# DATA MODEL

After downloading the excel files, I noticed the following issues:

1. The column names across all excel files are not uniform.
2. The data across most fields are not standardized, so normalizing it can be very difficult. Such fields include GL Account, Merchant Name etc.
3. There are a lot of rows that are used for calculating subtotals and totals which will be cleaned by removing them from the data imported.
4. Currency codes are included but the names were not included. As a result, I need to create a currency list table that has all the currencies, including the codes and descriptions.

As a result of the issues listed above, I try as much as possible to clean the data and normalize it where possible. From the data imported into the staging table (pcard\_staging\_table), I was able to create the following tables and their relationships:



**currency\_list:** This is created and populated the first time the code is executed. It contains currency codes and their description. The currency\_code is the primary key.

**division:** This contains the list of all divisions. The division\_id is autogenerated and it is also the primary key.

**merchant\_type:** This contains the list of merchant types and their description. The merchant\_type column is the primary key.

**transactions:** This table holds all the transactions imported from the excel files into the staging table. It is the main table. It has many-to-one relationships with the aforementioned three tables as shown in the previous diagram. There is also an index against these three columns to improve performance: division\_id, transaction\_date, and card\_posting\_dt.

# Data Pipeline and Warehouse

The program maintains two folders that can be configured in the *config.py* file. The two folders can be configured on setup, and they are:

UNPROCESSED\_FOLDER: This folder will hold all the Microsoft Excel files that will be processed. When the program is running, it checks this folder and after upload, moves the file to the processed folder below.

PROCESSED\_FOLDER: This folder will store all files that have been processed by the program.

Also included in the *config.*py file is the configuration value for the Microsoft SQL Server instance name as well as the database name. This should also be configured before running the program. During processing, the data is moved to the staging table and awaits further processing (data cleaning and normalization).

If there are any Excel files to the processed, they should be moved to the unprocessed folder, and the program will handle the import and processing. You can keep track of relevant events by setting the debug value in *config.py* to the preferred value.

Once the program starts running, it will not stop running until the program is stopped or closed as the case may be. While running, it will be checking the unprocessed folder for excel files, and then process them if it sees any file. Also, all relevant events that happen during the time the program is running are logged into a file that can be configured in the *config.py* file.

The ETL engine is being handled by both Python and SQL Server Stored Procedure. The procedure (p\_normalize\_data) for the normalization of data will be required to be installed after database creation and before the program can be executed. The ETL process follows the following steps:

* The program checks for the necessary tables and creates them if missing.
* The program reads excel files from the unprocessed folder.
* The program then cleans the data by excluding records that do not have value in the division field.
* After cleaning, the program inserts the cleaned data into the staging table.
* The program then calls the stored procedure (p\_normalize\_data) to normalize the data by populating the 4 tables shown in the mapping diagram on the previous page.