etl

December 2, 2022

1 ETL Processes

Use this notebook to develop the ETL process for each of your tables before completing the `etl.py` file to load the whole datasets.

```
In [1]: import os
        import glob
        import psycopg2
        import pandas as pd
        from sql_queries import *
```

```
In [2]: conn = psycopg2.connect("host=127.0.0.1 dbname=sparkifydb user=student password=student"
        cur = conn.cursor()
```

```
In [3]: def get_files(filepath):
            all_files = []
            for root, dirs, files in os.walk(filepath):
                print(root)
                files = glob.glob(os.path.join(root,'*.json'))

                for f in files :
                    all_files.append(os.path.abspath(f))

            return all_files
```

```
In [ ]:
```

2 Process `song_data`

In this first part, you'll perform ETL on the first dataset, `song_data`, to create the `songs` and `artists` dimensional tables.

Let's perform ETL on a single song file and load a single record into each table to start. - Use the `get_files` function provided above to get a list of all song JSON files in `data/song_data` - Select the first song in this list - Read the song file and view the data

```
In [4]: song_files = get_files('data/song_data')
```

```
data/song_data
data/song_data/A
data/song_data/A/A
data/song_data/A/A/A
data/song_data/A/A/B
data/song_data/A/A/.ipynb_checkpoints
data/song_data/A/A/C
data/song_data/A/B
data/song_data/A/B/A
data/song_data/A/B/B
data/song_data/A/B/.ipynb_checkpoints
data/song_data/A/B/C
data/song_data/A/.ipynb_checkpoints
data/song_data/.ipynb_checkpoints
```

```
In [5]: filepath = song_files[0]
        filepath
```

```
Out[5]: '/home/workspace/data/song_data/A/A/A/TRAAAAW128F429D538.json'
```

```
In [6]: ! cat '/home/workspace/data/song_data/A/A/A/TRAAAAW128F429D538.json'
```

```
{"num_songs": 1, "artist_id": "ARD7TVE1187B99BFB1", "artist_latitude": null, "artist_longitude":
```

```
In [7]: df = pd.read_json(filepath, lines=True)
        df.head()
```

```
Out[7]:            artist_id  artist_latitude  artist_location  artist_longitude  \
        0  ARD7TVE1187B99BFB1              NaN  California - LA               NaN

          artist_name   duration  num_songs             song_id             title  \
        0      Casual  218.93179          1  SOMZWCG12A8C13C480  I Didn't Mean To

           year
        0     0
```

```
In [8]: df.columns
```

```
Out[8]: Index(['artist_id', 'artist_latitude', 'artist_location', 'artist_longitude',
               'artist_name', 'duration', 'num_songs', 'song_id', 'title', 'year'],
              dtype='object')
```

2.1 #1: songs Table

Extract Data for Songs Table

- Select columns for song ID, title, artist ID, year, and duration
- Use `df.values` to select just the values from the dataframe
- Index to select the first (only) record in the dataframe

- Convert the array to a list and set it to `song_data`

```
In [9]: song_data = df[['song_id', 'title', 'artist_id', 'year', 'duration']].values[0].tolist()
        song_data
```

```
Out[9]: ['SOMZWCG12A8C13C480', "I Didn't Mean To", 'ARD7TVE1187B99BFB1', 0, 218.93179]
```

Insert Record into Song Table Implement the `song_table_insert` query in `sql_queries.py` and run the cell below to insert a record for this song into the `songs` table. Remember to run `create_tables.py` before running the cell below to ensure you've created/resetted the `songs` table in the sparkify database.

```
In [10]: cur.execute(song_table_insert, song_data)
         conn.commit()
```

Run `test.ipynb` to see if you've successfully added a record to this table.

2.2 #2: `artists` Table

Extract Data for Artists Table

- Select columns for artist ID, name, location, latitude, and longitude
- Use `df.values` to select just the values from the dataframe
- Index to select the first (only) record in the dataframe
- Convert the array to a list and set it to `artist_data`

```
In [11]: artist_data = df[['artist_id', 'artist_name', 'artist_location', 'artist_latitude', 'ar
         artist_data
```

```
Out[11]: ['ARD7TVE1187B99BFB1', 'Casual', 'California - LA', nan, nan]
```

Insert Record into Artist Table Implement the `artist_table_insert` query in `sql_queries.py` and run the cell below to insert a record for this song's artist into the `artists` table. Remember to run `create_tables.py` before running the cell below to ensure you've created/resetted the `artists` table in the sparkify database.

```
In [12]: cur.execute(artist_table_insert, artist_data)
         conn.commit()
```

Run `test.ipynb` to see if you've successfully added a record to this table.

3 Process `log_data`

In this part, you'll perform ETL on the second dataset, `log_data`, to create the `time` and `users` dimensional tables, as well as the `songplays` fact table.

Let's perform ETL on a single log file and load a single record into each table. - Use the `get_files` function provided above to get a list of all log JSON files in `data/log_data` - Select the first log file in this list - Read the log file and view the data

```
In [13]: log_files = get_files('data/log_data')

data/log_data
data/log_data/2018
data/log_data/2018/.ipynb_checkpoints
data/log_data/2018/11
data/log_data/2018/11/.ipynb_checkpoints


In [14]: filepath = log_files[0]

In [15]: df = pd.read_json(filepath, lines=True)
         df.head(10)

Out[15]:                       artist       auth firstName gender  itemInSession  \
         0           Stephen Lynch  Logged In    Jayden      M              0
         1                 Manowar  Logged In     Jacob      M              0
         2               Morcheeba  Logged In     Jacob      M              1
         3               Maroon 5  Logged In     Jacob      M              2
         4                   Train  Logged In     Jacob      M              3
         5                   LMFAO  Logged In     Jacob      M              4
         6                DJ Dizzy  Logged In     Jacob      M              5
         7  Fish Go Deep & Tracey K  Logged In     Jacob      M              6
         8                    None  Logged In    Alivia      F              0
         9                     M83  Logged In     Jacob      M              7

           lastName     length level                            location method  \
         0     Bell  182.85669  free      Dallas-Fort Worth-Arlington, TX    PUT
         1    Klein  247.56200  paid  Tampa-St. Petersburg-Clearwater, FL    PUT
         2    Klein  257.41016  paid  Tampa-St. Petersburg-Clearwater, FL    PUT
         3    Klein  231.23546  paid  Tampa-St. Petersburg-Clearwater, FL    PUT
         4    Klein  216.76363  paid  Tampa-St. Petersburg-Clearwater, FL    PUT
         5    Klein  227.99628  paid  Tampa-St. Petersburg-Clearwater, FL    PUT
         6    Klein  221.15220  paid  Tampa-St. Petersburg-Clearwater, FL    PUT
         7    Klein  377.41669  paid  Tampa-St. Petersburg-Clearwater, FL    PUT
         8  Terrell        NaN  free             Parkersburg-Vienna, WV    GET
         9    Klein   96.18240  paid  Tampa-St. Petersburg-Clearwater, FL    PUT

                page  registration  sessionId  \
         0  NextSong  1.540992e+12        829
         1  NextSong  1.540558e+12       1049
         2  NextSong  1.540558e+12       1049
         3  NextSong  1.540558e+12       1049
         4  NextSong  1.540558e+12       1049
         5  NextSong  1.540558e+12       1049
         6  NextSong  1.540558e+12       1049
         7  NextSong  1.540558e+12       1049
         8      Home  1.540505e+12       1070
         9  NextSong  1.540558e+12       1049
```

```
                                      song  status             ts  \
0                       Jim Henson's Dead     200  1543537327796
1                             Shell Shock     200  1543540121796
2     Women Lose Weight (Feat: Slick Rick)     200  1543540368796
3                  Won't Go Home Without You     200  1543540625796
4                           Hey_ Soul Sister     200  1543540856796
5                      I'm In Miami Bitch     200  1543541072796
6                              Sexy Bitch     200  1543541299796
7  The Cure & The Cause (Dennis Ferrer Remix)     200  1543541520796
8                                    None     200  1543541644796
9                            Staring At Me     200  1543541897796

                                         userAgent userId
0  Mozilla/5.0 (compatible; MSIE 10.0; Windows NT...     91
1  "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4...     73
2  "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4...     73
3  "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4...     73
4  "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4...     73
5  "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4...     73
6  "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4...     73
7  "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4...     73
8  "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebK...      4
9  "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4...     73
```

3.1 #3: `time` Table

Extract Data for Time Table

- Filter records by `NextSong` action
- Convert the `ts` timestamp column to datetime
- Hint: the current timestamp is in milliseconds
- Extract the timestamp, hour, day, week of year, month, year, and weekday from the `ts` column and set `time_data` to a list containing these values in order
- Hint: use pandas' `dt` attribute to access easily datetimelike properties.
- Specify labels for these columns and set to `column_labels`
- Create a dataframe, `time_df`, containing the time data for this file by combining `column_labels` and `time_data` into a dictionary and converting this into a dataframe

```
In [16]: df = df[(df.page == 'NextSong') & (df.level == "free")]
         df.head()
```

```
Out[16]:          artist       auth firstName gender  itemInSession lastName  \
         0   Stephen Lynch  Logged In    Jayden      M              0     Bell
         23   Jack Johnson  Logged In     Aiden      M              1     Hess
         24  Iron And Wine  Logged In     Aiden      M              2     Hess
         25         The xx  Logged In     Aiden      M              3     Hess
         26    The Antlers  Logged In     Aiden      M              4     Hess
```

```
       length level                          location method      page  \
0   182.85669  free  Dallas-Fort Worth-Arlington, TX    PUT  NextSong
23  240.06485  free        La Crosse-Onalaska, WI-MN    PUT  NextSong
24  153.05098  free        La Crosse-Onalaska, WI-MN    PUT  NextSong
25  158.24934  free        La Crosse-Onalaska, WI-MN    PUT  NextSong
26  328.88118  free        La Crosse-Onalaska, WI-MN    PUT  NextSong

    registration  sessionId              song  status             ts  \
0   1.540992e+12        829  Jim Henson's Dead     200  1543537327796
23  1.540829e+12        986             Taylor     200  1543547190796
24  1.540829e+12        986    Naked As We Can     200  1543547430796
25  1.540829e+12        986            Fantasy     200  1543547583796
26  1.540829e+12        986           Epilogue     200  1543547741796

                                            userAgent userId
0   Mozilla/5.0 (compatible; MSIE 10.0; Windows NT...     91
23  "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4...     86
24  "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4...     86
25  "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4...     86
26  "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4...     86
```

```
In [17]: df['ts'] = pd.to_datetime(df['ts'], unit='ms')
         t = df.copy()
         t.head()
```

```
Out[17]:           artist       auth firstName gender  itemInSession lastName  \
         0   Stephen Lynch  Logged In    Jayden      M              0     Bell
         23   Jack Johnson  Logged In     Aiden      M              1     Hess
         24  Iron And Wine  Logged In     Aiden      M              2     Hess
         25         The xx  Logged In     Aiden      M              3     Hess
         26    The Antlers  Logged In     Aiden      M              4     Hess

                length level                          location method      page  \
         0   182.85669  free  Dallas-Fort Worth-Arlington, TX    PUT  NextSong
         23  240.06485  free        La Crosse-Onalaska, WI-MN    PUT  NextSong
         24  153.05098  free        La Crosse-Onalaska, WI-MN    PUT  NextSong
         25  158.24934  free        La Crosse-Onalaska, WI-MN    PUT  NextSong
         26  328.88118  free        La Crosse-Onalaska, WI-MN    PUT  NextSong

             registration  sessionId              song  status  \
         0   1.540992e+12        829  Jim Henson's Dead     200
         23  1.540829e+12        986             Taylor     200
         24  1.540829e+12        986    Naked As We Can     200
         25  1.540829e+12        986            Fantasy     200
         26  1.540829e+12        986           Epilogue     200

                                 ts                                          userAgent  \
         0  2018-11-30 00:22:07.796  Mozilla/5.0 (compatible; MSIE 10.0; Windows NT...
23 2018-11-30 03:06:30.796  "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4...
24 2018-11-30 03:10:30.796  "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4...
25 2018-11-30 03:13:03.796  "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4...
26 2018-11-30 03:15:41.796  "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4...

   userId
0      91
23     86
24     86
25     86
26     86
```

```
In [18]: import datetime

In [19]: var = datetime.datetime.now()

In [20]: var.hour, var.month, var.weekday()

Out[20]: (12, 11, 2)

In [21]: time_data = (t.ts, t.ts.dt.hour , t.ts.dt.day , t.ts.dt.dayofweek , t.ts.dt.month , t.t
         column_labels = ['start_time', 'hour', 'day', 'week', 'month', 'year', 'weekday']

In [22]: time_df = pd.DataFrame(columns=column_labels)

         for index, column_label in enumerate(column_labels):
             time_df[column_label] = time_data[index]

         time_df.head()

Out[22]:                 start_time  hour  day  week  month  year  weekday
         0  2018-11-30 00:22:07.796     0   30     4     11  2018        4
         23 2018-11-30 03:06:30.796     3   30     4     11  2018        4
         24 2018-11-30 03:10:30.796     3   30     4     11  2018        4
         25 2018-11-30 03:13:03.796     3   30     4     11  2018        4
         26 2018-11-30 03:15:41.796     3   30     4     11  2018        4
```

Insert Records into Time Table Implement the `time_table_insert` query in `sql_queries.py` and run the cell below to insert records for the timestamps in this log file into the `time` table. Remember to run `create_tables.py` before running the cell below to ensure you've created/resetted the `time` table in the sparkify database.

```
In [23]: for i, row in time_df.iterrows():
             print(row)
             cur.execute(time_table_insert, list(row))
             conn.commit()

start_time    2018-11-30 00:22:07.796000
hour                                   0
```

```
day                                 30
week                                 4
month                               11
year                              2018
weekday                              4
Name: 0, dtype: object
start_time    2018-11-30 03:06:30.796000
hour                                 3
day                                 30
week                                 4
month                               11
year                              2018
weekday                              4
Name: 23, dtype: object
start_time    2018-11-30 03:10:30.796000
hour                                 3
day                                 30
week                                 4
month                               11
year                              2018
weekday                              4
Name: 24, dtype: object
start_time    2018-11-30 03:13:03.796000
hour                                 3
day                                 30
week                                 4
month                               11
year                              2018
weekday                              4
Name: 25, dtype: object
start_time    2018-11-30 03:15:41.796000
hour                                 3
day                                 30
week                                 4
month                               11
year                              2018
weekday                              4
Name: 26, dtype: object
start_time    2018-11-30 03:21:09.796000
hour                                 3
day                                 30
week                                 4
month                               11
year                              2018
weekday                              4
Name: 27, dtype: object
start_time    2018-11-30 03:58:21.796000
hour                                 3
```

```
day                                30
week                                4
month                              11
year                             2018
weekday                             4
Name: 40, dtype: object
start_time    2018-11-30 04:04:13.796000
hour                                4
day                                30
week                                4
month                              11
year                             2018
weekday                             4
Name: 42, dtype: object
start_time    2018-11-30 04:09:20.796000
hour                                4
day                                30
week                                4
month                              11
year                             2018
weekday                             4
Name: 44, dtype: object
start_time    2018-11-30 04:13:07.796000
hour                                4
day                                30
week                                4
month                              11
year                             2018
weekday                             4
Name: 46, dtype: object
start_time    2018-11-30 04:21:20.796000
hour                                4
day                                30
week                                4
month                              11
year                             2018
weekday                             4
Name: 51, dtype: object
start_time    2018-11-30 04:24:50.796000
hour                                4
day                                30
week                                4
month                              11
year                             2018
weekday                             4
Name: 53, dtype: object
start_time    2018-11-30 04:27:14.796000
hour                                4
```

```
day                                   30
week                                   4
month                                 11
year                                2018
weekday                                4
Name: 55, dtype: object
start_time    2018-11-30 04:32:02.796000
hour                                   4
day                                   30
week                                   4
month                                 11
year                                2018
weekday                                4
Name: 57, dtype: object
start_time    2018-11-30 04:36:30.796000
hour                                   4
day                                   30
week                                   4
month                                 11
year                                2018
weekday                                4
Name: 58, dtype: object
start_time    2018-11-30 04:56:11.796000
hour                                   4
day                                   30
week                                   4
month                                 11
year                                2018
weekday                                4
Name: 62, dtype: object
start_time    2018-11-30 05:31:25.796000
hour                                   5
day                                   30
week                                   4
month                                 11
year                                2018
weekday                                4
Name: 79, dtype: object
start_time    2018-11-30 07:29:09.796000
hour                                   7
day                                   30
week                                   4
month                                 11
year                                2018
weekday                                4
Name: 125, dtype: object
start_time    2018-11-30 10:56:25.796000
hour                                  10
```

```
day                                   30
week                                   4
month                                 11
year                                2018
weekday                                4
Name: 157, dtype: object
start_time    2018-11-30 11:00:01.796000
hour                                  11
day                                   30
week                                   4
month                                 11
year                                2018
weekday                                4
Name: 159, dtype: object
start_time    2018-11-30 11:06:18.796000
hour                                  11
day                                   30
week                                   4
month                                 11
year                                2018
weekday                                4
Name: 160, dtype: object
start_time    2018-11-30 11:09:49.796000
hour                                  11
day                                   30
week                                   4
month                                 11
year                                2018
weekday                                4
Name: 161, dtype: object
start_time    2018-11-30 12:10:57.796000
hour                                  12
day                                   30
week                                   4
month                                 11
year                                2018
weekday                                4
Name: 177, dtype: object
start_time    2018-11-30 12:57:53.796000
hour                                  12
day                                   30
week                                   4
month                                 11
year                                2018
weekday                                4
Name: 186, dtype: object
start_time    2018-11-30 13:20:44.796000
hour                                  13
```

```
day                                   30
week                                   4
month                                 11
year                                2018
weekday                                4
Name: 194, dtype: object
start_time    2018-11-30 13:57:24.796000
hour                                  13
day                                   30
week                                   4
month                                 11
year                                2018
weekday                                4
Name: 213, dtype: object
start_time    2018-11-30 13:59:31.796000
hour                                  13
day                                   30
week                                   4
month                                 11
year                                2018
weekday                                4
Name: 215, dtype: object
start_time    2018-11-30 14:00:38.796000
hour                                  14
day                                   30
week                                   4
month                                 11
year                                2018
weekday                                4
Name: 216, dtype: object
start_time    2018-11-30 14:03:21.796000
hour                                  14
day                                   30
week                                   4
month                                 11
year                                2018
weekday                                4
Name: 219, dtype: object
start_time    2018-11-30 14:04:47.796000
hour                                  14
day                                   30
week                                   4
month                                 11
year                                2018
weekday                                4
Name: 220, dtype: object
start_time    2018-11-30 14:33:53.796000
hour                                  14
```

```
day                                   30
week                                   4
month                                 11
year                                2018
weekday                                4
Name: 239, dtype: object
start_time    2018-11-30 14:37:56.796000
hour                                  14
day                                   30
week                                   4
month                                 11
year                                2018
weekday                                4
Name: 241, dtype: object
start_time    2018-11-30 14:42:45.796000
hour                                  14
day                                   30
week                                   4
month                                 11
year                                2018
weekday                                4
Name: 246, dtype: object
start_time    2018-11-30 14:43:50.796000
hour                                  14
day                                   30
week                                   4
month                                 11
year                                2018
weekday                                4
Name: 247, dtype: object
start_time    2018-11-30 14:47:35.796000
hour                                  14
day                                   30
week                                   4
month                                 11
year                                2018
weekday                                4
Name: 250, dtype: object
start_time    2018-11-30 14:51:37.796000
hour                                  14
day                                   30
week                                   4
month                                 11
year                                2018
weekday                                4
Name: 252, dtype: object
start_time    2018-11-30 16:18:18.796000
hour                                  16
```

```
day                                   30
week                                   4
month                                 11
year                                2018
weekday                                4
Name: 306, dtype: object
start_time    2018-11-30 16:21:23.796000
hour                                  16
day                                   30
week                                   4
month                                 11
year                                2018
weekday                                4
Name: 308, dtype: object
start_time    2018-11-30 16:35:57.796000
hour                                  16
day                                   30
week                                   4
month                                 11
year                                2018
weekday                                4
Name: 317, dtype: object
start_time    2018-11-30 16:39:12.796000
hour                                  16
day                                   30
week                                   4
month                                 11
year                                2018
weekday                                4
Name: 319, dtype: object
start_time    2018-11-30 16:43:28.796000
hour                                  16
day                                   30
week                                   4
month                                 11
year                                2018
weekday                                4
Name: 325, dtype: object
start_time    2018-11-30 18:16:20.796000
hour                                  18
day                                   30
week                                   4
month                                 11
year                                2018
weekday                                4
Name: 367, dtype: object
start_time    2018-11-30 18:21:19.796000
hour                                  18
```

```
day                                 30
week                                 4
month                               11
year                              2018
weekday                              4
Name: 370, dtype: object
start_time    2018-11-30 19:54:24.796000
hour                                19
day                                 30
week                                 4
month                               11
year                              2018
weekday                              4
Name: 387, dtype: object
```

Run `test.ipynb` to see if you've successfully added records to this table.

3.2 #4: `users` Table

Extract Data for Users Table

- Select columns for user ID, first name, last name, gender and level and set to `user_df`

```
In [24]: user_df =  df[['userId', 'firstName', 'lastName', 'gender', 'level']]
```

Insert Records into Users Table Implement the `user_table_insert` query in `sql_queries.py` and run the cell below to insert records for the users in this log file into the `users` table. Remember to run `create_tables.py` before running the cell below to ensure you've created/resetted the `users` table in the sparkify database.

```
In [25]: user_df
```

```
Out[25]:     userId  firstName  lastName gender level
         0       91     Jayden      Bell      M  free
         23      86      Aiden      Hess      M  free
         24      86      Aiden      Hess      M  free
         25      86      Aiden      Hess      M  free
         26      86      Aiden      Hess      M  free
         27      86      Aiden      Hess      M  free
         40      26       Ryan     Smith      M  free
         42      26       Ryan     Smith      M  free
         44      26       Ryan     Smith      M  free
         46      26       Ryan     Smith      M  free
         51      26       Ryan     Smith      M  free
         53      26       Ryan     Smith      M  free
         55      26       Ryan     Smith      M  free
         57      26       Ryan     Smith      M  free
         58      26       Ryan     Smith      M  free
```

```
62       57  Katherine       Gay      F  free
79       92      Ryann     Smith      F  free
125      74     Braden    Parker      M  free
157      92      Ryann     Smith      F  free
159      92      Ryann     Smith      F  free
160      92      Ryann     Smith      F  free
161      92      Ryann     Smith      F  free
177      12     Austin   Rosales      M  free
186      61     Samuel  Gonzalez      M  free
194      43     Jahiem     Miles      M  free
213      50        Ava  Robinson      F  free
215      50        Ava  Robinson      F  free
216      26       Ryan     Smith      M  free
219      50        Ava  Robinson      F  free
220      26       Ryan     Smith      M  free
239     101     Jayden       Fox      M  free
241     101     Jayden       Fox      M  free
246     101     Jayden       Fox      M  free
247     101     Jayden       Fox      M  free
250     101     Jayden       Fox      M  free
252     101     Jayden       Fox      M  free
306      78      Chloe      Roth      F  free
308      78      Chloe      Roth      F  free
317      33    Bronson    Harris      M  free
319      33    Bronson    Harris      M  free
325      33    Bronson    Harris      M  free
367      91     Jayden      Bell      M  free
370      91     Jayden      Bell      M  free
387       5     Elijah     Davis      M  free
```

```
In [26]: for i, row in user_df.iterrows():
             cur.execute(user_table_insert, row)
         conn.commit()
```

Run `test.ipynb` to see if you've successfully added records to this table.

3.3 #5: `songplays` Table

Extract Data and Songplays Table This one is a little more complicated since information from the songs table, artists table, and original log file are all needed for the `songplays` table. Since the log file does not specify an ID for either the song or the artist, you'll need to get the song ID and artist ID by querying the songs and artists tables to find matches based on song title, artist name, and song duration time. - Implement the `song_select` query in `sql_queries.py` to find the song ID and artist ID based on the title, artist name, and duration of a song. - Select the timestamp, user ID, level, song ID, artist ID, session ID, location, and user agent and set to `songplay_data`

Insert Records into Songplays Table

- Implement the `songplay_table_insert` query and run the cell below to insert records for the songplay actions in this log file into the `songplays` table. Remember to run `create_tables.py` before running the cell below to ensure you've created/resetted the `songplays` table in the sparkify database.

```
In [27]: for index, row in df.iterrows():

             # get songid and artistid from song and artist tables
             cur.execute(song_select, (row.song, row.artist, row.length))
             results = cur.fetchone()

             if results:
                 songid, artistid = results
             else:
                 songid, artistid = None, None

             # insert songplay record
             songplay_data = (row.ts, row.userId, row.level, songid, artistid, row.sessionId, ro
             cur.execute(songplay_table_insert, songplay_data)
             conn.commit()
```

Run `test.ipynb` to see if you've successfully added records to this table.

4 Close Connection to Sparkify Database

```
In [28]: conn.close()
```

5 Implement `etl.py`

Use what you've completed in this notebook to implement `etl.py`.

```
In [ ]:
```