# CONVERSION OF tbl FILE TO CSV AND ADDING HEADERS

import pandas as pd  
import random  
import numpy as np  
  
  
#converting tbl file to csv file  
f1 = open('/home/kaur/Downloads/ordersTiny1\_SF0.1\_0.12M.tbl', 'r')  
f2 = open('/home/kaur/Downloads/ordersTiny1\_SF1\_1.2M.csv', 'w')  
for line in f1:  
    line = line.replace(',', ' ')  
    f2.write((line.rstrip()[:-1].replace('|', ','))+"\n")  
f1.close()  
f2.close()  
  
from pandas import read\_csv  
  
df = read\_csv('/home/kaur/Downloads/ordersTiny1\_SF1\_1.2M.csv')  
df.columns = ['ORDERKEY','CUSTKEY','ORDERSTATUS','TOTALPRICE','ORDERDATE','ORDER-PRIORITY','SHIP-PRIORITY','CLERK','COMMENT']  
df.to\_csv('/home/kaur/Downloads/ordersTiny.csv',index=False)  
  
  
#####################################################################  
  
# CONVERSION OF tbl FILE TO CSV AND ADDING HEADERS

import pandas as pd  
import random  
import numpy as np  
  
  
#converting tbl file to csv file  
f1 = open('/home/kaur/Downloads/lineitemTiny1\_SF0.1\_0.55M.tbl', 'r')  
f2 = open('/home/kaur/Downloads/lineitemTiny1\_SF0.1\_0.55M.csv', 'w')  
for line in f1:  
    line = line.replace(',', ' ')  
    f2.write((line.rstrip()[:-1].replace('|', ','))+"\n")  
f1.close()  
f2.close()  
  
from pandas import read\_csv  
  
df = read\_csv('/home/kaur/Downloads/lineitemTiny1\_SF0.1\_0.55M.csv')  
df.columns = ['ORDERKEY','PARTKEY','SUPPKEY','LINENUMBER','EXTENDEDPRICE','DISCOUNT','TAX','QUANTITY','RETURNFLAG','LINESTATUS',  
'SHIPDATE','COMMITDATE','RECEIPTDATE','SHIPINSTRUCT','SHIPMODE','COMMENT']  
df.to\_csv('/home/kaur/Downloads/lineitemTiny.csv',index=False)  
  
  
##########################################################################  
  
  
#KEY VALUE STORES

import pandas as pd  
from pandas import DataFrame  
import csv  
import numpy as np  
import random  
import time  
import matplotlib.pyplot as plt  
  
# relational  
df = pd.read\_csv('/home/kaur/Downloads/ordersTiny.csv',index\_col='ORDERKEY')  
keys = []  
Values = []  
OrderList = list(df.index)  
ColumnList = list(df.columns)  
  
m = 101  
TimeR = 0.0  
Eval\_timeR = 0.0  
EvaluationTimeRel=0.0  
for y in range(1,m):  
    #start\_time = time.clock()  
    rand\_OIndex = OrderList[random.randrange(len(OrderList))]  
    rand\_CIndex = ColumnList[random.randrange(len(ColumnList))]  
    start\_time = time.clock()  
    ValuesR= df.get\_value(rand\_OIndex,rand\_CIndex)  
    TimeR = time.clock()  
    Eval\_timeR = Eval\_timeR + (TimeR - start\_time)  
  
print "Eval\_time-RDBMS",Eval\_timeR  
print "Eval\_timeR/100 -RDBMS",Eval\_timeR/100  
  
#Key Value  
for row in df.index:  
    count = 0  
    for column in df.columns:  
        key1= row.astype(str)+':'+ column  
        keys.append(key1)  
        Value= df.get\_value(row,column)  
        Values.append(Value)  
        count = count+1  
  
df1 = pd.DataFrame({})  
df2 = pd.DataFrame({})  
df3 = pd.DataFrame({})  
df1 = pd.DataFrame({'key':keys,'Value':Values},columns = ['key','Value'])  
df3 = df1.set\_index('key')['Value'].to\_dict()  
Random\_Index = rand\_OIndex.astype(str)+':'+ rand\_CIndex  
  
n = 101  
Time = 0.0  
Eval\_time = 0.0  
for x in range(1,n):  
    #start\_time = time.clock()  
    rand\_OIndex = OrderList[random.randrange(len(OrderList))]  
    rand\_CIndex = ColumnList[random.randrange(len(ColumnList))]  
    Random\_Index = rand\_OIndex.astype(str)+':'+ rand\_CIndex  
    start\_time = time.clock()  
    a = df3.get(Random\_Index)  
    Time = time.clock()  
    Eval\_time = Eval\_time + (Time - start\_time)  
print "Eval\_time-KVS",Eval\_time  
print "Eval\_time/100 -KVS", Eval\_time/100  
  
  
  
Models = ['RDBMS','Key Value Stores']  
Execution\_Time = [Eval\_timeR,Eval\_time]  
df4 = pd.DataFrame({})  
df4 = pd.DataFrame({'Models':['RDBMS','Key Value Stores',],'Execution\_Time':[Eval\_timeR,Eval\_time]},columns = ['Models','Execution\_Time'])  
ax = df4[['Models','Execution\_Time']].plot(kind='bar', title ="Execution Time comp", figsize=(15, 10), legend=True, fontsize=12)  
ax.set\_xticklabels(Models)  
ax.set\_xlabel("Models", fontsize=12)  
ax.set\_ylabel("Execution Time", fontsize=12)  
plt.show()  
  
  
  
############################################################################################  
  
#DOCUMENT STORES  
  
import pandas as pd  
import random  
import numpy as np  
import time  
import matplotlib.pyplot as plt  
  
import pandas as pd  
df = pd.read\_csv('/home/kaur/Downloads/lineitemTiny.csv')  
df1 = pd.read\_csv('/home/kaur/Downloads/ordersTiny.csv')  
OrderList = list(df1.ORDERKEY)  
ColumnList = list(df.columns)  
position = []  
Orders = []  
  
  
#Simple RDBMS Count function  
m = 101  
TimeR = 0.0  
Eval\_timeR = 0.0  
EvaluationTimeRel=0.0  
for y in range(1,m):  
    rand\_OIndex = OrderList[random.randrange(len(OrderList))]  
    count = 0  
    start\_time = time.clock()  
    for column in df.ORDERKEY:  
        if (column == rand\_OIndex):  
            count = count+1  
TimeR = time.clock()  
Eval\_timeR = Eval\_timeR + (TimeR - start\_time)  
  
print "Eval\_time-RDBMS",Eval\_timeR  
print "Eval\_timeR/100 -RDBMS",Eval\_timeR/100  
  
#Document Stores  
  
n = 101  
TimeDS = 0.0  
Eval\_timeDS = 0.0  
df3 = df.groupby('ORDERKEY')['LINENUMBER'].apply(list)  
df4 = df3.to\_frame().reset\_index()  
df6 = pd.merge(df4, df1, on='ORDERKEY', how='inner')  
#del df6['ORDERKEY']  
df5 = df4.set\_index('ORDERKEY')['LINENUMBER'].to\_dict()  
for y in range(1,n):  
    rand\_OIndex = OrderList[random.randrange(len(OrderList))]  
    start\_time = time.clock()  
    a = df6.loc[df6['ORDERKEY'] == rand\_OIndex, 'LINENUMBER']  
    length =  len(a)  
    TimeDS = time.clock()  
    #print a,length  
    Eval\_timeDS = Eval\_timeDS + (TimeDS - start\_time)  
  
print "Eval\_timeDS",Eval\_timeDS  
print "Eval\_timeDS/100",Eval\_timeDS/100  
  
Models = ['RDBMS','DocumentStores']  
Execution\_Time = [Eval\_timeR,Eval\_timeDS]  
df12 = pd.DataFrame({})  
df12 = pd.DataFrame({'Models':['RDBMS','Document Stores'],'Execution\_Time':[Eval\_timeR,Eval\_timeDS]},columns = ['Models','Execution\_Time'])  
ax = df12[['Models','Execution\_Time']].plot(kind='bar', title ="Execution Time comp", figsize=(15, 10), legend=True, fontsize=12)  
ax.set\_xticklabels(Models)  
ax.set\_xlabel("Models", fontsize=12)  
ax.set\_ylabel("Execution Time", fontsize=12)  
plt.show()  
  
  
#################################################################################