

# DB project4 Report

2016024893 오성준

## <수정한 부분1>

- › 주민등록 번호를 받는 대신에 생일을 받는 것으로 대체를 했습니다. 어차피 key로 쓰지 않을 거라면 이렇게 하는 편이 구현하기에 나을 것이라 생각했습니다.
- › 아티스트들의 데뷔 년도가 확실하지 않은 관계로, 아티스트의 생일로 대체했습니다.
- › 굳이 관리자와 사용자의 relationship이 필요 없을 거라고 생각이 들었습니다. 그래서 그 값을 삭제 했습니다. Db가 작기 때문에 굳이 관리자를 많이 만들 필요가 없는 상태라고 생각했습니다. 아마도 향후에 규모가 커지면 필요할 것 같습니다.
- › 관리자와 사용자의 권한 필드를 없애고 로그인을 하는 순간 분리하도록 했습니다. 권한 필드가 굳이 있을 필요가 없다는 생각이었습니다.

## <수정한 부분2>

- › 음원에도 같은 아티스트의 같은 노래가 다른 앨범에 각각 수록될 수 있다는 생각이 들어서 음원에 album 이름을 primary key로 추가했습니다. 그 와 동시에 foreign key로 앨범 영역에서 값을 가져오게 했습니다. 그와 동시에 in\_playlist db에서도 music에서 이 album또한 primary key이자 foreign key로 가져오게 되었습니다.
- › 스트리밍 부분에서 달별 스트리밍을 삭제하고 일자별 스트리밍만 남겼습니다. 그 와 동시에 일자를 년 월 일로 나누지 않고 한꺼번에 date 형식의 데이터로 일자를 받는 것으로 수정했습니다.
- › 한글이 db를 쓰기에 불편한 점이 있어서 모든 이름을 영어로 변경했습니다.

# <수정한 부분 최종- Manager/User>

이름:

코멘트:

열: + 추가 × 제거 ▲ 위로 ▼ 아래로

#	이름	데이터 유형	길이/설정	부호 ...	NULL ...	0으...	기본값
1	m_index	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음
2	email	VARCHAR	40	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음
3	phone_num	CHAR	13	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음
4	name	VARCHAR	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음
5	bdate	DATE		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음
6	sex	CHAR	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음
7	passwd	VARCHAR	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음

이름:

코멘트:

열: + 추가 × 제거 ▲ 위로 ▼ 아래로

#	이름	데이터 유형	길이/설정	부호 ...	NULL ...	0으...	기본값
1	u_index	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음
2	email	VARCHAR	40	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
3	phone_num	CHAR	13	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음
4	name	VARCHAR	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음
5	bdate	DATE		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
6	sex	CHAR	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
7	password	VARCHAR	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음

# <수정한 부분 최종- Music/Album/Artist>

이름:

코멘트:

열: + 추가 × 제거 ▲ 위로 ▼ 아래로

#	이름	데이터 유형	길이/설정	부호 ...	NULL ...	0으...	기본값
1	a_index	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음
2	name	VARCHAR	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음
3	b_date	DATE		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL

이름:

코멘트:

열: + 추가 × 제거 ▲ 위로 ▼ 아래로

#	이름	데이터 유형	길이/설정	부호 ...	NULL ...	0으...	기본값
1	album_name	VARCHAR	50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음
2	enroll_date	DATE		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
3	artist	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음

이름:

코멘트:

열: + 추가 × 제거 ▲ 위로 ▼ 아래로

#	이름	데이터 유형	길이/설정	부호 ...	NULL ...	0으...	기본값
1	music_name	VARCHAR	50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음
2	enroll_date	DATE		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
3	artist	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음
4	album	VARCHAR	50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음

# <수정한 부분 최종- Date\_stream/Playlist/In\_playlist>

이름:

코멘트:

열: + 추가 ✕ 제거 ▲ 위로 ▼ 아래로

#	이름	데이터 유형	길이/설정	부호 ...	NULL ...	0으...	기본값
1	str_cnt	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음
2	music	VARCHAR	50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음
3	artist	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음
4	album	VARCHAR	50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음
5	str date	DATE		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음

이름:

코멘트:

열: + 추가 ✕ 제거 ▲ 위로 ▼ 아래로

#	이름	데이터 유형	길이/설정	부호 ...	NULL ...	0으...	기본값
1	playlist	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음
2	producer	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음
3	music	VARCHAR	50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음
4	album	VARCHAR	50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음
5	artist	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음

이름:

코멘트:

열: + 추가 ✕ 제거 ▲ 위로 ▼ 아래로

#	이름	데이터 유형	길이/설정	부호 ...	NULL ...	0으...	기본값
1	pl_name	VARCHAR	50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음
2	pl_index	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음
3	producer	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음

## <구현한 기능 (User)>

### › User Function

- User login
- Search music by string ( by music, album, artist name or ALL)
- Search music and play or add it to user's playlist
- Create user's playlist and Delete user's playlist
- Search playlist and streaming playlist
- Show Chart (today, this month, the date user want)

## <구현한 기능 (Manager)>

- › Manager Function
  - Manager login
  - Manipulate Music DB (Add, Delete)
  - Manipulate Album DB (Add, Delete)
  - Manipulate Artist DB (just Add)
  - Manipulate User (Add, Delete, Update)



## <이후의 보고서 구성>

- › 앞에 페이지에 언급한 기능들에 한가지 기능당 2장의 pdf를 차지합니다.
- › 우선에 첫번째 장에는 해당하는 코드의 사진과 그 코드가 수행하는 기능에 대한 설명이 있습니다.
- › 그리고 두번째 장에는 해당 기능이 수행되는 스크린샷과 그 과정에서 사용하는 SQL문 명세가 기재 될 예정입니다.

# User/Manager Login

## [PROGRAM CODE]

```
login=False
manager=0 #0:user 1:manager

while login==False:
    print("You have to log in first")
    print("0: for login as user")
    print("1: for login as manager")

    input_a=input("숫자를 입력하세요: ")
    manager=int(input_a)#항후 애러처리 해주기

    input_name=input("Name: ")
    input_passwd=input("Passwd: ")

    if manager==1:
        sql_login=("SELECT* FROM manager WHERE name='%s' AND passwd='%s'"%(input_name,input_passwd))
        cursor.execute(sql_login)
        result_login=cursor.fetchall()
        if result_login!=(): login=True

    if manager==0:
        sql_login=("SELECT* FROM user WHERE name='%s' AND password='%s'"%(input_name,input_passwd))
        cursor.execute(sql_login)
        result_login=cursor.fetchall()
        if result_login!=(): login=True
    print()

if manager==1: print("You log in as manager!")
else: print("You log in as user!")
```

## [EXPLANATION]

- › 0을 입력하면 user로 1을 입력하면 manager로 로그인이 됩니다.
- › 그 이후에 name과 password를 입력 받고 select 구문을 통해서 해당하는 user나 manager가 있는지 확인합니다.
- › 로그인이 됐다면 그 다음 기능들을 사용할 수 있습니다.

# User/Manager Login

## [CMD CAPTURE]

```
C:\Users\joon2\Desktop>py DBproj4.py
You have to log in first
0: for login as user
1: for login as manager
숫자를 입력하세요: 0
Name: ta
Passwd: ta123
```

```
You log in as user!
What Kind of Service do you want?
1. Search All
2. Search by Music
3. Search by Artist
4. Search by Album
5. Playlist
6. Show Chart
0. Finish Program
Enter Number: 0
```

Program Finish!

```
C:\Users\joon2\Desktop>py DBproj4.py
You have to log in first
0: for login as user
1: for login as manager
숫자를 입력하세요: 1
Name: ta
Passwd: ta123
```

```
You log in as manager!
What Kind of Service do you want?
1. Manipulate Music
2. Manipulate Artist
3. Manipulate Album
4. Manipulate User
0. Finish Program
Enter Number: 
```

## [SQL TEXT]

- › sql\_login=("SELECT\* FROM user WHERE name='%s' AND password='%s'"%(input\_name,input\_passwd))
- › sql\_login=("SELECT\* FROM manager WHERE name='%s' AND passwd='%s'"%(input\_name,input\_passwd))

# (User) SEARCH MUSIC

## [PROGRAM CODE]

```
def search_all():
    global today_date, result_login
    init=True
    while init==True:
        tmp=input("Search Music by All Method: ")
        #string을 input으로 입력가능!
        tmp=tmp.strip()
        sql_tmp=("SELECT a.name as artist, m.music_name as music, m.album FROM music AS m, artist AS a WHERE a.a_index=m.artist AND (music_name LIKE '%s' OR a.name LIKE '%s' OR m.album LIKE '%s')")
        cursor.execute(sql_tmp)
        result=cursor.fetchall()
        if result==():
            print("There is no result")
            continue
        another=False
        while another==False:
            result_tmp=pd.DataFrame(result_)
            print(result_tmp)
            print()
            print("0: Finish Search")
            print("1: Select One")
            print("2: Search Another")
            select=int(input("Enter: "))
            print()
            if select==0:
                init=False
                another=True
            elif select==2:
                another=True
            elif select==1:
                music_num=int(input("Enter Music Num: "))
                select_music=result_tmp.loc[[music_num],:]
                print(select_music)
                print("\nSelect Your Service")
                print("1: Play")
                print("2: Put into your playlist")
                print("0: Back")
                service_num=int(input("Enter Service Num: "))
                if service_num==1:
                    cnt=0
                    for row in result_:
                        if cnt==music_num: break
                        else: cnt+=1
                    sql_tmptmp=("SELECT a_index From artist WHERE name='%s'" %(row['artist']))
                    cursor.execute(sql_tmptmp)
                    artist_tmp=cursor.fetchall()
                    artist_num=artist_tmp[0]['a_index']
                    #print(artist_num)
                    str_update(row['music'],artist_num,row['album'],today_date)
                    streaming()
                    #print(select_music['name'])
                    time.sleep(5)
                    print("Finish Music\n")
```

```
                if service_num==2:
                    row=result_[music_num]
                    sql_tmptmp=("SELECT a_index From artist WHERE name='%s'" %(row['artist']))
                    cursor.execute(sql_tmptmp)
                    artist_tmp=cursor.fetchall()
                    artist_num=artist_tmp[0]['a_index']
                    sql=("SELECT* FROM playlist WHERE producer=%s"%result_login[0]['u_index'])
                    cursor.execute(sql)
                    result_plpl=cursor.fetchall()
                    if result_plpl==():
                        print("There is no your playlist")
                        continue
                    frame=pd.DataFrame(result_plpl)
                    print(frame)
                    back=input("Keep going? yes or no: ")
                    if back=="no": continue
                    pl_n=int(input("Enter the number playlist: "))
                    pl=result_plpl[pl_n]
                    tmp=(pl['pl_index'], pl['producer'], row['music'],row['album'],artist_num)
                    sql=("SELECT* FROM in_playlist WHERE playlist=%s AND producer=%s AND music=%s AND album=%s AND artist=%s"%tmp)
                    cursor.execute(sql)
                    ch=cursor.fetchall()
                    if ch==():
                        print("Already IN!")
                        continue
                    else:
                        sql=("Insert into in_playlist(playlist,producer,music,album,artist) values (%s,%s,%s,'%s','%s')"%tmp)
                        cursor.execute(sql)
                        db.commit()
                        print("Finish!!!")
```

```
                    if service_num==2:
                        row=result_[music_num]
                        sql_tmptmp=("SELECT a_index From artist WHERE name='%s'" %(row['artist']))
                        cursor.execute(sql_tmptmp)
                        artist_tmp=cursor.fetchall()
                        artist_num=artist_tmp[0]['a_index']
                        sql=("SELECT* FROM playlist WHERE producer=%s"%result_login[0]['u_index'])
                        cursor.execute(sql)
                        result_plpl=cursor.fetchall()
                        if result_plpl==():
                            print("There is no your playlist")
                            continue
                        frame=pd.DataFrame(result_plpl)
                        print(frame)
                        back=input("Keep going? yes or no: ")
                        if back=="no": continue
                        pl_n=int(input("Enter the number playlist: "))
                        pl=result_plpl[pl_n]
                        tmp=(pl['pl_index'], pl['producer'], row['music'],row['album'],artist_num)
                        sql=("SELECT* FROM in_playlist WHERE playlist=%s AND producer=%s AND music=%s AND album=%s AND artist=%s"%tmp)
                        cursor.execute(sql)
                        ch=cursor.fetchall()
                        if ch==():
                            print("Already IN!")
                            continue
                        else:
                            sql=("Insert into in_playlist(playlist,producer,music,album,artist) values (%s,%s,%s,'%s','%s')"%tmp)
                            cursor.execute(sql)
                            db.commit()
                            print("Finish!!!")
```

## [EXPLANATION]

- 가장 기능이 많은 SEARCH ALL기능을 통해서 다른 SEARCH들을 함께 설명하겠습니다.
- 우선에 들어가서 string을 치면 음악, 앨범, 아티스트 이름에서 해당 string이 들어 있는 노래들을 모두 select합니다. 다른 것들은 각각의 부분에 해당 string이 있으면 노래를 select합니다.
- 그 이후에 노래를 하나를 선택합니다.
- 그리고 play를 하면, update 구문을 통해서 streaming 횟수가 1 올라갑니다.(streaming과 str\_cnt라는 함수를 통해서 update합니다.
- (Search all에서만 가능) 선택한 노래를 본인의 playlist에 추가할 수 있습니다.

# (User) SEARCH MUSIC

## [CMD CAPTURE]

```

You log in as user!
What Kind of Service do you want?
1: Search All
2: Search by Music
3: Search by Artist
4: Search by Album
5: Playlist
6: Show Chart
0: Finish Program
Enter Number: 1
Search Music by All Method: ed
  artist      music      album
0  Asap Rocky  Distorted Records  TESTING
1  Daniel Caesar  Blessed  Freudian
2  Kanye West  All Of The Lights  My Beautiful Dark Twisted Fantasy
3  Kanye West  Blame Game  My Beautiful Dark Twisted Fantasy
4  Kanye West  So Appalled  My Beautiful Dark Twisted Fantasy

0: Finish Search
1: Select One
2: Search Another
Enter: 1

Enter Music Num: 2
  artist      music      album
2  Kanye West  All Of The Lights  My Beautiful Dark Twisted Fantasy

Select Your Service
1: Play
2: Put into your playlist
0: Back
Enter Service Num: 1
play!
play!
play!
play!
play!
Finish Music!

  artist      music      album
0  Asap Rocky  Distorted Records  TESTING
1  Daniel Caesar  Blessed  Freudian
2  Kanye West  All Of The Lights  My Beautiful Dark Twisted Fantasy
3  Kanye West  Blame Game  My Beautiful Dark Twisted Fantasy
4  Kanye West  So Appalled  My Beautiful Dark Twisted Fantasy

0: Finish Search
1: Select One
2: Search Another
Enter: 1

Enter Music Num: 3
  artist      music      album
3  Kanye West  Blame Game  My Beautiful Dark Twisted Fantasy

Select Your Service
1: Play
2: Put into your playlist
0: Back
Enter Service Num: 2
  pl_name  pl_index  producer
0  pl1 by user2  0  4
Keep going? yes or no: yes
Enter the number playlist: 0
Finish!!!

```

## [SQL TEXT]

> sql\_tmp=("SELECT a.name as artist,  
 m.music\_name as music, m.album FROM music  
 AS m, artist AS a WHERE a.a\_index=m.artist AND  
 (music\_name LIKE '%s' OR a.name LIKE '%s' OR  
 m.album LIKE '%s')"% (tmp,tmp,tmp))

-> 노래 검색하는 sql

> sql=("SELECT\* FROM in\_playlist WHERE  
 playlist=%s AND producer=%s AND music='%s'  
 AND album='%s' AND artist=%s"%tmp)

-> Playlist에 노래가 이미 있는지 확인하는 sql

> sql=("insert into  
 in\_playlist(playlist,producer,music,album,artist)  
 values (%s,%s,'%s','%s',%s)"%tmp)

-> Playlist에 노래를 추가하는 sql

> sql=("UPDATE date\_stream SET str\_cnt=str\_cnt+1  
 WHERE music='%s' AND artist='%s' AND  
 album='%s' AND str\_date='%s'"%inputt)

-> Streaming 횟수 update하는 sql

-> 이 sql은 playlist play에서도 사용됩니다.

# (User) Create user's playlist

## [PROGRAM CODE]

```
def create_playlist():
    global result_login
    check=input("If You really want make new Playlist, enter YES or No\n")
    if check=="No": return

    pl_name=input("Input Playlist Name: ")
    producer=result_login[0]['u_index']
    sql_pl=("SELECT MAX(pl_index) as max FROM playlist WHERE producer=%s"%producer)
    cursor.execute(sql_pl)
    result_pl=cursor.fetchall()
    pl_index=0
    if result_pl[0]['max']!=None: pl_index=result_pl[0]['max']+1
    tmp=(pl_name,pl_index,producer)
    sql_pl_insert=("insert into playlist(pl_name, pl_index, producer) values ('%s','%s', %s)"%tmp)
    cursor.execute(sql_pl_insert)
    db.commit()
    #pl_name,pl_index,producer 까지 완료 이제 들어갈 노래 생성

    global today_date
    init=True
    while init==True:
        tmp=input("Looking for music for new Playlist: ")
        #string 일부만으로도 검색가능!
        tmp="%s"%tmp
        sql_tmp=("SELECT a.name as artist, m.music_name as music, m.album FROM music AS m, artist AS a WHERE a.a_index=m.artist AND (music_name LIKE '%s' OR a")
        cursor.execute(sql_tmp)
        result_tmp=cursor.fetchall()
        if result==():
            print("There is no result")
            nono=input("Done? Yes or No\n")
            if nono=="Yes": init=False
            continue

        another=False
        while another==False:
            result_tmp=pd.DataFrame(result_)
            print(result_tmp)
            print()

            print("0: Finish Search")
            print("1: Select Musics")
            print("2: Search Another")
            select=int(input("Enter: "))
            print()
            if select==0:
                init=False
                another=True
            elif select==2: another=True
            elif select==1:
                songs=input("Enter the number you want\nYou have to use comma to write\n->")
                songs=songs.split(',')
                songs=[int(i) for i in songs]
                n=len(songs)

                for i in songs:
                    sql_artist=("SELECT a_index FROM artist WHERE name='%s'%result_[i]['artist'])
                    cursor.execute(sql_artist)
                    result_tmp=cursor.fetchall()
                    artist_n=result_tmp[0]['a_index']
                    tmp=(pl_index,producer,result_[i]['music'],result_[i]['album'],artist_n)
                    sql_inpl=("insert into in_playlist(playlist, producer, music, album, artist) values (%s,%s,'%s','%s',%s)"%tmp)
                    cursor.execute(sql_inpl)
                    db.commit()
```

## [EXPLANATION]

- ▶ Playlist를 만들 때, 우선 playlist의 이름을 input으로 받고, max함수를 사용해서 pl\_index 값의 최대값을 구한 뒤에 1을 더한 후 값을 설정합니다.
- ▶ 그 이후에 노래들을 검색해서 넣고 싶은 노래를 선택할 수 있습니다. 이 때, comma로 여러 노래를 한꺼번에 넣을 수 있도록 했습니다.
- ▶ 노래들까지 선정되면 insert sql로 추가해줍니다.

# (User) Create user's playlist

## [CMD CAPTURE]

```
Enter Number: 5
1. Create New Playlist
2. Show Your Playlist
3. Search Playlist
0. Back
Enter Number: 1
If You really want make new Playlist, enter YES or No
yes
Input Playlist Name: ex5
Looking for music for new Playlist: ed
      artist      music      album
0   Asap Rocky   Distorted Records   TESTING
1   Daniel Caesar   Blessed   Freudian
2   Kanye West   All Of The Lights   My Beautiful Dark Twisted Fantasy
3   Kanye West   Blame Game   My Beautiful Dark Twisted Fantasy
4   Kanye West   So Appalled   My Beautiful Dark Twisted Fantasy

0: Finish Search
1: Select Musics
2: Search Another
Enter: 1

Enter the number you want
You have to use comma to write
=>0,1,2,3,4
      artist      music      album
0   Asap Rocky   Distorted Records   TESTING
1   Daniel Caesar   Blessed   Freudian
2   Kanye West   All Of The Lights   My Beautiful Dark Twisted Fantasy
3   Kanye West   Blame Game   My Beautiful Dark Twisted Fantasy
4   Kanye West   So Appalled   My Beautiful Dark Twisted Fantasy

0: Finish Search
1: Select Musics
2: Search Another
Enter: 0
```

## [SQL TEXT]

› sql\_pl=("SELECT MAX(pl\_index) as max FROM playlist WHERE producer=%s"%producer)

-> pl\_index값의 최대값을 받아서 새로운 playlist의 index값을 설정합니다.

› sql\_pl\_insert=("""insert into playlist(pl\_name, pl\_index, producer) values ('%s',%s, %s)"""%tmp)

-> playlist를 insert합니다

› sql\_tmp=("SELECT a.name as artist, m.music\_name as music, m.album FROM music AS m, artist AS a WHERE a.a\_index=m.artist AND (music\_name LIKE '%s' OR a.name LIKE '%s' OR m.album LIKE '%s')"% (tmp,tmp,tmp))

-> 노래를 검색합니다.

› sql\_inpl=("""insert into in\_playlist(playlist, producer, music, album, artist) values (%s,%s,'%s',%s,%s)"""%tmp)

-> playlist에 노래들을 추가합니다.

# (User) Show(Search) user's playlist (play and delete)

## [PROGRAM CODE]

```
def show_my_playlist(): #여기서 playlist채널과 내 playlist삭제 구현
    global result_login
    global today_date
    init=True
    while init==True:
        sql_sp=("SELECT pl_name, u.name as creator, pl_index, producer FROM playlist as pl,user as u WHERE u.u_index=pl.producer AND u.u_index=%s"%result_login)
        cursor.execute(sql_sp)
        result=cursor.fetchall()
        result=pd.DataFrame(result_)
        if result==():
            print("\nThere is no result")
            init=False
            continue
        print(result)
        print()
        print("1. Choose a Playlist")
        print("0. Back")
        cmd=int(input("Enter Number: "))

        if cmd==0:
            init=False
            continue
        elif cmd==1:
            pln=int(input("Enter the number playlist you want: "))
            select_pl=result.loc[[pln],:]
            print(select_pl)
            print()
            tmp=(result[pln]['pl_index'],result[pln]['producer'])
            sql=("SELECT* FROM in_playlist WHERE playlist=%s AND producer=%s"%tmp)
            cursor.execute(sql)
            rr=cursor.fetchall()
            if rr==():
                print("There is no result")
                continue
            rr=pd.DataFrame(rr_)
            print(rr)
            print("1. Play this Playlist")
            print("2. Delete this Playlist")
            print("0. Back")
            play=int(input("Enter the Number: "))
```

```
            print("1. Play this Playlist")
            print("2. Delete this Playlist")
            print("0. Back")
            play=int(input("Enter the Number: "))

            if play==1:
                for row in rr_:
                    print("play",row['music'])
                    str_update(row['music'],row['artist'],row['album'],today_date)
                    streaming()
                    time.sleep(5)

            elif play==2:
                ch=input("Are you really going to delete this playlist? Yes or No\n")
                if ch=='yes':
                    sql_del=("DELETE FROM in_playlist WHERE playlist=%s AND producer=%s"%tmp)
                    cursor.execute(sql_del)
                    db.commit()
                    sql_del=("DELETE FROM playlist WHERE pl_index=%s AND producer=%s"%tmp)
                    cursor.execute(sql_del)
                    db.commit()
                    print("Delete done!")
```

## [EXPLANATION]

- Playlist를 보여주는 function은 show playlist와 search playlist 두 개가 있고, 여기서 search playlist는 전체 playlist에서 제목을 검색해서 playlist를 찾은 후 재생까지 할 수 있는 function이고 show playlist는 자신의 playlist만 보여주며 재생과 삭제까지 가능합니다.
- 리스트를 보여준 후에 playlist를 선택하고 그 이후에 play를 하거나 삭제를 할 수 있습니다.
- 삭제를 할 때는 in\_playlist에 있는 data들을 삭제 이후에, playlist를 삭제합니다.



# (User) Show(Search) user's playlist (play and delete)

## [CMD CAPTURE]

```
1. Choose a Playlist
0. Back
Enter Number: 1
Enter the number playlist you want: 1
pl_name creator pl_index producer
1 pl_ex TA 1 0

playlist producer music album artist
0 1 0 Distorted Records TESTING 101
1 1 0 Martini Blue Her 104
2 1 0 White Ferrari Blonde 107
3 1 0 Abu Dhabi BENTLEY 2 112
1. Play this Playlist
2. Delete this Playlist
0. Back
Enter the Number: 1
play Distorted Records
play!
play!
play!
play!
play!
play!
play Martini Blue
play!
play!
play!
play!
play!
play White Ferrari
play!
play!
play!
play!
play!
play Abu Dhabi
play!
play!
play!
play!
play!
pl_name creator pl_index producer
0 Ta playlist TA 0 0
1 pl_ex TA 1 0
2 exex TA 2 0
3 ex5 TA 3 0
1. Choose a Playlist
0. Back
Enter Number: 1
Enter the number playlist you want: 3
pl_name creator pl_index producer
3 ex5 TA 3 0

playlist producer music album artist
0 3 0 Distorted Records TESTING 101
1 3 0 Blessed Freudian 103
2 3 0 All Of The Lights My Beautiful Dark Twisted Fantasy 108
3 3 0 Blame Game My Beautiful Dark Twisted Fantasy 108
4 3 0 So Appalled My Beautiful Dark Twisted Fantasy 108
1. Play this Playlist
2. Delete this Playlist
0. Back
Enter the Number: 2
Are you really going to delete this playlist? Yes or No
yes
Delete done!
```

## [SQL TEXT]

› `sql_sp=("SELECT pl_name, u.name as creator, pl_index, producer FROM playlist as pl,user as u WHERE u.u_index=pl.producer AND u.u_index=%s"%result_login[0]['u_index'])`

-> 본인의 playlist를 검색합니다.

› `sql=("SELECT* FROM in_playlist WHERE playlist=%s AND producer=%s"%ttmp)`

-> 선택한 playlist에 수록된 노래를 검색합니다.

› `sql_del=("DELETE FROM in_playlist WHERE playlist=%s AND producer=%s"%ttmp)`

-> 우선 수록된 노래에 관한 db를 삭제합니다.

› `sql_del=("DELETE FROM playlist WHERE pl_index=%s AND producer=%s"%ttmp)`

-> playlist를 삭제합니다.

# (User) Show Chart

## [PROGRAM CODE]

```
def show_chart():
    global today_date

    servant=True
    while servant==True:
        print("1. Today's Chart")
        print("2. Monthly Chart")
        print("3. You wish to Pick spcific day or month")
        print("0. Back")
        service=int(input("Enter the num: "))

        if service==1: #today
            keep=True
            while keep==True:
                top=int(input("Data Entry: "))
                tmp=(today_date,top)
                sql=("SELECT a.name, str.music, str.album FROM date_stream AS str, artist AS a WHERE a.a_index=str.artist AND str_date LIKE '%s' ORDER BY str."
                    +str(top))
                cursor.execute(sql)
                result=cursor.fetchall()
                result=pd.DataFrame(result)
                print(result)
                print()
                nnn=int(input("Enter 0 for Back: "))
                if nnn==0: keep=False
            if nnn==0: keep=False
```

```
if service==2: #this month
    keep=True
    while keep==True:
        year=str(datetime.today().year)
        month=str(datetime.today().month)
        mo_year+'-'+month+'-'+'_'
        top=int(input("Data Entry: "))
        tmp=(mo,top)
        sql=("SELECT a.name as artist, str.music, str.album FROM date_stream AS str, artist AS a WHERE a.a_index=str.artist AND str.str_date LIKE '%s'"
            +str(tmp))
        cursor.execute(sql)
        result=cursor.fetchall()
        if result==():
            print("There is no result")
            continue
        result=pd.DataFrame(result)
        print(mo, ' Chart')
        print(result)
        print()
        nnn=int(input("Enter 0 for Back: "))
        if nnn==0: keep=False
```

```
if service==3: #wish day
    keep=True
    while keep==True:
        year=input("input year ex)2020,2019\n")
        month=input("input month ex)01,02,04,12\n")
        date=input("input date ex)15,06\n if you dont wanna input data enter '_'\n")
        mo_year+'-'+month+'-'+date
        top=int(input("Data Entry: "))
        tmp=(mo,top)
        sql=("SELECT a.name as artist, str.music, str.album FROM date_stream AS str, artist AS a WHERE a.a_index=str.artist AND str.str_date LIKE '%s'"
            +str(tmp))
        cursor.execute(sql)
        result=cursor.fetchall()
        if result==():
            print("There is no result")
            continue
        result=pd.DataFrame(result)
        print(mo, ' Chart')
        print(result)
        print()
        nnn=int(input("Enter 0 for Back: "))
        if nnn==0: keep=False

if service==0: servant=False
```

## [EXPLANATION]

- 3가지 방식으로 차트를 보여줍니다.
- 첫번째는 오늘의 차트를 보여줍니다
- 두번째는 이번달의 차트를 보여줍니다.
- 세번째는 user가 원하는 날짜의 차트를 보여줍니다. 이 때, 일자를 입력할 때, \_\_를 입력하면 해당하는 달의 차트도 볼 수 있습니다.
- 차트를 보여줄 때는 숫자를 입력받아서 top에서 원하는 수만큼을 보여줍니다.

# (User) Delete user's playlist

## [CMD CAPTURE]

```

1. Today's Chart
2. Monthly Chart
3. You wish to Pick speific day or month
0. Back
Enter the num: 1
Data Entry: 5
      name                music                album
0 Daniel Caesar          Blessed          Freudian
1      Fromm Us On a Young Night Midnight Candy
2 The Quiett              Abu Dhabi      BENTLEY 2
3 Frank Ocean             White Ferrari  Blonde
4 BlackSkirts             EVERYTHING    TEAM BABY

Enter 0 for Back: 0
1. Today's Chart
2. Monthly Chart
3. You wish to Pick speific day or month
0. Back
Enter the num: 2
Data Entry: 5
2020-12-__ Chart
      artist                music                album
0      Logic              The Return        YSIV
1      Asap Rocky          OG Beeper         TESTING
2 Thorn Apple Pomegranate taste Seoul Disease
3 The Quiett              Abu Dhabi      BENTLEY 2
4 Kendrick Lamar          Alright          To Pimp A Butterfly

Enter 0 for Back: 0
1. Today's Chart
2. Monthly Chart
3. You wish to Pick speific day or month
0. Back
Enter the num: 3
input year ex)2020,2019
2020
input month ex)01,02,04,12
12
input date ex)15,06
if you dont wanna input data enter '_'
05
Data Entry: 5
2020-12-05 Chart
      artist                music                album
0 Daniel Caesar          GET YOU          Freudian
1 Justhis                Atelier 2 MANY HOMES 4 1 KID
2 Logic                  The Return        YSIV
3 The Quiett             Came From The Bottom Stormy Friday
4 Daniel Caesar          Blessed          Freudian
  
```

## [SQL TEXT]

› sql=("SELECT a.name, str.music, str.album  
FROM date\_stream AS str, artist AS a  
WHERE a.a\_index=str.artist AND str\_date  
LIKE '%s' ORDER BY str.str\_cnt DESC  
LIMIT %s"%tmp)

-> 오늘의 chart를 가져옵니다.

› sql=("SELECT a.name as artist, str.music,  
str.album FROM date\_stream AS str, artist  
AS a WHERE a.a\_index=str.artist AND  
str.str\_date LIKE '%s' GROUP BY  
str.music, str.artist, str.album ORDER BY  
SUM(str.str\_cnt) DESC LIMIT %s"%tmp)

-> 이번달의 chart를 가져옵니다.

› sql=("SELECT a.name as artist, str.music,  
str.album FROM date\_stream AS str, artist  
AS a WHERE a.a\_index=str.artist AND  
str.str\_date LIKE '%s' GROUP BY  
str.music, str.artist, str.album ORDER BY  
SUM(str.str\_cnt) DESC LIMIT %s"%tmp)

-> 원하는 날짜의 char를 가져옵니다.

# (Admin) Manipulate User

## [PROGRAM CODE]

```
def manipulate_user(): #user add, delete, update
    mani=True
    while mani==True:
        sql=("SELECT * FROM user")
        cursor.execute(sql)
        show=cursor.fetchall()
        show=pd.DataFrame(show)
        print(show)
        print("\n1. User add")
        print("\n2. User delete")
        print("\n3. User info Update")
        print("\n0. Back")
        serr=int(input("Enter the Num: "))

        if serr==0: mani=False
        elif serr==1:
            sql=("SELECT MAX(u_index) AS max FROM user")
            cursor.execute(sql)
            user_num=cursor.fetchall()
            user_num=user_num[0]['max']
            u_index=user_num+1
            name=input("name (necessary): ")
            email=input("email: ")
            phone_num=input("phone num (necessary): ")
            bdate=input("birth date ('--/--' form) (necessary): ")
            sex=input("sex (necessary): ")
            password=input("password (necessary): ")

            tmp=(u_index,email,phone_num,name,bdate,sex,password)
            sql=("insert into user(u_index,email,phone_num,name,bdate,sex,password) values (%s, '%s', '%s', '%s', '%s', '%s', '%s')"%tmp)
            cursor.execute(sql)
            db.commit()
            print("Finish!\n")

        elif serr==2:
            del_u=int(input("Enter the index: "))
            sql1=("DELETE FROM in_playlist WHERE producer=%s"%del_u)
            sql2=("DELETE FROM playlist WHERE producer=%s"%del_u)
            sql3=("DELETE FROM user WHERE u_index=%s"%del_u)
            cursor.execute(sql1)
            cursor.execute(sql2)
            cursor.execute(sql3)
            db.commit()
            print("Finish!\n")

        elif serr==3:
            up_u=int(input("Enter the index: "))
            area=input("Update Area: ")
            new=input("Update Contents: ")
            tmp=(area,new,up_u)
            sql=("UPDATE user SET %s='%s' WHERE u_index=%s"%tmp)
            cursor.execute(sql)
            db.commit()
            print("Finish Update!")
```

## [EXPLANATION]

- › 총 3가지의 기능을 제공합니다.
- › User를 추가/삭제 할 수 있고 또한 어떤 user의 특정한 필드를 수정할 수도 있습니다.
- › User에 대한 정보들을 정해진 형식으로 친다면, user를 add할 수 있습니다.
- › User 목록을 보면서 서비스를 선택하고 표에 나오는 user index를 통해서 삭제할 user를 선택할 수 있습니다.
- › 정보 수정을 원한다면 user index를 통해서 user를 선택한 다음에 수정할 필드이름을 치고 수정할 값을 치면서 user info를 수정할 수 있습니다.

# (Admin) Manipulate User

## [CMD CAPTURE]

```
Enter Number: 4
u_index  email      phone_num  name  bdate  sex  password
0         0      abc@abc.com  ****-****-****  TA      None  F      TA123
1         1      user1@naver.com  010-1234-1234  user1  1997-03-03  M      12341234
2         3      user3@google.com  010-9876-9876  user3  2001-01-01  M      98769876
3         4      user4@hanyang.ac.kr  010-7845-7845  user2  2010-12-26  F      78457845
4         5      delu@delu.net  010-4444-4444  delu   2020-12-07  M      delu123

1. User add
2. User delete
3. User info Update
0. Back
Enter the Num: 1
name (necessary): aaa
email: aaa@aaa.com
phone num (necessary): 010-9999-9999
birth date ('____-__-__' form) (necessary): 2020-12-07
sex (necessary): F
password (necessary): aaa123
Finish!

u_index  email      phone_num  name  bdate  sex  password
0         0      abc@abc.com  ****-****-****  TA      None  F      TA123
1         1      user1@naver.com  010-1234-1234  user1  1997-03-03  M      12341234
2         3      user3@google.com  010-9876-9876  user3  2001-01-01  M      98769876
3         4      user4@hanyang.ac.kr  010-7845-7845  user2  2010-12-26  F      78457845
4         5      delu@delu.net  010-4444-4444  delu   2020-12-07  M      delu123
5         6      aaa@aaa.com  010-9999-9999  aaa    2020-12-07  F      aaa123
```

```
1. User add
2. User delete
3. User info Update
0. Back
Enter the Num: 2
Enter the index: 1
Finish!

u_index  email      phone_num  name  bdate  sex  password
0         0      abc@abc.com  ****-****-****  TA      None  F      TA123
1         3      user3@google.com  010-9876-9876  user3  2001-01-01  M      98769876
2         4      user4@hanyang.ac.kr  010-7845-7845  user2  2010-12-26  F      78457845
3         5      delu@delu.net  010-4444-4444  delu   2020-12-07  M      delu123
4         6      aaa@aaa.com  010-9999-9999  aaa    2020-12-07  F      aaa123

1. User add
2. User delete
3. User info Update
0. Back
Enter the Num: 3
Enter the index: 6
Update Area: name
Update Contents: bbb
Finish Update!

u_index  email      phone_num  name  bdate  sex  password
0         0      abc@abc.com  ****-****-****  TA      None  F      TA123
1         3      user3@google.com  010-9876-9876  user3  2001-01-01  M      98769876
2         4      user4@hanyang.ac.kr  010-7845-7845  user2  2010-12-26  F      78457845
3         5      delu@delu.net  010-4444-4444  delu   2020-12-07  M      delu123
4         6      aaa@aaa.com  010-9999-9999  bbb    2020-12-07  F      aaa123
```

## [SQL TEXT]

› sql=("SELECT MAX(u\_index) AS max FROM user")

-> 새로운 user의 u\_index를 구하기 위해서 최대값을 가져와서 1을 더해줍니다.

› sql=("insert into user(u\_index,email,phone\_num,name,bdate,sex,password) values (%s, '%s', '%s', '%s', '%s', '%s', '%s')"%tmp)

-> 새로운 user를 insert합니다.

› sql1=("DELETE FROM in\_playlist WHERE producer=%s"%del\_u)

› sql2=("DELETE FROM playlist WHERE producer=%s"%del\_u)

› sql3=("DELETE FROM user WHERE u\_index=%s"%del\_u)

-> user가 가지고 있는 playlist와 그 수록곡들 그리고 user data를 삭제하는 sql입니다.

› sql=("UPDATE user SET %s='%s' WHERE u\_index=%s"%tmp)

-> 원하는 필드의 값에 새로운 콘텐츠를 삽입합니다.

# (Admin) Manipulate Music

## [PROGRAM CODE]

```
if man==0: mani=False

elif man==1: #Add music!
    artist=input("Artist: ")
    album=input("Album: ")
    music_name=input("Music: ")

    sql=("SELECT* FROM artist WHERE NAME='%s'%artist)
    cursor.execute(sql)
    tuple_artist=cursor.fetchall()
    if tuple_artist==():
        print("There is NO artist")
    else:
        sql=("SELECT* FROM album WHERE album_name='%s' AND artist='%s'%(album,tuple_artist[0]['a_index'])")
        cursor.execute(sql)
        tuple_album=cursor.fetchall()

        if tuple_album==(): print("There is NO album")
        else:
            artist=tuple_artist[0]['a_index']
            tmp=(music_name,artist,tuple_album[0]['album_name'])
            sql=("SELECT* FROM music WHERE music_name='%s' AND artist='%s' AND album='%s'%tmp)
            cursor.execute(sql)
            tuple_music=cursor.fetchall()

            if tuple_music!=(): print("There is music already")
            else:
                tmp=(music_name,today_date,artist,tuple_album[0]['album_name'])
                sql=("insert into music(music_name,enroll_date,artist,album) values ('%s','%s','%s','%s'%tmp)
                cursor.execute(sql)
                tmp=(0,music_name,artist,tuple_album[0]['album_name'],today_date)
                sql=("insert into date_stream(str_cnt,music,artist,album,str_date) values (%s,'%s','%s','%s','%s'%tmp)
                cursor.execute(sql)
                db.commit()
                print("Add Finish!")

elif man==2: #Delete music!
    artist=input("Artist: ")
    album=input("Album: ")
    music_name=input("Music: ")

    sql=("SELECT* FROM artist WHERE NAME='%s'%artist)
    cursor.execute(sql)
    tuple_artist=cursor.fetchall()

    if tuple_artist==():
        print("There is NO artist")
    else:
        sql=("SELECT* FROM album WHERE album_name='%s'%album)
        cursor.execute(sql)
        tuple_album=cursor.fetchall()

        if tuple_album==(): print("There is NO album")
        else:
            artist=tuple_artist[0]['a_index']
            tmp=(music_name,artist,album)
            sql=("SELECT* FROM music WHERE music_name='%s' AND artist='%s' AND album='%s'%tmp)
            cursor.execute(sql)
            tuple_music=cursor.fetchall()

            if tuple_music==(): print("There is no music already")
            else:
                tmp=(music_name,artist,album)
                sql0=("DELETE FROM date_stream WHERE music='%s' AND artist='%s' AND album='%s'%tmp)
                sql1=("DELETE FROM in_playlist WHERE music='%s' AND artist='%s' AND album='%s'%tmp)
                sql2=("DELETE FROM music WHERE music_name='%s' AND artist='%s' AND album='%s'%tmp)
                cursor.execute(sql0)
                cursor.execute(sql1)
                cursor.execute(sql2)
                db.commit()
                print("Delete Finish!")
```

## [EXPLANATION]

- › Add와 Delete 2가지 기능을 제공합니다.
- › Music add시에는 artist와 album이 존재해야 하며 존재하지 않다면 add할 수 없습니다.
- › 존재한다면 값을 받은 노래 제목까지 받은 이후에 노래를 등록합니다.
- › Music delete시에는 마찬가지로 정보를 받아서 in playlist에서 노래들을 삭제하고 그 이후에 노래를 삭제합니다.

# (Admin) Manipulate Music

## [CMD CAPTURE]

```
1. Manipulate Music
2. Manipulate Artist
3. Manipulate Album
4. Manipulate User
0. Finish Program
Enter Number: 1
1. Add new Music
2. Delete Music
0. Back
Enter the Num: 1
Artist: del
Album: del album
Music: del album12
There is NO album
1. Add new Music
2. Delete Music
0. Back
Enter the Num: 1
Artist: del
Album: del music
Music: del music123
Add Finish!
```

```
1. Add new Music
2. Delete Music
0. Back
Enter the Num: 2
Artist: del
Album: del music
Music: del music123
Delete Finish!
```

## [SQL TEXT]

- sql=("SELECT\* FROM artist WHERE NAME='%s'"%artist)
- sql=("SELECT\* FROM album WHERE album\_name='%s' AND artist=%s"%(album,tuple\_artist[0]['a\_index']))
- sql=("SELECT\* FROM music WHERE music\_name='%s' AND artist=%s AND album='%s'"%tmp)  
-> 각각 아티스트,앨범,음악을 검색합니다. 이미 존재하는지 확인합니다.
- Sql=( " insert into music(music\_name,enroll\_date,artist,album) values ('%s','%s','%s','%s')"%tmp)  
-> 음악을 추가합니다.
- Sql=("insert into date\_stream(str\_cnt,music,artist,album,str\_date) values (%s,'%s','%s','%s','%s')"%tmp)  
-> 음악을 추가했기 때문에 스트리밍 횟수에 대한 db도 추가합니다.
- sql0=("DELETE FROM date\_stream WHERE music='%s' AND artist=%s AND album='%s'"%tmp)
- sql1=("DELETE FROM in\_playlist WHERE music='%s' AND artist=%s AND album='%s'"%tmp)
- sql2=("DELETE FROM music WHERE music\_name='%s' AND artist=%s AND album='%s'"%tmp)  
-> 각각 streamig과 playlist 그리고 music을 삭제하빈다.

# (Admin) Manipulate Album

## [PROGRAM CODE]

```
if tuple_album==(): print("There is no Album already")
else:
    artist=tuple_artist[0]['a_index']
    tmp=(artist,album)
    sql_music=("SELECT* FROM music WHERE artist=%s AND album='%s'"%tmp)
    cursor.execute(sql_music)
    result_music=cursor.fetchall()

    for row in result_music:
        tmptmp=(row['music_name'],album,artist)
        sql_str_=("DELETE FROM date_stream WHERE music='%s' AND album='%s' AND artist=%s"%tmptmp)
        sql_pl_del=("DELETE FROM in_playlist WHERE music='%s' AND album='%s' AND artist=%s"%tmptmp)
        cursor.execute(sql_str_)
        cursor.execute(sql_pl_del)

    db.commit()
```

## [EXPLANATION]

- › Manipulate music과 마찬가지로 add와 delete를 제공합니다.
- › Album add시에는 artis를 확인하며 존재하지 않을 시 add할 수 없습니다.
- › Delete시에는 해당 앨범에 수록된 노래들을 모두 검사하고 각각에 대해서 delete music절차를 실행하고 그리고 난 후에 album을 삭제합니다.
- › Manipulate music과 다르거나 중요한 부분만 추가했습니다.



# (Admin) Manipulate Album

[CMD CAPTURE]

```
You log in as manager!  
What Kind of Service do you want?  
1. Manipulate Music  
2. Manipulate Artist  
3. Manipulate Album  
4. Manipulate User  
0. Finish Program  
Enter Number: 3  
1. Add new Album  
2. Delete Album  
0. Back  
Enter the Num: 2  
Artist: asap rocky  
Album: testing  
Finish Delete Album  
1. Add new Album  
2. Delete Album  
0. Back  
Enter the Num: 1  
Artist: asap rocky  
Album: testing  
Add Finish!
```

[SQL TEXT]

› sql\_album\_del=("DELETE FROM  
album WHERE artist=%s AND  
album\_name='%s'"%tmp)

-> 앨범을 삭제합니다

=> 나머지 sql은 manipulate music과 동일하게 사용되었습니다.

# (Admin) Manipulate Artist

## [PROGRAM CODE]

```
def manipulate_artist(): #artist add만
    man=True
    while man==True:
        print("You can just add new artist")
        print("1. Keep going")
        print("0. Back")

        keep=int(input("Enter the Num: "))

        if keep==1:
            artist=input("Input the new artist name: ")
            sql=("SELECT* FROM artist WHERE name='%s'"%artist)
            cursor.execute(sql)
            result_a=cursor.fetchall()

            if result_a!=(): print("There is artist already")
            else:
                sql=("SELECT MAX(a_index) as max FROM artist")
                cursor.execute(sql)
                max_index=cursor.fetchall()
                a_index=max_index[0]['max']+1
                b_date=input("Input birth form (____-__-__) or None: ")

                if b_date.lower()=='none': b_date='NULL'
                else: b_date=""+b_date+" "

                tmp=(a_index,artist,b_date)

                sql=("insert into artist(a_index,name,b_date) values (%s,'%s',%s)"%tmp)
                cursor.execute(sql)
                db.commit()

                print("Add artist Finish!")
        else: man=False
```

## [EXPLANATION]

- › 오로지 add만 가능합니다. 아티스트를 delete할 만한 상황이 생 각이 나지 않았습니다.
- › 아티스트 이름을 받고 해당하는 이름과 같은 아티스트가 없다면 아티스트를 추가합니다.
- › 생일도 입력 받습니다. None을 치면 Null로 값이 들어갑니다.

# (Admin) Manipulate Artist

[CMD CAPTURE]

```
What Kind of Service do you want?
1. Manipulate Music
2. Manipulate Artist
3. Manipulate Album
4. Manipulate User
0. Finish Program
Enter Number: 2
You can just add new artist
1. Keep going
0. Back
Enter the Num: 1
Input the new artist name: asap rocky
There is artist already
You can just add new artist
1. Keep going
0. Back
Enter the Num: 1
Input the new artist name: oh sung joon
Input birth form (____-__-__) or None: 1997-03-17
Add artist Finish!
```

[SQL TEXT]

› sql=("SELECT\* FROM artist  
WHERE name='%s'"%artist)

-> 해당하는 이름의 아티스트가 이  
미 있는지 확인합니다.

› sql=("SELECT MAX(a\_index) as  
max FROM artist")

-> 새로운 artist index를 위해서 최  
대값을 받아옵니다.

› Sql=("insert into  
artist(a\_index,name,b\_date) values  
(%s,'%s',%s)"%tmp)

-> 새로운 아티스트를 db에 추가해  
줍니다.

# (자동)Date Stream 추가

## [PROGRAM CODE]

```
db=pymysql.connect(user='root',
    passwd='Curry5407!',
    host='127.0.0.1',
    db='db',
    charset='utf8')

cursor=db.cursor(pymysql.cursors.DictCursor)

#프로그램 실행할 때 마다 날짜가 지나있으면 자동으로 date 생성하는 코드
today_date=None
if datetime.today().day<10:today_date=str(datetime.today().year)+'-'+str(datetime.today().month)+'-0'+str(datetime.today().day)
else: today_date=str(datetime.today().year)+'-'+str(datetime.today().month)+'-'+str(datetime.today().day)
sql_date=("SELECT* FROM date_stream WHERE str_date LIKE '%s';"%today_date)
cursor.execute(sql_date)
result_date=cursor.fetchall()
result_date=pd.DataFrame(result_date)
if result_date.empty:
    sql_music="SELECT* FROM music;"
    cursor.execute(sql_music)
    result_music=cursor.fetchall()
    sql_str_insert="""insert into date_stream(str_cnt,music,artist,album,str_date)
    values (%s, %s, %s, %s, %s)"""
    for row in result_music:
        cnt=random.randrange(10,100)
        a=(cnt, row['music_name'], row['artist'], row['album'], today_date)
        cursor.execute(sql_str_insert,a)
    db.commit()
```

## [EXPLANATION]

- ▶ 프로그램이 시작된 이후에 오늘의 날짜를 받아 온 후에 date형의 string으로 바꿉니다.
- ▶ 그 이후에 그 날짜에 해당하는 data가 date\_stream에 없을 시에 db에 있는 모든 노래들에 관해서 해당날짜에 해당하는 date\_stream을 사용합니다.
- ▶ 원래는 0을 초기값으로 넣어야 하지만, random함수를 사용해서 스트리밍 횟수를 넣었습니다. 제가 하나하나 스트리밍해서 노래들의 스트리밍 횟수를 올리고 그것을 통해서 chart를 보는 것은 거의 불가능하다고 생각했기 때문입니다. 이를 통해서 좀 더 다양한 차트를 볼 수 있었습니다.
- ▶ 만약에 해당하는 날짜에 data가 있다면 insert를 수행하지 않습니다.

# (자동)Date Stream 추가

## [CMD CAPTURE]

- › 자동으로 실행되기 때문에 cmd결과 창은 없습니다. 프로그램 실행 후 오늘 날의 chart를 볼 수 있는 것으로 대신합니다.

## [SQL TEXT]

- › sql\_date=("SELECT\* FROM date\_stream WHERE str\_date LIKE '%s';"%today\_date)  
-> 오늘 날짜에 해당하는 data를 검색합니다. 있다면 inser는 작동하지 않습니다.
- › Sql\_music="SELECT\* FROM music;"  
-> 모든 음악 data를 가져와서 아래 sql문에서 사용합니다.
- › Sql\_str\_insert="""insert into date\_stream(str\_cnt,music,artist,album,str\_date) values (%s, %s, %s, %s, %s)""  
-> date\_stream에 data를 모두 넣어줍니다. For문을 돌면서 실행했습니다.

$\pi$

감사합니다!