**SPARK SQL:**

Spark SQL, Spark’s interface for working with structured and semi-structured data. Structured data is any data that has a schema—that is, a known set of fields for each record.

Spark SQL provides three main capabilities

1. It provides a DataFrame abstraction in Python, Java, and Scala that simplifies working with structured datasets. DataFrames are similar to tables in a relational database.

2. It can read and write data in a variety of structured formats.

3. It lets you query the data using SQL, both inside a Spark program and from external tools that connect to Spark SQL through standard database connectors (JDBC/ODBC), such as business intelligence tools like Tableau.

To use Spark SQL, we construct a based HiveContext on our SparkContext. This context provides additional functions for querying and interacting with Spark SQL data.

First, we need to Import Spark SQL

import org.apache.spark.sql.hive.HiveContext

// Or if you can't have the hive dependencies

import org.apache.spark.sql.SQLContext

Then create HiveContext later import implicits.

Basic Example Query:

val input = hiveCtx.jsonFile(inputFile)

// Register the input schema RDD

input.registerTempTable("tweets")

// Select tweets based on the retweetCount

val topTweets = hiveCtx.sql("SELECT text, retweetCount FROM tweets ORDER BY retweetCount LIMIT 10")