**프로젝트 4:**

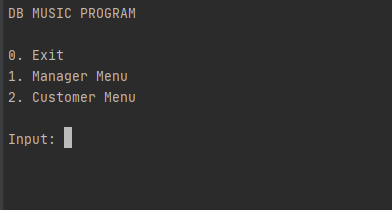
**DBMS를 활용한 프로그램 개발**

컴퓨터소프트웨어학부 2019067429 한승우

1. **프로그램 코드 설명**

* **타이머 기능**
* 쓰레드 타이머를 이용하여 일정 시간마다 모든 user의 남은 구독 일 수가 줄어들게 하였다.
* 만약, user의 남은 구독 날짜가 0이 되면 자동으로 프로그램에서 삭제되게 하였다.
* user와 관련한 playlist를 삭제할 때, playlist가 enrollment와의 fk관계가 있으므로 fk관계를 끊어 준 후 삭제를 진행한다.
* 사용한 query 문

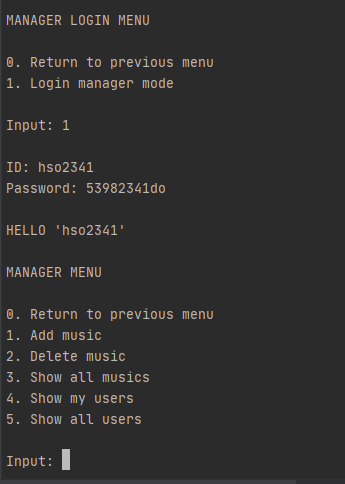
1. Update -> 일정 시간마다 user의 remain 값이 1씩 줄어들게 하였다.
2. Alter table drop -> enrollment와 playlist의 fk관계를 끊기 위해 사용
3. Delete -> playlist와 enrollment에 있는 user와 관련한 정보들을 삭제하기 위해 사용, 또한 user자체를 삭제하기 위해서도 사용
4. Alter table add -> enrollment와 playlist의 fk관계를 다시 이어주기 위해 사용



* **Manager**

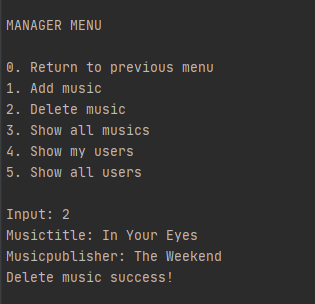
1. **로그인 기능**

* 매니저가 매니저 관련 메뉴에 접속할 수 있도록 로그인 기능을 넣어주었다.
* select exists를 이용해서, manager 테이블에 없으면 접근을 불가하게 하였다.
* 사용한 query 문 –

1. Select exists(select) -> 아이디가 존재하는지 확인하는 query
2. **음악 추가 기능**

* 매니저가 음원 추가를 담당할 수 있게 해주었다.
* 음원 번호는 현재 존재하는 음원의 번호 중 가장 큰 번호 +1 을 해주어서 음원 번호가 겹치지 않도록 하였다.
* 사용한 query 문

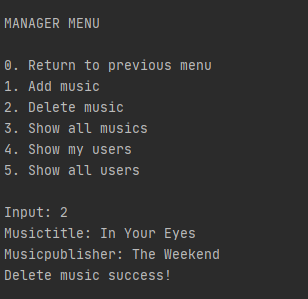
1. Select max(musicnum) from music -> 음원 번호를 선택하는 query
2. Insert into music -> 음원을 추가하는 query



1. **음악 삭제 기능**

* 매니저가 음원 삭제를 담당할 수 있게 해주었다.
* 이때, 음원은 플레이리스트와 enrollment table에 속해 있을 수 있으므로, fk관계를 끊어 준 후 삭제를 진행하였다.
* 사용한 query 문

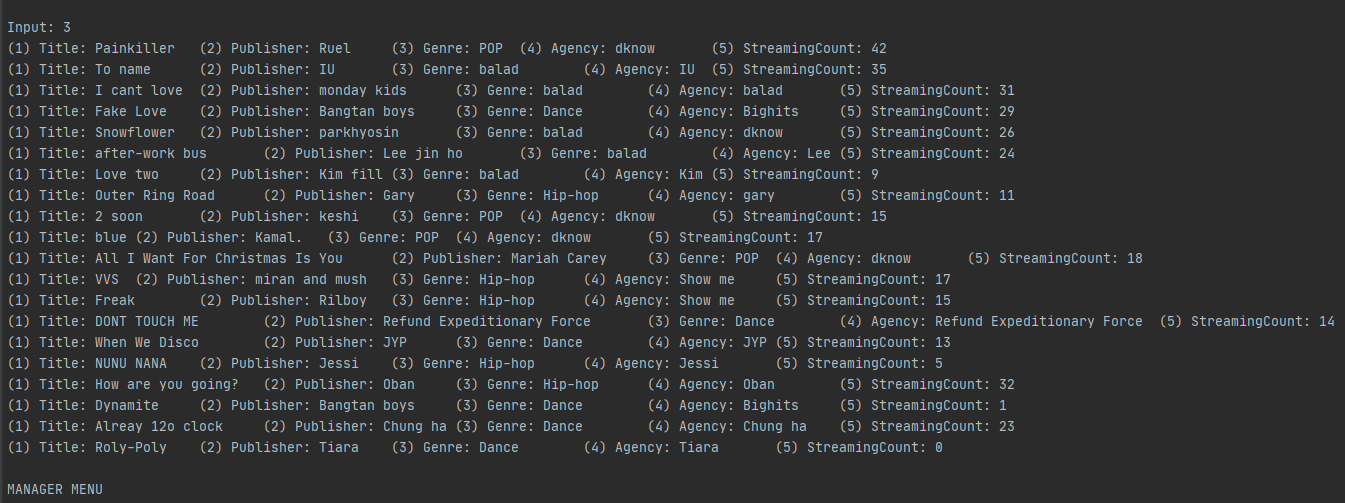
1. Select -> enrollment에 접근하기 위해 music 테이블에서 select
2. Alter table drop -> enrollment와 music의 fk관계를 끊는다.
3. Delete from music -> 음원 삭제
4. Delete from enrollment -> 플레이리스트에 포함되어 있으면 삭제
5. Alter table add -> enrollment와 music의 fk관계를 다시 잇는다.



1. **모든 음원 출력 기능**

* 매니저는 현재 존재하는 모든 음원을 볼 수 있다.
* 사용한 query 문

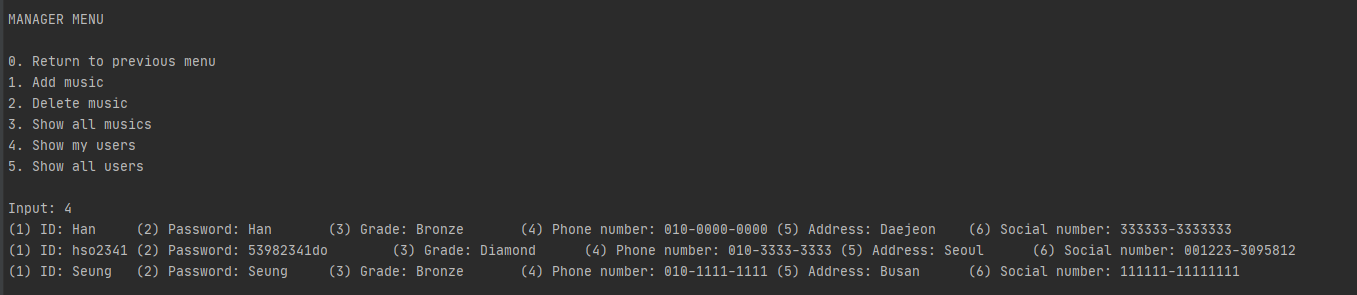
1. Select from music -> music table에서 모든 음원을 가져온다.



1. **자신이 담당하고 있는 user를 출력**

* 매니저는 자신이 담당하고 있는 user의 정보를 볼 수 있다.
* 사용한 query 문

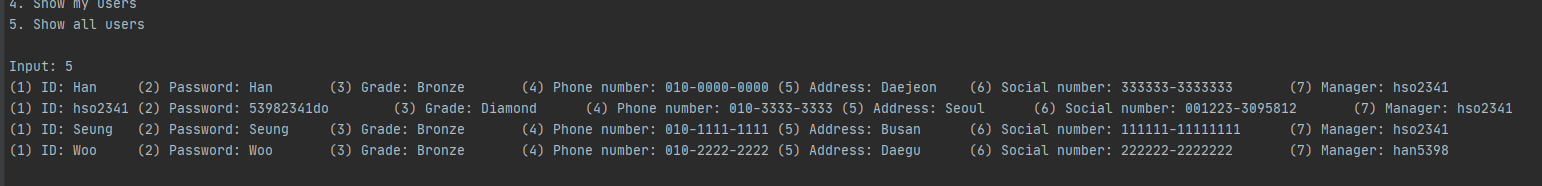
1. Select – where 현재 매니저 -> user 테이블에서 Mgr\_UID가 자신인 tuple을 가져온다.



1. **모든 user를 출력**

* 매니저는 현재 음원 관리 프로그램에 등록된 모든 user의 정보를 볼 수 있다.
* 사용한 query 문

1. Select \* -> user table에서 모든 user를 가져온다.

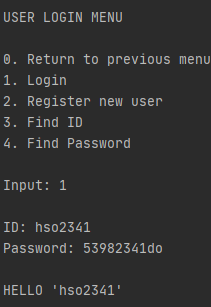


* **User 로그인 메뉴**

1. **로그인 기능**

* 사용자가 사용자 메뉴에 접근할 수 있도록 로그인 기능을 만들어 주었다.
* select exists를 이용해서, user 테이블에 없으면 접근을 불가하게 하였다.
* 사용한 query 문 –

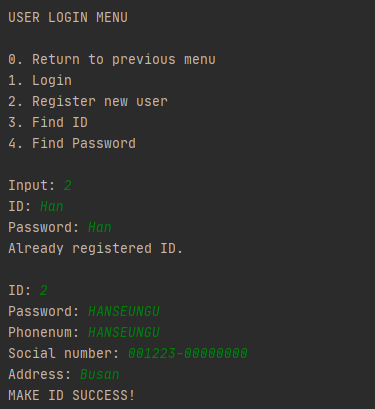
1. Select exists(select) -> 아이디가 존재하는지 확인하는 query



1. **새 사용자 등록 기능**

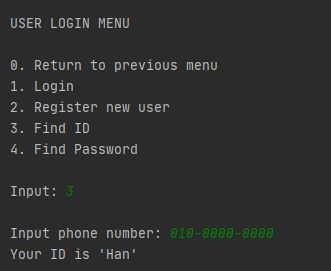
* 사용자 로그인 메뉴에서 새로운 사용자가 프로그램에 등록 할 수 있게 해 주었다.
* 이때, select exists를 사용해서, 이미 존재하는 아이디이면 등록 할 수 없게 하였다.
* 또한, 자신이 담당하는 매니저를 rand()함수를 이용해서, 랜덤으로 선택 되게 해주었다.
* 사용한 query 문

1. Select exists(select) -> 이미 존재하는 아이디가 있는지 확인하는 query
2. Select order by rand() -> 담당 매니저 랜덤 선택
3. Insert into -> user table에 생성한 user를 넣는다.



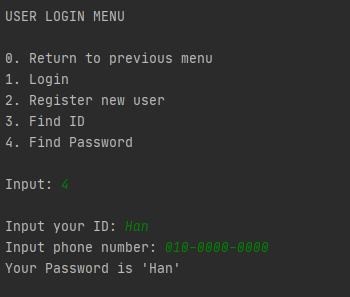
1. **아이디 찾기 기능**

* 이용자가 자신의 아이디를 잃어버렸을 때, 찾을 수 있게 하였다.
* 자신의 phone number을 매개체로 이용해서 아이디를 찾을 수 있도록 하였다.
* 사용한 query 문

1. Select -> phone number가 일치하는 유저 tuple을 반환한다. 
2. **비밀번호 찾기 기능**

* 이용자가 자신의 비밀번호를 잃어버렸을 때, 찾을 수 있도록 하였다.
* 자신의 ID와 phone number을 매개체로 비밀번호를 찾는다.
* 사용한 query 문

1. Select -> phone number와 ID가 일치하는 유저를 반환한다.

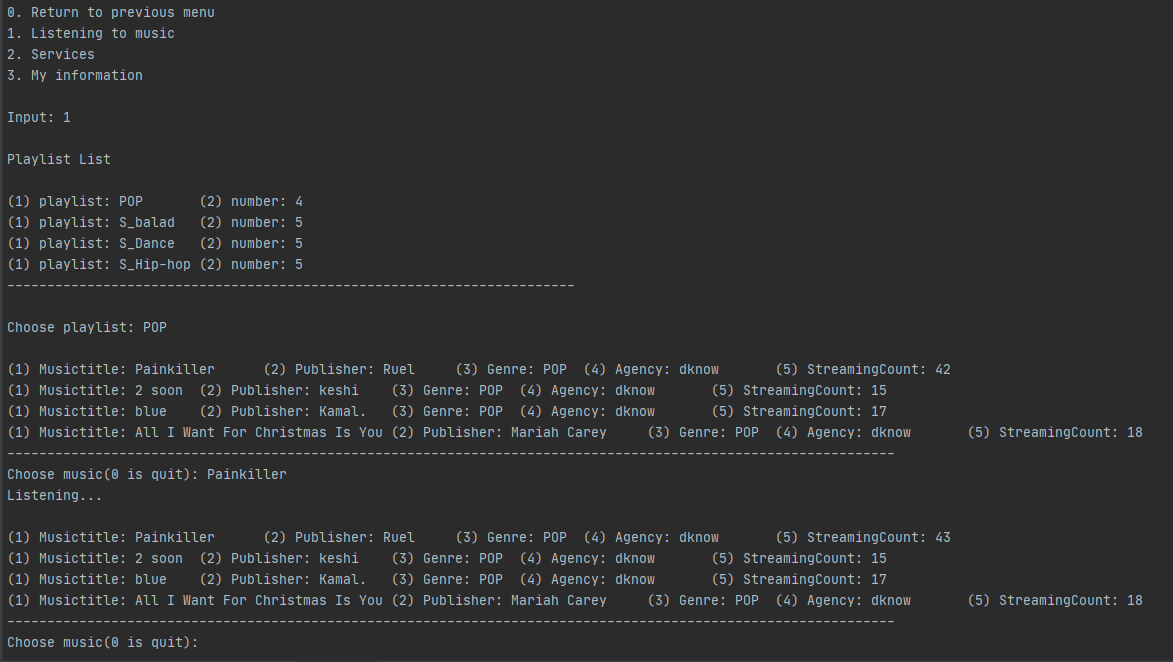


* **User 메인 메뉴**

1. **음악 듣기 기능**

* 사용자가 음악을 감상할 수 있도록 하였다.
* ‘사용자 자신’ 만을 한정하여 자신이 가지고 있는 플레이리스트 안에 속해 있는 음원을 들을 수 있도록 하였다.
* 음악 듣기는 2가지 단계로 구성된다.
* 플레이 리스트 고르기
* 음원 고르기 (반복)
* 음원을 성공적으로 감상 했으면, 해당 음원의 StreamingCount를 하나 늘린 후 계속해서 음원 고르는 메뉴를 출력한다.
* 사용한 query 문 –

1. Select -> 자신이 가지고 있는 플레이리스트를 출력하기 위해서 사용했다. 또한 프로그램에 존재하는 모든 음원을 출력하기 위해서 사용했다.
2. Update -> 음악 감상한 음원의 streamingcount를 하나 늘려주기 위해서 사용했다.

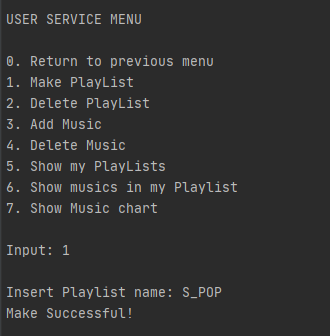


* **User 서비스 메뉴**

1. **플레이 리스트 만들기 기능**

* 사용자가 자신의 playlist를 생성할 수 있도록 하였다.
* 사용한 query 문 –

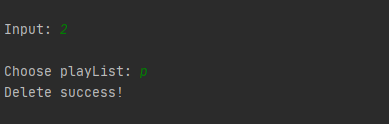
1. Insert into -> playlist table에 새로 만든 playlist를 넣어주기 위해 사용



1. **플레이 리스트 삭제 기능**

* 사용자가 자신의 playlist를 삭제할 수 있도록 하였다.
* 이때, playlist table과 enrollment table은 fk관계를 가지므로 fk관계를 끊어 준 후 삭제를 진행하였다.
* 사용한 query 문 –

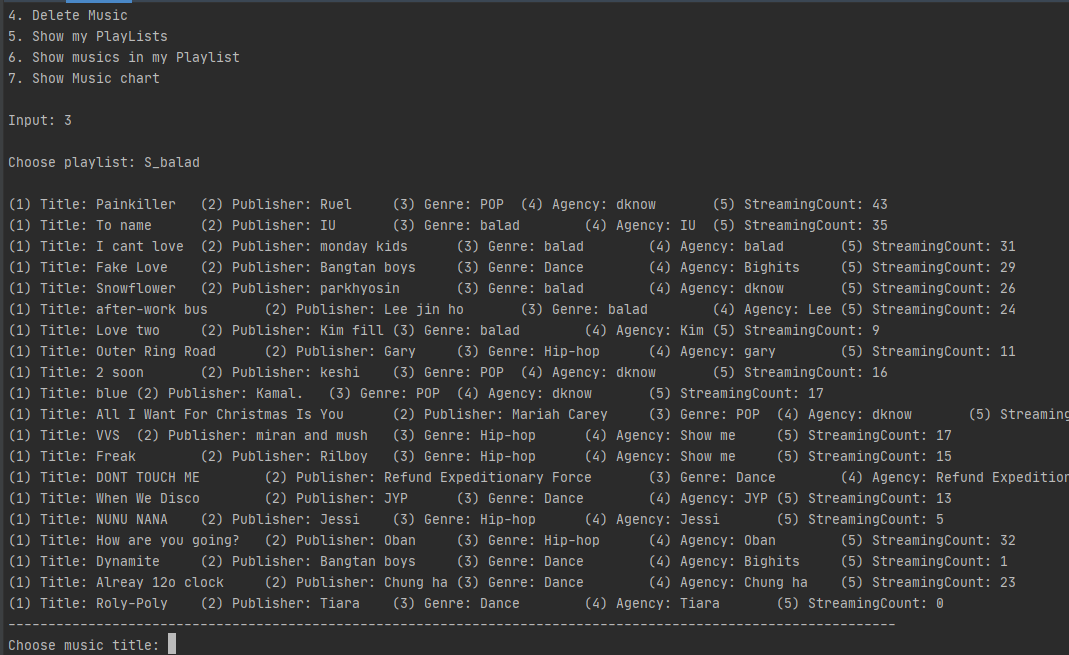
1. Alter table drop -> playlist와 enrollment 간의 fk관계를 끊어주기 위해 사용
2. Delete -> playlist에 포함되어 있는 음원과, playlist를 삭제하기 위해 사용
3. Select -> playlist table에서 삭제할 playlist의 tuple을 가져오기 위해 사용
4. Alter table add -> playlist와 enrollment 간의 fk관계를 다시 이어주기 위해 사용



1. **플레이 리스트에 음악을 추가하는 기능**

* 사용자가 자신의 playlist에 음원을 추가할 수 있도록 하였다.
* 이미 플레이리스트에 존재하는 음원이면 추가 할 수 없도록 하였다.
* 음원 추가 기능은 2단계로 구성된다.
  + - 플레이 리스트 선택
    - 음원 선택
* 성공적으로 플레이 리스트에 음원이 추가되면, playlist에 속해있는 음원의 개수를 의미하는 musicnumber을 하나 늘려주었다.
* 사용한 query 문 –

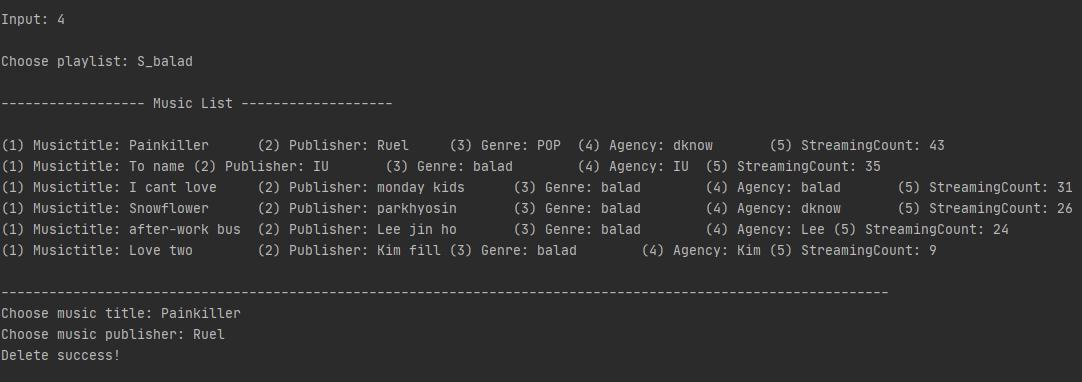
1. Select -> music table과 playlist에 존재하는 attribute인지 확인하기 위해 사용
2. Insert -> music과 playlist의 관계를 표현하는 table인 enrollment에 tuple을 추가하기 위해 사용
3. Update -> playlist에 속해있는 음원의 개수를 의미하는 musicnumber을 1 늘려주기 위해 사용



1. **플레이 리스트에 있는 음악을 삭제하는 기능**

* 사용자가 자신의 playlist에 속해 있는 음원을 삭제할 수 있도록 하였다.
* 음악 삭제 기능은 2가지 단계로 구성된다.
  + - 플레이 리스트 선택
    - 음원 선택
* 이때, 음원은 자신이 선택한 playlist에 속해 있는 음원만 보여주면 된다.
* 삭제 후 musicnumber을 하나 줄여준다.
* 사용한 query 문 –

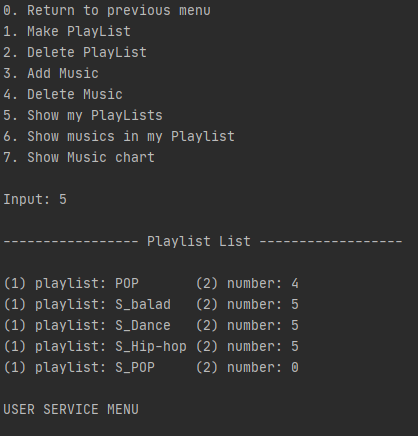
1. Select -> music table과 playlist table에 속해 있는 attribute인지 확인하기 위해 사용
2. Delete -> enrollment table에서 tuple을 삭제하기 위해 사용
3. Update -> 삭제 후 musicnumber을 1 줄이기 위해 사용



1. **플레이 리스트 출력 기능**

* 사용자가 자신의 playlist 목록을 볼 수 있게 하였다.
* 사용한 query 문 –

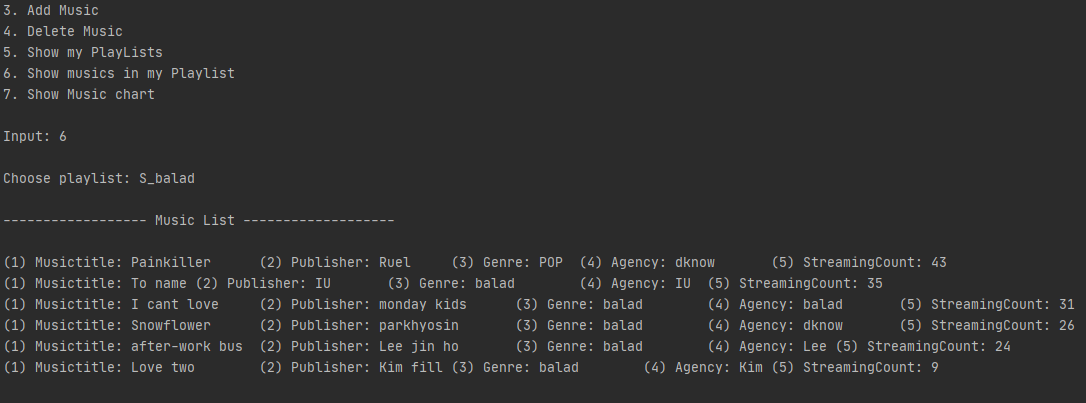
1. Select -> 자신에게 속해 있는 playlist를 출력하기 위해 사용



1. **플레이 리스트에 속해 있는 음원 출력**

* 사용자가 자신의 playlist에 속해 있는 음원 목록을 볼 수 있게 하였다.
* Playlist 이름을 입력 받은 후 해당 playlist에 속해 있는 음원의 목록을 보여준다.
* 사용한 query 문 –

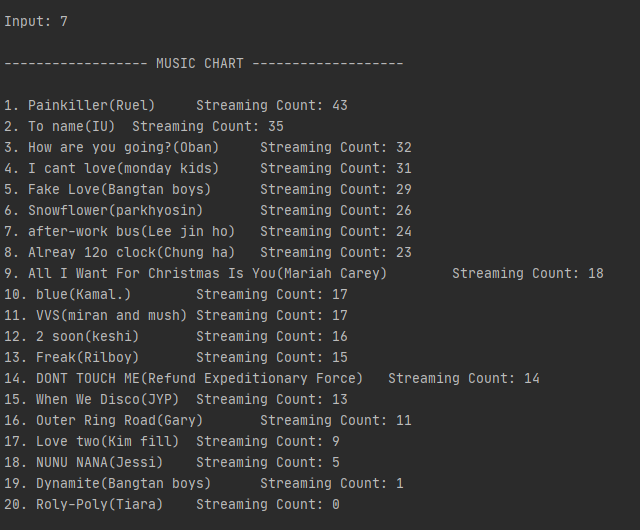
1. Select -> playlist table에 존재하는 playlist이름인지 확인하기 위해 사용 또한, 포함 되어 있는 음원들을 반복해서 출력하기 위해 사용



1. **음원 차트**

* 사용자는 프로그램에 존재하는 음원들의 순위를 볼 수 있다.
* 이때, 음원들은 스트리밍 된 횟수를 기준으로 순위가 매겨진다.
* 사용한 query 문 –

1. Select order -> streamingcount를 기준으로 음원들을 반복해서 출력하기 위해 사용

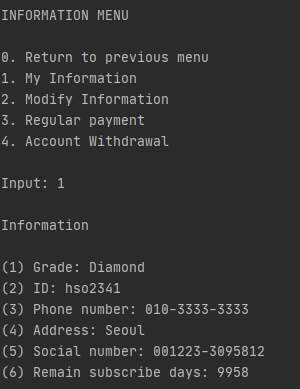


* **User 정보 메뉴**

1. **사용자 정보 출력 기능**

* 사용자가 자신의 정보를 볼 수 있도록 하였다.
* 사용한 query 문 –

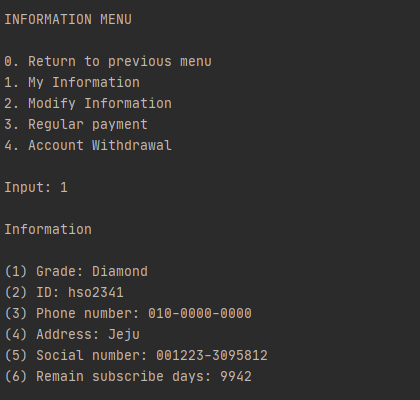
1. Select -> 사용자가 자신의 정보를 볼 수 있도록 하기 위해서 사용



1. **사용자 정보 변경 기능**

* 사용자가 자신의 정보를 변경 할 수 있도록 하였다.
* Phone number과 Address 그리고 비밀번호를 바꿀 수 있도록 하였다.
* 사용한 query 문 –

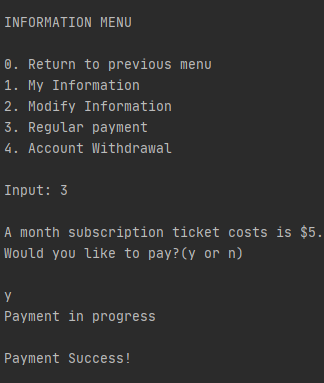
1. Update -> 자신의 정보를 바꾸어 주기 위해 사용



1. **정기 구독권 구매 기능**

* 사용자가 프로그램을 계속 이용하기 위해서 한달 구독권을 구매 할 수 있도록 하는 기능이다.
* 정기 구독권을 구매한 횟수에 따라서 등급이 정해지는데, 1~6번은 Bronze 6~11번은 Silver, 12~35번은 Gold, 35~59번은 Platinum, 60번 이상은 Diamond 등급으로 구분 하였다.
* 사용한 query 문 –

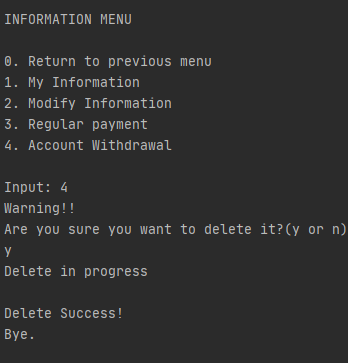
1. Select -> 현재 어떤 유저가 접속해 있는지 확인하기 위해 사용
2. Update -> user table의 remain 값과 grade 값을 재설정 하기 위해 사용



1. **사용자 계정 삭제 기능**

* 사용자가 자신의 계정을 삭제 할 수 있도록 하였다.
* user와 관련한 playlist를 삭제할 때, playlist가 enrollment와의 fk관계가 있으므로 fk관계를 끊어 준 후 삭제를 진행한다.
* 사용한 query 문 –

1. Alter table drop -> enrollment와 playlist의 fk관계를 끊기 위해 사용
2. Delete -> playlist와 enrollment에 있는 user와 관련한 정보들을 삭제하기 위해 사용, 또한 user자체를 삭제하기 위해서도 사용
3. Alter table add -> enrollment와 playlist의 fk관계를 다시 이어주기 위해 사용



1. **변경 사항**
2. **정기 구독**

* 정기 구독권 항목을 추가하였다. 남은 구독 일 수 는 쓰레드로 인해 자동으로 줄어들며, user table에 새로운 attribute로 추가해 주었다.

1. **차트**

* 플레이 리스트 차트를 삭제하였다. 사용자가 다른 사용자가 가지고 있는 플레이 리스트를 사용한 다는 것이 모순이므로 필요 없다고 생각했다.
* 음원 차트 테이블을 삭제했다. 음원 table을 이용해서 구현 할 수 있는 기능이므로 필요 없는 table이다.

1. **구독 table**

* 구독 테이블을 삭제했다. 사용자는 플레이 리스트를 이용하여 음악 감상을 하는데, 사용자와 음원 간의 fk관계가 필요 없기 때문이다.

1. **음원 table**

* 음원 table의 pk를 바꾸어 주었다. 음원 제목을 pk로 두게 되면 음원 제목이 같다면, 추가 될 수 없기 때문에 음원 번호 항목을 만들어 구별해 주었다.