Gramedia Digital
Data Engineer Take-Home Test

Scenario

You are part of a data team building an analytics platform for an e-commerce company. The team needs a data pipeline that extracts product and transaction data, cleans it, and prepares it for analytical use. Your task is to design and implement a simple ETL pipeline using any tools or frameworks you prefer.

Requirements

1. Data Pipeline

- Extracts product and sales transaction data (from API, CSV, or JSON files).
- Transforms the data by cleaning and joining relevant fields:
  - Normalize product categories.
  - Convert timestamps to consistent timezone.
  - Calculate total_sales (quantity × price).
- Loads the cleaned data into a database or data warehouse (e.g., PostgreSQL, BigQuery, SQLite).

Your final dataset should include:

Field	Description
transaction_id	Unique ID for each transaction
product_id	Product identifier
product_name	Product name
category	Cleaned category name
quantity	Units sold
price	Price per unit
total_sales	Derived metric (quantity × price)
transaction_date	Date/time of transaction

2. Data Quality & Validation

- No missing product names or prices.
- Quantity and price must be greater than zero.
- No duplicate transaction IDs.
- You may use Python (pandas, Great Expectations, PySpark), SQL queries, or any validation framework you're familiar with.

3. Deliverables

- ETL code (Python, SQL, or other language).
- Sample output dataset (cleaned_data.csv or SQL table dump).
- README.md including:
- - Setup instructions.
- - Description of ETL logic.
- - Explanation of data validation.
- - (Optional) Diagram of your pipeline.

Technical Expectations

- Use Python (pandas, SQLAlchemy, Airflow, or similar).
- Use a relational database (SQLite, PostgreSQL, MySQL) or local file output.
- Organize code modularly (extract, transform, load functions).
- Use Git for version control.
- Use Docker (optional bonus).

🧮 Sample Data Sources (Free APIs / Datasets)

- DummyJSON — https://dummyjson.com/products
- FakeStore API — https://fakestoreapi.com
- Kaggle Datasets — e.g., "E-commerce Sales Data"
- Or create your own synthetic CSVs.

Evaluation Criteria

Category	Description
Code Quality	Clean, modular, and maintainable ETL code.
Data Accuracy	Correct transformations and validation rules.
Performance	Efficient data handling for medium-scale datasets.

Documentation	Clear explanations and pipeline instructions.
Analytical Thinking	Logical transformation flow and meaningful metrics.
Reproducibility	Code can run easily in another environment.
Bonus	Use of orchestration (Airflow, Prefect) or cloud data tools.

Optional Enhancements (Bonus)

- Add data quality reports (e.g., Great Expectations summary).
- Add Airflow DAG or Prefect flow for orchestration.
- Containerize the pipeline with Docker.
- Deploy output to a cloud service (BigQuery, AWS RDS, etc.).
- Include visualizations or metrics summary in a notebook.