Backslash

Python Project

Avirup Roy Chowdhury

Pranathi Varma M.

Niharika Gupta

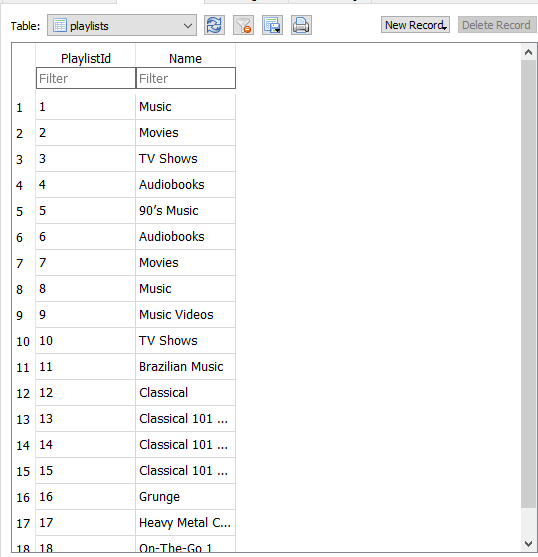
Chaitanya Mohite

Diptajit Chaurangi

# Problem Statement

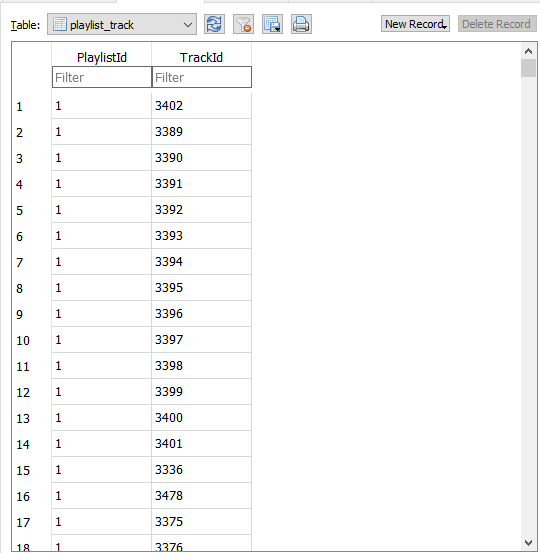
To design a python application which will be able to find common columns between multiple tables of a database without using SQL joins and retrieve data, this data will later be written to a file using IO operations.

In this case we’ll be using a sample database called chinook for the operations, to better demonstrate the implementation we will be searching the entire database for a column called **PlaylistID** without manually finding out where the column exists.



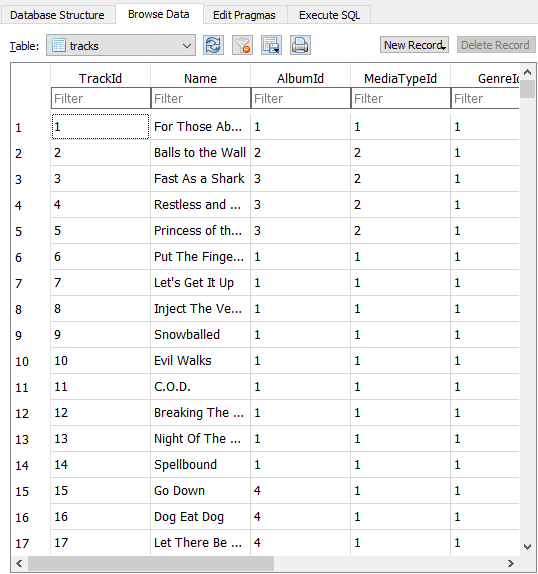
**Fig 1 :** **PlaylistID** exists in *playlists*

Once **PlaylistID** is found which should exist in a table called *playlists* we will take up 4 data members and find their corresponding **TrackID** in a table called *playlist\_track.*



**Fig 2 :** **TrackID** exists in *playlist\_tracks*

The **TrackID** is further referenced to another table called *tracks* from where **Name** and **AlbumID** will be then further exported to a file called *export.txt.*

**

**Fig 3 :** Comparison Table

# Technologies Used

The project base and structure will be revolving around **Python 3** and will be implemented using two libraries i.e

1. **Pandas -**  an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming
2. **Sqlite3** -  a relational database management system contained in a C library. In contrast to many other database management systems, SQLite is not a client–server database engine. Rather, it is embedded into the end program.

# Scope of the Project

The project will revolve around finding a general column name in multiple tables of a database and find relations among different columns name related to first and finally printing the data in a file.

# Source Code and Algorithm

1. **Import the required library**

**import** **sqlite3**

**import** **pandas** **as** **pd**

con = sqlite3.connect('chinook.db')

cur = con.cursor()

1. **Adding all database to a list**

a=['albums','artists','playlists','playlist\_track','tracks']

d={}

1. **Getting column names and table names in which these columns are present in a dictionary**

**for** val **in** a:

print(val)

cur.execute("select \* from **{}**".format(val))

col\_list=cur.description

l1=[]

**for** i **in** range(len(col\_list)):

l1.append(col\_list[i][0])

print(l1)

**for** col **in** l1:

**if** col **in** d:

d[col].append(val)

**else**:

d[col]=[val]

1. **Fetching table name in which columns ‘trackID’ and ‘playlistID’ both are present into a list**

l1=d['TrackId'] l2=d['PlaylistId'] print(l1) print(l2) l3=list(set(l1)&set(l2)) print(l3)

1. **Creating a dictionary for TrackID and their corresponding 10 track ids**

**for** i **in** range(len(playlist)):

cur.execute("select \* from **{}** where PlaylistId=**{}**".format(l3[0],playlist[i]))

list\_c=cur.fetchall()

print(list\_c)

**for** j **in** range(10):

**if** list\_c[j][0] **in** dict\_of\_tracks:

dict\_of\_tracks[list\_c[j][0]].append(list\_c[j][1])

**else**:

dict\_of\_tracks[list\_c[j][0]]=[list\_c[j][1]]

1. **Fetching table name in which column names ‘trackID’ and ‘albumID’ are both present in a list**

tableb=list(set(l1)&set(d['AlbumId']))

1. **Storing it into a file**

f=open('f\_out\_final.txt','a')

f.write("trackId **\t**")

f.write("albumid **\t**")

f.write("name **\t**")

f.write("**\n**")

**for** i **in** range(len(dict\_of\_tracks)):

**for** j **in** range(len(dict\_of\_tracks[playlist[i]])): cur.execute("select TrackId,AlbumId,Name from **{}** where TrackId=**{}**".format(tableb[0],dict\_of\_tracks[playlist[i]][j])) listd=cur.fetchall()

print(listd) f.write(str(listd[0][0]) +'**\t**') f.write(str(listd[0][1]) +'**\t**') f.write(listd[0][2]) f.write("**\n**")

f.close()

|  |  |
| --- | --- |
|  |  |
|  |  |

# Screenshots

# **C:\Users\admin\AppData\Local\Microsoft\Windows\INetCache\Content.Word\code.png**

**Fig 4 :** CODE



**Fig 5 :** output

# Result

The final output has been achieved using all the specified constraints and specifications i.e without execution of SQL queries and the results have been verified to work.

# Libraries and Functions Used:

1. **SQLite3**
   1. ***Connect() –*** *a connection objet to connect to database*
   2. ***Cursor() –*** *object to call execute method to execute queries*
   3. ***Execute() –*** *to perform SQL commands*
   4. ***Fetchall() –*** *to get a list of matching rows*
   5. ***Close() –*** *to close the database*
   6. ***Commit() –*** *to save the changes permanently in the database*