val data=1 to 10000

//Create an RDD

val distDate = sc.parallelize(data)

distData.filter(\_<10).collect()

Spark -work with JSON

val lines=sc.sqlContext.read.json("?.json”)

df.show()

df.printSchema

//show all column

df.select(“\*”).show()

//select a column

df.select(df(“name”)).show()

//filter age more than 20

df.filter(df(“age”)>20).show()

//save table above

df.registerTempTable(“people”)

val expEmployees = sqlContext.sql(“SELECT name FROM people where age>20”) //write sql in

expEmployees.collect().foreach(println)

Spark as ETL tool-Write to Parquet file using Spark

//Create external table in hive

CREATE EXTERNAL TABLE person ( name String, age Int, sex String)  
STORED as PARQUET  
LOCATION '/tmp/person'

//Create sample data

case class Person(name: String, age: Int, sex:String)  
val data = Seq(Person("Jack", 25,"M"), Person("Jill", 25,"F"), Person("Jess", 24,"F"))  
val df = data.toDF()

//Save data to HDFS  
import org.apache.spark.sql.SaveMode  
df.select("name", "age", "sex").write.mode(SaveMode.Append).format("parquet").save("/tmp/person")  
  
//Add new data  
val data = Seq(Person("John", 25,"M"))  
val df = data.toDF()  
df.select("name", "age", "sex").write.mode(SaveMode.Append).format("parquet").save("/tmp/person")

//Query the data from Hive

SELECT \* FROM preson;