

assignment_2

January 11, 2023

1 Assignment 2: Introduction to SQL

1.1 Overview

The bread and butter of utilizing SQL is writing queries. In this assignment, we will be using the Chinook database, which is data containing data from a music store. More information can be found [here](#).

1.1.1 Question 0

Load in any necessary python modules, then connect to the `chinook.sqlite` database file.

```
[ ]: import sqlite3
import pandas as pd

connection = sqlite3.connect("../data/chinook.sqlite")
cursor = connection.cursor()
```

Next, run this code cell to output every table name and corresponding column name in the database to confirm that the imports are functioning correctly.

```
[ ]: def output_tables_and_labels():
    # This function pulls the table names from sqlite, then outputs
    # the column name for each, removing all unnecessary information

    cursor.execute("SELECT name FROM sqlite_master WHERE type='table';")
    table_names = pd.DataFrame(cursor.fetchall())
    for i in table_names[0]:
        print(i)
        cursor.execute("PRAGMA table_info("+str(i)+")")
        print(list(pd.DataFrame(cursor.fetchall())[1]))

output_tables_and_labels()
```

Album

['AlbumId', 'Title', 'ArtistId']

Artist

['ArtistId', 'Name']

Customer

['CustomerId', 'FirstName', 'LastName', 'Company', 'Address', 'City', 'State',

```

'Country', 'PostalCode', 'Phone', 'Fax', 'Email', 'SupportRepId']
Employee
['EmployeeId', 'LastName', 'FirstName', 'Title', 'ReportsTo', 'BirthDate',
'HireDate', 'Address', 'City', 'State', 'Country', 'PostalCode', 'Phone', 'Fax',
'Email']
Genre
['GenreId', 'Name']
Invoice
['InvoiceId', 'CustomerId', 'InvoiceDate', 'BillingAddress', 'BillingCity',
'BillingState', 'BillingCountry', 'BillingPostalCode', 'Total']
InvoiceLine
['InvoiceLineId', 'InvoiceId', 'TrackId', 'UnitPrice', 'Quantity']
MediaType
['MediaTypeId', 'Name']
Playlist
['PlaylistId', 'Name']
PlaylistTrack
['PlaylistId', 'TrackId']
Track
['TrackId', 'Name', 'AlbumId', 'MediaTypeId', 'GenreId', 'Composer',
'Milliseconds', 'Bytes', 'UnitPrice']

```

1.1.2 Question 1

Print out all columns of the “Track” table.

```

[ ]: cursor.execute("SELECT * FROM track")
pd.DataFrame(cursor.fetchall())

```

```

[ ]:
      0      1      2      3      4  \
0      1      For Those About To Rock (We Salute You)      1      1      1
1      2      Balls to the Wall      2      2      1
2      3      Fast As a Shark      3      2      1
3      4      Restless and Wild      3      2      1
4      5      Princess of the Dawn      3      2      1
...    ...
3498  3499  Pini Di Roma (Pinien Von Rom) \ I Pini Della V...  343      2      24
3499  3500  String Quartet No. 12 in C Minor, D. 703 "Quar...  344      2      24
3500  3501      L'orfeo, Act 3, Sinfonia (Orchestra)  345      2      24
3501  3502  Quintet for Horn, Violin, 2 Violas, and Cello ...  346      2      24
3502  3503      Koyaanisqatsi      347      2      10

      5      6      7  \
0      Angus Young, Malcolm Young, Brian Johnson  343719  11170334
1      None  342562  5510424
2      F. Baltes, S. Kaufman, U. Dirkschneider & W. Ho...  230619  3990994
3      F. Baltes, R.A. Smith-Diesel, S. Kaufman, U. D...  252051  4331779
4      Deaffy & R.A. Smith-Diesel  375418  6290521

```

...
3498	None	286741	4718950
3499	Franz Schubert	139200	2283131
3500	Claudio Monteverdi	66639	1189062
3501	Wolfgang Amadeus Mozart	221331	3665114
3502	Philip Glass	206005	3305164

	8
0	0.99
1	0.99
2	0.99
3	0.99
4	0.99

...	...
3498	0.99
3499	0.99
3500	0.99
3501	0.99
3502	0.99

[3503 rows x 9 columns]

1.1.3 Question 2

Print out all of the “short” songs from the “Track” table - tracks under 180000 milliseconds (or 3 minutes). Only print the Name and Milliseconds columns.

```
[ ]: cursor.execute("SELECT Name, Milliseconds FROM track WHERE Milliseconds <= 180000")
pd.DataFrame(cursor.fetchall())
```

		0	1
0	Right Through You	176117	
1	We Die Young	152084	
2	Samba De Uma Nota Só (One Note Samba)	137273	
3	Por Causa De Você	169900	
4	Fotografia	129227	
...
475	Music for the Funeral of Queen Mary: VI. "Thou...	142081	
476	Sing Joyfully	133768	
477	Étude 1, In C Major - Preludio (Presto) - Liszt	51780	
478	String Quartet No. 12 in C Minor, D. 703 "Quar...	139200	
479	L'orfeo, Act 3, Sinfonia (Orchestra)	66639	

[480 rows x 2 columns]

1.1.4 Question 3

Using the Album and Artist tables, print out the columns for an album and its artist's name.

HINT: What column do Album and Artist share?

```
[ ]: cursor.execute('''
      SELECT Album.Title, Artist.Name
      FROM Album
      INNER JOIN Artist
      ON Album.ArtistId = Artist.ArtistId
      ''')
pd.DataFrame(cursor.fetchall())
```

```
[ ]:                                     0 \
0          For Those About To Rock We Salute You
1                      Balls to the Wall
2                      Restless and Wild
3                      Let There Be Rock
4                      Big Ones
..                      ...
342                      Respighi:Pines of Rome
343 Schubert: The Late String Quartets & String Qu...
344                      Monteverdi: L'Orfeo
345                      Mozart: Chamber Music
346 Koyaanisqatsi (Soundtrack from the Motion Pict...

                                     1
0                      AC/DC
1                      Accept
2                      Accept
3                      AC/DC
4                      Aerosmith
..                      ...
342                      Eugene Ormandy
343                      Emerson String Quartet
344 C. Monteverdi, Nigel Rogers - Chiaroscuro; Lon...
345                      Nash Ensemble
346                      Philip Glass Ensemble

[347 rows x 2 columns]
```

1.1.5 Question 4

Finally, let's make a table that contains the album name, artist name, track name, and genre name. Feel free to use the answer to question 3 as a baseline here. This is tricky, so don't be afraid to ask your mentor for help!

```
[ ]: cursor.execute('''
    SELECT Album.Title, Artist.Name, Track.Name, Genre.Name
    FROM Album
    INNER JOIN Artist
        ON Album.ArtistId = Artist.ArtistId
    INNER JOIN Track
        ON Track.AlbumId = Album.AlbumId
    INNER JOIN Genre
        ON Genre.GenreId = Track.GenreId
    ''')
pd.DataFrame(cursor.fetchall())
```

```
[ ]:
0          For Those About To Rock We Salute You
1          Balls to the Wall
2          Restless and Wild
3          Restless and Wild
4          Restless and Wild
...
3498          Respighi:Pines of Rome
3499 Schubert: The Late String Quartets & String Qu...
3500          Monteverdi: L'Orfeo
3501          Mozart: Chamber Music
3502 Koyaanisqatsi (Soundtrack from the Motion Pict...

0          AC/DC
1          Accept
2          Accept
3          Accept
4          Accept
...
3498          Eugene Ormandy
3499          Emerson String Quartet
3500 C. Monteverdi, Nigel Rogers - Chiaroscuro; Lon...
3501          Nash Ensemble
3502          Philip Glass Ensemble

0          For Those About To Rock (We Salute You)
1          Balls to the Wall
2          Fast As a Shark
3          Restless and Wild
4          Princess of the Dawn
...
3498 Pini Di Roma (Pinien Von Rom) \ I Pini Della V...
3499 String Quartet No. 12 in C Minor, D. 703 "Quar...
```

3500	L'orfeo, Act 3, Sinfonia (Orchestra)	Classical
3501	Quintet for Horn, Violin, 2 Violas, and Cello ...	Classical
3502	Koyaanisqatsi	Soundtrack

[3503 rows x 4 columns]

[]: