

1. 기능 스크린샷

```
C:\Users\pc\IdeaProjects\musicApp\src>java -cl

-----
Who Are You?
-----
1. administrator
2. listener
3. not a member - sign up
-----
>>> 1
input your ID : 12345678
input password : 000803
Login succeeded. Hello 12345678!

-----
Menu For Admin
-----
1. add artist
2. add album
3. delete album
4. user list
5. delete user
0. quit
-----
>>> 1
Insert the artist's name : Mimi
Does he/she belong to a group? (y or n) : y
Group name : omygirl
addArtist() done.

-----
Menu For Admin
-----
1. add artist
2. add album
3. delete album
4. user list
5. delete user
0. quit
-----
>>> 1
Insert the artist's name : Chungha
Does he/she belong to a group? (y or n) : n
addArtist() done.
```

```
-----
Menu For Admin
-----
1. add artist
2. add album
3. delete album
4. user list
5. delete user
0. quit
-----
>>> 2
Insert the album's name : Hands on me
Insert artist : Chungha
Do you want to insert additional info? (y or n) : y
Insert publisher : genie music
Insert release date : 2017-06-07
New album added.
The number of songs in this album : 2
Insert song title 1 : why dont you know
Do you want to insert additional info? (y or n) : y
Insert age limit : 0
Lyrics : why dont you know?
song writer : oreo
Number of genre : 1
Input genre 1 : dance pop
Insert song title 2 : make a wish
Do you want to insert additional info? (y or n) : n
addSong() done.

-----
Menu For Admin
-----
1. add artist
2. add album
3. delete album
4. user list
5. delete user
0. quit
-----
>>> 3
Insert album name : BANANA
Insert the artist : BanHana
Delete album BANANA succeeded.

-----
Menu For Admin
-----
1. add artist
2. add album
3. delete album
4. user list
5. delete user
0. quit
-----
>>> 4

-----
hyunsoo
gustn
-----

-----
Menu For Admin
```

```

-----
Menu For Admin
-----
1. add artist
2. add album
3. delete album
4. user list
5. delete user
0. quit
-----
>>> 5
Insert the user ID : gustn
Delete user gustn succeeded.
-----
Menu For Admin
-----
1. add artist
2. add album
3. delete album
4. user list
5. delete user
0. quit
-----
>>> 4
-----
hyunsoo
-----
Menu For Admin
-----
1. add artist
2. add album
3. delete album
4. user list
5. delete user
0. quit
-----
>>> 0
Log out succeeded.

C:\Users\pc\IdeaProjects\musicApp\src>

```

```

-----
Who Are You?
-----
1. administrator
2. listener
3. not a member - sign up
-----
>>> 3
-----Sign up Page-----
Input name : Gustn Kim
Input social security number : 0008037654321
Input your age : 15
Input your sex : M
Input ID : gustn
Input password : 0803
Input comment : Hi~!
Welcome gustn!
-----
Menu For Member
-----
1. search song
2. search album
3. search artist
4. create playlist
5. see my playlists
6. play playlist
7. search other users
8. see my bookmarks
9. statistics
0. quit
-----
>>> 1
-----
Search Song
-----
1. song
2. song + artist
-----
>>> 1
Input the song title : why
-----
song : why dont you know / album : Hands on me
-----
additional menu
-----
1. listen to the song
2. see the song info
0. exit
-----
>>> 2
Input the song title : why dont you know
Input the artist : Chungha
-----
Info : why dont you know
-----
Belonging album : Hands on me
Sung by : Chungha
ageLimit : 0
genre : dance pop
[song writer info]
oreo
[lyrics]
why dont you know?
-----

```

```

-----
Menu For Member
-----
1. search song
2. search album
3. search artist
4. create playlist
5. see my playlists
6. play playlist
7. search other users
8. see my bookmarks
9. statistics
0. quit
-----
>>> 1
-----
Search Song
-----
1. song
2. song + artist
-----
>>> 2
Input the song title : make a wish
Input the artist : Chungha
-----
song : make a wish / album : Hands on me / artist : Chungha
-----
additional menu
-----
1. listen to the song
2. see the song info
0. exit
-----
>>> 1
Input the song title : make a wish
Input the artist : Chungha
-----
Listening : make a wish for the first time.
-----

```

```

-----
Menu For Member
-----
1. search song
2. search album
3. search artist
4. create playlist
5. see my playlists
6. play playlist
7. search other users
8. see my bookmarks
9. statistics
0. quit
-----
>>> 2
-----
Search Album
-----
1. album
2. album + artist
-----
>>> 1
Input the album title : PINK
-----
PINK OCEAN sung by omygirl
-----
knock knock
one step two step
B612
liar liar
I found love
-----

```

```

-----
additional menu
-----
1. listen to this album
2. see the album info
0. exit
-----
>>> 2
Input the album title : PINK OCEAN
Input the artist : omygirl
-----
PINK OCEAN sung by omygirl (2016-03-28)
published by sony music
Classification : EP
-----
knock knock
one step two step
B612
liar liar
I found love
-----
Menu For Member
-----
1. search song
2. search album
3. search artist
4. create playlist
5. see my playlists
6. play playlist
7. search other users
8. see my bookmarks
9. statistics
0. quit
-----
>>> 2
-----
Search Album
-----
1. album
2. album + artist
-----
>>> 1
Input the album title : NONSTOP
-----
NONSTOP sung by omygirl
-----
dolphin
nonstop
-----
additional menu
-----
1. listen to this album
2. see the album info
0. exit
-----
>>> 1
Input the album title : NONSTOP
Input the artist : omygirl
-----
Listening : dolphin for the first time.
Listening : nonstop for the first time.
-----

```

Menu For Member

1. search song
2. search album
3. search artist
4. create playlist
5. see my playlists
6. play playlist
7. search other users
8. see my bookmarks
9. statistics
0. quit

>>> 3

Input artist : omygirl

< omygirl >

album : NONSTOP
album : Remember Me
album : PINK OCEAN

additional menu

1. see group members
2. listen to his/her album
3. see his/her album info
0. exit

>>> 1

Input artist : omygirl
Members of Group <omygirl>

BanHana
YooA
Mimi

Menu For Member

1. search song
2. search album
3. search artist
4. create playlist
5. see my playlists
6. play playlist
7. search other users
8. see my bookmarks
9. statistics
0. quit

>>> 4

Input name of a new playlist : omgPlay
public or private : public
How many songs will you put into the playlist? : 4

Input the song title 1 : dolphin
Input the artist : omygirl
Input the song title 2 : twilight
Input the artist : omygirl
Input the song title 3 : nonstop
Input the artist : omygirl
Input the song title 4 : liar liar
Input the artist : omygirl

Menu For Member

1. search song
2. search album
3. search artist
4. create playlist
5. see my playlists
6. play playlist
7. search other users
8. see my bookmarks
9. statistics
0. quit

>>> 5

omgPlay

dolphin - omygirl
liar liar - omygirl
twilight - omygirl
nonstop - omygirl

```

-----
Menu For Member
-----
1. search song
2. search album
3. search artist
4. create playlist
5. see my playlists
6. play playlist
7. search other users
8. see my bookmarks
9. statistics
0. quit
-----
>>> 6
Input playlist name : omgPlay
Input creator ID : gustn
Listening : dolphin for the 2th time.
Listening : liar liar for the first time.
Listening : twilight for the first time.
Listening : nonstop for the 2th time.
-----

```

```

-----
Menu For Member
-----
1. search song
2. search album
3. search artist
4. create playlist
5. see my playlists
6. play playlist
7. search other users
8. see my bookmarks
9. statistics
0. quit
-----
>>> 7
Input member ID for search : hyunsoo
-----
ID : hyunsoo
comment :
[open playlists]
omypli
-----
additional menu
-----
1. bookmark this user
2. play his/her playlist
0. exit
-----
>>> 2
Input playlist name : omypli
Input creator ID : hyunsoo
Listening : Bon voyage for the first time.
-----

```

```

-----
Menu For Member
-----
1. search song
2. search album
3. search artist
4. create playlist
5. see my playlists
6. play playlist
7. search other users
8. see my bookmarks
9. statistics
0. quit
-----
>>> 7
Input member ID for search : hyunsoo
-----
ID : hyunsoo
comment :
[open playlists]
nonstoppli
omypli
-----
additional menu
-----
1. bookmark this user
2. play his/her playlist
0. exit
-----
>>> 2
Input playlist name : nonstoppli
Input creator ID : hyunsoo
Listening : dolphin for the 3th time.
Listening : nonstop for the 3th time.
-----
Menu For Member
-----
1. search song
2. search album
3. search artist
4. create playlist
5. see my playlists
6. play playlist
7. search other users
8. see my bookmarks
9. statistics
0. quit
-----
>>> 7
Input member ID for search : hyunsoo
-----
ID : hyunsoo
comment :
[open playlists]
nonstoppli
omypli
-----
additional menu
-----
1. bookmark this user
2. play his/her playlist
0. exit
-----
>>> 1
Bookmark done.
-----

```

Menu For Member

1. search song
2. search album
3. search artist
4. create playlist
5. see my playlists
6. play playlist
7. search other users
8. see my bookmarks
9. statistics
0. quit

>>> 8

my following

hyunsoo

my followers

Menu For Member

1. search song
2. search album
3. search artist
4. create playlist
5. see my playlists
6. play playlist
7. search other users
8. see my bookmarks
9. statistics
0. quit

>>> 9

statistics

1. total ranking
2. ranking by category

>>> 1

total ranking

- 1) dolphin in NONSTOP sung by omygirl
- 2) nonstop in NONSTOP sung by omygirl
- 3) energetic in TO BE ONE sung by wannaone
- 4) wanna be in TO BE ONE sung by wannaone
- 5) Bon voyage in Bon Voyage sung by YooA
- 6) make a wish in Hands on me sung by Chungha
- 7) liar liar in PINK OCEAN sung by omygirl
- 8) twilight in Remember Me sung by omygirl
- 9) burn it up in TO BE ONE sung by wannaone

Menu For Member

1. search song
2. search album
3. search artist
4. create playlist
5. see my playlists
6. play playlist
7. search other users
8. see my bookmarks
9. statistics
0. quit

>>> 9

statistics

1. total ranking
2. ranking by category

>>> 2

category

1. by age
2. by gender
3. by age & gender

>>> 1

ranking by age

Input age (ex 20) : 20

- 1) energetic in TO BE ONE sung by wannaone
- 2) wanna be in TO BE ONE sung by wannaone
- 3) nonstop in NONSTOP sung by omygirl
- 4) dolphin in NONSTOP sung by omygirl
- 5) burn it up in TO BE ONE sung by wannaone

```

-----
Menu For Member
-----
1. search song
2. search album
3. search artist
4. create playlist
5. see my playlists
6. play playlist
7. search other users
8. see my bookmarks
9. statistics
0. quit
-----
>>> 9
-----
statistics
-----
1. total ranking
2. ranking by category
-----
>>> 2
-----
category
-----
1. by age
2. by gender
3. by age & gender
-----
>>> 2
-----
ranking by gender
-----
Input gender (M/F) : M
1) nonstop in NONSTOP sung by omygirl
2) dolphin in NONSTOP sung by omygirl
3) twilight in Remember Me sung by omygirl
4) Bon voyage in Bon Voyage sung by YooA
5) make a wish in Hands on me sung by Chungha
6) liar liar in PINK OCEAN sung by omygirl
-----

```

```

-----
Menu For Member
-----
1. search song
2. search album
3. search artist
4. create playlist
5. see my playlists
6. play playlist
7. search other users
8. see my bookmarks
9. statistics
0. quit
-----
>>> 9
-----
statistics
-----
1. total ranking
2. ranking by category
-----
>>> 2
-----
category
-----
1. by age
2. by gender
3. by age & gender
-----
>>> 3
-----
ranking by age & gender
-----
Input age (ex 20) : 20
Input gender (M/F) : F
1) energetic in TO BE ONE sung by wannaone
2) wanna be in TO BE ONE sung by wannaone
3) nonstop in NONSTOP sung by omygirl
4) dolphin in NONSTOP sung by omygirl
5) burn it up in TO BE ONE sung by wannaone
-----
-----
Menu For Member
-----
1. search song
2. search album
3. search artist
4. create playlist
5. see my playlists
6. play playlist
7. search other users
8. see my bookmarks
9. statistics
0. quit
-----
>>> 0
Log out succeeded.

C:\Users\pc\IdeaProjects\musicApp\src>

```

2. 사용된 함수 설명

2-1) 함수의 구조

int whoAreYou() : admin인지, 평범한 user인지, 회원가입해야 할 방문자인지 입력 받기

void Login(int who) : admin/user인 경우 login

void signUp() : 방문자인 경우 회원가입

void menuForAdmin() : administrator의 권한 보여주기

void addArtist() : artist 추가하기

void addAlbum() : album 추가하기

void addSong(String albumIdx) : 추가할 album에 수록곡 song 추가하기

void deleteAlbum() : album 삭제하기 -> album에 수록된 song들도 모두 삭제됨

void userList() : administrator가 관리하고 있는 user목록 보기

void deleteUser() : user 삭제하기

void menuForMember() : user의 권한 보여주기

void searchSong() : song의 정보 찾아보기

void playSong(String songidx) : 찾은 song을 play하기

void songInfo(String songidx) : 찾은 song의 상세정보 보기

void searchAlbum() : album의 정보 찾아보기

void playAlbum(String albumidx) : 찾은 album을 play하기

void playSong(String songidx) : album의 song들을 play하기

void albumInfo(String albumidx) : 찾은 album의 상세정보 보기

void searchArtist() : artist의 정보 찾아보기

void memOfGroup() : 찾은 artist가 group이라면 구성원 artist들 보기

void playAlbum(String albumidx) : 찾은 artist의 album을 play하기

void playSong(String songidx) : album의 song들을 play하기

void albumInfo(String albumidx) : 찾은 artist의 album의 상세정보 보기

void createPlaylist() : 내 playlist 만들기

void playPlaylist() : 내 playlist 또는 다른 사람의 공개 playlist를 play하기

void playSong(String songidx) : playlist의 song들을 play

void seeMyPlaylists() : 내 playlist의 목록을 보기

void searchMember() : 다른 user를 검색하기

void playPlaylist() : 찾은 user의 공개 playlist를 play하기

void playSong(String songidx) : playlist의 song들을 play

void seeMyBookmarks() : 내가 follow한 user의 ID, 나를 follow한 user의 ID를 보기

void statistics() : 카테고리별 음원차트 보기

2-2) 함수의 자세한 설명

int whoAreYou()

: admin/user/visitor 입력받아서 구분하기

void Login(int who)

: admin인지 user인지 매개변수로 넘겨서 각각 알맞게 로그인하기. 전역변수 userID에 로그인한 ID 저장.

void signUp()

: 방문자인 경우 회원가입하기. 필수정보를 입력하면 고유번호가 부여된다. 역시나 설정한 ID로 userID 셋팅

void menuForAdmin()

: administrator의 권한 명령을 입력 받아서 수행. artist, album추가, 삭제, user삭제 등의 권한이 있다.

void addArtist()

: 새 artist의 이름을 입력 받고 추가한다.

void addAlbum()

: 새 album의 이름과 artist 이름을 필수적으로 입력 받고 선택정보로 발매일, 발매사, 가사를 입력 받아 추가

void addSong(String albumidx)

: 넘겨받은 album에 수록곡 추가. 필수정보는 song title, 선택정보는 가사, 제한연령, 작사작곡정보.

void deleteAlbum()

: album의 이름과 artist를 입력 받아 해당하는 album을 지운다. 앨범에 수록된 song도 함께 지워진다.

void userList()

: 현재 로그인 되어있는 administrator가 관리하고 있는 user목록 보기

void deleteUser()

: user의 ID를 입력 받아 관리 권한 확인 후 삭제한다.

void menuForMember()

: 평범한 user에게 주어진 권한을 보여주고 명령을 입력 받아 그에 맞는 함수들을 실행한다.

void searchSong()

: song의 이름과 artist를 입력 받아 입력 받은 이름이 이름에 들어가는 song의 목록을 보여준다.

void playSong(String songidx)

: song의 index를 넘겨받아 해당하는 song을 play한다. 이때 연령제한에 걸리는 song은 play할 수 없다.

void songInfo(String songidx)

: song의 index를 넘겨받아 해당하는 song의 정보를 본다. 소속앨범, 가사, 연령제한, 작곡가 정보.

void searchAlbum()

: album의 이름과 artist를 입력 받아 입력 받은 이름이 이름에 들어가는 album의 목록을 보여준다.

void playAlbum(String albumidx)

: album의 index를 넘겨받아 해당 album에 수록된 song들의 index를 playSong()에 넘겨준다.

void albumInfo(String albumidx)

: album의 index를 넘겨받아 해당 album의 정보를 본다. 발매사, 발매일, 장르 등

void searchArtist()

: artist의 이름을 입력 받아 입력 받은 이름이 이름에 들어가는 artist의 목록과 각각의 album목록을 보임

void memOfGroup()

: artist의 이름을 입력 받아 그게 그룹이라면 구성원의 목록을 보여줌

void createPlaylist()

: playlist의 이름 입력 받아 현재 user index와 playlist이름을 primary key로 가지는 weak entity, playlist 생성.

void playPlaylist()

: playlist이름과 playlist의 creator ID를 입력 받아서 해당하는 playlist를 재생할 권한이 있다면 playSong()을 호출하여 재생한다.

void seeMyPlaylists()

: 현재 로그인 되어있는 user의 playlist의 목록을 본다

void searchMember()

: 다른 user의 ID를 입력하여 검색한다. search한 user를 bookmark에 추가할 수 있다.

void seeMyBookmarks()

: 내가 follow한 user의 ID, 나를 follow한 user의 ID를 본다.

void statistics()

: 데이터베이스에 저장된 각 음원의 재생횟수 데이터를 활용하여 전체 chart 순위와 나이별, 성별별 카테고리
에 맞는 chart 순위를 보여준다. ex) 20대 여성 인기순, 10대 인기순, 남성 인기순 등

3. 사용되는 SQL문 명세

```
"SELECT adminIdx, password FROM administrator  
WHERE adminIdx='" + id+ "' AND password='" + password + "'"
```

(Login) admin의 아이디와 password를 입력 받아 해당하는 tuple에서 adminIdx와 password를 가져온다

```
("INSERT INTO members" +  
    "(name, ssn, age, ID, comment, password, memberIdx, sex) " +  
    "VALUES ('%s', '%s', %d, '%s', '%s', '%s', '%s', '%c')",  
    name, ssn, age, id, comment, pw, memIdx, sex)
```

(sign up) 회원가입 하려는 사람의 정보들을 입력 받아 회원을 저장하는 table인 members에 새 tuple을 INSERT
한다

```
("INSERT INTO artist VALUES ('%s', '%s')", name, idx)
```

(add artist) 입력 받은 artist 이름과 랜덤으로 생성한 artist index를 새 tuple로 만들어 artist를 저장하는 table인
artist에 INSERT

```
"SELECT artistIdx FROM artist WHERE name='" + groupname + "'"
```

groupname에 해당하는 artist의 index를 SELECT

```
("INSERT INTO memberof VALUES ('%s', '%s')", groupId, idx)
```

소속그룹-구성원 정보를 저장하는 table인 memberof에 artist와 그가 속하는 group의 index를 tuple로
만들어 INSERT

```
"SELECT artistIdx FROM artist WHERE name='" + artist + "'"
```

(add album) artist 이름에 해당하는 artist index를 가져온다.

```
("INSERT INTO album " +  
    "VALUES ('%s', '%s', '%s', '%s', '%s', '%s')",  
    name, publisher, idx, reldate, artistIdx, userID)
```

앨범 상세 정보를 입력 받아 앨범 정보를 저장하는 table album에 tuple로 INSERT

```
("INSERT INTO album(name,albumIdx,artistNum) " +  
    "VALUES ('%s', '%s', '%s')", name, idx, artistIdx)
```

앨범 필수 정보만 tuple로 album에 INSERT

```
("INSERT INTO song VALUES ('%s', '%s', %d, '%s', '%s', '%s')",  
    name, lyrics, ageLimit, idx, writer, albumIdx)
```

(add song) song 상세 정보를 song table에 tuple로 INSERT

```
("INSERT INTO song(name, songIdx, albumNum) VALUES ('%s', '%s', '%s')",  
name, idx, albumIdx)
```

song 필수 정보만 song에 INSERT

```
"INSERT INTO genre VALUES ('" + g + "', '" + idx + "')
```

song의 index와 장르 정보를 저장하는 table인 genre에 장르와 song index tuple을 INSERT

```
("SELECT albumIdx, admNum " +  
"FROM album JOIN artist ON artistNum=artistIdx " +  
"WHERE album.name='%s' AND artist.name='%s'", album, artist)
```

(delete album) 입력 받은 album name과 artist name에 해당하는 album index와 앨범을 관리하는 admin의 index를 SELECT

```
("DELETE FROM album WHERE albumIdx='%s'", rs.getString("albumIdx"))
```

delete권한이 확인되면 해당되는 앨범 지우기

```
"SELECT ID, memberId " +  
"FROM members WHERE admNum = '" + userID + "'"
```

(user list) 로그인 되어있는 관리자가 권한을 가진 user들의 ID와 index를 members테이블에서 골라 옴

```
"SELECT admNum FROM members WHERE ID='" + user + "'"
```

(delete user) 삭제하려는 user를 관리하는 admin index를 SELECT

```
"DELETE FROM members WHERE ID='" + user + "'"
```

delete권한이 확인되면 해당되는 user 지우기

```
"SELECT songName, albumName, artistName " +  
"FROM artist_album_songs " +  
"WHERE songName LIKE '%" + song + "%'"
```

(search song) artist와 album, song을 미리 모두 join시켜둔 view인 artist_album_songs로부터 입력 받은 song name과 일치하는 tuple에서 song name, album name, artist name을 SELECT

```
"SELECT playtime " +  
"FROM playsong " +  
"WHERE listenerNum='" + memidx + "' AND songNum='" + songidx + "'"
```

(play song) user와 song 그리고 그 user가 그 song을 play한 횟수인 playtime을 저장하는 table인 playsong에서 listener와 song에 해당하는 playtime을 SELECT

```
"UPDATE playsong SET playtime = playtime + 1 " +  
"WHERE listenerNum='" + memidx + "' AND songNum='" + songidx + "'"
```

playsong에서 listener와 song에 해당하는 playtime을 1 증가시킨다. UPDATE.

```
"SELECT songName, albumName, artistName " +  
"FROM artist_album_songs " +  
"WHERE albumName LIKE '%" + albumname + "%' AND artistName LIKE '%" + artist +  
"%'" " +  
"ORDER BY albumName, artistName"
```

(search album) artist와 album, song을 미리 모두 join시켜둔 view인 artist_album_songs로부터 입력받은 artist name과 album name이 일치하는 tuple의 song name, album name, artist name을 SELECT 하되 albumName, artistName 순서로 정렬하여 SELECT

```
"SELECT count(*) " +  
  "FROM artist_album_songs " +  
  "WHERE albumIdx = '" + albumidx + "'"
```

(album info) artist_album_songs view로부터 album index가 일치하는 tuple의 개수를 SELECT

```
"SELECT g.name, m.name " +  
  "FROM artist AS g, artist AS m, memberof " +  
  "WHERE g.artistIdx = memberof.groupNum AND m.artistIdx = memberof.memberNum AND  
  g.name='" + artistname + "'"");
```

(mem of group) artist group과 구성원의 관계를 저장하는 table인 memberof와 artist들을 recursive하게 join하여 group name이 찾는 artistname인 tuple을 찾고 group name과 member name 쌍을 SELECT

```
"SELECT i.songNum FROM included AS i, playlist AS p " +  
  "WHERE p.name=i.PLName AND p.creatorNum=i.creatorNum AND p.public=1 AND  
  p.name='" + plName + "'"");  
}
```

(play playlist) playlist에 song이 포함된 관계를 나타내는 table인 included와 playlist를 JOIN하고 공개범위가 public이면서 찾는 playlist name과 일치하는 tuple의 song index를 SELECT

```
"SELECT p.name, m.memberIdx , m.comment " +  
  "FROM members m JOIN playlist p ON m.memberIdx=p.creatorNum " +  
  "WHERE m.ID='" + otherID + "' AND public = 1"
```

(search member) member와 playlist를 JOIN해서 찾는 user의 공개 playlist의 name, user의 index, user의 comment를 SELECT

```
"INSERT INTO bookmark VALUES('%s', '%s')", otherIdx, userIdx)
```

즐거찾기한 user들의 관계를 저장하는 table인 bookmark에 즐겨찾는 user의 index와 자신의 index를 tuple로 INSERT

```
"SELECT theyID " +  
  "FROM follow WHERE themID='" + userID + "'"
```

(see my bookmarks) members와 bookmark table을 recursive하게 JOIN하여 어떤 유저가 어느 유저를 bookmark 했는지 저장하는 view인 follow로부터 로그인 된 유저를 bookmark한 유저들의 ID를 SELECT

```
"SELECT themID " +  
  "FROM follow WHERE theyID='" + userID + "'"
```

follow에서 로그인 된 유저가 bookmark한 유저들의 ID를 SELECT

```
"SELECT songNum " +  
  "FROM playsong GROUP BY songNum ORDER BY SUM(playtime) DESC"
```

(statistics) 어떤 유저가 어느 노래를 몇 번 play했는지 저장하는 table인 playsong에서 song index를 기준으로 그

roup으로 묶어 playtime을 합산하고 합산한 playtime을 기준으로 내림차순 정렬한 song num을 SELECT

```
"SELECT songNum " +  
  "FROM playsong JOIN members ON listenerNum=memberIdx " +  
  "WHERE (age - age % 10) = " + (age-age%10) +  
  " GROUP BY songNum ORDER BY sum(playtime) DESC"
```

playsong과 members를 JOIN하여 입력 받은 age의 나이대(20대, 30대 등)가 들은 song 중 song index를 기준으로 묶어 playtime을 합산한 결과를 기준으로 내림차순 정렬하여 song num을 SELECT

4. 이전 과제에서 변경된 점

- 기존의 작사 작곡 정보 table은 song의 attribute로 들어가는 것이 낫다고 판단하여 table을 삭제하고 attribute로 넣었다.
- 음원 관리자는 고유번호를 ID로 쓴다.
- 구독자는 닉네임을 ID로 대체하였다.
- 음원 관리자와 구독자 모두 password attribute가 추가되었다.
- 즐겨찾기 table에서 나를 즐겨 찾는 사람 수는 derived attribute로 처리하였다.
- 음원과 구독자의 연령제한을 나타내던 제한됨 table은 존재 자체가 불필요하다고 판단하여 삭제하였다.
- 구독자가 playlist를 play한다는 정보를 나타내던 감상함 table은 역시 존재의 필요성이 없어 삭제하였다.
- playlist에 song이 담김을 저장하던 담김 table의 attribute였던 담긴 음원 수는 derived attribute로 처리하였다.
- 음원 table에 수록곡 수 attribute도 derived attribute로 처리하였다.
- 음원에 있던 관리자no foreign key가 빠지고 album에 관리자no가 추가되어 관리자가 앨범을 통해 음원을 관리하도록 바꾸었다.
- 아티스트와 음원의 관계를 나타내던 참여함 table과 발매함 table은 의미가 중복되기도 하고 존재의 필요성도 없어서 삭제하였다. 발매함의 attribute였던 발매일은 앨범의 attribute로 들어갔다.
- 아티스트와 앨범의 관계를 앨범 table에 artist를 참조하는 foreign key로 나타냈다.