Advanced Databases

- M. K. Valvi

Data Warehouse:
ETL

ETL Overview

- Data Warehouse environment is divided into three functional areas:
  - Data Acquisition,
  - Data Storage and
  - Information Delivery
- Data extraction, data transformation, and data loading encompass the areas of data acquisition and data storage
- ETL functions reshape the relevant data from the source systems into useful information to be stored in the data warehouse

ETL Overview

- Back-end processes and functions that covers:
  - extraction of data from the source systems
  - Changing the source data into the exact formats and structures appropriate for storage in DW
  - Physically moving the data into the DW repository
- Data extraction : Scope and Extent?
- Data extraction presupposes a selection process
- The extent and complexity of the back-end processes differ from one data warehouse to another

ETL Overview

- Each of the ETL function is important and essential. Each function fulfils a significant purpose
- All of the functions must be performed in sequence for successfully transforming data into strategic information or business intelligence
- ETL functions are challenging primarily because of the nature of the source systems
- ETL processes are Time Consuming and Arduous

List of reasons for the types of difficulties in ETL functions

- Source systems are very diverse and disparate
- Need to deal with source systems on multiple platforms and different operating systems
- older legacy applications running on obsolete database technologies
- Historical data on changes in values are not preserved
- Quality of data is dubious in many old source systems
- Source system structures keep changing over time
- Gross lack of consistency among source systems is prevalent
- Lack of means for resolving mismatches escalates the problem of inconsistency
- Most source systems do not represent data in types or formats that are meaningful to the users

ETL Steps

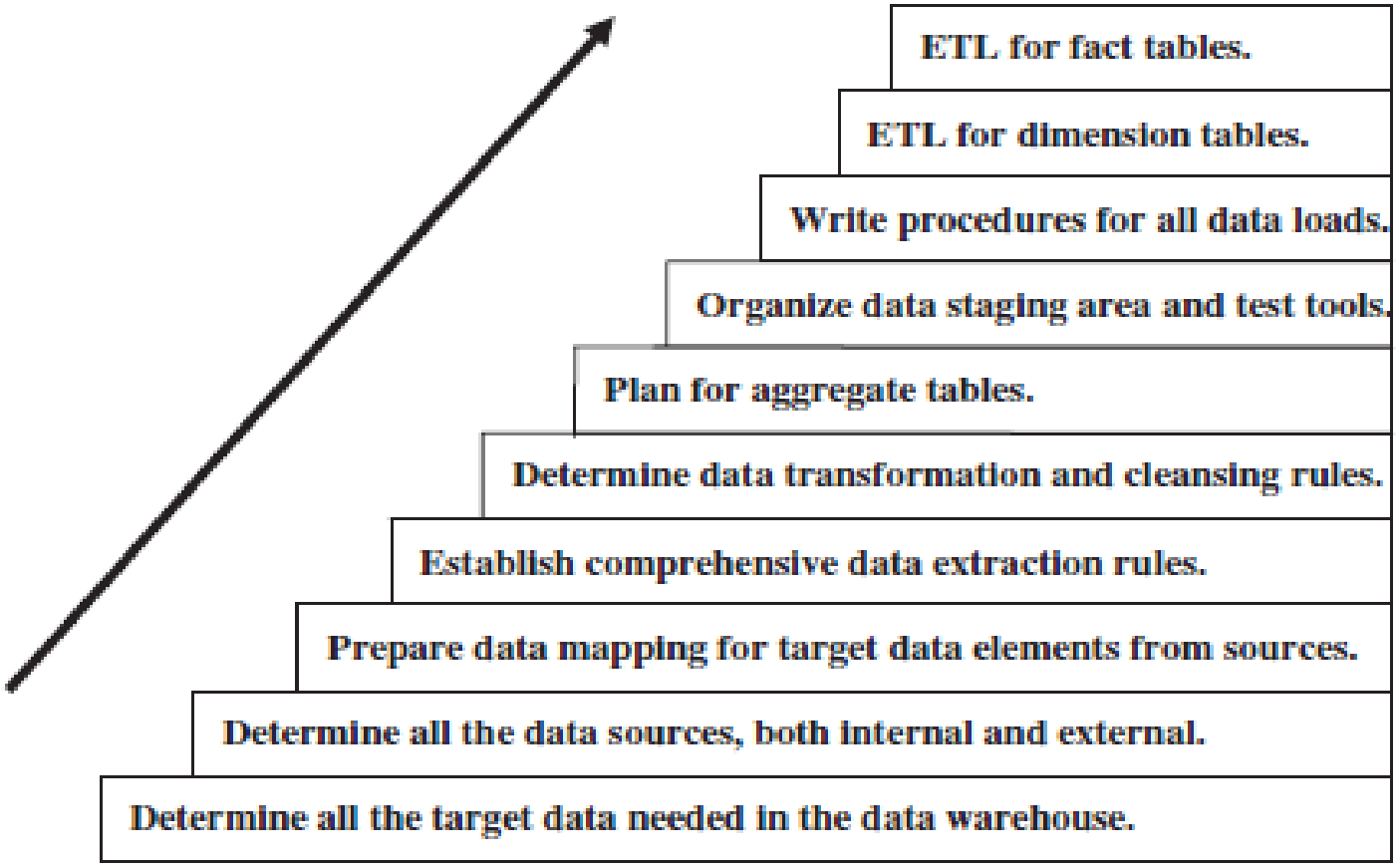

Figure 12-1 Major steps in the ETL process.

Data Extraction

- Factors increasing complexity of Data Extraction
  - First, for a data warehouse, you have to extract data from many disparate sources
  - Second, for a data warehouse, you have to extract data on the changes for ongoing incremental loads as well as for a one-time initial full load
- Use of third-party data extraction tools in addition to in-house programs or scripts is recommended

Data Extraction

- 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 be done: daily, weekly, quarterly, and so on
  - Time window
    for each data source, denote the time window for the extraction process

Data Extraction

- Data Extraction issues
  - 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

Identification of Data Sources

- Nature of the source data and its intended use must be understood
- Business transactions keep changing the data in the source systems
- Data in the source systems are said to be time-dependent or temporal, value of a single variable varies over time
- Every change is the source system must be identified by data Warehouse (Capturing the history)
- Can be done by knowing how source systems store the data

Identification of Data Sources

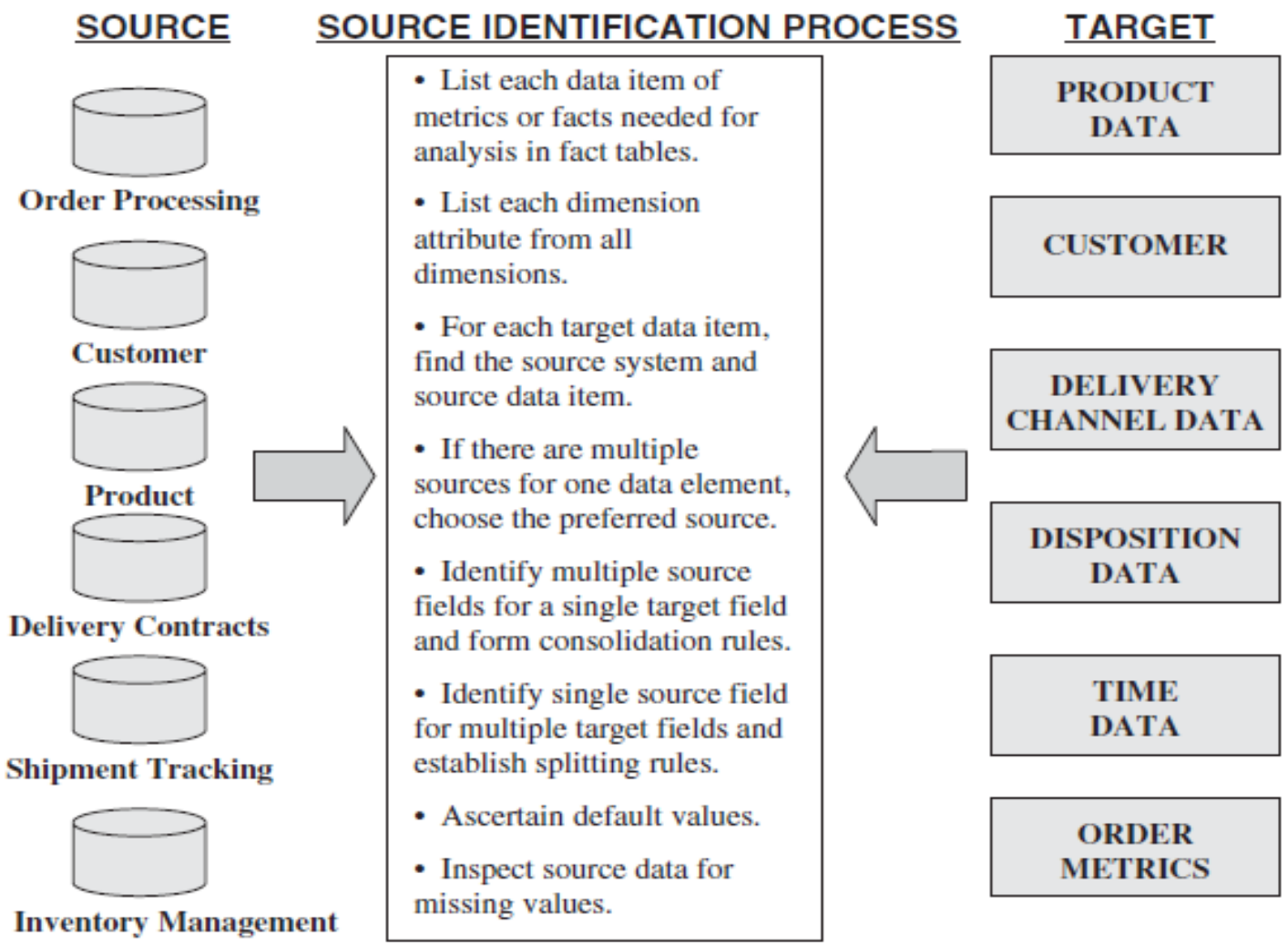

Figure 12-2 Source identification: a stepwise approach.

Data in Operational Sources

- Current Value
  - The stored value of an attribute represents the value of the attribute at this moment of time
  - Transient/Transitory values, changes with business transaction
  - Change in value cannot be predicted
  - Data extraction for preserving the history of the changes in the data warehouse gets quite involved for this category of data
  - Most of the attributes in the source systems fall into this category
  - E.g. Customer name and address, bank account balances, and outstanding amounts on individual orders etc.

Data in Operational Sources

- Periodic Status
  - Not as common as Current Value
  - The value of the attribute is preserved as the status every time a change occurs
  - The status value is stored with reference to the time when the new value became effective
  - For operational data in this category, the history of the changes is preserved in the source systems themselves
  - Whether it is status data or data about an event, the source systems contain data at each point in time when any change occurred
  - E.g. Data about an Insurance policy, bonus accumulated over time

Data in Operational Sources

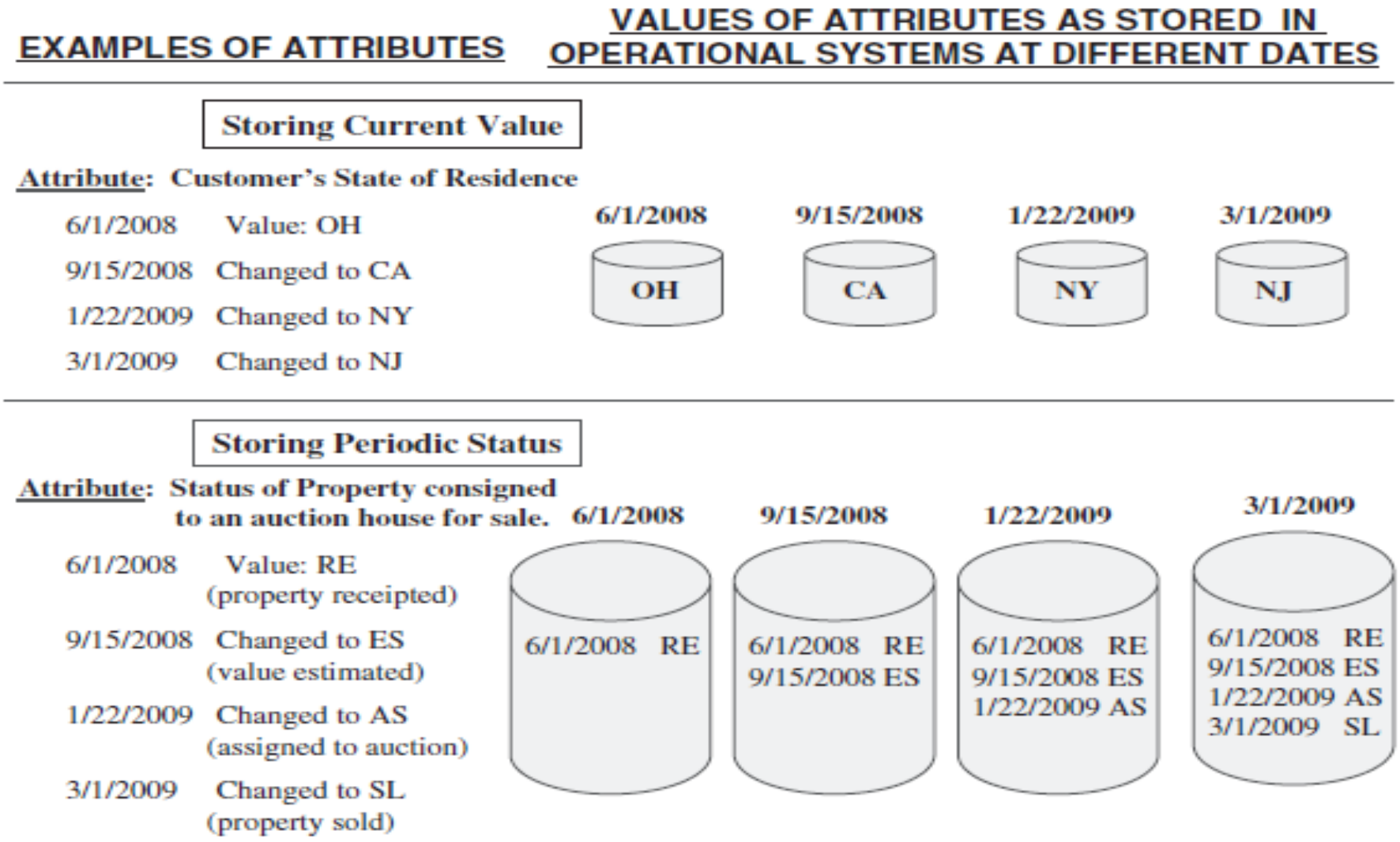

Figure 12-3 Data in operational systems.

Types of Data Extractions

- "as is" (static) Data
  - Capture of data at a given point in time
  - Static data capture is primarily used for the initial load of the data warehouse
- Data of revisions
  - also known as incremental data capture
  - Strictly not incremental but revisions since last data capture
  - Difficult if source of data is transient
  - Incremental data capture may be immediate or deferred

Immediate Data Extraction

- Data extraction is real time
- It occurs as the transactions happen at the source databases and files
- Three options of immediate extractions
  - Capture through transaction logs
  - Capture through database triggers
  - Capture in source applications

Immediate Data Extraction

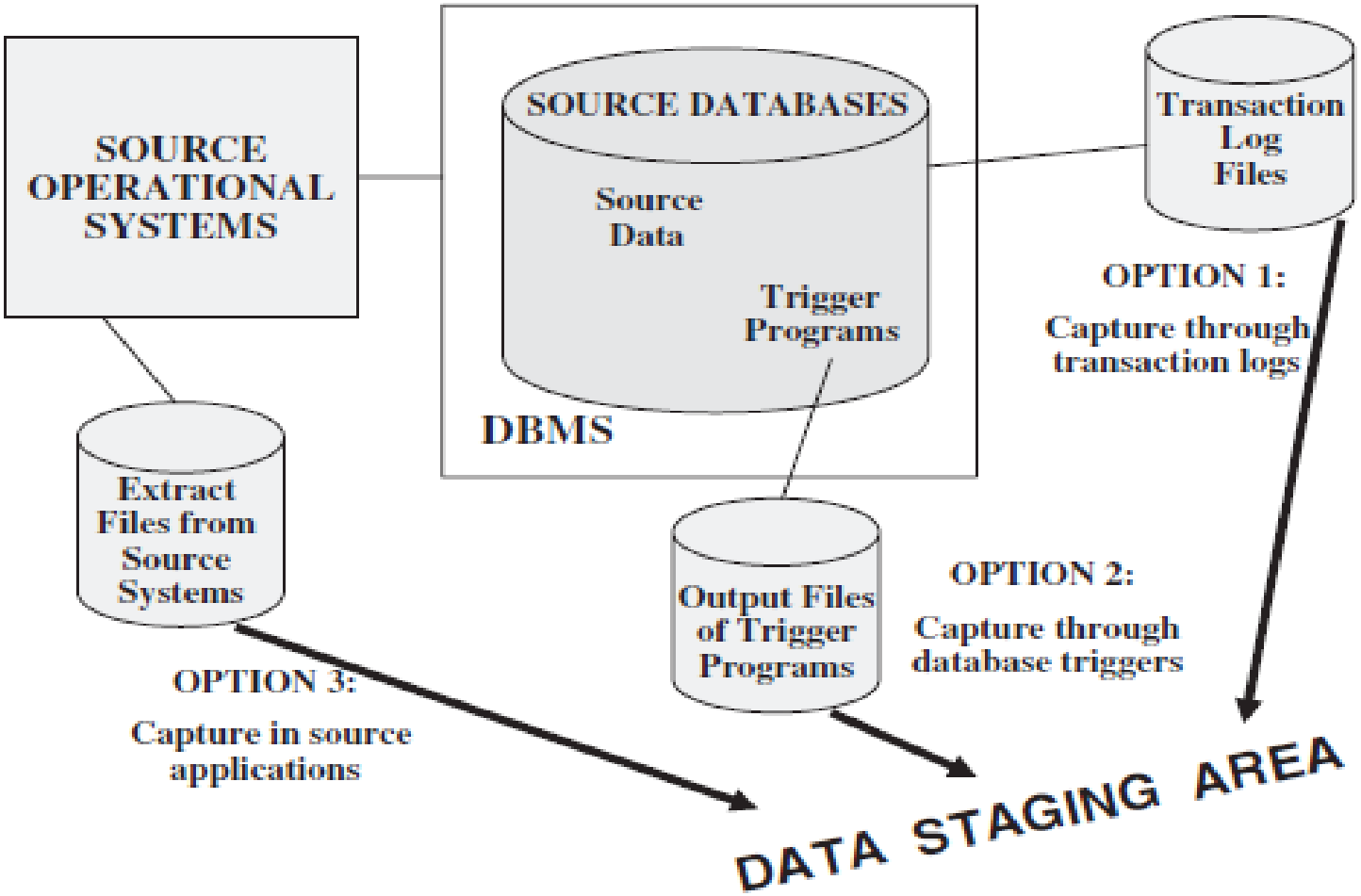

Figure 12-4 Options for immediate data extraction.

Immediate Data Extraction

- Capture through transaction logs
  - Uses the transaction logs of the DBMSs maintained for recovery from possible failures
  - This data extraction technique reads the transaction log and selects all the committed transactions
  - No extra overhead in the operational systems because logging is already part of the transaction processing
  - All log transactions must be extracted for DW
  - This option won't work if source system data is on indexed and other flat files as no log files for non-database applications

Immediate Data Extraction

- Capture through Database Triggers
  - This option is also applicable to the source systems that are database applications
  - Trigger programs can be created for all events for which data needs to be captured
  - The output of the trigger programs is written to a separate file that will be used to extract data for the data warehouse
  - Data capture through database triggers occurs right at the source and is therefore quite reliable
  - Additional burden of building and maintaining triggers and execution of it during transaction processing

Immediate Data Extraction

- Capture in Source Applications
  - Also referred to as application assisted data capture
  - The source application is made to assist in the data capture for the data warehouse
  - Relevant application programs needs to be modified that write to source files and databases
  - This technique may be used for all types of source data irrespective of whether it is in databases, indexed files, or other flat files
  - Programs in the source operational systems needs to be revised and keep them maintained
  - Difficult if number of source system program is large, also may degrade the performance of source applications

Deferred Data Extraction

- Data extraction is not real time
- Two options of deferred extractions
  - Capture Based on Date and Time Stamp
  - Capture by Comparing Files

Deferred Data Extraction

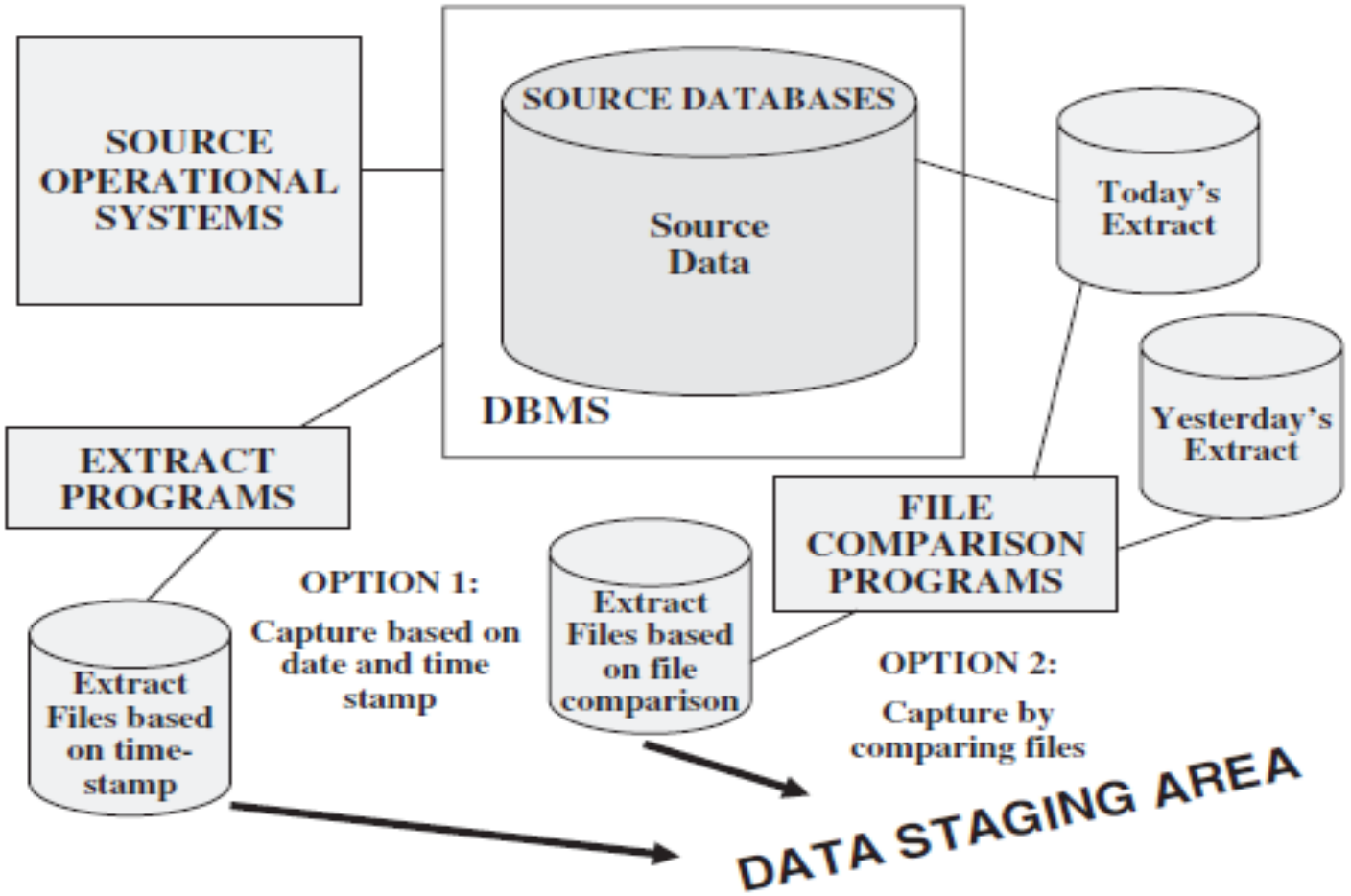

Figure 12-6 Options for deferred data extraction.

Deferred Data Extraction

- Capture Based on Date and Time Stamp
  - Every time a source record is created or updated it may be marked with a stamp showing the date and time. The time stamp provides the basis for selecting records for data extraction
  - This technique works well if the number of revised records is small
  - This technique presupposes that all the relevant source records **contain date and time stamps**
  - Works for any type of source file, given date and time stamps are recorded
  - Captures the latest state of the source data. Any intermediary states between two data extraction runs are lost

Deferred Data Extraction

- Capture Based on Date and Time Stamp
  - Extra logic needs to be added for certain cases like deletion of source record
- Capture by comparing Files
  - Last resort, if none of the techniques mentioned previously work for any of the source files
  - Also called the snapshot differential technique because it compares two snapshots of the source data
  - Full file comparison is done between the two copies (Prior and Later), then changes between the two is captured
  - This technique necessitates the keeping of prior copies of all the relevant source data

Deferred Data Extraction

- Capture by comparing Files
  - Simple and straightforward
  - Inefficient for large files
  - This may be the only feasible option for some legacy data sources that do not have transaction logs or time stamps on source records

Data Extraction Techniques

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.

Data Transformation

Basic Tasks

- Selection or filtering of records
  - Takes place at the beginning
  - Either whole records or parts of several records from the source systems are selected to meet the predefined criterian
  - Task of selection usually forms part of the extraction function itself
  - In the cases where composition of the source structure may not be amenable to selection of the necessary parts during data extraction, it is prudent to extract the whole record and then do the selection as part of the transformation function
- Data cleansing
  - Removing duplicates from a dataset.
  - the duplicates can skew analysis results and waste storage space. Removing them ensures data integrity and accuracy.

Basic Tasks

- Splitting / Joining
  - Includes the types of data manipulation needed to perform on the selected parts of source records
  - Sometimes, uncommonly, splitting of the selected parts even further during data transformation is needed
  - Joining of parts selected from many source systems is more widespread in the data warehouse environment

splitting

Address 123 Main St, Springfield, IL 62701
456 Elm St, Anytown, NY 12345

Street Address | City | State | Zip Code
---
123 Main St | Springfield| IL | 62701
456 Elm St | Anytown | NY | 12345

joining

Order ID | Product ID | Quantity | Total
---
101 | P001 | 2 | $100
102 | P002 | 1 | $50

Product ID | Product Name | Category | Price -

P001 | Laptop | Electronics | $500
P002 | Smartphone | Electronics | $400

rder ID | Product ID | Quantity | Total | Product Name | Category | Price
---
101 | P001 | 2 | $100 | Laptop | Electronics | $500
102 | P002 | 1 | $50 | Smartphone | Electronics | $400

Basic Tasks

- Conversion(eg. Name of person, date etc.)
  - An all-inclusive task
  - Includes a large variety of rudimentary conversions of single fields for two primary reasons
    - ✓ to standardize among the data extractions from disparate source systems, and
    - ✓ to make the fields usable and understandable to the users

Country
United States
USA
U.S.
United Kingdom
UK
United Kingd

Standardized Country
United States
United States
United States
United Kingdom
United Kingdom
United Kingdom

Converting units of measurement (e.g., converting pounds to kilograms) for consistency and comparability.

Basic Tasks

- Summarization
  - Sometimes it is not feasible to store the data at the lowest level of detail in DW, for such cases summarization is used
  - Maybe none of the users ever need the data at the lowest granularity for analysis or querying
  - For example, for a grocery chain, sales data at the lowest level of detail for every transaction at the checkout may not be needed, storing sales by product by store by day in the data warehouse is sufficient

Calculating total sales by summing up individual sales transactions.
such aggregation of data allows for the analysis of trends and patterns at a higher level of granularity.

Basic Tasks

- Enrichment
  - This task is the rearrangement and simplification of individual fields to make them more useful for the data warehouse environment
  - One or more fields from the same input record can be used 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

Example: Adding geographical information (e.g., latitude and longitude) based on postal codes enhances its value for analysis and reporting purposes.

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.

Implementing Transformation

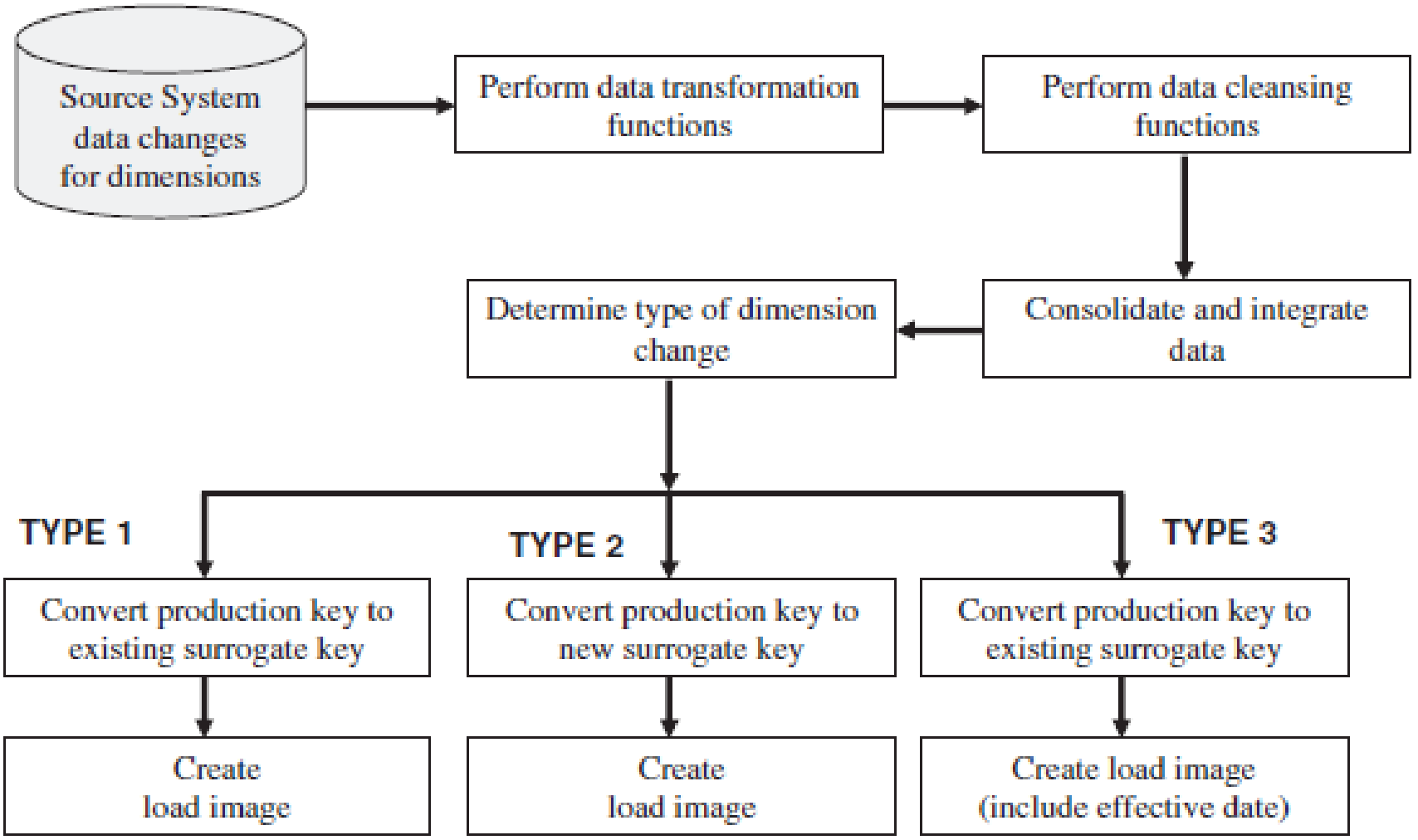

Figure 12-10 Transformed for dimension changes.

Data Loading

Data Loading Types of applying data

- 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).

Data Loading

- Of great concern as loading may take inordinate amount of time
- Data warehouse goes offline during loads so window of time must be identified for scheduling loads without affecting the work
- Whole load process can be divided into smaller chunks and few files can be populated at a time, with this
  - Smaller loads can run in parallel
  - Some parts of DW can be up and running while other part are being loaded
- Its hard to estimate running times of loads, especially during initial load or a complete refresh

Data Loading

- Specific procedure needs to be provided for the records not being successfully applied to DW
  - Maybe having wrong concatenated key or not corresponding to the dimension tables
- Transport of load images need to be handled if staging area and DW database are not on same server
  - Option needs to be selected (web, FTP, Database links). Also bandwidth needed needs to be considered. Have contingency plans and data compressions.
- Load utilities provided by DBMSs can be used instead of special load programs. DW size must be considered
- Project team must be capable of identifying and handling challenges in loading

Applying Data

- A file of data is created which is applied to the product dimension table in the data warehouse

- Data maybe applied in four modes
  - Load
  - Append
  - Destructive Merge
  - Constructive Merge

Modes of Applying Data

- Load
  - If target table already exists with some data, the data is wiped out and data from incoming file is applied
  - If target table is empty, load process will simply apply the data from the incoming file
- Append
  - Can be seen as an extension of Load
  - If target table already exists with some data, the append process unconditionally adds the data from incoming file, existing data is preserved

Modes of Applying Data

- Append
  - If incoming record is duplicate of existing record, it can be handled as
    - ✓ Incoming record is added as duplicate
    - ✓ Incoming duplicate record is rejected during append process
- Destructive Merge
  - Incoming data is applied to the target data
  - If primary key of an incoming record matches with the key of an existing record, matching target record is updated
  - If incoming record is a new record, it is added to the target table

Modes of Applying Data

- Constructive Merge
  - If primary key of an incoming record matches with the key of an existing record, existing record is kept as it is and new record is added as superseding the old record

Modes of Applying Data

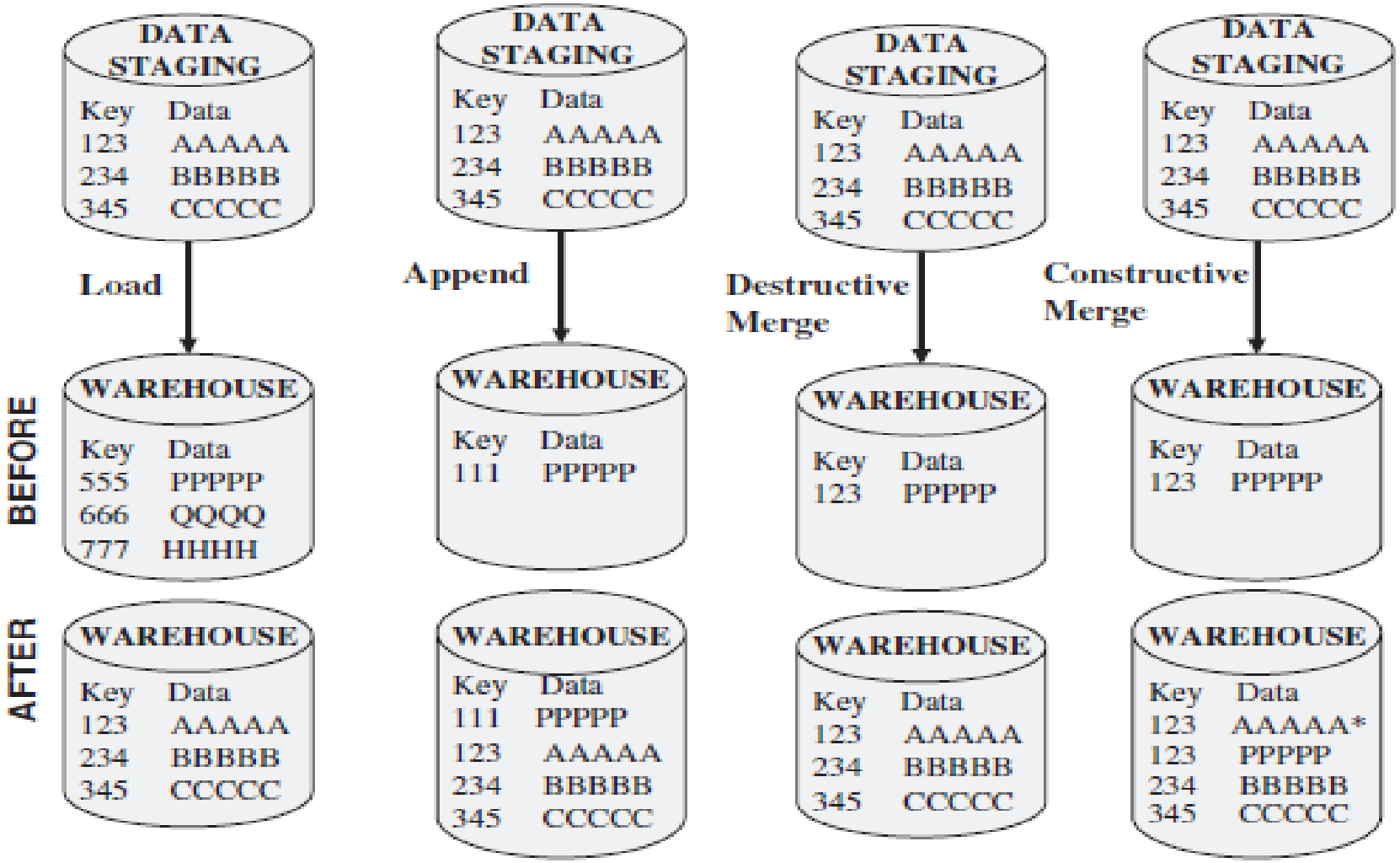

Figure 12-11 Modes of applying data.

Modes of Applying Data to Types of Loads

- Initial Load
  - In case of single load OR multiple sub-loads where every load run is creating database tables from scratch : Load mode can be used
  - In case, more than one runs are needed for creating single table and run is scheduled to run on several days
    - ✓ For the first run of the initial load uses Load mode
    - ✓ All further loads can apply the data using Append mode
  - Index creation for mass loads is time consuming. Indexes can be rebuilt or regenerated when the loads are complete

Modes of Applying Data to Types of Loads

- Incremental Load
  - For such type of loads, a method is needed to preserve the periodic nature of the changes in the data warehouse
  - **Constructive merge mode** is an appropriate method for incremental loads, as it preserves the periodic nature of the changes
  - **Destructive merge can be used for Type 1 changes** in dimensions
    - ✓ The mode can also be used if historical perspective is not important

Modes of Applying Data to Types of Loads

- Full Refresh
  - Entire data warehouse or partial refresh to specific table is done. However partial refreshes are rare as every dimension table is intricately tied to the fact table
  - Similar to initial load, except data exists in target tables
  - Load mode or Append mode are applicable

Data Refresh vs Update

- After initial load, data warehouse is maintained and kept up-to-date by
  - Update: application of incremental changes in the data sources
  - Refresh: complete reload at specified intervals

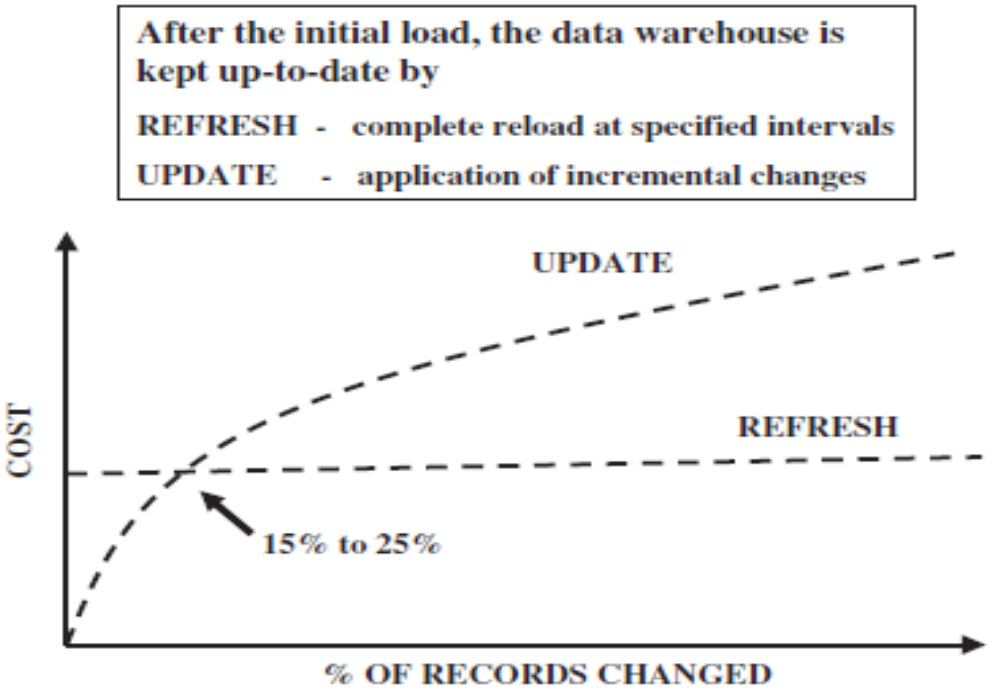

Figure 12-12 Refresh versus update.

Loading Dimension Tables

- The procedure for maintaining the dimension tables includes two functions
  - first, the initial loading of the tables
  - thereafter, applying the changes on an ongoing basis
- Two issues to be addressed in loading dimension tables
  - first one is about the keys of the records in the source systems and the keys of the records in the data warehouse
    - Before source data can be applied to the dimension tables, whether for the initial load or for ongoing changes, the production keys must be converted to the system-generated keys in the data warehouse.

Loading Dimension Tables

- Two issues to be addressed in loading dimension tables
  - first one is about the keys of the records in the source systems and the keys of the records in the data warehouse
    - Key conversion can be a part of transformation function or separate key translation can be used
    - Later is preferable
  - Another issue relates to the application of the type 1, type 2, and type 3 dimension changes to the data warehouse

Loading Dimension Tables

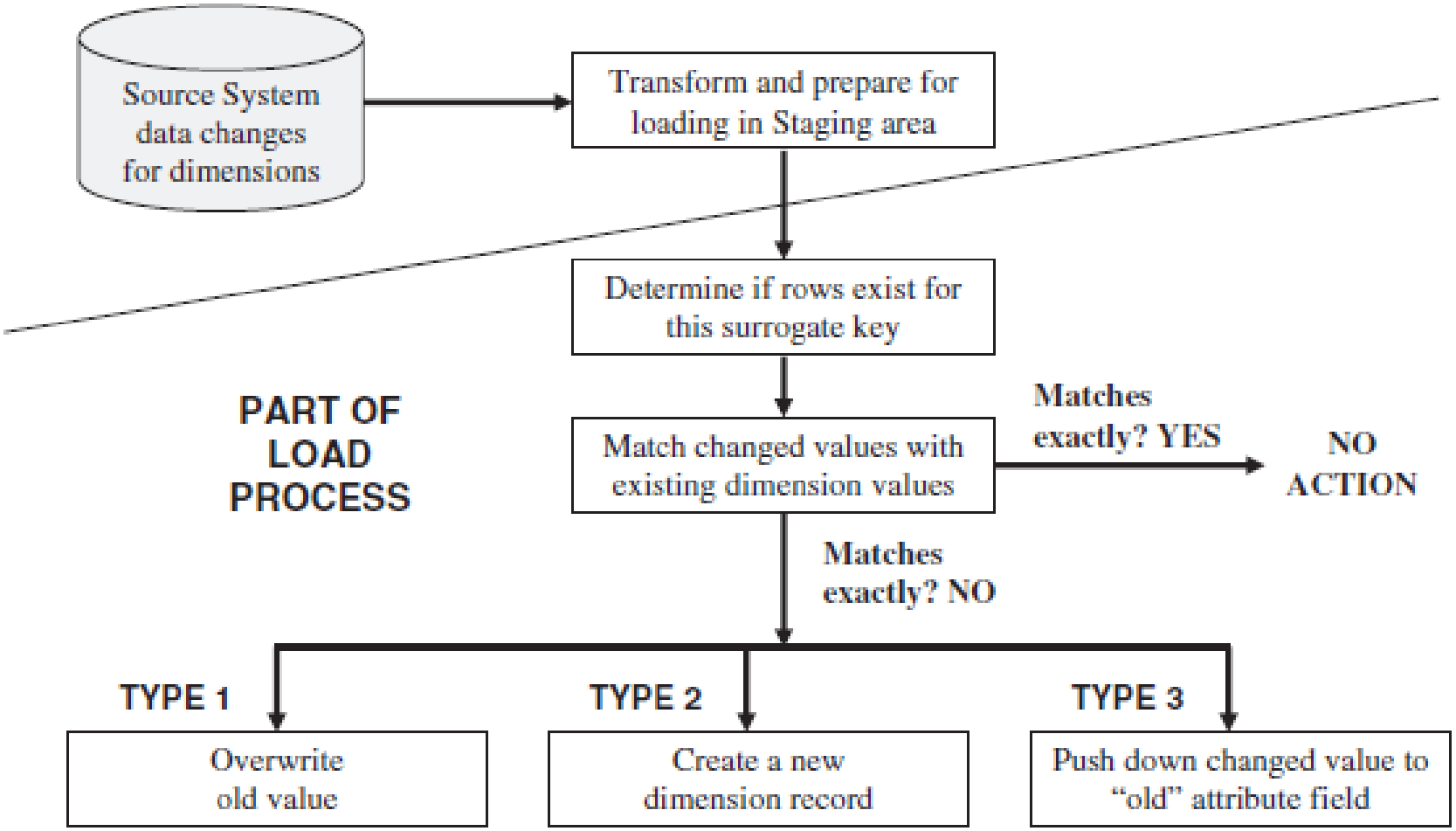

Figure 12-13 Loading changes to dimension tables.

Loading Fact Tables

- The key of the fact table is the concatenation of the keys of the dimension tables
  - So, dimension records are loaded first and then before loading fact table, concatenated key needs to be created from the keys of corresponding dimension tables
  - thereafter, applying the changes on an ongoing basis
- Fact table loads can be of two types
  - History Loads
  - Incremental Loads

Fact Tables History Loads

- Tips for history loads of the fact tables
  - Identify historical data useful and interesting for the data warehouse
  - Define and refine extract business rules
  - Capture audit statistics to tie back to the operational systems
  - Perform fact table surrogate key look-up
  - Improve fact table content
  - Restructure the data
  - Prepare the load files

Fact Tables Incremental Loads

Useful remarks about incremental loads for Fact Tables

- Incremental extracts for fact tables
  - Consist of new transactions
  - Consist of update transactions
  - Use database transaction logs for data capture
- Incremental loads for fact tables
  - Load as frequently as feasible
  - Use partitioned files and indexes
  - Apply parallel processing techniques

References

- *Paulraj Ponniah, "Data Warehousing Fundamentals For It Professionals", Second Edition, Wiley Publication, 2010*