Run spark-shell

|  |
| --- |
| [shalaj@mac127 ~]$ spark-shell  Setting default log level to "WARN".  To adjust logging level use sc.setLogLevel(newLevel).  Welcome to  \_\_\_\_ \_\_  / \_\_/\_\_ \_\_\_ \_\_\_\_\_/ /\_\_  \_\ \/ \_ \/ \_ `/ \_\_/ '\_/  /\_\_\_/ .\_\_/\\_,\_/\_/ /\_/\\_\ version 1.6.0  /\_/  Using Scala version 2.10.5 (Java HotSpot(TM) 64-Bit Server VM, Java 1.7.0\_67)  Type in expressions to have them evaluated.  Type :help for more information.  Spark context available as sc (master = yarn-client, app id = application\_1496836281414\_0002).  SQL context available as sqlContext. |

Default sqlContext available as hiveContext

|  |
| --- |
| scala> sqlContext  res2: org.apache.spark.sql.SQLContext = org.apache.spark.sql.hive.HiveContext@dea6f53 |

|  |
| --- |
| scala> sqlContext.sql("show tables").show()  +----------+-----------+  | tableName|isTemporary|  +----------+-----------+  | raw\_lines| false|  | t1| false|  | t2| false|  | t3| false|  |word\_count| false|  +----------+-----------+ |

We can create our own hiveContext

|  |
| --- |
| scala> import org.apache.spark.sql.hive.HiveContext  import org.apache.spark.sql.hive.HiveContext  scala> val hc = new HiveContext(sc)  hc: org.apache.spark.sql.hive.HiveContext = org.apache.spark.sql.hive.HiveContext@3cd28223  scala> hc.sql("show tables").show  +----------+-----------+  | tableName|isTemporary|  +----------+-----------+  | raw\_lines| false|  | t1| false|  | t2| false|  | t3| false|  |word\_count| false|  +----------+-----------+ |

|  |
| --- |
| scala> hc.sql("create table Record (key int,value string)")  res0: org.apache.spark.sql.DataFrame = [result: string] |

Now Record table will be created under hive

Note: in Spark version 2.1 , the SqlContext is replaced by SparkSession , please go through the latest API