



Theme & idea

DATE.

< DB 설계 >

① Songs

제목	가수	장르	발매일	곡길이
SPICY	aespa	dance	2023-05-24	180

② Singers

이름	소속사	데뷔년도	인원수
aespa	SM	2020	4

③ agencies (소속사)

이름	위치	설립년도
SM	성동구	1990

④ Ranking (음원순위)

노래	가수	순위
SPICY	aespa	1

⑤ view는 자유롭게 설정

- index 설정은 select문에 따라 설정해야 될 것 같아요

MainMenu

```
1 import java.util.Scanner;
2
3 public class MainMenu {
4     public static void main(String[] args) {
5         int exit=0;
6         Scanner input=new Scanner(System.in);
7         while(exit==0) {
8             System.out.println("##### Music Chart DB #####");
9             System.out.println("---- Please select a menu ----");
10            System.out.println("1. SelectMenu1");
11            System.out.println("2. SelectMenu2");
12            System.out.println("3. SelectMenu3");
13            System.out.println("4. InsertMenu");
14            System.out.println("5. UpdateMenu");
15            System.out.println("6. DeleteMenu");
16            System.out.println("0. exit");
17            System.out.print("menu:");
18            int menu=input.nextInt();
19            switch(menu) {
20                case 1:
21                    SelectMenu1 a=new SelectMenu1();
22                    break;
23                case 2:
24                    SelectMenu2 b=new SelectMenu2();
25                    break;
26                case 3:
27                    SelectMenu3 c=new SelectMenu3();
28                    break;
29                case 4:
30                    InsertMenu d=new InsertMenu();
31                    break;
32                case 5:
33                    UpdateMenu e=new UpdateMenu();
34                    break;
35                case 6:
36                    DeleteMenu f=new DeleteMenu();
37                    break;
38                case 0:
39                    System.out.println("exit!");
40                    exit=1;
41                    break;
42                default:
43                    System.out.println("메뉴가 존재하지 않습니다. 다시 선택해주세요!");
44                    System.out.println("");
45                    continue;
46            }
47        }
48        System.out.println("");
49    }
50}
51}
```

Classes and Methods:

MainMenu: This class is the entry point for the given code. Contains the main method.

main method: This is the method where execution of the program begins. Displays the menu and instantiates each menu class based on the user's selection.

Key variables and objects:

exit: Integer variable indicating whether the program is exiting.

input: A Scanner object to receive input from the user.

menu: Integer variable that stores the menu selected by the user.

Execution flow:

Initialize the exit variable.

The while loop lets you select menus repeatedly until the program ends.

Displays a menu and accepts choices from the user.

Use a switch statement to create an object corresponding to the menu selected by the user.

Performs an action on the selected menu object.

If the user wants to exit by selecting 0, he ends the loop by changing the exit variable.

→ 2 menus(queries) : ask for user input text, contain join, and use 'view' → 채린

Searching for a given singer name, and outputting the singer and agency information

```
*MainMenu.java  DeleteMenu.java  InsertMenu.java  SelectMenu1.java  SelectMenu2.java
1
2 import java.sql.*;
3
4 public class SelectMenu2 {
5     public SelectMenu2() {
6
7         try {
8             //연결 확인
9             Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/teamdb?serverTimezone=UTC", "testuser", "testpw");
10            System.out.println("DB connect success!");
11            Statement stmt = conn.createStatement();
12
13
14             // Singer 입력하면 Agency 를 알 수 있음
15             Scanner scanner = new Scanner(System.in);
16             System.out.print("Enter the Singer to search: ");
17             String inputName = scanner.nextLine();
18
19             // 뷰 생성
20             String createViewQuery = "CREATE OR REPLACE VIEW view_singer_agency AS SELECT singers.name, agencies.name AS agency_name FROM singers JOIN agencies ON singers.agency = agencies.name";
21             PreparedStatement createViewStatement = conn.prepareStatement(createViewQuery);
22             createViewStatement.executeUpdate();
23
24             // 조인과 뷰를 사용한 검색
25             String searchQuery = "SELECT name, agency_name FROM view_singer_agency WHERE name LIKE ?";
26             PreparedStatement searchStatement = conn.prepareStatement(searchQuery);
27             searchStatement.setString(1, "%" + inputName + "%");
28             ResultSet resultSet = searchStatement.executeQuery();
29
30             // 검색 결과 출력
31             while (resultSet.next()) {
32                 String singerName = resultSet.getString("name");
33                 String agencyName = resultSet.getString("agency_name");
34                 System.out.println("Singer: " + singerName + ", Agency: " + agencyName);
35             }
36
37             // 끝
38             resultSet.close();
39             searchStatement.close();
40             createViewStatement.close();
41             conn.close();
42
43
44         } catch (SQLException sqle) {
45             System.out.println("SQLException: " + sqle);
46         }
47
48
49     }
50 }
```

Classes and Methods:

SelectMenu2: This is the class defined in the given code. It has a constructor.

Execution flow:

The object is created when the SelectMenu2 constructor is called.

Connect to database.

Enter the singer's name from the user.

Execute the SQL query that creates the view.

Executes a SQL query for retrieval and stores the results in a ResultSet object.

Outputs search results using a ResultSet object.

Closes used objects and closes database connection.

→ 2 menus(queries) : ask for user input text, contain join, and use 'view' → 시온

After receiving the location of the agency from the user,
the names of singers belonging to that location are retrieved and output.

```
1 import java.sql.*;
2
3 public class SelectMenu1 {
4     public SelectMenu1() {
5         try {
6
6             // 데이터베이스 연결
7             Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/teamdb?severTimezone=UTC", "testuser", "testpw");
8             System.out.println("DB connect success!");
9             Statement stmt = conn.createStatement();
10
11
12             // 사용자로부터 입력 받기
13             java.util.Scanner scanner = new java.util.Scanner(System.in);
14             System.out.print("Enter the location of the agency: ");
15             String location = scanner.nextLine();
16
17             // 뷰 생성 쿼리 작성
18             String createViewQuery = "CREATE OR REPLACE VIEW SingerNamesView AS "
19                         + "SELECT Singers.name FROM Singers JOIN Agencies ON Singers.agency = Agencies.name WHERE Agencies.location = ?;";
20             PreparedStatement createViewStmt = conn.prepareStatement(createViewQuery);
21             createViewStmt.setString(1, location);
22             createViewStmt.executeUpdate();
23
24             // 뷰 조회 쿼리 작성 및 실행
25             String selectViewQuery = "SELECT * FROM SingerNamesView";
26             Statement selectViewStmt = conn.createStatement();
27             ResultSet rs = selectViewStmt.executeQuery(selectViewQuery);
28
29             // 결과 출력
30             System.out.println("Singer names:");
31             while (rs.next()) {
32                 String singerName = rs.getString("name");
33                 System.out.println(singerName);
34             }
35
36             // 연결 및 자원 해제
37             rs.close();
38             selectViewStmt.close();
39             createViewStmt.close();
40             conn.close();
41         } catch (Exception e) {
42             e.printStackTrace();
43         }
44     }
45 }
46 }
```

Classes and Methods:

SelectMenu1: This is the class defined in the given code. has a constructor

Execution flow:

The object is created when the SelectMenu1 constructor is called.

Connect to database.

Receives the location of the organization from the user.

Execute the SQL query that creates the view.

Executes a SQL query to query the view and stores the results in a ResultSet object.

Displays the result of the view using the ResultSet object.

Closes used objects and closes database connection.

→ 1 menu(query) should contain aggregation, group by → 윤재

Calculate the number of songs released by singer
in the database and output the result

```
1 *MainMenu.java  2 DeleteMenu.java  3 InsertMenu.java  4 SelectMenu1.java  5 SelectMenu2.java  6 SelectMenu3.java
1+ import java.sql.Connection;
2
3 public class SelectMenu3 {
4     public SelectMenu3() {
5         try {
6             //DB연결(teamdb이용)
7             Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/teamdb?serverTimezone=Asia/Seoul",
8                 "testuser", "testpw");
9             System.out.println("DB connect success!");
10            Statement stmt = conn.createStatement();
11
12            //Select문(aggregation function with group by)
13            //현 차트 안에서 각 가수가 발매한 음원수를 추출하는 select
14            ResultSet rset=stmt.executeQuery("select singer,count(*) from songs group by singer");
15
16            //select문 결과 출력
17            while(rset.next())
18            {
19                System.out.println("가수:"+rset.getString(1)+"  음원 수:"+rset.getString(2));
20            }
21
22        } catch (SQLException sqle) {
23            System.out.println("SQLException" + sqle);
24        }
25    }
26}
27
28
29
30
31
32 }
```

Classes and Methods:

SelectMenu3: This is the class defined in the given code. It has a constructor.

Execution flow:

The object is created when the SelectMenu3 constructor is called.

Connect to database.

Execute SQL queries using stmt.

Save the execution result to rset.

Use rset to loop through the results and output them.

Closes used objects and closes database connection.

4 insert : 소속사 이름만 입력(그 외 null값 처리) - 채린

Insert the agency name entered from the user into the Agencies table of the database

```
1+ import java.sql.*;
2
3 public class InsertMenu {
4     public InsertMenu() {
5
6         try {
7             //연결 확인
8             Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/teamdb?serverTimezone=UTC", "testuser", "testpw");
9             System.out.println("DB connect success!");
10            Statement stmt = conn.createStatement();
11
12            //input을 내용
13            Scanner scanner = new Scanner(System.in);
14            System.out.print("Enter the name of the Agencies: ");
15            String AgenciesName = scanner.nextLine();
16
17            // INSERT 문 실행
18            String sql = "INSERT INTO Agencies (name) VALUES (?)";
19            PreparedStatement statement = conn.prepareStatement(sql);
20            statement.setString(1, AgenciesName);
21            statement.executeUpdate();
22
23            System.out.println("Agencies inserted successfully.");
24
25            // 종료
26            statement.close();
27            conn.close();
28
29        } catch(SQLException ex) {
30            System.out.println("SQLException"+ ex);
31        }
32    }
33
34 }
```

Classes and Methods:

InsertMenu: The class defined in the given code. It has a constructor.

Execution flow:

The object is created when the InsertMenu constructor is called.

Connect to database.

It uses scanner to get input from the user.

Executes an INSERT statement using the entered information.

If the INSERT statement executes successfully, it prints a success message.

Closes used objects and closes database connection.

5 update : Singer(새 멤버 영입을 원하는 SingerGrop의 이름을 입력하시오. 그룹 인원 +1) - 시온

Find the singer name entered from the user in the Singers table of the database and increase the numMembers value of the singer by 1

```
1④ import java.sql.Connection;④
2
3 public class UpdateMenu {
4     public UpdateMenu() {
5         try {
6             // JDBC 드라이버 로드
7             Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/teamdb?severTimezone=UTC", "testuser", "testpw");
8             System.out.println("DB connect success!");
9             Statement stmt = conn.createStatement();
10
11             // 사용자로부터 입력 받기
12             java.util.Scanner scanner = new java.util.Scanner(System.in);
13             System.out.print("Enter the name of the singer: ");
14             String singerName = scanner.nextLine();
15
16             // 가수의 numMembers 값을 1 증가시키는 쿼리 작성 및 실행
17             String query = "UPDATE Singers SET numMembers = numMembers + 1 WHERE name = ?";
18             PreparedStatement pstmt = conn.prepareStatement(query);
19             pstmt.setString(1, singerName);
20             int rowsAffected = pstmt.executeUpdate();
21
22             // 업데이트 결과 출력
23             if (rowsAffected > 0) {
24                 String selectQuery = "SELECT numMembers FROM Singers WHERE name = ?";
25                 PreparedStatement selectStmt = conn.prepareStatement(selectQuery);
26                 selectStmt.setString(1, singerName);
27                 ResultSet rs = selectStmt.executeQuery();
28                 if (rs.next()) {
29                     int updatedNumMembers = rs.getInt("numMembers");
30                     System.out.println("Successfully updated numMembers for singer: " + singerName);
31                     System.out.println("Updated numMembers value: " + updatedNumMembers);
32                 }
33                 rs.close();
34                 selectStmt.close();
35             } else {
36                 System.out.println("No rows affected. Singer not found.");
37             }
38
39             // 연결 및 자원 해제
40             pstmt.close();
41             conn.close();
42         } catch (Exception e) {
43             e.printStackTrace();
44         }
45     }
46 }
47 }
```

Classes and Methods:

UpdateMenu: A class defined in the given code. It has a constructor.

Execution flow:

An object is created when the UpdateMenu constructor is called.
Connect to database.

I am using the scanner to get the singer's name from the user.
Create and execute an UPDATE statement that increases the numMembers value of the entered singer by 1.

If the UPDATE statement is executed successfully, the update result is output.

If there is a result, the numMembers value of the corresponding mantissa is searched and output.

If there are no results, "No rows affected. Singer not found." Prints a message.

Closes used objects and closes database connection.

6 delete : (song & ranking → 같이 삭제) song에서 노래 삭제하기. 랭킹에서도 같이 삭제됨 - 윤재

The song entered by the user is deleted from the Songs table and updated in Ranking.

```
1+ import java.sql.Connection;
2 |
3 public class DeleteMenu {
4     public DeleteMenu() {
5
6     }
7
8     try {
9
10        Scanner input=new Scanner(System.in);
11        //DB연결(teamdb 이용)
12        Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/teamdb?serverTimezone=Asia/Seoul",
13            "testuser","testpw");
14        System.out.println("DB connect success!");
15        Statement stmt = conn.createStatement();
16
17        //사용자에게 삭제할 노래 제목 입력받기
18        System.out.print("Please enter the title of the song you want to delete:");
19        String title=input.nextLine();
20
21        try {
22            //1.삭제될 노래를 제외한 나머지 노래들 순위 조정
23            //삭제할 노래의 순위 추출
24            int rank=0;
25            String sql="select ranking from ranking where song='"+title+"'";
26            ResultSet rset=stmt.executeQuery(sql);
27            if(rset.next())
28            {
29                rank=rset.getInt(1);
30            }
31            else{
32                System.out.println("노래가 존재하지 않습니다");
33                System.exit(1);
34            }
35        };
36        //삭제할 노래를 제외한 나머지 노래들의 ranking 조정
37        PreparedStatement pStmt2=conn.prepareStatement("update ranking set ranking=ranking-1 where ranking>" );
38        pStmt2.setInt(1, rank);
39        pStmt2.executeUpdate();
40
41        //2.선택한 노래 삭제 진행
42        PreparedStatement pStmt= conn.prepareStatement("delete from songs where title=?");
43        pStmt.setString(1,title);
44        int result=pStmt.executeUpdate();
45
46        if(result>0) {System.out.println("delete success!");}
47
48    }catch(SQLException e) {
49        e.printStackTrace();
50    }
51
52    } catch (SQLException sqle) {
53        System.out.println("SQLException" + sqle);
54    }
55
56
57
58    }
59 }
```

Classes and Methods:

DeleteMenu: The class defined in the given code. It has a constructor.

Execution flow:

The object is created when the DeleteMenu constructor is called.

Connect to database.

Use input to get the song title to be deleted from the user.

Extract the rank of the input song.

Adjust the ranks of songs with ranks greater than the ranks extracted.

Delete the entered song from the songs table.

Print the delete result.

Closes used objects and closes database connection.

Initial value

```
mysql> select *  
-> from Agencies;  
+-----+-----+-----+  
| name | location | established |  
+-----+-----+-----+  
| Big Planet Made Enter | Gangnam-gu, Seoul | 2021 |  
| Cube Entertainment | Seongdong-gu, Seoul | 2013 |  
| Mulgogi Music | Mapo-gu, Seoul | 2021 |  
| Pledis Entertainment | Yongsan-gu, Seoul | 2010 |  
| SM Entertainment | Seongdong-gu, Seoul | 1995 |  
| Source Music | Gangnam-gu, Seoul | 2014 |  
| Starship Entertainment | Gangnam-gu, Seoul | 2008 |  
| YG Entertainment | Mapo-gu, Seoul | 1998 |  
+-----+-----+-----+  
8 rows in set (0.00 sec)  
  
mysql> select *  
-> from Singers;  
+-----+-----+-----+-----+  
| name | agency | debut | numMembers |  
+-----+-----+-----+-----+  
| (G)I-DLE | Cube Entertainment | 2018 | 5 |  
| aespa | SM Entertainment | 2020 | 4 |  
| Huh Gak | Big Planet Made Enter | 2010 | 1 |  
| IVE | Starship Entertainment | 2021 | 6 |  
| JISOO | YG Entertainment | 2023 | 1 |  
| LE SSERAFIM | Source Music | 2022 | 5 |  
| Lim Young Woong | Mulgogi Music | 2016 | 1 |  
| SEVENTEEN | Pledis Entertainment | 2015 | 13 |  
+-----+-----+-----+-----+  
8 rows in set (0.00 sec)  
  
mysql> select *  
-> from Songs;  
+-----+-----+-----+-----+-----+  
| title | singer | genre | releaseDate | length |  
+-----+-----+-----+-----+-----+  
| Flower | JISOO | dance | 2023-03-31 | 173 |  
| I AM | IVE | dance | 2023-04-10 | 183 |  
| Kitsch | IVE | dance | 2023-04-10 | 195 |  
| London Boy | Lim Young Woong | rock, metal | 2022-11-15 | 245 |  
| Love Always Run Away | Lim Young Woong | ballad | 2021-10-11 | 240 |  
| Polaroid | Lim Young Woong | rock, metal | 2022-11-15 | 207 |  
| Queencard | (G)I-DLE | dance | 2023-05-15 | 161 |  
| Spicy | aespa | dance | 2023-05-08 | 197 |  
| Super | SEVENTEEN | dance | 2023-04-24 | 200 |  
| UNFORGIVEN | LE SSERAFIM | dance | 2023-05-01 | 182 |  
| With You | Huh Gak | ballad | 2023-04-28 | 223 |  
+-----+-----+-----+-----+-----+  
11 rows in set (0.00 sec)  
  
mysql> select *  
-> from Ranking;  
+-----+-----+-----+  
| song | singer | ranking |  
+-----+-----+-----+  
| Flower | JISOO | 10 |  
| I AM | IVE | 3 |  
| Kitsch | IVE | 5 |  
| London Boy | Lim Young Woong | 7 |  
| Love Always Run Away | Lim Young Woong | 9 |  
| Polaroid | Lim Young Woong | 8 |  
| Queencard | (G)I-DLE | 2 |  
| Spicy | aespa | 1 |  
| Super | SEVENTEEN | 6 |  
| UNFORGIVEN | LE SSERAFIM | 4 |  
| With You | Huh Gak | 11 |  
+-----+-----+-----+  
11 rows in set (0.00 sec)
```

execution result

1

```
Console ×  
<terminated> MainMenu [Java Application] C:\Users\samsung\p2\pool\pl  
##### Music Chart DB #####  
--- Please select a menu ---  
1. SelectMenu1  
2. SelectMenu2  
3. SelectMenu3  
4. InsertMenu  
5. UpdateMenu  
6. DeleteMenu  
0. exit  
menu:1  
DB connect success!  
Enter the location of the agency: Gangnam-gu, Seoul  
Singer names:  
Huh Gak  
LE SSERAFIM  
IVE
```

2

```
##### Music Chart DB #####  
--- Please select a menu ---  
1. SelectMenu1  
2. SelectMenu2  
3. SelectMenu3  
4. InsertMenu  
5. UpdateMenu  
6. DeleteMenu  
0. exit  
menu:2  
DB connect success!  
Enter the Singer to search: aespa  
Singer: aespa, Agency: SM Entertainment
```

3

```
##### Music Chart DB #####  
--- Please select a menu ---  
1. SelectMenu1  
2. SelectMenu2  
3. SelectMenu3  
4. InsertMenu  
5. UpdateMenu  
6. DeleteMenu  
0. exit  
menu:3  
DB connect success!  
가수:(G)I-DLE 음원 수:1  
가수:aespa 음원 수:1  
가수:Huh Gak 음원 수:1  
가수:IVE 음원 수:2  
가수:JISOO 음원 수:1  
가수:LE SSERAFIM 음원 수:1  
가수:Lim Young Woong 음원 수:3  
가수:SEVENTEEN 음원 수:1
```

Initial value

```
mysql> select * from Agencies;
+-----+-----+-----+
| name | location | established |
+-----+-----+-----+
| Big Planet Made Enter | Gangnam-gu, Seoul | 2021 |
| Cube Entertainment | Seongdong-gu, Seoul | 2013 |
| Mulgogi Music | Mapo-gu, Seoul | 2021 |
| Pledis Entertainment | Yongsan-gu, Seoul | 2010 |
| SM Entertainment | Seongdong-gu, Seoul | 1995 |
| Source Music | Gangnam-gu, Seoul | 2014 |
| Starship Entertainment | Gangnam-gu, Seoul | 2008 |
| YG Entertainment | Mapo-gu, Seoul | 1998 |
+-----+-----+-----+
8 rows in set (0.00 sec)
```

4execution

```
##### Music Chart DB #####
---- Please select a menu ----
1. SelectMenu1
2. SelectMenu2
3. SelectMenu3
4. InsertMenu
5. UpdateMenu
6. DeleteMenu
0. exit
menu:4
DB connect success!
Enter the name of the Agencies: JYP
Agencies inserted successfully.
```

result

```
mysql> select * from agencies;
+-----+-----+-----+
| name | location | established |
+-----+-----+-----+
| Big Planet Made Enter | Gangnam-gu, Seoul | 2021 |
| Cube Entertainment | Seongdong-gu, Seoul | 2013 |
| JYP | NULL | NULL |
| Mulgogi Music | Mapo-gu, Seoul | 2021 |
| Pledis Entertainment | Yongsan-gu, Seoul | 2010 |
| SM Entertainment | Seongdong-gu, Seoul | 1995 |
| Source Music | Gangnam-gu, Seoul | 2014 |
| Starship Entertainment | Gangnam-gu, Seoul | 2008 |
| YG Entertainment | Mapo-gu, Seoul | 1998 |
+-----+-----+-----+
```

Initial value

```
mysql> select * from Singers;
+-----+-----+-----+-----+
| name | agency | debut | numMembers |
+-----+-----+-----+-----+
| (G)I-DLE | Cube Entertainment | 2018 | 5 |
| aespa | SM Entertainment | 2020 | 4 |
| Huh Gak | Big Planet Made Enter | 2010 | 1 |
| IVE | Starship Entertainment | 2021 | 6 |
| JISOO | YG Entertainment | 2023 | 1 |
| LE SSERAFIM | Source Music | 2022 | 5 |
| Lim Young Woong | Mulgogi Music | 2016 | 1 |
| SEVENTEEN | Pledis Entertainment | 2015 | 13 |
+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

5execution

```
##### Music Chart DB #####
---- Please select a menu ----
1. SelectMenu1
2. SelectMenu2
3. SelectMenu3
4. InsertMenu
5. UpdateMenu
6. DeleteMenu
0. exit
menu:5
DB connect success!
Enter the name of the singer: aespa
Successfully updated numMembers for singer: aespa
Updated numMembers value: 5
```

result

```
mysql> select * from singers;
+-----+-----+-----+-----+
| name | agency | debut | numMembers |
+-----+-----+-----+-----+
| (G)I-DLE | Cube Entertainment | 2018 | 5 |
| aespa | SM Entertainment | 2020 | 5 |
| Huh Gak | Big Planet Made Enter | 2010 | 1 |
| IVE | Starship Entertainment | 2021 | 6 |
| JISOO | YG Entertainment | 2023 | 1 |
| LE SSERAFIM | Source Music | 2022 | 5 |
| Lim Young Woong | Mulgogi Music | 2016 | 1 |
| SEVENTEEN | Pledis Entertainment | 2015 | 13 |
+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

Initial value

```
mysql> select *  
-> from Songs;  
+-----+-----+-----+-----+-----+  
| title | singer | genre | releaseDate | length |  
+-----+-----+-----+-----+-----+  
| Flower | JISOO | dance | 2023-03-31 | 173 |  
| I AM | IVE | dance | 2023-04-10 | 183 |  
| Kitsch | IVE | dance | 2023-04-10 | 195 |  
| London Boy | Lim Young Woong | rock, metal | 2022-11-15 | 245 |  
| Love Always Run Away | Lim Young Woong | ballad | 2021-10-11 | 240 |  
| Polaroid | Lim Young Woong | rock, metal | 2022-11-15 | 207 |  
| QueenCard | (G)I-DLE | dance | 2023-05-15 | 161 |  
| Spicy | aespa | dance | 2023-05-08 | 197 |  
| Super | SEVENTEEN | dance | 2023-04-24 | 200 |  
| UNFORGIVEN | LE SSERAFIM | dance | 2023-05-01 | 182 |  
| With You | Huh Gak | ballad | 2023-04-28 | 223 |  
+-----+-----+-----+-----+-----+  
11 rows in set (0.00 sec)  
  
mysql> select *  
-> from Ranking;  
+-----+-----+-----+  
| song | singer | ranking |  
+-----+-----+-----+  
| Flower | JISOO | 10 |  
| I AM | IVE | 3 |  
| Kitsch | IVE | 5 |  
| London Boy | Lim Young Woong | 7 |  
| Love Always Run Away | Lim Young Woong | 9 |  
| Polaroid | Lim Young Woong | 8 |  
| QueenCard | (G)I-DLE | 2 |  
| Spicy | aespa | 1 |  
| Super | SEVENTEEN | 6 |  
| UNFORGIVEN | LE SSERAFIM | 4 |  
| With You | Huh Gak | 11 |  
+-----+-----+-----+  
11 rows in set (0.00 sec)
```

Execution

```
menu:6  
DB connect success!  
Please enter the title of the song you want to delete:I AM  
delete success!  
##### Music Chart DB #####  
---- Please select a menu ----  
1. SelectMenu1  
2. SelectMenu2  
3. SelectMenu3  
4. InsertMenu  
5. UpdateMenu  
6. DeleteMenu  
0. exit  
menu:0  
exit!
```

result

```
mysql> select * from songs;  
+-----+-----+-----+-----+-----+  
| title | singer | genre | releaseDate | length |  
+-----+-----+-----+-----+-----+  
| Flower | JISOO | dance | 2023-03-31 | 173 |  
| Kitsch | IVE | dance | 2023-04-10 | 195 |  
| London Boy | Lim Young Woong | rock, metal | 2022-11-15 | 245 |  
| Love Always Run Away | Lim Young Woong | ballad | 2021-10-11 | 240 |  
| Polaroid | Lim Young Woong | rock, metal | 2022-11-15 | 207 |  
| QueenCard | (G)I-DLE | dance | 2023-05-15 | 161 |  
| Spicy | aespa | dance | 2023-05-08 | 197 |  
| Super | SEVENTEEN | dance | 2023-04-24 | 200 |  
| UNFORGIVEN | LE SSERAFIM | dance | 2023-05-01 | 182 |  
| With You | Huh Gak | ballad | 2023-04-28 | 223 |  
+-----+-----+-----+-----+-----+  
10 rows in set (0.00 sec)  
  
mysql> select * from ranking order by ranking;  
+-----+-----+-----+  
| song | singer | ranking |  
+-----+-----+-----+  
| Spicy | aespa | 1 |  
| QueenCard | (G)I-DLE | 2 |  
| UNFORGIVEN | LE SSERAFIM | 3 |  
| Kitsch | IVE | 4 |  
| Super | SEVENTEEN | 5 |  
| London Boy | Lim Young Woong | 6 |  
| Polaroid | Lim Young Woong | 7 |  
| Love Always Run Away | Lim Young Woong | 8 |  
| Flower | JISOO | 9 |  
| With You | Huh Gak | 10 |  
+-----+-----+-----+  
10 rows in set (0.00 sec)
```

Ranking changes as 'I AM' song disappears

Team name: K251

김채린 (1898021), 오윤재(2171030), 김시온(2176060)

1. Creating dB, ER Diagram - 김시온
 2. Report creation, README text file, execution environment setting - 김채린
 4. DB idea conception, presentation (video presentation?) - 오윤재
-

Sharing 6 menus:

1. 2. 3. : 3select
 - 2 menus(queries) : ask for user input text, contain join, and use 'view' (!execute a single query!) - 김채린, 김시온
 - 1 menu(query) should contain aggregation, group by(!Execute a single query!) - 오윤재

4 insert: enter only the agency name (other null values are processed) - 김채린

5 update : Singer (Please enter the name of the SingerGrop you want to recruit. +1) - 김시온

6 delete : (song & ranking → delete together) Delete a song from song. Also deleted from the rankings - 오윤재