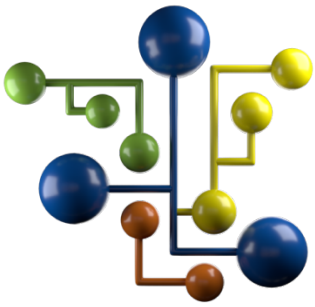


Data Science Academy

www.datascienceacademy.com.br

Engenharia de Dados com Hadoop e Spark

Componentes do Apache Spark SQL

Uma das principais vantagens do Spark SQL é que você não precisa reaprender nada. Podemos usar os mesmos conceitos de SQL que usamos em bancos de dados relacionais, extrair dados para o Spark e então nos beneficiarmos do processamento paralelo e distribuído fornecido pelo framework Spark através de clusters de computadores. O Spark SQL é uma biblioteca poderosa que Cientistas e Analistas de Dados, podem utilizar para realizar análise de dados em suas empresas ou clientes.

Vejamos os componentes do Spark SQL.

DataFrame

Um DataFrame é uma coleção de dados distribuídos e organizados em forma de colunas nomeadas. É baseado no conceito de estrutura de dados da linguagem R, dataframes do Pandas e similar a uma tabela de um banco de dados relacional.

Nas primeiras versões, o componente DataFrame era chamado de SchemaRDD. DataFrames podem ser transformados em RDDs. DataFrames podem ser criados a partir de diferentes fontes de dados e embora tenham métodos diferentes para manipulação, internamente o Spark trata um DataFrame como um RDD. Mas usar DataFrames facilita o trabalho de quem está construindo as aplicações de análise de dados.

DataFrames suportam operações: filter, join, groupby, agg e aninhamento de operações.

Spark Session

Criamos um Spark Session para acessar as funcionalidades do Spark SQL. Dataframes são criados a partir de Spark Sessions, que permitem registrar um Dataframes como tabelas temporárias e executar queries SQL (muito útil no processamento de streams de dados).

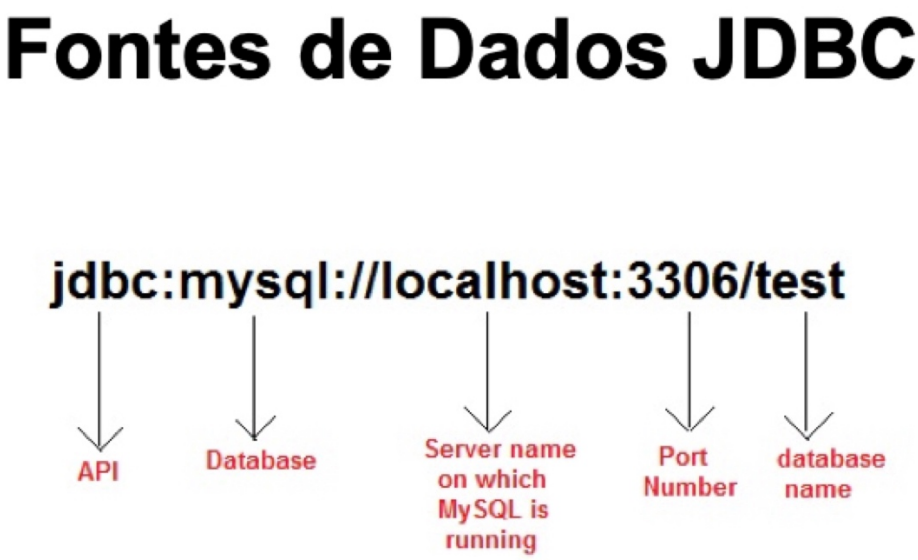
SQL Context

O Spark SQL fornece o componente SQLContext para encapsular todas as funcionalidades relacionais no Spark. É possível criar os SQLContext a partir do Spark Context. Existe também um componente HiveContext o qual fornece um conjunto maior de funcionalidades para o SQLContext. Este componente pode ser utilizado para escrever consultas utilizando o HiveQL e com isto, ler dados de tabelas Hive, a partir do Spark.

Fontes de Dados JDBC

Outras funcionalidades na biblioteca Spark SQL incluem fontes de dados que fazem uso de JDBC (Java Database Connection) como interface de integração. Interfaces de integração JDBC podem ser utilizadas para ler informações armazenadas em banco de dados relacionais. O JDBC é um conjunto de classes e interfaces (API) escritas em Java para execução e manipulação de resultados de instruções SQL para qualquer banco de dados relacional. Para cada Banco de dados há um driver JDBC. Apenas para que você tenha ideia da importância de se utilizar JDBC como conexão a bancos de dados, todos os principais bancos comerciais, como Oracle, SQL Server, DB2 e MySQL suportam conexão JDBC.

Para realizar uma conexão via JDBC, precisamos definir uma string de conexão, onde especificamos a API, o banco de dados, o nome do servidor, a porta e o nome do banco de dados que iremos conectar. Abaixo um exemplo de como seria essa string:



Tabelas Temporárias

As tabelas temporárias são outro importante recurso do Spark SQL. Podemos usar operações SQL em tabelas temporárias e embora sejam estruturas simples, são muito poderosas. Uma query executada em uma tabela temporária retorna um outro Dataframe.