

프로젝트 4: DBMS를 활용한 프로그램 개발

제출일: 2020년 12월 7일

컴퓨터소프트웨어학부

2019000982 이지원

프로그램 코드에 대한 설명

1. **try** {
 Class.forName(driver);
 con = DriverManager.getConnection("jdbc:mariadb://localhost:3306/music", "root", "gusah");
 if(con != null) {
 System.out.println("DB 접속 성공");
 Statement stmt = con.createStatement();
con = DriverManager.getConnection("jdbc:mariadb://localhost:3306/music",
"root", "password")를 통해 mariadb와 접속한다. 접속에 성공할 시 이후의 것들을
진행한다.

2. **while**(true) {
 System.out.println("----- Wellcome to My Music APP -----");
 System.out.println("1. Manager Menu\n2. Subscriber Menu\n0. Exit");
 System.out.print("> ");
 firstMenu = sc.nextInt();

접속에 성공하면 main menu를 보여준다. 이 menu에서 사용자 입력에 따라 다음을
진행한다.

3. Manager menu를 선택한 경우

```
if(rs.next()) {  
    System.out.println("\n***** Wellcome " + rs.getString("Name") + " *****");  
    String mgrSsn = rs.getString("Ssn");  
    while(true) {  
        System.out.println("----- Manager Menu -----");  
        System.out.println("1. Add Songs\n2. Delete Songs\n3. Add Artist\n4. Delete Subscribers\n5. Edit Mana");  
        System.out.print(">> ");  
        secondMenu = sc.nextInt();  
        sc.nextLine();  
    }
```

(1) nick name의 입력을 통해 로그인을 진행한다. 쿼리를 이용해 manager로서 유효한
nick name인지 확인하고 만약 그렇다면 환영한다는 message와 함께 manager menu를
보여준다.

(2) 1번 Add song을 선택하면

```
if(secondMenu == 1) {
    int songNum;
    String title, lyrics, singer, writer, composer;
    int time;
    rs = stmt.executeQuery("SELECT MAX(Number) FROM SONG");
    rs.next();
    int current = rs.getInt("MAX(Number)");
    System.out.print("How many songs will you add? ");
    songNum = sc.nextInt();
    for(int i = 1; i <= songNum; i++) {
        System.out.print("Enter " + i + "th song name: ");
        title = sc.nextLine();
        sc.nextLine();
        System.out.print("Enter " + i + "th song's lyrics: ");
        lyrics = sc.nextLine();
        sc.nextLine();
        System.out.print("Enter " + i + "th song's play time: ");
        time = sc.nextInt();
        sc.nextLine();
        System.out.print("Enter " + i + "th song's singer: ");
        singer = sc.nextLine();
        sc.nextLine();
        System.out.print("Enter " + i + "th song's writer: ");
        writer = sc.nextLine();
        sc.nextLine();
        System.out.print("Enter " + i + "th song's composer: ");
        composer = sc.nextLine();
        sc.nextLine();

        int error = 0;
        pstmt = con.prepareStatement("select Number from ARTIST where Name = ?");
        pstmt.setString(1, singer);
        rs = pstmt.executeQuery();
        int singerNum = 0;
        if(rs.next()) {
            singerNum = rs.getInt("Number");
        } else {
            System.out.println("No such artist!!!");
            error = 1;
        }

        pstmt = con.prepareStatement("select Number from ARTIST where Name = ?");
        pstmt.setString(1, writer);
        rs = pstmt.executeQuery();
        int writerNum = 0;
        if(rs.next()) {
            writerNum = rs.getInt("Number");
        } else {
            System.out.println("No such artist!!!");
            error = 1;
        }

        pstmt = con.prepareStatement("select Number from ARTIST where Name = ?");
        pstmt.setString(1, composer);
        rs = pstmt.executeQuery();
        int composerNum = 0;
        if(rs.next()) {
            composerNum = rs.getInt("Number");
        } else {
            System.out.println("No such artist!!!");
            error = 1;
        }

        pstmt = con.prepareStatement("insert into SING values (?, ?, ?, ?, 0, ?)");

        if(error == 1) {
            System.out.println("Add song fail!!!");
        } else {
            pstmt = con.prepareStatement("insert into SONG values (?, ?, ?, ?, 0, ?)");
            pstmt.setString(1, title);
            pstmt.setInt(2, current + i);
            pstmt.setString(3, lyrics);
            pstmt.setInt(4, time);
            pstmt.setString(5, mgrSsn);
            rs = pstmt.executeQuery();

            pstmt = con.prepareStatement("insert into SING values(?, ?)");
            pstmt.setInt(1, current + i);
            pstmt.setInt(2, singerNum);
            rs = pstmt.executeQuery();

            pstmt = con.prepareStatement("insert into WRITE_SONG values(?, ?)");
            pstmt.setInt(1, current + i);
            pstmt.setInt(2, writerNum);
            rs = pstmt.executeQuery();

            pstmt = con.prepareStatement("insert into COMPOSE values(?, ?)");
            pstmt.setInt(1, current + i);
            pstmt.setInt(2, composerNum);
            rs = pstmt.executeQuery();

            System.out.println("Add song success!");
        }
    }
}
```

-> 추가할 곡의 개수에 따라 제목, 가사, 플레이시간(초단위로), 가수, 작사가, 작곡가를 입력 받은 뒤 이 항목들과 쿼리를 이용해서 새로운 노래와 'COMPOSE', 'WRITE_SONG',

‘SING’에 tuple을 추가한다.

(3) 2번 Delete songs를 선택하면

```
} else if(secondMenu == 2) {
    System.out.print("How many songs will you delete? ");
    int songNum = sc.nextInt();
    sc.nextLine();
    for(int i = 1; i <= songNum; i++) {
        System.out.print("Enter " + i + "th song number: ");
        int number = sc.nextInt();
        sc.nextLine();

        pstmt = con.prepareStatement("select Number from SONG as s, MANAGER as m where s.MgrSsn = ?");
        pstmt.setString(1, mgrSsn);
        rs = pstmt.executeQuery();

        int error = 1;

        while(rs.next()) {
            if(rs.getInt("Number") == number) {
                error = 0;
                break;
            }
            error = 1;
        }

        if(error == 0) {
            pstmt = con.prepareStatement("delete from SONG where Number = ?");
            pstmt.setInt(1, number);
            rs = pstmt.executeQuery();
            System.out.println("Delete success!");
        } else {
            System.out.println("No such song or no permission");
        }
    }
}
```

-> 삭제할 곡의 개수에 따라 곡의 번호를 입력받고 쿼리를 통해 얻은 현재 로그인 중인 관리자가 관리하고 있는 곡과 입력받은 곡의 번호를 비교한다. 만약 일치하는 항목이 있다면 제거하고 없다면 권한이 없다는 에러 메시지를 띄운다.

(4) 3번 add artist를 선택하면

```
System.out.println(songNum + " songs is/ delete success: ");
} else if(secondMenu == 3) {
    System.out.print("Enter artist name: ");
    String aName = sc.next();
    sc.nextLine();
    System.out.print("Enter artist debut(ex. 1900-03-28 \\if you don't know, enter 0): ");
    String debut = sc.next();
    sc.nextLine();

    //TODO: check if the artist exists

    pstmt = con.prepareStatement("select max(Number) from ARTIST");
    rs = pstmt.executeQuery();
    rs.next();
    int current = rs.getInt("max(Number)");

    if(debut.equals("0")) {
        pstmt = con.prepareStatement("insert into ARTIST(Name, Number, MgrSsn) values(?, ?, ?)");
        pstmt.setString(1, aName);
        pstmt.setInt(2, current + 1);
        pstmt.setString(3, mgrSsn);
        rs = pstmt.executeQuery();
    } else {
        Date d = Date.valueOf(debut);
        pstmt = con.prepareStatement("insert into ARTIST values(?, ?, ?, ?)");
        pstmt.setString(1, aName);
        pstmt.setInt(2, current + 1);
        pstmt.setDate(3, d);
        pstmt.setString(4, mgrSsn);
        rs = pstmt.executeQuery();
    }
    System.out.println("Add artist success!");
}
```

-> artist의 이름과 debut날짜를 입력받아 쿼리를 이용해 ARTIST에 tuple을 추가한다. 만약 debut 날짜를 모른다면 이를 제외하고 추가한다.

(5) 4번 Delete Subscriber를 선택하면

```
} else if(secondMenu == 4) {
    System.out.print("Enter nick name of subscriber: ");
    String deleteSub = sc.next();
    sc.nextLine();

    pstmt = con.prepareStatement("select s.NickName from SUBSCRIBER as s, MANAGER as m where s.MgrSsn
    pstmt.setString(1, mgrSsn);
    rs = pstmt.executeQuery();

    if(rs.next()) {
        pstmt = con.prepareStatement("delete from SUBSCRIBER where NickName = ?");
        pstmt.setString(1, deleteSub);
        rs = pstmt.executeQuery();
        System.out.println("Delete success!");
    } else {
        System.out.println("No such subscriber or no permission");
    }
}
```

-> 삭제하고 싶은 구독자의 nick name을 입력받고 쿼리를 이용해 현재 로그인 중인 manager의 관리 목록에 있는 구독자인지 확인한다. 만약 그렇다면 해당 subscriber를 삭제하고 아니라면 에러 메시지를 띄운다.

(6) 5번 Edit Manager Information을 선택하면

```
} else if(secondMenu == 5) {
    System.out.println("Which one will you edit?");
    System.out.println("1. Name\n2. NickName");
    System.out.print(">> ");
    int edit = sc.nextInt();
    sc.nextLine();
    if(edit == 1) {
        System.out.print("Enter new name: ");
        String newName = sc.next();
        sc.nextLine();
        pstmt = con.prepareStatement("update MANAGER set Name = ? where Ssn = ?");
        pstmt.setString(1, newName);
        pstmt.setString(2, mgrSsn);
        rs = pstmt.executeQuery();
    } else if(edit == 2) {
        System.out.print("Enter new nick name: ");
        String newNickName = sc.next();
        sc.nextLine();
        pstmt = con.prepareStatement("update MANAGER set NickName = ? where Ssn = ?");
        pstmt.setString(1, newNickName);
        pstmt.setString(2, mgrSsn);
        rs = pstmt.executeQuery();
    } else {
        System.out.println("Wrong Input!!!");
    }
}
```

-> 이름과 nick name 중 어떤 항목을 수정할지 입력받은 뒤 해당 항목의 새로 바꿀 내용을 입력받는다. 쿼리를 이용해 새로운 정보를 update한다.

(7) 6번 Show All Managing을 선택하면

```
    } else if(secondMenu == 6) {
        System.out.println("All Managing");
        pstmt = con.prepareStatement("select s.Name, s.NickName from music.MANAGER as m, music.SUBSCRIBER as s where m.Ssn = ?");
        pstmt.setString(1, mgrSsn);
        rs = pstmt.executeQuery();
        if(rs.next()) {
            System.out.println("Subscriber:");
            do {
                System.out.println("Name: " + rs.getString("Name") + ", Nick Name: [" + rs.getString("NickName") + "]");
            } while(rs.next());
        } else {
            System.out.println("No subscriber managing");
        }

        pstmt = con.prepareStatement("select song.Title, song.Number from music.MANAGER as m, music.SONG as song where m.Ssn = ?");
        pstmt.setString(1, mgrSsn);
        rs = pstmt.executeQuery();
        if(rs.next()) {
            System.out.println("Song:");
            do {
                System.out.println("Title: \"\" + rs.getString("Title") + "\", Number: " + rs.getString("Number"));
            } while(rs.next());
        } else {
            System.out.println("No song managing");
        }
    }
}
```

-> 현재 로그인중인 관리자가 관리하는 노래, 구독자, 아티스트의 정보를 출력한다. 없을 시에는 없다는 문구를 출력한다.

(8) 0을 입력하면 break를 통해 다시 이전 메뉴로 돌아가고 이외의 입력이 들어온다면 잘못된 input이라는 메시지를 출력한 뒤 다시 메뉴를 띄운다.

4. Subscriber menu를 선택한 경우

```
    } else if(firstMenu == 2) {
        System.out.print("Enter your nick name: ");
        String nickName = sc.next();
        pstmt = con.prepareStatement("SELECT Name, Ssn FROM music.SUBSCRIBER WHERE NickName = ?");
        pstmt.setString(1, nickName);
        rs = pstmt.executeQuery();

        if(rs.next()) {
            System.out.println("\n***** Wellcome " + rs.getString("Name") + " *****");
            String subSsn = rs.getString("Ssn");

            while(true) {
                System.out.println("----- Subscriber Menu -----");
                System.out.println("1. Make New Playlist\n2. Delete Playlist\n3. Edit Playlist\n4. Play songs\n5. Edit Su");
                System.out.print(">> ");
                secondMenu = sc.nextInt();
                sc.nextLine();
            }
        }
    }
}
```

(1) nick name을 입력 받은 뒤 쿼리를 이용해 로그인을 진행한다. 만약 존재하지 않는 nick name이라면 유효하지 않다는 메시지를 띄운다. 존재한다면 환영한다는 문구와 함께 Subscriber Menu를 출력한다.

(2) 1번 Make New Play List를 선택하면

```
    if(secondMenu == 1) {
        rs = stmt.executeQuery("SELECT MAX(Number) FROM PLAYLIST");
        rs.next();
        int current = rs.getInt("MAX(Number)");

        System.out.print("Enter new playlist name: ");
        String pName = sc.next();
        sc.nextLine();

        pstmt = con.prepareStatement("select Number from PLAYLIST where Name = ?");
        pstmt.setString(1, pName);
        rs = pstmt.executeQuery();

        if(rs.next()) {
            System.out.println("Name Already exists.");
        } else {
            pstmt = con.prepareStatement("insert into PLAYLIST values (?, ?, ?)");
            pstmt.setString(1, pName);
            pstmt.setInt(2, current + 1);
            pstmt.setString(3, subSsn);
            rs = pstmt.executeQuery();
            System.out.println("Add playlist success!");
        }
    }
}
```

-> 새로 만들 play list의 이름을 입력 받은 뒤 쿼리를 이용해 존재하는 이름인지

확인한다. 만약 존재 한다면 에러메시지를 출력하고 아니라면 insert를 이용해 새로운 play list를 생성한다.

(3) 2번 Delete Play List를 선택하면

```
} else if(secondMenu == 2) {
    System.out.print("Enter playlist name: ");
    String pName = sc.next();
    sc.nextLine();

    pstmt = con.prepareStatement("select Number from PLAYLIST where Name = ?");
    pstmt.setString(1, pName);
    rs = pstmt.executeQuery();

    if(rs.next()) {
        pstmt = con.prepareStatement("delete from PLAYLIST where Name = ?");
        pstmt.setString(1, pName);
        rs = pstmt.executeQuery();
        System.out.println("Delete playlist success!");
    } else {
        System.out.println("No such playlist!!!");
    }
}
```

-> playlist의 이름을 입력 받은 뒤 이를 이용해 현재 로그인한 subscriber에게 존재하는 playlist인지 확인하고 존재한다면 해당 playlist를 삭제, 존재하지 않는다면 에러 메시지를 출력한다.

(4) 3번 Edit Play List를 선택하면

```
} else if(secondMenu == 3) {
    System.out.print("Enter play list name: ");
    String pName = sc.next();
    sc.nextLine();

    pstmt = con.prepareStatement("select Number from PLAYLIST where Name = ?");
    pstmt.setString(1, pName);
    rs = pstmt.executeQuery();

    if(rs.next()) {
        int pNum = rs.getInt("Number");
        System.out.println("1. Edit playlist name\n2. Add songs\n3. Delete songs");
        int pMenu = sc.nextInt();
        sc.nextLine();

        if(pMenu == 1) {
            System.out.print("Enter new name: ");
            String newName = sc.next();
            sc.nextLine();

            pstmt = con.prepareStatement("update PLAYLIST set Name = ? where Number = ?");
            pstmt.setString(1, newName);
            pstmt.setInt(2, pNum);
            pstmt.executeQuery();

            System.out.println("Update name success");
        } else if(pMenu == 2) {
            System.out.print("Enter song name: ");
            String songName = sc.next();
            sc.nextLine();

            pstmt = con.prepareStatement("select Number from SONG where Title = ?");
            pstmt.setString(1, songName);
            rs = pstmt.executeQuery();
            rs.next();
            int songNum = rs.getInt("Number");

            pstmt = con.prepareStatement("insert into CONTAIN values(?, ?, ?)");
            pstmt.setInt(1, songNum);
            pstmt.setInt(2, pNum);
            pstmt.setString(3, subSsn);
            rs = pstmt.executeQuery();

            System.out.println("Add song success!");
        } else if(pMenu == 3) {
            System.out.print("Enter song name: ");
            String songName = sc.next();
            sc.nextLine();

            pstmt = con.prepareStatement("select Number from SONG where Title = ?");
            pstmt.setString(1, songName);
            rs = pstmt.executeQuery();
            if(rs.next()) {
                int songNum = rs.getInt("Number");

                pstmt = con.prepareStatement("delete from CONTAIN where SNum = ?");
                pstmt.setInt(1, songNum);
                pstmt.executeQuery();
            } else {
                System.out.println("No such song");
            }
        } else {
            System.out.println("No such play list!!!");
        }
    }
}
```

-> play list 이름 변경, 곡추가, 곡 삭제 중 원하는 항목을 입력 받고 각 항목에 따라

필요한 것들을 입력 받은 뒤 쿼리를 이용해 이를 실행한다.

(5) 4번 Play Songs를 선택하면

```
    } else if(secondMenu == 4) {
        pstmt = con.prepareStatement("select Name from PLAYLIST");
        rs = pstmt.executeQuery();
        while(rs.next()) {
            System.out.println("Playlist name : " + rs.getString("Name"));
        }
        System.out.print("Enter play list name: ");
        String pName = sc.next();
        sc.nextLine();

        pstmt = con.prepareStatement("select song.Title, song.PlayTime, a.Name, song.PlayCount, song.Number from music.PLAYLIST, music.SONG, music.ARTIST where song.PlayListName = ? and song.Number = ? and a.Name = ?");
        pstmt.setString(1, pName);
        rs = pstmt.executeQuery();
        int count = 0;
        System.out.println("Play " + pName);

        ResultSet tmpRs;
        PreparedStatement tmpPstmt;

        while(rs.next()) {
            count++;
            System.out.println("(" + count + ") Title: " + rs.getString("song.Title") + ", Singer: " + rs.getString("a.Name"));
            tmpPstmt = con.prepareStatement("update SONG set PlayCount = ? where Number = ?");
            tmpPstmt.setInt(1, rs.getInt("song.PlayCount") + 1);
            tmpPstmt.setInt(2, rs.getInt("song.Number"));
            tmpRs = tmpPstmt.executeQuery();
        }

        if(count == 0) {
            System.out.println("Empty playlist");
        } else {
            pstmt = con.prepareStatement("select sum(song.PlayTime), count(song.Number) from music.PLAYLIST as p, music.SONG as s where p.Name = ?");
            pstmt.setString(1, pName);
            rs = pstmt.executeQuery();
            rs.next();

            System.out.println("The number of song: " + rs.getInt("count(song.Number)"));
            System.out.println("Total time: " + rs.getInt("sum(song.PlayTime)") / 60 + "minutes " + rs.getInt("sum(song.PlayTime)") % 60 + "seconds");
        }
    }
```

-> 현재 로그인 중인 subscriber가 가지고 있는 모든 playlist들을 출력해 보여준 뒤 재생하고 싶은 playlist를 입력받는다. 입력받은 play list가 존재하지 않거나 비어 있으면 에러 메시지를 출력하고 존재하고 비어있지 않다면 출력되는 노래의 제목과 가수를 출력하고 총 재생시간을 출력한다. 또한 각 음원들의 재생된 횟수를 1씩 증가시킨다.

(6) 5번 Edit Subscriber Information을 선택하면

```
    } else if(secondMenu == 5) {
        System.out.println("Which one will you edit?");
        System.out.println("1. Name\n2. NickName");
        System.out.print(">> ");
        int edit = sc.nextInt();
        sc.nextLine();

        if(edit == 1) {
            System.out.print("Enter new name: ");
            String newName = sc.next();
            sc.nextLine();
            pstmt = con.prepareStatement("update SUBSCRIBE set Name = ? where Ssn = ?");
            pstmt.setString(1, newName);
            pstmt.setString(2, subSsn);
            rs = pstmt.executeQuery();
        } else if(edit == 2) {
            System.out.print("Enter new nick name: ");
            String newNickName = sc.next();
            sc.nextLine();
            pstmt = con.prepareStatement("update SUBSCRIBE set NickName = ? where Ssn = ?");
            pstmt.setString(1, newNickName);
            pstmt.setString(2, subSsn);
            rs = pstmt.executeQuery();
        } else {
            System.out.println("Wrong Input!!!");
        }
    }
```

-> 이름과 nick name 중 어떤 항목을 변경할지 입력 받고 이에 따라 쿼리를 이용해 새로운 정보로 update한다.

(7) 6번 Search Song을 선택하면

```

} else if(secondMenu == 6) {
    System.out.println("1. Search with title\n2. Search with artist\n3. Search with lyrics");
    System.out.print(">> ");
    int tmp = sc.nextInt();
    sc.nextLine();

    if(tmp == 1) {
        System.out.print("Enter title: ");
        String title = sc.next();
        sc.nextLine();
        pstmt = con.prepareStatement("select song.Title, a.Name from SONG as song, SING as sing, ARTIST as a");
        pstmt.setString(1, "%" + title + "%");
        rs = pstmt.executeQuery();

        if(rs.next()) {
            do {
                System.out.println("Title: " + rs.getString("song.Title") + ", Singer: " + rs.getString("a.Na");
            } while (rs.next());
        } else {
            System.out.println("No result!");
        }
    } else if(tmp == 2) {
        System.out.print("Enter artist name: ");
        String aName = sc.next();
        sc.nextLine();
        pstmt = con.prepareStatement("select song.Title, a.Name from SONG as song, SING as sing, ARTIST as a");
        pstmt.setString(1, "%" + aName + "%");
        rs = pstmt.executeQuery();

        if(rs.next()) {
            do {
                System.out.println("Title: " + rs.getString("song.Title") + ", Singer: " + rs.getString("a.Na");
            } while (rs.next());
        } else {
            System.out.println("No result!");
        }
    } else if(tmp == 3) {
        System.out.print("Enter lyrics name: ");
        String lyrics = sc.next();
        sc.nextLine();
        pstmt = con.prepareStatement("select song.Title, a.Name from SONG as song, SING as sing, ARTIST as a");
        pstmt.setString(1, "%" + lyrics + "%");
        rs = pstmt.executeQuery();

        if(rs.next()) {
            do {
                System.out.println("Title: " + rs.getString("song.Title") + ", Singer: " + rs.getString("a.Na");
            } while (rs.next());
        } else {
            System.out.println("No result!");
        }
    } else {
        System.out.println("Wrong Input!!!");
    }
}

```

-> 제목, 가수, 가사로 찾기 중 한 개를 선택받는다. 입력받은 결과를 바탕으로 이를 포함하고 있는 곡이 있을 경우 play count의 수가 큰 곡부터 내림차순으로 출력한다.

(8) 0을 입력받으면 이전 메뉴로 돌아가고 이외의 것들이 입력으로 들어올 시에는 에러 문구를 출력하고 다시 Subscriber Menu를 출력한다.

수행되는 기능

1. MANAGER의 경우 (Name: Donald, Nick name: Duck인 경우)

```
DB 접속 성공
----- Wellcome to My Music APP -----
1. Manager Menu
2. Subscriber Menu
0. Exit
>> 1
Enter your nick name: Duck

***** Wellcome Donald *****
----- Manager Menu -----
1. Add Songs
2. Delete Songs
3. Add Artist
4. Delete Subscribers
5. Edit Manager Information
6. Show All Managing
0. Return to Previous Menu
>> 3
Enter artist name: Maroon 5
Enter artist debut(ex. 1900-03-28 \if you don't know, enter 0): 2004-03-19
Add artist success!
----- Manager Menu -----
1. Add Songs
2. Delete Songs
3. Add Artist
4. Delete Subscribers
5. Edit Manager Information
6. Show All Managing
0. Return to Previous Menu
>> 1
How many songs will you add? 2
Enter 1th song name: Sugar
Enter 1th song's lyrics: sugar sugar power
Enter 1th song's play time: 200
Enter 1th song's singer: Maroon 5
Enter 1th song's writer: Maroon 5
Enter 1th song's composer: Maroon 5
Add song success!
Enter 2th song name: Memories
Enter 2th song's lyrics: And the memories bring back memories bring back you
Enter 2th song's play time: 190
Enter 2th song's singer: Maroon 5
Enter 2th song's writer: Maroon 5
Enter 2th song's composer: Maroon 5
Add song success!
----- Manager Menu -----
1. Add Songs
2. Delete Songs
3. Add Artist
4. Delete Subscribers
5. Edit Manager Information
6. Show All Managing
0. Return to Previous Menu
>> 2
How many songs will you delete? 1
Enter 1th song number: 4
No such song or no permission
----- Manager Menu -----
1. Add Songs
2. Delete Songs
3. Add Artist
4. Delete Subscribers
5. Edit Manager Information
6. Show All Managing
0. Return to Previous Menu
>> 6
No subscriber managing
Song:
Title: "Sugar", Number: 5
Title: "Memories", Number: 6
.. ..
```

```
----- Manager Menu -----
1. Add Songs
2. Delete Songs
3. Add Artist
4. Delete Subscribers
5. Edit Manager Information
6. Show All Managing
0. Return to Previous Menu
>> 2
How many songs will you delete? 1
Enter 1th song number: 6
Delete success!
```

```
----- Manager Menu -----
1. Add Songs
2. Delete Songs
3. Add Artist
4. Delete Subscribers
5. Edit Manager Information
6. Show All Managing
0. Return to Previous Menu
>> 4
Enter nick name of subscriber: Bear
No such subscriber or no permission
```

```
----- Manager Menu -----
1. Add Songs
2. Delete Songs
3. Add Artist
4. Delete Subscribers
5. Edit Manager Information
6. Show All Managing
0. Return to Previous Menu
```

```
>> 6
No subscriber managing
Song:
Title: "Sugar", Number: 5
```

```
----- Manager Menu -----
1. Add Songs
2. Delete Songs
3. Add Artist
4. Delete Subscribers
5. Edit Manager Information
6. Show All Managing
0. Return to Previous Menu
```

```
>> 0
----- Wellcome to My Music APP -----
1. Manager Menu
2. Subscriber Menu
0. Exit
```

2. SUBSCRIBER의 경우 (Name: Snow White, Nick name: Apple인 경우)

```
Enter your nick name: Apple

***** Wellcome Snow White *****
----- Subscriber Menu -----
1. Make New Playlist
2. Delete Playlist
3. Edit Playlist
4. Play songs
5. Edit Subscriber Information
6. Search Song
0. Return to Previous Menu
>> 4
Playlist name : Cristmas
Playlist name : Tell
Enter play list name: Cristmas
Play Cristmas
(1) Title: All I Want For Christmas Is You, Singer: Mariah Carey
(2) Title: Santa Tell Me, Singer: Ariana Grande
(3) Title: Santa Clause is Comin' to Town, Singer: Mariah Carey
The number of song: 3
Total time: 10minutes 49seconds
----- Subscriber Menu -----
1. Make New Playlist
2. Delete Playlist
3. Edit Playlist
4. Play songs
5. Edit Subscriber Information
6. Search Song
0. Return to Previous Menu
>> 2
Enter playlist name: Tell
Delete playlist success!

----- Subscriber Menu -----
1. Make New Playlist
2. Delete Playlist
3. Edit Playlist
4. Play songs
5. Edit Subscriber Information
6. Search Song
0. Return to Previous Menu
>> 5
Which one will you edit?
1. Name
2. NickName
>> 2
Enter new nick name: apple

----- Subscriber Menu -----
1. Make New Playlist
2. Delete Playlist
3. Edit Playlist
4. Play songs
5. Edit Subscriber Information
6. Search Song
0. Return to Previous Menu
>> 6
1. Search with title
2. Search with artist
3. Search with lyrics
>> 2
Enter artist name: Ariana
Title: Santa Tell Me, Singer: Ariana Grande
----- Subscriber Menu -----
1. Make New Playlist
2. Delete Playlist
3. Edit Playlist
4. Play songs
5. Edit Subscriber Information
6. Search Song
0. Return to Previous Menu
>> 6
1. Search with title
2. Search with artist
3. Search with lyrics
>> 3
Enter lyrics name: sugar
Title: Sugar, Singer: Maroon
```

사용된 SQL 명세

1. "SELECT Name, Ssn FROM music.MANAGER WHERE NickName = ?"
2. "SELECT MAX(Number) FROM SONG"
3. "select Number from ARTIST where Name = ?"
4. "select Number from ARTIST where Name = ?"
5. "select Number from ARTIST where Name = ?"
6. "insert into SONG values (?, ?, ?, ?, 0, ?)"
7. "insert into SING values(?, ?)"
8. "insert into WRITE_SONG values(?, ?)"
9. "insert into COMPOSE values(?, ?)"
10. "select Number from SONG as s, MANAGER as m where s.MgrSsn = ?"
11. "delete from SONG where Number = ?"
12. "select max(Number) from ARTIST"
13. "insert into ARTIST(Name, Number, MgrSsn) values(?, ?, ?)"
14. "insert into ARTIST values(?, ?, ?, ?)"
15. "select s.NickName from SUBSCRIBER as s, MANAGER as m where s.MgrSsn = ?"
16. "delete from SUBSCRIBER where NickName = ?"
17. "update MANAGER set Name = ? where Ssn = ?"
18. "update MANAGER set NickName = ? where Ssn = ?"
19. "select s.Name, s.NickName from music.MANAGER as m, music.SUBSCRIBER as s where m.Ssn = ? and m.Ssn = s.MgrSsn"
20. "select song.Title, song.Number from music.MANAGER as m, music.SONG as song where m.Ssn = ? and m.Ssn = song.MgrSsn"
21. "SELECT Name, Ssn FROM music.SUBSCRIBER WHERE NickName = ?"
22. "SELECT MAX(Number) FROM PLAYLIST"
23. "select Number from PLAYLIST where Name = ?"
24. "insert into PLAYLIST values (?, ?, ?)"
25. "select Number from PLAYLIST where Name = ?"
26. "delete from PLAYLIST where Name = ?"
27. "select Number from PLAYLIST where Name = ?"
28. "select Name from PLAYLIST"
29. "select song.Title, song.PlayTime, a.Name, song.PlayCount, song.Number from music.PLAYLIST as p, music.SING as sing, music.SONG as song, music.ARTIST as a, music.CONTAIN as c where c.PNum = p.`Number` and c.SNum = song.Number and song.Number = sing.SNum and sing.ArtNum = a.`Number` and p.name = ?"
30. "update SONG set PlayCount = ? where Number = ?"
31. "select sum(song.PlayTime), count(song.Number) from music.PLAYLIST as p, music.SING as sing, music.SONG as song, music.ARTIST as a, music.CONTAIN as c where c.PNum = p.`Number` and c.SNum = song.Number and song.Number = sing.SNum and sing.ArtNum = a.`Number` and p.name = ?"

32. "update SUBSCRIBE set Name = ? where Ssn = ?"
33. "update SUBSCRIBE set NickName = ? where Ssn = ?"
34. "select song.Title, a.Name, song.PlayCount from music.SONG as song, music.SING as sing, music.ARTIST as a where song.Title like '%Santa%' and song.`Number` = sing.SNum and sing.ArtNum = a.`Number` order by PlayCount DESC"

지난 Project 수정

1. MANAGER의 attribute를 Name, Ssn으로 줄이고 Ssn을 key attribute으로 설정하였다.
2. SUBSCRIBER의 attribute를 Name, Ssn, MgrSsn으로 줄이고 key attribute을 Ssn으로 foreign key를 MgrSsn으로 설정하였다.
3. PLAYLIST의 attribute을 Name, Number, SubSsn으로 줄이고 Number를 key attribute으로 SubSsn을 foreign key로 설정하였다.
4. ALBUM과 PARTICIPATE_IN, BELONG_TO table들을 삭제하였다.