# Assessment Overview

The candidate will design, build, and optimize a small-scale data pipeline that processes JSON data representing sales transactions for a fictional e-commerce company. The project involves data extraction, transformation, and loading (ETL) from JSON, database management, data quality checks, and visualization using Power BI.

## Dataset

For this assessment, candidates will work with a JSON dataset provided below, simulating e-commerce sales transactions. The dataset contains nested structures, missing values, and inconsistencies to test data cleaning and transformation skills. Candidates should use this sample data and are encouraged to scale it to 100-500 records by duplicating or modifying entries for testing purposes.

## Sample JSON Dataset (sales\_data.json):

[

{"transaction\_id": "T001", "customer\_id": "C001", "product": {"id": "P01", "name": "Laptop", "category": "Electronics", "price": 999.99}, "quantity": 2, "date": "2023-01-15T10:30:00Z", "region": "North"},

{"transaction\_id": "T002", "customer\_id": "C002", "product": {"id": "P02", "name": "Mouse", "category": "Accessories", "price": 19.99}, "quantity": 5, "date": "2023-02-10T14:15:00Z", "region": "South"},

{"transaction\_id": "T003", "customer\_id": null, "product": {"id": "P01", "name": "Laptop", "category": "Electronics", "price": 999.99}, "quantity": -1, "date": "2023-03-05", "region": "East"},

{"transaction\_id": "T004", "customer\_id": "C003", "product": {"id": "P03", "name": "Monitor", "category": "Electronics", "price": 299.50}, "quantity": 3, "date": "2023-04-20T09:00:00Z", "region": "West"},

{"transaction\_id": "T005", "customer\_id": "C001", "product": {"id": "P02", "name": "Mouse", "category": "Accessories", "price": 19.99}, "quantity": 10, "date": "2023-05-12", "region": "North"}

]

**Instructions:** Save this JSON as sales\_data.json. Scale it to 100-500 records by duplicating or modifying entries to test your pipeline. The dataset includes nested product data, missing customer\_id, negative quantity, and inconsistent date formats for cleaning practice.

# Task Description

**1. Data Pipeline Development**

**Objective:** Build an ETL pipeline to extract data from the JSON file, transform it into a clean and structured format, and load it into a relational database.

**Details:**

Extract data from sales\_data.json using Python (e.g., json library or Pandas).

**Perform transformations:**

* Flatten the nested product object into columns (e.g., product\_id, product\_name, category, price).
* Standardize date to a consistent format (e.g., YYYY-MM-DD).
* Calculate total\_value (quantity \* price) for each transaction.
* Handle missing or invalid data (e.g., null customer\_id, negative quantity).
* Load the transformed data into a relational database (e.g., SQLite, PostgreSQL, or MySQL).

**Deliverables:**

* A Python script or Jupyter notebook implementing the ETL process.
* SQL script for creating the database schema (e.g., CREATE TABLE sales (...)).
* **Upload the above to a GitHub repo and share the URL. The repo should be publicly available.**

**2. Database Design and Querying**

**Objective:** Design an optimized database schema to store the transformed JSON data and enable efficient querying.

**Details:**

* Create a table (e.g., sales) or normalized tables (e.g., transactions, products, customers) with appropriate relationships (e.g., transaction\_id, product\_id as keys).
* Add an index on frequently queried fields (e.g., date or region).

**Write SQL queries to answer:**

* Total sales value by region.
* Top 5 products by total sales value (using total\_value).
* Monthly sales trends (aggregate total\_value by month).

**Deliverables:**

* SQL scripts for the required queries with sample results.
* **Upload the above to the same GitHub repo with ETL pipeline script.**

**3. Data Quality Checks**

**Objective:** Implement checks to ensure data accuracy and integrity throughout the pipeline.

**Details:**

* Validate JSON data for missing fields (e.g., replace null customer\_id with "Unknown").
* Handle negative quantity values (e.g., set to 0 or flag for review).
* Ensure no duplicate transaction\_id entries are loaded.
* Log anomalies (e.g., to a text file or console) during processing.
* Deliverables:
* Code snippets or documentation in the ETL script explaining the implemented quality checks.

**4. Performance Optimization**

**Objective:** Optimize the pipeline for efficiency.

**Details:**

* Use batch processing to load JSON data into the database (e.g., Pandas to\_sql with chunks).
* Optimize one SQL query (e.g., add an index and explain its impact).
* Deliverables:
* Documentation or code comments detailing optimizations applied.

**5. Data Visualization with Power BI**

**Objective:** Connect the database to Power BI and create a dashboard with DAX queries to visualize and analyse insights.

**Details:**

Connect Power BI to the database (e.g., via ODBC or direct connector for SQLite/PostgreSQL).

**Create at least 4 visuals:**

**Bar chart:** Total sales by region.

**Line chart:** Monthly sales trend (using date).

**Table:** Top 5 products by total\_value.

**Card:** Display a key metric (e.g., average sales per transaction or total sales).

Add a slicer (e.g., filter by category, region, or date).

Use DAX to create the following five measures:

* Calculate total sales across all transactions for use in visuals like the bar chart.
* Compute the average sales value per transaction, useful for a card visual.
* Calculate year-to-date sales for time-based analysis in the line chart.
* Count transactions with a total value exceeding $1000, highlighting significant sales.
* Calculate total sales for the "Electronics" category, useful for category-specific insights.

**Deliverables:**

* Power BI file (.pbix) containing the dashboard with visuals, slicers, and all five DAX measures.
* Brief description of visuals, DAX measures, and insights (e.g., in a text file or Power BI comments).
* **Attach the pbix file to the email**