Spark SQL

Spark SQL

01 Contexto
02 DataFrames

01

Introducción

Spark SQL

¿Qué es?

Spark SQL es el módulo de Spark para el procesamiento de datos estructurados, es decir, datos con filas y columnas.

Utilidades:

- programar y trabajar sobre DataFrames.
- motor de consultas SQL distribuido.

SQLContext

¿Qué es?

SQLContext es el punto de entrada para todas las funcionalidades relacionales de Spark SQL.

- El objeto de SQLContext se instancia a través del SparkContext.
- Permite crear objetos de datos y hacer consultas en lenguaje SQL.

SQL Queries

¿Cómo se lanzan?

Lo que permite ejecutar consultas SQL al trabajar en Spark es la función ***sql*** de la clase SQLContext. Esto siempre devuelve los resultados en forma de DataFrame.

SparkSession

¿Qué es?

SparkSession representa un punto de entrada único a todas las funcionalidades de Spark e incluye las variables de contexto necesarias para manipular datos.

Al inicializar una **SparkSession** también se está inicializando a la par un **SparkContext.**

El **SparkContext** representa la conexión al cluster, diciéndole a Spark cómo y dónde acceder a él.

PySpark

¿Qué es?

PySpark es la API de Python para Spark. Por lo tanto, permite el procesamiento de *big data* con Spark mediante el lenguaje de programación Python.

Al trabajar con DataFrames, las funcionalidades de PySpark se construyen a partir de Spark SQL.

Módulo y clases de pyspark.sql

Hay una serie de módulos y clases dentro de *pyspark.sql*. El listado entero se puede encontrar en el siguiente enlace:

http://spark.apache.org/docs/2.1.0/api/python/pyspark.sql.html#module-pyspark.sql.functions

Entre todos ellos, hay dos módulos especialmente importantes y útiles:

- *pyspark.sql.functions*
- *pyspark.sql.types*

pyspark.sql.functions

Características

- En este módulo se halla una colección de funciones disponibles para DataFrames.
- En muchas ocasiones el parámetro de entrada para estas funciones es un objeto de tipo columna.

pyspark.sql.types

Características

- En este módulo se hallan los distintos tipos de datos disponibles.
- Hay 14 tipos de datos distintos aceptados.

02

DataFrames

DataFrames

Definición

Un DataFrame es una colección inmutable y distribuida de datos organizados en columnas de forma tabular que permite una abstracción de los datos a alto nivel.

DataFrames

Estructura de datos relacional en Spark

- Equivale a una tabla en una bbdd relacional
- Equivalente a un dataframe de pandas o R
- Es un objeto distribuido
- Construido sobre un RDD
- Las operaciones también son de tipo Lazy

DataFrames

Estructura de datos relacional en Spark

```
: columnas = ["Permit Number","Permit Type Definition","Street Name","Street Number"]
buildings_DF.select(columnas).show(3)
```

```
+-------------+---------------------+-----------+-------------+
|Permit Number|Permit Type Definition|Street Name|Street Number|
+-------------+---------------------+-----------+-------------+
| 201505065519|          sign - erect|      Ellis|          140|
| 201604195146|          sign - erect|      Geary|          440|
| 201605278609|  additions alterat...|    Pacific|         1647|
+-------------+---------------------+-----------+-------------+
only showing top 3 rows
```

DataFrames

Estructuran los datos entorno a un SchemaRDD

```
root
 |-- Permit Number: string (nullable = true)
 |-- Permit Type: string (nullable = true)
 |-- Permit Type Definition: string (nullable = true)
 |-- Permit Creation Date: string (nullable = true)
 |-- Block: string (nullable = true)
 |-- Lot: string (nullable = true)
 |-- Street Number: string (nullable = true)
 |-- Street Number Suffix: string (nullable = true)
 |-- Street Name: string (nullable = true)
 |-- Street Suffix: string (nullable = true)
 |-- Unit: string (nullable = true)
 |-- Unit Suffix: string (nullable = true)
 |-- Description: string (nullable = true)
 |-- Current Status: string (nullable = true)
```

DataFrames

Transformaciones

map	filter	flatMap	mapPartitions	mapPartitionswithIndex
sample	union	intersection	distinct	aggregateByKey
groupByKey	reduceByKey	sortByKey	join	repartitionAndSortWithinPartitions
repartition	coalesce	pipe	cartesian	cogroup

DataFrames

Acciones

reduce	collect	count	first
take	takeSample	takeOrdered	saveAsTextFile
saveAsSequenceFile	saveAsObjectFile	countByKey	foreach

DataFrames

Acciones poco recomendadas

- ***collect()***:

  se trata de una acción muy poco recomendada y que sólo se debería utilizar en ocasiones muy puntuales. Se debería evitar siempre que sea posible puesto que un *collect()* vierte todos los datos en el mismo nodo. Como consecuencia, es una acción muy pesada con la que se pierde la paralelización de los datos.

NOTEBOOK

Conceptos básicos de Spark SQL con ejemplos de las transformaciones y acciones más típicas

Spark SQL

Lazy Evaluation Acciones y Transformaciones

NOTEBOOK

Ejercicios prácticos para aplicar las transformaciones y acciones más típicas en Spark

Spark SQL

Lazy Evaluation Ejercicios

Thanks