2020 음원 데이터베이스 애플리케이션

Database Systems (ITE2038)

컴퓨터소프트웨어학부

2019054957 이용우

1. Environment

OS: Windows 10

Language: Java (jdk 1.8.0._261)

2. Configuration Files

Source/ Main.java

- 본 프로그램을 실행시키는 클래스이다.

Source/ Menu.java

- 메뉴창을 모아 놓은 클래스로 CLI의 동작을 제어한다.

Source/ Entity/ Admin.java Music.java Playlist.java User.java

DAO/ AdminDAO.java MusicDAO.java PlaylistDAO.java UserDAO.java

Action/ AdminAction.java MusicAction.java PlaylistAction.java UserAction.java

- [TableName].java: DB의 해당 table에서의 tuple을 하나의 클래스로 정의한다.
- [TableName]DAO.java: Data Access Object(DAO)로 실제로 DB에 접근하는 객체이며, SQL 쿼리를 모아 놓은 클래스이다.
- [TableName]Function.java: 해당 테이블의 기능을 구현한 함수들을 모아 놓은 클래스이다.

music_streaming_service.sql

- DB를 백업한 Sql 덤프 파일이다.

3. Modifications

- User 테이블에 Deleted라는 Attribute를 추가했다.
    - 어느 사이트에서 회원을 탈퇴해도 그 회원정보는 일시적으로 보관된다고 알고 있어서 삭제 여부에 관한 Attribute을 추가했다. 또한, 실수로 다른 회원을 탈퇴시키는 경우가 발생할 수 있으므로 해당 Attribute은 필요하다고 생각했다.
    - Deleted는 Boolean type으로 True면 삭제된 것으로 로그인할 수 없다. Deleted 정보는 Admin이 다시 Restore할 수 있다.
- 음원의 streaming 조건을 삭제했다.
    - 기존) 스트리밍 구독자에 한정해서 2분 이상 재생된 곡은 플레이 횟수를 1만큼 올린다. 단, 재생 시간이 2분이 안되는 곡은 1분 이상 재생시에 플레이 횟수를 1만큼 올린다.
    - 본 프로젝트의 목적과 Demo의 시간 제한 등을 고려했을 때 음원을 직접 스트리밍하는 기능은 꼭 필요하지 않는 것처럼 느껴졌다. 그래서 streaming시 음원이 재생되지 않으며, 구독자에 한해 플레이 횟수를 1만큼 올리는 것으로 수정했다.
- User가 포함할 수 있는 Playlist의 개수 제한과 Playlist에 넣을 수 있는 음원의 개수 제한을 삭제했다.
    - 많은 User가 사용할 경우에 DB의 과부화를 막기 위해 넣은 제한사항이지만, 본 프로젝트에는 꼭 필요한 기능은 아니라고 생각했다.

4. Description of Codes

(TableName)Function.java에 구현된 Create, Read, Update, Delete(CRUD)의 기능을 위주로 설명한다. 함수에서 input을 받는 등의 불필요한 부분은 제거하고 핵심 부분의 코드만 가져와 설명한다. DB에서 가져오는 정보는 DAO class에서 호출하는 함수의 SQL 쿼리문을 참고한다.

■ Create

- Admin, User – Register(회원가입)

```
public int register() {
    int result = adminDAO.register(ID, PW, name, SSN, address, phoneNumber, email);

    if(result >= 0) {
        System.out.println("Register Success\n");
```

```
        return 1;
    }
    return -1;
}
```

회원정보를 입력 받아(해당 부분은 길어져서 표기하지 않았다) DB table에 등록(INSERT)한다. 보안을 위해 PW는 SHA-256으로 암호화하여 저장한다. Admin과 User의 Register는 DB table만 다를 뿐 동일한 코드이다.

- Admin – addMusic

```
public int addMusic(Admin admin) {
    int result = adminDAO.musicInsert(title, album, lyrics, playTime, composer,
lyricist, arranger);
    int idx = adminDAO.getMusicLastIdx();
    int result2 = adminDAO.insertManagingMusic(admin, idx);

    if(contains) {
        for (int i = 0; i < singers.length; i++) {
            adminDAO.singerInsert(idx, singers[i]);
        }
    }
    else {
        adminDAO.singerInsert(idx, singer);
    }

    if(contains2) {
        for(int i = 0; i < genres.length; i++) {
            adminDAO.genreInsert(idx, genres[i]);
        }
    }
    else {
        adminDAO.genreInsert(idx, genre);
    }

    if(result > 0 && result2 > 0) {
        System.out.println("Insert Success\n");
        return 1;
    }
    return -1;
}
```

Admin은 음원을 관리하여 등록할 수 있다. 음원의 정보를 music table에 등록(INSERT)한다. 어떤 Admin이 어떤 음원을 등록했는지 확인하기 위해 관련 index들을 manages_music table에 등록(INSERT)한다. 음원의 가수를 입력할 때, 여러 가수가 참여했는지를 contains로 구분하여 singer table에 등록(INSERT)한다. 같은 방식으로 음원의 장르도 genre table에 등록(INSERT)한다.

- Playlist – createPlaylist

```
public void createPlaylist(User user) {
    int result = playlistDAO.createPlaylist(user, name);
    int result2 = playlistDAO.increaseListCount(user);
    if(result > 0 && result2 > 0) {
        System.out.println("Create Success\n");
    }
}
```

parameter로 전달받은 user의 새로운 playlist를 등록(INSERT)한다. Playlist에 들어있는 곡

의 수를 나타내는 ListCount attribute을 1만큼 증가시킨다.

- Playlist – insertMusicIntoPlaylist

```
public void insertMusicIntoPlaylist(User user, Playlist playlist, Music music) {
    playlistDAO.insertMusicIntoPlaylist(user, playlist, music);
}
```

SELECT한 music 객체의 정보를 contains table에 등록(INSERT)한다.

■ Read

- Admin, User – login

```
public Admin login() {
    int result = adminDAO.login(ID, PW);
    if(result > 0) {
        admin = adminDAO.getInfo(result);
        System.out.println("Login Success\n");
        return admin;
    }
    return null;
}
```

ID와 PW를 입력 받아(해당 부분은 길어져서 표기하지 않았다.) DB table의 Data를 가져와서(SELECT) 서로 일치하는지 검사한다. 이때 입력한 PW는 SHA-256으로 암호화하여 저장된 PW hash와 같은 지 비교한다. Admin과 User의 login은 DB table만 다를 뿐 동일한 코드이다.

- Admin – showUserList, showDeletedUserList

```
public void showUserList() {
    adminDAO.showUserList();
}

public void showDeletedUserList() {
    adminDAO.showDeletedUserList();
}
```

showUserList – User의 정보를 가져와서(SELECT) 리스트로 나열하여 출력한다.

showDeletedUserList – Deleted가 True인 User의 정보만 가져와서(SELECT) 출력한다.

- Music – searchMusicList, searchMusicByTitle, searchMusicBySinger, searchMusicByGenre, showStatistics

```
public void searchMusicList() {
    musicDAO.searchMusicList();
}
```

searchMusicList – 모든 음원 목록과 해당 음원의 가수 목록을 가져와서(SELECT) 출력한다.

searchMusicByTitle – Title을 input으로 받는다. SQL의 LIKE 키워드를 이용하여, input이 포함된 Title이 존재하는 음원 목록과 해당 음원의 가수 목록을 가져와서(SELECT) 출력한다.

searchMusicBySinger – Singer의 name을 input으로 받는다. input이 포함된 Singer가 존재하는 음원의 목록과 해당 음원의 가수 목록을 가져와서(SELECT) 출력한다.

searchMusicByGenre – Genre를 input으로 받는다. Input이 포함된 Genre가 존재하는 음원 목록과 해당 음원의 가수 목록을 가져와서(SELECT) 출력한다.

showStatistics – 음원 플레이 횟수를 기준으로 내림차순으로 정렬하여 출력한다.

5가지 함수 모두 입력의 여부와 DAO의 호출 함수만 다르고 구조는 같기 때문에 나머지 코드는 생략한다.

- Music – selectMusic

```
public Music selectMusic() {
    System.out.print("Title: ");
    String title = keyboard.next();
    return musicDAO.selectMusic(title);
}
```

Title이 일치하는 music의 모든 정보를 가져와(SELECT) 하나의 객체로 반환한다.

- Playlist – showPlaylists

```
public void showPlaylists(User user) {
    playlistDAO.showPlaylists(user);
}
```

parameter로 전달된 user의 playlist 목록을 가져와(SELECT) 출력한다.

- Playlist - selectPlaylist

```
public Playlist selectPlaylist(User user) {
    Playlist playlist = playlistDAO.selectPlaylist(user, name);
    if(playlist != null) {
        return playlist;
    }
    return null;
}
```

Playlist의 이름을 input으로 받아 일치하는 Playlist의 정보를 가져와서(SELECT) 하나의 객체로 반환한다.

- Playlist - showMusicInPlaylist

```
public void showMusicInPlaylist(User user, Playlist playlist) {
    playlistDAO.showMusicInPlaylist(user, playlist);
}
```

Playlist안에 들어있는 모든 음원의 목록을 가져와(SELECT) 출력한다.

- Playlist – selectMusicInPlaylist

```
public Music selectMusicInPlaylist(User user, Playlist playlist) {
    Music music = playlistDAO.selectMusicInPlaylist(user, playlist, title);
    if(music != null) {
        return music;
    }
    return null;
}
```

playlist에 들어있는 음원 중 하나를 선택(SELECT)하여 객체로 반환한다.

■ Update

- Admin – deleteUser, restoreUser

```
public void deleteUser(Admin admin) {
    adminDAO.deleteUser(index);
    adminDAO.insertManagingUser(admin, index);
}
public void restoreUser(Admin admin) {
    adminDAO.restore(index);
    adminDAO.deleteManagingUser(admin, index);
}
```

deleteUser - User의 Deleted Attribute를 True로 변경(UPDATE)한다. Admin과 삭제된 User의 index는 모두 manages_user table에 저장된다.

restoreUser – User의 Deleted Attribute를 False로 변경(UPDATE)한다. 해당하는 내용을 manages_user에서 삭제한다.

- User – musicStreaming

```
public void musicStreaming(User user) {
    if(!user.isSubscription()) {
        System.out.println("Subscribe First\n");
        return;
    }
    System.out.print("Title (cancel c): ");
    String title = keyboard.next();
```

```
    if(title.equalsIgnoreCase("c")) {
        return;
    }
    int result = userDAO.musicStreaming(title);
    if(result > 0) {
        System.out.println("Streaming Success\n");
    }
}
```

구독을 한 user에 한정해서 음원을 스트리밍한다. 해당 음원은 Play된 횟수인 PlayCount attribute을 1만큼 증가시킨다.

- User - subscribe

```
public void subscribe(User user) {
    if(user.isSubscription()) {
        System.out.println("You're already subscribed\n");
        return;
    }

    System.out.print("AutoSubscribe (Y/N): ");
    String answer = keyboard.next();
    boolean flag;

    if(answer.equalsIgnoreCase("Y")) {
        flag = true;
    }
    else if(answer.equalsIgnoreCase("N")) {
        flag = false;
    }
    else {
        System.out.println("Wrong input\n");
        return;
    }
    int result = userDAO.subscribe(user.getIndex(), flag);
    if(result > 0) {
        System.out.println("Subscribe Success\n");
    }
```

user의 Subscribe, ExpirationDate, AutoSubscribe의 attribute을 UPDATE한다. 구독 시 ExpirationDate는 한 달 뒤로 설정되고, AutoSubscribe가 True면 ExpirationDate가 지날 때 자동으로 재구독한다.

- User – unsubscribe

```
public void unsubscribe(User user) {
    int result = userDAO.unsubscribe(user.getIndex());
    if(result > 0) {
        user.setSubscription(false);
        user.setExpirationDate(null);
        user.setAutoSubscription(false);
        System.out.println("Unsubscribed\n");
    }
}
```

Parameter로 전달받은 user의 구독을 해지(UPDATE)한다.

■ Delete

- Admin – deleteMusic

```
public int deleteMusic(Admin admin) {
    int deleteIdx = adminDAO.findIndexByTitle(title);
    int result = adminDAO.deleteMusic(deleteIdx);
    if(result > 0) {
        System.out.println("Delete Success\n");
        return 1;
    }
    return -1;
}
```

Admin은 음원을 관리하여 삭제할 수 있다. 음원의 Title로 검색하여 index를 가져오고, 이를 이용하여 해당 음원을 music table에서 삭제(DELETE)한다.

- Playlist – deletePlaylist

```
public void deletePlaylist(User user) {
    int result = playlistDAO.deletePlayList(user, name);
    int result2 = playlistDAO.decreaseListCount(user);
    if(result > 0 && result2 > 0) {
        System.out.println("Delete Success\n");
    }
}
```

Playlist 이름을 input으로 받아 parameter로 전달받은 user의 playlist와 일치하는 것을 삭제(DELETE)한다. Playlist에 들어있는 곡의 수를 나타내는 ListCount attribute을 1만큼 감소시킨다.

- Playlist – deleteMusicFromPlaylist

```
public void deleteMusicFromPlaylist(User user, Playlist playlist, Music music) {
    playlistDAO.deleteMusicFromPlaylist(user, playlist, music);
}
```

SELECT한 음원을 Playlist에서 삭제(DELETE)한다.

5. Screenshots

- Register & Login

```
0. Return to previous menu
1. Sign In
2. Sign Up
2

ID: root
PW: root
Name: root
SSN: 1111
Address: A_Street
PhoneNumber: 1111
Email: root@root.com
Register Success

0. Return to previous menu
1. Sign In
2. Sign Up
1

ID: root
PW: root
Login Success

0. Return to previous menu
1. Manage Users
2. Manage Music
```

- Admin Page / Manage Users

```
0. Return to previous menu
1. Show all Users
2. Show Deleted Users
3. Restore Deleted User
4. Delete User
1

-------------------------------------------------------------------------------------------------------------------------------------------
    Index        ID       Name         PhoneNumber              Email         Subscription      ExpirationDate    AutoSubscription            Deleted
-------------------------------------------------------------------------------------------------------------------------------------------
        1      user       user            11111111   userEmail@user.com             false                null               false              false
        2   qwer1234      James            22222222  user2Email@user.com             false                null               false              false
        3   q1w2e3r4       John            33333333  user3Email@user.com             false                null               false              false
-------------------------------------------------------------------------------------------------------------------------------------------

0. Return to previous menu
1. Show all Users
2. Show Deleted Users
3. Restore Deleted User
4. Delete User
4

User Index: 1
0. Return to previous menu
1. Show all Users
2. Show Deleted Users
3. Restore Deleted User
4. Delete User
2

-------------------------------------------------------------------------------------------------------------------------------------------
    Index        ID       Name         PhoneNumber              Email         Subscription      ExpirationDate    AutoSubscription            Deleted
-------------------------------------------------------------------------------------------------------------------------------------------
        1      user       user            11111111   userEmail@user.com             false                null               false               true
-------------------------------------------------------------------------------------------------------------------------------------------
```

- Admin Page / Manage Music

```
Title: Song
Singer(if many artists participated, separate with spacebar): aa
Genre(if music has many genres, seperate with spacebar): ballad
Album: SongAlbum
Lyrics: Skip
Playtime (s): 120
Composer: aa
Lyricist: aa
Arranger: aa
Insert Success

0. Return to previous menu
1. Show all Music
2. Add Music
3. Delete Music
1

--------------------------------------------------------------------------------
     Title             Artists
--------------------------------------------------------------------------------
  Dynamite                  BTS
Can't_sleep        Jang-Bum-Joon
   Dolphin           Oh-my-girl
       VVS         Khundi_Panda Miranni Munchman Mushvenom
      Song                   aa
--------------------------------------------------------------------------------
```

- User Page

```
0. Return to previous menu
1. Music menu
2. PlayList Menu
3. User info
4. Subscribe
4

AutoSubscribe (Y/N): Y
Subscribe Success

0. Return to previous menu
1. Music menu
2. PlayList Menu
3. User info
4. Subscribe
3

User{index=1, ID='user', PW='04f8996da763b7a969b1028ee3007569eaf3a635486ddab211d512c85b9df8fb', name='user', SSN='111111
11', address='userAddress', phoneNumber='11111111', email='userEmail@user.com', subscription=true, expirationDate='2021-
01-06', autoSubscription=true, listCount=2}
```

- User Page / Music menu

```
7. Music Details
2

Title: D
-------------------------------------------------------------------------------
         Title             Artists
-------------------------------------------------------------------------------
       Dynamite                 BTS
        Dolphin          Oh-my-girl
-------------------------------------------------------------------------------

0. Return to previous menu
1. Show All Music
2. Search Music by title
3. Search Music by singer
4. Search Music by genre
5. Music Chart
6. Music Streaming
7. Music Details
5

-------------------------------------------------------------------------------
         Title          Play Count             Artists
-------------------------------------------------------------------------------
        Dolphin                   2          Oh-my-girl
       Dynamite                   1                 BTS
    Can't_sleep                   0         Jang-Bum-Joon
            VVS                   0          Khundi_Panda Miranni Munchman Mushvenom
-------------------------------------------------------------------------------
```

- User Page / Playlist menu

```
-------------------------------------------------------------------------------
  Playlist          MusicCount          TotalLength
-------------------------------------------------------------------------------
 Playlist1                   2                  421
 Playlist2                   0                    0
 Playlist3                   0                    0
-------------------------------------------------------------------------------

0. Return to previous menu
1. Show all Playlists
2. Make a new Playlist
3. Select a Playlist
4. Delete a Playlist
4

Playlist Name (Cancel c): Playlist3
Delete Success

0. Return to previous menu
1. Show all Playlists
2. Make a new Playlist
3. Select a Playlist
4. Delete a Playlist
1

-------------------------------------------------------------------------------
  Playlist          MusicCount          TotalLength
-------------------------------------------------------------------------------
 Playlist1                   2                  421
 Playlist2                   0                    0
-------------------------------------------------------------------------------
```

- User Page / Playlist Page

```
0. Return to previous menu
1. Show all Music in Playlist
2. Add Music
3. Delete Music
2

--------------------------------------------------------------------------------
     Title             Artists
--------------------------------------------------------------------------------
  Dynamite                  BTS
Can't_sleep        Jang-Bum-Joon
   Dolphin           Oh-my-girl
       VVS        Khundi_Panda Miranni Munchman Mushvenom
--------------------------------------------------------------------------------

Title: VVS

0. Return to previous menu
1. Show all Music in Playlist
2. Add Music
3. Delete Music
1

--------------------
     Title
--------------------
  Dynamite
   Dolphin
       VVS
--------------------
```