1)

SQL script:

CREATE TABLE hw10\_1(

nct\_id varchar(1000),

brief\_title varchar(1000),

sponsors\_lead\_sponsor\_agency varchar(1000),

sponsors\_lead\_sponsor\_agency\_class varchar(1000),

sponsors\_collaborator\_1\_agency varchar(1000),

sponsors\_collaborator\_1\_agency\_class varchar(1000),

sponsors\_collaborator\_2\_agency varchar(1000),

sponsors\_collaborator\_2\_agency\_class varchar(1000),

sponsors\_collaborator\_3\_agency varchar(1000),

sponsors\_collaborator\_3\_agency\_class varchar(1000),

start\_date varchar(1000),

completion\_date varchar(1000),

study\_design\_info\_observational\_model varchar(1000),

study\_design\_info\_time\_perspective varchar(1000),

primary\_outcome\_measure varchar(1000),

primary\_outcome\_time\_frame varchar(1000),

secondary\_outcome\_1\_measure varchar(1000),

secondary\_outcome\_1\_time\_frame varchar(1000),

secondary\_outcome\_2\_measure varchar(1000),

secondary\_outcome\_2\_time\_frame varchar(1000),

intervention\_type varchar(1000),

intervention\_name varchar(1000),

intervention\_description varchar(1000),

intervention\_arm\_group\_label varchar(1000),

location\_1\_facility\_name varchar(1000),

location\_1\_facility\_address\_city varchar(1000),

location\_1\_facility\_state varchar(1000),

location\_1\_facility\_zip varchar(1000),

location\_1\_facility\_country varchar(1000),

location\_2\_facility\_name varchar(1000),

location\_2\_facility\_address\_city varchar(1000),

location\_2\_facility\_state varchar(1000),

location\_2\_facility\_zip varchar(1000),

location\_2\_facility\_country varchar(1000),

location\_3\_facility\_name varchar(1000),

location\_3\_facility\_address\_city varchar(1000),

location\_3\_facility\_state varchar(1000),

location\_3\_facility\_zip varchar(1000),

location\_3\_facility\_country varchar(1000))

Python script:

import xml.etree.ElementTree as ET

from os import listdir, path

mypath = 'D:/Own Drive/all docs/сабак/Kaspi Lab Data Eng/hw10/HW\_10\_task\_1'

files = [path.join(mypath, f) for f in listdir(mypath) if f.endswith('.xml')]

totallist=[]

for file in files:

tree = ET.parse(file)

sqllist=[]

p = 0

if tree.findall('id\_info'):

for it in tree.findall('id\_info'):

sqllist.append(it.find('nct\_id').text)

else:

sqllist.append('NULL')

if tree.findall('brief\_title'):

sqllist.append(tree.find('brief\_title').text)

else:

sqllist.append('NULL')

if tree.findall('sponsors'):

for it in tree.findall('sponsors'):

if it.findall('lead\_sponsor'):

for it in it.findall('lead\_sponsor'):

sqllist.append(it.find('agency').text)

sqllist.append(it.find('agency\_class').text)

else:

sqllist.append('NULL')

sqllist.append('NULL')

else:

sqllist.append('NULL')

sqllist.append('NULL')

if tree.findall('sponsors'):

for it in tree.findall('sponsors'):

if it.findall('collaborator'):

for it in it.findall('collaborator'):

if p==6:

break

else:

pass

sqllist.append(it.find('agency').text)

sqllist.append(it.find('agency\_class').text)

p+=2

if p==2:

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

elif p==4:

sqllist.append('NULL')

sqllist.append('NULL')

else:

pass

else:

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

else:

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

if tree.findall('start\_date'):

sqllist.append(tree.find('start\_date').text)

else:

sqllist.append('NULL')

if tree.findall('completion\_date'):

sqllist.append(tree.find('completion\_date').text)

else:

sqllist.append('NULL')

if tree.findall('study\_design\_info'):

for it in tree.findall('study\_design\_info'):

sqllist.append(it.find('observational\_model').text) if it.find('observational\_model') else sqllist.append('NULL')

sqllist.append(it.find('time\_perspective').text) if it.find('time\_perspective') else sqllist.append('NULL')

else:

sqllist.append('NULL')

sqllist.append('NULL')

if tree.findall('primary\_outcome'):

for it in tree.findall('primary\_outcome'):

sqllist.append(it.find('measure').text)

sqllist.append(it.find('time\_frame').text)

break

else:

sqllist.append('NULL')

sqllist.append('NULL')

p=0

if tree.findall('secondary\_outcome'):

for it in tree.findall('secondary\_outcome'):

if p==4:

break

else:

pass

sqllist.append(it.find('measure').text)

sqllist.append(it.find('time\_frame').text)

p+=2

if p==2:

sqllist.append('NULL')

sqllist.append('NULL')

else:

pass

else:

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

if tree.findall('intervention'):

for it in tree.findall('intervention'):

sqllist.append(it.find('intervention\_type').text) if it.findall('intervention\_type') else sqllist.append('NULL')

sqllist.append(it.find('intervention\_name').text) if it.findall('intervention\_name') else sqllist.append('NULL')

sqllist.append(it.find('description').text) if it.findall('description') else sqllist.append('NULL')

sqllist.append(it.find('arm\_group\_label').text) if it.findall('arm\_group\_label') else sqllist.append('NULL')

break

else:

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

p = 0

if tree.findall('location'):

for it in tree.findall('location'):

if p==15:

break

else:

pass

if it.findall('facility'):

for it in it.findall('facility'):

sqllist.append(it.find('name').text)

if it.findall('address'):

for it in it.findall('address'):

sqllist.append(it.find('city').text) if it.findall('city') else sqllist.append('NULL')

sqllist.append(it.find('state').text) if it.findall('state') else sqllist.append('NULL')

sqllist.append(it.find('zip').text) if it.findall('zip') else sqllist.append('NULL')

sqllist.append(it.find('country').text) if it.findall('country') else sqllist.append('NULL')

p+=5

else:

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

else:

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

if p==5:

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

elif p==10:

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

else:

pass

else:

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

sqllist.append('NULL')

totallist.append(sqllist)

##########################################################################

import cx\_Oracle as orcCon

from cx\_Oracle import DatabaseError

try:

conn = orcCon.connect('C##MARFINI/250598@//localhost:1521/xe')

if conn:

print("cx\_Oracle version:", orcCon.version)

print("Database version:", conn.version)

print("Client version:", orcCon.clientversion())

cursor = conn.cursor()

print("You're connected: ")

print('Inserting data into table hw10\_1....')

for i in totallist:

query = "INSERT INTO hw10\_1 VALUES(:1,:2,:3,:4,:5,:6,:7,:8,:9,:10,:11,:12,:13,:14,:15,:16,:17,:18,:19,:20,:21,:22,:23,:24,:25,:26,:27,:28,:29,:30,:31,:32,:33,:34,:35,:36,:37,:38,:39)"

cursor.execute(query,i)

# the connection is not autocommitted by default, so we must commit to save our changes

conn.commit()

print("Record inserted succesfully")

except DatabaseError as e:

err, = e.args

print("Oracle-Error-Code:", err.code)

print("Oracle-Error-Message:", err.message)

finally:

cursor.close()

conn.close()

2)

SQL script:

CREATE TABLE hw10\_2(

table\_number varchar(100),

table\_name varchar(100),

table\_data\_row\_number varchar(100),

table\_data\_label varchar(100),

table\_data\_value\_year\_2008 varchar(100),

table\_data\_value\_year\_2009 varchar(100),

table\_data\_value\_year\_2010 varchar(100),

table\_data\_value\_year\_2011 varchar(100),

table\_data\_value\_year\_2012 varchar(100),

table\_data\_value\_year\_2013 varchar(100),

table\_data\_value\_year\_2014 varchar(100),

table\_data\_value\_year\_2015 varchar(100),

table\_data\_value\_year\_2016 varchar(100),

table\_data\_value\_year\_2017 varchar(100),

table\_data\_value\_year\_2018 varchar(100),

table\_data\_value\_year\_2019 varchar(100),

table\_data\_value\_year\_2020 varchar(100),

table\_data\_value\_year\_2021 varchar(100),

table\_data\_value\_year\_2022 varchar(100),

table\_data\_value\_year\_2023 varchar(100),

table\_data\_value\_year\_2024 varchar(100),

table\_data\_value\_year\_2025 varchar(100),

table\_data\_value\_year\_2026 varchar(100),

table\_data\_value\_year\_2027 varchar(100),

table\_data\_value\_year\_2028 varchar(100),

table\_data\_value\_year\_2029 varchar(100),

table\_data\_value\_year\_2030 varchar(100),

table\_data\_value\_year\_2031 varchar(100),

table\_data\_value\_year\_2032 varchar(100),

table\_data\_value\_year\_2033 varchar(100),

table\_data\_value\_year\_2034 varchar(100),

table\_data\_value\_year\_2035 varchar(100))

Python script:

import xml.etree.ElementTree as ET

import cx\_Oracle as orcCon

from cx\_Oracle import DatabaseError

tree = ET.parse('D:/Own Drive/all docs/сабак/Kaspi Lab Data Eng/hw10/HW\_10\_task\_2/AEO 2011 Final.xml')

root = tree.getroot()

sqllist=[]

for elem in root.iter('data'):

for elem in elem.findall('row'):

sqllist.append(elem.attrib)

try:

conn = orcCon.connect('C##MARFINI/250598@//localhost:1521/xe')

if conn:

print("cx\_Oracle version:", orcCon.version)

print("Database version:", conn.version)

print("Client version:", orcCon.clientversion())

cursor = conn.cursor()

print("You're connected: ")

print('Inserting data into table hw10\_2....')

for i in sqllist:

query = "INSERT INTO hw10\_2 VALUES(:1,:2,:3,:4,:5,:6,:7,:8,:9,:10,:11,:12,:13,:14,:15,:16,:17,:18,:19,:20,:21,:22,:23,:24,:25,:26,:27,:28,:29,:30,:31,:32)"

cursor.execute(query,i)

# the connection is not autocommitted by default, so we must commit to save our changes

conn.commit()

print("Record inserted succesfully")

except DatabaseError as e:

err, = e.args

print("Oracle-Error-Code:", err.code)

print("Oracle-Error-Message:", err.message)

finally:

cursor.close()

conn.close()