Data Extraction in ETL

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

ETL (Extract, Transform, Load) systems use various data sources to collect data for analysis and reporting.

1. Relational Databases:
   Store structured data in tables.
   Examples: MySQL, Oracle, SQL Server.
2. Flat Files:
   Simple file-based data sources used for data exchange.
   Examples: CSV, TXT, Excel files.
3. APIs / Web Services:
   Used to extract data from external applications, often in real time.
   Examples: REST APIs returning JSON or XML.
4. Cloud Storage:
   Stores large volumes of data in the cloud.
   Examples: AWS S3, Google Cloud Storage.
5. NoSQL Databases:
   Store semi-structured or unstructured data with flexible schemas.
   Examples: MongoDB, Cassandra.

Question 2 : What is data extraction? Explain its role in the ETL pipeline.
Data extraction is the first step in the ETL (Extract, Transform, Load) process. It involves collecting data from various source systems such as databases, files, APIs, and cloud platforms and transferring it to a staging area for further processing.
Role in the ETL Pipeline

1. Initiates the ETL process by making source data available for transformation.
2. Collects data from multiple heterogeneous sources into a common staging area.
3. Ensures data integrity by extracting data accurately without loss or corruption.
4. Supports different extraction methods such as full, incremental, and real-time extraction.
5. Minimizes impact on source systems by scheduling and optimizing extraction jobs.

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

- CSV (Comma-Separated Values) is a simple text-based format that is easy and fast to extract, making it highly suitable for automated ETL pipelines and large datasets.
- Excel files contain formatting, formulas, and multiple sheets, which makes extraction more complex and slower, requiring special tools or libraries.
- In ETL, CSV is preferred for large-scale processing, while Excel is mainly used for small datasets and manual reporting.

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

Steps Involved in Extracting Data from a Relational Database

1. Establish Connection:
   Connect to the relational database using credentials and JDBC/ODBC drivers.
2. Define and Execute Query:
   Write and run SQL queries to select required tables, columns, and records.
3. Store in Staging Area:
   Save the extracted data into a staging area or temporary storage for further ETL processing.

Question 5 : Explain three common challenges faced during data extraction.
Three Common Challenges in Data Extraction

1. Data Quality Issues:
   Source data may contain missing values, duplicates, or inconsistent formats.
2. Performance Impact:
   Extracting large volumes of data can slow down source systems and affect operations.
3. Heterogeneous Data Sources:
   Data comes from different sources with varying formats and structures, increasing complexity.

Question 6 : What are APIs? Explain how APIs help in real-time data extraction.
APIs (Application Programming Interfaces) are interfaces that allow different software applications to communicate and exchange data. APIs help in real-time data extraction by providing continuous or on-demand access to live data from external systems, usually in formats like JSON or XML.

Question 7 : Why are databases preferred for enterprise-level data extraction?
Databases are preferred for enterprise-level data extraction due to their reliability, scalability, and security.

1. High Data Integrity:
   Databases enforce constraints and rules to maintain accurate and consistent data.
2. Scalability:
   They efficiently handle large volumes of enterprise data using indexing and partitioning.
3. Security and Access Control:
   Databases provide authentication, authorization, and auditing mechanisms.
4. Efficient Extraction:
   Support incremental extraction and optimized SQL queries.
5. Reliability:
   Databases offer backup, recovery, and transaction management.

Question 8 : What steps should an ETL developer take when extracting data from large CSV files (1GB+)?
Extracting very large CSV files requires careful handling to ensure efficiency and reliability.

1. Use Chunk-Based or Streaming Read:
   Read data in batches instead of loading the entire file into memory.

2. Validate File Structure:
   Check delimiters, headers, and column consistency before processing.
3. Define Schema and Data Types:
   Explicitly set data types to avoid incorrect parsing and errors.
4. Implement Error Handling and Logging:
   Track processed rows and handle corrupt records without stopping the process.
5. Store Data in a Staging Area:
   Load extracted data into temporary storage for further transformation.