Assignment 2

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
authenticationcheck LDAP: LDAP/AD based authentication
KERBEROS: Kerberos/GSSAPIauthentication 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
```

```
import sys
from hive import ThriftHive
from hive.ttypes import HiveServerException
from thrift import Thrift
from thrift.transport import TSocket
from thrift.transport import TTransport
from thrift.protocol import TBinaryProtocol
try:
transport = TSocket.TSocket('localhost', 10000)
transport = TTransport.TBufferedTransport(transport)
protocol = TBinaryProtocol.TBinaryProtocol(transport)
client = ThriftHive.Client(protocol)

transport.open()
client.execute("CREATE TABLE r(a STRING, b INT, c DOUBLE)")
client.execute("LOAD TABLE LOCAL INPATH '/path' INTO TABLE r")
client.execute("SELECT * FROM r")
while (1):
row = client.fetchOne()
if (row == None):
break
print row
client.execute("SELECT * FROM r")
print client.fetchAll()
transport.close()
except Thrift.TException, tx:
print '%s' % (tx.message)
```