###################################################################################################

# Tasty Bytes is a fictitious, global food truck network, that is on a mission to serve unique food

# options with high quality items in a safe, convenient and cost effective way. In order to drive

# forward on their mission, Tasty Bytes is beginning to leverage the Snowflake Data Cloud.

# In this Python Worksheet, we will walk through the end to end process required to load a CSV file

# containing Menu specific data that is currently hosted in Blob Storage.

# Please click Run and see details below on what each step it doing. The final output will return

# a Dataframe

###################################################################################################

###################################################################################################

# Step 1 - To start, we must first import our Snowpark Package and a few Functions and Types

###################################################################################################

## Note: You can add more packages by selecting them using the Packages control and then importing them.

import snowflake.snowpark as snowpark

from snowflake.snowpark.functions import col

from snowflake.snowpark.types import StructField, StructType, IntegerType, StringType, VariantType

###################################################################################################

# Step 2 - Let's now define our Session and Create our Database, Schema and Blob Stage

###################################################################################################

def main(session: snowpark.Session):

# Use SQL to create our Tasty Bytes Database

session.sql('CREATE OR REPLACE DATABASE tasty\_bytes\_sample\_data;').collect()

# Use SQL to create our Raw POS (Point-of-Sale) Schema

session.sql('CREATE OR REPLACE SCHEMA tasty\_bytes\_sample\_data.raw\_pos;').collect()

# Use SQL to create our Blob Stage

session.sql('CREATE OR REPLACE STAGE tasty\_bytes\_sample\_data.public.blob\_stage url = "s3://sfquickstarts/tastybytes/" file\_format = (type = csv);').collect()

###################################################################################################

# Step 3 - Now, we will define the Schema for our CSV file

###################################################################################################

# Define our Menu Schema

menu\_schema = StructType([StructField("menu\_id",IntegerType()),\

StructField("menu\_type\_id",IntegerType()),\

StructField("menu\_type",StringType()),\

StructField("truck\_brand\_name",StringType()),\

StructField("menu\_item\_id",IntegerType()),\

StructField("menu\_item\_name",StringType()),\

StructField("item\_category",StringType()),\

StructField("item\_subcategory",StringType()),\

StructField("cost\_of\_goods\_usd",IntegerType()),\

StructField("sale\_price\_usd",IntegerType()),\

StructField("menu\_item\_health\_metrics\_obj",VariantType())])

###################################################################################################

# Step 4 - Using the Schema, let's Create a Dataframe from the Menu file and Save it as a Table

###################################################################################################

# Create a Dataframe from our Menu file from our Blob Stage

df\_blob\_stage\_read = session.read.schema(menu\_schema).csv('@tasty\_bytes\_sample\_data.public.blob\_stage/raw\_pos/menu/')

# Save our Dataframe as a Menu table in our Tasty Bytes Database and Raw POS Schema

df\_blob\_stage\_read.write.mode("overwrite").save\_as\_table("tasty\_bytes\_sample\_data.raw\_pos.menu")

###################################################################################################

# Step 5 - With the table saved, let's create a new, filtered Dataframe and Return the results

###################################################################################################

# Create a new Dataframe reading from our Menu table and filtering for the Freezing Point brand

df\_menu\_freezing\_point = session.table("tasty\_bytes\_sample\_data.raw\_pos.menu").filter(col("truck\_brand\_name") == 'Freezing Point')

# return our Dataframe

return df\_menu\_freezing\_point