A diagram of knowledge and wisdom

AI-generated content may be incorrect.**Contents of Human Mind**

**• Data: Symbols**

• Data represents a fact or a statement of event without relation to other things.

• Data is raw. It simply exists and has no significance beyond its existence (in and of itself). It can exist in any form, usable or not. It does not have meaning of itself. In computer parlance, a spreadsheet generally starts out by holding data.

• Ex: It is raining. Content

**• Information:**

Data that are processed to be useful; provides answer to “who”, “what”, “where”, and “when” questions.

• Information is data that has been given meaning by way of relational connection. This "meaning" can be useful, but does not have to be.

• Information embodies the understanding of a relationship of some sort, possibly cause and effect.

• Example The temperature dropped 15 degrees and then it started raining. • In computer parlance, a relational database makes information from the data stored within it.

**• Knowledge:**

application of data and information; answers “how” questions.

• **Knowledge** = application of data + information; answers **“how”**.

• Knowledge is the appropriate collection of information, such that it's intent is to be useful. Knowledge is a deterministic process.

• Example: *High humidity + sudden temperature drop → it rains.*

**• Understanding:** appreciation of why

**• Wisdom:** evaluated understanding

• Wisdom embodies more of an understanding of fundamental principles embodied within the knowledge that are essentially the basis for the knowledge being what it is. Wisdom is essentially systemic.

Ex: *Rain happens because of evaporation, air currents, temperature gradients*

***Data mining definition and KDD process***

What Is Data Mining?

 **Definition**: Extracting **interesting, non-trivial, implicit, previously unknown, potentially useful patterns/knowledge** from data.

 Alternative names: KDD, knowledge extraction, pattern analysis, BI, etc.

 Not just search/query.

Knowledge discovery (mining) in databases (KDD)

**KDD (Knowledge Discovery in Databases)** = process with stages:

1. Domain understanding
2. Data cleaning
3. Data integration
4. Data selection
5. Data reduction/transformation
6. Data mining (classification, clustering, etc.)
7. Pattern evaluation & presentation

***Multidimensional view of data mining***

 Data types: relational DBs, streams, time-series, text, graphs, multimedia, web.

 Knowledge mined: summarization, association, clustering, classification, outlier, trend.

 Techniques: OLAP, machine learning, statistics, visualization.

 Applications: retail, banking, telecom, fraud detection, bioinformatics.

***Data Mining Functionalities***

• Generalization/Summarization: characterization and discrimination

• Pattern mining, Association mining, correlation

• Classification

• Clustering

• Outlier analysis

• Sequential, trend and evolution analysis

• Structure and network analysis

**🔹 1. Summarization / Generalization**

**Definition:** Producing a compact, high-level description of a dataset.

* **Characterization:** describe general properties of a class of data.
  + *Example:* Summarizing “graduate students” → avg. age 23, mostly CS dept, 70% from India.
* **Discrimination:** compare/contrast two classes of data.
  + *Example:* Customers who churn vs. customers who stay.

**Methods:**

* Statistical summaries (mean, median, mode).
* Attribute-Oriented Induction (AOI).
* Data cube/OLAP summarization.

👉 **Purpose:** to simplify large data into understandable summaries for decision-making.

**🔹 2. Association & Correlation (Pattern Mining)**

**Definition:** Finding relationships among items that frequently occur together in a dataset.

* **Association Rule Example:**
  + Diaper → Beer [support=0.5%, confidence=75%]
  + Meaning: in 0.5% of transactions, diapers and beer were bought together, and in 75% of diaper purchases, beer was also bought.
* **Support:** proportion of transactions containing both A and B.
* **Confidence:** probability of B given A.

👉 **Use cases:** Market basket analysis, recommender systems, cross-selling.

⚠️ **Important:** Correlation ≠ causation (just because two items are linked doesn’t mean one causes the other).

**🔹 3. Classification**

**Definition:** Supervised learning — build a model that maps input attributes → discrete labels.

* **Training data:** labeled examples.
* **Goal:** predict unknown labels for new data.

**Examples:**

* Spam filtering (spam / not spam).
* Medical diagnosis (disease / no disease).
* Credit risk (low / medium / high).

**Methods:**

* Decision Trees
* Naïve Bayes
* Logistic Regression
* Neural Networks
* Support Vector Machines
* k-Nearest Neighbors
* Ensembles (Random Forests, Boosting).

👉 **Purpose:** prediction and classification into predefined categories.

**🔹 4. Clustering**

**Definition:** Unsupervised learning — grouping data objects so that objects in the same cluster are similar, and those in different clusters are dissimilar.

* **No labels given** — structure discovered automatically.

**Examples:**

* Customer segmentation in marketing.
* Grouping documents by topics.
* Image compression (pixels clustered).

**Methods:**

* Partitioning (K-means, K-medoids).
* Hierarchical clustering (Agglomerative, Divisive).
* Density-based (DBSCAN).
* Grid-based (STING).
* Model-based (Gaussian Mixtures, EM).

👉 **Purpose:** discover hidden groupings or patterns.

**🔹 5. Outlier Analysis (Anomaly Detection)**

**Definition:** Detecting data objects that do not conform to the general behavior of the data.

**Examples:**

* Fraud detection in credit cards.
* Intrusion detection in cybersecurity.
* Fault detection in sensors.

**Methods:**

* Statistical tests (3σ rule).
* Distance-based (far from neighbors).
* Density-based (low-density regions in DBSCAN).
* Residual analysis in regression.

👉 **Purpose:** find rare, unusual but potentially important events.

**🔹 6. Sequential Pattern, Trend, and Evolution Analysis**

**Definition:** Discovering patterns over time and ordering.

**Examples:**

* Customer shopping sequences: “Buy camera → then buy SD card → then buy tripod.”
* Web clickstream analysis.
* Biological sequence motifs.
* Stock market trends.

**Tasks:**

* Sequential pattern mining.
* Time-series analysis (forecasting).
* Periodicity analysis.
* Stream data mining (continuous, infinite, fast).

👉 **Purpose:** understand order and evolution of behavior.

**🔹 7. Structure & Network Analysis**

**Definition:** Mining structured, relational, and networked data (graphs, social networks, web).

**Examples:**

* **Graph mining:** frequent subgraphs in chemical compounds.
* **Social networks:** community detection, influence spread.
* **Link mining:** PageRank, HITS algorithm (Google search).
* **Web mining:** user behavior, community discovery, opinion mining.

👉 **Purpose:** exploit structural relationships, not just attributes.

**🔹 Summary Table (All Functionalities)**

| **Functionality** | **Definition** | **Example** |
| --- | --- | --- |
| Summarization | High-level description (characterization/discrimination) | Avg. salary by dept |
| Association & Correlation | Find co-occurring items & relationships | Diaper → Beer |
| Classification | Supervised prediction of labels | Spam detection |
| Clustering | Unsupervised grouping | Customer segmentation |
| Outlier Analysis | Detect anomalies | Credit card fraud |
| Sequential/Trend | Order/time-based patterns | Shopping sequences |
| Structure/Network | Mining graphs & networks | PageRank, community detection |

A diagram of data mining

AI-generated content may be incorrect.

**About Big Data**

**Definition (slide):** data whose scale, diversity, and complexity require new architectures, techniques, algorithms, and analytics to manage and extract value.

**Key:** not just size — complexity and heterogeneity matter

Characteristics of Big data :

**volume :**

* **Definition:** sheer amount of data (bytes). New data collectors (phones, sensors, video) massively increase volume.
* **Implication:** storage & compute must scale; analytics must handle huge datasets.
* **Slide point:** data volume increased dramatically (slide cites a 44x increase 2009→2020, 0.8 ZB → 35 ZB).
* **Implication:** exponential growth forces new storage/processing models.

**Variety :**

* **Definition:** multiple data formats/types — text, numeric, images, audio, video, time series, graphs, social media.
* **Challenge:** integrating information across heterogeneous representations (requires feature extraction & alignment).

**Velocity**

* **Definition:** speed of data generation & required processing (real-time/near-real-time).
* **Examples:** location-based promotions, healthcare sensor alerts.
* **Implication:** late processing may miss opportunities — need stream processing.

**Veracity**

* **Definition:** trustworthiness / quality / uncertainty of data.
* **Implication:** data cleaning, master data management, governance are crucial — garbage in → garbage out.

**Value**

* **Definition:** data by itself is not useful unless turned into value (ROI).
* **Implication:** focus on analytics that produce actionable insights.

**Big Data: 3V’s**

* **Summarizes:** Volume, Variety, Velocity (slide also discussed Veracity & Value — sometimes called 5Vs).
* **Takeaway:** design and algorithm choices driven by these characteristics.

A diagram of data processing

AI-generated content may be incorrect.

When we deal with **data management systems**, traditionally there were two major categories:

1. **OLTP (Online Transaction Processing)**
   * Used for day-to-day business operations.
   * Handles *lots of small, fast transactions*.
   * Example: ATM withdrawal, flight booking, e-commerce checkout.
   * Focus: **speed & reliability** of single transactions.
2. **OLAP (Online Analytical Processing)**
   * Used for **decision support & analysis**.
   * Data is collected in a **Data Warehouse**, integrated from multiple OLTP systems.
   * Queries are more complex: aggregations, summaries, comparisons.
   * Example: *“What were sales in Q1 2025 by region and product category?”*
   * Focus: **historical analysis, summarization, trends**.

**🔹 Big Data Challenge**

* With **Big Data** (social media streams, IoT sensors, financial tick data, clickstreams):
  + Data arrives **fast and continuously**.
  + Waiting for OLAP batch processing is **too slow**.
  + Businesses want to react **in real time** (e.g., fraud detection, stock trading, personalized ads).

**🔹 RTAP (Real-Time Analytics Processing)**

* A **new layer** needed beyond OLTP and OLAP.
* **RTAP definition:** systems designed to analyze large-scale data streams in *real time* (as the data is being generated).
* **Examples:**
  + A bank detecting fraudulent credit card activity within seconds.
  + A hospital analyzing patient sensor readings instantly to prevent emergencies.
  + An e-commerce site recommending products while the customer is still browsing.

**🔹 Implication**

* Traditional Data Warehouses (OLAP) **alone are insufficient** for Big Data.
* To truly **harness Big Data**, we need:
  + OLTP for operations,
  + OLAP for batch analysis,
  + **RTAP for streaming, real-time decisions**.

✅ **In simple words:**

* OLTP = “What just happened?” (transaction done).
* OLAP = “What happened in the past?” (historical reports).
* RTAP = “What is happening *right now*, and what should we do?” (real-time insights).

**What’s Driving Big Data**

Big Data didn’t just happen because storage became cheap — it’s driven by **business and scientific needs**:

1. **Ad-hoc querying**
   * Users want the ability to ask *any* question about the data, not just pre-defined reports.
   * Example: *“Show me how sales of product X changed in cities with rainfall above 10 cm last week.”*
2. **Advanced analytics**
   * Beyond simple reporting: machine learning, predictive modeling, text mining, image analysis.
   * Example: Netflix predicting which show you’ll watch next.
3. **Large and real-time datasets**
   * Data isn’t just large, it’s **continuous** (streams from sensors, social media, financial markets).
   * Example: Twitter firehose = millions of tweets per minute.
4. **Complex statistical analysis**
   * Businesses want deep insights — correlations, causal modeling, predictive analytics.
   * Example: analyzing medical data to detect disease risks early.
5. **Multiple data sources**
   * Data now comes from diverse origins: databases, logs, sensors, social media, videos, GPS.
   * Example: Uber combines driver GPS, traffic, weather, and rider requests simultaneously.

👉 **Summary:** Big Data is driven by the need for **flexibility, sophistication, scale, and real-time insights.**

**Value of Big Data Analytics**

Big Data Analytics is valuable because it supports **faster, more adaptive decision-making**:

* **Compared to classic Data Warehousing (DW):**
  + DW systems → batch, overnight processing → insights are delayed.
  + Big Data apps → **real-time or near real-time**.
* **Architectures for Big Data:**
  + **Shared-nothing MPP (Massively Parallel Processing):**
    - Each machine works independently with its own CPU, memory, storage.
    - Results are combined at the end.
    - Scales well — just add more machines.
    - Example: Amazon Redshift, Teradata.
  + **Scale-out clusters:**
    - Instead of buying one huge server (scale-up), you connect many smaller commodity servers.
    - Example: Hadoop, Spark clusters.

👉 **Why valuable:** Big Data analytics allows companies to make **timely, data-driven decisions** that were impossible with traditional systems.

**Challenges in Handling Big Data**

Even with great potential, Big Data has serious **challenges**:

1. **Technological Bottlenecks**
   * Need new storage/processing frameworks (HDFS, Spark, Kafka).
   * Algorithms must be redesigned to handle distributed, large-scale, and streaming data.
   * Challenge: balance **speed, accuracy, and cost**.
2. **Skill Bottlenecks**
   * Shortage of **data engineers** (who build pipelines) and **data scientists** (who analyze & model data).
   * Skills required: statistics, machine learning, big-data platforms, domain knowledge.
   * This talent gap makes it harder for organizations to adopt Big Data effectively.
3. **Implication**
   * It’s not enough to have tools; you need skilled people + organizational commitment.
   * Without this, Big Data projects often fail to deliver value.

👉 **Summary:** Handling Big Data requires both **technological innovation** and **human expertise**.

About Big Data: Conclusion

• Beginning of big data economy.

• Big data and data science will bring a major social change.

• Companies which fail to exploit big data runs the risk of left behind.

• Should be exploited for sustainable development and equitable society

**Applications of Data Mining**

 Examples: web analysis (classification, PageRank), recommender systems, basket analysis, medical/bioinformatics (microarrays), software engineering applications.

 Point: data mining is cross-domain and present in many products/services.

**Major Issues in Data Mining (1)**

**1. Mining Methodology**

* **Mining various and new kinds of knowledge**
  + Not just frequent itemsets — but sequences, time-series, graphs, multimedia, text, spatiotemporal, bioinformatics patterns.
* **Mining in multidimensional space**
  + Data has many attributes (e.g., age, gender, location, time). Mining must capture relationships across multiple dimensions.
* **Interdisciplinary nature**
  + Data mining uses **databases, statistics, AI, machine learning, visualization, high-performance computing, domain knowledge**.
* **Boosting discovery in a networked environment**
  + Modern data is **distributed across networks/clouds** — mining must work in distributed settings (e.g., federated learning).
* **Handling noise, uncertainty, incompleteness**
  + Real-world data = messy. Techniques needed: data cleaning, missing value handling, robust algorithms.
* **Pattern evaluation & constraint-guided mining**
  + Not all patterns are interesting. Research focuses on:
    - *Interestingness measures* (novelty, coverage, usefulness).
    - *Constraint-based mining* (user specifies what kind of patterns are useful).

**2. User Interaction**

* **Interactive mining**
  + Users should be able to explore results iteratively (drill down, filter, zoom in/out).
* **Use of background knowledge**
  + Incorporating domain knowledge (hierarchies, ontologies, rules) improves relevance of results.
* **Presentation & visualization**
  + Patterns must be shown in a form humans can understand: charts, dashboards, visual summaries.

👉 **Takeaway for Part 1:** Research must ensure mining is flexible, relevant, interpretable, and works on complex, noisy, distributed data.

**🔹 Major Issues in Data Mining (2)**

**1. Efficiency and Scalability**

* **Efficiency of algorithms**
  + Must handle *very large* datasets (terabytes–petabytes).
* **Parallel and distributed methods**
  + Run on clusters (Hadoop, Spark) or cloud platforms.
* **Stream & incremental mining**
  + Data is arriving continuously — need one-pass or online updating algorithms (not full recomputation).

**2. Diversity of Data Types**

* **Complex data:**
  + Text, images, audio, video, graphs, social networks, sequences, spatiotemporal.
* **Dynamic & networked repositories:**
  + Social media, sensor networks, IoT, web logs — constantly changing.
* **Global repositories:**
  + Data is worldwide, heterogeneous, and often inconsistent.
* **Legacy systems:**
  + Old DBs in incompatible formats must still be integrated into mining.

**3. Data Mining and Society**

* **Social impacts:**
  + Mining decisions influence jobs, finance, healthcare, governance.
* **Privacy-preserving data mining (PPDM):**
  + Must protect sensitive info (anonymization, encryption, differential privacy).
* **Invisible data mining:**
  + Mining is embedded in everyday applications (Google search, YouTube recommendations, e-commerce) often without user awareness.
* **Ethical concerns:**
  + Bias in models, discrimination, surveillance misuse.

👉 **Takeaway for Part 2:** Research must ensure mining is **scalable, capable of handling diverse data, socially responsible, and privacy-friendly.**

**🔹 Important Characteristics of Structured Data**

* **Dimensionality:**
  + Too many features (high-dimensionality) → “curse of dimensionality” (distance measures become less meaningful).
  + Solution: *feature selection / dimensionality reduction (PCA, LDA)*.
* **Sparsity:**
  + Data is often sparse (e.g., document-term matrices, recommender systems).
  + Sparsity can be exploited for efficiency.
* **Resolution:**
  + The level of granularity matters — too coarse = lose detail, too fine = overwhelming.
* **Distribution:**
  + Data distribution (central tendency, variance, skewness) affects algorithm performance.

**🔹 Multi-Dimensional View of Data Mining**

* **Data to be mined:** relational DBs, warehouses, streams, text, web, multimedia, graphs.
* **Knowledge mined:** summarization, association, classification, clustering, outlier/trend analysis.
* **Techniques:** OLAP, statistics, machine learning, visualization.
* **Applications:** retail, banking, telecom, fraud, bioinformatics, web mining.

👉 Wrap-up view: data mining is multi-dimensional in **data, methods, and applications**.

2nd Lecture Done ✅

Lecture 3

**Data Objects and Attribute Types**

* **Data set = collection of data objects.**
  + A **data object** represents an entity in the real world (customer, product, student, transaction, gene, image, etc.).
  + Objects are also called: **records, points, cases, tuples, samples, examples, instances.**
* **Attributes (features, dimensions, variables):**
  + Properties used to describe a data object.
  + Example: *Customer* → (CustomerID, Name, Age, Gender, Income).
  + Example: *Student* → (Roll No., CGPA, Branch, Credits earned).

👉 **Key:** Data objects = rows; Attributes = columns.

* **Attribute value:** the specific data that fills a property for a given object.
  + Example: Student Roll No. = 101, CGPA = 8.2, Branch = CSE.
* **Important:** The **attribute type** (not the value) determines:
  + Which *statistical summaries* apply.
  + Which *distance/similarity measures* are valid.
  + Which *mining algorithms* are appropriate.

👉 Example: You can compute an *average* of Age (numerical), but not of Gender (categorical).

A white text with black text

AI-generated content may be incorrect.A black and white text on a white background

AI-generated content may be incorrect.A diagram of attributes

AI-generated content may be incorrect.A close-up of a document

AI-generated content may be incorrect.

***Basic Statistical Descriptions of Data***

Statistical methods are used to **understand, summarize, and explore** datasets.  
Two main categories:

1. **Measures of Central Tendency** (middle or centre of where values “tend” to cluster).
   1. Mean, Median, Mode, Geometric Mean, Harmonic Mean
2. **Measures of Dispersion/Spread** (how far values are spread out).
   1. Range, Quartile Deviation, Mean Deviation,
   2. Standard Deviation, Coefficient of Variation
3. Skewness
4. Constructing a boxplot

**Measures of Central Tendency**

A group of data is represented with a single number and It brings very important information from itA white paper with black text and black text

AI-generated content may be incorrect.

**(b) Median :** Middle value when data is sorted.

* Example: {10, 20, 30, 40, 1000} → median = 30 (not influenced by extreme outlier 1000).
* **Good for skewed data.**

**(c) Mode**

* Most frequent value.
* Example: {1, 1, 2, 3, 3, 3, 4} → mode = 3.

A close-up of a text

AI-generated content may be incorrect.

**🔹 2. Measures of Dispersion / Spread (p.76–77)**

These tell how **scattered or consistent** the data values are.

**(a) Range**

* Max – Min.
* Example: Heights {150, 160, 170, 180, 190} → Range = 190 – 150 = 40.
* **Cons:** very sensitive to outliers.

**(b) Variance (σ²)**

* Average squared deviation from the mean.
* Formula: A black text on a white background

  AI-generated content may be incorrect.
* Sigma is standard deviation
* Variation =sigma squared deviation
* Example: data close to mean → low variance; far apart → high variance.

**(c) Standard Deviation (σ)**

* Square root of variance.
* Easier to interpret because it has same unit as data.
* **Example:** if average salary = ₹50,000 and σ = 5,000 → most salaries lie within ₹45,000–55,000.

**(d) Quartiles & Interquartile Range (IQR)**

* **Quartiles:** divide data into 4 equal parts.
  + Q1 = 25th percentile, Q2 = 50th percentile (median), Q3 = 75th percentile.
* **IQR = Q3 – Q1.**
* Example: incomes = {10, 12, 14, 15, 18, 20, 100}
  + Q1=12, Q2=15, Q3=20 → IQR=8.
* **Good for detecting outliers.**

Coefficient of Variation

• CV is the percentage ratio of S.D to mean.

• CV= (SD/Mean)\*100

• For a player scores: if CV is high he/she is inconsistent

A math problem with numbers and equations

AI-generated content may be incorrect.A black and white text with black text

AI-generated content may be incorrect.**Covariance and correlation analysis**

A white text with black text

AI-generated content may be incorrect.

**🔹 Correlation Analysis (for Categorical Data)**

Correlation means: **Do two attributes depend on each other? Or are they independent?**

For **categorical attributes** (Nominal or Ordinal), the main tool is the **Chi-Square (χ²) Test of Independence**.

**1. Contingency Table (Cross Tabulation)**

* A **contingency table** shows the frequency counts for combinations of two categorical attributes.
* Example: Study if **Gender** and **Likes Ice Cream** are related.

| **Gender** | **Likes Ice Cream = Yes** | **Likes Ice Cream = No** | **Row Total** |
| --- | --- | --- | --- |
| Male | 30 | 10 | 40 |
| Female | 20 | 20 | 40 |
| **Col Total** | **50** | **30** | **80** |

👉 Question: *Is preference for ice cream independent of gender?*

**2. Expected Frequency**

* If two attributes are **independent**, then
* A black background with white text

  AI-generated content may be incorrect.

So expected table =

| **Gender** | **Yes** | **No** | **Row Total** |
| --- | --- | --- | --- |
| Male | 25 | 15 | 40 |
| Female | 25 | 15 | 40 |
| Total | 50 | 30 | 80 |

A black background with white text

AI-generated content may be incorrect.A black background with white text

AI-generated content may be incorrect.A white text with black text

AI-generated content may be incorrect.**3. Chi-Square Statistic (χ²)**

A math problem with numbers and lines

AI-generated content may be incorrect.A screenshot of a test

AI-generated content may be incorrect.Example :

A black text with black text

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

Graphic Displays of Basic Statistical Descriptions

• Boxplot: graphic display of five-number summary

• Histogram: x-axis are values, y-axis repres. frequencies

• Quantile plot: each value xi is paired with fi indicating that approximately 100 fi % of data are ≤ xi

• Quantile-quantile (q-q) plot: graphs the quantiles of one univariant distribution against the corresponding quantiles of another

• Scatter plot: each pair of values is a pair of coordinates and plotted as points in the plane

**🔹 Constructing a Box-and-Whisker Plot**

A **boxplot** is drawn based on the **five-number summary** of data:

1. **Minimum** (lowest value, excluding outliers)
2. **Q1 (First Quartile = 25th percentile)**
3. **Median (Q2 = 50th percentile)**
4. **Q3 (Third Quartile = 75th percentile)**
5. **Maximum** (highest value, excluding outliers)

**🔸 Steps to Construct**

1. **Order the data** (ascending).
2. **Find quartiles (Q1, Q2, Q3).**
   * Q1 = 25% mark.
   * Q2 = median.
   * Q3 = 75% mark.
3. **Draw a box** from Q1 to Q3.
   * Inside the box, draw a line at the **median (Q2)**.
4. **Draw whiskers (lines)** from the box out to the minimum and maximum values (not counting extreme outliers).
5. **Outliers:**
   * Defined as values beyond **1.5 × IQR** from the quartiles.
   * They are plotted as separate points.

**🔸 Example**

Dataset: {7, 8, 5, 5, 10, 12, 14, 15, 18, 21}

1. **Sorted:** {5, 5, 7, 8, 10, 12, 14, 15, 18, 21}
2. **Median (Q2):** middle of dataset = (10 + 12)/2 = 11.
3. **Q1 (25th percentile):** lower half = {5, 5, 7, 8, 10} → median = 7.
4. **Q3 (75th percentile):** upper half = {12, 14, 15, 18, 21} → median = 15.
5. **Min = 5, Max = 21.**
6. **IQR = Q3 – Q1 = 15 – 7 = 8.**
   * Outlier cutoff = 1.5 × IQR = 12.
   * Lower bound = Q1 – 12 = -5 → none below.
   * Upper bound = Q3 + 12 = 27 → none above.
   * ✅ No outliers.

**So the boxplot will look like:**

* Box = from 7 (Q1) to 15 (Q3).
* Median line at 11.
* Whiskers at 5 (min) and 21 (max).

**🔸 Interpretation**

* **Box length (IQR):** shows spread of middle 50% of data.
* **Median line:** central tendency.
* **Whiskers:** data range.
* **Outliers:** unusual/extreme values (plotted separately).

**🔹 Why Boxplots are Useful**

* Summarize data distribution **visually**.
* Show **skewness**:
  + If median closer to Q1 → data skewed right (positive skew).
  + If median closer to Q3 → skewed left (negative skew).
* Show **spread & outliers** clearly.
* Useful for **comparing distributions** across groups.

✅ **In simple words:**  
A boxplot is like a “data fingerprint.” It shows **where most values lie, how spread out they are, and if there are any unusual values (outliers).**

**🔹 Graphing the Data**

There are several standard ways to visualize data:

**1. Histogram**

* A **bar-like graph** showing how data is distributed across intervals (called *bins*).
* **X-axis:** ranges of values (bins).
* **Y-axis:** frequency/count in each bin.

**Example:**

* Test scores of 100 students grouped into bins:
  + 0–10: 2 students, 10–20: 5 students, …, 90–100: 8 students.
* Histogram shows where most scores lie (say between 50–70).

**Usefulness:**

* Shows **shape** of distribution (normal, skewed, uniform, etc.).
* Easy to spot outliers.

**2. Scatter Plot**

* Each point = one object with 2 attributes (X, Y).
* Shows **relationship between two variables**.

**Example:**

* X = Hours studied, Y = Exam score.
* Scatter plot shows a **positive trend** (more study → higher score).

**Usefulness:**

* Identifies **correlation**, clusters, or unusual points.

**3. Boxplot (Box-and-Whisker Plot)**

* Already explained earlier.
* Based on five-number summary (Min, Q1, Median, Q3, Max).
* Shows **spread, skewness, outliers**.

**Example:**

* Compare monthly income distributions of IT vs Non-IT workers.
* IT boxplot might be shifted higher, with more outliers (very high earners).

**Usefulness:**

* Great for **comparing groups**.

**4. QQ Plot (Quantile-Quantile Plot)**

* Compares data distribution to a theoretical distribution (often **Normal distribution**).
* Points falling on a straight line → data is close to normal.
* Deviations → skewness or heavy tails.

**Example:**

* If you plot exam marks vs a normal distribution → large deviations at extremes indicate marks are not perfectly normal.

Data Visualization

• Why data visualization?

• Gain insight into an information space by mapping data onto graphical primitives

• Provide qualitative overview of large data sets

• Search for patterns, trends, structure, irregularities, relationships among data

• Help find interesting regions and suitable parameters for further quantitative analysis

• Provide a visual proof of computer representations derived

**Categorization of visualization methods:**

• Pixel-oriented visualization techniques .

• Geometric projection visualization techniques

• Icon-based visualization techniques

• Hierarchical visualization techniques

• Visualizing complex data and relations

Theres alots of shit after this but I don’t feel this will be asked in exam skipppppp……..

Leecture 3 also done ✅

Lecture 4

Proximity: Similarity or Dissimilarity

* **Similarity:** A numerical score of how alike two objects are. A higher value means the objects are more similar, typically scaled between 0 and 1.
* **Dissimilarity (or Distance):** A numerical score of how different two objects are. A lower value means the objects are more alike. The minimum is usually 0.

**Proximity** is the general term used to refer to either similarity or dissimilarity

Dissimilarity/Similarity metric :

**Explanation:** This slide formalizes the concept of dissimilarity as a distance function, denoted by d(i,j). It makes three key points:

1. The choice of distance function depends heavily on the type of data (e.g., numeric, categorical, binary).
2. Different attributes can be given different weights based on their importance for a specific application.
3. A simple way to convert dissimilarity to similarity is using the formula:

sim(i,j)=1−d(i,j)

What is Metric ?

* A distance that satisfies the following properties is a metric
  + Non-negativity: d(i, j) > 0 if i ≠ j, and d(i, i) = 0 (Positive definiteness)
    - Distance is non-negative number
  + Identify: d(i,j) >= 0
    - The distance of an object to itself is 0
  + d(i, j) = d(j, i)(Symmetry)
    - Distance is a symmetric function
  + d(i, j) ≤ d(i, k) + d(k, j)(Triangle Inequality)
* **Data Matrix:** The standard format where rows are data objects (e.g., people) and columns are attributes (e.g., age, height).
* **Dissimilarity Matrix:** A square matrix where both rows and columns represent data objects. The value in each cell is the dissimilarity (distance) between the two corresponding objects. This is often a triangular matrix because the distance from object A to B is the same as from B to A. Many algorithms, like clustering, work directly with this matrix.
* Data matrix
  + n data points with p dimensions
* Dissimilarity matrix
  + n data points, but registers only the distance
  + A triangular matrix
  + Single mode
* Most clustering and nearest neighbor algorithms operate on dissimilarity matrix

Types of Attributes :

**1. Nominal Attributes**

* **Definition**: Attributes that represent **categories** with **no order**.
* They are just “names” or “labels”.
* **Examples**:
  + Hair color = {Black, Brown, Blonde}
  + City = {Hyderabad, Delhi, Mumbai}
  + Gender = {Male, Female, Other}

👉 We cannot say “Delhi > Hyderabad”. It’s just a category, not an order.

**2. Binary Attributes**

* **Definition**: A special case of nominal attributes with only **two states** (yes/no, 0/1).
* Two types:
  + **Symmetric binary**: Both states are equally important.  
    Example: Gender (M/F).
  + **Asymmetric binary**: One state is more important.  
    Example: Medical test result (Positive/Negative). Positive matters more.

**3. Numeric Data**

* **Definition**: Attributes with **quantitative values** (numbers).
* Two types:
  + **Interval-scaled**: Differences are meaningful, but ratio is not.  
    Example: Temperature in Celsius (20°C is 10° more than 10°C, but not “twice as hot”).
  + **Ratio-scaled**: Both differences and ratios are meaningful.  
    Example: Weight (60 kg is twice 30 kg).

**4. Ordinal Data**

* **Definition**: Attributes with **order/rank**, but differences are not meaningful.
* Example:
  + Student grade = {A, B, C, D} → A is higher than B, but the gap between A and B is not the same as B and C.
  + Survey rating = {Poor, Average, Good, Excellent}.

👉 We know order, but not exact “distance” between ranks.

**5. Mixed Data**

* **Definition**: Many real-world datasets contain a **mix of attribute types** (nominal + numeric + ordinal + binary).
* Example:
  + Patient dataset:
    - Name (Nominal)
    - Gender (Binary)
    - Age (Numeric, ratio-scaled)
    - Health rating (Ordinal: Poor/Fair/Good)

👉 In such cases, **special formulas** are used to compute similarity/distance (by combining contributions from each attribute type).

**Methods to measure proximity**

**Method 1: Simple Matching**

* Suppose we have **p attributes** (all nominal).
* Let **m = number of matches** (same value for i and j).
* Then,
* Similarity(I,j)=m/p
* Dissimilarity =1-m/p

**Example:**  
Objects (students) with attributes:

* Student A: {Gender=M, Dept=ECE, Hostel=Yes}
* Student B: {Gender=M, Dept=CSE, Hostel=Yes}

Here p = 3 attributes.  
Matches (m) = 2 (Gender + Hostel).  
Similarity = 2/3 = 0.67.  
Dissimilarity = 1 – 0.67 = 0.33.

**Method 2: Convert Nominal → Binary**

* For each category value, create a **binary attribute** (0/1).
* Example: Car color = {Red, Blue, Green}.
  + New binary features:
    - IsRed? (Yes=1, No=0)
    - IsBlue?
    - IsGreen?

Car1 (Red) → {1,0,0}  
Car2 (Green) → {0,0,1}

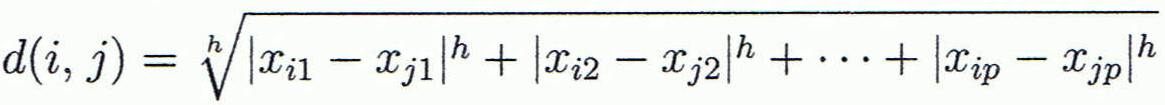
👉 Now we can apply **binary similarity measures** (like Jaccard, Hamming distance, etc.).

A white paper with black text and numbers

AI-generated content may be incorrect.A screenshot of a math equation

AI-generated content may be incorrect.

* *Minkowski distance*: A popular distance measure



where *i* = (*x*i1, *x*i2, …, *x*ip) and *j* = (*x*j1, *x*j2, …, *x*jp) are two *p*-dimensional data objects, and *h* is the order (the distance so defined is also called L-*h* norm)

* *h* = 1: Manhattan (city block, L1 norm) distance
  + E.g., the Hamming distance: the number of bits that are different between two binary vectors



* *h* = 2: (L2 norm) Euclidean distance



* *h* → ∞. “supremum” (Lmax norm, L∞ norm) distance.
  + This is the maximum difference between any component (attribute) of the vectors



Example :

Let’s take 2D points:

* p1 = (0, 2)
* p2 = (2, 0)

**1. Manhattan distance (h=1)**

d(p1,p2) = |0-2| + |2-0| = 2 + 2 = **4**

**2. Euclidean distance (h=2)**

d(p1,p2) = sqrt((0-2)^2 + (2-0)^2) = sqrt(4+4) = sqrt(8) = **2.828**

**3. Supremum distance (h=∞)**

d(p1,p2) = max(|0-2|, |2-0|) = max(2,2) = **2**

**Standardizing Numeric Data**

* **Why?** Attributes like "salary" (range 10,000-100,000) will dominate "age" (range 0-100) in distance calculations. We need to put them on the same scale.
* **Z-score normalization:** z = (x - μ) / σ
  + x: original value, μ: mean of the attribute, σ: standard deviation.
  + This tells you how many standard deviations away from the mean the value is.
* **Mean Absolute Deviation (MAD):** A more robust alternative to standard deviation. z = (x - μ) / MAD.

**Ordinal Variables :**

**order is important**

* **How to handle:** Treat them like numeric data after mapping them to ranks.
  1. Replace the ordinal value (e.g., "small", "medium", "large") with its rank (e.g., 1, 2, 3).
  2. Map the ranks to the [0, 1] interval: z\_i = (rank\_i - 1) / (M - 1), where M is the maximum rank.
  3. Now use numeric distance measures (e.g., Euclidean) on these normalized ranks z\_i.

**Attributes of Mixed Type**

* A dataset may contain all attribute types
  1. Nominal, symmetric binary, asymmetric binary, numeric, and ordinal
  2. 
* One may use a weighted formula to combine their effects:
  1. If *f* is numeric: Use the normalized distance
  2. If *f* is binary or nominal: dij(f) = 0 if xif = xjf; or dij(f) = 1 otherwise
  3. If *f* is ordinal
     + Compute ranks zif  (where )
     + Treat zif as interval-scaled

**Cosine Similarity**

* **Use Case:** Perfect for comparing **text documents** represented as high-dimensional vectors of word frequencies (Term Frequency).
* **Formula:** cos(d1, d2) = (d1 • d2) / (||d1|| \* ||d2||)
  + • is the vector dot product.
  + ||d|| is the magnitude (length) of the vector.
* **Interpretation:** Measures the angle between two vectors. A value of 1 means the vectors are identical in orientation (perfectly similar), 0 means they are orthogonal (completely dissimilar). It ignores magnitude and focuses on direction.

Example :

* cos(*d1, d2*) = (*d1* • *d2*) /||*d1*|| ||*d2*|| ,

where

• indicates vector dot product, ||*d*|: the length of vector *d*

* Ex: Find the **similarity** between documents 1 and 2.

*d1* **=** (5, 0, 3, 0, 2, 0, 0, 2, 0, 0)

*d2* **=** (3, 0, 2, 0, 1, 1, 0, 1, 0, 1)

*d1*•*d2* = 5\*3+0\*0+3\*2+0\*0+2\*1+0\*1+0\*1+2\*1+0\*0+0\*1 = 25

||*d1*||= (5\*5+0\*0+3\*3+0\*0+2\*2+0\*0+0\*0+2\*2+0\*0+0\*0)**0.5**=(42)**0.5** = 6.481

||*d2*||= (3\*3+0\*0+2\*2+0\*0+1\*1+1\*1+0\*0+1\*1+0\*0+1\*1)**0.5**=(17)**0.5** = 4.12

cos(*d1, d2* ) = 0.94

**KL Divergence: Comparing Two Probability Distributions**

* **Key Concept:** Measures how one probability distribution q(x) diverges from a second, expected ("true") distribution p(x). It measures the **information lost** when q(x) is used to approximate p(x).
* **It is NOT a distance metric** because it is **not symmetric** (D\_KL(P||Q) != D\_KL(Q||P)) and does not satisfy the triangle inequality.

**More on KL Divergence**

* **Interpretation:** In machine learning, P is often the true data distribution, and Q is the model's predicted distribution. KL Divergence measures how bad the model's prediction is.
* **Other Names:** Relative entropy, information gain.

**KDD process Data preprocessing is a major step**

 This is the famous KDD (Knowledge Discovery in Databases) pipeline. Data Preprocessing is a major step (often taking 60% of the effort!) that comes after understanding the domain and selecting data, but before applying the actual mining algorithms.

* Learning the application domain:
  + relevant prior knowledge and goals of application
* Creating a target data set: data selection
* **Preprocessing: (may take 60% of effort!)**
* **Data reduction and transformation:**
  + **Find useful features, dimensionality/variable reduction, invariant representation.**
* Choosing functions of data mining
  + summarization, classification, association, clustering.
* Choosing the mining algorithm(s)
* Data mining: search for patterns of interest
* Pattern evaluation and knowledge presentation
  + visualization, transformation, removing redundant patterns, etc.
* Use of discovered knowledge

**Why Data Preprocessing?**

* **Real-world data is dirty:**
  + Incomplete: Missing values (e.g., occupation="").
  + Noisy: Contains errors or outliers (e.g., Salary="-10").
  + Inconsistent: Contains discrepancies (e.g., Age="42", Birthday="03/07/2010").

**Why Is Data Dirty?**

* **Reasons for Incomplete Data:** "Not applicable" values, data not considered important at collection time, equipment malfunctions.
* **Reasons for Noisy Data:** Faulty sensors, human data entry errors, transmission errors.
* **Reasons for Inconsistent Data:** Integrating multiple data sources, violating business rules (e.g., changing a product code but not the related descriptions).

**Why Is Data Preprocessing Important?**

* **The Golden Rule:** **"No quality data, no quality mining results."** Garbage In, Garbage Out (GIGO). The quality of your analysis is directly dependent on the quality of your input data.

***Data Preprocessing***

* Data Preprocessing: An Overview
  + **Data Quality**
  + Major tasks in Data Preprocessing
* Data Cleaning
* Data Integration
* Data Transformation
* Dimensionality Reduction
* Summary

**Data Quality: Why Preprocess the Data?**

* **Key Concept:** Data quality is multi-dimensional. It's not just about accuracy.
  + **Accuracy:** Is the data correct?
  + **Completeness:** Is all the data present?
  + **Consistency:** Is the data conflict-free?
  + **Timeliness:** Is the data up-to-date?
  + **Believability:** Can the data be trusted?
  + **Interpretability:** Is the data understandable?

Timeliness: Out-of-date data can be useless or harmful (e.g., for monthly bonuses).

Believability: Too many errors destroy user trust.

Interpretability: Data encoded with obscure codes is hard to use.

**Major tasks in data pre-processing :**

* **Data cleaning**
  + Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies
  + Apply data cleaning techniques
* **Data integration**
  + Integration of multiple databases, data cubes, or files
  + Examples:
  + 1. customer identification may be referred as customer-id in one database and cust\_id in another database
  + 2. first name may be Bill on one database and William in another database
  + How to create a common database by integrating Facebook and LinkedIn users?
* **Data reduction**
  + Question: Data set is very huge. How to reduce the data set size without jeopardizing the data mining results?
  + Dimensionality reduction
    - Compressed representation
  + Numerosity reduction
    - Regression or log-linear models
  + Data compression
* **Data transformation and data discretization**
  + Better results if the data is transformed, i.e., normalized
  + Normalization: Example age and salary: transform all age and salary values between 0 and 1.
  + Discretization: mining data at different abstraction levels
  + Concept hierarchy generation

***Data Cleaning***

**How to Handle Missing Data?**

* Ignore the tuple: Not good if many tuples have missing values.
* Manual fill-in: Impractical for large datasets.
* Use a global constant: Fill with "unknown" or "N/A". Simple but can be misleading.
* Use attribute mean/mode: Better than a global constant.
* Use attribute mean/mode per class: Smarter. For a customer with missing "income", use the average income of other customers in the same "profession" class.
* **Use most probable value: The most sophisticated method. Use regression, inference, or decision trees to predict the missing value.**

**How to Handle Noisy Data?**

Methods:

* Binning: Smooth ordered data by consulting its neighbors.
* Regression: Fit data to a function to smooth it.
* Clustering: Detect and remove outliers that fall outside clusters.
* Human Inspection: Manually check suspicious values

**Simple Discretization Methods: Binning**

* **Discretization:** Converting numeric values into categorical intervals or bins.
* **Equal-width partitioning:** Divide the range (max-min) into N intervals of equal size. Simple, but sensitive to outliers.
* **Equal-depth (frequency) partitioning:** Divide the data into N intervals, each containing approximately the same number of data points. Handles skewed data better.

**Regression & Cluster Analysis**

* **Regression:** A method for modeling the relationship between attributes to smooth data or predict missing values. The line (or curve) represents the trend.
* **Clustering:** Groups similar data points together. Points that do not belong to any cluster (outliers) can be identified as noise and removed. This is a very effective way to handle noise.

Lecture 5

**Data Cleaning as a Process**

* **Key Concept:** Data cleaning is not a one-off task; it's a process with two main steps:
* 1) Detect discrepancies,
* 2) Transform the data to fix them.

**Step 1: Data discrepancy detection**

* **How to find errors?**
  + Use **metadata** (data about data): know the allowed domain, range, and data types.
  + Use **knowledge** about the data: know business rules (e.g., age cannot be negative).
  + Write **scripts** to automatically check for anomalies.
  + Check for **field overloading** (using a field for a purpose it wasn't intended for).
  + **Rules to check:**
  + **Uniqueness Rule:** Every value should be unique (e.g., primary keys).
  + **Consecutive Rule:** No missing values in a sequence (e.g., invoice numbers).
  + **Null Rule:** Define how null/missing values are represented.
* **Tools:** Data scrubbing (using simple domain checks) and data auditing (using statistical analysis to find outliers).

**Step 2: data transformation & Integration**

* **ETL Tools:** (Extraction, Transformation, Loading) are used in data warehousing to specify and execute data cleaning and transformation rules.
* **Crucial Point:** **Update metadata** whenever you transform the data. If you change a field, the documentation for that field must be updated.

**Data integration**:

* + Combines data from multiple sources into a coherent store

**Step 1: data discrepancy detection**

A white text on a black background

AI-generated content may be incorrect.A white text on a black background

AI-generated content may be incorrect.

**Handling Redundancy in Data Integration**

* **Redundancy:** An attribute may be repeated across tables or be derivable from another attribute (e.g., annual\_revenue can be derived from monthly\_revenue).
* **Detection:** Can be detected using **correlation analysis** (e.g., if two attributes are highly correlated, one might be redundant).

A white paper with black text and black text

AI-generated content may be incorrect.

From here too mathematical so skipped

Till data transformation

**Data Transformation**

* **Definition:** Converting data into forms suitable for mining.
* **Approaches:** Normalization, Discretization, Data Compression, Sampling.

A black and white text with black text

AI-generated content may be incorrect.

**Discretization**

* **Definition:** Replacing raw numeric values with interval labels or conceptual labels (e.g., "Youth", "Adult").
* **Why?** Allows mining at multiple levels of abstraction, reduces data size, and prepares data for certain algorithms (like some classifiers).**Types :**
* **Supervised:** Uses class label information to decide split/merge points (e.g., Decision Trees, Chi-merge).
* **Unsupervised:** Does not use class labels (e.g., Binning, Histograms, Clustering).
* **Split (Top-down):** Start with one interval and split it.
* **Merge (Bottom-up):** Start with each value as its own interval and merge them.

**Data Discretization Methods**

* **List of Techniques:** Binning,
* Histogram Analysis
* Clustering
* Decision-Tree Analysis
* Correlation (χ²) Analysis.

**Simple Discretization: Binning**

* Equal-width (distance) partitioning
  + Divides the range into *N* intervals of equal size: uniform grid
  + if *A* and *B* are the lowest and highest values of the attribute, the width of intervals will be: *W* = (*B* –*A*)/*N.*
  + The most straightforward, but outliers may dominate the presentation
  + Skewed data is not handled well
* Equal-depth (frequency) partitioning
  + Divide the range into *N* intervals, each containing approximately same number of samples
  + Good data scaling
  + Managing categorical attributes can be tricky
* Sorted data for price (in dollars): 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34

\* Partition into equal-frequency (**equi-depth**) bins:

- Bin 1: 4, 8, 9, 15

- Bin 2: 21, 21, 24, 25

- Bin 3: 26, 28, 29, 34

\* Smoothing by **bin means**:

- Bin 1: 9, 9, 9, 9

- Bin 2: 23, 23, 23, 23

- Bin 3: 29, 29, 29, 29

\* Smoothing by **bin boundaries**:

- Bin 1: 4, 4, 4, 15

- Bin 2: 21, 21, 25, 25

- Bin 3: 26, 26, 26, 34

**Discretization by Classification & Correlation Analysis**

* Classification (e.g., decision tree analysis)
  + Supervised: Given class labels, e.g., cancerous vs. benign
  + Using *entropy* to determine split point (discretization point)
  + Top-down, recursive split
  + Details to be covered in Chapter 7
* Correlation analysis (e.g., Chi-merge: χ2-based discretization)
  + Supervised: use class information
  + Bottom-up merge: find the best neighboring intervals (those having similar distributions of classes, i.e., low χ2 values) to merge
  + Merge performed recursively, until a predefined stopping condition

**Concept Hierarchy Generation**

* **Concept hierarchy** organizes concepts (i.e., attribute values) hierarchically and is usually associated with each dimension in a data warehouse
* Concept hierarchies facilitate drilling and rolling in data warehouses to view data in multiple granularity
* Concept hierarchy formation: Recursively reduce the data by collecting and replacing low level concepts (such as numeric values for *age*) by higher level concepts (such as *youth, adult*, or *senior*)
* Concept hierarchies can be explicitly specified by domain experts and/or data warehouse designers
* Concept hierarchy can be automatically formed for both numeric and nominal data. For numeric data, use discretization methods shown.

**Concept Hierarchy Generation   
for Nominal Data**

* Specification of a partial/total ordering of attributes explicitly at the schema level by users or experts
  + *street* < *city* < *state* < *country*
* Specification of a hierarchy for a set of values by explicit data grouping
  + {Urbana, Champaign, Chicago} < Illinois
* Specification of only a partial set of attributes
  + E.g., only *street* < *city*, not others
* Automatic generation of hierarchies (or attribute levels) by the analysis of the number of distinct values
  + E.g., for a set of attributes: {*street, city, state, country*}

**Data Compression**

* **String Compression:** Lossless (e.g., ZIP files). Original data can be perfectly reconstructed.
* **Audio/Video Compression:** Lossy (e.g., MP3, JPEG). Some information is lost for much higher compression.
* **Dimensionality/Numerosity Reduction** can also be seen as a form of data compression.

**Data Cube Aggregation**

* **Data Cube:** A multidimensional representation of data.
* **Aggregation:** Precomputing and storing summaries (sum, count, average) at different levels of detail (e.g., sales per day, per month, per year). This is a prime method for data reduction, as queries can be answered quickly from the cube without scanning the raw data.

**Automatic Concept Hierarchy Generation**

* **Key Concept:** A simple heuristic for automatic hierarchy generation: sort attributes by the number of distinct values they have. The attribute with the **most distinct values** is at the lowest level of the hierarchy (most detailed), and the one with the **fewest distinct values** is at the highest level (most general).

***SAMPLING***

* **Why?** To run mining algorithms on a small, representative **sample** of the large dataset N. This reduces processing time and complexity.
* **Principle:** The sample must be representative of the whole.

**Types of Sampling**

* **Simple random sampling: equal probability of selecting any particular item**
* **Sampling without replacement**
  + **Once an object is selected, it is removed from the population**
* **Sampling with replacement**
  + **A selected object is not removed from the population**
* **Stratified sampling**
  + **Partition (or cluster) the data set, and draw samples from each partition (proportionally, i.e., approximately the same percentage of the data)**

Lecture 6

WAREHOUSE DA

Chapter 3

**What is a Data Warehouse?**

A **data warehouse** is a **separate database** built for **decision making**, not daily transactions. It is:

* **Subject-oriented** (organized by topics like Customer, Product, Sales),
* **Integrated** (combines multiple sources consistently),
* **Time-variant** (stores history over years), and
* **Non-volatile** (loaded and read; not updated like a normal app DB).  
  (Definition attributed to W. H. Inmon.)

**Example:** Your college may have many systems—admissions, hostel, academics, payments. A warehouse merges them so you can ask: “How did average GPA change across batches from 2018–2025?”

**Data Warehouse — Subject-Oriented**

Data is grouped around **major subjects** such as **customer**, **product**, **sales**. The goal is **analysis for managers**, not running daily transactions. It gives a **clean view** of just the subject you care about.

**Example:** A “Sales” subject may exclude HR data because it’s not useful for sales analysis.

**Data Warehouse — Integrated**

It **integrates different sources** (databases, files, logs) using **cleaning and standardization** so names, codes, and units are consistent.

**Example:** “Hotel price” from one source might include tax and be in EUR; another excludes tax and is in INR. The warehouse converts both to **INR before tax** so numbers are comparable.

**Data Warehouse — Time-Variant**

Warehouses keep **long histories** (e.g., 5–10+ years). **Time** is part of keys or context, so you can compare across periods. Operational DBs usually keep only **current** values.

**Example:** You can ask, “What were monthly sales for the last 8 years?”

**Data Warehouse — Non-Volatile**

The warehouse is **physically separate**. It is **loaded** (ETL) and then **queried**. No frequent updates, transactions, or locks like OLTP systems. Two main operations: **initial load** and **read**.

**Example:** Yesterday’s sales are appended; you don’t edit last year’s sales.

**Data Warehouse vs. Heterogeneous DBMS**

Heterogeneous DB integration is **query-driven** (wrappers/mediators translate on the fly)—it can be slow and complex. A warehouse is **update-driven**: sources are integrated **in advance** and stored for **fast querying**.

**Example:** Instead of hitting 5 systems every time you ask a question, you query one consolidated store.

**Data Warehouse vs. Operational DBMS**

* OLTP (on-line transaction processing)
  + Major task of traditional relational DBMS
  + Day-to-day operations: purchasing, inventory, banking, manufacturing, payroll, registration, accounting, etc.
* OLAP (on-line analytical processing)
  + Major task of data warehouse system
  + Data analysis and decision making
* Distinct features (OLTP vs. OLAP):
  + User and system orientation: customer vs. market
  + Data contents: current, detailed vs. historical, consolidated
  + Database design: ER + application vs. star + subject
  + View: current, local vs. evolutionary, integrated
  + Access patterns: update vs. read-only but complex queries

**Why Separate a Warehouse?**

* **Performance**: OLTP tuned for transactions; warehouse tuned for analytics.
* **Different needs**: OLAP needs **history**, **consolidation** from many sources, and **consistent/clean** data. Trying to mix both hurts both.

**Example:** Indexing for fast inserts (OLTP) vs. pre-aggregations for fast scans (OLAP).

**Data Warehouse: A Multi-Tiered Architecture**

Sources (operational DBs/files) → **ETL** (Extract-Transform-Load) → **Storage** (warehouse + possible **data marts**) → **OLAP server/engine** → **Front-end tools** (reports, dashboards, data mining). **Metadata** describes everything.

**Example:** A nightly ETL job loads yesterday’s sales; analysts view dashboards in the morning.

**Three Data Warehouse Models**

* Enterprise warehouse
  + collects all of the information about subjects spanning the entire organization
* Data Mart
  + a subset of corporate-wide data that is of value to a specific groups of users. Its scope is confined to specific, selected groups, such as marketing data mart
    - Independent vs. dependent (directly from warehouse) data mart
* Virtual warehouse
  + A set of views over operational databases
  + Only some of the possible summary views may be materialized

**Extraction, Transformation, and Loading (ETL)**

* **Data extraction**
  + get data from multiple, heterogeneous, and external sources
* **Data cleaning**
  + detect errors in the data and rectify them when possible
* **Data transformation**
  + convert data from legacy or host format to warehouse format
* **Load**
  + sort, summarize, consolidate, compute views, check integrity, and build indices and partitions
* **Refresh**
  + propagate the updates from the data sources to the warehouse
* **Metadata Repository**
* Stores **what’s in the warehouse** (schemas, dimensions, hierarchies), **data lineage** (where data came from, transformations), **currency** (active/archived), performance stats, business definitions/owners. It’s the warehouse’s **dictionary & logbook**.
* **Example:** “dollars\_sold = gross selling price in INR after discount, before tax.”

**Data Lake (and Layers)**

A **data lake** stores **all** types of data (structured/unstructured) in raw form at any scale; you run analytics/ML directly on it. Typical layers: **raw → standardized → cleansed → application → sandbox**, with connectors and catalogs around.

**Two Purposes & Ecosystems → Towards Lakehouse**

* **Analytic (warehouses)**: structured data, SQL/Python
* **Operational/Intelligence (lakes)**: raw data, multiple languages  
  A **lakehouse** lets both workloads run on the lake **without duplicating** into another rigid DB—everyone uses **the most up-to-date** data.

**Example:** BI reports (warehouse-style) and ML feature engineering (lake-style) on the **same** storage.

**Data Warehouse Modeling: Data Cube and OLAP**

**CUBE Operator (Motivation)**

GROUP BY can’t directly produce multi-level totals like **histograms, roll-ups, cross-tabs**. The **CUBE** operator generalizes GROUP BY to compute **all** necessary aggregates across **N dimensions** (including “ALL” totals).

**Example:** For **(Make, Year, Color)**, CUBE gives subtotals by Make×Year, Make×Color, Year×Color, each single dimension, and the grand total.

* N-dimensional Aggregate [sum(), max(),...]
  + Fits relational model exactly**:**
    - **a1, a2, ...., aN, f()**
* Super-aggregate over *N-1* Dimensional sub-cubes
  + - **ALL, a2, ...., aN , f()**
    - **a3 , ALL, a3, ...., aN , f()**
    - **...**
    - **a1, a2, ...., ALL, f()**
  + This is the *N-1* Dimensional cross-tab**.**
* Super-aggregate over *N-2* Dimensional sub-cubes
  + - **ALL, ALL, a3, ...., aN , f()**
    - **...**
    - **a1, a2 ,...., ALL, ALL, f()**
* …

**Slide 30: Conceptual Modeling of Data Warehouses**

* **How we design a data warehouse schema:**
  1. **Star Schema** – One **fact table** (measures like sales) in center, connected to multiple **dimension tables** (time, item, location).
  2. **Snowflake Schema** – A more normalized version of star schema (dimension tables are broken into smaller related tables).
  3. **Fact Constellation (Galaxy Schema)** – Multiple fact tables share dimension tables (like stars forming a galaxy).

👉 **Example:**

* **Star:** Sales fact table connected to Time, Product, Location.
* **Snowflake:** Location table split into City, State, Country tables.
* **Constellation:** Sales fact + Shipping fact both share Time and Item dimensions.