**<데이터베이스시스템 및 응용>**

**데이터베이스 어플리케이션 개발\_과제4(DBMS 프로그램 개발)**

**컴퓨터공학부 컴퓨터전공**

**2016025096 강수진**

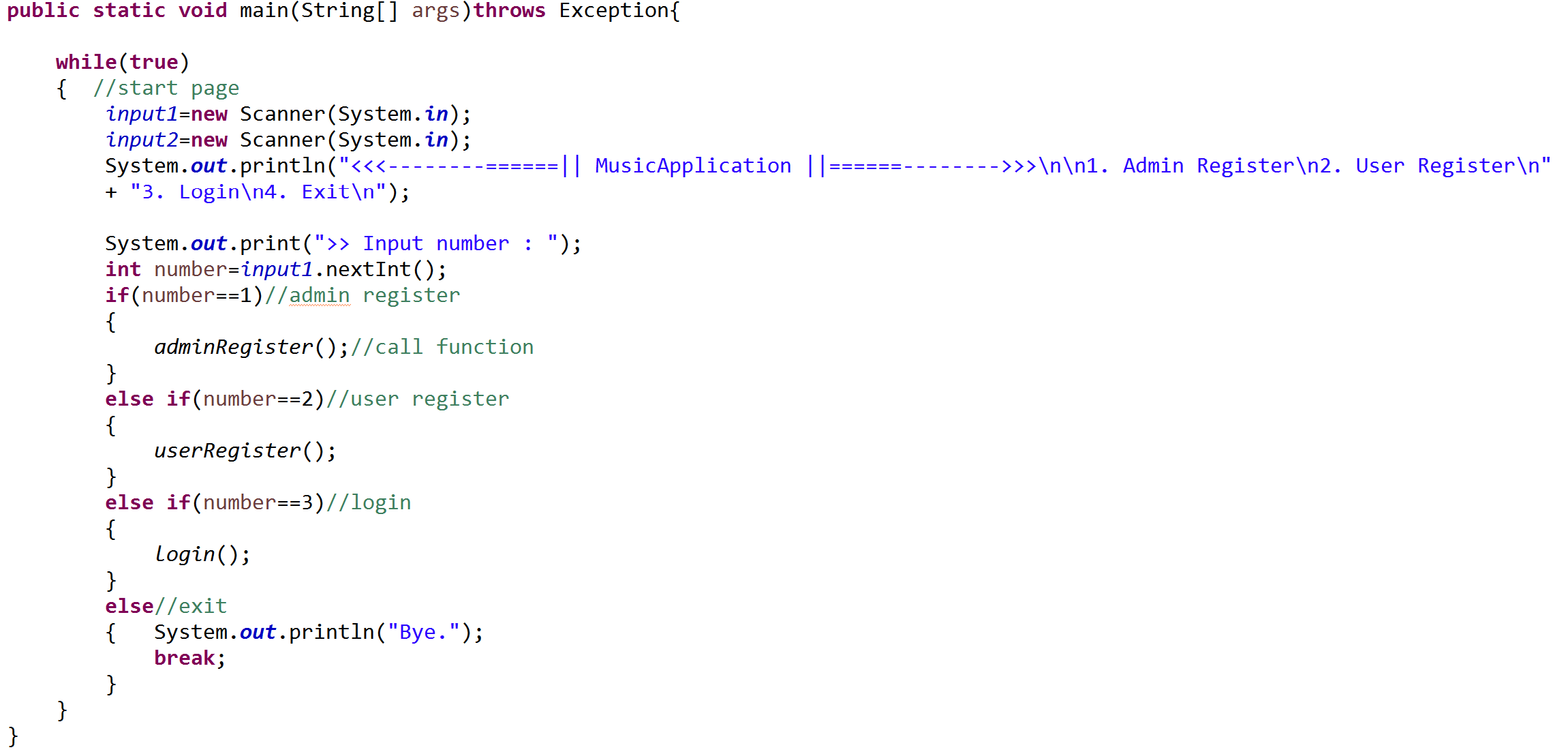
**\*프로그램 코드에 대한 자세한 설명/ 사용되는 SQL문 명세에 대한 것은 1번에, 수행되는 기능 스크린 샷에 대한 설명은 2번에 서술하겠습니다.**

**1. 프로그램 코드에 대한 자세한 설명 및 사용되는 SQL문 명세**

****

idNum은 사용자일 경우 사용자의 번호를 저장하는 용도이며 input1, input2는 각각 int형 입력, 문자열 입력하는 용도의 Scanner이다.

**1) main**

****

메인 문에서는 처음에 ‘MusicApplication’을 실행할 때의 화면이 나타나고 이때에는

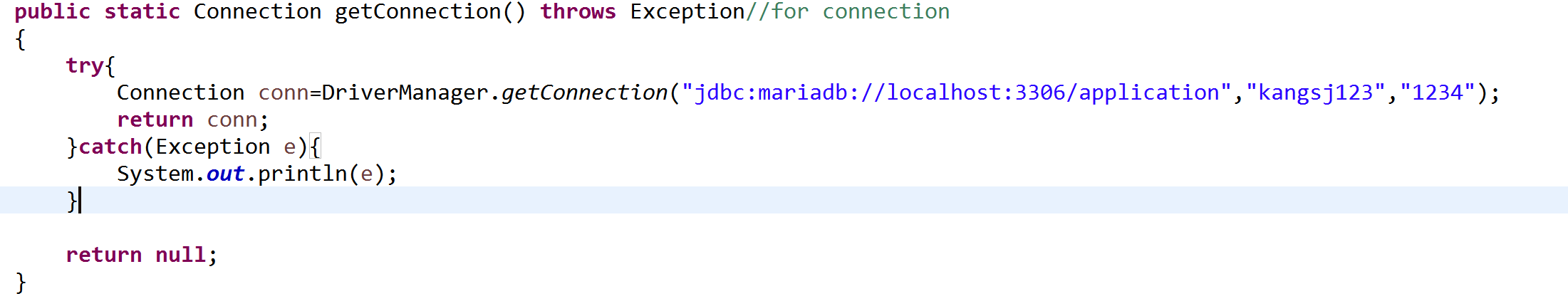
1. 관리자를 등록하거나 (Admin Register)

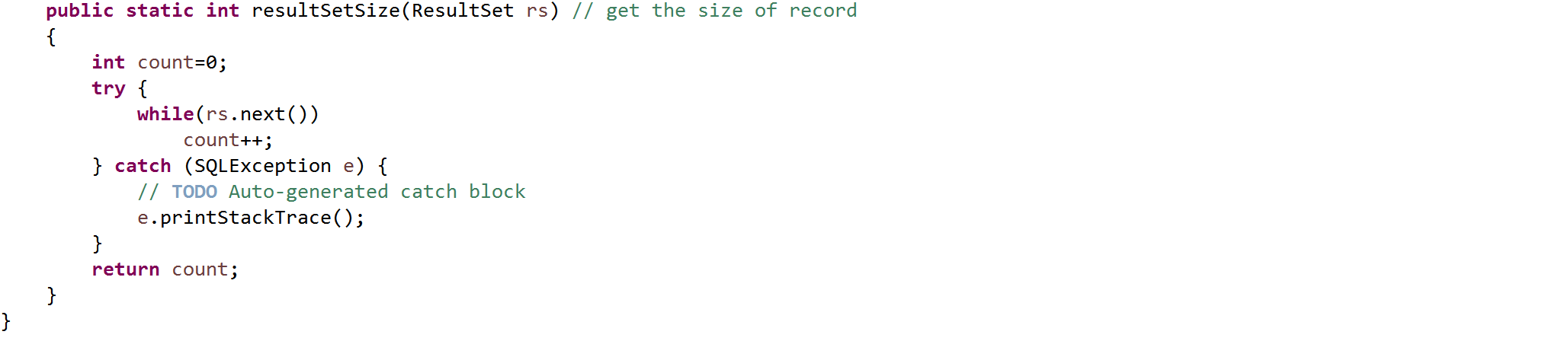
2. 사용자를 등록하거나 (User Register)

3. 등록된 관리자나 사용자가 로그인을 하거나(Login)

4. 프로그램을 종료할 수 있다.

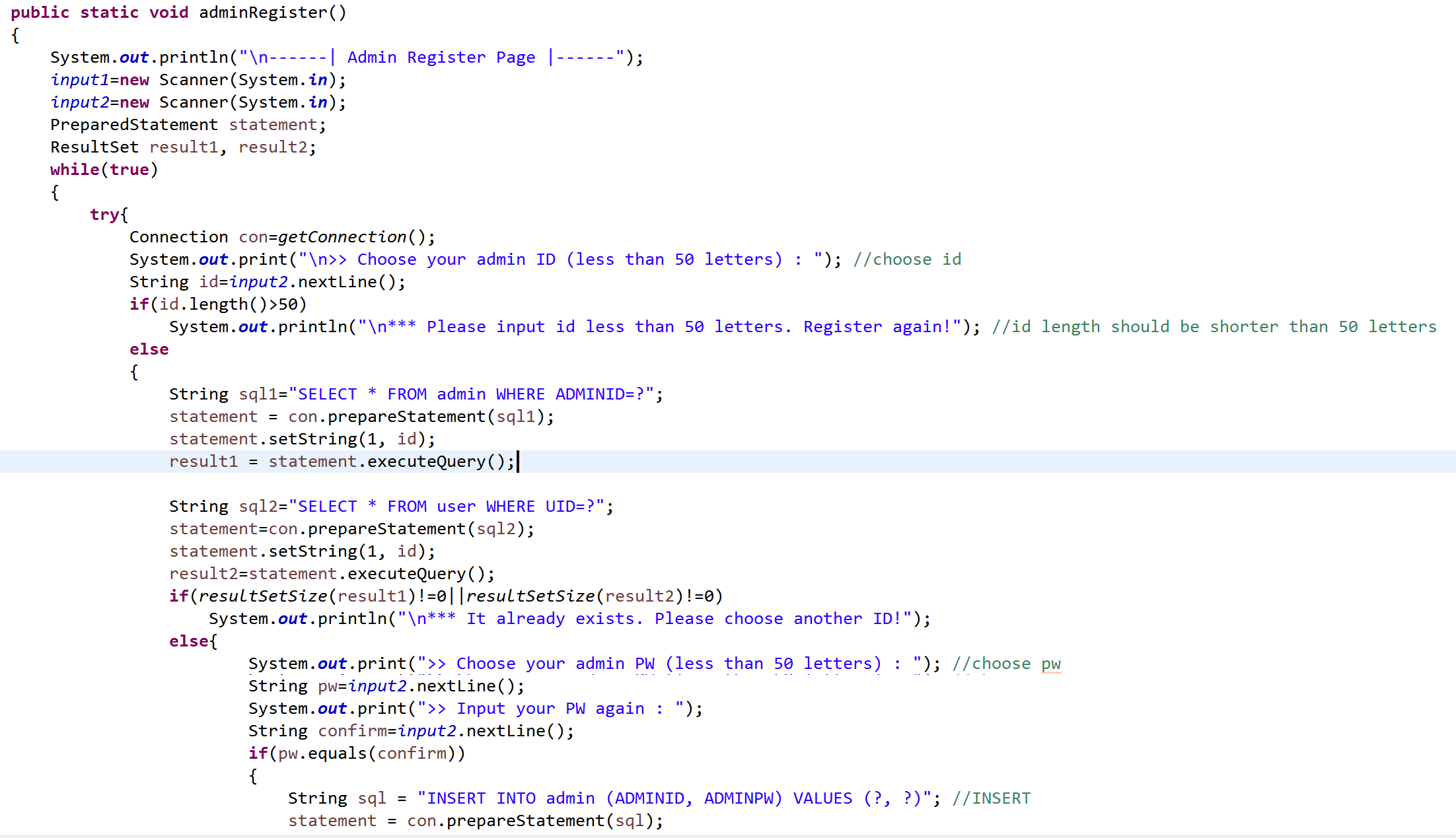
**2) getConnection**

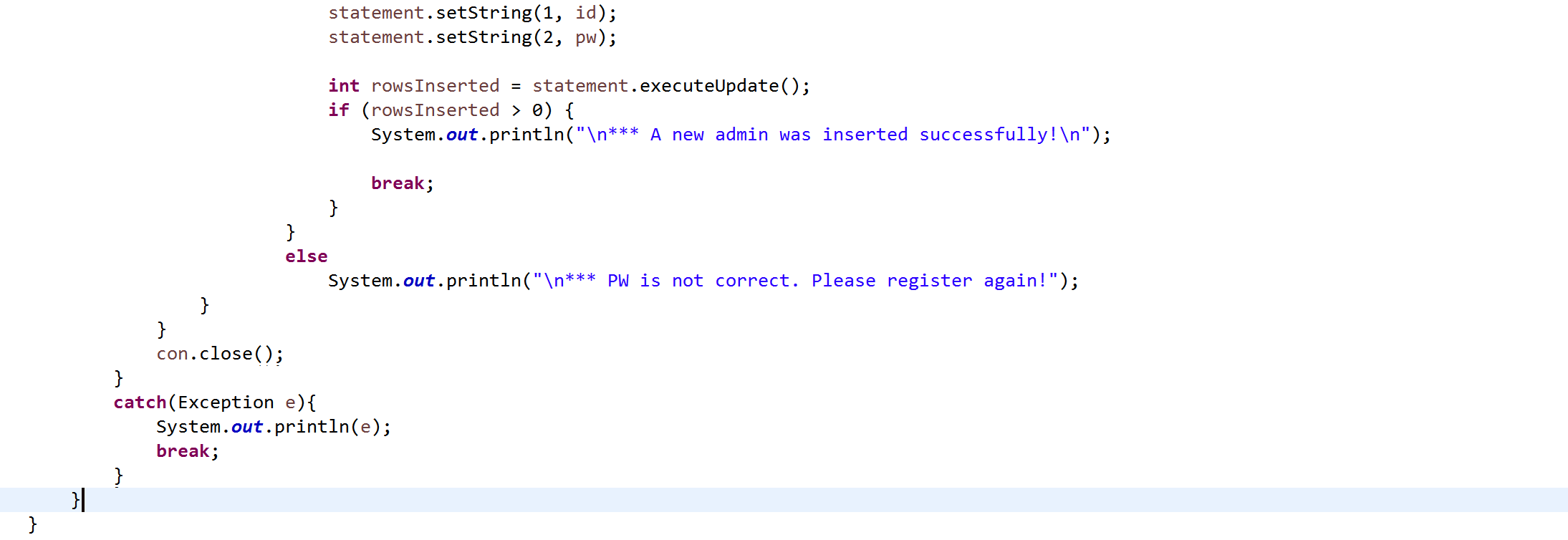
 해당 데이터베이스 application과 연결하는 메소드로, sql을 실행하는 메소드 안에 이를 호출한다.



select하여 얻은 ResultSet rs에 저장된 레코드의 개수를 의미한다. rs.next()가 존재할 때까지 끝까지 count를 구해서 return하였다.

**3) Admin Register**





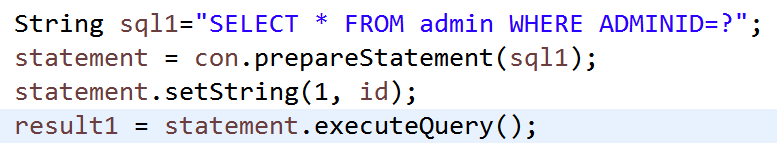
**관리자를 등록하는 페이지이다**.

등록할 id를 입력하고 그 id가 이미 admin이나 user테이블에 존재하면 다른 id를 입력하도록 한다.

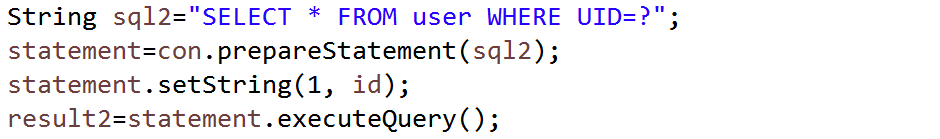
그 id가 유일한 값이면 pw를 입력하도록 하고 한 번 더 확인의 의미로 pw를 입력한 후 서로 일치하면 admin테이블에 insert하면 된다.

**[SQL문]**

-위 코드에 존재하는 sql문에 대해 설명을 덧붙였습니다.



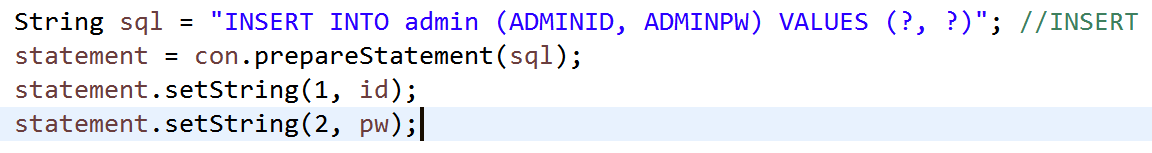
등록하려는 아이디가 admin 테이블에 있는지 확인하기 위해 ADMINID=id인 레코드를 admin테이블에서 SELECT한다.



등록하려는 아이디가 user 테이블에 있는지 확인하기 위해 UID=id인 레코드를 user테이블에서

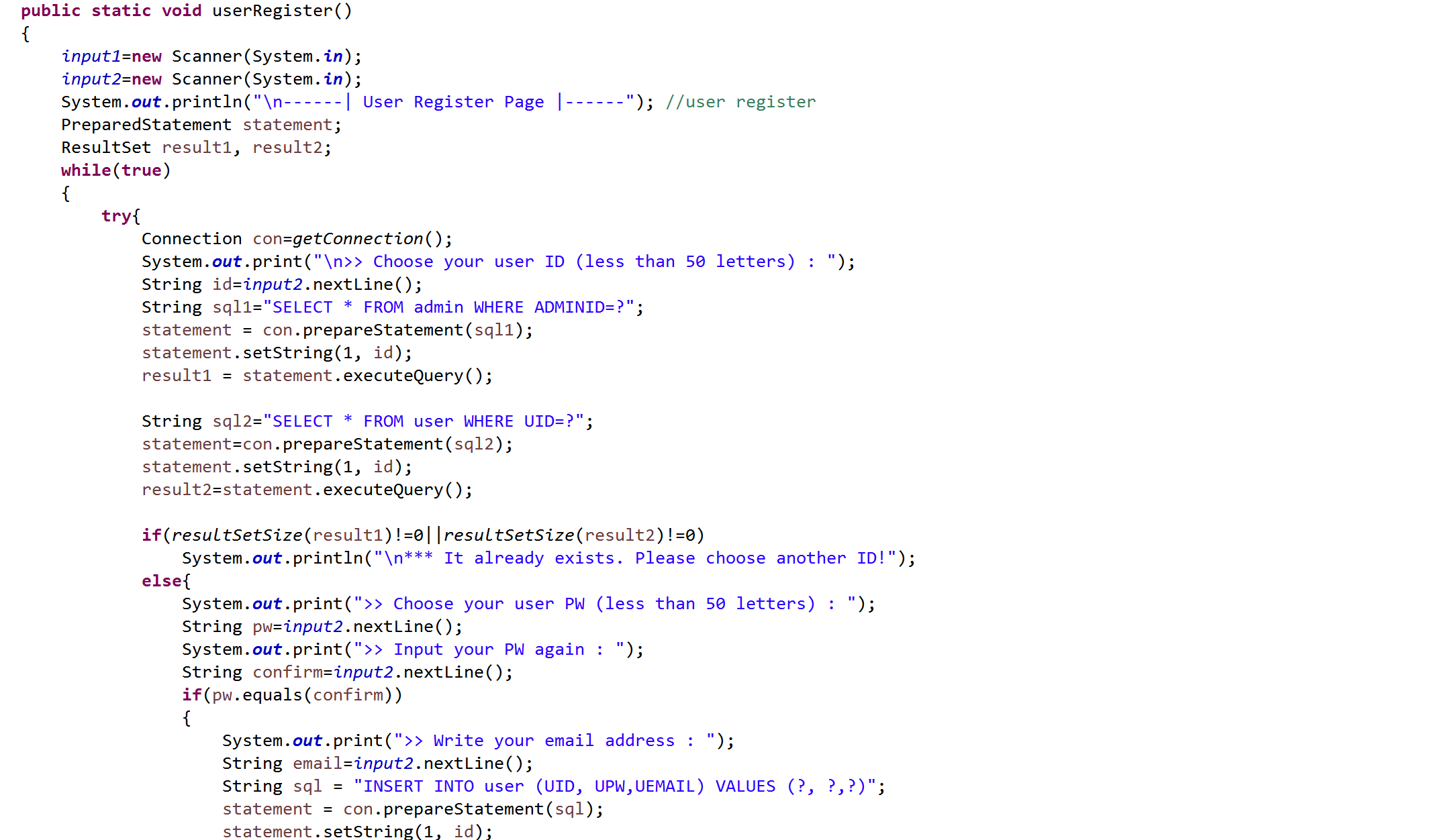
SELECT한다.

이 두 테이블에 모두 없는 id만이 관리자의 등록될 수 있다. (사용자도 마찬가지)

****

관리자 등록 페이지에서 입력된 id와 비밀번호를 admin테이블에 insert하는 sql문이다.

**4) user Register**





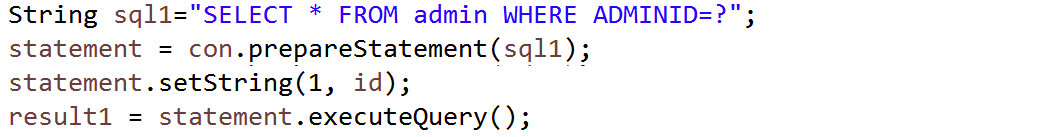
**사용자를 등록하는 페이지이다**.

관리자와 마찬가지로 id를 입력하고 그 id와 pw가 user나 admin 테이블에 존재하지 않으면 pw와 email주소를 입력하고 insert하였으며 (pw와 다시 입력한 pw가 일치할 경우에)

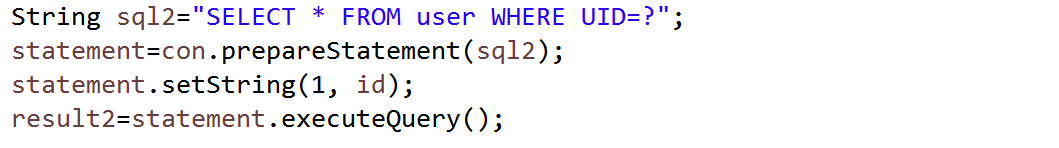
User나 admin테이블에 존재하면 id를 다시 입력하도록 화면을 전환했다.

**[SQL문]**

-위 코드에 존재하는 sql문에 대해 설명을 덧붙였습니다.



로그인하는 과정은 관리자와 사용자 모두 같은 페이지에서 이뤄지므로 입력한 아이디가 admin테이블에 존재하는지 여부를 파악해야 한다. 따라서 SELECT로 해당 레코드를 살펴본다.



입력한 아이디와 일치하는 레코드가 user테이블에 존재하는지를 살펴보기 위해 user테이블에서

SELECT를 통해 살펴본다.

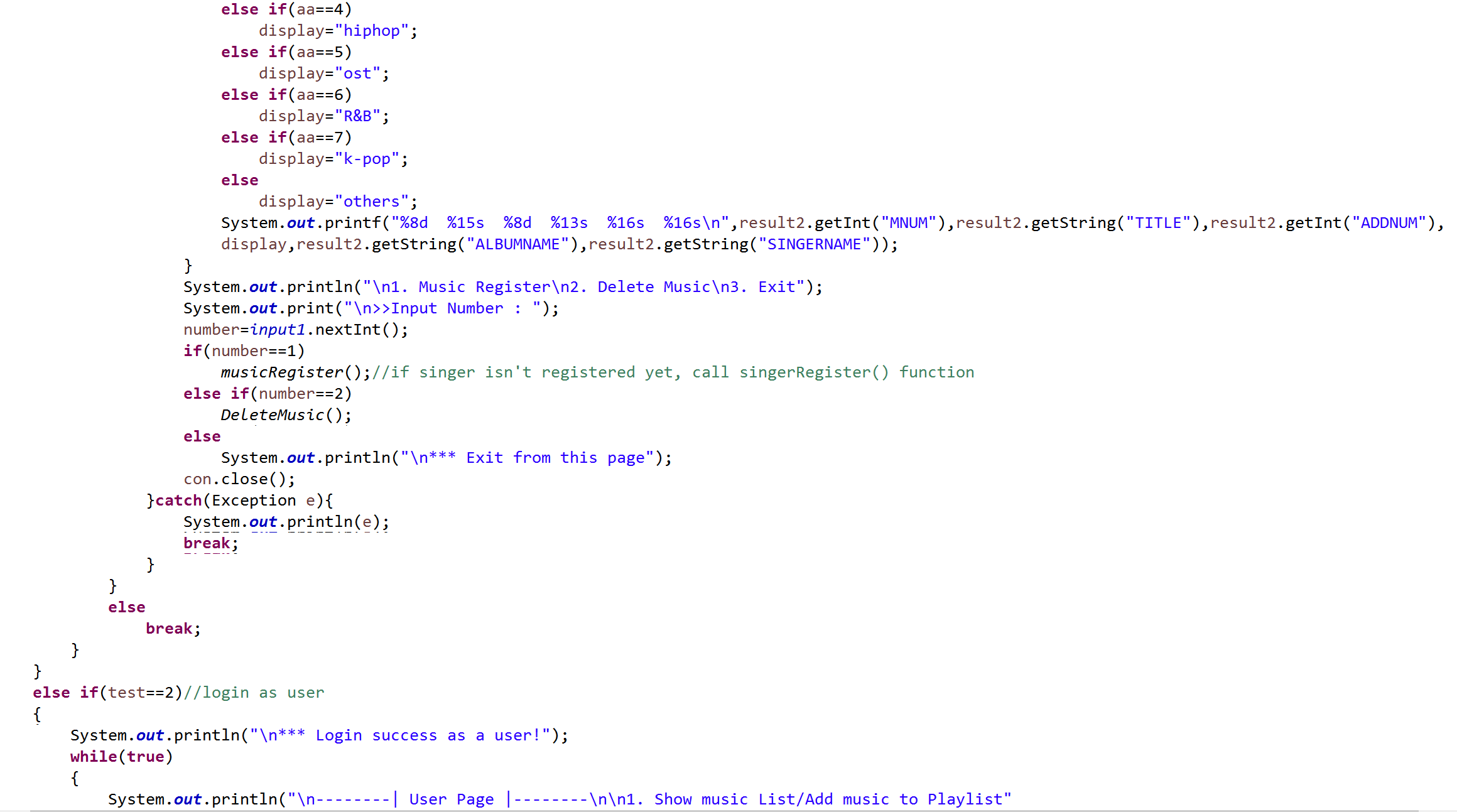
**5) login**

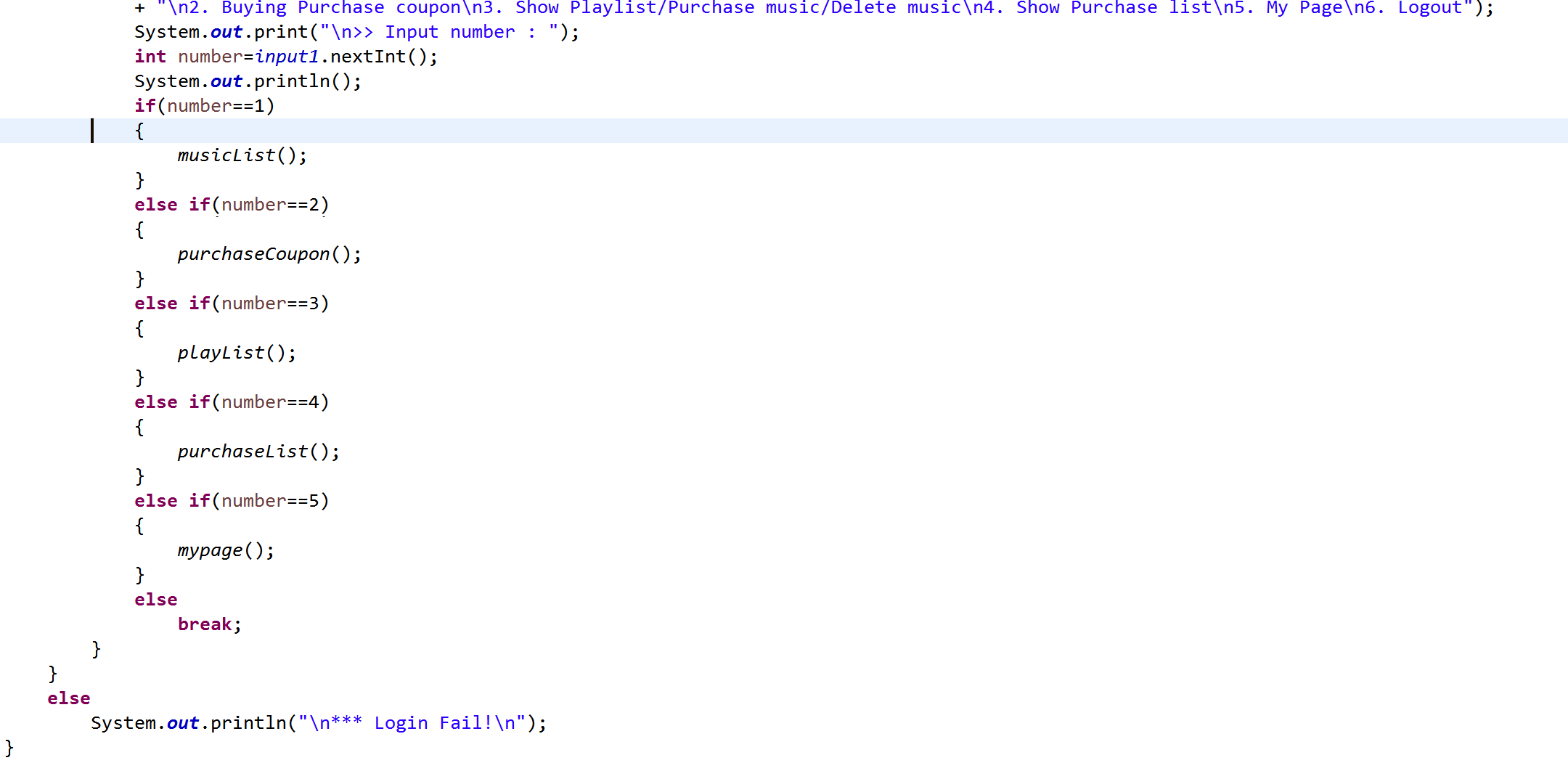












로그인할 때의 페이지를 구성하였다. 로그인 페이지는 관리자나 사용자 모두 접속이 가능하며

**관리자의 경우** 로그인에 성공하면 관리자의 권한이 모인 페이지로,

**사용자의 경우** 로그인에 성공하면 사용자의 권한이 모인 페이지로 이동한다.

따라서 로그인을 하기 위해 id와 pw를 입력한 후, user 테이블과 admin 테이블을 모두 찾아보아야 한다.

사용자 테이블에서 동일한 id와 pw의 레코드가 발견되면 user로 로그인 되는 것이며

관리자 테이블에서 동일한 id와 pw의 레코드가 발견되면 admin으로 로그인 되는 것이다.

이때 user로 로그인 되는 경우에는 idNum을 사용자 테이블의 UNUM으로 바꿔주어 이후에 select나 insert 등의 기능을 구현할 때 사용한다.

**<관리자>**

**test==1이면 관리자로 로그인 되는 것이며**

**1. 가수 등록 및 삭제**

**2. 음악 등록 및 삭제**

**3. 종료**

**의 선택지가 있다.**

<각각 설명>

**1. 가수 등록 및 삭제**

일단 먼저 **전체 가수의 목록을 나열**하고 1. 가수를 등록하거나 2. 가수를 삭제할 수 있다.

숫자를 1을 입력하면 **가수를 등록하는 페이지**로,

2를 입력하면 **가수를 삭제하는 페이지**로

이동한다. (이동은 함수 호출로 이뤄진다.)

**2. 음악 등록 및 삭제**

일단 전체 음악 목록을 나열하고 1. 음악을 등록하거나 2. 음악을 삭제할 수 있다.

숫자를 1을 입력하면 **음악을 새로 등록하는 페이지**로,

2를 입력하면 **음악을 삭제하는 페이지**로

이동한다. (이동은 함수 호출로 이뤄진다.)

**3. 종료**

이외에는 이 페이지를 종료시키며(while문이었으므로 break를 통해 종료) 이전 페이지로 돌아간다.

**<사용자>**

**test==2이면 사용자로 로그인 되는 것이며**

**1. 음악 조회/ 플레이리스트에 음악 추가**

**2. 쿠폰 구매**

**3. 플레이리스트 보기/ 음악 구매/ 음악 삭제**

**4. 구매리스트 조회**

**5. 마이페이지**

**6. 로그아웃**

**의 선택지가 있다.**

<각각 설명-자세한 내용은 해당 메소드에서 설명>

**1. 음악 조회/ 플레이리스트에 음악 추가 – musicList() 메소드 호출**

음악을 1. 등록순 2. 인기순 3. 장르별로 조회가 가능하며

조회만 하고 종료할 수도 있고 원하는 곡을 추가할 수도 있다.

계속 곡을 추가할 수 있도록 하기 위해 종료를 원하는 경우에만 break를 실행하였다.

**2. 쿠폰 구매 – purchaseCoupon() 메소드 호출**

음악을 구매하기를 원하는 경우에는 쿠폰이 필요하기 때문에 쿠폰 구매 페이지로 이동한다.

**3. 플레이리스트 보기/ 음악 구매/ 음악 삭제 – playList() 메소드 호출**

그 사용자가 추가한 플레이리스트 목록을 조회할 수 있으며

이전 보고서에서 명시했듯이 플레이리스트 목록 중에서만 음악을 구매할 수 있으며(인기도 측정할 때 기록이 의미가 있으려면)

플레이리스트에서 음악을 삭제할 수 있다.

**4. 구매리스트 조회**

그 사용자가 구매한 곡 목록을 조회할 수 있다.

**5. 마이페이지**

사용자의 정보를 확인할 수 있다.

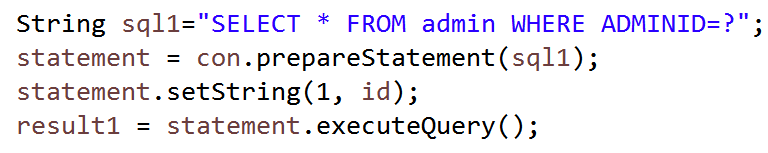
사용자 아이디, 비밀번호, 이메일, 구매쿠폰 개수 정보를 조회할 수 있다.

**6. 로그아웃**

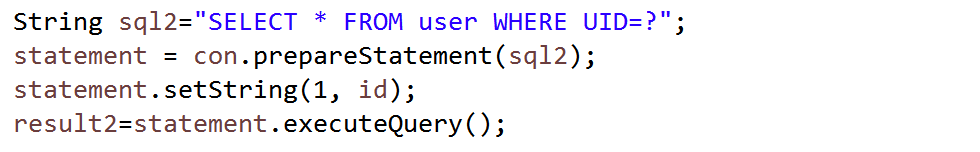
처음 페이지로 돌아간다.

**[SQL문]**

-위 코드에 존재하는 sql문에 대해 설명을 덧붙였습니다.



로그인을 관리자로 로그인 하는지 사용자로 로그인 하는지 알기 위해 먼저 admin에서 입력한 id와 일치하는 ADMINID가 존재하는지 SELECT를 통해 살펴보고 레코드의 개수를 파악한다.



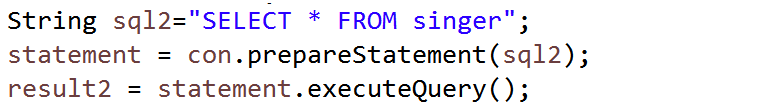
사용자로 로그인 하는지 알기 위해 user에서 입력한 id와 일치하는 UID가 존재하는지 SELECT를 통해 살펴보고 레코드의 개수를 파악한다.

이렇게 살펴본 후에 메소드 resultSetSize를 통해 admin테이블에서 파악한 레코드 개수가 1개이면 test변수의 값을 1로 바꾸고

user테이블에서 파악한 레코드 개수가 1개이면 test변수의 값을 2로 바꾼다.

test변수의 초기값은 0이므로 이 두 가지 경우 모두 해당하지 않을 경우에는 ‘Login Fail’의 메시지가 뜬다.

이후에는 test값이 1인 경우에는 admin권한에 해당하는 페이지를 보여주고 test값이 2인 경우에는 user권한에 해당하는 페이지를 보여주어 다음 페이지로 이동하도록 한다.



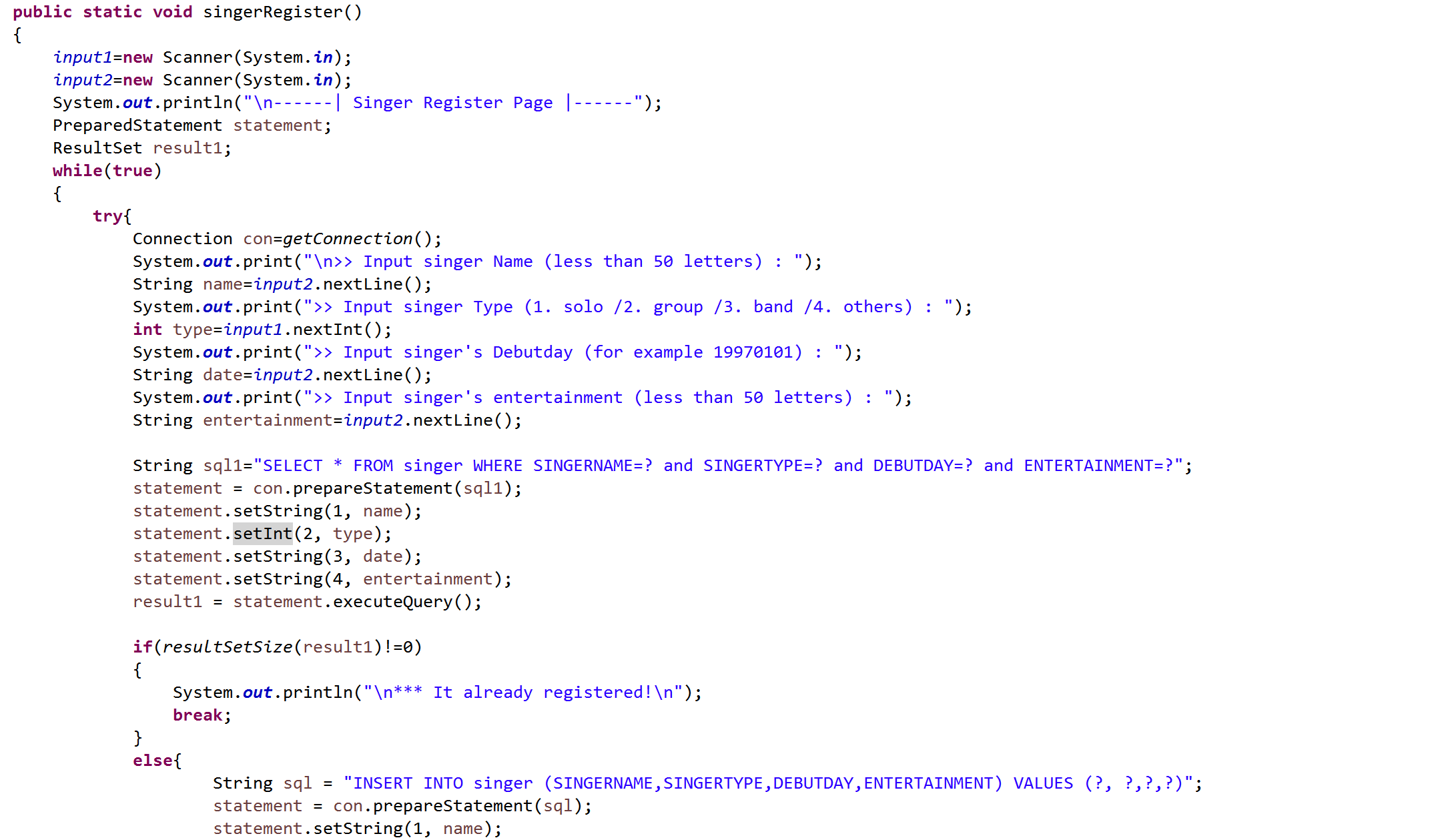
관리자 페이지에서 singer Register/Delete singer페이지로 이동한 경우에 일단 모든 가수의 목록을 보여준 후, 새로 등록하고 싶은 가수를 등록하거나 삭제하는 결정을 하는 것이 효과적이므로 SELECT를 통해서 singer테이블에 있는 모든 데이터를 가져온 후 화면에 한 줄씩 나타낸다.

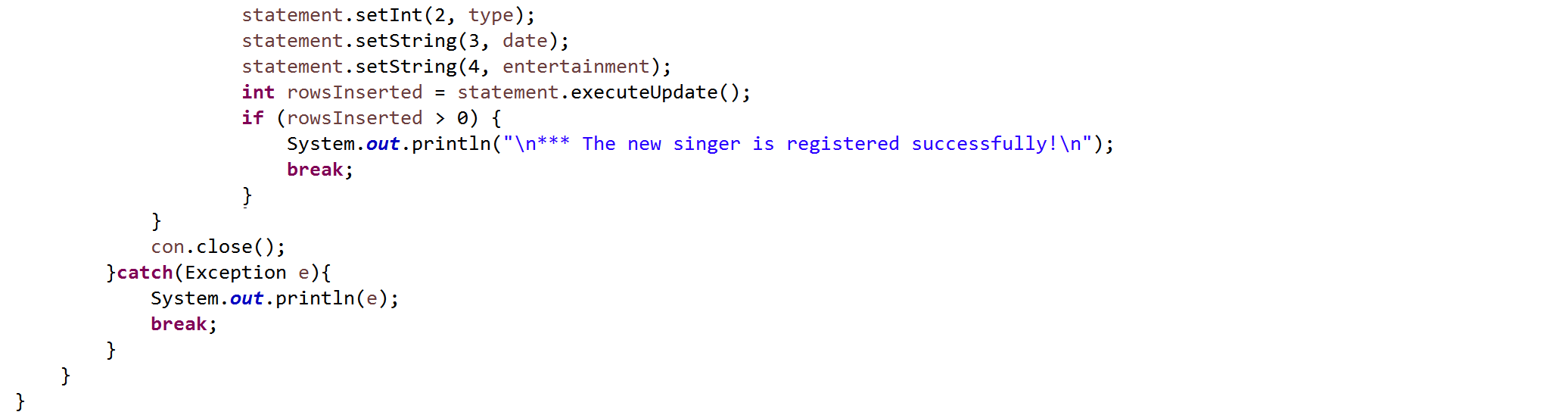


관리자 페이지에서 Music Register/Delete Music페이지로 이동한 경우에 일단 모든 음악의 목록을 보여준 후에 새로운 음악을 추가하거나 삭제한다.

이때 필요하므로 SELECT를 통해 music의 모든 데이터를 나타내며 이때 music테이블에서 SINGERN(가수 번호)를 외래키로 갖기 때문에 가수의 이름도 같이 나타내기 위해서 singer테이블의 SINGERNUM=SINGERN을 만족하는 데이터를 가져온다.

**6) singerRegister**

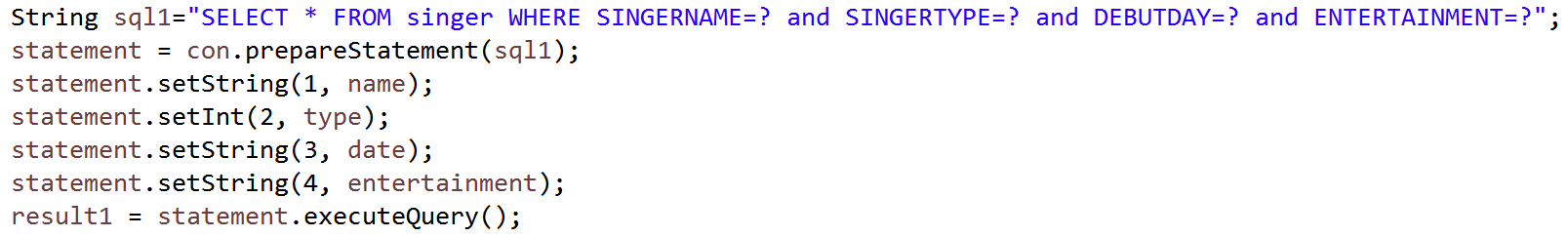
****

****

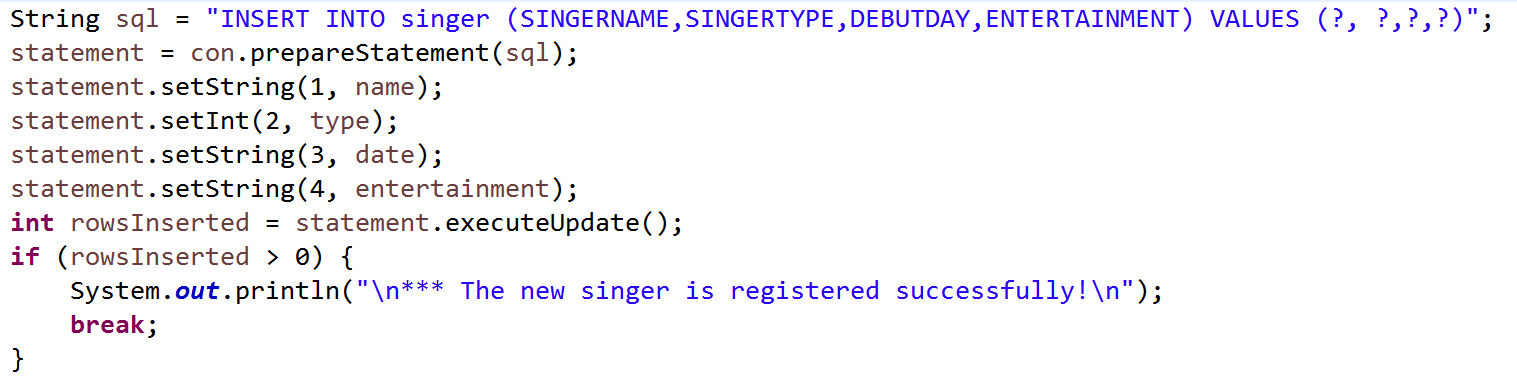
가수를 새로 등록하기 위해서 이름, 유형, 데뷔날짜, 엔터테인먼트를 입력하고 이미 존재하는 레코드이면 이미 등록되었다고 명시를 해준다.

아닐 경우에는 insert를 통해서 새로 가수를 등록한다.

**[SQL문]**

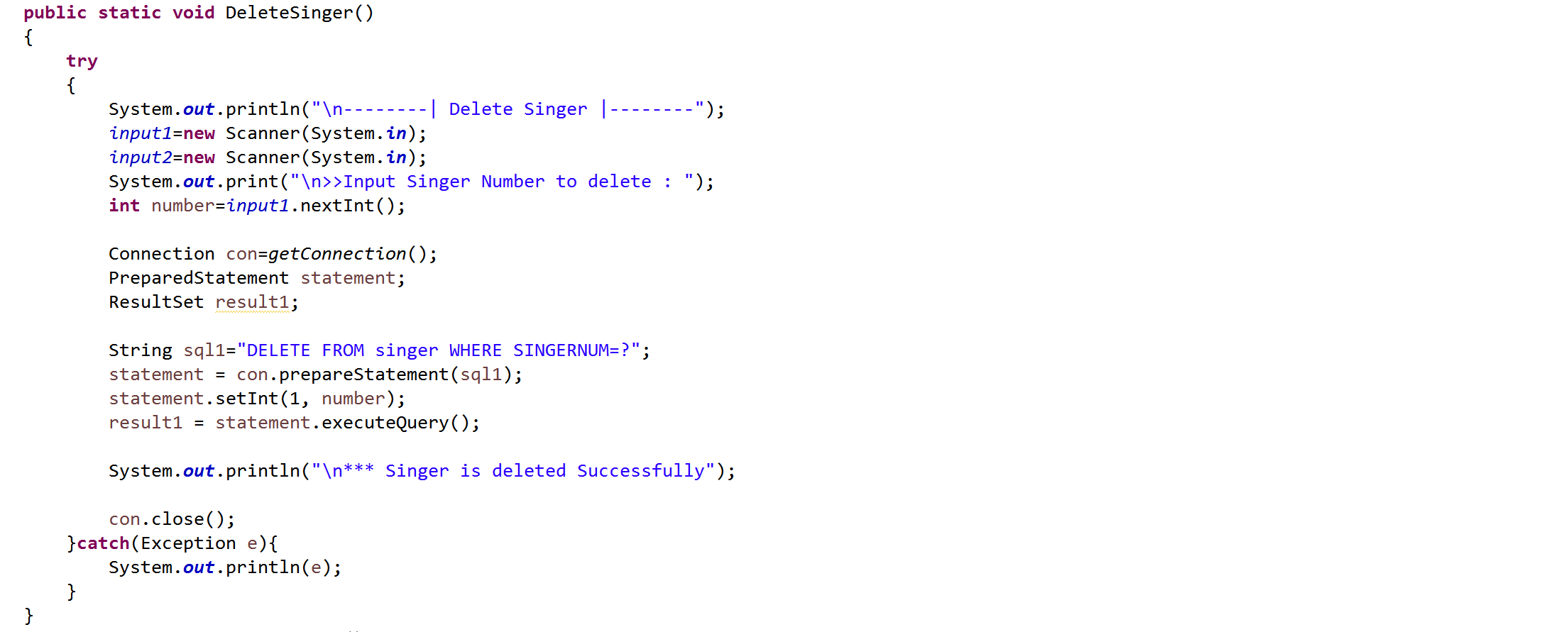
****

등록하려는 가수의 정보와 일치하는 데이터가 singer테이블에 없는지 확인하기 위해서 SELECT를 통해 해당 데이터가 존재하는지 확인한다. 이후에 resultSetSize메소드를 통해 아직 해당 데이터가 없다는 것을 알게 되면 아래와 같이



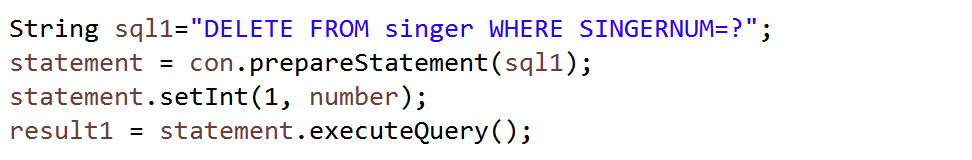
singer테이블에 해당 데이터를 insert한다.

**7) DeleteSinger**

****

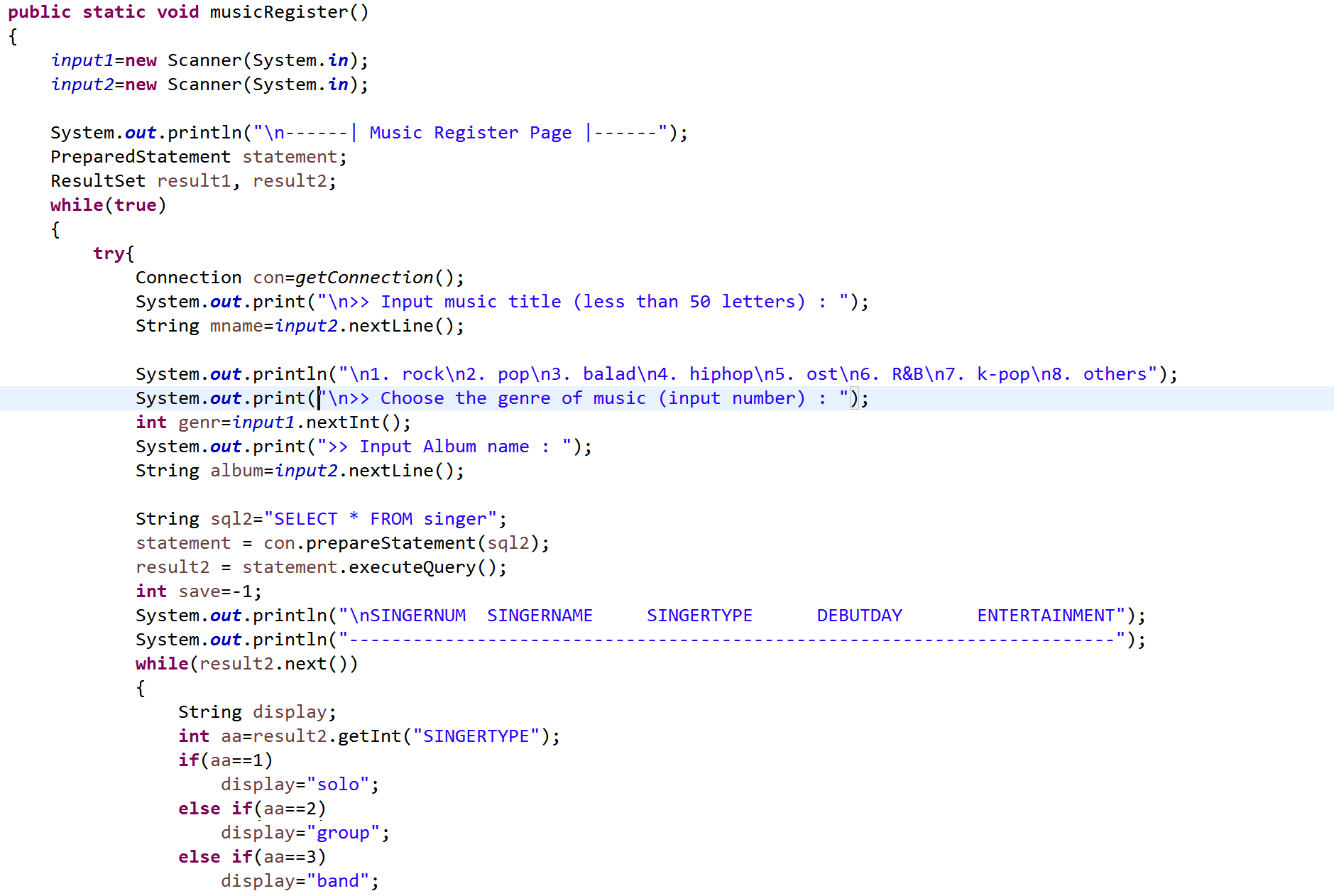
지우려는 가수의 번호를 입력하여 delete하는 과정을 거친다.

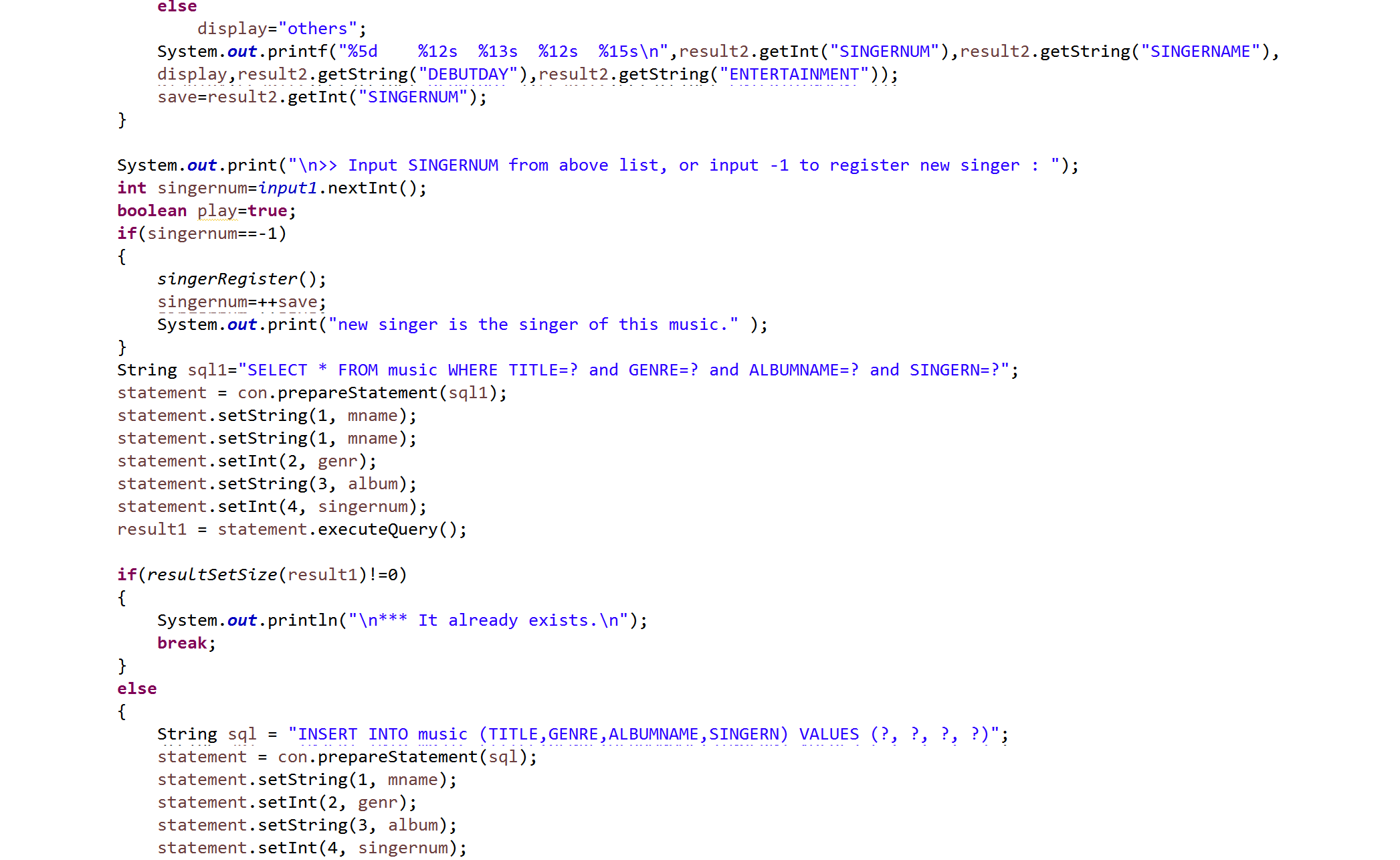
**[SQL문]**

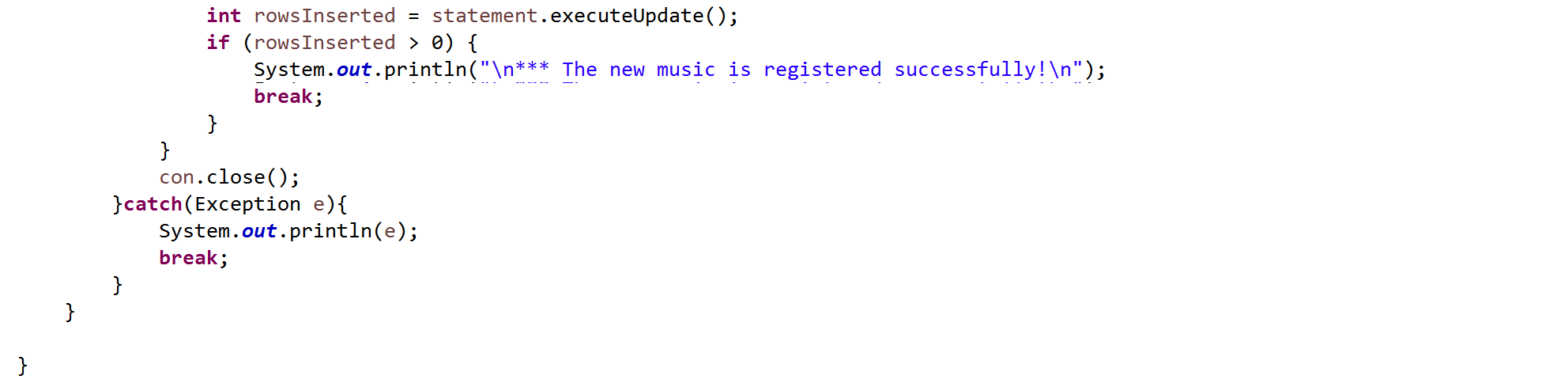
****

해당 가수를 삭제한다. DELETE를 이용하여 SINGERNUM이 입력한 number와 일치하는 데이터를 지운다.

**8) musicRegister**

****

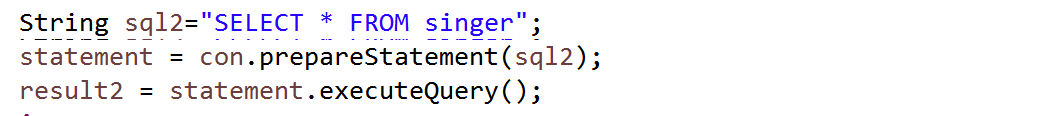
****

****

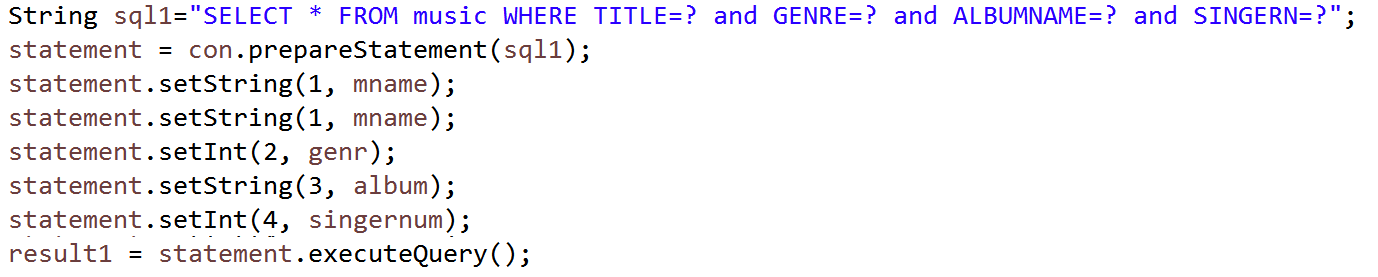
새로운 음악을 등록하기 위해서는 제목, 장르, 앨범 이름을 일단 입력한 후에 가수 목록을 보여준 후 그 음악의 가수가 존재하지 않으면 -1을 입력하여 새로운 가수를 등록할 수 있는 페이지로 전환된다.

새로운 가수가 등록되는 경우에는 가수가 등록됨과 동시에 음악이 등록된다.

**[SQL문]**

****

등록할 음악의 가수를 선택하기 위해서 전체 목록을 나열해야 한다. 따라서 SELECT를 통해서 데이터를 가져온다.

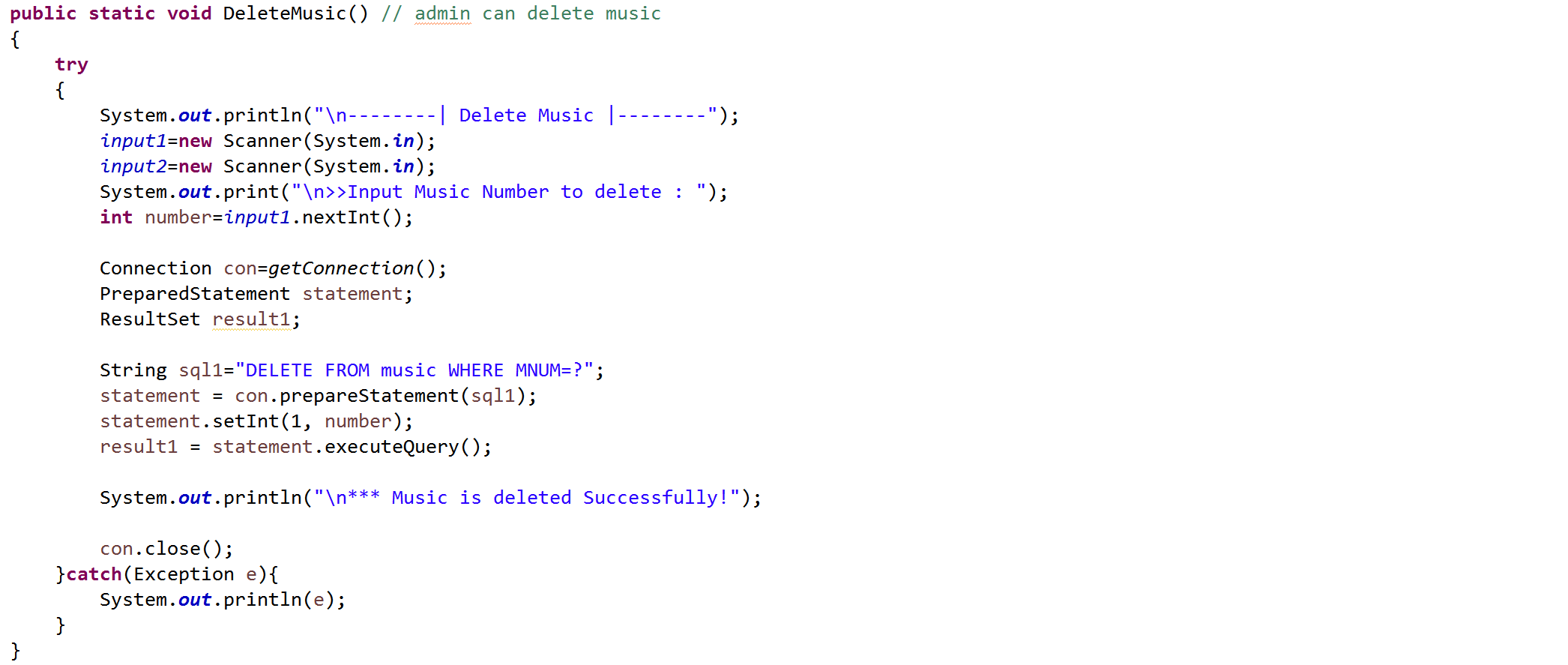
****

등록하려는 데이터가 중복되면 안되므로 이미 그 데이터가 존재하는지 SELECT를 통해서 데이터를 가져오고 개수가 0이 아니면 이미 존재하는 데이터라고 메시지를 띄운다.

****

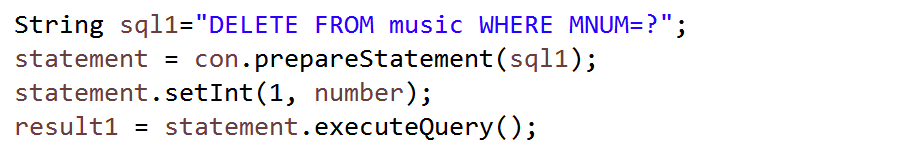
중복이 안되었다는 것이 확인이 되면 INSERT를 통해서 음악을 새로 등록한다.

**9) DeleteMusic**

****

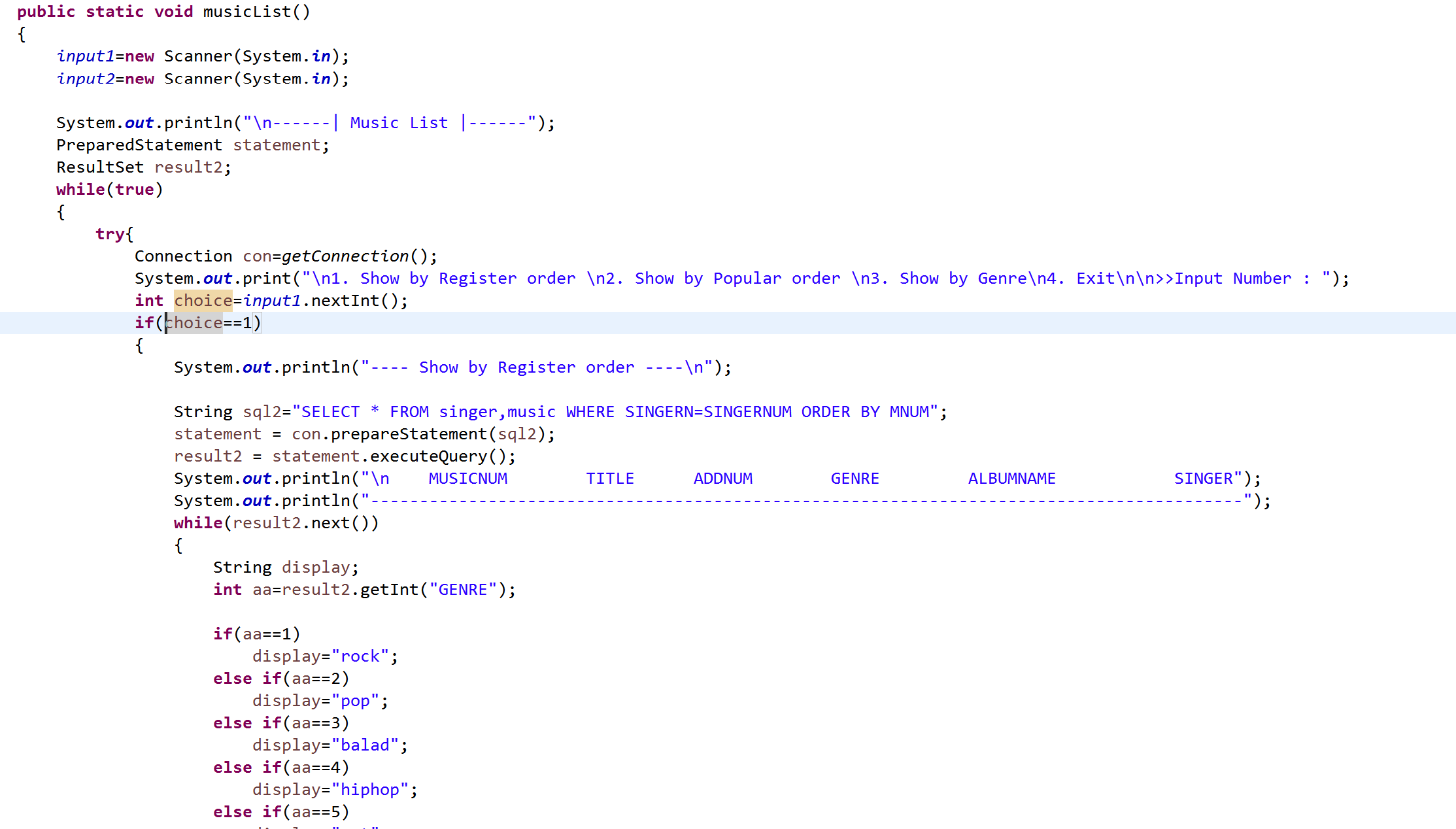
음악을 지우는 페이지로, 이전 함수에서 음악 목록을 전체 제시해주었으므로 원하는 음악 번호를 입력하면 그 테이블에서 해당 음악을 삭제한다.

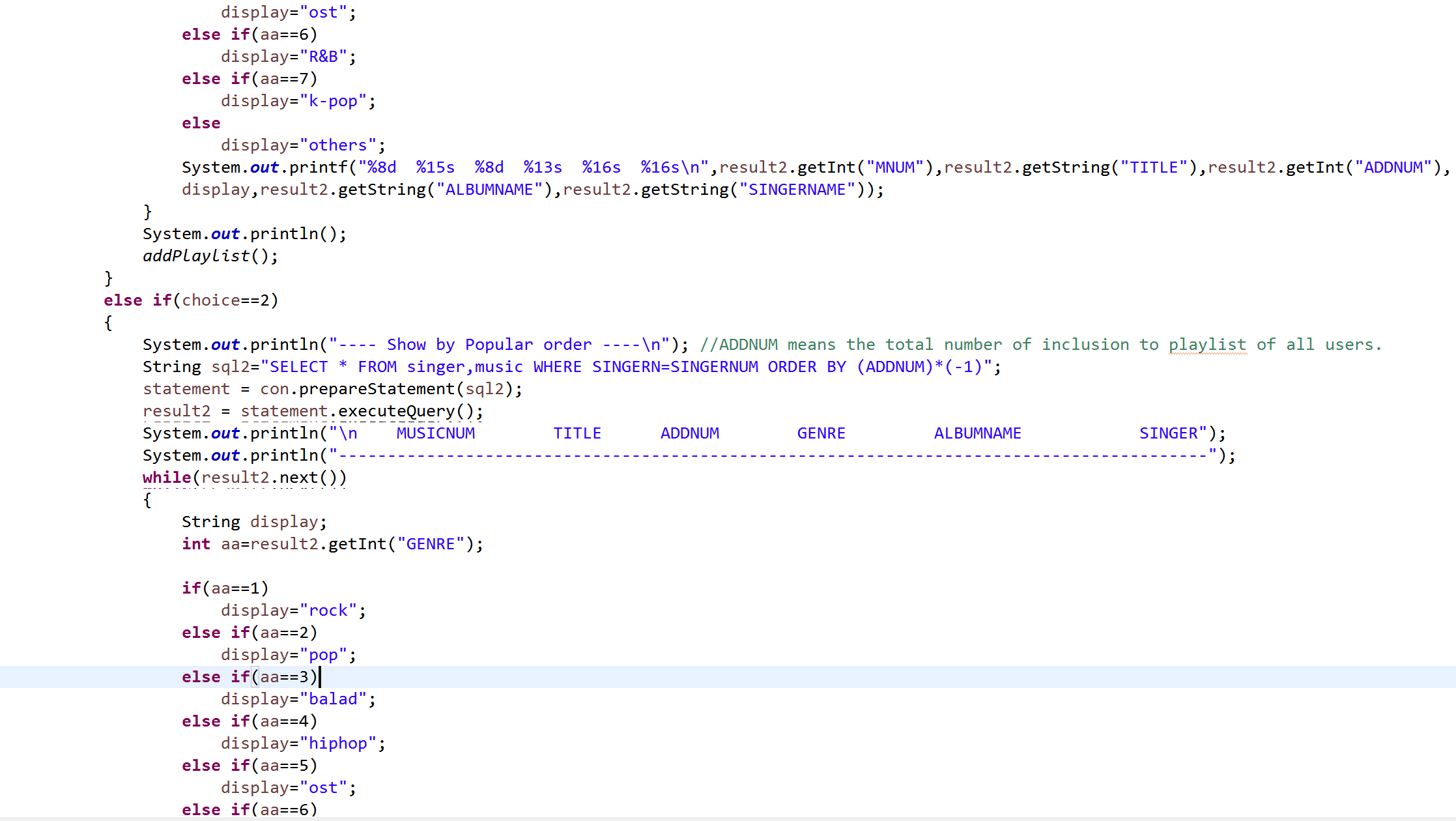
**[SQL문]**

****

지우려는 음악의 번호 MNUM이 입력한 숫자 number와 일치하는 데이터를 music테이블에서 지운다.(DELETE)

**10) musicList**

****

****

****

****

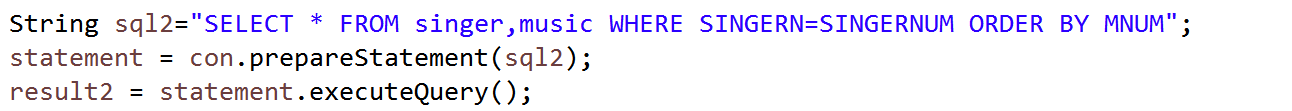
음악을 1. 등록 순 2. 인기 순 3. 장르별로 나열하는 기능을 가지고 있다. 각 번호를 입력하면 해당하는 음악의 목록이 순서대로 나열된다.

또한 addPlaylist 메소드를 호출하여 원하는 음악을 플레이리스트에 추가할 수 있으며

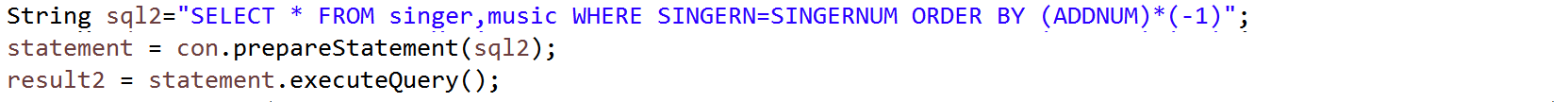
혹시 원하는 가수의 정보를 상세히 조회하고 싶을 때에는 조회가 가능하다.

장르는 번호를 입력하여 int형으로 저장하였는데 화면에 제시할 때에는 문자열로 변환하여 제시할 수 있도록 구현했다.

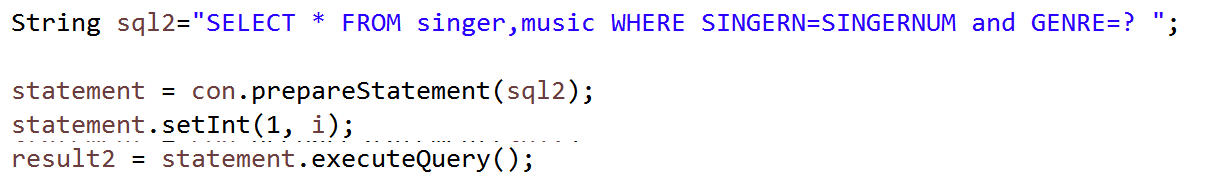
**[SQL문]**

****

등록순으로 음악을 조회할 때의 sql문으로 ORDER BY MNUM으로 음악 번호 순으로 나열하면 된다.

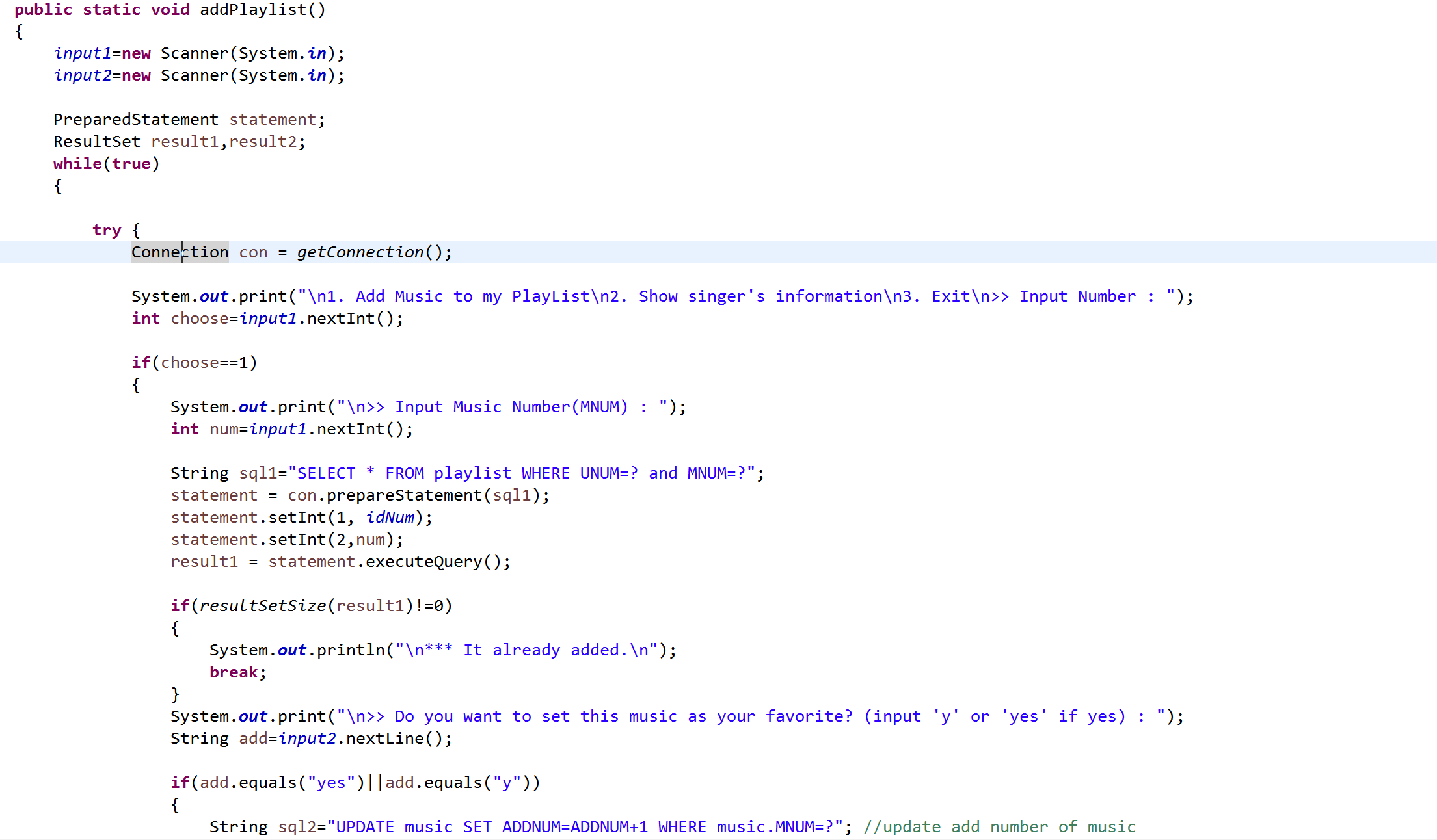
****

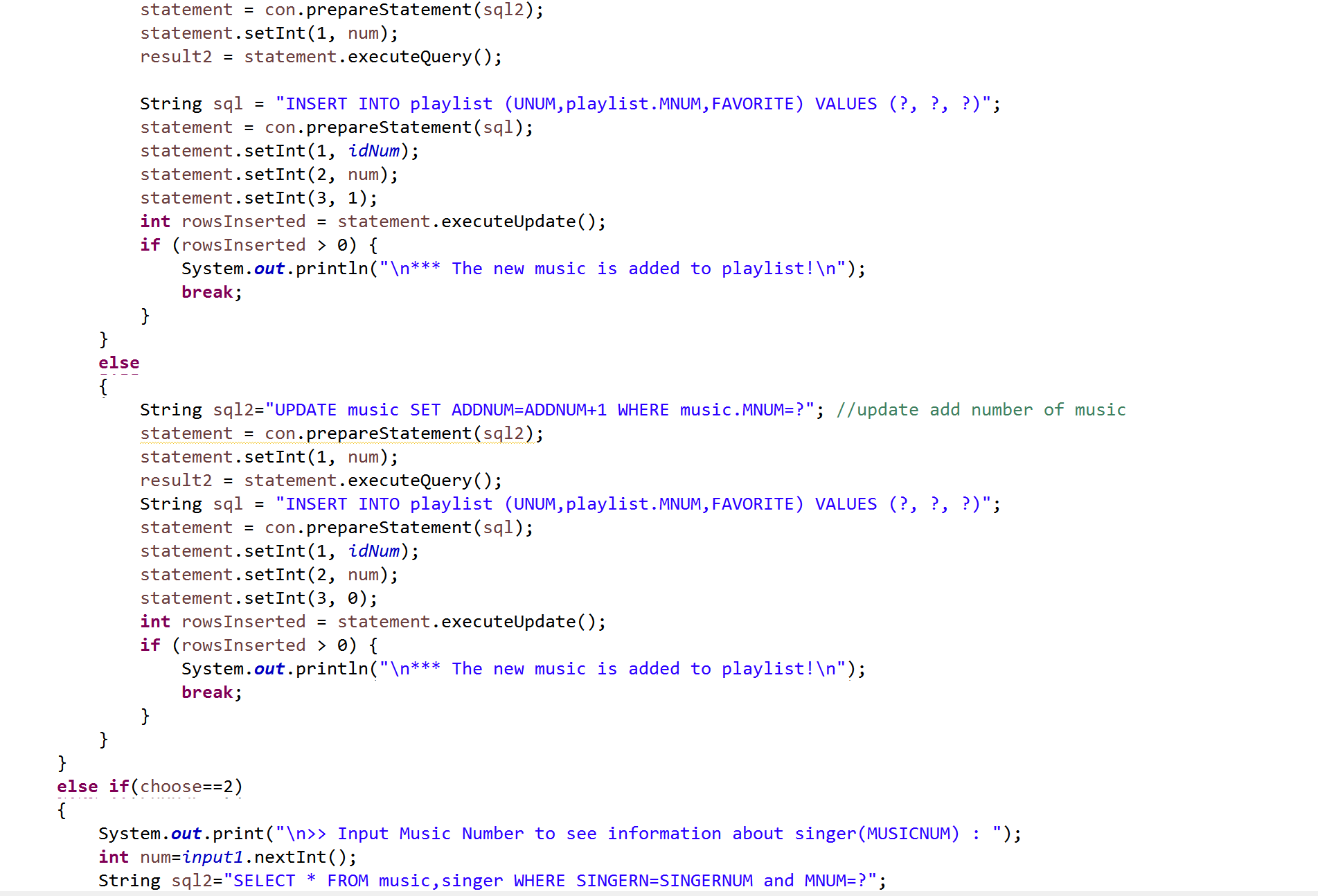
사용자들이 플레이리스트에 곡을 추가할 때마다 그 곡의 ADDNUM을 1씩 증가시키고 이는 결국 인기도를 의미하므로 인기순으로 음악을 조회할 때에는 ORDER BY (ADDNUM)\*(-1)로 나열하여 조회한다.

****

장르마다 곡의 목록을 조회할 때에는 포문을 이용하여 각 장르마다 GENRE=1, 2, 3, 4, 5, 6, 7, 8인 경우의 음악을 조회하여 나열한다.

**11) addPlaylist**







이 메소드에서는

1. 플레이리스트에 곡 추가

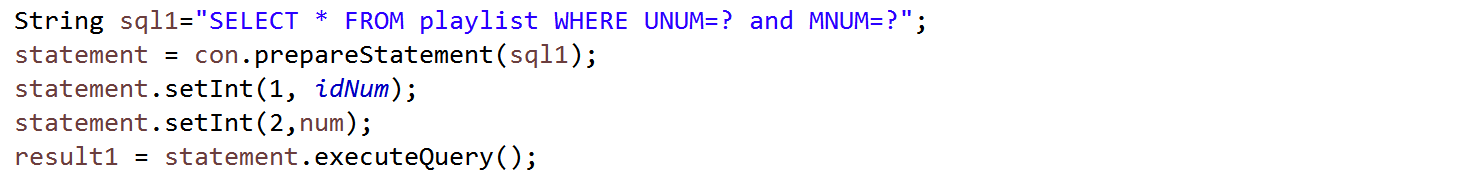
2. 가수 상세 조회

3. 종료

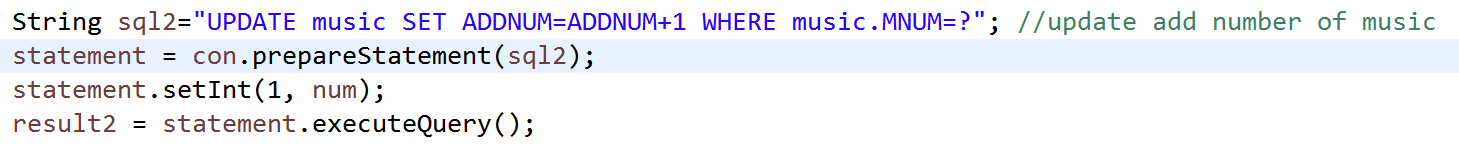
의 기능을 가진다.

가수 상세 조회에서는 MUSICNUM을 입력하면 그 음악의 가수에 대한 상세한 정보를 얻을 수 있다.

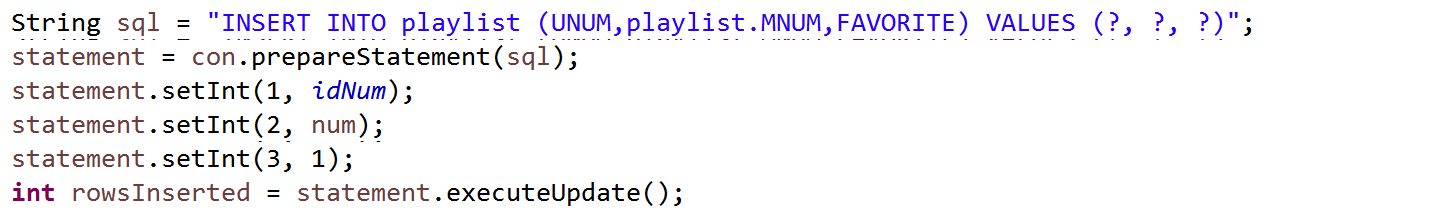
**[SQL문]**



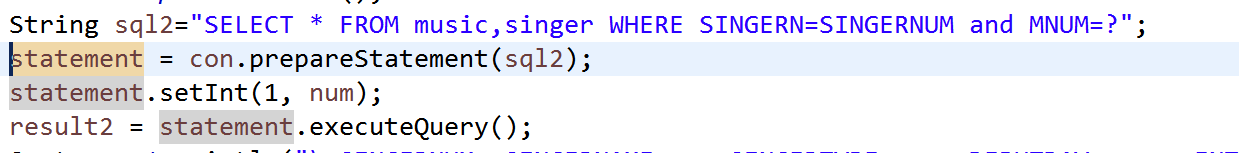
중복된 데이터를 넣지 않기 위해 SELECT를 통해 데이터를 모은 후 확인한다.



추가한 음악의 ADDNUM을 1증가시켜서 저장한다.(UPDATE)

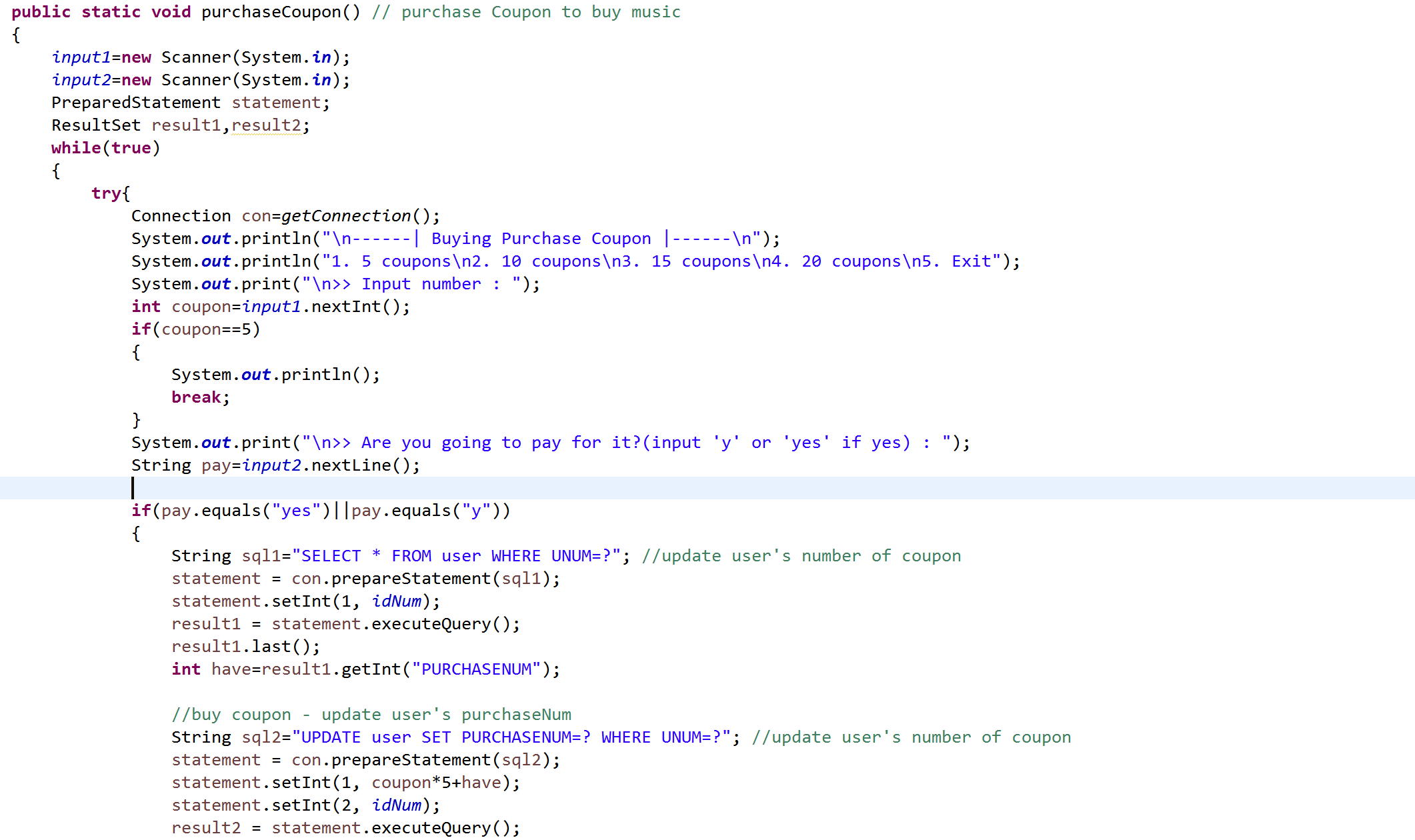


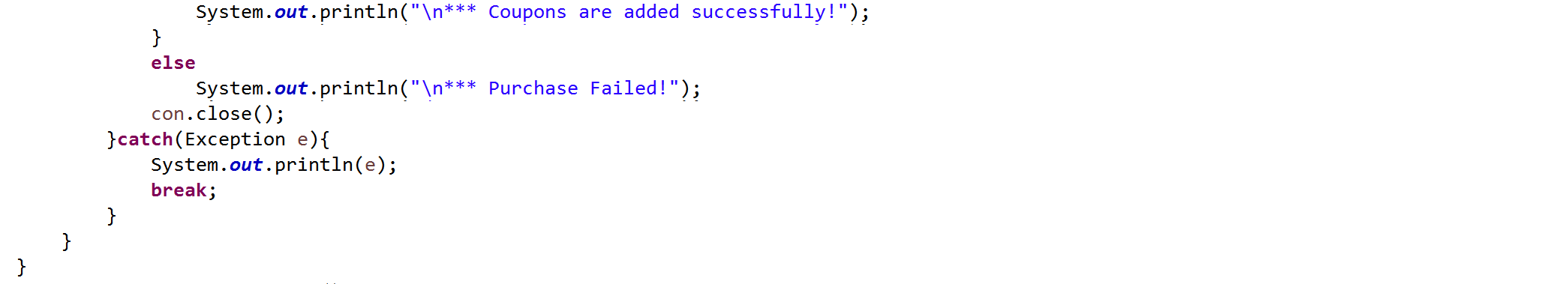
중복되지 않은 정보라는 것이 확인되면 playlist에 데이터를 INSERT한다. 이때 FAVORITE값은 0이냐 1이냐에 따라 0을 넣을 수도 있고 1을 넣을 수도 있다. (if로 구분하여 저장했다.)



상세히 보고 싶은 가수의 정보를 열람하기 위한 sql문이다. 따라서 SINGERN과 SINGERNUM이 같고 MNUM이 num과 같은 데이터를 music, singer로부터 가져온다.

**11) purchaseCoupon**

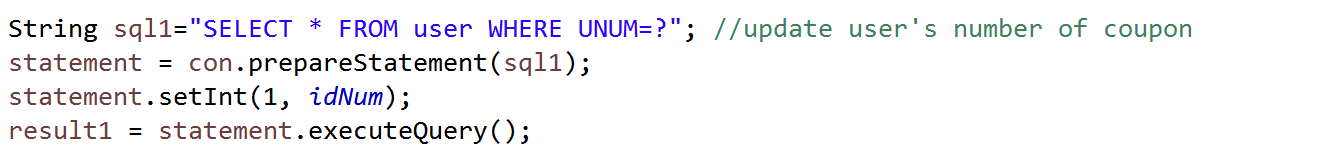




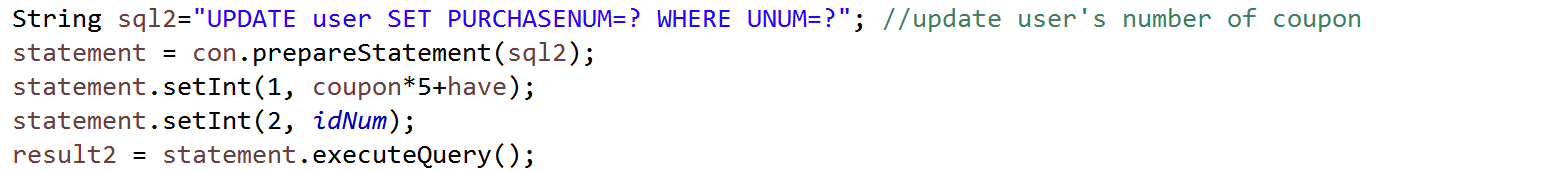
쿠폰을 구매하는 페이지로 5개, 10개, 15개, 20개 중에 선택하여 쿠폰을 구매할 수 있다. 이는 해당 사용자의 PURCHASENUM을 기존 것에 더한 값으로 update시키는 기능을 한다.

PURCHASENUM이란 현재 소유하고 있는 쿠폰의 개수로, 구매할 수 있는 음악의 수와 일치한다. 이후 음악을 구입하면 PURCHASENUM값을 1씩 감소시킨다.

**[SQL문]**

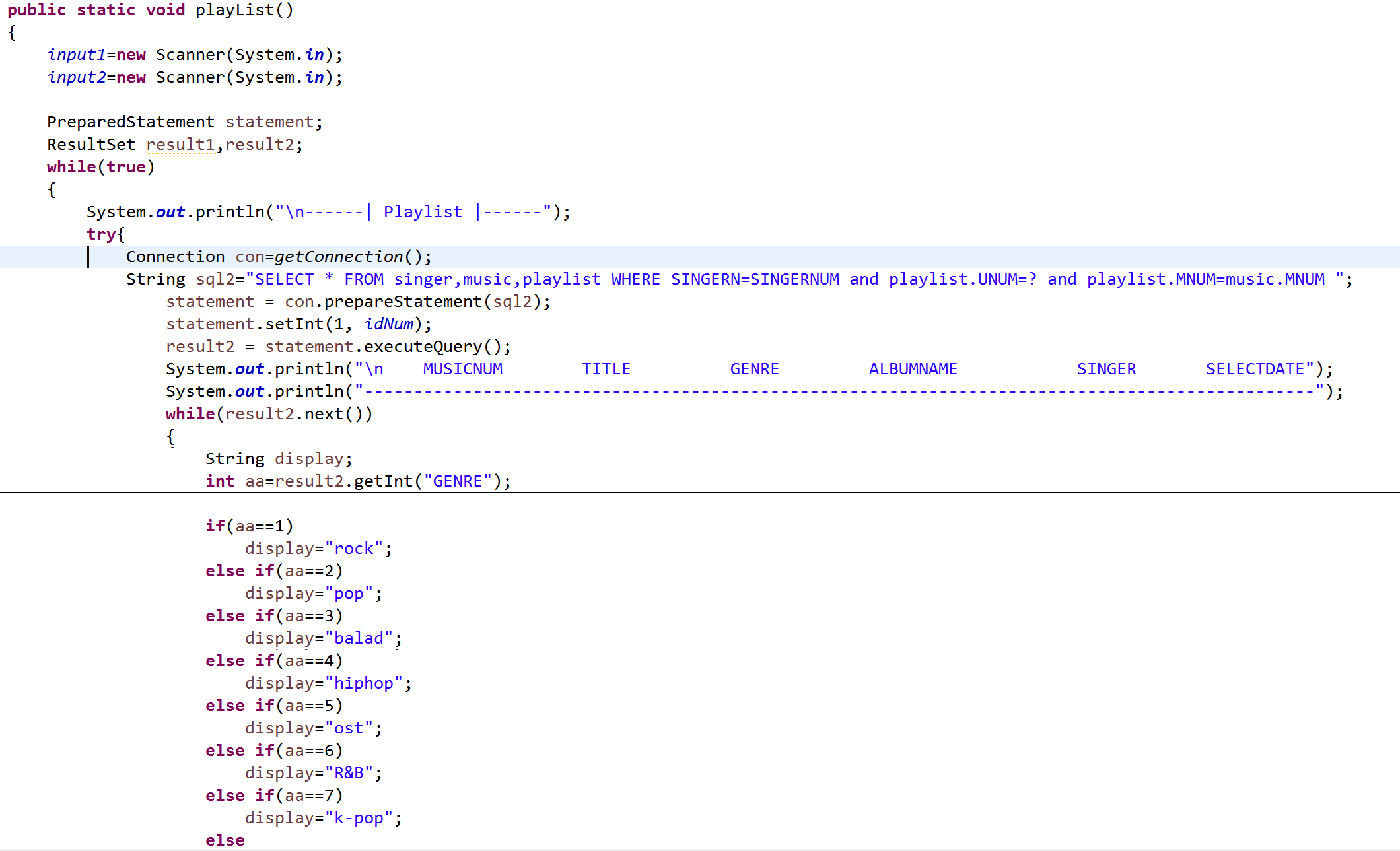


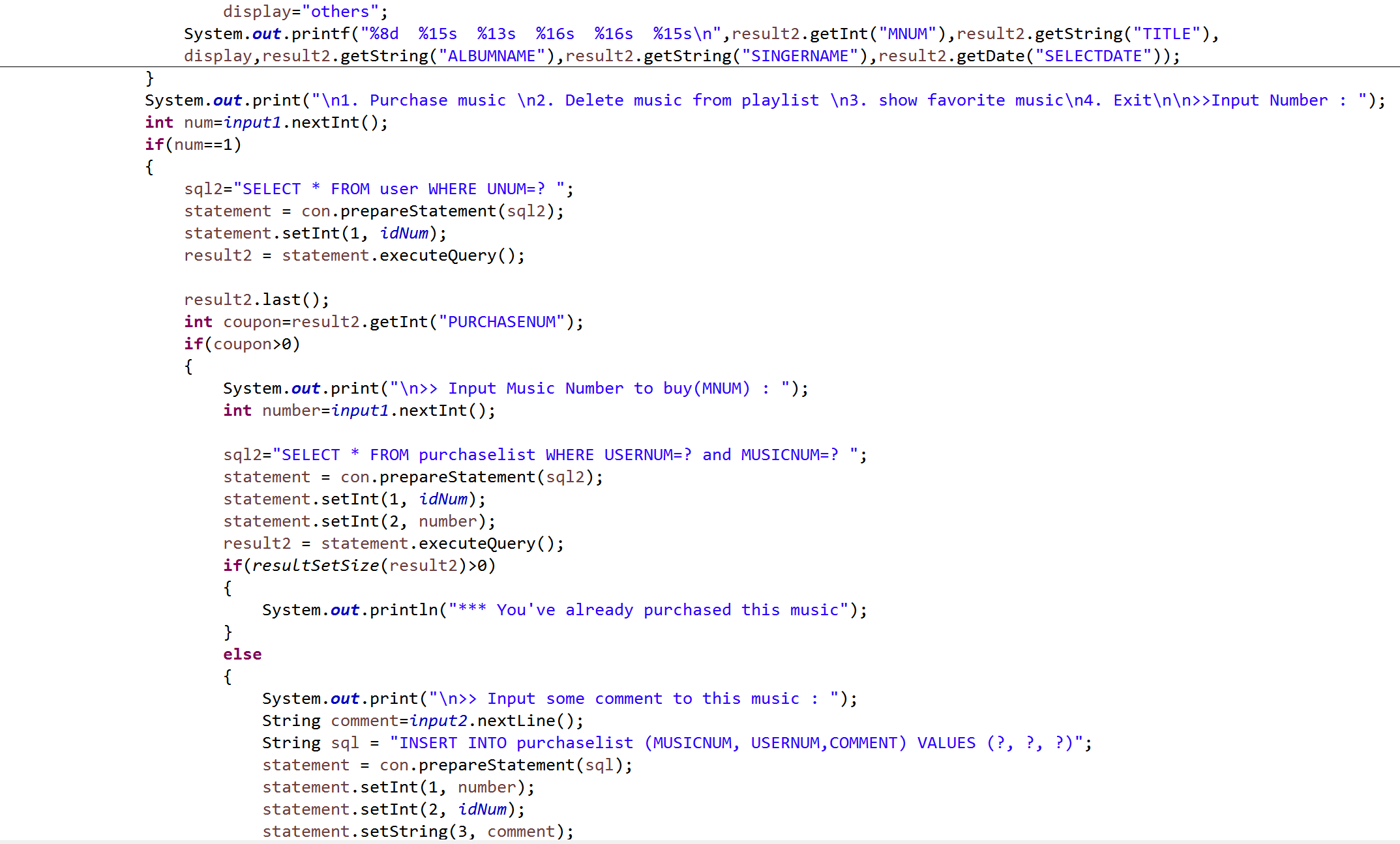
쿠폰 개수를 업데이트 하기 위해서 원래 값을 가져 오기 위해서 SELECT를 이용했다.

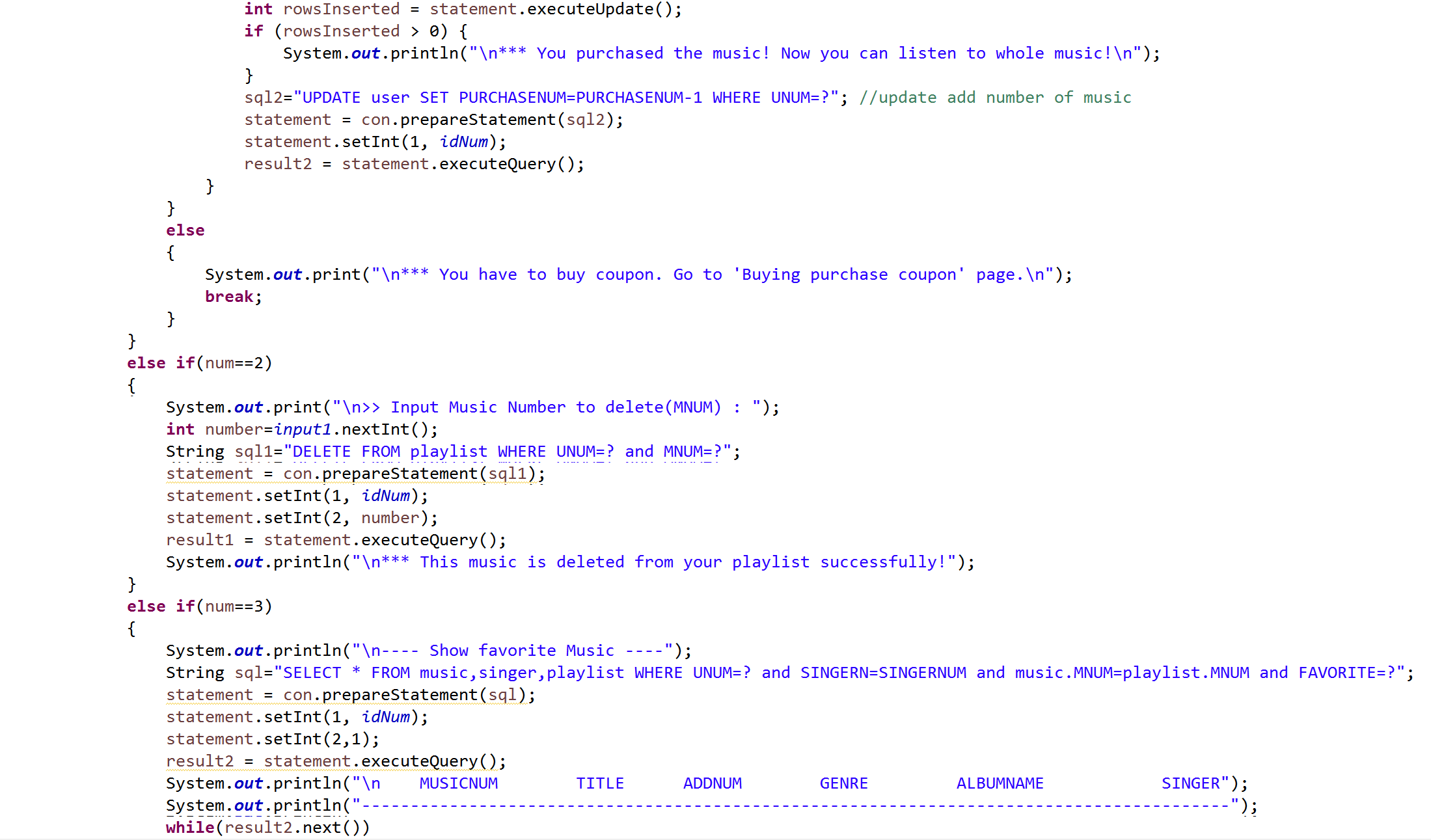


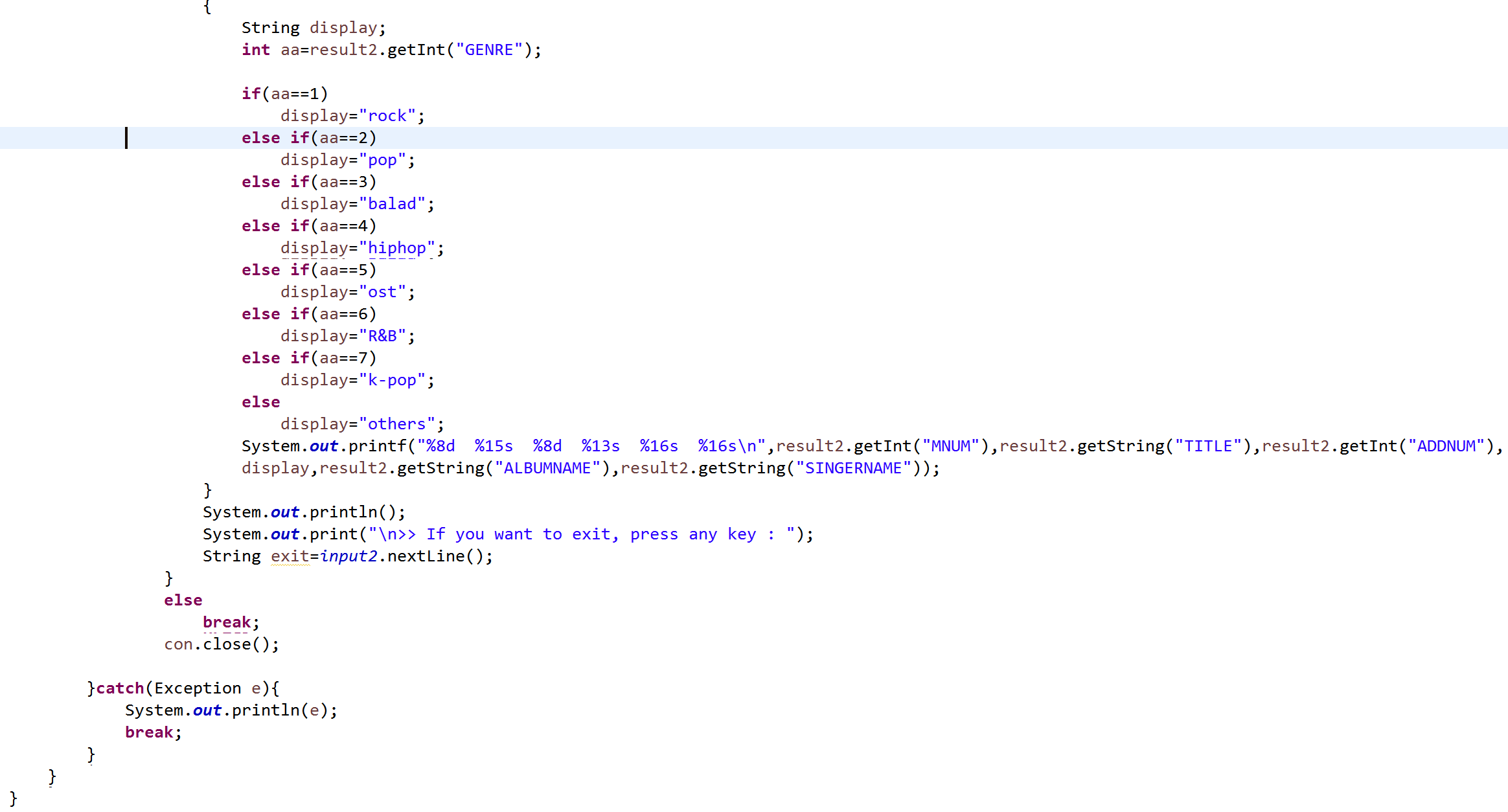
구매한 쿠폰의 개수만큼 더해서 UPDATE했다.

**12) playList**

****

****

****

****

playList 메소드에서는 먼저 로그인 상태의 사용자가 갖는 플레이리스트 목록을 나열하였으며

1. 곡 구매

2. 플레이리스트에서 곡 삭제

3. favorite으로 설정한 곡 조회

4. 종료

로 이루어져 있다.

만약 쿠폰의 수가 부족한 경우에는 쿠폰을 구매하는 페이지로 이동하라고 지시하였으며

쿠폰 개수가 충분한 경우에는 곡 구매 페이지로 넘어갔다.

곡 구매는 구매하고자 하는 음악 번호를 통해 이뤄지며 곡을 이미 구매한 것을 또 구매하지 못하도록 코드를 구현하였고 플레이리스트 중에서 아직 구매하지 않은 곡을 구매할 수 있도록 하였다.

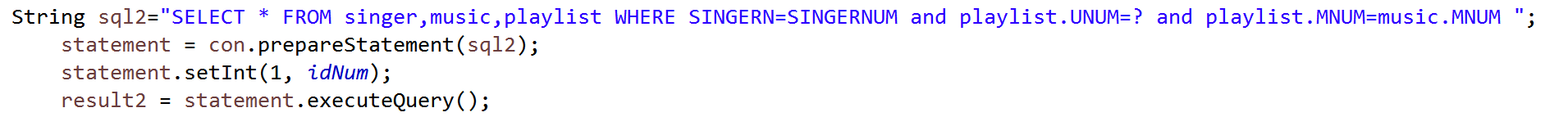
곡을 삭제하는 것은 플레이리스트에서 지우려는 음악의 번호를 입력하면 playlist테이블에서 지운다.

플레이리스트에 곡을 추가할 때 각 사용자마다 그 곡을 favorite으로 설정할 수 있었는데

플레이리스트 곡 중에서 favorite으로 설정한 곡들만 모아서 조회할 수 있는 것이 3번이다.

그리고 이외의 경우는 종료를 통해 이전 페이지로 돌아갔다. (break문으로 while문을 빠져나갔다.)

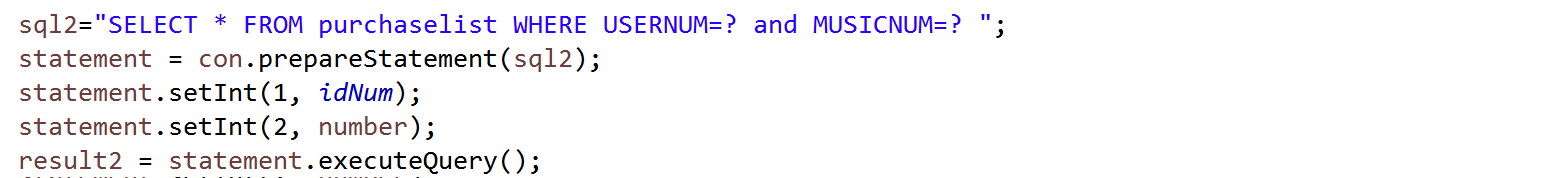
**[SQL문]**

****

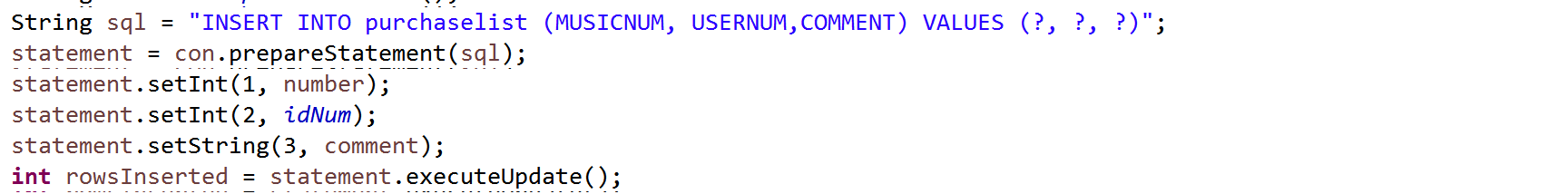
로그인 되어있는 사용자의 플레이리스트를 조회하기 위해 singer, music, playlist를 엮어서 WHERE문을 통해 조건을 명시하여 걸러낸다.

****

구매할 수 있는 쿠폰이 존재해야 하므로 UNUM이 idNUM과 같은 데이터의 PURCHASENUM의 개수가 0보다 큰 지 확인해야 한다. (SELECT문 사용)

****

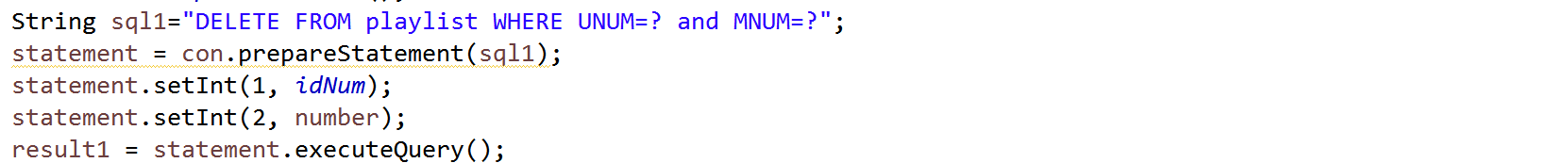
중복된 노래를 구매하지 않기 위해 해당 데이터가 이미 purchaselist에 존재하는지 확인할 수 있는 sql문을 사용한다.

****

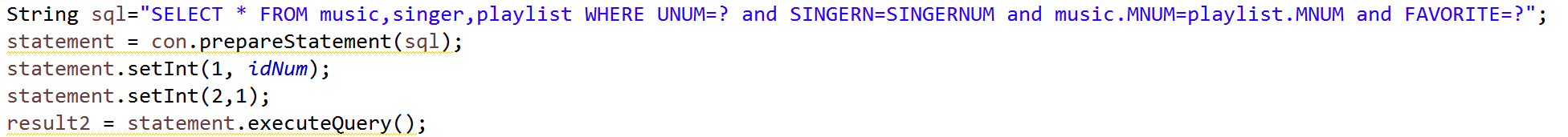
중복되는 데이터가 없으면 해당 데이터를 purchaselist에 넣는다.

****

음악을 구매하면 PURCHASENUM을 1감소시켜야 하므로 UPDATE문을 통해 정보를 변경한다.

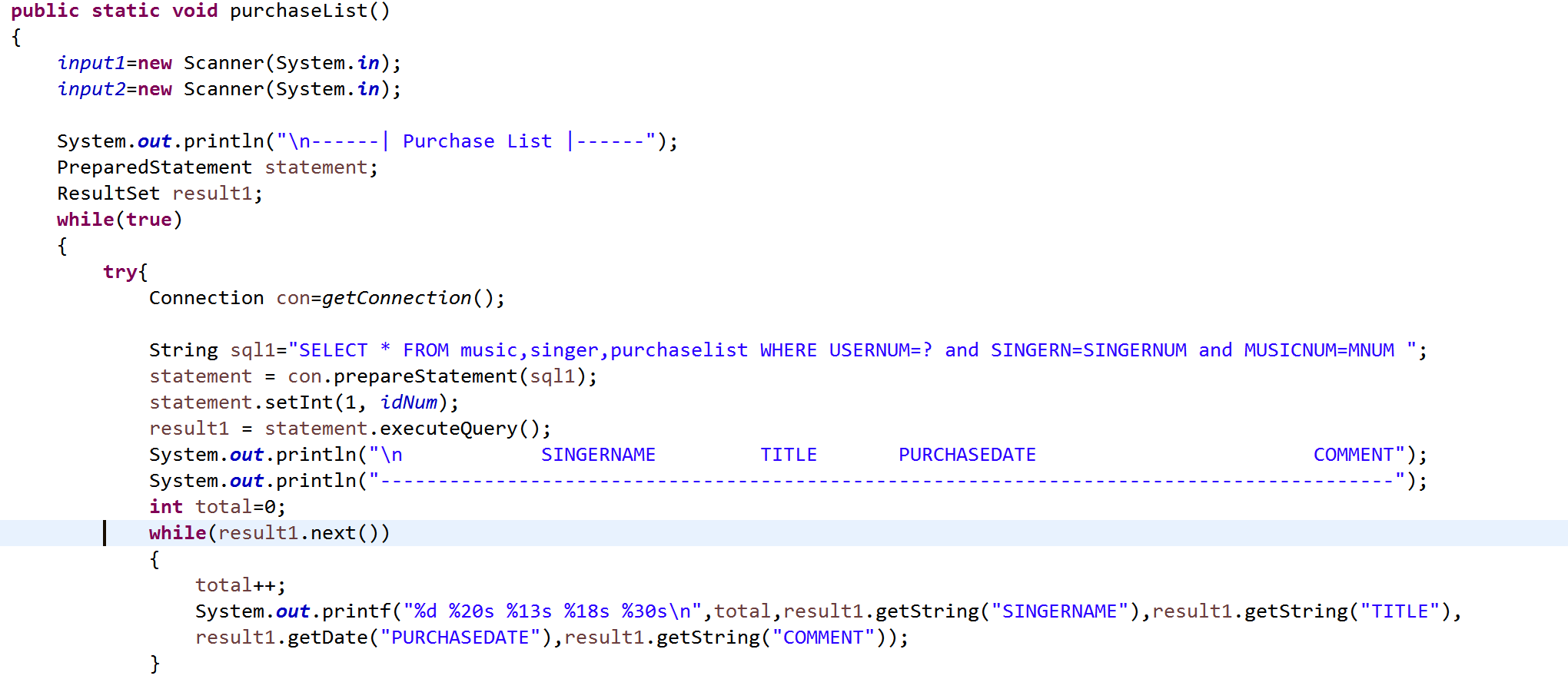
****

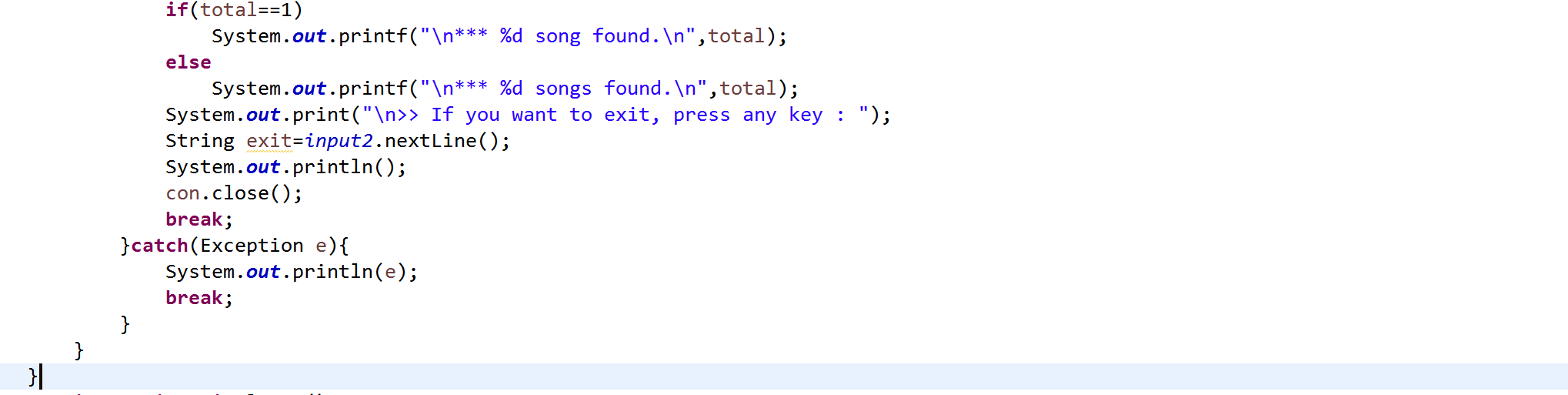
플레이리스트에서 곡을 삭제하고 싶을 때에는 해당 데이터의 조건을 명시하여 DELETE문을 사용한다.

****

Favorite 음악으로 설정된 음악만을 보여주기 위한 페이지에서는 사용자의 플레이리스트에서 FAVORITE=1로 설정된 데이터를 골라낸다. 다만 이때 SINGERN=SINGERNUM이고 music.MNUM=playlist.MNUM으로 설정한다.

**13) purchaseList**



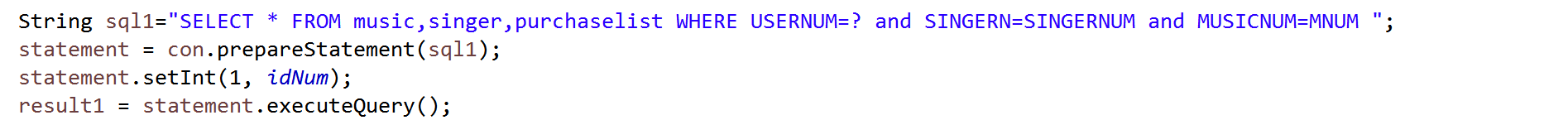


플레이리스트 목록의 곡 중에서 구매한 곡을 모아놓은 리스트이다.

곡을 구매하면 곡을 처음부터 끝까지 들을 수 있으며

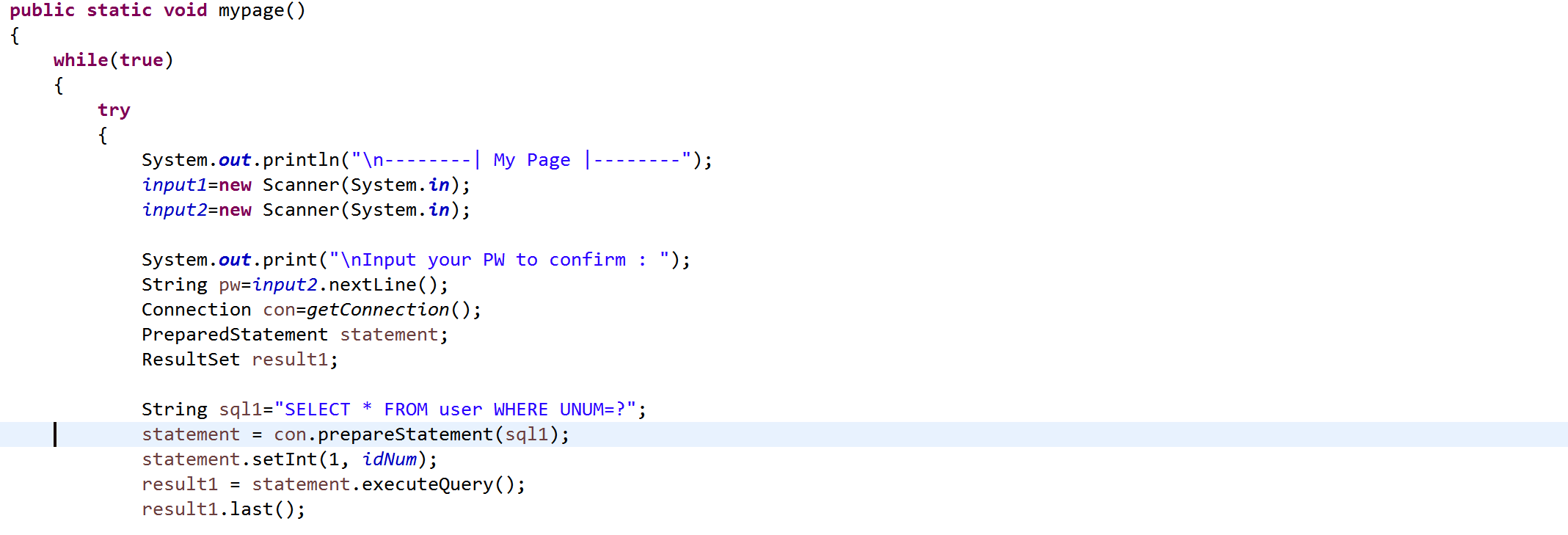
곡을 구매할 때마다 comment를 입력할 수 있어서 나에게 어떠한 의미를 지니는 곡인지 어떠한 이유로 이 곡을 구매하였는지 등을 기록할 수 있다.

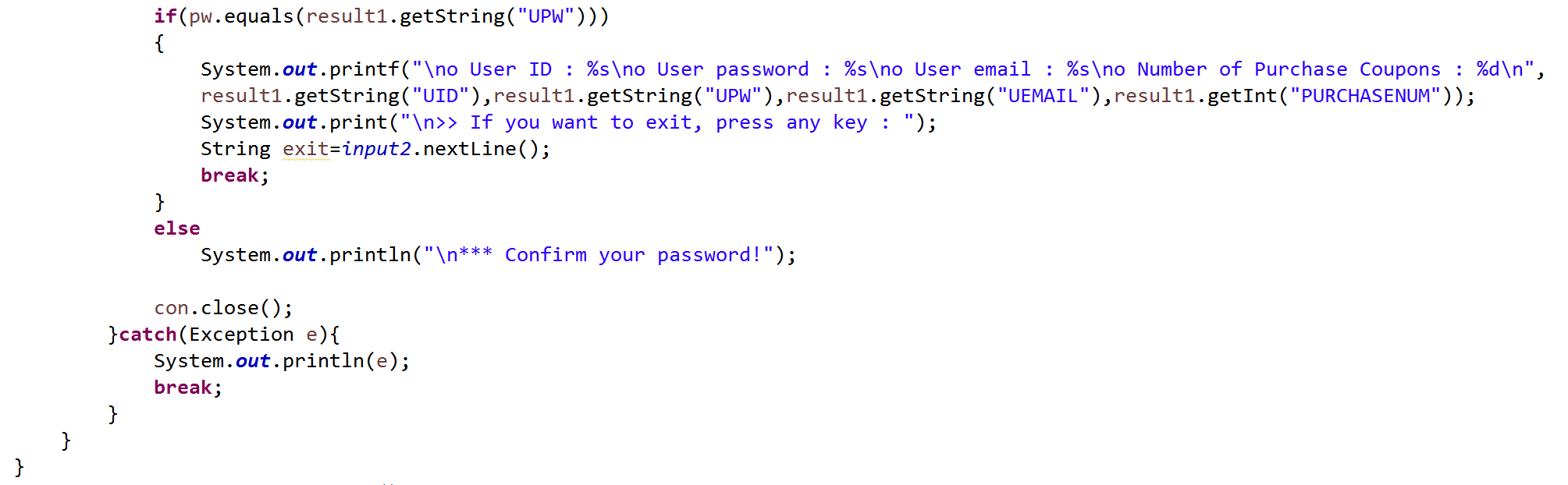
**[SQL문]**

****

구매한 음악을 나열하기 위한 sql문이다.

**14) myPage**



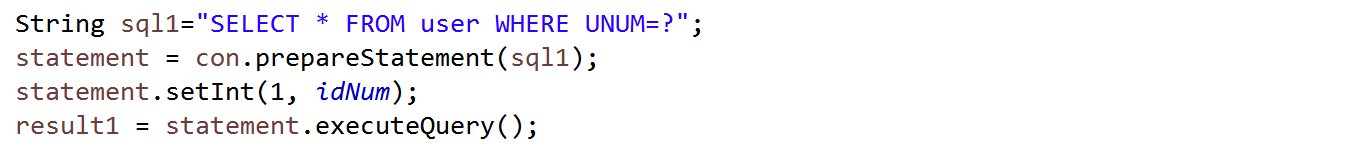


내 개인 정보를 조회할 수 있는 페이지로

비밀번호를 입력한 후에 현재 로그인 되어있는 사용자의 비밀번호와 일치할 경우에만 개인 정보를 조회할 수 있으며

아이디, 비밀번호, 이메일, 남아있는 쿠폰을 개수(=구매할 수 있는 곡의 개수)를 확인할 수 있다.

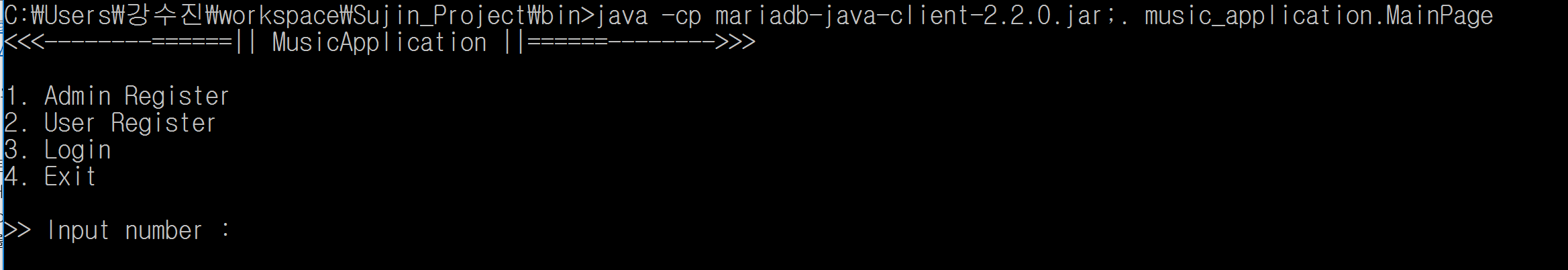
**[SQL문]**



마이페이지에서 사용자의 정보를 조회하기 위함으로 user테이블에서 UNUM=idNum인 데이터를 가져온다.

**2. 수행되는 기능 스크린 샷**

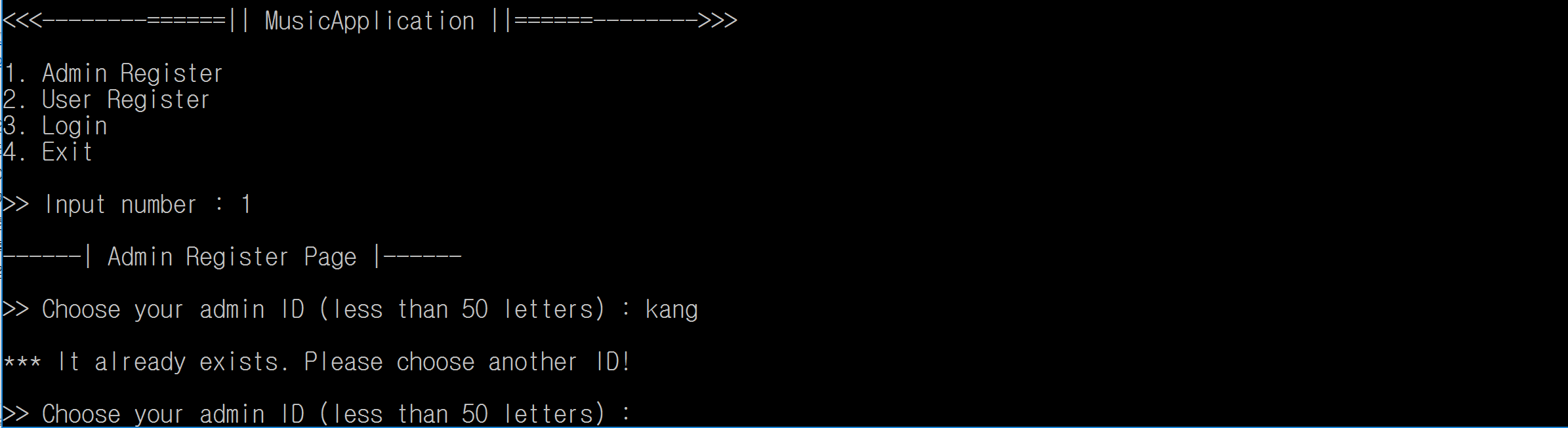
<처음 화면>



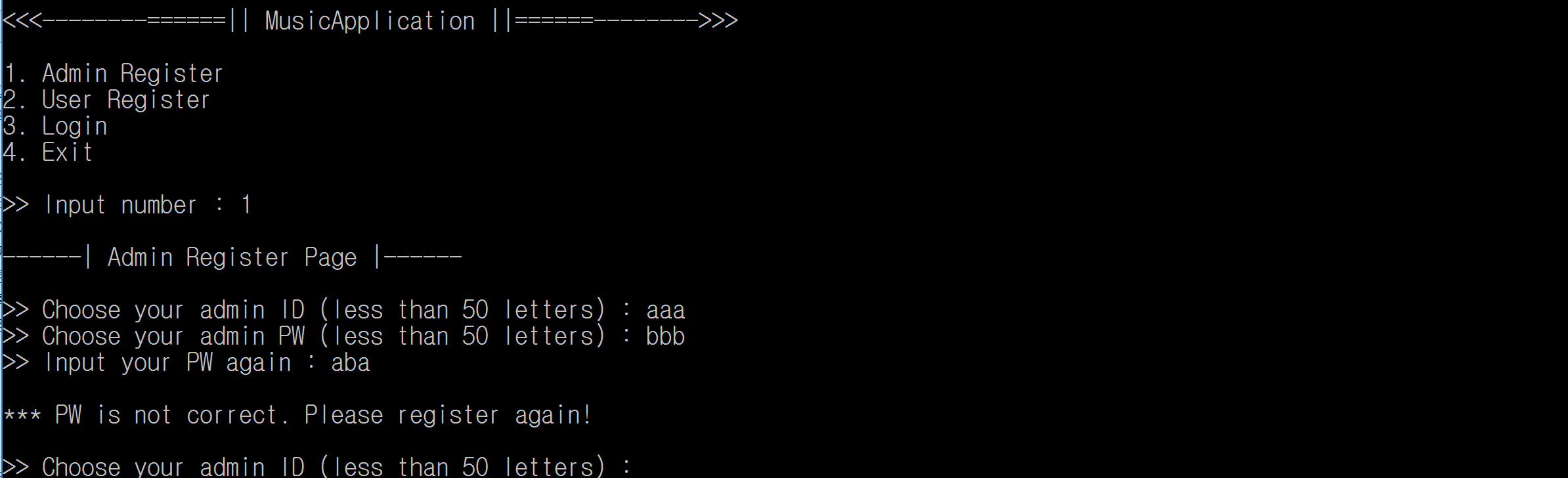
<관리자 등록>

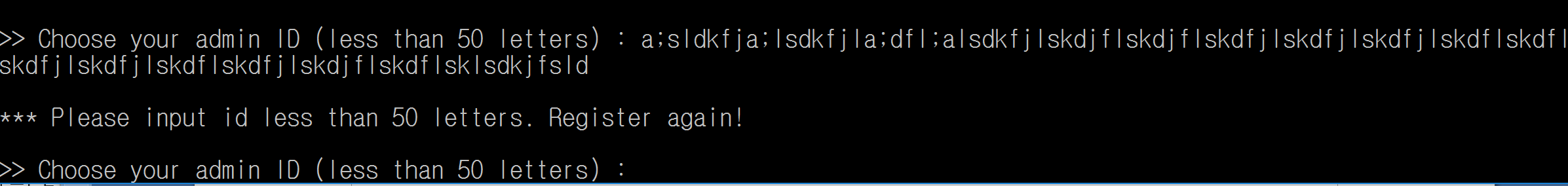


새롭게 관리자를 등록할 수 있으며 만약 이미 입력한 아이디가 user테이블이나 admin테이블에 존재할 경우에는 아래와 같다.



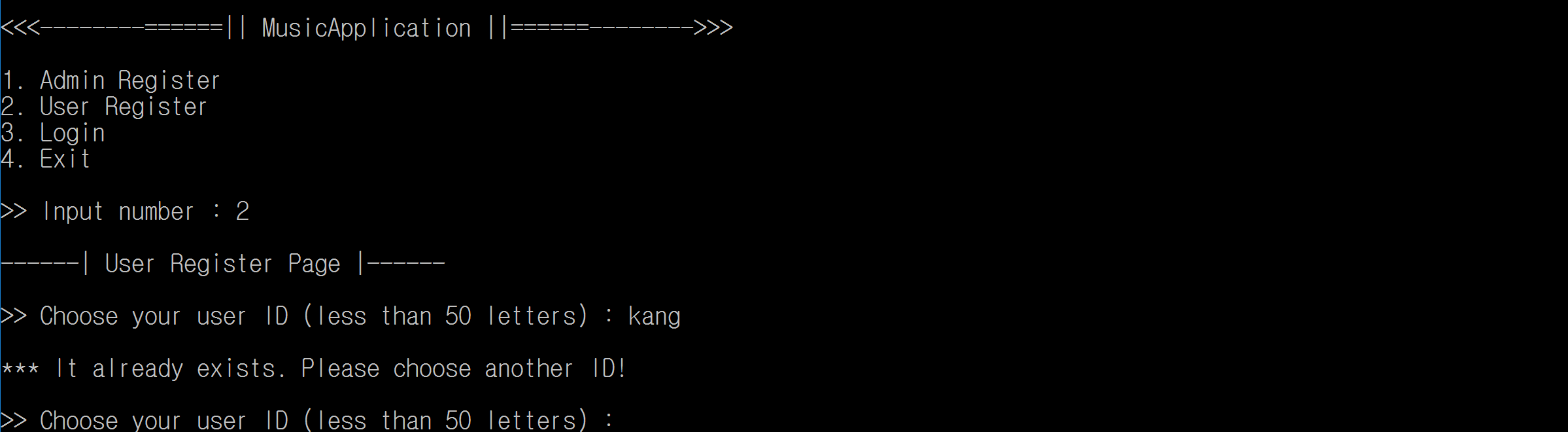
관리자와 사용자 모두 로그인은 같은 공간에서 진행하므로 모두 아이디가 겹쳐서는 안 된다. 위 예제에서는 kang이라는 아이디가 user테이블에 존재하기 때문에 다른 아이디를 설정하라고 지시하고 있다.

만약 비밀번호를 다시 입력하라고 했을 때 일치하지 않은 비밀번호를 입력한 경우에는 다시 등록하라고 알림 메세지가 뜬다.

50글자 넘게 입력했을 경우 역시 다시 등록하라는 메시지가 뜬다.

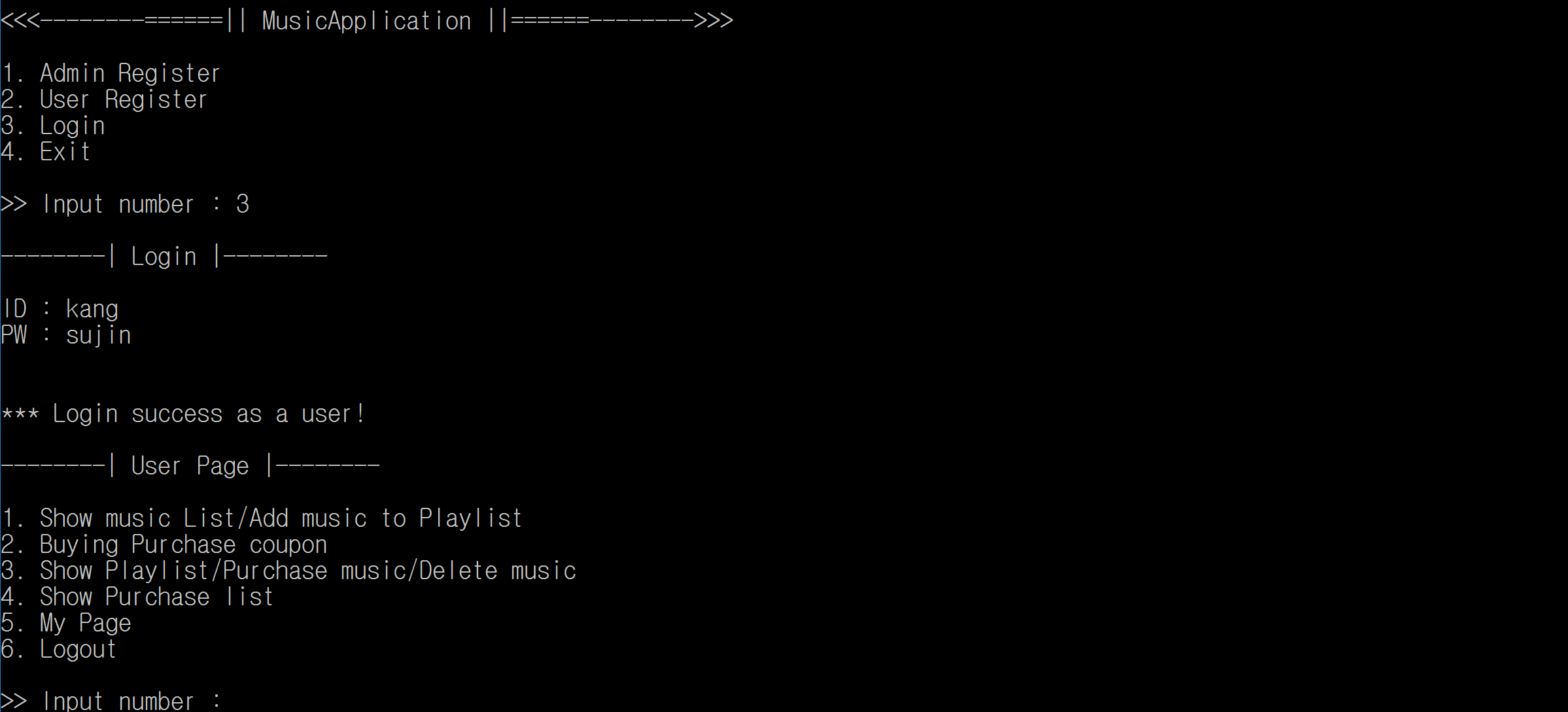
<사용자 등록>

 사용자 등록 역시 user테이블과 admin테이블에 존재하지 않는 아이디를 등록할 수 있으며 이미 존재하는 아이디를 입력했을 경우에는 아래와 같다.

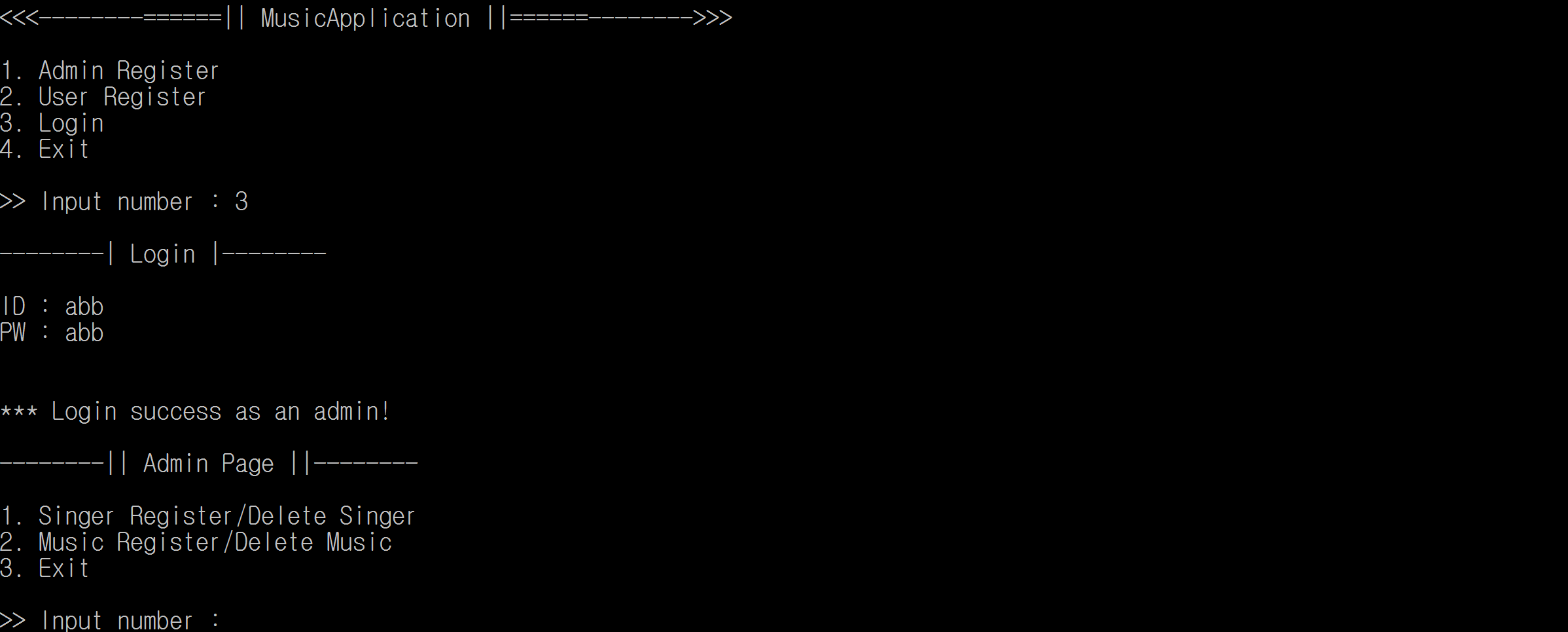


이 외에도 비밀번호를 다시 입력할 때 일치하지 않게 입력하면 경고 메시지를 제시하며 50글자가 넘을 때에도 관리자 등록 페이지에서와 동일하게 경고 메시지를 띄운다.

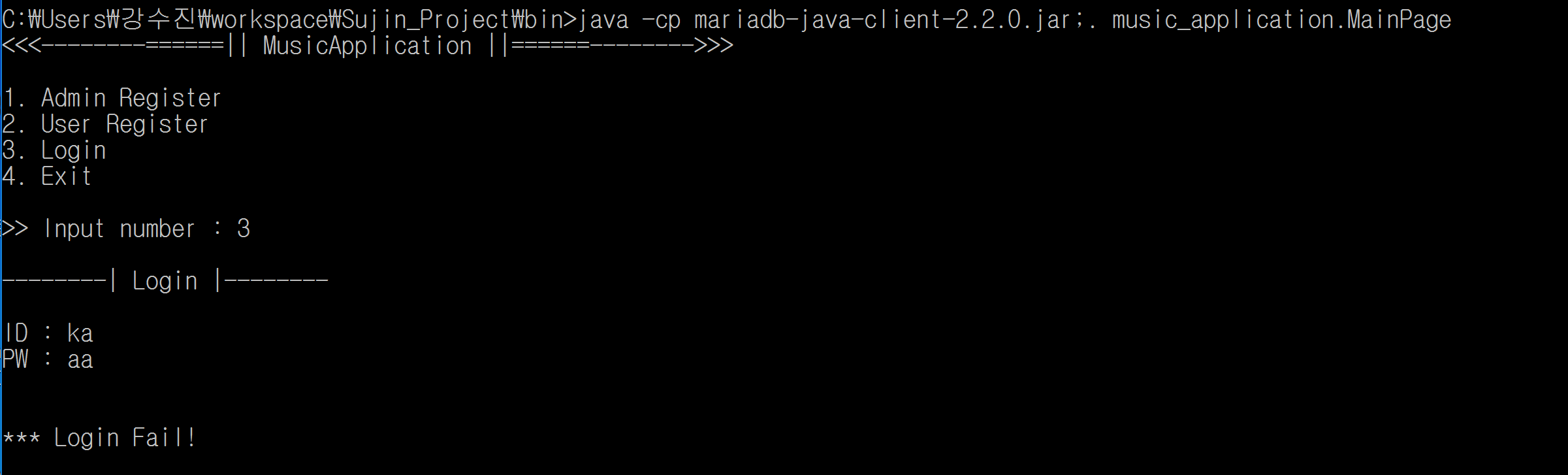
<로그인>



사용자로 로그인한 경우에는 이렇게 메시지가 뜬다.

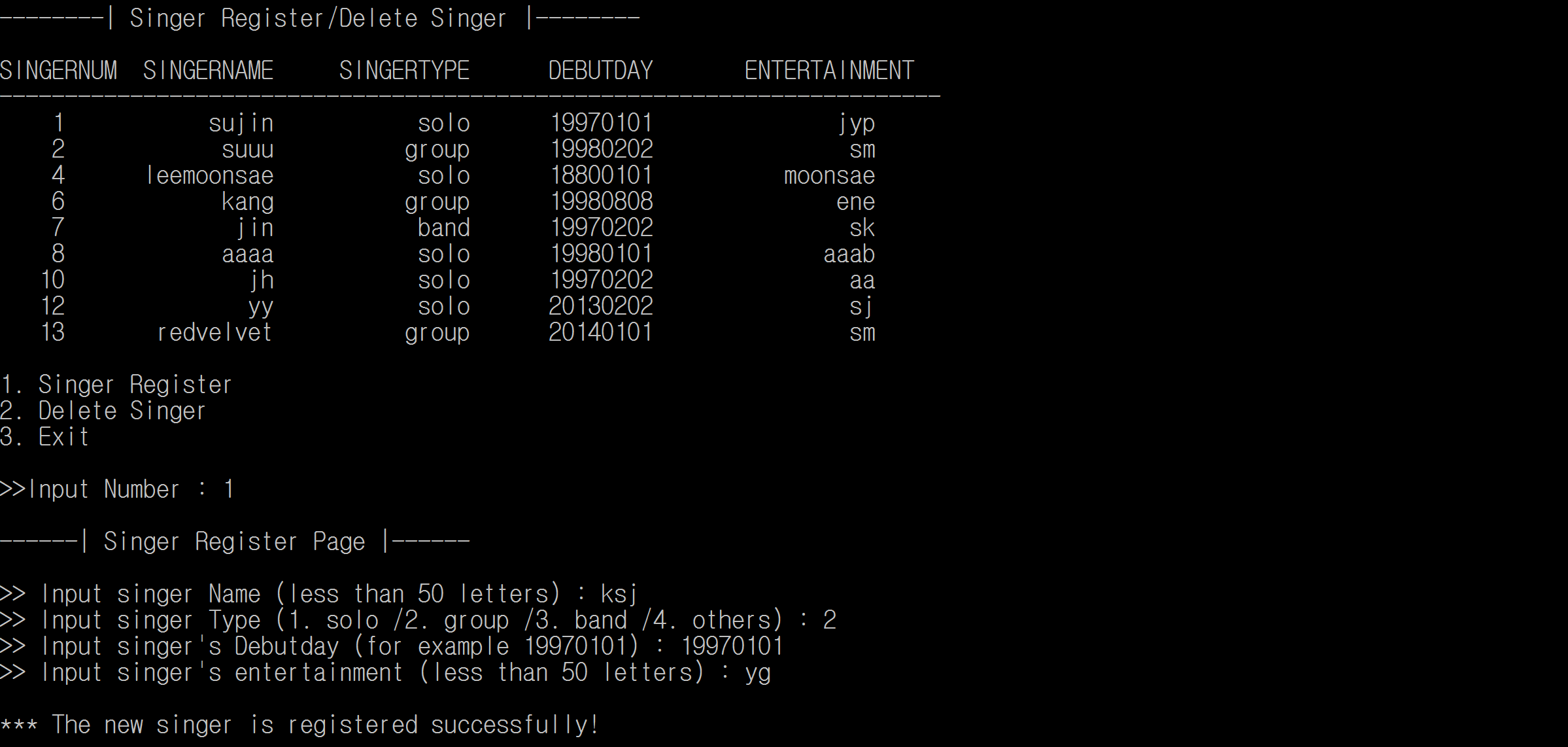


관리자로 로그인했을 때에는 이렇게 메시지가 뜬다.

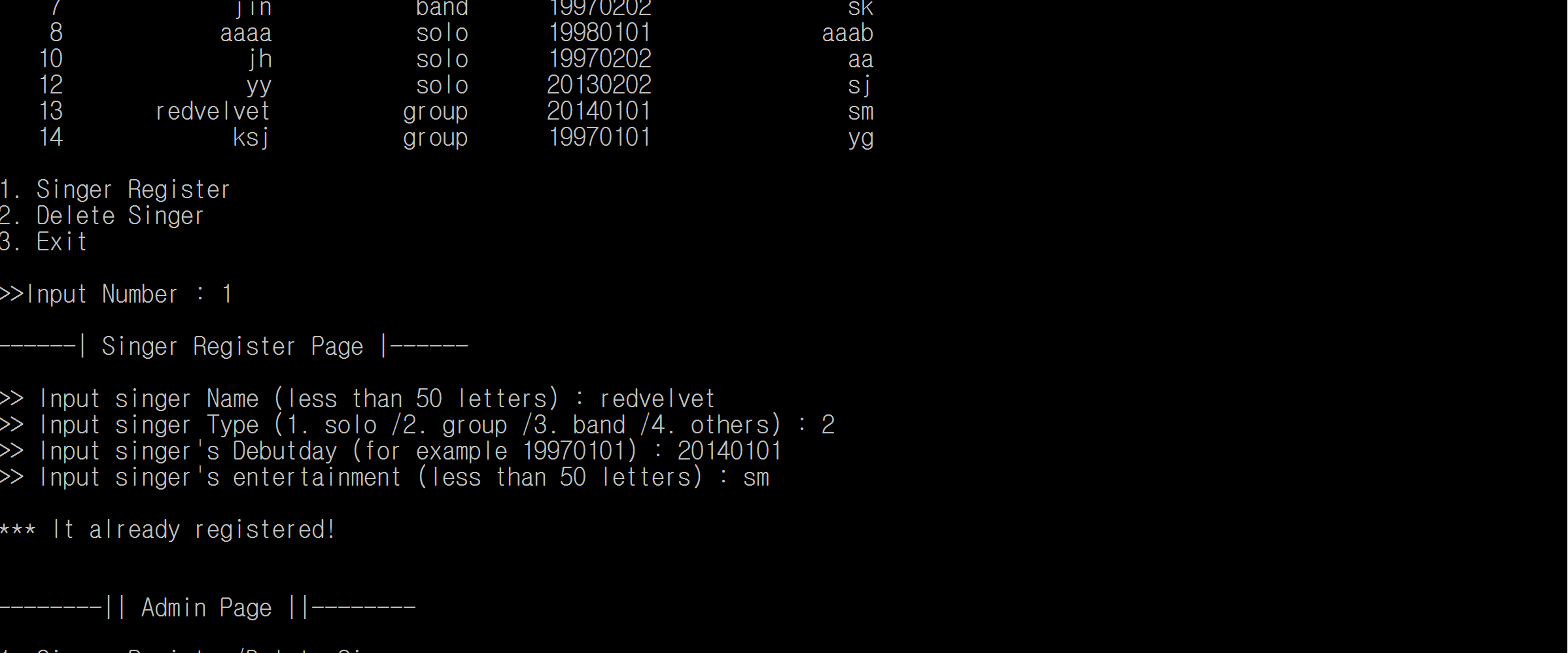


로그인 실패했을 때 메시지

<가수 등록>

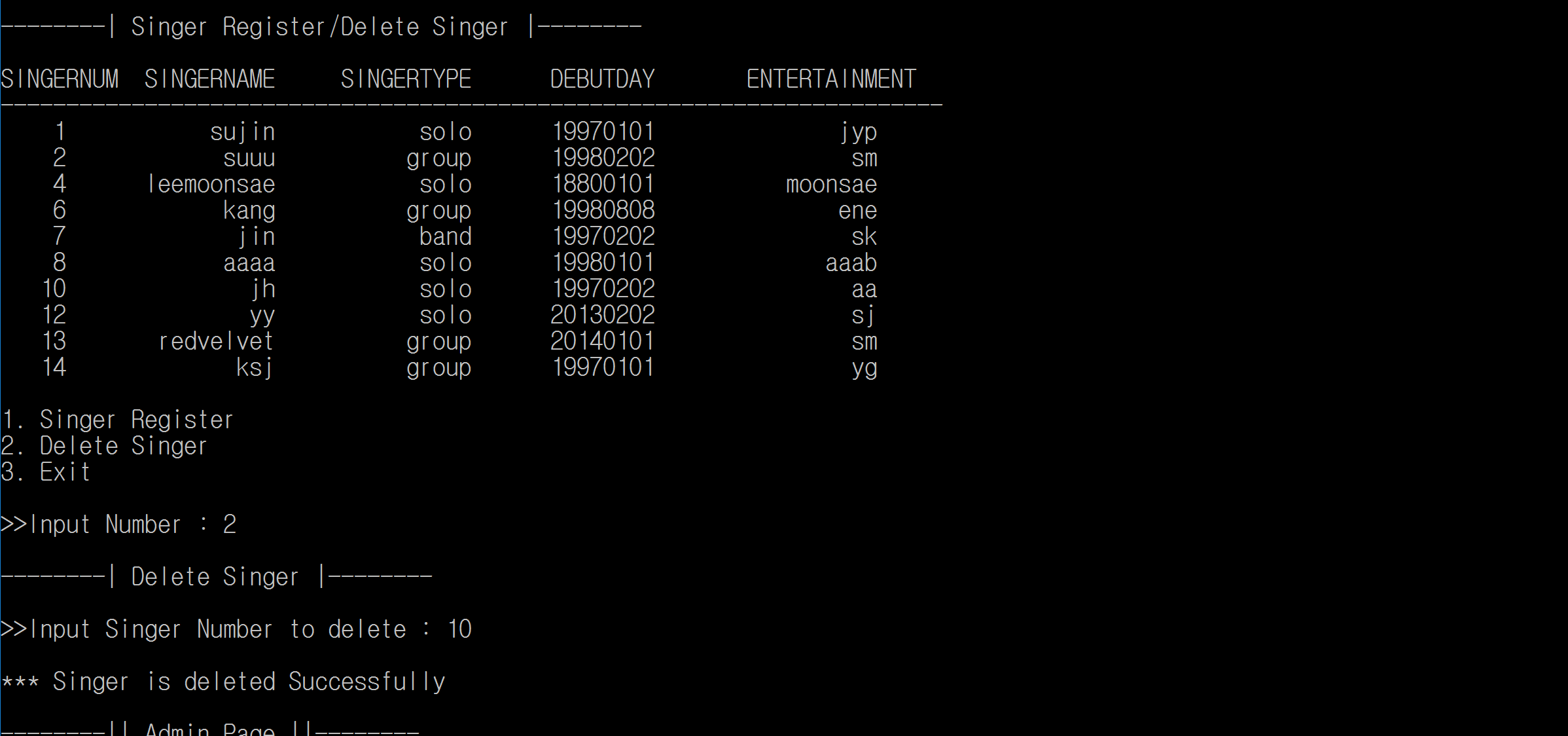


관리자의 경우 가수를 등록할 수 있으며 위와 같이 가수를 등록한다.



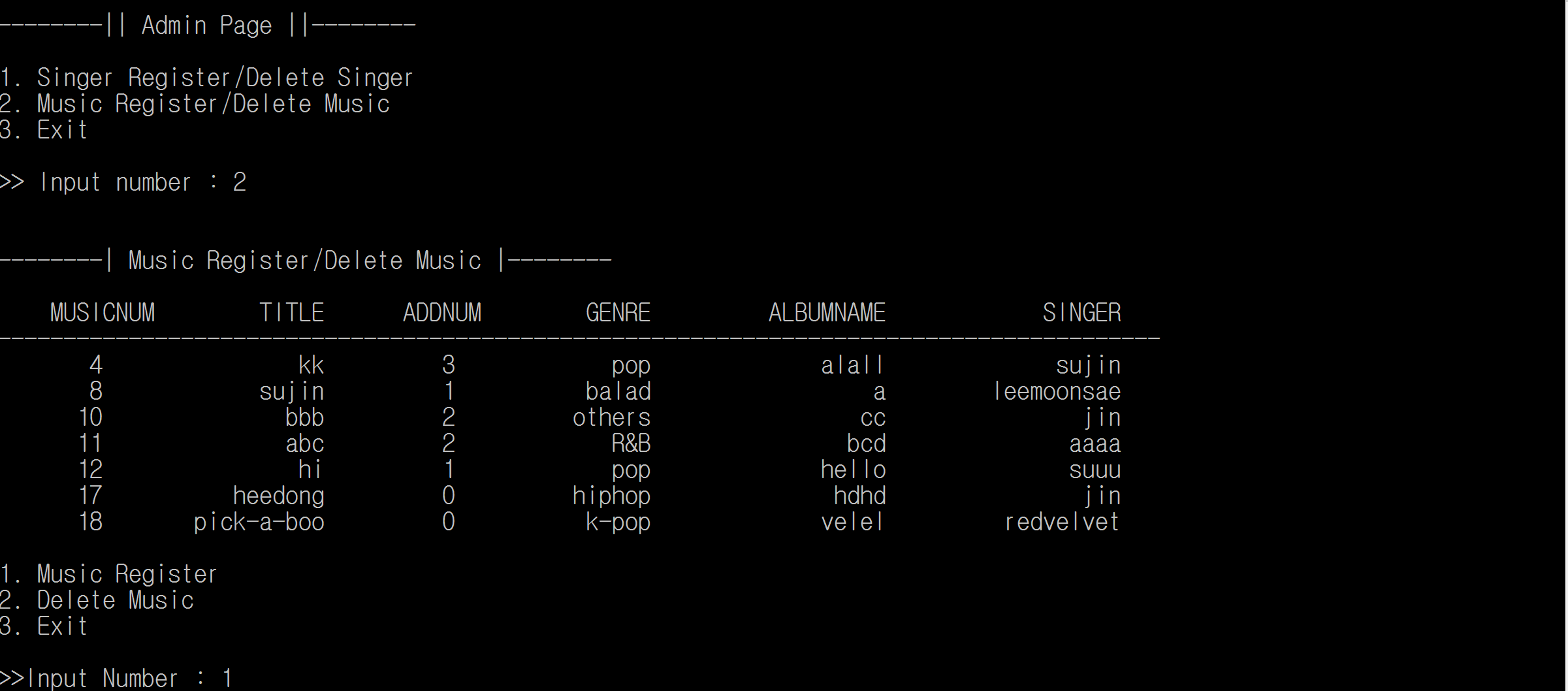
이미 등록된 정보와 동일한 값을 입력하면 이렇게 나타난다.

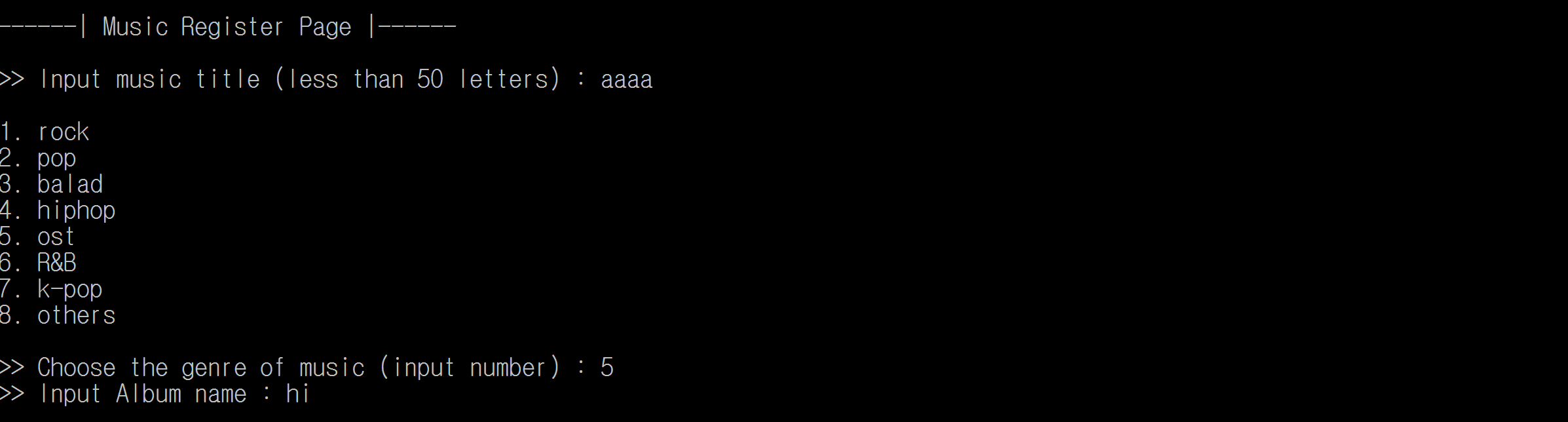
<가수 삭제>

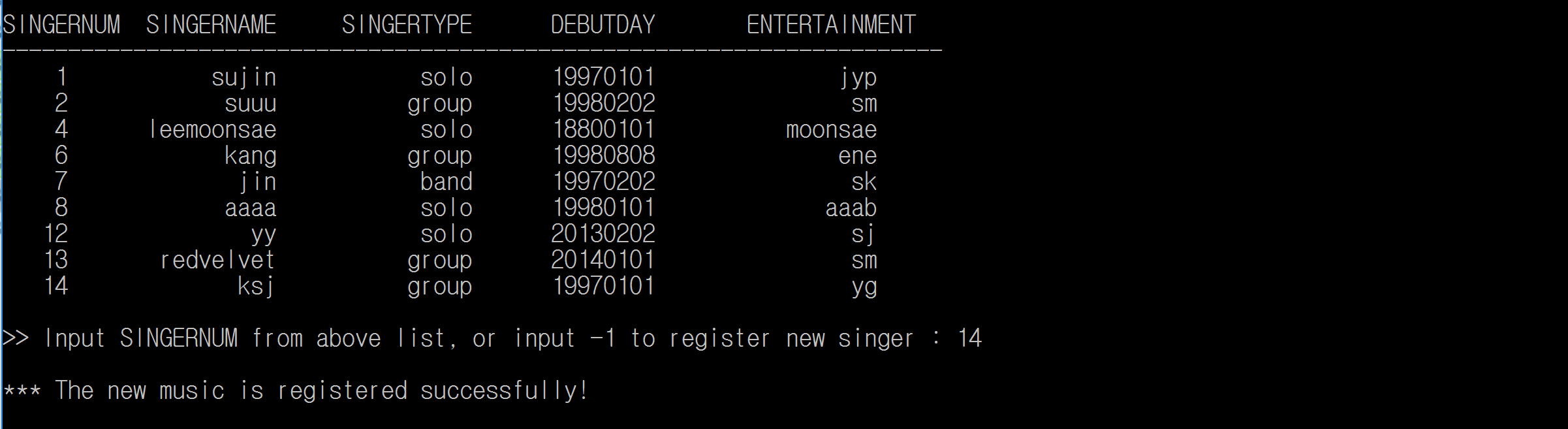


관리자가 가수를 삭제하는 과정이다.

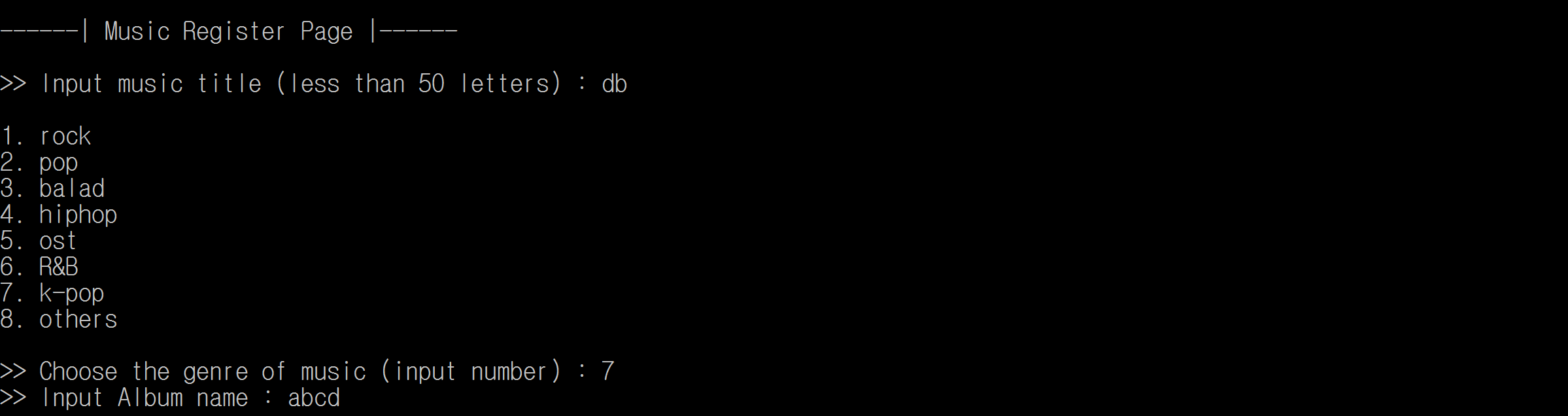
<음악 등록>



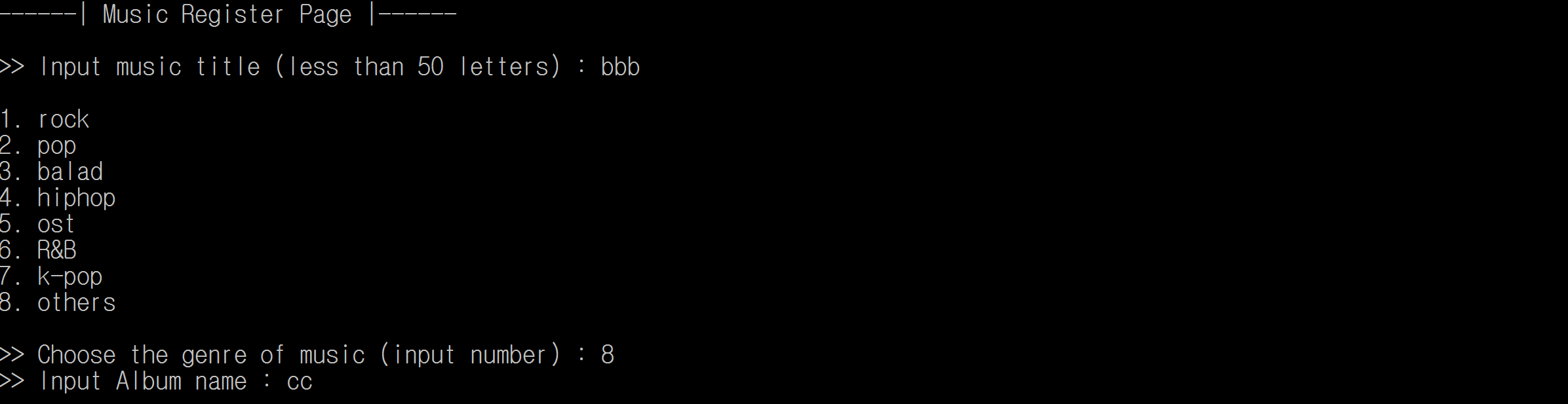


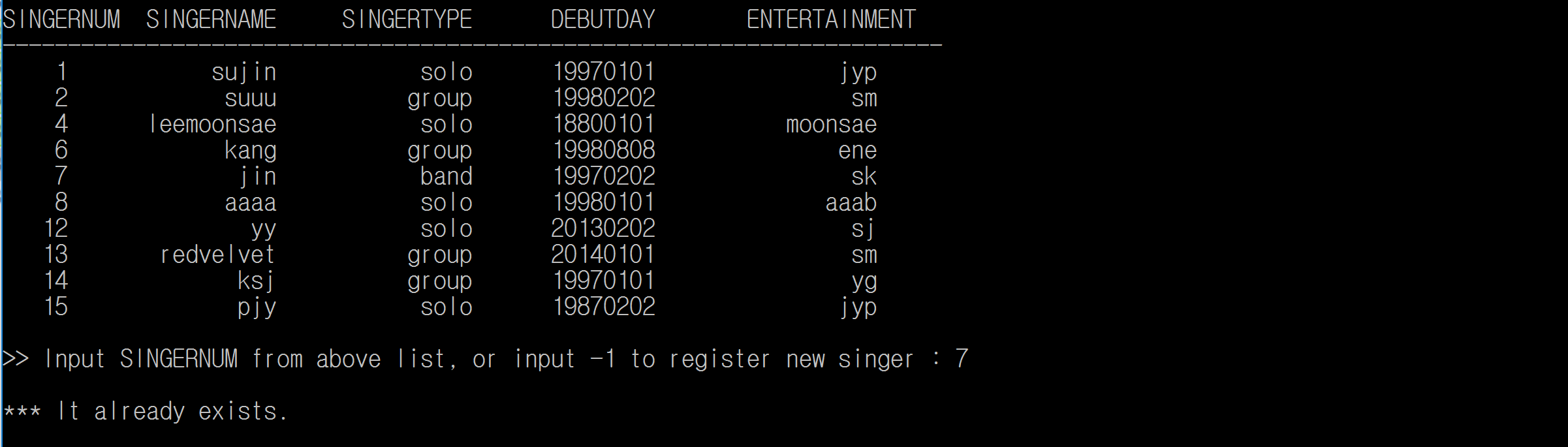


음악을 새로 등록하는 과정이다. 기존 음악 목록에 해당 가수가 존재하면 해당 가수의 번호를 입력하면 된다. 만약 존재하지 않을 경우에는 가수를 새로 등록하는 과정을 거쳐야 하는데 이는 아래와 같다.

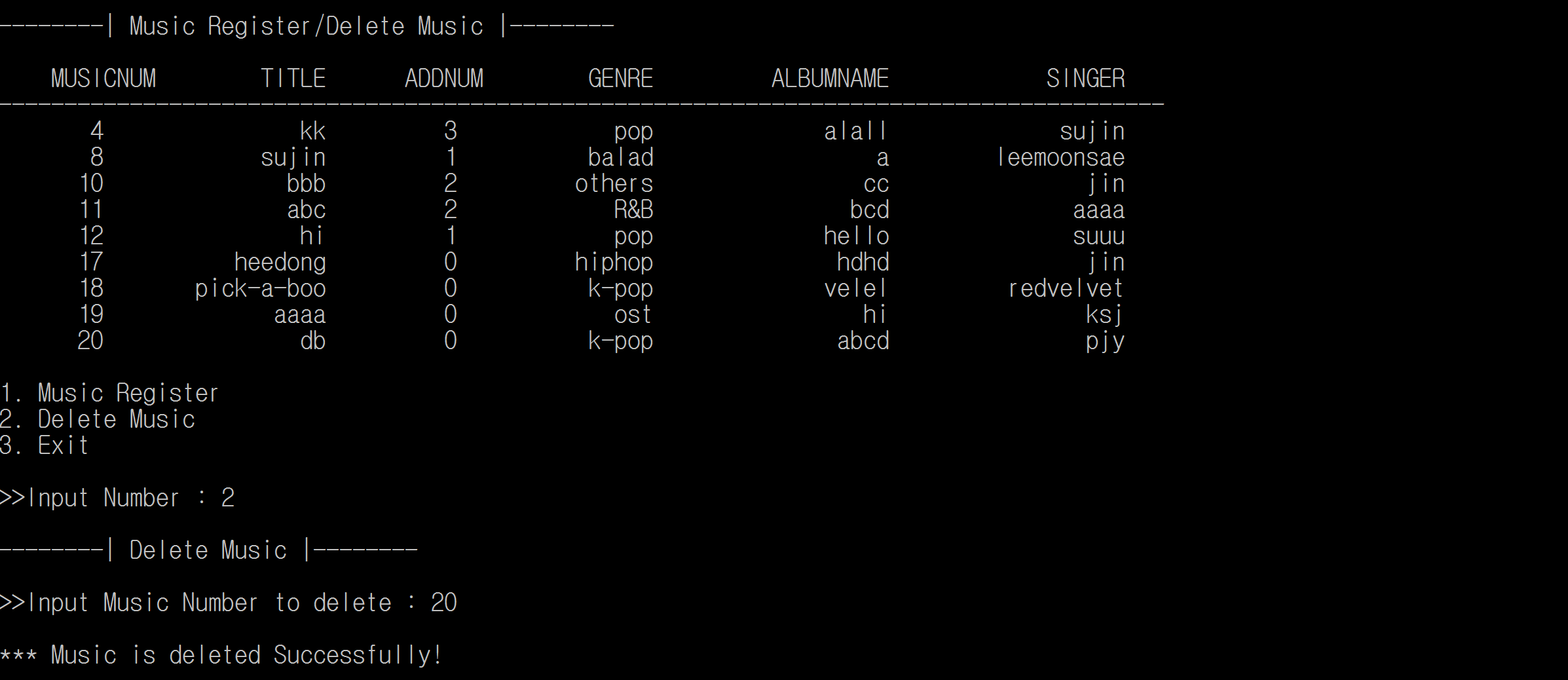


가수 목록에 해당 가수가 존재하지 않으면 가수 등록페이지로 이동해서 다시 가수를 등록한다.



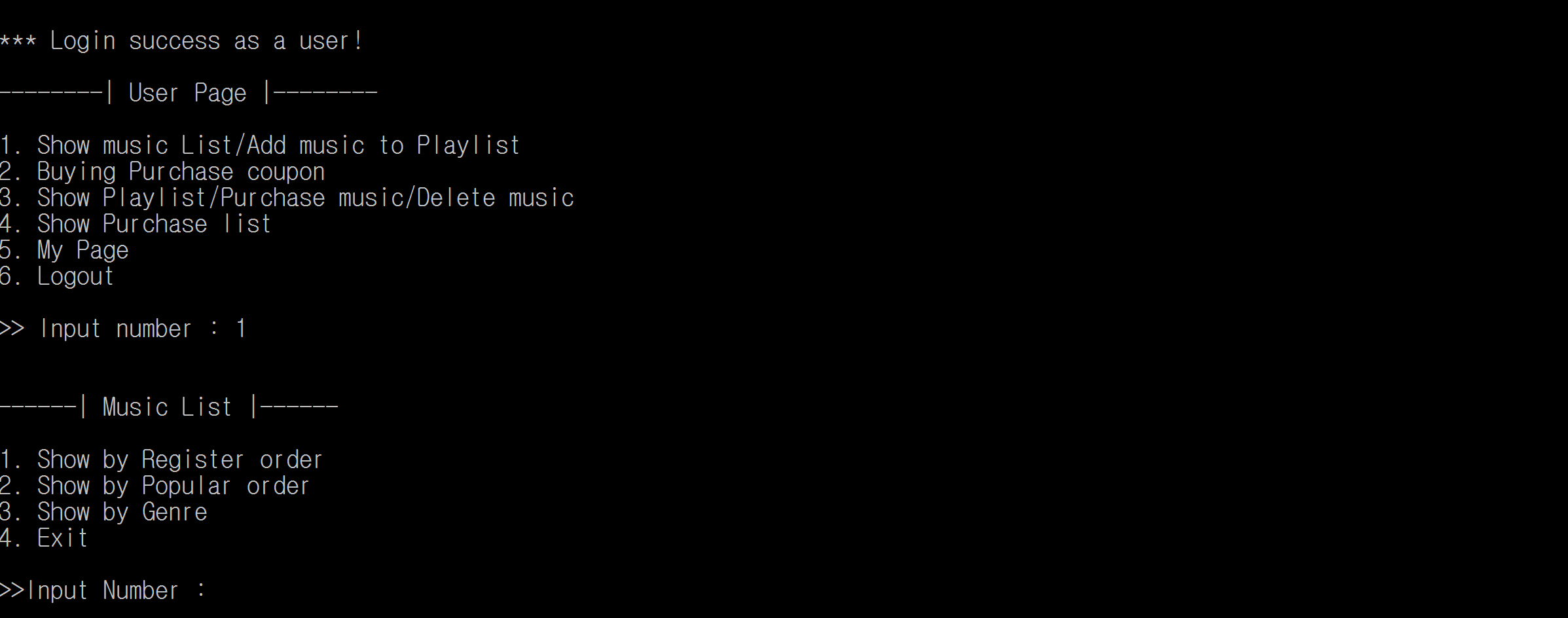
이미 등록된 음악의 경우 이렇게 메시지가 뜬다.

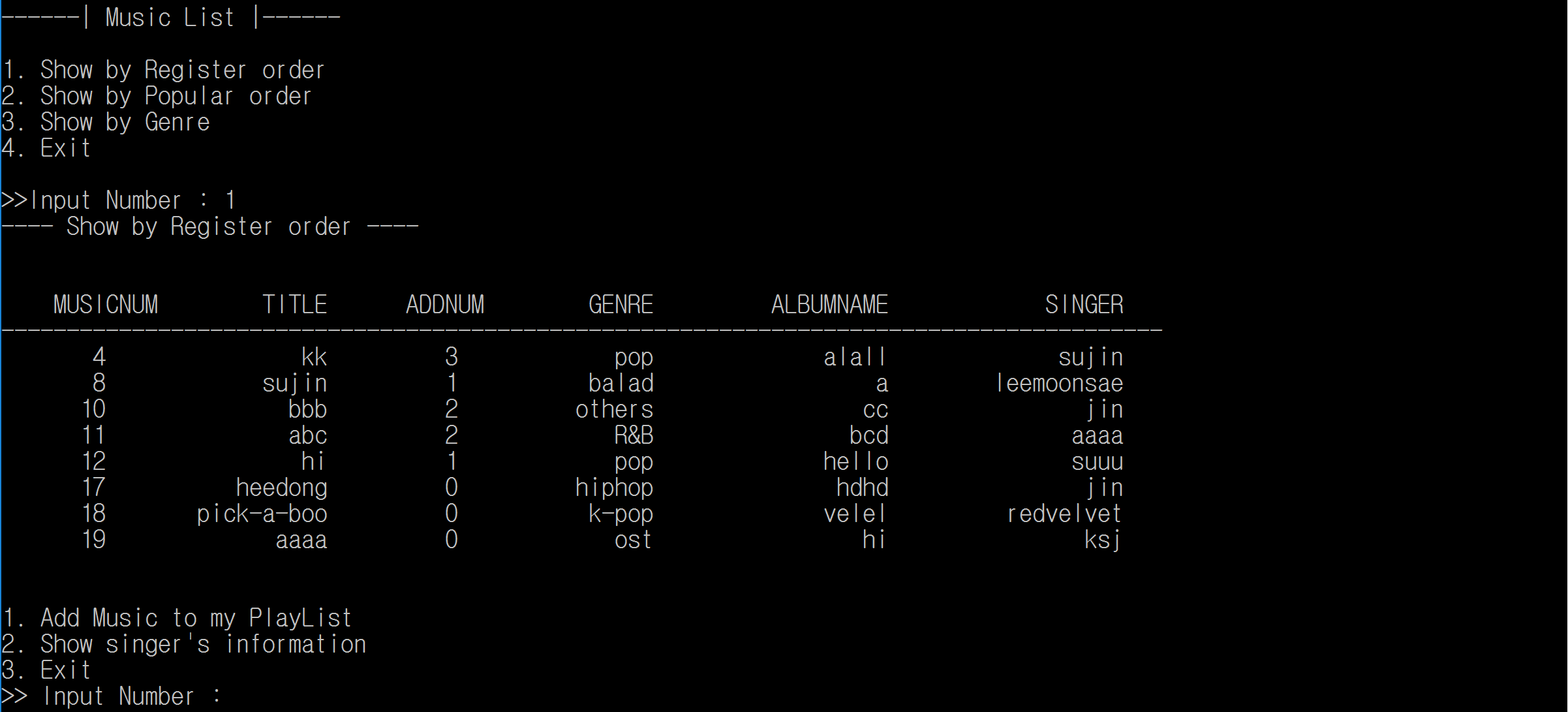
<가수 삭제>



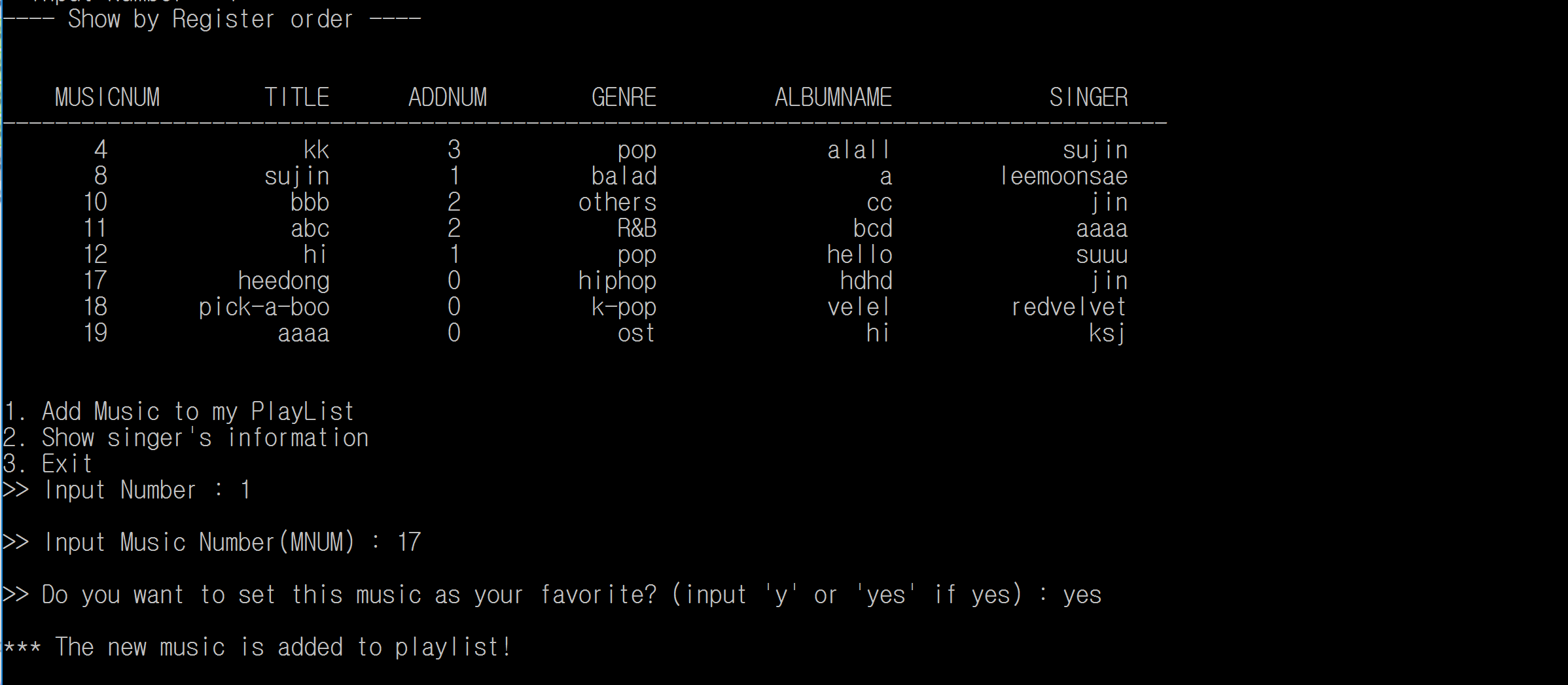
지우려는 곡의 번호를 입력 받아서 곡을 지운다.

<음악 전체 리스트 조회>

 처음 나타나는 화면



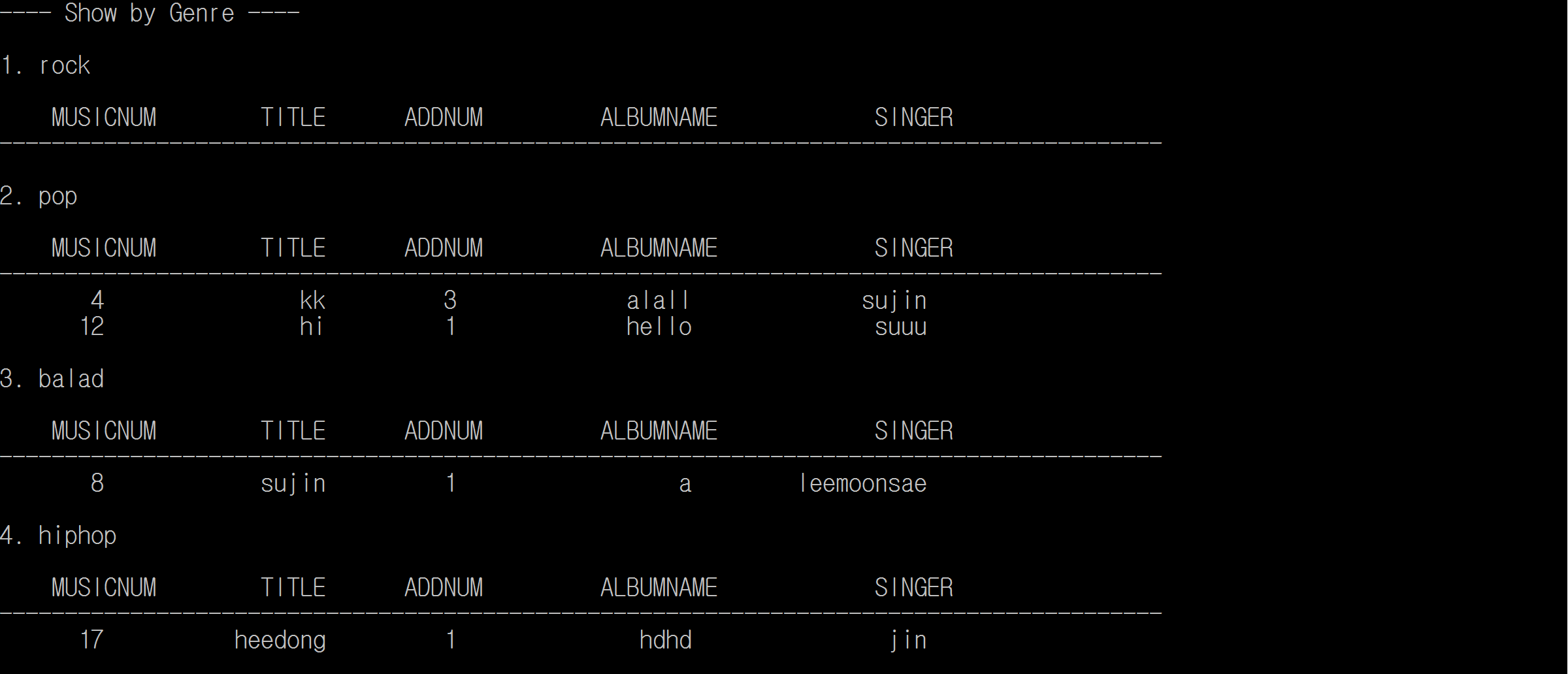
등록순으로 나열할 때

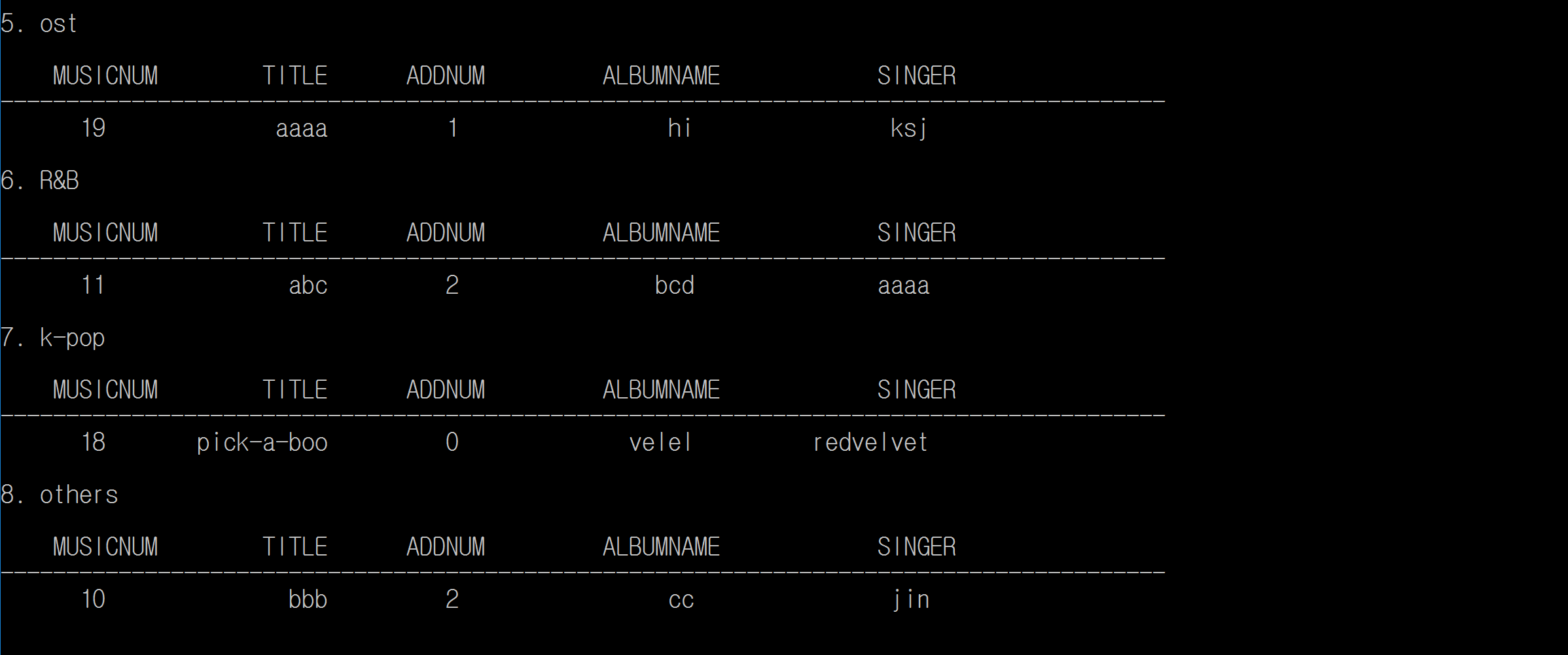


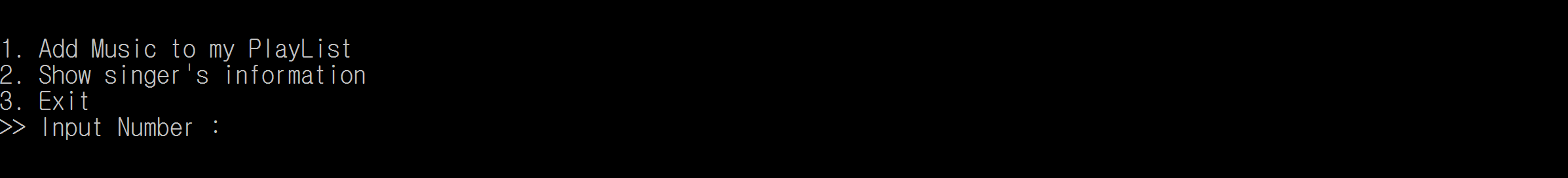
플레이리스트에 곡 추가



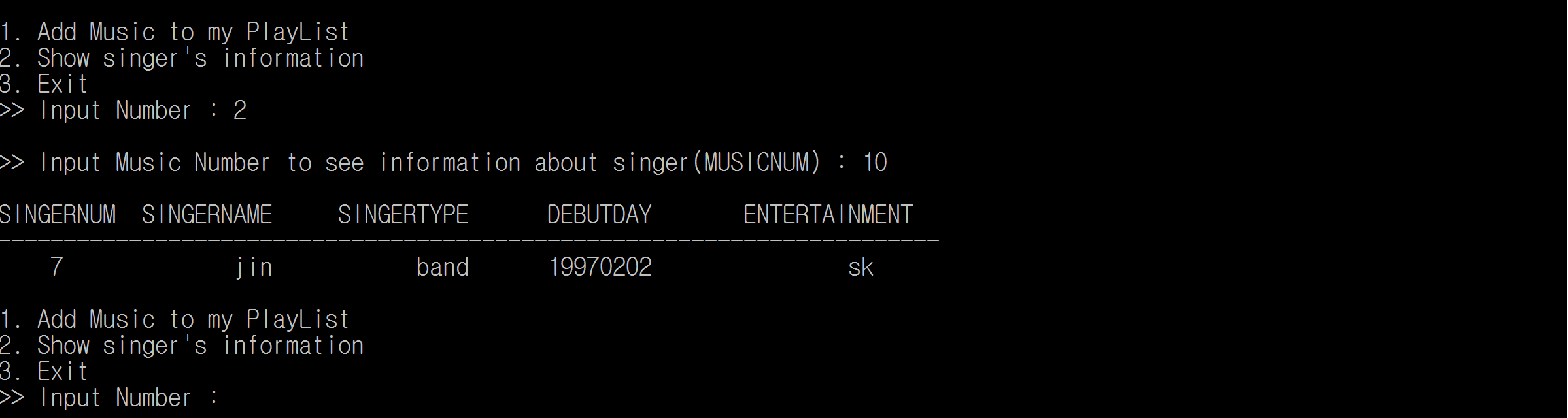
인기 순으로 나열할 때





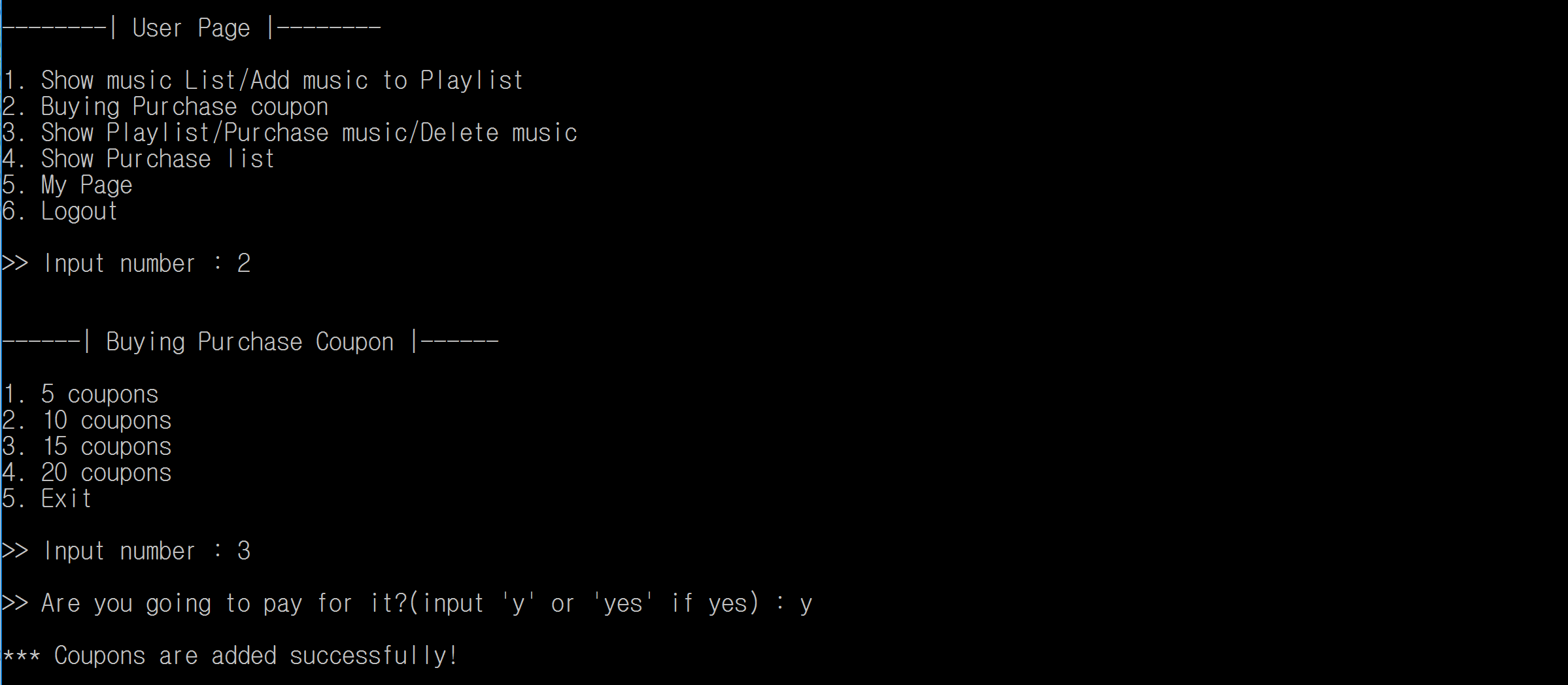


장르별로 나열할 때



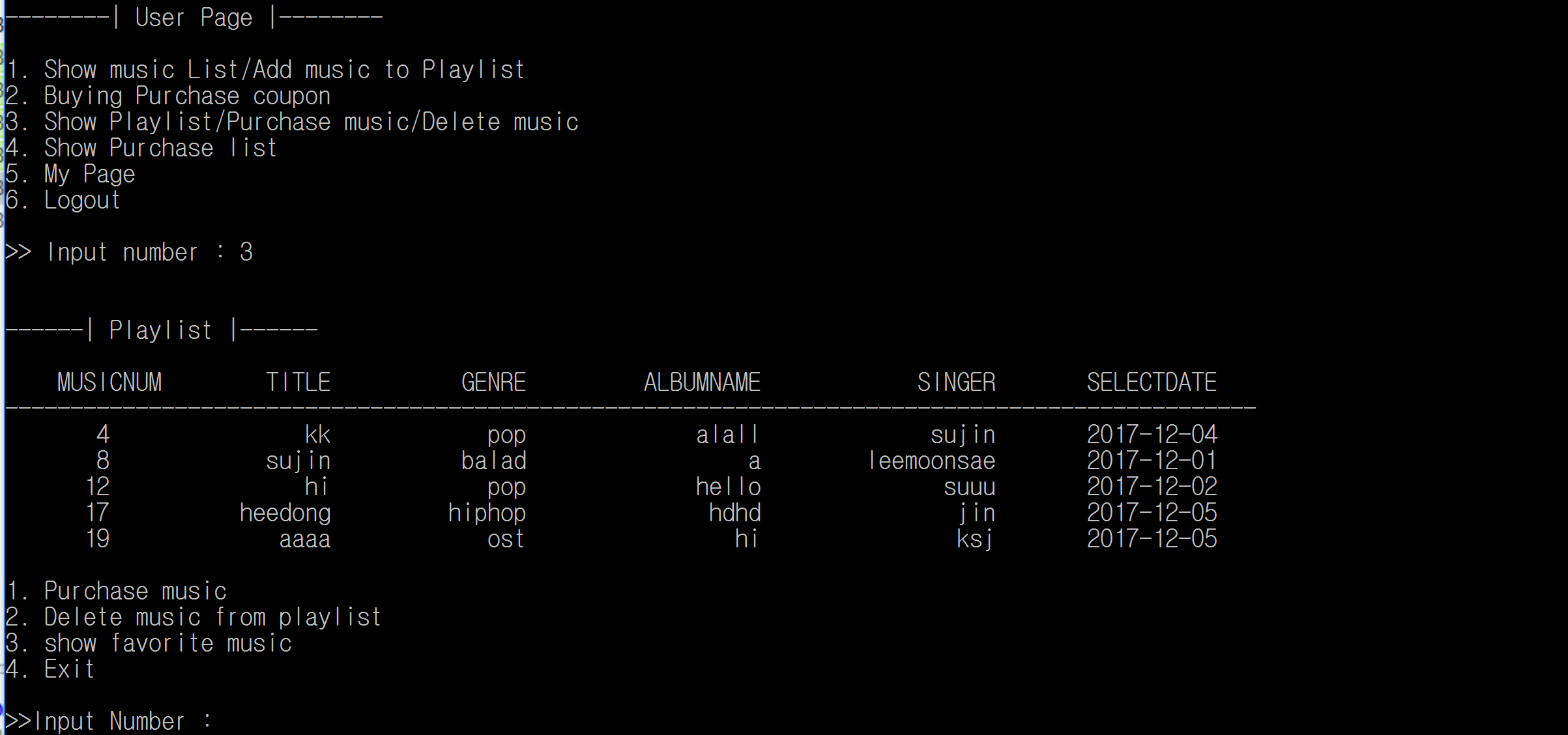
상세하게 가수의 정보를 확인하고 싶을 때

<쿠폰 구매>



쿠폰을 구매하는 과정이다. 쿠폰의 개수만큼 곡을 구매할 수 있다.

<플레이리스트 조회>



로그인 중인 사용자의 플레이리스트에 포함된 곡이 조회된다. 각 사용자마다 플레이리스트는 한 개이다.

<곡 구매>



곡을 구매할 수 있으며 이후부터 전체 듣기가 가능하다.

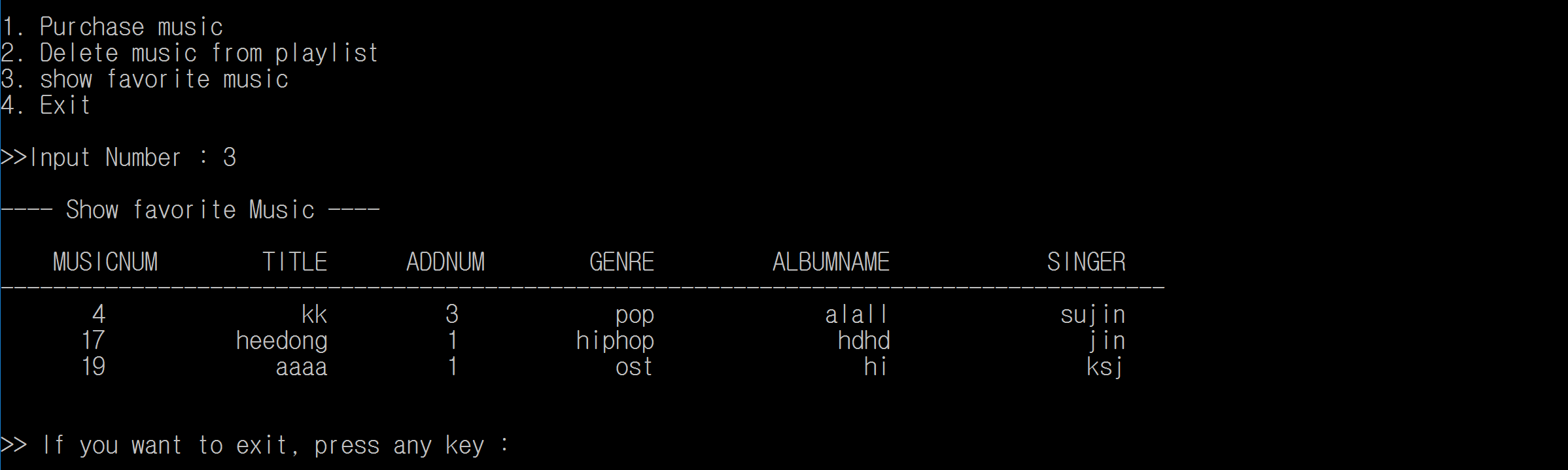


이미 구매한 곡의 경우 이렇게 메시지가 뜬다.

<플레이리스트에서 곡 삭제>



플레이리스트에서 곡을 삭제한다.



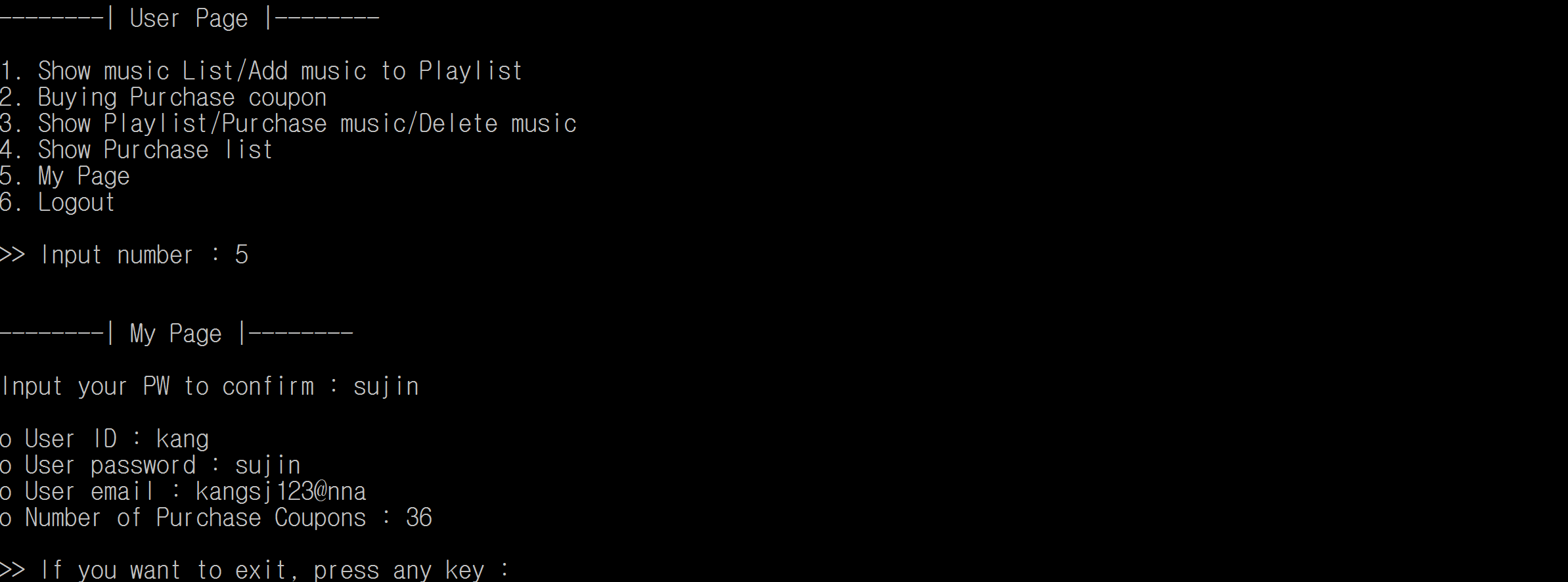
Favorite music으로 설정한 음악의 목록이 조회된다.

<구매한 곡 조회>



구매한 곡의 목록을 조회할 수 있으며 구매한 날의 날짜까지 조회할 수 있고 해당 곡에 대한 comment를 달 수 있게 하여 곡마다 자신만의 의견을 담을 수 있게 했다.

<마이페이지>

자신의 정보를 조회할 수 있으며 purchase coupon의 개수를 확인할 수 있다. 이 사용자의 경우 36개의 곡을 더 살 수 있다. 또한 쿠폰을 사면 구입한 쿠폰만큼 쿠폰의 개수가 늘어나며 곡을 사면 이 값이 하나씩 줄어든다.

<종료>

