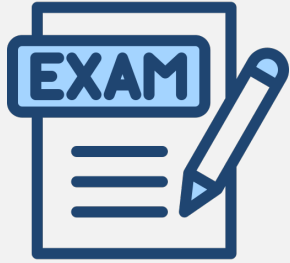


Sales PK	Product id	Product FK	Date
3	P533	3	01.01.2022
4	P252	2	01.01.2022
5	P755	5	02.01.2022
6	P648	4	02.01.2022
7	P755	5	02.01.2022

- Example:**

- Product ID 533 in the fact table corresponds to the primary key 3 in the dimension table.
- After joining, the fact table now includes the foreign key value 3 for product ID 533.
- This process works like a lookup to integrate the foreign key into the fact table.

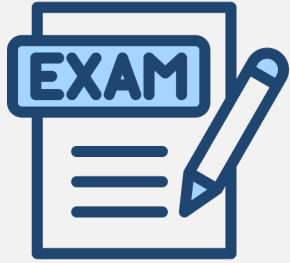




# ADVANCED TRANSFORMATION HANDLING SLOWLY CHANGING DIMENSIONS

Product PK	product_id	name	category	Eff_Date	Exp_Date
1	P521	Almonds 150g	Nuts	2021-01-01	2121-01-01
2	P252	Garlic	Fruits & Vegetables	2021-01-01	2121-01-01
3	P533	Banana	Fruits & Vegetables	2021-01-01	2121-01-01
4	P684	Chocolate	Sweets & Snacks	2021-01-01	2121-01-01
5	P755	Spicy Chips	Sweets & Snacks	2021-01-01	2121-01-01

- **Effective and Expiry Dates**
  - Use effective and expiry dates for slowly changing dimensions.
  - Ensure that the transaction date falls between these dates.
- **Filtering with Dates**
  - Prevent multiple values for the same natural key by filtering data.
  - Only include records where the transaction date is between the effective and expiry dates.
- **Specific Use Case**
  - Applicable when working with slowly changing dimensions.
  - For other scenarios, a simple join is sufficient.
- **Example Process**
  - Join tables based on a common column.
  - Apply additional filter conditions for effective and expiry dates to ensure accurate results.



# MERGING TABLES FOR ADDITIONAL COLUMNS

Product PK	Product id	name	category
1	P521	Almonds 150	1
2	P252	Garlic	2
3	P533	Banana	2
4	P684	Chocolate	3
5	P755	Spicy Chips	3

Catergory id	category
1	Nuts
2	Fruits & Vegetable
3	Sweets & Snacks

## Adding Additional Columns

- Merge separate tables to include additional columns.
- Example: Combine product and category tables into one.

## User-Friendliness

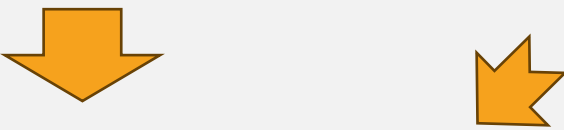
- Create a compact, single product table for ease of use.
- Simplifies access and improves user experience.

## Performance Improvement

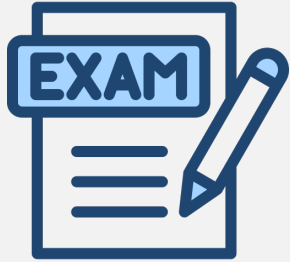
- Reduce the need for manual joins, saving compute resources.
- Perform the join using a common column (e.g., category column).

## Resulting Table

- A consolidated product dimension table.
- Enhances performance and usability for end users.



Product PK	Product id	name	category
1	P521	Almonds 150	Nuts
2	P252	Garlic	Fruits & Vegetable
3	P533	Banana	Fruits & Vegetable
4	P684	Chocolate	Sweets & Snacks
5	P755	Spicy Chips	Sweets & Snacks



# SPLITTING DATA

Store Dimenion

Store_id	Location
1	New York, NY 10011
2	Orland Park, IL 60462
3	Houston, TX 77002

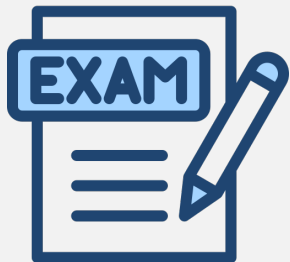


Store_id	City	Location
1	New York	NY 10012
2	Orland Park	IL 60463
3	Houston	TX 77003



Store_id	City	State	Location
1	New York	NY	10013
2	Orland Park	IL	60464
3	Houston	TX	77004

- **Splitting Columns**
  - Example: Store dimension with city, state, and zip code in one column.
  - Split data into separate columns for independent use.
- **Splitting by Delimiter**
  - Use a specific delimiter (e.g., comma) to split data.
  - Example: Split by comma to extract city names.
- **Splitting by Whitespace**
  - Use whitespace as a delimiter to split data.
- **Splitting by Length or Position**
  - Define split points by character length or specific positions.
  - Example: Split after two characters for state abbreviations and zip codes.
- **Practical Application**
  - Extract relevant parts into separate columns.
  - Improves data organization and accessibility.



## CHANGING GRANULARITY AND AGGREGATING DATA

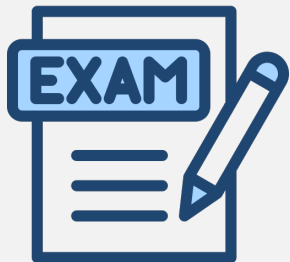
Sales Date	Name	Amount
2022-06-06	Sun lases TR-7	\$25
2022-06-06	Chocolate bar 70% cacao	\$3
2022-06-07	Oat meal biscuits	\$4
2022-06-07	Chocolate bar 70% cacao	\$3
2022-06-08	Oat meal biscuits	\$4



Sales Date	No, of sales	Amount
06.06.2022	2	\$28
07.06.2022	2	\$7
08.06.2022	1	\$4


- SUM
- COUNT
- DISTINCT COUNT
- AVERAGE

- **Aggregation for Changing Granularity**
  - Adjust the granularity of data to meet specific needs.
  - Example: Aggregate daily sales data.
- **Types of Aggregations**
  - **Sum:**
    - Sum all sales amounts for each day.
  - **Count:**
    - Count the number of sales transactions.
    - Each row represents one sale or transaction.
  - **Distinct Count:**
    - Count the number of unique products sold.
  - **Average:**
    - Calculate the average of a given metric, such as sales amount.
- **Practical Examples**
  - Sum: Total daily sales amount.
  - Count: Total number of sales transactions per day.
  - Distinct Count: Number of different products sold.
  - Average: Average sales amount per transaction.



# CALCULATING ADDITIONAL VALUES

Sales Date	Name	Amount	Tax	
06.06.2022	Sun lases TR-7	\$25	17%	
06.06.2022	Chocolate bar 70% cacao	\$3	6%	
07.06.2022	Oat meal biscuits	\$4	6%	
07.06.2022	Chocolate bar 70% cacao	\$3	6%	



Sales Date	Name	Amount	Tax	Tax_amount
06.06.2022	Sun lases TR-7	\$25	17%	\$4,25
06.06.2022	Chocolate bar 70% cacao	\$3	6%	\$0,18
07.06.2022	Oat meal biscuits	\$4	6%	\$0,24
07.06.2022	Chocolate bar 70% cacao	\$3	6%	\$0,18
08.06.2022	Oat meal biscuits	\$4	6%	\$0,24

- **Performing Calculations**

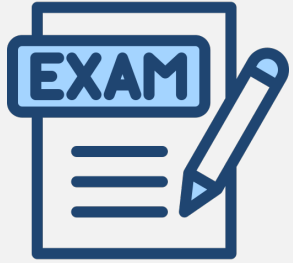
- Example: Calculate tax amount using percentage and sales amount.
- Tax percentage is non-additive, but absolute tax amount can be computed.

- **Creating New Columns**

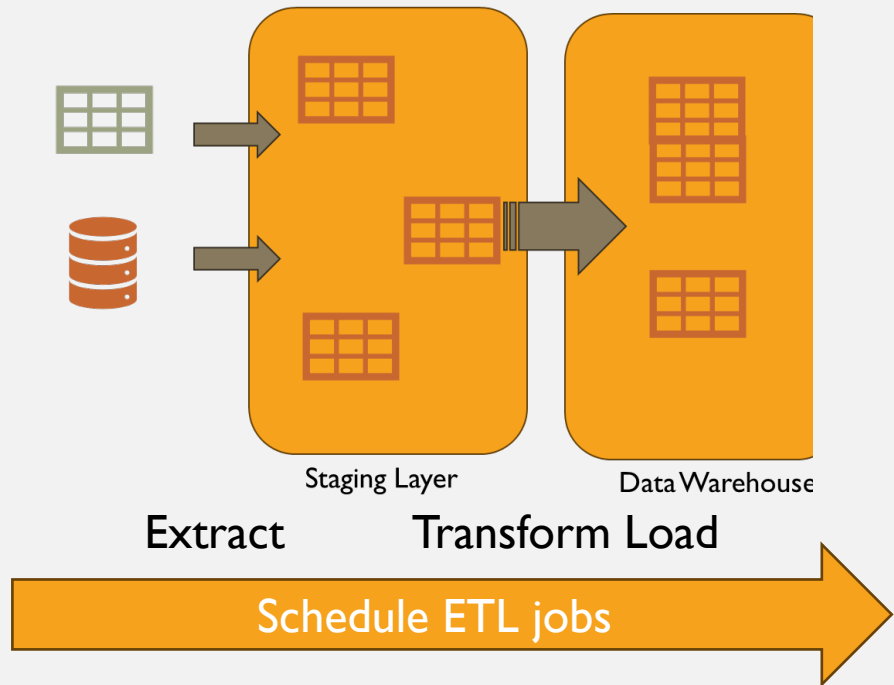
- Derive new insights by multiplying or manipulating existing columns.
- Example: Compute tax amount as a product of tax percentage and sales amount.

- **Types of Calculations**

- **Subtraction:** Subtract one value from another.
- **Multiplication:** Multiply values to derive new metrics.
- **Grouping:** Group data to analyze subsets collectively.



# SCHEDULING ETL JOBS



- **Continuous Data Processing**

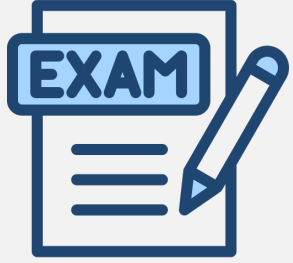
- Ensure data is continuously extracted, transformed, and loaded (ETL) to keep the data warehouse up to date.

- **Packaging Workflows**

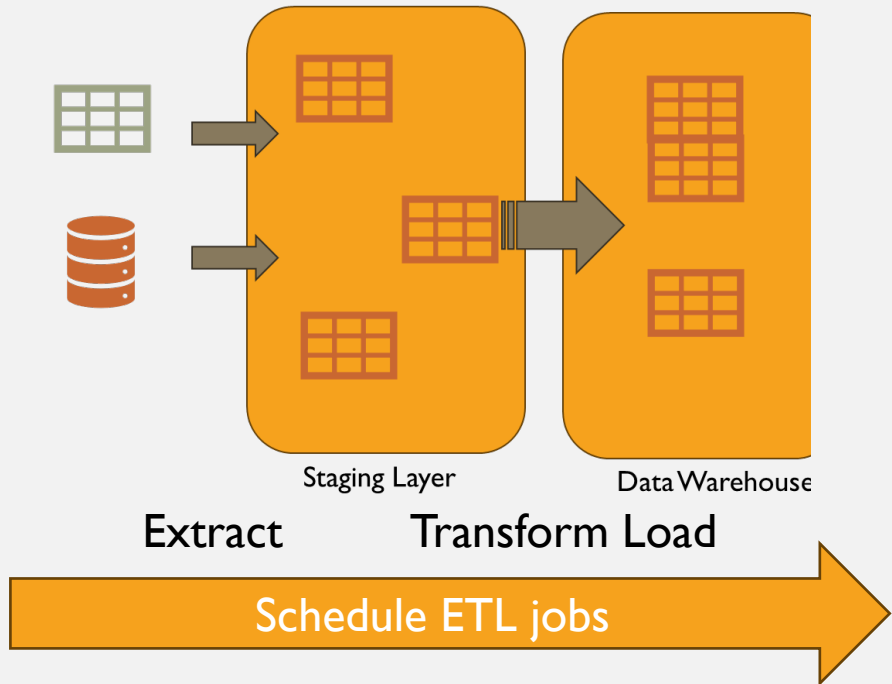
- Group transformations into jobs or packages depending on the ETL tool used.

- **Scheduling**

- Schedule jobs or packages to run at specific times or frequencies.
- Automate the ETL process to maintain data currency in the data warehouse.

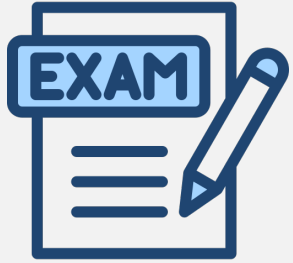


# SCHEDULING ETL JOBS



- **Best Practices**

- **Frequency:** Determine how often data needs to be updated (e.g., hourly, daily).
- **Dependencies:** Manage job dependencies to ensure tasks run in the correct sequence.
- **Monitoring:** Implement monitoring to track job execution and detect any issues.
- **Error Handling:** Include error handling mechanisms to manage and resolve failures promptly.
- **Logging:** Maintain logs for auditing and troubleshooting purposes.
- **Performance:** Optimize job schedules to minimize impact on system resources.



# IMPLEMENTATION IN ETL PROCESS

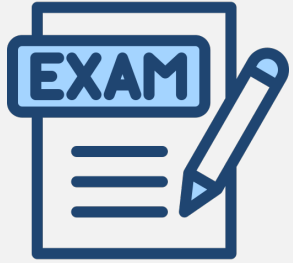
Perform scheduling in...

..in ETL tools ...

... or using external tools

- **Using ETL Tool**
  - Schedule jobs directly within the ETL tool itself, typically available in the enterprise version.
  - Example: Pentaho offers scheduling features in the enterprise edition.
- **Alternative Methods**
  - **External Scheduling Tools:** If scheduling isn't available in the ETL tool (e.g., using a free version), use external tools.
    - Utilize tools like Windows Scheduler or similar for job execution.
    - External tools trigger the execution of ETL jobs at specified times.
- **Deployment on Server**
  - Deploy ETL packages or jobs onto a dedicated server for execution.
  - Ensure the server is optimized for handling ETL processes efficiently.





# IMPLEMENTATION IN ETL PROCESS

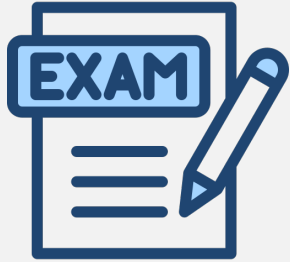
Perform scheduling in...

..in ETL tools ...

... or using external tools

- **Best Practices for Scheduling**

- **Selection of Tools:** Choose scheduling methods based on the capabilities of the ETL tool and organizational needs.
- **Execution Control:** Manage job execution timing and frequency to meet data refresh requirements.
- **Monitoring and Management:** Implement monitoring to track job execution status and performance.
- **Scalability:** Ensure scalability of scheduling solutions as data volumes and complexity grow.
- **Documentation:** Document schedules and dependencies for clarity and maintenance.



# GUIDELINES FOR SCHEDULING ETLs

What are the requirements?	How long does it take?	What is a good time
3x1 day?	1 hour?	Initial Load vs. Delta Load
1x1 day?	5 min?	Effect on productive system
Every 30 min?		Short read access
		Night? Morning?

## Understanding Business Requirements

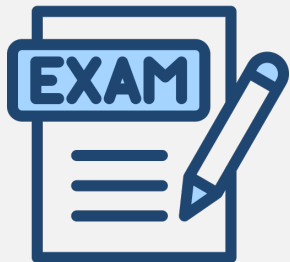
Start by identifying the data update frequency needed by business users.  
Communicate with stakeholders to determine how often data updates are required.

## Matching Requirements with ETL Performance

Assess the actual duration of the ETL process.  
ETLs can vary in duration from minutes to hours depending on data volume and complexity.  
Align the frequency of updates with the time it takes for the ETL to complete.

## Finding a Balance

Balance business requirements with technical feasibility.  
If there's a discrepancy between update frequency and ETL duration, negotiate a compromise or find ways to optimize the ETL process.



# GUIDELINES FOR SCHEDULING ETLs

What are the requirements?	How long does it take?	What is a good time
3x1 day?	1 hour?	Initial Load vs. Delta Load
1x1 day?	5 min?	Effect on productive system
Every 30 min?		Short read access
		Night? Morning?

## Choosing Execution Time

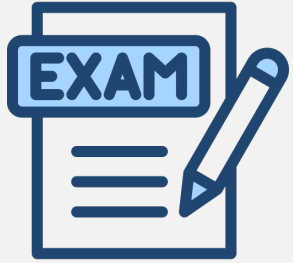
- Consider the impact on productive source systems when scheduling ETL jobs.
- Test data extractions to understand resource utilization and potential impacts on source systems.
- Coordinate with responsible personnel to determine an appropriate execution time that minimizes disruption to productive operations.

## Differentiating Initial Load and Delta Load

- Recognize the difference in resource demand between initial data loads and ongoing delta loads.
- Initial loads may require more resources and time, while delta loads are typically less resource-intensive.

## Ideal Execution Times

- Nighttime, weekends, or early morning hours before peak usage times are often suitable for ETL execution.
- Consult with relevant stakeholders to decide on the best time based on system usage patterns and business needs.



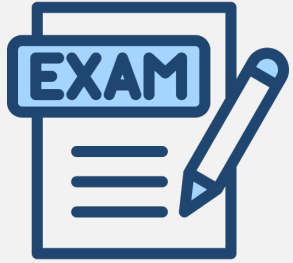
## GUIDELINES FOR SCHEDULING ETLs

What are the requirements?	How long does it take?	What is a good time
3x1 day?	1 hour?	Initial Load vs. Delta Load
1x1 day?	5 min?	Effect on productive system
Every 30 min?		Short read access
		Night? Morning?

### Collaborative Approach

Engage in discussions and testing with system administrators and stakeholders to validate and finalize the timing of ETL executions.

Ensure alignment between ETL scheduling and overall business operations to minimize disruption and maximize efficiency.



# GUIDELINES FOR SCHEDULING ETLs

What are the requirements?	How long does it take?	What is a good time
3x1 day?	1 hour?	Initial Load vs. Delta Load
1x1 day?	5 min?	Effect on productive system
Every 30 min?		Short read access
		Night? Morning?

- **Implementation Strategy**

- Document agreed-upon scheduling parameters and rationale.
- Regularly review and adjust schedules based on changing business needs and system performance.
- Maintain open communication channels with stakeholders to address any issues or adjustments in scheduling.

# ETL TOOLS

Enterprise	Open-source	Cloud-native	Custom
Commercial	Source code	Cloud technology	Own development
Most mature	Often free	Data already in cloud?	Customized
Graphical interface	Graphical interface	Efficiency	Internal resources
Architectural needs	Support?	Flexibility?	Maintainance?
Support	Ease of use?		Training?

- **Commercial ETL Tools**

- Developed and offered by companies for a price.
- Known for maturity, robust interfaces, and user-friendliness.
- Typically support a wide range of data sources and offer comprehensive customer support.
- Ideal for large companies where ETL processes are critical to business success.

# ETL TOOLS

Enterprise	Open-source	Cloud-native	Custom
Commercial	Source code	Cloud technology	Own development
Most mature	Often free	Data already in cloud?	Customized
Graphical interface	Graphical interface	Efficiency	Internal resources
Architectural needs	Support?	Flexibility?	Maintainance?
Support	Ease of use?		Training?

- **Open-Source ETL Tools**
  - Code is publicly accessible, fostering transparency and trust in how the tool operates.
  - Often free to use, but not always.
  - Some enterprise tools also have open-source versions with accessible code.
  - Mature open-source tools increasingly offer graphical interfaces but may lack formal support and consistency in ease of use.

# ETL TOOLS

Enterprise	Open-source	Cloud-native	Custom
Commercial	Source code	Cloud technology	Own development
Most mature	Often free	Data already in cloud?	Customized
Graphical interface	Graphical interface	Efficiency	Internal resources
Architectural needs	Support?	Flexibility?	Maintainance?
Support	Ease of use?		Training?

- **Cloud-Native Solutions**
  - Offered by major cloud providers like AWS, Azure, etc.
  - Efficient for organizations already leveraging cloud services.
  - Consider flexibility in accessing and working with data across different cloud providers.



# ETL TOOLS

Enterprise	Open-source	Cloud-native	Custom
Commercial	Source code	Cloud technology	Own development
Most mature	Often free	Data already in cloud?	Customized
Graphical interface	Graphical interface	Efficiency	Internal resources
Architectural needs	Support?	Flexibility?	Maintainance?
Support	Ease of use?		Training?

- **Custom Developed Solutions**
  - Developed internally to meet specific needs, often stemming from historical necessity.
  - Requires significant resources for development, maintenance, and training.
  - May lack maturity and robustness compared to commercial or open-source solutions.

# ETL TOOLS

## **Enterprise**

- Alteryx
- Informatica
- Oracle Data Integrator
- Microsoft SSIS

## **Open-source**

- Tatend Open Studio
- Pentaho Data Integration
- Hadoop

## **Cloud-native**

- Azure Data Factory
- AWS Glue
- Google Cloud Data Flow