DSC 650

Week 5 Assignment

Eyram Kueviakoe

April 9, 2024

Screenshot of the results obtained from the SparkSQL commands in Scala

*spark.sql("SHOW TABLES").show()*

A screenshot of a computer

Description automatically generated

*spark.sql("SELECT \* FROM df WHERE Final > 50").show()*

A screenshot of a computer program

Description automatically generated

*spark.sql("SELECT \* FROM grades").show()*

A screenshot of a computer

Description automatically generated

**Screenshot of your 3 other SQL query results**

**Query 1: Top 5 students with the highest scores in Test 1**

*spark.sql("SELECT `Last name`, `First name`, Test1 FROM df ORDER BY Test1 DESC LIMIT 5").show()*

A black screen with white text

Description automatically generated

**Query 2: Students with highest Final exam score**

*spark.sql("SELECT `Last name`, `First name`, MAX(Final) AS Highest\_Final FROM df GROUP BY `Last name`, `First name` ORDER BY Highest\_Final DESC").show()*

A black screen with white text

Description automatically generated

**Query 3: List of students who scored less than the average final exam score**

*spark.sql("SELECT `Last name`, `First name`, Final FROM df WHERE Final < (SELECT AVG(Final) FROM df)").show()*

A black screen with white text

Description automatically generated

**Screenshot of the results obtained from the SparkSQL commands in Python.**

*spark.sql('SHOW TABLES').show()*

*A screen shot of a computer

Description automatically generated*

*spark.sql('SELECT \* FROM df WHERE Final > 50').show()*

*A screenshot of a computer program

Description automatically generated*

*spark.sql('SELECT \* FROM df').show()*

*A screenshot of a computer

Description automatically generated*

***Run 3 other SQL queries in the PySpark Shell***

***Query 1:* Top 5 students with the highest scores in Test 1**

*spark.sql("SELECT `Last name`, `First name`, Test1 FROM df ORDER BY Test1 DESC LIMIT 5").show()*

*A black screen with white text

Description automatically generated*

***Query 2:* Students with highest Final exam score**

*spark.sql("SELECT `Last name`, `First name`, MAX(Final) AS Highest\_Final FROM df GROUP BY `Last name`, `First name` ORDER BY Highest\_Final DESC").show()*

*A black screen with white text

Description automatically generated*

**Query 3: List of students who scored less than the average final exam score**

*spark.sql("SELECT `Last name`, `First name`, Final FROM df WHERE Final < (SELECT AVG(Final) FROM df)").show()*

*A black screen with white text

Description automatically generated*

**3- SparkSQL with custom data set**

Our dataset from assignment 3 is world\_pop\_data.csv.

Loading data into Spark

*val df = spark.read.format("csv").option("header", "true").load("/data/world\_pop\_data.csv")*

*df.createOrReplaceTempView("df")*

**Query 1: 10 most populated countries in 2023**

*spark.sql("SELECT Country, Continent, Population\_2023 FROM df ORDER BY Population\_2023 DESC LIMIT 10").show()*

A screenshot of a computer

Description automatically generated

**Query 2: What are the top 5 countries with highest density in 2023**

*spark.sql("SELECT Country, Continent, Population\_2023/Area\_km2 as Density FROM df ORDER BY Density DESC LIMIT 5").show()*

A screen shot of a computer

Description automatically generated

**Query 3: Find countries with a population greater than 100 million in 1970:**

*spark.sql("SELECT Country, Population\_1970 FROM df WHERE Population\_1970> 100000000").show()*

A computer screen with text on it

Description automatically generated

**Query 4: Find countries with a population greater than 100 million in 2023:**

*spark.sql("SELECT Country, Population\_2023 FROM df WHERE Population\_2023 > 100000000").show()*

A screenshot of a computer

Description automatically generated

**Running the same queries using PySpark**

**Query 1: 10 most populated countries in 2023**

*spark.sql("SELECT Country, Continent, Population\_2023 FROM df ORDER BY Population\_2023 DESC LIMIT 10").show()*

A screenshot of a computer

Description automatically generated

**Query 2: What are the top 5 countries with highest density in 2023**

*spark.sql("SELECT Country, Continent, Population\_2023/Area\_km2 as Density FROM df ORDER BY Density DESC LIMIT 5").show()*

A screen shot of a computer

Description automatically generated

**Query 3: Find countries with a population greater than 100 million in 1970:**

*spark.sql("SELECT Country, Population\_1970 FROM df WHERE Population\_1970> 100000000").show()*

A black screen with white text

Description automatically generated

**Query 4: Find countries with a population greater than 100 million in 2023:**

*spark.sql("SELECT Country, Population\_2023 FROM df WHERE Population\_2023 > 100000000").show()*

A screenshot of a computer

Description automatically generated