NAAN MUDHALVAN

BIG DATA ANALYTICS

ASSESSMENT – VI

Spark SQL and DataFrames

Homework

1.What is Spark SQL?

Spark SQL is Apache Spark's module for working with structured data. Spark SQL brings native support for SQL to Spark and streamlines the process of querying data stored both in RDDs (Spark's distributed datasets) and in external sources. Spark SQL conveniently blurs the lines between RDDs and relational tables. Unifying these powerful abstractions makes it easy for developers to intermix SQL commands querying external data with complex analytics, all within in a single application. Concretely, Spark SQL will allow developers to:

- Import relational data from Parquet files and Hive tables
- Run SQL queries over imported data and existing RDDs
- Easily write RDDs out to Hive tables or Parquet files

Spark SQL also includes a cost-based optimizer, columnar storage, and code generation to make queries fast. At the same time, it scales to thousands of nodes and multi-hour queries using the Spark engine, which provides full mid-query fault tolerance, without having to worry about using a different engine for historical data.

2. Is there a module to implement SQL in Spark? How does it work?

Spark SQL is Apache Spark's module for working with structured data. Seamlessly mix SQL queries with Spark programs.

Spark SQL lets you query structured data inside Spark programs, using either SQL or a familiar DataFrame API. Usable in Java, Scala, Python and R.

```
results = spark.sql(
  "SELECT * FROM people")
names = results.map(lambda p: p.name)
```

Connect to any data source the same way.

DataFrames and SQL provide a common way to access a variety of data sources, including Hive, Avro, Parquet, ORC, JSON, and JDBC. You can even join data across these sources.

```
spark.read.json("s3n://...")
  .registerTempTable("json")
results = spark.sql(
  """SELECT *
     FROM people
     JOIN json ...""")
```

3. What is a Parquet file?

Parquet is a columnar format that is supported by many other data processing systems. Spark SQL provides support for both reading and writing Parquet files that automatically preserves the schema of the original data. When reading Parquet files, all columns are automatically converted to be nullable for compatibility reasons.

It has the following features:

3.1. Loading Data

3.2. Schema Merging

3.3. Metadata Refreshing

3.4. Columnar Encryption

4. List the functions of Spark SQL.

Spark SQL String Functions

String functions are used to perform operations on String values such as computing numeric values, calculations and formatting etc. The String functions are grouped as " string_funcs" in spark SQL. The following given are some of the String functions in Spark:

- concat_ws(sep: String, exprs: Column*): Column
- encode(value: Column, charset: String): Column
- length(e: Column): Column
- instr(str: Column, substring: String): Column
- initcap(e: Column): Column
- decode(value: Column, charset: String): Column

Date and Time Functions

The Date and Time Functions in Spark help in performing operations like returning the current date as a date column, returning the number of days from beginning till end or converting a column into a ‘DateType’ with a specific date format. Some of the Date and Time functions used in Spark are as follows :

- current_date () : Column
- to_date(e: Column): Column
- to_date(e: Column, fmt: String): Column
- add_months(startDate: Column, numMonths: Int): Column
- date_add(start: Column, days: Int): Column
- date_sub(start: Column, days: Int): Column

Collection Functions

Collection Functions in Spark SQL are basically used to perform operations on groups or arrays. Some of the important Collection functions in Spark SQL are:

- array_contains(column: Column, value: Any)
- array_except(col1: Column, col2: Column)
- array_join(column: Column, delimiter: String, nullReplacement: String)
- array_join(column: Column, delimiter: String)
- array_remove(column: Column, element: Any)
- array_repeat(left: Column, right: Column)
- arrays_zip(e: Column*)

Math Functions

Math Functions are used to perform calculations such as Trigonometry (Sin, Cos, Tan), Hyperbolic statements etc. Following mentioned are some of the Math Functions used in Spark SQL:

- sin (e : Column) : Column
- sin(columnName: String): Column

- cosh(e: Column): Column
- cosh(columnName: String): Column

Aggregate Functions

Aggregate functions are used to perform aggregate operations on DataFrame columns. The working of aggregate functions is on the basis of the groups and rows. Following are some of the aggregate functions in Spark SQL:

- approx_count_distinct(e: Column)
- approx_count_distinct(e: Column, rsd: Double)
- avg(e: Column)
- collect_set(e: Column)
- countDistinct(expr: Column, exprs: Column*)

Window Functions

The use of Window functions in Spark is to perform operations like calculating the rank and row number etc. on large sets of input rows. These Window functions are available by importing ‘org.apache.spark.sql.’ functions. Let us now have a look at some of the important Window functions available in Spark SQL :

- row_number(): Column
- rank(): Column
- dense_rank(): Column
- cume_dist(): Column
- ntile(n: Int): Column

5. How is Spark SQL different from HQL and SQL?

Basis of Comparison	Apache Hive	Apache Spark SQL
Structure	An open source data warehousing system which is built on top of Hadoop	Mainly used for structured data processing where more information is retrieved by using structured query language.
Processing	Large datasets which are stored in hadoop files are analyzed and queried. Processing is mainly performed using SQL.	The processing of Apache Spark SQL involves heavy computations performed due to which a right optimization technique is required. Interaction with Spark SQL is possible in different ways such as

		Dataset and DataFrame API.
Initial Release	Hive was first released in 2012	Spark SQL was first released in 2014
Latest Release	The latest version of Hive is released on 18th November 2017 : release 2.3.2	The latest version of Apache Spark SQL is released on 28th February 2018: 2.3.0
Licensing	It is Apache version 2 open sourced	Open sourced through Apache version 2
Implementation language	Java language primarily can be used to implement apache Hive	Spark SQL can be implemented on Scala, Java, R as well as Python
Database model	Primarily its database model is RDBMS	Though Spark SQL is capable of integrating with any NoSQL

		database but primarily its database model is RDBMS
Additional Database Models	Additional database model is a key-value store which can take data in the form of JSON	Key-value store is the additional database model
Development	Hive was originally developed by Facebook but later on donated to Apache Software foundation	It was originally developed by Apache Software Foundation itself
Server Operating System	It supports all operating system with a Java Virtual machine environment	It supports several operating systems such as Windows, X, Linux etc.
Access Methods	It supports ODBC, JDBC and Thrift	It only supports ODBC and JDBC

Programming Language Support	Several programming languages such as C++, PHP, Java, Python, etc. are supported	Several programming languages such as Java, R, Python, and Scala is supported
Partitioning Methods	Data sharding method is used to store data on various nodes	It makes use of Apache Spark Core for storing data on various nodes

6. Why is Spark SQL used?

Spark SQL is a Spark module for structured data processing. It provides a programming abstraction called DataFrames and can also act as a distributed SQL query engine. It enables unmodified Hadoop Hive queries to run up to 100x faster on existing deployments and data. It also provides powerful integration with the rest of the Spark ecosystem (e.g., integrating SQL query processing with machine learning).

7. Is Spark SQL faster than Hive?

Speed: – The operations in Hive are slower than Apache Spark in terms of memory and disk processing as Hive runs on top of Hadoop.
Read/Write operations: – The number of read/write operations in Hive are greater than in Apache Spark. This is because Spark performs its intermediate operations in memory itself.
Memory Consumption: – Spark is highly expensive in terms of memory than Hive due to its in-memory processing.
From the above we can say that Spark SQL is faster than Hive.