DATA EXTRACTION TRANSFORMATION AND LOADING

Chapter #12 Part I

IBA-FCS

Imran Khan

Introduction

- As information technology professionals, we are fully aware of the **Futile Attempts In The Past Decades To Provide Strategic Information From Operational Systems.**
- These attempts did not work.
- Data warehousing had begun to fulfill that pressing need for strategic information.
- Mostly, the information contained in a warehouse flows from the same operational systems that could not be directly used to provide strategic information.
- **What constitutes the difference between THE DATA IN THE SOURCE OPERATIONAL SYSTEMS and THE INFORMATION IN THE DATA WAREHOUSE?**
- **It is THE SET OF FUNCTIONS that fall under the broad group of data extraction, transformation, and loading (ETL).**

Introduction

- ETL functions **reshape the relevant data** from the source systems into useful information to be stored in the data warehouse.
- If the source data is not **extracted correctly, cleansed,** and **integrated** in the proper formats, query processing and delivery of business intelligence, the backbone of the data warehouse, could not happen.
- ETL functions form the prerequisites for the data warehouse information content.

ETL Overview

- The DW environment is divided into three functional areas:
  - Data acquisition
  - Data storage
  - Information delivery.
- Data extraction, of course, precedes all other functions. But what is the scope and extent of the data you will extract from the source systems?
- Do you not think that the users of your data warehouse are interested in all of the operational data for some type of query or analysis?
- Data extraction presupposes a selection process.
- Select the needed data based on the user requirements.

Most Important and Most Challenging

- Take as an example an analysis your user wants to perform.
  - The user wants to **compare and analyze sales by store, by product, and by month.**
  - The sales figures are available in the several sales applications in your company.
  - Also, you have a product master file.
  - Further, each sales transaction refers to a specific store.
  - All these are pieces of data in the source operational systems.
  - For doing the analysis, you have to provide information about the sales in the data warehouse database.
  - You have to provide the sales units and dollars in a fact table, the products in a product dimension table, the stores in a store dimension table, and months in a time dimension table.

Most Important and Most Challenging

- Howdo you do this?
- **Extract** the data from each of the operational systems, **reconcile** the variations in data representations among the source systems, and **transform** all the sales of all the products. Then **load** the sales into the fact and dimension tables.
- Now, after completion of these three functions, the extracted data is sitting in the data warehouse, transformed into strategic information, ready for delivery as business intelligence for analysis.
- Notice that it is important for each function to be performed, and performed in sequence.
- ETL functions are challenging primarily because of the nature of the source systems.
- Most of **the challenges in ETL arise from the disparities among the source operational systems.**

Most Important and Most Challenging

- Review the following list of reasons for **the types of difficulties in ETL functions.** Consider each carefully and relate it to your environment so that you may find proper resolutions.
  - **Source systems are very diverse and disparate.**
  - There is usually a need to deal **with source systems on multiple platforms and different operating systems.**
  - Many **source systems are older legacy applications running on obsolete database technologies.**
  - Generally, **historical data on changes in values are not preserved in source operational systems.** Historical information is critical in a data warehouse.
  - **Quality of data is dubious in many old source systems** that have evolved over time.

Most Important and Most Challenging

- **Source system structures keep changing over time because of new business conditions.** ETL functions must also be modified accordingly.
- **Gross lack of consistency among source systems** is prevalent. Same data is likely to be represented differently in the various source systems. For example, data on salary may be represented as monthly salary, weekly salary, and bimonthly salary in different source payroll systems.
- Even **when inconsistent data is detected among disparate source systems, lack of a 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.** Many representations are cryptic and ambiguous.

Time Consuming and Arduous

- It is not uncommon for a project team to spend as much as 50% to 70% of the project effort on ETL functions.
- **Data Extraction** itself can be quite involved depending on the nature and complexity of the source systems.
  - The metadata on the source systems must contain information on every database and every data structure that are needed from the source systems.
- **Data Transformation Function** can run the gamut of transformation methods.
  - You have to reformat internal data structures, resequence data, apply various forms of conversion techniques, supply default values wherever values are missing, and you must design the whole set of aggregates that are needed for performance improvement.
  - In many cases, you need to convert from EBCDIC to ASCII formats.

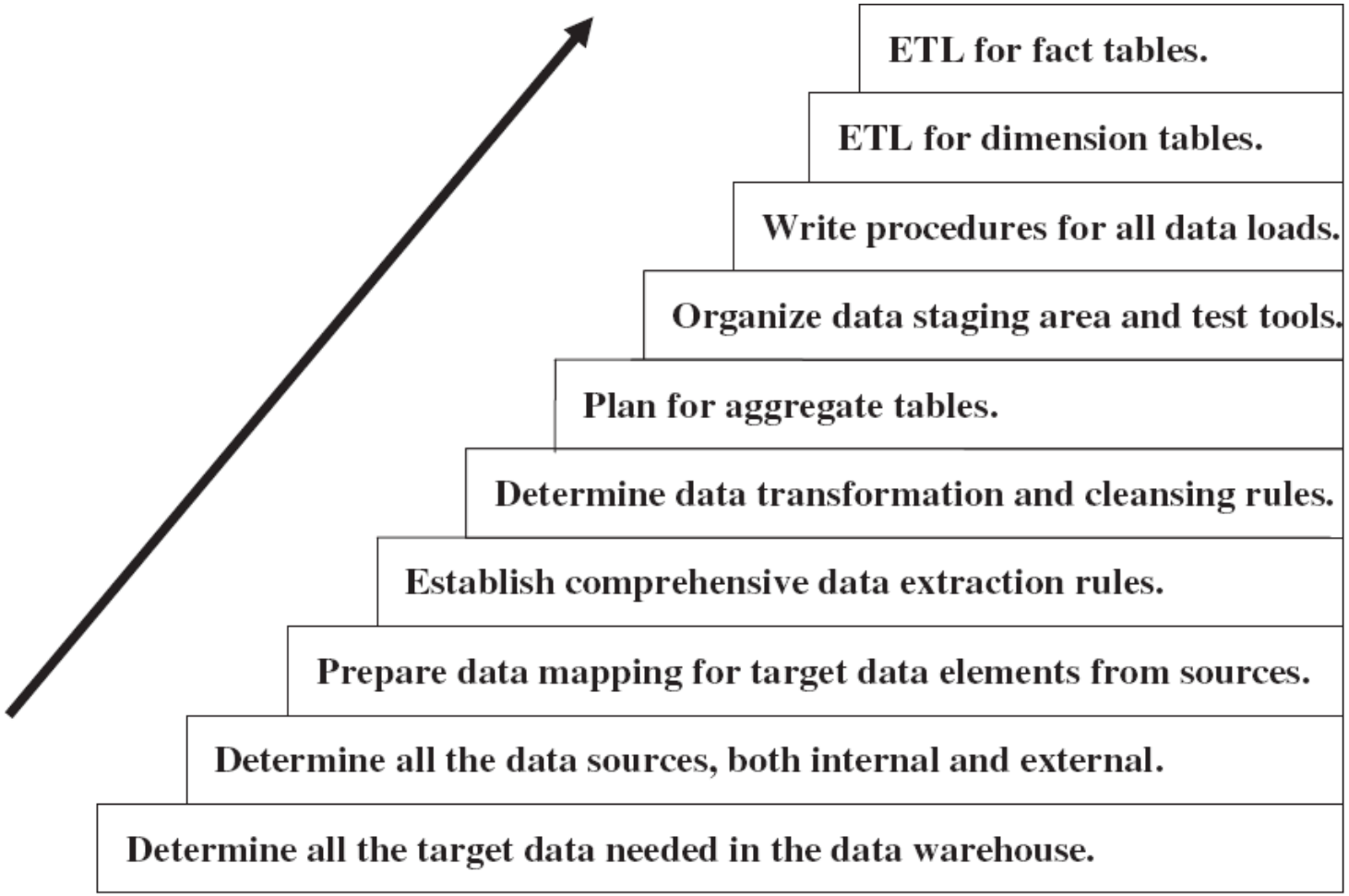
Time Consuming and Arduous

- **Data Loading Function, t**he sheer massive size of the initial loading can populate millions of rows in the data warehouse database.
  - Creating and managing load images for such large numbers are not easy tasks.
  - Even more difficult is the task of testing and applying the load images to actually populate the physical files in the data warehouse.
  - Sometimes, it may take two or more weeks to complete the initial physical loading.

ETL REQUIREMENTS AND STEPS

- let us review the functional steps for initial bulk refresh as well as for the incremental data loads, the sequence is simply as noted here:
  - triggering for incremental changes
  - filtering for refreshes and incremental loads
  - Data extraction
  - transformation
  - integration
  - cleansing
  - applying to the data warehouse database.
- This list is by no means complete for every data warehouse, but it gives a good insight into what is involved to complete the ETL process.

ETL REQUIREMENTS AND STEPS



Major steps in the ETL process .

Key Factors

- The first relates to the complexity of the data extraction and transformation functions.
- The second is about the data loading function.
- You need to pay special attention to the various sources and begin with a complete inventory of the source systems
- The difficulties encountered in the data transformation function also relate to the heterogeneity of the source systems.
- Data Loading Issues:
  - the mass refreshes, whether for initial load or for periodic refreshes, cause difficulties, not so much because of complexities, but because these load
  - jobs run too long.
  - You will have to find the proper time to schedule these full refreshes.

Key Factors

- Incremental loads have some other types of difficulties.
  - First, you have to determine the best method to capture the ongoing changes from each source system.
  - Next, you have to execute the capture without impacting the source systems.
  - After that, at the other end, you have to schedule the incremental loads without impacting use of the data warehouse by the users.

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.
  - **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.

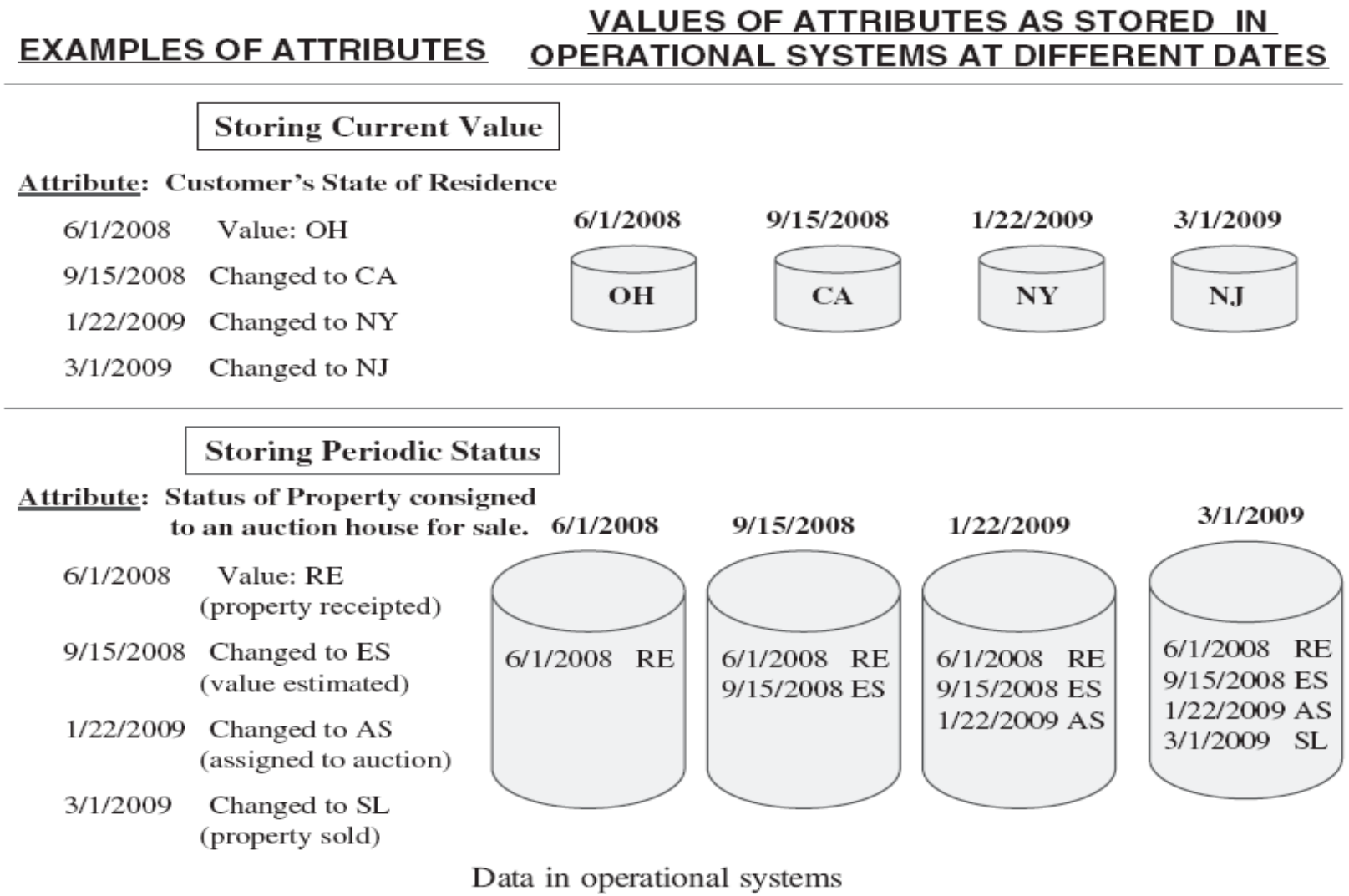
Data in Operational Systems

- Current Value
  - Most of the attributes in the source systems fall into this category.
  - Here the stored value of an attribute represents the value of the attribute at this moment of time.
  - The values are transient or transitory.
  - As business transactions happen, the values change.
  - There is no way to predict how long the present value will stay or when it will get changed next.
  - Customer name and address, bank account balances, and outstanding amounts on individual orders are some examples of this category.

Data in Operational Systems

- Periodic Status
  - This category is not as common as the previous category.
  - In this category, the value of the attribute is preserved as the status every time a change occurs.
  - At each of these points in time, the status value is stored with reference to the time when the new value became effective.
  - This category also includes events stored with reference to the time when each event occurred.
  - Look at the way data about an insurance policy is usually recorded in the operational systems of an insurance company.
  - The operational databases store the status data of the policy at each point of time when something in the policy changes.
  - Similarly, for an insurance claim, each event, such as claim initiation, verification, appraisal, and settlement, is recorded with reference to the points in time.

Data in Operational Systems



Data in operational systems

Data in Operational Systems

- Broadly, there are two major types of data extractions from the source operational systems:
  - "as is" (static) data
  - data of revisions.

“As is” or static data

- “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.
- For current or transient data, this capture would include all transient data identified for extraction.
- In addition, for data categorized as periodic, this data capture would include each status or event at each point in time as available in the source operational systems.

Data of Revisions

- Data of revisions is also known as incremental data capture.
- Strictly, it is not incremental data but the revisions since the last time data was captured.
- If the source data is transient, the capture of the revisions is not easy.
- For periodic status data or periodic event data, the incremental data capture includes the values of attributes at specific times.
- Extract the statuses and events that have been recorded since the last data extraction.
- Incremental data capture may be immediate or deferred.

Immediate Data Extraction

- In this option, the data extraction is real-time.
- It occurs as the transactions happen at the source databases and files.
- three options for immediate data extraction.
  - Capture through Transaction Logs
  - Capture through Database Triggers
  - Capture in Source Applications

Capture through Transaction Logs

- This option uses the transaction logs of the DBMSs maintained for recovery from possible failures.
- As each transaction adds, updates, or deletes a row from a database table, the DBMS immediately writes entries on the log file.
- This data extraction technique reads the transaction log and selects all the committed transactions.
- There is no extra overhead in the operational systems because logging is already part of the transaction processing.

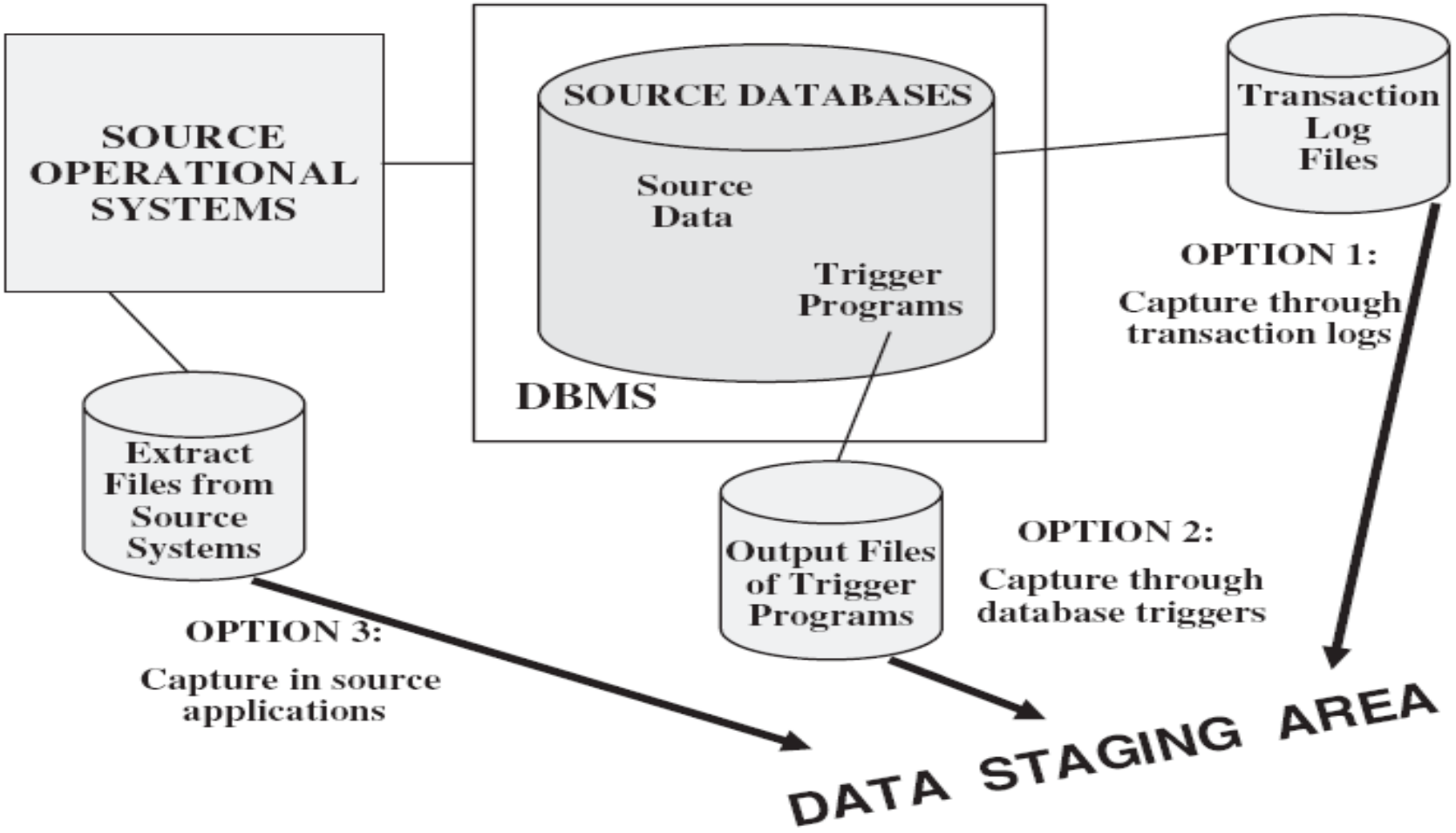
Capture through Database Triggers

- this option is applicable to your source systems that are database applications.
- As you know, triggers are special stored procedures (programs) that are stored on the database and fired when certain predefined events occur.
- You can create trigger programs for all events for which you need data 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.
- For example, if you need to capture all changes to the records in the customer table, write a trigger program to capture all updates and deletes in that table.

Capture in Source Applications

- This technique is also referred to as application assisted data capture.
- In other words, the source application is made to assist in the data capture for the data warehouse.
- You have to modify the relevant application programs that write to the source files and databases.
- You revise the programs to write all adds, updates, and deletes to the source files and database tables.
- Then other extract programs can use the separate file containing the changes to the source data.

Immediate Data Extraction



Options for immediate data extraction.

Deferred Data Extraction

- The techniques under deferred data extraction do not capture the changes in real time.
- The capture happens later.
- Two options for deferred data extraction
  - Capture Based on Date and Time Stamp
  - Capture by Comparing Files

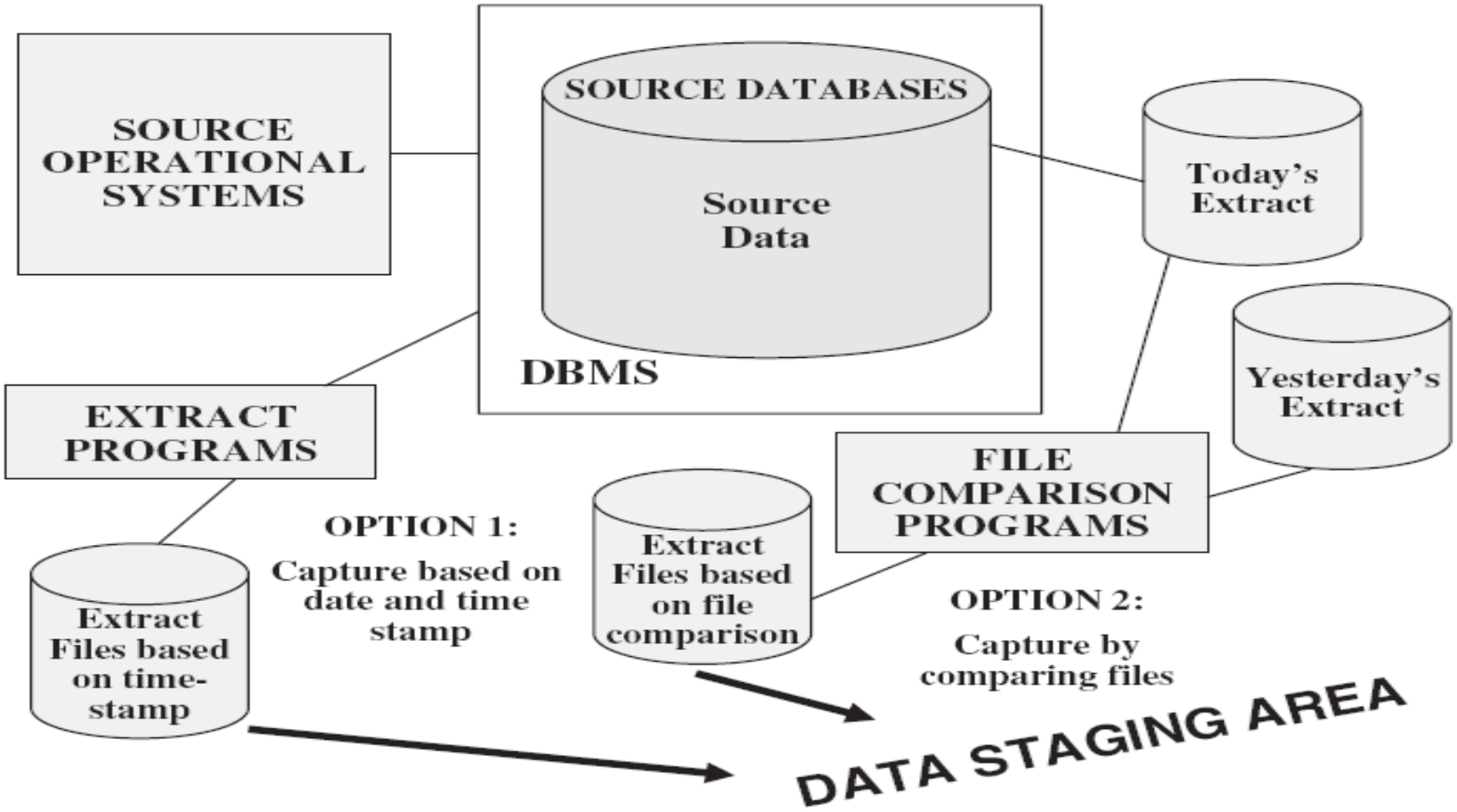
Capture Based on Date and Time Stamp

- 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.
- Here the data capture occurs at a later time, not while each source record is created or updated.
- If you run your data extraction program at midnight every day, each day you will extract only those with the date and time stamp later than midnight of the previous day.
- This technique works well if the number of revised records is small.
- 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.
- This technique necessitates the keeping of prior copies of all the relevant source data.
- Though simple and straightforward, comparison of full rows in a large file can be very inefficient.
- this may be the only feasible option for some legacy data sources that do not have transaction logs or time stamps on source records.

Capture by Comparing Files



Options for deferred data extraction.

Data Capture Techniques: Adv & Disadvantages

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 capture techniques: advantages and disadvantages

Evaluation of the Techniques

- To summarize, the following options are available for data extraction:
  - Capture of static data
  - Capture through transaction logs
  - Capture through database triggers
  - Capture in source applications
  - Capture based on date and time stamp
  - Capture by comparing files