Session 19: SPARK SQL

Assignment 19.3

Student Name: Abarajithan SA

Course: Big Data Hadoop & Spark Training

Start Date: 2017-09-09

End Date: 2017-11-26

Assignment 19.3– Introduction on parquet file

Contents

Introduction

In this assignment, we are going to see a little introduction on parquet file.

Problem Statement

1. Create a **dataframe** with **1 to 100** and save as parquet file.

Task - Create a dataframe with 1 to 100 and save as parquet file.

- Creating a RDD which has numbers from 1 to 100.
- Creating a **dataframe** with above RDD.
- Writing a parquet file from above defined dataframe and then reading it a desired location.

Creating a RDD which has numbers from 1 to 100.

- ***val numbers = sc.parallelize(1 to 100)***
- ***numbers.collect()***

```
scala> val numbers = sc.parallelize(1 to 100)
numbers: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[0] at parallelize at <console>:24

scala> numbers.collect()
18/01/12 17:54:58 WARN SizeEstimator: Failed to check whether UseCompressedOops is set; assuming yes
res0: Array[Int] = Array(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32,
 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68
, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100)
```

Creating a dataframe with above RDD.

- **val numbersDF = numbers.toDF()**
- **numbers.show()**

```
scala> val numbersDF = numbers.toDF()
18/01/12 17:57:34 WARN ObjectStore: Failed to get database global_temp, returning NoSuchObjectException
numbersDF: org.apache.spark.sql.DataFrame = [value: int]

scala> numbersDF.show()
+-----+
|value|
+-----+
|    1|
|    2|
|    3|
|    4|
|    5|
|    6|
|    7|
|    8|
|    9|
|   10|
|   11|
|   12|
|   13|
|   14|
|   15|
|   16|
|   17|
|   18|
|   19|
|   20|
+-----+
only showing top 20 rows
```

Writing a parquet file from above defined dataframe and then reading it a desired location.

- ***numbersDF.write.parquet("/home/acadgild/hadoop/numbers.parquet")***
- ***val numbersRead = spark.read.parquet("/home/acadgild/hadoop/numbers.parquet")***
- ***numbersRead.show()***

Expected output

```
scala> numbersDF.write.parquet("/home/acadgild/hadoop/numbers.parquet")

scala> val numbersRead = spark.read.parquet("/home/acadgild/hadoop/numbers.parquet")
numbersRead: org.apache.spark.sql.DataFrame = [value: int]

scala> numbersRead.show()
+-----+
|value|
+-----+
|    1|
|    2|
|    3|
|    4|
|    5|
|    6|
|    7|
|    8|
|    9|
|   10|
|   11|
|   12|
|   13|
|   14|
|   15|
|   16|
|   17|
|   18|
|   19|
|   20|
+-----+
only showing top 20 rows
```