In [59]:

```
!pip install pymysql
!pip install pandas
!pip install numpy
!pip install seaborn
!pip install matplotlib
!pip install cryptography
```

```
Requirement already satisfied: pymysql in c:\users\zhiza\anaconda3\lib\site-packages (1.0.2)
Requirement already satisfied: pandas in c:\users\zhiza\anaconda3\lib\site-packages (1.3.4)
Requirement already satisfied: pytz>=2017.3 in c:\users\zhiza\anaconda3\lib\site-packages (from pandas) (2021.3)
Requirement already satisfied: numpy>=1.17.3 in c:\users\zhiza\anaconda3\lib\site-packages (from pandas) (1.20.3)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\users\zhiza\anaconda3\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\zhiza\anaconda3\lib\site-packages (from python-dateutil>=2.7.3->pandas) (1.16.0)
Requirement already satisfied: numpy in c:\users\zhiza\anaconda3\lib\site-packages (1.20.3)
Requirement already satisfied: seaborn in c:\users\zhiza\anaconda3\lib\site-packages (0.11.2)
Requirement already satisfied: numpy>=1.15 in c:\users\zhiza\anaconda3\lib\site-packages (from seaborn) (1.20.3)
Requirement already satisfied: matplotlib>=2.2 in c:\users\zhiza\anaconda3\lib\site-packages (from seaborn) (3.4.3)
Requirement already satisfied: pandas>=0.23 in c:\users\zhiza\anaconda3\lib\site-packages (from seaborn) (1.3.4)
Requirement already satisfied: scipy>=1.0 in c:\users\zhiza\anaconda3\lib\site-packages (from seaborn) (1.7.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\zhiza\anaconda3\lib\site-packages (from matplotlib>=2.2->seaborn) (8.4.0)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\zhiza\anaconda3\lib\site-packages (from matplotlib>=2.2->seaborn) (2.
8.2)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\zhiza\anaconda3\lib\site-packages (from matplotlib>=2.2->seaborn) (1.3.
1)
Requirement already satisfied: cycler>=0.10 in c:\users\zhiza\anaconda3\lib\site-packages (from matplotlib>=2.2->seaborn) (0.10.0)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\zhiza\anaconda3\lib\site-packages (from matplotlib>=2.2->seaborn) (3.0.4)
Requirement already satisfied: six in c:\users\zhiza\anaconda3\lib\site-packages (from cycler>=0.10->matplotlib>=2.2->seaborn) (1.16.
0)
Requirement already satisfied: pytz>=2017.3 in c:\users\zhiza\anaconda3\lib\site-packages (from pandas>=0.23->seaborn) (2021.3)
Requirement already satisfied: matplotlib in c:\users\zhiza\anaconda3\lib\site-packages (3.4.3)
Requirement already satisfied: numpy>=1.16 in c:\users\zhiza\anaconda3\lib\site-packages (from matplotlib) (1.20.3)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\zhiza\anaconda3\lib\site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\zhiza\anaconda3\lib\site-packages (from matplotlib) (3.0.4)
Requirement already satisfied: pillow>=6.2.0 in c:\users\zhiza\anaconda3\lib\site-packages (from matplotlib) (8.4.0)
Requirement already satisfied: cycler>=0.10 in c:\users\zhiza\anaconda3\lib\site-packages (from matplotlib) (0.10.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\zhiza\anaconda3\lib\site-packages (from matplotlib) (1.3.1)
Requirement already satisfied: six in c:\users\zhiza\anaconda3\lib\site-packages (from cycler>=0.10->matplotlib) (1.16.0)
Requirement already satisfied: cryptography in c:\users\zhiza\anaconda3\lib\site-packages (3.4.8)
Requirement already satisfied: cffi>=1.12 in c:\users\zhiza\anaconda3\lib\site-packages (from cryptography) (1.14.6)
Requirement already satisfied: pycparser in c:\users\zhiza\anaconda3\lib\site-packages (from cffi>=1.12->cryptography) (2.20)
```

User direction:

==

This code is designed to be runned

from here to bottom,

with the last block of code works as a GUI system.

Only the last block of code is suppose to be rerun during the whole process, no exceptions.

Make sure you've all the pre-requisite installed,

if not, run the first block of code before running any other ones.

Remember to change your username, password and all other information that are different in the block below

=============================== = =
*==

Change these BELOW if your username, password and any other informations are different from provided information

In [1]:

```
host = '127.0.0.1'
port = 3306
database = 'final0420'
userg = 'root'
passg = 'abcabc2004'
#Change these two lines above if your username and password is different
```

Change these ABOVE if your username, password and any other informations are different from provided information

In [2]:

```
import pymysql
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import cryptography
```

In [3]:

```
tablename = ['comment','creator','customerservice','moderator','report','staff','subscriber','user','video','viewer']
guestopt = '1'
```

In [4]:

```
def get_tableg(select_statement):
    host = '127.0.0.1'
    port = 3306
    database = 'final0420'
    username = userg
    password = passg
    conn = pymysql.connect(database = database, user = username, password = password, host = host, port = port)

    cursor = conn.cursor()
    cursor.execute(select_statement)

    dataset = cursor.fetchall()
    dataset = pd.DataFrame(dataset, columns=[desc[0] for desc in cursor.description])

    conn.close()
    return dataset
```

In [5]:

```
    scm = '''SELECT * FROM comment'''
    scr = '''SELECT * FROM creator'''
    scs = '''SELECT * FROM customerservice'''
    sm = '''SELECT * FROM moderator'''
    sre = '''SELECT * FROM report'''
    sst = '''SELECT * FROM staff'''
    ssub = '''SELECT * FROM subscriber'''
    sus = '''SELECT * FROM user'''
    svid = '''SELECT * FROM video'''
    sviewer = '''SELECT * FROM viewer'''
    d_comment = get_tableg(scm)
    d_creator = get_tableg(scr)
    d_cust = get_tableg(scs)
    d_mod = get_tableg(sm)
    d_report = get_tableg(sre)
    d_staff = get_tableg(sst)
    d_subs = get_tableg(ssub)
    d_user = get_tableg(sus)
    d_video = get_tableg(svid)
    d_viewer = get_tableg(sviewer)
```

Function: get_data_guest, retrive data from Python isolated Pandas database

In [6]:

```
datalist = ['1. comment','2. creator','3. customer service','4. moderator','5. report','6. staff','7. subs','8. user','9. video','10.
dfs = {'1': d_comment, '2': d_creator, '3': d_cust, '4': d_mod, '5': d_report, '6': d_staff, '7': d_subs, '8': d_user, '9': d_video,
def get_data_guest():
    print(datalist)
    while True:
        wantd = input ('Select the number of the dataset you want from the list above (1-10): ')
        if wantd in dfs: #assuming this is the right answer
            display(dfs[wantd])
            break
        else:
             print('Input is incorrect, please re-enter a valid input')
```

In [7]:

```
def video_revenue_guest():
    df0 = d_video
    df1 = df0.sort_values(by = 'revenue')
    display(df1.head(3))
    in1 = input('Please enter the name of the first NUMERIC column you want to add to the revenue compare plot: ')
    in2 = input('Please enter the name of the second NUMERIC column you want to add to the revenue compare plot: ')
    revenue = df1['revenue'].tolist()
    a   = df1[in1].tolist()
    b   = df1[in2].tolist()


    plt.figure(figsize = (15,6))
    plt.plot(a, revenue,   label = 'Revenue vs {}'.format(in1), marker='o', linewidth=3)
    plt.plot(b, revenue,   label = 'Revenue vs {}'.format(in2),  marker='o', linewidth=3)



    plt.legend(loc='upper left')
    plt.ylabel('Revenue')
    plt.title('Revenue')
    plt.show()
    return df1[['revenue',in1, in2]]
```

In [8]:

```
def video_theme_count_guest():
    sns.catplot(data=d_video, x = 'theme', kind='count')
```

In [9]:

```
def guest_list():
    print('As a guest, you can call upon the following functions: \n 1. Obtain a table of Data \n 2. Video Revenue comparison \n 3. s
```

In [10]:

```
def user_list():
    print('As a user, you can call upon the following functions: \n 4. Obtain up-to-date Data \n 5. up-to-date Video Revenue Comparis
    print('6. up-to-date video theme  \n 7. Search a video by its url or 8. by its title or 9. by its creator \n 10. Filter a number
    print('11. Filter all videos by their views \n 12. insert a new Video using your current user \n 17. delete a Video under your na
    print('19. make a comment under a designated video. \n 22. change your video title')
```

Function: get_data(), retrives up-to-date database from mySQL server

In [11]:

```
def get_data():
    if guestopt =='1':
        print('Guest does not have access to this function')
    else:
        host = '127.0.0.1'
        port = 3306
        database = 'final0420'
        username = userg
        password = passg
        conn = pymysql.connect(database = database, user = username, password = password, host = host, port = port)
        cursor = conn.cursor()
        print(tablename)
        while True:
            dbtable = input('Enter database table name from above:')
            if dbtable in tablename: #assuming this is the right answer
                cursor.execute('SELECT * FROM %s' %(dbtable))
                result = cursor.fetchall()
                result2 = pd.DataFrame(result, columns=[desc[0] for desc in cursor.description])
                display(result2)
                break
            else:
                print('Input is incorrect, please re-enter a valid input')
```

In [12]:

```
def video_revenue():
    if guestopt == '1':
        print('Guest does not have access to this function')
    else:
        df2 = get_tableg('''SELECT * FROM video''')
        df22 = df2.sort_values(by = 'revenue')
        display(df22.head(3))
        in1 = input('Please enter the name of the first NUMERIC column you want to add to the revenue compare plot: ')
        in2 = input('Please enter the name of the second NUMERIC column you want to add to the revenue compare plot: ')
        revenue = df22['revenue'].tolist()
        a   = df22[in1].tolist()
        b   = df22[in2].tolist()


        plt.figure(figsize = (15,6))
        plt.plot(a, revenue,    label = 'Revenue vs {}'.format(in1), marker='o', linewidth=3)
        plt.plot(b, revenue,    label = 'Revenue vs {}'.format(in2),  marker='o', linewidth=3)



        plt.legend(loc='upper left')
        plt.ylabel('Revenue')
        plt.title('Revenue')
        plt.show()
        return df22[['revenue',in1, in2]]
```

In [13]:

```
def video_theme_count():
    if guestopt =='1':
        print('Guest does not have access to this function')
    else:
        df = get_tableg('''SELECT * FROM video''')
        sns.catplot(data=df, x = 'theme', kind='count')
```

In [14]:

```
def get_videos_by_url():
    if guestopt =='1':
        print('Guest does not have access to this function')
    else:
        conn = pymysql.connect(database = database, user = userg, password = passg, host = host, port = port)
        cursor = conn.cursor()
        in1 = input('Please enter your Url: ')
        try:
            query = "Call get_videos_by_url(%s)"
            df = pd.read_sql(query, conn,  params=([in1]))
            display(df)
        except Exception as e:
            print("exception occured: {}".format (e))
        finally:
            conn.close()
```

In [15]:

```
def get_videos_by_title():
    if guestopt =='1':
        print('Guest does not have access to this function')
    else:
        conn = pymysql.connect(database = database, user = userg, password = passg, host = host, port = port)
        cursor = conn.cursor()
        in1 = input('Please enter your title: ')
        try:
            query = "Call get_videos_by_title(%s)"
            df = pd.read_sql(query, conn,  params=([in1]))
            display(df)
        except Exception as e:
            print("exception occured: {}".format (e))
        finally:
            conn.close()
```

In [16]:

```
def get_videos_by_creator():
    if guestopt =='1':
        print('Guest does not have access to this function')
    else:
        conn = pymysql.connect(database = database, user = userg, password = passg, host = host, port = port)
        cursor = conn.cursor()
        in1 = input('Please enter the creator username: ')
        try:
            query = "Call get_videos_by_creator(%s)"
            df = pd.read_sql(query, conn,  params=([in1]))
            display(df)
        except Exception as e:
            print("exception occured: {}".format (e))
        finally:
            conn.close()
```

In [17]:

```
def filter_vid_by_update():
    if guestopt =='1':
        print('Guest does not have access to this function')
    else:
        conn = pymysql.connect(database = database, user = userg, password = passg, host = host, port = port)
        cursor = conn.cursor()
        in1 = input('Please enter the number of videos on display: ')
        try:
            query = "Call filter_videos_by_upload_date(%s)"
            df = pd.read_sql(query, conn,  params=([in1]))
            display(df)
        except Exception as e:
            print("exception occured: {}".format (e))
        finally:
            conn.close()
```

In [18]:

```
def filter_vid_by_view():
    if guestopt =='1':
        print('Guest does not have access to this function')
    else:
        conn = pymysql.connect(database = database, user = userg, password = passg, host = host, port = port)
        cursor = conn.cursor()
        try:
            query = "Call filter_videos_by_view()"
            df = pd.read_sql(query, conn)
            display(df)
        except Exception as e:
            print("exception occured: {}".format (e))
        finally:
            conn.close()
```

In [19]:

```
def insert_video():
    if guestopt =='1':
        print('Guest does not have access to this function')
    elif guestopt =='2':
        conn = pymysql.connect(database=database, user=userg, password=passg, host=host, port=port)
        cursor = conn.cursor()
        inurl = input('input your custom Url: ')
        inuid = userid
        intitle = input('input your Title: ')
        intheme = input('input your Theme from: Sports, Music, Technology, Food, \n Beauty, Comedy, Politics, Science. case sensitive
        inlength = input('input your video total length in minute: ')
        dmod = get_tableg('''SELECT * FROM moderator''')
        inmod = dmod.loc[dmod['mod_theme'] == intheme, 'modid'].iloc[0]
        try:
            # Call stored procedure
            cursor.callproc('insert_video', (inurl, inuid, intitle, intheme, inlength, inmod))
            conn.commit()
            df = pd.read_sql("SELECT * FROM video WHERE videoUrl=%s", conn, params=[inurl])
            display(df)
        except Exception as e:
            print("exception occured: {}".format(e))
        finally:
            conn.close()
    else:
        print('!!! Admin, please use function 13 instead of 12 !!!')
```

In [20]:

```
def delete_vid():
    if guestopt =='1':
        print('Guest does not have access to this function')
    elif guestopt == '2':
        try:
            conn = pymysql.connect(database = database, user = userg, password = passg, host = host, port = port)
            cursor = conn.cursor()
            df2 = pd.read_sql("SELECT * FROM video WHERE vid_creatorId = %s", conn, params = [userid])
            display(df2)
            delvid = input('Type the URL of the video you are deleting: ')

            if delvid in df2['videoUrl'].tolist():
                query = "SELECT final0420.delete_video('%s')"%delvid
                cursor.execute(query)
                conn.commit()
                disp = input('Deletion successful, do you want to see the updated video list? type yes if you do: ')
                if disp == 'yes':
                    df = pd.read_sql("SELECT * FROM video WHERE vid_creatorId = %s", conn, params = [userid])
                    display(df)
                else:
                    print(' ')
            else:
                print('Input failed, you do not have that video under your name.')
        except Exception as e:
            print("exception occured: {}".format (e))
        finally:
            conn.close()
    else:
        print('!!! Admin, please use the admin function 16 instead !!!')
```

In [21]:

```
def create_comment():
    if guestopt =='1':
        print('Guest does not have access to this function')
    elif guestopt == '2':
        try:
            conn = pymysql.connect(database = database, user = userg, password = passg, host = host, port = port)
            cursor = conn.cursor()
            vidcom = input('Type the URL of the video the comment is going to place under: ')
            usercom = userid
            conten = input('Please input what you want to say: ')
            df2 = pd.read_sql("SELECT * FROM video", conn)
            if vidcom in df2['videoUrl'].tolist():
                cursor.callproc('add_comment', (usercom, vidcom, conten))
                conn.commit()
                disp = input('Comment successfully posted, do you want to see the updated video comment list? type yes if you do: ')
                if disp == 'yes':
                    df = pd.read_sql("SELECT * FROM comment WHERE c_vidUrl=%s", conn, params = [vidcom])
                    display(df)
                else:
                    print(' ')
            else:
                print('Input failed, no such url in database.')
        except Exception as e:
            print("exception occured: {}".format (e))
        finally:
            conn.close()
    else:
        print('!!! Admin, please use the admin function 18 instead !!!')
```

In [22]:

```
def change_title():
    if guestopt =='1':
        print('Guest does not have access to this function')
    elif guestopt == '2':
        try:
            conn = pymysql.connect(database = database, user = userg, password = passg, host = host, port = port)
            cursor = conn.cursor()
            df2 = pd.read_sql("SELECT * FROM video WHERE vid_creatorId=%s", conn, params = [userid])
            display(df2)
            vidurl = input('Type the URL of the video you are changing: ')
            vidtitle = input('please input the new title: ')
            if vidurl in df2['videoUrl'].tolist():
                cursor.callproc('update_video_title', (vidurl, vidtitle))
                conn.commit()
                disp = input('Modification successful, do you want to see the updated video? type yes if you do: ')
                if disp == 'yes':
                    df = pd.read_sql("SELECT * FROM video WHERE videoUrl=%s", conn, params = [vidurl])
                    display(df)
                else:
                    print(' ')
            else:
                print('Input failed, no such url under your video list.')
        except Exception as e:
            print("exception occured: {}".format (e))
        finally:
            conn.close()
    else:
        print('!! Please use admin function instead. !!')
```

In [ ]:

```

```

In [23]:

```
def admin_list():
    print('As an admin, you can call upon the following functions: \n 4. Obtain up-to-date Data \n 5. up-to-date Video Revenue Compar
    print('6. up-to-date video theme  \n 7. Search videos by its url or 8. by its title or 9. by its creator \n 10. Filter a number c
    print('11. Filter all videos by their views \n 13. insert a new Video \n 14. create a new user \n 15. print log \n 16. delete a v
    print('18. Create a comment under an existing video and an existing user \n 20. Add a new user \n 21. Change a video title')
```

In [24]:

```
def insert_video_admin():
    if guestopt =='1':
        print('Guest does not have access to this function')
    elif guestopt == '2':
        print('Users does not have access to this function')
    else:
        conn = pymysql.connect(database = database, user = userg, password = passg, host = host, port = port)
        cursor = conn.cursor()
        inurl = input('input custom Url: ')
        inuid = input('input the UserID:')
        intitle = input('input your Title: ')
        intheme = input('input your Theme from: Sports, Music, Technology, Food, \n Beauty, Comedy, Politics, Science. case sensitive
        inlength = input('input your video total length in minute: ')
        dmod = get_tableg('''SELECT * FROM moderator''')
        inmod = dmod.loc[dmod['mod_theme'] == intheme, 'modid'].iloc[0]
        try:
            cursor.callproc('insert_video', (inurl, inuid, intitle, intheme, inlength, inmod))
            conn.commit()
            df = pd.read_sql("SELECT * FROM video WHERE videoUrl=%s", conn, params=[inurl])
            display(df)
        except Exception as e:
            print("exception occured: {}".format (e))
        finally:
            conn.close()
```

In [25]:

```
def insert_user():
    if guestopt =='1':
        print('Guest does not have access to this function')
    elif guestopt == '2':
        print('Users does not have access to this function')
    else:
        conn = pymysql.connect(database = database, user = userg, password = passg, host = host, port = port)
        cursor = conn.cursor()
        inuid = input('input the UserID: ')
        inname = input('input the Username: ')
        inemail = input('enter the User email address: ')
        increator = input('is user a Creator? Enter 1 if Yes, 0 if No: ')
        inviewer = input('is user a Viewer? Enter 1 if Yes, 0 if No: ')
        try:
            cursor.callproc('add_new_user', (inuid, inname, inemail, increator, inviewer))
            conn.commit()
            df = pd.read_sql("SELECT * FROM user WHERE userId=%s", conn, params=[inuid])
            display(df)
        except Exception as e:
            print("exception occured: {}".format (e))
        finally:
            conn.close()
```

In [26]:

```
def print_log():
    if guestopt =='1':
        print('Guest does not have access to this function')
    elif guestopt == '2':
        print('Users does not have access to this function')
    else:
        conn = pymysql.connect(database = database, user = userg, password = passg, host = host, port = port)
        query = "SELECT * FROM video_insert_log"
        df = pd.read_sql(query, conn)
        display(df)
```

In [27]:

```
def delete_vid_admin():
    if guestopt =='1':
        print('Guest does not have access to this function')
    elif guestopt == '2':
        print('Users does not have access to this function')
    else:
        try:
            conn = pymysql.connect(database = database, user = userg, password = passg, host = host, port = port)
            cursor = conn.cursor()
            delvid = input('Type the URL of the video you are deleting: ')
            df2 = pd.read_sql("SELECT * FROM video", conn)
            if delvid in df2['videoUrl'].tolist():
                query = "SELECT final0420.delete_video('%s')"%delvid
                cursor.execute(query)
                conn.commit()
                disp = input('Deletion successful, do you want to see the updated video list? type yes if you do: ')
                if disp == 'yes':
                    df = pd.read_sql("SELECT * FROM video", conn)
                    display(df.tail(5))
                else:
                    print(' ')
            else:
                print('Input failed, no such url in video database.')
        except Exception as e:
            print("exception occured: {}".format (e))
        finally:
            conn.close()
```

In [28]:

```
def create_comment_admin():
    if guestopt =='1':
        print('Guest does not have access to this function')
    elif guestopt == '2':
        print('Users does not have access to this function')
    else:
        try:
            conn = pymysql.connect(database = database, user = userg, password = passg, host = host, port = port)
            cursor = conn.cursor()
            vidcom = input('Type the URL of the video the comment is going to place under: ')
            usercom = input('Type the userId the comment is under: ')
            conten = input('Please input the content of the comment: ')
            df2 = pd.read_sql("SELECT * FROM video", conn)
            df3 = pd.read_sql("SELECT * FROM user", conn)
            if vidcom in df2['videoUrl'].tolist() and int(usercom) in df3['userId'].tolist():
                cursor.callproc('add_comment', (usercom, vidcom, conten))
                conn.commit()
                disp = input('Addition successful, do you want to see the updated video comment list? type yes if you do: ')
                if disp == 'yes':
                    df = pd.read_sql("SELECT * FROM comment WHERE c_vidUrl=%s", conn, params = [vidcom])
                    display(df)
                else:
                    print(' ')
            elif vidcom in df2['videoUrl'].tolist() and usercom not in df3['userId'].tolist():
                print('Input failed, no such user in database.')
            else:
                print('Input failed, no such url in database.')
        except Exception as e:
            print("exception occured: {}".format (e))
        finally:
            conn.close()
```

In [29]:

```
def add_new_u():
    if guestopt =='1':
        print('Guest does not have access to this function')
    elif guestopt == '2':
        print('Users does not have access to this function')
    else:
        try:
            conn = pymysql.connect(database = database, user = userg, password = passg, host = host, port = port)
            cursor = conn.cursor()
            newuid = int(input('Type the new userid of the user: '))
            newuname = input('Type the username of the user: ')
            newumail = input('Type the email of the user: ')
            newuc = input('Is user a content creator? type 1 if yes, 0 if no: ')
            newuv = input('Is user a viewer? type 1 if yes, 0 if no: ')
            df2 = pd.read_sql("SELECT userId FROM user", conn)
            if newuid not in df2['userId'].tolist():
                cursor.callproc('add_new_user', (newuid, newuname, newumail, newuc, newuv))
                conn.commit()
                disp = input('Addition successful, do you want to see the updated user list? type yes if you do: ')
                if disp == 'yes':
                    df = pd.read_sql("SELECT * FROM user", conn,)
                    display(df)
                else:
                    print(' ')
            else:
                print('Input failed, user id duplicated in database.')
        except Exception as e:
            print("exception occured: {}".format (e))
        finally:
            conn.close()
```

In [30]:

```
def change_title_admin():
    if guestopt =='1':
        print('Guest does not have access to this function')
    elif guestopt == '2':
        print('Users does not have access to this function')
    else:
        try:
            conn = pymysql.connect(database = database, user = userg, password = passg, host = host, port = port)
            cursor = conn.cursor()
            vidurl = input('Type the URL of the video you are changing: ')
            vidtitle = input('please input the new title: ')
            df2 = pd.read_sql("SELECT * FROM video", conn)
            if vidurl in df2['videoUrl'].tolist():
                cursor.callproc('update_video_title', (vidurl, vidtitle))
                conn.commit()
                disp = input('Modification successful, do you want to see the updated video? type yes if you do: ')
                if disp == 'yes':
                    df = pd.read_sql("SELECT * FROM video WHERE videoUrl=%s", conn, params = [vidurl])
                    display(df)
                else:
                    print(' ')
            else:
                print('Input failed, no such url in video database.')
        except Exception as e:
            print("exception occured: {}".format (e))
        finally:
            conn.close()
```

The following function is for executing all the functions above based on user input

In [31]:

```
def run_funcs():
    funcs = {'1': get_data_guest, '2': video_revenue_guest, '3': video_theme_count_guest, '4': get_data, '5': video_revenue,
             '6':  video_theme_count, '7': get_videos_by_url, '8': get_videos_by_title, '9': get_videos_by_creator,
             '10': filter_vid_by_update, '11': filter_vid_by_view,'12': insert_video,'13': insert_video_admin,'14': insert_user, '15'
             '16': delete_vid_admin, '17': delete_vid, '18': create_comment_admin, '19': create_comment, '20': add_new_u,
            '21': change_title_admin, '22': change_title}
    while True:
        execfunc = input ('Select the number of the functions you want to run from the list above: ')
        if execfunc in funcs: #assuming this is the right answer
            return funcs[execfunc]()
            break
        else:
             print('Input is incorrect, please re-enter a valid input')
```

In case of failure or no further input options, rerun the codeblock below and re-login to database.

In [42]:

```
guestopt =  input ('If you are a guest, enter 1; If you are a user, enter 2. Else, enter anything else and you will be redirected to
if guestopt == '1':
    guest_list()
    run_funcs()
elif guestopt == '2':
    userid = int(input('Enter your userId: '))
    username = input('Enter your username: ')
    LoU = get_tableg('''SELECT * FROM user''')
    corrname = LoU.loc[LoU['userId'] == userid, 'username'].iloc[0]
    if corrname == username:
        print('Welcome,', username)
        user_list()
        run_funcs()
    else:
        print('certification invalid, please re-run the code block and re-enter.')
        guestopt = '1'
```

```
else:
    admin1 = input('Enter username:')
    adpass1 = input('Enter password:')
    # Username is root
    # Password is abcabc2004
    #change these for your own database
    print('Username: ',admin1,'Password: ',adpass1)
    if admin1 == userg and adpass1 == passg:
        print('Welcome, Admin')
        admin_list()
        run_funcs()
    else:
        print('certification invalid, please re-run the code block and re-enter.')
        guestopt = '1'
```

```
If you are a guest, enter 1; If you are a user, enter 2. Else, enter anything else and you will be redirected to login page.1
As a guest, you can call upon the following functions:
 1. Obtain a table of Data
 2. Video Revenue comparison
 3. show number of video themes

Select the number of the functions you want to run from the list above: 1
['1. comment', '2. creator', '3. customer service', '4. moderator', '5. report', '6. staff', '7. subs', '8. user', '9. video', '10. v
iewer']
Select the number of the dataset you want from the list above (1-10): 1
```

	c_userId	c_vidUrl	comment_date	content	likes	reply_num
0	1	https://www.youtube.com/watch?v=fJ9rUzIMcZQ	2023-04-03	This is my favorite video on the channel so far.	18	2
1	2	https://www.youtube.com/watch?v=5qap5aO4i9A	2023-03-20	This video was very helpful!	12	3
2	3	https://www.youtube.com/watch?v=I8dG4Gu1J_E	2023-03-25	Can you make a video about X topic?	5	1
3	4	https://www.youtube.com/watch?v=ArHc-7XpPBo	2023-03-28	I appreciate the effort you put into this vide...	10	4
4	5	https://www.youtube.com/watch?v=5qap5aO4i9A	2023-03-15	Great video, thanks for sharing!	8	2
5	6	https://www.youtube.com/watch?v=fJ9rUzIMcZQ	2023-04-04	Can you do a video about Y topic next?	6	1
6	7	https://www.youtube.com/watch?v=I8dG4Gu1J_E	2023-03-22	I love this channel, keep up the good work!	20	5
7	8	https://www.youtube.com/watch?v=fJ9rUzIMcZQ	2023-04-02	Your videos have helped me a lot in my studies...	15	6
8	9	https://www.youtube.com/watch?v=I8dG4Gu1J_E	2023-03-27	Interesting video, but I have some disagreemen...	3	0
9	10	https://www.youtube.com/watch?v=IbYOfjLDPYo	2023-04-05	I found this video very insightful, thank you!	7	1

	c_userId	c_vidUrl	comment_date	content	likes	reply_num
10	11	https://www.youtube.com/watch?v=ArHc-7XpPBo	2023-04-10	I just discovered your channel and I love it a...	5	1
11	12	https://www.youtube.com/watch?v=7dqJTP6l9y0	2023-04-11	The examples you gave in this video were reall...	12	3
12	13	https://www.youtube.com/watch?v=4twN6NnmZNU	2023-04-13	Your videos have helped me a lot in my researc...	8	2
13	14	https://www.youtube.com/watch?v=wQXZptfRUsE	2023-04-15	I appreciate how clearly you explain complex c...	15	4
14	15	https://www.youtube.com/watch?v=t1vqjiGI3wU	2023-04-16	I have been struggling with this topic for a w...	10	2
15	16	https://www.youtube.com/watch?v=iPmCxQuWwck	2023-04-17	I disagree with some of your points in this vi...	2	0
16	17	https://www.youtube.com/watch?v=ifxMxrd-Elc	2023-04-19	I always look forward to your videos, they are...	20	5
17	18	https://www.youtube.com/watch?v=zFgpNbsO-CE	2023-04-20	I learned so much from this video, thank you f...	15	3
18	19	https://www.youtube.com/watch?v=xT8J6TBJIjw	2023-04-21	This video was a great introduction to the top...	7	1
19	20	https://www.youtube.com/watch?v=lJQKtp03X9M	2023-04-23	I would love to see more videos about this top...	4	1

```
In [ ]:
```