# Assignment Title: ETL Pipeline Development

Objective**:** The goal of this assignment is to assess the intern's ability to design and implement an ETL (Extract, Transform, Load) pipeline using Python, SQL, and other data engineering technologies. The candidate will be responsible for extracting data from a source, performing transformations on it, and loading it into a target destination.

Instructions**:**

`

1. Data Source: You will be provided with a CSV file (`Updated\_sales\_data.csv`) containing fictional sales data. The data includes columns like `order\_id`, `product`, `quantity`, `price`, and `order\_date`. You can assume the CSV file is located in the same directory as your code.

2. ETL Pipeline: Your task is to create a Python script or Jupyter Notebook that performs the following tasks:

a. **Extract**: Read the data from the ` Updated\_sales\_data.csv ` file into a pandas DataFrame or any other appropriate data structure.

b. **Transform**: Perform the following transformations on the data:

* Calculate the total sales amount for each order (quantity \* price) and add a new column `total\_sales`.
* Calculate the year and month of each order from the `order\_date` column and add new columns `order\_year` and `order\_month`.
* Aggregate the data to calculate the total sales amount for each product in each month and store it in a separate DataFrame.

c. **Load**: Create a SQL database (you can use SQLite or any other SQL database) and define tables to store the transformed data. Load the transformed data into appropriate SQL tables. You should have at least two tables - one for orders and another for product sales by month.

3. SQL Queries: Write SQL queries to answer the following questions:

- What was the total sales amount for each product in the last quarter (last 3 months)?

- List the top 5 products by total sales amount for the entire dataset.

- Calculate the monthly average sales for each product over the entire dataset.

4. Documentation: Provide clear documentation in the form of comments in your code explaining each step of the ETL process, including how to run the script and any assumptions made.

# Bonus Question: Data Validation and Error Handling

Data Validation: Add data validation checks to your ETL pipeline to ensure data quality. Specifically, perform the following validations:

a. Check for missing values in essential columns (e.g., order\_id, product\_id, quantity, price, order\_date) and handle them appropriately (e.g., skip the row or log the issue).

b. Check for any anomalies or outliers in the data (e.g., negative quantities or prices) and handle them appropriately (e.g., correct the values or log the issue).

Error Handling: Implement error handling mechanisms to gracefully handle exceptions or errors during the ETL process. For example:

a. If there is an issue with reading the source data file (e.g., file not found or corrupted), log the error and gracefully exit the script with an error message.

b. If there are issues with loading data into the SQL database (e.g., table already exists or data type mismatches), handle these errors and log appropriate messages.

**Submission:**   
Please submit the following:

* Your Python script or Jupyter Notebook with the ETL pipeline.
* A separate SQL file containing the table creation and data insertion statements.
* A document explaining your approach, any challenges faced, and any additional improvements or optimizations you would make if given more time.

**Evaluation Criteria:**

Assignment will be evaluated on the following criteria:

* Accuracy of the SQL queries and data transformations.
* Correctness of the exported CSV files.
* Clarity and organization of the documentation.
* Code readability and adherence to best practices.