In [ ]:
```
#Install libraries for ETL
!pip install pyodbc
!pip install sqlite3
!pip install sqlalchemy
```

In [4]:
```
#Import Libraries
import sqlite3
import pyodbc
import pandas as pd
import numpy as np
from sqlalchemy import create_engine
```

In [5]:
```
#For Display
pd.set_option('display.width', 140)
pd.set_option('display.max_rows', 10000)
pd.set_option('display.max_columns', 20)
pd.set_option('display.max_colwidth', 100)
pd.options.display.width
pd.options.display.max_rows
```

Out[5]:
```
10000
```

In [6]:
```
#Create connection to the source database
dbconn = sqlite3.connect('C:/Users/mneme/Desktop/source.db')

#Extract Today's Shippers Data from the source database
shippers_extract = pd.read_sql_query("SELECT * FROM Shippers", dbconn)
shippers_extract

#Close the Source Connection, since we already extracted the data
dbconn.close()
```

In [7]:
```
#Conect to the Data Warehouse System
dwconn = sqlite3.connect('C:/Users/mneme/Desktop/datawarehouse.db')

#Establish cursor ---> needed for executing SQL statements
c = dwconn.cursor()
```

In [8]:
```
#Insert into Staging Tables (Staging tables are in the form of the Source)
#Remove existing data inside the staging table for shippers: s_shippers
delete_sshippers = c.execute('DELETE FROM S_Shippers')
```

In [9]:
```
#Insert data into S_Shippers
c.execute('''INSERT INTO S_Shippers( ShipperID, CompanyName, Phone) VALUES ('ERLJ-101', 'Speedy International', '(503) 555-9831'
c.execute('''INSERT INTO S_Shippers( ShipperID, CompanyName, Phone) VALUES ('ERLJ-102', 'United Package', '(503) 555-8712')''')
```

In [7]:
```
#Conect to the Data Warehouse System
dwconn = sqlite3.connect('C:/Users/mneme/Desktop/datawarehouse.db')

#Establish cursor ---> needed for executing SQL statements
c = dwconn.cursor()
```

In [8]:
```
#Insert into Staging Tables (Staging tables are in the form of the Source)
#Remove existing data inside the staging table for shippers: s_shippers
delete_sshippers = c.execute('DELETE FROM S_Shippers')
```

In [9]:
```
#Insert data into S_Shippers
c.execute('''INSERT INTO S_Shippers( ShipperID, CompanyName, Phone) VALUES ('ERLJ-101', 'Speedy International', '(503) 555-9831')
c.execute('''INSERT INTO S_Shippers( ShipperID, CompanyName, Phone) VALUES ('ERLJ-102', 'United Package', '(503) 555-8712')''')
c.execute('''INSERT INTO S_Shippers( ShipperID, CompanyName, Phone) VALUES ('ERLJ-103', 'Federal Shipping', '(503) 555-9931')''')
c.execute('''INSERT INTO S_Shippers( ShipperID, CompanyName, Phone) VALUES ('ERLJ-104', 'Lazaro Bulk Corporation', '(503) 555-239
c.execute('''INSERT INTO S_Shippers( ShipperID, CompanyName, Phone) VALUES ('ERLJ-105', 'Pacia Estates', '(503) 555-5564')''')
c.execute('''INSERT INTO S_Shippers( ShipperID, CompanyName, Phone) VALUES ('ERLJ-106', 'We Like to Move it', '(503) 555-1234')'
c.execute('''INSERT INTO S_Shippers( ShipperID, CompanyName, Phone) VALUES ('ERLJ-107', 'Transpo Updates', '(503) 555-6520')''')
c.execute('''INSERT INTO S_Shippers( ShipperID, CompanyName, Phone) VALUES ('ERLJ-107', 'Transpo Updates', '(503) 555-6520')''')
c.execute('''INSERT INTO S_Shippers( ShipperID, CompanyName, Phone) VALUES ('ERLJ-108', 'Log Logistics', '(503) 555-2374')''')
c.execute('''INSERT INTO S_Shippers( ShipperID, CompanyName, Phone) VALUES ('ERLJ-109', 'Fast and the Slow', '(503) 555-7712')''
c.execute('''INSERT INTO S_Shippers( ShipperID, CompanyName, Phone) VALUES ('ERLJ-110', 'Tangera Express', '(503) 555-3344')''')
c.execute('''INSERT INTO S_Shippers( ShipperID, CompanyName, Phone) VALUES ('ERLJ-111', NULL, '(503) 555-2334')''')
c.execute('''INSERT INTO S_Shippers( ShipperID, CompanyName, Phone) VALUES ('ERLJ-112', 'IDLF Lofistics', NULL)''')
c.execute('''INSERT INTO S_Shippers( ShipperID, CompanyName, Phone) VALUES ('ERLJ-113', 'United Package', '(503) 555-8712')''')
c.execute('SELECT * FROM S_Shippers')
```

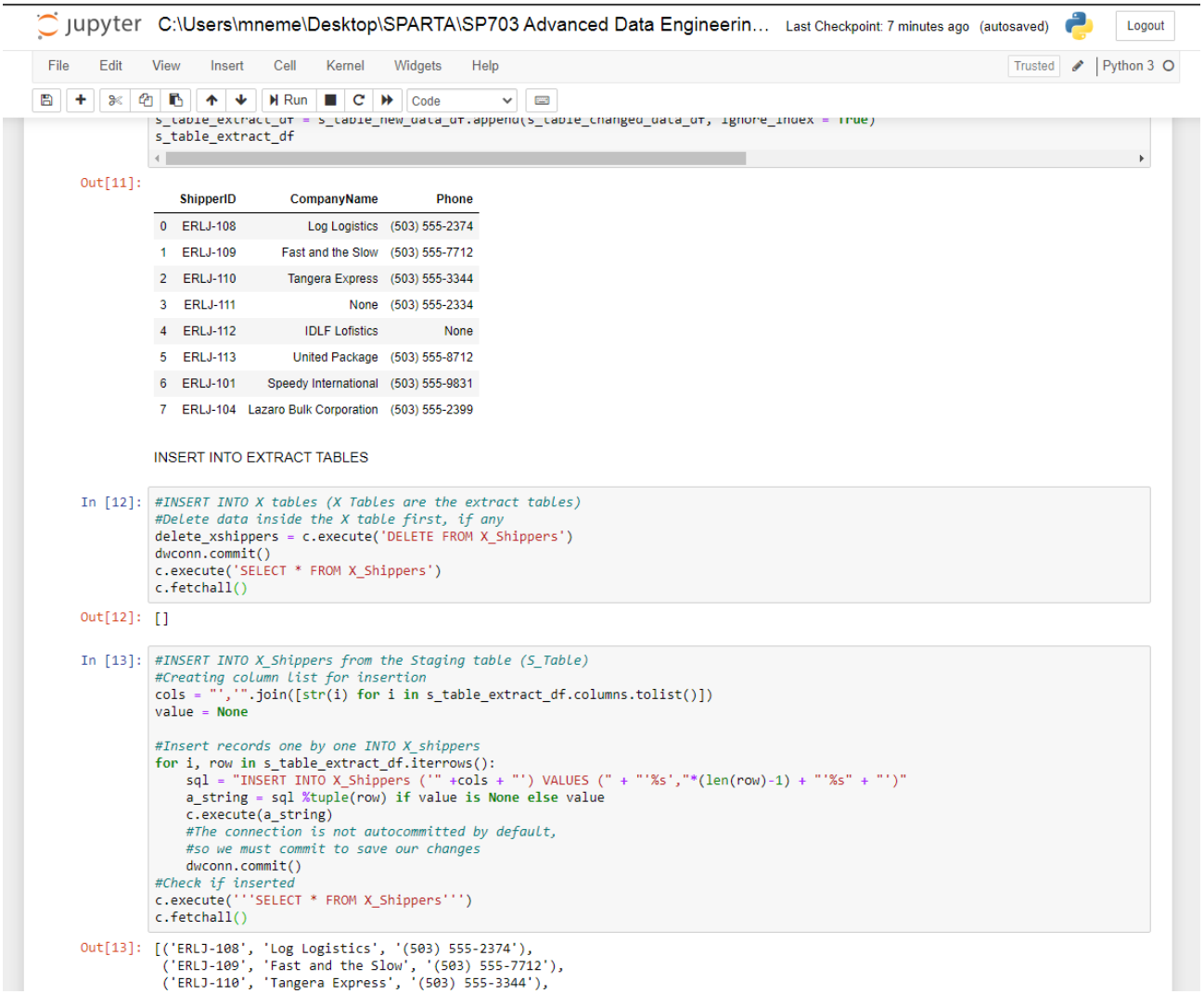
Out[9]:
```
<sqlite3.Cursor at 0x49f9ae8960>
```

In [10]:
```
#Check to see if data is already inside S_Shipper
#We use fetchall() as part of the cursor
c.fetchall()
dwconn.commit()
```

COMPARE NEW AND CHANGED DATA

In [11]:
```
#Get New and Changed Data from the Staging Table compared with the Master Data
s_table_new_data_df = pd.read_sql('''SELECT * FROM S_Shippers WHERE ShipperID NOT IN (SELECT ShipperID FROM M_Shippers)''', dwco
s_table_changed_company_name_df = pd.read_sql('''SELECT s.ShipperID, s.CompanyName, s.Phone FROM S_Shippers s INNER JOIN M_Shipp
s_table_changed_phone_number_df = pd.read_sql('''SELECT s.ShipperID, s.CompanyName, s.Phone FROM S_Shippers s INNER JOIN M_Shipp
s_table_changed_data_df = s_table_changed_company_name_df.append(s_table_changed_phone_number_df, ignore_index = True)
s_table_changed_data_df
s_table_extract_df = s_table_new_data_df.append(s_table_changed_data_df, ignore_index = True)
s_table_extract_df
```

Out[11]:


jupyter C:\Users\mneme\Desktop\SPARTA\SP703 Advanced Data Engineerin… Last Checkpoint: 7 minutes ago (autosaved) Logout
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3
Run Code
s_table_extract_df = s_table_new_data_df.append(s_table_changed_data_df, ignore_index = True)
s_table_extract_df
Out[11]:
ShipperID CompanyName Phone
0 ERLJ-108 Log Logistics (503) 555-2374
1 ERLJ-109 Fast and the Slow (503) 555-7712
2 ERLJ-110 Tangera Express (503) 555-3344
3 ERLJ-111 None (503) 555-2334
4 ERLJ-112 IDLF Lofistics None
5 ERLJ-113 United Package (503) 555-8712
6 ERLJ-101 Speedy International (503) 555-9831
7 ERLJ-104 Lazaro Bulk Corporation (503) 555-2399
INSERT INTO EXTRACT TABLES
In [12]:
#INSERT INTO X tables (X Tables are the extract tables)
#Delete data inside the X table first, if any
delete_xshippers = c.execute('DELETE FROM X_Shippers')
dwconn.commit()
c.execute('SELECT * FROM X_Shippers')
c.fetchall()
Out[12]: []
In [13]:
#INSERT INTO X_Shippers from the Staging table (S_Table)
#Creating column list for insertion
cols = "','".join([str(i) for i in s_table_extract_df.columns.tolist()])
value = None
#Insert records one by one INTO X_shippers
for i, row in s_table_extract_df.iterrows():
sql = "INSERT INTO X_Shippers ('" +cols + "') VALUES (" + "'%s'"*(len(row)-1) + "'%s" + "')"
a_string = sql %tuple(row) if value is None else value
c.execute(a_string)
#The connection is not autocommitted by default,
#so we must commit to save our changes
dwconn.commit()
#Check if inserted
c.execute('''SELECT * FROM X_Shippers''')
c.fetchall()
Out[13]: [('ERLJ-108', 'Log Logistics', '(503) 555-2374'),
('ERLJ-109', 'Fast and the Slow', '(503) 555-7712'),
('ERLJ-110', 'Tangera Express', '(503) 555-3344'),

In [13]:
```
#INSERT INTO X_Shippers from the Staging table (S_Table)
#Creating column list for insertion
cols = "','".join([str(i) for i in s_table_extract_df.columns.tolist()])
value = None

#Insert records one by one INTO X_shippers
for i, row in s_table_extract_df.iterrows():
    sql = "INSERT INTO X_Shippers ('" +cols + "') VALUES (" + "'%s'"*(len(row)-1) + "'%s" + "')"
    a_string = sql %tuple(row) if value is None else value
    c.execute(a_string)
    #The connection is not autocommitted by default,
    #so we must commit to save our changes
    dwconn.commit()
#Check if inserted
c.execute('''SELECT * FROM X_Shippers''')
c.fetchall()
```

Out[13]:
```
[('ERLJ-108', 'Log Logistics', '(503) 555-2374'),
 ('ERLJ-109', 'Fast and the Slow', '(503) 555-7712'),
 ('ERLJ-110', 'Tangera Express', '(503) 555-3344'),
 ('ERLJ-111', 'None', '(503) 555-2334'),
 ('ERLJ-112', 'IDLF Lofistics', 'None'),
 ('ERLJ-113', 'United Package', '(503) 555-8712'),
 ('ERLJ-101', 'Speedy International', '(503) 555-9831'),
 ('ERLJ-104', 'Lazaro Bulk Corporation', '(503) 555-2399')]
```

CLEAN X TABLE and INSERT INTO ERROR Table

In [17]:
```
#CLEAN X TABLE and INSERT INTO ERROR TABLE
#Select companies with null names
x_table_no_companyname_df = pd.read_sql('''SELECT * FROM X_Shippers WHERE CompanyName = 'None' ''', dwconn)
x_table_no_companyname_df['ErrorType'] = 'No Company Name'

#Select Duplicate Companies
x_table_duplicate_companies_df = pd.read_sql('''SELECT * FROM X_Shippers WHERE CompanyName IN (SELECT CompanyName FROM S_Shipper
x_table_duplicate_companies_df['ErrorType'] = 'Duplicate Company Name'
x_table_errors_df = pd.concat([x_table_no_companyname_df, x_table_duplicate_companies_df])

#Set Unknown to MIssing Phone Numbers
update_xshippers = c.execute('''UPDATE X_Shippers SET Phone = 'Unknown Phone Number' WHERE Phone = 'None' ''')
c.execute("SELECT * FROM X_Shippers")
c.fetchall()

#INSERT Uknown Phone NUmber into Error Table (E Table)
#Delete data inside E_Shipper first
delete_eshippers = c.execute('DELETE FROM E_Shippers')
c.execute("SELECT * FROM E_Shippers")
c.fetchall()
```

In [18]:
```
#Creating column list for insertion
cols = "','".join([str(i) for i in x_table_errors_df.columns.tolist()])

#Insert records one by one INTO E_shippers
for i, row in x_table_errors_df.iterrows():
    sql = "INSERT INTO E_Shippers ('" +cols + "') VALUES (" + "'%s'"*(len(row)-1) + "'%s" + "')"
    a_string = sql %tuple(row) if value is None else value
    c.execute(a_string)
    #The connection is not autocommitted by default,
    #so we must commit to save our changes
    dwconn.commit()
#Check if inserted
c.execute('''SELECT * FROM E_Shippers''')
c.fetchall()
```

Out[18]:
```
[('ERLJ-111', 'None', '(503) 555-2334', 'No Company Name'),
 ('ERLJ-113', 'United Package', '(503) 555-8712', 'Duplicate Company Name')]
```

PROCESS CLEAN DATA AND INSERT INTO C TABLE

In [20]:
```
#Process Clean Data
#Select Clean Data
x_table_clean_data_df = pd.read_sql('''SELECT * FROM X_Shippers WHERE ShipperID NOT IN (SELECT ShipperID FROM E_Shippers)''', dw

#DELETE existing data in C table
delete_cshippers = c.execute('DELETE FROM C_Shippers')
c.execute("SELECT * FROM C_Shippers")
c.fetchall()
```

Out[20]:
```
[]
```

In [22]:
```
#Actual INSERT INTO C Table
#Creating column list for insertion
cols = "','".join([str(i) for i in x_table_clean_data_df.columns.tolist()])

#Insert records one by one INTO E_shippers
for i, row in x_table_clean_data_df.iterrows():
    sql = "INSERT INTO C_Shippers ('" +cols + "') VALUES (" + "'%s'"*(len(row)-1) + "'%s" + "')"
    a_string = sql %tuple(row) if value is None else value
    c.execute(a_string)
    #The connection is not autocommitted by default,
    #so we must commit to save our changes
    dwconn.commit()
#Check if inserted
c.execute('''SELECT * FROM C_Shippers''')
c.fetchall()
```

In [22]:
```
#Actual INSERT INTO C Table
#Creating column list for insertion
cols = "','".join([str(i) for i in x_table_clean_data_df.columns.tolist()])

#Insert records one by one INTO E_shippers
for i, row in x_table_clean_data_df.iterrows():
    sql = "INSERT INTO C_Shippers ('" +cols + "') VALUES (" + "'%s'"*(len(row)-1) + "'%s" + "')"
    a_string = sql %tuple(row) if value is None else value
    c.execute(a_string)
    #The connection is not autocommitted by default,
    #so we must commit to save our changes
    dwconn.commit()
#Check if inserted
c.execute('''SELECT * FROM C_Shippers''')
c.fetchall()
```

Out[22]:
```
[('ERLJ-108', 'Log Logistics', '(503) 555-2374'),
 ('ERLJ-109', 'Fast and the Slow', '(503) 555-7712'),
 ('ERLJ-110', 'Tangera Express', '(503) 555-3344'),
 ('ERLJ-112', 'IDLF Lofistics', 'Unknown Phone Number'),
 ('ERLJ-101', 'Speedy International', '(503) 555-9831'),
 ('ERLJ-104', 'Lazaro Bulk Corporation', '(503) 555-2399')]
```

UPDATE MASTER TABLE (M TABLE)

In [24]:
```
#UPDATE M TABLE
#Select All NEW From C Tables
c_table_new_date_df = pd.read_sql('''SELECT * FROM C_Shippers c WHERE ShipperID NOT IN (SELECT m.ShipperID FROM M_Shippers m)'''

#INSERT Clean data into M Table
#Creating column list for insertion
cols = "','".join([str(i) for i in c_table_new_date_df.columns.tolist()])

#Insert records one by one INTO M_shippers
for i, row in c_table_new_date_df.iterrows():
    sql = "INSERT INTO M_Shippers_Test ('" +cols + "') VALUES (" + "'%s'"*(len(row)-1) + "'%s" + "')"
    a_string = sql %tuple(row) if value is None else value
    c.execute(a_string)
    #The connection is not autocommitted by default,
    #so we must commit to save our changes
    dwconn.commit()
#Check if inserted
c.execute('''SELECT * FROM M_Shippers_Test''')
c.fetchall()
```

Out[24]:
```
[('ERLJ-101', 'Speedy Express', '(503) 555-9831'),
 ('ERLJ-102', 'United Package', '(503) 555-8712'),
 ('ERLJ-103', 'Federal Shipping', '(503) 555-9931'),
 ('ERLJ-104', 'Lazaro Bulk Corporation', '(503) 555-2388'),
```

```
 ('ERLJ-107', 'Transpo Updates', '(503) 555-6520'),
 ('ERLJ-108', 'Log Logistics', '(503) 555-2374'),
 ('ERLJ-109', 'Fast and the Slow', '(503) 555-7712'),
 ('ERLJ-110', 'Tangera Express', '(503) 555-3344'),
 ('ERLJ-112', 'IDLF Lofistics', 'Unknown Phone Number')]
```

In [25]:
```
#Processing Changed Data and Update the master Data
#Select All Changed from C Table
c_table_changed_data_df = pd.read_sql('''SELECT c.* FROM C_Shippers c, M_Shippers m WHERE c.ShipperID = m.ShipperID
                                    AND (c.CompanyName <> m.CompanyName or c.Phone <> m.Phone)''', dwconn)

delete_mshippertest = c.execute('''DELETE FROM M_Shippers_Test
                                WHERE ShipperID IN (SELECT m.ShipperID FROM C_Shippers c, M_Shippers m
                                WHERE c.ShipperID = m.ShipperID
                                AND (c.CompanyName <> m.CompanyName or c.Phone <> m.Phone))''')

c.execute("SELECT * FROM M_Shippers_Test")
c.fetchall()
dwconn.commit()
```

In [26]:
```
#INSERT Clean data into M Table with changed data
#Creating column list for insertion
cols = "','".join([str(i) for i in c_table_changed_data_df.columns.tolist()])

#Insert records one by one INTO M_shippers
for i, row in c_table_changed_data_df.iterrows():
    sql = "INSERT INTO M_Shippers_Test ('" +cols + "') VALUES (" + "'%s'"*(len(row)-1) + "'%s" + "')"
    a_string = sql %tuple(row) if value is None else value
    c.execute(a_string)
    #The connection is not autocommitted by default,
    #so we must commit to save our changes
    dwconn.commit()
#Check if inserted
c.execute('''SELECT * FROM M_Shippers_Test''')
c.fetchall()
```

Out[26]:
```
[('ERLJ-102', 'United Package', '(503) 555-8712'),
 ('ERLJ-103', 'Federal Shipping', '(503) 555-9931'),
 ('ERLJ-105', 'Pacia Estates', '(503) 555-5564'),
 ('ERLJ-106', 'We Like to Move it', '(503) 555-1234'),
 ('ERLJ-107', 'Transpo Updates', '(503) 555-6520'),
 ('ERLJ-108', 'Log Logistics', '(503) 555-2374'),
 ('ERLJ-109', 'Fast and the Slow', '(503) 555-7712'),
 ('ERLJ-110', 'Tangera Express', '(503) 555-3344'),
 ('ERLJ-112', 'IDLF Lofistics', 'Unknown Phone Number'),
 ('ERLJ-101', 'Speedy International', '(503) 555-9831'),
 ('ERLJ-104', 'Lazaro Bulk Corporation', '(503) 555-2399')]
```

INITIATE TRANSFORM PROCESSES

INITIATE TRANSFORM PROCESSES

In [29]:
```
#Transform Processes
#Select data from C and Transform to DW Format
c_table_data_df = pd.read_sql('''SELECT ShipperID as [Shipper_ID],
                                 CompanyName as [Shipper_Name], Phone as [Current_Shipper_Phone],
                                 DATE() as [Effective_Date] FROM C_Shippers''', dwconn)
c_table_data_df['Previous_Shipper_Phone'] = "Previous_Shipper_Phone"
c_table_data_df = c_table_data_df[['Shipper_ID', 'Shipper_Name', 'Current_Shipper_Phone',
                                   'Previous_Shipper_Phone', 'Effective_Date']]
```

In [30]:
```
#INSERT INTO T Table
#DELETE existing data in T table
delete_cshippers = c.execute("SELECT * FROM T_Shipper")
c.fetchall()
dwconn.commit()
```

In [31]:
```
#Actual INSERT C Table data into T table (C_Shipper into T_Shipper)
#Creating column list for insertion
cols = "','".join([str(i) for i in c_table_data_df.columns.tolist()])

#Insert records one by one INTO M_shippers
for i, row in c_table_data_df.iterrows():
    sql = "INSERT INTO T_Shipper ('" +cols + "') VALUES (" + "'%s'"*(len(row)-1) + "'%s" + "')"
    a_string = sql %tuple(row) if value is None else value
    c.execute(a_string)
    #The connection is not autocommitted by default,
    #so we must commit to save our changes
    dwconn.commit()
#Check if inserted
pd.read_sql('''SELECT * FROM T_Shipper''', dwconn)
```

Out[31]:

	Shipper_ID	Shipper_Name	Current_Shipper_Phone	Previous_Shipper_Phone	Effective_Date
0	ERLJ-108	Log Logistics	(503) 555-2374	Previous_Shipper_Phone	2023-10-04
1	ERLJ-109	Fast and the Slow	(503) 555-7712	Previous_Shipper_Phone	2023-10-04
2	ERLJ-110	Tangera Express	(503) 555-3344	Previous_Shipper_Phone	2023-10-04
3	ERLJ-112	IDLF Lofistics	Unknown Phone Number	Previous_Shipper_Phone	2023-10-04
4	ERLJ-101	Speedy International	(503) 555-9831	Previous_Shipper_Phone	2023-10-04
5	ERLJ-104	Lazaro Bulk Corporation	(503) 555-2399	Previous_Shipper_Phone	2023-10-04

SELECT DATA FROM T TABLE AND INSERT TO I AND U TABLE

SELECT DATA FROM T TABLE AND INSERT TO I AND U TABLE

In [32]:
```
#SELECT New data from the T Table
t_table_new_data_df = pd.read_sql('''SELECT t.* FROM t_shipper t
                                LEFT JOIN d_shipper d ON t.Shipper_ID = d.Shipper_ID
                                WHERE d.Shipper_ID IS NULL''', dwconn)
t_table_new_data_df['Current_Row_Ind'] = 'Y'
```

In [33]:
```
#INSERT New data INTO I Table
#DELETE existing data in I table
delete_ishippers = c.execute('DELETE FROM I_Shipper')
c.execute("SELECT * FROM I_Shipper")
c.fetchall()

#Creating column list for insertion
cols = "','".join([str(i) for i in t_table_new_data_df.columns.tolist()])

#Insert records one by one INTO M_shippers
for i, row in t_table_new_data_df.iterrows():
    sql = "INSERT INTO I_Shipper ('" +cols + "') VALUES (" + "'%s'"*(len(row)-1) + "'%s" + "')"
    a_string = sql %tuple(row) if value is None else value
    c.execute(a_string)
    #The connection is not autocommitted by default,
    #so we must commit to save our changes
    dwconn.commit()
#Check if inserted
pd.read_sql('''SELECT * FROM I_Shipper''', dwconn)
```

Out[33]:

	Shipper_ID	Shipper_Name	Current_Shipper_Phone	Previous_Shipper_Phone	Effective_Date	Current_Row_Ind
0	ERLJ-108	Log Logistics	(503) 555-2374	Previous_Shipper_Phone	2023-10-04	Y
1	ERLJ-109	Fast and the Slow	(503) 555-7712	Previous_Shipper_Phone	2023-10-04	Y
2	ERLJ-110	Tangera Express	(503) 555-3344	Previous_Shipper_Phone	2023-10-04	Y
3	ERLJ-112	IDLF Lofistics	Unknown Phone Number	Previous_Shipper_Phone	2023-10-04	Y

In [34]:
```
#SELECT New data from the T Table
t_table_changed_data_df = pd.read_sql('''SELECT t.* FROM t_shipper t
                                inner join d_shipper d ON t.Shipper_ID = d.Shipper_ID
                                WHERE (NOT t.Shipper_Name = d.Shipper_Name or NOT t.Current_Shipper_Phone = d.Current_Shipper_Pho
                                AND d.Current_Row_Ind IN ('Y')''', dwconn)
t_table_changed_data_df['Current_Row_Ind'] = 'Y'

#DELETE existing data from the U table first
delete_ushippers = c.execute('DELETE FROM U_Shipper')
c.execute("SELECT * FROM U_Shipper")
c.fetchall()
```

2	ERLJ-110	Tangera Express	(503) 555-3344	Previous_Shipper_Phone	2023-10-04	Y
3	ERLJ-112	IDLF Lofistics	Unknown Phone Number	Previous_Shipper_Phone	2023-10-04	Y

In [34]:
```
#SELECT New data from the T Table
t_table_changed_data_df = pd.read_sql('''SELECT t.* FROM t_shipper t
                                inner join d_shipper d ON t.Shipper_ID = d.Shipper_ID
                                WHERE (NOT t.Shipper_Name = d.Shipper_Name or NOT t.Current_Shipper_Phone = d.Current_Shipper_Pho
                                AND d.Current_Row_Ind IN ('Y')''', dwconn)
t_table_changed_data_df['Current_Row_Ind'] = 'Y'

#DELETE existing data from the U table first
delete_ushippers = c.execute('DELETE FROM U_Shipper')
c.execute("SELECT * FROM U_Shipper")
c.fetchall()
```

Out[34]: []

In [35]:
```
#Actual Insert of Changed data into U table
#INSERT Changed Data INTO U
#Creating column list for insertion
cols = "','".join([str(i) for i in t_table_changed_data_df.columns.tolist()])

#Insert records one by one INTO M_shippers
for i, row in t_table_changed_data_df.iterrows():
    sql = "INSERT INTO U_Shipper ('" +cols + "') VALUES (" + "'%s'"*(len(row)-1) + "'%s" + "')"
    a_string = sql %tuple(row) if value is None else value
    c.execute(a_string)
    #The connection is not autocommitted by default,
    #so we must commit to save our changes
    dwconn.commit()
#Check if inserted
pd.read_sql('''SELECT * FROM U_Shipper''', dwconn)
```

Out[35]:

	Shipper_ID	Shipper_Name	Current_Shipper_Phone	Previous_Shipper_Phone	Effective_Date	Current_Row_Ind
0	ERLJ-101	Speedy International	(503) 555-9831	Previous_Shipper_Phone	2023-10-04	Y
1	ERLJ-104	Lazaro Bulk Corporation	(503) 555-2399	Previous_Shipper_Phone	2023-10-04	Y

INSERT I TABLE DATA INTO D TABLE

In [36]:
```
#INSERT I INTO D Table
#Get Max Warehouse Key
maxkey = pd.read_sql('''SELECT MAX(Shipper_Key) as MAX FROM D_Shipper''', dwconn)

#Select Data to be INSERTED from I Table
i_table_data_df = pd.read_sql('''SELECT * FROM I_Shipper''', dwconn)
```

In [36]:
```
#INSERT I INTO D Table
#Get Max Warehouse Key
maxkey = pd.read_sql('''SELECT MAX(Shipper_Key) as MAX FROM D_Shipper''', dwconn)

#Select Data to be INSERTED from I Table
i_table_data_df = pd.read_sql('''SELECT * FROM I_Shipper''', dwconn)

#Identify the next set of Shipper_Key's to be assigned to the New Data from I Table
i_table_data_df['Shipper_Key'] = np.arange(pd.to_numeric(maxkey.iloc[0]).values+1,
                                           (pd.to_numeric(maxkey.iloc[0].values)+len(i_table_data_df)+1))
#Rearrange according to the D table format of columns
i_table_data_df = i_table_data_df[['Shipper_Key', 'Shipper_ID', 'Shipper_Name',
                                   'Current_Shipper_Phone', 'Previous_Shipper_Phone',
                                  'Effective_Date', 'Current_Row_Ind']]
```

In [37]:
```
#Now INSERT into D Table
#Creating column list for insertion
cols = "','".join([str(i) for i in i_table_data_df.columns.tolist()])

#Insert records one by one INTO D_shippers
for i, row in i_table_data_df.iterrows():
    sql = "INSERT INTO D_Shipper ('" +cols + "') VALUES (" + "'%s'"*(len(row)-1) + "'%s" + "')"
    a_string = sql %tuple(row) if value is None else value
    c.execute(a_string)
    #The connection is not autocommitted by default,
    #so we must commit to save our changes
    dwconn.commit()
#Check if inserted
pd.read_sql('''SELECT * FROM D_Shipper''', dwconn)
```

Out[37]:

	Shipper_Key	Shipper_ID	Shipper_Name	Current_Shipper_Phone	Previous_Shipper_Phone	Effective_Date	Current_Row_Ind
0	1	ERLJ-101	Speedy Express	(503) 555-9831	Previous Phone Unknown	10/31/2014	Y
1	2	ERLJ-102	United Package	(503) 555-8712	Previous Phone Unknown	10/31/2014	Y
2	3	ERLJ-103	Federal Shipping	(503) 555-9931	Previous Phone Unknown	10/31/2014	Y
3	4	ERLJ-104	Lazaro Bulk Corporation	(503) 555-2388	Previous Phone Unknown	10/31/2014	Y
4	5	ERLJ-105	Pacia Estates	(503) 555-5564	Previous Phone Unknown	10/31/2014	Y
5	6	ERLJ-106	We Like to Move it	(503) 555-1234	Previous Phone Unknown	10/31/2014	Y
6	7	ERLJ-107	Transpo Updates	(503) 555-6520	Previous Phone Unknown	10/31/2014	Y
7	8	ERLJ-108	Log Logistics	(503) 555-2374	Previous_Shipper_Phone	2023-10-04	Y
8	9	ERLJ-109	Fast and the Slow	(503) 555-7712	Previous_Shipper_Phone	2023-10-04	Y
9	10	ERLJ-110	Tangera Express	(503) 555-3344	Previous_Shipper_Phone	2023-10-04	Y
10	11	ERLJ-112	IDLF Lofistics	Unknown Phone Number	Previous_Shipper_Phone	2023-10-04	Y

INSERT U TABLE DATA (TYPE 3) INTO D TABLE

In [38]:
```
u_table_type3_data_df = pd.read_sql('''SELECT u.* FROM U_Shipper u
                                    INNER JOIN D_Shipper d on u.Shipper_ID = d.Shipper_ID
                                    WHERE NOT (u.Current_Shipper_Phone = d.Current_Shipper_Phone)
                                    AND d.Current_Row_Ind IN ('Y')''', dwconn)
d_table_type3_data_df = pd.read_sql('''SELECT d.* FROM U_Shipper u INNER JOIN D_Shipper d
                                    on u.Shipper_ID = d.Shipper_ID
                                    WHERE NOT (u.Current_Shipper_Phone = d.Current_Shipper_Phone)
                                    AND d.Current_Row_Ind IN ('Y')''', dwconn)

u_table_type3_data_df['Previous_Shipper_Phone'] = d_table_type3_data_df['Current_Shipper_Phone']
u_table_type3_data_df['Shipper_Key'] = d_table_type3_data_df['Shipper_Key']
u_table_type3_data_df = u_table_type3_data_df[['Shipper_Key', 'Shipper_ID', 'Shipper_Name',
                                               'Current_Shipper_Phone', 'Previous_Shipper_Phone',
                                               'Effective_Date', 'Current_Row_Ind']]
pd.read_sql("SELECT * FROM D_Shipper", dwconn)
```

Out[38]:

	Shipper_Key	Shipper_ID	Shipper_Name	Current_Shipper_Phone	Previous_Shipper_Phone	Effective_Date	Current_Row_Ind
0	1	ERLJ-101	Speedy Express	(503) 555-9831	Previous Phone Unknown	10/31/2014	Y
1	2	ERLJ-102	United Package	(503) 555-8712	Previous Phone Unknown	10/31/2014	Y
2	3	ERLJ-103	Federal Shipping	(503) 555-9931	Previous Phone Unknown	10/31/2014	Y
3	4	ERLJ-104	Lazaro Bulk Corporation	(503) 555-2388	Previous Phone Unknown	10/31/2014	Y
4	5	ERLJ-105	Pacia Estates	(503) 555-5564	Previous Phone Unknown	10/31/2014	Y
5	6	ERLJ-106	We Like to Move it	(503) 555-1234	Previous Phone Unknown	10/31/2014	Y
6	7	ERLJ-107	Transpo Updates	(503) 555-6520	Previous Phone Unknown	10/31/2014	Y
7	8	ERLJ-108	Log Logistics	(503) 555-2374	Previous_Shipper_Phone	2023-10-04	Y
8	9	ERLJ-109	Fast and the Slow	(503) 555-7712	Previous_Shipper_Phone	2023-10-04	Y
9	10	ERLJ-110	Tangera Express	(503) 555-3344	Previous_Shipper_Phone	2023-10-04	Y
10	11	ERLJ-112	IDLF Lofistics	Unknown Phone Number	Previous_Shipper_Phone	2023-10-04	Y

In [40]:
```
#Now INSERT CHANGED Data (TYPE 3) from U Table into D Table
#Creating column list for insertion
cols = "','".join([str(i) for i in u_table_type3_data_df.columns.tolist()])

#Insert records one by one INTO D_shippers
for i, row in u_table_type3_data_df.iterrows():
    sql = "INSERT INTO D_Shipper ('" +cols + "') VALUES (" + "'%s',"*(len(row)-1) + "'%s" + "')"
    a_string = sql %tuple(row) if value is None else value
    c.execute(a_string)
    #The connection is not autocommitted by default,
```

In [41]:
```
#Select Type2 from U table and then update the D Table
#Get Max Warehouse Key
maxkey = pd.read_sql('''SELECT MAX(Shipper_Key) as MAX FROM D_Shipper''', dwconn)

#Select Data to be INSERTED from U Table
u_table_type2_data_df = pd.read_sql('''SELECT u.* FROM u_shipper u INNER JOIN d_shipper d
                                    ON u.Shipper_ID = d.Shipper_ID
                                    WHERE NOT (u.Shipper_Name = d.Shipper_Name)
                                    AND d.Current_Row_Ind IN ('Y')''', dwconn)

#Identify the next set of Shipper_Key's to be assigned to the New Data from I Table
u_table_type2_data_df['Shipper_Key'] = np.arange(pd.to_numeric(maxkey.iloc[0]).values+1,
                                        (pd.to_numeric(maxkey.iloc[0].values)+len(u_table_type2_data_df)+1))
#Rearrange according to the D table format of columns
u_table_type2_data_df = u_table_type2_data_df[['Shipper_Key', 'Shipper_ID', 'Shipper_Name',
                                    'Current_Shipper_Phone', 'Previous_Shipper_Phone',
                                   'Effective_Date', 'Current_Row_Ind']]
```

In [42]:
```
#Now INSERT CHANGED Data (TYPE 2) from U Table into D Table
#Creating column list for insertion
cols = "','".join([str(i) for i in u_table_type2_data_df.columns.tolist()])

#Insert records one by one INTO D_shippers
for i, row in u_table_type2_data_df.iterrows():
    sql = "INSERT INTO D_Shipper ('" +cols + "') VALUES (" + "'%s'"*(len(row)-1) + "'%s" + "')"
    a_string = sql %tuple(row) if value is None else value
    c.execute(a_string)
    #The connection is not autocommitted by default,
    #so we must commit to save our changes
    dwconn.commit()
#Check if inserted
pd.read_sql('''SELECT * FROM D_Shipper''', dwconn)
```

Out[42]:

	Shipper_Key	Shipper_ID	Shipper_Name	Current_Shipper_Phone	Previous_Shipper_Phone	Effective_Date	Current_Row_Ind
0	1	ERLJ-101	Speedy Express	(503) 555-9831	Previous Phone Unknown	10/31/2014	Y
1	2	ERLJ-102	United Package	(503) 555-8712	Previous Phone Unknown	10/31/2014	Y
2	3	ERLJ-103	Federal Shipping	(503) 555-9931	Previous Phone Unknown	10/31/2014	Y
3	4	ERLJ-104	Lazaro Bulk Corporation	(503) 555-2388	Previous Phone Unknown	10/31/2014	Y
4	5	ERLJ-105	Pacia Estates	(503) 555-5564	Previous Phone Unknown	10/31/2014	Y
5	6	ERLJ-106	We Like to Move it	(503) 555-1234	Previous Phone Unknown	10/31/2014	Y
6	7	ERLJ-107	Transpo Updates	(503) 555-6520	Previous Phone Unknown	10/31/2014	Y
7	8	ERLJ-108	Log Logistics	(503) 555-2374	Previous_Shipper_Phone	2023-10-04	Y
8	9	ERLJ-109	Fast and the Slow	(503) 555-7712	Previous_Shipper_Phone	2023-10-04	Y
9	10	ERLJ-110	Tangera Express	(503) 555-3344	Previous_Shipper_Phone	2023-10-04	Y

```
cols = ",".join([str(i) for i in u_table_type2_data_df.columns.tolist()])

#Insert records one by one INTO D_shippers
for i, row in u_table_type2_data_df.iterrows():
    sql = "INSERT INTO D_Shipper (" +cols + ") VALUES (" + "'%s'"*(len(row)-1) + "'%s" + "')"
    a_string = sql %tuple(row) if value is None else value
    c.execute(a_string)
    #The connection is not autocommitted by default,
    #so we must commit to save our changes
    dwconn.commit()
#Check if inserted
pd.read_sql('''SELECT * FROM D_Shipper''', dwconn)
```

Out[42]:

	Shipper_Key	Shipper_ID	Shipper_Name	Current_Shipper_Phone	Previous_Shipper_Phone	Effective_Date	Current_Row_Ind
0	1	ERLJ-101	Speedy Express	(503) 555-9831	Previous Phone Unknown	10/31/2014	Y
1	2	ERLJ-102	United Package	(503) 555-8712	Previous Phone Unknown	10/31/2014	Y
2	3	ERLJ-103	Federal Shipping	(503) 555-9931	Previous Phone Unknown	10/31/2014	Y
3	4	ERLJ-104	Lazaro Bulk Corporation	(503) 555-2388	Previous Phone Unknown	10/31/2014	Y
4	5	ERLJ-105	Pacia Estates	(503) 555-5564	Previous Phone Unknown	10/31/2014	Y
5	6	ERLJ-106	We Like to Move it	(503) 555-1234	Previous Phone Unknown	10/31/2014	Y
6	7	ERLJ-107	Transpo Updates	(503) 555-6520	Previous Phone Unknown	10/31/2014	Y
7	8	ERLJ-108	Log Logistics	(503) 555-2374	Previous_Shipper_Phone	2023-10-04	Y
8	9	ERLJ-109	Fast and the Slow	(503) 555-7712	Previous_Shipper_Phone	2023-10-04	Y
9	10	ERLJ-110	Tangera Express	(503) 555-3344	Previous_Shipper_Phone	2023-10-04	Y
10	11	ERLJ-112	IDLF Lofistics	Unknown Phone Number	Previous_Shipper_Phone	2023-10-04	Y
11	12	ERLJ-101	Speedy International	(503) 555-9831	Previous_Shipper_Phone	2023-10-04	Y

UPDATE INDICAORS TO CURRENT IN D TABLE

In [43]:
```
#Update Current Indicators in D Table
#Update Shipper Current Value
for_update_data_df = pd.read_sql('''SELECT d.* FROM u_shipper u INNER JOIN d_shipper d ON
                                u.Shipper_Id = d.Shipper_Id WHERE d.Current_Row_Ind = 'Y'
                                AND NOT u.Shipper_Name = d.Shipper_Name''', dwconn)
for_update_data_df['Current_Row_Ind'] = 'N'

#Delete from D Shipper first
delete_dshipper = c.execute('''DELETE FROM d_shipper WHERE Shipper_Key
                            IN (SELECT d.Shipper_Key FROM u_shipper u INNER JOIN d_shipper d ON
                            u.Shipper_Id = d.Shipper_Id
                            WHERE d.Current_Row_Ind = 'Y'
                            AND Not u.Shipper_Name = d.Shipper_Name)''')
```

```
                                u.Shipper_Id = d.Shipper_Id WHERE d.Current_Row_Ind = 'Y'
                                AND NOT u.Shipper_Name = d.Shipper_Name''', dwconn)
for_update_data_df['Current_Row_Ind'] = 'N'

#Delete from D Shipper first
delete_dshipper = c.execute('''DELETE FROM d_shipper WHERE Shipper_Key
                            IN (SELECT d.Shipper_Key FROM u_shipper u INNER JOIN d_shipper d ON
                            u.Shipper_Id = d.Shipper_Id
                            WHERE d.Current_Row_Ind = 'Y'
                            AND Not u.Shipper_Name = d.Shipper_Name)''')
```

In [44]:

```
#Actual UPDATE of current indicator
#Creating column list for insertion
cols = "','".join([str(i) for i in for_update_data_df.columns.tolist()])

#Insert records one by one INTO D_shippers
for i, row in for_update_data_df.iterrows():
    sql = "INSERT INTO D_Shipper ('" +cols + "') VALUES (" + "'%s'"*(len(row)-1) + "'%s" + "')"
    a_string = sql %tuple(row) if value is None else value
    c.execute(a_string)
    #The connection is not autocommitted by default,
    #so we must commit to save our changes
    dwconn.commit()
#Check if inserted
pd.read_sql('''SELECT * FROM D_Shipper''', dwconn)
```

Out[44]:

	Shipper_Key	Shipper_ID	Shipper_Name	Current_Shipper_Phone	Previous_Shipper_Phone	Effective_Date	Current_Row_Ind
0	1	ERLJ-101	Speedy Express	(503) 555-9831	Previous Phone Unknown	10/31/2014	N
1	2	ERLJ-102	United Package	(503) 555-8712	Previous Phone Unknown	10/31/2014	Y
2	3	ERLJ-103	Federal Shipping	(503) 555-9931	Previous Phone Unknown	10/31/2014	Y
3	4	ERLJ-104	Lazaro Bulk Corporation	(503) 555-2388	Previous Phone Unknown	10/31/2014	Y
4	5	ERLJ-105	Pacia Estates	(503) 555-5564	Previous Phone Unknown	10/31/2014	Y
5	6	ERLJ-106	We Like to Move it	(503) 555-1234	Previous Phone Unknown	10/31/2014	Y
6	7	ERLJ-107	Transpo Updates	(503) 555-6520	Previous Phone Unknown	10/31/2014	Y
7	8	ERLJ-108	Log Logistics	(503) 555-2374	Previous_Shipper_Phone	2023-10-04	Y
8	9	ERLJ-109	Fast and the Slow	(503) 555-7712	Previous_Shipper_Phone	2023-10-04	Y
9	10	ERLJ-110	Tangera Express	(503) 555-3344	Previous_Shipper_Phone	2023-10-04	Y
10	11	ERLJ-112	IDLF Lofistics	Unknown Phone Number	Previous_Shipper_Phone	2023-10-04	Y
11	12	ERLJ-101	Speedy International	(503) 555-9831	Previous_Shipper_Phone	2023-10-04	Y

In [ ]: