

Задание 4

ХУАН ЦЗИНЬЯНЬ

4.1 Create a new DataFrame that deletes all rows prior to January 1, 1950 in it.

```
4.1.py x
1 from pyspark.sql import SparkSession
2 from pyspark.sql.functions import col
3 import pyspark.sql.functions as F
4
5 # initialization Spark Session
6 spark = SparkSession.builder \
7     .appName("Temperature Data Processing") \
8     .getOrCreate()
9
10 # Reading CSV files
11 df = spark.read .csv( path: "D:\Задание 4\GlobalLandTemperaturesByMajorCity.csv", header=True, inferSchema=True)
12
13 # Show raw data structure
14 df.printSchema()
15
16 # Filter out data after 1950-01-01
17 filtered_df = df.filter(col("dt") >= "1950-01-01")
18
19 # View Results
20 filtered_df.show()
21
22 # If desired, you can save a new DataFrame
23 # filtered_df.write.csv("FilteredTemperatureData.csv")
24
25 # closure Spark Session
26 spark.stop()
27
28
```

4.2 Find the city with the highest variance of temperature samples in the given data.

```
4.2.py x
1 from pyspark.sql import SparkSession
2 from pyspark.sql.functions import var_samp
3
4 # initialization Spark Session
5 spark = SparkSession.builder \
6     .appName("Max Variance City in Temperature Data") \
7     .getOrCreate()
8
9 # Reading CSV files
10 df = spark.read .csv( path: "D:\Задание 4\Data after 1950 GlobalLandTemperaturesByMajorCity.csv", header=True, inferSchema=True)
11
12 # Calculate the sample variance of temperature for each city
13 variance_df = df.groupBy("City").agg(var_samp("AverageTemperature").alias("Variance"))
14
15 # Find the city with the highest sample variance
16 max_variance_city = variance_df.orderBy( "cols: "Variance", ascending=False).first()
17
18 # Print results
19 print(f"The city with the maximum temperature variance is {max_variance_city['City']} with a variance of {max_variance_city['Variance']}")
20
21 # closure Spark Session
22 spark.stop()
23
```

4.3 Calculate the average annual temperature data for St. Petersburg. Find the years in which the average temperature was higher than the previous and the following year.

```
4.3.py ×
1 from pyspark.sql import SparkSession
2 from pyspark.sql.functions import year, avg, col, lead, lag
3 from pyspark.sql.window import Window
4
5 # initialization Spark Session
6 spark = SparkSession.builder \
7     .appName("St. Petersburg Average Temperature Analysis") \
8     .getOrCreate()
9
10 # Reading CSV files
11 df = spark.read.option("header", "true").csv(path: "D:\Задание 4\Data after 1950 GlobalLandTemperaturesByMajorCity.csv", header=True, inferSchema=True)
12
13 # Filtering out St. Petersburg and extracting the year
14 stp_df = df.filter(col("City") == "Saint Petersburg").withColumn(colName="Year", year(col("dt")))
15
16 # Calculate the average temperature for each year
17 yearly_avg_temp = stp_df.groupBy("Year").agg(avg("AverageTemperature").alias("AvgTemp"))
18
19 # Use the window function to compare average temperatures of neighboring years
20 windowSpec = Window.orderBy("Year")
21 yearly_avg_temp = yearly_avg_temp.withColumn(colName="PrevYearTemp", lag(col="AvgTemp", offset=1).over(windowSpec))
22 yearly_avg_temp = yearly_avg_temp.withColumn("NextYearTemp", lead(col="AvgTemp", offset=1).over(windowSpec))
23
24 # Identify years with higher average temperatures than the previous and subsequent years
25 result = yearly_avg_temp.filter((col("AvgTemp") > col("PrevYearTemp")) & (col("AvgTemp") > col("NextYearTemp")))
26
27 # output result
28 result.select("Year").show()
29
30 # close Spark Session
31 spark.stop()
32
```

4.4 Find out which cities:

1. the difference between the maximum and minimum mean annual temperatures in the sample is the greatest.
2. has the greatest difference between the mean temperature in January and the mean temperature in July.
3. has the greatest number of months with negative mean annual temperatures.

```

4.4.py
1 from pyspark.sql import SparkSession
2 from pyspark.sql.functions import col, year, month, avg, abs, to_date
3
4 # initialization Spark Session
5 spark = SparkSession.builder \
6     .appName("City Temperature Analysis") \
7     .getOrCreate()
8
9 # Reading CSV files
10 df = spark.read.option("header", "true").csv(path="D:\Задание 4\Data after 1950 GlobalLandTemperaturesByMajorCity.csv", header=True, inferSchema=True)
11
12 # Convert a date string to a date type and extract the year and month
13 df = df.withColumn("date", to_date(col("date"), format="yyyy-MM-dd")) \
14     .withColumn("year", year(col("date"))) \
15     .withColumn("month", month(col("date")))
16
17 # Registering a DataFrame as a SQL Queryable View
18 df.createOrReplaceTempView("temperature_data")
19
20 # SQL Query 1: Cities with the largest difference between the maximum and minimum annual average temperature
21 query1 = """
22 SELECT City, MAX(YearlyAvgTemp) - MIN(YearlyAvgTemp) AS MaxDiff
23 FROM (
24     SELECT City, Year, AVG(AverageTemperature) AS YearlyAvgTemp
25     FROM temperature_data
26     GROUP BY City, Year
27 )
28 GROUP BY City
29 ORDER BY MaxDiff DESC
30 LIMIT 1;
31 """
32
33 # SQL Query 2: Cities with the largest difference between the mean temperature in January and the mean temperature in July
34 query2 = """
35 SELECT City, ABS(JanTemp - JulTemp) AS MaxDifference
36 FROM (
37     SELECT City,
38         AVG(CASE WHEN Month = 1 THEN AverageTemperature ELSE NULL END) AS JanTemp,
39         AVG(CASE WHEN Month = 7 THEN AverageTemperature ELSE NULL END) AS JulTemp
40     FROM temperature_data
41     GROUP BY City
42 )
43 ORDER BY MaxDifference DESC
44 LIMIT 1;
45 """
46
47 # SQL Query 3: Cities with the highest number of months with negative average annual temperatures
48 query3 = """
49 SELECT City, COUNT(*) AS NegativeMonths
50 FROM (
51     SELECT City, Month, AVG(AverageTemperature) AS MonthlyAvgTemp
52     FROM temperature_data
53     GROUP BY City, Month
54     HAVING MonthlyAvgTemp < 0
55 )
56 GROUP BY City
57 ORDER BY NegativeMonths DESC
58 LIMIT 1;
59 """
60
61 # Execute a SQL query and print the results
62 result1 = spark.sql(query1)
63 result2 = spark.sql(query2)
64 result3 = spark.sql(query3)
65
66 print("The city with the largest difference between the maximum and minimum annual average temperature of the city:")
67 result1.show()
68
69 print("Cities with the largest difference between the average temperature in January and the average temperature in July:")
70 result2.show()
71
72 print("Cities with the highest number of months with negative average temperatures throughout the year:")
73 result3.show()
74
75 # closure Spark Session
76 spark.stop()
77

```