ASSIGNMENT 3

EXTRACT,TRANSFORM,LOAD

NAHUSHA ACHARYA PES1201700044

KEERTHAN G PES1201700963

PUSHPENDAR SINGH PES1201700243

ETL is short for ***e****xtract,* ***t****ransform,* ***l****oad*, three database functions that are combined into one tool to pull data out of one database and place it into another database.

- **Extract** is the process of *reading data* from a database. In this stage, the data is collected, often from multiple and different types of sources.
- **Transform** is the process of *converting the extracted data* from its previous form into the form it needs to be in so that it can be placed into another database. Transformation occurs by using rules or lookup tables or by combining the data with other data.
- **Load** is the process of *writing the data* into the target database.

Code snippet:

```
import pandas as pd #import pandas library

############################ Concatenate Data Frames
#############################################################
dummy_data1 = {
        'id': ['1', '2', '3', '4', '5'],
        'Feature1': ['A', 'C', 'E', 'G', 'I'],
        'Feature2': ['B', 'D', 'F', 'H', 'J']}
df1 = pd.DataFrame(dummy_data1, columns = ['id', 'Feature1', 'Feature2'])
dummy_data2 = {
        'id': ['1', '2', '6', '7', '8'],
        'Feature1': ['K', 'M', 'O', 'Q', 'S'],
```

```
        'Feature2': ['L', 'N', 'P', 'R', 'T']}
df2 = pd.DataFrame(dummy_data2, columns = ['id', 'Feature1', 'Feature2'])
#print(df1)
#print(df2)
dummy_data3 = {
        'id': ['1', '2', '3', '4', '5', '7', '8', '9', '10', '11'],
        'Feature3': [12, 13, 14, 15, 16, 17, 15, 12, 13, 23]}
#df3 = pd.DataFrame(dummy_data3, columns = ['id', 'Feature3'])
#print(df3)
#concatenate on rows
df_row = pd.concat([df1, df2],ignore_index=True)
#print(df_row)
#label concatenated data frames
frames = [df1,df2]
df_keys = pd.concat(frames, keys=['x', 'y'])
#print(df_keys)
#check y dataframe of the concatenated data
#df_keys.loc['y']

#concatenate dataframes over column
df_col = pd.concat([df1,df2], axis=1)
#df_col

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


#----------------------------------Merge Data Frames ---------------------------------
#
df_merge_col = pd.merge(df_row, df3, on='id')
#df_merge_col
df_merge_difkey = pd.merge(df_row, df3, left_on='id', right_on='id')
#df_merge_difkey
#append rows if you want using append() function
add_row = pd.Series(['10', 'X1', 'X2', 'X3'],
                    index=['id','Feature1', 'Feature2', 'Feature3'])

df_add_row = df_merge_col.append(add_row, ignore_index=True)
```

```
#--------------------------Integration using SQL Operations-------------------------
----------------------
#Full Outer Join
df_outer = pd.merge(df1, df2, on='id', how='outer')
#df_outer #NaN indicates missing values
#Full Inner  Join
df_inner = pd.merge(df1, df2, on='id', how='inner')
df_inner #joined result of matched parameters
#right join
df_right = pd.merge(df1, df2, on='id', how='right')
#df_right

#left join
df_left = pd.merge(df1, df2, on='id', how='left')
#df_left

#joining on index
df_index = pd.merge(df1, df2, right_index=True, left_index=True)
df_index
#-----------------------------------------------------------------------------------------
-------------
trades = pd.DataFrame({
    'time': pd.to_datetime(['20160525 13:30:00.023',
                            '20160525 13:30:00.038',
                            '20160525 13:30:00.048',
                            '20160525 13:30:00.048',
                            '20160525 13:30:00.048']),
    'ticker': ['MSFT', 'MSFT','GOOG', 'GOOG', 'AAPL'],
    'price': [51.95, 51.95,720.77, 720.92, 98.00],
    'quantity': [75, 155,100, 100, 100]},
    columns=['time', 'ticker', 'price', 'quantity'])

quotes = pd.DataFrame({
    'time': pd.to_datetime(['20160525 13:30:00.023',
                            '20160525 13:30:00.023',
                            '20160525 13:30:00.030',
                            '20160525 13:30:00.041',
                            '20160525 13:30:00.048',
                            '20160525 13:30:00.049',
                            '20160525 13:30:00.072',
```

```
                        '20160525 13:30:00.075']),
    'ticker': ['GOOG', 'MSFT', 'MSFT','MSFT', 'GOOG', 'AAPL',
'GOOG','MSFT'],
    'bid': [720.50, 51.95, 51.97, 51.99,720.50, 97.99, 720.50, 52.01],
    'ask': [720.93, 51.96, 51.98, 52.00,720.93, 98.01, 720.88, 52.03]},
    columns=['time', 'ticker', 'bid', 'ask'])
#print(trades)
#print(quotes)
df_merge_asof = pd.merge_asof(trades, quotes,
            on='time',
            by='ticker')
df_merge_asof
```

OUTPUT: