*'''*

*Created on Mar 10, 2017*

**@author:** *Mason Cooper, 140328200*

*'''*

import mysql.connector

# Main Function, This functions handle all calls towards other function in the program. Also runs the main loop and updates the database

def **Main**():

db = logIn() #Calls Login function, and returns database if successful

cursor = db.cursor() #Sets database cursor

if db != *""*:

print(*'Welcome Back!'*)

ask = input(*'What would you like to do\n Add Album(A)\n Add Song(s)\n Quit(Q)\n-> '*) #Prompt user for answer, if they want to add album, add song, or quit

while ask.lower() != *'q'*: #Main Loop, continuously prompts user to add songs or albums until they decide to quit

if ask.lower() == *'a'*: #This if-elif-else statement checks if they want to add an album, a song, or if they entered invalid input

addAlbum(cursor) #Calls the function to add an album

elif ask.lower() == *'s'*:

song\_id = addSong(cursor) #Calls function to add a song

addPerformer(cursor, song\_id) #Calls function to insert into perform

else:

print(*'Invalid Input\n'*)

ask = input(*'What would you like to do\n Add Album(A)\n Add Song(s)\n Quit(Q)\n-> '*)

db.commit() #Updates database

print(*'Finished!'*)

cursor.close() #Closes the cursor

db.close() #Closes the database

return

#This Function prompts the user for their user name and passwords. Then validates the user name and password. If incorrect information entered the user may continue to attempt until they decide to quit

def **logIn**():

cont = True

askchk = True

chk = *''*

while cont: #Loops until the user successfully logins in or decides to quit

print(*'Login'*)

username = input(*'Username: '*) #Prompt user for user name

passW = input(*'Password: '*) #Prompt user for password

try: #Attempts to log user in with given credentials

db = mysql.connector.connect(user = username,

password = passW,

host = *'localhost'*,

database = *'notown'*)

cont = False

except: #Handles failure of login

askchk = True

print(*'Username or Password is Incorrect'*)

while askchk:

chk = input(*'Would you like to continue(Y/N): '*) #Prompts user if the wish to try again

if chk.lower() == *'n'*: #This if-elif-else statement checks if they want to try again, quit, or if they entered invalid input

db = *''*

cont = False

askchk = False

elif chk.lower() == *'y'*:

askchk = False

else:

print(*'Invalid Input'*)

return db #returns database. db is empty string, if login was unsuccessful

#This function handles, obtaining the information of albums and entering albums into the album table

def **addAlbum**(cursor):

print(*'Album Information\n'*)

album\_id = input(*'Album ID: '*) #Prompts user for album id

producer\_id = input(*'Producer ID: '*) #Prompts user for producer id

producer\_Name = input(*'Producer Name: '*) #Prompts user for producer name

release\_date = input(*'Release Date(yyyy-mm-dd): '*) #Prompts user for release date

album\_title = input(*'Album Title: '*) #Prompts user for album title

price = input(*'Price: '*) #Prompts user for price

try: #Adds producer to musician table, will be successful if producer does not already exist in the musician table and will go to exception otherwise

stmt = *"INSERT INTO musicians (sin, name) VALUES (%s, '%s')"* % (producer\_id, producer\_Name)

cursor.execute(stmt)

except: #This exception just passes, because producer is already contained within musician and their is no issue

pass

try: #Attempts to add new album based on given information, if album already exists will go to exception

stmt = *"INSERT INTO albums (album\_id, producer\_id, release\_date, album\_title, price) VALUES (%s, %s, '%s', '%s', '%s')"* % (album\_id, producer\_id, release\_date, album\_title, price)

cursor.execute(stmt)

except: #Lets user know album already exists

print(*'Album already exists'*)

return

#This function handles, obtaining the information of songs and entering songs into the songs table

def **addSong**(cursor):

print(*'Song Information\n'*)

song\_id = input(*'Song ID: '*) #Prompts user for song id

author\_id = input(*'Author ID: '*) #Prompts user for author id

author\_name = input(*'Author Name: '*) #Prompts user for author name

song\_title = input(*'Song Title: '*) #Prompts user for song title

album\_id = input(*'Album ID: '*) #Prompts user for album id

try: #Adds author to musician table, will be successful if author does not already exist in the musician table and will go to exception otherwise

stmt = *"INSERT INTO musicians (sin, name) VALUES (%s, '%s')"* % (author\_id, author\_name)

cursor.execute(stmt)

except: #This exception just passes, because author is already contained within musician and their is no issue

pass

try: #Attempts to add new song based on given information

stmt = *"INSERT INTO songs (song\_id, author\_id, song\_title, album\_id) VALUES (%s, %s, '%s', %s)"* % (song\_id, author\_id, song\_title, album\_id)

cursor.execute(stmt)

except Exception as e: #Lets user know of any errors that may occur if the attempt to add song fails

print(str(e))

return song\_id

#This function adding songs into the perform table

def **addPerformer**(cursor, song\_id):

stmt = *'SELECT album\_id from songs WHERE song\_id = %s'* % (song\_id) #The following block obtains album\_id from songs table using song\_id

cursor.execute(stmt)

album\_id = cursor.fetchone()[0]

stmt = *'SELECT author\_id from songs WHERE song\_id = %s'* % (song\_id) #The following block obtains author\_id from songs table using song\_id

cursor.execute(stmt)

song\_musician\_id = cursor.fetchone()[0]

stmt = *'SELECT producer\_id from albums WHERE album\_id = %s'* % (album\_id) #The following block obtains producer\_id from songs table using album\_id

cursor.execute(stmt)

album\_musician\_id = cursor.fetchone()[0]

if song\_musician\_id == album\_musician\_id: #This if statement adds one record into perform table if author id and producer id are the same

stmt = *'INSERT INTO perform (song\_id, musician\_id, album\_id) VALUES (%s, %s, %s)'* % (song\_id, song\_musician\_id, album\_id)

cursor.execute(stmt)

else: #This else statement adds two record into perform table if author id and producer id are not the same, one entry for each author and producer

stmt = *'INSERT INTO perform (song\_id, musician\_id, album\_id) VALUES (%s, %s, %s)'* % (song\_id, song\_musician\_id, album\_id)

cursor.execute(stmt)

stmt = *'INSERT INTO perform (song\_id, musician\_id, album\_id) VALUES (%s, %s, %s)'* % (song\_id, album\_musician\_id, album\_id)

cursor.execute(stmt)

return

Main() #Runs Main Function