First, I found data sources to supplement my project. I used the example classic models database (in mysqlsampledatabase.sql), and an API call to get another data file. I used a public API called FFXIV Collect to simulate in-game items as products. This data warehouse seeks to simulate the relationship between players of a game and the parent company.

I made the API call with the command-line command:

curl 'https://ffxivcollect.com/api/mounts' -o “mounts.json”

This created the json file containing information on in-game items known as mounts. I converted this file into a sql file to insert a new table into the database, and also added an id column.

After this, I began to use the jupyter notebook to develop the ETL pipeline to modify the data warehouse.

To create the ETL pipeline, we first run the two sql files in order (mysqlsampledatabase.sql and mountproducts.sql), and perform some basic transforms, and make sure both are using the classicmodels database. This creates the product_db database, which will function as our warehouse. At this point, run through each cell individually until you reach dim_date transformations. There is a try except block at that point in case the dim_date table is not created.

After running some other transformations, before we get to dim_date transformations, the Create_Populate_DimDate.sql file should be run to create the dim_date table in the new project_db data warehouse.

As in a usual ETL pipeline, we modify the number of columns and insert new keys into each dim table. We create the dim_customers table, dim_mounts table, dim_employees, dim_date, and the fact_orders table. Dim_mounts essentially replaced the dim_products table, so I created a key for both tables and merged them. I also merged the orders table and order_details table to create the basic fact_orders table. From there, I replaced the id columns with the corresponding key colums, and finally reorganized the column order. Lastly, I wrote the new fact_orders table back into my data warehouse. A sql query was also written to test the data warehouse, which merges several tables.