Data Warehousing

Data Preprocessing

- The real-world data and Descriptive Data Summarization
  - Noisy, Incomplete, Inconsistent and Redundant
  - It is important to get overall picture of data
- Data Cleaning
- Data Integration
- Data Transformation
- Data Reduction

Why ?

- Noisy data comes from the process of data
  - Collection (faulty of collection system)
  - Entry (human mistake)
  - Transmission (communication noise)
- Incomplete data comes from
  - data value is unavailable when collected
  - human/hardware/software problems (errors)
- Inconsistent data comes from
  - Different data sources (different schemas and hierarchies)
  - different consideration between the time when the data was collected and when it is analyzed.

Data Preprocessing is Important

- No quality data, no quality mining results!
- *From the view of building Data Warehouse*

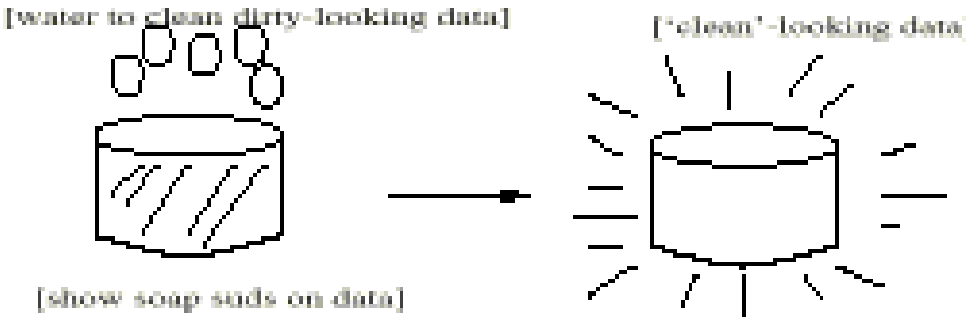
.

Major Tasks in Data Preprocessing

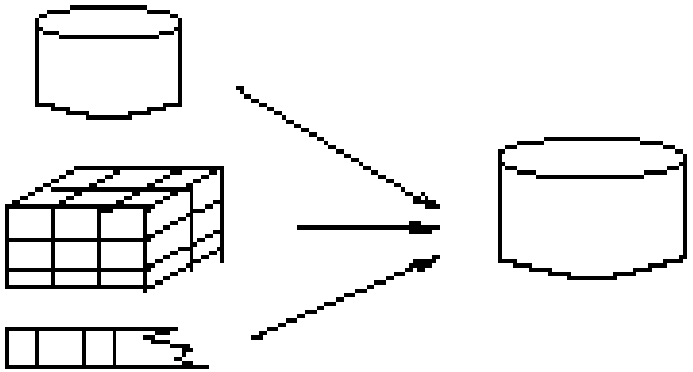
- Data cleaning
  - Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies
- Data integration
  - Integration of multiple databases, data cubes, files, or notes
- Data transformation
  - Normalization (scaling to a specific range)
  - Aggregation
- Data reduction
  - Obtains reduced representation in volume but produces the same or similar analytical results
  - Data discretization: with particular importance, especially for numerical data
  - Data aggregation, dimensionality reduction, data compression,generalization

Forms of data preprocessing

Data Cleaning



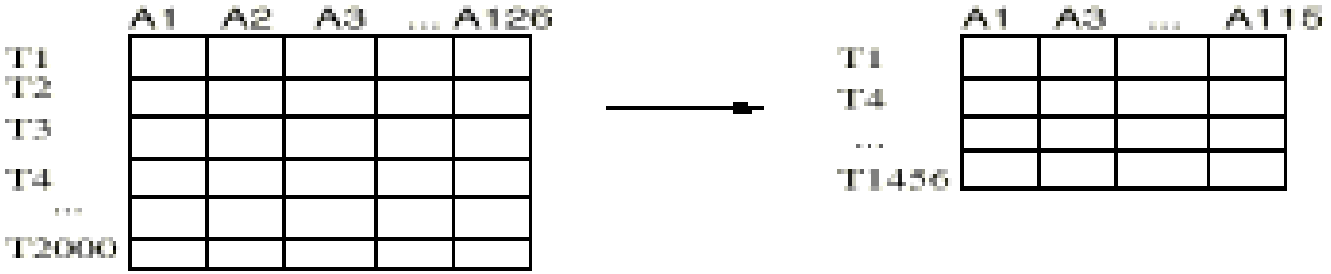
Data Integration



Data Transformation

-2, 32, 100, 59, 48 → -0.02, 0.32, 1.00, 0.59, 0.48

Data Reduction



Data Cleaning

- Data cleaning tasks
  - Fill in missing values
  - Identify outliers and smooth out noisy data
  - Correct inconsistent data

Missing Values

- Data is not always available
  - E.g., many tuples have no recorded value for several attributes, such as customer income in sales data
- Missing data may be due to
  - equipment malfunction
  - inconsistent with other recorded data and thus deleted
  - data not entered due to misunderstanding
  - certain data may not be considered important at the time of entry

–not register history or changes of the data In

databases, there exists Null values in tuples.

Noisy Data

- Noise: random error or variance in a measured variable

- Incorrect attribute values may due to
  - faulty data collection instruments
  - data entry problems
  - data transmission problems
  - technology limitation
  - inconsistency in naming convention

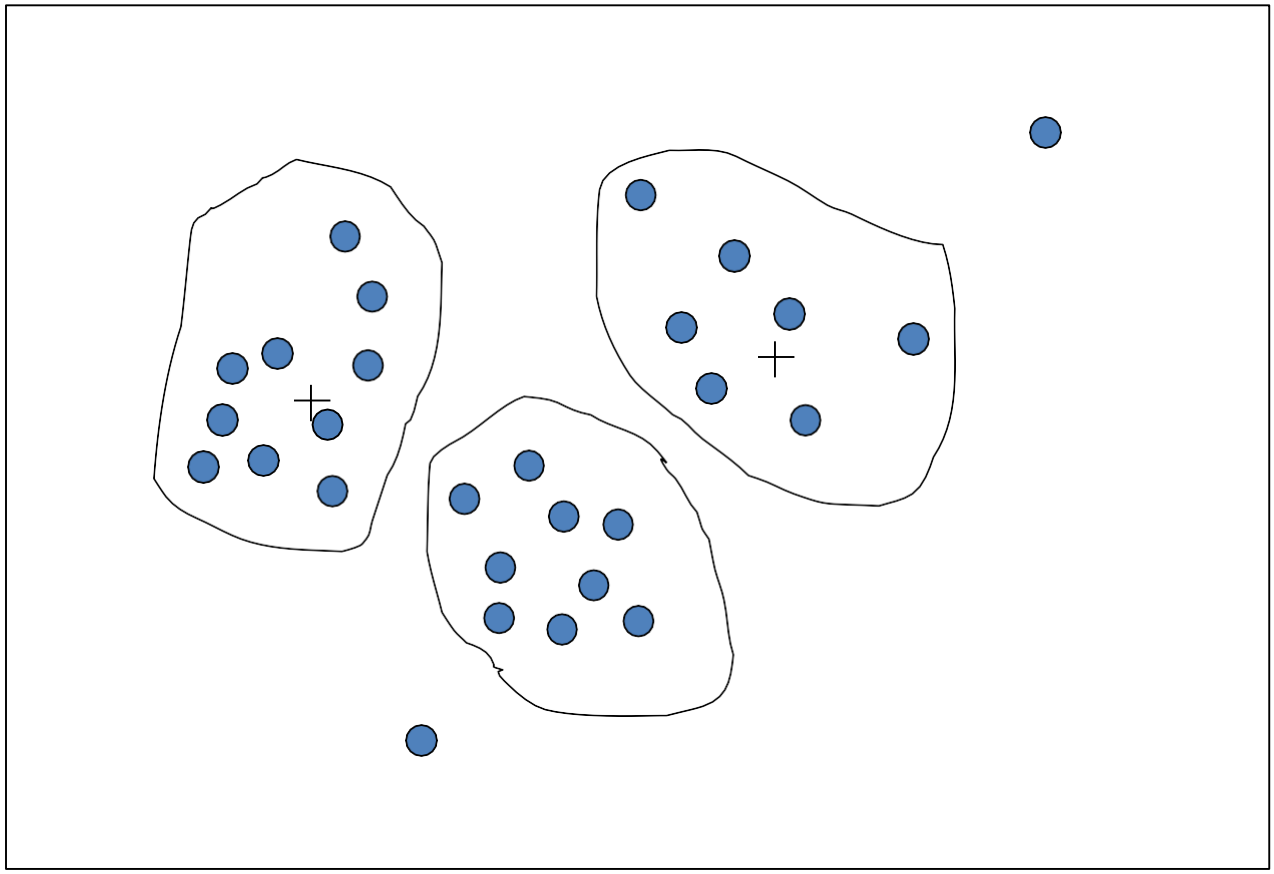
- Other data problems which requires data cleaning
  - duplicate records
  - incomplete data
  - inconsistent data

How to Handle Noisy Data?

- Binning method by consulting its "neighborhood":
  - first sort data and partition into (equi-depth) bins
  - then one can smooth by bin means, smooth by bin median, smooth by bin boundaries, etc.
- Clustering
  - detect and remove outliers
- Regression
  - smooth by fitting the data into regression functions

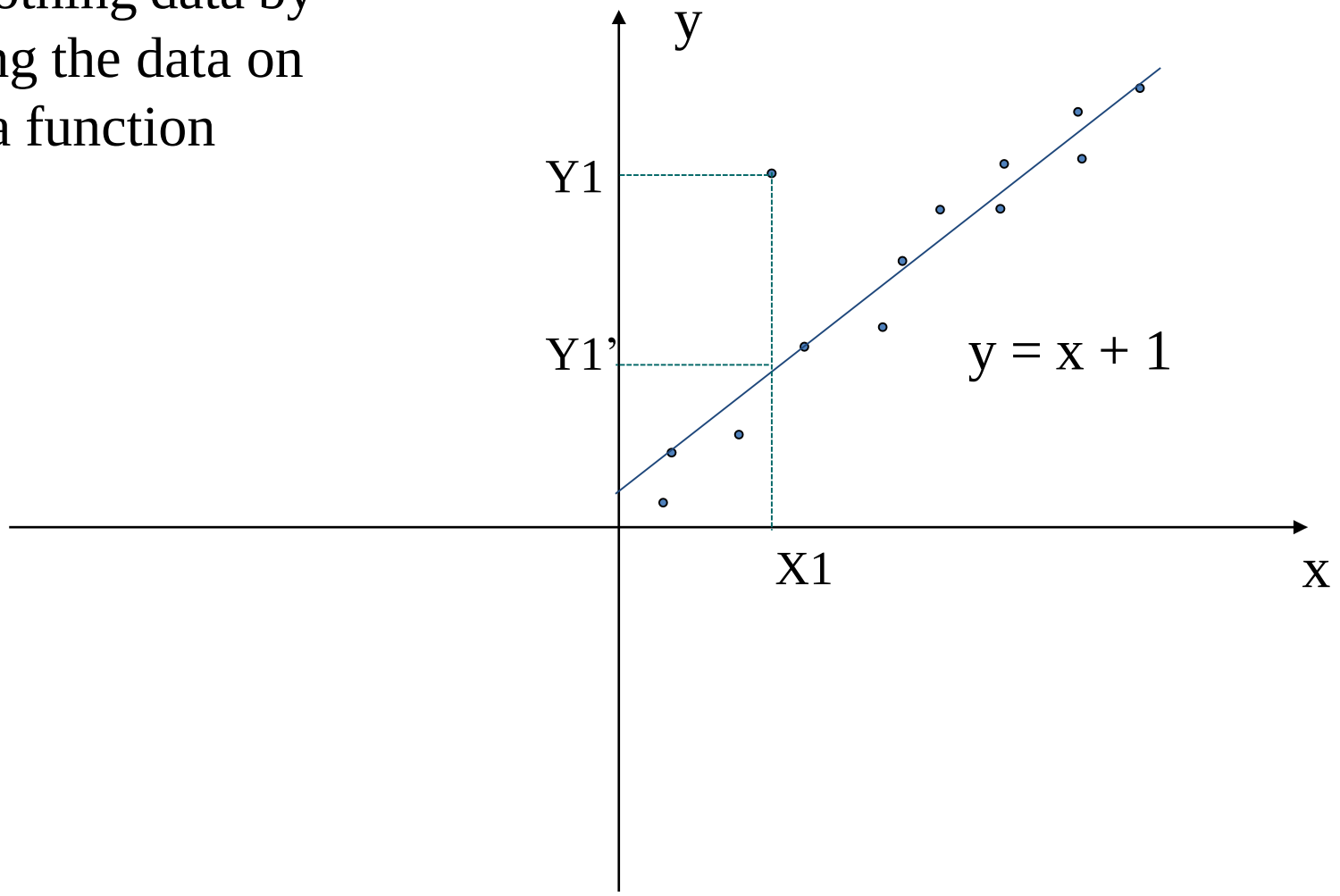
Cluster Analysis

Similar values are organized into groups, or “clusters”. Values that fall outside of clusters may be consider outliers (noise)



Regression

Smoothing data by fitting the data on a function



Data Integration

- Data integration:
  - combines data from multiple sources into a coherent data source or data warehouse
- *Schema integration*
  - *integrate metadata (schemas) from different sources*
  - *can not be automatic*
  - *Entity identification problem: identify real world entities from multiple data sources, e.g., ? Customer("J. Doe") ≡ Customer("John Doe")*
- Detecting and resolving data inconsistency
  - for the same real world entity, attribute values from different sources are different or contradiction (age vs. birthdate)
  - possible reasons: different representations, different scales, e.g., US dollar vs. Canadian dollar

Redundancy in Data Integration

- Redundant data occur often when integration of multiple databases
  - The same attribute may have different names but same values in different databases, e.g. name vs. fullname
  - One attribute may be a “derived” attribute in another table, e.g., annual revenue <= quarter revenues
- Redundant data may be able to be detected by correlation analysis $r_{A,B} = \Sigma(A - \bar{A})(B - \bar{B})/(n-1)\sigma_A \sigma_B$
- Careful integration of the data from multiple sources may help reduce/avoid redundancies and inconsistencies

Data Transformation

- Aggregation and Generalization: summarization, concept hierarchy climbing
- Normalization: scaled to fall within a small, specified range
  - min-max normalization
  - z-score normalization
  - normalization by decimal scaling
- Attribute construction
  - New attributes constructed from the given ones
    - E.g. construct *Area* from *Height* and *Width, Age* from *birthdate*

Normalization

Min-max

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

Z-score

$$v' = \frac{v - \bar{A}}{\sigma_A}$$

Decimal scaling

$$v' = \frac{v}{10^j}$$ Where j is the smallest integer such that Max($|v'|$)<1

Data Reduction

- Problem:

  Data Warehouse may store terabytes of data: Complex data analysis/mining may take a very long time to run on the complete data set

- Solution?
  - Data reduction…

Data Reduction

- Obtains a reduced representation of the data set that is much smaller in volume but yet produces the same (or almost the same) analytical results
- Data reduction strategies
  - Data cube aggregation
  - Dimensionality reduction
  - Data compression
  - Numerosity reduction

Data Cube Aggregation

- Multiple levels of aggregation in data cubes
  - Further reduce the size of data to deal with
- Reference appropriate levels
  - Use the smallest representation capable to solve the task
- Queries regarding aggregated information should be answered using data cube, when possible

Dimensionality Reduction

- Feature selection (i.e., attribute subset selection):
  - Select a minimum set of features such that the probability distribution of different classes given the values for those features is as close as possible to the original distribution given the values of all features
  - reduce # of patterns, easier to understand
- Heuristic methods (due to exponential # of choices):
  - step-wise forward selection
  - step-wise backward elimination
  - combining forward selection and backward elimination
  - decision-tree induction

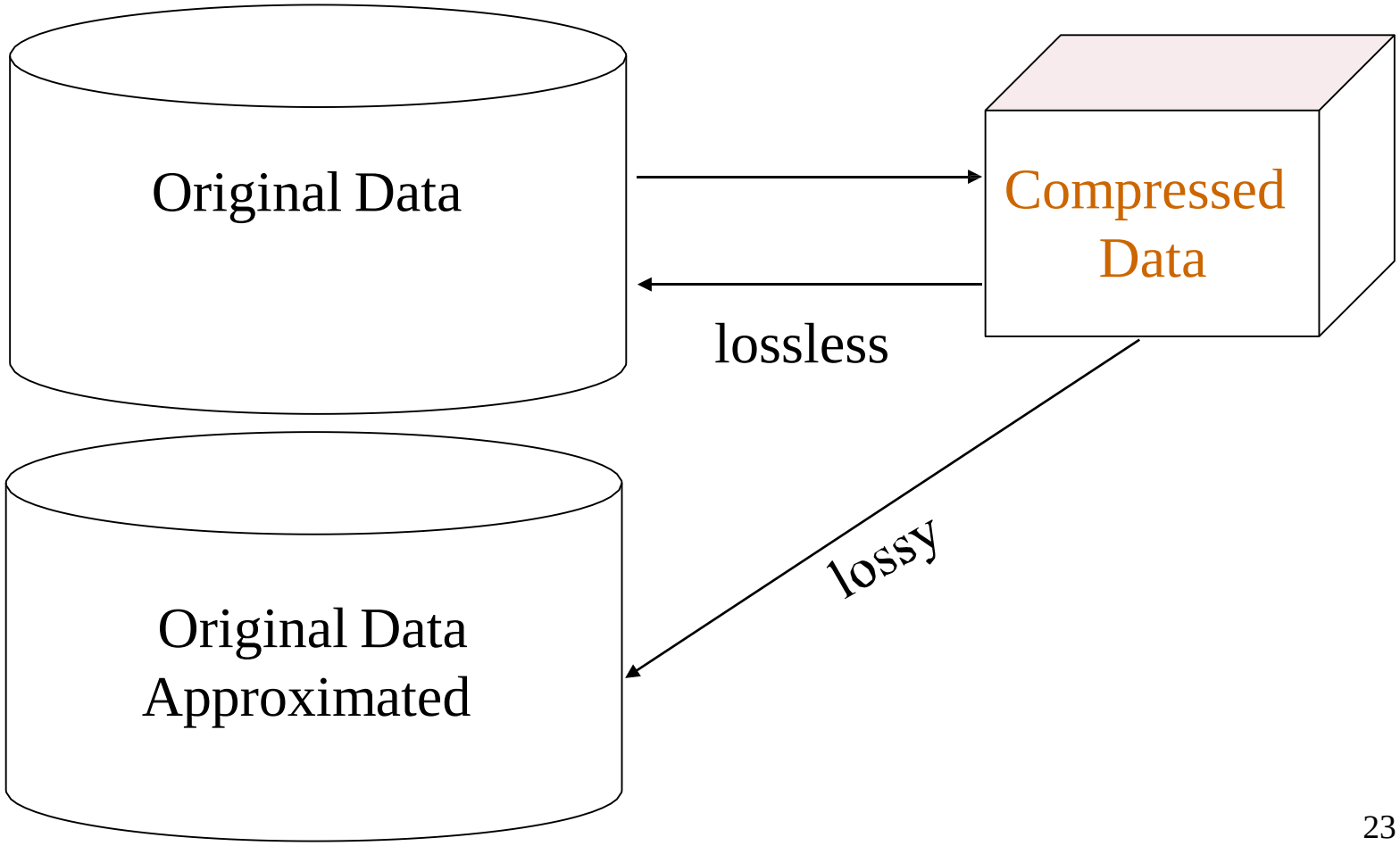
Heuristic Feature Selection Methods

- There are 2^d possible sub-features of d features
- Several heuristic feature selection methods:
  - Stepwise forward selection:
    - Starts with an empty set of attributes
    - The best single-feature is picked first
    - Then next best feature condition to the first, ...
  - Stepwise backward elimination:
    - Starts with full set of attributes
    - Repeatedly eliminate the worst feature
  - Best combined feature selection and elimination:
  - Decision tree induction

Data Compression

- Data encoding or transformation are applied to obtain a reduced or “compressed” representation of the original data.
- String compression
  - There are extensive theories and well-tuned algorithms
  - Typically lossless (*can be restructured without loss*)
- *Audio/video compression*
  - *Typically lossy compression, with progressive refinement*
  - *Sometimes small fragments of signal can be reconstructed w*

Data Compression



Numerosity Reduction

- Parametric methods
  - Assume the data fits some model, estimate model parameters, store only the parameters, and discard the data (except possible outliers)
    - E.g. Log-linear models estimate discrete multidimensional probability distributions.
- Non-parametric methods
  - Do not assume models
  - Major families: histograms, clustering, sampling

Summary

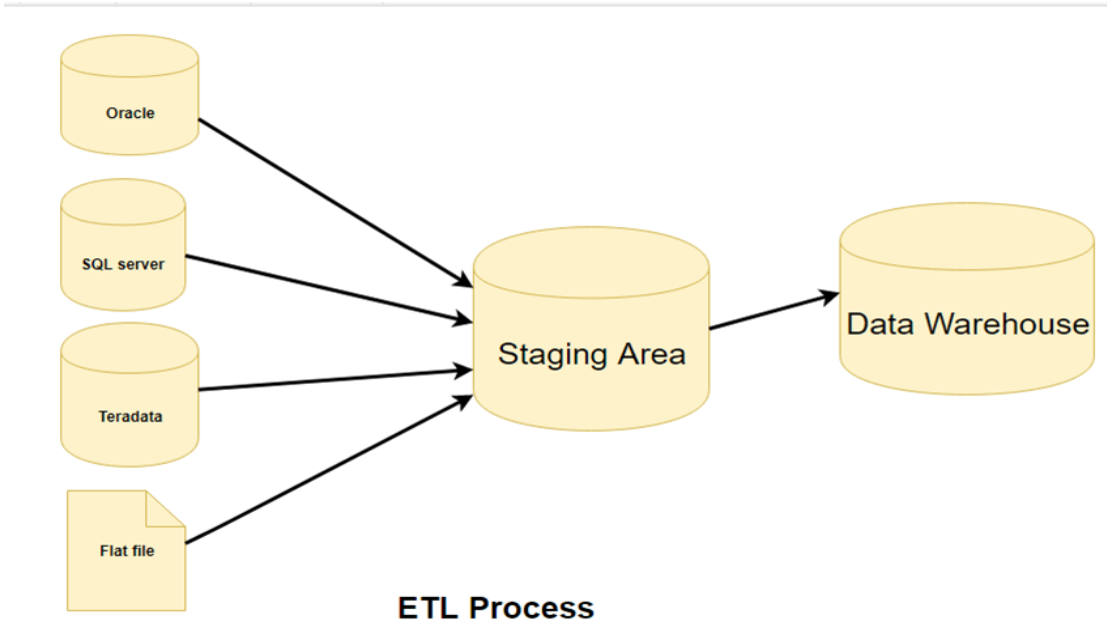
- Data preparation is a big issue for both warehousing and mining
- Data preparation includes
  - Data cleaning and data integration
  - Data reduction and feature selection
  - Discretization
- A lot a methods have been developed but still an active area of research

- ETL stands for Extract, Transform and Load.
- An ETL tool extracts the data from different RDBMS source systems, transforms the data like applying calculations, concatenate, etc. and then load the data to Data Warehouse system.
- The data is loaded in the DW system in the form of dimension and fact tables.

- **Why do you need ETL?**
- It helps companies to analyze their business data for taking critical business decisions.
- provides a common data repository
- ETL provides a method of moving the data from various sources into a data warehouse.

- As data sources change, the Data Warehouse will automatically
  update.
- Allow verification of data transformation, aggregation and calculations rules.
- allows sample data comparison between the source and the
  target system.
- ETL process can perform complex transformations and requires the extra area to store the data. (Data Lake)

- ETL is a 3-step process
- **Step 1) Extraction**
- **Step 2) Transformation**
- **Step 3) Loading**



ETL Process

- ETL is a 3-step process
- **Step 1) Extraction**
- **Three Data Extraction methods:**

1. Full Extraction
2. Partial Extraction- without update notification.
3. Partial Extraction- with update notification

Step 2) Transformation

- this is the key step where ETL process adds value and changes data such that insightful BI reports can be generated.
- Data that does not require any transformation is called as **direct move** or **pass through data**.
- In transformation step, you can perform customized operations on data. For instance, if the user wants sum-of-sales revenue which is not in the database , you can apply the **SUM** formula during transformation and load the data

Step 3) Loading

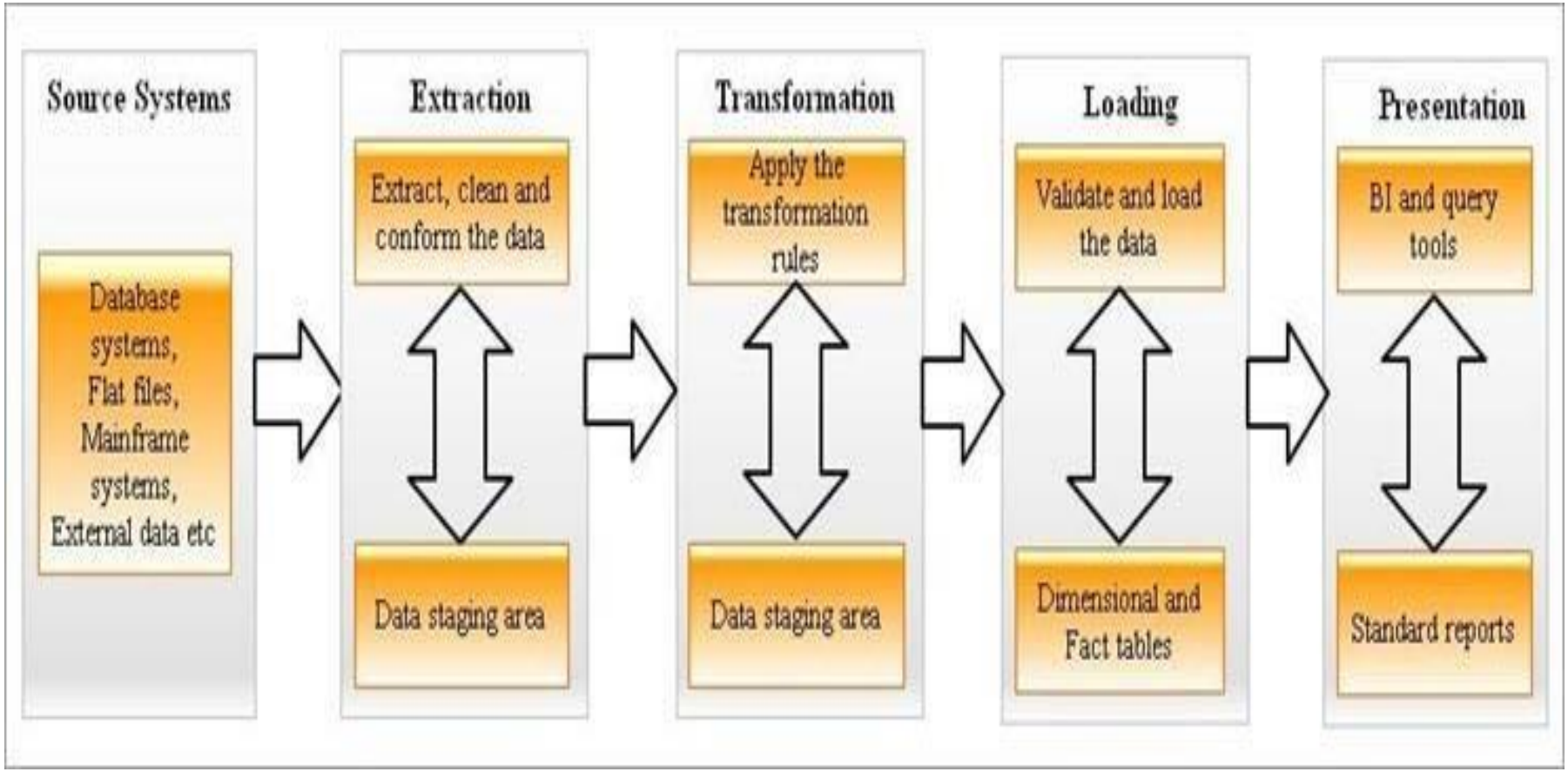
- it is the last step of the ETL process. In a typical Data warehouse, huge volume of data needs to be loaded in a relatively short period (nights Hence, load process should be optimized for performance.
- In case of load failure, recover mechanisms should be configured to restart from the point of failure without data integrity loss. Data Warehouse admins need to monitor, resume, cancel loads as per prevailing server performance
- **Types of Loading:**

➢ **Initial Load** — populating all the Data Warehouse tables

➢ **Incremental Load** — applying ongoing changes as when needed periodically.

➢ **Full Refresh** —erasing the contents of one or more tables and reloading with fresh data.

- **ETL Process Flow Diagram:**



- **ETL Tools**

some most prominent one:

1.MarkLogic: MarkLogic is a data warehousing solution which makes data integration easier and faster using an array of enterprise features. It can query different types of data like documents, relationships, and metadata.

2.Oracle: Oracle is the industry-leading database. It offers a wide range of choice of Data Warehouse solutions for both on-premises and in the cloud. It helps to optimize customer experiences by increasing operational efficiency.

3.Amazon RedShift: Amazon Redshift is Datawarehouse too It is a simple and cost-effective tool to analyze all types of data using standard SQL and existing BI tools. It also allows running complex queries against petabytes of structured data.

Best practices ETL process

- Following are the best practices for ETL Process steps:
- **Never try to cleanse all the data:**
- **Never cleanse Anything:** Always plan to clean something because the biggest reason for building the Data Warehouse is to offer cleaner and more reliable data.
- **Determine the cost of cleansing the data:**
- **To speed up query processing, have auxiliary views and indexes:**

	ELT	ETL
Order of Processes	Extract Load Transform	Extract Transform Load
Flexibility	Because transformation is not dependent on extraction, ELT is more flexible than ETL for adding more extracted data in the future.	More upfront planning should be conducted to ensure that all relevant data is being integrated.
Administration	More administration may be required as multiple tools may need to be adopted.	Typically, a single tool is used for all three stages perhaps simplifying administration effort.
Development Time	With a more flexible approach, development time may expand depending upon requirements and approach.	ETL requires upfront design planning, which can result in less overhead and development time because only relevant data is processed.
End Users	Data scientists and advanced analysts	Users reading reports and SQL coders

	ELT	ETL
Maturity	ELT is a relatively new practice, and as such there is less expertise and fewer best practices available.	ETL is a mature practice that has existed since the 1990s. There are many skilled technicians, best practices exist, and there are many useful ETL tools on the market.
Data Stores	Mostly Hadoop, perhaps NoSQL database. Rarely relational database.	Almost exclusively relational database.
Use Cases	Best for unstructured data and nonrelational data. Ideal for data lakes. Can work for homogeneous relational data, too. Well-suited for very large amounts of data.	Best for relational and structured data. Better for small to medium amounts of data.