Hands-on Lab: Final Project

Estimated Effort: 75 minutes

Introduction

In this final project, you will apply all the knowledge gained throughout this course using a real-world project scenario and data that simulates the various tasks that a data engineer must perform. Consider the following scenario.

Scenario

You are a data engineer hired by a European online retail company to design a data workflow for their operations. You are required to perform all of the following tasks for them:

- Propose a detailed data architecture for the whole data process.
- Propose a detailed data warehouse schema and design its entity relationship diagram (ERD).
- Propose the infrastructure requirements for the required data architecture.
- Create an ETL pipeline to clean, process, and load the data to an SQL server for analysis. Test the pipeline on a sample database.
- Query the SQL database to access data from the server.
- Implement data analysis and data mining strategies on the final data.

Data set

This lab uses the Online Retail data set available in the UCI ML library, available publically under the CC BY 4.0 license.

About generative AI classroom lab

▶ Click here

Notes:

1. The prompts used in this lab are for your reference only. You can create your own prompts and generate responses using generative AI.
2. Since AI-generated outputs are dynamic, you may receive different responses even though you've used the same prompt from this lab.

Test environment

The testing environment for this project lab is linked and available within the course immediately following this lab link. Please open the lab environment in a separate window to the side and complete the lab setup process.

You can test the code generated in this lab within the testing interface using **GPT-5 Nano**.

Data architecture

To propose a data architecture for the retail system, you will set up the following guidelines based on the client specifications.

The client does not want to go for cloud based processing resources. They want an SQL-based central data repository that their employees from multiple countries can access for their use.

You can use the GenAI model to propose a data architecture. Try to create a prompt that will give you the expected response.

▶ Click here for a sample prompt
▶ Click here for a sample response

Data warehouse schema and ERD

First, you need to set up the data warehouse schema and its ERD diagram. For that task, you need clearly defined requirements from the client as to the kind of data they want recorded. Assume that you received the following information from the client.

1. The client wants to record customer information, seller information, inventory information, and transaction invoice information.
2. The client wants the final data prepared such that the final record of sales invoices displayes the headers `InvoiceNo`, `StockCode`, `Description`, `Quantity`, `InvoiceDate`, `UnitPrice`, `CustomerID` and `Country`.

To define the schema of the data warehouse that meets these requirements, you can now write a prompt on the GenAI platform.

▶ Click here for a sample prompt
▶ Click here for a sample response

You can further use the generative AI platform to generate the SQL codes for creating this warehouse, and use this code to create the ERD on the DbDiagram interface, as explained earlier in the course.

▶ Click here for a sample prompt
▶ Click here for a sample response
▶ Click here for a sample ERD

Infrastructure requirements

You now need to define the infrastructure requirements for such a setup. You can make use of the same chat for making this prompt as the GenAI will draw context from the previous responses and give you a tailored response.

You will frame a prompt that can use the context from the previous steps and create the infrastructure requirements for your design.

▶ Click here for a sample prompt
▶ Click here for a sample response

The ETL workflow

At this point, you can assume that the central data collection has taken place and the invoice details are available as a CSV file on remote server. You need to create an ETL pipeline that reads this file, cleans the data based on the client specifications and loads the data to a SQL server. Following specifications for this task have been shared.

1. The recorded data is available on the following URL:

   ```
   https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMSkillsNetwork-AI0273EN-SkillsNetwork/labs/v1/m3/data/Project_c
   ```

2. InvoiceNo starting with the character `C` is a credit entry and should be removed from the record before analysis.
3. StockCode values of `C2`, `D`, `M`, and `POST` correspond to Carraige, Discount, Manual and Postage entries, all of which are not required for our analysis.
4. There are a few entries where the CustomerID is missing. Such entries can be removed from the data before your analysis.
5. Load the final transaction record to an SQLite3 database `Invoice_Records` under the table `Purchase_transactions`.

Create a prompt and use it on the generative AI platform to generate a Python code that can create the required data processing pipeline.

▶ Click here for a sample prompt
▶ Click here for a sample response
▶ Sample content of testing code
▶ Click here for a sample output

Querying the database

After the data is available on the central data repository, you can use SQL queries to extract the data directly into your Python coding interface. For the next part of your project, you are required to extract the data of a specific country–let's use Germany. Use the generative AI model to create a code snippet that you can add to your previous code in the testing interface and that will run a SQL query on the data to extract the details transactions for the specified country.

▶ Click here for a sample prompt
▶ Click here for a sample response
▶ Sample code for testing
▶ Click here for a sample output

Data analysis and data mining

One of the most relevant and important techniques for analyzing transactional data is association rule mining. You are required to implement an Apriori algorithm to mine association rules from the data extracted in the previous step. As a result, you will be able to identify the items that have the most likelihood to be purchased together. This information is necessary for the company to efficiently develop their marketing and advertising strategies.

You can use generative AI to create the code for implementation. You can write a prompt that creates code for the implementation of Apriori algorithm for association rule mining on the extracted data frame.

Consider the following prompt that describes the different steps involved in implementing an Apriori algorithm.

```
For the data frame extracted here, write a brief python code to execute the apriori algorithm and extract association rules for the
1. Group the records by InvoiceID and Description, along with their total quantities
2. Unpack the data into a table, making the InvoiceNo, the row indexes, Unique descriptions as column indexes and the total quantiti
3. Apply one-hot encoding on this table, making the value True if the Item description existed in the invoice and False if it didn't
4. Perfrom Apriori algorithm on this data and extract the required association rules.
```

You can expect to see a respnese similar to the following sample response:

▶ Click here for a sample response

You can comment out all previous print statements and combine these code snippets with the existing test code file, making completed code for reference.

▶ Click here for sample final code
▶ Click here for sample output

You can infer that if an item from the antecedent column is picked, then it can be said with the shown value of confidence that the corresponding consequent is also going to be picked for purchase in the same invoice.

Conclusion

Congratulations on completing this project!

You are now trained in using generative AI for end-to-end data engineering applications, including but not limited to:

- Data architecture design
- Data warehouse and schema design
- Infrastructure requirements determination
- ETL pipeline integration
- Querying databases
- Data analysis and mining

Author(s)

Abhishek Gagneja