ETL

Dr. Yashvardhan Sharma

Department of Computer Science & Information Systems

BITS, Pilani

L8 (Lec 13)

Purpose of ETL

- **ETL functions reshape the relevant data from the source systems into useful information to be stored in the data warehouse.**
- Without these functions, there would be no strategic information in the data warehouse.
- If the source data is not extracted correctly, cleansed, and integrated in the proper formats, query processing, the backbone of the data warehouse, could not happen.

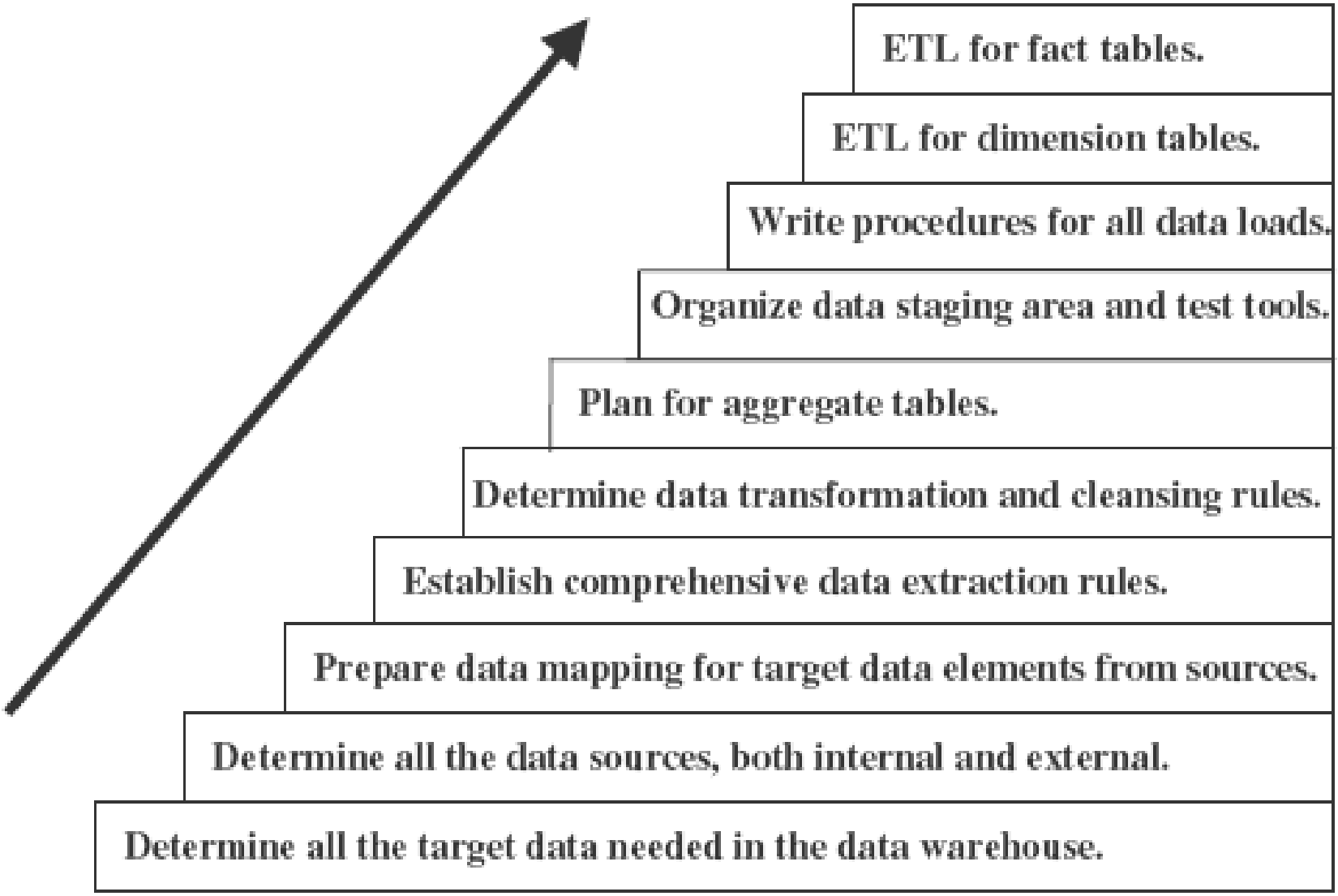


Figure 12-1 Major steps in the ETL process.

Types of activities and tasks that compose the ETLprocess

- Split one source data structure into several structures to go into several rows of the target database.
- Read data from data dictionaries and catalogs of source systems.
- Read data from a variety of file structures including flat files, indexed files (VSAM), and legacy system databases (hierarchical/network).
- Load details for populating atomic fact tables.
- Aggregate for populating aggregate or summary fact tables.
- Transform data from one format in the source platform to another format in the target platform.
- Derive target values for input fields (example: age from date of birth).
- Change cryptic values to values meaningful to the users (example: 1 and 2 to male and female).

DATA EXTRACTION

DATA EXTRACTION

- For operational systems upgrade, all you need is one-time extractions and data conversions.
- For a data warehouse, you have to extract (increased complexity, 3rd party tools):
  - data from many disparate sources.
  - data on the changes for ongoing incremental loads as well as for a one-time initial full load.

List of data extraction issues

- ***Source Identification***—identify source applications and source structures.
- ***Method of extraction***—for each data source, define whether the extraction process is manual or tool-based.
- ***Extraction frequency***—for each data source, establish how frequently the data extraction must by done—daily, weekly, quarterly, and so on.
- ***Time window***—for each data source, denote the time window for the extraction process.
- ***Job sequencing***—determine whether the beginning of one job in an extraction job stream has to wait until the previous job has finished successfully.
- ***Exception handling***—determine how to handle input records that cannot be extracted

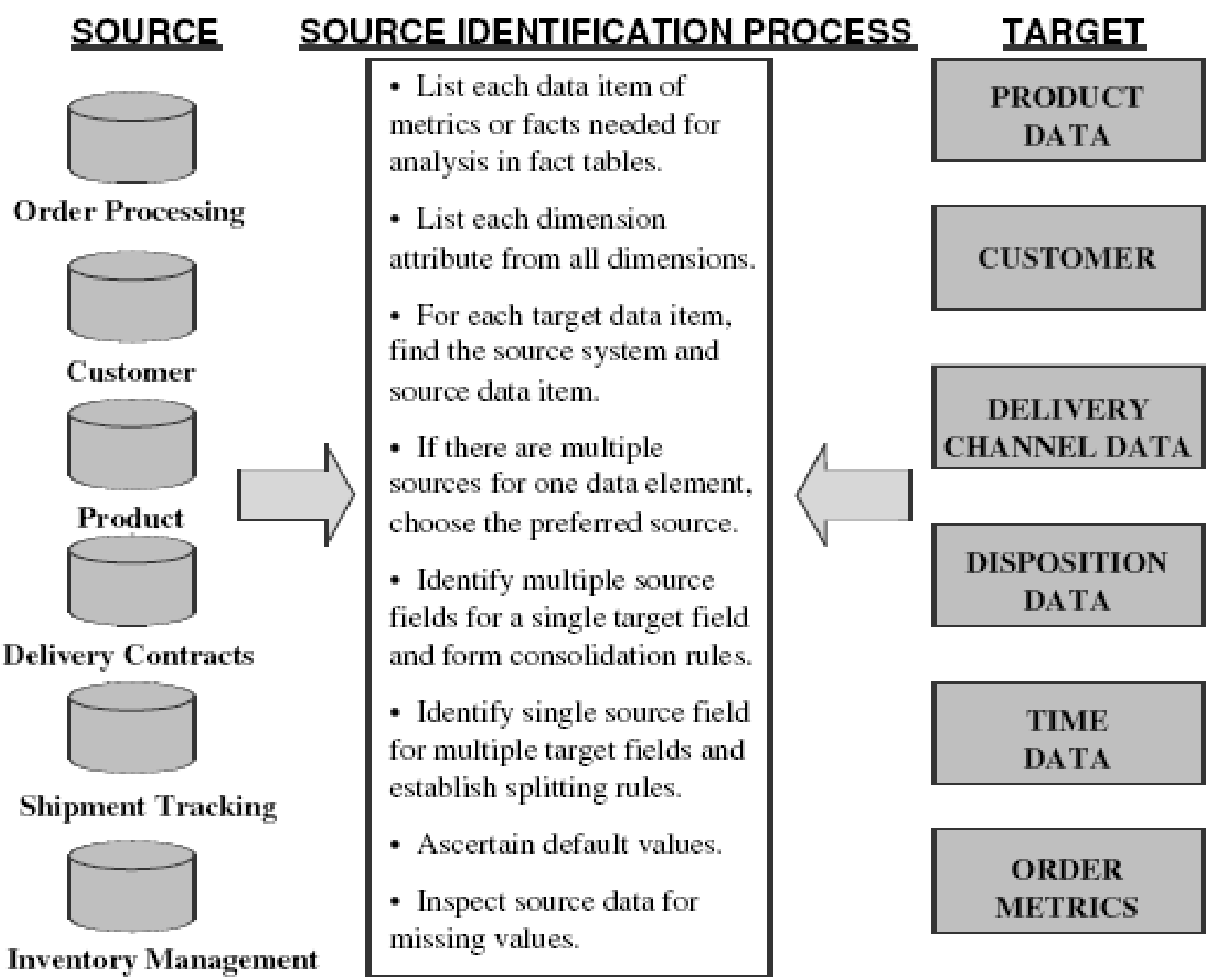


Figure 12-2 Source identification: a stepwise approach.

Data in Operational Systems.

Two categories:

- **Current Value**. (most of the attributes) The value of an attribute remains constant only until a business transaction changes it. Data extraction for preserving the history of the changes in the data warehouse gets quite involved for this category of data.
- **Periodic Status**. (not as common as the previous category) The history of the changes is preserved in the source systems themselves. Therefore, data extraction is relatively easier.

EXAMPLES OF ATTRIBUTES

VALUES OF ATTRIBUTES AS STORED IN OPERATIONAL SYSTEMS AT DIFFERENT DATES

Storing Current Value

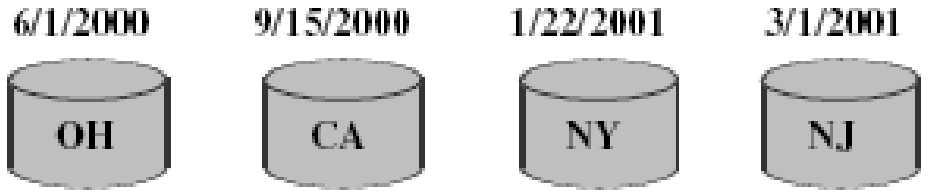
Attribute: Customer's State of Residence

6/1/2000 Value: OH

9/15/2000 Changed to CA

1/22/2001 Changed to NY

3/1/2001 Changed to NJ



Storing Periodic Status

Attribute: Status of Property consigned to an auction house for sale.

6/1/2000 Value: RE (property receipted)

9/15/2000 Changed to ES (value estimated)

1/22/2001 Changed to AS (assigned to auction)

3/1/2001 Changed to SL (property sold)

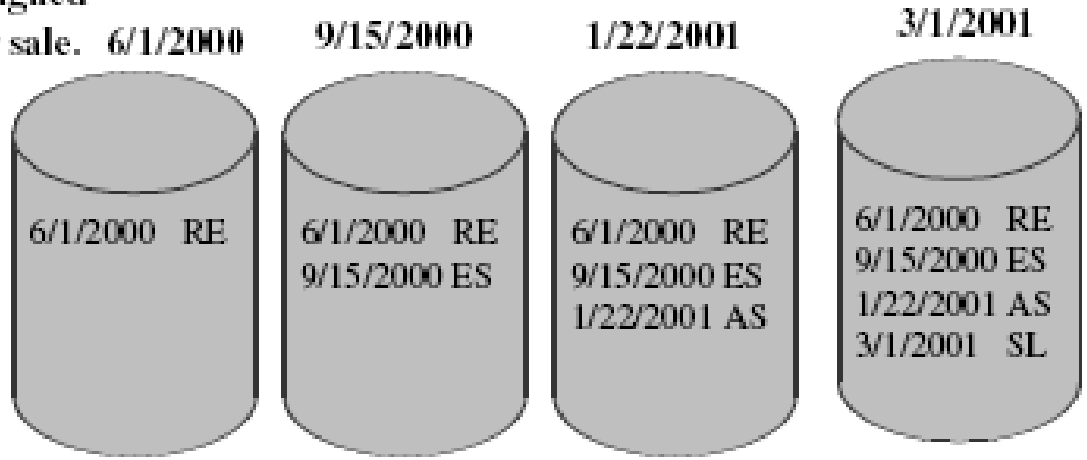


Figure 12-3 Data in operational systems.

Data in Operational Systems.

- When you deploy your data warehouse, the initial data as of a certain time must be moved to the data warehouse to get it started. This is the initial load.
- After the initial load, your data warehouse must be kept updated so the history of the changes and statuses are reflected in the data warehouse. There are two major types of data extractions from the source operational systems:
  - "as is" (static) data
  - data of revisions. (explanation follows)

Data in Operational Systems.

- "As is" or static data is the capture of data at a given point in time. It is like taking a snapshot of the relevant source data at a certain point in time.
- Data of revisions is also known as incremental data capture. Incremental data capture may be immediate or deferred. Within the group of immediate data capture there are three distinct options.

Options for data capture

- Immediate Data Extraction.
  - Capture through Transaction Logs.
  - Capture through Database Triggers.
  - Capture in Source Applications.
- Deferred Data Extraction.
  - Capture Based on Date and Time Stamp.
  - Capture by Comparing Files.

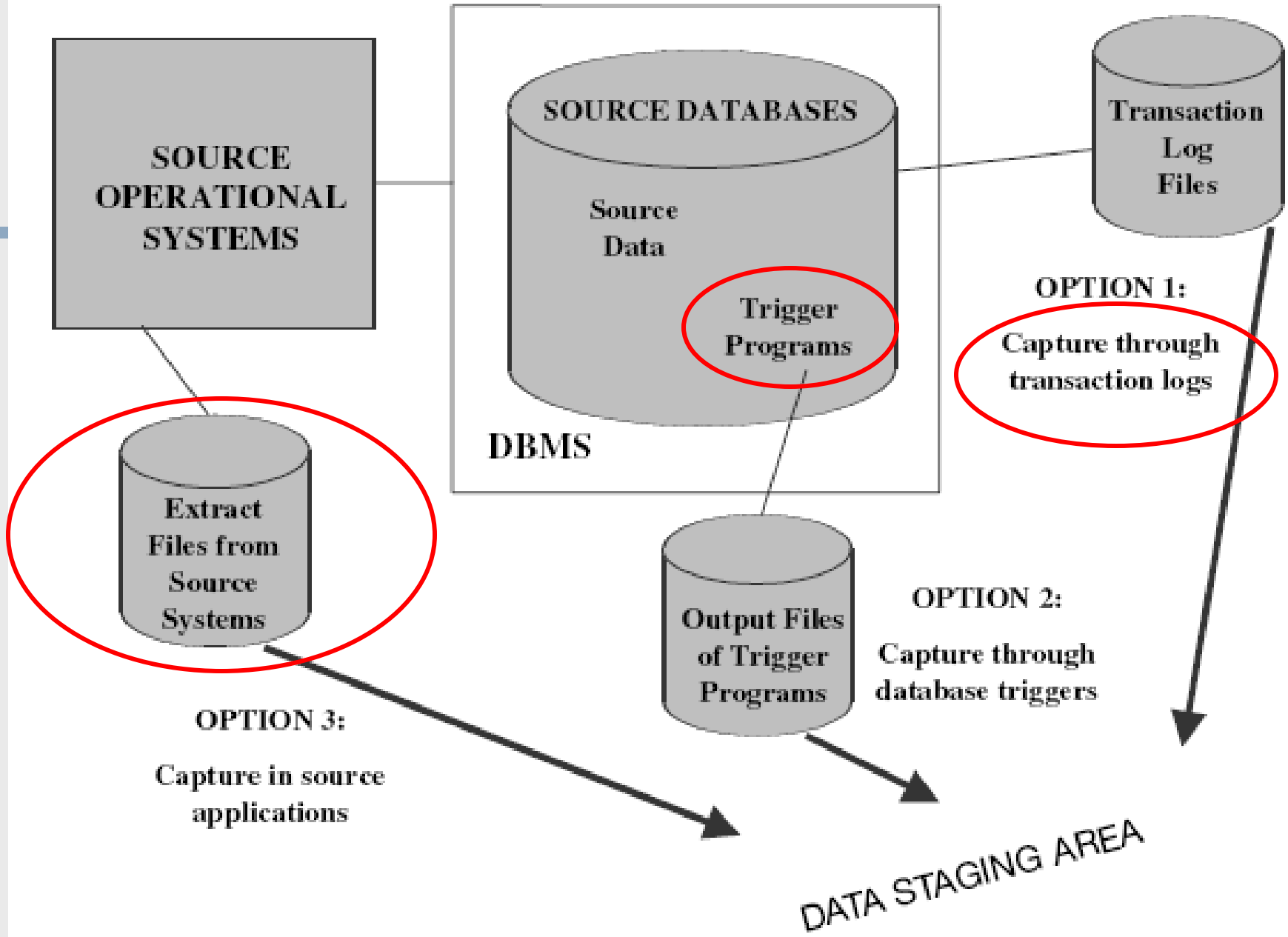


Figure 12-4 Immediate data extraction: options.

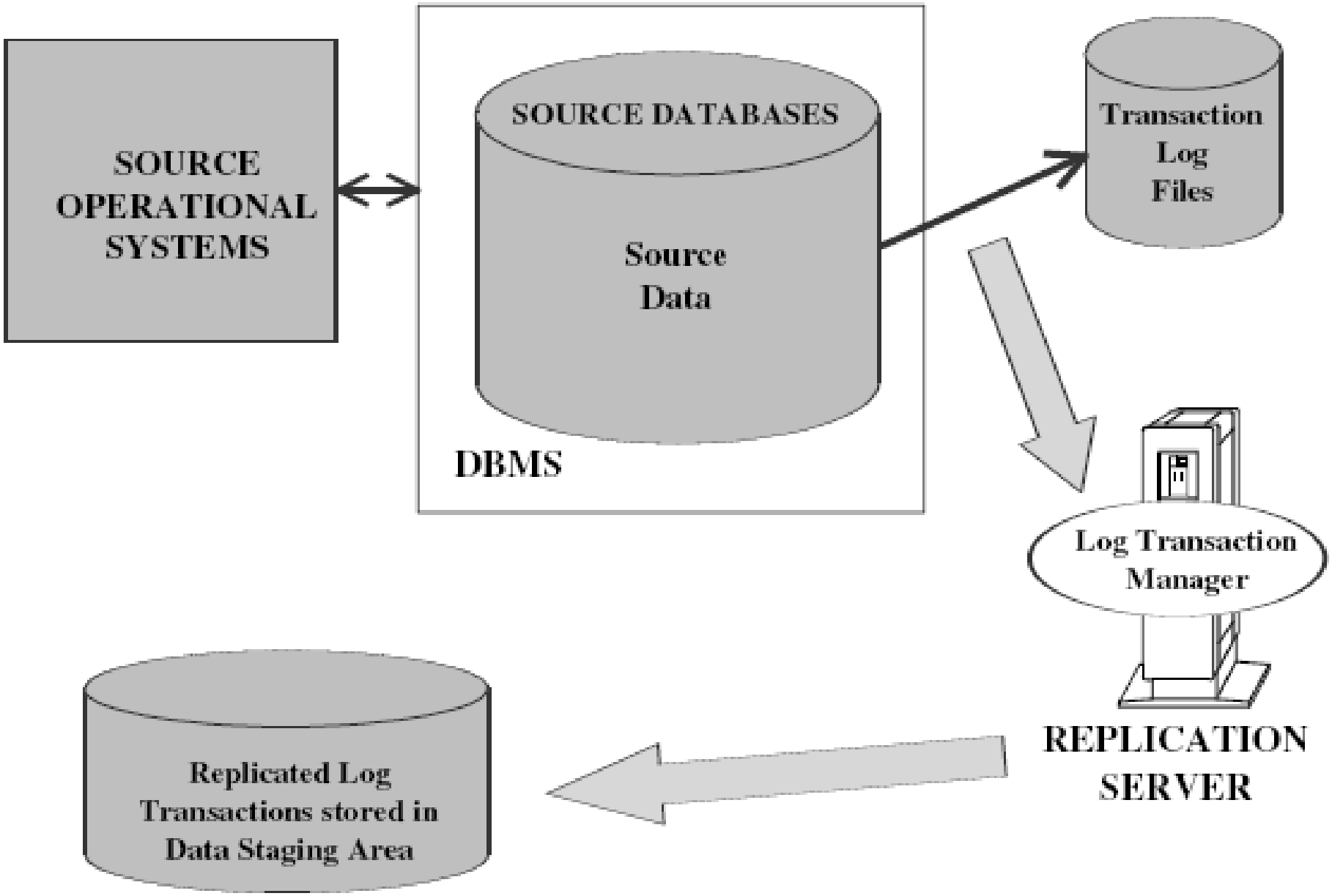


Figure 12-5 Data extraction: using replication technology.

Data capture through database triggers

- Data capture through database triggers occurs right at the source and is therefore quite reliable.
- You can capture both before and after images.
- Building and maintaining trigger programs puts an additional burden on the development effort.
- Execution of trigger procedures during transaction processing of the source systems puts additional overhead on the source systems.
- This option is applicable only for source data in databases.

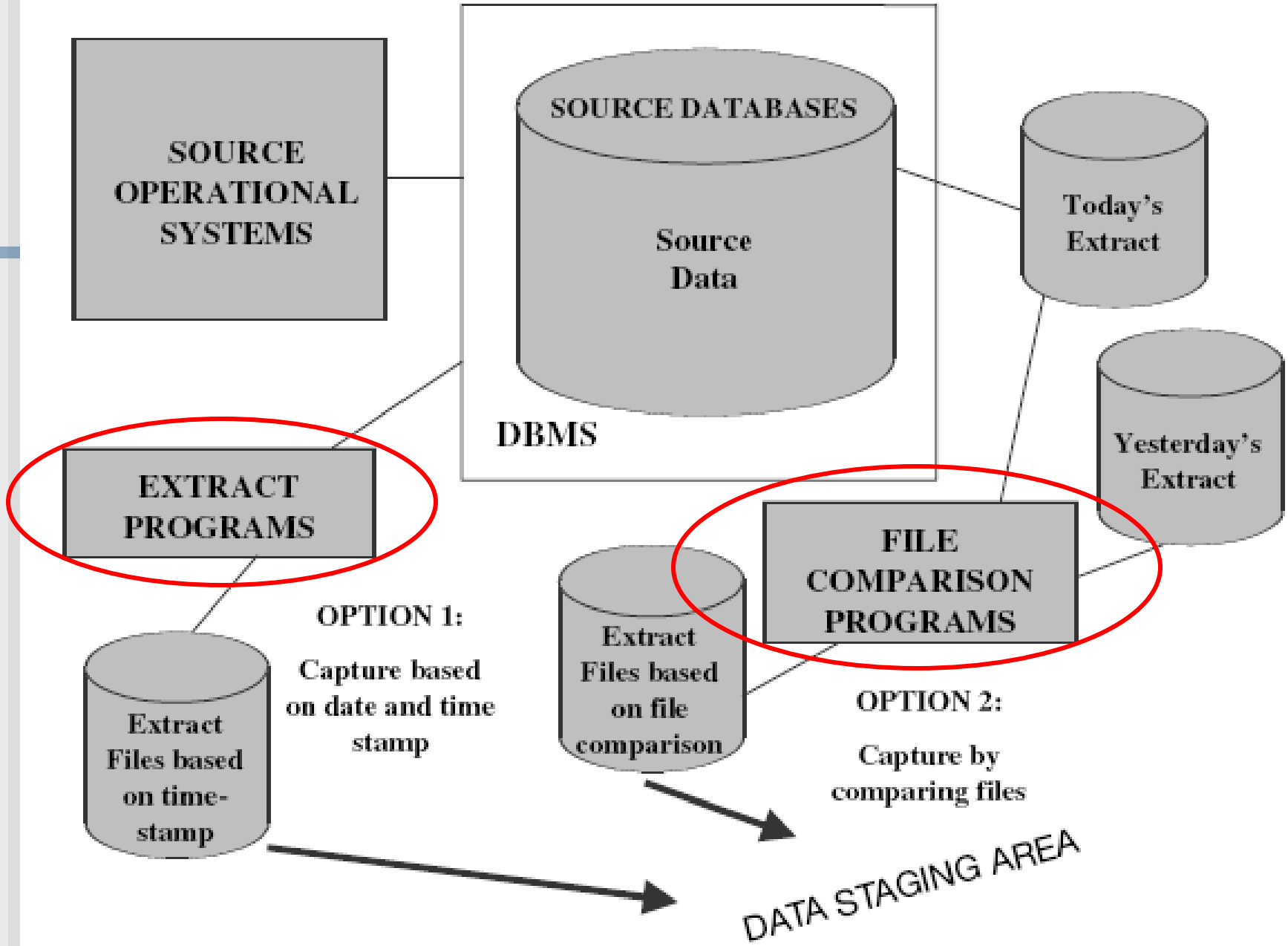


Figure 12-6 Deferred data extraction: options.

Capture Based on Date and Time Stamp

- Deletion of source records presents a special problem. If a source record gets deleted in between two extract runs, the information about the delete is not detected.
- You can get around this by marking the source record for delete first, do the extraction run, and then go ahead and physically delete the record.
- This means you have to add more logic to the source applications.

Capture by Comparing Files

- If none of the above techniques are feasible for specific source files in your environment, then consider this technique as the last resort.
- This technique is also called the snapshot differential technique because it compares two snapshots of the source data.

Capture of static data

Good flexibility for capture specifications.
Performance of source systems not affected.
No revisions to existing applications.
Can be used on legacy systems.
Can be used on file-oriented systems.
Vendor products are used. No internal costs.

Capture in source applications

Good flexibility for capture specifications.
Performance of source systems affected a bit.
Major revisions to existing applications.
Can be used on most legacy systems.
Can be used on file-oriented systems.
High internal costs because of in-house work.

Capture through transaction logs

Not much flexibility for capture specifications.
Performance of source systems not affected.
No revisions to existing applications.
Can be used on most legacy systems.
Cannot be used on file-oriented systems.
Vendor products are used. No internal costs.

Capture based on date and time stamp

Good flexibility for capture specifications.
Performance of source systems not affected.
Major revisions to existing applications likely.
Cannot be used on most legacy systems.
Can be used on file-oriented systems.
Vendor products may be used.

Capture through database triggers

Not much flexibility for capture specifications.
Performance of source systems affected a bit.
No revisions to existing applications.
Cannot be used on most legacy systems.
Cannot be used on file-oriented systems.
Vendor products are used. No internal costs.

Capture by comparing files

Good flexibility for capture specifications.
Performance of source systems not affected.
No revisions to existing applications.
May be used on legacy systems.
May be used on file-oriented systems.
Vendor products are used. No internal costs.

Figure 12-7 Data capture techniques: advantages and disadvantages.

DATA TRANSFORMATION

DATA TRANSFORMATION

- Extracted data is raw data and it cannot be applied to the data warehouse
- All the extracted data must be made usable in the data warehouse.

Quality of data

- Major effort within data transformation is the improvement of data quality.
- This includes filling in the missing values for attributes in the extracted data.
- Data quality is of *paramount importance* in the data warehouse because the effect of strategic decisions based on incorrect information can be devastating.

Basic tasks in data transformation

- **Selection -** beginning of the whole process of data transformation. Select either whole records or parts of several records from the source systems.
- **Splitting/joining -** types of data manipulation needed to be performed on the selected parts of source records. Sometimes (uncommonly), you will be splitting the selected parts even further during data transformation. Joining of parts selected from many source systems is more widespread in the data warehouse environment.
- **Conversion -** all-inclusive task. It includes a large variety of rudimentary conversions of single fields for two primary reasons—one to standardize among the data extractions from disparate source systems, and the other to make the fields usable and understandable to the users.

Basic tasks in data transformation(2)

- **Summarization.** Sometimes it is not feasible to keep data at the lowest level of detail in the data warehouse. It may be that none of users ever need data at the lowest granularity for analysis or querying.
- **Enrichment -** rearrangement and simplification of individual fields to make them more useful for the data warehouse environment. You may use one or more fields from the same input record to create a better view of the data for the data warehouse. This principle is extended when one or more fields originate from multiple records, resulting in a single field for the data warehouse.

Major Transformation Types

- Format Revisions
- Decoding of Fields
- Calculated and Derived Values.
- Splitting of Single Fields.
- Merging of Information.
- Character Set Conversion.
- Conversion of Units of Measurements
- Date/Time Conversion.
- Summarization.
- Key Restructuring.
- Deduplication.

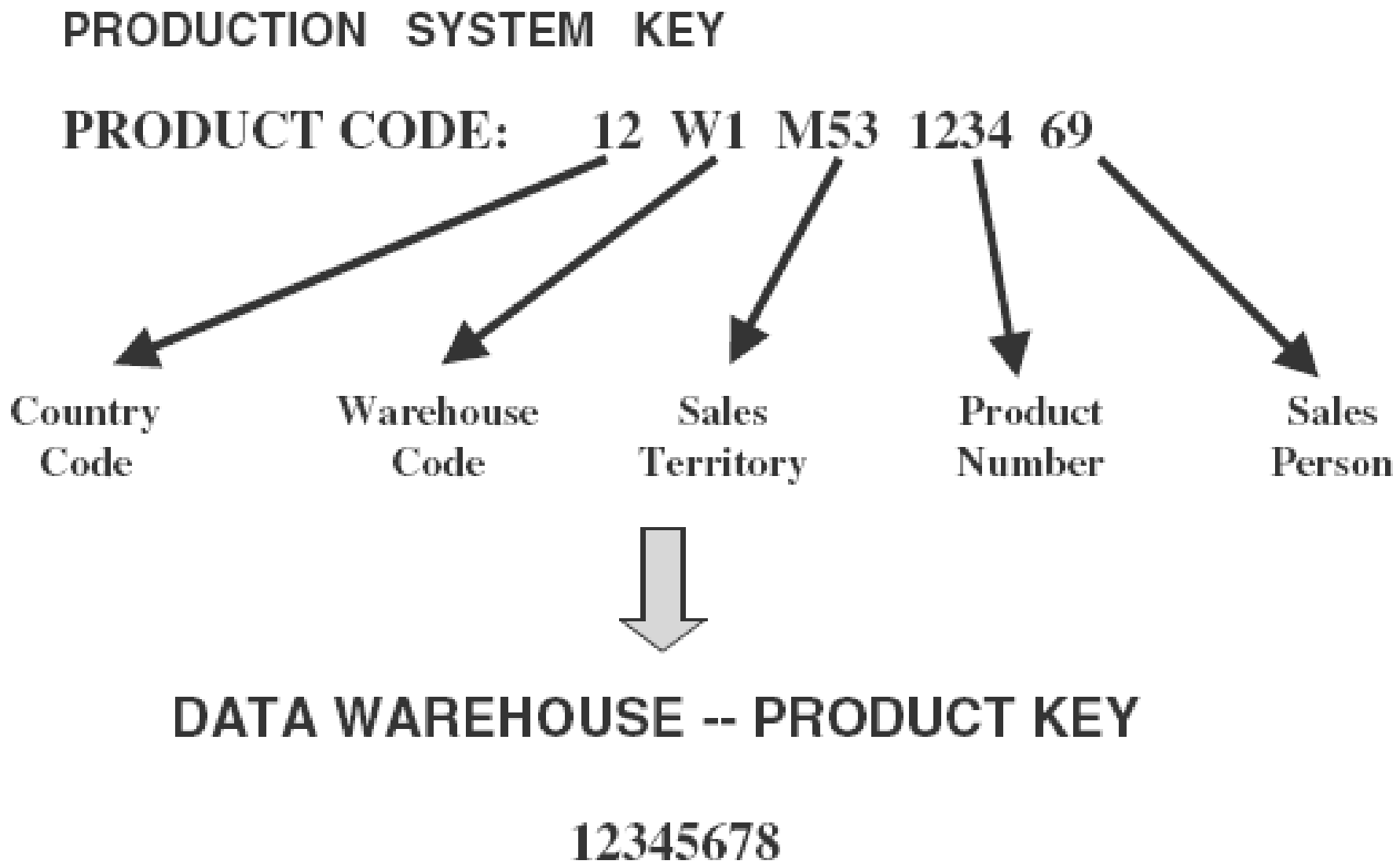


Figure 12-8 Data transformation: key restructuring.

Data Integration and Consolidation

- ***Entity Identification Problem***
- ***Multiple Sources Problem***

Entity Identification Problem

- If you have three different legacy applications developed in your organization at different times in the past, you are likely to have three different customer files supporting those systems.
- Most of the customers will be common to all three files.
- The same customer on each of the files may have a unique identification number.
- These unique identification numbers for the same customer may not be the same across the three systems.

- **Solution** - complex algorithms have to be designed to match records from all the three files and form groups of matching records. No matching algorithm can completely determine the groups. If the matching criteria are too tight, then some records will escape the groups. On the other hand, if the matching criteria are too loose, a particular group may include records of more than one customer.

Multiple Sources Problem

- Single data element having more than one source.
- A **straightforward solution** is to assign a higher priority to one of the two sources and pick up the data element from that source. Sometimes, a straightforward solution such as this may not sit well with needs of the data warehouse users. You may have to select from either of the files based on the last update date. Or, in some other instances, your determination of the appropriate source depends on other related fields

DATA LOADING

DATA LOADING

- Data loading takes the prepared data, applies it to the data warehouse, and stores it in the database
- Terminology:
  - **Initial Load** — populating all the data warehouse tables for the very first time
  - **Incremental Load** — applying ongoing changes as necessary in a periodic manner
  - **Full Refresh** — completely erasing the contents of one or more tables and reloading with fresh data (initial load is a refresh of all the tables)

Applying Data: Techniques and Processes

- load,
- append,
- destructive merge,
- constructive merge.

Load

- If the target table to be loaded already exists and data exists in the table, the load process wipes out the existing data and applies the data from the incoming file.
- If the table is already empty before loading, the load process simply applies the data from the incoming file.

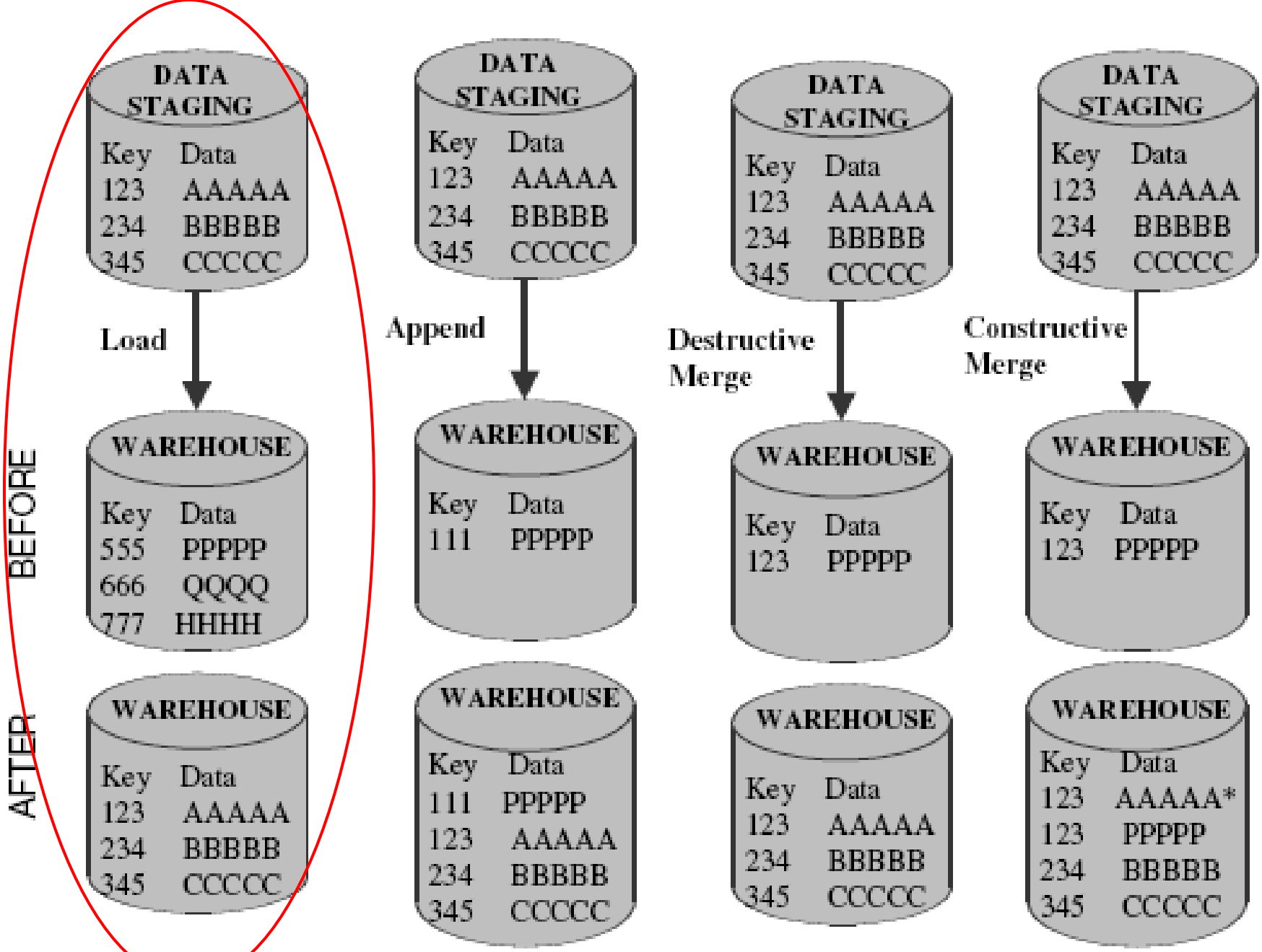


Figure 12-11 Modes of applying data.

Append

- extension of the load.
- If data already exists in the table, the append process unconditionally adds the incoming data, preserving the existing data in the target table.
- When an incoming record is a duplicate of an already existing record, you may define how to handle an incoming duplicate:
  - The incoming record may be allowed to be added as a duplicate.
  - In the other option, the incoming duplicate record may be rejected during the append process.

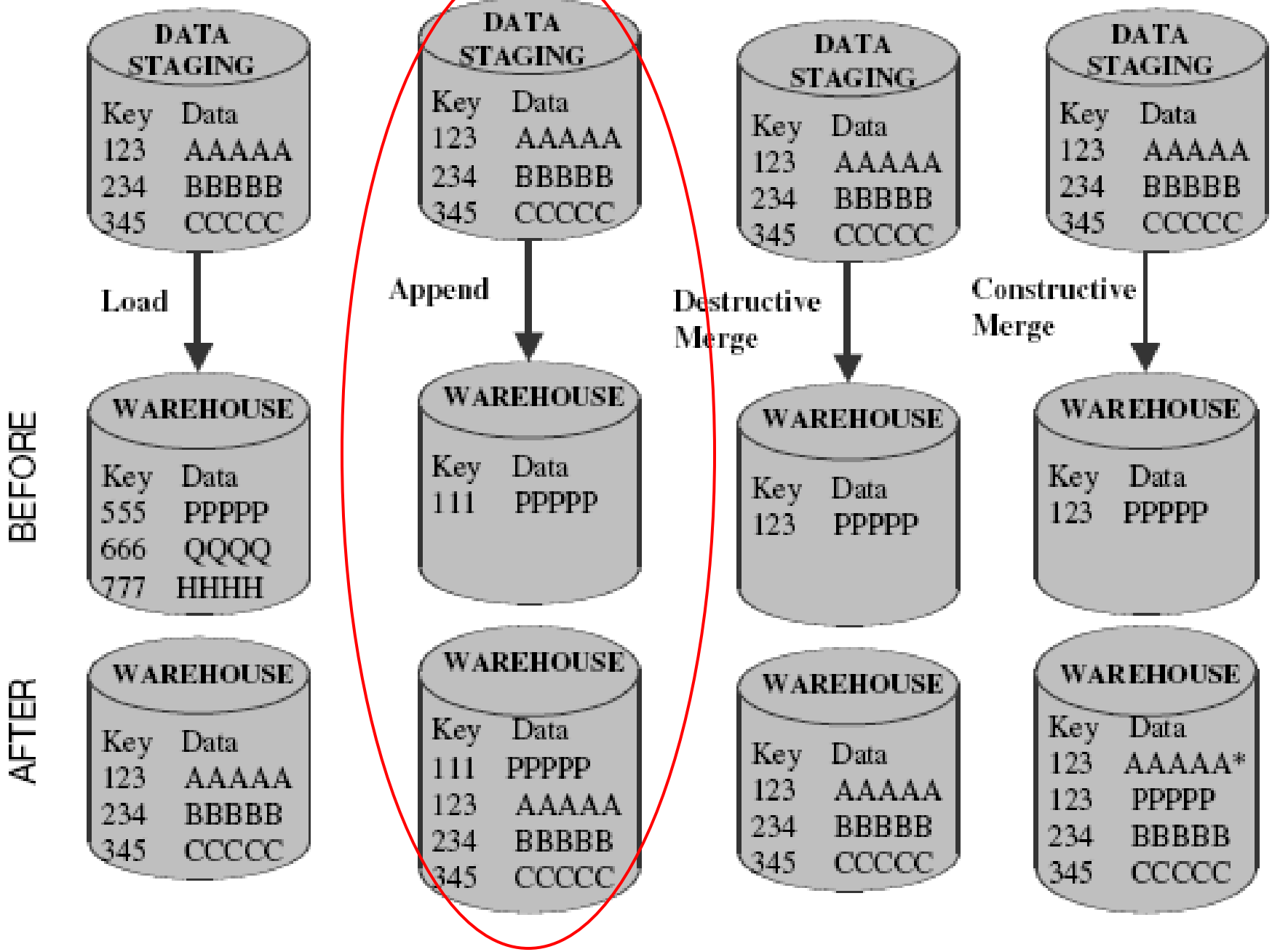


Figure 12-11 Modes of applying data.

Destructive Merge

- Applies incoming data to the target data.
- If the primary key of an incoming record matches with the key of an existing record, update the matching target record.
- If the incoming record is a new record without a match with any existing record, add the incoming record to the target table.

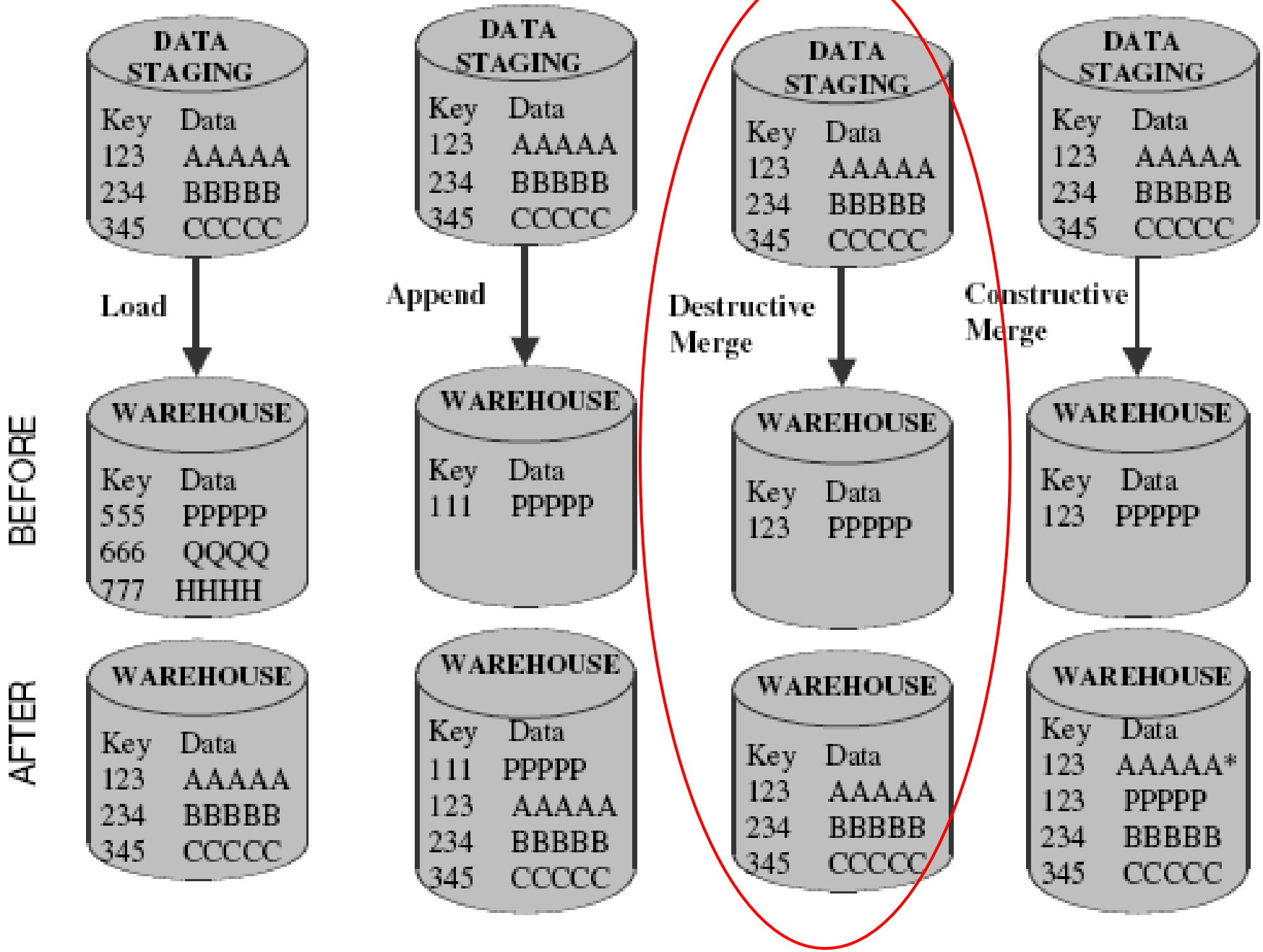


Figure 12-11 Modes of applying data.

Constructive Merge

- Slightly different from the destructive merge.
- If the primary key of an incoming record matches with the key of an existing record, leave the existing record, add the incoming record, and mark the added record as superceding the old record.

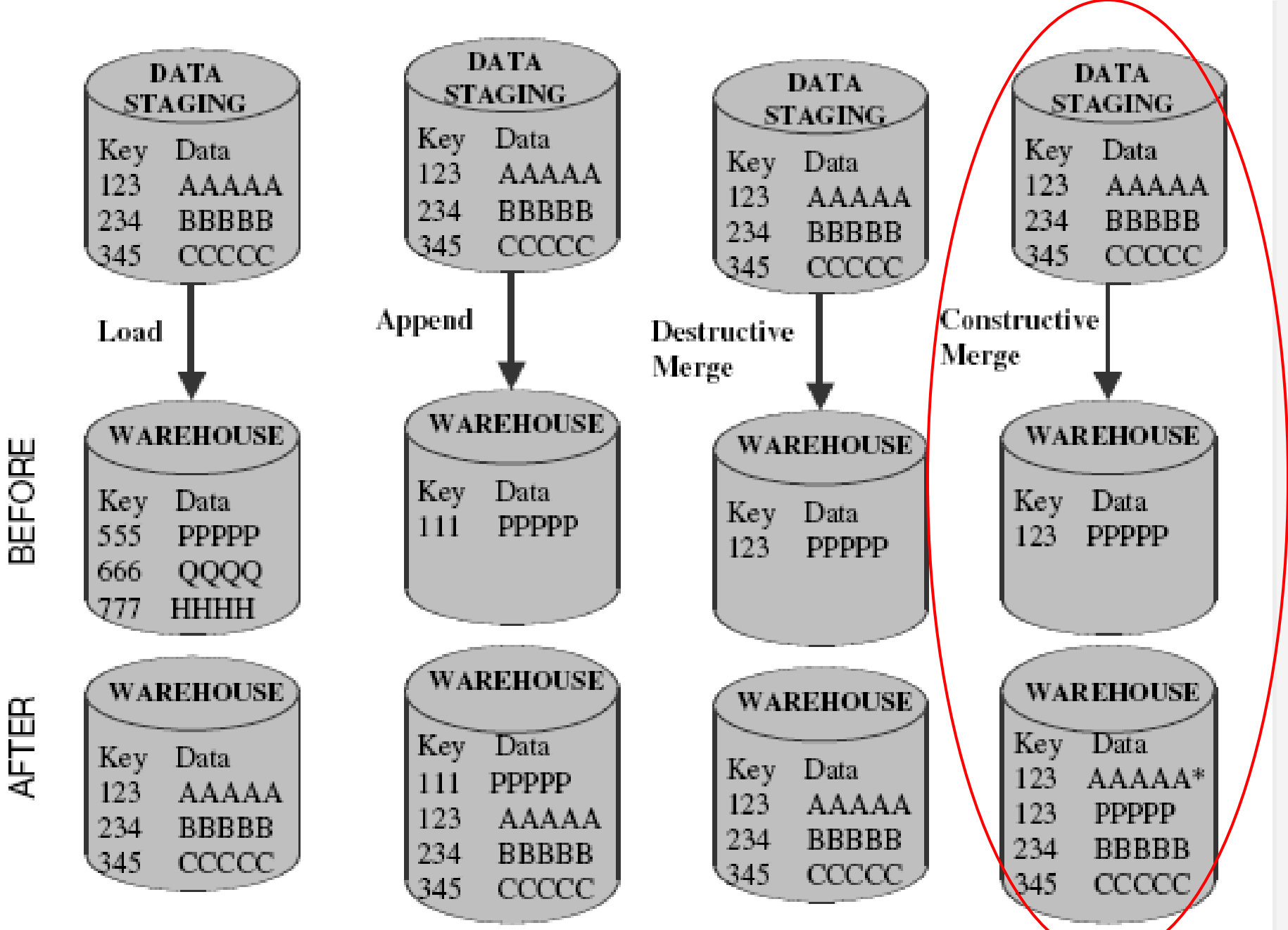


Figure 12-11 Modes of applying data.

ETL Tools Options

- Data transformation engines
- Data capture through replication
- Code generators

Data transformation engines

- Consist of dynamic and sophisticated data manipulation algorithms.
- The tool suite captures data from a designated set of source systems at user-defined intervals, performs elaborate data transformations, sends the results to a target environment, and applies the data to target files.
- These tools provide maximum flexibility for pointing to various source systems, to select the appropriate data transformation methods, and to apply full loads and incremental loads.
- The functionality of these tools sweeps the full range of the ETL process.

Data capture through replication

- Most of these tools use the transaction recovery logs maintained by the DBMS.
- The changes to the source systems captured in the transaction logs are replicated in near real time to the data staging area for further processing.
- Some of the tools provide the ability to replicate data through the use of database triggers. These specialized stored procedures in the database signal the replication agent to capture and transport the changes.

Code generators

- Tools that directly deal with the extraction, transformation, and loading of data.
- The tools enable the process by generating program code to perform these functions.
- Code generators create 3GL/4GL data extraction and transformation programs.
- The tools generate most of the program code in some of the common programming languages.
- Own program code can be addedd, also.
- The code automatically generated by the tool has exits at which points you may add your code to handle special conditions.

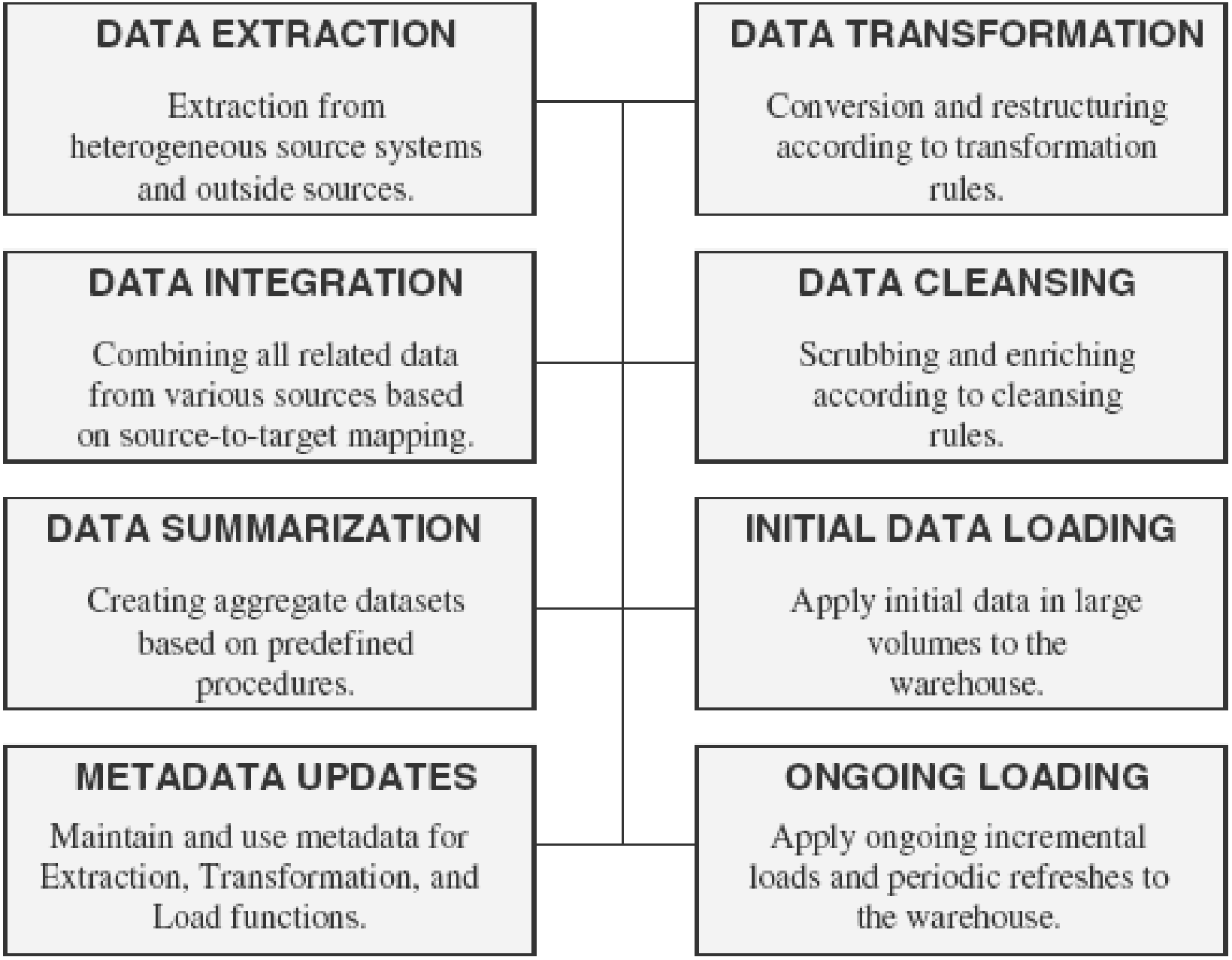


Figure 12-14 ETL summary.