1. Auto Creating Staged Table and Importing data from Excel File to Oracle DB

In [7]:
```
# Program 1: Auto Creating Staged Table and Importing data from Excel File
# Import modules
import cx_Oracle
import pandas as pd
import os
import csv
# Set up the connection and the target database/schema
# Note: type 'sudo hostname <hostname>' in terminal to match your host name
# Close the excel files opened that are going to be used in this program wh
username = 'MASY_LC4013'
password = 'MASY_LC4013'
server_ip = 'localhost:1522'
service_name = 'app12c'
DB = 'MASY_LC4013'
# Define a function to confirm drop
def confirm_bef_drop(prompt, complaint='yes or no, please'): #complaint:key
    while True:
        confirm = input(prompt)   # In Python 2, it is called raw_input()
        if confirm in ['y', 'ye', 'yes', 'YES']:  #same as: if ok == yes or
            try:
                cur.execute('DROP TABLE ' + table)
            except:
                print('No table with the same name existed, no need to drop
            return True
        if confirm in ['n', 'no', 'nop', 'nope', 'NO']:
            raise IOError('Please change your excel name')
        print(complaint)

# Create the connection and the cursor object
con = cx_Oracle.connect(username + '/' + password + '@' + server_ip + '/' +
print('The connection to DB: %s' % con)
cur = con.cursor()
print('Cursor object created: %s' % cur)
# Iterate through all the excel files in xlsx format under the current work
# Note: This will set up the name of your staged table as the UPPER CASE of
for excel in os.listdir(os.getcwd()):
    if excel.endswith('.xlsx'):
        table = excel.split('.')[0].upper()
        print('Excel File: %s' % excel)
        print('Staged Table: %s' % table)
        df = pd.read_excel(excel)
        # Confirm with user before drop the existed table with the same nam
        confirm_bef_drop('If you currently have a table named %s, ' % table
        # Create a Staged table with the table name of 'table' and set the
        sql_create_tbl = 'CREATE TABLE ' + table + ' ('
        varchar2_size = '50'
        number_size = '38'
        lis = []
        dtypes = list(zip(df.dtypes.index, df.dtypes.values))
        for elem in dtypes:
            if str(elem[1]).find('datetime') != -1:
                lis.append(str(elem[0] + ' DATE'))
            elif str(elem[1]) == 'float64' or str(elem[1]) == 'int64':
                lis.append(str(elem[0] + ' NUMBER'+ '(' + number_size + ')'
            elif str(elem[1]) == 'object':
                lis.append(str(elem[0] + ' VARCHAR2'+ '(' + varchar2_size +
        cols_dtypes = ','.join(lis)
        sql_create_tbl = sql_create_tbl + cols_dtypes + ')'
        print('-'*100)
        print('CREATE command: %s' % sql_create_tbl)
        print('-'*100)
        df = df.where((pd.notnull(df)), None)
        cur.execute(sql_create_tbl)
#         for error in cur.getbatcherrors():
#             print("Error", error.message, "at row offset", error.offset)
        # Insert data from excel files end with xlsx to the staged table
        col_name = list(df.columns)
        sql_col_name = ','.join(col_name)
        sql_insert = 'INSERT INTO ' + table + ' (' + sql_col_name + ') VALU
        sql_values = []
        for i in range(1, len(col_name) + 1):
            sql_values.append(':' + str(i))
        sql_values = ','.join(sql_values) + ')'
        sql_insert += sql_values
        print('INSERT command: %s' % sql_insert)
        print('-'*100)
        rows = [tuple(x) for x in df.values]
        print('Data Sample: %s' % rows[0:2])
        print('-'*100)
        cur.executemany(sql_insert, rows, batcherrors=True, arraydmlrowcoun
        #Return rows affected: Oracle Client library needs to be version 12
        #rowCounts = cur.getarraydmlrowcounts()
        #for count in rowCounts:
            #print("Inserted", count, "rows.")
        for error in cur.getbatcherrors():
            print("Error", error.message, "at row offset", error.offset)
        con.commit()
        continue
    else:
        continue
cur.close()
con.close()
```

```
The connection to DB: <cx_Oracle.Connection to MASY_LC4013@localhost:15
22/app12c>
Cursor object created: <cx_Oracle.Cursor on <cx_Oracle.Connection to MA
SY_LC4013@localhost:1522/app12c>>
Excel File: HR_Data.xlsx
Staged Table: HR_DATA
If you currently have a table named HR_DATA, it will be dropped, type n
o if you don't want to do so, this program will breaks. You can re-run
this after you changed your excel file name. Otherwise, type yes.yes
No table with the same name existed, no need to drop
-----------------------------------------------------------------------
-----------------------------
CREATE command: CREATE TABLE HR_DATA (EMPLOYEE_ID NUMBER(38),FIRST_NAME
VARCHAR2(50),LAST_NAME VARCHAR2(50),EMAIL VARCHAR2(50),PHONE_NUMBER VAR
CHAR2(50),HIRE_DATE DATE,JOB_ID VARCHAR2(50),SALARY NUMBER(38),COMMISSI
ON_PCT NUMBER(38),MANAGER_ID NUMBER(38),DEPARTMENT_ID NUMBER(38),DEPART
MENT_ID_1 NUMBER(38),DEPARTMENT_NAME VARCHAR2(50),MANAGER_ID_1 NUMBER(3
8),LOCATION_ID NUMBER(38),JOB_ID_1 VARCHAR2(50),JOB_TITLE VARCHAR2(50),
MIN_SALARY NUMBER(38),MAX_SALARY NUMBER(38),EMPLOYEE_ID_1 NUMBER(38),ST
ART_DATE DATE,END_DATE DATE,JOB_ID_2 VARCHAR2(50),DEPARTMENT_ID_2 NUMBE
R(38),LOCATION_ID_1 NUMBER(38),STREET_ADDRESS VARCHAR2(50),POSTAL_CODE
NUMBER(38),CITY VARCHAR2(50),STATE_PROVINCE VARCHAR2(50),COUNTRY_ID VAR
CHAR2(50),COUNTRY_ID_1 VARCHAR2(50),COUNTRY_NAME VARCHAR2(50),REGION_ID
NUMBER(38),REGION_ID_1 NUMBER(38),REGION_NAME VARCHAR2(50))
-----------------------------------------------------------------------
-----------------------------
INSERT command: INSERT INTO HR_DATA (EMPLOYEE_ID,FIRST_NAME,LAST_NAME,E
MAIL,PHONE_NUMBER,HIRE_DATE,JOB_ID,SALARY,COMMISSION_PCT,MANAGER_ID,DEP
ARTMENT_ID,DEPARTMENT_ID_1,DEPARTMENT_NAME,MANAGER_ID_1,LOCATION_ID,JOB
_ID_1,JOB_TITLE,MIN_SALARY,MAX_SALARY,EMPLOYEE_ID_1,START_DATE,END_DAT
E,JOB_ID_2,DEPARTMENT_ID_2,LOCATION_ID_1,STREET_ADDRESS,POSTAL_CODE,CIT
Y,STATE_PROVINCE,COUNTRY_ID,COUNTRY_ID_1,COUNTRY_NAME,REGION_ID,REGION_
ID_1,REGION_NAME) VALUES (:1,:2,:3,:4,:5,:6,:7,:8,:9,:10,:11,:12,:13,:1
4,:15,:16,:17,:18,:19,:20,:21,:22,:23,:24,:25,:26,:27,:28,:29,:30,:31,:
32,:33,:34,:35)
-----------------------------------------------------------------------
-----------------------------
Data Sample: [(200.0, 'Jennifer', 'Whalen', 'JWHALEN', '515.123.4444',
Timestamp('1987-09-17 00:00:00'), 'AD_ASST', 4400.0, None, 101.0, 10.0,
10.0, 'Administration', 200.0, 1700.0, 'AD_ASST', 'Administration Assis
tant', 3000.0, 6000.0, 200.0, Timestamp('1987-09-17 00:00:00'), Timesta
mp('1993-06-17 00:00:00'), 'AD_ASST', 90.0, 1700.0, '2004 Charade Rd',
98199.0, 'Seattle', 'Washington', 'US', 'US', 'United States of Americ
a', 2.0, 2.0, 'Americas'), (200.0, 'Jennifer', 'Whalen', 'JWHALEN', '51
5.123.4444', Timestamp('1987-09-17 00:00:00'), 'AD_ASST', 4400.0, None,
101.0, 10.0, 10.0, 'Administration', 200.0, 1700.0, 'AD_ASST', 'Adminis
tration Assistant', 3000.0, 6000.0, 200.0, Timestamp('1995-09-17 00:00:
00'), Timestamp('2001-06-17 00:00:00'), 'AD_ASST', 90.0, 1700.0, '2004
Charade Rd', 98199.0, 'Seattle', 'Washington', 'US', 'US', 'United Stat
es of America', 2.0, 2.0, 'Americas')]
-----------------------------------------------------------------------
-----------------------------
```

2. Insert Data from Staged table to Relational Tables

- Functionalities not fully realized yet. Might cause mistakes in some circumstances. Will be imporved at the next update

In [8]:
```
# Program 2: 1. Auto Creating Staged Table and Importing data from Excel Fi
# Import modules
import cx_Oracle
import pandas as pd
import os
import csv
# Set up the connection and the target database/schema
# Note: type 'sudo hostname <hostname>' in terminal to match your host name
# Close the excel files opened that are going to be used in this program wh
username = 'MASY_LC4013'
password = 'MASY_LC4013'
server_ip = 'localhost:1522'
service_name = 'app12c'
DB = 'MASY_LC4013'
#
staged_tables = ['HR_DATA']
target_tables = [['LC_JOB', 'LC_JOB_HISTORY', 'LC_LOCATION', 'LC_REGION', '
insert_dic = dict(list(zip(staged_tables, target_tables)))
# Define a function to confirm drop
def confirm_bef_drop(prompt, complaint='yes or no, please'): #complaint:key
    while True:
        confirm = input(prompt)   # In Python 2, it is called raw_input()
        if confirm in ['y', 'ye', 'yes', 'YES']:  #same as: if ok == yes or
            try:
                cur.execute('DROP TABLE ' + table)
            except:
                print('No table with the same name existed, no need to drop
            return True
        if confirm in ['n', 'no', 'nop', 'nope', 'NO']:
            raise IOError('Please change your excel name')
        print(complaint)

# Create the connection and the cursor object
con = cx_Oracle.connect(username + '/' + password + '@' + server_ip + '/' +
print('The connection to DB: %s' % con)
cur = con.cursor()
print('Cursor object created: %s' % cur)
# Iterate through all the excel files in xlsx format under the current work
# Note: This will set up the name of your staged table as the UPPER CASE of
for excel in os.listdir(os.getcwd()):
    if excel.endswith('.xlsx'):
        table = excel.split('.')[0].upper()
        print('Excel File: %s' % excel)
        print('Staged Table: %s' % table)
        df = pd.read_excel(excel)
        # Confirm with user before drop the existed table with the same nam
        confirm_bef_drop('If you currently have a table named %s, ' % table
        # Create a Staged table with the table name of 'table' and set the
        sql_create_tbl = 'CREATE TABLE ' + table + ' ('
        varchar2_size = '50'
        number_size = '38'
        lis = []
        dtypes = list(zip(df.dtypes.index, df.dtypes.values))
        for elem in dtypes:
            if str(elem[1]).find('datetime') != -1:
                lis.append(str(elem[0] + ' DATE'))
            elif str(elem[1]) == 'float64' or str(elem[1]) == 'int64':
                lis.append(str(elem[0] + ' NUMBER'+ '(' + number_size + ')'
            elif str(elem[1]) == 'object':
                lis.append(str(elem[0] + ' VARCHAR2'+ '(' + varchar2_size +
        cols_dtypes = ','.join(lis)
        sql_create_tbl = sql_create_tbl + cols_dtypes + ')'
        print('-'*100)
        print('CREATE command: %s' % sql_create_tbl)
        print('-'*100)
        df = df.where((pd.notnull(df)), None)
        cur.execute(sql_create_tbl)
#           for error in cur.getbatcherrors():
#               print("Error", error.message, "at row offset", error.offset)
        # Insert data from excel files end with xlsx to the staged table
        col_name = list(df.columns)
        sql_col_name = ','.join(col_name)
        sql_insert = 'INSERT INTO ' + table + ' (' + sql_col_name + ') VALU
        sql_values = []
        for i in range(1, len(col_name) + 1):
            sql_values.append(':' + str(i))
        sql_values = ','.join(sql_values) + ')'
        sql_insert += sql_values
        print('INSERT command: %s' % sql_insert)
        print('-'*100)
        rows = [tuple(x) for x in df.values]
        cur.executemany(sql_insert, rows, batcherrors=True, arraydmlrowcoun
        #Return rows affected: Oracle Client library needs to be version 12
        #rowCounts = cur.getarraydmlrowcounts()
        #for count in rowCounts:
            #print("Inserted", count, "rows.")
        for error in cur.getbatcherrors():
            print("Error", error.message, "at row offset", error.offset)
            # Insert Data into the relational tables. The functionality is
        try:
            lis = []
            for row in cur.execute("SELECT column_name FROM USER_TAB_COLUMN
                lis.append(row[0])
            cols = ','.join(lis)
            print(cols)
            print('-'*100)
            fail = 'table not existed in dict, no target tables to insert'
            targets = insert_dic.get(table, fail)
            print(targets)
            print('-'*100)
            if targets == fail:
                con.commit()
                continue
            else:
                dic = {}
                for tbl in targets:
                    sql = "SELECT column_name FROM USER_TAB_COLUMNS WHERE t
                    lis = []
                    cur.execute(sql)
                    rows = cur.fetchall()
                    for i in rows:
                        lis.append(','.join(i))
                    dic[tbl] = lis
                print(dic)
                print('-'*100)
                for key in dic:
                    print(key)
                    print('-'*100)
                    lis = []
                    table_4pk = "'" + key + "'"
                    query_pk = "SELECT cols.column_name FROM all_constraint
                    for row in cur.execute(query_pk):
                        lis.append(','.join(row))
                    where = ' AND '.join([i + ' IS NOT NULL' for i in lis])
                    columns = ','.join(lis)
                    sql_select_stg = "SELECT DISTINCT " + ','.join(dic[key]
                    sql_insert_tbl = "INSERT INTO " + key + '(' + ','.join(
                    print(sql_insert_tbl)
                    cur.execute(sql_insert_tbl)
        except Exception as e:
            print(type(e),e)
        con.commit()
        continue
    else:
        continue
cur.close()
con.close()
```

```
The connection to DB: <cx_Oracle.Connection to MASY_LC4013@localhost:152
2/app12c>
Cursor object created: <cx_Oracle.Cursor on <cx_Oracle.Connection to MASY
_LC4013@localhost:1522/app12c>>
Excel File: HR_Data.xlsx
Staged Table: HR_DATA
If you currently have a table named HR_DATA, it will be dropped, type no
if you don't want to do so, this program will breaks. You can re-run this
after you changed your excel file name. Otherwise, type yes.yes
-------------------------------------------------------------------------
---------------------------
CREATE command: CREATE TABLE HR_DATA (EMPLOYEE_ID NUMBER(38),FIRST_NAME V
ARCHAR2(50),LAST_NAME VARCHAR2(50),EMAIL VARCHAR2(50),PHONE_NUMBER VARCHA
R2(50),HIRE_DATE DATE,JOB_ID VARCHAR2(50),SALARY NUMBER(38),COMMISSION_PC
T NUMBER(38),MANAGER_ID NUMBER(38),DEPARTMENT_ID NUMBER(38),DEPARTMENT_ID
_1 NUMBER(38),DEPARTMENT_NAME VARCHAR2(50),MANAGER_ID_1 NUMBER(38),LOCATI
ON_ID NUMBER(38),JOB_ID_1 VARCHAR2(50),JOB_TITLE VARCHAR2(50),MIN_SALARY
NUMBER(38),MAX_SALARY NUMBER(38),EMPLOYEE_ID_1 NUMBER(38),START_DATE DAT
E,END_DATE DATE,JOB_ID_2 VARCHAR2(50),DEPARTMENT_ID_2 NUMBER(38),LOCATION
_ID_1 NUMBER(38),STREET_ADDRESS VARCHAR2(50),POSTAL_CODE NUMBER(38),CITY
VARCHAR2(50),STATE_PROVINCE VARCHAR2(50),COUNTRY_ID VARCHAR2(50),COUNTRY_
ID_1 VARCHAR2(50),COUNTRY_NAME VARCHAR2(50),REGION_ID NUMBER(38),REGION_I
D_1 NUMBER(38),REGION_NAME VARCHAR2(50))
-------------------------------------------------------------------------
---------------------------
INSERT command: INSERT INTO HR_DATA (EMPLOYEE_ID,FIRST_NAME,LAST_NAME,EMA
IL,PHONE_NUMBER,HIRE_DATE,JOB_ID,SALARY,COMMISSION_PCT,MANAGER_ID,DEPARTM
ENT_ID,DEPARTMENT_ID_1,DEPARTMENT_NAME,MANAGER_ID_1,LOCATION_ID,JOB_ID_1,
JOB_TITLE,MIN_SALARY,MAX_SALARY,EMPLOYEE_ID_1,START_DATE,END_DATE,JOB_ID_
2,DEPARTMENT_ID_2,LOCATION_ID_1,STREET_ADDRESS,POSTAL_CODE,CITY,STATE_PRO
VINCE,COUNTRY_ID,COUNTRY_ID_1,COUNTRY_NAME,REGION_ID,REGION_ID_1,REGION_N
AME) 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)
--------------------------------------------------------------------------
-------------------------
EMPLOYEE_ID,FIRST_NAME,LAST_NAME,EMAIL,PHONE_NUMBER,HIRE_DATE,JOB_ID,SALA
RY,COMMISSION_PCT,MANAGER_ID,DEPARTMENT_ID,DEPARTMENT_ID_1,DEPARTMENT_NAM
E,MANAGER_ID_1,LOCATION_ID,JOB_ID_1,JOB_TITLE,MIN_SALARY,MAX_SALARY,EMPLO
YEE_ID_1,START_DATE,END_DATE,JOB_ID_2,DEPARTMENT_ID_2,LOCATION_ID_1,STREE
T_ADDRESS,POSTAL_CODE,CITY,STATE_PROVINCE,COUNTRY_ID,COUNTRY_ID_1,COUNTRY
_NAME,REGION_ID,REGION_ID_1,REGION_NAME
--------------------------------------------------------------------------
-------------------------
['LC_JOB', 'LC_JOB_HISTORY', 'LC_LOCATION', 'LC_REGION', 'LC_EMPLOYEE',
'LC_DEPARTMENT', 'LC_COUNTRY']
--------------------------------------------------------------------------
-------------------------
{'LC_JOB': ['JOB_ID', 'JOB_TITLE', 'MIN_SALARY', 'MAX_SALARY'], 'LC_JOB_H
ISTORY': ['EMPLOYEE_ID', 'START_DATE', 'END_DATE', 'DEPARTMENT_ID', 'JOB_
ID'], 'LC_LOCATION': ['LOCATION_ID', 'STREET_ADDRESS', 'POSTAL_CODE', 'CI
TY', 'STATE_PROVINCE', 'COUNTRY_ID'], 'LC_REGION': ['REGION_ID', 'REGION_
NAME'], 'LC_EMPLOYEE': ['EMPLOYEE_ID', 'FIRST_NAME', 'LAST_NAME', 'EMAI
L', 'PHONE_NUMBER', 'HIRE_DATE', 'SALARY', 'COMMISSION_PCT', 'MANAGER_I
D', 'JOB_ID', 'DEPARTMENT_ID'], 'LC_DEPARTMENT': ['DEPARTMENT_ID', 'DEPAR
TMENT_NAME', 'MANAGER_ID', 'LOCATION_ID'], 'LC_COUNTRY': ['COUNTRY_ID',
'COUNTRY_NAME', 'REGION_ID']}
--------------------------------------------------------------------------
-------------------------
LC_JOB
--------------------------------------------------------------------------
-------------------------
INSERT INTO LC_JOB(JOB_ID,JOB_TITLE,MIN_SALARY,MAX_SALARY) SELECT DISTINC
T JOB_ID,JOB_TITLE,MIN_SALARY,MAX_SALARY FROM HR_DATA WHERE JOB_ID IS NOT
NULL
LC_JOB_HISTORY
--------------------------------------------------------------------------
-------------------------
INSERT INTO LC_JOB_HISTORY(EMPLOYEE_ID,START_DATE,END_DATE,DEPARTMENT_ID,
JOB_ID) SELECT DISTINCT EMPLOYEE_ID,START_DATE,END_DATE,DEPARTMENT_ID,JOB
_ID FROM HR_DATA WHERE EMPLOYEE_ID IS NOT NULL AND START_DATE IS NOT NULL
LC_LOCATION
--------------------------------------------------------------------------
-------------------------
INSERT INTO LC_LOCATION(LOCATION_ID,STREET_ADDRESS,POSTAL_CODE,CITY,STATE
_PROVINCE,COUNTRY_ID) SELECT DISTINCT LOCATION_ID,STREET_ADDRESS,POSTAL_C
ODE,CITY,STATE_PROVINCE,COUNTRY_ID FROM HR_DATA WHERE LOCATION_ID IS NOT
NULL
LC_REGION
--------------------------------------------------------------------------
-------------------------
INSERT INTO LC_REGION(REGION_ID,REGION_NAME) SELECT DISTINCT REGION_ID,RE
GION_NAME FROM HR_DATA WHERE REGION_ID IS NOT NULL
LC_EMPLOYEE
--------------------------------------------------------------------------
-------------------------
INSERT INTO LC_EMPLOYEE(EMPLOYEE_ID,FIRST_NAME,LAST_NAME,EMAIL,PHONE_NUMB
ER,HIRE_DATE,SALARY,COMMISSION_PCT,MANAGER_ID,JOB_ID,DEPARTMENT_ID) SELEC
T DISTINCT EMPLOYEE_ID,FIRST_NAME,LAST_NAME,EMAIL,PHONE_NUMBER,HIRE_DATE,
SALARY,COMMISSION_PCT,MANAGER_ID,JOB_ID,DEPARTMENT_ID FROM HR_DATA WHERE
```

```
EMPLOYEE_ID IS NOT NULL
LC_DEPARTMENT
--------------------------------------------------------------------------
---------------------------
INSERT INTO LC_DEPARTMENT(DEPARTMENT_ID,DEPARTMENT_NAME,MANAGER_ID,LOCATI
ON_ID) SELECT DISTINCT DEPARTMENT_ID,DEPARTMENT_NAME,MANAGER_ID,LOCATION_
ID FROM HR_DATA WHERE DEPARTMENT_ID IS NOT NULL
LC_COUNTRY
--------------------------------------------------------------------------
---------------------------
INSERT INTO LC_COUNTRY(COUNTRY_ID,COUNTRY_NAME,REGION_ID) SELECT DISTINCT
COUNTRY_ID,COUNTRY_NAME,REGION_ID FROM HR_DATA WHERE COUNTRY_ID IS NOT NU
LL
```

```
In [ ]:
```