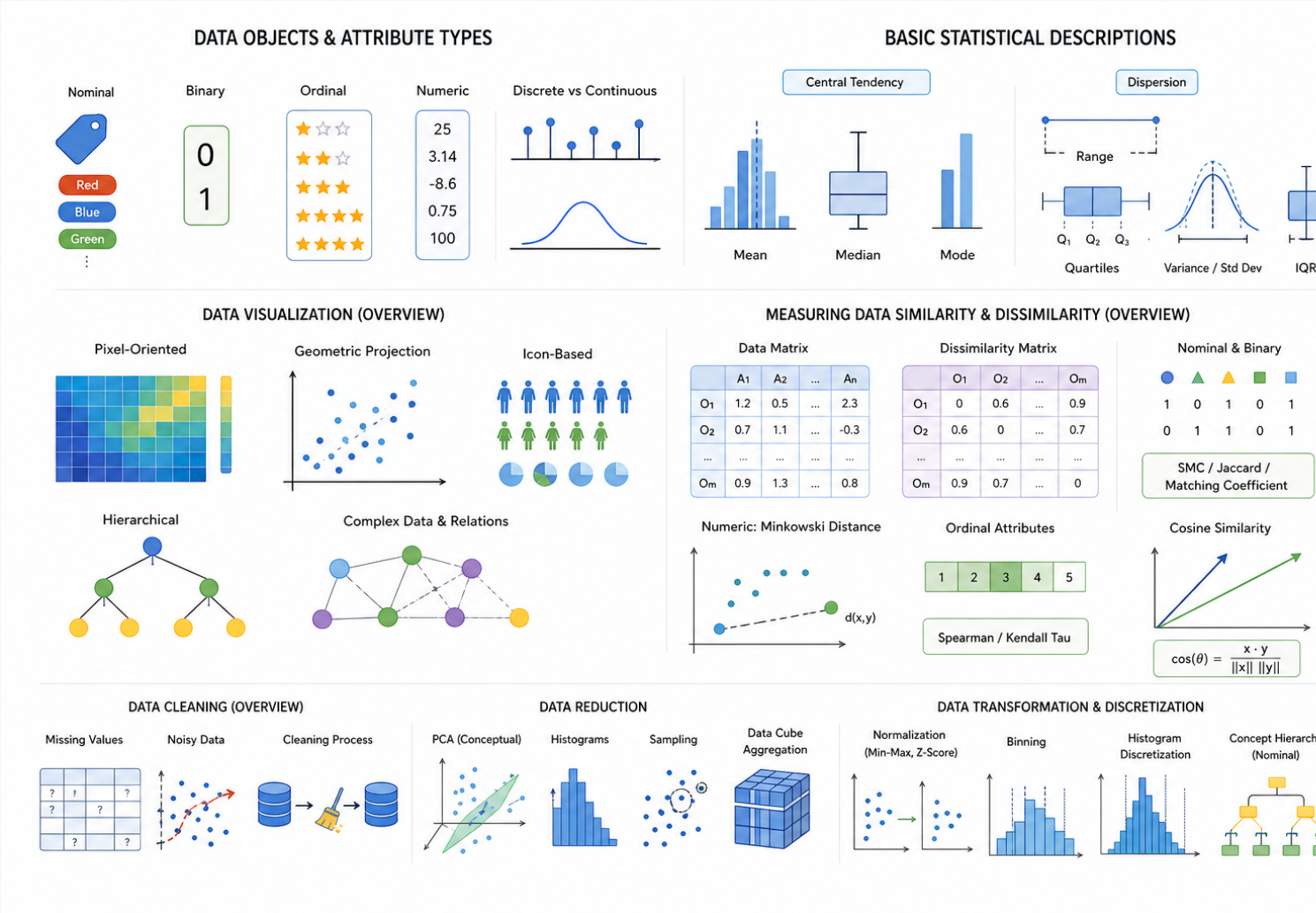
GATE – Data Science & Artificial Intelligence (DA)

Data Warehousing



GateXAIML

Contents

About the Book

Artificial Intelligence and Machine Learning (AI/ML) are transforming industries across the globe — from healthcare and finance to transportation and education. From medical diagnosis systems and fraud detection to personalized recommendations and autonomous vehicles, AI/ML is shaping the way we live, work, and interact with technology.

To support this rapidly growing field, the GATE Data Science and Artificial Intelligence (DA) exam was introduced as a national-level gateway to higher studies, research, and employment opportunities in top institutions and organizations. The exam tests a candidate's proficiency in mathematics, programming, data handling, machine learning, and AI fundamentals.

This book is a compact and comprehensive guide for GATE DA aspirants. It is designed to help learners build a strong conceptual foundation while developing the problem-solving skills required for the exam. Many solved examples are included to illustrate key concepts, and each chapter features carefully crafted problems for practice.

Solutions to selected problems and topic-wise lectures will be discussed in detail on my YouTube channel (@GATEX-Aiml). All the concepts covered in the book will also be taught step-by-step through video tutorials, making this a complete learning resource for GATE DA preparation.

This book is designed for aspirants of the GATE DA exam focusing on **Data Warehousing**. It systematically covers theory, solved examples, and practice problems aligned with the official syllabus.

Dedicated to all my Gurus and Students.
"Knowledge grows only when shared — and it must remain free, for that is how it thrives."

Data Warehousing – Syllabus

Data transformation such as normalization, discretization, sampling, compression; data warehouse modelling: schema for multidimensional data models, concept hierarchies, measures: categorization and computations.

STOP!

Attention!

Some examples solved in video lectures are different from those given in this book.

The procedure to solve problems and examples is well explained in the video lectures, and it is highly recommended to go through the video lectures for complete understanding.

Official Video Playlist

Watch on YouTube

Chapter 1

The Data

1.1 Data Objects and Attribute Types

Data Object

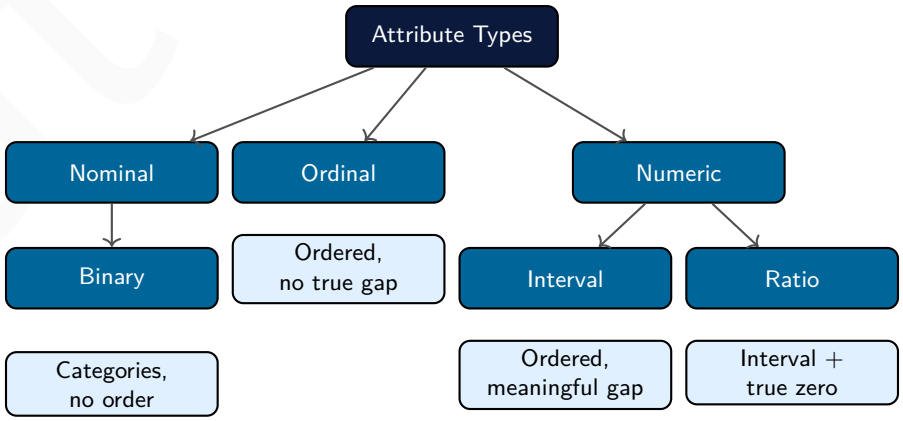
A **data object** represents an entity in a dataset.

- Also called: *samples, instances, data points, tuples*
- Each row in a database table → one data object
- Described by a set of **attributes**

Attribute

An **attribute** is a data field representing a characteristic of a data object.

- Also called: *feature, variable, dimension, field*
- Example: `Age`, `Name`, `Salary`, `Zip Code`



1.1.1 Nominal Attributes

Nominal Attribute

A **nominal attribute** takes values that are **names or labels** for categories.

- No ordering between values — cannot say one is greater or lesser
- No arithmetic meaningful — mean, sum have no sense
- Only valid operations: **equality** ($=$) and **inequality** ($\neq$)
- Mode is the only meaningful central tendency measure

Properties

- Values act as **labels only** — even if coded as numbers (e.g. 1=red, 2=blue), arithmetic is meaningless
- Can be **multi-valued**: one object may belong to multiple categories
- Useful statistics: **frequency count**, **mode**, **contingency tables**

Example 1: Hair Color

Attribute: `HairColor` **Domain:** {Black, Brown, Blonde, Red, Gray}

- Person A $\rightarrow$ Black Person B $\rightarrow$ Blonde Person C $\rightarrow$ Red
- We can say: A $\neq$ B
- We **cannot** say: Black $>$ Blonde, or Average(Black, Red) $=$?
- Most frequent value (mode) $\rightarrow$ valid summary statistic

Example 2: Profession / Occupation

Attribute: `Occupation` **Domain:** {Doctor, Engineer, Teacher, Lawyer, Artist}

- Encoded internally as: Doctor=1, Engineer=2, Teacher=3, Lawyer=4, Artist=5
- Even though stored as numbers, $1 + 2 = 3$ (Doctor + Engineer = Teacher?)
- Only meaningful question: "How many people are Engineers?" $\rightarrow$ frequency count
- Ordering like Engineer $>$ Doctor is **completely invalid**

Example 3: Zip / Pin Code

Attribute: `PinCode` Example values: {110001, 560001, 400001}

- Looks numeric — but arithmetic is meaningless
- $560001 - 110001 = 450000$ has **no geographic meaning**
- Used only to **identify a region** (label), not to measure anything
- Correct use: grouping, frequency, equality checks only

1.1.2 Binary Attributes

Binary Attribute

A **binary attribute** is a special case of nominal with exactly **two possible states**: 0 and 1 (or True/False, Yes/No).

- **Symmetric binary** — both outcomes carry equal weight/importance
- **Asymmetric binary** — one outcome is more significant (usually coded as 1)

Symmetric vs Asymmetric

- **Symmetric:** Neither state is more important
  Example: `Gender` — Male and Female are equally weighted
- **Asymmetric:** State 1 (positive) is **rare and more important**
  Example: `Disease Diagnosis` — Positive is rare, but critical
- For **similarity measures**, asymmetric binary attributes ignore **0-0 matches** (both negative = uninformative)

Example 4: Medical Test — Asymmetric

Attribute: `HIV_Test` States: {Positive=1, Negative=0}

- Most patients test Negative (0) — this is the common, uninformative state
- A Positive (1) result is **rare and highly significant**
- When comparing two patients, a **1-1 match** (both positive) is very meaningful
- A **0-0 match** (both negative) is **ignored** in similarity computation — it adds no information
- Jaccard coefficient used instead of simple matching coefficient

Example 5: Gender — Symmetric

Attribute: `Gender` States: {Male=1, Female=0} (or vice versa)

- Both states are **equally important** — no state is rarer or more critical
- A 0-0 match (both Female) is just as meaningful as a 1-1 match (both Male)
- Simple matching coefficient is appropriate for similarity
- Recoding Male=0, Female=1 does **not** change the analysis

Example 6: Purchase Decision — Asymmetric

Attribute: `Bought_Item` States: {Yes=1, No=0}

- In a large transaction database, most entries are 0 (item not bought)
- A **1-1 co-occurrence** (two customers both buying the same item) → meaningful pattern
- **0-0 co-occurrence** (both did not buy) → uninformative, ignored
- Used heavily in **market basket analysis** and association rule mining

1.1.3 Ordinal Attributes

Ordinal Attribute

An **ordinal attribute** has values with a **meaningful rank order**, but the **magnitude of difference** between successive values is unknown.

- Valid operations: $=$, $\neq$, $<$, $>$, $\leq$, $\geq$
- **Cannot** compute meaningful differences: $A - B$ has no fixed meaning
- Central tendency: **median** and **mode** are valid; mean is **not**

Ranking & Normalization of Ordinal Values

If an ordinal attribute has M states (ordered 1 to M), values can be mapped to $[0, 1]$:

$$z_i = \frac{r_i - 1}{M - 1}, \quad r_i \in \{1, 2, \ldots, M\}$$

This allows ordinal data to be used in certain numeric computations (e.g., distance measures).

Example 7: Academic Grades

Attribute: `Grade` **Order:** F $<$ D $<$ C $<$ B $<$ A

- Student X: B Student Y: A Student Z: C
- We know: A $>$ B $>$ C
- We **cannot** say: the gap between A and B equals the gap between B and C
- Median grade of {B, A, C, B, D} = **B** (after sorting: D, C, B, B, A)
- Mean of {B, A, C} = ? **undefined and meaningless**

Example 8: Customer Satisfaction Survey

Attribute: `Satisfaction` **Order:** Very Dissatisfied $<$ Dissatisfied $<$ Neutral $<$ Satisfied $<$ Very Satisfied

- $M = 5$ states, mapped: Very Dissatisfied=1, ..., Very Satisfied=5
- Normalized: $z = \{0,\ 0.25,\ 0.5,\ 0.75,\ 1.0\}$
- "Satisfied" is better than "Neutral" — but by **how much**? Unknown
- Report median satisfaction, **not** average — averaging ordinal labels is statistically incorrect

Example 9: Military / Job Rank

Attribute: `Rank` **Order:** Private $<$ Corporal $<$ Sergeant $<$ Lieutenant $<$ Captain

- Clear ordering of authority and responsibility
- Sergeant is **higher than** Corporal
- (Captain $-$ Lieutenant) $=$ (Sergeant $-$ Corporal)? **No such claim valid**
- Useful in sorting, ranking comparisons, and ordinal regression models

1.1.4 Numeric Attributes

Numeric Attribute

A **numeric attribute** is quantitative — represented by **measurable numbers** with meaningful arithmetic.

Two subtypes:

- **Interval-scaled** — ordered, meaningful differences, **no true zero**
- **Ratio-scaled** — interval + **true zero**, all arithmetic valid

Interval-Scaled Attributes

Interval-Scaled

- Measured on a scale with **equal-sized units**
- Differences are meaningful; ratios are **not**
- **No absolute zero** — zero is just another point on the scale
- Valid operations: $+, -$ Invalid: $\times, \div$ for ratio comparisons

Example 10: Temperature in Celsius

Attribute: `Temperature_C`

- $30°C - 20°C = 10°C$ difference is meaningful
- $40°C$ is **NOT** twice as hot as $20°C$
- Proof: $40°C = 104°F$, $20°C = 68°F$, $104/68 \neq 2$ — ratio changes with scale
- $0°C$ does not mean *absence of heat* — water still has thermal energy

Example 11: Calendar Year

Attribute: `Year`

- $2024 - 2020 = 4$ years difference
- Year 2024 is **not** "twice" as recent as Year 1012
- Year 0 is an **arbitrary reference**, not the start of time
- Differences (durations) are meaningful; ratios of years are not

Ratio-Scaled Attributes

Ratio-Scaled

- Has all properties of interval **plus a true zero**
- Zero means **complete absence** of the quantity
- **All arithmetic valid:** $+, -, \times, \div$
- Can make statements like "X is *twice* as much as Y"

Example 12: Weight

Attribute: `Weight_kg`

- Person A: 80 kg Person B: 40 kg
- A is **exactly twice** as heavy as B $80/40 = 2$
- 0 kg = truly no weight
- Difference: $80 - 40 = 40$ kg meaningful
- Mean, variance, standard deviation all valid

Example 13: Salary

Attribute: `Annual_Salary`

- Salary of \$0 = **no income** (true zero)
- \$80,000 is **four times** \$20,000
- Meaningful to compute: average salary, salary ratio, percentage increase
- Transformations (log scale) commonly applied for skewed salary distributions

Example 14: Age

Attribute: `Age_years`

- Age 0 = just born — true absence of elapsed life time
- Person aged 60 is **twice** as old as person aged 30
- Difference: $60 - 30 = 30$ years, fully meaningful
- Mean age, age ratio — all statistically valid operations

1.1.5 Discrete versus Continuous Attributes

Discrete Attribute

A **discrete attribute** has a **finite** or **countably infinite** set of values.

- Values are **distinct, separate** — no value exists between two adjacent values
- Often represented as **integers**
- Nominal and ordinal attributes are **always discrete**
- Some numeric attributes are also discrete (e.g., count data)

Continuous Attribute

A **continuous attribute** can take any **real value** within a range.

- Infinite possible values between any two points

- Represented as **floating-point numbers** in practice
- Measurements (height, weight, temperature) are typically continuous
- In practice, **precision of instrument** limits the recorded values

Key Distinction

Property	Discrete	Continuous
Value set	Finite / countable	Uncountably infinite (real)
Between vals	No value exists in between	Infinite values exist
Represented	Integer	Float / double
Example	No. of children: 0,1,2,3	Height: 170.52, 170.521...

Example 15: Number of Items Purchased — Discrete

Attribute: `Items_Purchased` **Domain:** $\{0, 1, 2, 3, \ldots\}$

- A customer can buy 3 or 4 items — **never 3.5 items**
- Set is countably infinite — no upper bound, but each value is a whole number
- Represented as integer; bar charts and frequency tables are natural visualizations
- Mean is technically computable but result (e.g., 2.7 items) must be interpreted carefully

Example 16: Zip Code / Pin Code — Discrete

Attribute: `PinCode` Example: {110001, 560001, 400001}

- Finite set of valid pin codes across the country
- No pin code exists *between* 110001 and 110002 — they are adjacent, distinct labels
- Although stored as numbers, this is a **nominal discrete** attribute
- Frequency analysis by pin code region is a valid operation

Example 17: Height — Continuous

Attribute: `Height_cm` Example: 170.5, 171.2, 165.87

- Between 170 cm and 171 cm, **infinitely many** real values exist: 170.1, 170.11, 170.111...
- Measured by instrument — precision limited (e.g., 0.1 cm), but the attribute itself is continuous
- Histograms (not bar charts) are the appropriate visualization
- All statistical operations valid: mean, variance, percentile, standard deviation

Example 18: Temperature — Continuous

Attribute: `Temperature_C`

- Can be 36.6, 36.61, 36.612... — infinite precision possible
- Thermometer records to 1 decimal, but true value is continuous
- In data preprocessing: often **discretized** into bins
  e.g., $[< 20°C$ = Cold], $[20–35°C$ = Warm], $[> 35°C$ = Hot]
- Discretization converts continuous → ordinal for certain algorithms

1.2 Basic Statistical Descriptions of Data

Why Statistical Descriptions?

- Helps understand the **distribution** and **spread** of data
- Identifies **outliers**, **noise**, and **skewness** early
- Guides preprocessing decisions: normalization, binning, transformation

1.2.1 Measuring the Central Tendency: Mean, Median, and Mode

Central Tendency

A **central tendency** measure gives a single value that best represents an entire dataset.
Three primary measures:

- **Mean** — arithmetic average
- **Median** — middle value when sorted
- **Mode** — most frequently occurring value

Mean

Arithmetic Mean

For N values $x_1, x_2, \ldots, x_N$:

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i$$

- Most common measure of central tendency
- **Sensitive to outliers** — a single extreme value pulls the mean strongly
- Valid only for **interval** and **ratio** attributes

Weighted Mean

When values have different importance (weights w_i):

$$\bar{x}_w = \frac{\sum_{i=1}^{N} w_i x_i}{\sum_{i=1}^{N} w_i}$$

Used when some observations are more reliable or frequent than others.

Trimmed Mean

- Remove a fixed percentage of extreme values (top and bottom) before computing mean
- Reduces effect of outliers
- Example: 5% trimmed mean — discard lowest 5% and highest 5% of values

Example 19: Arithmetic Mean — Exam Scores

Scores of 7 students: {52, 60, 65, 70, 72, 80, 95}

$$\bar{x} = \frac{52 + 60 + 65 + 70 + 72 + 80 + 95}{7} = \frac{494}{7} \approx 70.57$$

- If one student scored 200 (data error), mean becomes ≈ 99.14 — heavily distorted
- This shows mean's **sensitivity to outliers**

Example 20: Weighted Mean — Course Grade

Three components with weights:

Component	Score	Weight
Assignments	80	0.20
Midterm	70	0.30
Final Exam	90	0.50

$$\bar{x}_w = \frac{(80 \times 0.20) + (70 \times 0.30) + (90 \times 0.50)}{0.20 + 0.30 + 0.50} = \frac{16 + 21 + 45}{1} = 82$$

Simple mean would give $\frac{80+70+90}{3} = 80$ — ignoring the higher importance of the final exam.

Median

Median

The **median** is the **middle value** of a sorted dataset.

$$\text{Median} = \begin{cases} x_{\left(\frac{N+1}{2}\right)} & \text{if } N \text{ is odd} \\ \dfrac{x_{\left(\frac{N}{2}\right)} + x_{\left(\frac{N}{2}+1\right)}}{2} & \text{if } N \text{ is even} \end{cases}$$

- **Not affected by outliers** — robust measure
- Preferred over mean when data is **skewed**
- Valid for **ordinal, interval,** and **ratio** attributes

Median for Grouped Data

When data is in frequency classes, median is estimated as:

$$\text{Median} = L_1 + \left(\frac{\frac{N}{2} - \sum f_l}{f_{\text{median}}} \right) \times \text{width}$$

where L_1 = lower boundary of median class, $\sum f_l$ = frequencies before median class, f_{median} = frequency of median class, width = class interval width.

Example 21: Median — Odd Count

Salaries (in thousands): $\{30, 45, 50, 60, 72, 85, 120\}$ $N = 7$

- Sorted (already sorted above)
- Middle position: $\frac{7+1}{2} = 4$th value
- Median $= 60$
- Note: even if the outlier 120 becomes 5000, median **remains 60**

Example 22: Median — Even Count

House prices (lakhs): $\{25, 40, 55, 70, 90, 130\}$ $N = 6$

- Positions $\frac{6}{2} = 3$rd and 4th values: 55 and 70
- Median $= \dfrac{55 + 70}{2} = 62.5$ lakhs
- Mean $= \dfrac{25 + 40 + 55 + 70 + 90 + 130}{6} = 68.3$ lakhs — pulled up by 130
- Median better represents the "typical" house price here

Mode

Mode

The **mode** is the value that occurs **most frequently** in a dataset.

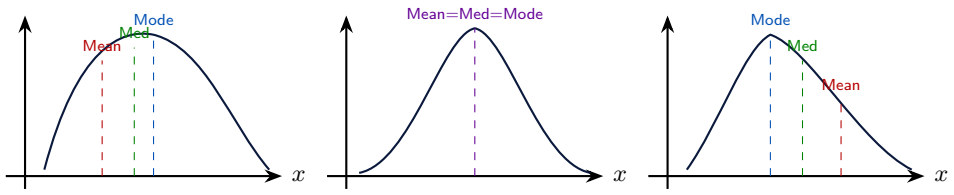
- A dataset can be **unimodal** (one mode), **bimodal** (two modes), or **multimodal**
- Valid for **all attribute types** including nominal
- For continuous data: mode is estimated from the most frequent class interval

Empirical Relation Mean Median Mode

For **moderately skewed** unimodal distributions:

$$\text{Mean} - \text{Mode} \approx 3 \times (\text{Mean} - \text{Median})$$

- **Symmetric distribution:** Mean = Median = Mode
- **Positively skewed (right):** Mode < Median < Mean
- **Negatively skewed (left):** Mean < Median < Mode



Negatively Skewed Symmetric Positively Skewed

Example 23: Mode — Shoe Sizes

Sizes sold in a day: {6, 7, 7, 8, 8, 8, 9, 9, 10}

- Size 8 appears 3 times — most frequent
- Mode $= 8$ (unimodal)
- A store manager stocks **more of size 8** based on mode — not mean or median
- Mode is the **only** valid central tendency for nominal data

Example 24: Bimodal — Age of Gym Members

Ages: {18, 19, 20, 20, 21, 35, 36, 36, 37, 38}

- Mode $= 20$ and 36 — **bimodal distribution**
- Reveals two distinct groups: college students and working professionals
- Mean ≈ 27.9 — falls in a gap between both groups, **misrepresenting** either group
- Bimodal patterns suggest the data may need to be **split into subgroups**

1.2.2 Measuring the Dispersion of Data

Dispersion

Dispersion measures how **spread out** the data values are around the central tendency.

- Low dispersion $\rightarrow$ values clustered tightly around the mean
- High dispersion $\rightarrow$ values scattered widely
- Key measures: **Range, Quartiles, IQR, Variance, Standard Deviation**

Range

Range

$$\text{Range} = x_{\max} - x_{\min}$$

- Simplest measure of spread
- Extremely **sensitive to outliers** — one bad value distorts it completely
- Gives no information about how values are distributed within the range

Example 25: Range — Monthly Rainfall

Rainfall (mm): {10, 25, 40, 55, 70, 120}

$$\text{Range} = 120 - 10 = 110 \text{ mm}$$

- If 120 is an outlier (measurement error), range is completely misleading
- IQR and standard deviation give a more reliable picture

Quartiles and IQR

Quartiles

Quartiles divide a **sorted** dataset into four equal parts:

- Q_1 (25th percentile) — 25% of data lies below this value
- Q_2 (50th percentile) — the **median**
- Q_3 (75th percentile) — 75% of data lies below this value

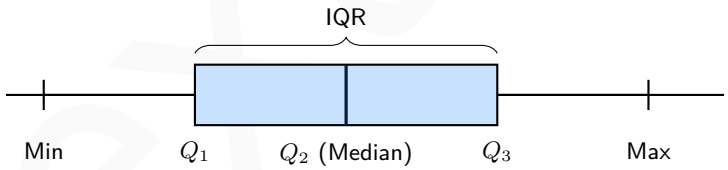
$$\text{IQR} = Q_3 - Q_1$$

Outlier Detection using IQR

A value is considered an **outlier** if it falls outside the fences:

$$\text{Lower fence} = Q_1 - 1.5 \times \text{IQR} \qquad \text{Upper fence} = Q_3 + 1.5 \times \text{IQR}$$

Any value beyond these fences is flagged as a potential outlier.



Example 26: Quartiles and IQR — Test Scores

Scores (sorted): {42, 55, 60, 65, 70, 75, 80, 88, 95} $N = 9$

- Q_2 = 5th value $= 70$
- Q_1 = median of lower half {42, 55, 60, 65} $= \dfrac{60 + 55}{2} = 57.5$
- Q_3 = median of upper half {75, 80, 88, 95} $= \dfrac{80 + 88}{2} = 84$
- IQR $= 84 - 57.5 = 26.5$
- Lower fence $= 57.5 - 1.5(26.5) = 17.75$ Upper fence $= 84 + 1.5(26.5) = 123.75$
- All values within fences — no outliers in this dataset

Example 27: Outlier Detection — Employee Salaries

Salaries (thousands): {30, 35, 38, 40, 42, 45, 48, 50, 200}

- $Q_1 = 38, \quad Q_3 = 48, \quad \text{IQR} = 10$
- Upper fence $= 48 + 1.5 \times 10 = 63$
- Salary of 200 exceeds 63 — flagged as **outlier**
- Without IQR, range $= 170$ makes the spread look enormous; IQR $= 10$ reveals true spread

Variance and Standard Deviation

Variance and Standard Deviation

Variance measures the average squared deviation from the mean:

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2$$

Standard Deviation is the square root of variance — in the **same units** as the data:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2}$$

- Low $\sigma \rightarrow$ values clustered near mean
- High $\sigma \rightarrow$ values spread widely
- **Sensitive to outliers** — squaring amplifies extreme deviations

Sample vs Population

- **Population variance:** divide by N
- **Sample variance:** divide by $N - 1$ (Bessel's correction — avoids underestimation)

$$s^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2$$

- In data mining, population formula is commonly used when entire dataset is available

Example 28: Variance — Daily Temperatures

Temperatures (°C): {20, 22, 24, 26, 28} $\quad \bar{x} = 24$

x_i	$x_i - \bar{x}$	$(x_i - \bar{x})^2$
20	-4	16
22	-2	4
24	0	0
26	2	4
28	4	16

$$\sigma^2 = \frac{16+4+0+4+16}{5} = \frac{40}{5} = 8 \qquad \sigma = \sqrt{8} \approx 2.83°C$$

Temperatures deviate on average by $2.83°C$ from the mean of $24°C$.

Example 29: Comparing Spread — Two Batsmen

Batsman A scores: $\{45, 48, 50, 52, 55\}$ $\quad \bar{x} = 50$
Batsman B scores: $\{10, 30, 50, 70, 90\}$ $\quad \bar{x} = 50$

- Both have the same mean: 50
- $\sigma_A = \sqrt{\frac{(-5)^2+(-2)^2+0^2+2^2+5^2}{5}} = \sqrt{11.6} \approx 3.4$
- $\sigma_B = \sqrt{\frac{(-40)^2+(-20)^2+0^2+20^2+40^2}{5}} = \sqrt{1200} \approx 34.6$
- Batsman A is **consistent** (low σ); Batsman B is **unpredictable** (high σ)
- Mean alone is **completely insufficient** to compare performance

1.3 Problems

Problem 1 *Which of the following is the most appropriate attribute type for* `Employee ID`*?*

(A) Ordinal

(B) Ratio

(C) Nominal

(D) Interval

Problem 2 *A dataset contains the attribute* `Temperature in Fahrenheit`*. Which type best describes it?*

(A) Ratio, because temperature can be zero

(B) Nominal, because units are arbitrary

(C) Interval, because differences are meaningful but zero is not absolute

(D) Ordinal, because temperatures can be ranked

Problem 3 *Which of the following attributes has a* ***true zero****?*

(A) Calendar Year

(B) Temperature in Celsius

(C) IQ Score

(D) Number of items sold

Problem 4 *Pin codes (e.g., 110001, 560001) are best classified as:*

(A) Ratio — because they are stored as numbers

(B) Ordinal — because higher pin codes are in different regions

(C) Nominal — because they serve only as region labels

(D) Continuous — because infinitely many values are possible

Problem 5 *For a* ***symmetric binary*** *attribute, which statement is correct?*

(A) The value 1 is always more important than 0

(B) Both states carry equal importance in similarity computation

(C) 0–0 matches are ignored in similarity measures

(D) It is a special case of ordinal attribute

Problem 6 *A doctor records whether a patient tested positive or negative for a rare disease. This attribute is best classified as:*

(A) Symmetric binary

(B) Asymmetric binary

(C) Ordinal

(D) Nominal with more than two values

Problem 7 *The* ***empirical relation*** *between mean, median, and mode for a moderately skewed distribution is:*

(A) $\text{Mode} - \text{Mean} = 3(\text{Mean} - \text{Median})$

(B) $\text{Mean} - \text{Mode} = 2(\text{Mean} - \text{Median})$

(C) $\textit{Mean} - \textit{Mode} = 3(\textit{Mean} - \textit{Median})$

(D) $\textit{Median} - \textit{Mode} = 3(\textit{Mean} - \textit{Median})$

Problem 8 *For the dataset* $\{3, 7, 7, 9, 12, 15, 15, 15, 20\}$*, the mode is:*

(A) 7

(B) 12

(C) 15

(D) The dataset is bimodal: 7 and 15

Problem 9 *Which measure of central tendency is* ***least affected*** *by outliers?*

(A) Arithmetic Mean

(B) Weighted Mean

(C) Median

(D) Trimmed Mean with 0% trimming

Problem 10 *Student grades* $\{A, B, C, D, F\}$ *are an example of which attribute type?*

(A) Nominal — labels with no order

(B) Binary — only pass or fail

(C) Ordinal — ranked but differences unknown

(D) Ratio — zero grade means no performance

Problem 11 *Which of the following is an example of a* ***continuous*** *attribute?*

(A) Number of login attempts

(B) Zip code

(C) Body temperature in Celsius

(D) Number of siblings

Problem 12 *The* ***Interquartile Range (IQR)*** *for the dataset* $\{5, 10, 15, 20, 25, 30, 35, 40\}$ *is:*

(A) 20

(B) 25

(C) 22.5

(D) 15

Problem 13 *A dataset has* $\bar{x} = 50$*, Median* $= 55$*, Mode* $= 60$*. The distribution is:*

(A) Symmetric

(B) Positively skewed (right skewed)

(C) Negatively skewed (left skewed)

(D) Bimodal

Problem 14 *Which operations are valid for an* ***ordinal*** *attribute?*

(A) Addition and subtraction only

(B) Equality and inequality only

(C) *Equality, inequality, and ordering ($<$, $>$)*

(D) *All arithmetic operations including multiplication*

Problem 15 *For which attribute type is the **mode the only valid** measure of central tendency?*

(A) *Ratio*

(B) *Interval*

(C) *Ordinal*

(D) *Nominal*

Problem 16 *An attribute records the **number of cars** owned by each household. It is:*

(A) *Continuous and ratio*

(B) *Discrete and ratio*

(C) *Discrete and ordinal*

(D) *Continuous and interval*

Problem 17 *Two attributes: `Blood Type` $\in$ {A, B, AB, O} and `Pain Level` $\in$ {None, Mild, Moderate, Severe}. Which pair correctly classifies them?*

(A) *Both nominal*

(B) *Nominal and ordinal respectively*

(C) *Ordinal and nominal respectively*

(D) *Both ordinal*

Problem 18 *For the data {10, 20, 30, 40, 200}, which measure changes **most dramatically** if 200 is replaced by 2000?*

(A) *Median*

(B) *Mode*

(C) *Mean*

(D) Q_1

Problem 19 *The **upper fence** for outlier detection using IQR is:*

(A) $Q_3 + 2 \times IQR$

(B) $Q_3 + 1.5 \times IQR$

(C) $Q_1 - 1.5 \times IQR$

(D) $Q_3 + 3 \times IQR$

Problem 20 ***[MSQ]*** *Which of the following attributes are **nominal**? Select all that apply.*

(A) *Marital status*

(B) *Military rank*

(C) *Nationality*

(D) *Eye colour*

Problem 21 ***[MSQ]*** *Which of the following are properties of a **ratio-scaled** attribute?*

(A) *Differences between values are meaningful*

(B) There exists a true zero point

(C) Ratios of values are meaningful

(D) Values can only be positive

Problem 22 ***[MSQ]*** *For a **positively skewed** distribution, which of the following hold?*

(A) Mean $>$ Median

(B) Mode $<$ Median $<$ Mean

(C) Median $>$ Mean

(D) The tail extends to the right

Problem 23 ***[MSQ]*** *Which central tendency measures are valid for an **ordinal** attribute?*

(A) Mean

(B) Median

(C) Mode

(D) Weighted Mean

Problem 24 ***[NAT]*** *The mean of 40 observations is 20. It is later discovered that one observation was recorded as 36 instead of the correct value 26. The corrected mean is _____.*

Problem 25 ***[NAT]*** *A dataset of 60 values has mean 15. A second dataset of 40 values has mean 20. The mean of the combined dataset of 100 values is _____.*

Problem 26 ***[NAT]*** *The variance of 10 observations is 4. Each observation is multiplied by 3. The variance of the new dataset is _____.*

Problem 27 ***[NAT]*** *For a distribution, the mean is 45 and the mode is 30. Assuming the distribution is moderately skewed, the median is _____ (rounded to 2 decimal places).*

Problem 28 ***[NAT]*** *A dataset has $n = 8$ values with $\sum x_i = 64$ and $\sum x_i^2 = 560$. The standard deviation of the dataset is _____.*

Problem 29 ***[NAT]*** *The mean of n observations is $\bar{x}$. If each observation is increased by 5, the new mean becomes 30. If instead each observation is doubled, the new mean becomes 50. The value of $\bar{x}$ is _____.*

Problem 30 ***[NAT]*** *A dataset of 9 observations sorted in ascending order is:*
$\{x$, 12, 18, 22, 25, 28, 33, 40, 50$\}$
The median is 25 and the mean is 26. The value of x is _____.

Problem 31 ***[MSQ]*** *Q_1 and Q_3 of a dataset are 18 and 42 respectively. A new value of 75 is added to the dataset. Using the original IQR,select that apply _____.*

(A) 75 is an outlier

(B) 75 is not an outlier

(C) The upper fence is 78

(D) The upper fence is -18

Problem 32 ***[NAT]*** *The mean and variance of 5 observations are 8 and 7.9 respectively. If three of the five observations are 3, 7, and 11, and the remaining two observations are equal, the value of each of the remaining two observations is _____.*

1.4 Try It Yourself

Exercise 1 ***[MCQ]*** *Which of the following correctly classifies the attribute* `Body Mass Index (BMI)`*?*

(A) Nominal, because BMI is a label

(B) Ordinal, because BMI can be ranked

(C) Interval, because BMI differences are meaningful but no true zero exists

(D) Ratio, because BMI has a true zero and all arithmetic is valid

Exercise 2 ***[MCQ]*** *A researcher encodes* `Gender` *as Male = 1, Female = 2, and computes the mean as 1.6. What is the most appropriate critique?*

(A) The mean is valid since the values are numeric

(B) The mean is meaningless because Gender is nominal and arithmetic does not apply

(C) The mean should be replaced by weighted mean

(D) The encoding should use 0 and 1 instead of 1 and 2

Exercise 3 ***[MCQ]*** *A dataset stores customer* `Satisfaction` *as* $\{$*Poor, Average, Good, Excellent*$\}$*. A data analyst wants to compute the average satisfaction. Which statement is correct?*

(A) Mean is valid since satisfaction can be numerically encoded

(B) Median is the appropriate measure; mean is not valid for ordinal data

(C) Mode is the only valid measure for this attribute

(D) Both mean and median are equally valid

Exercise 4 ***[MCQ]*** *For the dataset* $\{2, 4, 4, 6, 8, 10, 10, 10, 14\}$*, which statement about central tendency is correct?*

(A) Mean = Median = Mode, symmetric distribution

(B) Mode = 10, which is greater than the mean, suggesting left skew

(C) Mode = 4 and the distribution is positively skewed

(D) Mean = Mode, hence the distribution is symmetric

Exercise 5 ***[MCQ]*** *An attribute records the* `Year of Birth` *of employees. Which type is most appropriate?*

(A) Ratio — because year zero is a true absence of time

(B) Ordinal — because years can be ordered

(C) Interval — because differences are meaningful but year zero is not a true zero

(D) Nominal — because years are just labels

Exercise 6 ***[MCQ]*** *A hospital uses the attribute* `Cancer_Detected`*: Yes = 1, No = 0. When computing patient similarity, 0–0 matches (both cancer-free) are ignored. This is because:*

(A) The attribute is symmetric binary

(B) The Jaccard coefficient requires ignoring all zero values

(C) The attribute is asymmetric binary and negative matches are uninformative

(D) The attribute should be re-encoded as ordinal

Exercise 7 ***[MCQ]*** *Which of the following datasets has the* ***highest variance****?*

(A) $\{10, 10, 10, 10, 10\}$

(B) $\{8, 9, 10, 11, 12\}$

(C) $\{1, 5, 10, 15, 19\}$

(D) $\{9, 10, 10, 10, 11\}$

Exercise 8 ***[MCQ]*** *The **IQR** is preferred over the **range** as a measure of dispersion because:*

(A) IQR is always larger than range

(B) IQR uses all data values in its computation

(C) IQR is not affected by extreme values and outliers

(D) IQR equals standard deviation for symmetric distributions

Exercise 9 ***[MCQ]*** *For a dataset with $Q_1 = 30$, $Q_3 = 60$, and IQR $= 30$, which value is flagged as an outlier?*

(A) 85

(B) 15

(C) 105

(D) 20

Exercise 10 ***[MCQ]*** *A temperature sensor records values to the nearest $0.5°C$. The attribute `Temperature` is:*

(A) Discrete, because the sensor only outputs a finite set of values

(B) Continuous, because the underlying phenomenon is continuous even if precision is limited

(C) Ordinal, because temperatures can be ranked

(D) Nominal, because temperature labels are arbitrarily assigned

Exercise 11 ***[MSQ]*** *Which of the following attributes would be classified as **ratio-scaled**? Select all that apply.*

(A) Annual income in rupees

(B) Temperature in Kelvin

(C) Calendar year

(D) Distance in kilometres

Exercise 12 ***[MSQ]*** *A dataset has attributes: `Age`, `ZipCode`, `PainLevel (Low/Med/High)`, `Weight_kg`, `Smoker (Yes/No)`.*
*Which of the following are **discrete** attributes?*

(A) Age (in completed years)

(B) ZipCode

(C) Weight_kg

(D) Smoker

Exercise 13 ***[MSQ]*** *Which of the following are valid operations for an **interval-scaled** attribute?*

(A) Computing the difference between two values

(B) Computing the ratio of two values

(C) Ordering values from smallest to largest

(D) Computing the mean

Exercise 14 ***[MSQ]*** *For a* ***negatively skewed*** *distribution, which of the following are true?*

(A) The tail extends to the left

(B) Mean $<$ Median $<$ Mode

(C) Mode $<$ Median $<$ Mean

(D) The median is greater than the mean

Exercise 15 ***[MSQ]*** *Which of the following changes will* ***not affect the median*** *of a dataset?*

(A) Doubling the largest value

(B) Adding a constant to every value

(C) Replacing the smallest value with an extremely small number

(D) Removing a value that is equal to the current median (odd N)

Exercise 16 ***[NAT]*** *For the dataset $\{12, 15, 20, 22, 25, 30, 35, 40, 45, 50\}$, compute the* ***IQR****.*

Exercise 17 ***[NAT]*** *The ages of 10 participants are $\{18, 21, 22, 22, 23, 25, 26, 28, 30, 75\}$.*
Compute the ***mean*** *(round to 2 decimal places).*

Exercise 18 ***[NAT]*** *Given Mean $= 60$ and Mode $= 72$ for a moderately skewed distribution.*
Using the empirical relation, compute the ***Median****.*

Exercise 19 ***[NAT]*** *A student scores $\{55, 60, 65, 70, 75\}$ in five tests.*
Compute the ***population standard deviation*** *σ (round to 2 decimal places).*

Exercise 20 ***[NAT]*** *From the frequency table below, compute the* ***mean****:*

Value	*Frequency*
10	*3*
20	*5*
30	*8*
40	*2*
50	*2*

Exercise 21 ***[NAT]*** *For an ordinal attribute with $M = 7$ states, a respondent is at state 5.*
Compute the ***normalized value*** $z = \dfrac{r-1}{M-1}$.

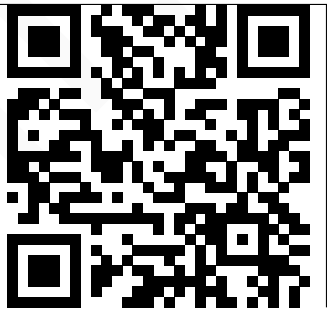
Exercise 22 ***[NAT]*** *Salaries (in thousands): $\{28, 32, 35, 38, 40, 42, 45, 48, 52, 300\}$.*
$Q_1 = 35$, $Q_3 = 48$. Is 300 an outlier by the ***IQR rule****?*
Compute the upper fence to justify. (Enter the upper fence value.)

Exercise 23 ***[NAT]*** *A dataset of $N = 100$ values has $\sum x_i = 5000$ and $\sum x_i^2 = 260000$.*
Compute the ***population variance*** *using* $\sigma^2 = \dfrac{\sum x_i^2}{N} - \bar{x}^2$.

Exercise 24 ***[NAT]*** *Two professors grade students:*
Professor A: $\{70, 71, 72, 73, 74\}$ Professor B: $\{50, 60, 72, 84, 94\}$
Both have mean $= 72$. Compute the ***population standard deviation*** *for Professor B's class (round to 2 decimal places).*

Exercise 25 ***[NAT]*** *A population variance $\sigma^2 = 36$ is computed for $N = 10$ values.*
If $\sum(x_i - \bar{x})^2$ increases by 24, what is the ***new population variance****?*

1.5 YouTube Links and QR Codes

Lecture	Details	YouTube Link	QR Code
01	Introduction to Data Warehousing — Attribute Types (Nominal, Ordinal, Numeric)	https://youtu.be/QLlfO08X2xo	
02	Statistical Description of Data — Data Warehousing	https://youtu.be/G-ttDpu6QlM	
03	Problem Solving on Attribute Types & Statistical Description — Data Warehousing	https://youtu.be/7R-hZ1EfLXM	

Chapter 2

Data Preprocessing

Data Preprocessing

Data Preprocessing is a crucial phase in the **Knowledge Discovery Process (KDD)** that involves transforming raw, "dirty" data into a clean, consistent format suitable for data mining and analysis.

Why Preprocess?

Real-world data is typically:

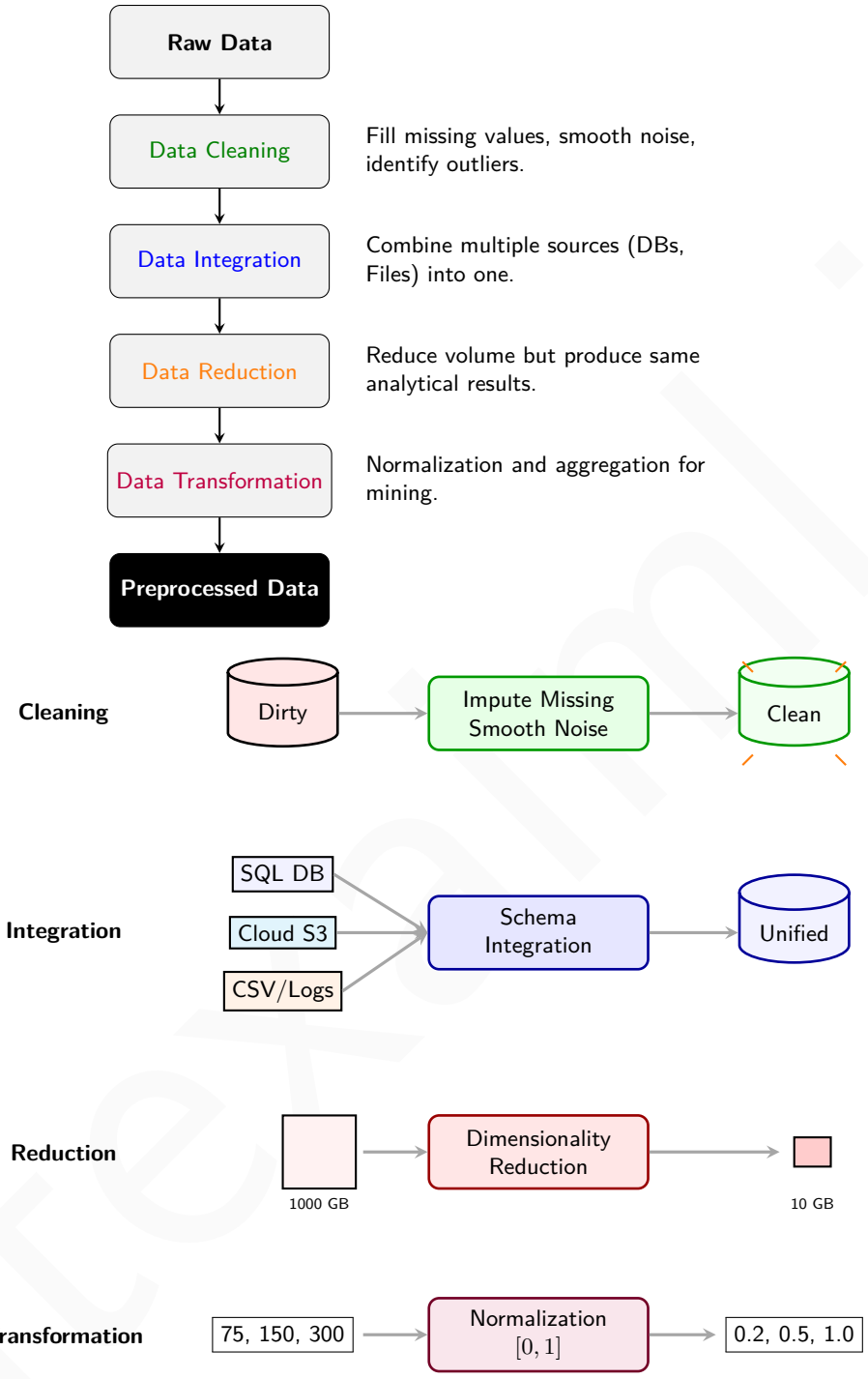
- **Incomplete:** Lacking attribute values or certain attributes of interest.
- **Noisy:** Containing errors, outliers, or random variance.
- **Inconsistent:** Discrepancies in codes or names (e.g., "Age=42" vs "Birthday=01/01/2010").

Impact: Quality decisions depend on quality data. Preprocessing improves accuracy, efficiency, and leads to significant payoffs in decision-making.

Example 30: The "Dirty Data" Scenario

Consider a database for a retail chain:

- Missing Values: A customer record with no "Email" field.
- Inconsistency: One branch records currency in USD, another in INR.
- Noise: A "Salary" field recorded as -100.



Core Tasks Breakdown

1. **Cleaning:** "Cleaning" the data by removing noise and correcting inconsistencies.
2. **Integration:** Merging data from multiple data stores.
3. **Reduction:** Obtaining a reduced representation of the data set that is much smaller in volume, yet yields the same (or almost the same) analytical results.
4. **Transformation:** Normalizing data (e.g., scaling values between 0.0 and 1.0) to ensure features contribute equally.

2.1 Data Cleaning

2.1.1 Handling Missing Values

The Challenge of "Null" Data

In real-world datasets like *AllElectronics*, tuples often lack recorded values for attributes such as *Customer Income*. Missing data can lead to biased results or algorithm failure.

Methods for Handling Missing Values

1. **Ignore the Tuple:**
   - Best used when the **class label** is missing.
   - Drawback: Highly ineffective if the missing rate is high; wastes other useful information in the same row.
2. **Manual Filling:**
   - Human intervention to find the correct value.
   - Drawback: Scalability. Impossible for Big Data.
3. **Global Constant:**
   - Replace all blanks with a label like `"Unknown"` or `-1`.
   - Drawback: The algorithm may wrongly cluster all "Unknowns" as a meaningful group.
4. **Measure of Central Tendency (Global):**
   - Use the **Mean** for symmetric (Normal) distributions.
   - Use the **Median** for skewed (offset) distributions.
5. **Class-Based Central Tendency:**
   - Instead of a global average, use the average of the specific group the tuple belongs to (e.g., mean income of "High Risk" customers only).
6. **Most Probable Value (Predictive):**
   - Use **Regression** or **Decision Trees** to guess the value based on other attributes.
   - **Note:** This is the most popular strategy as it uses all available information.

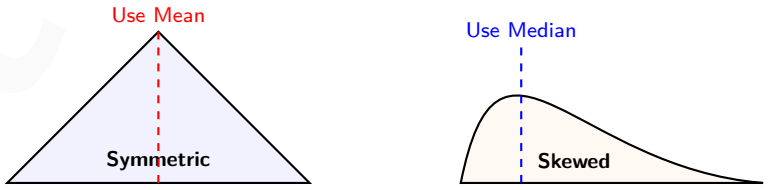


Figure: Choosing Imputation Strategy based on Data Distribution

Example 31: Example: Income Imputation

Suppose you have a missing income for a customer who is a **"Software Engineer"**.

- **Global Mean:** 56,000 (Uses everyone from interns to CEOs).
- **Class-based Mean:** 95,000 (Uses only the mean of other "Software Engineers").
- **Regression:** 98,500 (Predicts value based on Education, Age, and Location).

When Missing is Not an Error

Sometimes a `NULL` is intentional.

- **Not Applicable:** A "Driver's License" field left blank by someone without a license.
- **Business Process:** Fields to be filled in a later stage of a workflow.

Pro-Tip: Good database design with specific "Null Rules" is better than any post-capture cleaning!

2.1.2 Noisy Data: Binning, Regression, and Clustering

What is Noise?

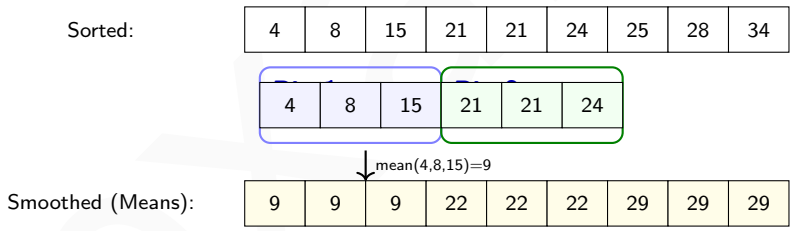
Noise is a **random error** or variance in a measured variable. While outliers can sometimes be valid extreme data points, noise is typically incorrect data that needs "smoothing" to reveal the underlying pattern.

1. Binning (Local Smoothing)

The Neighborhood Concept

Binning methods smooth sorted data by consulting its **neighborhood** (the values around it).

- **Step 1:** Sort the data.
- **Step 2:** Partition into bins (Equal-frequency or Equal-width).
- **Step 3:** Apply smoothing logic (Means, Medians, or Boundaries).



Example 32: Smoothing by Bin Boundaries

In **Bin 1: [4, 8, 15]**, the boundaries are 4 and 15.

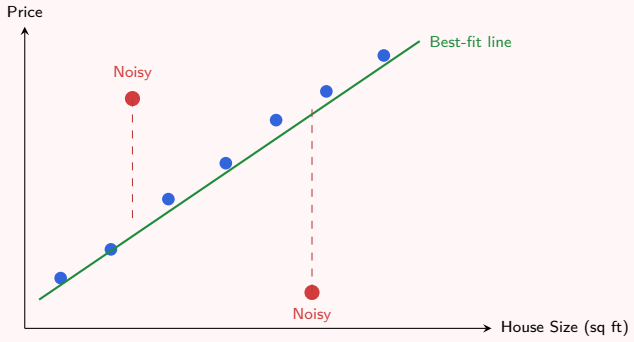
- 8 is closer to 4 than to 15.
- **Resulting Bin:** [4, 4, 15].

2. Regression & Outlier Analysis

Regression for Noise Smoothing

Fit a **mathematical function** to the data — points that deviate far from the function are noise.

- **Linear Regression:** fits a straight line $\hat{y} = \beta_0 + \beta_1 x$
- Noisy points are **smoothed** — replaced by their predicted value on the line
- The line captures the **true trend**; scatter around it is treated as noise



Noisy points lie far from the line → flagged & smoothed

Example 33: Exam Score vs Study Hours

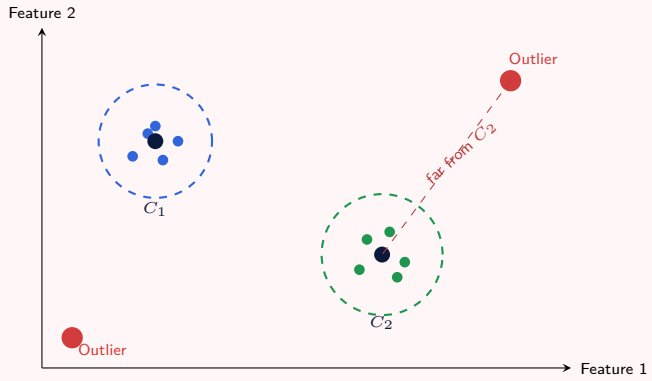
A teacher records study hours and exam scores for 30 students.
One student studied 2 hours but scored 95 (unusually high — maybe copied).
Another studied 8 hours but scored 20 (unusually low — maybe sick).

- Regression fits a line: Score $= 40 + 6 \times$ Hours
- Both students deviate **far from the line** → flagged as noisy records
- Options: **smooth** them to their predicted value, or **remove** them entirely

Outlier Analysis via Clustering

Group data into clusters — points that **do not belong to any cluster** are outliers (noise).

- Similar records form tight clusters
- Points **far from all cluster centres** → flagged as anomalies
- No mathematical model needed — purely distance-based



Example 34: Bank Transactions

A bank has **1 million transaction records**.

- Clustering groups them into normal spending clusters:
  Groceries cluster, *Rent cluster*, *Entertainment cluster*
- A transaction of **$50,000 at 3 AM** falls far outside every cluster
- ⇒ Flagged as an **outlier** — potential fraud or data entry error

Key insight: No rule was written manually — the data itself revealed what is "unusual."

Regression vs Clustering for Noise — Quick Contrast

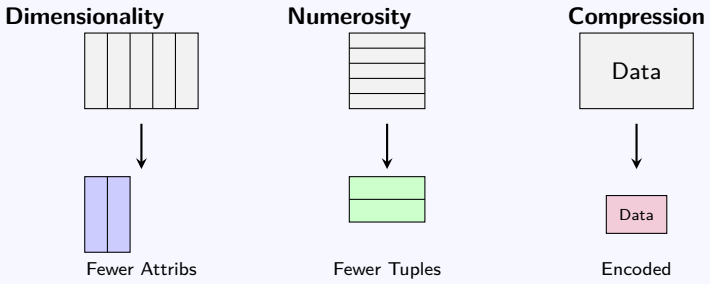
Aspect	Regression	Clustering
Detects noise via	Deviation from fit line	Distance from cluster
Needs	Input–output pairs	Only input features
Output	Smooth predicted value	Outlier flag
Best for	Continuous trends	Grouped / categorical data

2.2 Data Reduction

Why Data Reduction?

Large-scale datasets can take an enormous amount of time to process. **Data Reduction** techniques obtain a reduced representation of the data set that is much smaller in volume, yet yields the same (or almost the same) analytical results.

Major Strategies



1. **Dimensionality Reduction:** Removing irrelevant or redundant attributes (e.g., Wavelet transforms, PCA, Feature subset selection).
2. **Numerosity Reduction:** Replacing the original data with smaller forms of data representation.
   - **Parametric:** Storing only model parameters (e.g., Regression).
   - **Non-parametric:** Storing reduced samples (e.g., Histograms, Clustering, Sampling).
3. **Data Compression:** Applying transformations (lossy or lossless) to reduce the file size.

2.2.1 Dimensionality Reduction

Dimensionality reduction

Dimensionality reduction is the process of reducing the number of random variables, attributes, or features under consideration in a dataset. This is typically achieved through two primary approaches:

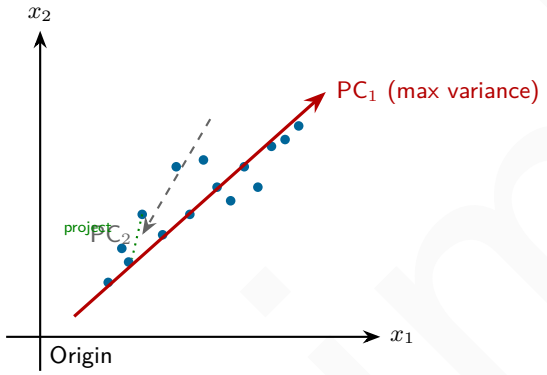
- **Principal Components Analysis (PCA):** A linear method that transforms or projects the original data onto a smaller space, preserving as much variance as possible.
- **Attribute Subset Selection:** A method that identifies and removes irrelevant, weakly relevant, or redundant attributes or dimensions from the dataset.

2.2.1.1 Principal Component Analysis

Principal Component Analysis (PCA)

Finds a **lower-dimensional** representation that captures maximum variance.

- Computes **principal components** — orthogonal directions of maximum variance
- Projects data onto the top k components, discarding the rest
- Original n dimensions reduced to $k \ll n$ dimensions
- Information loss is controlled by choosing k such that retained variance is high (e.g., 95%)



Example 35: PCA for Dimension Reduction

A student dataset has 10 attributes:
`Math, Physics, Chemistry, Biology, History, Geography, English, CS, Economics, PE`

- PCA applied $\rightarrow$ finds that the first 2 principal components explain 88% of total variance
- Data projected from 10 dimensions $\rightarrow$ 2 dimensions
- PC$_1$ may represent "Science aptitude"; PC$_2$ may represent "Humanities aptitude"
- Remaining 8 components discarded — small information loss, major storage gain
- Clustering and classification now run on 2 features instead of 10

2.2.1.2 Feature Subset Selection

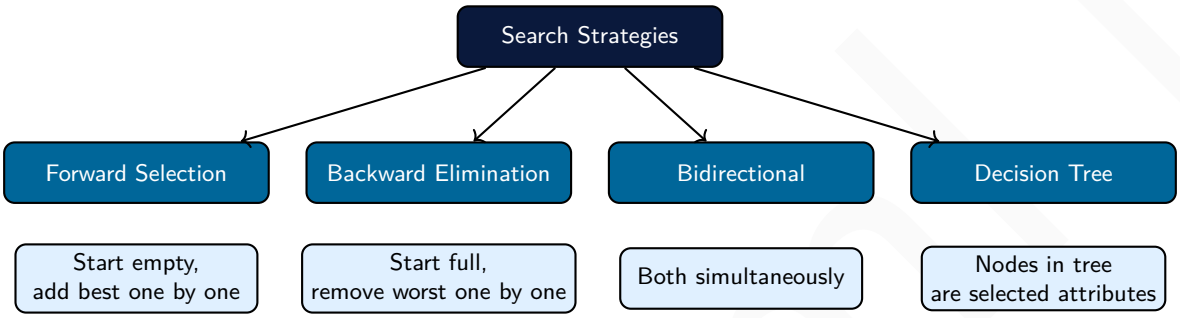
Attribute Subset Selection

Attribute subset selection reduces the dataset by retaining only the **relevant attributes** and removing redundant or irrelevant ones.

- Also called **feature selection**
- Irrelevant attributes: no useful information for the mining task
- Redundant attributes: information already present in another attribute
- Goal: find the **minimum set of attributes** that best represents the data

Why Attribute Subset Selection?

- Reduces **dimensionality** — fewer attributes to process
- Speeds up **learning algorithms** — less computation
- Improves **model accuracy** — removes noise from irrelevant features
- Avoids the **curse of dimensionality** — too many features degrade performance
- Produces **simpler, more interpretable** models



Forward Selection

- Begin with an **empty set** of attributes
- At each step, add the attribute that **most improves** model performance
- Stop when adding more attributes no longer improves performance significantly
- Efficient when the **number of relevant attributes is small**

Example 36: Forward Selection

Dataset has attributes: {Age, Income, ZipCode, Gender, Purchase_History, WebClicks}
Task: Predict whether a customer will buy.

- Step 1: Test each alone → `Purchase_History` gives best accuracy → selected
- Step 2: Add one more → `Income` improves accuracy → selected
- Step 3: `Age` adds marginal gain → selected
- Step 4: `ZipCode`, `Gender`, `WebClicks` add no improvement → stopped
- Final subset: {`Purchase_History, Income, Age`}

Backward Elimination

- Begin with the **full set** of attributes
- At each step, remove the attribute whose **removal least affects** performance
- Stop when removing any remaining attribute causes a significant drop
- Better when **most attributes are relevant**

Example 37: Backward Elimination

Same dataset: {Age, Income, ZipCode, Gender, Purchase_History, WebClicks}

- Step 1: Remove `ZipCode` → accuracy unchanged → eliminated
- Step 2: Remove `Gender` → accuracy unchanged → eliminated
- Step 3: Remove `WebClicks` → slight drop, but acceptable → eliminated
- Step 4: Removing any of {`Age`, `Income`, `Purchase_History`} causes significant drop → stop
- Final subset: {`Age`, `Income`, `Purchase_History`} — same result as forward selection

Decision Tree Induction for Attribute Selection

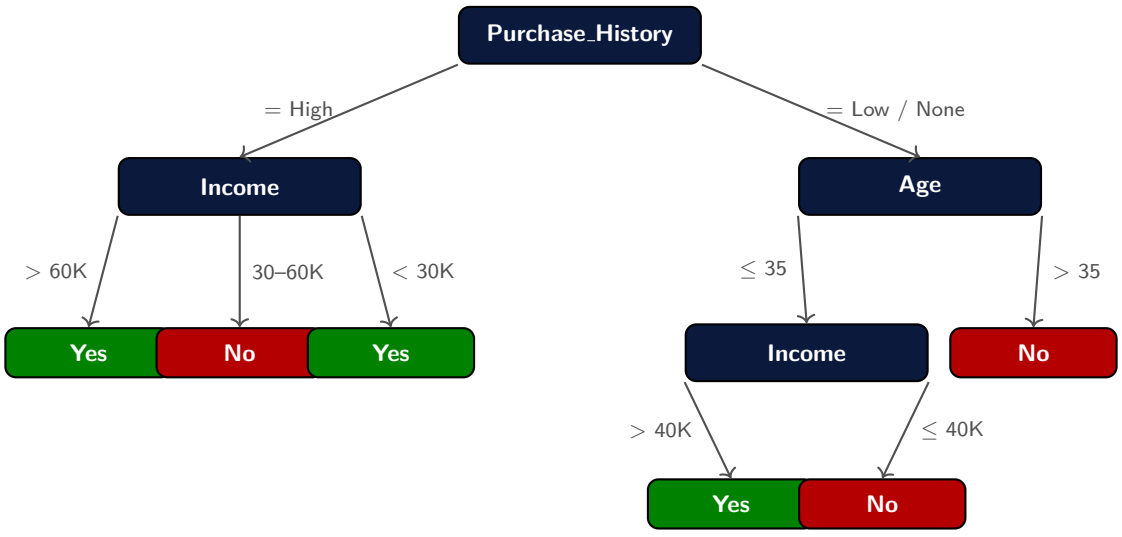
A **decision tree** can be used to identify relevant attributes:

- Build a decision tree on the training data
- Attributes that **appear as nodes** in the tree are selected
- Attributes **not used** in any split are considered irrelevant
- Efficient — no separate search process needed

Example 38: Decision Tree for Customer Purchase Prediction

Dataset attributes: `Age`, `Income`, `ZipCode`, `Gender`, `Purchase_History`, `WebClicks`
Target: Will the customer buy? (Yes / No)

After building the decision tree on training data:



Attributes SELECTED (appear in tree):
`Purchase_History`, `Income`, `Age`

Attributes DISCARDED (never split on):
`ZipCode`, `Gender`, `WebClicks`

- **Root split:** `Purchase_History` — most informative attribute; splits data most cleanly
- **Level 1:** `Income` (left branch) and `Age` (right branch) — next most relevant

- **Level 2:** `Income` appears again under `Age` — still relevant in that context
- **Final selected subset:** {`Purchase_History`, `Income`, `Age`} — 3 of 6 attributes
- `ZipCode`, `Gender`, `WebClicks` were **never chosen** for any split — discarded as irrelevant

Attribute Selection Measures

Criteria used to evaluate which attributes to keep or discard:

- **Information Gain** — reduction in entropy after splitting on an attribute
- **Gain Ratio** — normalizes information gain to avoid bias toward many-valued attributes
- **Gini Index** — measures impurity of a split
- **Correlation** — removes attributes highly correlated with others (redundancy)

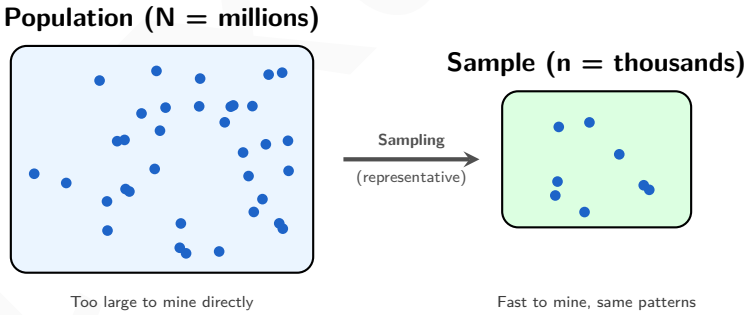
2.2.2 Numerosity Reduction

2.2.2.1 Sampling

Sampling — Core Idea

Sampling is a **numerosity reduction** technique. It allows us to represent a massive dataset using a much smaller random subset.

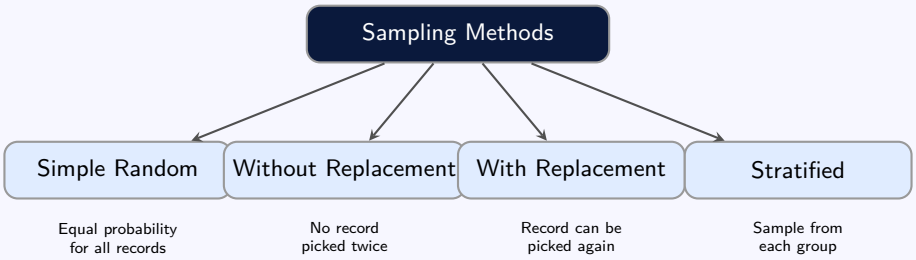
- If the sample is **representative**, patterns found in the small set hold true for the entire population
- Key assumption: *the distribution of the sample* $\approx$ *distribution of the population*
- Goal: minimize data processed while maximizing accuracy of results



Why Sampling Works — The Representative Principle

- A sample is **representative** if it has approximately the same property (distribution) as the original population
- **Sample size matters:** larger $n \Rightarrow$ lower sampling error, but higher cost
- **Randomness matters:** without it, the sample becomes biased

Common Sampling Methods — Overview



Simple Random Sampling (SRS)

Every record in the dataset has an **equal and independent** probability of being selected.

$$P(\text{any record selected}) = \frac{n}{N}$$

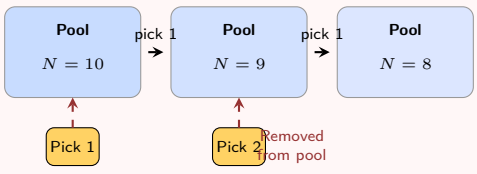
where n = sample size, N = population size.

- Simplest and most unbiased method
- Works best when population is **homogeneous** (similar records)
- May miss rare subgroups in heterogeneous data

Sampling Without Replacement (SRSWOR)

Once a record is selected, it is **removed from the pool** — it cannot be selected again.

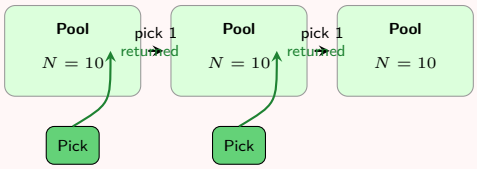
- Each draw changes the pool size: $N \to N-1 \to N-2 \ldots$
- Probability of selection increases slightly each round
- **More commonly used** in practice — avoids redundancy
- Every selected record is *unique*



Sampling With Replacement (SRSWR)

After a record is selected, it is **put back** into the pool — it may be selected again.

- Pool size stays constant at N every draw
- Same record can appear **multiple times** in the sample
- Used in **bootstrapping** — a powerful statistical technique
- Probability of any record $= \frac{1}{N}$ always



Example 39: The Logic of With vs. Without Replacement

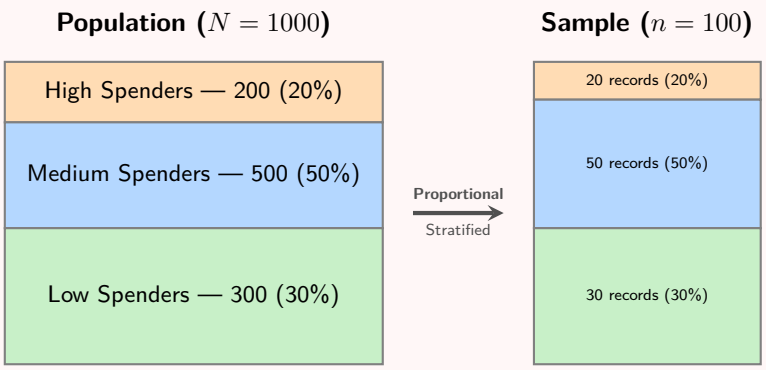
Imagine a bag with **10 coloured balls**. You want to pick 3.

- **SRSWR:** Pick a **Red** ball → record it → **put it back**.
  Next turn, the same Red ball can be picked again. Bag always has 10 balls.
- **SRSWOR:** Pick a **Red** ball → **keep it out**.
  Now only 9 balls remain. That Red ball is gone forever from the pool.

Stratified Sampling

The population is first divided into **non-overlapping groups (strata)**, then a random sample is drawn from **each stratum**.

- Ensures all subgroups are **proportionally represented**
- Reduces risk of missing rare but important subgroups
- Two variants:
  - **Proportional:** sample size from each stratum $\propto$ stratum size
  - **Equal:** same number drawn from every stratum (regardless of size)



Example 40: Stratified Proportional Sampling — University

A university has **1,000 students** across three streams:

Stream	Students	Proportion	Pick from 100
Engineering	600	60%	$100 \times 0.60 = \mathbf{60}$
Arts	300	30%	$100 \times 0.30 = \mathbf{30}$
Science	100	10%	$100 \times 0.10 = \mathbf{10}$
Total	**1000**	**100%**	**100**

Result: The 100-student sample looks exactly like the full university — same proportions, smaller size.

Proportional vs Equal Stratified — When to Use Which

Proportional
Mirrors population ratios

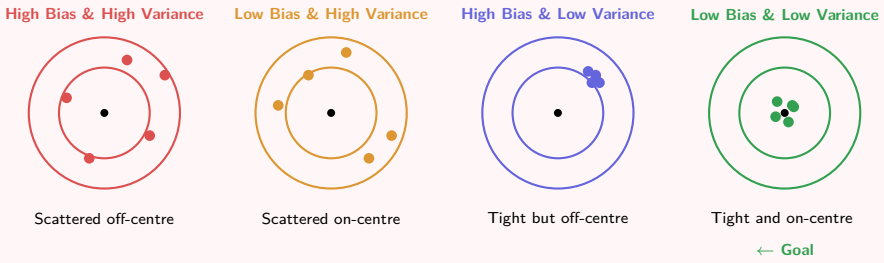
Use when: majority class is important to represent correctly

Equal
Same n from each stratum

Use when: minority class must not be drowned out

Sampling Error — Bias vs Variance

- **Sampling error** = difference between sample statistic and true population value
- Two sources of error:
  - **Bias** — sample is not random; some records are systematically favoured
  - **Variance** — sample size too small; results fluctuate a lot
- Both reduce as sampling is done more carefully and with larger n



Example 41: The Soup Pot — Sampling Intuition

You want to know if a giant pot of soup needs more salt. You taste 5 spoonfuls.

- **Scenario A — Good Sample (Stirred):**
  Stir the pot thoroughly first, then take spoons from different depths and sides.
  $\Rightarrow$ The 5 spoonfuls represent the whole pot accurately.
- **Scenario B — Biased Sample (Unstirred):**
  Take all 5 spoons from the top where salt has settled.
  $\Rightarrow$ You conclude the soup is too salty — but the bottom is bland.
  This is **sampling bias**.
- **Scenario C — Too Small a Sample:**
  Taste only 1 spoon from a 100-litre pot.
  $\Rightarrow$ High variance — one spoon may not be representative at all.

Lesson: A good sample must be both **random (stirred)** and **large enough (5+ spoons)**.

Comparison — All Sampling Methods

Method	Key Idea	Best For	Risk
SRS	Equal probability	Homogeneous data	May miss rare groups
SRSWOR	No duplicates	General purpose	Slightly complex prob.
SRSWR	Duplicates allowed	Bootstrapping	Redundant records
Stratified	Sample per group	Heterogeneous data	Need known strata

2.2.2.2 Clustering & Regression

Numerosity Reduction via Clustering

Instead of storing all N records, **replace each cluster of similar records with just its centroid** (center point).

- Data is grouped into clusters of similar objects
- Only the **cluster representatives** (centroids) are stored — not the raw data
- Massive reduction: N records $\rightarrow$ k centroids $\quad (k \ll N)$
- Works well when data has **natural groupings**

Example 42: City Zones Instead of Every House

A dataset has the location (lat, long) of **10,000 houses** in a city.

- Clustering groups them into **5 city zones** (North, South, East, West, Centre)
- Each zone is represented by its **centroid** — the average location
- Analysis now runs on **5 points** instead of 10,000
- Patterns like "crime is high in the East zone" are still discoverable

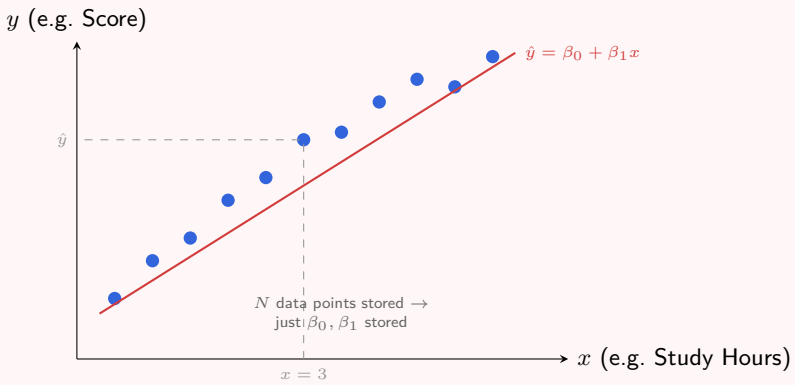
Trade-off: Fine-grained detail within a zone is lost — only the zone-level pattern survives.

Numerosity Reduction via Regression

Regression fits a mathematical model to the data — then **the model replaces the data**.

- **Linear Regression:** fits a straight line $y = \beta_0 + \beta_1 x$
- Store only **2 parameters** (β_0, β_1) instead of N data points
- **Multiple Regression:** multiple predictors $\Rightarrow$ store $p + 1$ parameters

- Any future y value is estimated from the model — raw data no longer needed



Example 43: Predicting House Prices

A dataset has **50,000 house sale records** with (area → price).

- Fit a regression: Price $= 5000 + 3200 \times$ Area (sq ft)
- Now store just **two numbers**: $\beta_0 = 5000$, $\beta_1 = 3200$
- For any new house: plug in area, get predicted price instantly
- 50,000 rows replaced by **one equation**

Trade-off: Works only if the relationship is truly linear — nonlinear patterns need more complex models.

Clustering vs Regression — Quick Contrast

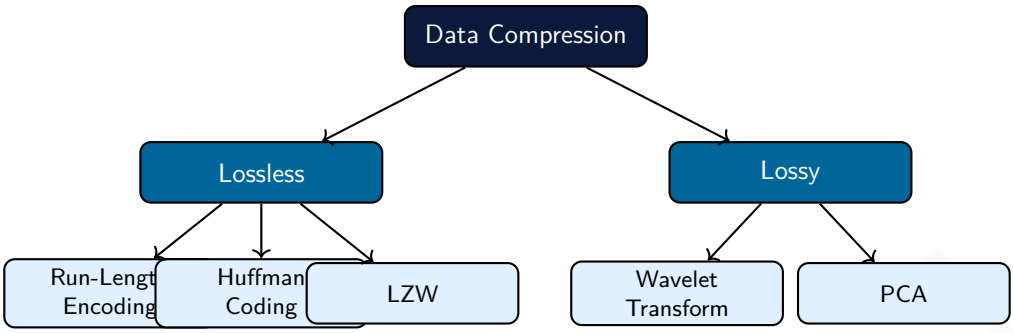
Aspect	Clustering	Regression
Replaces data with	Cluster centroids	Model parameters (βs)
Data type	Any (no target variable)	Needs input–output pairs
Storage	k centroids	$p + 1$ coefficients
Assumption	Natural groupings exist	Linear (or known) relationship

2.2.3 Data Compression

Data Compression

Data compression reduces the size of data by encoding it in fewer bits without losing (or with controlled loss of) information.

- Reduces **storage space** and **transmission cost**
- Two major types: **Lossless** and **Lossy**
- In data mining, compression is a key **dimensionality reduction** strategy



2.2.3.1 Lossless Compression

Lossless Compression

Lossless compression reduces data size such that the *original data can be reconstructed exactly*.

- No information is discarded — decompressed output $\equiv$ original input bit-for-bit.
- Exploits **statistical redundancy**: repeated symbols, predictable patterns, skewed frequencies.
- Theoretical lower bound: **Shannon entropy** $H = -\sum_i p_i \log_2 p_i$ bits/symbol.
- Applications: text files, source code, database columns, lossless image formats (PNG), archiving.

Run-Length Encoding (RLE)

Run-Length Encoding

RLE replaces consecutive runs of the same symbol with a *(count, symbol)* pair.

- **Encoding rule:** scan left to right; emit $(\text{count}, \text{symbol})$ for each maximal run.
- **Decoding rule:** for each pair $(\text{count}, \text{symbol})$, repeat symbol count times.
- **Best case:** long uniform runs (binary bitmaps, fax images, sparse columns).
- **Worst case:** alternating symbols — output *larger* than input (e.g. `ABAB` $\rightarrow$ `1A1B1A1B`).
- **Complexity:** $O(n)$ encode and decode.

Diagram 1 — RLE Encoding Pipeline

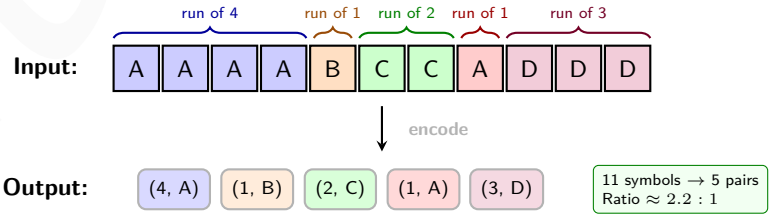
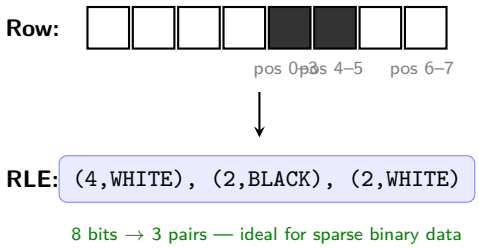


Diagram 2 — RLE on Binary Bitmap (Fax / Sparse Column)



Example 44: RLE on a Database Bitmap Index

Scenario: A column `Gender` with values `M/F` sorted by gender.

Raw bitmap for `M`: `1 1 1 1 1 0 0 0 0 0 0 0` (5 M's then 7 F's)

Representation	Storage	Notes
Raw bitmap	12 bits	one bit per row
RLE encoded	`(5,1)(7,0)` — 2 pairs	$6\times$ fewer units
WAH compressed	word-aligned RLE	standard in Oracle, DB2

Decode: `(5,1)` → `11111`, `(7,0)` → `0000000` — original bitmap restored exactly.

Example 45: RLE Worst Case — Alternating Input

Input: `A B A B A B A B` (8 symbols)
RLE output: `(1,A)(1,B)(1,A)(1,B)(1,A)(1,B)(1,A)(1,B)` — 16 units

Observation: Output is **twice** the size of input.
Mitigation: Use a *literal run* escape: sequences of run-length 1 are grouped as a raw literal block instead of individual `(1,x)` pairs — used in PackBits (Apple) and PCX formats.

Example 46: RLE on OLAP Column Store — Sorted Dimension Key

Scenario: `FACT_SALES` sorted by `store_key`. Store 1 has 50 000 rows, store 2 has 30 000 rows.

```
-- Raw column (80,000 integers):
1,1,1,...,1,  2,2,2,...,2
-- RLE encoded (2 pairs):
(50000, 1), (30000, 2)
```

Compression ratio: $80\,000 \div 2 = 40\,000\times$ reduction for this column.
Bonus: Aggregate queries (`COUNT(*) WHERE store_key = 1`) answered *directly from the RLE pairs* without decompressing.

Example 47: RLE Decode Trace

Compressed stream: `(3,X)(1,Y)(4,Z)(2,X)`

Step-by-step decode:

Step	Pair	Action	Output so far
1	(3,X)	emit X three times	`X X X`
2	(1,Y)	emit Y once	`X X X Y`
3	(4,Z)	emit Z four times	`X X X Y Z Z Z Z`
4	(2,X)	emit X twice	`X X X Y Z Z Z Z X X`

Restored string: `XXXYZZZZZXX` — identical to original.

Huffman Coding

Huffman Coding

Huffman Coding is an *optimal prefix-free* variable-length code where:

- **Frequent symbols** receive *shorter* codewords.
- **Rare symbols** receive *longer* codewords.
- **Prefix-free property:** no codeword is a prefix of another — enables unambiguous decoding.
- **Optimality:** produces the shortest possible average code length for a given symbol probability distribution (proven optimal among all prefix-free codes).
- Average code length: $\bar{L} = \sum_i p_i \ell_i \geq H$ (Shannon entropy lower bound).

Huffman Tree Construction Algorithm

Input: symbol frequency table. **Output:** binary tree defining the code.

1. Create a **leaf node** for each symbol; insert into a min-priority queue keyed by frequency.
2. While queue size > 1:
   a) Extract two nodes with **minimum frequency** f_1, f_2.
   b) Create an **internal node** with frequency $f_1 + f_2$; left child $= f_1$, right child $= f_2$.
   c) Insert internal node back into queue.
3. Remaining node is the **root**.
4. Assign **0** to every left edge, **1** to every right edge.
5. Codeword for symbol s = path of bits from root to leaf s.

Complexity: $O(n \log n)$ using a heap.

Diagram 3 — Huffman Tree Construction

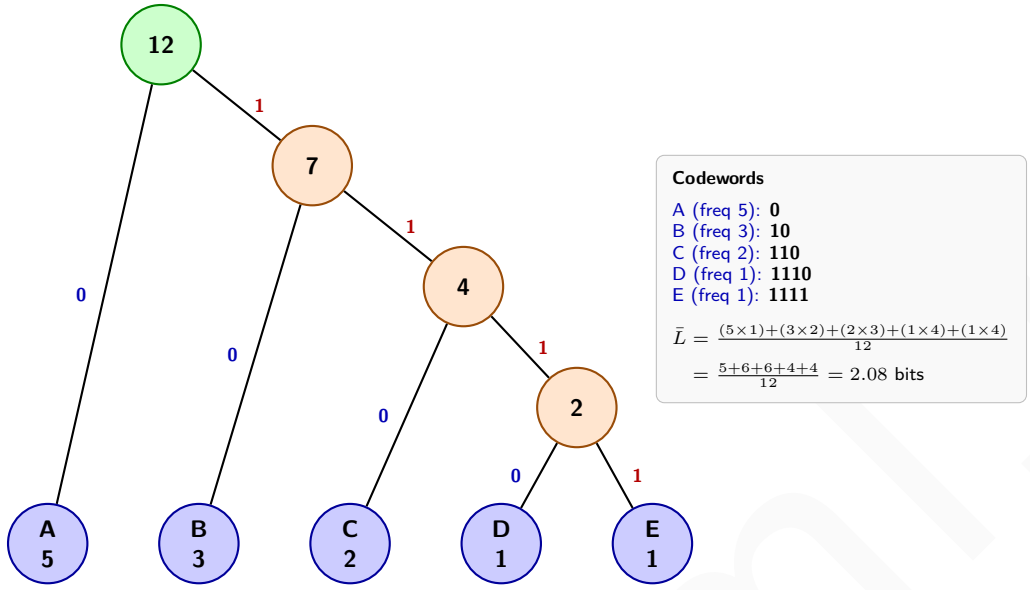
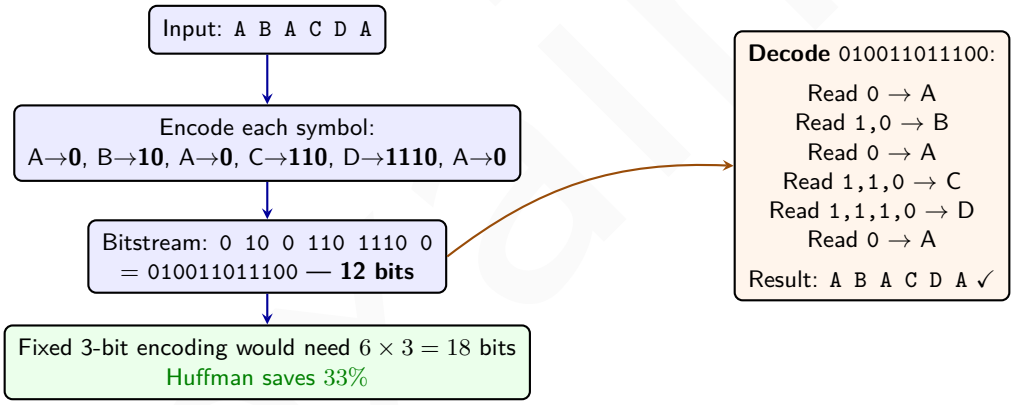


Diagram 4 — Encode and Decode Pipeline



Example 48: Huffman for English Text Compression

Approximate English letter frequencies (top 5):

Symbol	Freq (%)	Huffman bits	Fixed (5-bit ASCII)
E	12.7	2	5
T	9.1	3	5
A	8.2	3	5
O	7.5	3	5
Z	0.07	8	5

Effect: Common letters cost 2–3 bits; rare letters cost 7–8 bits. Weighted average ≈ 4.2 bits vs. 5-bit fixed — ∼16% compression on English text.

Example 49: Canonical Huffman — Used in DEFLATE / ZIP

Problem: Huffman tree must be transmitted with the compressed data — expensive.

Canonical Huffman: Fix codewords by two rules only:

1. Shorter codes have numerically smaller values.
2. Codes of the same length are consecutive integers.

Transmit only: the list of codeword *lengths* per symbol (e.g. `[1,2,3,4,4]` for A,B,C,D,E).
Decoder reconstructs canonical codewords from lengths alone — no tree storage needed.

Used in: ZIP, gzip, PNG, HTTP/2 HPACK header compression.

Example 50: Adaptive Huffman — Online Streaming

Scenario: Compress a data stream where symbol frequencies are unknown in advance.

Algorithm (FGK / Vitter):

- Both encoder and decoder maintain identical Huffman trees, updated *after every symbol*.
- New symbols are emitted with an escape code + raw bits; tree restructured immediately.
- No separate frequency pre-scan or tree transmission required.

Trade-off: Slightly worse compression than static Huffman (tree adapts with delay); enables *single-pass streaming* compression.

Example 51: Huffman Suboptimality — Skewed Distribution

Input: Single symbol A with probability $p_A = 1.0$ (only one symbol exists).
Huffman output: codeword `0` (1 bit).
Shannon entropy: $H = -1 \cdot \log_2 1 = 0$ bits/symbol.

Observation: Huffman must assign at least 1 bit — cannot reach the 0-bit theoretical limit.
Solution: **Arithmetic coding** can approach entropy arbitrarily closely, even for skewed distributions — at the cost of higher computational complexity.

LZW (Lempel–Ziv–Welch)

LZW Compression

LZW is a *dictionary-based* lossless algorithm that:

- Builds a **dictionary** (string table) *on the fly* during encoding — no pre-scan needed.
- Replaces recurring substrings with **integer codes** (shorter than the substrings).
- Decoder reconstructs the *same dictionary* independently — dictionary need not be transmitted.
- Dictionary initialised with all single-symbol entries (e.g. codes 0–255 for ASCII).
- **Complexity:** $O(n)$ encode and decode.
- **Used in:** GIF images, TIFF (optional), PDF (FlateDecode predecessor), Unix `compress`.

LZW Encoding Algorithm

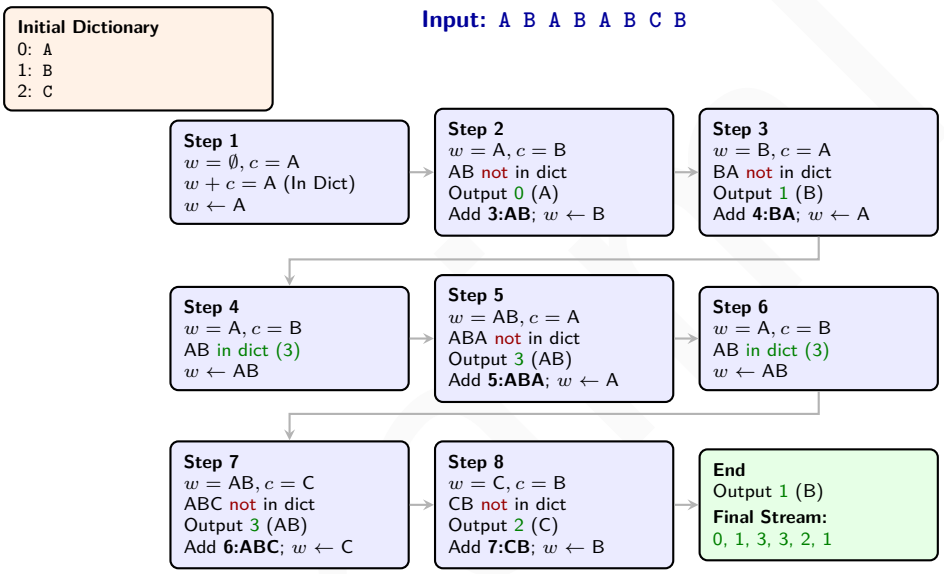
Initialise: dictionary $\leftarrow$ {each single symbol $\rightarrow$ code}.
Set: $w \leftarrow \varepsilon$ (empty string).

For each symbol c in input:

1. If $w + c$ **is in dictionary**: set $w \leftarrow w + c$ (extend current match).
2. Else:
   a) **Output** code for w.
   b) **Add** $w + c$ to dictionary with next available code.
   c) Set $w \leftarrow c$.

End: output code for w.

Diagram 5 — LZW Encoding Trace



Example 52: LZW Example

Input String:
BABAABAAA

Initial Dictionary:

Symbol	Code
A	0
B	1

LZW Encoding Trace:

Step	w	c	$w + c$ in dict?	Output	New Entry
1	B	A	No	1	2 : BA
2	A	B	No	0	3 : AB
3	B	A	Yes	-	-
4	BA	A	No	2	4 : BAA
5	A	B	Yes	-	-
6	AB	A	No	3	5 : ABA
7	A	A	No	0	6 : AA
8	A	A	Yes	-	-
9	AA	A	No	6	7 : AAA

Final Output Code Stream:

$$\boxed{1,\ 0,\ 2,\ 3,\ 0,\ 6}$$

Final Dictionary:

Code	Entry
0	A
1	B
2	BA
3	AB
4	BAA
5	ABA
6	AA
7	AAA

Example 53: LZW Example

Input String:
ABABABCB

Initial Dictionary:

Symbol	Code
A	0
B	1
C	2

LZW Encoding Trace:

Step	w	c	$w + c$ in dict?	Output	New Entry
1	A	B	No	0	3 : AB
2	B	A	No	1	4 : BA
3	A	B	Yes	-	-
4	AB	A	No	3	5 : ABA
5	A	B	Yes	-	-
6	AB	C	No	3	6 : ABC
7	C	B	No	2	7 : CB
8	B	-	-	1	-

Final Output Code Stream:

$$\boxed{0,\ 1,\ 3,\ 3,\ 2,\ 1}$$

Final Dictionary:

Code	Entry
0	A
1	B
2	C
3	AB
4	BA
5	ABA
6	ABC
7	CB

Example 54: LZW vs. RLE vs. Huffman — Comparison on Same Input

Input: `AAABBBCCDDDDAA` (14 symbols, alphabet {A,B,C,D})

Method	Encoded output	Size
Raw (2-bit fixed)	00 00 00 01 01 01 10 10 11 11 11 11 00 00	28 bits
RLE	(3,A)(3,B)(2,C)(4,D)(2,A)	5 pairs
Huffman	A→0, B→10, C→110, D→111	$3+6+6+12+2 = \mathbf{29}$ bits
LZW	codes: 0,0,0,1,1,1,2,2,3,3,7,0,...	improves with repeats

Insight:

- RLE wins on long uniform runs.
- Huffman wins when frequencies are skewed but runs are short.

- LZW wins on long files with recurring *substrings* (not just characters).
- Real compressors (gzip, bzip2) *chain* multiple methods.

2.2.3.2 Lossy Compression

Lossy Compression

Lossy compression permanently discards *perceptually insignificant* information to achieve much higher compression ratios than lossless methods allow.

- **Irreversible:** decompressed output $\neq$ original; reconstruction error is accepted.
- **Governed by:** a quality / distortion parameter — higher compression = more loss.
- **Core principle:** exploit **perceptual redundancy** — the human eye/ear is insensitive to fine high-frequency detail or subtle colour differences.
- **Applications:** images (JPEG), audio (MP3, AAC), video (H.264, H.265), scientific data cubes.
- **Error metrics:** PSNR (Peak Signal-to-Noise Ratio), SSIM (Structural Similarity Index).

Wavelet Transform (Overview)

- Decomposes a signal into **approximation** (low-frequency) and **detail** (high-frequency) components at multiple *resolution levels*.
- **Key idea:** most energy concentrates in low-frequency coefficients; high-frequency coefficients are small and can be **quantised or zeroed** with minimal perceptual loss.
- **Steps:**
  1. Apply wavelet transform (e.g. Haar, Daubechies) $\rightarrow$ coefficient array.
  2. **Quantise**: round small coefficients to zero (lossy step).
  3. **Entropy-code** the sparse coefficient array (lossless step).
- **Used in:** JPEG 2000, FBI fingerprint database, EEG/seismic signal compression.
- **Advantage over JPEG (DCT):** no block boundary artefacts; better at low bit-rates.

PCA — Principal Component Analysis (Overview)

- Projects high-dimensional data onto a **lower-dimensional subspace** defined by the directions of maximum variance (**principal components**).
- **Steps:**
  1. Centre data: subtract mean from each dimension.
  2. Compute **covariance matrix** Σ.
  3. Compute **eigenvectors** (principal components) and **eigenvalues** of Σ.
  4. Project data onto top-k eigenvectors (largest eigenvalues).
- **Lossy:** information in the dropped $(d - k)$ dimensions is permanently lost.
- **Lossless special case:** $k = d$ — no compression, full reconstruction.
- **Applications:** dimensionality reduction, feature extraction, face recognition (Eigenfaces), noise removal, data cube aggregation.

Diagram — PCA: 2-D to 1-D Projection

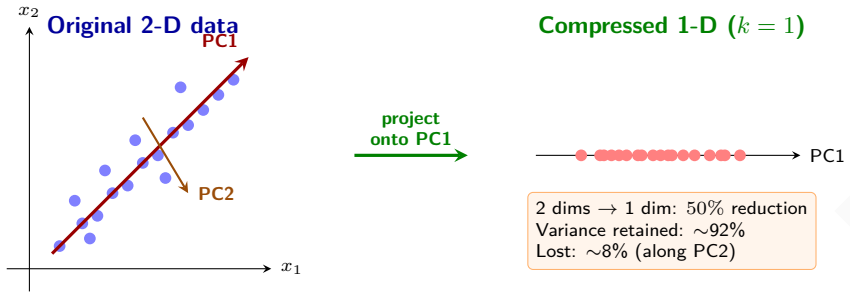
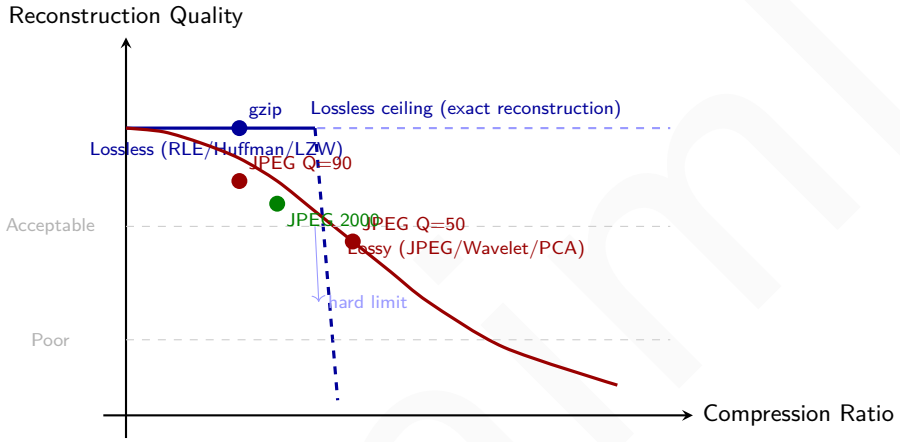


Diagram — Quality vs. Compression Ratio Trade-off



Wavelet vs. PCA — Summary Comparison

Property	Wavelet	PCA
Domain	Spatial / temporal signals	Tabular / feature data
Basis	Fixed (Haar, Daubechies)	Data-adaptive (eigenvectors)
Locality	Localised in time & freq	Global (whole dataset)
Computation	$O(n)$ fast wavelet transform	$O(d^2n + d^3)$ SVD
Optimality	Near-optimal for signals	Optimal linear reduction
Use in DM	Time-series, images	Feature reduction, noise removal
Reconstruction	Partial (drop detail)	Partial (drop components)

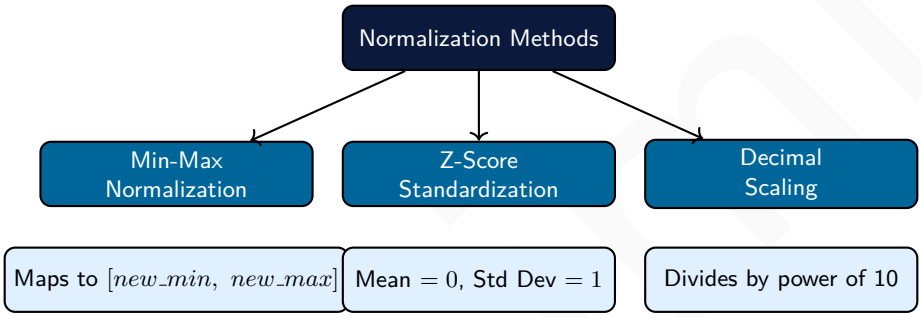
2.3 Data Transformation

2.3.1 Data Transformation by Normalization

Normalization

Normalization transforms numeric attribute values into a **specified range or distribution** to remove scale bias.

- Different attributes have different units and ranges
  Example: `Age` $\in [0, 100]$, `Salary` $\in [10000, 500000]$
- Distance-based algorithms (KNN, K-Means, SVM) are **biased toward higher-range attributes**
- Normalization ensures **equal contribution** from all attributes
- Three standard methods: **Min-Max**, **Z-Score**, **Decimal Scaling**



2.3.1.1 Min-Max Normalization

Min-Max Normalization

Maps original values linearly to a **target range** $[new\_min,\ new\_max]$:

$$v' = \frac{v - \min_A}{\max_A - \min_A} \times (new\_max - new\_min) + new\_min$$

- Most common target range: $[0,\ 1]$
- **Preserves relationships** between original values exactly
- **Sensitive to outliers** — one extreme value compresses all others
- If a future value falls **outside** $[\min_A, \max_A]$, result is outside $[0, 1]$

Special Case: Mapping to [0, 1]

When $new\_min = 0$ and $new\_max = 1$, the formula simplifies to:

$$v' = \frac{v - \min_A}{\max_A - \min_A}$$

Example 55: Min-Max: Basic Mapping to [0,1]

Attribute: `Age` with values {20, 30, 40, 50, 60}
$\min_A = 20, \quad \max_A = 60$

Original Age	Normalized Value
20	$\frac{20-20}{60-20} = \frac{0}{40} = 0.00$
30	$\frac{30-20}{60-20} = \frac{10}{40} = 0.25$
40	$\frac{40-20}{60-20} = \frac{20}{40} = 0.50$
50	$\frac{50-20}{60-20} = \frac{30}{40} = 0.75$
60	$\frac{60-20}{60-20} = \frac{40}{40} = 1.00$

Example 56: Min-Max: Mapping to a Custom Range [-1, 1]

Attribute: `Temperature` values: $\{-10, 0, 15, 25, 40\}$
$\min_A = -10, \quad \max_A = 40, \quad new\_min = -1, \quad new\_max = 1$

Formula:

$$v' = \frac{v-(-10)}{40-(-10)} \times (1-(-1)) + (-1) = \frac{v+10}{50} \times 2 - 1$$

Original	Normalized to $[-1, 1]$
-10	$\frac{0}{50} \times 2 - 1 = -1.00$
0	$\frac{10}{50} \times 2 - 1 = -0.60$
15	$\frac{25}{50} \times 2 - 1 = 0.00$
25	$\frac{35}{50} \times 2 - 1 = 0.40$
40	$\frac{50}{50} \times 2 - 1 = 1.00$

Example 57: Min-Max: Numerical

The income attribute has $\min = 10{,}000$ and $\max = 90{,}000$.
Using min-max normalization to $[0, 1]$, the value $v = 40{,}000$ maps to:

$$v' = \frac{40000-10000}{90000-10000} = \frac{30000}{80000} = 0.375$$

If a new value of $95{,}000$ arrives, the mapped value would be:

$$v' = \frac{95000-10000}{80000} = \frac{85000}{80000} = 1.0625 > 1$$

This falls **outside** $[0, 1]$ — showing min-max normalization does not handle out-of-range values well.

Example 58: Min-Max: Effect of Outlier

Attribute: `Salary (thousands)` $\{20, 25, 30, 35, 200\}$
$\min = 20, \quad \max = 200$

Salary	Normalized
20	$0/180 = 0.000$
25	$5/180 = 0.028$
30	$10/180 = 0.056$
35	$15/180 = 0.083$
200	$180/180 = 1.000$

Values 20–35 are all compressed into $[0,\ 0.083]$ due to the outlier 200.
This illustrates the major weakness of min-max normalization.

2.3.1.2 Z-Score Normalization (Standardization)

Z-Score Normalization (Standardization)

Transforms values so the resulting attribute has **mean = 0** and **standard deviation = 1**:

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

where $\bar{A}$ = mean of attribute A, σ_A = standard deviation of A

- Output is **not bounded** — values can go negative or exceed 1
- **Robust to outliers** compared to min-max
- Preferred when **distribution is approximately normal**
- Required when algorithm assumes **zero mean, unit variance** (e.g., PCA, SVM)

Using Sample Standard Deviation

When working with a **sample** (not full population), replace σ_A with sample std dev s_A:

$$v' = \frac{v - \bar{A}}{s_A}, \qquad s_A = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{N-1}}$$

In data mining, population std dev (N in denominator) is typically used when the full dataset is available.

Example 59: Z-Score: Basic Calculation

Attribute: `Marks` {50, 60, 70, 80, 90}

$$\bar{A} = \frac{50 + 60 + 70 + 80 + 90}{5} = 70$$

$$\sigma_A = \sqrt{\frac{(50-70)^2 + (60-70)^2 + (70-70)^2 + (80-70)^2 + (90-70)^2}{5}} = \sqrt{\frac{400 + 100 + 0 + 100 + 400}{5}} = \sqrt{200} \approx 14.14$$

Marks	$v - \bar{A}$	Z-score v'
50	-20	$-20/14.14 \approx -1.41$
60	-10	$-10/14.14 \approx -0.71$
70	0	$0/14.14 = 0.00$
80	$+10$	$10/14.14 \approx +0.71$
90	$+20$	$20/14.14 \approx +1.41$

Mean of normalized values $= 0$, Std Dev $= 1$ (verified)

Example 60: Z-Score: Numerical

An attribute `Weight_kg` has $\bar{A} = 65$ kg and $\sigma_A = 10$ kg.

(a) Normalize $v = 80$ kg:

$$v' = \frac{80 - 65}{10} = \frac{15}{10} = 1.5$$

(b) Normalize $v = 45$ kg:

$$v' = \frac{45 - 65}{10} = \frac{-20}{10} = -2.0$$

(c) What original value maps to $v' = 0.5$?

$$0.5 = \frac{v - 65}{10} \implies v = 65 + 5 = 70 \text{ kg}$$

(d) A value of $v' = -2.0$ means the weight is **2 standard deviations below** the mean.

Example 61: Z-Score: Finding Parameters from Normalized Values

After Z-score normalization, two values are known:
Original value 40 maps to $v' = -1.5$
Original value 70 maps to $v' = 1.5$

Find $\bar{A}$ and σ_A:

From the two equations:

$$-1.5 = \frac{40 - \bar{A}}{\sigma_A} \quad \text{and} \quad 1.5 = \frac{70 - \bar{A}}{\sigma_A}$$

Adding both equations:

$$0 = \frac{(40 + 70) - 2\bar{A}}{\sigma_A} \implies \bar{A} = 55$$

Substituting back:

$$1.5 = \frac{70 - 55}{\sigma_A} = \frac{15}{\sigma_A} \implies \sigma_A = 10$$

Example 62: Z-Score: Comparison with Min-Max on Same Data

Attribute: `Salary (thousands)` {20, 25, 30, 35, 200}

$\bar{A} = \frac{20+25+30+35+200}{5} = 62$, $\sigma_A = \sqrt{\frac{(20-62)^2+(25-62)^2+(30-62)^2+(35-62)^2+(200-62)^2}{5}}$

$= \sqrt{\frac{1764+1369+1024+729+19044}{5}} = \sqrt{\frac{23930}{5}} = \sqrt{4786} \approx 69.18$

Salary	Min-Max	Z-Score
20	0.000	$(20-62)/69.18 = -0.607$
25	0.028	$(25-62)/69.18 = -0.535$
30	0.056	$(30-62)/69.18 = -0.463$
35	0.083	$(35-62)/69.18 = -0.390$
200	1.000	$(200-62)/69.18 = +1.995$

- Min-max compresses 20–35 into $[0,\ 0.083]$; Z-score spreads them into $[-0.607,\ -0.390]$
- Z-score handles the outlier (200) more gracefully

2.3.1.3 Decimal Scaling

Decimal Scaling

Normalizes by moving the decimal point of attribute values:

$$v' = \frac{v}{10^j}$$

where j is the smallest integer such that $\max(|v'|) < 1$.

- Simple to compute — no mean or std dev required
- Range of result depends on actual data values
- Less commonly used in modern data mining pipelines

Example 63: Decimal Scaling

Attribute: `Votes` {350, 800, 1200, 4750, 9600}

$\max(|v|) = 9600$ — 5 digits, so $j = 4$ makes $\max(|v'|) < 1$?
$9600/10^4 = 0.96 < 1$ so $j = 4$

Original	After Decimal Scaling ($j = 4$)
350	$350/10000 = 0.0350$
800	$800/10000 = 0.0800$
1200	$1200/10000 = 0.1200$
4750	$4750/10000 = 0.4750$
9600	$9600/10000 = 0.9600$

Comparison of All Three Methods

Property	Min-Max	Z-Score	Decimal Scaling
Output range	$[new\_min,\ new\_max]$	$(-\infty, +\infty)$	$(0,\ 1)$ approx
Requires	Min, Max	Mean, Std Dev	Max value
Outlier effect	High (severe)	Moderate	Moderate
Preserves zero	No	No	Yes
Bounded output	Yes	No	Yes
When to use	Known bounded range	Normal dist.	Large integer values

Example 64: Reverse Mapping

After min-max normalization to $[0, 1]$, a value maps to $v' = 0.6$.
If $\min_A = 100$ and $\max_A = 600$, find the original value v.

$$0.6 = \frac{v - 100}{600 - 100} = \frac{v - 100}{500}$$
$$v - 100 = 0.6 \times 500 = 300 \implies v = 400$$

Example 65: Two Attributes Distance Comparison

Two data points $P_1 = (20,\ 10000)$ and $P_2 = (50,\ 40000)$
Attributes: `Age` $\in [20, 80]$, `Salary` $\in [10000, 70000]$

Before normalization (Euclidean distance):

$$d = \sqrt{(50-20)^2 + (40000-10000)^2} = \sqrt{900 + 9 \times 10^8} \approx 30000$$

Salary completely **dominates** the distance.

After min-max normalization to $[0, 1]$**:**
Age: $P_1 \to \frac{20-20}{60} = 0.0, \quad P_2 \to \frac{50-20}{60} = 0.5$
Salary: $P_1 \to \frac{10000-10000}{60000} = 0.0, \quad P_2 \to \frac{40000-10000}{60000} = 0.5$

$$d' = \sqrt{(0.5-0.0)^2 + (0.5-0.0)^2} = \sqrt{0.25 + 0.25} = \sqrt{0.5} \approx 0.707$$

Both attributes now contribute **equally** to the distance.

2.4 Data Discretization

2.4.1 Discretization by Binning

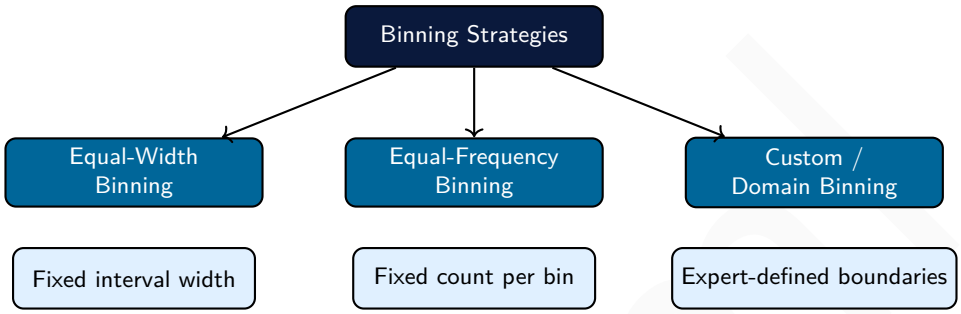
Discretization

Discretization converts a **continuous** attribute into a **discrete** one by dividing the value range into intervals called **bins**.

- Continuous attribute $\to$ finite set of intervals (ordinal categories)
- Each bin is assigned a label: `Low`, `Medium`, `High` or bin numbers
- Required by algorithms that work only on **discrete or categorical** data
- Also reduces **noise** by smoothing out minor fluctuations in values

Why Discretization?

- Some algorithms (e.g., Naive Bayes, Apriori) require **categorical** input
- Reduces the number of distinct values — speeds up learning
- Smooths noisy data — small variations within a bin are treated as equal
- Enables **concept hierarchy** generation for data cube roll-up
- Makes patterns more **interpretable**: "Age 20–35" is clearer than exact values



Equal-Width Binning

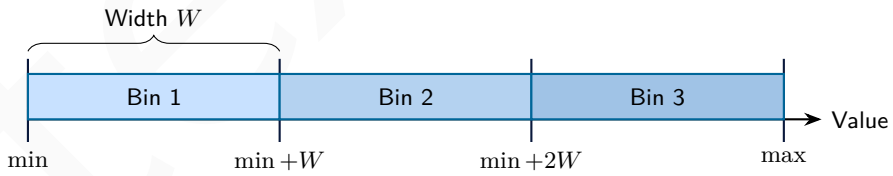
Divides the range $[\min_A,\ \max_A]$ into k bins of **equal width**:

$$W = \frac{\max_A - \min_A}{k}$$

Bin boundaries:

$$\min_A,\quad \min_A + W,\quad \min_A + 2W,\quad \ldots,\quad \max_A$$

- Simple to compute — only min, max, and k needed
- **Sensitive to outliers** — one extreme value creates a very wide range, leaving most data in one bin
- May produce **empty bins** if data is skewed



Example 66: Equal-Width: 3 Bins

Data (sorted): $\{5, 10, 15, 20, 25, 30, 35, 40, 45\}$ $\quad k = 3$

$$W = \frac{45 - 5}{3} = \frac{40}{3} \approx 13.33$$

Boundaries: $5,\quad 18.33,\quad 31.67,\quad 45$

Bin	Range	Values
Bin 1	$[5,\ 18.33)$	5, 10, 15
Bin 2	$[18.33,\ 31.67)$	20, 25, 30
Bin 3	$[31.67,\ 45]$	35, 40, 45

Each bin has 3 values here — balanced only because data is uniformly distributed.

Example 67: Equal-Width: Effect of Outlier

Data: {10, 12, 13, 15, 16, 18, 20, 22, 200} $k = 3$

$$W = \frac{200 - 10}{3} = \frac{190}{3} \approx 63.33$$

Boundaries: $10, \quad 73.33, \quad 136.67, \quad 200$

Bin	Range	Values
Bin 1	$[10,\ 73.33)$	10, 12, 13, 15, 16, 18, 20, 22
Bin 2	$[73.33,\ 136.67)$	(empty)
Bin 3	$[136.67,\ 200]$	200

The outlier 200 pulls the range wide, creating an empty bin and packing 8 of 9 values into Bin 1. This is the key **weakness** of equal-width binning.

Example 68: Equal-Width

Attribute: `Score` $\min = 0$, $\max = 100$, $k = 5$

$$W = \frac{100 - 0}{5} = 20$$

Bins: $[0, 20),\ [20, 40),\ [40, 60),\ [60, 80),\ [80, 100]$

Question: A score of 72 falls in which bin, and what is the bin number?

$72 \in [60,\ 80) \rightarrow$ **Bin 4**

Question: How many bins contain the values {5, 35, 60, 85, 100}?

- $5 \in [0, 20)$ — Bin 1
- $35 \in [20, 40)$ — Bin 2
- $60 \in [60, 80)$ — Bin 4
- $85 \in [80, 100]$ — Bin 5
- $100 \in [80, 100]$ — Bin 5

4 distinct bins are occupied.

Equal-Frequency Binning

Divides data so that each bin contains **approximately the same number of values**:

$$\text{Count per bin} = \frac{N}{k}$$

- Also called **equi-depth** or **quantile-based** binning
- Bin widths vary — determined by the data distribution
- **No empty bins** — guaranteed roughly equal occupancy
- **Better for skewed data** than equal-width

- Boundaries are set at the appropriate **quantile values**

Example 69: Equal-Frequency: 3 Bins

Data (sorted): $\{5, 10, 15, 20, 25, 30, 35, 40, 45\}$ $N = 9$, $k = 3$

Count per bin $= 9/3 = 3$

Bin	Values	Range
Bin 1	5, 10, 15	$[5,\ 15]$
Bin 2	20, 25, 30	$[20,\ 30]$
Bin 3	35, 40, 45	$[35,\ 45]$

For uniform data, equal-width and equal-frequency produce the same result.

Example 70: Equal-Frequency: Skewed Data

Data (sorted): $\{2, 3, 4, 5, 6, 50, 55, 60, 100\}$ $N = 9$, $k = 3$

Count per bin $= 3$

Bin	Values	Width
Bin 1	2, 3, 4	$100 - 2 = 4$ (narrow)
Bin 2	5, 6, 50	$50 - 5 = 45$
Bin 3	55, 60, 100	$100 - 55 = 45$ (wide)

- Each bin has exactly 3 values — frequency balanced
- Bin widths differ significantly — equal-frequency adapts to skew
- Equal-width would have put 2,3,4,5,6 all in one bin here

Example 71: Equal-Frequency

Data (sorted): $\{4, 8, 15, 16, 23, 25, 30, 42, 55, 60\}$ $N = 10$, $k = 2$

Count per bin $= 10/2 = 5$

Bin	Values
Bin 1	4, 8, 15, 16, 23
Bin 2	25, 30, 42, 55, 60

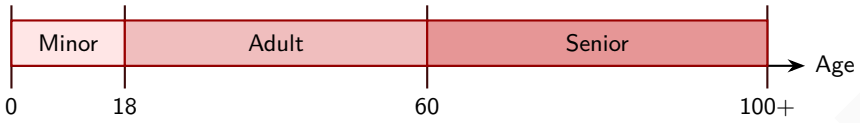
Bin boundary is set between 23 and 25 (between 5th and 6th sorted value).
New value 18: falls in Bin 1 $[4,\ 23]$.
New value 24: boundary ambiguity — assign to Bin 1 or Bin 2 based on convention.

Custom / Domain Binning

Divides the range based on **expert knowledge**, predefined standards, or specific business logic rather than mathematical distributions:

- **Meaningful Thresholds** — Boundaries represent real-world categories (e.g., Age groups, Credit scores, Temperature zones).

- **Unequal Widths** — Bins can have completely different sizes based on the importance of specific ranges.
- **Interpretability** — Results are immediately actionable for decision-makers.



Example 72: Domain Binning: Academic Grading

Data (Scores): {42, 55, 68, 72, 85, 91, 94} **Logic:** Institutional Policy

Boundaries are fixed by the department: $0, 50, 75, 100$

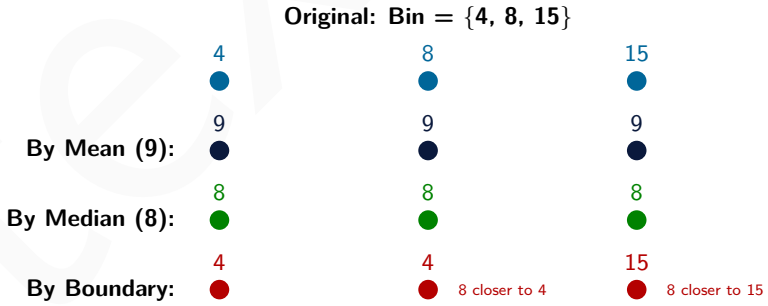
Category	Range	Width	Values
Fail	$[0, 50)$	50	42
Pass	$[50, 75)$	25	55, 68, 72
Distinction	$[75, 100]$	25	85, 91, 94

Bin widths are inconsistent (50 vs 25), but the division is **meaningful** for the domain.

Smoothing Methods within Bins

After assigning values to bins, the bin values can be **smoothed** (replaced) by a representative:

- **Smoothing by bin mean:** Replace each value with the **mean** of its bin
- **Smoothing by bin median:** Replace each value with the **median** of its bin
- **Smoothing by bin boundaries:** Replace each value with the **nearest boundary** (min or max of the bin)



Example 73: Smoothing by All Three Methods

Sorted data divided into 3 bins:

- Bin 1: {4, 8, 15}
- Bin 2: {21, 24, 25}
- Bin 3: {28, 34, 40}

Smoothing by bin mean:

- Bin 1 mean $= (4 + 8 + 15)/3 = 9 \quad \rightarrow \{9, 9, 9\}$

- Bin 2 mean $= (21 + 24 + 25)/3 = 23.33 \quad \rightarrow \{23.33, 23.33, 23.33\}$
- Bin 3 mean $= (28 + 34 + 40)/3 = 34 \quad \rightarrow \{34, 34, 34\}$

Smoothing by bin median:

- Bin 1 median $= 8 \quad \rightarrow \{8, 8, 8\}$
- Bin 2 median $= 24 \quad \rightarrow \{24, 24, 24\}$
- Bin 3 median $= 34 \quad \rightarrow \{34, 34, 34\}$

Smoothing by bin boundaries:
(Each value snapped to nearest of bin min or bin max)

- Bin 1: boundaries $=$ 4 and 15
  - $4 \rightarrow 4$ (distance to 4: 0, to 15: 11) $\rightarrow$ 4
  - $8 \rightarrow 4$ (distance to 4: 4, to 15: 7) $\rightarrow$ 4
  - $15 \rightarrow 15$ (distance to 4: 11, to 15: 0) $\rightarrow$ 15
- Bin 2: boundaries $=$ 21 and 25 $\quad \rightarrow \{21, 25, 25\}$
- Bin 3: boundaries $=$ 28 and 40 $\quad \rightarrow \{28, 28, 40\}$

Example 74: Full Binning Pipeline

Attribute: `Price` (in hundreds): {10, 20, 30, 40, 50, 60, 70, 80, 90}
Apply **equal-width binning** with $k = 3$ and smooth by **bin mean**.

Step 1 — Compute width:

$$W = \frac{90 - 10}{3} = \frac{80}{3} \approx 26.67$$

Step 2 — Assign bins:

Bin	Range	Values
Bin 1	$[10,\ 36.67)$	10, 20, 30
Bin 2	$[36.67,\ 63.33)$	40, 50, 60
Bin 3	$[63.33,\ 90]$	70, 80, 90

Step 3 — Smooth by bin mean:

- Bin 1 mean $= (10 + 20 + 30)/3 = 20 \quad \rightarrow \{20, 20, 20\}$
- Bin 2 mean $= (40 + 50 + 60)/3 = 50 \quad \rightarrow \{50, 50, 50\}$
- Bin 3 mean $= (70 + 80 + 90)/3 = 80 \quad \rightarrow \{80, 80, 80\}$

Smoothed dataset: {20, 20, 20, 50, 50, 50, 80, 80, 80}

Equal-Width vs Equal-Frequency — Comparison

Property	Equal-Width	Equal-Frequency
Bin width	Fixed (same for all bins)	Varies by data
Values per bin	Varies	Fixed (approximately)
Empty bins	Possible	Not possible
Outlier effect	High (distorts widths)	Low (absorbed into a bin)
Good for	Uniform distributions	Skewed distributions
Boundary rule	$\min + i \times W$	Quantile positions

2.4.2 Discretization by Histogram Analysis

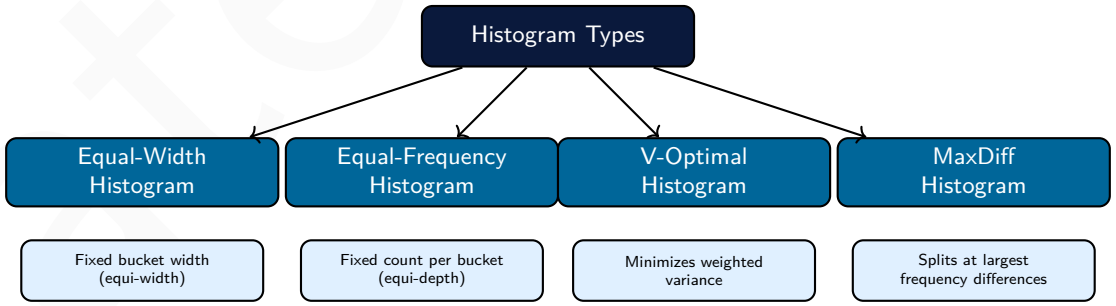
Histogram Analysis

A **histogram** partitions the value range of an attribute into **adjacent, non-overlapping intervals** and displays the **frequency** of values in each interval.

- Each interval is called a **bucket** or **bin**
- Height of each bar represents **frequency** (count) or **relative frequency** (proportion)
- Used for both **visualization** and **discretization**
- The shape of a histogram reveals the **distribution** of the attribute

Histogram vs Binning

- Binning (previous section) focuses on **assigning values to bins** and smoothing
- Histogram analysis focuses on **analyzing the frequency distribution** to find **natural partitions**
- Histogram-based discretization identifies bin boundaries based on **data density** rather than fixed width or count rules
- Boundaries are placed where **frequency drops** — natural valleys in the histogram



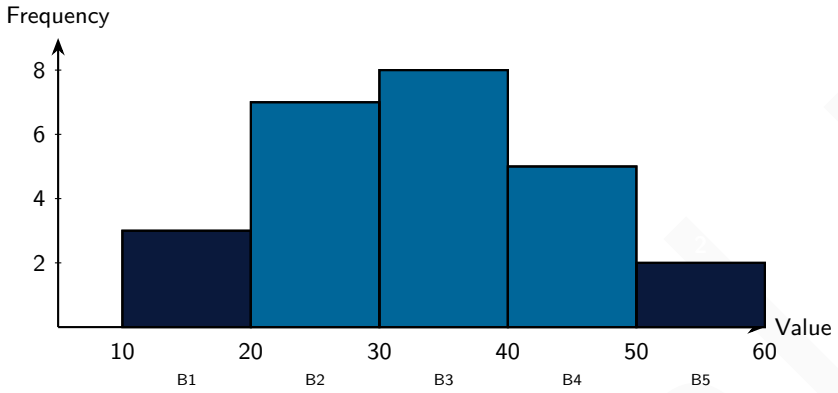
Equal-Width Histogram

Partitions the range $[\min_A, \max_A]$ into k buckets of **equal width**:

$$W = \frac{\max_A - \min_A}{k}$$

- Each bucket spans the same range of values
- Bucket heights (frequencies) vary according to data distribution

- Same as equal-width binning — the histogram **visualizes** the result
- Easy to construct; sensitive to outliers and skew



Example 75: Equal-Width Histogram Construction

Data: 25 salary values (in thousands) ranging from 10 to 60. $k = 5$ buckets.

$$W = \frac{60 - 10}{5} = 10$$

Bucket	Range	Frequency	Relative Freq.
B1	$[10,\ 20)$	3	$3/25 = 0.12$
B2	$[20,\ 30)$	7	$7/25 = 0.28$
B3	$[30,\ 40)$	8	$8/25 = 0.32$
B4	$[40,\ 50)$	5	$5/25 = 0.20$
B5	$[50,\ 60]$	2	$2/25 = 0.08$
Total		**25**	**1.00**

- Distribution is **roughly bell-shaped** — peak at B3
- B3 $[30, 40)$ is the most frequent bucket — modal class
- Discretization assigns: B1 $\rightarrow$ Low, B2–B3 $\rightarrow$ Medium, B4–B5 $\rightarrow$ High

Equal-Frequency (Equi-Depth) Histogram

Each bucket contains approximately the **same number of data points**:

$$\text{Count per bucket} = \frac{N}{k}$$

- Bucket **widths vary** to accommodate equal counts
- No bucket is empty — more informative for skewed data
- Boundary of each bucket set at **quantile positions**
- Better captures shape of skewed distributions

Example 76: Equal-Frequency Histogram

Sorted data (20 values):
$\{5, 8, 10, 12, 15, 18, 20, 22, 25, 28, 30, 35, 40, 45, 50, 55, 60, 70, 80, 100\}$
$k = 4$ buckets, count per bucket $= 20/4 = 5$

Bucket	Values	Range	Width
B1	5, 8, 10, 12, 15	$[5,\ 15]$	10
B2	18, 20, 22, 25, 28	$[18,\ 28]$	10
B3	30, 35, 40, 45, 50	$[30,\ 50]$	20
B4	55, 60, 70, 80, 100	$[55,\ 100]$	45

- B1 and B2 are narrow — data is dense at lower values
- B4 is very wide — data is sparse at higher values
- All buckets have exactly 5 values — frequency is balanced

V-Optimal Histogram

Partitions data into k buckets to **minimize total weighted variance**:

$$\text{Cost} = \sum_{i=1}^{k} n_i \cdot \sigma_i^2$$

where n_i = number of values in bucket i, $\quad \sigma_i^2$ = variance of values in bucket i

- Produces the **most accurate** histogram approximation
- Uses **dynamic programming** to find the optimal partition
- Computationally expensive — $O(N^2 k)$ time complexity
- Groups values that are **close to each other** in the same bucket

Example 77: V-Optimal: Concept with 2 Buckets

Data: $\{1, 2, 3, 20, 21, 22\}$ $\quad k = 2$

Candidate partitions and their weighted variance:

Partition A: $\{1,2,3\}$ and $\{20,21,22\}$

$$\bar{x}_1 = 2, \quad \sigma_1^2 = \frac{(1-2)^2 + (2-2)^2 + (3-2)^2}{3} = \frac{2}{3}$$

$$\bar{x}_2 = 21, \quad \sigma_2^2 = \frac{(20-21)^2 + (21-21)^2 + (22-21)^2}{3} = \frac{2}{3}$$

$$\text{Cost}_A = 3 \times \frac{2}{3} + 3 \times \frac{2}{3} = 2 + 2 = \mathbf{4}$$

Partition B: $\{1,2,3,20\}$ and $\{21,22\}$

$$\bar{x}_1 = 6.5, \quad \sigma_1^2 = \frac{(1-6.5)^2 + (2-6.5)^2 + (3-6.5)^2 + (20-6.5)^2}{4} = \frac{30.25 + 20.25 + 12.25 + 182.25}{4} = \frac{245}{4} = 61.25$$

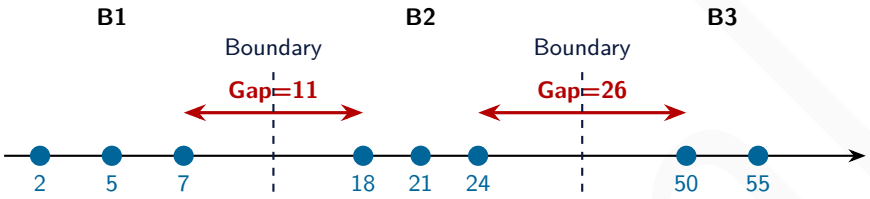
$$\text{Cost}_B = 4 \times 61.25 + 2 \times 0.25 = 245 + 0.5 = \mathbf{245.5}$$

Partition A has **much lower cost** — V-Optimal selects it.
This matches natural clusters in the data: $\{1,2,3\}$ and $\{20,21,22\}$.

MaxDiff Histogram

Places **bucket boundaries** at the $k-1$ largest **consecutive differences** in the sorted data.

- Sort the data values
- Compute difference between each pair of adjacent (consecutive) values
- Place boundaries at the positions of the $k-1$ **largest differences**
- Natural: splits where **data gaps are largest**
- Simple to compute — no variance calculation needed



Example 78: MaxDiff Histogram: Step-by-Step

Sorted data: $\{2, 5, 7, 18, 21, 24, 50, 55\}$ $k = 3$ buckets

Step 1 — Compute consecutive differences:

Pair	Values	Difference
d_1	$(2, 5)$	3
d_2	$(5, 7)$	2
d_3	$(7, 18)$	**11**
d_4	$(18, 21)$	3
d_5	$(21, 24)$	3
d_6	$(24, 50)$	**26**
d_7	$(50, 55)$	5

Step 2 — Select $k-1=2$ largest differences:
$d_6 = 26$ (largest) and $d_3 = 11$ (second largest)

Step 3 — Place boundaries after 7 and after 24:

Bucket	Values
B1	$\{2, 5, 7\}$
B2	$\{18, 21, 24\}$
B3	$\{50, 55\}$

MaxDiff correctly identifies the **three natural clusters** in the data.

Example 79: MaxDiff with 4 Buckets

Sorted data: $\{3, 4, 5, 15, 16, 40, 41, 42, 43, 90\}$ $k = 4$

Consecutive differences:

Pair	Values	Difference
d_1	$(3, 4)$	1
d_2	$(4, 5)$	1
d_3	$(5, 15)$	**10**
d_4	$(15, 16)$	1
d_5	$(16, 40)$	**24**
d_6	$(40, 41)$	1
d_7	$(41, 42)$	1
d_8	$(42, 43)$	1
d_9	$(43, 90)$	**47**

Top $k - 1 = 3$ differences: $d_9 = 47$, $d_5 = 24$, $d_3 = 10$

Boundaries placed after: 43, 16, 5
Final buckets:

Bucket	Values
B1	{3, 4, 5}
B2	{15, 16}
B3	{40, 41, 42, 43}
B4	{90}

Discretization from Histogram Analysis

Once a histogram is built, bin boundaries serve as **discretization cut points**:

- Each bucket becomes a discrete interval (category)
- Values within a bucket are assigned the **same discrete label**
- Cut points are the **boundary values** between adjacent buckets
- For k buckets, there are $k - 1$ cut points and k discrete categories

Example 80: Discretization Cut Points from Histogram

An equal-width histogram is built for `Age` with $\min = 0$, $\max = 80$, $k = 4$ buckets.

$$W = \frac{80 - 0}{4} = 20$$

Buckets: $[0, 20)$, $[20, 40)$, $[40, 60)$, $[60, 80]$

Cut points: $\{20, 40, 60\} \rightarrow k - 1 = 3$ cut points

Discretized labels:

Bucket	Range	Label
B1	$[0, 20)$	Young
B2	$[20, 40)$	Adult
B3	$[40, 60)$	Middle-aged
B4	$[60, 80]$	Senior

A new value of 35 $\rightarrow$ falls in B2 $\rightarrow$ labeled `Adult`.

Comparison of Histogram Types

Type	Boundary Rule	Accuracy	Cost
Equal-Width	Fixed width W	Low (skew-sensitive)	$O(N)$
Equal-Frequency	Fixed count per bucket	Medium	$O(N \log N)$
V-Optimal	Min weighted variance	Highest	$O(N^2 k)$
MaxDiff	Largest adjacent gaps	High (cluster-aware)	$O(N \log N)$

2.5 Problems

Problem 33 ***[MCQ]*** *A data warehouse analyst notices that the attribute* `Age` *contains the value* ***999*** *for several records. This is most likely an example of:*

(A) Noise

(B) Missing value encoded as a sentinel

(C) Outlier arising from natural variation

(D) Duplicate record

Problem 34 ***[MSQ]*** *Which of the following are valid strategies for handling* ***missing values*** *in a dataset?*

(A) Ignore the tuple entirely

(B) Fill in the missing value manually

(C) Use the attribute mean/median/mode for imputation

(D) Use a learning algorithm (e.g., Bayesian classifier) to infer the value

Problem 35 ***[MCQ]*** *Consider a sorted attribute partition:* $\{4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34\}$ *divided into bins of depth 4. After* ***smoothing by bin means****, the first bin is replaced by:*

(A) $\{9, 9, 9, 9\}$

(B) $\{4, 4, 15, 15\}$

(C) $\{9, 9, 9, 9\}$

(D) $\{8, 8, 8, 8\}$ *(rounded)*

Problem 36 ***[NAT]*** *The sorted data values are:* $5,\ 10,\ 11,\ 13,\ 15,\ 35,\ 50,\ 55,\ 72,\ 92$. *They are placed into* ***equal-depth bins*** *of size 5. After* ***smoothing by bin boundaries****, the value* ***13*** *is replaced by* _____.

Problem 37 ***[MCQ]*** *In* ***smoothing by bin medians****, the sorted bin* $\{7, 12, 16, 20, 25\}$ *is smoothed to:*

(A) $\{12, 12, 12, 12, 12\}$

(B) $\{16, 16, 16, 16, 16\}$

(C) $\{7, 12, 16, 20, 25\}$ *(unchanged)*

(D) $\{7, 7, 16, 25, 25\}$

Problem 38 ***[MSQ]*** *Regarding* ***regression-based smoothing*** *for noisy data, which statements are correct?*

(A) Linear regression fits a straight line to minimize squared error between observed and predicted values.

(B) Multiple linear regression allows more than one predictor variable.

(C) Regression can be used to detect outliers by examining residuals.

(D) Regression smoothing is a non-parametric *method requiring no model assumptions.*

Problem 39 ***[MCQ]*** *Which statement* ***best*** *distinguishes* noise *from an* outlier*?*

(A) Noise is always intentionally introduced; outliers arise naturally.

(B) Noise is a random error or variance in a measured variable; an outlier is a data object that deviates remarkably from the rest.

(C) Outliers are always errors; noise can be valid data.

(D) Noise affects categorical attributes; outliers affect numerical attributes.

Problem 40 ***[MSQ]*** *Clustering-based methods for noisy data detection work on the principle that:*

(A) Values that fall outside all clusters are likely noise or outliers.

(B) Every data point must belong to exactly one cluster.

(C) Small or sparse clusters may represent noise.

(D) Cluster centroids always represent the "clean" version of the data.

Problem 41 ***[NAT]*** *A dataset of 200 records has 15 records with missing values for attribute A. If the missing values are imputed using the* ***global mean*** *of the non-missing values of A, the percentage of records modified is _____%.*

Problem 42 ***[MCQ]*** *Principal Component Analysis (PCA) reduces dimensionality by:*

(A) Selecting a subset of original features based on a filter criterion.

(B) Projecting data onto orthogonal directions that maximize variance.

(C) Clustering features into groups and selecting one from each group.

(D) Removing features with more than 50% missing values.

Problem 43 ***[MSQ]*** *Which of the following properties hold for the* ***principal components*** *in PCA?*

(A) Each component is a linear combination of the original attributes.

(B) Principal components are mutually orthogonal.

(C) The first principal component captures the maximum variance.

(D) Principal components are always axis-aligned with the original feature axes.

Problem 44 ***[NAT]*** *A dataset in 5 dimensions has the following eigenvalues of the covariance matrix: $\lambda_1 = 50,\ \lambda_2 = 30,\ \lambda_3 = 12,\ \lambda_4 = 6,\ \lambda_5 = 2$. The* ***percentage of variance*** *explained by the first* ***two*** *principal components is _____%.*

Problem 45 ***[MCQ]*** *In* ***forward feature selection****, the algorithm:*

(A) Starts with all features and removes the least important one at each step.

(B) Starts with no features and greedily adds the most informative feature at each step.

(C) Evaluates all 2^n subsets and picks the best one.

(D) Randomly selects a subset and iteratively swaps features.

Problem 46 ***[MCQ] Backward feature elimination*** *differs from forward selection primarily in:*

(A) It uses a different evaluation metric (information gain vs. accuracy).

(B) It begins with the complete set of features and removes one at each step, whereas forward selection begins with an empty set.

(C) It is only applicable to categorical features.

(D) It requires a validation set, while forward selection does not.

Problem 47 ***[MSQ]*** *Using a* ***decision tree*** *for feature selection:*

(A) Features appearing near the root of the tree are generally more informative.

(B) Features never selected by the tree can be considered candidates for removal.

(C) The depth of the tree determines the number of features selected.

(D) Decision tree-based selection is an example of an embedded *method.*

Problem 48 ***[MCQ]*** *Consider a 100×50 data matrix (100 samples, 50 features). After PCA, you retain components explaining 95% of variance, resulting in 8 principal components. The* ***compression ratio*** *(original dimensions : reduced dimensions) is:*

(A) $50 : 8$

(B) $100 : 8$

(C) $8 : 50$

(D) $100 : 50$

Problem 49 ***[MCQ]*** *In* ***simple random sampling without replacement*** *(SRSWOR) from a population of* $N = 1000$ *records, if a sample of size* $n = 100$ *is drawn, the probability that a specific record is included in the sample is:*

(A) $\frac{1}{1000}$

(B) $\frac{1}{100}$

(C) $\frac{100}{1000}$

(D) $\frac{99}{999}$

Problem 50 ***[NAT]*** *A population of 5000 records is divided into 4 strata of sizes 1000, 1500, 2000, and 500 respectively. Using* ***proportional stratified sampling*** *with a total sample size of 400, the number of records drawn from stratum 3 (size 2000) is* _____.

Problem 51 ***[MSQ]*** *In* ***sampling with replacement*** *(SRSWR), which of the following are true?*

(A) The same record can appear multiple times in the sample.

(B) The sample size can exceed the population size.

(C) Each draw is independent of previous draws.

(D) It guarantees a more representative sample than SRSWOR.

Problem 52 ***[MCQ]*** *In* ***equal-allocation stratified sampling****, a population of 6000 records split into 3 strata (2000, 1500, 2500) is sampled with a total sample of 300. How many records are drawn from stratum 2 (size 1500)?*

(A) 75

(B) 100

(C) 125

(D) 150

Problem 53 ***[NAT]*** *A database of 8000 tuples is to be reduced using* ***simple random sampling without replacement*** *to a 10% sample. The probability that any particular tuple is* ***not*** *selected (rounded to two decimal places) is* _____.

Problem 54 ***[MCQ]*** *Which sampling technique is* ***most appropriate*** *when the population has several distinct sub-groups (e.g., different classes of customers) and you want to ensure each sub-group is proportionally represented?*

(A) Simple random sampling without replacement

(B) Simple random sampling with replacement

(C) Systematic sampling

(D) Proportional stratified sampling

Problem 55 ***[MSQ]*** *Regarding* ***clustering*** *as a numerosity reduction technique, which statements are correct?*

(A) The original dataset is replaced by cluster centroids (or medoids) for storage/processing.

(B) Clustering for numerosity reduction is a lossy process.

(C) The number of clusters determines the degree of reduction.

(D) Clustering guarantees a reduction in numerical error compared to the original data.

Problem 56 ***[MCQ] Run-Length Encoding (RLE)*** *is most effective when the data:*

(A) Has a uniform distribution of values across the attribute range.

(B) Contains long sequences of repeated identical values.

(C) Has many distinct values with no repetition.

(D) Is stored in a sorted B-tree index.

Problem 57 ***[NAT]*** *Apply RLE to the binary string:* `AAAABBBCCDAAAA`*. Express as* $(value, count)$ *pairs. The* ***total number of*** $(value, count)$ ***pairs*** *in the RLE encoding is _____.*

Problem 58 ***[MCQ]*** *In* ***Huffman coding****, which character gets the* ***shortest*** *codeword?*

(A) The character with the lowest frequency

(B) The character with the highest frequency

(C) The character with the longest original encoding

(D) The character appearing last in alphabetical order

Problem 59 ***[NAT]*** *Consider characters with frequencies:* $A = 5,\ B = 9,\ C = 12,\ D = 13,\ E = 16,\ F = 45$*. Using Huffman coding, the* ***minimum number of bits*** *required to encode the entire message (using the optimal Huffman tree) for a message of 100 characters distributed as above is _____.*

Problem 60 ***[MSQ]*** *Which of the following are properties of* ***Huffman encoding****?*

(A) It is a lossless compression technique.

(B) More frequent symbols are assigned shorter codewords.

(C) The generated code is a prefix-free code (no codeword is a prefix of another).

(D) None of these.

Problem 61 ***[MCQ] LZW (Lempel-Ziv-Welch)*** *compression builds its dictionary:*

(A) Statically, before encoding begins, based on symbol frequencies.

(B) Dynamically, during encoding, by adding new patterns encountered in the input.

(C) Using a fixed-size sliding window of recent symbols.

(D) By exhaustively enumerating all possible substrings up to a fixed length.

Problem 62 ***[MCQ]*** *The key distinction between* ***lossless*** *and* ***lossy*** *data compression is:*

(A) Lossless compression achieves higher compression ratios than lossy compression.

(B) Lossless compression allows exact reconstruction of the original data; lossy compression does not.

(C) Lossy compression is only applicable to textual data.

(D) Lossless compression always uses dictionary-based methods.

Problem 63 ***[NAT]*** *A string* `ABABABAB` *is encoded using RLE on* ***pairs****: The encoding* `AB` $\times$ *4 uses 3 tokens. If standard character-level RLE is applied (encoding only consecutive identical characters), the number of* $(char, count)$ *pairs in the RLE output is _____.*

Problem 64 ***[MCQ]*** *PCA used as a* ***lossy compression*** *technique achieves compression by:*

(A) Quantizing each feature to a fixed number of bits.

(B) Retaining only the top-k *principal components and discarding the rest.*

(C) Replacing each data point with its nearest cluster centroid.

(D) Encoding runs of repeated values.

Problem 65 ***[NAT]*** *A value* $v = 73$ *from the attribute* `Score` *with* $\min = 50$ *and* $\max = 100$ *is to be normalized to the range* $[0, 1]$ *using* ***min-max normalization****. The normalized value (rounded to two decimal places) is _____.*

Problem 66 ***[MCQ]*** *Min-max normalization maps a value* v *to* v' *in the range* $[new\_\min,\ new\_\max]$ *using:*

(A) $v' = \dfrac{v - \min_A}{\max_A - \min_A}$

(B) $v' = \dfrac{v - \min_A}{\max_A - \min_A}(new\_\max - new\_\min) + new\_\min$

(C) $v' = \dfrac{v - \mu_A}{\sigma_A}$

(D) $v' = \dfrac{v}{10^j}$ *where* j *is chosen to bring* $|v'| < 1$

Problem 67 ***[NAT]*** *The attribute* `Income` *has* $\mu = 56{,}000$ *and* $\sigma = 16{,}000$*. Using* ***z-score normalization****, the standardized value of Income* $= 73{,}000$ *(rounded to two decimal places) is* _____.

Problem 68 ***[MSQ]*** *Z-score normalization (standardization) is appropriate when:*

(A) The actual minimum and maximum of an attribute are unknown.

(B) The data is known to follow a Gaussian distribution.

(C) You want to preserve the relative distances between data points.

(D) The attribute contains outliers that heavily skew the min/max.

Problem 69 ***[NAT]*** *Using* ***decimal scaling****, the value* $v = 0.00372$ *is normalized so that* $|v'| < 1$ *with the smallest possible* j *(where* $v' = v/10^j$*). The value of* j *is* _____.

Problem 70 ***[MCQ]*** *A dataset has values:* $200,\ 300,\ 400,\ 600,\ 1000$*. After* ***decimal scaling*** *with* $j = 3$*, the transformed values are:*

(A) $0.002,\ 0.003,\ 0.004,\ 0.006,\ 0.01$

(B) $0.2,\ 0.3,\ 0.4,\ 0.6,\ 1.0$

(C) $2,\ 3,\ 4,\ 6,\ 10$

(D) $20,\ 30,\ 40,\ 60,\ 100$

Problem 71 ***[NAT]*** *An attribute has values:* $10,\ 20,\ 30,\ 40,\ 50$*. It is min-max normalized to the range* $[0,\ 10]$*. The normalized value of* $x = 35$ *is* _____.

Problem 72 ***[MSQ]*** *Which normalization techniques are* ***sensitive to outliers****?*

(A) Min-max normalization

(B) Z-score normalization

(C) Decimal scaling

(D) Normalization using inter-quartile range (IQR)

Problem 73 ***[MCQ]*** *After min-max normalization to* $[0, 1]$*, a new data point arrives with a value* greater than the previous maximum*. The normalized value of this new point will be:*

(A) Between 0 and 1

(B) Exactly 1

(C) Greater than 1

(D) Exactly 0

Problem 74 ***[NAT]*** *A dataset consists of values:* $4,\ 8,\ 15,\ 16,\ 23,\ 42$*. After* ***z-score normalization****, the standardized value of* $x = 16$ *is* _____ *(rounded to two decimal places).*

Problem 75 ***[MCQ]*** *In* ***decimal scaling normalization****, the number of decimal places by which a value is shifted (i.e.,* j*) is determined by:*

(A) The mean of the attribute

(B) The smallest integer such that $\max(|v'|) < 1$

(C) The standard deviation of the attribute

(D) The number of distinct values in the attribute

Problem 76 ***[NAT]*** *Values of an attribute are: $-200,\ -150,\ 0,\ 125,\ 300$. Using min-max normalization to $[0, 1]$, the normalized value of $x = 125$ (rounded to two decimal places) is _____.*

Problem 77 ***[MCQ]*** *In **equal-width binning** with k bins for data in $[\min, \max]$, the width of each bin is:*

(A) $\dfrac{\max - \min}{k}$

(B) $\dfrac{n}{k}$ *where n is the number of data points*

(C) $\dfrac{\max + \min}{k}$

(D) $\dfrac{\max - \min}{k - 1}$

Problem 78 ***[NAT]*** *Data values: $5,\ 10,\ 11,\ 13,\ 15,\ 35,\ 50,\ 55,\ 72,\ 92$. Using **equal-width binning** with $k = 3$ bins, the width of each bin is _____(rounded to one decimal if needed).*

Problem 79 ***[MCQ]*** *For the data $\{5, 10, 11, 13, 15, 35, 50, 55, 72, 92\}$ with **equal-width binning** ($k = 3$), which bin does the value **55** fall into?*

(A) Bin 1: $[5, 34)$

(B) Bin 2: $[34, 63)$

(C) Bin 3: $[63, 92]$

(D) Bin 2 and Bin 3 (it is a boundary value)

Problem 80 ***[NAT]*** *Data: $3,\ 7,\ 8,\ 10,\ 14,\ 18,\ 20,\ 24,\ 28,\ 30$ (10 values). After **equal-frequency binning** with $k = 2$ bins, the **mean** of Bin 2 (second bin) is _____.*

Problem 81 ***[MSQ]*** *Which of the following are advantages of **equal-frequency (equi-depth) binning** over **equal-width binning**?*

(A) Each bin contains the same number of data points, avoiding heavily skewed bins.

(B) Equal-frequency binning handles skewed data distributions better.

(C) Equal-frequency binning always produces more accurate models.

(D) The bin boundaries in equal-frequency binning are data-driven, not fixed.

Problem 82 ***[MCQ]*** *In **equal-width binning**, a major drawback is:*

(A) It requires sorting the data before binning.

(B) Bins can have very unequal frequencies if the data is skewed.

(C) The number of bins must be specified as a power of 2.

(D) Each bin must contain at least one value.

Problem 83 ***[MSQ] Domain knowledge-based binning*** *(concept hierarchy):*

(A) Allows discretization based on semantically meaningful intervals (e.g., age groups: youth, adult, senior).

(B) Is completely automatic and requires no user input.

(C) Can yield bins of unequal width and unequal frequency.

(D) Is an example of a top-down discretization approach.

Problem 84 ***[MCQ]*** *In a **V-Optimal histogram** with k partitions, the objective is to:*

(A) Minimize the number of distinct values within each partition.

(B) Minimize the weighted variance of the values within each partition.

(C) Maximize the number of data points in the first partition.

(D) Minimize the total number of bucket boundaries.

Problem 85 ***[MCQ]*** *The **MaxDiff histogram** places partition boundaries between adjacent values v_i and v_{i+1} where:*

(A) The difference in cumulative frequency is maximum.

(B) The difference $v_{i+1} - v_i$ is the largest among all adjacent pairs.

(C) The number of data points in v_i's interval is maximum.

(D) The bin containing v_i has the highest variance.

Problem 86 ***[NAT]*** *Consider the sorted data values and their frequencies:*

Value	5	10	20	25	50	80
Freq	3	5	2	8	1	4

*Using a **MaxDiff histogram** with $k = 3$ buckets (i.e., 2 boundaries), partition boundaries are placed at the two largest adjacent **value** differences. The **sum** of the two values at which boundaries are placed is _____.*

Problem 87 ***[MSQ]*** *An **equal-width histogram** partitions the value range into k intervals of equal width. Which limitations apply?*

(A) Sparse regions waste bins while dense regions may be poorly represented.

(B) The histogram is sensitive to outliers that extend the value range.

(C) It cannot be used for numerical attributes.

(D) The number of bins k must be determined by the user or via a heuristic.

Problem 88 ***[MCQ]*** *Which histogram type is theoretically **optimal** in minimizing approximation error (variance) for a given number of buckets, but is also computationally expensive?*

(A) Equal-width histogram

(B) Equal-frequency histogram

(C) V-Optimal histogram

(D) MaxDiff histogram

Problem 89 ***[MCQ]*** *Consider the following pipeline:*

$$\textit{Raw Data} \xrightarrow{(1)} \textit{Clean Data} \xrightarrow{(2)} \textit{Transformed Data} \xrightarrow{(3)} \textit{Reduced Data}$$

*Which ordering of operations is **most appropriate** in practice?*

(A) Normalization → Missing value imputation → PCA

(B) Missing value imputation → Normalization → PCA

(C) PCA → Missing value imputation → Normalization

(D) PCA → Normalization → Missing value imputation

Problem 90 ***[MSQ]*** *Which of the following operations are **lossless** with respect to the original data?*

(A) Min-max normalization

(B) Equal-width binning followed by smoothing by bin mean

(C) Huffman encoding

(D) Replacing a data attribute with its PCA projection onto the top-2 components (from 10 original dimensions)

Problem 91 ***[MCQ]*** *A PCA-based lossy compression retains 3 of 10 original principal components. A new query point is projected into this 3D space and then reconstructed back into the original 10D space. The reconstructed point:*

(A) Exactly equals the original point.

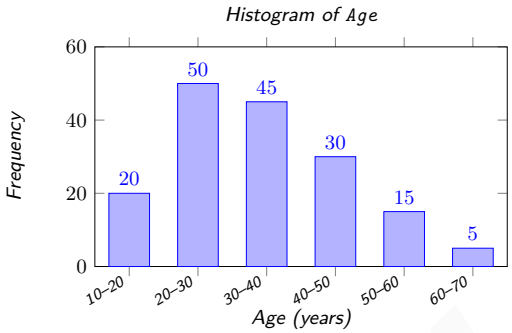
(B) Is the closest point in the 3D subspace to the original point, but differs from it in the other 7 dimensions.

(C) Has zero values in the 7 discarded dimensions.

(D) Cannot be reconstructed at all.

Problem 92 ***[NAT]*** *Data values in a bin (before smoothing):* $12,\ 17,\ 23,\ 29,\ 33$. *After* ***smoothing by bin boundaries****, the number of values that* change *from their original value is* _____.

(Bin boundaries are the minimum and maximum values in the bin.)



Problem 93 ***[MCQ]***

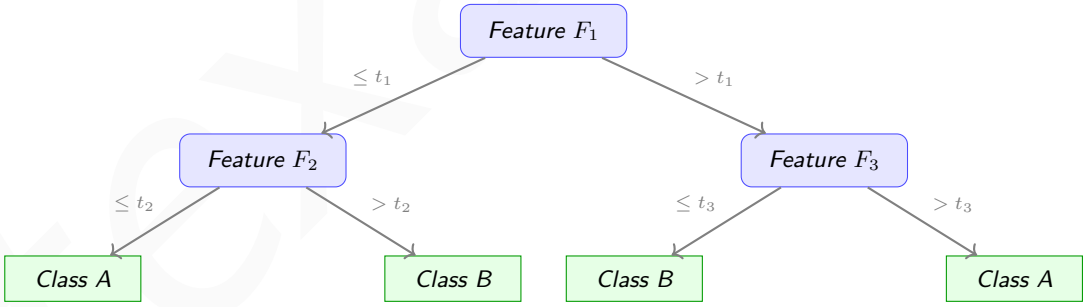
The histogram above represents `Age` *discretized into equal-width bins of width 10. Which bin contains the* ***modal class****?*

(A) 10–20

(B) 20–30

(C) 30–40

(D) 40–50



Problem 94 ***[MCQ]***

In the decision tree shown above, which feature is considered ***most informative*** *for the classification task, based on the tree's structure?*

(A) F_2*, because it appears twice in the tree.*

(B) F_1*, because it is the root node and therefore selected first due to its highest information gain.*

(C) F_3*, because it correctly classifies more samples.*

(D) All features are equally informative.

Problem 95 ***[NAT]*** *A dataset has 5 features:* F_1, F_2, F_3, F_4, F_5. *A forward selection procedure evaluated subsets as follows:*

Subset Added	*Accuracy*
$\{F_3\}$	*72%*
$\{F_3, F_1\}$	*78%*
$\{F_3, F_1, F_5\}$	*81%*
$\{F_3, F_1, F_5, F_2\}$	*80%*

If the stopping criterion is accuracy must not decrease, *the* ***number of features*** *selected is* _____.

Problem 96 ***[MSQ]*** *Which of the following statements correctly distinguish* ***data reduction*** *from* ***data transformation****?*

(A) Data reduction decreases the volume of data (fewer tuples, attributes, or a compressed representation); data transformation changes the form of the data without necessarily reducing its size.

(B) Normalization is a data transformation technique, not a data reduction technique.

(C) PCA can be classified under both data reduction (dimensionality reduction) and lossy data compression.

(D) Sampling is a transformation technique that normalizes the data distribution.

Problem 97 ***[MCQ]*** *Consider a Huffman tree built from symbol frequencies* $\{a : 1, b : 1, c : 2, d : 4\}$*. The* ***total weighted path length*** *(i.e.,* $\sum f_i \cdot \ell_i$*) of the optimal Huffman tree is:*

(A) 14

(B) 17

(C) 19

(D) 16

Problem 98 ***[MSQ]*** *Which of the following normalization-related statements are* ***TRUE****?*

(A) After z-score normalization, the mean of the transformed attribute is 0 and variance is 1.

(B) After min-max normalization to $[0, 1]$*, the minimum value maps to 0 and maximum to 1.*

(C) Decimal scaling changes the relative order of values.

(D) Z-score normalization can produce values outside $[-1, 1]$*.*

Problem 99 ***[MCQ]*** *Which of the following* ***correctly*** *identifies a scenario where* ***RLE is inefficient****?*

(A) A black-and-white scanned document with large white margins.

(B) A dense random noise image where pixel values alternate rapidly.

(C) A binary attribute column in a data warehouse with long runs of 0s.

(D) A video frame background region with a uniform colour.

Problem 100 ***[NAT]*** *A dataset contains* $n = 1000$ *records. After applying proportional stratified sampling:*

Stratum	*Size*	*Sample size*
1	*300*	*60*
2	*200*	*40*
3	x	*100*

The value of x *(stratum 3 size) is* _____.

Problem 101 ***[NAT]*** *The following bitmap (read row by row, left to right) represents a 4×8 binary image:*

Row 1	0	0	0	1	1	1	0	0
Row 2	0	0	1	1	1	0	0	0
Row 3	1	1	1	0	0	0	1	1
Row 4	1	1	0	0	1	1	1	0

*Apply standard **run-length encoding** to the entire 32-bit stream as $(value,\ run\text{-}length)$ pairs. The **total number of bits** required to store the RLE output, if each pair uses **1 bit** for the value and **4 bits** for the run-length is _____ bits.*

Problem 102 ***[NAT]*** *A text file contains exactly **six symbols** with the following frequencies:*

Symbol	*P*	*Q*	*R*	*S*	*T*	*U*
Frequency	*2*	*3*	*7*	*8*	*15*	*25*

*Construct the **optimal Huffman tree** (break ties by placing the new merged node on the right/higher position). The **total number of merge operations** required to build the Huffman tree is _____.*

Problem 103 ***[NAT]*** *A communication channel transmits messages using symbols $\{A, B, C, D, E\}$ with probabilities:*

Symbol	*A*	*B*	*C*	*D*	*E*
Probability	*0.40*	*0.20*	*0.15*	*0.15*	*0.10*

*A message of **1000 symbols** drawn i.i.d. from the above distribution is encoded using this Huffman code. The **expected total number of bits** in the encoded message is _____.*

Problem 104 ***[NAT]*** *The **LZW algorithm** is initialized with a dictionary containing single-character entries:*

$$A \to 1, \quad B \to 2, \quad C \to 3$$

*New entries are assigned codes starting from **4**, in the order they are discovered.*

*Let **sum of output codes** be S and the **final dictionary size** (total entries including the 3 initial ones) is D, produced by LZW encoding of `ABABCABABC`. Then $S + D$ is: _____.*

Problem 105 ***[NAT]*** *The LZW dictionary is initialized with:*

$$A \to 1, \quad B \to 2, \quad C \to 3, \quad D \to 4$$

*New codes are assigned starting from **5**. The string to encode is: $\mathbf{ABCDABCDAB}$. Suppose the dictionary codes use **4 bits** each. The total output size in bits is _____.*

Problem 106 ***[NAT]*** *Three attributes of a dataset are to be min-max normalized to $[0,\ 1]$:*

Attribute	***Min***	***Max***	***Value to Normalize***
X_1	-50	150	25
X_2	0.001	0.999	0.5
X_3	1000	9000	4600

Let (Y_1, Y_2, Y_3) be the attribute values after normalization. The Manhattan distance (L_1 norm) of the point (Y_1, Y_2, Y_3) from the origin $(0, 0, 0)$ is _____.(rounded to four decimal places).

2.6 Try it Yourself

Exercise 26 *A dataset has attributes: {Age, Salary, ID, ZipCode}. Which attribute is most likely to be removed during preprocessing?*

(A) Age

(B) Salary

(C) ID

(D) ZipCode

Exercise 27 *Which of the following is NOT a goal of data preprocessing?*

(A) Improve data quality

(B) Reduce dimensionality

(C) Increase noise

(D) Improve model accuracy

Exercise 28 *A dataset has 20% missing values in attribute A. Which strategy is MOST inappropriate?*

(A) Replace with mean

(B) Ignore tuples

(C) Replace with most probable value using model

(D) Replace with random values

Exercise 29 *Which of the following methods can be used to handle missing values? (MSQ)*

(A) Mean substitution

(B) Regression imputation

(C) Clustering-based filling

(D) Dropping all attributes

Exercise 30 *If missing values are NOT random, which method is more appropriate?*

(A) Mean substitution

(B) Model-based imputation

(C) Random filling

(D) Ignoring records

Exercise 31 *Which technique reduces noise by grouping values into bins?*

(A) Regression

(B) Clustering

(C) Binning

(D) Normalization

Exercise 32 *Which smoothing method replaces values using bin boundaries?*

(A) Mean smoothing

(B) Median smoothing

(C) Boundary smoothing

(D) Regression smoothing

Exercise 33 *Binning is MOST effective when:*

(A) Data is categorical

(B) Data is continuous

(C) Data is missing

(D) Data is binary

Exercise 34 *Regression-based noise reduction assumes:*

(A) Random grouping

(B) Underlying functional relationship

(C) Discrete bins

(D) Uniform distribution

Exercise 35 *Min-max normalize value 50 from range [0,100] to [0,1].* ***Answer type: NAT***

Exercise 36 *Z-score normalization: If mean=20, std=5, value=30, find normalized value.* ***Answer type: NAT***

Exercise 37 *Decimal scaling: Normalize 987 such that* $|v'| < 1$. *Find j.* ***Answer type: NAT***

Exercise 38 *Which normalization is affected by outliers?*

(A) Z-score

(B) Decimal scaling

(C) Min-max

(D) None

Exercise 39 *In simple random sampling without replacement, probability of selecting same element twice is:*

(A) 0

(B) 1

(C) Depends

(D) Infinite

Exercise 40 *Stratified sampling ensures:*

(A) Equal probability

(B) Representation of subgroups

(C) Randomness only

(D) Bias

Exercise 41 *Dataset size = 1000. Sample size = 100. Stratified proportional sampling. Class A=400. Samples from A?* ***Answer type: NAT***

Exercise 42 *Sampling with replacement implies: (MSQ)*

(A) Same element may appear multiple times

(B) Sample size reduces

(C) *Independence*

(D) *No duplicates*

Exercise 43 *Apply RLE on: AAAABBBCCDAA. What is encoded length?* ***Answer type: NAT***

Exercise 44 *Which compression is lossless? (MSQ)*

(A) *Huffman*

(B) *RLE*

(C) *LZW*

(D) *Wavelet*

Exercise 45 *Huffman coding guarantees:*

(A) *Fixed length codes*

(B) *Optimal prefix codes*

(C) *Random encoding*

(D) *Equal frequency*

Exercise 46 *Which is NOT lossless?*

(A) *LZW*

(B) *Huffman*

(C) *Wavelet*

(D) *RLE*

Exercise 47 *Data: $\{1,2,3,4,5,6,7,8\}$. Equal-width binning into 2 bins. Width?* ***Answer type: NAT***

Exercise 48 *Equal-frequency binning ensures:*

(A) *Equal width*

(B) *Equal number of points*

(C) *Equal variance*

(D) *Equal mean*

Exercise 49 *Bin values: $\{2,4,6\}$. Mean smoothing result?* ***Answer type: NAT***

Exercise 50 *Which discretization is data-driven?*

(A) *Domain binning*

(B) *Equal width*

(C) *Histogram-based*

(D) *Manual*

Exercise 51 *PCA maximizes:*

(A) *Mean*

(B) *Variance*

(C) *Distance*

(D) Error

Exercise 52 *Principal components are:*

(A) Correlated

(B) Orthogonal

(C) Random

(D) Parallel

Exercise 53 *Which step is required before PCA?*

(A) Sorting

(B) Normalization

(C) Compression

(D) Sampling

Exercise 54 *First principal component captures:*

(A) Least variance

(B) Maximum variance

(C) Mean

(D) Noise

Exercise 55 *Forward selection starts with:*

(A) All features

(B) No features

(C) Random features

(D) Half features

Exercise 56 *Backward elimination removes:*

(A) Best features

(B) Worst features

(C) Random features

(D) None

Exercise 57 *Decision tree feature selection uses:*

(A) Entropy

(B) Variance

(C) Mean

(D) Distance

Exercise 58 *Histogram equal-width: Range=100, bins=5. Width?* ***Answer type: NAT***

Exercise 59 *MaxDiff histogram focuses on:*

(A) Largest differences

(B) Equal frequency

(C) Equal width

(D) Random bins

Exercise 60 *V-optimal histogram minimizes:*

(A) Variance within bins

(B) Width

(C) Count

(D) Mean

Exercise 61 *Which combination reduces dimensionality? (MSQ)*

(A) PCA

(B) Feature selection

(C) Sampling

(D) Compression

Exercise 62 *Which is TRUE about normalization? (MSQ)*

(A) Required for distance-based models

(B) Removes noise

(C) Scales data

(D) Removes outliers

Exercise 63 *If all values are same, z-score normalization results in:*

(A) 0

(B) 1

(C) Undefined

(D) Infinite

Exercise 64 *Sampling reduces:*

(A) Noise

(B) Size

(C) Dimensions

(D) Features

Exercise 65 *Which technique is BOTH compression and dimensionality reduction?*

(A) PCA

(B) Huffman

(C) RLE

(D) Sampling

Exercise 66 *Given dataset: {10,20,30,40,50}. Median?* ***Answer type: NAT***

Exercise 67 *Which method is robust to outliers?*

(A) Mean

(B) Median

(C) Min-max

(D) Z-score

Exercise 68 *Which preprocessing step is mandatory for KNN?*

(A) Sampling

(B) Normalization

(C) Compression

(D) Discretization

Exercise 69 *Entropy-based splitting is used in:*

(A) PCA

(B) Decision trees

(C) Sampling

(D) Compression

Exercise 70 *Which is unsupervised? (MSQ)*

(A) PCA

(B) Clustering

(C) Regression

(D) Decision tree

Exercise 71 ***[MCQ]***

A dataset contains an attribute `Temperature` *with values:*

$$12,\ 14,\ 15,\ 18,\ 19,\ 100$$

A preprocessing engineer replaces missing values using the ***mean*** *of the attribute.*
Which statement best explains why this strategy may be problematic for this dataset?

(A) Mean imputation cannot be applied to numerical attributes.

(B) The attribute contains an outlier that can significantly distort the mean.

(C) Mean imputation changes the dimensionality of the dataset.

(D) The dataset must first be normalized before computing the mean.

Exercise 72 ***[MSQ]***

Which of the following preprocessing operations are generally considered ***lossy****?*

(A) Smoothing by bin means

(B) Huffman encoding

(C) PCA retaining only top-k components

(D) Equal-width discretization

Exercise 73 ***[NAT]***

A dataset contains the following sorted values:

$$5,\ 8,\ 12,\ 14,\ 17,\ 21,\ 24,\ 27,\ 30$$

Equal-frequency binning with 3 bins is applied.
After smoothing by ***bin medians****, the transformed value corresponding to the original value* 24 *is* _____.

Exercise 74 ***[MCQ]***

Consider the following statements regarding PCA:

1. *Principal components are mutually orthogonal.*
2. *PCA always preserves interpretability of original attributes.*
3. *PCA maximizes variance along projected directions.*

Which one of the following is correct?

(A) Only 1 and 2 are true

(B) Only 1 and 3 are true

(C) Only 2 and 3 are true

(D) 1, 2 and 3 are true

Exercise 75 ***[NAT]***

A value $x = 45$ *from an attribute with minimum value* 20 *and maximum value* 70 *is normalized to the range* $[-1, 1]$ *using min-max normalization.*
The normalized value is _____.

Exercise 76 ***[MCQ]***

Which of the following situations is most suitable for ***stratified sampling****?*

(A) The dataset is already uniformly distributed across all classes.

(B) Minority classes are important and must be adequately represented.

(C) The dataset contains only numerical attributes.

(D) The dataset size is very small.

Exercise 77 ***[MSQ]***

Which of the following statements regarding Huffman coding are TRUE?

(A) Huffman codes are prefix-free.

(B) Huffman coding guarantees fixed-length encoding.

(C) Symbols with larger frequencies tend to get shorter codewords.

(D) Huffman coding is always lossy.

Exercise 78 ***[NAT]***

Consider the binary string:

$$11111000000000011111000$$

Using standard Run-Length Encoding (RLE), the number of runs generated is _____.

Exercise 79 ***[MCQ]***

In equal-width discretization, the major effect of a large outlier is that it:

(A) Causes all bins to contain equal frequencies

(B) Stretches the overall range, potentially producing sparse bins

(C) Makes the histogram V-optimal

(D) Eliminates skewness automatically

Exercise 80 ***[NAT]***

The following values are discretized using equal-width binning into 4 *bins:*

$$2,\ 4,\ 6,\ 8,\ 10,\ 12,\ 14,\ 16,\ 18,\ 20$$

The width of each bin is _____.

Exercise 81 ***[MCQ]***

Backward elimination in feature subset selection starts with:

(A) An empty feature set

(B) A random subset of features

(C) The complete set of features

(D) Only the highest variance feature

Exercise 82 ***[MSQ]***

Which of the following techniques may reduce the impact of noisy data?

(A) Binning

(B) Regression smoothing

(C) Clustering-based outlier detection

(D) Decimal scaling

Exercise 83 ***[NAT]***

A dataset contains 5 classes with tuple counts:

$$100,\ 200,\ 300,\ 250,\ 150$$

A proportional stratified sample of size 200 *is drawn.*
The number of tuples selected from the class containing 300 *tuples is* _____.

Exercise 84 ***[MCQ]***

Which one of the following is TRUE regarding z-score normalization?

(A) It always maps values to the range $[0, 1]$

(B) It depends only on minimum and maximum values

(C) It standardizes data using mean and standard deviation

(D) It cannot produce negative values

Exercise 85 ***[MCQ]***

A compression algorithm replaces repeated occurrences of strings using references to a dynamically constructed dictionary.

The algorithm described above is:

(A) Huffman coding

(B) Decimal scaling

(C) LZW compression

(D) Equal-frequency binning

Exercise 86 ***[NAT]***

For the values

$$3,\ 5,\ 7,\ 9,\ 11$$

the mean is used for smoothing within the bin.

The smoothed value replacing 11 *is* _____.

Exercise 87 ***[MSQ]***

Which of the following are advantages of dimensionality reduction?

(A) Reduced storage requirements

(B) Faster learning algorithms

(C) Reduced risk of overfitting

(D) Guaranteed increase in classification accuracy

Exercise 88 ***[MCQ]***

Which histogram method places bucket boundaries at locations where adjacent differences between sorted values are largest?

(A) Equal-width histogram

(B) Equal-frequency histogram

(C) MaxDiff histogram

(D) V-Optimal histogram

Exercise 89 ***[NAT]***

Consider the sorted values:

$$1,\ 2,\ 3,\ 20,\ 21,\ 22,\ 23$$

Using MaxDiff histogram partitioning with 2 buckets, the partition boundary is placed immediately after the value _____.

Exercise 90 ***[MCQ]***

Which statement best distinguishes data transformation from data reduction?

(A) Transformation changes representation, whereas reduction decreases data volume.

(B) Reduction changes representation, whereas transformation decreases dimensionality.

(C) Transformation is always lossless, reduction is always lossy.

(D) Reduction applies only to numerical data.

Exercise 91 ***[MCQ]***

Suppose PCA retains only the first principal component from a 20-dimensional dataset.
Which statement is MOST accurate?

(A) All information is preserved exactly.

(B) The retained component captures the direction of maximum variance.

(C) The dimensionality increases from 20 to 21.

(D) PCA removes all correlation and all noise completely.

Exercise 92 ***[MSQ]***

Which of the following normalization methods are sensitive to extreme outliers?

(A) Min-max normalization

(B) Decimal scaling

(C) Z-score normalization

(D) Median-based scaling

Exercise 93 ***[NAT]***

A dataset contains the values:

$$10,\ 20,\ 30,\ 40,\ 50,\ 60$$

Equal-frequency discretization with 3 bins is applied.
The number of values in each bin is _____.

Exercise 94 ***[MCQ]***

In Huffman coding, the weighted path length is minimized because:

(A) More frequent symbols are placed closer to the root

(B) All leaves occur at identical depths

(C) Rare symbols are removed

(D) The tree is always perfectly balanced

Exercise 95 ***[MCQ]***

Which preprocessing technique is MOST directly intended to reduce the number of attributes in a dataset?

(A) Sampling

(B) PCA

(C) Huffman coding

(D) Equal-frequency binning

Exercise 96 ***[NAT]***

The values

$$15,\ 18,\ 22,\ 25,\ 30$$

form a single bin.

After smoothing by bin boundaries, the transformed value corresponding to 22 *is* _____.

Exercise 97 ***[MCQ]***

Which statement about equal-frequency binning is TRUE?

(A) All bins have equal width.

(B) All bins contain approximately equal numbers of tuples.

(C) It cannot handle skewed distributions.

(D) Bin boundaries are independent of the data distribution.

Exercise 98 ***[MSQ]***

Which of the following are examples of numerosity reduction?

(A) Sampling

(B) Histograms

(C) Regression models

(D) PCA

Exercise 99 ***[NAT]***

A value $x = 250$ *is decimal-scaled using:*

$$v' = \frac{v}{10^j}$$

such that $|v'| < 1$.

The minimum integer value of j *is* _____.

Exercise 100 ***[MCQ]***

A V-Optimal histogram attempts to:

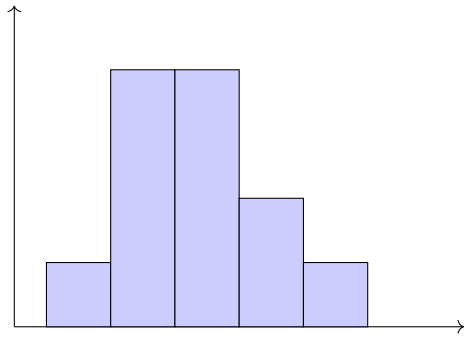
(A) Maximize bucket width

(B) Minimize variance within buckets

(C) Ensure equal frequencies in all buckets

(D) Place boundaries at largest adjacent gaps

Exercise 101 ***[MCQ]***



The histogram shown above is MOST likely to represent:

(A) A highly right-skewed distribution

(B) A symmetric distribution

(C) Uniform random noise

(D) Perfectly equal-frequency bins

Exercise 102 ***[MSQ]***

Consider the following statements about sampling with replacement:

(A) A tuple may appear multiple times in the sample.

(B) The effective population size decreases after each selection.

(C) Selections are independent.

(D) The same tuple can never be selected twice.

Exercise 103 ***[NAT]***

A dataset initially contains 80 attributes. After dimensionality reduction, only 20 attributes remain. The percentage reduction in dimensionality is _____.

Exercise 104 ***[MCQ]***

Which preprocessing operation is generally performed FIRST in a practical data mining pipeline?

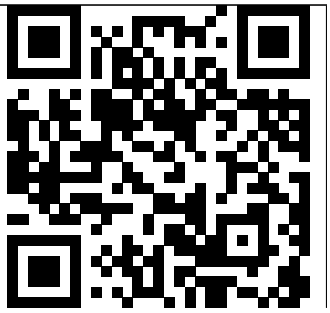
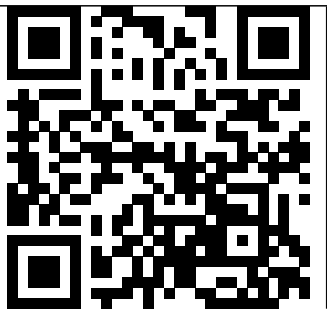
(A) PCA

(B) Normalization

(C) Missing value handling

(D) Huffman encoding

2.7 YouTube Links and QR Codes

Lecture	Details	YouTube Link	QR Code
04	Data Cleaning — Missing Values, Noise & Outliers — Data Preprocessing	https://youtu.be/0RQJ4dpmUV0	
05	Data Reduction — Dimensionality Reduction— PCA, Attribute Subset, Decision Trees	https://youtu.be/Mnc_0_vrLtc	
06	Data Reduction — Numerosity Reduction & Sampling — Data Warehousing	https://youtu.be/LEHO47htStY	
07	Data Compression — Lossless (RLE, LZW, Huffman) & Lossy (PCA) — Data Warehousing	https://youtu.be/rK6YOhT9yA0	
08	Data Transformation — Normalization (Min-Max, Z-Score, Decimal Scaling) — Data Warehousing	https://youtu.be/2qs14ERx-qM	

09	Data Discretization — Binning Techniques Explained — Data Warehousing	https://youtu.be/51RgEGfOjqM	
10	Data Discretization — Histogram Analysis — Data Warehousing	https://youtu.be/oc3yyCz1Td4	
11	Problem Solving on Data Preprocessing — Data Warehousing	https://youtu.be/VpH49WkvPNE	

Chapter 3

Data Warehousing and Online Analytical Processing

3.1 Data Warehouse: Basic Concepts

Data Warehouse

A **data warehouse** is a **subject-oriented, integrated, time-variant, and non-volatile** collection of data that supports **management decision making**.

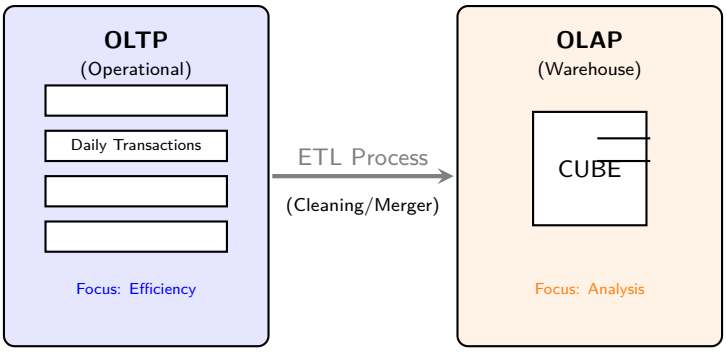
- **Subject-oriented:** Organized around major subjects (sales, customers, products) not transactions
- **Integrated:** Data from multiple heterogeneous sources consolidated into a unified format
- **Time-variant:** Stores **historical data** — every record has a time element
- **Non-volatile:** Data is **loaded and accessed**, never updated or deleted in place

3.1.1 OLTP versus OLAP

The Fundamental Difference

The major distinction lies between **Operational** systems and **Informational** systems:

- **OLTP (Online Transaction Processing):** Systems used for day-to-day operations (e.g., banking, retail checkout).
- **OLAP (Online Analytical Processing):** Systems used for data analysis and decision making (e.g., trend analysis, sales forecasting).



Data Warehouse vs Operational Database

Property	Operational DB (OLTP)	Data Warehouse (OLAP)
Purpose	Day-to-day transactions	Decision support, analysis
Data	Current, detailed	Historical, summarized
Operations	INSERT, UPDATE, DELETE	SELECT, aggregation
Query type	Simple, frequent, short	Complex, infrequent, long
Users	Clerks, front-end apps	Analysts, executives
Schema	Normalized (3NF)	Denormalized (Star, Snowflake)
Size	GB	TB to PB
Update	Continuous (real-time)	Periodic (batch ETL)

Example 81: OLTP vs OLAP in Retail

- **OLTP Task:** A customer buys a laptop. The database updates the stock count from 10 to 9 and records the payment. *(High frequency, row-level access)*
- **OLAP Task:** A manager asks, "What were the sales trends for laptops in South India over the last 3 summers compared to the rainy season?" *(Low frequency, massive data aggregation)*

Example 82: Step-by-Step Logic: The "ATM" Analogy

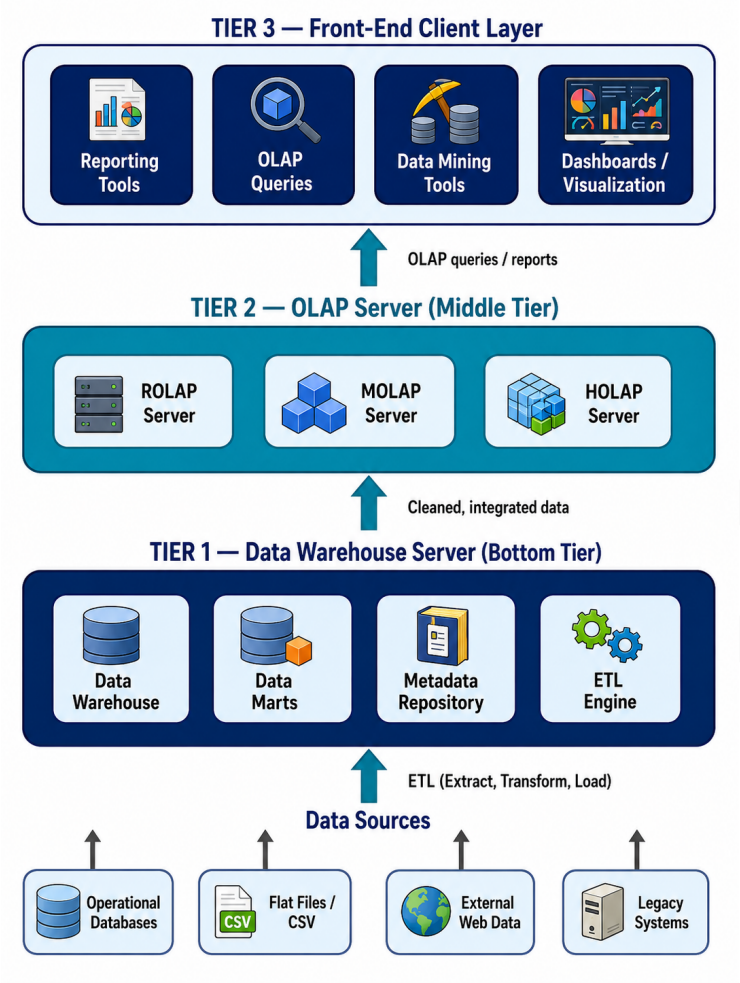
1. **Step 1 (OLTP):** You withdraw money from an ATM. This is a single, fast transaction that must be consistent (ACID).
2. **Step 2 (Data Flow):** At the end of the day, your transaction data is moved to a central Warehouse.
3. **Step 3 (OLAP):** The bank analyzes the "Withdrawal Patterns" of all customers in your city to decide if they need to install more ATMs in that area.

3.1.2 Data Warehousing: A Multitiered Architecture

Three-Tier Data Warehouse Architecture

The standard data warehouse architecture has **three tiers**:

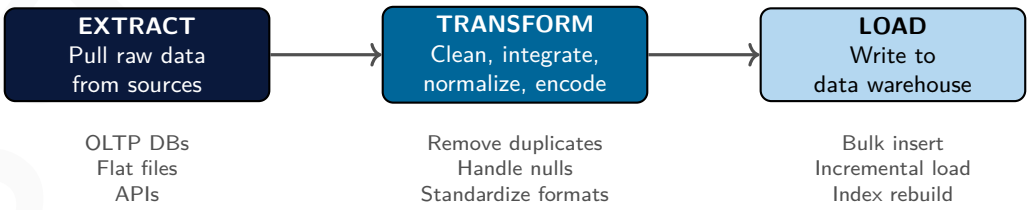
- **Tier 1 — Bottom Tier:** Data sources and data warehouse server (ETL layer)
- **Tier 2 — Middle Tier:** OLAP server (multidimensional analysis engine)
- **Tier 3 — Top Tier:** Front-end client tools (reporting, mining, visualization)



Tier 1 — Bottom Tier (Data Warehouse Server)

The **foundation layer** responsible for **data storage and ETL processing**.

- **Data Sources:** Operational databases, flat files, external feeds, legacy systems
- **ETL Engine:** Extract, Transform, Load — the data integration pipeline
- **Data Warehouse Server:** Central relational database storing integrated, historical data
- **Data Marts:** Smaller, subject-specific subsets of the warehouse
- **Metadata Repository:** Stores information *about* the data (schema, lineage, transformations)



ETL Extract Transform Load

- **Extract:** Pull data from heterogeneous sources — OLTP databases, CSV files, APIs, web scraping
- **Transform:** Apply data cleaning and integration rules:

  - Handle **missing values** and **duplicates**
  - **Standardize formats**: Date formats, unit conversion, encoding
  - **Aggregate** data to required granularity
  - **Resolve conflicts**: Different source systems may store same data differently
- **Load:** Insert transformed data into the warehouse — full or incremental load
- ETL runs on a **periodic schedule** (nightly, weekly) — not real-time

Example 83: ETL in a Retail Chain

Sources:

- Store A database: Sales in USD, date format MM/DD/YYYY
- Store B database: Sales in INR, date format DD-MM-YYYY
- Online portal: JSON logs with Unix timestamps

Transform steps:

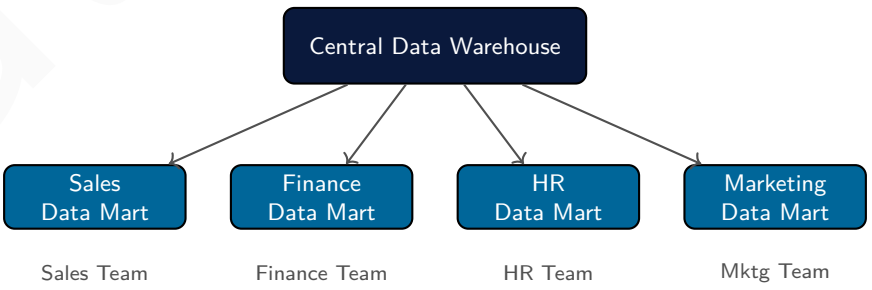
- Convert all currencies to INR using exchange rates
- Standardize all dates to YYYY-MM-DD format
- Parse Unix timestamps to readable date-time
- Remove duplicate transaction IDs
- Fill missing product categories using product lookup table

Load: Nightly batch job inserts transformed records into the central warehouse.

Data Mart

A **data mart** is a **smaller, subject-specific** subset of a data warehouse.

- Serves a specific **business unit** or department: Sales, Finance, HR, Marketing
- Faster to build and query than the full warehouse
- Two types:
  - **Dependent data mart:** Created from the central data warehouse (top-down)
  - **Independent data mart:** Built directly from source systems without a central warehouse



Metadata Repository

The **metadata repository** stores information *about* the data warehouse — it is the "data about data."

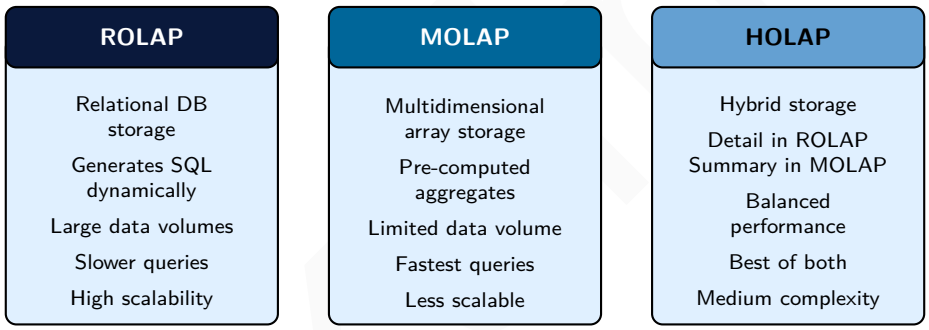
- **Business metadata:** Definitions, business rules, ownership
- **Technical metadata:** Schema, data types, ETL mappings, transformations applied
- **Operational metadata:** ETL job history, load timestamps, data quality logs
- Enables **data lineage**: track where each piece of data came from and how it was transformed
- Critical for **auditing, debugging, and governance**

Tier 2 — Middle Tier (OLAP Server)

The **analytical engine** that processes multidimensional queries on warehoused data.

Three types of OLAP servers:

- **ROLAP (Relational OLAP):** Implements multidimensional data on top of a relational database using SQL
- **MOLAP (Multidimensional OLAP):** Stores data in a proprietary **multidimensional array** (data cube) directly
- **HOLAP (Hybrid OLAP):** Combines ROLAP and MOLAP — detailed data in relational, aggregates in multidimensional



ROLAP vs MOLAP — Detailed Comparison

Property	ROLAP	MOLAP
Storage	Relational tables	Multidimensional arrays
Query language	SQL (extended)	MDX or proprietary
Query speed	Slower (computed on-fly)	Faster (pre-aggregated)
Data volume	Very large (TB scale)	Limited (GB scale)
Aggregation	Computed at query time	Pre-computed and stored
Sparsity handling	Good (only stores facts)	Poor (stores all cells)
Scalability	High	Low
Example systems	Microsoft SQL Server	Microsoft SSAS, Essbase

Tier 3 — Top Tier (Front-End Client Tools)

The **user-facing layer** — tools that query, visualize, and mine the data warehouse.

- **Reporting tools:** Generate fixed-format reports for management (Crystal Reports, SSRS)
- **OLAP query tools:** Ad-hoc multidimensional analysis (drill-down, slice, dice)
- **Data mining tools:** Discover patterns, build predictive models

- **Dashboards:** Real-time KPI visualization (Tableau, Power BI, Looker)
- **Statistical tools:** In-depth statistical analysis (R, Python)

Example 84: Three-Tier Architecture in a Banking System

Tier 1 — Data Sources and Warehouse Server:

- Sources: Core banking system (Oracle DB), ATM logs (CSV), Mobile app (JSON API), Credit card system (MySQL)
- ETL: Nightly batch — extract transactions, standardize currency, clean nulls, load into central warehouse
- Warehouse: Stores 10 years of transaction history (Star Schema with Time, Branch, Product, Customer dimensions)

Tier 2 — OLAP Server:

- MOLAP cube pre-aggregated for: monthly deposits by branch, loan defaults by region, product-wise revenue
- Supports roll-up (branch $\rightarrow$ zone $\rightarrow$ region), drill-down, slice (product $=$ home loan)

Tier 3 — Client Tools:

- Executives: Power BI dashboard showing real-time KPIs
- Risk analysts: Python-based data mining for default prediction
- Compliance: Crystal Reports for audit reports (fixed format)
- Branch managers: Ad-hoc OLAP queries for local performance analysis

Summary — Three-Tier Architecture

Tier	Name	Components	Function
1	Bottom	Sources, ETL, DW Server, Data Marts	Store and integrate
2	Middle	ROLAP / MOLAP / HOLAP Server	Analyze and aggregate
3	Top	Reports, Mining, Dashboards, Queries	Present and discover

3.2 Data Warehouse Modeling: Data Cube and OLAP

3.2.1 Data Cube: A Multidimensional Data Model

Multidimensional Data Model

A **multidimensional data model** views data as a **data cube** — a structure that organizes data by **dimensions** and stores **measures** at each intersection.

- Designed for **OLAP** and data warehouse analysis
- Replaces the flat 2D table view with an n-dimensional perspective
- Each **point** in the cube represents an aggregated measure for a unique combination of dimension values
- Enables efficient **roll-up, drill-down, slice, and dice** operations

Formal Definition of a Data Cube

An n-dimensional data cube is defined by:

$$\text{Cube} = (D_1,\ D_2,\ \ldots,\ D_n,\ M)$$

where:

- $D_i = i$-th dimension with domain $\text{dom}(D_i)$
- $M =$ measure function mapping each cell $(d_1, d_2, \ldots, d_n)$ to a numeric value
- Total number of cells $= |\text{dom}(D_1)| \times |\text{dom}(D_2)| \times \cdots \times |\text{dom}(D_n)|$
- Many cells may be **empty (sparse)** if not all combinations exist in the data

Core Components

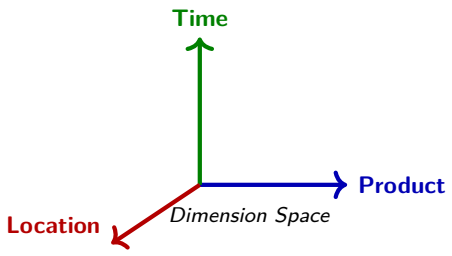
- **Dimension:** An attribute or perspective used to organize data
  Examples: `Time`, `Location`, `Product`
- **Dimension Hierarchy:** Levels of abstraction within a dimension
  `Time`: Day $\rightarrow$ Month $\rightarrow$ Quarter $\rightarrow$ Year
- **Measure:** A numeric value stored at each cell of the cube
  Examples: `Sales`, `Profit`, `Count`, `Average Price`
- **Cell:** One point in the cube identified by a unique combination of dimension values
  Example: (Q1, Bengaluru, Electronics) $\rightarrow$ Sales $= 120$
- **Cuboid:** One particular aggregation of the full cube across a subset of dimensions

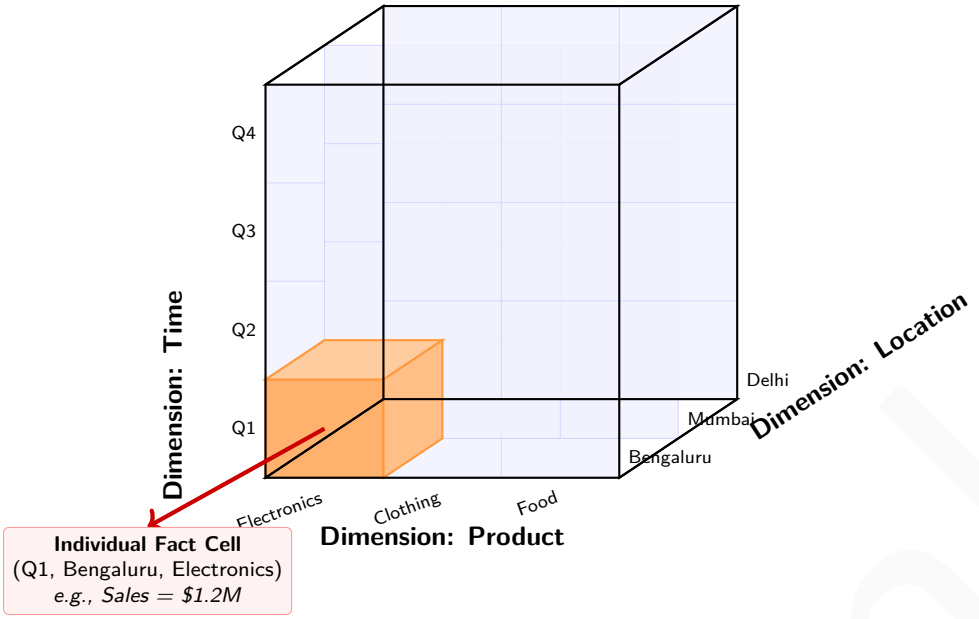
Example 85: Formal Structure of a 3D Sales Cube

Dimensions and their domains:

Dimension	Domain Values	Cardinality
`Time`	Q1, Q2, Q3, Q4	4
`Location`	Bengaluru, Mumbai, Delhi	3
`Product`	Electronics, Clothing, Food	3

- Total cells $= 4 \times 3 \times 3 = 36$
- Measure: `Sales (lakhs)` — one value per cell
- If only 20 of 36 combinations have actual sales data $\rightarrow$ cube is **sparse** (44% empty)





Example 86: Find Domain Size from Cell Count

Q: A 3-dimensional data cube has Time(12), Location(x), and Product(50) as dimensions. If the total number of cells is 36,000, what is the domain size of the Location dimension?

$$\text{Total cells} = 12 \times x \times 50 = 36{,}000$$

$$600x = 36{,}000 \implies x = 60$$

Answer: The Location dimension has $\mathbf{60}$ distinct values (e.g., 60 cities).

Q: A 4-dimensional cube has equal domain size d for all dimensions. If the total number of cells is 10,000, what is d?

$$d^4 = 10{,}000 \implies d = \sqrt[4]{10{,}000} = 10$$

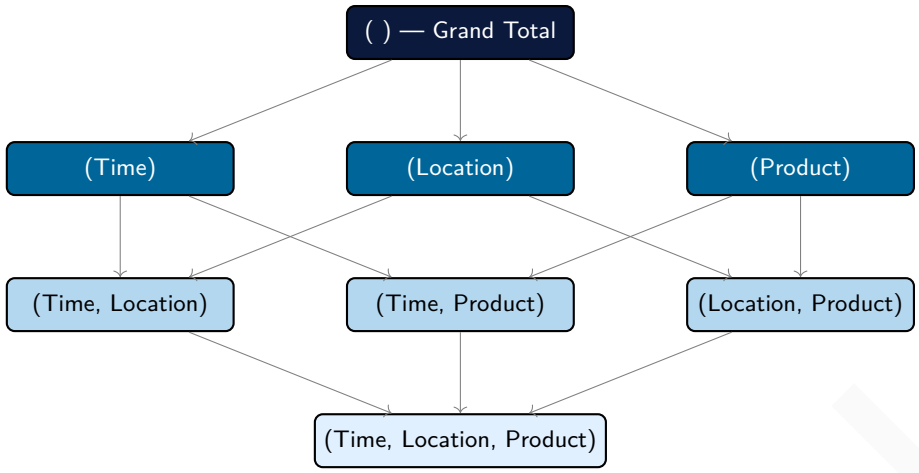
Answer: Each dimension has $\mathbf{10}$ distinct values.

Cuboid and Cuboid Lattice

For an n-dimensional cube, there exist 2^n **cuboids** — each representing a different level of aggregation.

- **Base cuboid:** All n dimensions present — most detailed, lowest aggregation
- **Apex cuboid (0-D):** All dimensions aggregated — single grand total value
- Each cuboid corresponds to a **GROUP BY** query on a subset of dimensions
- The set of all 2^n cuboids forms a **cuboid lattice**

$$\text{Number of cuboids} = 2^n$$



Example 87: Cuboid Lattice — SQL Equivalents

Each cuboid corresponds to a SQL `GROUP BY` query on a different subset of dimensions:

Cuboid	SQL Equivalent
( ) — Apex	`SELECT SUM(Sales) FROM Cube`
(Time)	`GROUP BY Time`
(Location)	`GROUP BY Location`
(Product)	`GROUP BY Product`
(Time, Location)	`GROUP BY Time, Location`
(Time, Product)	`GROUP BY Time, Product`
(Location, Product)	`GROUP BY Location, Product`
(Time, Location, Product)	`GROUP BY Time, Location, Product`

Cuboid Count by Level — General Formula

For an n-dimensional cube, the number of cuboids at each level of the lattice is given by the binomial coefficient:

$$\text{Number of } k\text{-dimensional cuboids} = \binom{n}{k} = \frac{n!}{k!\,(n-k)!}$$

The total across all levels:

$$\sum_{k=0}^{n} \binom{n}{k} = 2^n$$

This follows directly from the Binomial Theorem: $\sum_{k=0}^{n} \binom{n}{k} 1^k 1^{n-k} = 2^n$.

n	0-D	1-D	2-D	3-D	4-D	5-D	Total 2^n
2	1	2	1	—	—	—	4
3	1	3	3	1	—	—	8
4	1	4	6	4	1	—	16
5	1	5	10	10	5	1	32

Example 88: Number of Cuboids

Q: A data cube has 5 dimensions: Time, Location, Product, Customer, Channel.
How many cuboids does the full data cube contain?

$$\text{Number of cuboids} = 2^5 = 32$$

- 1 apex cuboid (0-D)
- $\binom{5}{1} = 5$ one-dimensional cuboids
- $\binom{5}{2} = 10$ two-dimensional cuboids
- $\binom{5}{3} = 10$ three-dimensional cuboids
- $\binom{5}{4} = 5$ four-dimensional cuboids
- $\binom{5}{5} = 1$ base cuboid (5-D)
- Total: $1 + 5 + 10 + 10 + 5 + 1 = 32$ cuboids

Example 89: Hierarchy-Aware Cell Count — Total Cells at Each Level

Given a Time dimension with the hierarchy: Day(365) → Month(12) → Quarter(4) → Year(1), and a Location dimension: City(50) → State(10) → Country(3) → ALL, combined with a Product dimension with a flat domain of 200 items, calculate the number of cells for each combination of hierarchy levels:

Time Level	Location Level	Product Level	Total Cells
Day (365)	City (50)	Product (200)	$365 \times 50 \times 200 = 3{,}650{,}000$
Month (12)	City (50)	Product (200)	$12 \times 50 \times 200 = 120{,}000$
Quarter (4)	State (10)	Product (200)	$4 \times 10 \times 200 = 8{,}000$
Year (1)	Country (3)	Product (200)	$1 \times 3 \times 200 = 600$
ALL (1)	ALL (1)	ALL (1)	$1 \times 1 \times 1 = 1$ (Apex)
Day (365)	Country (3)	Product (200)	$365 \times 3 \times 200 = 219{,}000$

Example 90: Cuboid Counting — Practice Problems with Solutions

A cube has 6 dimensions. How many 3-D cuboids exist? What is the total cuboid count?

$$\binom{6}{3} = \frac{6!}{3!\,3!} = 20 \quad \text{three-dimensional cuboids}$$

$$\text{Total cuboids} = 2^6 = 64$$

Example 91:

A cube has n dimensions. The number of 2-D cuboids equals 21. Find n.

$$\binom{n}{2} = \frac{n(n-1)}{2} = 21 \implies n(n-1) = 42 \implies n = 7$$

Example 92:

How many cuboids does a 10-dimensional cube have? At which level is the number of cuboids maximised?

$$2^{10} = 1024 \quad \text{total cuboids}$$

The maximum occurs at $k = \lfloor n/2 \rfloor = 5$:

$$\binom{10}{5} = 252 \quad \text{(largest single-level count)}$$

Example 93:

A data cube has Time(T), Product(P), Store(S), Promotion(Pr), and Customer(C) as dimensions ($n = 5$). An analyst pre-materialises all cuboids involving **at least** Product and at most 3 dimensions total. How many cuboids satisfy this condition?

- The only 1-D cuboid containing P is $\{P\} = 1$.
- Cuboids of size exactly 2 containing P: choose 1 more from $\{T, S, Pr, C\} = \binom{4}{1} = 4$
- Cuboids of size exactly 3 containing P: choose 2 more from $\{T, S, Pr, C\} = \binom{4}{2} = 6$
- Total: $1 + 4 + 6 = \mathbf{11}$ cuboids

Example 94:

For an n-dimensional cube, prove that the number of cuboids containing a specific dimension D_j is 2^{n-1}.

- Fix D_j as present. The remaining $n - 1$ dimensions can each independently be included or excluded.
- This gives 2^{n-1} distinct combinations $\Rightarrow$ 2^{n-1} cuboids contain D_j.
- **Verification ($n = 3$):** Each dimension appears in $2^2 = 4$ cuboids. For T: (T), (T, L), (T, P), (T, L, P) — indeed 4. ✓

Concept

Each dimension with L_i levels contributes L_i+1 choices (including the "ALL" choice). Total cuboids $= \prod_i (L_i+1)$.

3.2.2 Types of Data Cubes

Sparse Data Cube

In practice, most data cubes are **sparse** — many cells are empty.

- Not all combinations of dimension values have corresponding data
- Example: Not every product is sold in every city in every quarter
- **Density** of a cube:

$$\text{Density} = \frac{\text{Number of non-empty cells}}{\text{Total possible cells}} \times 100\%$$

- Storing all cells wastes memory — sparse representation techniques used
- **MOLAP** stores the full array (wastes space for sparse cubes)
- **ROLAP** stores only non-empty cells (efficient for sparse cubes)

Example 95: Cube Density Calculation

A 4-dimensional cube has dimensions:
Time (12 months), Location (50 cities), Product (200 items), Customer (1000 segments)

Total possible cells:

$$12 \times 50 \times 200 \times 1000 = 1{,}200{,}000{,}000 \quad \text{(1.2 billion)}$$

Actual non-empty cells: 3,000,000 (3 million)

$$\text{Density} = \frac{3{,}000{,}000}{1{,}200{,}000{,}000} \times 100\% = 0.25\%$$

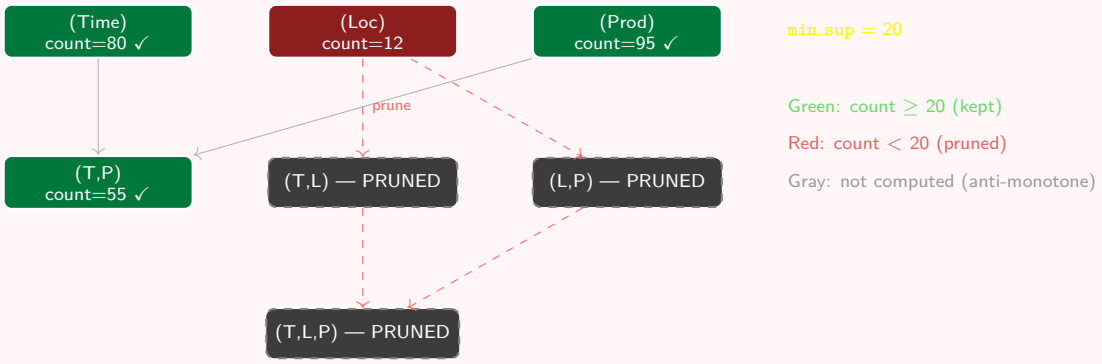
The cube is **99.75% sparse** — ROLAP is preferred; MOLAP would waste enormous memory.

Iceberg Cube

An **iceberg cube** computes and stores only those cuboid cells whose **aggregate value satisfies a minimum threshold**.

$$\text{Store cell } (d_1, d_2, \ldots, d_k) \text{ only if measure}(d_1, \ldots, d_k) \geq \text{min\_sup}$$

- Avoids materializing cells with very low counts — reduces storage dramatically
- Named after the iceberg metaphor — only the **tip** (significant cells) is visible
- Threshold called **minimum support** (often count $\geq$ some value)
- Useful for **association rule mining** on large sparse cubes



Example 96: Iceberg Cube

Sales cube with 36 cells (4 quarters $\times$ 3 locations $\times$ 3 products).
`min_sup` $= 100$ lakhs (store only cells where Sales ≥ 100).

Time	Location	Product	Sales	Stored?
Q1	Bengaluru	Electronics	120	Yes
Q1	Bengaluru	Clothing	80	No (below 100)
Q1	Mumbai	Electronics	150	Yes
Q1	Mumbai	Clothing	90	No
Q4	Mumbai	Electronics	190	Yes
...				

Only cells meeting the threshold are materialized — the rest are **pruned**.

Example 97: Sparse Cube Density

A 3-D cube has Time(4), Location(3), Product(3) with 22 non-empty cells.

$$\text{Total cells} = 4 \times 3 \times 3 = 36, \qquad \text{Density} = \frac{22}{36} \times 100\% \approx 61.1\%$$

Moderately dense; either storage approach is viable.

Example 98:

A 4-D retail cube has Month(12), Store(500), SKU(10,000), Channel(5) with 2,000,000 actual sales records.

$$\text{Total cells} = 12 \times 500 \times 10{,}000 \times 5 = 300{,}000{,}000$$

$$\text{Density} = \frac{2{,}000{,}000}{300{,}000{,}000} \times 100\% \approx 0.67\%$$

Extremely sparse (99.33% empty) — ROLAP is strongly preferred.

Example 99:

A cube has density 2% and 5×10^8 total cells. How many cells are non-empty?

$$N = 0.02 \times 5 \times 10^8 = 10{,}000{,}000 \quad \text{(10 million non-empty cells)}$$

Example 100:

(MOLAP vs. ROLAP storage): For Problem 2 above, with $b = 8$ bytes/cell in MOLAP, and in ROLAP each record has 4 dimension keys (4 bytes each) + 1 measure (8 bytes):

$$\text{MOLAP} = 300{,}000{,}000 \times 8 = 2{,}400{,}000{,}000 \text{ bytes} \approx 2.4 \text{ GB}$$

$$\text{ROLAP} = 2{,}000{,}000 \times (4 \times 4 + 8) = 2{,}000{,}000 \times 24 = 48{,}000{,}000 \text{ bytes} \approx 48 \text{ MB}$$

$$\text{Space saving} = \frac{2400 - 48}{2400} \times 100\% = 98\%$$

Example 101:

(Find density from storage constraint): A 3-D cube has 1 billion total cells. MOLAP requires 8 GB (8 bytes/cell). At what minimum density does ROLAP *require more* storage than MOLAP? (ROLAP record size = 20 bytes: 3 × 4-byte keys + 8-byte measure.)

$$N \times 20 = 10^9 \times 8 \implies N = \frac{8 \times 10^9}{20} = 4 \times 10^8$$

$$\text{Break-even density} = \frac{4 \times 10^8}{10^9} \times 100\% = 40\%$$

At density $> 40\%$, MOLAP is more storage-efficient; below 40%, ROLAP wins.

Closed Cube — Formal Definition

A cuboid cell $(d_1, d_2, \ldots, d_k)$ with measure value v is **closed** if and only if there is **no** more specific cell (with additional dimension constraints) that has the same measure value:

$$\nexists\, (d_1, \ldots, d_k, d_{k+1}) : \quad \text{measure}(d_1, \ldots, d_k, d_{k+1}) = v$$

The **closed cube** stores only the closed cells, eliminating redundancy while preserving the ability to reconstruct all other cells by roll-up.

- If $\text{measure}(d_1, \ldots, d_k) = \text{measure}(d_1, \ldots, d_k, d_{k+1})$ for every value of d_{k+1}, then the coarser cell is **redundant** (the finer cell already captures the same information)
- Closed cube $\subseteq$ Iceberg cube $\subseteq$ Full cube
- Used primarily with COUNT; a cell is not closed if all its refinements have the same count (i.e., adding a dimension doesn't split the group)

Example 102: Closed Cube — Identifying and Pruning Redundant Cells

Consider a 2-D cube with dimensions: Location $\in \{North, South\}$ and Customer $\in \{Retail, Wholesale\}$. The base cuboid (2-D) has the following COUNT values:

Location \ Customer	Retail	Wholesale	(Location) cuboid
North	15	0	15
South	10	25	35
(Customer) cuboid	25	25	**50 (Apex)**

Checking closedness of (North) in the 1-D (Location) cuboid:
The cell $(North)$ has count $= 15$. Its refinements in the 2-D cuboid are: $(North, Retail) = 15$ and $(North, Wholesale) = 0$.
Since the refinement $(North, Retail)$ has the *same count* (15) as its parent $(North)$, the cell $(North)$ **is not closed** — adding the Customer dimension does not change the count.
Checking closedness of (Wholesale) in the 1-D (Customer) cuboid:
The cell $(Wholesale)$ has count $= 25$. Its refinements in the 2-D cuboid are: $(North, Wholesale) = 0$ and $(South, Wholesale) = 25$.
Since the refinement $(South, Wholesale)$ has the *same count* (25) as its parent $(Wholesale)$, the cell $(Wholesale)$ **is not closed**.
Case of a closed cell:
Checking $(South)$ in the 1-D (Location) cuboid: $(South)$ has count $= 35$. Its refinements are $(South, Retail) = 10$ and $(South, Wholesale) = 25$. Since neither refinement equals 35, the cell $(South)$ *is* closed.
Summary: Non-closed cells (like North and Wholesale in this table) are redundant. In a Closed Cube representation, we prune these and only store cells where every refinement results in a different count value, effectively compressing the data without losing information.

Example 103: Closed Cube — Identifying and Pruning Redundant Cells

Consider a small 2-D cube: Time $\in \{Q1, Q2\}$, Product $\in \{Electronics, Clothing\}$. The base cuboid (2-D) has the following COUNT values:

Time \ Product	Electronics	Clothing	(Time) cuboid
Q1	5	5	10
Q2	8	3	11
(Product) cuboid	13	8	**21 (Apex)**

Checking closedness of (Q1) in the 1-D (Time) cuboid:
$(Q1)$ has count $= 10$. Its refinements in the 2-D cuboid are: $(Q1, \text{Electronics}) = 5$ and $(Q1, \text{Clothing}) = 5$.
Since $5 \neq 10$, the cell $(Q1)$ *is* closed — adding the Product dimension changes the count.
Checking closedness of (Electronics) in the 1-D (Product) cuboid:
$(Electronics)$ has count $= 13$. Its refinements: $(Q1, Electronics) = 5$, $(Q2, Electronics) = 8$.
Since $5 \neq 13$ and $8 \neq 13$, $(Electronics)$ is closed.
Hypothetical non-closed case:
If all sales of Clothing occurred only in Q2 and there were no other data, then: $(Q2, Clothing) =$ all Clothing sales, and the (Clothing) cuboid cell would equal $(Q2, Clothing)$ — making (Clothing) *not closed* (redundant with its 2-D refinement).
Summary: In the example above, all cells happen to be closed. Non-closed cells arise when the same count value is shared by a cell and one of its direct 1-step refinements — signalling that the refinement adds no information.

3.3 Schemas for Multidimensional Data Models

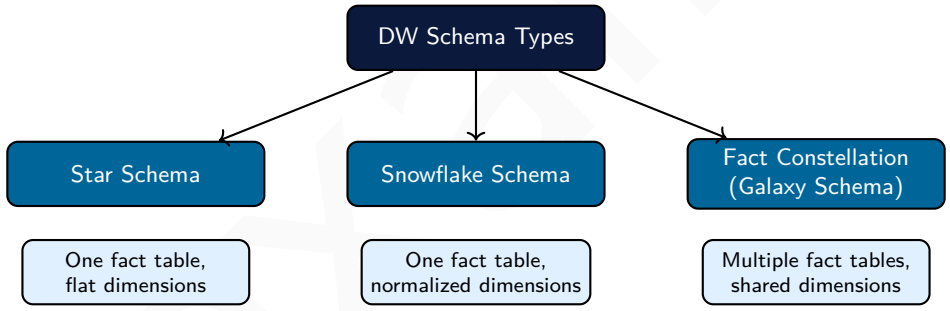
Schema for Multidimensional Data Model

A **schema** in a data warehouse defines how **fact tables** and **dimension tables** are organized and related.

- **Fact table:** Central table storing **measures** (numeric, quantitative data) and **foreign keys** to dimension tables
- **Dimension table:** Stores **descriptive attributes** of a dimension — the context for analysis
- Three standard schemas: **Star**, **Snowflake**, **Fact Constellation (Galaxy)**

Fact Table vs Dimension Table

Property	Fact Table	Dimension Table
Content	Measures + Foreign Keys	Descriptive attributes
Size	Very large (millions of rows)	Small (thousands of rows)
Updates	Append only (ETL)	Slowly changing
Keys	Composite primary key (FKs)	Surrogate primary key
Normalization	Not normalized	Varies by schema
Examples	Sales_Fact, Order_Fact	Time_Dim, Product_Dim



Star Schema

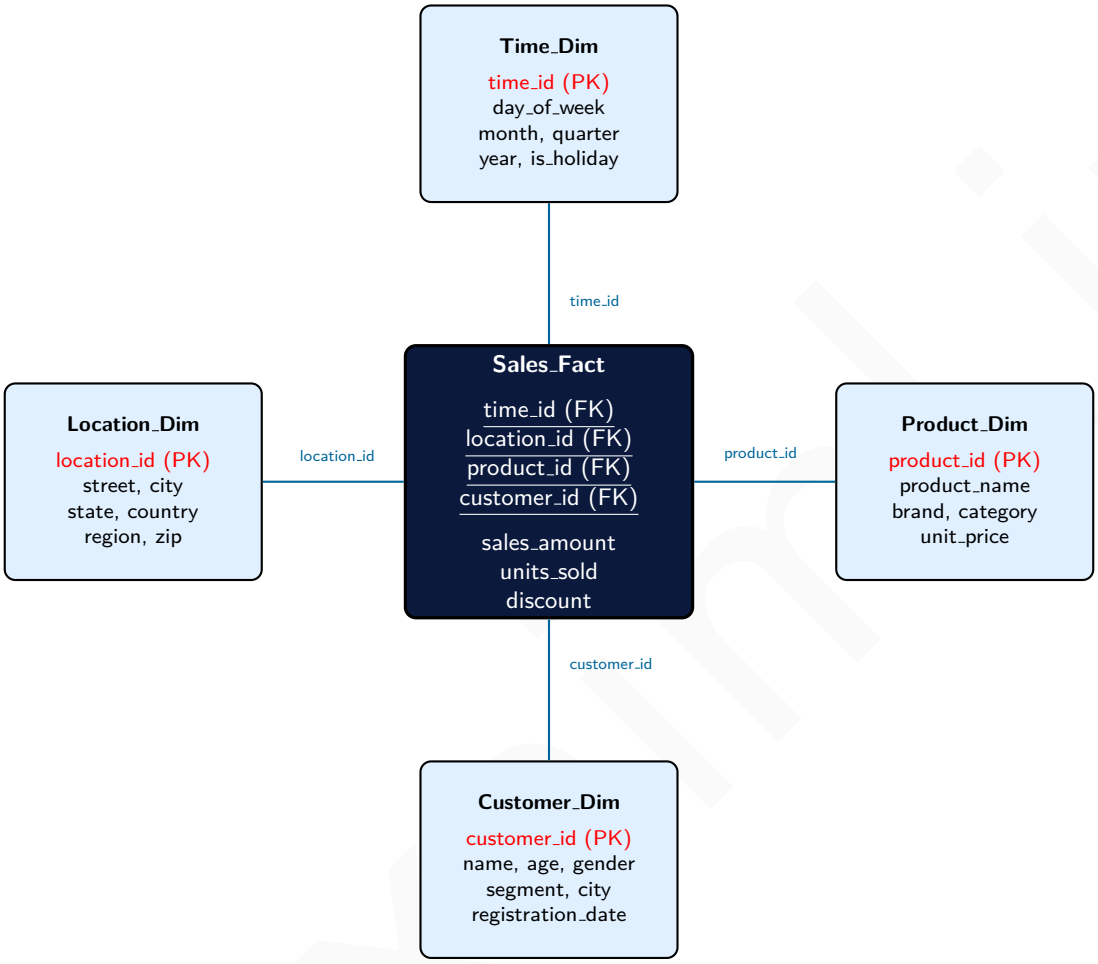
A **star schema** has a single **central fact table** connected to multiple **denormalized dimension tables** — resembling a star shape.

- Dimension tables are **flat** — all attributes in one table, not further decomposed
- Simple structure — **fewer joins** required for queries
- **Fast query performance** — optimized for OLAP
- **Data redundancy** in dimension tables (denormalized)
- Most widely used schema in data warehouses

Star Schema Structure

- **Fact table:** Contains measures and foreign keys to all dimension tables
  Primary key = composite of all foreign keys
- **Dimension tables:** One per dimension; contain all attributes of that dimension in a single flat table
- Number of joins in a query = number of dimensions involved

- **Denormalization:** Dimension attributes are stored redundantly to avoid joins



Example 104: Star Schema: Sales Query

Query: Total sales per product category per quarter for 2024.

SQL on Star Schema:

```
SELECT  T.quarter, P.category, SUM(F.sales_amount)
FROM    Sales_Fact F
JOIN    Time_Dim T     ON F.time_id     = T.time_id
JOIN    Product_Dim P  ON F.product_id  = P.product_id
WHERE   T.year = 2024
GROUP BY T.quarter, P.category;
```

- Only **2 joins** needed (Time and Product) — Location and Customer not touched
- All product attributes (name, brand, category) are in **one flat table** — no further joins
- Fast execution — typical OLAP query pattern

Example 105: Star Schema: Denormalization Trade-off

Product_Dim stores category alongside individual product:

product_id	product_name	brand	category	unit_price
101	Galaxy S24	Samsung	Electronics	79999
102	iPhone 15	Apple	Electronics	89999
103	Levi Jeans	Levi's	Clothing	3999
104	Nike T-Shirt	Nike	Clothing	1499

- "Electronics" is stored **twice** — redundancy due to denormalization
- In a normalized schema, category would be in a separate table
- Star schema accepts this redundancy to **avoid extra joins** at query time

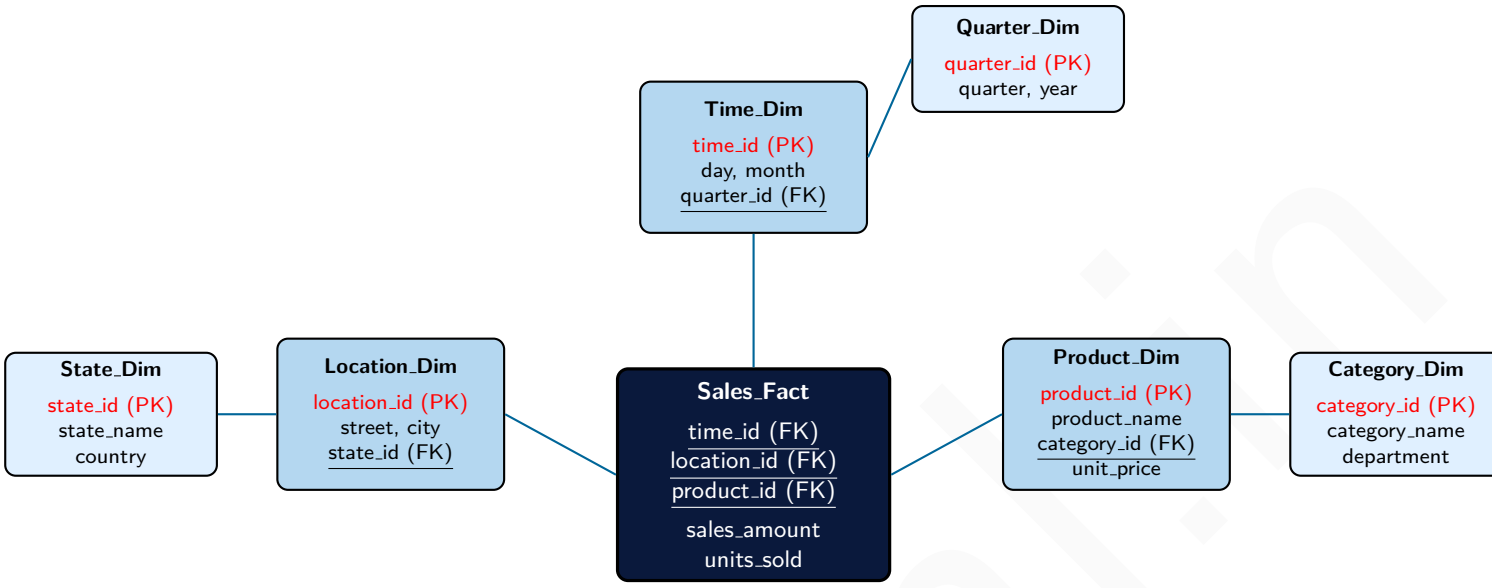
Snowflake Schema

A **snowflake schema** is a **normalized** extension of the star schema where dimension tables are **further decomposed** into multiple related tables.

- Dimension tables are **normalized** — no redundancy
- Resembles a snowflake shape — dimensions branch out into sub-dimensions
- **Reduces storage** — eliminates redundant attribute values
- **More joins** required for queries — slower than star schema
- Better for **frequent dimension updates** (normalized tables easier to maintain)

Star Schema vs Snowflake Schema

Property	Star Schema	Snowflake Schema
Dimension tables	Flat (denormalized)	Normalized (multiple tables)
Number of tables	Fewer	More
Number of joins	Fewer (faster queries)	More (slower queries)
Data redundancy	Higher	Lower
Storage	More	Less
Maintenance	Harder (anomalies possible)	Easier (normalized)
Preferred for	Query performance	Storage efficiency



Example 106: Snowflake Schema: Query with Extra Joins

Query: Total sales per department per year.

SQL on Snowflake Schema:

```
SELECT  Q.year, C.department, SUM(F.sales_amount)
FROM    Sales_Fact F
JOIN    Product_Dim P   ON F.product_id  = P.product_id
JOIN    Category_Dim C  ON P.category_id = C.category_id
JOIN    Time_Dim T      ON F.time_id     = T.time_id
JOIN    Quarter_Dim Q   ON T.quarter_id  = Q.quarter_id
GROUP BY Q.year, C.department;
```

- **5 joins** required — compared to 2 joins in equivalent star schema query
- `department` is not in `Product_Dim` directly — must traverse `Category_Dim`
- `year` is not in `Time_Dim` directly — must traverse `Quarter_Dim`
- Snowflake pays for normalization with **extra join overhead**

Example 107: Snowflake: Normalization Removes Redundancy

Star Schema — Product_Dim (denormalized):

product_id	product_name	category_name	department
101	Galaxy S24	Electronics	Technology
102	iPhone 15	Electronics	Technology
103	Pixel 8	Electronics	Technology

"Electronics" and "Technology" repeated 3 times.

Snowflake Schema — normalized into two tables:

Product_Dim:

product_id	product_name	category_id
101	Galaxy S24	10
102	iPhone 15	10
103	Pixel 8	10

Category_Dim:

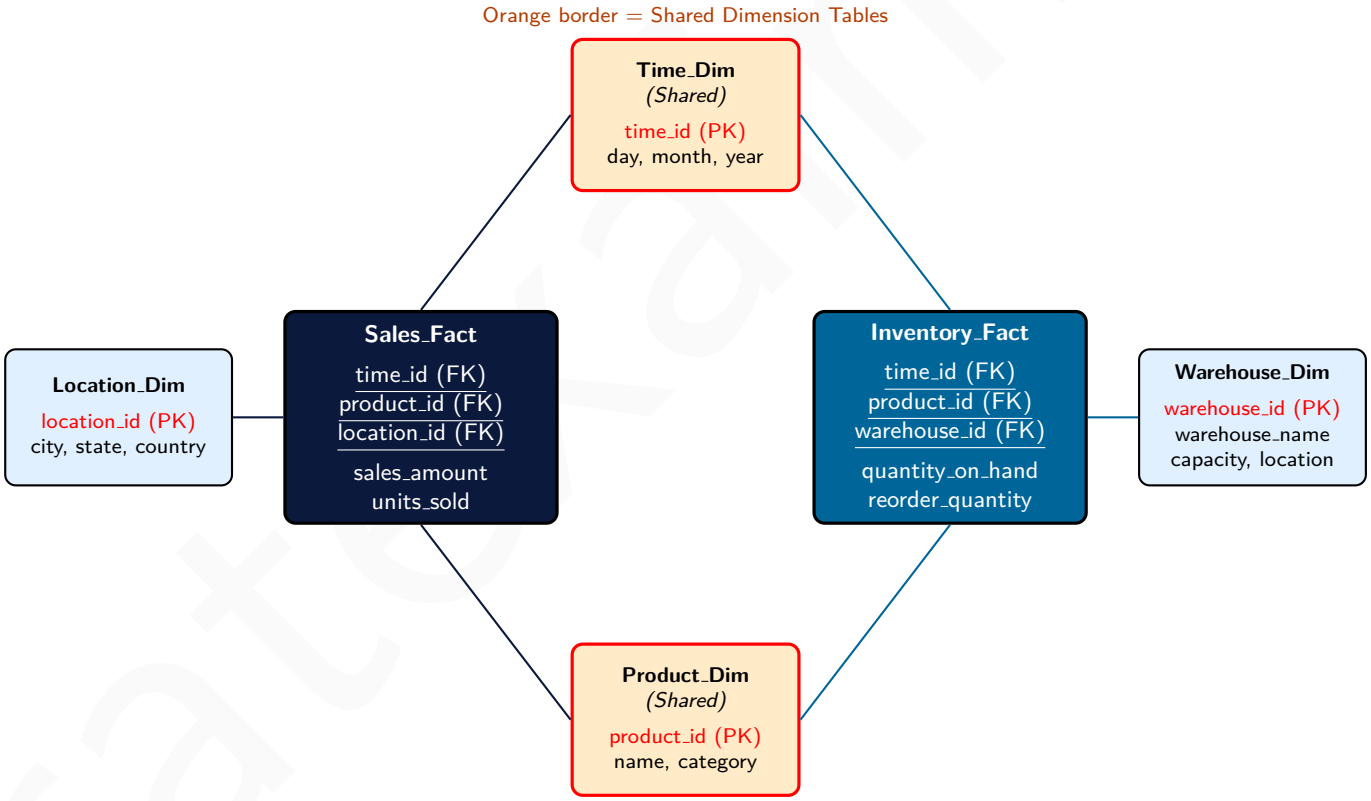
category_id	category_name	department
10	Electronics	Technology

"Electronics" and "Technology" stored **once only** — no redundancy.

Fact Constellation Schema (Galaxy Schema)

A **fact constellation** (also called **galaxy schema**) contains **multiple fact tables** that **share dimension tables**.

- Models **complex real-world scenarios** where multiple subjects need analysis
- Shared dimension tables — consistency across fact tables
- Each fact table can have **different measures** and **different grains**
- Most flexible but **most complex** schema to design and maintain
- Looks like a collection of stars — hence "galaxy" schema



Example 108: Fact Constellation: Two Fact Tables

A retail company needs to analyze both **Sales** and **Inventory**:

Property	Sales_Fact	Inventory_Fact
Measures	sales_amount, units_sold	quantity_on_hand, reorder_qty
Grain	One row per sale	One row per product per day
Shared dims	Time_Dim, Product_Dim	Time_Dim, Product_Dim
Exclusive dims	Location_Dim	Warehouse_Dim

- Cross-fact query: "Which products have high sales but low inventory?" — requires joining both fact tables via shared `Product_Dim`
- Shared dimension ensures **consistent product definitions** across both facts

Example 109: Identifying Schema Type

Q: A data warehouse has the following tables:
`Sales_Fact`, `Time_Dim`, `Customer_Dim`, `Product_Dim`, `Category_Dim`
`Product_Dim` has a foreign key to `Category_Dim`.
What schema does this represent?

- One fact table (`Sales_Fact`) $\rightarrow$ not Fact Constellation
- `Product_Dim` connects to `Category_Dim` — dimension table is normalized
- This is a **Snowflake Schema** — dimension hierarchy is normalized out into sub-dimensions

Answer: Snowflake Schema

If `Category_Dim` were merged into `Product_Dim` (no sub-table), it would be a **Star Schema**.

Example 110: Fact Table Primary Key

Q: A star schema has a fact table `Orders_Fact` with foreign keys:
`customer_id, product_id, time_id, store_id`
What is the primary key of `Orders_Fact`?

- No single attribute uniquely identifies a row in a fact table
- Primary key = **composite key** of all foreign keys:
  PK = (`customer_id, product_id, time_id, store_id`)
- This assumes one fact per unique combination of all dimensions
- If multiple orders can exist for the same combination, a **surrogate key** (`order_id`) is added

Summary — Three Schema Types

Property	Star	Snowflake	Fact Constellation
Fact tables	1	1	2 or more
Dimension structure	Flat	Normalized	Flat or Normalized
Joins required	Minimum	More	Most
Query speed	Fastest	Moderate	Complex
Redundancy	High	Low	Low (shared dims)
Storage	More	Less	Moderate
Complexity	Simple	Moderate	High
Use case	OLAP queries	Storage-opt	Multi-subject analysis

3.4 Concept Hierarchy

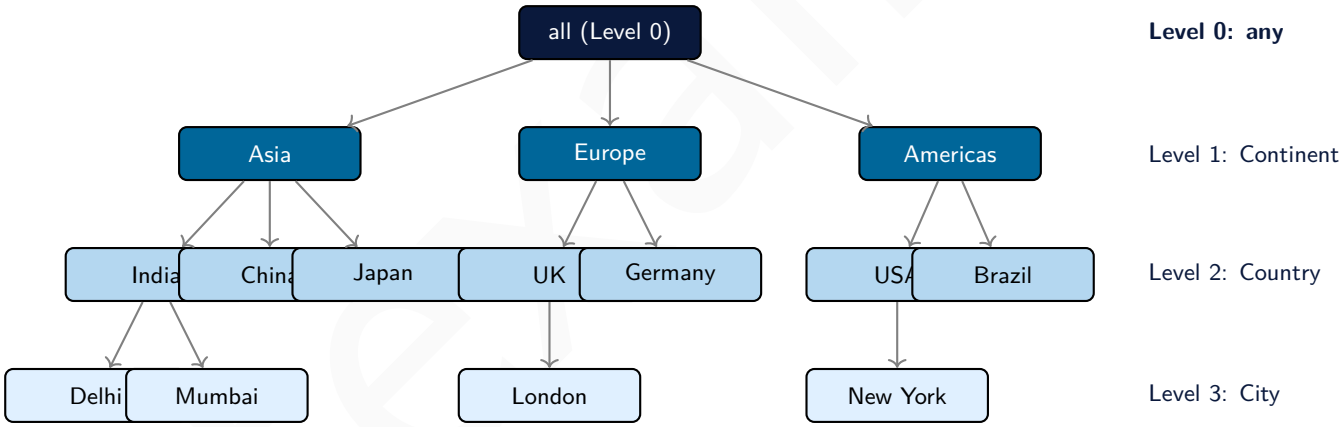
Concept Hierarchy

A **concept hierarchy** defines a sequence of mappings from a set of **low-level concepts** to **higher-level, more general concepts**.

- Organizes nominal attribute values into **levels of abstraction**
- Enables **roll-up** and **drill-down** operations in data cubes
- Allows mining at **multiple levels of granularity**
- Example: `City` → `State` → `Country` → `Continent`

Why Concept Hierarchies for Nominal Data?

- Nominal attributes have **many distinct values** at fine levels (e.g., thousands of cities)
- Patterns at fine level are **too sparse** to be statistically meaningful
- Rolling up to higher levels **groups values**, increasing support for patterns
- Enables **generalization**: specific value → broader category
- Reduces cardinality: fewer distinct values → faster mining



Methods for Generating Concept Hierarchies for Nominal Data

Four main approaches:

- **Method 1:** Specified explicitly by a user or domain expert
- **Method 2:** Specified by the order of attributes (schema hierarchy)
- **Method 3:** Automatically generated by number of distinct values
- **Method 4:** Specified by explicit grouping of values

Method 1 — Explicitly Specified by User/Expert

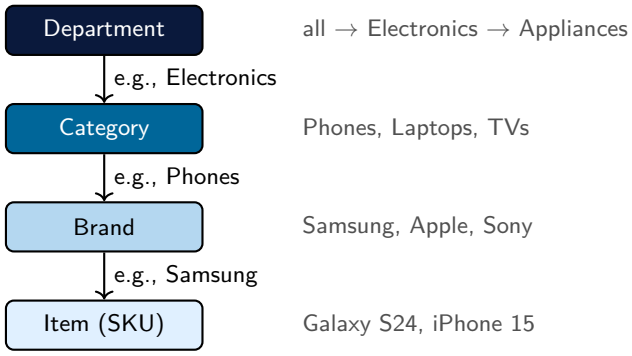
A domain expert **manually defines** the hierarchy levels and mappings.

- Most accurate — captures domain knowledge
- Time-consuming and requires expert involvement

- Stored as **metadata** in the data warehouse schema
- Example: A retail expert defines product hierarchy:
  `Item` → `Brand` → `Category` → `Department`

Example 111: Explicit Hierarchy — Product

A retail database stores individual products. An expert defines:



Roll-up from `Item` to `Category`: instead of analyzing 10,000 SKUs, analyst sees 5 categories.

Method 2 — Specified by Order of Attributes (Schema Hierarchy)

The hierarchy is implied by the **order of attributes listed** in the database schema.

- User specifies: `street` < `city` < `state` < `country`
- System automatically infers the hierarchy from this order
- A **total ordering** of attributes defines the levels
- Simple to implement — no explicit value-to-value mapping needed
- The attribute with **fewer distinct values** is at a higher level

Schema Hierarchy Rule

If a user specifies the partial order:

`day` < `month` < `quarter` < `year`

Then the system knows:

- `day` is the most specific level (many distinct values)
- `year` is the most general level (few distinct values)
- Rolling up moves **left to right** in the ordering
- Drilling down moves **right to left**

Example 112: Schema Hierarchy — Location

Database table `Customer` has attributes:
`Street, City, District, State, Country`

User specifies order: `Street` < `City` < `District` < `State` < `Country`

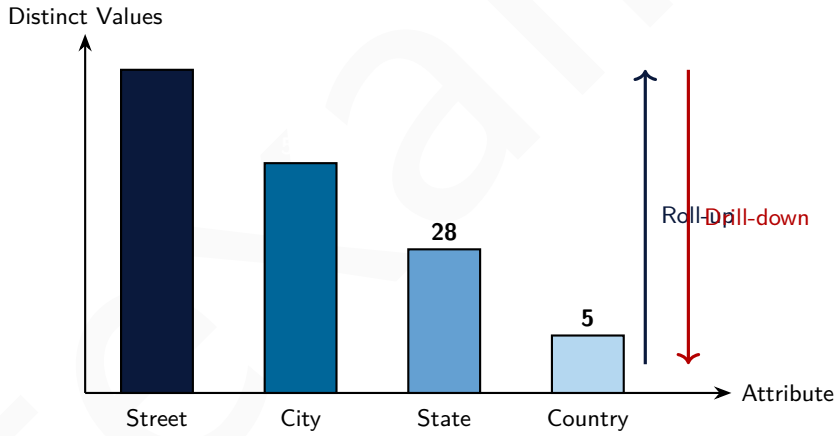
Attribute	Distinct Values	Example
Street	50,000	MG Road, Brigade Road
City	500	Bengaluru, Chennai
District	200	Bengaluru Urban, Mysuru
State	28	Karnataka, Tamil Nadu
Country	5	India, USA

Fewer distinct values → higher in hierarchy.
Mining at `City` level instead of `Street` reduces cardinality from 50,000 to 500.

Method 3 — Automatic Generation by Number of Distinct Values

System **automatically generates** the hierarchy by sorting attributes on the number of distinct values:

- Attribute with the **most distinct values** → lowest level (most specific)
- Attribute with the **fewest distinct values** → highest level (most general)
- No user specification of order needed
- Works well when attributes have a **natural parent-child relationship**
- May produce incorrect hierarchies if attributes are **unrelated**



Example 113: Automatic Hierarchy Generation

A data warehouse has the following attributes and their distinct value counts:

Attribute	Distinct Values	Auto-assigned Level
`ProductID`	12,000	Level 4 (lowest)
`ProductName`	8,500	Level 3
`SubCategory`	120	Level 2
`Category`	15	Level 1
`Department`	4	Level 0 (highest)

- System automatically orders: `ProductID` < `ProductName` < `SubCategory` < `Category` < `Department`
- No user input needed — purely driven by cardinality
- Risk: `ProductID` and `ProductName` may not have a true parent-child relationship

Method 4 — Explicit Value Grouping

User **manually groups specific values** of an attribute into higher-level concepts.

- Most flexible — handles arbitrary, non-geographic groupings
- Example: Group countries by economic classification:
  {India, China, Brazil} → `Developing`
  {USA, UK, Germany} → `Developed`
- Requires domain knowledge — groups may not be derivable from data alone
- Stored as a **mapping table** in the data warehouse

Example 114: Explicit Grouping — Job Role to Department

Attribute: `JobRole` with many values. User defines groupings:

JobRole (Level 0)	Department (Level 1)
Data Analyst, Data Scientist, ML Engineer	`Analytics`
Frontend Dev, Backend Dev, DevOps	`Engineering`
HR Manager, Recruiter, Payroll Officer	`Human Resources`
CFO, Accountant, Financial Analyst	`Finance`

Mining at `Department` level: "Analytics department has highest project completion rate" — a pattern invisible at the `JobRole` level due to data sparsity.

Example 115: Explicit Grouping — Academic Grades to Performance Band

Attribute: `Grade` $\in$ {A+, A, A–, B+, B, B–, C+, C, D, F}

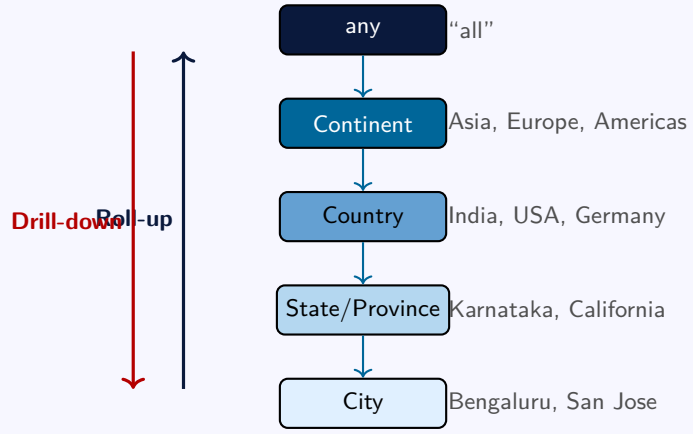
User defines grouping:

Grade Values	Performance Band
A+, A, A–	`Distinction`
B+, B, B–	`Merit`
C+, C	`Pass`
D, F	`Fail`

Roll-up from 10 grade values to 4 performance bands — meaningful for policy decisions.
Pattern: "80% of scholarship holders are in `Distinction`" is clear at rolled-up level.

Multi-Level Concept Hierarchy — Combined Example

A `Location` attribute supports a 4-level hierarchy:



Concept Hierarchy vs Discretization — Key Difference

Property	Concept Hierarchy	Discretization
Attribute type	Primarily **nominal**	Primarily **numeric**
Basis	Semantic grouping	Value range splitting
Output	Abstract category labels	Ordered discrete intervals
Knowledge	Domain knowledge required	Statistical/algorithmic
Example	City $\rightarrow$ Country	Age $\in [20, 40) \rightarrow$ "Young"

3.4.1 Measures: Categorization and Computation

Measure Classification

A **measure** is a numeric function defined over the cells of a data cube. Measures are classified by how they *behave under aggregation* across cuboids:

- **Distributive** — computable from sub-aggregate results alone.
- **Algebraic** — computable from a *fixed finite* set of distributive sub-aggregates.
- **Holistic** — require the *entire* partition; no bounded sub-aggregate suffices.

This classification directly determines **how efficiently** a measure can be pre-computed and maintained in a data cube or OLAP system.

Measures in a Data Cube

Measures are classified by how they behave during **aggregation**:

- **Distributive:** Can be computed from sub-aggregates
  $\text{agg}(S) = f(\text{agg}(S_1), \text{agg}(S_2), \ldots)$
  Examples: `SUM`, `COUNT`, `MIN`, `MAX`
- **Algebraic:** Computed from a fixed number of distributive aggregates
  Examples: `AVG` (= SUM / COUNT), `Standard Deviation`
- **Holistic:** Cannot be computed from sub-aggregates — require the full set
  Examples: `MEDIAN`, `MODE`, `RANK`

Summary of Measures in a Data Cube

Measure	Classification	Sub-aggregates needed	Formula
SUM	Distributive	SUM only	$\text{SUM}(S) = \sum_j \text{SUM}(S_j)$
COUNT	Distributive	COUNT only	$\text{COUNT}(S) = \sum_j \text{COUNT}(S_j)$
MIN	Distributive	MIN only	$\text{MIN}(S) = \min_j \text{MIN}(S_j)$
MAX	Distributive	MAX only	$\text{MAX}(S) = \max_j \text{MAX}(S_j)$
AVG	Algebraic	SUM, COUNT	SUM/COUNT
σ (Std Dev)	Algebraic	SUM(x^2), SUM(x), COUNT	See formula above
MEDIAN	Holistic	Full sorted list	$n/2$-th value
MODE	Holistic	Full frequency table	Most frequent value
RANK	Holistic	Full ordered list	Position of element
PERCENTILE	Holistic	Full sorted list	p-th percentile value

Distributive
SUM, COUNT
MIN, MAX

Algebraic
AVG, Std Dev
Min-Max Range

Holistic
MEDIAN, MODE
RANK, Percentile

Example 116: Distributive vs Algebraic vs Holistic

Consider partitioned sales data split by region:
Region A: {10, 20, 30} Region B: {40, 50, 60}

SUM (Distributive):

$$\text{SUM(A)} = 60, \quad \text{SUM(B)} = 150, \quad \text{SUM(All)} = 60 + 150 = 210 \quad \text{(computable from sub-aggregates)}$$

AVG (Algebraic):

$$\text{AVG(A)} = 20, \quad \text{AVG(B)} = 50$$

$$\text{AVG(All)} \neq \frac{20 + 50}{2} = 35 \quad \text{(wrong!)}$$

$$\text{AVG(All)} = \frac{\text{SUM(All)}}{\text{COUNT(All)}} = \frac{210}{6} = 35 \quad \text{(correct — needs both SUM and COUNT)}$$

MEDIAN (Holistic):

$$\text{MEDIAN(A)} = 20, \quad \text{MEDIAN(B)} = 50$$

$$\text{MEDIAN(All)} = \text{MEDIAN}(\{10, 20, 30, 40, 50, 60\}) = \frac{30 + 40}{2} = 35$$

Cannot compute this from MEDIAN(A) and MEDIAN(B) alone — requires the **full sorted list**.

Example 117: AVG Aggregation Trap — Why AVG Is Not Distributive

Three branch offices report quarterly sales:

Branch	SUM (lakhs)	COUNT (transactions)	AVG
North	240	8	30
South	150	15	10
East	360	12	30

Incorrect global AVG:

$$\text{AVG}_{\text{wrong}} = \frac{30 + 10 + 30}{3} = \frac{70}{3} \approx 23.33 \quad 55$$

Correct global AVG (algebraic combination):

$$\text{AVG}_{\text{correct}} = \frac{240 + 150 + 360}{8 + 15 + 12} = \frac{750}{35} \approx 21.43 \quad \checkmark$$

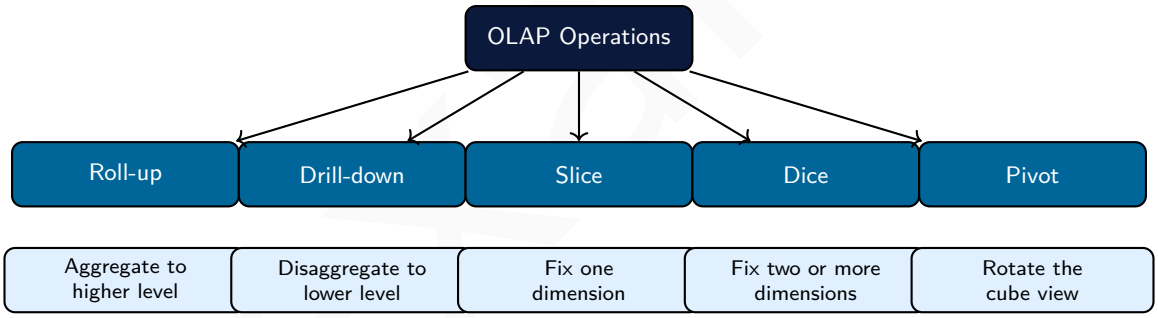
Conclusion: Simply averaging the branch averages *ignores differing counts*. The correct approach requires carrying both SUM and COUNT as sub-aggregates — confirming AVG is *algebraic*, not distributive.

3.5 Typical OLAP Operations

OLAP Operations

OLAP (Online Analytical Processing) operations allow analysts to **navigate and explore** data cubes at different levels of abstraction and from different perspectives.

- Operate on **multidimensional data cubes**
- Each operation produces a **new view** of the data
- Five fundamental operations: **Roll-up, Drill-down, Slice, Dice, Pivot**
- Supported by data warehouses for fast **decision support queries**



Running Example — Sales Data Cube

All five operations are demonstrated on the following base data cube:

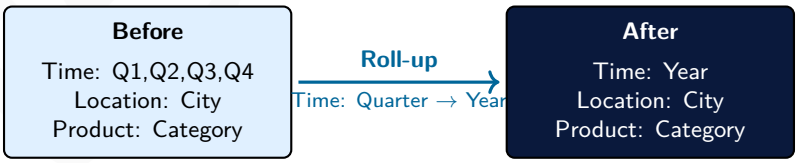
Dimensions: `Time` (Quarter), `Location` (City), `Product` (Category)
Measure: `Sales` (in lakhs)

Time	Location	Product	Sales (Lakhs)
Q1	Bengaluru	Electronics	120
Q1	Bengaluru	Clothing	80
Q1	Mumbai	Electronics	150
Q1	Mumbai	Clothing	90
Q2	Bengaluru	Electronics	130
Q2	Bengaluru	Clothing	85
Q2	Mumbai	Electronics	160
Q2	Mumbai	Clothing	95
Q3	Bengaluru	Electronics	140
Q3	Bengaluru	Clothing	75
Q3	Mumbai	Electronics	170
Q3	Mumbai	Clothing	100
Q4	Bengaluru	Electronics	160
Q4	Bengaluru	Clothing	95
Q4	Mumbai	Electronics	190
Q4	Mumbai	Clothing	110

Roll-up (Drill-up)

Roll-up aggregates data by **climbing up** a concept hierarchy or by **reducing a dimension**.

- Moving to a **higher, more general** level of abstraction
- Applies an **aggregate function** (SUM, AVG, COUNT) across the rolled-up level
- Two ways to roll up:
  - **Along a hierarchy:** Day → Month → Quarter → Year
  - **By dimension reduction:** Remove a dimension entirely
- Result has **fewer rows and fewer details**



Example 118: Roll-up: Quarter → Year (along Time hierarchy)

Roll up `Time` from **Quarter** to **Year** — aggregate Sales by SUM.

Time	Location	Product	Sales (Lakhs)
2024	Bengaluru	Electronics	$120 + 130 + 140 + 160 = 550$
2024	Bengaluru	Clothing	$80 + 85 + 75 + 95 = 335$
2024	Mumbai	Electronics	$150 + 160 + 170 + 190 = 670$
2024	Mumbai	Clothing	$90 + 95 + 100 + 110 = 395$

16 rows → 4 rows. Detail lost; annual trends visible.

Example 119: Roll-up: City → State (along Location hierarchy)

Roll up `Location` from **City** to **State**:
Bengaluru and Mumbai both belong to different states (Karnataka, Maharashtra).

Time	State	Product	Sales (Lakhs)
Q1	Karnataka	Electronics	120
Q1	Karnataka	Clothing	80
Q1	Maharashtra	Electronics	150
Q1	Maharashtra	Clothing	90
... (Q2, Q3, Q4 similarly)			

If both cities belonged to the same state, their sales would be **summed** into one row per state.

Example 120: Roll-up: Dimension Reduction — Remove Product

Remove `Product` dimension entirely — sum across all products:

Time	Location	Total Sales (Lakhs)
Q1	Bengaluru	$120 + 80 = 200$
Q1	Mumbai	$150 + 90 = 240$
Q2	Bengaluru	$130 + 85 = 215$
Q2	Mumbai	$160 + 95 = 255$
Q3	Bengaluru	$140 + 75 = 215$
Q3	Mumbai	$170 + 100 = 270$
Q4	Bengaluru	$160 + 95 = 255$
Q4	Mumbai	$190 + 110 = 300$

Product breakdown lost — now shows total sales per city per quarter.

Drill-down

Drill-down is the **reverse of roll-up** — navigates from **summarized data to more detailed data**.

- Moves **down** a concept hierarchy to a lower, more specific level
- Or **adds a new dimension** to the current view
- Result has **more rows and finer detail**
- Used when a summary shows an interesting pattern that warrants deeper investigation

Example 121: Drill-down: Year → Quarter

Starting from the rolled-up annual view, drill down `Time` back to **Quarter**:

Time	Location	Total Sales (Lakhs)
Q1	Bengaluru	200
Q2	Bengaluru	215
Q3	Bengaluru	215
Q4	Bengaluru	255
Q1	Mumbai	240
Q2	Mumbai	255
Q3	Mumbai	270
Q4	Mumbai	300

Insight: Mumbai's Q4 sales (300) are consistently highest — visible only after drilling down.

Example 122: Drill-down: Adding a New Dimension

Starting from a 2-dimension view (Time, Location), add `Product` dimension:

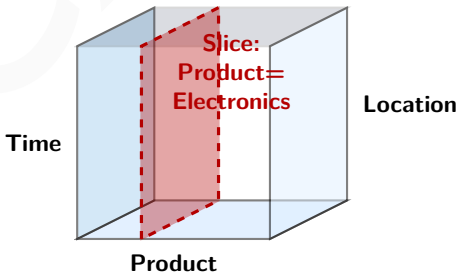
Time	Location	Product	Sales
Q4	Mumbai	Electronics	190
Q4	Mumbai	Clothing	110

Mumbai Q4 total was 300 — drilling into Product reveals Electronics (190) dominates Clothing (110).

Slice

Slice selects a **single value** for one dimension, reducing the cube by one dimension.

- Fixes **exactly one dimension** to a **single specific value**
- Result is a **2D table** (if original was 3D cube)
- Equivalent to a **WHERE** clause on one dimension in SQL
- Does **not aggregate** — all remaining rows at the same granularity are retained



Example 123: Slice: Product

Fix `Product = Electronics` — remove Product dimension:

Time	Location	Sales (Lakhs)
Q1	Bengaluru	120
Q1	Mumbai	150
Q2	Bengaluru	130
Q2	Mumbai	160
Q3	Bengaluru	140
Q3	Mumbai	170
Q4	Bengaluru	160
Q4	Mumbai	190

Equivalent SQL:

```
SELECT Time, Location, SUM(Sales)
FROM SalesCube
WHERE Product = 'Electronics'
GROUP BY Time, Location;
```

Example 124: Slice: Time

Fix `Time = Q1` — analyze only first quarter performance:

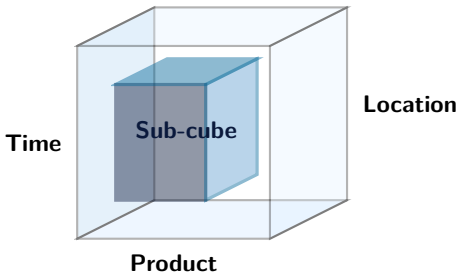
Location	Product	Sales (Lakhs)
Bengaluru	Electronics	120
Bengaluru	Clothing	80
Mumbai	Electronics	150
Mumbai	Clothing	90

3D cube (Time $\times$ Location $\times$ Product) $\rightarrow$ 2D table (Location $\times$ Product) for Q1 only.

Dice

Dice selects a **range of values** for **two or more dimensions**, extracting a **sub-cube**.

- Fixes a **subset of values** (not just one) on multiple dimensions
- Result is a **smaller cube** — same number of dimensions, fewer values per dimension
- Equivalent to a **WHERE** clause on **multiple dimensions** in SQL
- Key difference from Slice: Dice retains dimensionality; Slice reduces it by one



Example 125: Dice

Select sub-cube where Time is Q1 or Q2 **and** Location is Mumbai:

Time	Location	Product	Sales (Lakhs)
Q1	Mumbai	Electronics	150
Q1	Mumbai	Clothing	90
Q2	Mumbai	Electronics	160
Q2	Mumbai	Clothing	95

16 rows $\rightarrow$ 4 rows. Still a 3D sub-cube (Time $\times$ Location $\times$ Product).

Equivalent SQL:

```
SELECT Time, Location, Product, Sales
FROM SalesCube
WHERE Time IN ('Q1','Q2') AND Location = 'Mumbai';
```

Example 126: Dice

Three dimensions constrained:

Time	Location	Product	Sales
Q3	Bengaluru	Electronics	140
Q4	Bengaluru	Electronics	160

This is the **smallest meaningful sub-cube** — 2 rows remain, all 3 dimensions still present.

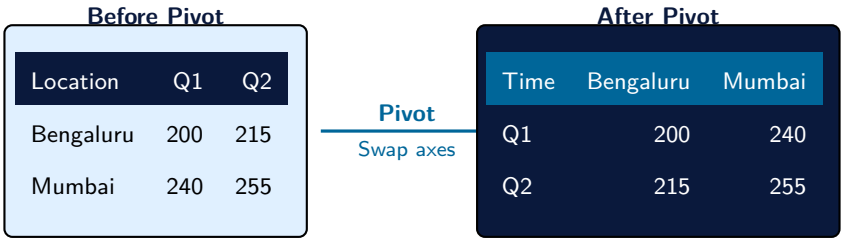
Slice vs Dice — Key Distinction

Property	Slice	Dice
Dimensions fixed	Exactly **one**	**Two or more**
Value constraint	Single value only	Range or set of values
Result dimensions	Original $-$ 1	Same as original
3D cube result	2D table	Smaller 3D sub-cube
SQL equivalent	`WHERE dim = value`	`WHERE dim1 IN (...) AND dim2 IN (...)`

Pivot (Rotate)

Pivot rotates the data cube to present an alternative view by **reorienting dimensions**.

- Does **not change the data** — only changes the **presentation axis**
- Swaps rows and columns in a 2D cross-tabulation
- Makes different relationships more visible
- Also called **rotate** in some systems
- Familiar as **Pivot Table** in Excel/spreadsheet tools



Example 127: Pivot: Location as Rows → Location as Columns

Before Pivot — rows indexed by Location, columns by Time:

Location	Q1	Q2	Q3	Q4
Bengaluru	200	215	215	255
Mumbai	240	255	270	300

After Pivot — rows indexed by Time, columns by Location:

Time	Bengaluru	Mumbai
Q1	200	240
Q2	215	255
Q3	215	270
Q4	255	300

Same data — different perspective. Quarterly growth trend per city is now easier to read.

Example 128: Pivot: Product as Rows → Product as Columns

Before Pivot (after Slice: Location = Bengaluru):

Time	Product	Sales
Q1	Electronics	120
Q1	Clothing	80
Q2	Electronics	130
Q2	Clothing	85

After Pivot:

Time	Electronics	Clothing
Q1	120	80
Q2	130	85

Comparing Electronics vs Clothing across quarters is now **immediate** from a single row.

Summary — All Five OLAP Operations

Operation	What it does	Effect on data	Example
Roll-up	Climb hierarchy / remove dim	Fewer rows, aggregated	Q $\rightarrow$ Year
Drill-down	Descend hierarchy / add dim	More rows, more detail	Year $\rightarrow$ Q
Slice	Fix one dim to single value	n-D $\rightarrow$ $(n-1)$-D	Product = Electronics
Dice	Fix 2+ dims to value ranges	Smaller n-D sub-cube	Q1–Q2, Mumbai
Pivot	Rotate axis orientation	Same data, new layout	Swap rows/cols

3.6 Problems

Problem 107 ***[MCQ]*** *A bank's system processes thousands of* `INSERT`*,* `UPDATE`*, and* `DELETE` *transactions per second on a normalized relational database serving ATMs and tellers. This system is best classified as:*

(A) OLAP, because it deals with financial data

(B) OLTP, because it handles short, atomic, real-time transactions

(C) A data warehouse, because it stores historical records

(D) A data mart, because it is domain-specific (banking)

Problem 108 ***[MSQ]*** *Which of the following characteristics are associated with* ***OLAP*** *systems (as opposed to OLTP)?*

(A) Queries are complex, involving aggregations over millions of records.

(B) The database schema is typically highly normalized (3NF or higher).

(C) Data is read-mostly; updates are periodic (e.g., nightly ETL loads).

(D) Response time for a single query may range from seconds to minutes.

Problem 109 ***[MCQ]*** *In an OLTP system, the primary design goal for the database schema is:*

(A) To minimize query response time for complex analytical queries

(B) To maximize data redundancy for faster reads

(C) To eliminate data redundancy and ensure update anomaly-free modifications

(D) To store data in a multidimensional cube structure

Problem 110 ***[MSQ]*** *A retailer maintains two systems: System X tracks every sale in real time (normalized schema, row-level locking, thousands of transactions/minute), and System Y stores 5 years of aggregated sales data for trend analysis (denormalized, refreshed nightly). Which of the following are* ***true****?*

(A) System X is an OLTP system; System Y is an OLAP/data-warehouse system.

(B) System Y would typically use a star or snowflake schema.

(C) System X is optimized for ad hoc *analytical queries spanning multiple years.*

(D) Both systems can serve the same purpose if the OLTP database is large enough.

Problem 111 ***[MCQ]*** *Which statement* ***best*** *describes the* subject orientation *of a data warehouse?*

(A) Data is organized around business processes such as invoicing and order entry.

(B) Data is organized around customers, products, and sales, rather than around application functions.

(C) Data is organized to minimize storage by heavy normalization.

(D) Data is organized so that OLTP transactions can be run directly on it.

Problem 112 ***[MSQ]*** *The defining characteristics of a data warehouse (Inmon's definition) are:*

(A) Subject-oriented

(B) Integrated

(C) Time-variant (non-volatile in historical sense)

(D) Volatile

Problem 113 ***[MCQ]*** *The* ***"integrated"*** *property of a data warehouse means:*

(A) Data from multiple heterogeneous sources is cleaned, transformed, and stored in a consistent format.

(B) The warehouse integrates both OLTP and OLAP operations simultaneously.

(C) All data in the warehouse is stored in a single flat table.

(D) The warehouse integrates with the organization's ERP system directly.

Problem 114 ***[MSQ]*** *Compared to OLTP databases, data warehouses typically exhibit which of the following storage and access characteristics?*

(A) Much larger data volumes (terabytes to petabytes vs. gigabytes).

(B) Predominantly read operations with infrequent bulk writes (ETL).

(C) High concurrency of short update transactions.

(D) Use of columnar storage or bitmap indexes for analytical query performance.

Problem 115 ***[MCQ]*** *The* ***"time-variant"*** *property of a data warehouse implies that:*

(A) The warehouse is updated in real time as transactions occur.

(B) Historical data is retained and every record is associated with a time horizon, enabling trend and evolution analysis.

(C) Data automatically expires after a fixed retention period.

(D) The schema changes over time to reflect business evolution.

Problem 116 ***[MSQ]*** *An analyst claims: "We can simply run our OLAP queries directly on the OLTP database and avoid building a separate data warehouse." Which of the following are valid* ***counter-arguments****?*

(A) OLAP queries on OLTP databases would compete with live transactions, degrading transactional throughput.

(B) OLTP schemas (normalized) are inefficient for multi-table aggregation queries.

(C) An OLTP database does not store historical snapshots needed for trend analysis.

(D) OLTP databases cannot store numerical data.

Problem 117 ***[MCQ]*** *In a* ***three-tier data warehouse architecture****, the correct order of tiers from bottom to top is:*

(A) OLAP Server $\rightarrow$ Data Sources $\rightarrow$ Front-End Tools

(B) Data Sources / Bottom Tier $\rightarrow$ Data Warehouse Server / Middle Tier $\rightarrow$ Front-End / Top Tier

(C) ETL Layer $\rightarrow$ Staging Area $\rightarrow$ Data Mart

(D) Metadata Repository $\rightarrow$ Fact Table $\rightarrow$ Dimension Table

Problem 118 ***[MSQ]*** *The* ***ETL (Extract, Transform, Load)*** *process in a data warehouse pipeline involves which of the following activities?*

(A) Extracting data from heterogeneous operational databases and external sources.

(B) Transforming data by cleaning, resolving inconsistencies, and converting formats.

(C) Loading the transformed data into the data warehouse (typically in bulk).

(D) Running OLAP queries on the extracted data before loading it.

Problem 119 ***[MCQ]*** *A* ***data mart*** *differs from a full data warehouse primarily in that:*

(A) A data mart uses OLTP while a data warehouse uses OLAP.

(B) *A data mart is a subject-specific subset of a data warehouse, containing data relevant to a particular department or business function.*

(C) *A data mart stores only raw, un-aggregated transaction data.*

(D) *A data mart requires no ETL process.*

Problem 120 ***[MSQ]*** *Which of the following are components typically found in the* ***bottom tier*** *(data source / warehouse server tier) of a three-tier data warehouse architecture?*

(A) *Operational databases (RDBMS)*

(B) *ETL tools and staging area*

(C) *OLAP server (ROLAP/MOLAP/HOLAP engine)*

(D) *External data feeds (flat files, web data)*

Problem 121 ***[MCQ]*** *In a* ***ROLAP*** *(Relational OLAP) system, multidimensional data is:*

(A) *Stored in pre-computed multidimensional arrays.*

(B) *Stored in relational tables and OLAP operations are implemented via extended SQL (e.g., `GROUP BY CUBE`).*

(C) *Stored in XML documents and queried using XQuery.*

(D) *Stored in a proprietary in-memory format incompatible with SQL.*

Problem 122 ***[MSQ]*** *Compare* ***MOLAP*** *(Multidimensional OLAP) with* ***ROLAP****. Which statements are* ***correct****?*

(A) *MOLAP stores data in multidimensional arrays, offering faster query response for pre-computed aggregations.*

(B) *ROLAP scales better to very large datasets because it leverages mature relational database technology.*

(C) *MOLAP typically requires more storage than ROLAP due to pre-computation of all aggregates.*

(D) *HOLAP is a hybrid that stores detail data relationally and aggregates in multidimensional arrays.*

Problem 123 ***[MCQ]*** *A* ***metadata repository*** *in a data warehouse stores:*

(A) *The actual fact and dimension tables.*

(B) *Descriptive information about data: schema, sources, transformations applied, data lineage, and refresh schedules.*

(C) *Raw operational transaction records before ETL.*

(D) *Only the dimension hierarchies used in drill-down operations.*

Problem 124 ***[MSQ]*** *Which of the following are valid reasons for maintaining a* ***staging area*** *between source systems and the data warehouse?*

(A) *To perform data cleansing and transformation without impacting source system performance.*

(B) *To allow rollback of a failed ETL load without corrupting the warehouse.*

(C) *To serve end-user OLAP queries with low latency.*

(D) *To integrate and reconcile data from multiple heterogeneous sources before loading.*

Problem 125 ***[MCQ]*** *A data cube has dimensions `Time`, `Product`, and `Location` with cardinalities 12, 50, and 30 respectively. The total number of cells in the* ***base cuboid*** *(ignoring ALL-dimension cells) is:*

(A) $12 + 50 + 30 = 92$

(B) $12 \times 50 \times 30 = 18{,}000$

(C) $12 \times 50 \times 30 \times 3 = 54{,}000$

(D) $2^3 \times 18{,}000 = 144{,}000$

Problem 126 ***[NAT]*** *A data cube has $n = 4$ dimensions with cardinalities $d_1 = 10,\ d_2 = 5,\ d_3 = 8,\ d_4 = 4$. The* ***total number of cuboids*** *in the full data cube lattice (including the apex/all cuboid and the base cuboid) is _____.*

Problem 127 ***[MCQ]*** *In a data cube with 3 dimensions (`Time`, `Product`, `Store`), the cuboid corresponding to aggregating over **all** dimensions (i.e., a single aggregate value for the entire dataset) is called:*

(A) Base cuboid

(B) Apex cuboid (0-D cuboid)

(C) Dimension cuboid

(D) Fact cuboid

Problem 128 ***[NAT]*** *A 3-D data cube has dimensions with cardinalities $|Time| = 4$, $|Product| = 6$, $|Location| = 5$. The measure is `Sales`. If the **base cuboid** is fully materialized, the number of cells it contains is _____.*

Problem 129 ***[MSQ]*** *In a data cube, the **lattice of cuboids** captures all possible group-by aggregations. For a 3-D cube with dimensions A, B, C, which of the following are valid cuboids in the lattice?*

(A) (A, B, C) — base cuboid

(B) (A, B) — aggregated over C

(C) $(\ )$ — apex cuboid (all dimensions aggregated)

(D) (A, B, C, D) — where D is a new dimension not in the cube

Problem 130 ***[MCQ]*** *A sales data cube has dimensions `City` (30 values), `Month` (12 values), and `Product` (100 values). The cuboid $(City,\ Month)$ — obtained by aggregating over `Product` — contains how many cells?*

(A) 142

(B) 360

(C) 1200

(D) 36,000

Problem 131 ***[MSQ]*** *Which of the following correctly describe the difference between a **sparse cuboid** and a **dense cuboid** in a data cube?*

(A) A sparse cuboid has many cells with no data (null or zero measure values).

(B) Higher-dimensional cuboids tend to be sparser than lower-dimensional cuboids.

(C) The base cuboid is always the densest cuboid in a data cube.

(D) Sparse cuboids benefit more from compression techniques than dense cuboids.

Problem 132 ***[NAT]*** *A data cube has 5 dimensions. The number of **2-D cuboids** (cuboids with exactly 2 dimensions) in the lattice is _____.*

Problem 133 ***[MCQ]*** *The **"curse of dimensionality"** in the context of data cubes refers to:*

(A) The difficulty of visualizing data in more than 3 dimensions.

(B) The exponential growth in the number of cuboids and cells as the number of dimensions increases, making full materialization infeasible.

(C) The inability of SQL to handle more than 3 `GROUP BY` columns.

(D) The fact that higher-dimensional data always has lower information content.

Problem 134 ***[NAT]*** *A 4-D data cube has dimensions A, B, C, D. The total number of **non-empty cuboids** (excluding the apex cuboid) that contain dimension A is _____.*

Problem 135 ***[MCQ]*** *A **fully materialized data cube** means:*

(A) Only the base cuboid is stored; all other cuboids are computed on demand.

(B) All cuboids in the lattice (all possible group-by aggregations) are pre-computed and stored.

(C) Only the apex cuboid is stored to minimize space.

(D) Cuboids are materialized only when queried for the first time.

Problem 136 ***[MSQ]*** *A **partial materialization** strategy for data cubes:*

(A) Pre-computes and stores only a selected subset of cuboids based on query frequency or storage constraints.

(B) Always materializes all cuboids below a certain dimension threshold.

(C) Can use a dependency graph (lattice) to compute un-materialized cuboids from their nearest materialized ancestors.

(D) Guarantees the same query response time as full materialization.

Problem 137 ***[NAT]*** *Consider a data cube with dimensions `Time` (3 levels: year, quarter, month), `Location` (2 levels: country, city), and `Product` (2 levels: category, item). Counting only dimension-level combinations (not including the ALL level at each dimension), the number of distinct cuboids in the lattice when each dimension can appear at any one of its levels (or be fully rolled up) is _____.*

Hint: Each dimension with L_i levels contributes $L_i + 1$ choices (including the "ALL" choice). Total cuboids $= \prod_i (L_i + 1)$.

Problem 138 ***[MCQ]*** *In the **iceberg cube** optimization, a cuboid cell is retained only if its aggregate value meets a minimum threshold (e.g., `count` ≥ 5). This technique primarily helps by:*

(A) Increasing the number of cells stored in each cuboid.

(B) Reducing the storage and computation cost by pruning cells with low support, which are often uninteresting.

(C) Ensuring all cells are stored for completeness.

(D) Replacing the need for a lattice structure entirely.

Problem 139 ***[MSQ]*** *Which of the following are valid **types of measures** in a data cube?*

(A) Distributive measure (e.g., `COUNT`, `SUM`, `MAX`)

(B) Algebraic measure (e.g., `AVG`, `standard_deviation`)

(C) Holistic measure (e.g., `MEDIAN`, `MODE`, `RANK`)

(D) Structural measure (e.g., schema version number)

Problem 140 ***[NAT]*** *A data cube has dimensions `Time` (12 months), `Product` (80 items), `Store` (25 stores). An **iceberg cube** with threshold `count` ≥ 2 prunes 40% of base cuboid cells. The number of cells **retained** in the base cuboid after iceberg pruning is _____.*

Problem 141 ***[MCQ]*** *A 3-D cube has dimensions A (cardinality 10), B (cardinality 6), C (cardinality 4). The ratio of cells in the **apex cuboid** to cells in the **base cuboid** is:*

(A) $1 : 240$

(B) $1 : 60$

(C) $240 : 1$

(D) $1 : 20$

Problem 142 ***[NAT]*** *A distributive measure f satisfies: $f(\{S_1, S_2, \ldots, S_n\}) = g(\{f(S_1), f(S_2), \ldots, f(S_n)\})$ for some function g. A data cube partition has 4 sub-partitions with `SUM` values: 120, 85, 200, 95. The overall `SUM` is _____.*

Problem 143 ***[NAT]*** *A 3-D cube with dimensions A ($|A| = 5$), B ($|B| = 4$), C ($|C| = 3$) is fully materialized. The total number of cells across **all cuboids** in the lattice (including the apex cuboid, which has 1 cell) is _____.*

Formula: $\prod_{i=1}^{n} (d_i + 1)$ where d_i is the cardinality of dimension i.

Problem 144 ***[MCQ]*** *A **ROLAP** cube over a star schema uses SQL extensions. The SQL clause `GROUP BY CUBE(Time, Product, Location)` generates:*

(A) Only the base cuboid (Time, Product, Location)

(B) All $2^3 = 8$ possible group-by combinations (all cuboids in the lattice)

(C) Only the apex cuboid (single grand total)

(D) Exactly 3 cuboids, one per dimension

Problem 145 ***[MSQ]*** *The SQL `ROLLUP` operator `GROUP BY ROLLUP(A, B, C)` generates which of the following groupings?*

(A) (A, B, C)

(B) (A, B)

(C) (A)

(D) $()$ *(grand total)*

Problem 146 ***[MSQ]*** *Which of the following correctly distinguish* ***ROLLUP*** *from* ***CUBE*** *in SQL?*

(A) `ROLLUP(A,B,C)` *generates* $n + 1 = 4$ *groupings;* `CUBE(A,B,C)` *generates* $2^n = 8$ *groupings (for* $n = 3$ *attributes).*

(B) `ROLLUP` *is order-sensitive (hierarchy matters);* `CUBE` *is order-insensitive (generates all subsets).*

(C) Both `ROLLUP` *and* `CUBE` *generate the same set of groupings for any input.*

(D) `ROLLUP(A,B,C)` *includes the grouping* (B, C) *without* (A).

Problem 147 ***[NAT]*** `GROUP BY CUBE(A, B, C, D)` *generates* x *groupings.* `GROUP BY ROLLUP(A, B, C, D)` *generates* y *groupings. The difference* $x - y$ *(CUBE groupings minus ROLLUP groupings) is* _____.

Problem 148 ***[MCQ]*** *A* ***closed cube*** *(or* closed data cube*) retains only cells that:*

(A) Have measure values equal to zero.

(B) Cannot be further rolled up without losing information, i.e., cells whose measure value is the same as the roll-up.

(C) Have the same measure value as their corresponding ancestor in all higher cuboids.

(D) Are in the base cuboid only.

Problem 149 ***[MSQ]*** *An* ***iceberg cube*** *with minimum threshold* `count` ≥ 3 *is computed over a dataset. Which of the following statements are* ***correct****?*

(A) Cells with `count` < 3 *in any cuboid are pruned and not stored.*

(B) The iceberg cube is always a subset of the full data cube.

(C) Pruning cells in higher-dimensional cuboids may allow pruning corresponding cells in lower-dimensional cuboids (anti-monotone property).

(D) The iceberg cube stores more cells than the full materialized cube.

Problem 150 ***[MCQ]*** *In a* ***star schema****, the central table is called the* ***fact table*** *and the surrounding tables are called* ***dimension tables****. The primary key of the fact table is:*

(A) A surrogate key generated sequentially.

(B) A composite key consisting of foreign keys referencing all dimension tables.

(C) The same as the primary key of the largest dimension table.

(D) A natural key derived from the business domain.

Problem 151 ***[MSQ]*** *A star schema for retail sales has a fact table `SALES` and dimension tables `TIME`, `PRODUCT`, `STORE`, `CUSTOMER`. Which of the following are* ***true*** *about this schema?*

(A) The fact table `SALES` contains foreign keys to all four dimension tables.

(B) Dimension tables in a star schema are not *in 3NF (they may be denormalized).*

(C) A join between `SALES` and `PRODUCT` requires traversing intermediate normalized tables.

(D) The star schema enables simpler (fewer joins) queries than a snowflake schema for the same data.

Problem 152 ***[MCQ]*** *A* ***snowflake schema*** *differs from a star schema primarily in that:*

(A) The fact table is replaced by multiple smaller fact tables.

(B) Dimension tables are normalized into multiple related tables, reducing redundancy but requiring more joins for queries.

(C) The snowflake schema does not use foreign keys.

(D) The snowflake schema stores pre-aggregated data at each level.

Problem 153 ***[MSQ]*** *Compare a* ***star schema*** *and a* ***snowflake schema****. Which statements are* ***correct****?*

(A) A star schema has denormalized dimension tables; a snowflake schema has normalized dimension tables.

(B) Queries on a star schema generally require fewer joins than equivalent queries on a snowflake schema.

(C) A snowflake schema uses less storage than a star schema due to normalization.

(D) A star schema is harder to maintain when dimension data changes frequently.

Problem 154 ***[NAT]*** *A star schema has a fact table and 5 dimension tables. Each dimension table is joined to the fact table on a single foreign key. A query requires data from the fact table and all 5 dimension tables. The minimum number of* `JOIN` *operations required to answer this query is _____.*

Problem 155 ***[MCQ]*** *The* ***galaxy schema*** *(also called* fact constellation schema*) is characterized by:*

(A) Multiple fact tables that share dimension tables.

(B) A single fact table surrounded by highly normalized dimension tables.

(C) Dimension tables that are themselves fact tables.

(D) No foreign key relationships between fact and dimension tables.

Problem 156 ***[MSQ]*** *Examine the following partial schema:*

Table	***Columns***
`SALES_FACT`	`time_id, prod_id, store_id, dollars_sold, units_sold`
`TIME_DIM`	`time_id, day, month, quarter, year`
`PRODUCT_DIM`	`prod_id, prod_name, category_id`
`CATEGORY_DIM`	`category_id, category_name, dept_id`
`DEPT_DIM`	`dept_id, dept_name`

Which of the following are ***correct*** *about this schema?*

(A) This is a snowflake schema because `PRODUCT_DIM` *references* `CATEGORY_DIM`*, which references* `DEPT_DIM`*.*

(B) `SALES_FACT` *stores two measures:* `dollars_sold` *and* `units_sold`*.*

(C) A query for total sales by department requires at least 4 joins.

(D) This schema wastes storage compared to a star schema with a single flat `PRODUCT_DIM`*.*

Problem 157 ***[NAT]*** *A snowflake schema has a fact table and a dimension* `Product` *normalized into 3 levels:* `Product` $\rightarrow$ `Category` $\rightarrow$ `Department`*. A query for total sales by department requires joining* `FACT`*,* `Product`*,* `Category`*, and* `Department` *tables. The number of* `JOIN` *operations for this single dimension chain is _____.*

Problem 158 ***[MCQ]*** *In a star schema fact table for sales with dimensions* `Date` *(1095 values, 3 years daily),* `Product` *(200 items),* `Store` *(50 stores), and assuming every combination has at least one sale, the maximum number of rows in the fact table is:*

(A) $1095 + 200 + 50 = 1345$

(B) $1095 \times 200 \times 50 = 10{,}950{,}000$

(C) $1095 \times 200 = 219{,}000$

(D) $200 \times 50 = 10{,}000$

Problem 159 ***[NAT]*** *A galaxy schema contains 2 fact tables: `SALES` and `INVENTORY`. `SALES` shares dimension tables `TIME`, `PRODUCT`, `STORE`. `INVENTORY` shares `TIME` and `PRODUCT` but has its own `WAREHOUSE` dimension. The total number of distinct tables in this galaxy schema (fact + dimension) is _____.*

Problem 160 ***[MCQ]*** *A concept hierarchy for the `Location` dimension is:*

$$\textit{street} \rightarrow \textit{city} \rightarrow \textit{state} \rightarrow \textit{country} \rightarrow \textit{all}$$

*The number of **levels** in this concept hierarchy (including all) is:*

(A) *4*

(B) *5*

(C) *6*

(D) *3*

Problem 161 ***[MSQ]*** *In OLAP, concept hierarchies serve which of the following purposes?*

(A) *They enable* drill-down *and* roll-up *operations along a dimension.*

(B) *They define the granularity levels at which data can be summarized.*

(C) *They eliminate the need for fact tables in a star schema.*

(D) *They support viewing data at different levels of abstraction.*

Problem 162 ***[MCQ]*** *The `Time` dimension has the hierarchy: second $\rightarrow$ minute $\rightarrow$ hour $\rightarrow$ day $\rightarrow$ week $\rightarrow$ month $\rightarrow$ quarter $\rightarrow$ year $\rightarrow$ all. A user is viewing sales at the `month` level and performs a **roll-up**. The next level of data they would see is:*

(A) *day*

(B) *week*

(C) *quarter*

(D) *year*

Problem 163 ***[NAT]*** *The `Time` hierarchy has 9 levels (including all at the top and second at the bottom). A user currently views data at the `day` level (which is the 4th level from the bottom, i.e., level 4, where level 1 = second). How many **drill-down** steps are needed to reach the `second` level? _____ steps.*

Problem 164 ***[MSQ]*** *A concept hierarchy for `Salary` is defined as:*

$$\textit{low}\ (< \$30K), \quad \textit{medium}\ (\$30K\text{–}\$60K), \quad \textit{high}\ (> \$60K)$$

This type of concept hierarchy is an example of:

(A) *A schema hierarchy defined by a total order on a continuous attribute.*

(B) *A discretization hierarchy mapping numerical ranges to categorical labels.*

(C) *A set-grouping hierarchy.*

(D) *A user-defined hierarchy based on domain knowledge.*

Problem 165 ***[NAT]*** *A concept hierarchy for `Location` is: city (200) $\rightarrow$ state (20) $\rightarrow$ country (4) $\rightarrow$ all (1). On average, how many **cities per country** does this hierarchy imply? _____ cities per country.*

Problem 166 ***[MSQ]*** *A **user-defined** concept hierarchy for `Season` is:*

$\{Dec, Jan, Feb\} \rightarrow$ *Winter*, $\{Mar, Apr, May\} \rightarrow$ *Spring*, $\{Jun, Jul, Aug\} \rightarrow$ *Summer*, $\{Sep, Oct, Nov\} \rightarrow$ *Fall*

*Which of the following are **true**?*

(A) *This is a set-grouping hierarchy mapping months to seasons.*

(B) A roll-up from month to season reduces 12 attribute values to 4.

(C) This hierarchy can be represented as a functional dependency: $\texttt{Month} \rightarrow \texttt{Season}$.

(D) Drilling down from Summer would expose the three months June, July, August.

Problem 167 ***[MCQ]*** *A measure M is **distributive** if there exists a function g such that $M(S_1 \cup S_2 \cup \ldots \cup S_n) = g(M(S_1), M(S_2), \ldots, M(S_n))$. Which of the following is a distributive measure?*

(A) `AVERAGE`

(B) `MEDIAN`

(C) `SUM`

(D) `STANDARD_DEVIATION`

Problem 168 ***[MSQ]*** *Which of the following measures are **distributive**?*

(A) `COUNT`

(B) `MAX`

(C) `MIN`

(D) `AVG`

Problem 169 ***[MCQ]*** *An **algebraic measure** can be computed by applying an algebraic function to a fixed number of distributive measures. Which of the following is an algebraic measure?*

(A) `COUNT`

(B) `AVG` *(=* `SUM` */* `COUNT`*)*

(C) `MEDIAN`

(D) `MODE`

Problem 170 ***[MSQ]*** *Which of the following measures are **holistic** (i.e., cannot be computed from a fixed number of sub-aggregate values)?*

(A) `MEDIAN`

(B) `MODE`

(C) `RANK`

(D) `SUM`

Problem 171 ***[MCQ]*** *A data analyst wants to compute the **median** of a very large dataset stored in a distributed data cube. She divides data into 10 partitions and computes the median of each partition. She then takes the median of the 10 partition medians. This approach:*

(A) Is correct, because median is a distributive measure.

(B) Is incorrect, because median is a holistic measure; the median of medians does not equal the overall median.

(C) Is correct only if all partitions have equal size.

(D) Always underestimates the true median.

Problem 172 ***[MSQ]*** *Which of the following statements about **distributive**, **algebraic**, and **holistic** measures are **correct**?*

(A) Every distributive measure is also algebraic.

(B) `standard_deviation` *is algebraic because it can be computed from* `SUM`*,* `SUM_OF_SQUARES`*, and* `COUNT`*.*

(C) A holistic measure requires examining all individual data values to compute exactly; it cannot be computed from fixed-size sub-aggregates.

(D) `MAX` *is holistic because taking the max of partition maxima does not always yield the global max.*

Problem 173 ***[MCQ]*** *The measure* ***profit margin*** *defined as* $\texttt{profit\_margin} = (\texttt{revenue} - \texttt{cost})/\texttt{revenue}$ *is classified as:*

(A) Distributive, because revenue and cost are distributive.

(B) Algebraic, because it is a ratio of two algebraic (or distributive) measures.

(C) Holistic, because ratios cannot be aggregated from sub-partitions.

(D) Neither; ratio measures are not supported in data cubes.

Problem 174 ***[MCQ]***

A sales data cube has dimensions `Time` *(hierarchy: day* $\rightarrow$ *month* $\rightarrow$ *quarter* $\rightarrow$ *year),* `Location` *(hierarchy: city* $\rightarrow$ *state* $\rightarrow$ *country), and* `Product` *(hierarchy: item* $\rightarrow$ *brand* $\rightarrow$ *category). The measure is* `total_sales`*. Currently the user views data at granularity: (quarter, state, brand)*

Starting from (quarter, state, brand), the user performs a ***roll-up*** *on* `Time` *to the next level. The new view granularity is:*

(A) (month, state, brand)

(B) (year, state, brand)

(C) (quarter, country, brand)

(D) (quarter, state, category)

Problem 175 ***[MCQ]***

A sales data cube has dimensions `Time` *(hierarchy: day* $\rightarrow$ *month* $\rightarrow$ *quarter* $\rightarrow$ *year),* `Location` *(hierarchy: city* $\rightarrow$ *state* $\rightarrow$ *country), and* `Product` *(hierarchy: item* $\rightarrow$ *brand* $\rightarrow$ *category). The measure is* `total_sales`*. Currently the user views data at granularity: (quarter, state, brand)*

From (quarter, state, brand), the user performs a ***drill-down*** *on* `Location`*. The new view granularity is:*

(A) (quarter, country, brand)

(B) (quarter, city, brand)

(C) (month, state, brand)

(D) (quarter, state, item)

Problem 176 ***[MCQ]***

A sales data cube has dimensions `Time` *(hierarchy: day* $\rightarrow$ *month* $\rightarrow$ *quarter* $\rightarrow$ *year),* `Location` *(hierarchy: city* $\rightarrow$ *state* $\rightarrow$ *country), and* `Product` *(hierarchy: item* $\rightarrow$ *brand* $\rightarrow$ *category). The measure is* `total_sales`*. Currently the user views data at granularity: (quarter, state, brand)*

The user applies a ***slice*** *operation by fixing* `Time = Q1 (first quarter)`*. The result is:*

(A) A 3-D cube restricted to only the Q1 slice, showing all states and brands for Q1 only.

(B) A 2-D table (state $\times$ *brand) for Q1, because one dimension is fixed.*

(C) All quarters are retained but sorted by sales.

(D) The Time dimension is removed from the schema entirely.

Problem 177 ***[MCQ]***

A sales data cube has dimensions `Time` *(hierarchy: day* $\rightarrow$ *month* $\rightarrow$ *quarter* $\rightarrow$ *year),* `Location` *(hierarchy: city* $\rightarrow$ *state* $\rightarrow$ *country), and* `Product` *(hierarchy: item* $\rightarrow$ *brand* $\rightarrow$ *category). The measure is* `total_sales`*. Currently the user views data at granularity: (quarter, state, brand)*

The user applies a ***dice*** *operation with conditions:* `Time` $\in \{Q1, Q2\}$ *AND* `Location` $\in \{$*California, Texas*$\}$*. The result is:*

(A) A single value (total sales for Q1+Q2 in CA+TX).

(B) A sub-cube retaining all three dimensions but restricted to the specified dimension value ranges.

(C) A 1-D vector of sales by brand.

(D) Identical to a slice operation.

Problem 178 ***[MSQ]***

*Which of the following correctly distinguish **slice** from **dice**?*

(A) Slice selects a single value on one dimension (reducing dimensionality by 1); dice selects a range of values on two or more dimensions (preserving all dimensions).

(B) Both slice and dice are selection operations on the cube.

(C) Dice always results in a lower-dimensional sub-space than slice.

(D) Slicing on all dimensions simultaneously is equivalent to looking up a single cell in the cube.

Problem 179 ***[MCQ]*** *The **pivot** (rotate) OLAP operation:*

(A) Aggregates data along one dimension to produce a lower-dimensional cube.

(B) Rotates the data axes, providing an alternative presentation of the data without changing the data values.

(C) Filters rows in the fact table based on a `WHERE` clause.

(D) Joins two fact tables on a shared dimension.

Problem 180 ***[MSQ]*** *Which of the following OLAP operations **change the granularity** of the data currently being viewed?*

(A) Roll-up

(B) Drill-down

(C) Slice (fixing one dimension to a single value)

(D) Pivot (rotate)

Problem 181 ***[NAT]*** *Starting from the base cuboid at granularity (day, city, item), the user performs: roll-up on `Time` by 2 levels, roll-up on `Location` by 1 level, roll-up on `Product` by 1 level. Determine the final granularity (write the levels as a triplet): After 2 roll-ups on Time from day: day $\rightarrow$ month $\rightarrow$ quarter. After 1 roll-up on Location from city: city $\rightarrow$ state. After 1 roll-up on Product from item: item $\rightarrow$ brand. The number of **total roll-up steps performed** is* _____.

Problem 182 ***[MCQ]*** *A user views a 3-D sales cube (Time $\times$ Location $\times$ Product). After applying a **slice** on `Location` = ``India'', the resulting data structure has:*

(A) 3 dimensions

(B) 2 dimensions

(C) 1 dimension

(D) 0 dimensions (a single scalar)

Problem 183 ***[NAT]*** *A sales cube at granularity (month, country, category) shows:*

Month	*Country*	*Category*	*Sales ($K)*
Jan	*India*	*Electronics*	*500*
Jan	*India*	*Clothing*	*200*
Jan	*USA*	*Electronics*	*800*
Jan	*USA*	*Clothing*	*350*
Feb	*India*	*Electronics*	*450*
Feb	*India*	*Clothing*	*180*
Feb	*USA*	*Electronics*	*900*
Feb	*USA*	*Clothing*	*400*

*After a **roll-up on Time** from month to all, the total `Sales` for (India, Electronics) is _____ $K.*

Problem 184 ***[NAT]*** *Using the same data cube as Q183, after a **slice** on `Country = ''USA''` and then a **roll-up on Time**, the total `Sales` for USA across **all** categories is _____ $K.*

Problem 185 ***[MCQ]*** *A user applies a **dice** operation: `Month` $\in \{$Jan$\}$, `Country` $\in \{$India, USA$\}$, `Category` $\in \{$Electronics$\}$. Using the data from Q183, the **total sales** in this dice sub-cube is:*

(A) $800K

(B) $1300K

(C) $1500K

(D) $1700K

Problem 186 ***[MSQ]*** *Which of the following sequences of OLAP operations are **logically valid**?*

(A) Drill-down followed by roll-up on the same dimension returns to the original view.

(B) Slice followed by pivot changes neither the data values nor the dimensions present.

(C) Roll-up to the apex cuboid followed by drill-down to any cuboid is always possible.

(D) Dice followed by roll-up is equivalent to first rolling up and then dicing.

Problem 187 ***[MCQ]*** *An analyst is exploring sales trends. She starts with yearly totals and wants to understand which specific day in a particular month had the highest sales. The correct sequence of operations is:*

(A) Roll-up $\rightarrow$ Slice $\rightarrow$ Roll-up

(B) Slice (fix year, month) $\rightarrow$ Drill-down to day level

(C) Pivot $\rightarrow$ Dice $\rightarrow$ Roll-up

(D) Drill-through $\rightarrow$ Pivot $\rightarrow$ Slice

Problem 188 ***[NAT]*** *A data cube at the (year, country, category) level has the following entries (showing only `COUNT` of transactions):*

Year	Country	Category	Count
2023	India	Electronics	1200
2023	India	Clothing	800
2023	USA	Electronics	2500
2023	USA	Clothing	1500

*After rolling up on **both** `Country` and `Category` (i.e., summing across all countries and all categories for year 2023), the total `COUNT` is _____.*

Problem 189 ***[NAT]*** *A 5-D data cube has equal cardinality d for every dimension. If the total number of cells across all cuboids is 7776, the value of d is _____.*

3.7 Try it Yourself

Exercise 105 ***[NAT]*** *A data cube has $n = 6$ dimensions. The total number of cuboids in the complete lattice (including the apex cuboid and the base cuboid) is _____.*

Exercise 106 ***[NAT]*** *A 4-D data cube has dimension cardinalities $|A| = 10$, $|B| = 8$, $|C| = 5$, $|D| = 4$. The total number of cells in the **base cuboid** (A, B, C, D) is _____.*

Exercise 107 ***[NAT]*** *Using the same cube as Q2, the total number of cells across* ***all cuboids*** *in the complete lattice is given by* $\prod_{i=1}^{4}(d_i+1)$*. This value is* _____.

Exercise 108 ***[MCQ]*** *A 3-D cube with* $|A|=5,\ |B|=4,\ |C|=3$ *is fully materialized. The 1-D cuboid* (B) *— obtained by aggregating over both* A *and* C *— contains how many cells?*

(A) 3

(B) 4

(C) 5

(D) 12

Exercise 109 ***[NAT]*** *A data cube has 4 dimensions with cardinalities* $5, 4, 3, 2$*. The number of* ***3-D cuboids*** *(cuboids involving exactly 3 dimensions) in the lattice is* _____.

Hint: $\binom{4}{3}=4$ *cuboids; count them.*

Exercise 110 ***[NAT]*** *A data cube has 5 dimensions* A, B, C, D, E *each with cardinality 3. The number of cells in the 2-D cuboid* (A, C) *is* _____.

Exercise 111 ***[NAT]*** *For a 4-D cube with dimensions of cardinalities* d_1, d_2, d_3, d_4*, the number of* ***1-D cuboids*** *is* $\binom{4}{1}=4$*. If* $d_1=6,\ d_2=4,\ d_3=3,\ d_4=5$*, the* ***total number of cells*** *across* all *1-D cuboids is* _____.

Exercise 112 ***[NAT]*** *A data cube has 3 dimensions with concept hierarchies:* `Time` *has 4 levels (day, month, quarter, year) + all,* `Location` *has 3 levels (city, state, country) + all,* `Product` *has 2 levels (item, category) + all. The total number of distinct cuboids in the concept-hierarchy lattice (each dimension can be at any one level or all) is* _____.

Exercise 113 ***[NAT]*** *A 3-D cube (*`Year`*: 5,* `Quarter`*: 4,* `Region`*: 8) is partially materialized. Only cuboids with* ***at most 2 dimensions*** *are stored. The total number of cells stored across all materialized cuboids (including the apex) is* _____.

Exercise 114 ***[NAT]*** *A star schema has a* `SALES` *fact table with 4 dimension tables:* `TIME` *(365 rows),* `PRODUCT` *(120 rows),* `STORE` *(40 rows),* `CUSTOMER` *(5000 rows). Assuming every combination of (Time, Product, Store) has exactly one fact row (customers are not part of the fact key), the number of rows in* `SALES` *is* _____.

Exercise 115 ***[NAT]*** *In the same schema as above, each row in* `SALES` *occupies 40 bytes (4 foreign keys* $\times$ *4 bytes each* $+$ *2 measures* $\times$ *8 bytes each* $+$ *8 bytes overhead). The total storage for the* `SALES` *fact table in* ***megabytes*** *(rounded to two decimal places) is* _____ *MB.*

Exercise 116 ***[MCQ]*** *A snowflake schema normalizes the* `PRODUCT` *dimension of a star schema into three tables:* `PRODUCT` *(500 rows, 3 cols),* `CATEGORY` *(50 rows, 2 cols),* `DEPARTMENT` *(10 rows, 2 cols). The equivalent denormalized* `PRODUCT` *dimension table in a star schema would have how many rows?*

(A) 10

(B) 50

(C) 500

(D) 560

Exercise 117 ***[NAT]*** *A galaxy schema has two fact tables:* `SALES` *and* `RETURNS`*. They share dimension tables* `TIME` *and* `PRODUCT`*.* `SALES` *additionally has* `STORE`*;* `RETURNS` *additionally has* `REASON`*. The total number of* ***distinct tables*** *(fact + dimension) in this galaxy schema is* _____.

Exercise 118 ***[MCQ]*** *In a star schema, the* `TIME` *dimension table has attributes:* `time_id` *(PK),* `day`*,* `month`*,* `quarter`*,* `year`*,* `day_of_week`*,* `is_holiday`*. This dimension table covers 3 years of daily data. The number of rows in* `TIME` *(assuming 365 days/year) is:*

(A) 12

(B) 36

(C) 365

(D) 1095

Exercise 119 ***[NAT]*** *A fact table has a composite primary key of 5 foreign keys (4 bytes each) plus 3 measure columns (8 bytes each). Ignoring page headers, the size of one fact row is _____ bytes.*

Exercise 120 ***[NAT]*** *A snowflake schema has `PRODUCT` → `SUBCATEGORY` → `CATEGORY` → `DEPARTMENT` as a 4-level chain. A query joining the fact table to all 4 product-chain tables requires _____ `JOIN` operations along this chain alone.*

Exercise 121 ***[NAT]*** *A star schema fact table has 10 million rows. Each row has 6 columns: 4 foreign keys (INT, 4 bytes each) and 2 measures (FLOAT, 8 bytes each). The total raw storage (uncompressed) for the fact table in **gigabytes** (rounded to three decimal places) is _____ GB.*

Exercise 122 ***[MCQ]*** *A slowly changing dimension (SCD) Type 2 implementation adds a new row for each change. A `CUSTOMER` dimension had 10,000 original rows. Over one year, 5% of customers changed their address (one change each), and 2% changed their address **twice**. The total number of rows in the `CUSTOMER` dimension after one year is:*

(A) 10,000

(B) 10,700

(C) 10,900

(D) 11,400

Exercise 123 ***[NAT]*** *The SQL statement `GROUP BY CUBE(A, B, C, D, E)` generates 2^5 grouping sets. The number of **non-empty grouping sets** (excluding the empty set / grand total) is _____.*

Exercise 124 ***[NAT]*** *`GROUP BY ROLLUP(Region, Country, City)` generates $n + 1$ groupings for $n = 3$ attributes. List how many result rows are produced if: Region has 2 values, Country has 4 values (2 per region), City has 8 values (2 per country). Total rows from `ROLLUP` (including subtotals and grand total) is _____.*

Rows: (Region, Country, City) level: 8; (Region, Country): 4; (Region): 2; (): 1.

Exercise 125 ***[NAT]*** *`GROUP BY GROUPING SETS((A,B),(A),(B),())` generates 4 grouping sets. A table has $|A| = 3$ and $|B| = 4$ distinct values. The **maximum** total number of output rows (assuming all combinations exist) is _____.*

(A, B): $3 \times 4 = 12$; (A): 3; (B): 4; $()$: 1.

Exercise 126 ***[MCQ]*** *How many grouping sets does `GROUP BY ROLLUP(A,B,C,D)` generate?*

(A) 4

(B) 5

(C) 8

(D) 16

Exercise 127 ***[NAT]*** *A sales table has the following data:*

Region	Product	Quarter	Sales
North	X	Q1	100
North	X	Q2	150
North	Y	Q1	200
South	X	Q1	120
South	Y	Q2	180

`SELECT Region, SUM(Sales) FROM sales GROUP BY ROLLUP(Region)` produces a grand total row. The value of the grand total `SUM(Sales)` is _____.

Exercise 128 ***[NAT]*** *Using the data above, the result of `SELECT Region, Product, SUM(Sales) FROM sales GROUP BY CUBE(Region, Product)` produces how many output rows (including subtotals and grand total)? _____.*

Groupings: (Region, Product), (Region), (Product), (); count distinct combinations at each level.

Exercise 129 ***[MCQ]*** *A table has attributes A (5 distinct values), B (3 distinct values), C (4 distinct values). The query `GROUP BY CUBE(A,B,C)` produces at most how many output rows in total across all grouping sets?*

(A) 60

(B) 119

(C) 120

(D) 96

Exercise 130 ***[NAT]*** *`GROUP BY CUBE(A,B)` produces at most $(|A|+1)(|B|+1)$ rows. For $|A|=4$ and $|B|=6$, this is _____.*

Exercise 131 ***[NAT]*** *Five partitions of a dataset have the following `COUNT` and `SUM_OF_SQUARES` (SS) and `SUM` of attribute X:*

Partition	COUNT	SUM	SS
P_1	10	50	310
P_2	20	100	620
P_3	15	90	580
P_4	5	20	120
P_5	10	40	200

*The overall **population variance** $\sigma^2 = \frac{\sum SS}{N} - \left(\frac{\sum SUM}{N}\right)^2$ (rounded to two decimal places) is _____.*

Exercise 132 ***[NAT]*** *Four regional sales cubes report:*

Region	COUNT	SUM (Revenue $K)
North	200	4000
South	150	2700
East	300	5400
West	100	1500

*The overall **average revenue** per transaction across all regions (rounded to two decimal places) is $_____K.*

Exercise 133 ***[NAT]*** *From the data above, the overall **MAX revenue** in $K is not directly computable from the regional SUMs and COUNTs — this illustrates that `MAX` is **distributive** (the global max is the max of partition maxima). If regional `MAX` values are: North = $35K, South = $28K, East = $42K, West = $31K, the overall `MAX` is _____ $K.*

Exercise 134 ***[MCQ]*** *Three data partitions have `AVG` values: $\bar{x}_1 = 50$ (size 10), $\bar{x}_2 = 80$ (size 20), $\bar{x}_3 = 60$ (size 30). The overall (weighted) **average** is:*

(A) 63.33

(B) 65.00

(C) 66.67

(D) 70.00

Exercise 135 ***[NAT]*** *A data cube partition is split into 3 sub-partitions with COUNT values $c_1 = 100,\ c_2 = 150,\ c_3 = 250$. The overall COUNT (a distributive measure) is _____.*

Exercise 136 ***[NAT]*** *Five sub-partitions of a data cube have MAX values: $28, 45, 33, 51, 39$. The global MAX is _____.*

Exercise 137 ***[NAT]*** *Three sub-partitions have MIN values: $12, 7, 19$. The global MIN is _____.*

Exercise 138 ***[NAT]*** *A measure is defined as $M = \texttt{SUM\_OF\_SQUARES}/\texttt{COUNT} - (\texttt{AVG})^2$ (population variance). Sub-partition data:*

Part.	*SUM*	*COUNT*
P_1	*300*	*6*
P_2	*420*	*7*

*The overall **AVG** across both partitions (rounded to two decimal places) is _____.*

Exercise 139 ***[MCQ]*** *A data warehouse computes the measure $\texttt{profit\_ratio} = \texttt{SUM(profit)}/\texttt{SUM(revenue)}$ separately for each region: North = 0.25, South = 0.30, East = 0.22. The overall profit ratio **cannot** be computed as the average of these three ratios because:*

(A) Ratios are always holistic measures that cannot be aggregated.

(B) The regional SUM(revenue) values differ, so a simple average of ratios is not the correct weighted combination.

(C) Profit ratios are ordinal and cannot be averaged.

(D) The SUM function is not distributive.

Exercise 140 ***[NAT]*** *Three regions have SUM(profit) and SUM(revenue) as follows: North: \$150K profit on \$600K revenue; South: \$90K profit on \$300K revenue; East: \$110K profit on \$500K revenue. The correct overall **profit ratio** (profit / revenue, as a percentage rounded to two decimal places) is _____%.*

Common Data for Q141–150.

A sales data cube at granularity (*Year, Quarter, Region, Product_Category*) contains the following aggregated sales figures (\$K):

Year	Quarter	Region	Category	Sales ($K)
2022	Q1	North	Electronics	400
2022	Q1	North	Clothing	150
2022	Q1	South	Electronics	300
2022	Q1	South	Clothing	200
2022	Q2	North	Electronics	500
2022	Q2	North	Clothing	180
2022	Q2	South	Electronics	350
2022	Q2	South	Clothing	220
2023	Q1	North	Electronics	450
2023	Q1	North	Clothing	170
2023	Q1	South	Electronics	320
2023	Q1	South	Clothing	210
2023	Q2	North	Electronics	550
2023	Q2	North	Clothing	190
2023	Q2	South	Electronics	380
2023	Q2	South	Clothing	240

Exercise 141 ***[NAT] Roll-up on Year*** *(aggregate 2022 and 2023 into a single grand total). The total sales across* ***all*** *years, regions, and categories ($K) is _____.*

Exercise 142 ***[NAT] Slice: Year = 2022****. After slicing on Year = 2022, perform a* ***roll-up on Quarter*** *(sum Q1 and Q2). The total sales ($K) for* ***(2022, North, Electronics)*** *is _____.*

Exercise 143 ***[NAT] Dice****: $\texttt{Year} \in \{2022\}$, $\texttt{Region} =$ North. The total sales ($K) in this sub-cube across both quarters and both categories is _____.*

Exercise 144 ***[NAT] Roll-up on Category*** *(aggregate Electronics + Clothing). For (Year=2023, Quarter=Q2, Region=South), the rolled-up total sales ($K) is _____.*

Exercise 145 ***[NAT] Roll-up on Region*** *(aggregate North + South). For (Year=2022, Quarter=Q1, Category=Electronics), the rolled-up total ($K) is _____.*

Exercise 146 ***[MCQ]*** *After rolling up* ***both*** $\texttt{Quarter}$ *and* $\texttt{Region}$ *(summing all quarters, summing North+South), the total sales ($K) for (Year=2022, Category=Clothing) is:*

(A) 550

(B) 630

(C) 750

(D) 780

Exercise 147 ***[NAT] Drill-down on Quarter****: the user is viewing (Year=2023, Region=North, Category=Electronics) with Quarter rolled up. The rolled-up value ($K) is _____ (sum of Q1 and Q2 for this slice).*

Exercise 148 ***[NAT] Percentage change****: After rolling up on Quarter and Region, total Electronics sales in* ***2023*** *vs.* ***2022*** *are compared.*

$$\%\ \textit{change} = \frac{\textit{2023 total} - \textit{2022 total}}{\textit{2022 total}} \times 100$$

Total Electronics sales (both regions, both quarters): 2022 $= 400 + 300 + 500 + 350 = 1550$*\$K; 2023* $= 450 + 320 + 550 + 380 = 1700$*\$K. The percentage change (rounded to two decimal places) is* _____*%.*

Exercise 149 ***[NAT] Pivot****: a 2-D table is displayed with* `Quarter` *as rows and* `Category` *as columns, after* ***slicing*** *on* `Year=2022` *and* `Region=North`*. The value in the cell (Q2, Clothing) is* _____ *\$K.*

Exercise 150 ***[NAT] Contribution analysis****: After rolling up all dimensions for Year=2022, the grand total is the answer to Q42 + Q43 adapted for 2022. Total 2022 sales* $= 400 + 150 + 300 + 200 + 500 + 180 + 350 + 220 = 2300$*\$K. The* ***percentage contribution*** *of (2022, Q1, North, Electronics) to the 2022 grand total (rounded to two decimal places) is* _____*%.*

3.8 YouTube Links and QR Codes

Lecture	Details	YouTube Link	QR Code
12	OLTP vs OLAP — Data Warehouse Architecture — Data Warehousing	https://youtu.be/hko7Ncxl_dA	
13	Data Cubes & Types of Cuboids — Data Warehousing	https://youtu.be/QJo1cXEAIns	
14	Schemas for Multidimensional Data Model — Star, Snowflake & Fact Constellation — DW	https://youtu.be/hT9GC7bJyJ0	
15	Concept Hierarchy — Data Generalization & Levels of Abstraction — DW	https://youtu.be/ND7B9ZHanwA	
16	Measures — Categorization & Computation of Measures — DW	https://youtu.be/BjQUV6dIqnI	

17	OLAP Operations — Roll-Up, Drill-Down, Slice, Dice & Pivot — DW	https://youtu.be/hrwwOibflXg	
18	Problem Solving on Data Warehousing	https://youtu.be/wtbJY3Wxx0w	

Chapter 4

Solutions to Practice Problems

Problems Covered	YouTube Link	QR Code
Problems 1–32 (Lecture 2)	https://youtu.be/7R-hZ1EfLXM	
Problems 33–106 (Lecture 11)	https://youtu.be/VpH49WkvPNE	
Problems 107–189 (Lecture 18)	https://youtu.be/wtbJY3Wxx0w	

Bibliography

[1] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 4th ed., Morgan Kaufmann, 2022.

[2] C. C. Aggarwal, *Data Mining: The Textbook*, Springer, Cham, Switzerland, 2015.

GateXAIML

Free GATE resources for Data Science & AI and CSE
Website: www.gatexaiml.in
Email: contact@gatexaiml.in