Data Warehouse and Data Mining

ETL-1

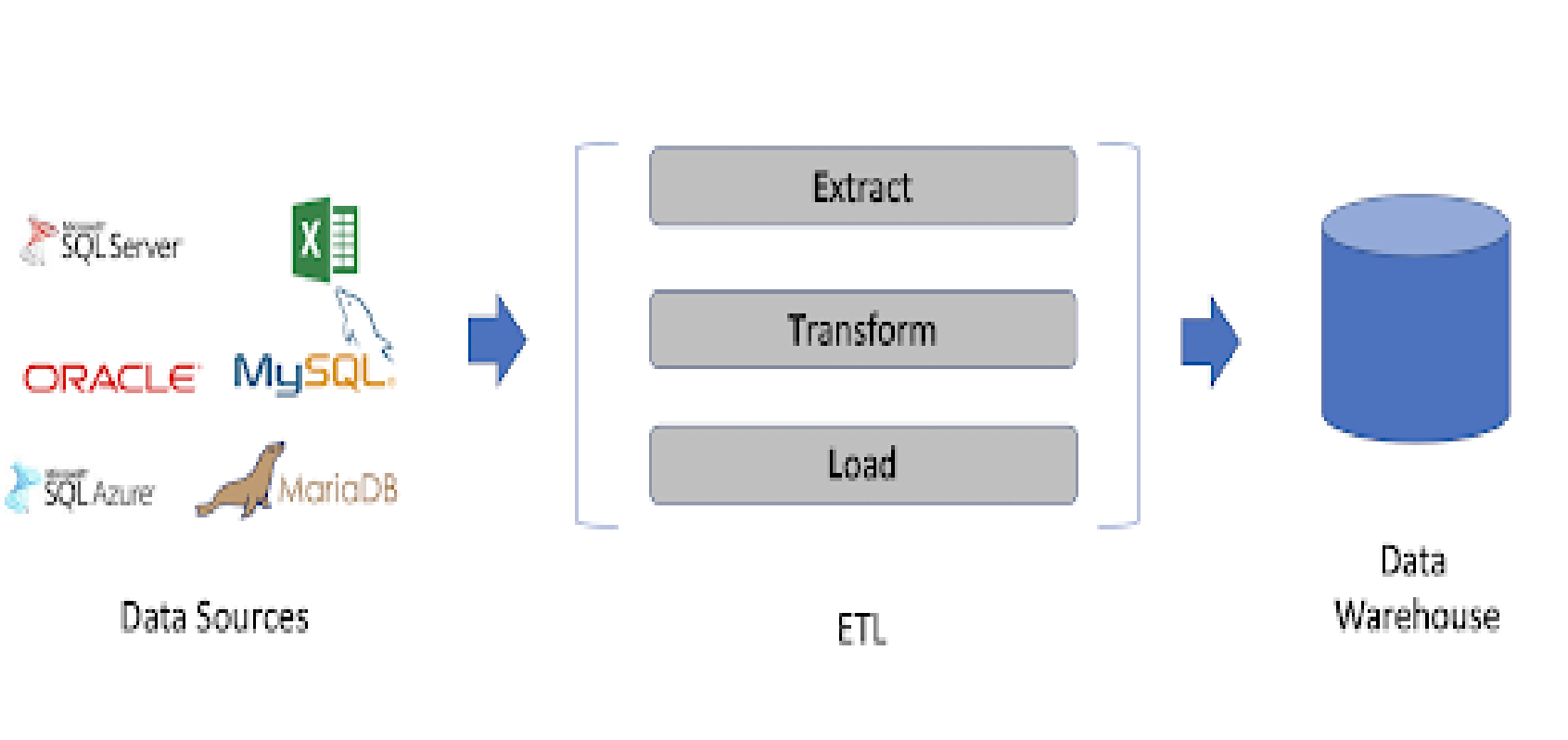
Alaa Khalaf Hamoud

Contents

1. *Introduction*
2. *ETL Requirements*
3. *Data Extraction*
   - *3.1 Source Identification*
   - *3.2 Data Extraction Techniques*
4. *Data Transformation*
   - *4.1 Data Transformation Tasks*
   - *4.2 Data Integration and Consolidation*
   - *4.3 Transformation for Dimension Attributs*
5. *Data Loading*

1. Introduction

- ETL ?



1. Introduction

- To **change** the **data** to **information** you need to **capture** the data.
- You **cannot** simply **dump** that data into the **DW** and call it **strategic information**.
- You have to **subject** the **extracted data** to all manner of **transformations**.
- You must perform all **three functions of ETL** for successfully transforming data into **strategic information** or **business intelligence**.

1. Introduction

- ETL functions are **challenging** primarily because of the **nature of the source systems**.
  - **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**.

1. Introduction

- Generally, **historical data** on **changes in values** **are not preserved** in source operational systems. Historical information is **critical** in a DW.

- **Quality of data** is **dubious** in many old source systems that have **evolved** over time.

- **Source system structures** keep **changing** over time because of **new business conditions** .ETL functions must also be modified accordingly.

1. Introduction

- **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.

- **Most** source systems **do not represent** data in **types** or **formats** that are **meaningful** to the users. Many representations are cryptic and ambiguous.

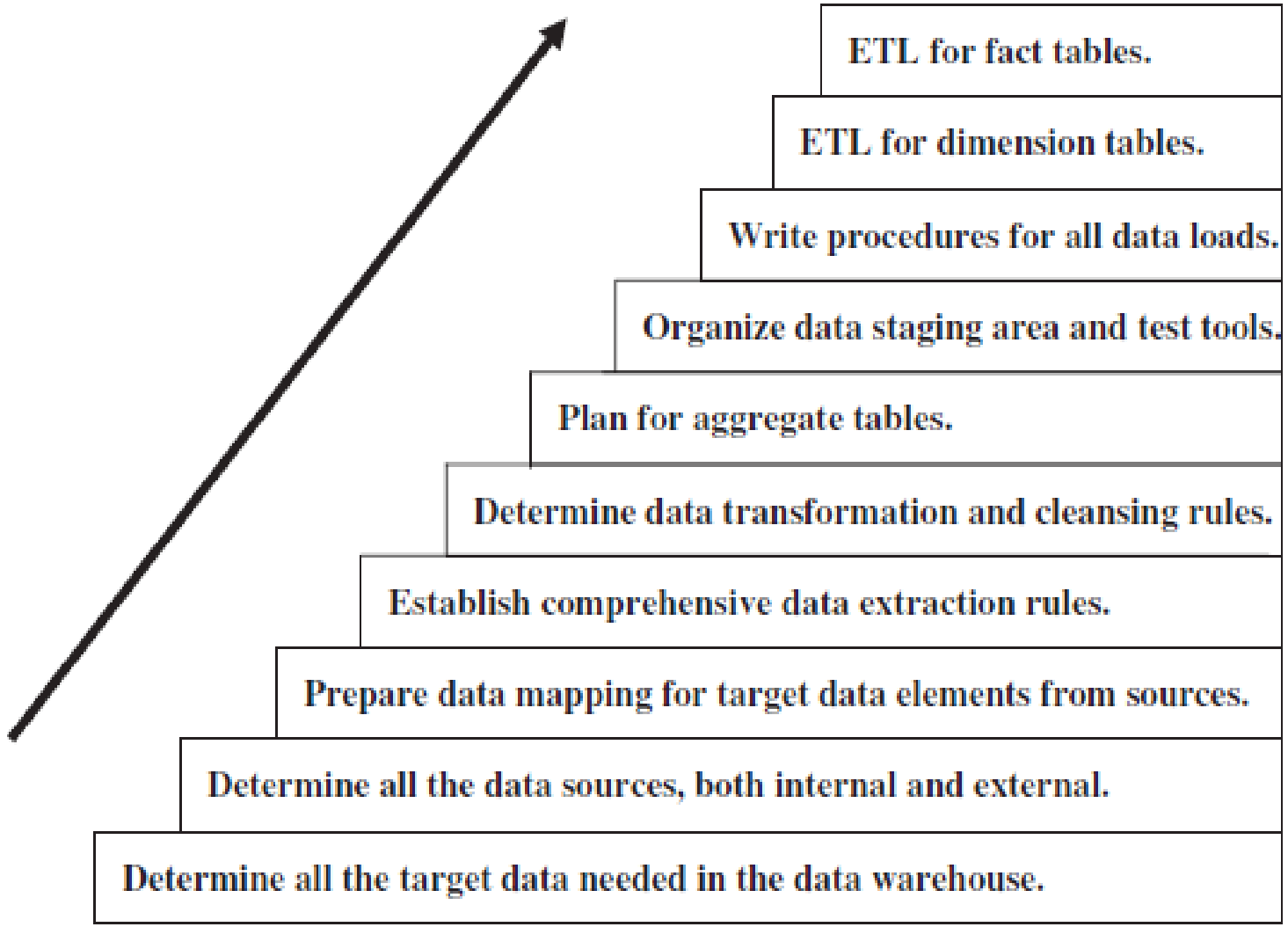
2. ETL Requirements

- The primary reason for the complexity of extraction and transformation function is the **tremendous diversity** of the **source systems**.
- For **initial bulk refresh** as well as for the **incremental data loads**, the sequence is:
  - Triggering for incremental changes
  - Filtering for refreshes and incremental loads
  - Data extraction
  - Data transformation
  - Integration, cleansing, and applying to the DW DBMS.

2. ETL Requirements

- In a **large enterprise**, we could have a bewildering combination of **computing platforms, operating systems, database management systems, network protocols, and source legacy systems**.
- Usually, the 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**.

2. ETL Requirements



3. **Data Extraction**

- Some data may be on other **legacy network** and **hierarchical data models**.
- **Many data sources** may still be in **flat files**.
- You may want to **include data from spreadsheets** and **local departmental data sets**.
- You may want to consider using **outside tools (market) suitable** for certain data sources.

3. Data Extraction

- For the other data sources, you may want to **develop in-house programs** to do the data extraction.

- Purchasing **outside tools** may **entail high initial costs**.

- **In-house programs**, on the other hand, may **mean ongoing costs** for **development** and **maintenance**.

3. Data Extraction

- Two major factors **differentiate** the data extraction in a **new operational system and DW.**
  - For a DW, you have to **extract data from many disparate sources**.
  - Next, you have to **extract data on the changes** for **ongoing incremental loads** as well as for a one-time initial full load.
  - For operational systems, all you need **is one-time extractions and data conversions**.
- **Effective data extraction** is a key to the success of your DW.
  - Pay special attention to the issues and formulate a **data extraction strategy** for your DW.

3. Data Extraction

- Here is a 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 be done: daily, weekly, quarterly, and so on.

3. Data Extraction

- **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.

3.1 Source Identification

- Encompasses the **identification** of all **the proper data sources**.
- It includes **examination** and **verification** that the identified sources will provide the **necessary value** to the DW.
  - Determine if the source systems have data needed for this **data mart**.
  - Then, you have to establish the **correct data source** for each **data element in the data mart**.

3.2 Data Extraction Techniques

- You should **understand** the **nature of source data** before examine extraction techniques.
- **Business transactions** keep changing the data in the source systems.
- In most cases, the **value of an attribute** in a source system is the value of that **attribute at that time**.
- If you look at every data structure in the source operational systems, the **day-to-day business** transactions **constantly change** the values of the attributes in these structures.

3.2 Data Extraction Techniques

- When a **customer moves** to **another state**, the data about that **customer changes** in the customer table in the source system.

- Data in the source systems are said to be **time-dependent** or **temporal**.

- This is because **source data changes with time**. The value of a single variable varies over time.

3.2 Data Extraction Techniques

- Operational data may falling into **two broad categories depends on the nature).**
  - **Current Value**: here the stored value of an attribute **represents** the value of the attribute at **this moment of time**.
  - **Periodic Status**: 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.

3.2 Data Extraction Techniques

EXAMPLES OF ATTRIBUTES

VALUES OF ATTRIBUTES AS STORED IN OPERATIONAL SYSTEMS AT DIFFERENT DATES

Storing Current Value

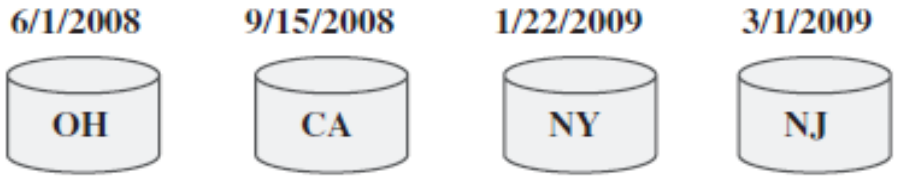
Attribute: Customer's State of Residence

6/1/2008 Value: OH

9/15/2008 Changed to CA

1/22/2009 Changed to NY

3/1/2009 Changed to NJ



Storing Periodic Status

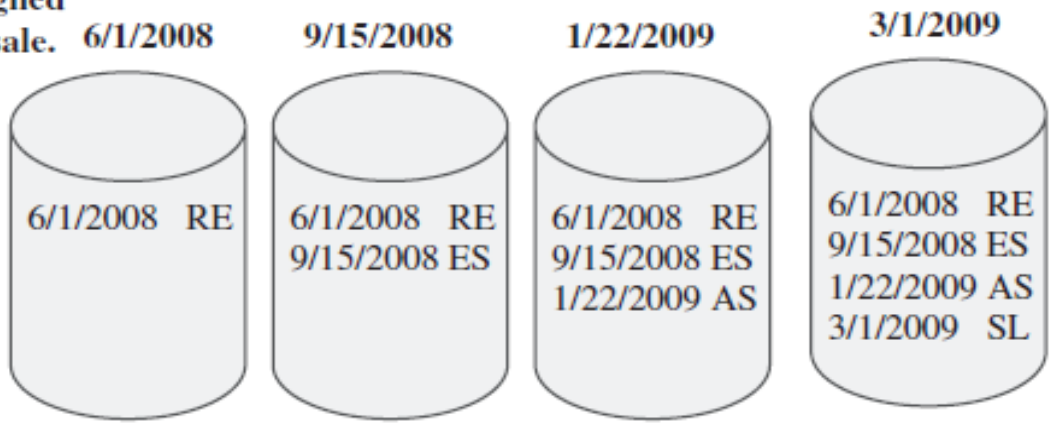
Attribute: Status of Property consigned to an auction house for sale.

6/1/2008 Value: RE (property receipted)

9/15/2008 Changed to ES (value estimated)

1/22/2009 Changed to AS (assigned to auction)

3/1/2009 Changed to SL (property sold)



3.2 Data Extraction Techniques

- Static data will be used for **the initial load** of the DW.
- Sometimes, you may want a **full refresh** of a dimension table.
  - For example, assume that the product master of your source application is completely revamped. In this case, you may find it easier to do a full refresh of the product dimension table of the target DW.
- **Data** of **revisions** is also known as **incremental data capture**.

3.2 Data Extraction Techniques

- 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.

3.2 Data Extraction Techniques

- **Immediate Data Extraction**
  - In this option, the data extraction is **real-time**.
  - It occurs as the **transactions happen** at the source databases and files.

3.2 Data Extraction Techniques

- **Capture through Transaction Logs**
  - Use the **transaction logs** of DBMSs maintained for **recovery** from possible failures.
  - Transaction logs for (**adds, updates, or deletes a row from a database table**).
  - 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**.

3.2 Data Extraction Techniques

- **Capture through Database Triggers**
  - Applicable to **database applications**.
  - 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 (**to capture all changes to the records in the customer table)**.
  - The output of the trigger programs is written to a **separate file** that will be used to extract data for the DW.

3.2 Data Extraction Techniques

- **Capture in Source Applications**
  - Referred to as **application assisted data capture**.
  - The source application is **made to assist in the data capture for the DW**. 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.

3.2 Data Extraction Techniques

- **Deferred Data Extraction**
  - In the cases discussed before, data capture takes place while the transactions occur in the source operational systems. The data capture is immediate or real-time.
  - The techniques under deferred data extraction **do not capture the changes in real time**. The capture happens later.

3.2 Data Extraction Techniques

- **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**.
  - The relevant source records Should contain **date and time stamps.**
  - Here the **data capture occurs** at a **later time**, not while each source record is created or updated.

3.2 Data Extraction Techniques

- **Capture Based on Date and Time Stamp**
  - This technique works well if the **number** of **revised records is small**.
  - This technique can work for **any type** of source file.
  - This technique captures the **latest state** of the source data.
  - Any intermediary states between **two data extraction** runs **are lost**.

3.2 Data Extraction Techniques

- **Capture by Comparing Files**
  - This technique is also called the **snapshot differential technique** because it **compares two snapshots** of the source data.
  - To apply this technique, while performing today's data extraction for changes to product data, you do a **full file comparison** between today's copy of the product data and yesterday's copy.
  - You also **compare the record keys** to find the inserts and deletes. Then you capture any changes between the two copies.

3.2 Data Extraction Techniques

- **Capture by Comparing Files**
  - This technique **necessitates** the keeping of **prior copies** of all the **relevant source data**.
  - **Not effective for large files**.
  - Considered as the only feasible option for **some legacy data sources** with no transaction logs or time stamps on source records.

4. Data Transformation

- **Data** in **the operational systems is not usable for DW purpose (quality, inconsistent** ...)

- First, all the extracted data must be made **usable** in the DW.

- Having information that is **usable** for **strategic decision making** is the underlying principle of the DW.

- You have to **enrich** and **improve** the quality of the data before it can be usable in the data warehouse.

4. Data Transformation

- **Various kinds of data transformations** should be applied into extracted data.
- Transformation should apply according to **standards** of source systems.
  - A wide variety of manipulations to change all the extracted source data into usable information
- After the data put together, the combined data **should not violate business rules**.
  - Data formats, data values, and the condition of the data quality
- Due to transformation complexity, many organizations start out with a **simple departmental data mart** as the pilot project.

4. Data Transformation

- One practitioner may refer to **data integration** as the process within the data transformation function that is some kind of preprocessing of the source data.

- Data integration may mean the **mapping** of the source fields to the **target fields** in the DW.

- One major effort within data transformation is the **improvement of data quality.**
  - Filling in the missing values for attributes in the extracted data.

4. Data Transformation

- **Data quality** is of paramount importance in the DW because the **effect of strategic decisions** based on **incorrect information** can be **devastating**

- First, you **clean the data extracted** from each source.

4. Data Transformation

- **Cleaning** may just be
  - **Correction** of misspellings
  - **Resolution** of **conflicts** between state codes and zip codes in the source data
  - **Deal with providing default values** for missing data elements
  - **Elimination** of duplicates when you bring in the same data from multiple source systems.

4. Data Transformation

- **Standardization** of data elements forms a large part of data transformation.

- You **standardize** the data types and field lengths for same data elements retrieved from the various sources.

- **Semantic standardization** is another major task. You resolve **synonyms** and **homonyms**.

4. Data Transformation

- When two or more terms from different source systems mean the same thing, you resolve the **synonyms**.
- When a single term means many different things in different source systems, you resolve the **homonym**.
- Data transformation involves **combining** processes; you combine data from single source record or related data elements from many source records.
- **Sorting** and **merging** of data takes place on a large scale in the data staging area.

4. Data Transformation

- In many cases, the **keys chosen** for the ODs are **field values with built-in meanings**.
- For example, the **product key value** may be a **combination of characters** indicating the **product category, the code of the warehouse where the product is stored, and some code to show the production batch**.
- **Primary keys** in the DW **cannot** have built-in meanings.

End of ETL-1