NAME : SAGAR KUMAR BARICK

EMAIL : sagarbarick01@gmail.com

ASSIGNMENT NAME : **Data Extraction in ETL**

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

Sol : In ETL (Extract, Transform, Load), data sources are the origins from where data is extracted. These sources can be structured, semi-structured, or unstructured. Below are the major types of data sources used in ETL.

- **Databases**
- **Files**
- **APIs**
- **Cloud platforms**
- **NoSQL systems**
- **Streaming systems**
- **Enterprise applications**

The choice of data source depends on business needs, data format, and system architecture.

Relational Databases (RDBMS) – Structured Data

These store data in tables with rows and columns.

Examples:

- **MySQL**
- **Oracle**
- **SQL Server**
- **PostgreSQL**

An e-commerce company extracts customer orders from a MySQL database into a data warehouse for reporting.

Flat Files

These are simple file-based data storage formats.

Examples:

- CSV (Comma-Separated Values)
- Excel (.xlsx)
- Text files (.txt)

  A company receives daily sales data in CSV format from regional branches and loads it into a centralized system.

APIs (Application Programming Interfaces)

APIs allow systems to fetch data from external platforms in real time.

Examples:

- REST APIs
- SOAP APIs
- Twitter API
- Google Maps API

  A marketing company extracts live social media engagement data using a REST API.

Cloud-Based Data Sources

Data stored on cloud platforms or SaaS applications.

Examples:

- Google BigQuery
- Amazon S3
- Snowflake
- Salesforce

  Sales data stored in Salesforce CRM is extracted into a data warehouse for analysis.

NoSQL Databases – Semi-Structured Data

Used for flexible, non-tabular data storage.

Examples:

- MongoDB
- Cassandra
- DynamoDB

A mobile app stores user activity logs in MongoDB, which are later extracted for analytics.

Data Streams / Real-Time Sources

Continuous flow of data.

Examples:

- Apache Kafka
- IoT sensors
- Log files

An IoT company extracts real-time temperature sensor data for monitoring systems.

Enterprise Applications

Business systems used by organizations.

Examples:

- SAP
- Oracle ERP
- Workday

HR data from Workday is extracted for payroll reporting.

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

Sol: Data Extraction is the first step in the ETL (Extract, Transform, Load) process.
It involves collecting or retrieving data from various source systems such as databases, files, APIs, or cloud platforms.

Collecting Data from Multiple Sources

Data may come from:

- **Relational databases (MySQL, Oracle)**
- **CSV or Excel files**
- **APIs**
- **Cloud systems**
- **NoSQL databases**

Extraction gathers all this data into one place.

Ensuring Data Accuracy

During extraction, ETL developers ensure:

- **Correct data is selected**
- **No duplication**
- **Proper filtering**
- **Required columns only are extracted**

Preparing Data for Transformation

Once data is extracted:

- **It moves to the Transformation stage**
- **Data cleaning, filtering, joining, and formatting happen there**

Without proper extraction, transformation cannot be accurate.

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

Sol:

Feature	CSV File	Excel File (.xlsx)
Full Form	Comma-Separated Values	Microsoft Excel Workbook
Structure	Plain text file	Structured spreadsheet
Data Format	Stores only raw data	Can store data, formulas, charts, formatting
Sheets	Only one dataset	Multiple sheets in one file
Complexity	Simple	More complex
File Size	Usually smaller	Can be larger
Extraction Speed	Faster	Slightly slower
ETL Processing	Easy to parse	Requires Excel-specific drivers/libraries

CSV in ETL

Advantages:

- Lightweight and simple
- Easy to load using SQL tools or Python
- Faster processing
- Widely supported

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

Sol: Understand the Source Database

- **Identify required tables**
- **Understand table relationships (Primary Key / Foreign Key)**
- **Check data types and structure**
- **Analyze volume of data**

Establish Database Connection

Use connection tools like:

- **JDBC / ODBC**
- **Python (PyMySQL, psycopg2)**
- **ETL tools (Informatica, Talend, SSIS)**

Choose Extraction Type

- **Full Load – Extract entire table**
- **Incremental Load – Extract only new/updated records**

Handle Data Validation

- **Check for NULL values**
- **Validate data types**
- **Remove duplicates**
- **Log extraction results**

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

Sol: Data Quality Issues

Problem:

- **Missing values (NULLs)**
- **Duplicate records**
- **Incorrect formats (date, numbers)**
- **Inconsistent data**

Solution:

- **Validate data during extraction**
- **Apply data cleaning rules**
- **Use constraints and filters**

Large Data Volume

Problem:

- **Extracting millions or billions of records**
- **Slow query performance**
- **High load on source system**

Solution:

- **Use incremental load**
- **Extract in batches**
- **Use indexing**
- **Schedule extraction during off-peak hours**

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

Sol: An API acts like a bridge that allows systems to request and receive data from other systems.

Fetching Live Data

APIs allow ETL systems to request real-time data directly from applications.

Example:
Fetching live stock market prices every minute.

Automated Data Retrieval

ETL jobs can automatically call APIs at scheduled intervals.

Example:

- **Every 5 minutes**
- **Every hour**
- **On-demand**

This removes the need for manual file transfers.

Real-Time Streaming

Some APIs support continuous data flow.

Example:

- **Social media feeds**
- **Payment transactions**
- **IoT sensor data**

This helps in:

- **Fraud detection**
- **Live dashboards**
- **Monitoring systems**

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

Sol: Databases are preferred for enterprise-level data extraction because they provide:

- **Structured data**
- **High performance**
- **Strong security**
- **Data integrity**
- **Scalability**
- **Support for incremental loading**

They are reliable and suitable for handling large-scale business data.

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

Sol : Use Efficient File Formats (If Possible)

After initial extraction, convert CSV to optimized formats like:

- **Parquet**
- **ORC**

These formats:

- **Reduce file size**
- **Improve processing speed**

Perform Data Validation During Extraction

- **Check for missing values**
- **Validate data types**
- **Handle corrupted rows**

Log errors instead of stopping the entire process.

Use Parallel Processing

If system allows:

- **Split file into smaller parts**
- **Process multiple chunks simultaneously**

This improves performance.

Use Proper Hardware Resources

- **Sufficient RAM**
- **SSD storage**
- **Proper CPU allocation**

Large files require good infrastructure.

Implement Incremental Processing (If Applicable)

If file is updated daily:

- **Process only new data**
- **Maintain checkpoint or last processed row**

Monitor and Log the Process

- **Track processing time**
- **Log errors**
- **Monitor memory usage**

This helps in troubleshooting.