M.Tech Program

Advanced Industry Integrated Programs

Jointly offered by University and LTIMindTree

Data Engineering

Knowledge partner

LTIMindtree

Implementation partner

L&T EduTech

Course Objective

- Recognize data types and structures.
- Grasp big data fundamentals and analytics.
- Master data ingestion processes and tools.
- Understand exploratory data analysis techniques.
- Learn storage methods and data flow.

Modules

- Data Types & Formats
- Data Ingestion techniques
- Data Profiling & Visual Representation via various tools (Pandas)
- Storage and retrieval methods
- Data Lineage Analysis

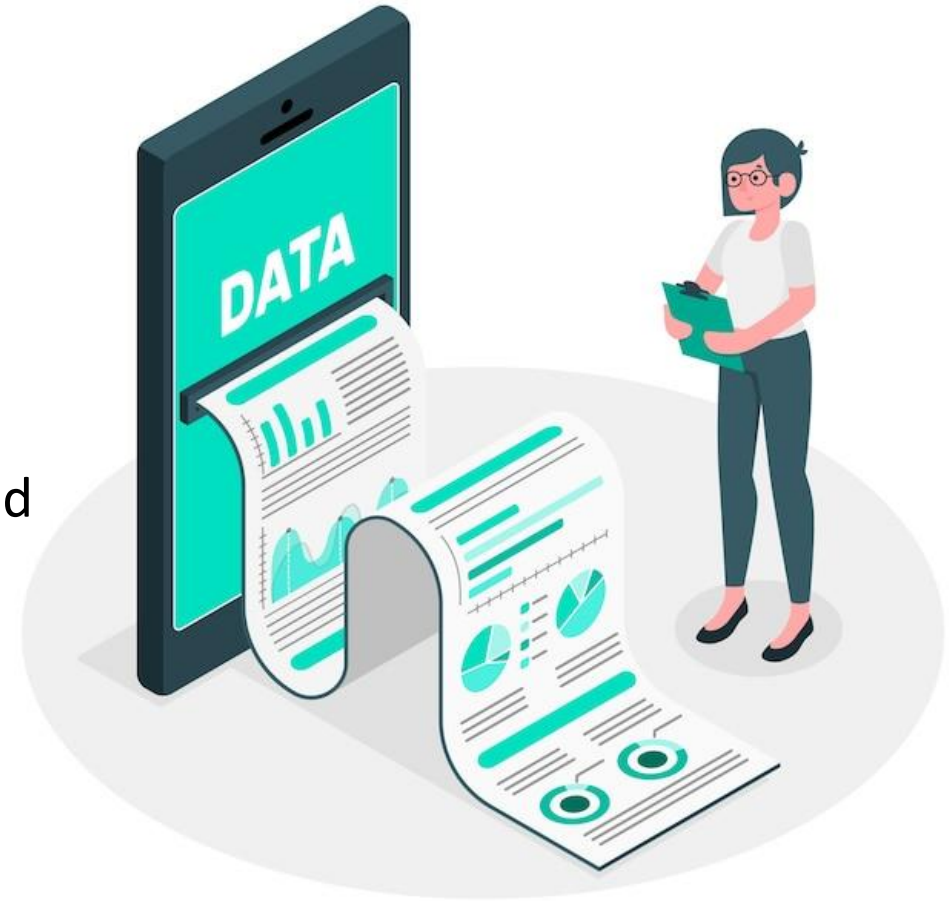
Data Types & Formats

Data Fundamentals

Data Fundamentals

Data

- **Raw Information**: Data represents raw facts, figures, and statistics collected through observation, measurement, or research.
- **Structured or Unstructured:** Data can be structured, organized in a specific format like databases or spreadsheets, making it easily searchable and analyzable.
- **Foundation of Insights:** Data serves as the foundation for generating insights and making informed decisions.



Data Fundamentals

Big Data

- **Large Volume**: Massive amounts of data collected.
- **Varied Types**: Different formats and structures present.
- **High Velocity**: Data arrives rapidly in real-time.
- **Complexity and Diversity**: Includes structured, unstructured, and semi-structured.
- **Challenges and Opportunities**: Requires advanced analytics for insights.

Data Fundamentals

Key Characteristics of Big Data

- **Volume:** Involves large amounts of data.
- **Velocity**: Data streams in rapidly, often in real-time.
- **Variety:** Includes diverse types of data formats.
- **Veracity:** Data quality and reliability may vary.
- **Value:** Extracting insights for decision-making.

Data Fundamentals

Types of Big Data

Structured Data

- Organized data with a defined format, often stored in relational databases.
- Examples include tables in SQL databases or spreadsheets.

Unstructured Data

- Data without a predefined format or organization
- Examples include text documents, social media posts, videos, and images

Semi-Structured Data

- Data that does not fit neatly into either structured or unstructured categories.
- Examples include XML or JSON files, emails, and log files.

Quasi Structured Data

- Textual data with erratic data formats, can be formatted with effort, tools, and time
- Example: We

Data Fundamentals

Structured vs Un-Structured Data

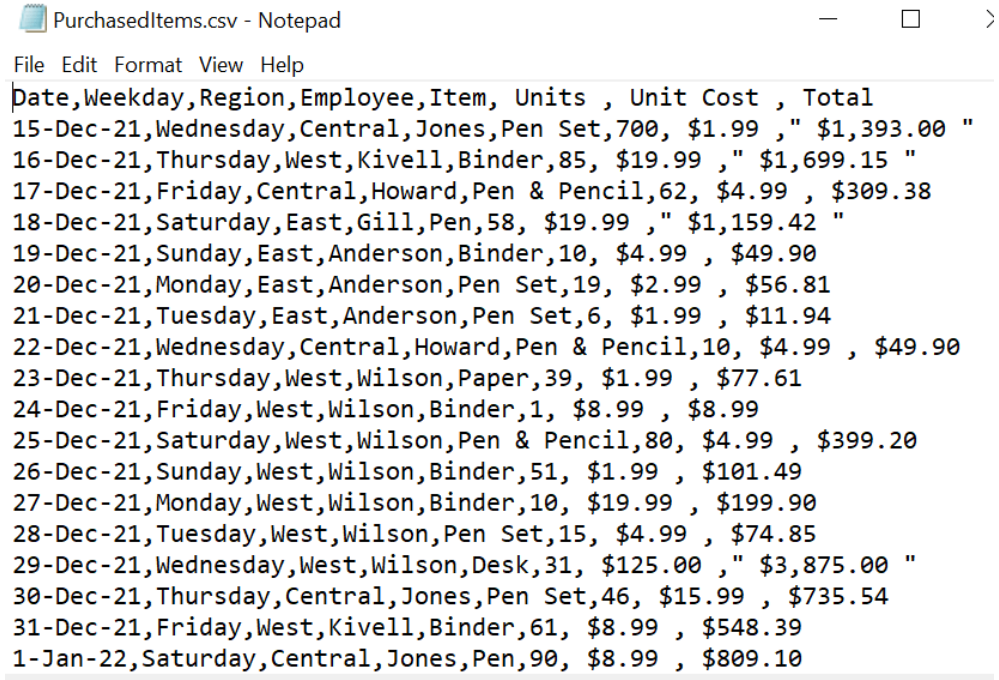
	Structured Data	Unstructured Data
Definition	Data That Fits Into a Table or Other Fixed Field	Data That Does Not Fit Into a Fixed Field or Consistent Structure
Examples	Quantitative Data, Categorized Data	Images, Audio files, Natural language/text, large text files Social Media Content, Digital Behavior Data
Storage	Relational Database Management System (RDBMS), Data Warehouse	NoSQL databases, Data lakes
Analysis	Conventional Methods and Tools, Excel, Google Sheets, Artificial Intelligence	Specialized tools, Natural language, Processing (NLP)Text Mining, Some manual analysis
Users	Business Professionals, Data Analysts	Data Scientists, Data Engineers

Data Representation

Data Representation

Formats of Data

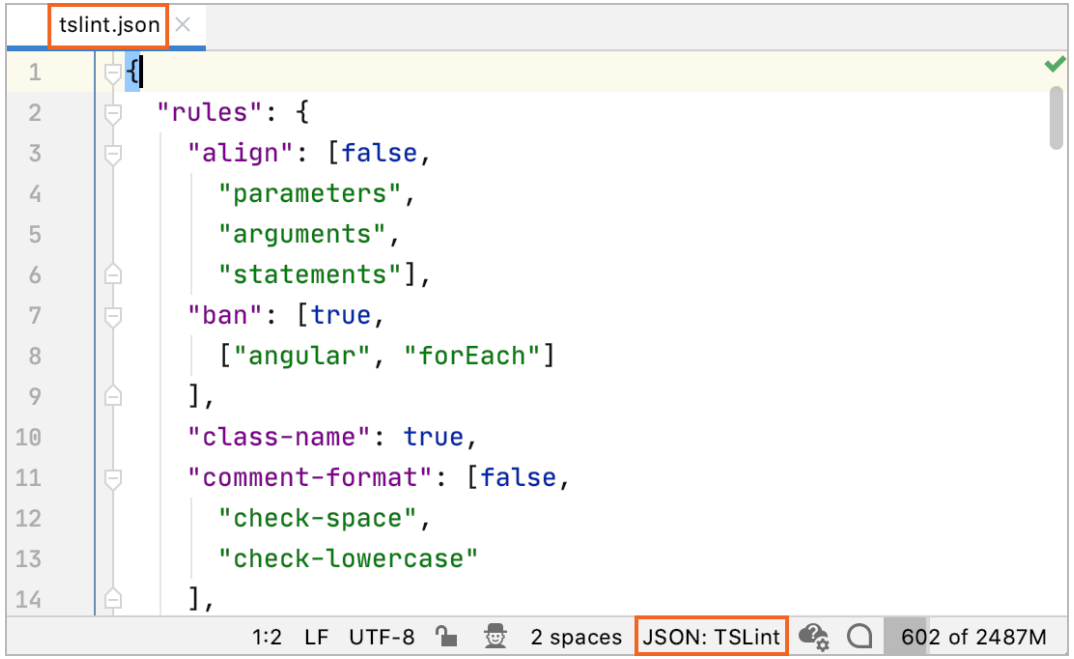
- **CSV (Comma-Separated Values)**
- **Description:** CSV files store tabular data in plain text format, with each line representing a row and fields separated by commas.
- **Attributes:**
- Simple and widely used.
- Lacks support for data types.
- Commonly used for exchanging tabular data between different systems or applications.



Data Representation

Formats of Data

- **JSON (JavaScript Object Notation)**
- **Description:** JSON is a lightweight data interchange format that is human-readable and easy for both humans and machines to understand.
- **Attributes:**
- Supports nested structures.
- Widely used in web development, APIs, and configurations.
- Useful for semi-structured data.



Data Representation

Formats of Data

- **Parquet**
- **Description:** Parquet is a columnar storage file format optimized for big data processing frameworks like Apache Hadoop and Apache Spark**.**
- **Attributes:**
- Stores data in a columnar fashion, improving query performance.
- Efficient column-wise compression.
- Well-suited for analytics workloads on large datasets.

Data Representation

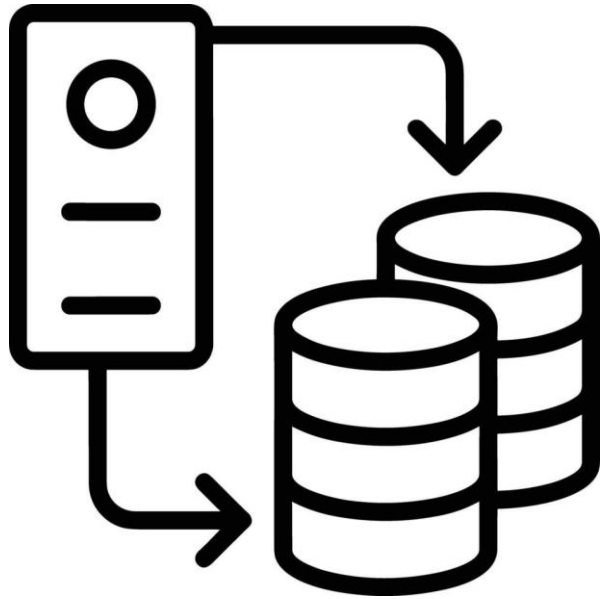
Semi-Structured

- Semi-structured data is a type of data that doesn't conform to the *structure of traditional relational databases.*
- It *lacks the rigid structure of tabular* data but may have tags, keys, or other markers that provide some structure.
- Unlike structured data, which fits neatly into tables with predefined schemas, *semi-structured data allows for more flexibility.*
- Examples: Semi-structured data examples include JSON (JavaScript Object Notation), XML (eXtensible Markup Language), and log files.

Data Representation

What is Data Type Conversion?

- **Definition:** Converting data from one type to another.
- **Types:**
  - **Explicit:** Manual conversion using functions.
  - **Implicit:** Automatic conversion by the programming language.
  - **Examples**: Converting integers to strings, strings to floats, lists to dictionaries.



Data Representation

Importance of Data Type Conversion

- **Interoperability:** Ensures data is usable across different platforms and systems.
- **Data Accuracy:** Maintains integrity and precision in calculations and analysis.
- **Storage and Memory Management:** Optimizes memory usage and application performance.
- **Error Prevention:** Avoids errors from incompatible data types.

Data Representation

Role in Data Management and Integration

- **Data Integration:** Standardizes data formats from multiple sources.
- **Data Cleaning and Preprocessing:** Ensures data is in the correct format for analysis.
- **Database Management:** Maintains consistency when importing/exporting data.
- **Data Interchange Formats:** Correctly interprets and manipulates data from formats like JSON, XML, CSV.

Data Representation

Implicit vs. Explicit Type Conversion

- **Implicit (Automatic) Type Conversion:**
- **Definition:** Conversion that is automatically performed by the compiler or interpreter without explicit instructions from the programmer.
- **Characteristics:**
  - Automatic: The language handles the conversion process.
  - Seamless: Often occurs without the programmer's awareness.
  - Potential for Errors: Can lead to unexpected behavior if not carefully managed.

Data Representation

Implicit vs. Explicit Type Conversion

- **Explicit (Manual) Type Conversion:**
- **Definition:** Conversion that is explicitly defined by the programmer using functions or casting mechanisms.
- **Characteristics:**
- **Manual:** Requires specific instructions from the programmer.
- **Controlled:** The programmer determines when and how the conversion occurs.
- **Less Error-Prone:** Reduces the likelihood of unexpected behavior by making conversions intentional.

Data Representation

Promotion and Demotion in Data Type Conversion

- **Promotion:**
- **Definition:** Also known as type widening, promotion refers to converting a value from a smaller data type to a larger data type. This process generally involves converting a type with a smaller range or less precision to one with a larger range or more precision.
- **Purpose:** To prevent data loss and maintain precision during operations.

Data Representation

Promotion and Demotion in Data Type Conversion

- **Demotion:**
- **Definition:** Also known as type narrowing, demotion refers to converting a value from a larger data type to a smaller data type. This process often involves a loss of precision or range and needs to be done explicitly to avoid data loss.

- **Purpose:** To fit a larger data type into a smaller domain, typically for memory efficiency or specific application requirements.

Data Representation

Best Practices and Considerations for Efficient Data Type Conversion

Best Practices:

- **Understand the Data:**
  - Know the source and destination data types.
  - Understand the data's range and precision requirements.
- **Use Built-in Functions:**
  - Leverage language-specific functions and libraries designed for type conversion.
  - Ensure they are optimized for performance and accuracy.

Data Representation

Best Practices and Considerations for Efficient Data Type Conversion

Best Practices:

- **Validate Data:**
  - Before conversion, validate data to ensure it meets the expected format and value ranges.
  - Use exception handling to manage conversion errors gracefully.
- **Prefer Explicit Conversion:**
  - Use explicit conversion to avoid ambiguity and ensure clarity in code.
  - Document the rationale for conversions to maintain code readability and maintainability.

Data Representation

Best Practices and Considerations for Efficient Data Type Conversion

Best Practices:

- **Minimize Demotion:**
  - Avoid unnecessary demotion to prevent data loss.
  - Where demotion is necessary, carefully handle potential loss of precision or range.
- **Consistent Data Handling:**
  - Maintain consistency in data types across your application to minimize the need for frequent conversions.
  - Standardize data formats and types within your systems and databases.

Data Representation

Best Practices and Considerations for Efficient Data Type Conversion

Best Practices:

- **Optimize for Performance:**
  - Profile and optimize conversion-heavy sections of your code.
  - Use efficient data structures and algorithms to manage large-scale conversions.
- **Test Thoroughly:**
  - Implement comprehensive testing for all type conversion logic.
  - Include edge cases, such as maximum and minimum values, and invalid inputs.

Data Representation

Data Transformation

- Data transformation is the process of converting data from one format, standard, or structure to another.
- It does not change the content but prepares data for app or user consumption and improves data quality.

Data Representation

Key Points : Data Transformation

- **Modification:**
  - Changes the format, organization, or values of data.
- **Timing in Data Pipeline:**
  - Applied during data analytics projects.
  - In on-premises data warehouses: Part of ETL (extract, transform, load).
  - In cloud-based data warehouses: Part of ELT (extract, load, transform)..

Data Representation

Key Points : Data Transformation

- **Use Cases:**
  - Data migration
  - Data warehousing
  - Data integration
  - Data wrangling
- **Importance for Organizations:**
  - Provides timely business insights.
  - Ensures data is accessible, consistent, and safe.
  - Makes data usable for targeted business users.

Data Representation

Key Points : Data Transformation

- **Methodologies:**
  - ETL: Transform data before loading into the warehouse.
  - ELT: Load raw data first, then transform upon query.
- **Benefits:**
  - Handles increased data volumes efficiently.
  - Expands computational and storage capacity quickly in cloud environments.
  - Ensures data quality and usability..

Data Representation

Types of Data Transformation

- Data Transformation through Scripting
- On-Premises ETL Tools
- Cloud-Based ETL Tools
- Constructive and Destructive Data Transformation
- Structural Data Transformation
- Aesthetic Data Transformation

Data Representation

Data Serialization

- **Definition:** Data serialization converts data objects into a format that can be easily stored or transmitted.
- **Purpose:**
  - Facilitates data exchange between different systems and applications.
  - Supports data persistence (long-term storage) and compression.
- **Function:**
  - Translates complex data structures into universally understandable formats.
- **Benefits:**
  - Ensures seamless integration and communication between heterogeneous platforms.
  - Bridges the gap between disparate systems.

Data Representation

Common Data Serialization Formats

XML (eXtensible Markup Language)	• A flexible and structured format widely used for web services and configurations.
JSON (JavaScript Object Notation)	• A lightweight, easy-to-parse format popular in web applications.
BSON (Binary JSON)	• An optimized binary format extending JSON's capabilities, ideal for storage and network transfers.
YAML (YAML Ain't Markup Language)	• A human-readable format commonly used for configuration files.
MessagePack	• A compact binary format efficient for both storage and network communication.
Protobuf (Protocol Buffers)	• Developed by Google, known for its efficiency and scalability

Data Representation

Key Applications of Data Serialization

Storing Data

- Serialization enables data to be saved in files or databases in a format understandable by different systems.
- Facilitates long-term storage and retrieval processes.

Transferring Data

- It allows efficient transmission of data over the network or between processes.
- Ensures reliable communication across different systems and machines.

Distributing Objects

- Serialization is crucial in component-based software engineering.
- It ensures seamless interaction and data exchange between components.

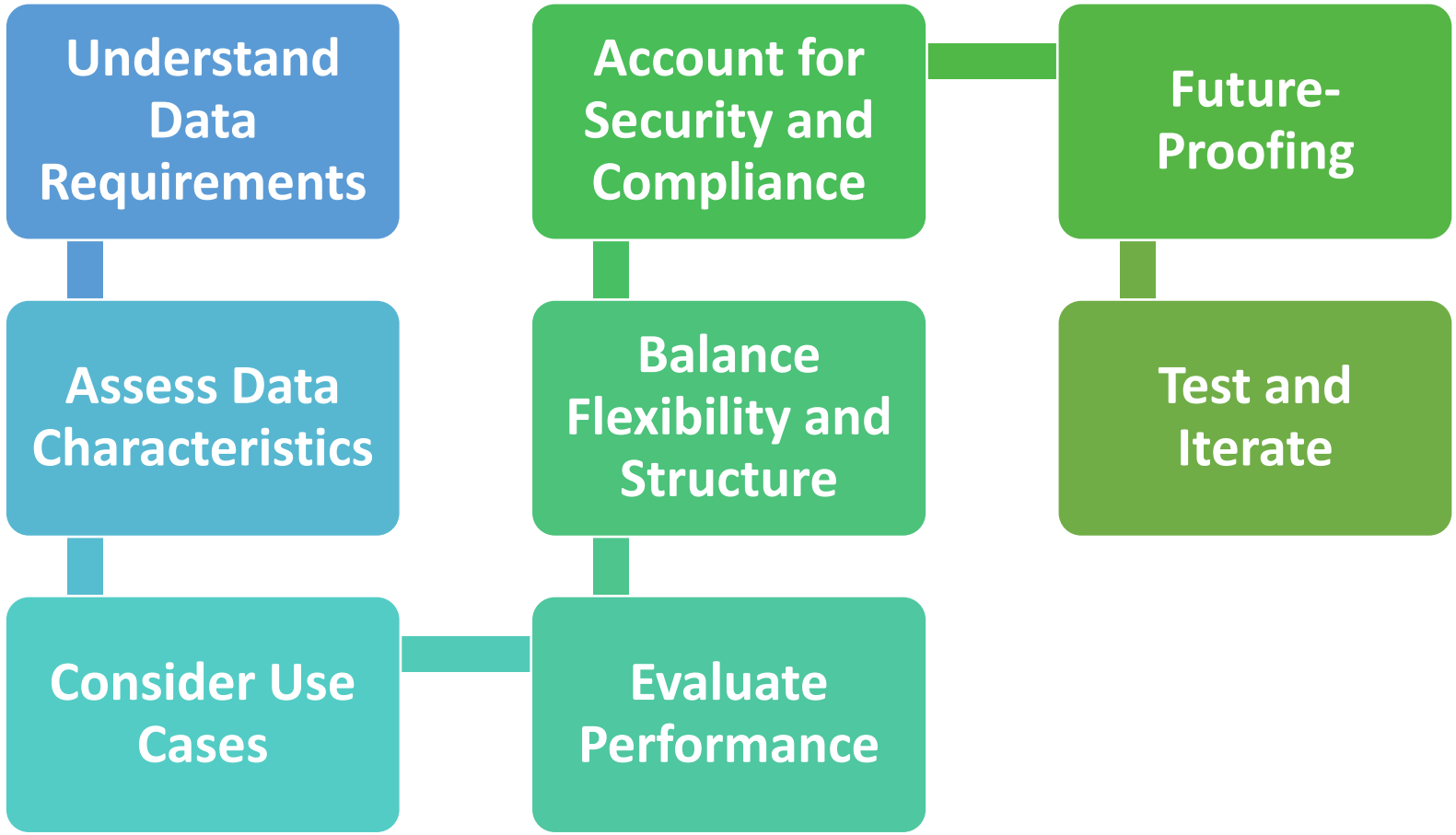
Change Detection

- By serializing data at different intervals, changes can be detected and monitored.
- Supports functionalities like versioning and audit trails.

Data Management

Data Management

Choosing the Right Data Type and Format



Data Management

Tools and Technologies for Data Types and Formats

Programming Languages

- **Python**: Widely used for data manipulation and transformation with libraries like Pandas for handling different data types.
- **Java**: Popular for enterprise-level applications and offers extensive support for various data formats.

Data Serialization Libraries

- **JSON (JavaScript Object Notation)**: Supported by most programming languages and frameworks for data interchange.
- **XML (eXtensible Markup Language)**: Used for structured data representation, with libraries like lxml and xml.etree.ElementTree in Python.
- **MessagePack**: Compact binary format with implementations available in multiple languages.

Data Management

Tools and Technologies for Data Types and Formats

Database Management Systems (DBMS)

- **SQL Databases (e.g., MySQL, PostgreSQL)**: Support structured data storage and retrieval with built-in data types.
- **NoSQL Databases (e.g., MongoDB, Cassandra)**: Handle semi-structured and unstructured data with flexible schema designs.

ETL (Extract, Transform, Load) Tools

- **Apache NiFi**: Provides powerful data routing, transformation, and system mediation capabilities.
- **Talend**: Offers comprehensive data integration and transformation solutions with a user-friendly interface.
- **Informatica PowerCenter**: Enterprise-grade ETL tool for data integration and management.

Data Management

Tools and Technologies for Data Types and Formats

Data Serialization Formats Conversion Tools

- **JSON to XML Converters**: Tools for converting JSON data to XML format and vice versa.
- **Protobuf Compiler**: Generates code for serializing and deserializing data in Protocol Buffers format.

Cloud Platforms

- **AWS (Amazon Web Services)**: Provides various services for data storage, processing, and serialization like Amazon S3 and AWS Glue.
- **Google Cloud Platform**: Offers services such as BigQuery and Dataflow for data processing and serialization.

Data Management

Tools and Technologies for Data Types and Formats

Frameworks and Libraries

- **Apache Avro**: Provides a compact, fast, binary serialization format with support for schema evolution.
- **Apache Parquet**: Columnar storage format optimized for efficient data storage and processing in big data environments.

Data Quality and Governance Tools

- **Collibra**: Offers data governance solutions for ensuring data quality, compliance, and security.
- **Informatica Data Quality**: Provides tools for profiling, cleansing, and enriching data to maintain quality standards.

Exercise

Data Types & Formats

Exercise

- In this exercise, you will work with various data formats commonly used in data engineering: CSV, JSON, AVRO, Parquet, XML, and HDF5. You will be provided with sample data representing employees' names, ages, and occupations. Your tasks will include reading data from these formats, processing the data, and performing simple analytical operations.
- Employee Data - Sample data is as follows :

Name	Age	Occupation
John Doe	28	Software Engineer
Jane Smith	34	Data Scientist
Emily Davis	41	Product Manager
Michael Brown	37	UX Designer

Data Types & Formats

Exercise

Tasks :

1. Data Preparation:

- Create files in the following formats containing the sample employee data:
  - CSV
  - JSON
  - AVRO
  - Parquet
  - XML
  - HDF5

Data Types & Formats

Exercise

Tasks :

2. Python Script for Reading and Printing Data:

- Write a Python script to read the data from each file format.
- Print the data in a tabular format.

3. Extracting Names and Ages:

- Modify the script to extract the names and ages of the employees from each file.
- Store the names in one list and the ages in another list.
- Print both lists.

4. Calculating Average Age:

- Modify the script to calculate the average age of the employees.
- Print the average age.

Data Types & Formats

Exercise

Instructions:

1. **CSV File:** Use libraries such as csv or pandas to handle CSV files.
2. **JSON File:** Use the json library or pandas for reading JSON files.
3. **AVRO File:** Use the fast Avro or avro-python3 library to work with AVRO files.
4. **Parquet File:** Use the pyarrow or pandas library to read Parquet files.
5. **XML File:** Use the xml.etree.ElementTree or pandas library to parse XML files.
6. **HDF5 File:** Use the h5py or pandas library to read HDF5 files.

Summary

Data Types & Formats

Summary

- **Structured Data:** Defined models, rows, columns, relational databases.
- **Unstructured Data:** No format, hard to analyze, text, videos.
- **Semi-Structured Data:** Mix, tags, XML, JSON, flexible.
- **Choosing Data Format:** Crucial for storage, processing, analysis.
- **Semi-Structured Flexibility:** JSON, XML, NoSQL, diverse data.

Knowledge Check

Data Types & Formats

Q1 : What type of data does social media posts, emails, and multimedia content represent?

a) Structured Data

b) Semi-Structured Data

c) Unstructured Data

d) Relational Data

Data Types & Formats

Q2 : Which of the following is an example of structured data?

a) JSON

b) XML

c) Text document

d) Relational Data

Data Types & Formats

Q3 : What does data transformation involve?

a) Storing data in its original format

b) Converting data from one type to another

c) Creating semi-structured data

d) Extracting data from structured sources

Data Types & Formats

Q4 : Which of the following is a tool for data transformation?

a) Hadoop

b) Spark

c) ETL (Extract, Transform, Load)

d) MongoDB

Data Types & Formats

Q5 : What is the purpose of data serialization?

a) Storing data in a database

b) Converting data into a binary format

c) Transforming semi-structured data

d) Processing unstructured data

Data Types & Formats

Q6 : Which of the following is an advantage of choosing the right data type and format?

a) Increased data complexity

b) Improved data quality

c) Reduced data processing speed

d) Enhanced data security

?

Thank You !!!