**Required Viewing**

Before starting this assignment, you must watch two videos:

1. **Week 5 Fundamentals Lecture Video**
   * Explains Spark SQL architecture and shows how to work with it using both Scala and Python.
   * Provides the background needed to understand what you are learning.
   * Link: <https://youtu.be/3xPLAfZTLZo>
2. **Week 5 Assignment Walkthrough Video**
   * Demonstrates step by step how to complete the tasks, including commands and expected outputs.
   * It is not enough to simply run commands. You must verify that your commands execute correctly. Incorrect or incomplete results will lose points.

Watching both videos is mandatory. The **fundamentals video** explains what you are learning and why it matters, while the **walkthrough video** shows how to complete the assignment.

**Submission Guidelines**

* Submit your work as a **single Word or PDF document** (no raw screenshots or multiple files).
* Include the following in your submission:
  + Screenshots of each required step.
  + A short explanation for each screenshot:
    - The command/action you ran.
    - What the output shows.
    - Whether the result matched your expectation.
* Organize your work in the **same order as the assignment guide** so it is easy to follow.
* This is a **master’s level course** – professionalism and clarity are expected. Well-structured submissions demonstrate your ability to communicate technical work effectively.

**Week 5 Assignment – Objectives and Points**

* **Objective 1 – Conceptual Foundations (SparkSQL Overview)**: 8 pts
* **Objective 2 – SparkSQL with Scala**: 20 pts
* **Objective 3 – SparkSQL with Python (PySpark)**: 20 pts
* **Objective 4 – SparkSQL with a Custom Dataset**: 40 pts

**Total: 88 points**

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

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** and **Python (PySpark).**

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.

# Objective 1 - Conceptual Foundations (8 points)

Before beginning the assignment, watch the instructor-led fundamentals video, which introduces and explains the key concepts for this week:  
<https://youtu.be/3xPLAfZTLZo>

**Deliverable:** Write a 3–4 paragraph summary in your own words. Your write-up should explain the main ideas, highlight why these concepts are important, and connect them to the SparkSQL tasks in this assignment.

# Objective 2 - SparkSQL with Scala (20 points):

#### **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()
* **Deliverable 1:** Screenshot of the results from the provided SparkSQL commands in Scala, plus 1–2 sentences explaining what the commands did and what the output shows.
* Run 3 other SQL queries in the Spark Shell:

**Deliverable 2:** Screenshots of three additional SQL queries you wrote in Scala, each with its results and a short explanation of what the query does and what the output means.

* Exit the Spark shell:

:quit

# Objective 3 - SparkSQL with Python (20 points):

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()
* **Deliverable 1:** Screenshot of the results from the provided SparkSQL commands in Python, plus 1–2 sentences explaining what the commands did and what the output shows.
* Run 3 other SQL queries in the PySpark Shell

**Deliverable 2:** Screenshots of three additional SQL queries you wrote in PySpark, each with its results and a short explanation of what the query does and what the output means.

* Exit the Spark shell:

exit()

# Objective 4 - SparkSQL with Custom Dataset (40 points):

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 1:** Your Scala or PySpark code used to load and query the custom dataset, plus 2–3 sentences explaining what the code does and how you confirmed it worked.

**Deliverable 2:** Screenshots of three SQL query results on your dataset, each with a short explanation of what the query is doing and what the output shows.

## 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.