**PYTHON + SPARK + CASSANDRA**

* **Install oracle jdk – ver 1.8**
  + **Set JAVA\_HOME environmental variable**
* **Install spark**
  + **Set environmental variables SPARK\_HOME and add %SPARK\_HOME%\bin to path and also add PYTHONPATH with %SPARK\_HOME%\python\lib**
* **Install datastax Cassandra 3.7**
  + **Set environmental variables CASSANDRA\_HOME**
* **Install anaconda 3 (I am working on python 3.5 version and using pycharm community edition)**
* **Need to download winutils**
  + **Set HADOOP\_HOME with winutils and %HADOOP\_HOME%\bin in path**

**pyspark-cassandra**

[**https://spark-packages.org/package/TargetHolding/pyspark-cassandra**](https://spark-packages.org/package/TargetHolding/pyspark-cassandra)

**Compatibility**

**Spark**

The current version of PySpark Cassandra is succesfully used with Spark version 1.5 and 1.6. Use older versions for Spark 1.2, 1.3 or 1.4.

**Cassandra**

PySpark Cassandra is compatible with Cassandra:

* 2.1.5 and higher
* 2.2
* 3

**Python**

PySpark Cassandra is used with python 2.7, python 3.3 and 3.4.

**Scala**

PySpark Cassandra is currently only packaged for Scala 2.10

**We will try to do a simple task of copying data from table in Cassandra to write to another table in Cassandra using pyspark-cassandra connector and pyspark package**

**We need to create a keyspace, table under the keyspace created and insert data**

* Type cqlsh

**create table user ( name text PRIMARY KEY, favorite\_food text );**

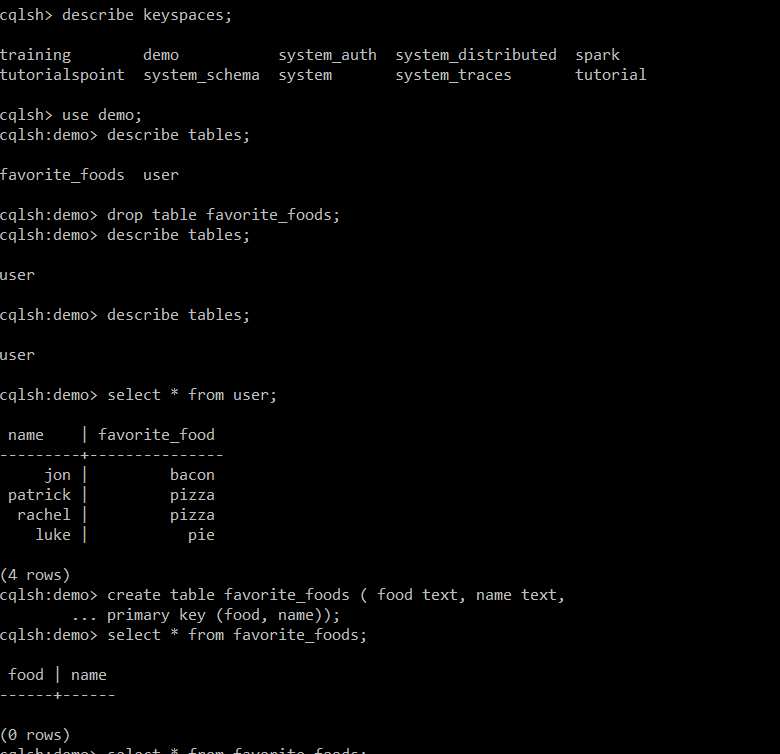
**insert into user (name, favorite\_food) values ('jon', 'bacon');**

**insert into user (name, favorite\_food) values ('luke', 'pie');**

**insert into user (name, favorite\_food) values ('patrick', 'pizza');**

**insert into user (name, favorite\_food) values ('rachel', 'pizza');**

**create table favorite\_foods ( food text, name text, primary key (food, name));**

****

**In a python file (sparkcassandra.py) include the below code:**

**from** pyspark\_cassandra **import** CassandraSparkContext, Row  
**from** pyspark **import** SparkContext, SparkConf  
conf = SparkConf() \  
 .setAppName(**"User Food Migration"**) \  
 .setMaster(**"local[\*]"**) \  
 .set(**"spark.cassandra.connection.host"**, **"127.0.0.1"**)  
sc = CassandraSparkContext(conf=conf)  
users = sc.cassandraTable(**"demo"**, **"user"**)  
favorite\_foods = users.map(**lambda** x:  
 {**"food"**:x[**'favorite\_food'**],  
 **"name"**:x[**'name'**]} )  
favorite\_foods.saveToCassandra(**"demo"**, **"favorite\_foods"**)

Execute the below command in cmd (Run as Administrator):

You could use Pyspark or spark-submit

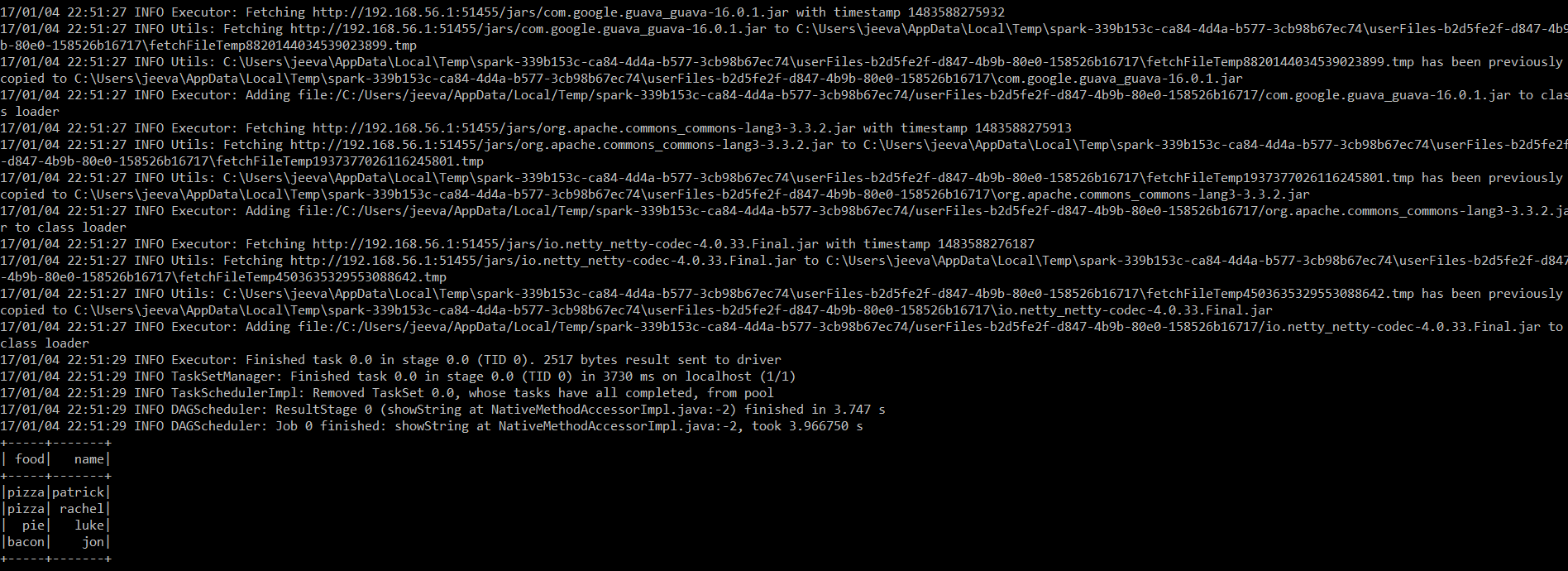
****

The task will be executed and disconnected from spark context, also disconnects from the Cassandra

* **To check the data from a Cassandra table which we created in the previous section on Spark console. In other Python file (sc.py) using “SQLContext”**   
  **from** pyspark **import** SparkContext, SparkConf  
  **from** pyspark.sql **import** SQLContext  
    
  conf = SparkConf()\  
   .setAppName(**"Python Script for getting the table ‘favorite foods’"**)\  
   .setMaster(**"local[\*]"**) \  
   .set(**"spark.cassandra.connection.host"**, **"127.0.0.1"**)  
  sc = SparkContext(conf=conf)  
  sqlContext = SQLContext(sc)  
  sqlContext.read.format(**"org.apache.spark.sql.cassandra"**).options(table=**"favorite\_foods"**, keyspace=**"demo"**).load().show()

Execute the below command in cmd (Run as Administrator):

****

****

* **To cross check whether the data displayed on Spark console is correct or not.**

Type cqlsh in cmd and go to the keyspace created before. Then execute the below command:

Select \* from favorite\_foods;

