BIG DATA HADOOP & SPARK TRAINING

Rashmi Krishna

Integrate tools.

Give your own input in output-screenshot of report.

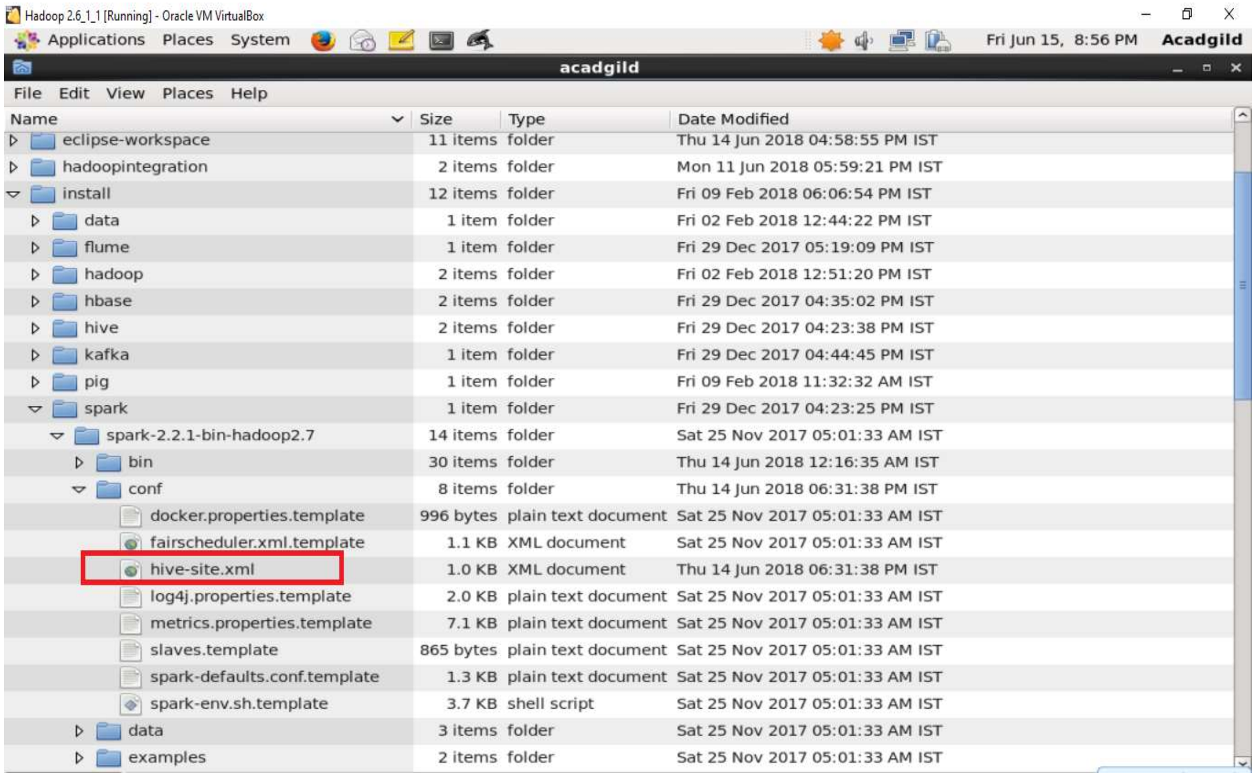
Task 1: As discussed in class integrate Spark Hive

Program to integrate spark with Hive, which displays the list of databases in Hive

```
import org.apache.spark.sql.SparkSession

object SparkHiveTest {

  def main (args: Array[String]) : Unit  = {

//creating a Spark Session with local master, app name, configure spark SQL warehouse
and hive metasore with thrift server on port: 9083. We also enable hive support

      val sparkSession = SparkSession.builder()

                                    .master("local")

                                    .appName("spark session example")

                                    .config("spark.sql.warehouse.dir","/user/hive/warehouse")

                                    .config("hive.metastore.uris", "thrift://localhost:9083")

                                    .enableHiveSupport()

                                    .getOrCreate()

//we create a function which shows first 8 list of databases present in  Hive

    val listOfDB = sparkSession.sqlContext.sql("show databases")

    listOfDB.show(8,false)

    println("test")  }}
```

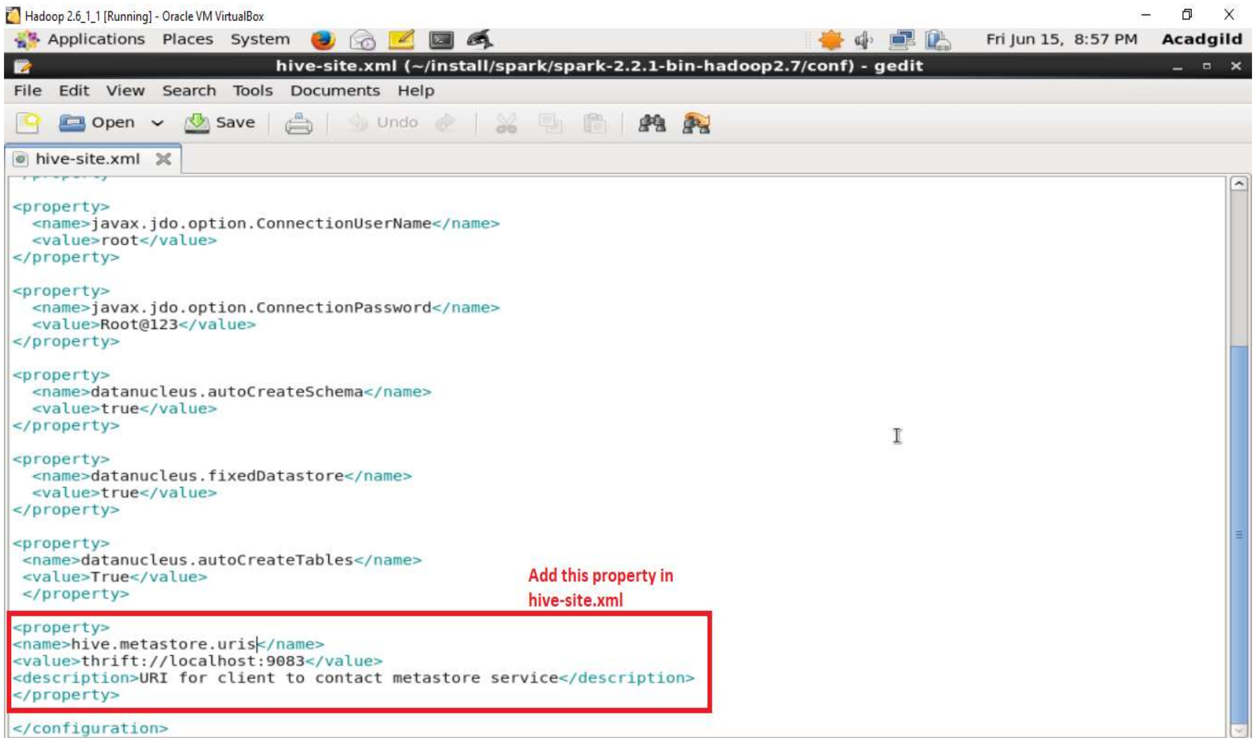
Before we execute the above program, we have to follow few steps, which are as follows: -

Step 1: We have to copy hive-site xml to spark-conf folder as shown below:


Hadoop 2.6_1_1 [Running] - Oracle VM VirtualBox
Applications Places System
Fri Jun 15, 8:56 PM Acadgild
acadgild
File Edit View Places Help

Name	Size	Type	Date Modified
eclipse-workspace	11 items	folder	Thu 14 Jun 2018 04:58:55 PM IST
hadoopintegration	2 items	folder	Mon 11 Jun 2018 05:59:21 PM IST
install	12 items	folder	Fri 09 Feb 2018 06:06:54 PM IST
data	1 item	folder	Fri 02 Feb 2018 12:44:22 PM IST
flume	1 item	folder	Fri 29 Dec 2017 05:19:09 PM IST
hadoop	2 items	folder	Fri 02 Feb 2018 12:51:20 PM IST
hbase	2 items	folder	Fri 29 Dec 2017 04:35:02 PM IST
hive	2 items	folder	Fri 29 Dec 2017 04:23:38 PM IST
kafka	1 item	folder	Fri 29 Dec 2017 04:44:45 PM IST
pig	1 item	folder	Fri 09 Feb 2018 11:32:32 AM IST
spark	1 item	folder	Fri 29 Dec 2017 04:23:25 PM IST
spark-2.2.1-bin-hadoop2.7	14 items	folder	Sat 25 Nov 2017 05:01:33 AM IST
bin	30 items	folder	Thu 14 Jun 2018 12:16:35 AM IST
conf	8 items	folder	Thu 14 Jun 2018 06:31:38 PM IST
docker.properties.template	996 bytes	plain text document	Sat 25 Nov 2017 05:01:33 AM IST
fairscheduler.xml.template	1.1 KB	XML document	Sat 25 Nov 2017 05:01:33 AM IST
hive-site.xml	1.0 KB	XML document	Thu 14 Jun 2018 06:31:38 PM IST
log4j.properties.template	2.0 KB	plain text document	Sat 25 Nov 2017 05:01:33 AM IST
metrics.properties.template	7.1 KB	plain text document	Sat 25 Nov 2017 05:01:33 AM IST
slaves.template	865 bytes	plain text document	Sat 25 Nov 2017 05:01:33 AM IST
spark-defaults.conf.template	1.3 KB	plain text document	Sat 25 Nov 2017 05:01:33 AM IST
spark-env.sh.template	3.7 KB	shell script	Sat 25 Nov 2017 05:01:33 AM IST
data	3 items	folder	Sat 25 Nov 2017 05:01:33 AM IST
examples	2 items	folder	Sat 25 Nov 2017 05:01:33 AM IST

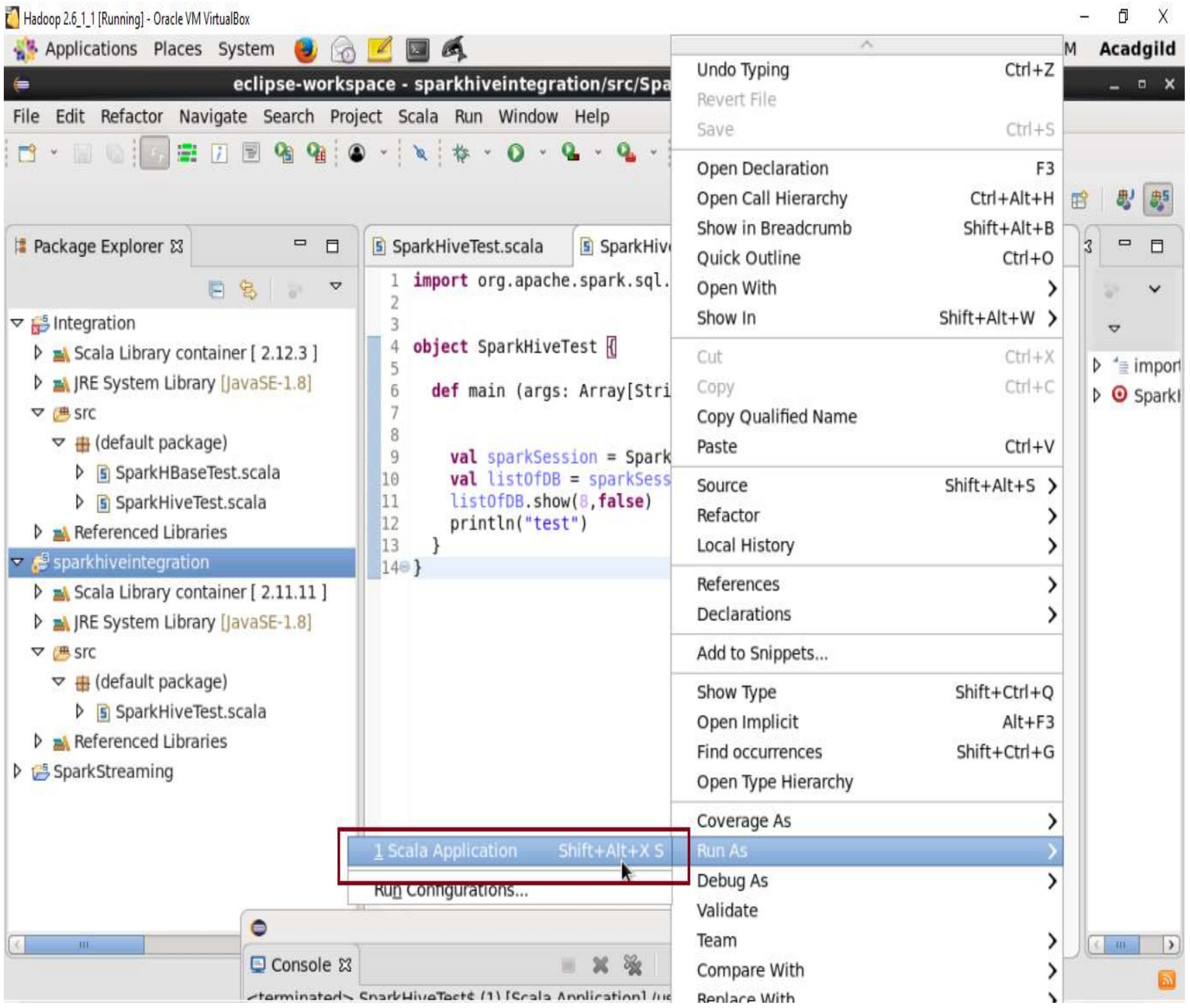
Step 2: We have add thrift server properties in hive-site xml as shown below:


Hadoop 2.6_1_1 [Running] - Oracle VM VirtualBox
Applications Places System
Fri Jun 15, 8:57 PM Acadgild
hive-site.xml (~/install/spark/spark-2.2.1-bin-hadoop2.7/conf) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
hive-site.xml
Add this property in hive-site.xml

```
<property>
  <name>javax.jdo.option.ConnectionUserName</name>
  <value>root</value>
</property>

<property>
  <name>javax.jdo.option.ConnectionPassword</name>
  <value>Root@123</value>
</property>

<property>
  <name>datanucleus.autoCreateSchema</name>
  <value>true</value>
</property>

<property>
  <name>datanucleus.fixedDatastore</name>
  <value>true</value>
</property>

<property>
 <name>datanucleus.autoCreateTables</name>
 <value>True</value>
 </property>

<property>
<name>hive.metastore.uris</name>
<value>thrift://localhost:9083</value>
<description>URI for client to contact metastore service</description>
</property>

</configuration>
```

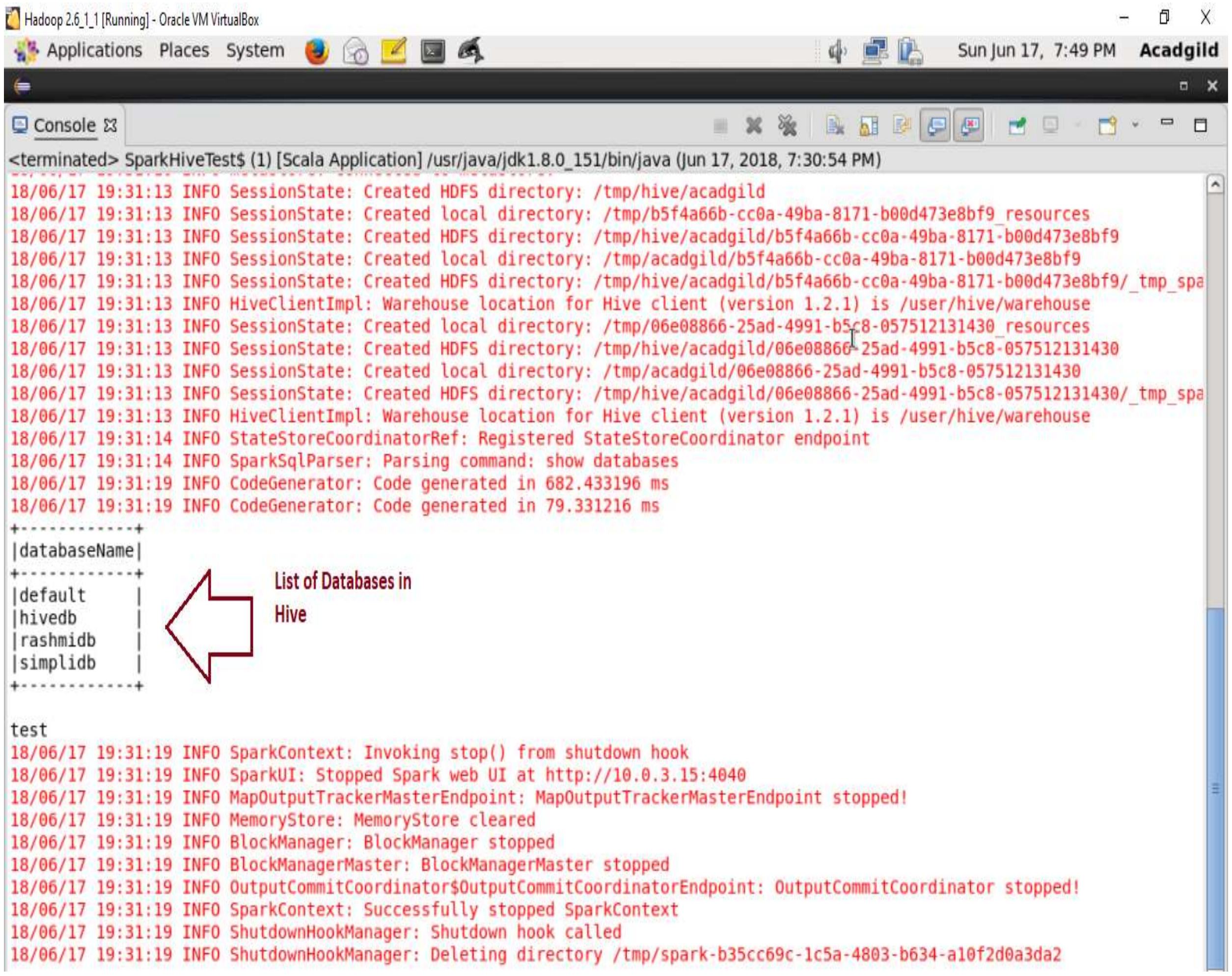
Once these two steps are done, we can execute the program to view the list of databases present in Hive using Spark Application.

We execute this program in eclipse as below:



```
import org.apache.spark.sql.


object SparkHiveTest {

  def main (args: Array[Stri


    val sparkSession = Spark
    val listOfDB = sparkSess
    listOfDB.show(8,false)
    println("test")
  }
}
```

We can see the list of databases in hive as below:


Hadoop 2.6_1_1 [Running] - Oracle VM VirtualBox
Applications Places System
Sun Jun 17, 7:49 PM Acadgild
Console
List of Databases in Hive

```
<terminated> SparkHiveTest$ (1) [Scala Application] /usr/java/jdk1.8.0_151/bin/java (Jun 17, 2018, 7:30:54 PM)
18/06/17 19:31:13 INFO SessionState: Created HDFS directory: /tmp/hive/acadgild
18/06/17 19:31:13 INFO SessionState: Created local directory: /tmp/b5f4a66b-cc0a-49ba-8171-b00d473e8bf9_resources
18/06/17 19:31:13 INFO SessionState: Created HDFS directory: /tmp/hive/acadgild/b5f4a66b-cc0a-49ba-8171-b00d473e8bf9
18/06/17 19:31:13 INFO SessionState: Created local directory: /tmp/acadgild/b5f4a66b-cc0a-49ba-8171-b00d473e8bf9
18/06/17 19:31:13 INFO SessionState: Created HDFS directory: /tmp/hive/acadgild/b5f4a66b-cc0a-49ba-8171-b00d473e8bf9/_tmp_spa
18/06/17 19:31:13 INFO HiveClientImpl: Warehouse location for Hive client (version 1.2.1) is /user/hive/warehouse
18/06/17 19:31:13 INFO SessionState: Created local directory: /tmp/06e08866-25ad-4991-b5c8-057512131430_resources
18/06/17 19:31:13 INFO SessionState: Created HDFS directory: /tmp/hive/acadgild/06e08866-25ad-4991-b5c8-057512131430
18/06/17 19:31:13 INFO SessionState: Created local directory: /tmp/acadgild/06e08866-25ad-4991-b5c8-057512131430
18/06/17 19:31:13 INFO SessionState: Created HDFS directory: /tmp/hive/acadgild/06e08866-25ad-4991-b5c8-057512131430/_tmp_spa
18/06/17 19:31:13 INFO HiveClientImpl: Warehouse location for Hive client (version 1.2.1) is /user/hive/warehouse
18/06/17 19:31:14 INFO StateStoreCoordinatorRef: Registered StateStoreCoordinator endpoint
18/06/17 19:31:14 INFO SparkSqlParser: Parsing command: show databases
18/06/17 19:31:19 INFO CodeGenerator: Code generated in 682.433196 ms
18/06/17 19:31:19 INFO CodeGenerator: Code generated in 79.331216 ms
+------------+
|databaseName|
+------------+
|default     |
|hivedb      |
|rashmidb    |
|simplidb    |
+------------+

test
18/06/17 19:31:19 INFO SparkContext: Invoking stop() from shutdown hook
18/06/17 19:31:19 INFO SparkUI: Stopped Spark web UI at http://10.0.3.15:4040
18/06/17 19:31:19 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
18/06/17 19:31:19 INFO MemoryStore: MemoryStore cleared
18/06/17 19:31:19 INFO BlockManager: BlockManager stopped
18/06/17 19:31:19 INFO BlockManagerMaster: BlockManagerMaster stopped
18/06/17 19:31:19 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
18/06/17 19:31:19 INFO SparkContext: Successfully stopped SparkContext
18/06/17 19:31:19 INFO ShutdownHookManager: Shutdown hook called
18/06/17 19:31:19 INFO ShutdownHookManager: Deleting directory /tmp/spark-b35cc69c-1c5a-4803-b634-a10f2d0a3da2
```

Task 2: As discussed in class integrate Spark Hbase

Program to integrate Spark with Hbase to create a table and insert contents to that table:

Imports required are as below: -

import org.apache.spark.SparkContext

import org.apache.hadoop.hbase.HBaseConfiguration

import org.apache.hadoop.hbase.mapreduce.TableInputFormat

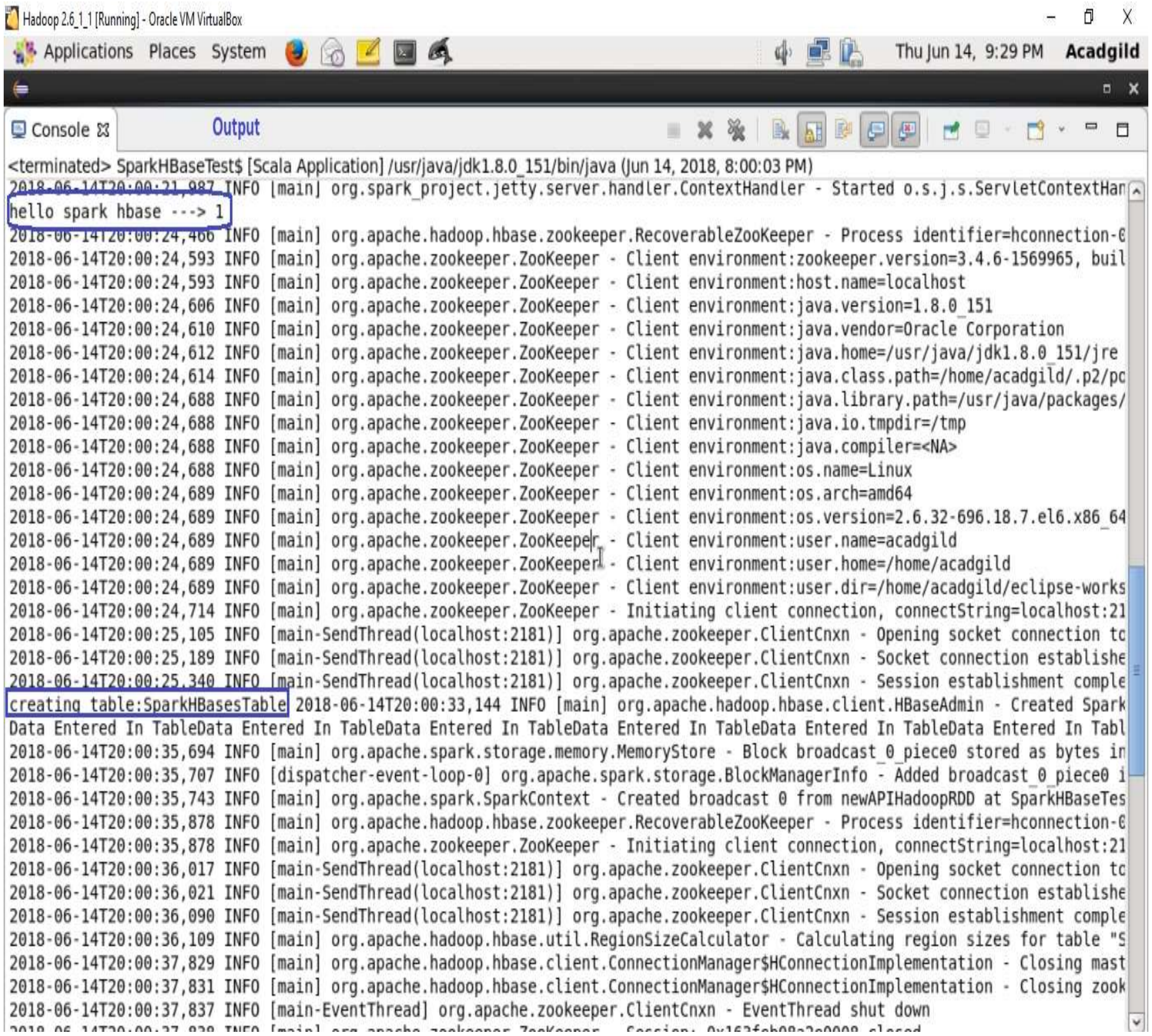
import org.apache.hadoop.hbase.client.HBaseAdmin

import org.apache.hadoop.hbase.{HTableDescriptor,HColumnDescriptor}

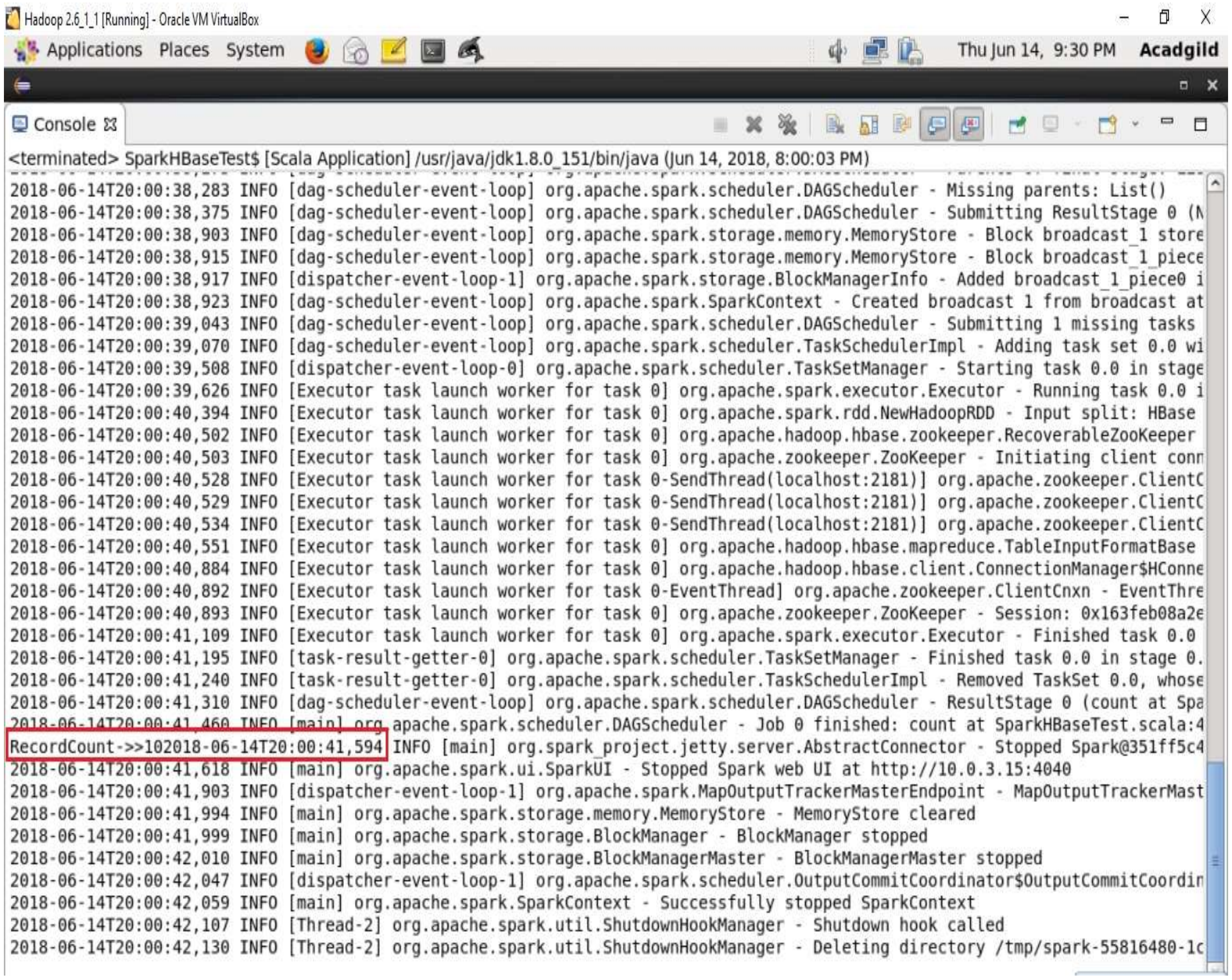
```
import org.apache.hadoop.hbase.util.Bytes

import org.apache.hadoop.hbase.client.{Put,HTable}

import org.apache.log4j._

import org.apache.hadoop.hbase.io.ImmutableBytesWritable

import org.apache.hadoop.hbase.client.Result

object SparkHBaseTest {

//main function

  def main(args: Array[String]) {

    // Create a SparkContext using every core of the local machine, named SparkHBaseTest

    val sc = new SparkContext("local[*]", "SparkHBaseTest")

    println("hello spark hbase ---> 1")

// creating HBase configuration and assigning a HBase table name as "SparkHBasesTable"

    val conf = HBaseConfiguration.create()

    val tablename = "SparkHBasesTable"

    conf.set(TableInputFormat.INPUT_TABLE,tablename)

    val admin = new HBaseAdmin(conf)

// checking if the table is already there in HBase, if table exists it will throw an error if not
//it will create a Hbase Table.

    if(!admin.isTableAvailable(tablename)){

      print("creating table:"+tablename+"\t")

      val tableDescription = new HTableDescriptor(tablename)

      tableDescription.addFamily(new HColumnDescriptor("cf".getBytes()));

      admin.createTable(tableDescription);

    } else {

      print("table already exists")

    }

//insert values into HBase table using for loop and print "data entered in the table" and
//print the record count

    val table = new HTable(conf,tablename);
```

```
   for(x <- 1 to 10){

     var p = new Put(new String("row" + x).getBytes());

     p.add("cf".getBytes(),"column1".getBytes(),new String("value" + x).getBytes());

     table.put(p);

     print("Data Entered In Table")

   }

   val hBaseRDD = sc.newAPIHadoopRDD(conf, classOf[TableInputFormat],
classOf[ImmutableBytesWritable],classOf[Result])

   print("RecordCount->>"+hBaseRDD.count())

//stop the spark context

   sc.stop()  }}
```

We will execute the above program in eclipse, and we can see the output as below:


Hadoop 2.6_1_1 [Running] - Oracle VM VirtualBox
Applications Places System
Thu Jun 14, 9:29 PM Acadgild
Console
Output

```
<terminated> SparkHBaseTest$ [Scala Application] /usr/java/jdk1.8.0_151/bin/java (Jun 14, 2018, 8:00:03 PM)
2018-06-14T20:00:21,987 INFO [main] org.spark_project.jetty.server.handler.ContextHandler - Started o.s.j.s.ServletContextHan
hello spark hbase ---> 1
2018-06-14T20:00:24,466 INFO [main] org.apache.hadoop.hbase.zookeeper.RecoverableZooKeeper - Process identifier=hconnection-0
2018-06-14T20:00:24,593 INFO [main] org.apache.zookeeper.ZooKeeper - Client environment:zookeeper.version=3.4.6-1569965, buil
2018-06-14T20:00:24,593 INFO [main] org.apache.zookeeper.ZooKeeper - Client environment:host.name=localhost
2018-06-14T20:00:24,606 INFO [main] org.apache.zookeeper.ZooKeeper - Client environment:java.version=1.8.0_151
2018-06-14T20:00:24,610 INFO [main] org.apache.zookeeper.ZooKeeper - Client environment:java.vendor=Oracle Corporation
2018-06-14T20:00:24,612 INFO [main] org.apache.zookeeper.ZooKeeper - Client environment:java.home=/usr/java/jdk1.8.0_151/jre
2018-06-14T20:00:24,614 INFO [main] org.apache.zookeeper.ZooKeeper - Client environment:java.class.path=/home/acadgild/.p2/po
2018-06-14T20:00:24,688 INFO [main] org.apache.zookeeper.ZooKeeper - Client environment:java.library.path=/usr/java/packages/
2018-06-14T20:00:24,688 INFO [main] org.apache.zookeeper.ZooKeeper - Client environment:java.io.tmpdir=/tmp
2018-06-14T20:00:24,688 INFO [main] org.apache.zookeeper.ZooKeeper - Client environment:java.compiler=<NA>
2018-06-14T20:00:24,688 INFO [main] org.apache.zookeeper.ZooKeeper - Client environment:os.name=Linux
2018-06-14T20:00:24,689 INFO [main] org.apache.zookeeper.ZooKeeper - Client environment:os.arch=amd64
2018-06-14T20:00:24,689 INFO [main] org.apache.zookeeper.ZooKeeper - Client environment:os.version=2.6.32-696.18.7.el6.x86_64
2018-06-14T20:00:24,689 INFO [main] org.apache.zookeeper.ZooKeeper - Client environment:user.name=acadgild
2018-06-14T20:00:24,689 INFO [main] org.apache.zookeeper.ZooKeeper - Client environment:user.home=/home/acadgild
2018-06-14T20:00:24,689 INFO [main] org.apache.zookeeper.ZooKeeper - Client environment:user.dir=/home/acadgild/eclipse-works
2018-06-14T20:00:24,714 INFO [main] org.apache.zookeeper.ZooKeeper - Initiating client connection, connectString=localhost:21
2018-06-14T20:00:25,105 INFO [main-SendThread(localhost:2181)] org.apache.zookeeper.ClientCnxn - Opening socket connection to
2018-06-14T20:00:25,189 INFO [main-SendThread(localhost:2181)] org.apache.zookeeper.ClientCnxn - Socket connection establishe
2018-06-14T20:00:25,340 INFO [main-SendThread(localhost:2181)] org.apache.zookeeper.ClientCnxn - Session establishment comple
creating table:SparkHBasesTable 2018-06-14T20:00:33,144 INFO [main] org.apache.hadoop.hbase.client.HBaseAdmin - Created Spark
Data Entered In TableData Entered In TableData Entered In TableData Entered In TableData Entered In TableData Entered In Tabl
2018-06-14T20:00:35,694 INFO [main] org.apache.spark.storage.memory.MemoryStore - Block broadcast_0_piece0 stored as bytes in
2018-06-14T20:00:35,707 INFO [dispatcher-event-loop-0] org.apache.spark.storage.BlockManagerInfo - Added broadcast_0_piece0 i
2018-06-14T20:00:35,743 INFO [main] org.apache.spark.SparkContext - Created broadcast 0 from newAPIHadoopRDD at SparkHBaseTes
2018-06-14T20:00:35,878 INFO [main] org.apache.hadoop.hbase.zookeeper.RecoverableZooKeeper - Process identifier=hconnection-0
2018-06-14T20:00:35,878 INFO [main] org.apache.zookeeper.ZooKeeper - Initiating client connection, connectString=localhost:21
2018-06-14T20:00:36,017 INFO [main-SendThread(localhost:2181)] org.apache.zookeeper.ClientCnxn - Opening socket connection to
2018-06-14T20:00:36,021 INFO [main-SendThread(localhost:2181)] org.apache.zookeeper.ClientCnxn - Socket connection establishe
2018-06-14T20:00:36,090 INFO [main-SendThread(localhost:2181)] org.apache.zookeeper.ClientCnxn - Session establishment comple
2018-06-14T20:00:36,109 INFO [main] org.apache.hadoop.hbase.util.RegionSizeCalculator - Calculating region sizes for table "S
2018-06-14T20:00:37,829 INFO [main] org.apache.hadoop.hbase.client.ConnectionManager$HConnectionImplementation - Closing mast
2018-06-14T20:00:37,831 INFO [main] org.apache.hadoop.hbase.client.ConnectionManager$HConnectionImplementation - Closing zook
2018-06-14T20:00:37,837 INFO [main-EventThread] org.apache.zookeeper.ClientCnxn - EventThread shut down
```



Hadoop 2.6_1_1 [Running] - Oracle VM VirtualBox

Applications Places System Thu Jun 14, 9:30 PM Acadgild

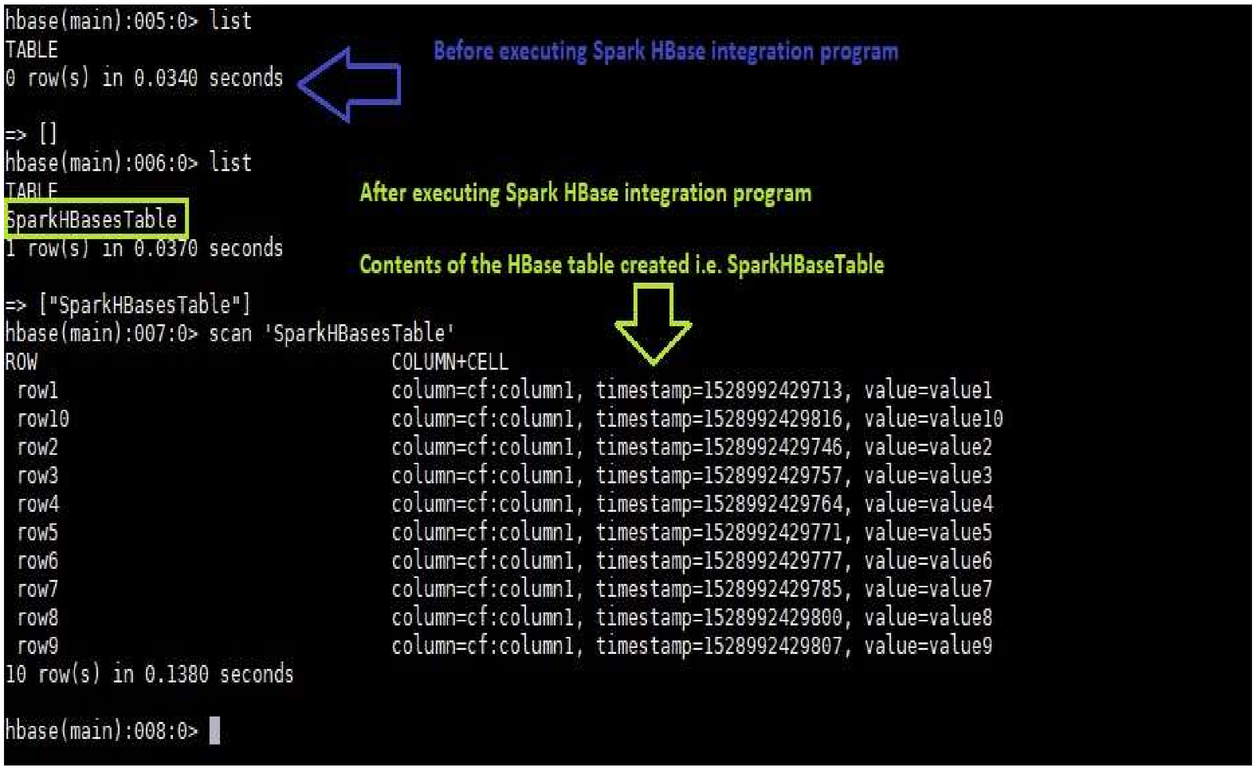
Console

```
<terminated> SparkHBaseTest$ [Scala Application] /usr/java/jdk1.8.0_151/bin/java (Jun 14, 2018, 8:00:03 PM)
2018-06-14T20:00:38,283 INFO [dag-scheduler-event-loop] org.apache.spark.scheduler.DAGScheduler - Missing parents: List()
2018-06-14T20:00:38,375 INFO [dag-scheduler-event-loop] org.apache.spark.scheduler.DAGScheduler - Submitting ResultStage 0 (M
2018-06-14T20:00:38,903 INFO [dag-scheduler-event-loop] org.apache.spark.storage.memory.MemoryStore - Block broadcast_1 store
2018-06-14T20:00:38,915 INFO [dag-scheduler-event-loop] org.apache.spark.storage.memory.MemoryStore - Block broadcast_1_piece
2018-06-14T20:00:38,917 INFO [dispatcher-event-loop-1] org.apache.spark.storage.BlockManagerInfo - Added broadcast_1_piece0 i
2018-06-14T20:00:38,923 INFO [dag-scheduler-event-loop] org.apache.spark.SparkContext - Created broadcast 1 from broadcast at
2018-06-14T20:00:39,043 INFO [dag-scheduler-event-loop] org.apache.spark.scheduler.DAGScheduler - Submitting 1 missing tasks
2018-06-14T20:00:39,070 INFO [dag-scheduler-event-loop] org.apache.spark.scheduler.TaskSchedulerImpl - Adding task set 0.0 wi
2018-06-14T20:00:39,508 INFO [dispatcher-event-loop-0] org.apache.spark.scheduler.TaskSetManager - Starting task 0.0 in stage
2018-06-14T20:00:39,626 INFO [Executor task launch worker for task 0] org.apache.spark.executor.Executor - Running task 0.0 i
2018-06-14T20:00:40,394 INFO [Executor task launch worker for task 0] org.apache.spark.rdd.NewHadoopRDD - Input split: HBase
2018-06-14T20:00:40,502 INFO [Executor task launch worker for task 0] org.apache.hadoop.hbase.zookeeper.RecoverableZooKeeper
2018-06-14T20:00:40,503 INFO [Executor task launch worker for task 0] org.apache.zookeeper.ZooKeeper - Initiating client conn
2018-06-14T20:00:40,528 INFO [Executor task launch worker for task 0-SendThread(localhost:2181)] org.apache.zookeeper.ClientC
2018-06-14T20:00:40,529 INFO [Executor task launch worker for task 0-SendThread(localhost:2181)] org.apache.zookeeper.ClientC
2018-06-14T20:00:40,534 INFO [Executor task launch worker for task 0-SendThread(localhost:2181)] org.apache.zookeeper.ClientC
2018-06-14T20:00:40,551 INFO [Executor task launch worker for task 0] org.apache.hadoop.hbase.mapreduce.TableInputFormatBase
2018-06-14T20:00:40,884 INFO [Executor task launch worker for task 0] org.apache.hadoop.hbase.client.ConnectionManager$HConne
2018-06-14T20:00:40,892 INFO [Executor task launch worker for task 0-EventThread] org.apache.zookeeper.ClientCnxn - EventThre
2018-06-14T20:00:40,893 INFO [Executor task launch worker for task 0] org.apache.zookeeper.ZooKeeper - Session: 0x163feb08a2e
2018-06-14T20:00:41,109 INFO [Executor task launch worker for task 0] org.apache.spark.executor.Executor - Finished task 0.0
2018-06-14T20:00:41,195 INFO [task-result-getter-0] org.apache.spark.scheduler.TaskSetManager - Finished task 0.0 in stage 0.
2018-06-14T20:00:41,240 INFO [task-result-getter-0] org.apache.spark.scheduler.TaskSchedulerImpl - Removed TaskSet 0.0, whose
2018-06-14T20:00:41,310 INFO [dag-scheduler-event-loop] org.apache.spark.scheduler.DAGScheduler - ResultStage 0 (count at Spa
2018-06-14T20:00:41,460 INFO [main] org.apache.spark.scheduler.DAGScheduler - Job 0 finished: count at SparkHBaseTest.scala:4
RecordCount->>102018-06-14T20:00:41,594 INFO [main] org.spark_project.jetty.server.AbstractConnector - Stopped Spark@351ff5c4
2018-06-14T20:00:41,618 INFO [main] org.apache.spark.ui.SparkUI - Stopped Spark web UI at http://10.0.3.15:4040
2018-06-14T20:00:41,903 INFO [dispatcher-event-loop-1] org.apache.spark.MapOutputTrackerMasterEndpoint - MapOutputTrackerMast
2018-06-14T20:00:41,994 INFO [main] org.apache.spark.storage.memory.MemoryStore - MemoryStore cleared
2018-06-14T20:00:41,999 INFO [main] org.apache.spark.storage.BlockManager - BlockManager stopped
2018-06-14T20:00:42,010 INFO [main] org.apache.spark.storage.BlockManagerMaster - BlockManagerMaster stopped
2018-06-14T20:00:42,047 INFO [dispatcher-event-loop-1] org.apache.spark.scheduler.OutputCommitCoordinator$OutputCommitCoordin
2018-06-14T20:00:42,059 INFO [main] org.apache.spark.SparkContext - Successfully stopped SparkContext
2018-06-14T20:00:42,107 INFO [Thread-2] org.apache.spark.util.ShutdownHookManager - Shutdown hook called
2018-06-14T20:00:42,130 INFO [Thread-2] org.apache.spark.util.ShutdownHookManager - Deleting directory /tmp/spark-55816480-1c
```

Program is executed successfully, now let's check in HBase shell:

- We can see that "SparkHBasesTable" was not there before we executed the program.
- After executing the above program, we can see Hbase table and also we can view the contents of the table.

Which are as shown below: -


Before executing Spark HBase integration program
After executing Spark HBase integration program
Contents of the HBase table created i.e. SparkHBaseTable

```
hbase(main):005:0> list
TABLE
0 row(s) in 0.0340 seconds

=> []
hbase(main):006:0> list
TABLE
SparkHBasesTable
1 row(s) in 0.0370 seconds

=> ["SparkHBasesTable"]
hbase(main):007:0> scan 'SparkHBasesTable'
ROW                                COLUMN+CELL
 row1                              column=cf:column1, timestamp=1528992429713, value=value1
 row10                             column=cf:column1, timestamp=1528992429816, value=value10
 row2                              column=cf:column1, timestamp=1528992429746, value=value2
 row3                              column=cf:column1, timestamp=1528992429757, value=value3
 row4                              column=cf:column1, timestamp=1528992429764, value=value4
 row5                              column=cf:column1, timestamp=1528992429771, value=value5
 row6                              column=cf:column1, timestamp=1528992429777, value=value6
 row7                              column=cf:column1, timestamp=1528992429785, value=value7
 row8                              column=cf:column1, timestamp=1528992429800, value=value8
 row9                              column=cf:column1, timestamp=1528992429807, value=value9
10 row(s) in 0.1380 seconds

hbase(main):008:0>
```

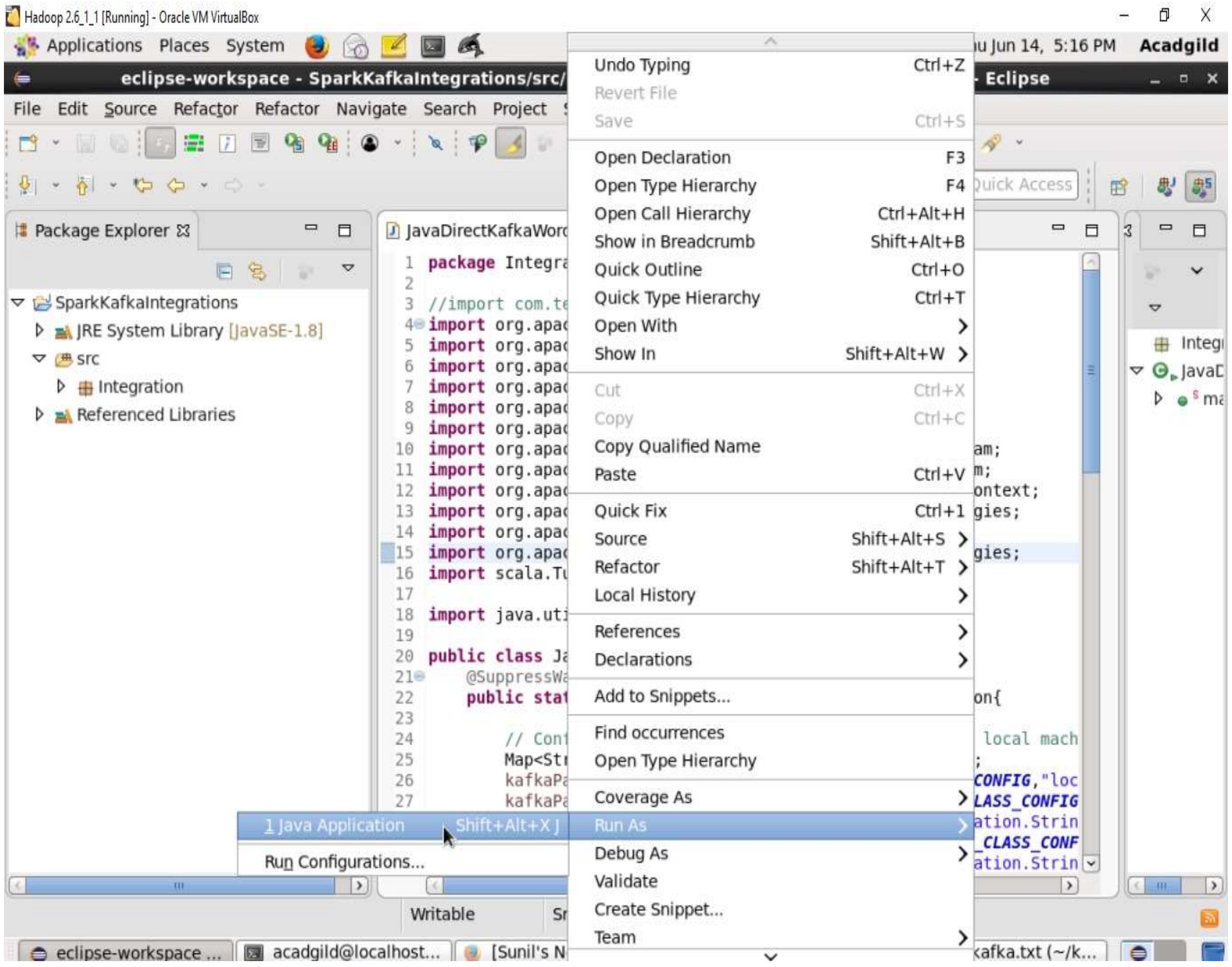
Task 3: As discussed in class integrate Spark Kafka

Program which runs the word count program by reading the contents from kafka and run in spark.

```
//imports required for the program

import com.test.schema.ContactType;
import org.apache.kafka.clients.consumer.ConsumerConfig;
import org.apache.kafka.clients.consumer.ConsumerRecord;
import org.apache.spark.SparkConf;
import org.apache.spark.api.java.JavaPairRDD;
import org.apache.spark.api.java.function.*;
import org.apache.spark.streaming.Durations;
import org.apache.spark.streaming.api.java.JavaDStream;
import org.apache.spark.streaming.api.java.JavaInputDStream;
import org.apache.spark.streaming.api.java.JavaPairDStream;
import org.apache.spark.streaming.api.java.JavaStreamingContext;
import org.apache.spark.streaming.kafka010.ConsumerStrategies;
import org.apache.spark.streaming.kafka010.KafkaUtils;
import org.apache.spark.streaming.kafka010.LocationStrategies;
import scala.Tuple2;
import java.util.*;


public class SparkKafka10 {
  public static void main(String[] argv) throws Exception{
    // Configure Spark to connect to Kafka running on local machine
    Map<String, Object> kafkaParams = new HashMap<>();

kafkaParams.put(ConsumerConfig.BOOTSTRAP_SERVERS_CONFIG,"localhost:
9092");
    kafkaParams.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG,
            "org.apache.kafka.common.serialization.StringDeserializer");
    kafkaParams.put(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG,
            "org.apache.kafka.common.serialization.StringDeserializer");
    kafkaParams.put(ConsumerConfig.GROUP_ID_CONFIG,"group1");

kafkaParams.put(ConsumerConfig.AUTO_OFFSET_RESET_CONFIG,"latest");

kafkaParams.put(ConsumerConfig.ENABLE_AUTO_COMMIT_CONFIG,true);
    //Configure Spark to listen messages in topic test
```

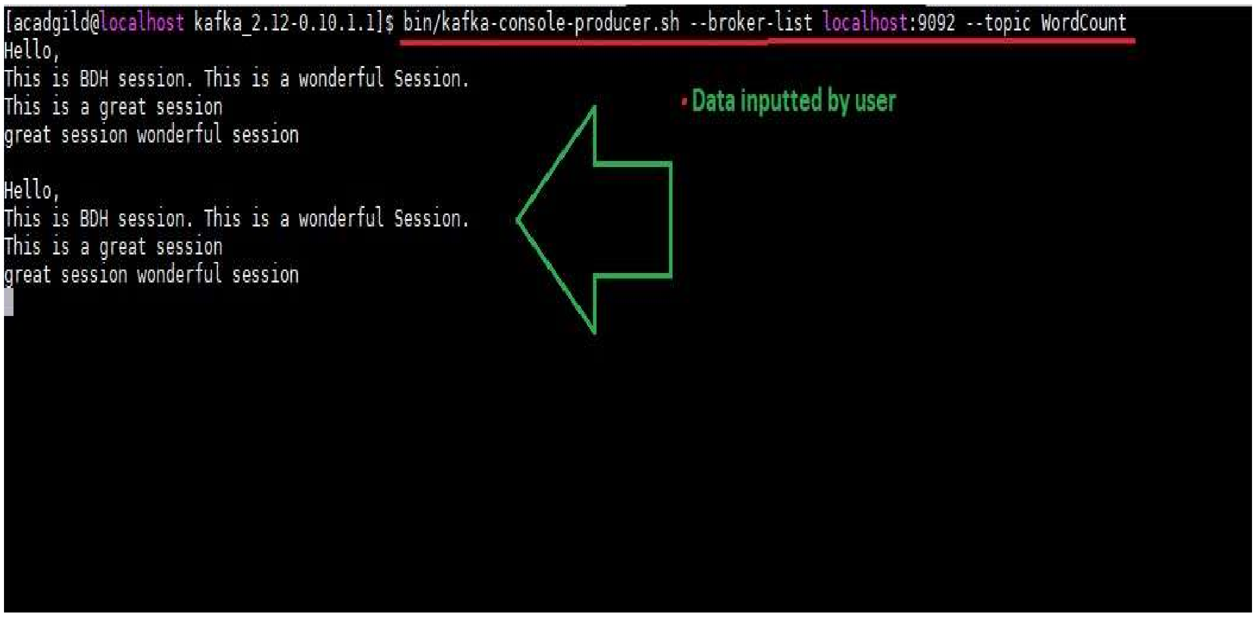
```
        Collection<String> topics = Arrays.asList("WordCount");
//setting the spark configuration with local master and setting the appname
//as "SparkKafkaWordCount"
        SparkConf conf = new
SparkConf().setMaster("local[2]").setAppName("SparkKafkaWordCount");
        //Read messages in batch of 30 seconds in realtime, by using console
//producer
        JavaStreamingContext jssc = new JavaStreamingContext(conf,
Durations.seconds(30));
        // Start reading messages from Kafka and get DStream
        final JavaInputDStream<ConsumerRecord<String, String>> stream =
                KafkaUtils.createDirectStream(jssc,
LocationStrategies.PreferConsistent(),

ConsumerStrategies.<String,String>Subscribe(topics,kafkaParams));



        // Read value of each message from Kafka and return it
        JavaDStream<String> lines = stream.map(new
Function<ConsumerRecord<String,String>, String>() {
            @Override
            public String call(ConsumerRecord<String, String> kafkaRecord) throws
Exception {
                return kafkaRecord.value();
            }
        });
        // Break every message into words and return list of words
        JavaDStream<String> words = lines.flatMap(new FlatMapFunction<String,
String>() {
            @Override
            public Iterator<String> call(String line) throws Exception {
                return Arrays.asList(line.split(" ")).iterator();
            }
        });


        // Take every word and return Tuple with (word,1)
        JavaPairDStream<String,Integer> wordMap = words.mapToPair(new
PairFunction<String, String, Integer>() {
            @Override
            public Tuple2<String, Integer> call(String word) throws Exception {
                return new Tuple2<>(word,1);
            }
        });


        // Count occurance of each word
        JavaPairDStream<String,Integer> wordCount =
wordMap.reduceByKey(new Function2<Integer, Integer, Integer>() {
            @Override
            public Integer call(Integer first, Integer second) throws Exception {
                return first+second;
            }
         });
        //Print the word count
        wordCount.print();


        jssc.start();
        jssc.awaitTermination();}}
```

We execute the program in eclipse as shown below:



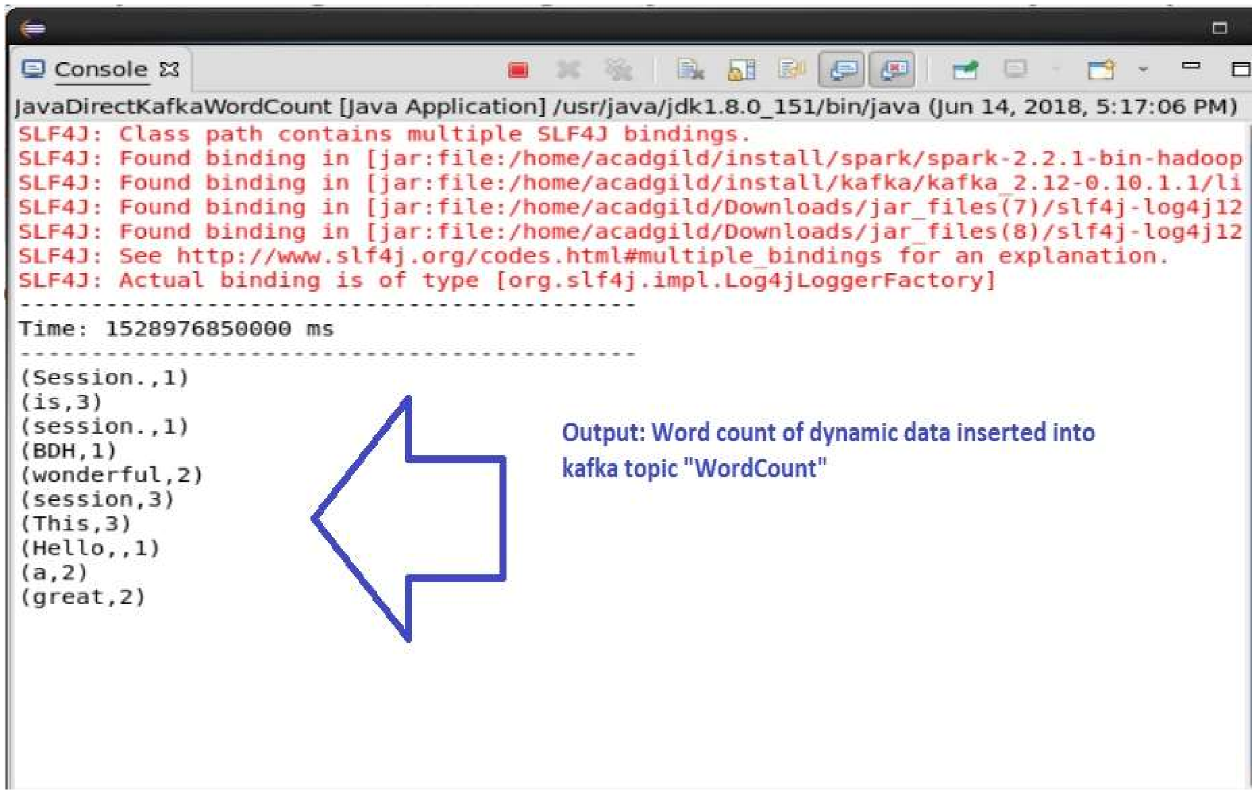
```
package Integra
//import com.te
import org.apac
import org.apac
import org.apac
import org.apac
import org.apac
import org.apac
import org.apac
import org.apac
import org.apac
import org.apac
import org.apac
import org.apac
import scala.Tu

import java.uti

public class Ja
    @SuppressWa
    public stat

        // Conf
        Map<Str
        kafkaPa
        kafkaPa
```

After executing the run command in eclipse, we open the console producer in the terminal, and input the data as shown below:



```
[acadgild@localhost kafka_2.12-0.10.1.1]$ bin/kafka-console-producer.sh --broker-list localhost:9092 --topic WordCount
Hello,
This is BDH session. This is a wonderful Session.
This is a great session
great session wonderful session

Hello,
This is BDH session. This is a wonderful Session.
This is a great session
great session wonderful session
```

After 30 seconds, we can see the word count for 30second batch of data. We can see the output in eclipse as shown below:



```
JavaDirectKafkaWordCount [Java Application] /usr/java/jdk1.8.0_151/bin/java (Jun 14, 2018, 5:17:06 PM)
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/acadgild/install/spark/spark-2.2.1-bin-hadoop
SLF4J: Found binding in [jar:file:/home/acadgild/install/kafka/kafka_2.12-0.10.1.1/li
SLF4J: Found binding in [jar:file:/home/acadgild/Downloads/jar_files(7)/slf4j-log4j12
SLF4J: Found binding in [jar:file:/home/acadgild/Downloads/jar_files(8)/slf4j-log4j12
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
-------------------------------------------
Time: 1528976850000 ms
-------------------------------------------
(Session.,1)
(is,3)
(session.,1)
(BDH,1)
(wonderful,2)
(session,3)
(This,3)
(Hello,,1)
(a,2)
(great,2)
```

Output: Word count of dynamic data inserted into kafka topic "WordCount"