In a DW, **ETL (Extract, Transform, Load)** processes play a crucial role in shaping data from source systems into valuable info stored in the DW.

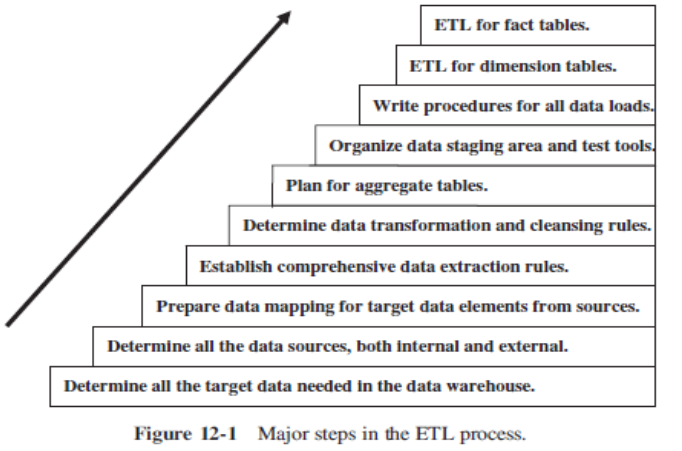
1. *Data Acquisition:* Involves extracting data from source systems. Extraction includes selecting relevant data & dealing with diverse source systems on different platforms.
2. *Data Storage:* Transforms extracted data into suitable formats for storage in the DW. This process deals with challenges like diverse source system structures and inconsistent data quality.
3. *Information Delivery:* Once data is stored in DW, it can be accessed for reporting and analysis, providing strategic insights.

**ETL Functions:**

1. *Extraction:* Selects and gathers data from source systems, often dealing with diverse and disparate (distinct in quality or character) platforms.
2. *Transformation:* Changes the extracted data into appropriate formats and structures for storage in the data warehouse.
3. *Loading:* Physically moves transformed data into the DW repository.

**Challenges in ETL Processes:**

1. Source systems come in different types and run on different platforms.
2. Some systems use old and outdated technology.
3. Historical data changes may not always be kept.
4. The quality of data in old systems might not be reliable.
5. Source system structures can change unexpectedly.
6. It's common to find inconsistencies between different source systems.
7. Fixing these inconsistencies can be tough.
8. Source systems may not present data in a way that's easy for users to understand.



**Complexity of data extraction:**

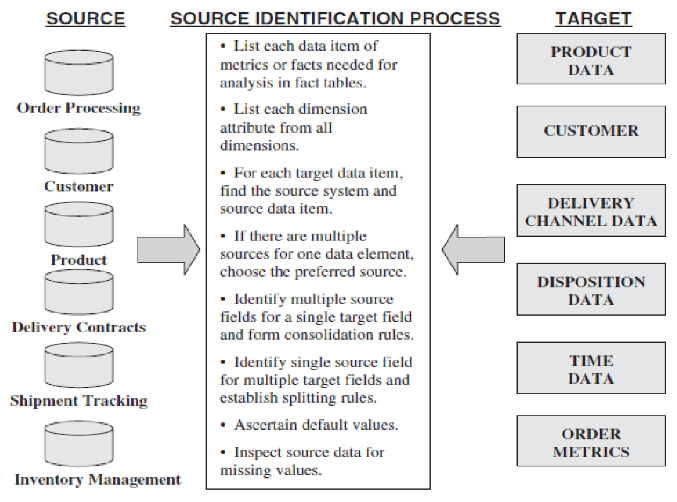
1. *Source Diversity:* Extracting data from many different sources adds complexity.
2. *Incremental and Full Loads:* It must handle ongoing updates & initial full data loads.
3. *Tools Usage:* Using third-party tools alongside in-house scripts is recommended for efficiency.

**Data Extraction Issues:**

1. *Source Identification:* Identifying source applications and their data structures.
2. *Extraction Method:* Determining if extraction is manual or tool-based for each source.
3. *Extraction Frequency:* Establishing how often data needs to be extracted (daily, weekly, etc.) for each source.
4. *Time Window:* Defining the time frame for extraction for each source.
5. *Job Sequencing:* Deciding if one extraction job must wait for another to finish.
6. *Exception Handling:* Figuring out how to manage records that can't be extracted.

**Identification of Data Sources:**

1. Know the nature of the data and how it will be used.
2. Source data changes with business transactions.
3. Source data is time-dependent, meaning values change over time.
4. DW must capture all changes to maintain a historical record by understanding how source systems store data helps identify changes.



**Types of Data Extractions:**

1. *"As Is" (Static) Data Extraction:*

This type of extraction captures data at a specific point in time, like taking a snapshot of the data as it exists at that moment. This method is commonly used for the initial loading of a DW when you want to establish a baseline of data.

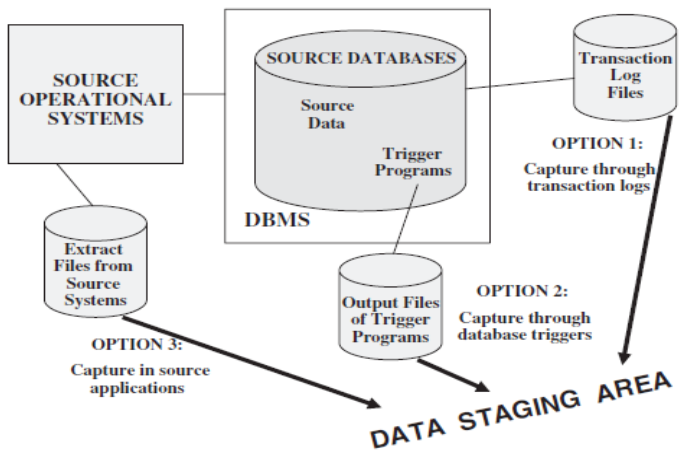
1. *Data of Revisions:*

This type involves capturing changes made to the data since the last extraction. It's about identifying what has been added, modified, or deleted. Depending on the requirements, this can be done in real-time (immediate) or at scheduled intervals (deferred).

**Immediate Data Extraction:**

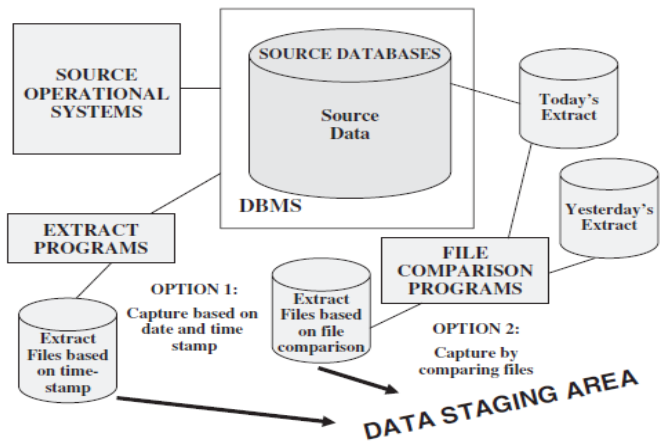
1. *Capture through Transaction Logs:*

Transaction logs maintained by dbs are used to identify changes. These logs record every transaction that occurs, including inserts, updates, and deletes. By reading these logs, it's possible to identify and capture changes as they happen.

1. *Capture through Database Triggers:*

DB triggers are special procedures that automatically execute in response to specific events, such as data changes. These triggers can be set up to capture changes and write them to a separate file or table, which can then be extracted for the data warehouse.

1. *Capture in Source Applications:*

Changes are made directly within the source appns themselves to help with data capture. When data is added or updated in the source application by users or processes, these applications are adjusted to perform extra actions alongside to ensure that any changes made to the data are also recorded or flagged in a way that makes them accessible for extraction.

**Deferred Data Extraction:**

1. *Capture Based on Date and Time Stamp:*

Records in source data are marked with a timestamp indicating when they were created or last modified. During extraction, only records with timestamps within a specified range (since the last extraction) are selected.

1. *Capture by Comparing Files:*

This method involves comparing two snapshots of the source data, taken at different times. By comparing the two snapshots, it's possible to identify changes and extract only the modified or new records.

