**Milestone 2**

Team: Owen Randolph, Gabe Tharp, Marcos Fernandez

1. Attempted to change the date format. The date format in CSV file from MM/DD/YYYY to YYYY-MM-DD in Order Date and Ship Date columns doesn’t work. Windows reverts to the MM/DD/YYYY date format even after saving. This has to be done using Python code in a Jupyter notebook and pushed directly into MySQL Workbench using a connector (see Jupyter notebook)
2. Create Schema for new database “ecommerce” in My SQL Workbench

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. Create Pre-normalized Table “ecommerce” in MySQL Workbench

A screenshot of a computer

AI-generated content may be incorrect.

1. Data Cleaning using Python Pandas library (see Jupyter notebook code)

* Removal of “$” before the numbers in the Sales, Profit, and Discount values
* Round those values to 2 significant figures to account for cents in the dollar amount
* Change the dates in Order Date and Ship Date columns to YYYY-MM-DD format for MySQL readability

1. Load data into MySQL workbench table and test to make sure it worked smoothly by checking the number of rows and trying a test query.

A screenshot of a computer

AI-generated content may be incorrect.

1. Normalize the Database
   1. 1NF : a table is normalized to the first form if each column has individual values, each row is distinct (no duplicates), and each column has only values of the same type. These were satisfied when we created the database. To check for these we use a Python for-loop that checks all values for commas or colons.

**The data table was already in first normalized form from the beginning, no adjustments necessary.**

* 1. 2NF: a table is normalized to the second form if it satisfies 1NF and has no partial dependencies. This is satisfied when the non-key columns are completely dependent on the entire primary key.

To get this data table into the second normalized form, we need to break three new tables:

* Customers: Customer\_Id (PK), Customer\_Name, Segment
* Orders: Order\_Id (PK), Customer\_Id (FK), Order\_Date, Ship\_Date, Ship\_Mode, Postal\_Code, Region
* Products: Product\_Id (PK), Product\_Name, Category, Sub-Category
* Order Details: Order\_Id (PK, FK), Product\_Id (PK, FK), Quantity, Discount, Sales, Profit

We use CRUD methods to create the new tables in MySQL Workbench and insert the correct data into each new table. See Appendix (CRUD methods)

* 1. 3NF: a table is normalized to the third form if there are no transitive dependencies, meaning there are no non-primary key fields what are reliant on only the primary key. Foreign keys have to also be added for the new tables:
* In Products table, Sub-Category is dependent on Category, so we break it into a new table and add a new ID for it (SubCategory\_ID)
* We create a new table called Locations because Postal code is dependent on region

**Reference**

Chat GPT used for writing python code to prove 1NF to check for non-atomic values, duplicate rows, and consistent data types

**Appendix**