Data Extraction in ETL

ETL Assignment Solutions

Question 1: Describe different types of data sources used in ETL with suitable examples.

In ETL (Extract, Transform, Load), data sources are the origins of the information being processed. Common types include:

- **Relational Databases:** Structured data stored in tables (e.g., MySQL, PostgreSQL, or Oracle).
- **Flat Files:** Plain text files containing data, such as **CSV** or **JSON** files.
- **Cloud Applications/SaaS:** Data from platforms like Salesforce or Google Analytics, often accessed via **APIs**.
- **NoSQL Databases:** Unstructured or semi-structured data sources like MongoDB.

Question 2: What is data extraction? Explain its role in the ETL pipeline.

Data extraction is the first step in the ETL pipeline.

- **Definition:** It involves retrieving data from various source systems, which may be in different formats.
- **Role:** Its primary role is to act as the intake valve for the pipeline, ensuring that all necessary raw data is successfully moved into a staging area without impacting the performance of the source systems.

Question 3: Explain the difference between CSV and Excel in terms of extraction and ETL usage.

While both are used for data, they differ significantly for ETL purposes:

- **CSV (Comma Separated Values):** Simple text files that are lightweight and highly portable. They are preferred for large datasets because they consume fewer resources during extraction.
- **Excel:** Binary files that support multiple sheets, formatting, and formulas. Extraction is more complex and slower compared to CSV because it requires specialized libraries to parse the formatting and metadata.

Question 4: Explain the steps involved in extracting data from a relational database.

Extracting data from databases typically involves these steps:

1. **Connection:** Establishing a secure connection using drivers like JDBC or ODBC.
2. **Schema Identification:** Identifying the specific tables and columns required for the task.
3. **Querying:** Executing SQL queries (e.g., `SELECT` statements) to pull the relevant data.
4. **Incremental/Full Load:** Deciding whether to pull all data or only the records that have changed since the last extraction.

Question 5: Explain three common challenges faced during data extraction.

Common hurdles include:

1. **Data Volume:** Handling massive amounts of data can strain network bandwidth and source system performance.
2. **Data Quality:** Dealing with inconsistent formats, missing values, or corrupted data at the source.
3. **Security and Access:** Navigating firewall restrictions and complex authentication protocols to reach the data.

Question 6: What are APIs? Explain how APIs help in real-time data extraction.

APIs (Application Programming Interfaces) act as a bridge between different software systems.

- **Real-time Role:** Unlike batch processing, APIs allow an ETL pipeline to request and receive data instantly as soon as an event occurs. This enables "streaming ETL," where data is extracted and moved in near real-time to support live dashboards or instant analytics.

Question 7: Why are databases preferred for enterprise-level data extraction?

Databases are preferred over flat files for large-scale operations because:

- **Concurrency:** They allow multiple users or processes to access data simultaneously.
- **Scalability:** They are designed to handle terabytes of data efficiently with indexing and partitioning.

- **Standardization:** Using SQL provides a uniform way to filter and join data before it even leaves the source.

Question 8: What steps should an ETL developer take when extracting data from large CSV files (1GB+)?

For large-scale file extraction, a developer should:

1. **Chunking:** Read the file in smaller "chunks" or blocks rather than loading the whole 1GB into memory at once.
2. **Parallel Processing:** If possible, split the file and process different sections simultaneously to save time.
3. **Schema Inference:** Explicitly define data types (e.g., telling the system a column is an integer) to avoid the overhead of the system guessing the type for millions of rows.