Data Extraction in ETL
Assignment Questions

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

Answer: In ETL, data can come from multiple sources depending on business needs.

1. Relational Databases
Structured data stored in tables with rows and columns.
Examples: MySQL, PostgreSQL, Oracle, SQL Server
Use case: Customer records, sales transactions

2. Flat Files
Simple file-based data storage.
Examples: CSV, TXT, Excel
Use case: Daily sales reports, employee attendance

3. APIs / Web Services
Data received from external systems in real time or batches.
Examples: REST APIs, SOAP APIs
Use case: Weather data, payment gateway transactions

4. Cloud Data Sources
Data stored on cloud platforms.
Examples: AWS S3, Google BigQuery, Azure Data Lake
Use case: Logs, clickstream data

5. NoSQL Databases
Semi-structured or unstructured data.
Examples: MongoDB, Cassandra
Use case: User activity logs, social media data

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

Answer: Data extraction is the process of collecting raw data from various source systems and moving it to a staging area for further processing.

Role in ETL Pipeline

- It is the first step of ETL (Extract → Transform → Load)
- Ensures accurate and complete data collection
- Maintains data consistency across systems
- Supports both full extraction and incremental extraction

Without proper extraction, transformation and loading cannot be done correctly.

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

Answer:

Feature	**CSV**	**Excel**
Format	Plain text	Binary
File size handling	Good for large files	Not suitable for very large files
Speed	Faster extraction	Slower extraction
Structure	Simple rows and columns	Multiple sheets, formulas
ETL usage	Highly preferred	Used for small datasets

CSV files are more efficient and scalable for ETL processes compared to Excel files.

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

Answer: 1. Understand Source Schema

Analyze tables, columns, and relationships

2. Establish Database Connection
Use JDBC/ODBC or ETL connectors

3. Write SQL Queries
Use `SELECT` queries with filters

4. Apply Extraction Type

- Full extraction
- Incremental extraction (using timestamps or IDs)

5. Extract Data to Staging Area
Store data temporarily for transformation

6. Validate Extracted Data
Check row counts, null values, and data consistency

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

Answer: 1. Data Quality Issues
Missing values, duplicates, or incorrect formats

2. Performance Problems
Large data volumes may slow down extraction

3. Schema Changes
Changes in source tables can break extraction logic

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

Answer: APIs (Application Programming Interfaces) allow systems to communicate and exchange data.

How APIs help in real-time extraction:

- Fetch data instantly when an event occurs
- Support JSON or XML formats
- Enable continuous data flow
- Used in streaming and near real-time ETL pipelines

Example: Extracting live stock prices or payment transaction data using REST APIs.

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

Answer: Databases are preferred because they:

- Handle large volumes of structured data
- Support ACID properties ensuring data reliability
- Allow complex querying
- Enable incremental extraction
- Provide better security and access control

This makes them suitable for enterprise data warehouses and analytics systems.

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

Answer: 1. Use Chunk-Based Reading
Read data in smaller chunks instead of loading all at once

2. Avoid GUI Tools
Prefer scripting tools like Python or Spark

3. Validate File Structure
Ensure delimiter and schema consistency

4. Compress Files
Use gzip or zip to reduce size

5. Parallel Processing
Split files and process simultaneously

6. Monitor Memory Usage
Prevent system crashes due to high memory consumption