

Assignment 2

Roll No : BCOB129

Configuring hive-site.xml

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration><property> <name>javax.jdo.option.ConnectionURL</name>
<value>jdbc:derby://localhost:1527/metastore_db;create=true</value>
<description>JDBC connect string for a JDBC metastore</description>
</property><property>
<name>javax.jdo.option.ConnectionDriverName</name>
<value>org.apache.derby.jdbc.ClientDriver</value>
<description>Driver class name for a JDBC metastore</description>
</property>
<property>
<name>hive.server2.enable.doAs</name>
<description>Enable user impersonation for HiveServer2</description>
<value>true</value>
</property>
<property>
<name>hive.server2.authentication</name>
<value>NONE</value>
<description> Client authentication types. NONE: no authentication
check LDAP: LDAP/AD based authentication KERBEROS: Kerberos/GSSAPI
authentication CUSTOM: Custom authentication provider (Use with
property hive.server2.custom.authentication.class) </description>
</property>
<property>
<name>datanucleus.autoCreateTables</name>
<value>True</value>
</property>
</configuration>
```

Initializing Hive Metastore

```
export HADOOP_HOME='/cygdrive/e/hadoop-env/hadoop-3.2.1'
export PATH=$PATH:$HADOOP_HOME/bin
export HIVE_HOME='/cygdrive/e/hadoop-env/apache-hive-3.1.2'
export PATH=$PATH:$HIVE_HOME/bin
export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$HIVE_HOME/lib/*.jar
```

```
1 import sys
2 from hive import ThriftHive
3 from hive.ttypes import HiveServerException
4 from thrift import Thrift
5 from thrift.transport import TSocket
6 from thrift.transport import TTransport
7 from thrift.protocol import TBinaryProtocol
8 try:
9     transport = TSocket.TSocket('localhost', 10000)
10    transport = TTransport.TBufferedTransport(transport)
11    protocol = TBinaryProtocol.TBinaryProtocol(transport)
12    client = ThriftHive.Client(protocol)
13
14    transport.open()
15    client.execute("CREATE TABLE r(a STRING, b INT, c DOUBLE)")
16    client.execute("LOAD TABLE LOCAL INPATH '/path' INTO TABLE r")
17    client.execute("SELECT * FROM r")
18    while (1):
19        row = client.fetchOne()
20        if (row == None):
21            break
22        print row
23    client.execute("SELECT * FROM r")
24    print client.fetchAll()
25    transport.close()
26 except Thrift.TException, tx:
27     print '%s' % (tx.message)
```

