

데이터베이스 시스템 project (음원 데이터베이스) - 2019061721 최성훈

1. Environment

- OS : Windows 10
- Language : JAVA (jdk 1.8.0_261)

2. Configuration Files

- Album.java
 - : main문이 포함되어 있는 클래스
- mariadb-java-client-2.7.1.jar
 - : classpath를 이용해 MariaDB jdbc driver를 사용할 수 있게 도와준다.
- Albulmdumpfinal.sql
 - : DB를 백업한 sql 덤프 파일

3. Modifications

- ① 플레이리스트 테이블을 삭제하였다.
: 애초에 playlist는 manager가 관리할 필요가 없고, user가 자기 자신의 playlist를 관리하면 충분하다고 판단했다.
- ② 음원 테이블 내에 있었던 삭제 여부 attribute를 삭제하였다.
: 어떤 생각으로 삭제 여부 attribute가 있었는지 잘 모르겠지만, 이번 project 4를 진행하면서 delete문을 통해 지우면 바로 해당 tuple이 삭제 되기 때문에 삭제 여부 attribute의 존재 이유가 없다고 판단했다.
- ③ 테이블명과 attribute 명을 영어로 바꾸어 주었다.
: 이는 SQL 구문에서 더욱 편하고 효율적으로 작성하기 위함이었다.

4. Code 설명 및 실행 결과

1. Manager Menu

- Register : Manager로 회원가입을 할 수 있다. 이 때, 회사 내에서 준 보안 번호 (1112)를 입력해야만 회원가입이 가능하다.
- Login : Manager로 로그인 가능하다. 이 때도 마찬가지로 회사에서 내어준 보안 번호를 입력하고 맞춰야만 로그인이 가능하다.

```

0. Exit
1. Manager Menu
2. User Menu
3. Singer Menu
1
-----
0. Return to previous menu
1. Login
2. Register
2
-----
Name : Hello
ID : finalboss
PW : finalboss
Phone # : 010-4512-5623
Your birthdate : 1976/06/23
Your home address : Busan
Secrity # from Company : 1112
-----
0. Return to previous menu
1. Login
2. Register
1
-----
ID : finalboss
PW : finalboss
Secrity # from Company : 1112
-----
0. Return to previous menu
1. Manages User
2. Manages Music
3. See statistics

```

- Manages User : 모든 유저를 확인하거나, 필요시에 유저를 삭제 할 수 있다.

```

0. Return to previous menu
1. Manages User
2. Manages Music
3. See statistics
1
-----
0. Return to previous menu
1. List All Users
2. Delete User
1
-----
      USER_ID,      USER_NAME,      USER BD,      USER_ADDRESS,      USER_PHONENUMBER,      REGISTER DATE,      USER_MANAGER
      sunghunchoi,      Sunghun, 2000-07-27,      Seoul,      010-5242-8770,      2020-12-07,      admin1
      user1,      user1, 2000-06-21,      Busan,      010-8956-5623,      2020-12-07,      admin1
      user2,      User2, 1978-05-21,      Jeju,      010-5623-8956,      2020-12-07,      admin1
-----
0. Return to previous menu
1. List All Users
2. Delete User
2
-----
Please type user id that you want to deleteuser2
-----
0. Return to previous menu
1. List All Users
2. Delete User
1
-----
      USER_ID,      USER_NAME,      USER BD,      USER_ADDRESS,      USER_PHONENUMBER,      REGISTER DATE,      USER_MANAGER
      sunghunchoi,      Sunghun, 2000-07-27,      Seoul,      010-5242-8770,      2020-12-07,      admin1
      user1,      user1, 2000-06-21,      Busan,      010-8956-5623,      2020-12-07,      admin1

```

- Manages Music : 모든 음악을 확인할 수 있고, 필요시에 음악을 추가하거나 삭제 할 수 있다.

```

1
MUSIC_NUM,      MUSIC_TITLE,      MUSIC_GENRE,      MUSIC_COMP,      MUSIC_LYRICIST,      MUSIC_COMPANY,      MUSIC_SINGER
3,              Night_Letter,      Ballade,          IU,              IU,              EDAM,              IU
4,              1,              1,              1,              1,              EDAM,              BTS
7,              Good_Day,      POP,              IU,              IU,              EDAM,              IU
8,              eight,      POP,              IU,              IU,              EDAM,              IU
9,              Love_Poem,      POP,              IU,              IU,              EDAM,              IU
10,             Palette,      POP,              IU,              IU,              EDAM,              IU

0. Return to previous menu
1. List All Musics
2. Add Music
3. Delete Music
4
Please type music title you want to add : Dynamite
Please type music genre you want to add : Dance
Please type music composer you want to add : David.Stewart
Please type music lyricist you want to add : David.Stewart
Please type company you want to add : BigHit
Please type music singer ID you want to add : BTS

0. Return to previous menu
1. List All Musics
2. Add Music
3. Delete Music
4
MUSIC_NUM,      MUSIC_TITLE,      MUSIC_GENRE,      MUSIC_COMP,      MUSIC_LYRICIST,      MUSIC_COMPANY,      MUSIC_SINGER
3,              Night_Letter,      Ballade,          IU,              IU,              EDAM,              IU
4,              1,              1,              1,              1,              EDAM,              BTS
7,              Good_Day,      POP,              IU,              IU,              EDAM,              IU
8,              eight,      POP,              IU,              IU,              EDAM,              IU
9,              Love_Poem,      POP,              IU,              IU,              EDAM,              IU
10,             Palette,      POP,              IU,              IU,              EDAM,              IU
11,             Dynamite,      Dance,          David_Stewart,      David_Stewart,      BigHit,              BTS

0. Return to previous menu
1. List All Musics
2. Add Music
3. Delete Music
4
Please type music title that you want to delete : Dynamite

0. Return to previous menu
1. List All Musics
2. Add Music
3. Delete Music
4
MUSIC_NUM,      MUSIC_TITLE,      MUSIC_GENRE,      MUSIC_COMP,      MUSIC_LYRICIST,      MUSIC_COMPANY,      MUSIC_SINGER
3,              Night_Letter,      Ballade,          IU,              IU,              EDAM,              IU
4,              1,              1,              1,              1,              EDAM,              BTS
7,              Good_Day,      POP,              IU,              IU,              EDAM,              IU
8,              eight,      POP,              IU,              IU,              EDAM,              IU
9,              Love_Poem,      POP,              IU,              IU,              EDAM,              IU
10,             Palette,      POP,              IU,              IU,              EDAM,              IU

```

- See statistics : 통계 자료를 확인 할 수 있다. (어떤 유저가 어떤 가수의 팬인지, 어떤 유저가 어떤 노래를 몇 번 들었는지)

```

0. Return to previous menu
1. See fans
2. Check number of streaming
2
          USER,      MUSIC,  NUM OF STREAMING
          user1,      3,          0
          user1,      4,          0

-----
0. Return to previous menu
1. See fans
2. Check number of streaming
1
SINGER,      USER

```

- 팬의 기준 : 그 가수가 만든 어떤 노래던 상관없이 총 만 번이상 스트리밍을 통해 들었을 경우
- 데이터 상의 총 만 번 이상 스트리밍한 유저가 없기에 실행창에는 그 누구도 팬이 아니다.

2. User Menu

- Register : User로 회원가입을 할 수 있다. * 실행사진은 manger와 비슷하여 생략
- Login : User로 로그인이 가능하다. * 실행사진은 manager와 비슷하여 생략
- List Musics : 음악을 검색하거나 전체 음악을 확인 할 수 있다. 또한 음악을 본인 playlist에 추가 할 수 있다.
- Add Music to playlist : 음악을 검색하거나 또는 음원번호를 안다면, 바로 음원번호를 통해 playlist에 값을 넣을 수 있다.

```

0. Return to previous menu
1. List Musics (You can search musics or add musics to your playlist in this section)
2. List Playlist I made (You can play music at this section)
1
MUSIC_NUM.      MUSIC_TITLE.      MUSIC_GENRE.      MUSIC_COMP.      MUSIC_LYRICIST.      MUSIC_COMPANY.      MUSIC_SINGER
3.              Night Letter.      Ballade.          IU.              IU.                  EDAM.              IU
4.              1.              1.              1.              1.                  1.                  BTS
7.              Good Day.          POP.              IU.              IU.                  EDAM.              IU
8.              eight.            POP.              IU.              IU.                  EDAM.              IU
9.              Love Poem.        POP.              IU.              IU.                  EDAM.              IU
10.             Palette.         POP.              IU.              IU.                  EDAM.              IU
0. Return to previous menu
1. Add music to playlist
1
Do you want to search music by title? (Y/N) : Y
You can type whole title or part of title : Good
MUSIC_NUM.      MUSIC_TITLE.      MUSIC_GENRE.      MUSIC_COMP.      MUSIC_LYRICIST.      MUSIC_COMPANY.      MUSIC_SINGER
7.              Good Day.          POP.              IU.              IU.                  EDAM.              IU
Type Music number you want to add : 7
Type the Playlist number that your wanted music to be in : 10
Please type your interest (in percentage without %) : 99

```

- List Playlist I made : 플레이리스트를 확인 할 수 있으며, 플레이리스트를 생성, 삭제가 가능하며 플레이리스트 내에 있는 노래를 재생하거나 노래를 삭제 할 수 있다.
- Play All musics in Playlist : 플레이리스트에 있는 모든 노래를 재생하며, interest표에서 알맞은 노래와 유저에 관련된 tuple에서 streaming이 1 더해진다.
- Create Playlist : 플레이리스트를 새롭게 생성한다.

```

-----
0. Return to previous menu
1. Play All musics in Playlist
2. Create Playlist
3. Delete Playlist
4. Delete musics in your playlist
2
Here is your playlist(s)
PLAYLIST_NUMBER
9
11
-----

```

- Delete Playlist : 플레이리스트를 삭제한다.

```

0. Return to previous menu
1. Play All musics in Playlist
2. Create Playlist
3. Delete Playlist
4. Delete musics in your playlist
3
Here is your playlist(s)
PLAYLIST_NUMBER
9
10
Please type Playlist number that you want to delete : 10

```

- Delete musics in your playlist : 플레이리스트 내에 있는 노래를 삭제한다.

```

0. Return to previous menu
1. Play All musics in Playlist
2. Create Playlist
3. Delete Playlist
4. Delete musics in your playlist
4
PLAYLIST_NUMBER,          MUSIC_TITLE
      8,                    1
      9,                    Love_Poem
Please type Playlistnumber that includes music that you want to delete. : 8
Here is list of your musics in number 8 playlist
MUSIC_NUMBER,          MUSIC_TITLE
      4                    1
Please type the music number you want to delete : 4

```

3. Singer Menu

- Register : Singer로 회원가입 할 수 있다. * 실행사진은 manager와 똑같이 생략
- Login : Singer로 로그인 가능하다. * 실행사진은 manager와 똑같이 생략
- Check my fans : 자신의 팬인 유저들을 확인 가능하다.

```

0. Return to previous menu
1. Check my fans
2. Manage my musics
1

```

- Manage my musics : 자신의 노래를 관리 할 수 있다. (삭제 또는 추가) *실행사진은 manager와 비슷하여 생략 (manager와 다른 부분은 가수 ID를 묻지 않고 manger의 ID를 묻습니다)

5. SQL 구문

SQL injection을 방지하기 위해서 JDBC 내에서도 PreparedStatement를 사용하였다. 따라서 ?로 처리되어 있는 부분들은 입력을 받는다는 뜻이다.

Delete from music where MUSIC_TITLE = ?

Delete from music where MUSIC_TITLE = ? and MUSIC_SINGER = ?

DELETE FROM MUSIC_IN_PLAYLIST WHERE PLAYLIST_NUM = ? AND IN_MUSIC = ?

Delete from playlist where PLAYLIST_NUM = ?

DELETE FROM PLAYLIST WHERE PLAYLIST_NUM = ? AND PLAYLIST_USER = ?

Delete from user where USER_ID = ?

INSERT IGNORE INTO FANS VALUES (?, ?)

INSERT IGNORE INTO INTEREST VALUES (?, ?, ?, ?)

INSERT IGNORE INTO MUSIC_IN_PLAYLIST VALUES (?,?)

INSERT INTO manager VALUES (?, ?, ?, ?, ?, ?, ?)

Insert into music (MUSIC_TITLE, MUSIC_GENRE, MUSIC_COMP, MUSIC_LYRICIST, MUSIC_COMPANY, MUSIC_MANAGER, MUSIC_SINGER) values (?, ?, ?, ?, ?, ?)

```

INSERT INTO PLAYLIST (PLAYLIST_USER) VALUES (?)
INSERT INTO singer VALUES ( ?, ?, ?, ?)
INSERT INTO user VALUES ( ?, ?, ?, ?, ?, ?, ?, ?)
SELECT * FROM fans
SELECT * FROM MANAGER WHERE MANAGER_ID = ? AND MANAGER_PW = ?
Select * from music where MUSIC_TITLE = ?
Select * from music where MUSIC_TITLE = ? and MUSIC_SINGER = ?
SELECT * FROM PLAYLIST
Select * from playlist where PLAYLIST_NUM = ?
SELECT * FROM SINGER WHERE SINGER_ID = ? AND SINGER_PW = ?
Select * from user where USER_ID = ?
SELECT * FROM USER WHERE USER_ID = ? AND USER_PW = ?
SELECT IN_MUSIC, MUSIC_TITLE FROM MUSIC_IN_PLAYLIST, MUSIC
WHERE PLAYLIST_NUM = ? AND MUSIC_NUM = IN_MUSIC

SELECT MUSIC_NUM, MUSIC_TITLE, MUSIC_GENRE, MUSIC_COMP,
MUSIC_LYRICIST, MUSIC_COMPANY, MUSIC_SINGER FROM MUSIC

SELECT MUSIC_NUM, MUSIC_TITLE, MUSIC_GENRE, MUSIC_COMP,
MUSIC_LYRICIST, MUSIC_COMPANY, MUSIC_SINGER FROM MUSIC
WHERE MUSIC_TITLE LIKE CONCAT('%',?,'%')

SELECT MUSIC_SINGER FROM MUSIC, INTEREST WHERE INTEREST.USER = ?
AND INTEREST.MUSIC = MUSIC.MUSIC_NUM
GROUP BY MUSIC_SINGER HAVING SUM(STREAMING) >= 10000

SELECT PLAYLIST_NUM FROM PLAYLIST WHERE PLAYLIST_USER = ?
SELECT PLAYLIST_NUM, MUSIC_TITLE
FROM MUSIC_IN_PLAYLIST, PLAYLIST, MUSIC
WHERE PLAYLIST.PLAYLIST_USER = ? AND MUSIC_NUM = IN_MUSIC

SELECT USER FROM FANS WHERE SINGER = ?
SELECT USER, MUSIC, STREAMING FROM INTEREST
SELECT USER_ID , USER_NAME, USER_BD, USER_ADDR, USER_PN,
REGISTER_DATE, USER_MANAGER FROM USER

UPDATE INTEREST, PLAYLIST, MUSIC_IN_PLAYLIST SET STREAMING = STREAMING + 1
WHERE USER = ? AND PLAYLIST_USER = ? AND MUSIC = IN_MUSIC

```

6. Compile 및 실행 과정

```

C:\Users\Sunghun\Desktop\Sunghun\Hanyang\Grade 2\2 semester\데이터베이스시스템및응용\2020_ite2038_2019061721\Album>javac
album.java

C:\Users\Sunghun\Desktop\Sunghun\Hanyang\Grade 2\2 semester\데이터베이스시스템및응용\2020_ite2038_2019061721\Album>mysql
-u root -p album < albu.mdmpfinal.sql
Enter password:

C:\Users\Sunghun\Desktop\Sunghun\Hanyang\Grade 2\2 semester\데이터베이스시스템및응용\2020_ite2038_2019061721\Album>java
-cp ".\;.\mariadb-java-client-2.7.1.jar" album

```

java 파일이 있는 위치로 가 javac album.java를 통해 컴파일 한다.

mysql -u root -p album < albu.mdmpfinal.sql를 통해 dump 받은 sql 파일을 넣어준다.

java -cp “..\mariadb-java-client-2.7.1.jar” album을 통해 mariadb를 사용할 것을 알려
줌과 동시에 코드를 실행시킨다.