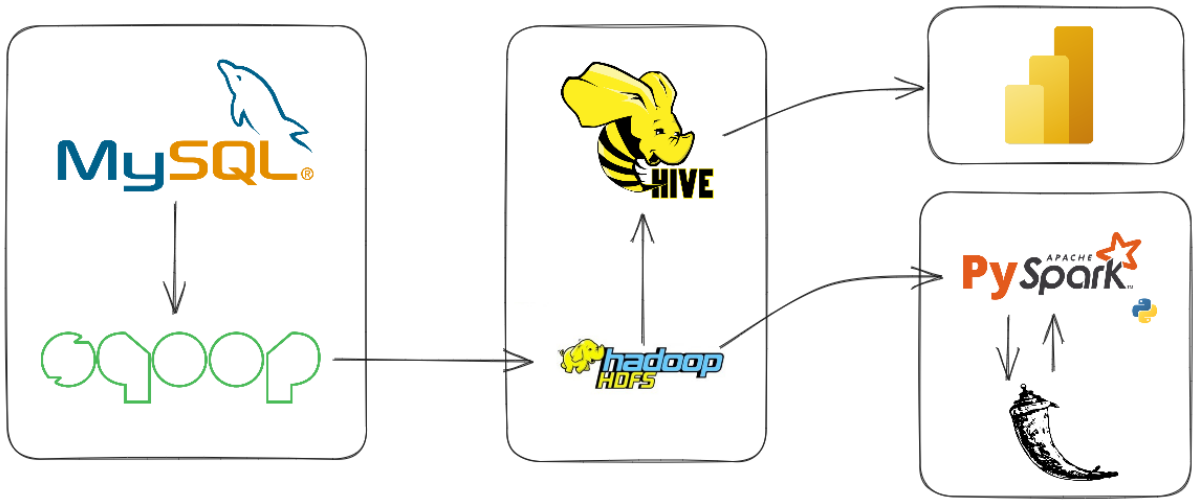
Retail-Store-BigData

Docker Solution : Incomplete

Project Architecture



Part l : Data Migration & Data Analysis

Importing a Table from MySQL to HDFS:

- Run the “retail_db.sql” script on MySQL to create the database.
- Use Sqoop to import the tables in the retail store database and save it in HDFS under “/user”
- Import the tables to a Parquet data format rather than the default file form (text file).

SQL Script for creating tables and inserting data :

https://drive.google.com/file/d/10VNmnnO4RMMZn4EgbDJyk8pAMLLGHt7l/view

dowloand the file from google drive through terminal

```
wget --no-check-certificate 'https://docs.google.com/uc?export=download&id=10VNmnnO4RMMZn4EgbDJyk8pAMLLGHt7l' -O retail.sql
```

Upload data to mysql

Connect to my SQL and run the script

```
mysql -u root -p
```

```
create database retaildb;
```

```
[root@sandbox-hdp maria_dev]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 5.7.22 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database retaildb;
Query OK, 1 row affected (0.00 sec)

mysql> show databases
    -> ;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| druid              |
| hive               |
| mysql              |
| performance_schema |
| ranger             |
| retaildb           |
| superset           |
| sys                |
+--------------------+
9 rows in set (0.02 sec)
```

let's do some configuration on mysql

```
SET NAMES 'utf8';
SET CHARACTER SET utf8;
use retaildb;
```

```
mysql> SET NAMES 'utf8';
Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> SET CHARACTER SET utf8;
Query OK, 0 rows affected (0.00 sec)

mysql> use retaildb;
Database changed
mysql>
```

run the script retail.sql

```
source /home/maria_dev/retail.sql
```

```
mysql> source /home/maria_dev/retail.sql
Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected, 1 warning (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

Check if the data is uploaded:

```
mysql> show tables
    -> ;
+--------------------+
| Tables_in_retaildb |
+--------------------+
| categories         |
| customers          |
| departments        |
| order_items        |
| orders             |
| products           |
+--------------------+
6 rows in set (0.00 sec)

mysql> select * from customers limit 10;
+-------------+---------------+---------------+----------------+------------------+-------------------------+---------------+----------------+-----------------+
| customer_id | customer_fname | customer_lname | customer_email | customer_password | customer_street         | customer_city | customer_state | customer_zipcode |
+-------------+---------------+---------------+----------------+------------------+-------------------------+---------------+----------------+-----------------+
|           1 | Richard       | Hernandez     | XXXXXXXXX      | XXXXXXXXX        | 6303 Heather Plaza      | Brownsville   | TX             | 78521           |
|           2 | Mary          | Barrett       | XXXXXXXXX      | XXXXXXXXX        | 9526 Noble Embers Ridge | Littleton     | CO             | 80126           |
|           3 | Ann           | Smith         | XXXXXXXXX      | XXXXXXXXX        | 3422 Blue Pioneer Bend  | Caguas        | PR             | 00725           |
|           4 | Mary          | Jones         | XXXXXXXXX      | XXXXXXXXX        | 8324 Little Common      | San Marcos    | CA             | 92069           |
|           5 | Robert        | Hudson        | XXXXXXXXX      | XXXXXXXXX        | 10 Crystal River Mall   | Caguas        | PR             | 00725           |
|           6 | Mary          | Smith         | XXXXXXXXX      | XXXXXXXXX        | 3151 Sleepy Quail Promenade | Passaic   | NJ             | 07055           |
|           7 | Melissa       | Wilcox        | XXXXXXXXX      | XXXXXXXXX        | 9453 High Concession    | Caguas        | PR             | 00725           |
|           8 | Megan         | Smith         | XXXXXXXXX      | XXXXXXXXX        | 3047 Foggy Forest Plaza | Lawrence      | MA             | 01841           |
|           9 | Mary          | Perez         | XXXXXXXXX      | XXXXXXXXX        | 3616 Quaking Street     | Caguas        | PR             | 00725           |
|          10 | Melissa       | Smith         | XXXXXXXXX      | XXXXXXXXX        | 8598 Harvest Beacon Plaza | Stafford    | VA             | 22554           |
+-------------+---------------+---------------+----------------+------------------+-------------------------+---------------+----------------+-----------------+
10 rows in set (0.00 sec)
```

```
mysql> describe products
    -> ;
+--------------------+--------------+------+-----+---------+----------------+
| Field              | Type         | Null | Key | Default | Extra          |
+--------------------+--------------+------+-----+---------+----------------+
| product_id         | int(11)      | NO   | PRI | NULL    | auto_increment |
| product_category_id | int(11)     | NO   |     | NULL    |                |
| product_name       | varchar(45)  | NO   |     | NULL    |                |
| product_description | varchar(255) | NO   |     | NULL    |                |
| product_price      | float        | NO   |     | NULL    |                |
| product_image      | varchar(255) | NO   |     | NULL    |                |
+--------------------+--------------+------+-----+---------+----------------+
6 rows in set (0.18 sec)
```

Use Apache sqoop to import data from mysql to hdfs

let's set the appropriate privilges first so that sqoop can access tables in mysql

```
GRANT ALL PRIVILEGES ON retaildb.* to root@localhost identified by 'hadoop';
```

```
mysql> GRANT ALL PRIVILEGES ON retaildb.* to root@localhost identified by 'hadoop';
Query OK, 0 rows affected, 1 warning (0.24 sec)

mysql> exit
Bye
```

```
sqoop import-all-tables --connect jdbc:mysql://localhost/retaildb --driver com.mysql.jdbc.Driver -m 1 --username root --password hadoo
```

```
[root@sandbox-hdp maria_dev]# sqoop import-all-tables --connect jdbc:mysql://localhost/retaildb --driver com.mysql.jdbc.Driver --username root --password hadoop --wareh
ouse-dir /home
Warning: /usr/hdp/2.6.5.0-292/accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
23/08/28 15:13:42 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6.2.6.5.0-292
23/08/28 15:13:42 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
23/08/28 15:13:42 WARN sqoop.ConnFactory: Parameter --driver is set to an explicit driver however appropriate connection manager is not being set (via --connection-mana
ger). Sqoop is going to fall back to org.apache.sqoop.manager.GenericJdbcManager. Please specify explicitly which connection manager should be used next time.
23/08/28 15:13:42 INFO manager.SqlManager: Using default fetchSize of 1000
Mon Aug 28 15:13:42 UTC 2023 WARN: Establishing SSL connection without server's identity verification is not recommended. According to MySQL 5.5.45+, 5.6.26+ and 5.7.6+
 requirements SSL connection must be established by default if explicit option isn't set. For compliance with existing applications not using SSL the verifyServerCertif
icate property is set to 'false'. You need either to explicitly disable SSL by setting useSSL=false, or set useSSL=true and provide truststore for server certificate ve
rification.
23/08/28 15:13:43 INFO tool.CodeGenTool: Beginning code generation
23/08/28 15:13:43 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM categories AS t WHERE 1=0
23/08/28 15:13:43 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM categories AS t WHERE 1=0
23/08/28 15:13:43 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/hdp/2.6.5.0-292/hadoop-mapreduce
Note: /tmp/sqoop-root/compile/a035a3b928a449f27d165397e7d32102/categories.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
23/08/28 15:13:44 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-root/compile/a035a3b928a449f27d165397e7d32102/categories.jar
23/08/28 15:13:44 INFO mapreduce.ImportJobBase: Beginning import of categories
23/08/28 15:13:44 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM categories AS t WHERE 1=0
23/08/28 15:13:45 INFO client.RMProxy: Connecting to ResourceManager at sandbox-hdp.hortonworks.com/172.18.0.2:8032
23/08/28 15:13:45 INFO client.AHSProxy: Connecting to Application History server at sandbox-hdp.hortonworks.com/172.18.0.2:10200
Mon Aug 28 15:13:51 UTC 2023 WARN: Establishing SSL connection without server's identity verification is not recommended. According to MySQL 5.5.45+, 5.6.26+ and 5.7.6+
 requirements SSL connection must be established by default if explicit option isn't set. For compliance with existing applications not using SSL the verifyServerCertif
icate property is set to 'false'. You need either to explicitly disable SSL by setting useSSL=false, or set useSSL=true and provide truststore for server certificate ve
rification.
23/08/28 15:13:51 INFO db.DBInputFormat: Using read commited transaction isolation
23/08/28 15:13:51 INFO db.DataDrivenDBInputFormat: BoundingValsQuery: SELECT MIN(category_id), MAX(category_id) FROM categories
23/08/28 15:13:51 INFO db.IntegerSplitter: Split size: 14; Num splits: 4 from: 1 to: 58
23/08/28 15:13:52 INFO mapreduce.JobSubmitter: number of splits:4
23/08/28 15:13:52 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1693231448681_0007
23/08/28 15:13:53 INFO impl.YarnClientImpl: Submitted application application_1693231448681_0007
23/08/28 15:13:53 INFO mapreduce.Job: The url to track the job: http://sandbox-hdp.hortonworks.com:8088/proxy/application_1693231448681_0007/
```

```
23/08/28 15:16:26 INFO mapreduce.JobSubmitter: number of splits:4
23/08/28 15:16:26 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1693231448681_0012
23/08/28 15:16:27 INFO impl.YarnClientImpl: Submitted application application_1693231448681_0012
23/08/28 15:16:27 INFO mapreduce.Job: The url to track the job: http://sandbox-hdp.hortonworks.com:8088/proxy/application_1693231448681_0012/
23/08/28 15:16:27 INFO mapreduce.Job: Running job: job_1693231448681_0012
23/08/28 15:16:35 INFO mapreduce.Job: Job job_1693231448681_0012 running in uber mode : false
23/08/28 15:16:35 INFO mapreduce.Job:  map 0% reduce 0%
23/08/28 15:16:48 INFO mapreduce.Job:  map 25% reduce 0%
23/08/28 15:16:49 INFO mapreduce.Job:  map 50% reduce 0%
23/08/28 15:16:54 INFO mapreduce.Job:  map 100% reduce 0%
23/08/28 15:16:55 INFO mapreduce.Job: Job job_1693231448681_0012 completed successfully
23/08/28 15:16:55 INFO mapreduce.Job: Counters: 30
        File System Counters
                FILE: Number of bytes read=0
                FILE: Number of bytes written=684540
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=458
                HDFS: Number of bytes written=173993
                HDFS: Number of read operations=16
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=8
        Job Counters
                Launched map tasks=4
                Other local map tasks=4
                Total time spent by all maps in occupied slots (ms)=50228
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=50228
                Total vcore-milliseconds taken by all map tasks=50228
                Total megabyte-milliseconds taken by all map tasks=12557000
        Map-Reduce Framework
                Map input records=1345
                Map output records=1345
                Input split bytes=458
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=1668
                CPU time spent (ms)=10640
                Physical memory (bytes) snapshot=572518400
```

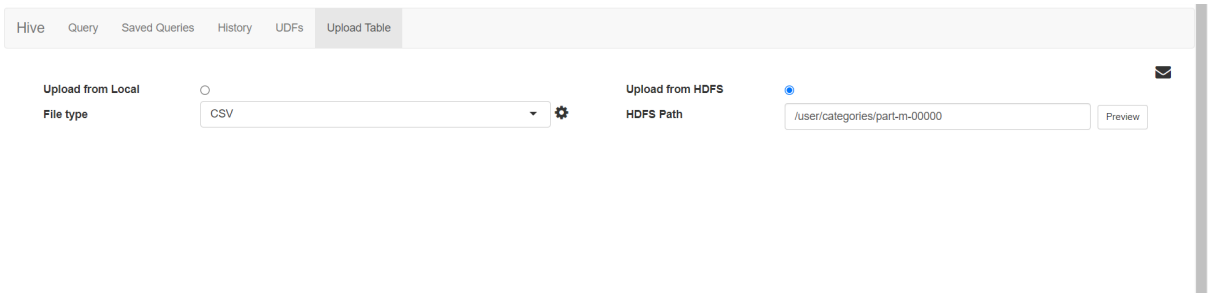
we can make sure that all the files are imported to hdfs using the command line

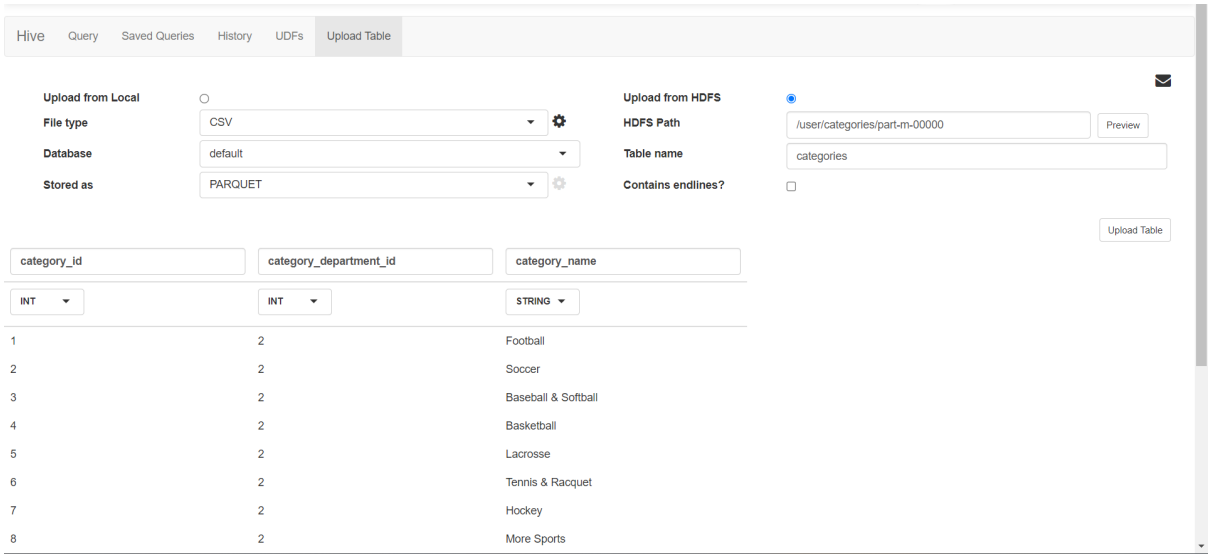
```
hdfs dfs -ls /home | grep '^drwx.* root '
```

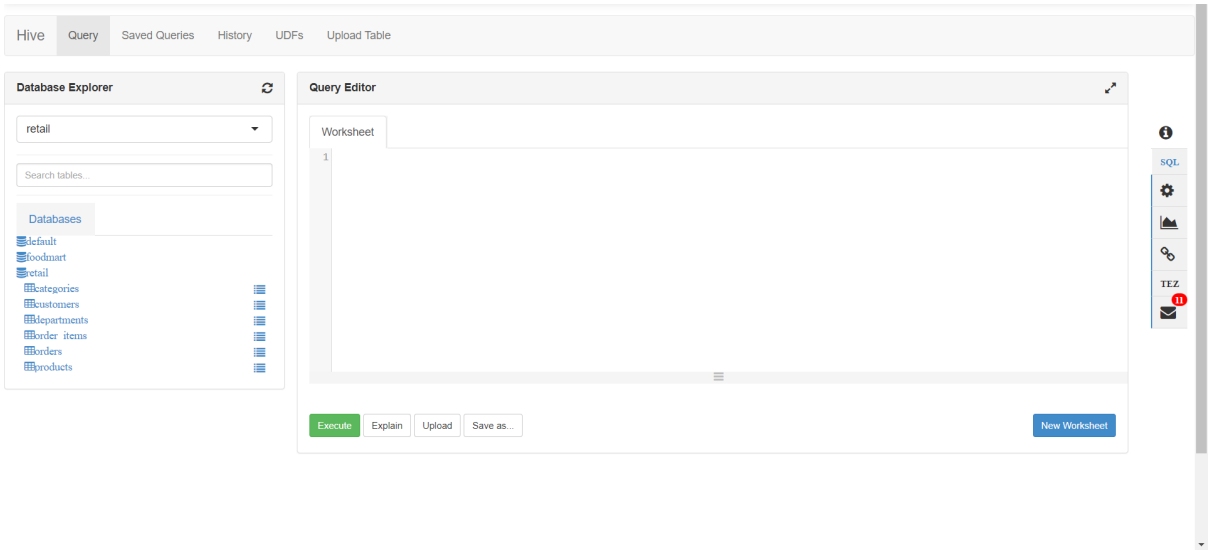
```
[root@sandbox-hdp maria_dev]# hdfs dfs -ls /user | grep '^drwx.* root '
drwxr-xr-x   - root      hdfs          0 2018-06-18 14:52 /user/anonymous
drwxr-xr-x   - root      hdfs          0 2023-08-28 15:21 /user/categories
drwxr-xr-x   - root      hdfs          0 2023-08-28 15:22 /user/customers
drwxr-xr-x   - root      hdfs          0 2023-08-28 15:22 /user/departments
drwxr-xr-x   - root      hdfs          0 2023-08-28 15:23 /user/order_items
drwxr-xr-x   - root      hdfs          0 2023-08-28 15:23 /user/orders
drwxr-xr-x   - root      hdfs          0 2023-08-28 15:24 /user/products
drwxr-xr-x   - root      hdfs          0 2023-08-28 15:21 /user/root
[root@sandbox-hdp maria_dev]#
```

Now let’s do some analysis on the data using Hive through Apache Ambari

First we gotta import all the tables from hdfs





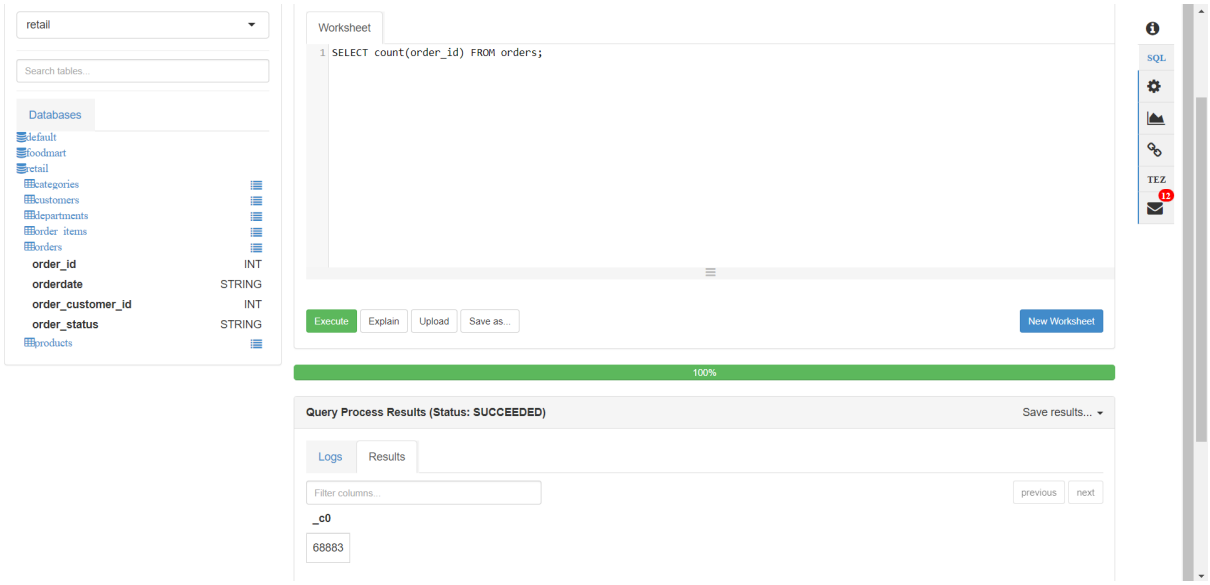


Data Analysis:

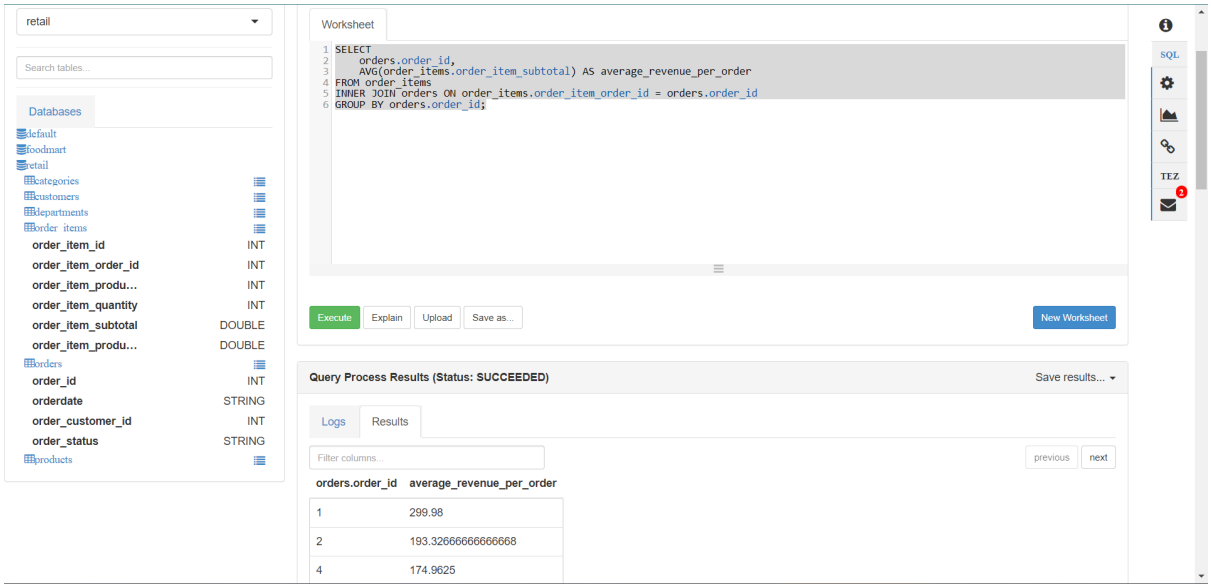
HiveQL is Hive's query language, a dialect of SQL for big data. By using HiveQL, write a query to:

- Get How many Orders were placed
- Get Average Revenue Per Order
- Get Average Revenue Per Day Per Product

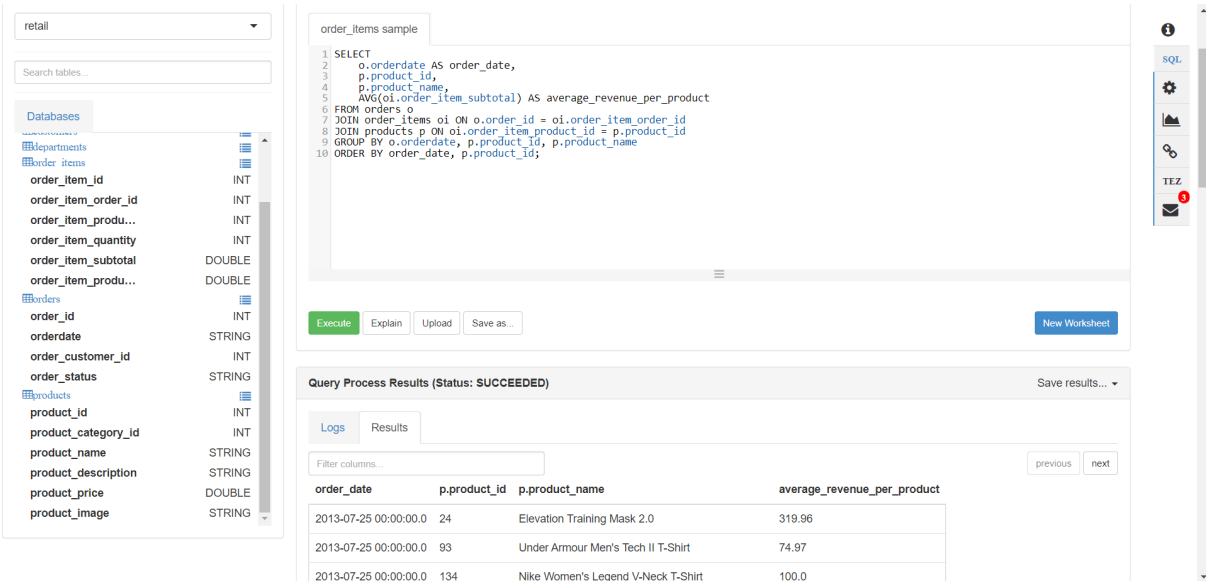
```
--Get How many Orders were placed
SELECT count(order_id) FROM orders;
```




```
--Get Average Revenue Per Order
SELECT
    orders.order_id,
    AVG(order_items.order_item_subtotal) AS average_revenue_per_order
FROM order_items
INNER JOIN orders ON order_items.order_item_order_id = orders.order_id
GROUP BY orders.order_id;
```




```
--Get Average Revenue Per Day Per Product
SELECT
    o.orderdate AS order_date,
    p.product_id,
    p.product_name,
    AVG(oi.order_item_subtotal) AS average_revenue_per_product
FROM orders o
JOIN order_items oi ON o.order_id = oi.order_item_order_id
JOIN products p ON oi.order_item_product_id = p.product_id
GROUP BY o.orderdate, p.product_id, p.product_name
ORDER BY order_date, p.product_id;
```



Part II : Spark SQL and PySpark

Creating a DataFrame from a Table:

- Create a dataframe to load the retail store data form Hdfs to Spark
- Examine the schema of the new DataFrame

import_data_schema.py

```
from pyspark.sql import SparkSession

# Create a Spark session

spark = SparkSession.builder.appName("RetailStoreData").getOrCreate()

# Define the HDFS base path
hdfs_path_starting = "hdfs://sandbox-hdp.hortonworks.com:8020//user/"
hdfs_path_finishing = "/part-m-00000"

# Read data from HDFS into DataFrames
customers_df = spark.read.format("csv").option("header", "false").load(hdfs_path_starting + "customers" + hdfs_path_finishing )
departments_df = spark.read.format("csv").option("header", "false").load(hdfs_path_starting + "departments" + hdfs_path_finishing)
categories_df = spark.read.format("csv").option("header", "false").load(hdfs_path_starting + "categories" + hdfs_path_finishing)
orders_df = spark.read.format("csv").option("header", "false").load(hdfs_path_starting + "orders" + hdfs_path_finishing)
order_items_df = spark.read.format("csv").option("header", "false").load(hdfs_path_starting + "order_items" + hdfs_path_finishing)
products_df = spark.read.format("csv").option("header", "false").load(hdfs_path_starting + "products" + hdfs_path_finishing)


# schemas examination
customers_df.printSchema()
departments_df.printSchema()
categories_df.printSchema()
orders_df.printSchema()
order_items_df.printSchema()
products_df.printSchema()

# You can perform further operations, such as joins, aggregations, etc., on these DataFrames as needed.
```

```
spark-submit import_data_schema.py
```

```
root
 |-- _c0: string (nullable = true)
 |-- _c1: string (nullable = true)
 |-- _c2: string (nullable = true)
 |-- _c3: string (nullable = true)
 |-- _c4: string (nullable = true)
 |-- _c5: string (nullable = true)
 |-- _c6: string (nullable = true)
 |-- _c7: string (nullable = true)
 |-- _c8: string (nullable = true)

root
 |-- _c0: string (nullable = true)
 |-- _c1: string (nullable = true)

root
 |-- _c0: string (nullable = true)
 |-- _c1: string (nullable = true)
 |-- _c2: string (nullable = true)

root
 |-- _c0: string (nullable = true)
 |-- _c1: string (nullable = true)
 |-- _c2: string (nullable = true)
 |-- _c3: string (nullable = true)

root
 |-- _c0: string (nullable = true)
 |-- _c1: string (nullable = true)
 |-- _c2: string (nullable = true)
 |-- _c3: string (nullable = true)
 |-- _c4: string (nullable = true)
 |-- _c5: string (nullable = true)

root
 |-- _c0: string (nullable = true)
 |-- _c1: string (nullable = true)
 |-- _c2: string (nullable = true)
 |-- _c3: string (nullable = true)
 |-- _c4: string (nullable = true)
 |-- _c5: string (nullable = true)
```

SparkSQL Data Analysis:

By using the SparkSQL with pySpark, write a script to:

- Get How many Orders were placed
- Get Average Revenue Per Order
- Get Average Revenue Per Day Per Product

```
# Get How many Orders were placed

spark_OrderPlaced = SparkSession.builder.appName("OrderPlaced").getOrCreate()
orders_df.createOrReplaceTempView("orders")


orders_df.createOrReplaceTempView("orders")
query = """
    SELECT COUNT(orders._c0) AS Number_of_Orders_Placed FROM orders
"""
result = spark.sql(query)

result.show()

# Stop the Spark session
spark.stop()
```

```
23/08/29 14:40:30 INFO DAGScheduler: Submitting 1 missing tasks from ResultStage 7 (MapPartitionsRDD[60] at showString at NativeMethodAccessorImpl.java:0) (fir
ks are for partitions Vector(0))
23/08/29 14:40:30 INFO TaskSchedulerImpl: Adding task set 7.0 with 1 tasks
23/08/29 14:40:30 INFO TaskSetManager: Starting task 0.0 in stage 7.0 (TID 7, localhost, executor driver, partition 0, ANY, 7754 bytes)
23/08/29 14:40:30 INFO Executor: Running task 0.0 in stage 7.0 (TID 7)
23/08/29 14:40:30 INFO ShuffleBlockFetcherIterator: Getting 1 non-empty blocks out of 1 blocks
23/08/29 14:40:30 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 38 ms
23/08/29 14:40:30 INFO Executor: Finished task 0.0 in stage 7.0 (TID 7). 1609 bytes result sent to driver
23/08/29 14:40:30 INFO TaskSetManager: Finished task 0.0 in stage 7.0 (TID 7) in 292 ms on localhost (executor driver) (1/1)
23/08/29 14:40:30 INFO TaskSchedulerImpl: Removed TaskSet 7.0, whose tasks have all completed, from pool
23/08/29 14:40:30 INFO DAGScheduler: ResultStage 7 (showString at NativeMethodAccessorImpl.java:0) finished in 0.307 s
23/08/29 14:40:30 INFO DAGScheduler: Job 6 finished: showString at NativeMethodAccessorImpl.java:0, took 1.619812 s
+----------------------+
|Number_of_Orders_Placed|
+----------------------+
|                 68883|
+----------------------+

23/08/29 14:40:30 INFO AbstractConnector: Stopped Spark@21009ab6{HTTP/1.1,[http/1.1]}{0.0.0.0:4040}
23/08/29 14:40:30 INFO SparkUI: Stopped Spark web UI at http://sandbox-hdp.hortonworks.com:4040
23/08/29 14:40:31 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
23/08/29 14:40:31 INFO MemoryStore: MemoryStore cleared
23/08/29 14:40:31 INFO BlockManager: BlockManager stopped
23/08/29 14:40:31 INFO BlockManagerMaster: BlockManagerMaster stopped
23/08/29 14:40:31 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
23/08/29 14:40:31 INFO SparkContext: Successfully stopped SparkContext
23/08/29 14:40:32 INFO ShutdownHookManager: Shutdown hook called
23/08/29 14:40:32 INFO ShutdownHookManager: Deleting directory /tmp/spark-edbe8d9d-66ce-471e-9eab-46229f12e758
23/08/29 14:40:32 INFO ShutdownHookManager: Deleting directory /tmp/spark-40962db2-fe5e-45ab-86ab-b437393dfcd9/pyspark-55964807-2154-45f9-aa82-a13e5ef1b0bd
23/08/29 14:40:32 INFO ShutdownHookManager: Deleting directory /tmp/spark-40962db2-fe5e-45ab-86ab-b437393dfcd9
[root@sandbox-hdp home]#
```

```
#Get Average Revenue Per Order

spark_AverageRevenuePerOrder = SparkSession.builder.appName("AverageRevenuePerOrder").getOrCreate()

orders_df.createOrReplaceTempView("orders")
order_items_df.createOrReplaceTempView("order_items")

query = """
    SELECT orders._c0 AS order_id, AVG(order_items._c4) AS average_revenue_per_order
    FROM order_items INNER JOIN orders ON order_items._c1 = orders._c0
    GROUP BY orders._c0
"""
result = spark_AverageRevenuePerOrder.sql(query)

result.show()

# Stop the Spark session
spark_AverageRevenuePerOrder.stop()
```

```
+--------+------------------------+
|order_id|average_revenue_per_order|
+--------+------------------------+
|     296|                   69.98|
|     467|       151.96800000000002|
|     675|                 264.985|
|     691|       255.97400000000002|
|    1090|                 233.958|
|    1159|                  62.475|
|    1512|       162.47250000000003|
|    2088|                  286.63|
|    2136|        279.9766666666667|
|    2162|       187.47750000000002|
|    2294|                299.9575|
|    2904|                  299.98|
|    3210|                 114.975|
|    3414|                237.4925|
|    3606|                  149.98|
|    3959|       122.47999999999999|
|    4032|                  324.99|
|    4821|                  299.98|
|    4937|       174.98000000000002|
|    5325|       166.65333333333334|
+--------+------------------------+
only showing top 20 rows
```

```
# Get Average Revenue Per Day Per Product

products_df.createOrReplaceTempView("products")

query = """
    SELECT
    o._c1 AS order_date,
    p._c0 AS Product_id,
    p._c2 AS Product_name,
    AVG(oi._c4) AS average_revenue_per_product
    FROM orders o
    JOIN order_items oi ON o._c0 = oi._c1
    JOIN products p ON oi._c2 = p._c0
    GROUP BY o._c1, p._c0, p._c2
    ORDER BY o._c1, p._c0
"""
result = spark.sql(query)

result.show()
```

```
spark.stop()
```

```
+--------------------+----------+--------------------+--------------------------+
|          order_date|Product_id|        Product_name|average_revenue_per_product|
+--------------------+----------+--------------------+--------------------------+
|2013-07-25 00:00:...|      1004|Field & Stream Sp...|          399.9799999999999|
|2013-07-25 00:00:...|      1014|O'Brien Men's Neo...|          164.0369230769231|
|2013-07-25 00:00:...|      1073|Pelican Sunstream...|         199.98999999999992|
|2013-07-25 00:00:...|       134|Nike Women's Lege...|                     100.0|
|2013-07-25 00:00:...|       191|Nike Men's Free 5...|          326.8903846153846|
|2013-07-25 00:00:...|       226|Bowflex SelectTec...|                    599.99|
|2013-07-25 00:00:...|        24|Elevation Trainin...|                    319.96|
|2013-07-25 00:00:...|       276|Under Armour Wome...|                    127.96|
|2013-07-25 00:00:...|       365|Perfect Fitness P...|          175.7846511627906|
|2013-07-25 00:00:...|        37|adidas Kids' F5 M...|                     69.98|
|2013-07-25 00:00:...|       403|Nike Men's CJ Eli...|         129.98999999999992|
|2013-07-25 00:00:...|       502|Nike Men's Dri-FI...|         130.76923076923077|
|2013-07-25 00:00:...|       572|TYR Boys' Team Di...|                    119.97|
|2013-07-25 00:00:...|       625|Nike Men's Kobe I...|                    199.99|
|2013-07-25 00:00:...|       627|Under Armour Girl...|                   143.964|
|2013-07-25 00:00:...|       666|Merrell Men's All...|                    109.99|
|2013-07-25 00:00:...|       691|MDGolf Pittsburgh...|                    239.97|
|2013-07-25 00:00:...|       705|Cleveland Golf Wo...|                    119.99|
|2013-07-25 00:00:...|       725|LIJA Women's Butt...|                     108.0|
|2013-07-25 00:00:...|       775|Clicgear 8.0 Shoe...|                     19.98|
+--------------------+----------+--------------------+--------------------------+
only showing top 20 rows

23/08/29 15:08:31 INFO AbstractConnector: Stopped Spark@6f395ae0{HTTP/1.1,[http/1.1]}{0.0.0.0:4040}
23/08/29 15:08:31 INFO SparkUI: Stopped Spark web UI at http://sandbox-hdp.hortonworks.com:4040
23/08/29 15:08:31 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
23/08/29 15:08:32 INFO MemoryStore: MemoryStore cleared
23/08/29 15:08:32 INFO BlockManager: BlockManager stopped
23/08/29 15:08:32 INFO BlockManagerMaster: BlockManagerMaster stopped
23/08/29 15:08:32 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
23/08/29 15:08:32 INFO SparkContext: Successfully stopped SparkContext
23/08/29 15:08:33 INFO ShutdownHookManager: Shutdown hook called
23/08/29 15:08:33 INFO ShutdownHookManager: Deleting directory /tmp/spark-7dcd00e6-53c5-4cfd-b743-3627bde9e40c/pyspark-8079013b-c644-460f-acdf-340e4b
23/08/29 15:08:33 INFO ShutdownHookManager: Deleting directory /tmp/spark-7dcd00e6-53c5-4cfd-b743-3627bde9e40c
```

Mlib Analysis and prediction:

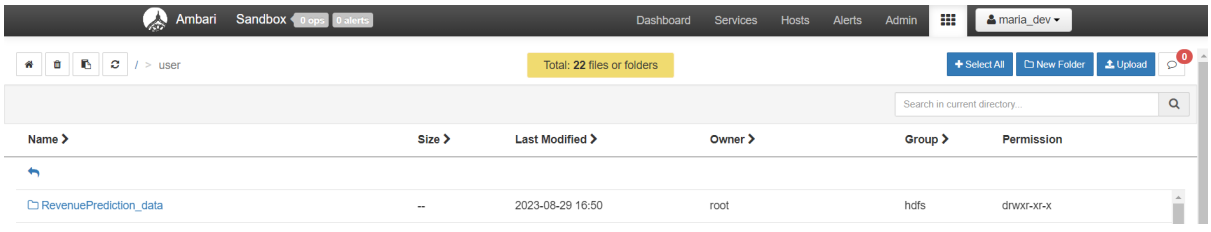
By using Pyspark and the Mllib package, write a script to:

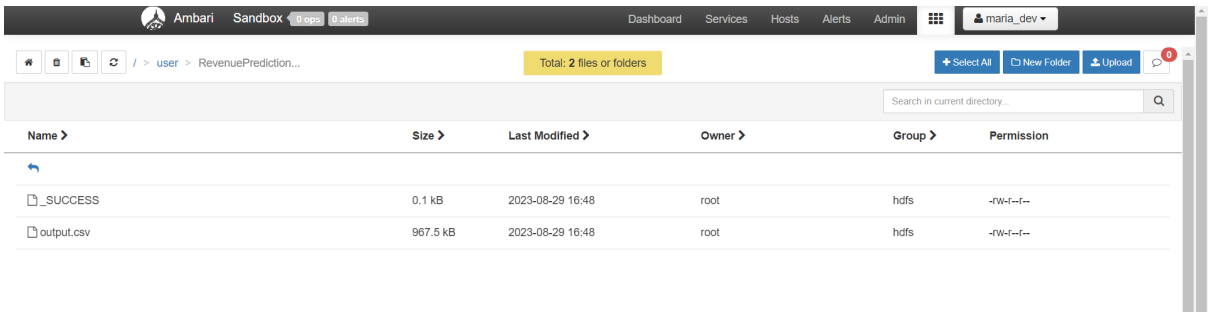
- Extract the list of product, day and average revenue per Day
- Save the result to Hdfs
- Write a second Mllib script to predict revenue for a given product and a day

```
from pyspark.sql import SparkSession

# Create a Spark session

spark = SparkSession.builder.appName("RevenuePrediction").getOrCreate()

# Define the HDFS base path
hdfs_path_starting = "hdfs://sandbox-hdp.hortonworks.com:8020//user/"
hdfs_path_finishing = "/part-m-00000"

# Read data from HDFS into DataFrames
customers_df = spark.read.format("csv").option("header", "false").load(hdfs_path_starting + "customers" + hdfs_path_finishing )
departments_df = spark.read.format("csv").option("header", "false").load(hdfs_path_starting + "departments" + hdfs_path_finishing)
categories_df = spark.read.format("csv").option("header", "false").load(hdfs_path_starting + "categories" + hdfs_path_finishing)
orders_df = spark.read.format("csv").option("header", "false").load(hdfs_path_starting + "orders" + hdfs_path_finishing)
order_items_df = spark.read.format("csv").option("header", "false").load(hdfs_path_starting + "order_items" + hdfs_path_finishing)
products_df = spark.read.format("csv").option("header", "false").load(hdfs_path_starting + "products" + hdfs_path_finishing)

orders_df.createOrReplaceTempView("orders")
order_items_df.createOrReplaceTempView("order_items")
products_df.createOrReplaceTempView("products")

query = """
    SELECT
          o._c1 as order_date,
        p._c2 as product_name,
        AVG(oi._c4) AS avg_revenue_per_day
    FROM
        orders o
    JOIN
        order_items oi ON o._c0 = oi._c1
    JOIN
        products p ON oi._c2 = p._c0
    GROUP BY
        o._c1, p._c2
"""

result = spark.sql(query)

result.show()
```

```
column_names = ["order_date", "product_name", "avg_revenue_per_day"]

result = result.toDF(*column_names)

# Specify the full HDFS path including the desired file name
output_path = "hdfs://sandbox-hdp.hortonworks.com:8020//user/RevenuePrediction_data/output.csv"

# Save the result to HDFS with the specified file name and column names
result.coalesce(1).write.option("header", "true").option("delimiter", ",").mode("overwrite").csv(output_path)

spark.stop()
```

```
23/08/29 15:47:16 INFO TaskSchedulerImpl: Removed TaskSet 9.0, whose tasks have all completed, from pool
23/08/29 15:47:16 INFO DAGScheduler: ResultStage 9 (showString at NativeMethodAccessorImpl.java:0) finished in 0.080 s
23/08/29 15:47:16 INFO DAGScheduler: Job 8 finished: showString at NativeMethodAccessorImpl.java:0, took 1.183919 s
+-------------------+--------------------+------------------+
|         order_date|        product_name|avg_revenue_per_day|
+-------------------+--------------------+------------------+
|2013-07-31 00:00:...|Diamondback Women...| 299.97999999999973|
|2013-08-02 00:00:...|Under Armour Kids...|             74.64|
|2013-08-03 00:00:...|O'Brien Men's Neo...| 147.05653846153837|
|2013-08-06 00:00:...|Nike Men's Deutsc...|              60.0|
|2013-08-07 00:00:...|Team Golf San Fra...|             49.98|
|2013-08-18 00:00:...|ENO Atlas Hammock...|            74.975|
|2013-08-18 00:00:...|Bridgestone e6 St...|             95.97|
|2013-08-23 00:00:...|Hirzl Women's Sof...| 62.964999999999996|
|2013-09-26 00:00:...|Hirzl Women's Sof...|             17.99|
|2013-09-30 00:00:...|Titleist Pro V1x ...|            191.96|
|2013-10-04 00:00:...|Columbia Men's PF...|              90.0|
|2013-10-14 00:00:...|Nike Men's Comfor...|            134.97|
|2013-10-20 00:00:...|Glove It Imperial...|             79.95|
|2013-10-21 00:00:...|Nike Dri-FIT Crew...|              33.0|
|2013-10-27 00:00:...|Titleist Pro V1 H...|           129.975|
|2013-10-31 00:00:...|Hirzl Men's Hybri...|             59.96|
|2013-11-02 00:00:...|Bridgestone e6 St...|            159.95|
|2013-11-04 00:00:...|Nike Men's Dri-FI...| 130.13698630136986|
|2013-11-05 00:00:...|Glove It Women's ...|             59.97|
|2013-11-06 00:00:...|Diamondback Women...| 299.97999999999973|
+-------------------+--------------------+------------------+
only showing top 20 rows

23/08/29 15:47:16 INFO AbstractConnector: Stopped Spark@458d1485{HTTP/1.1,[http/1.1]}{0.0.0.0:4040}
23/08/29 15:47:16 INFO SparkUI: Stopped Spark web UI at http://sandbox-hdp.hortonworks.com:4040
23/08/29 15:47:16 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
23/08/29 15:47:16 INFO MemoryStore: MemoryStore cleared
23/08/29 15:47:16 INFO BlockManager: BlockManager stopped
23/08/29 15:47:16 INFO BlockManagerMaster: BlockManagerMaster stopped
23/08/29 15:47:16 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
23/08/29 15:47:16 INFO SparkContext: Successfully stopped SparkContext
```






```
from pyspark.sql import SparkSession
from pyspark.ml.regression import LinearRegression
from pyspark.ml.feature import StringIndexer, OneHotEncoder
from pyspark.ml.feature import VectorAssembler
from pyspark.ml import Pipeline
from pyspark.ml.evaluation import RegressionEvaluator
from pyspark.sql.functions import dayofweek

spark = SparkSession.builder.appName("RevenuePrediction").getOrCreate()
training_data_path = "hdfs://sandbox-hdp.hortonworks.com:8020//user/RevenuePrediction_data/output.csv"
data = spark.read.csv(training_data_path , header=True, inferSchema=True)


# Convert order_date to day of the week (1-7, where 1=Sunday, 7=Saturday)
data = data.withColumn("day_of_week", dayofweek(data["order_date"]))

# Assuming 'product_name' is a categorical variable
```

```
product_indexer = StringIndexer(inputCol="product_name", outputCol="product_index")
product_encoder = OneHotEncoder(inputCol="product_index", outputCol="product_vector")

# Prepare features vector
feature_cols = ["product_vector", "day_of_week"]
feature_assembler = VectorAssembler(inputCols=feature_cols, outputCol="features")

# Split data into training and testing sets
train_data, test_data = data.randomSplit([0.7, 0.3], seed=42)

lr = LinearRegression(featuresCol="features", labelCol="avg_revenue_per_day")
pipeline = Pipeline(stages=[product_indexer, product_encoder, feature_assembler, lr])
model = pipeline.fit(train_data)
predictions = model.transform(test_data)

evaluator = RegressionEvaluator(labelCol="avg_revenue_per_day", predictionCol="prediction", metricName="mse")
mse = evaluator.evaluate(predictions)
print("Mean Squared Error: {}".format(mse))

spark.stop()
```

```
+-----------+
|[empty row]|
+-----------+
23/08/29 17:49:38 INFO Executor: Finished task 0.0 in stage 10.0 (TID 10). 2300 bytes result sent to driver
23/08/29 17:49:38 INFO TaskSetManager: Finished task 0.0 in stage 10.0 (TID 10) in 189 ms on localhost (executor driver) (1/1)
23/08/29 17:49:38 INFO TaskSchedulerImpl: Removed TaskSet 10.0, whose tasks have all completed, from pool
23/08/29 17:49:38 INFO DAGScheduler: ResultStage 10 (treeAggregate at RegressionMetrics.scala:57) finished in 0.208 s
23/08/29 17:49:38 INFO DAGScheduler: Job 8 finished: treeAggregate at RegressionMetrics.scala:57, took 0.210926 s
Mean Squared Error: 2042.89897802
23/08/29 17:49:38 INFO ContextCleaner: Cleaned accumulator 262
23/08/29 17:49:38 INFO ContextCleaner: Cleaned accumulator 291
23/08/29 17:49:38 INFO ContextCleaner: Cleaned accumulator 269
23/08/29 17:49:38 INFO ContextCleaner: Cleaned accumulator 275
23/08/29 17:49:38 INFO ContextCleaner: Cleaned accumulator 304
23/08/29 17:49:38 INFO ContextCleaner: Cleaned accumulator 160
23/08/29 17:49:38 INFO ContextCleaner: Cleaned accumulator 248
23/08/29 17:49:38 INFO ContextCleaner: Cleaned accumulator 252
23/08/29 17:49:38 INFO ContextCleaner: Cleaned accumulator 302
23/08/29 17:49:38 INFO ContextCleaner: Cleaned accumulator 171
23/08/29 17:49:38 INFO ContextCleaner: Cleaned accumulator 292
23/08/29 17:49:38 INFO ContextCleaner: Cleaned accumulator 172
23/08/29 17:49:38 INFO AbstractConnector: Stopped Spark@458d1485{HTTP/1.1,[http/1.1]}{0.0.0.0:4040}
23/08/29 17:49:38 INFO BlockManagerInfo: Removed broadcast_10_piece0 on sandbox-hdp.hortonworks.com:37975 in memory (size: 33.3 KB, free: 366.1 MB)
23/08/29 17:49:38 INFO SparkUI: Stopped Spark web UI at http://sandbox-hdp.hortonworks.com:4040
23/08/29 17:49:38 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
23/08/29 17:49:38 INFO MemoryStore: MemoryStore cleared
23/08/29 17:49:38 INFO BlockManager: BlockManager stopped
23/08/29 17:49:38 INFO BlockManagerMaster: BlockManagerMaster stopped
23/08/29 17:49:38 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
23/08/29 17:49:38 INFO SparkContext: Successfully stopped SparkContext
23/08/29 17:49:38 INFO ShutdownHookManager: Shutdown hook called
23/08/29 17:49:38 INFO ShutdownHookManager: Deleting directory /tmp/spark-0e7277d1-83eb-4464-b0d3-0d3cee677040/pyspark-20c0b73a-2541-464e-9c18-f8b1cae64b2d
23/08/29 17:49:38 INFO ShutdownHookManager: Deleting directory /tmp/spark-e2df14f8-49f5-424b-b818-9b92947ec9f6
23/08/29 17:49:38 INFO ShutdownHookManager: Deleting directory /tmp/spark-0e7277d1-83eb-4464-b0d3-0d3cee677040
[root@sandbox-hdp home]#
```