

데이터베이스시스템

프로젝트 4 : DBMS 프로그램 개발

컴퓨터전공

2013011695 정태화

1. 목표

지금까지 설계한 DB 스키마를 바탕으로, 이 데이터베이스를 기반으로 하는 어플리케이션 프로그램을 개발한다. 프로그램의 주제는 음악 데이터베이스를 관리하는 DBMS 프로그램이다.

2. Program Overview

Mode	Manager	User	Register
기능	<ul style="list-style-type: none">1. Music<ul style="list-style-type: none">- Show Music- Insert Music- Delete Music- Modify Music2. Album<ul style="list-style-type: none">- Show Album- Insert Album- Delete Album- Modify Album3. Artist<ul style="list-style-type: none">- Show Artist- Insert Artist- Delete Artist- Modify Artist4. User<ul style="list-style-type: none">- Show User- Delete User5. Manager<ul style="list-style-type: none">- Show Manager- Add Manager- Modify Manager Info- Resign Manager	<ul style="list-style-type: none">1. Search Music by Title2. Search Music by Album3. Search Music by Artist4. Show Playlists<ul style="list-style-type: none">- Select Playlist- Create Playlist- Delete Playlist- Add Music to Playlist- Delete Music from Playlist5. Show Favorite Artists6. Modify User Information7. Show TOP 10<ul style="list-style-type: none">- Music TOP 10- Artist TOP 108. Show Newly Released9. Delete User Account	<ul style="list-style-type: none">1. Register

3. 개발 환경

개발 OS : Windows 7, Mac OS X

실행 OS : Windows 7, Mac OS X

사용 언어 : JAVA

사용 IDE : IntelliJ IDEA

사용 DBMS : MySQL

4. 코드 설명

코드는 크게 3개의 소스파일로 구성되어 있다. 기본 초기 DB 세팅을 해주는 DB_connect.java, 쿼리를 이용한 동작들을 함수로 구현한 Query_operations.java, 그리고 메인함수인 Main.java이다.

[DB_connect.java]

프로그램 시작 시 초기 세팅 함수와 DB에 오류가 없는지 체크해주는 역할이 구현되어 있다.

* public Connection DB_setup() : 어플리케이션과 MySQL을 연결해주는 세팅을 해주는 함수이다. jdbc로 연결 할 localhost의 mysql 주소와, mysql root 계정의 아이디와 비밀번호를 체크하여 연결한다. 그 후, 이 프로그램이 사용할 데이터베이스인 ‘musicapp’ DB를 연결하고, 만약 없을시 생성한다.

* public void table_setup(Connection con) : 기본으로 필요한 테이블들의 존재 여부를 체크해주는 함수이다. ‘information_schema’ DB에 담겨있는 ‘table’ 테이블에 모든 테이블들의 이름이 담겨 있어 SELECT문을 이용 하여 해당 테이블들이 존재하는지 여부를 체크하였다. 없으면 CREATE로 생성한다. 테이블 별로 각각 Primary Key와 UNIQUE, 그리고 Foreign Key가 지정되어 있으며, Foreign Key는 전부 ON DELETE와 ON UPDATE 시 CASCADE로 지정되어 있다.

* public void data_ex(Connection con) : 기본 데이터들을 입력해주는 함수이다. 하드코딩으로 일일이 기본으로 들어갈 manager, user, music, album, artist 등 모든 데이터를 체크하고 없을 경우 INSERT하게 구현하였다. 그러나 최종 평가때는 data dump를 이용하여 들고오므로 그냥 검사의 의미만 있을 듯 하다.

[Query_operations.java]

기본적으로 모든 쿼리문을 이용한 함수들은 전부 PreparedStatement 자료형을 이용하였다. 이 자료형을 이용하면 조건을 이용한 쿼리문에서 조건값을 ‘?’ 로 받아온 후 물음표 안에 원하는 값을 넣어주는 setString() 함수를 이용하여 쿼리의 불필요한 중복을 줄일 수 있다.

* public void create_table(Connection con, PreparedStatement pst, ResultSet rs, String tname, String query) : 테이블 생성을 해주는 함수이다. 테이블의 존재여부를 검사하기 위한 SELECT 쿼리인 String query 와 조건값으로 들어갈 String tname 전부 받아와 테이블이 존재하는지 여부를 확인 후, 없을 경우 새로운 statement 를 만들어 테이블을 생성해준다.

* public int get_artistID(String name, Connection con) : Artist의 이름을 인자로 받아와 Artist 테이블의

Primary Key인 Artist_ID를 리턴해주는 함수이다.

- * `public int get_albumID(String name, String artist, Connection con)` : Album의 이름과 Artist의 이름을 인자로 받아와 Album 테이블의 Primary Key인 Album_ID를 리턴해주는 함수이다.
- * `public int get_musicID(String name, String artist, Connection con)` : Music의 이름과 Artist의 이름을 인자로 받아와 Music 테이블의 Primary Key인 Music_ID를 리턴해주는 함수이다.
- * `public int get(userID(String name, Connection con)` : User의 이름을 인자로 받아와 User 테이블의 Primary Key인 User_ID를 리턴해주는 함수이다.
- * `public int get_playlistID(String name, int usrid, Connection con)` : Playlist의 이름과 user의 id를 인자로 받아와 Playlist 테이블의 Primary Key인 Playlist_ID를 리턴해주는 함수이다.
- * `public void insert_person(Connection con, PreparedStatement pst_check, ResultSet rs, String query_insert, String name, String sex, String login, String pw)` : User 테이블과 Manager 테이블에 데이터를 넣기 위한 함수이다. 먼저 pst_check 쿼리로 존재여부를 검사한 후, 중복이 아닐 경우 query_insert 문자열에 있는 INSERT를 수행한다.
- * `public void insert_artist(Connection con, PreparedStatement pst_check, ResultSet rs, String query_insert, String name, String sex, String date)` : Artist 테이블에 데이터를 넣기 위한 함수이다. 먼저 pst_check 쿼리로 존재여부를 검사한 후, 중복이 아닐 경우 query_insert 문자열에 있는 INSERT를 수행한다.
- * `public void insert_album(Connection con, PreparedStatement pst_check, ResultSet rs, String query_insert, String title, String date, String artist)` : Album 테이블에 데이터를 넣기 위한 함수이다. get_artistID() 함수를 이용하여 Artist의 ID를 알아내 존재여부를 검사한 후 INSERT를 수행한다.
- * `public void insert_music(Connection con, PreparedStatement pst_check, ResultSet rs, String query_insert, String title, String Genre, String album, String artist)` : Music 테이블에 데이터를 넣기 위한 함수이다. get_artistID() 함수와 get_albumID() 함수를 이용하여 둘 다 ID를 알아내 존재여부를 검사한 후 INSERT를 수행한다.
- * `public void update_music(Connection con, PreparedStatement pst, String query, String newtitle, int mid)` : music 테이블에서 attribute들의 UPDATE를 수행하기 위한 함수이다.
- * `public void update_SS(Connection con, PreparedStatement pst, String query, String newtitle, String curid)` : manager와 user 테이블에서 attribute들의 UPDATE를 수행하기 위한 함수이다.
- * `public void update_ll(Connection con, PreparedStatement pst, String query, int artid, int mid)` : Artist 테이블에서 attribute들의 UPDATE를 수행하기 위한 함수이다.
- * `public void sync_album_artist(Connection con, PreparedStatement pst)` : album이나 artist 테이블의 UPDATE 수행 후 서로 연결되어 있는 foreign key로 인해 둘 다 문제없게 UPDATE 되었는지 동기화 해주는 함수이다.
- * `public void delete_artist(Connection con, String query, String name)` : Artist 테이블에서 원하는 데이터를 찾아 DELETE 연산을 수행하는 함수이다. String name으로 get_artistID() 함수를 이용하여 id를 가져와 DELETE의 조건으로 사용한다.
- * `public void delete_album(Connection con, String query, String title, String artist)` : Album 테이블에서 원하는 데이터를 찾아 DELETE 연산을 수행하는 함수이다. String title과 artist로 get_albumID() 함수를 이용하여 id를 가져와 DELETE의 조건으로 사용한다.
- * `public void delete_music(Connection con, String query, String title, String artist)` : Music 테이블에서 원하는 데이터를 찾아 DELETE 연산을 수행하는 함수이다. String title과 artist로 get_musicID() 함수를 이용하여 id를 가져와 DELETE의 조건으로 사용한다.

- * public void delete_user(Connection con, String query, String login) : user 테이블에서 원하는 데이터를 찾아 DELETE 연산을 수행하는 함수이다. String login으로 get(userID) 함수를 이용하여 id를 가져와 DELETE의 조건으로 사용한다.
- * public void show_music(Connection con, PreparedStatement pst, ResultSet rs) : music table의 원하는 음악들을 출력하는 SELECT 연산을 수행하는 함수이다. table의 틀이 그려져 있으며 title, genre, album, artist를 출력한다.
- * public void show_pl_music(Connection con, PreparedStatement pst, ResultSet rs) : playlist_music 테이블에서 해당 playlist_ID를 가진 플레이리스트에 속한 music을 출력하는 SELECT 연산을 수행하는 함수이다. table의 틀이 그려져 있으며 title, album, artist, playlist name을 출력한다.
- * public void make_playlist(Connection con, PreparedStatement pst_check, ResultSet rs, String query_insert, String name, String login) : String login에 해당하는 login_ID를 가진 user에 대해 playlist를 CREATE하는 연산을 수행하는 함수이다.
- * public void delete_playlist(Connection con, PreparedStatement pst_check, ResultSet rs, String query_delete, String pname, String usr) : String login에 해당하는 login_ID를 가진 user에 대해 playlist를 DELETE하는 연산을 수행하는 함수이다.
- * public void add_to_playlist(Connection con, PreparedStatement pst_check, ResultSet rs, String query_insert, String playlist, int usrid, String music, String artist) : String playlist에 해당하는 Name을 가진 playlist_ID를 찾아 해당 playlist에 String music에 해당하는 노래를 INSERT하는 함수이다.
- * public void fav_artist(Connection con, PreparedStatement pst_check, ResultSet rs, String query_insert, String user, String artist) : String artist에 해당하는 가수를 찾아 그 가수를 접속중인 user의 favorite_artist로 추가하는 함수이다. INSERT를 수행하기 전에 먼저 중복이 존재하는지 체크하고 없을 경우에만 INSERT를 수행한다.
- * public void fav_delete(Connection con, PreparedStatement pst_check, ResultSet rs, String query_insert, String user, String artist) : String artist에 해당하는 가수를 찾아 그 가수를 접속중인 user의 favorite_artist에서 제거하는 함수이다. INSERT를 수행하기 전에 먼저 해당 가수가 존재하는지 체크하고 없을 경우에만 DELETE를 수행한다.
- * public void newRelease(Connection con, PreparedStatement pst, ResultSet rs, String query) : newly released 기능을 수행하기 위한 함수로, 앨범의 발매일 기준으로 정렬하여 새로나온 음악을 10개까지 출력해주는 함수이다.
- * public void pop_artist(Connection con, PreparedStatement pst, ResultSet rs) : Artist의 인기순위를 출력하는 함수로, popularity 기준으로 정렬하여 내림차순으로 10명까지 출력한다.
- * public void pop_music(Connection con, PreparedStatement pst, ResultSet rs) : Music의 인기순위를 출력하는 함수로, popularity 기준으로 정렬하여 내림차순으로 10명까지 출력한다.

[Main.java]

main 함수 내에 프로그램의 인터페이스를 전부 구현하였다. 전체적인 구성은 input으로 원하는 메뉴의 숫자를 입력 받아 if ~ else 구문으로 들어가 해당 기능을 실행하는 방식이다. Main.java의 코드는 절차적으로 구성되어 있어 코드 설명도 절차적으로 작성하였다. 또한 모든 메뉴는 while(true) 안에 구현되어 있어 원하는 조건을 만족하지 않는 한 해당 메뉴가 계속 반복되게 되어있다. 모든 메뉴에서는 back to prev 선택지가 존재하여 while문을 탈출하여 이전 메뉴로 돌아갈 수 있다.

1. Manager

매니저 모드로 들어가게 되면 ID와 password를 입력받는 로그인 창이 나타난다. 매니저의 ID와 password는 user 와 별개로 Manager 테이블에서 찾는다. 로그인에 성공할 경우 매니저 모드에 들어가게 되며 실패할 경우 break로 다시 초기 메뉴로 돌아가게 된다. 매니저 모드에 접속하게 되면 매니저가 접근할 수 있는 데이터베이스 table들의 선택지가 나타난다.

1-1) Music

Music 테이블에 대한 SELECT, INSERT, DELETE, UPDATE 기능을 제공한다.

- Show Music : SELECT 연산을 이용하여 모든 음악의 목록을 출력한다.
- Insert Music : Title, Genre, Album, Artist를 입력받아 중복여부 체크 후 음악을 추가한다.
- Delete Music : Title, Artist를 입력받아 DELETE로 해당 음악을 삭제한다.
- Modify Music : Title, Artist를 입력받아 해당 음악의 attribute를 갱신한다. 음악의 Title, Genre, Album 의 UPDATE가 가능하다. Artist의 수정은 음악이 아닌 Album 메뉴에서 이루어진다.

1-2) Album

Album 테이블에 대한 SELECT, INSERT, DELETE, UPDATE 기능을 제공한다.

- Show Album : SELECT 연산을 이용하여 모든 앨범의 목록을 출력한다.
- Insert Album : Title, Released Date, Artist를 입력받아 중복여부 체크 후 앨범을 추가한다.
- Delete Album : Title, Artist를 입력받아 DELETE로 해당 앨범을 삭제한다.
- Modify Album : Title, Artist를 입력받아 해당 앨범의 attribute를 갱신한다. 앨범의 Title, Released Date, Artist의 UPDATE가 가능하다.

1-3) Artist

Artist 테이블에 대한 SELECT, INSERT, DELETE, UPDATE 기능을 제공한다.

- Show Artist : SELECT 연산을 이용하여 모든 아티스트의 목록을 출력한다.
- Insert Artist : Name, Sex, Debut Date를 입력받아 중복여부 체크 후 아티스트를 추가한다.
- Delete Artist : Title, Artist를 입력받아 DELETE로 해당 아티스트를 삭제한다.
- Modify Album : Artist의 이름을 입력받아 해당 앨범의 attribute를 갱신한다. 아티스트의 Name, Sex, DebutDate의 UPDATE가 가능하다.

1-4) User

User 테이블에 대한 SELECT, DELETE 기능을 제공한다. INSERT는 유저들이 각자 가입함으로써 수행하고, UPDATE도 유저 본인이 자신의 정보를 바꿀 때만 수행한다.

- Show User : SELECT 연산을 이용하여 모든 유저의 목록을 출력한다.
- Delete User : 삭제할 유저의 login id를 입력하여 해당 유저를 삭제한다.

1-5) Manager

Manager 테이블에 대한 SELECT, INSERT, DELETE, UPDATE 기능을 제공한다. DELETE와 UPDATE는 자기 자신에 대해서만 수행이 가능하다.

- Show Manager : SELECT 연산을 이용하여 모든 매니저의 목록을 출력한다.

- Add Manager : ID, Password, Name, Sex를 입력받아 중복 체크 후 매니저를 새로 추가한다.
- Modify Manager : 현재 접속중인 매니저 본인의 정보를 수정한다. Name, Sex, Password의 UPDATE 가 가능하다.
- Resign Manager : 현재 접속중인 매니저 본인의 계정 정보를 삭제한다. ‘RESIGN’을 입력받으면 진짜로 DELETE를 수행한다.

2. User

유저 모드도 역시 접속 인증을 위한 ID와 password를 입력받는다. 유저도 역시 매니저와는 별개로 user 테이블에서 정보를 찾는다. 로그인에 성공할 경우 유저 모드에 들어가게 되며 실패할 경우 break로 다시 초기 메뉴로 돌아가게 된다. 유저 모드에 접속하게 되면 유저가 수행할 수 있는 기능들의 선택지가 나타난다. 유저는 매니저와는 달리 테이블에 대한 직접적인 삽입, 삭제, 수정보다는 개인의 플레이리스트 추가나 좋아하는 가수 추가, 또한 조건 별로 원하는 음악들을 볼 수 있는 SELECT 연산의 수행이 가능하다.

2-1) Search Music by Title

입력한 문자열에 해당하는 제목을 가진 음악을 SELECT 연산으로 모두 출력한다. 같은 제목이지만 가수가 다르면 모두 출력된다.

2-2) Search Music by Album

입력한 문자열에 해당하는 앨범 이름을 가진 음악을 SELECT 연산으로 모두 출력한다. 다시 말해 해당 앨범의 수록곡을 모두 출력한다.

2-3) Search Music by Artist

입력한 문자열에 해당하는 가수 이름을 가진 음악을 SELECT 연산으로 모두 출력한다. 다시 말해 해당 가수의 모든 음악을 출력한다.

2-4) Show Playlists

현재 접속한 사용자의 user_ID를 가진 모든 플레이리스트를 SELECT 연산으로 보여준다. 그 후 이어지는 동작의 선택지가 제공된다.

- Select Playlist : 추가 연산을 수행할 Playlist를 선택한다. 그 후 해당 플레이리스트에 등록된 모든 노래가 출력되고, Add Music 과 Delete Music 기능을 수행할 수 있다.
- Create Playlist : 접속한 사용자의 새로운 Playlist를 생성한다. 서로 다른 사용자들 간에는 플레이리스트 이름이 같을 수는 있지만 한 사용자의 서로 다른 두 플레이리스트 이름이 같을 수는 없게 중복을 체크한다.
- Delete Playlist : 문자열을 입력받아 접속한 사용자의 Playlist 중 해당 이름을 가진 Playlist를 삭제한다.

2-5) Show Favorite Artists

현재 접속한 사용자가 설정해둔 Favorite Artist를 SELECT 연산으로 전부 출력한다. 그 후 이어지는 동작의 선택지가 제공된다.

- Add Artist : 문자열을 입력받아 이미 설정되었는지 중복 체크 후 해당 이름을 가진 Artist를 현재 접속중인 사용자의 선호 아티스트로 추가한다.
- Delete Artist : 문자열을 입력받아 해당 이름을 가진 Artist를 현재 접속중인 사용자의 선호 아티스트에서 제거한다.

2-6) Modify User Info

현재 접속한 사용자의 정보를 수정할 수 있다. 해당 사용자의 Name, Sex, Password의 UPDATE가 가능하다.

2-7) Show Ranks

현재 데이터베이스에서의 Music 테이블과 Artist 테이블에 대한 Ranking을 SELECT 연산으로 보여준다. 랭크는 TOP 10으로 출력되며, popularity 앤트리뷰트 기준으로 Music TOP 10과 Artist TOP 10을 선택할 수 있다.

- Music : 1위부터 10위까지 음악 차트를 출력한다.
- Artist : 1위부터 10위까지 인기 가수 순위를 출력한다.

2-8) Newly Released

현재 데이터베이스에서의 Music 테이블에 대해 해당 노래들이 속한 Album의 ReleasedDate를 기준으로 출력하여 새로 나온 노래를 SELECT 연산으로 보여준다. 새로 나온 노래는 10개까지 출력된다.

2-9) Delete User Account

현재 접속중인 사용자 정보를 user 테이블에서 삭제한다. 다시 말해 탈퇴 기능이다. 이 메뉴를 선택한 후 DEACTIVATE를 입력하면 해당 사용자에 대한 DELETE 연산을 수행한다.

3. Register

새로운 user로 회원가입을 하는 메뉴이다. 먼저 가입하고 싶은 ID와 PW만 입력하여 user 테이블에서 중복 여부를 체크한 후, 중복이 아닐 경우 나머지 정보인 Name, Sex, login_ID, password를 입력받아 가입을 완료한다.

4. 사용된 SQL문 명세

정말 많은 수의 SELECT, INSERT, DELETE, UPDATE 문이 구현되어 있다. 단순 쿼리문 뿐 아니라 SELECT와 함께 JOIN들도 이곳 저곳에서 사용되었다. [Main.java]를 따라가며 기능별로 사용된 SQL문에 대한 명세를 작성한다. (WHERE 등의 조건에서 입력받은 값마다 바뀌는 부분은 ?로 표기하였다)

1. Database Setup

기본적으로 관계 타입에 의해 생긴 playlist_music과 favorite_artist 테이블을 제외하고는 모든 테이블은 AUTO_INCREMENT 기능을 이용한 id를 Primary Key로 갖는다. 또한, Foreign Key를 갖는 모든 테이블은 제약조건에 ON DELETE와 ON UPDATE 모두 cascade로 설정하여 원래 해당 키의 변경을 따라가게 설정하였다.

```
* CREATE DATABASE musicapp
* CREATE TABLE manager (Name varchar(30) NOT NULL, Sex char, MgrStartDate date, Mgr_ID int NOT NULL AUTO_INCREMENT, Login_ID varchar(30) NOT NULL, Password varchar(30) NOT NULL,
PRIMARY KEY(Mgr_ID),
UNIQUE(Login_ID))
* CREATE TABLE user (Name varchar(30) NOT NULL, Sex char, JoinedDate date, User_ID int NOT NULL AUTO_INCREMENT, Login_ID varchar(30) NOT NULL, Password varchar(30) NOT NULL,
PRIMARY KEY(User_ID),
```

```

UNIQUE(Login_ID))

* CREATE TABLE artist (Name varchar(30) not null, Sex char, DebutDate date, Artist_ID int NOT
NULL AUTO_INCREMENT,
PRIMARY KEY(Artist_ID),
UNIQUE(Name))

* CREATE TABLE album (Title varchar(30) not null, ReleasedDate date, Album_ID int NOT NULL
AUTO_INCREMENT, Artist_ID int,
PRIMARY KEY(Album_ID),
FOREIGN KEY(Artist_ID) REFERENCES artist(Artist_ID) ON DELETE cascade ON UPDATE
cascade)

* CREATE TABLE music (Title varchar(30) not null, Genre varchar(10), Music_ID int NOT NULL
AUTO_INCREMENT, Album_ID int, Artist_ID int,
PRIMARY KEY(Music_ID),
FOREIGN KEY(Album_ID) REFERENCES album(Album_ID) ON DELETE cascade ON UPDATE
cascade,
FOREIGN KEY(Artist_ID) REFERENCES artist(Artist_ID) ON DELETE cascade ON UPDATE
cascade)

* CREATE TABLE playlist (Name varchar(30) not null, Playlist_ID int NOT NULL AUTO_INCREMENT,
Owner_ID int,
PRIMARY KEY(Playlist_ID),
FOREIGN KEY(Owner_ID) REFERENCES user (User_ID) ON DELETE cascade ON UPDATE
cascade)

* CREATE TABLE playlist_music (Playlist_ID int NOT NULL, Music_ID int NOT NULL,
PRIMARY KEY(Playlist_ID, Music_ID),
FOREIGN KEY(Playlist_ID) REFERENCES playlist(Playlist_ID) ON DELETE cascade ON UPDATE
cascade,
FOREIGN KEY(Music_ID) REFERENCES music (Music_ID) ON DELETE cascade ON UPDATE
cascade)

* CREATE TABLE favorite_artist (User_ID int NOT NULL, Artist_ID int NOT NULL,
PRIMARY KEY(User_ID, Artist_ID),
FOREIGN KEY(User_ID) REFERENCES user(User_ID) ON DELETE cascade ON UPDATE
cascade,
FOREIGN KEY(Artist_ID) REFERENCES artist(Artist_ID) ON DELETE cascade ON UPDATE
cascade)

```

2. Manager

< Login Operation >

```
* SELECT * FROM manager WHERE Login_ID = ? AND Password = ?
```

2-1) Music

```
* SELECT music.Title, Genre, album.Title as Album, artist.Name as Artist FROM music
JOIN album ON music.Album_ID = album.Album_ID
```

```

JOIN artist ON album.Artist_ID = artist.Artist_ID
* SELECT * FROM music WHERE Title = ? AND Album_ID = ? AND Artist_ID = ?
* INSERT INTO music(Title, Genre, Album_ID, Artist_ID) VALUES (?, ?, ?, ?)
* DELETE FROM music WHERE Music_ID = ?
* UPDATE music SET (    ) = ? WHERE Music_ID = ?

```

2-2) Album

```

* SELECT Title, ReleasedDate, artist.name AS Artist FROM album
JOIN artist ON album.artist_ID = artist.artist_ID
* SELECT * FROM album WHERE Title = ? AND Artist_ID = ?\n
* INSERT INTO album(Title, ReleasedDate, Artist_ID) VALUES (?, ?, ?)
* DELETE FROM album WHERE Album_ID = ?
* UPDATE album SET (    ) = ? WHERE Album_ID = ?

```

2-3) Artist

```

* SELECT Name, Sex, DebutDate FROM artist
* INSERT INTO artist(Name, Sex, DebutDate) VALUES (?, ?, ?)
* DELETE FROM artist WHERE Name = ?
* UPDATE artist SET (    ) = ? WHERE Artist_ID = ?

```

2-4) User

```

* SELECT Name, Sex, JoinedDate, Login_ID, Password from user
* DELETE FROM user WHERE Login_ID = ?

```

2-5) Manager

```

* SELECT Name, Sex, MgrStartDate, Login_ID, Password FROM manager
* SELECT * FROM manager WHERE Name = ? AND Login_ID = ?
* INSERT INTO manager(Name, Sex, MgrStartDate, Login_ID, Password) VALUES (?, ?, now(), ?, ?)
* UPDATE manager SET (    ) = ? WHERE Login_ID = ?
* DELETE FROM manager WHERE Login_ID = ?

```

3. User

< Login Operation >

```
* SELECT * FROM user WHERE Login_ID = ? AND Password = ?  
* SELECT User_ID FROM user WHERE Login_ID =?
```

3-1) Search Music by Title

```
* SELECT music.Title, Genre, album.Title as Album, artist.Name as Artist FROM music  
JOIN album ON music.Album_ID = album.Album_ID  
JOIN artist ON album.Artist_ID = artist.Artist_ID " + "WHERE music.Title = ?
```

3-2) Search Music by Album

```
* SELECT music.Title, Genre, album.Title as Album, artist.Name as Artist FROM music  
JOIN album ON music.Album_ID = album.Album_ID  
JOIN artist ON album.Artist_ID = artist.Artist_ID " + "WHERE album.Title = ?
```

3-3) Search Music by Artist

```
* SELECT music.Title, Genre, album.Title as Album, artist.Name as Artist FROM music  
JOIN album ON music.Album_ID = album.Album_ID  
JOIN artist ON album.Artist_ID = artist.Artist_ID " + "WHERE album.Title = ?
```

3-4) Show Playlists

```
* SELECT Name as Playlist FROM playlist WHERE Owner_ID = ?  
  
* SELECT music.Title as Title, album.Title as Album, artist.Name as Artist, playlist.Name as Playlist  
FROM music  
JOIN album ON music.Album_ID = album.Album_ID  
JOIN artist ON music.Artist_ID = artist.Artist_ID  
JOIN playlist_music ON music.Music_ID = playlist_music.Music_ID  
JOIN playlist ON playlist_music.Playlist_ID = playlist.Playlist_ID  
WHERE playlist_music.Playlist_ID = ?  
  
* SELECT * FROM playlist_music WHERE Playlist_ID = ? AND Music_ID = ?  
  
* INSERT INTO playlist_music(Playlist_ID, Music_ID) VALUES (?, ?)  
  
* DELETE FROM playlist_music WHERE Playlist_ID = ? AND Music_ID = ?  
  
* SELECT * FROM playlist WHERE Name = ? AND Owner_ID = ?  
  
* INSERT INTO playlist(Name, Owner_ID) VALUES(?, ?)  
  
* DELETE FROM playlist WHERE Name = ? AND Owner_ID = ?
```

3-5) Show Favorite Artists

```
* SELECT artist.Name FROM favorite_artist, artist WHERE favorite_artist.Artist_ID = artist.Artist_ID  
AND User_ID = ?  
  
* INSERT INTO favorite_artist(User_ID, Artist_ID) VALUES (?, ?)
```

```
* DELETE FROM favorite_artist WHERE User_ID = ? AND Artist_ID = ?
```

3-6) Modify User Information

```
* UPDATE manager SET ( ) = ? WHERE Login_ID = ?  
: User 테이블에서 해당 Login_ID 값을 갖는 튜플의 attribute를 ?에 지정된 값으로 갱신한다.
```

3-7) Show TOP 10

```
* SELECT music.Title as Music, album.Title as Album, artist.Name as Artist,  
COUNT(if(playlist_music.Music_ID=music.Music_ID,1,null)) as popularity  
FROM music, playlist_music, album, artist  
WHERE music.Album_ID=album.Album_ID AND album.Artist_ID=artist.Artist_ID  
GROUP BY music.Music_ID ORDER BY popularity DESC LIMIT 10  
: music이 playlist에서 참조된 횟수만큼 인기도를 COUNT로 측정하여 popularity라는 attribute로 만들어 출력하였다. Music_ID 별로 그룹화하여 인기도를 계산할 수 있게 하였고 popularity 순으로 10개까지 정렬하였다.  
  
* SELECT Name, Sex, DebutDate, COUNT(if(favorite_artist.Artist_ID=artist.Artist_ID, 1, null)) as Popularity FROM artist, favorite_artist GROUP BY artist.Artist_ID ORDER BY popularity DESC LIMIT 10  
: artist가 user들에 의해 favorite_artist로 설정되어 참조된 횟수만큼 인기도를 COUNT로 측정하여 popularity라는 attribute로 만들어 출력하였다. Artist_ID 별로 그룹화하여 인기도를 계산할 수 있게 하였고 popularity 순으로 10개까지 정렬하였다.
```

3-8) Show Newly Released

```
* SELECT music.Title AS Music, album.Title AS Album, artist.Name AS Artist, album.ReleasedDate  
FROM music, album, artist  
WHERE music.Album_ID=album.Album_ID AND album.Artist_ID=artist.Artist_ID  
ORDER BY album.ReleasedDate DESC LIMIT 10  
: album을 ReleasedDate 기준으로 정렬하여 신곡을 10개까지 출력하였다.
```

3-9) Delete User Account

```
* DELETE FROM user WHERE Login_ID = ?  
: 현재 접속중인 user의 Login_ID 값을 만족하는 튜플을 찾아 삭제한다. 탈퇴 용도로 쓰이는 쿼리이다.
```

4. Register

```
* INSERT INTO user(Name, Sex, JoinedDate, Login_ID, Password) VALUES(?, ?, now(), ?, ?)  
: user 테이블에서 해당 attribute들의 값을 정하여 데이터를 삽입한다. 가입 용도로 쓰이는 쿼리이다.
```

5. 기능 스크린샷

<Mode Select>

: 처음 진입 시에 나오는 화면이다. 매니저모드, 유저모드, 가입모드를 선택할 수 있다.

<Manager Mode>

```
<Mode select>
1. Manager
2. User
3. Register
0. Exit
Input : 1

<Log in>
ID : □ a
PW : a
ID or Password is wrong. Please check again.

<Mode select>
1. Manager
2. User
3. Register
0. Exit
Input : 1

<Log in>
ID : admin
PW : 1234
Manager1 Permission Granted.

[MANAGER MODE]
Select Database to control.
1. Music
2. Album
3. Artist
4. User
5. Manager
0. Log out
Input : 1
```

: 매니저모드로 login에 성공하면 '<manager name> Permission Granted' 가 출력되고 매니저모드에 진입한다. 로그인 정보가 맞지 않을 경우 틀렸다는 멘트를 출력하고 다시 이전 메뉴로 돌아가게 된다. 매니저 모드에 진입하면 위와 같이 5개의 메뉴와 로그아웃의 선택지가 나온다.

<Manager Mode> - Music

```
[MANAGER MODE]
Select command to execute.
1. Show Music
2. Insert Music
3. Delete Music
4. Modify Music
0. Back to previous menu
Input : 1
```

: Music DB 메뉴로 들어가면 위와 같이 4개의 선택지가 나온다.

<Manager Mode> - Music - Show Music

[MANAGER MODE]			
Select command to execute.			
1. Show Music			
2. Insert Music			
3. Delete Music			
4. Modify Music			
0. Back to previous menu			
Input : 1			
Title	Genre	Album	Artist
Chasing Pavements	Ballad	19	Adele
Rolling In The Deep	Ballad	21	Adele
Set Fire To The Rain	Ballad	21	Adele
Someone Like You	Ballad	21	Adele
Hello	Ballad	25	Adele
When We Were Young	Ballad	25	Adele
Marvin Gaye	R&B	Some Type of Love	Charlie Puth
Some Type of Love	R&B	Some Type of Love	Charlie Puth
One Call Away	Pop	Nine Track Mind	Charlie Puth
We Don't Talk Anymore	Pop	Nine Track Mind	Charlie Puth
River	Pop	Nine Track Mind	Charlie Puth
Good Day	Dance	Real	IU
Voice Mail	Ballad	Modern Times	IU
23	Ballad	CHAT-SHIRE	IU
This Love	Rock	Songs About Jane	Maroon 5
She Will Be Loved	Rock	Songs About Jane	Maroon 5
Sunday Morning	Rock	Songs About Jane	Maroon 5
Makes Me Wonder	Rock	It Won't Be Soon Before Long	Maroon 5
Wake Up Call	Rock	It Won't Be Soon Before Long	Maroon 5
Misery	Rock	Hands All Over	Maroon 5
Give A Little More	Rock	Hands All Over	Maroon 5
Stutter	Rock	Hands All Over	Maroon 5
Moves Like Jagger	Rock	Hands All Over	Maroon 5
One More Night	Rock	Overexposed	Maroon 5
Payphone	Rock	Overexposed	Maroon 5
Lucky Strike	Rock	Overexposed	Maroon 5
Happiness	Dance	Happiness	Red Velvet
Dumb Dumb	Dance	The Red	Red Velvet
Rookie	Dance	Rookie	Red Velvet
CHEER UP	Dance	PAGE TWO	TWICE
Touchdown	Pop	PAGE TWO	TWICE
TT	Dance	TWICEcoaster : LANE 1	TWICE
JELLY JELLY	Dance	TWICEcoaster : LANE 1	TWICE
LIKEY	Dance	twicetagram	TWICE
MISSING U	Dance	twicetagram	TWICE

: Show Music 선택 시 Music 테이블의 모든 음악을 출력한다. 장르와 앨범, 아티스트명도 함께 출력한다.

<Manager Mode> - Music - Insert Music

[MANAGER MODE]			
Select command to execute.			
1. Show Music			
2. Insert Music			
3. Delete Music			
4. Modify Music			
0. Back to previous menu			
Input : 2			
[INSERT Music]			
Title : Tickets			
Genre : Pop			
Album : Overexposed			
Artist : a			
ERROR : Artist does not exist.			

[INSERT Music]			
Title : Tickets			
Genre : Pop			
Album : Overexposed			
Artist : Maroon 5			
[Tickets] INSERT DONE			

[MANAGER MODE]			
Select command to execute.			
1. Show Music			
2. Insert Music			
3. Delete Music			
4. Modify Music			
0. Back to previous menu			
Input : 2			
Title	Genre	Album	Artist
One More Night	Rock	Overexposed	Maroon 5
Payphone	Rock	Overexposed	Maroon 5
Lucky Strike	Rock	Overexposed	Maroon 5
Tickets	Pop	Overexposed	Maroon 5
Happiness	Dance	Happiness	Red Velvet
Dumb Dumb	Dance	The Red	Red Velvet

: Insert Music 선택 시 위와 같이 music 테이블에 삽입할 데이터의 정보를 입력받는다. 이 때, 앨범이나 아티스트가

존재하지 않는 데이터일 경우 에러를 출력한다.

위와 같이 Show Music으로 다시 들어가면 데이터가 삽입됐음을 확인할 수 있다.

<Manager Mode> - Music - Delete Music

```
[DELETE Music]
Title : □ a
Artist : a
ERROR : Artist does not exist.
ERROR : Music does not exist.
```

```
[DELETE Music]
Title : Tickets
Artist : Maroon 5
[Tickets] DELETED
```

One More Night	Rock	Overexposed	Maroon 5	
Payphone	Rock	Overexposed	Maroon 5	
Lucky Strike	Rock	Overexposed	Maroon 5	
Happiness	Dance	Happiness	Red Velvet	
Dumb Dumb	Dance	The Red	Red Velvet	

: Delete Music 선택 시 삭제할 음악의 제목과 아티스트를 입력받는다. 존재하지 않는 정보일 경우 에러메시지를 출력한다. 존재하는 정보일 경우 삭제가 완료된다.

위와 같이 Show Music으로 다시 들어가면 데이터가 삭제됐음을 확인할 수 있다.

<Manager Mode> - Music - Modify Music

```
[UPDATE Music]
Select Music to update
Title : □ a
Artist : a
ERROR : Artist does not exist.
ERROR : Music does not exist.
```

```
[UPDATE Music]
Select Music to update
Title : Touchdown
Artist : TWICE

Select attribute to update
1. Title
2. Genre
3. Album
0. Back to previous menu
Input : 2
```

```
Input new genre to update
Genre : Dance
```

```
MODIFIED TO [Dance]
```

Rookie	Dance	Rookie	Red Velvet	
CHEER UP	Dance	PAGE TWO	TWICE	
Touchdown	Dance	PAGE TWO	TWICE	
TT	Dance	TWICEcoaster : LANE 1	TWICE	

: Modify Music 선택 시 수정할 음악의 제목과 아티스트를 입력받는다. 존재하지 않는 정보일 경우 에러메시지를 출력한다. 존재하는 정보일 경우 수정할 attribute의 선택지를 제공한다. 예시로 Genre를 변경해보았다.

<Manager Mode> - Album

```
[MANAGER MODE]
Select command to execute.
1. Show Album
2. Insert Album
3. Delete Album
4. Modify Album
0. Back to previous menu
Input : 1
```

<Manager Mode> - Album - Show Album

Title	Released Date	Artist
19	2008-01-27	Adele
21	2011-01-21	Adele
25	2015-11-20	Adele
Some Type of Love	2015-05-06	Charlie Puth
Nine Track Mind	2016-11-11	Charlie Puth
Real	2010-12-09	IU
Modern Times	2013-10-08	IU
CHAT-SHIRE	2015-10-23	IU
Songs About Jane	2003-09-06	Maroon 5
It Won't Be Soon Before Long	2007-05-22	Maroon 5
Hands All Over	2011-07-14	Maroon 5
Overexposed	2012-06-25	Maroon 5
Happiness	2014-08-04	Red Velvet
The Red	2015-09-09	Red Velvet
Rookie	2017-02-01	Red Velvet
PAGE TWO	2016-04-25	TWICE
TWICEcoaster : LANE 1	2016-10-24	TWICE
twicetagram	2017-10-30	TWICE

: Show Album 선택 시 모든 앨범을 출력한다.

<Manager Mode> - Album - Insert Album

[INSERT Album]
Title : d
Released Date : d
Artist : d
ERROR : Artist does not exist.
[INSERT Album]
Title : aaaaaaa
Released Date : 11111
Artist : Adele
[aaaaaaa] INSERT DONE
25 2015-11-20 Adele
aaaaaaa 2011-11-01 Adele
Some Type of Love 2015-05-06 Charlie Puth

: Insert Album 선택 시 위와 같이 music 테이블에 삽입할 데이터의 정보를 입력받는다. 이 때, 앨범이나 아티스트가 존재하지 않는 데이터일 경우 에러를 출력한다.

<Manager Mode> - Album - Delete Album

[DELETE Album]
Title : aaa
Artist : aaa
ERROR : Artist does not exist.
ERROR : Album does not exist.
[DELETE Album]
Title : aaaaaaa
Artist : Adele
[aaaaaaa] DELETED

: Delete Album 선택 시 삭제할 음악의 제목과 아티스트를 입력받는다. 존재하지 않는 정보일 경우 에러메시지를 출력한다. 존재하는 정보일 경우 삭제가 완료된다.

<Manager Mode> - Album - Modify Album

```
[MANAGER MODE]
Select command to execute.
1. Show Album
2. Insert Album
3. Delete Album
4. Modify Album
0. Back to previous menu
Input : 4

[UPDATE Album]
Select Album to update
Title : 21
Artist : Adele

Select attribute to update
1. Title
2. Released Date
3. Artist
0. Back to previous menu
Input : 1

Input new title to update
Title : 21

MODIFIED TO [21]
```

: Modify Album 선택 시 수정할 앨범의 제목과 아티스트를 입력받는다. 존재하지 않는 정보일 경우 에러메시지를 출력한다. 존재하는 정보일 경우 수정할 attribute의 선택지를 제공한다. 예시로 Title을 변경해보았다.

<Manager Mode> - Artist

```
[MANAGER MODE]
Select command to execute.
1. Show Artist
2. Insert Artist
3. Delete Artist
4. Modify Artist
0. Back to previous menu
Input : 1
```

<Manager Mode> - Artist - Show Artist

Name	Sex	Debut
IU	F	2008-09-18
Red Velvet	G	2014-08-01
Charlie Puth	M	2015-02-26
TWICE	G	2015-10-20
Adele	F	2008-01-20
Maroon 5	G	2003-09-06

<Manager Mode> - Artist - Insert Artist

```
[INSERT Artist]
Name : Oasis
Sex : G
Debut Date : 20130111
[Oasis] INSERT DONE
```

<Manager Mode> - Artist - Delete Artist

```
[DELETE Artist]
Name : Oasis
[Oasis] DELETED
```

<Manager Mode> - Artist - Modify Artist

```
[UPDATE Artist]
Select Artist to update
Name : TWICE

Select attribute to update
1. Name
2. Sex
3. Debut Date
0. Back to previous menu
Input : 2

Input new Sex to update
Sex : G

MODIFIED TO [G]
```

<Manager Mode> - User

```
[MANAGER MODE]
Select command to execute.
1. Show User
2. Delete User
0. Back to previous menu
Input : 1
```

<Manager Mode> - User - Show User

Name	Sex	Joined Date	ID	Password	
User1	M	2010-01-01	user1	1234	
User2	M	2010-01-01	user2	1234	
User3	F	2010-01-01	user3	1234	
User4	F	2010-01-01	user4	1234	

<Manager Mode> - User - Delete User

```
[DELETE User]
ID : User4
USER [User4] DELETED
```

<Manager Mode> - Manager

```
[MANAGER MODE]
Select command to execute.
1. Show Manager
2. Add Manager
3. Modify Manager Info
4. Resign Manager
0. Back to previous menu
Input : 1
```

<Manager Mode> - Manager - Show Manager

Name	Sex	Manager Start Date	ID	Password	
Manager1	M	2007-01-01	admin	1234	
Manager2	F	2007-01-01	admin2	1234	
Manager3	M	2007-01-01	admin3	1234	

<Manager Mode> - Manager - Add Manager

```
[ADD Manager]
ID : Manager3
Password : s
Name : s
Sex : s
[s] INSERT DONE
```

<Manager Mode> - Manager - Modify Manager Info

```
[UPDATE Manager Information]
Select attribute to update
1. Name
2. Sex
3. Password
0. Back to previous menu
Input : 2

Input new Sex to update
Sex : F

MODIFIED TO [F]
```

<Manager Mode> - Manager - Resign Manager

```
[RESIGN Manager]
WARNING : This command deletes your information as manager.
Input "RESIGN" if you're sure with this.
If you're not, please input anything else.
Input : RESIGN
MANAGER RESIGNED
```

<User Mode>

```
<Mode select>
1. Manager
2. User
3. Register
0. Exit
Input : 2

<Log in>
ID : user1
PW : 1234
User1 Permission Granted.

[USER MODE]
1. Search Music by Title
2. Search Music by Album
3. Search Music by Artist
4. Show Playlists
5. Show Favorite Artists
6. Modify User Information
7. Show TOP 10
8. Show Newly Released
9. Delete User Account
0. Log out
Input : 1
```

<User Mode> - Search Music by Title

```
[USER MODE]
Input music title to search
Title : Rookie

-----| Title | Genre | Album | Artist |-----
| Rookie | Dance | Rookie | Red Velvet |-----
```

<User Mode> - Search Music by Album

[USER MODE]				
Input album title to search				
Album : 25				
Title	Genre	Album	Artist	
Hello	Ballad	25	Adele	
When We Were Young	Ballad	25	Adele	

<User Mode> - Search Music by Artist

[USER MODE]				
Input artist name to search				
Artist : Maroon 5				
Title	Genre	Album	Artist	
This Love	Rock	Songs About Jane	Maroon 5	
She Will Be Loved	Rock	Songs About Jane	Maroon 5	
Sunday Morning	Rock	Songs About Jane	Maroon 5	
Makes Me Wonder	Rock	It Won't Be Soon Before Long!	Maroon 5	
Wake Up Call	Rock	It Won't Be Soon Before Long!	Maroon 5	
Misery	Rock	Hands All Over	Maroon 5	
Give A Little More	Rock	Hands All Over	Maroon 5	
Stutter	Rock	Hands All Over	Maroon 5	
Moves Like Jagger	Rock	Hands All Over	Maroon 5	
One More Night	Rock	Overexposed	Maroon 5	
Payphone	Rock	Overexposed	Maroon 5	
Lucky Strike	Rock	Overexposed	Maroon 5	

<User Mode> - Show Playlists

<PLAYLISTS>	
Playlist	
Playlist_1_A	
Playlist_1_B	

<User Mode> - Show Playlists - Select Playlist

[USER MODE] - Playlist				
1. Select Playlist				
2. Create Playlist				
3. Delete Playlist				
0. Back to previous menu				
Input : 1				
[USER MODE] - Playlist				
Input playlist name to use				
Name : Playlist_1_A				
Title	Album	Artist	Playlist	
Voice Mail	Modern Times	IU	Playlist_1_A	
Dumb Dumb	The Red	Red Velvet	Playlist_1_A	
TT	TWICEcoaster : LANE 1	TWICE	Playlist_1_A	
LIKEY	twicetagram	TWICE	Playlist_1_A	
Rolling In The Deep	21	Adele	Playlist_1_A	
Set Fire To The Rain	21	Adele	Playlist_1_A	
Hello	25	Adele	Playlist_1_A	
[USER MODE] - Playlist				
1. Add Music to Playlist				
2. Delete Music from Playlist				
0. Back to previous menu				
Input :				

<User Mode> - Show Playlists - Create Playlist

```
[USER MODE] - Playlist
Input playlist name to create
Name : ex1

[ex1] CREATED
<PLAYLISTS>
-----
| Playlist          |
-----
| Playlist_1_A     |
| Playlist_1_B     |
| ex1              |
-----
```

<User Mode> - Show Playlists - Delete Playlist

```
[USER MODE] - Playlist
Input playlist name to delete
Name : ex1

[ex1] DELETED
<PLAYLISTS>
-----
| Playlist          |
-----
| Playlist_1_A     |
| Playlist_1_B     |
-----
```

<User Mode> - Show Favorite Artists

```
<FAVORITE ARTISTS>
-----
| Artist           |
-----
| Adele           |
| Charlie Puth   |
| IU              |
| Maroon 5        |
| Red Velvet      |
| TWICE           |
-----
```



```
[USER MODE] - Favorite Artist
1. Add Artist
2. Delete Artist
0. Back to previous menu
Input : |
```

<User Mode> - Show Favorite Artists - Add Artist

```
[USER MODE] - Favorite Artist
Input artist name to add
Artist : IU

[IU] ADDED INTO favorite_artist
<FAVORITE ARTISTS>
-----
| Artist           |
-----
| Adele           |
| Charlie Puth   |
| IU              |
| Maroon 5        |
| Red Velvet      |
| TWICE           |
-----
```

<User Mode> - Show Favorite Artists - Delete Artist

```
[USER MODE] - Favorite Artist
Input artist name to delete
Artist : IU

[IU] DELETED

<FAVORITE ARTISTS>
-----
| Artist          |
-----
| Red Velvet    |
| Charlie Puth   |
| TWICE          |
| Adele          |
| Maroon 5       |
```

<User Mode> - Modify User Information

```
[UPDATE Manager Information]

Select attribute to update
1. Name
2. Sex
3. Password
0. Back to previous menu
Input : 1
```

<User Mode> - Show TOP 10

```
[Ranking]
Select TOP10 to see
1. Music
2. Artist
0. Back to previous menu
Input : 1

<MUSIC TOP10>
-----
| Music           | Album        | Artist      | Popularity |
-----
| Moves Like Jagger | Hands All Over | Maroon 5    | 4
| Rookie          | Rookie        | Red Velvet  | 3
| Dumb Dumb       | The Red       | Red Velvet  | 2
| TT              | TWICEcoaster : LANE 1 | TWICE       | 1
| Rolling In The Deep | 21           | Adele       | 1
| LIKEY           | twicetagram  | TWICE       | 1
| Set Fire To The Rain | 21           | Adele       | 1
| River            | Nine Track Mind | Charlie Puth | 1
| Hello            | 25           | Adele       | 1
| Voice Mail       | Modern Times  | IU          | 1
-----

[Ranking]
Select TOP10 to see
1. Music
2. Artist
0. Back to previous menu
Input : 2

<ARTIST TOP10>
-----
| Name            | Sex        | Debut      | Popularity |
-----
| Charlie Puth   | M          | 2015-02-26 | 3
| Maroon 5       | G          | 2003-09-06 | 3
| Red Velvet     | G          | 2014-08-01 | 3
| Adele          | F          | 2008-01-20 | 3
| TWICE          | G          | 2015-10-20 | 2
| IU              | F          | 2008-09-18 | 1
```

<User Mode> - Show Newly Released

<NEWLY RELEASED>				
Music	Album	Artist	Released	
LIKEY	twicetagram	TWICE	2017-10-30	
MISSING U	twicetagram	TWICE	2017-10-30	
Rookie	Rookie	Red Velvet	2017-02-01	
One Call Away	Nine Track Mind	Charlie Puth	2016-11-11	
We Don't Talk Anymore	Nine Track Mind	Charlie Puth	2016-11-11	
River	Nine Track Mind	Charlie Puth	2016-11-11	
TT	TWICEcoaster : LANE 1	TWICE	2016-10-24	
JELLY JELLY	TWICEcoaster : LANE 1	TWICE	2016-10-24	
CHEER UP	PAGE TWO	TWICE	2016-04-25	
Touchdown	PAGE TWO	TWICE	2016-04-25	

<User Mode> - Delete User Account

```
[DELETE Account]
WARNING : This command deletes your user information as a member.
Input "DEACTIVATE" if you're sure with this.
If you're not, please input anything else.
Input : DEACTIVATE
USER ACCOUNT DELETED
```

<Register Mode>

```
<User Register>
ID : aa
PW : aa
Input your information.
Name : aa
Sex : a
aa register succeeded.
```