Unit 2: Data Munging

Dipesh Koirala

Outline

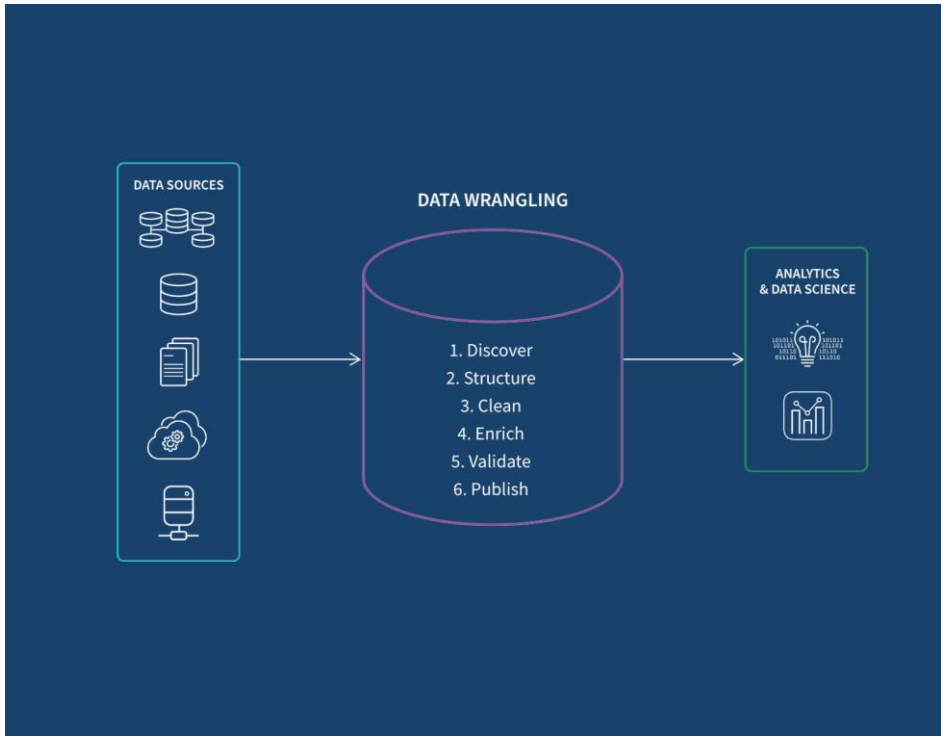
- Data quality, common issues with real world data: Duplicates, Missing Data, Non standard data, Unit mismatch;
- Ways to clean up and standardize data;
- Data enrichment: Need for data enrichment; Common ways to enrich data: correction, extrapolation, augmentation;
- Data Validation: Common methods of data validation: type check, range & constraint check, consistency check;
- Data format conversion: Commonly used formats: JSON, XML, Tabular, Relational - their strengths and weaknesses;
- Motivation behind format conversion. General methods of conversion between data formats. Tabular data: Row based vs column based (Parquet, ORC, CSV). Wide vs narrow(long) table format. Converting between wide vs narrow formats

Data Munging

- Also known as data wrangling, is the process of cleaning, structuring and transforming raw data into a more usable form.

- **Involves tasks such as** handling missing, inconsistent data, formatting data types, integrating datasets to prepare data for further exploration and modeling in data analysis or machine learning projects.

- The goal of data munging is to prepare data, ensuring it is accurate, complete, and in a format that can be easily used.



https://www.qlik.com/us/data-management/data-wrangling

Data Quality

- Data quality refers to the state of data that determine how well suited the data is to meet the purpose.
- It is a measure of the condition of data in terms of following key dimensions

Key Dimensions:

- Data Accuracy
- Completeness
- Consistency
- Relevance
- Uniqueness
- Validity

- **High data quality** ensures that analyses are valid and actionable, leading to better insights and outcomes.
- Poor data quality, on the other hand, can result in flawed analyses, wasted resources, and *misguided strategies.*

Common Issues with Real World Data

i. Duplicates
ii. Incomplete Data/Missing Data
iii. Non-standard Data
iv. Unit Mismatch
v. Inaccurate Data
vi. Outliers
vii. Noisy Data
viii. Irrelevant Data
ix. Data security and Privacy Issues

Data Cleaning Techniques

- Is a crucial step in data science, ensuring that datasets are accurate, consistent and ready for analysis.
- Some common techniques used are mentioned here:

1. Removing Duplicates:

- Identify and eliminate duplicate entries *to ensure each data point is unique.* This helps prevent skewed results in analyses.

Data Cleaning Techniques

2. Handling Missing Data:

- **Deletion**: Remove records with missing values if they are not significant to the analysis.

- **Imputation:**
  - Heuristic-based
  - Mean Value/Mode Value
  - Random Value
  - By nearest neighbor

- Interpolation
- Forward Fill
- Backward Fill

Data Cleaning Techniques

3. Handling Non-Standard Data:

- Non-standard data are those that doesn't conform to consistent format, structure or convention.
- **E.g.,**
  - Variations in date formats,
  - inconsistent naming conventions,
  - mixed data types within a single field

- Non-standard data leads to inconsistency
- **Inconsistent data** contradicts itself or varies in representation. (e.g., misspelling)

Data Cleaning Techniques

3. Handling Non – Standard Data:

- Fix *typos, mislabeling, and inconsistencies* in data formats to standardize the dataset.
- Standard Date formats
- Single Data type
- Same convention for values

Standard Date format

```
df["Date"] = pd.to_datetime(df["Date"], format="%Y-%m-%d")
```

```
salarydata['Education Level'] = salarydata['Education Level'].replace({
    "Bachelor's" : 'bachelors',
    "Bachelor's Degree" : 'bachelors',
    "Master's" : 'masters',
    "Master's Degree" : 'masters',
    "PhD" : 'phd',
    "phD" : 'phd'
})
```

Take numbers only for Age

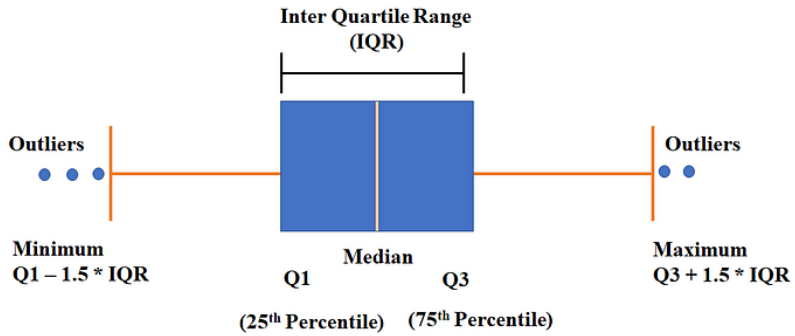
```
df["Age"] = df["Age"].str.extract("(\d+)").astype(int)
```

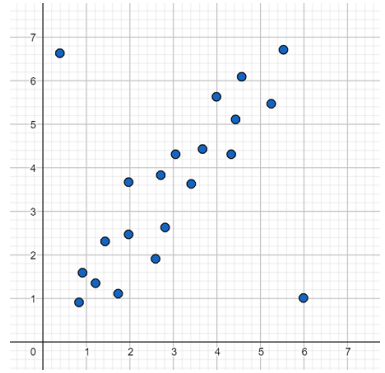
Data Cleaning Techniques

4. Outlier Detection and Treatment:

- Outliers are extreme values that deviate significantly from the rest of the data.

- Statistical Method for Detection
  - Z – Score Method
  - IQR Method
- Visualization Method
  - Boxplot
  - Scatterplot





Data Cleaning Techniques

4. Outlier Detection and Treatment:

- Remove
- Capping or Flooring – If outliers should be kept
- Imputation – If outliers indicate missing data points

Data Cleaning Techniques

5. Filtering Irrelevant Data:

- Remove data that does not contribute to the analysis objectives, helping to focus on relevant information.

6. Data Integration:

- *Combine data from different sources* while also resolving inconsistencies and ensuring that the merged dataset is coherent and accurate.

Data Cleaning Techniques

5. Data Transformation:

- Refers to the process of converting the format or structure of a data set to match that of a target system.
- Converting a raw data source into a ready-to-use format.

Analogy

Import

Clean

Transform

Data Cleaning Techniques

5. Data Transformation:

i. Format Conversion – Changing data types, file format conversion

ii. Aggregation or Splitting – Summarize data at a higher level (daily sales to monthly)

iii. Discretization – Convert continuous variables into discrete bins

iv. Feature Scaling

v. Feature Engineering – Create new features based on domain knowledge

vi. Encode Categorical Variables

vii. Simple Manipulations – Deriving new attributes, sorting, ordering and indexing data

Data Cleaning Techniques

5. Data Transformation:

- Aggregation or Splitting – Summarize data at a higher level (daily sales to monthly)

Raw data

Date	Order	Item	Total	State
1-Jan	1001	Shorts	$32.00	CA
1-Jan	1001	Hoodie	$34.00	CA
2-Jan	1002	T-shirt	$18.00	CA
3-Jan	1003	T-shirt	$18.00	CA
3-Jan	1004	Shorts	$32.00	OR
3-Jan	1004	Hoodie	$34.00	OR
3-Jan	1004	Hat	$16.00	OR
4-Jan	1005	Shorts	$32.00	ID
4-Jan	1005	Hoodie	$34.00	ID
5-Jan	1006	T-shirt	$18.00	CA
5-Jan	1006	T-shirt	$18.00	CA
7-Jan	1007	T-shirt	$18.00	CA
7-Jan	1007	Hat	$16.00	CA

Pivot table

Item	Count	Total
Hat	46	$736
Hoodie	59	$2,006
Sandals	36	$864
Shorts	25	$800
T-shirt	47	$846
Grand Total	**213**	**$5,252**

Data Cleaning Techniques

5. Data Transformation:

- Discretization – Convert continuous variables into discrete bins

Age range	Count
0–5	36
6–10	19
11–15	18
16–20	99
21–25	139
26–30	121

Age range	Count
31–35	76
36–40	74
41–45	54
46–50	50
51–55	26
56–60	22

Age range	Count
61–65	16
66–70	3
71–75	3

Data Cleaning Techniques

5. Data Transformation:

Feature Scaling

- Is a step that involves transforming the features of a dataset to similar scale.
- Is important because many machine learning models are sensitive to the scale of the input data.
- Prevents convergence and improves convergence.

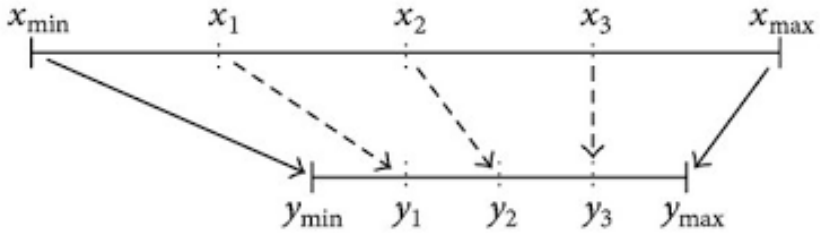
	Student	CGPA	Salary '000
0	1	3.0	60
1	2	3.0	40
2	3	4.0	40
3	4	4.5	50
4	5	4.2	52

	Student	CGPA	Salary '000
0	1	-1.184341	1.520013
1	2	-1.184341	-1.100699
2	3	0.416120	-1.100699
3	4	1.216350	0.209657
4	5	0.736212	0.471728

Data Cleaning Techniques

5. Data Transformation:

Feature Scaling



- **Normalization (Min – max scaling):** Adjust values to a common scale without distorting differences.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

- **Standardization:** Center and scale data (e.g., z-score normalization) to make it suitable for analysis, especially in machine learning.

$$Z = \frac{x - \mu}{\sigma}$$

Score, Mean, SD

Data Enrichment

- Data enrichment is a data management process that involves enhancing an existing data set by incorporating additional information.
- Data is enriched for several reasons, many of which revolve around improving its quality, depth, and usefulness.

	Symbol	Date	Open	High	Low	Close	Percent Change
0	ADBL	2024-07-11	288.1	291.8	286.0	291.0	1.57
1	ADBL	2024-07-10	283.0	288.0	282.0	286.5	0.46
2	ADBL	2024-07-09	286.1	287.5	278.0	285.2	1.68
3	ADBL	2024-07-08	282.0	286.4	279.4	280.5	-0.21
4	ADBL	2024-07-07	271.0	283.0	271.0	281.1	3.00

	Symbol	Date	Open	High	Low	Close	Percent Change	Volume
0	ADBL	2024-07-11	288.1	291.8	286.0	291.0	1.57	139159.0
1	ADBL	2024-07-10	283.0	288.0	282.0	286.5	0.46	42559.0
2	ADBL	2024-07-09	286.1	287.5	278.0	285.2	1.68	274360.0
3	ADBL	2024-07-08	282.0	286.4	279.4	280.5	-0.21	49865.0
4	ADBL	2024-07-07	271.0	283.0	271.0	281.1	3.00	106759.0

Data Enrichment

- Businesses carry out Data Enrichment to improve the information they currently have so they can make better-informed decisions
- **E.g.,** If you have a customer list with *just names and email addresses*, you could enrich it by *adding demographic information like age, location, and income level* from a third-party data provider.

Used for:

- Enhancing data quality
- Personalizing customer experiences
- Enhanced decision-making
- Market Research
- Customer Segmentation

Data Enrichment

Techniques

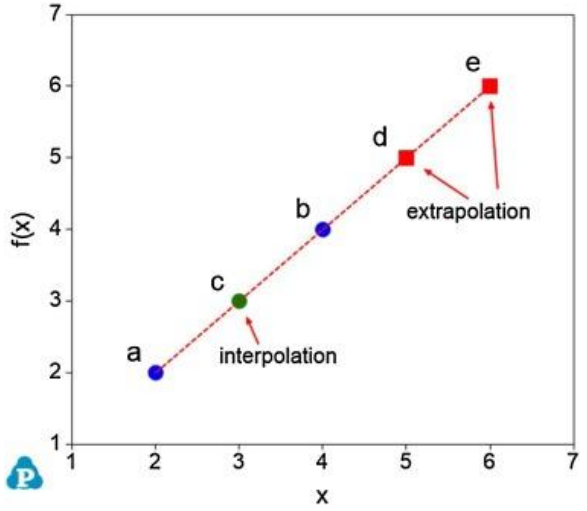
1. Data Segmentation: Dividing data into different groups is segmentation

- Examples of data segmentation:
  - Geographic: Categorizing data based on state, city, county, etc.
  - Demographic: based on attributes like age, gender, income, occupation, marital status, number of children, etc.
  - Behavioral: based on spending habits, browsing history, order habits, consumption habits, session frequency, time spent on a website, etc.
  - Technographic: on technological preferences like mobile devices, favorite browsers, software and other behavior.
  - Psychographic: attitudes, interests, values, personalities and archetypes are common psychographic segment criteria.

Data Enrichment

Techniques

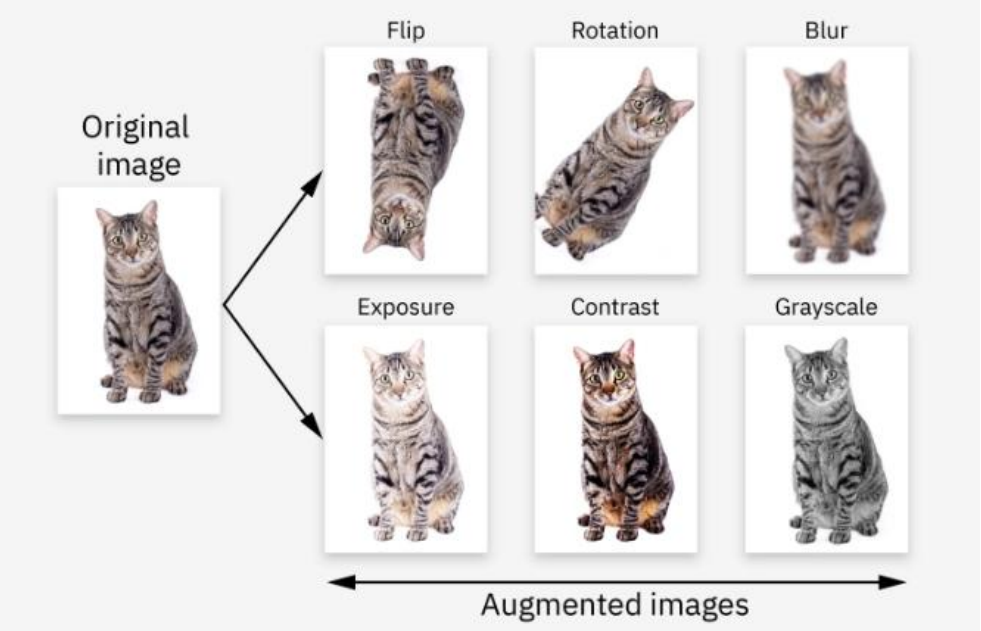
2. Data Imputation: Fixed values, Most frequent values, Nearest Neighbor, Next or previous
3. Appending Data:
4. Derived Attributes: Finding total sales by calculating of all sales made by a customer
5. Correction: Fixing errors, inconsistencies, or inaccuracies in the data.
6. Extrapolation: Estimating or predicting missing values based on existing data.



Data Enrichment

7. Data Augmentation

- is a set of techniques to artificially increase the amount of data by generating new data points from existing data.
- This is achieved by making small changes to data or using deep learning models to generate new data points.



https://research.aimultiple.com/data-augmentation/

Data Validation

- involves verifying the accuracy and consistency of your data against predefined rules, criteria, or external sources.
- process of determining whether a particular piece of information falls within the acceptable range of values for a given field.

i. **Type Check:** Ensures that the data entered is of the correct data type (e.g., integer, string, date).
   - **E.g.,** A field expecting a numeric value should not accept text.
   - A date field should only accept valid date formats.

Data Validation

ii. Range and Constraint Check

- Verifies that the data falls within a specified range or meets certain constraints.
- **Range Check:** An age field should only accept positive values in some specific range.
- **Constraint Check:** A email without '@' symbol is not a email.

iii. Consistency Check:

- Ensures that the data is logically consistent across different fields or datasets.
  - If a person's birth date is entered as 1990, their age should not be recorded as 50.
  - If a customer's shipping address is in the United States, the country field should not be set to Canada.

Data Formats

- Commonly used formats:

ID	Name	Age	Degree
1	John	18	B.Sc.
2	David	31	Ph.D.
3	Robert	51	Ph.D.
4	Rick	26	M.Sc.
5	Michael	19	B.Sc.

```
<University>
 <Student ID="1">
  <Name>John</Name>
  <Age>18</Age>
  <Degree>B.Sc.</Degree>
 </Student>
 <Student ID="2">
  <Name>David</Name>
  <Age>31</Age>
  <Degree>Ph.D. </Degree>
 </Student>
 ....
</University>
```

```
{
  "University" : {
    "Student" : [
      {
        "ID" : "1",
        "Name" : "John",
        "Age" : 18,
        "Degree" : "B.Sc."
      },
      {
        "ID" : "2",
        "Name" : "David",
        "Age" : 31,
        "Degree" : "Ph.D."
      }
    ]
  }
}
```

Data Formats

- Data is everywhere. Every click, every purchase, and every online interaction generates data.
- But how does all this information get organized, shared, and stored efficiently? This is where data formats come into play.

- Data formats determine how data is structured and made readable for systems and humans alike.
- How data is organized and presented?
- Same information can be organized in different ways.

Data Formats

Comma-Separated Values (CSV)

- stores tabular data with each line representing a row, and values separated by commas.
- This format is especially popular for importing and exporting data from spreadsheets and relational databases.
- CSV files are lightweight and easy to open in tools like Excel, making them accessible to technical and non-technical users alike.

```
ID,Name,Age,Degree
1,John,18,B.Sc.
2,David,31,Ph.D.
3,Robert,51,Ph.D.
4,Rick,26,M.Sc.
5,Michael,19,B.Sc.
```

- While efficient for straight forward datasets, CSV's flat structure becomes a limitation when dealing with nested or complex information.
- It also lacks built-in validation, which can result in inconsistencies if not managed carefully.

Data Formats

JSON (JavaScript Object Notation)

- was developed to provide a lightweight, readable way to store structured data. Unlike CSV, JSON supports nested structures, arrays, and multiple data types.
- Its key-value format makes it ideal for APIs, enabling seamless communication between servers and applications.

- However, JSON's versatility can lead to larger file sizes, which may impact performance in large-scale systems.
- JSON files are also more prone to inconsistencies if the data isn't properly validated, as there is no enforced schema.

Data Formats

XML (Extensible Markup Language)

- XML came before JSON and aimed to structure data in a machine- and human-readable way.
- **It uses custom tags** to organize information hierarchically, making it well-suited for complex datasets.

- XML is still used in many industries where data integrity and strict validation are required, such as healthcare and finance.
- Unlike JSON, XML supports schemas (XSD) to validate data, ensuring consistency across applications

Data Formats

Tabular

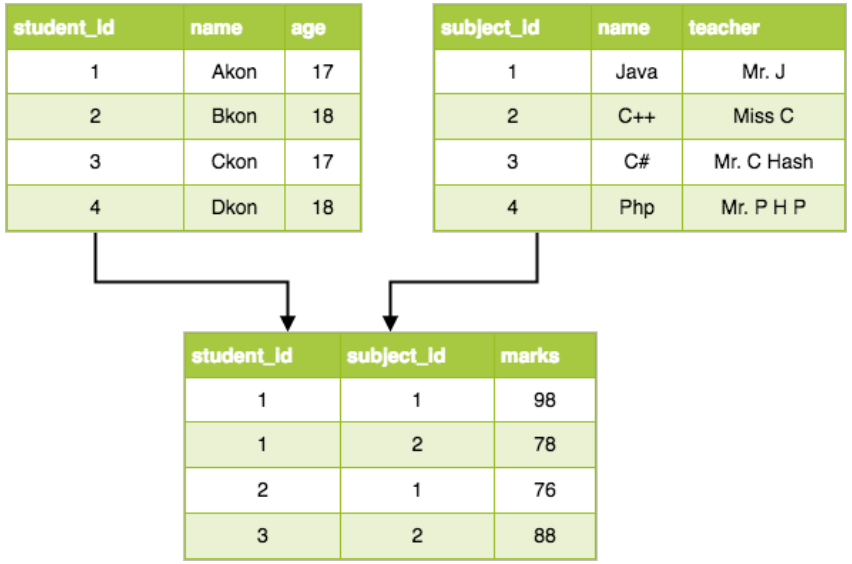
- is simply information presented in the form of a table with rows and columns.
- It represents the structured data format.
- A data table is a neat and convenient way to present a large body of information that includes repeating data elements.

- **For e.g.,** each entry in a list of company clients contains the **client's name, title, address, phone** number and other identifying information.
- This information can be listed in tabular format --that is, in rows and columns--by using separate columns for each data element.

Data Formats

Relational

- A relational model organizes data into one or more tables (or "relations") of columns and rows, with a unique key identifying each row.
- Rows are also called records or tuples.
- Columns are also called attributes.



student_id	name	age
1	Akon	17
2	Bkon	18
3	Ckon	17
4	Dkon	18

subject_id	name	teacher
1	Java	Mr. J
2	C++	Miss C
3	C#	Mr. C Hash
4	Php	Mr. P H P

student_id	subject_id	marks
1	1	98
1	2	78
2	1	76
3	2	88

- Generally, each table/relation represents one "entity type" (such as customer or product).
- A relationship is maintained between tables.
- For e.g., each row of a class table corresponds to a class, and a class corresponds to multiple students, so the relationship between the class table and the student table is "one to many".

Data Formats

Motivations behind format conversion

- Source data can come in many different data formats.

- To run analytics effectively, a data scientist must *first convert that source data to a common format* for each model to process

General Methods of Format Conversion

Data Formats

- While **JSON and CSV** files are still common for storing data, they were never designed for the massive scale of big data and tend to eat up resources unnecessarily (JSON files with nested data can be very CPU-intensive, for example).
- They are in text format and therefore human readable. But they lack the efficiencies offered by binary options.

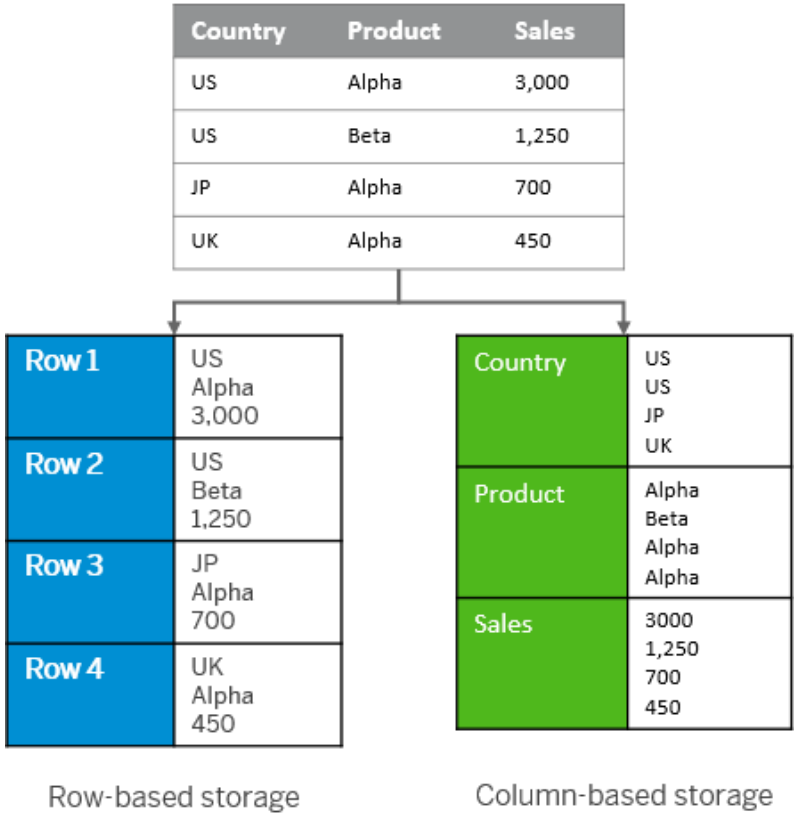
- So as data has grown, file formats have evolved.
- File format impacts speed and performance, and can be a key factor in determining whether you must wait an hour for an answer – or milliseconds.
- Matching your file format to your needs is crucial for minimizing the time it takes to find the relevant data

Tabular Data

Row-based vs. Column-based

- There are two main ways in which tabular data can be organized your data: rows and columns.

- **Row-Based :** Data is stored by rows (records). Each row contains all columns for that record.
- **Column-Based :** Data is stored by columns. Each column contains all values for that column across different records.



Tabular Data

Row-based:

Pros

- Modifying data is easier
- Ideal for OLTP

Cons

- Slower data aggregation
- Insufficient compression
- Requires additional space

https://www.fivetran.com/learn/columnar-database-vs-row-database

Tabular Data

Row-based

1	Michael	Jones	Dallas	32	2	Preston	James	Boston	25

1	Michael	Jones	Dallas	32	2	Preston	James	Boston	25	3	Amy	Clarke	Denver	37

Column-based

1	2	Michael	Preston	Jones	James	Dallas	Boston	32	25

1	2	3	Michael	Preston	Amy	Jones	James	Carke	Dallas	Boston	Denver	32	25	37

Tabular Data

Row-based

Disk 1				
ID	First Name	Last Name	City	Age
1	Michael	Jones	Dallas	32
Disk 2				
ID	First Name	Last Name	City	Age
2	Preston	James	Boston	25
Disk 3				
ID	First Name	Last Name	City	Age
3	Amy	Clarke	Denver	37

Column-based

Disk 1		
ID		
1	2	3

Disk 2		
First Name		
Michael	Preston	Amy

Disk 3		
Last Name		
Jones	James	Clarke

Disk 4		
City		
Dallas	Boston	Denver

Disk 5		
Age		
32	25	37

Tabular Data

Column-based

- A columnar database stores all the data from each column as a single block.
- Think of it as vertical partitioning compared to the horizontal partitioning of a row store.

Pros

- Best for OLAP applications
- Faster data aggregation
- High compression speeds
- Requires less space

Cons

- Data modification is slower

Tabular Data

- Parquet
- ORC

https://www.upsolver.com/blog/the-file-format-fundamentals-of-big-data

Tabular Data

Wide vs. Narrow(Long) Table Formats

Wide Format

Team	Points	Assists	Rebounds
A	88	12	22
B	91	17	28
C	99	24	30
D	94	28	31

Each value is unique in first column

Long Format

Team	Variable	Value
A	Points	88
A	Assists	12
A	Rebounds	22
B	Points	91
B	Assists	17
B	Rebounds	28
C	Points	99
C	Assists	24
C	Rebounds	30
D	Points	94
D	Assists	28
D	Rebounds	31

The values in the first column repeat

Tabular Data

Wide Table Formats

- Each row contains many columns.
- Suitable for denormalized data where all related information is stored in a single row.

- Good for small datasets or when all columns are needed.
- Can lead to redundancy and increased storage size.
- Less flexible for adding new attributes.

Name	Height	Weight
John	160	67
Christopher	182	78

Name	Attribute	Value
John	Height	160
John	Weight	67
Christopher	Height	182
Christopher	Weight	78

Tabular Data

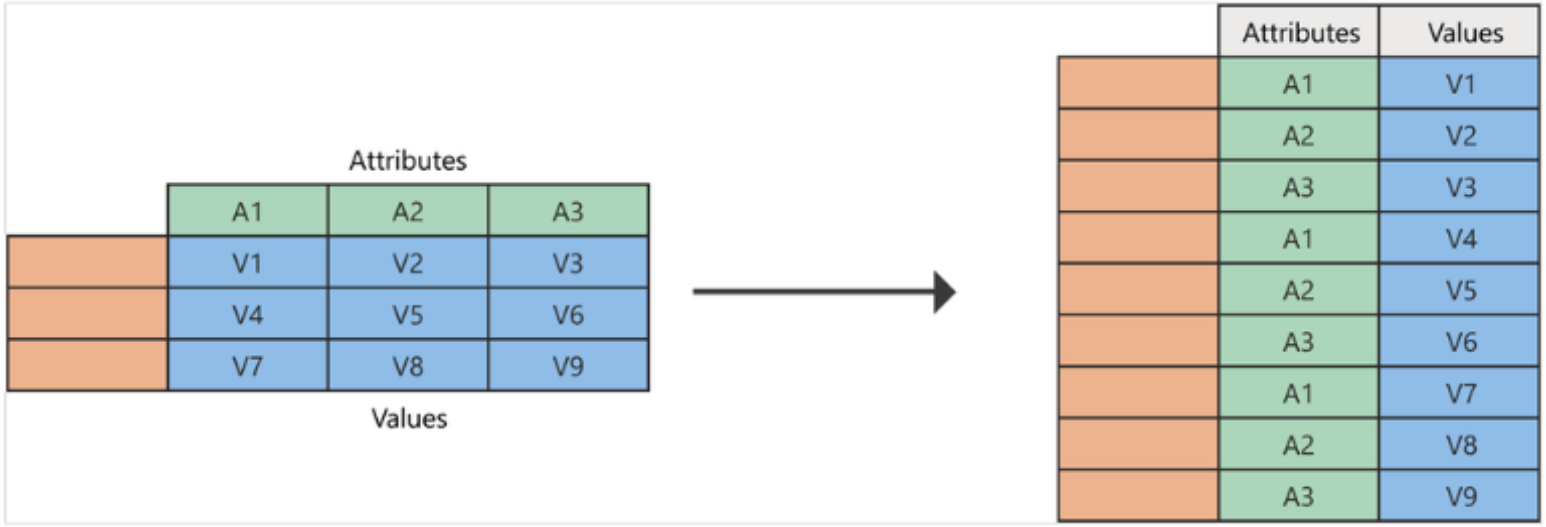
Narrow (Long) Format

- Each row contains fewer columns, often with a key-value or attribute-value structure.
- Suitable for normalized data where attributes are stored in separate rows.

- Reduces redundancy and storage size.
- Requires more complex queries for analysis.

Tabular Data

- Wide to Narrow: Melting or unpivoting
- Narrow to wide: Pivoting



End of Unit 2

Thank you