# Retail Business

Dataset is:

- Customers
- Products
- Category
- Orders

Seven Scenarios were asked
( mainly ingest -> a simple query -> save job)

Scenario 1:

Customers tab delimited

1) Reading

```
from pyspark.sql import types

schema = types.StructType([
types.StructField('id', types.StringType(), True),
types.StructField('first_name', types.StringType(), True),
types.StructField('last_name', types.StringType(), True),
types.StructField('email', types.StringType(), True),
types.StructField('password', types.StringType(), True),
types.StructField('street', types.StringType(), True),
types.StructField('city', types.StringType(), True),
types.StructField('state', types.StringType(), True),
types.StructField('zipcode', types.StringType(), True)

])
```

```
df =  spark.read.format('csv')\
.schema(schema)\
.option('sep', r'\t')\
.load(path)
```

2) Filtering by state:

```
result = df.filter(df.state == "CA")
```

3) Concat:

```
]: from pyspark.sql import functions as F
result=result.withColumn("row",concat( '
    F.col('first_name'), F.lit(' '), \
    F.col('last_name')))
```

4) Saving:

```
result.write.mode('Overwrite').text
```

Scenario 2:

Orders Parquet

The downloaded parquet files didn't work.

Solution: parquetize order csv

```
df_orders =  spark.read.format('parquet')\
            .load(path_orders_pq)
```

```
: result_orders = df_orders.filter(df_orders.order_status == "COMPLETE")
```

```
result_orders.write.mode('Overwrite') \
.option("compression", "gzip") \
.json('C:/Users/DevAdmin/MohyWorkSpace/Retail_Proj/result/scenario2/solution')
```

## Scenario 3:

## Customers tab delimited

```
df =  spark.read.format('csv')\
.schema(schema)\
.option('sep', r'\t')\
.load(path)
```

```
result = df.filter(df.city == "Caguas")
```

```
result.write.mode('Overwrite') \
.option("compression", "snappy") \
.orc('C:/Users/DevAdmin/MohyWorkSpace/Retail_Proj/result/scenario3/solution')
```

## Scenario 4:

## Categories

```
schema = types.StructType([
types.StructField('id', types.IntegerType(), True),
types.StructField('dept_id', types.IntegerType(), True)
types.StructField('name', types.StringType(), True)

])
```

```
df =  spark.read.format('csv')\
        .schema(schema)\
        .load(path)
```

```
df.write.mode('Overwrite') \
.option("compression", "lz4") \
.csv('C:/Users/DevAdmin/MohyWorkSpace
```

## Scenario 5:

```
!spark-submit --packages org.apache.spark:spark-avro_2.12:3.3.1 avro_5.py
```

```
spark = SparkSession.builder \
        .master("local[*]") \
        .getOrCreate()

df = spark.read.format("avro").load(path)
df_filter = df.filter(df.product_price > 1000.0)
df_filter.write.mode('Overwrite') \
.option("compression", "snappy") \
.parquet('C:/Users/DevAdmin/MohyWorkSpace/Retail_Proj/result/scenario5/solution')
```

## Scenario 6:

```
!spark-submit --packages org.apache.spark:spark-avro_2.12:3.3.1 avro_6.py
```

```
path = 'C:/Users/DevAdmin/MohyWorkSpace/data-files/products_avro/*.avro'
df = spark.read.format("avro").load(path)
df_filter = df.filter(df.product_price > 1000.0) \
.filter(F.col('product_name').like("%Treadmill%"))

df_filter.write.mode('Overwrite') \
.option("compression", "gzip") \
.parquet('C:/Users/DevAdmin/MohyWorkSpace/Retail_Proj/result/scenario6/sol
```

## Scenario 7:

```
df_orders =  spark.read.format('parquet')\
            .load(path_orders_pq)
```

```
result_orders = df_orders.filter(df_orders.order_status == "PENDING_PAYMENT") \
.filter((F.year('order_date') == 2013) & (F.month('order_date') == 7))
```

```
result_orders.write.mode('Overwrite') \
.option("compression", "snappy") \
.json('C:/Users/DevAdmin/MohyWorkSpace/Retail_Proj/result/scenario7/solution')
```