databricks ASP 2.2L - Ingesting Data Lab

Ingesting Data Lab

Read in CSV files containing products data.

Tasks

1. Read with infer schema
2. Read with user-defined schema
3. Read with schema as DDL formatted string
4. Write using Delta format

```
%run ../../Includes/Classroom-Setup
```

```
Deleted the working directory dbfs:/user/ajaypanday678@hotmail.com/dbacademy/aspwd/asp_2_2l_ingesting_data_lab
```

```
Your working directory is
dbfs:/user/ajaypanday678@hotmail.com/dbacademy/aspwd

The source for this dataset is
wasbs://courseware@dbacademy.blob.core.windows.net/apache-spark-programming-with-databricks/v02/

Skipping install of existing dataset to
dbfs:/user/ajaypanday678@hotmail.com/dbacademy/aspwd/datasets

Out[5]: DataFrame[key: string, value: string]
```

1. Read with infer schema

- View the first CSV file using DBUtils method `fs.head` with the filepath provided in the variable `single_product_cs_fil_path`
- Create `products_df` by reading from CSV files located in the filepath provided in the variable `products_csv_path`
  - Configure options to use first line as header and infer schema

```
# TODO
single_product_csv_file_path = f"{datasets_dir}/products/products.csv/part-00000-tid-1663954264736839188-daf30e86-
5967-4173-b9ae-d1481d3506db-2367-1-c000.csv"
# print(single_product_csv_file_path)
dbutils.fs.head(single_product_csv_file_path)

products_csv_path = f"{datasets_dir}/products/products.csv"
products_df = spark.read.csv(products_csv_path,header=True,inferSchema=True)

products_df.printSchema()

root
 |-- item_id: string (nullable = true)
```

```
 |-- name: string (nullable = true)
 |-- price: double (nullable = true)
```

1.1: CHECK YOUR WORK

```
assert(products_df.count() == 12)
```

2. Read with user-defined schema

Define schema by creating a `StructType` with column names and data types

```
# TODO
from pyspark.sql.types import DoubleType, StringType, StructType, StructField

user_defined_schema = StructType([
    StructField("item_id", StringType(), True),
    StructField("name", StringType(), True),
    StructField("price", DoubleType(), True)
])

products_df2 = spark.read.csv(products_csv_path,schema=user_defined_schema,header=True)
```

2.1: CHECK YOUR WORK

```
assert(user_defined_schema.fieldNames() == ["item_id", "name", "price"])
```

```
from pyspark.sql import Row

expected1 = Row(item_id="M_STAN_Q", name="Standard Queen Mattress", price=1045.0)
result1 = products_df2.first()

assert(expected1 == result1)
```

3. Read with DDL formatted string

```
# TODO
ddl_schema = "item_id string,name string,price double"

products_df3 = spark.read.csv(products_csv_path,schema=ddl_schema,header=True)
```

3.1: CHECK YOUR WORK

```
assert(products_df3.count() == 12)
```

4. Write to Delta

Write `products_df` to the filepath provided in the variable `products_output_path`

```
# TODO
products_output_path = working_dir + "/delta/products"
products_df.write.format("delta").mode("overwrite").save(products_output_path)
```

4.1: CHECK YOUR WORK

```
verify_files = dbutils.fs.ls(products_output_path)
verify_delta_format = False
verify_num_data_files = 0
for f in verify_files:
    if f.name == "_delta_log/":
        verify_delta_format = True
    elif f.name.endswith(".parquet"):
        verify_num_data_files += 1

assert verify_delta_format, "Data not written in Delta format"
assert verify_num_data_files > 0, "No data written"
del verify_files, verify_delta_format, verify_num_data_files
```

Clean up classroom

```
classroom_cleanup()
```

```
Dropped the database dbacademy_odl_user_534131_databrickslabs_com_aspwd_asp_2_2l_ingesting_data_lab
Deleted the working directory dbfs:/user/odl_user_534131@databrickslabs.com/dbacademy/aspwd/asp_2_2l_ingesting_data_la
b
```