## ! STOP!

## You are required to watch the walkthrough video for this week's assignment. In the video, I guide you through each command, showing both the input and the expected output. Since you are working with new technology for the first time, it’s important not to just run a command and assume it worked. You will lose points if the commands are not executed correctly.

## Week 5 Assignment: Mastering SparkSQL with Scala, Python

### Objective: Gain experience in querying datasets using SparkSQL across multiple languages - Scala, Python, and R.

In this assignment, you will explore **SparkSQL**, a powerful module in Apache Spark for querying structured data using SQL. SparkSQL allows you to use familiar SQL syntax to query datasets stored in Spark, making it easier to integrate with other data processing systems and languages. You will gain hands-on experience querying data in Spark across multiple programming environments, including **Scala**, **Python (PySpark)**, and **R**.

By the end of this assignment, you will:

* Understand how to load and query datasets using **SparkSQL**.
* Learn how to write and execute SQL queries within **Scala** and **PySpark** environments.
* Practice applying SQL queries to custom datasets and experiment with different query techniques.
* Explore how SparkSQL integrates with the Spark framework to enable distributed query execution across clusters.

#### **1. Environment Initialization**

* Navigate to the required directory and start your Docker containers:
* cd dsc650-infra/bellevue-bigdata/hadoop-hive-spark-hbase
* docker-compose up -d
* Access the master container:
* docker-compose exec master bash
* Load the grades.csv into HDFS:

hdfs dfs -mkdir /data

* hdfs dfs -put /data/grades.csv /data/grades.csv

#### **2. SparkSQL with Scala**

You will now enter the **Spark Shell** to run SQL queries using Scala. By loading the dataset into Spark, you’ll be able to create a **temporary view** and run SQL queries against it. This will help you explore the dataset and extract meaningful insights using standard SQL syntax in a distributed environment.

* Enter the Spark shell:
* spark-shell
* Run the following SparkSQL commands in Scala:
* val df = spark.read.format("csv").option("header", "true").load("/data/grades.csv")  
  df.createOrReplaceTempView("df")  
    
  spark.sql("SHOW TABLES").show()  
  spark.sql("SELECT \* FROM df WHERE Final > 50").show()  
  spark.sql("SELECT \* FROM df").show()
* Run 3 other SQL queries in the Spark Shell:
* Exit the Spark shell:

:quit

**Deliverable:**

* Screenshot of the results obtained from the SparkSQL commands in Scala.
* Screenshot of your 3 other SQL query results.

#### **3. SparkSQL with Python (PySpark)**

Next, you will use **PySpark** to perform the same tasks using Python. PySpark provides an interface to SparkSQL through Python, enabling you to query structured data in the same way as with Scala but using a language that might be more familiar.

* Enter the PySpark environment:  
    
  pyspark
* Run the following SparkSQL commands in Python:
* df = spark.read.format('csv').option('header', 'true').load('/data/grades.csv')  
  df.show()  
    
  df.createOrReplaceTempView('df')  
  spark.sql('SHOW TABLES').show()  
  spark.sql('SELECT \* FROM df WHERE Final > 50').show()  
  spark.sql('SELECT \* FROM df).show()
* Run 3 other SQL queries in the PySpark Shell:
* Exit the Spark shell:

exit()

**Deliverable:**

* Screenshot of the results obtained from the SparkSQL commands in Python.
* Screenshot of your 3 other SQL query results

#### **3. SparkSQL with Custom Data Set**

In this section, you will apply what you’ve learned to a custom dataset that you’ve previously worked with (from Assignment 3). This gives you the opportunity to work with real-world data, apply SQL queries, and extract insights.

1. **Data Loading into Spark**: Use Spark to load the dataset from Assignment 3, or you may choose a different dataset if preferred. Be sure to download your dataset and upload it to HDFS, following the same process you used in Assignment 3. You might find methods like spark.read.csv or spark.read.text useful, depending on the dataset format.
2. **SQL Queries**: Once you’ve loaded the data into Spark, please run three SQL queries on this dataset. Remember to first create a temporary view of your data in Spark using createOrReplaceTempView (for Scala) or a similar method in PySpark, so you can query it using SparkSQL.
3. **Language Selection**: You have the flexibility to use either Scala or PySpark for this exercise. Please choose whichever you’re more comfortable with.

**Deliverable:**

* Your Scala or PySpark Code.
* Screenshot of your 3 SQL query results.

## Shutting Down

Ensure all Docker containers are turned off with docker-compose down for each directory. If you’re using google cloud, please shut down your virtual machine to preserve cloud costs.