# HDFS to Hive in PySpark Example

## Step 1: Load the Data into HDFS

Exec into the master container using:

docker-compose exec master bash

The load grades.csv from the /data folder into HDFS using:

hdfs dfs -put /data/grades.csv /

## Step 2: Use Pyspark to Read the HDFS Data, create a Hive Table, and load the Data

from pyspark.sql.functions import col

# Step 1: Create a Spark session with Hive support

spark = SparkSession.builder \

.appName("Grades CSV to Hive") \

.enableHiveSupport() \

.getOrCreate()

# Step 2: Read the CSV file from HDFS

grades\_df = spark.read.option("header", True) \

.csv("hdfs:///grades.csv") # Adjust the path as necessary

# Step 3: Rename columns to valid Hive column names (no spaces, special characters)

grades\_df = grades\_df \

.withColumnRenamed("Last name", "last\_name") \

.withColumnRenamed("First name", "first\_name") \

.withColumnRenamed("SSN", "ssn") \

.withColumnRenamed("Test1", "test1") \

.withColumnRenamed("Test2", "test2") \

.withColumnRenamed("Test3", "test3") \

.withColumnRenamed("Test4", "test4") \

.withColumnRenamed("Final", "final\_score") \

.withColumnRenamed("Grade", "grade")

# Step 4: Write the DataFrame to a new Hive table

grades\_df.write.mode("overwrite").saveAsTable("default.newgrades")

# Step 5: Verify the data loaded into the Hive table

spark.sql("SELECT \* FROM default.newgrades").show()

# Step 6: Stop the Spark session

spark.stop()