

Database System

2019_ project4

Music_Player

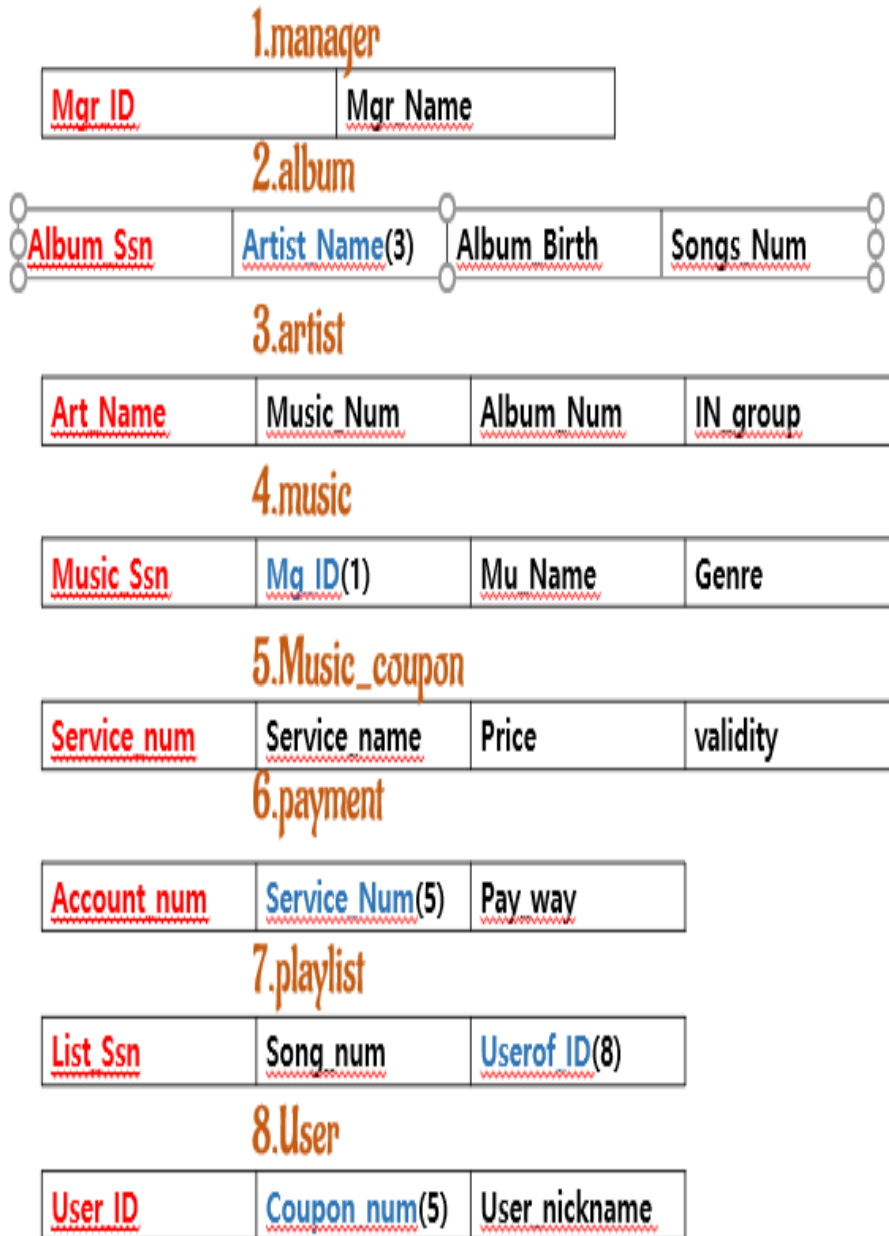
11 월 30 일

2018008069

김재형



1. Relational model 과 그에 대한 설명



빨간 색 글씨로 되어있는 부분은 primary key 이며 파란색 글씨로 되어있는 부분은 foreign key 이다.

위의 기본적인 구조로 music_player 를 작성했으며
각각의 객체에 대한 설명은 아래에 기재해놓았다.

1. Manager

매니저는 ID 와 매니저 이름으로 구성되어있으며 Music_player 실행할시 ID 를 입력하고 music _player 에 등록되어있는 ID 와 일치해야만 로그인이 가능하며 그 이후에 관리자 권한을 이용하여 객체를 생성 또는 삭제할수 있다.

2. Album

앨범은 앨범고유 번호인 Ssn 과 외래키인 Artist_Name, 앨범발매일, 수록되어있는 노래수가 있으며 앨범이 존재하기 위해서는 아티스트가 사전에 등록되어있어야한다.

3. Artist

아티스트는 primary key 인 아티스트 이름과 아티스트의 노래개수,앨범개수 그리고 소속 그룹의 이름으로 구성되어 있다.

4. Music

음악은 primarykey 인 음악 고유번호와 외래키인 관리자의 ID, 음악이름 ,장르 로 구성되어있으며 음악은 관리자만이 등록 혹은 삭제할수있다.

5. Music_coupon

음악이용권은 primary key 인 Service_num 과 음악이용권의 이름, 그것에 대한 가격,그리고 유효기간이 적혀있다.

6. Payment

Payment 는 지불수단으로 결제하려고하는 음악이용권의 번호가 외래키이며 계좌번호가 primary key 이고 결제방법을 속성키로 가지고있다.

7. Playlist

사용자는 playlist 를 가지고 있으며 외래키로 user 의 ID 를 가지고있다. 또한 primary key 로 고유번호 또한 수록노래수또한 가지고 있다. 이때 사용자가 노래수를 삭제하거나 삽입할시 노래수는 +1, -1 등을 한다.

8. User

사용자는 관리자가 사용자를 등록 혹은 삭제할수있으며 사용자가 이용하는 음악이용권을 고유키로 가지고있으며 사용자의 닉네임을 속성키로 가지고 있다.

2. 기능과 그에 대한 스크린샷

1. Manager mode – 1. 로그인하기

우선 music_player 가 실행되기 이전부터 manaer 인 (Aza1200,Jae-Hyeong) 과 (dienstar,Dien) 은 이미 등록되어있는 상태다.

```
-----Music Player-----  
Welcome to music_player what do you want to do? :)  
0. Exit  
1. Manager_menu  
2. User_menu  
Input : 1  
What is you manager ID? : aza1200  
Welcome aza1200 glad to enter this program!  
  
-----Manager menu-----  
0. Return to previous menu  
1. Go into register menu  
2. Go into delete menu  
3. View List  
Input : 0  
  
-----Music Player-----  
Welcome to music_player what do you want to do? :)  
0. Exit  
1. Manager_menu  
2. User_menu  
Input : 2  
  
What do you want? :)  
0. Return to previous menu  
1. Make new Playlist Something  
2. Delete or add music to playlist  
3. change Music_coupon  
4. Regist/Delete payment  
Input : 0  
|  
  
-----Music Player-----  
Welcome to music_player what do you want to do? :)  
0. Exit  
1. Manager_menu  
2. User_menu  
Input :
```

Music player 실행시

0 번은 프로그램 종료

1 번은 Manager 의 메뉴로 들어가는것인데 이를 위해서는 기존에 등록되어있는 매니저의 ID 와 동일해야 로그인이 가능하다.

(매니저 매뉴에서 0 번 입력시 이전 메뉴로 돌아감!)

2 번을 누르면 user 의 메뉴로 들어갈수있는데 user 의 메뉴에서는 매니저의 메뉴에서와 다르게 여러가지 기능들이 있다. (자세한 기능들에 대한 설명은 이후 페이지에서 다룰 예정)

```

-----Music Player-----
Welcome to music_player what do you want to do? :)
0. Exit
1. Manager_menu
2. User_menu
Input : 1
What is you manager ID? : aaa
Wrong Input 1 times Press '0' If you want to return
What is you manager ID? : aaa
Wrong Input 2 times Press '0' If you want to return
What is you manager ID? : aa
Wrong Input 3 times Press '0' If you want to return
What is you manager ID? : aaa
Wrong Input 4 times Press '0' If you want to return
What is you manager ID? : aaa
It's the last time!!! Please be careful!!
Wrong Input 5 times Press '0' If you want to return
What is you manager ID? : aaa
Access denied Please Debug again

```

로그인 에러시

다음과 같이 로그인 에러시
다시 입력하라는 메시지가
나오면 5 번 을 초과해서
매니저 아이디가 틀릴경우
프로그램을 강제종료 한다.

```

-----Music Player-----
Welcome to music_player what do you want to do? :)
0. Exit
1. Manager_menu
2. User_menu
Input : 1
What is your Mg_ID? : aza1200
Welcome Jae-Hyeong Nice to meet you!!

```

하지만 로그인 성공시

밑에 문장과 같이

Welcome Jae_Hyeong 과같이

매니저의 ID 인 aza1200 에 해당하는

매니저의 이름을 SELECT 하여

매니저의 이름을 출력함으로써

환영하는 문구또한 센스있게 출력해준다.

```

if(idcheck==false) {
    System.out.print("What is your Mg_ID? : ");
    Mg_ID=Sinput.nextLine();
    StringBuilder hey=new StringBuilder();
    String sb111=hey.append("SELECT * FROM manager WHERE Mgr_ID=")
        .append(""+Mg_ID+"")
        .toString();
    if(Mg_ID.equals("0") && error>0 ) {
        check=true;
        continue;
    }
    ResultSet rs11=stmt.executeQuery(sb111);
    if(!rs11.next()) {
        if(error==5) {
            System.out.println("\n It is your last time be careful!!");
        }
        if(error>5) {
            System.out.println("Access denied Please debug again");
            System.exit(0);
        }
        error++;
        System.out.println("Wrong Input "+error+" times Press 0 if you want to get back");
        check=false;
        Input=1;
        continue;
    }
}

String name="SELECT * FROM manager WHERE Mgr_ID='"+Mg_ID+"'";
ResultSet rs12=stmt.executeQuery(name);
rs12.next();
System.out.println("Welcome "+rs12.getString("Mgr_Name")+" Nice to meet you!!" );
idcheck=true;
error=0;
//1. Manager menu

```

위에서 설명한 바와 같이 Select * from manager where Mgr_ID= Mg_ID 를 하는데 Rs11.next() 에서 null 값이 나오게 되면 매니저 아이디에 해당하는 값이 없다는 것을 의미한다. 즉, 이때에는 다시 입력하라는 문구와 에러 메시지가 출력되며 만일 rs.next() 값이 null 값이 아닐시 해당 rs.next() 에서 해당 매니저의 아이디에 해당하는 이름을 select 하여 서 이름을 출력하며 환영 메시지 를 센스있게 출력한다.

2. 관리자 메뉴(삽입,삭제,데이터확인)

```
-----Manager menu-----
0. Return to previous menu
1. Go into register menu
2. Go into delete menu
3. View List
Input :
```

메니저 메뉴에서는 register 메뉴 Delete 메뉴 ,view 리스트가 있는데 말 그대로 register menu 는 쿼리문의 insert 기능을 수행하는 것이며 Delete 메뉴는 쿼리 문의 delete 기능을 수행하는 것이며 view 는 전체 등록된 데이터를 조회하는것으로 Select 문을 보는 것이다.

```
-----Manager menu-----
0. Return to previous menu
1. Go into register menu
2. Go into delete menu
3. View List
Input : 1

-----Manager -> register-----
What do you want to register?
0.Return  1.music  2.user  3.artist  4.album
5.Coupn
Input :
```

1 번 입력시 다음과 같이 Music, user, artist, album ,coupon 등의 데이터들을 입력 할것이다. 밑에 스크린샷을 확인할수있을것이다.

```
error=0;
//1. Manager menu
System.out.println("\n-----Manager menu-----\n");
System.out.println("0. Return to previous menu");
System.out.println("1. Go into register menu");
System.out.println("2. Go into delete menu");
System.out.println("3. View List");
System.out.print("Input : ");
int Input1=Sinput.nextInt();
String trash1=Sinput.nextLine();
```

매니저문의 메뉴에 해당하고 정수를 입력받는 코드는 다음과 같다.

이때 string trash 를 하는 이유는 정수를 입력받을 때 enter 키값역시 입력받기에 해당 trash1 은 ""라는 값을 가지게 된다.

(각각의 데이터 insert 하기)

각 데이터들마다 key 의 이름이 다르기 때문에 입력받을 때 각각의 이름역시 다르게 해서
입력받았고 , manager 는 playlist 와 payment 제외하고 다 등록이 가능하기에 아래의
스크린샷을 보면 쉽게 이해할수 있다.

```
in -----Manager -> register-----
in
in What do you want to register?
is 0.Return    1.music    2.user    3.artist    4.album
is 5.Coupn
is Input : 1
is Music_Ssn : 1
is Mg_ID : aza1200
is Mu_Name : Keeping_your_head_up
is Genre : Pop
is
is -----Music Player-----
```

←Music 등록

```
-----Manager -> register-----
What do you want to register?
0.Return    1.music    2.user    3.artist    4.album
5.Coupn
Input : 3
Artist_name : Zico
Music num : 20
Album_num : 20
IN_Group : SOLO
```

←artist 등록

```
-----Music Player-----
-----Manager -> register-----
What do you want to register?
0.Return    1.music    2.user    3.artist    4.album
5.Coupn
Input : 5
Coupon_num : 1
Coupn name : One_month_free
Price : 3000
Validate : 2020-12-31
```

←Coupon 등록

```
-----Manager -> register-----
What do you want to register?
0.Return    1.music    2.user    3.artist    4.album
5.Coupn
Input : 4
Album_Ssn : 1
Artist_Name : Zico
Album_Birth : 2019-11-30
Songs_Num : 20
```

←album 등록

이때 외래키인 artist 의 이름이
들어가므로 album 을 등록할시에는
그전에 아티스트를 먼저 등록해놓아야
한다.

-----Manager -> register-----

```
What do you want to register?
0.Return 1.music 2.user 3.artist 4.album
5.Coupn
Input : 2
User_ID : aza1200
Coupon_num : 1
User_nickname : Kim_Jae_Hyeong
```

```
else if(Input1_1==2) {
    System.out.print("User_ID : ");
    String User_ID=Sinput.nextLine();
    System.out.print("Coupon_num : ");
    String Coupon_num=Sinput.nextLine();
    System.out.print("User_nickname : ");
    String User_nickname=Sinput.nextLine();
    StringBuilder sb1=new StringBuilder();
    String sql30 = sb1.append("Insert into " + "User"+" VALUES(")
        .append("'" + User_ID + "'",")")
        .append("'" + Coupon_num + "'",")")
        .append("'" + User_nickname + "'"")
        .append(");")
        .toString();

    try {
        stmt.executeUpdate(sql30);
    }catch (SQLException e) {
        e.printStackTrace();
    }
}
else if(Input1_1==3) {
```

대표적으로 user 를 insert 한것 코드를 캡쳐한것이다.

다른 table 역시 위와 같으므로 생략하겠다.

자바 코드는 다음과 같다. 이때 string builder() 를 사용해 문장을 만들었으며 마지막에 toString 을 사용해 executeUpdate 의 매개변수인 string 형식으로 바꿔주었다. 이때 쿼리문은 Insert 문을 사용해 데이터 값을 사용했으며 위와 같은 방식으로 music,playlist,payment 등 insert 를하였다.

←User 등록

이때 외래키인 music_coupon 의 번호가 들어가므로 user 를 등록할시에는 그전에 music_coupon 을 먼저 등록해놓아야 한다.

(각각의 데이터 view 하기)

Manager 의 메뉴에서는 각각의 데이터를 관찰할수 있는데 이때는 Select 문을 써서 Select * from table_name 등으로 표현을 하여서 실행을 했으며 , 이를 통해서 각각의 테이블들에 어떠한 데이터가 들어가 있는지 확인할수 있다.

1. Artist View

다음과같이 구별하기 편하게 key 별로 나누어서 출력을 하였으며 key 값들을 한눈에 알아보기 쉽다

```
-----Manager -> view-----  
  
0.Return    1.music    2.user    3.artist    4.album  
5.playlist  6.payment  7.coupon  
Input : 4  
|  
-----  
1 : (1)Album_Ssn : 1 (2)Artist_Name : Zico (3)Album_Birth : 2019-11-30 (4)Songs_Num : 20  
2 : (1)Album_Ssn : 2 (2)Artist_Name : Big_Bang (3)Album_Birth : 2019-20-02 (4)Songs_Num : 399  
-----
```

다른 view 들 역시 위와 같은 방법으로 동일하며 스크린샷을 볼시 이해 하기 쉬울 것이다.

```
-----Manager -> view-----  
  
0.Return    1.music    2.user    3.artist    4.album  
5.playlist  6.payment  7.coupon  
Input : 1  
|  
-----  
1 : (1)Music_Ssn : 1 (2)Mg_ID : aza1200 (3)Mu_Name : Keeping_your_head_up (4)Genre : Pop  
2 : (1)Music_Ssn : 2 (2)Mg_ID : aza1200 (3)Mu_Name : Loser (4)Genre : K-POP  
-----
```

```
-----Manager -> view-----  
  
0.Return    1.music    2.user    3.artist    4.album  
5.playlist  6.payment  7.coupon  
Input : 3  
|  
-----  
1 : (1)Artist_Name : Big_Bang (2)Music_Num : 30 (3)Albumnum : 40 (4)IN-gourp : ToOP  
2 : (1)Artist_Name : Zico (2)Music_Num : 20 (3)Albumnum : 20 (4)IN-gourp : SOLO  
-----
```

```
-----Manager -> view-----  
  
0.Return    1.music    2.user    3.artist    4.album  
5.playlist  6.payment  7.coupon  
Input : 2  
|  
-----  
1 : (1)User_ID : aza1200 (2)Coupon_num : 1 (3)User_nickname : Kim_Jae_Hyeong  
2 : (1)User_ID : azzzz (2)Coupon_num : 1 (3)User_nickname : Babe  
-----
```

```
-----Manager -> view-----
```

```
0.Return    1.music    2.user    3.artist    4.album
5.playlist  6.payment  7.coupon
Input : 7
```

```
1 : (1)Service_num : 1 (2)Service_Name : One_month_free (3)Price : 3000 (4)validate : 2020-12-31
2 : (1)Service_num : 2 (2)Service_Name : Free (3)Price : 2000000 (4)validate : 3000
-----
```

```
else if(Input1==3) {
    System.out.println("\n-----Manager -> view-----\n");
    System.out.println("0.Return    1.music    2.user    3.artist    4.album");
    System.out.println("5.playlist  6.payment  7.coupon");
    System.out.print("Input : ");
    int Input1_3=Sinput.nextInt();
    String tCash3=Sinput.nextLine();
    if(Input1_3==0) {
        check=false;
        Input=1;
        continue;
    }
    else if(Input1_3==1) {
        ResultSet rs = stmt.executeQuery("SELECT * FROM music");
        System.out.println("\n-----");
        int k=1;
        while(rs.next()) {
            System.out.print(k+" : ");
            String Music_Ssn=rs.getString("Music_Ssn");
            String Mg_IDs=rs.getString("Mg_ID");
            String Mu_Name=rs.getString("Mu_Name");
            String Genre=rs.getString("Genre");
            System.out.println("(1)Music_Ssn : " + Music_Ssn + " (2)Mg_ID : " + Mg_IDs+ " (3)Mu_Name "
                + " : "+Mu_Name+ " (4)Genre : "+Genre);
            k++;
        }
        System.out.println("-----");
    }
}
else if(Input1_3==2) {
```

다음은 view 에 해당하는 코드를 나타낸것이다.1 번부터 7 번까지 모두 비슷한 형식이므로 하나만 설명하고 나머지것들은 생략하겠다.

View 를 할 때 select 문을 사용하였으며 insert 와 마찬가지로 string builder 를 사용하여 실행하였다. 그리고 출력시 무작정 출력하면 눈이 아플수도 있으므로 출력시 보기 편하라고 번호를 써서 좀더 보기 편하게 출력하도록 코드를 짰다.

각각의 데이터 delete 하기

데이터들을 삭제할 때 막 삭제하면 에러가 날 수 있다. 외래키 참조하는 데이터의 경우 해당 외래키를 먼저 삭제해야 그 데이터를 삭제할 수 있기 때문이다.

따라서 외래키를 가지고 있는 데이터들을 먼저 삭제하는 방향으로 삭제를 하였으며 이후 데이터를 삭제할 때에는 굳이 모든 key 값들을 입력할 필요없이 primary key 만 입력함으로써 삭제하였다.

```
-----Manager -> delete-----
```

```
What do you want to delete?
0.Return    1.music    2.user    3.artist    4.album
5.Music_Coupon
Input : 4
Enter the Album's Number -> (Primary key) : 1
```

```
-----Music Player-----
```

```
Welcome to music_player what do you want to do? :)
```

```
0. Exit
1. Manager_menu
2. User_menu
Input : 1
Welcome Jae-Hyeong Nice to meet you!!
```

```
-----Manager menu-----
```

```
0. Return to previous menu
1. Go into register menu
2. Go into delete menu
3. View List
Input : 3
```

```
-----Manager -> view-----
```

```
0.Return    1.music    2.user    3.artist    4.album
5.playlist  6.payment  7.coupon
Input : 4
```

```
-----
```

```
1 : (1)Album_Ssn : 2 (2)Artist_Name : Big_Bang (3)Album_Birth : 2019-20-02 (4)Songs_Num : 399
-----
```

```
-----Music Player-----
```

다음그림은 처음에 album 의 개수가 두개였고 album 의 number 가 1 인 album 의 primarykey 값을 제거하였더니 album 이 한 개만 보이는 것을 나타낸것이다 . 다른 테이블들역시 위와 같은 방법으로 설계하였다.

```

else if(Input1==2) {
    System.out.println("\n-----Manager -> delete-----\n");
    System.out.println("What do you want to delete?");
    System.out.println("0.Return    1.music    2.user    3.artist    4.album");
    System.out.println("5.Music_Coupon ");
    System.out.print("Input : ");
    int Input1_2=Sinput.nextInt();
    String trash2=Sinput.nextLine();
    if(Input1_2==0) {
        check=false;
        Input=1;
        continue;
    }
    else if(Input1_2==1) {
        System.out.print("Enter the musci's Number -> (Primary key) : ");
        String Music_Ssn =Sinput.nextLine();
        StringBuilder sb=new StringBuilder();
        String sql=sb.append("Delete FROM music "
            + "WHERE Music_Ssn=")
            .append("'" +Music_Ssn+"'")
            .toString();

        try {
            stmt.executeUpdate(sql);
        }
        catch(SQLException e){
            e.printStackTrace();
        }
    }
}
else if(Input1_2==2) {

```

Delete 문의 코드는 다음과 같다. 나머지 6 개의 attribute 역시 위와 같은 방식이므로 생략하겠다. 이때 Primary key 를 입력받고 그곳에 해당하는 table 을 찾아서 삭제하는 방식으로 삭제를 하였으며 이때 DELETE 문을 사용하였다.

2. User Mode

```
What do you want? :)
0. Return to previous menu
1. Make new Playlist Something
2. Delete or add music to playlist
3. change Music_coupon
4. Regist/Delete payment
Input :
```

User menu 는 다음과 같이 기본적으로 4 개의 메뉴로 이루어져 있는데 1 번부터 설명하자면 1 번은 간단히 playlist 를 등록하는것이다.

2 번은 playlist 의 음악을 등록하거나 삭제하는것인데 이 때 음악을 등록할시 Playlist 에 있는 음악의 개수가 1 이 증가하며 음악을 삭제할시 playlist 의 음악의 개수가 1 개가 줄어든다.

3 번은 음악 쿠폰을 변경하는것인데 사용자가 원할시 사용자가 현재 사용하고 있는 음악 이용권을 변경하는것이다.

4 번은 사용자가 음악이용권을 결제할 때 결제수단 즉 결제할 방식을 등록하는 것이다. 이때 payment 는 primary key 인 계좌번호와 결제수단 , 그리고 결제할 이용권 의 번호를 외래 키로 가지고있다.

Playlist 등록은 위에 있는 manager 모드와 동일 하므로 생략하도록 하겠다.

2 번의 playlist 의 음악을 보자

```
-----Manager -> view-----

0.Return    1.music    2.user    3.artist    4.album
5.playlist  6.payment  7.coupon
Input : 5

-----
1 : (1)List_Ssn : 1 (2)Songs_num : 20 (3)UserofID : aza1200
-----
```

그림에서 확인할수 있듯이 이때 노래의 개수는 20 개이다.

```

What do you want to do? 1.Delete 2.Add
Input : 2
Enter the playlist's Ssn : 1

-----Music Player-----

Welcome to music_player what do you want to do? :)

0. Exit
1. Manager_menu
2. User_menu
Input : 1
Welcome Jae-Hyeong Nice to meet you!!

-----Manager menu-----

0. Return to previous menu
1. Go into register menu
2. Go into delete menu
3. View List
Input : 3

-----Manager -> view-----

0.Return 1.music 2.user 3.artist 4.album
5.playlist 6.payment 7.coupon
Input : 5

-----
1 : (1)List_Ssn : 1 (2)Songs_num : 21 (3)UserofID : aza1200
-----

else if(Input2_1==2) {
    System.out.println("What do you want to do? 1.Delete 2.Add");
    System.out.print("Input : ");
    int choose = Sinput.nextInt();
    trash=Sinput.nextLine();
    System.out.print("Enter the playlist's Ssn : ");
    String Ssn=Sinput.nextLine();
    if(choose==1) {
        StringBuilder sb=new StringBuilder();
        String sql = sb.append("UPDATE playlist SET Song_num=Song_num-1 "
            + "WHERE List_Ssn=")
            .append("'" + Ssn + "'")
            .toString();

        try {
            stmt.executeUpdate(sql);
        }catch (SQLException e) {
            e.printStackTrace();
        }
    }
    else if(choose==2) {
        StringBuilder sb=new StringBuilder();
        String sql = sb.append("UPDATE playlist SET Song_num=Song_num+1 "
            + "WHERE List_Ssn=")
            .append("'" + Ssn + "'")
            .toString();

        try {
            stmt.executeUpdate(sql);
        }catch (SQLException e) {
            e.printStackTrace();
        }
    }
}

```

하지만 다음 사진과같이
Playlist 의 음악을 1 번을 눌러서
추가할시 플레이 리스트 의
음악의 개수 가 1 개 증가한다.
이때 음악의 개수가 증가할 때
쿼리의 Update 문을 사용해서
음악의 개수를 증가시키거나
감소시킨다.

다음 쿼리문은 다음과 같다.
쿼리문은 UPDATE 로
표현하였으며 만일 add 할시
+1 을 하여서 노래의 개수를
1 개 증가시켰으며 만일 delete
할시 해당 playlist 의 노래를
1 개 감소시켰다.

```

-----Manager -> view-----
0.Return    1.music    2.user    3.artist    4.album
5.playlist  6.payment  7.coupon
Input : 7

-----
1 : (1)Service_num : 1 (2)Service_Name : One_month_free (3)Price : 3000 (4)validate : 2020-12-31
2 : (1)Service_num : 2 (2)Service_Name : Free (3)Price : 2000000 (4)validate : 3000
-----

-----Music Player-----

Welcome to music_player what do you want to do? :)

0. Exit
1. Manager_menu
2. User_menu
Input : 1
Welcome Jae-Hyeong Nice to meet you!!

-----Manager menu-----

0. Return to previous menu
1. Go into register menu
2. Go into delete menu
3. View List
Input : 3

-----Manager -> view-----

0.Return    1.music    2.user    3.artist    4.album
5.playlist  6.payment  7.coupon
Input : 2
|
-----
1 : (1)User_ID : aza1200 (2)Coupon_num : 1 (3)User_nickname : Kim_Jae_Hyeong
2 : (1)User_ID : azzzz (2)Coupon_num : 1 (3)User_nickname : Babe
-----

```

3 번인 음악 쿠폰 변경을 보자 다음과 같이 음악 쿠폰 2 개가 있고 user 역시 두명이 있다.
 이때 user aza1200 의 음악쿠폰을 2 번으로 변경시켜보자.


```

What do you want? :)
0. Return to previous menu
1. Make new Playlist Something
2. Delete or add music to playlist
3. change Music_coupon
4. Regist/Delete payment
Input : 3
What is your user_ID? : aza1200
What coupon do you want to change? : 2

-----Music Player-----

Welcome to music_player what do you want to do? :)

0. Exit
1. Manager_menu
2. User_menu
Input : 1
Welcome Jae-Hyeong Nice to meet you!!

-----Manager menu-----

0. Return to previous menu
1. Go into register menu
2. Go into delete menu
3. View List
Input : 3

-----Manager -> view-----

0.Return    1.music    2.user    3.artist    4.album
5.playlist  6.payment  7.coupon
Input : 2
|
-----
1 : (1)User_ID : aza1200 (2)Coupon_num : 2 (3)User_nickname : Kim_Jae_Hyeong
2 : (1)User_ID : azzzz (2)Coupon_num : 1 (3)User_nickname : Babe
-----

```

다음과 같이 변경하려는 사용자의 IDEA 인 aza1200 과

변경하려는 쿠폰 의 번호를 입력하고 manager mode 로 들어가 관찰시 coupon number 가 1 에서 2 로 변경된 것을 확인할수있다. 이때 변경은 Update 구문을 사용해서 변경하였다.

```

    }
    else if(Input2_1==3) {
        System.out.print("What is your user_ID? : ");
        String ID=Sinput.nextLine();
        System.out.print("What coupon do you want to change? : ");
        String Coupon_num=Sinput.nextLine();
        StringBuilder sb=new StringBuilder();
        String sql = sb.append("UPDATE User SET Coupon_num=")
            .append("'" + Coupon_num + "'")
            .append(" WHERE User_ID=")
            .append("'" + ID + "'")
            .toString();

        try {
            stmt.executeUpdate(sql);
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}

```

3 번에 해당하는 change 는 다음과 같이 자바 코드를 짰다. 이때 변경할려는 사용자의 ID 를 입력하였으며 사용자가 음악 쿠폰을 어떤것으로 바꾸고 싶은지 입력을 할시 자동으로 바꿀수있도록 Update 문을 사용하였다.

다른 insert,delete 와 같은 방식으로 stringBuilder() 를 사용하였으며 try catch 문 역시 사용하였다.

4 번인 Payment 의 등록과 삭제 역시 위의 다른 table 들과 등록방식은 똑같다 .단지 payment 와 paylist 의 등록권한은 manager 에 있는 것이 아니라 user 한테 있다.

