Design and Build Data Processing Application

Mrs. ALF. Sajeetha

APDP

Unstructured data

I work with a wide variety of data kinds in different formats as an owner and data analysts of an IT company. I have to sort them and examine the information I've collected. There are two categories of data sources: semi-structured and unstructured. Unstructured data includes files like audio, video, and picture files. In today's expanding digital world, unstructured data makes up 90% of all data. Relational databases are not appropriate for storing this data, so scenario created a database using NoSQL to house it. Currently, NoSQL databases come in four varieties: document-oriented, key-value, column-oriented, and graph-oriented. These days, the majority of well-known companies—including Google, YouTube, Amazon, LinkedIn, Facebook, and LinkedIn—deal with NoSQL statistics and have transitioned from conventional database structures to NoSQL databases. Unstructured data isn't predicated on schemas. Relational databases cannot handle it, but 94% of data is currently unstructured and growing. It includes Word and PDF documents, digital media files, and NoSQL database-stored documents.

Semi Structure data

Semi-structured data comprises of files like emails, XML, and JSON. Relational databases, which express data using edges, designations, and tree structures, are not appropriate places for semi-structured data. They have properties and labels and are represented as trees and graphs. These are unstructured data. Graph-based data structures can be used to store semi-structured data. A NOSQL database that supports JSON (semi-structured data) is called MongoDB. Self-descriptive tags constitute semi-structured data. They differentiate between data that is organized and data that is not. Objects Exchange Model, Data Guide, and Data Object Model are popular models for representing semi-structured data. Concepts of semi-structured data models include write down instance, document schema, variables details, and elements relationship sets. (Google.com, 2019) Data that is semi-structured is not schema-based. It is created from multiple websites and is labeled and has edges. It has a number of traits. I can therefore appropriately arrange such data at my company

Python Libraries to build Data Processing Application

- JSON
- Pandas, Numpy
- Matplotlib
- Web browsers, Html, Boostrap framework

Design & Implementation of Data Processing Application

1. Utilize an appropriate language and development tools, incorporate security and maintainability expectations.
2. Produce program code that implements a design based on SOLID principles, clean coding techniques and programming patterns.
3. Understand and interpret design features, meet requirements, input, output, processing security, portability and maintainability

What is Data Processing Application

A data processing application, also known as a data app, is a software application that processes and analyzes large volumes of data to rapidly deliver insights or take autonomous action. These applications are designed to handle specific types of data processing tasks

Ex: A timesharing system is optimized to run timesharing processing

Data processing applications utilize techniques such as data science, machine learning, artificial intelligence, automation, and other advanced data techniques to transform raw data into usable information. This transformation often involves a series of steps including collection, filtering, sorting, processing, analyzing, storing, and then presenting the data in a readable format

Types of data processing methods that Data Processing Application implements

1. **Transaction Processing**: Real-time handling of individual operations such as data entry and retrieval. This is commonly used in applications like banking and online transactions.
2. **Distributed Processing**: Distribution of data processing tasks across multiple interconnected computers or servers for parallel processing. This enhances efficiency in large-scale systems and big data applications.
3. **Real-time Processing**: Immediate processing of data as it is generated or received. This requires low latency and quick response times and is used in applications like monitoring systems and financial trading.
4. **Batch Processing**: Execution of a series of data processing tasks in a batch or group, collected over time and processed in large volumes, typically used for non-real-time tasks like data backups and report generation.
5. **Multiprocessing**: Utilizing multiple processors or computing units to execute tasks concurrently, dividing tasks into smaller subtasks for simultaneous processing, used to improve performance in high-performance computing and parallel computing applications.

What is Data Processing Application

A data processing application is a software program designed to handle the complete lifecycle of data, from its raw state to a usable and informative format.

1. Takes in Raw Data: Data processing applications can acquire data from various sources like files (CSV, Excel), databases, web APIs, or even user input.

2. Transforms and Cleans Data: Raw data is often messy and unorganized. The application may clean the data by handling missing values, correcting inconsistencies, and converting formats to ensure consistency.

3. Processes the Data: This is the heart of the application. Here, the data is manipulated according to specific needs.

- Sorting and filtering data to focus on relevant subsets.
- Performing calculations and aggregations (e.g., finding averages, totals).
- Merging or joining data from multiple sources.

4. Outputs the Results: The processed data is presented in a user-friendly format. This could be:

- Writing data to files (CSV, Excel) for further analysis.
- Loading the data into a database for storage and retrieval.
- Generating reports or visualizations (charts, graphs) to make trends and patterns easier to understand.

Examples of Data Processing Applications

1. **Sales data analysis** to understand customer behavior and buying trends.
2. **Financial data processing** for accounting, budgeting, and forecasting.
3. **Scientific data analysis** to interpret research results and make discoveries.
4. **Log data processing** to monitor system health and identify potential issues.

Benefits of Data Processing Applications

- **Improved Decision Making:** By transforming raw data into actionable insights, these applications empower users to make data-driven decisions.
- **Enhanced Efficiency:** Automating data processing tasks saves time and reduces manual errors.
- **Better Insights:** Data processing helps identify patterns, trends, and relationships within data that might not be evident otherwise.

Developing Main Modules in Data Processing Application

- **The specific modules you'll build for a data processing application will depend on the application's functionality and complexity.**
- **Some common modules found in many data processing applications include,**

1. Data Input Module

- Handles fetching data from various sources like files (CSV, Excel), databases, APIs, or user input.
- May include functionalities for data validation and transformation
    1. converting data formats and data cleansing
    2. handling missing values

2. Data Processing Module

- This is the core module where the actual data manipulation happens.
- Depending on the application, this could involve tasks like:
  - Sorting and filtering data
  - Performing calculations and aggregations
  - Joining datasets
  - Data normalization

3. Data Output Module

- Handles presenting the processed data in a desired format.
- This could involve:
  - Writing data to files
    - Text Report (Excel)
    - CSV Report
    - Row data Report
    - HTML Web Report
  - Loading data to a database
  - Generating reports or visualizations

4. Error Handling Module

- Catches and logs errors that occur during data processing.
- May implement mechanisms for data recovery or retries in case of failures.

5. User Interface Module

- Provides a user interface for interacting with the data processing application.
- This could be a simple command-line interface or a more complex graphical user interface (GUI) depending on the application's target audience.

Additional Modules depending on Complexity of the Data Processing Application

6. Scheduling Module: Automates data processing tasks to run at specific times or intervals.

7. Security Module: Implements security measures to protect sensitive data during processing.

8. Logging Module: Tracks the application's activity and data processing steps for auditing purposes.