

# DB\_Project\_4



Database Systems

DBMS 프로그램 개발

2016025469

서건식

- 개발 환경

Windows 10 x64 / Python 3 / IDLE 로 개발하였습니다.

- 구현한 기능

```
-----
0. 종료
1. 사용자 로그인
2. 관리자 로그인
3. 사용자 회원가입
4. 사용자 회원탈퇴
5. 관리자 등록
-----
Input:
```

처음 프로그램을 실행하면 나오는 화면 입니다.

메뉴별로 구현사항을 설명하겠습니다.

1. 사용자 로그인

```
Input: 1
아이디: id
비밀번호: pw
```

```
-----
플레이리스트 목록:
```

```
p : EDM
1 : 발라드
```

```
-----
0. 이전화면으로
1. 전체 음악 순위 보기
2. 플레이리스트에 음악 넣기
3. 플레이리스트에서 음악 삭제
4. 플레이리스트 열기
5. 플레이리스트 공유하기
6. 공유받은 플레이리스트 보기
7. 이용권 구매하기
8. 구매한 이용권 보기
-----
Input:
```

사용자 로그인을 누르면 아이디와 비밀번호를 입력해야 합니다. 입력할 시 사용자 메뉴를 출력합니다.

현재 자신이 소유한 플레이리스트 목록을 보여줍니다.

```
Input: 1
아이디: qwf
비밀번호: waf
존재하지 않는 아이디와 비밀번호 입니다.
```

없는 아이디와 비밀번호 일시 로그인이 불가능 합니다.

기능들은 코드 description 에서 자세히 설명하고, 기능 수행 화면만 스크린샷을 첨부하겠습니다.

- 전체 음악 순위 보기

```

Input: 1
-----
정보 : 순위 / 제목 / 아티스트 / 장르 / 플레이시간 / 발매일자 / 앨범제목 / 뮤직비디오
-----
0 : 1 | 포장마차 | 황인옥 | 발라드 | 4분10초 | 2019-04-22 | 포장마차 | https://www.youtube.com/watch?v=Sfsf3WdJ8xo |
1 : 2 | 이별행동 | 이우 | 발라드 | 3분22초 | 2019-08-21 | 이우 | https://www.youtube.com/watch?v=D1PvIWdJ124 |
2 : 3 | Blueming | IU | 락 | 3분28초 | 2019-11-30 | Love Poem | https://www.youtube.com/watch?v=D1PvIWdJ8xo |
3 : 4 | 늦은 밤 너의 집 앞 골목길에서 | 노을 | 발라드 | 4분20초 | 2019-11-07 | 디지털싱글 | https://www.youtube.com/watch?v=D1P

```

- 플레이리스트에 음악 넣기

```

플레이리스트 이름을 입력하세요: 락
-----
전체 음악 목록:
-----
정보 : 제목 / 아티스트 / 장르 / 플레이시간 / 발매일자 / 앨범제목 / 뮤직비디오 / 순위
-----
0 : 포장마차 | 황인옥 | 발라드 | 4분10초 | 2019-04-22 | 포장마차 | https://www.youtube.com/watch?v=Sfsf3WdJ8xo | 1 |
1 : 이별행동 | 이우 | 발라드 | 3분22초 | 2019-08-21 | 이우 | https://www.youtube.com/watch?v=D1PvIWdJ124 | 2 |
2 : Blueming | IU | 락 | 3분28초 | 2019-11-30 | Love Poem | https://www.youtube.com/watch?v=D1PvIWdJ8xo | 3 |
3 : 늦은 밤 너의 집 앞 골목길에서 | 노을 | 발라드 | 4분20초 | 2019-11-07 | 디지털싱글 | https://www.youtube.com/watch?v=D1P
-----

플레이리스트에 담긴 음악 목록:
-----
정보 : 제목 / 아티스트 / 장르 / 플레이시간 / 발매일자 / 앨범제목 / 뮤직비디오 / 순위
-----
0 : Blueming | IU | 락 | 3분28초 | 2019-11-30 | Love Poem | https://www.youtube.com/watch?v=D1PvIWdJ8xo | 3 |
-----

입력을 취소하시려면 exit을 눌러주세요.
넣을 음악의 제목을 선택하세요: Blueming
음악이 이미 존재합니다.

입력을 취소하시려면 exit을 눌러주세요.
넣을 음악의 제목을 선택하세요: ..?
음악이 존재하지 않습니다.

입력을 취소하시려면 exit을 눌러주세요.
넣을 음악의 제목을 선택하세요: 이별행동

```

플레이리스트 이름을 입력하면 만약 존재하지 않는 플레이리스트면 입력받은 값으로 새로 생성하고, 존재하는 플레이리스트면 기존 플레이리스트에 음악을 추가합니다.

전체 음악 목록을 보여주고, 선택한 플레이리스트에 담긴 음악도 보여줍니다.

입력을 넣을 때 exit을 넣으면 넣기를 종료합니다.

이미 들어있는 음악의 제목이거나 존재하지 않는 음악을 입력할 시에 입력이 되지 않습니다.

전체음악목록에 있는 음악의 제목을 선택할 시 플레이리스트에 음악이 삽입됩니다.

결과:

-----  
플레이리스트에 담긴 음악 목록:

정보 : 제목 / 아티스트 / 장르 / 플레이시간 / 발매일자 / 앨범제목 / 뮤직비디오 / 순위

0 : 이별행동 | 이우 | 발라드 | 3분22초 | 2019-08-21 | 이우 | <https://www.youtube.com/watch?v=D1PvIWdJ124> | 2 |  
1 : Blueming | IU | 락 | 3분28초 | 2019-11-30 | Love Poem | <https://www.youtube.com/watch?v=D1PvIWdJ8xo> | 3 |  
-----

## ● 플레이리스트에서 음악 삭제

Input: 3  
입력을 취소하시려면 exit을 눌러주세요.  
플레이리스트 이름을 입력하세요: 팝  
존재하지 않는 플레이리스트입니다.

-----  
입력을 취소하시려면 exit을 눌러주세요.  
플레이리스트 이름을 입력하세요: 락

플레이리스트에 담긴 음악 목록:

-----  
정보 : 제목 / 아티스트 / 장르 / 플레이시간 / 발매일자 / 앨범제목 / 뮤직비디오 / 순위

0 : 이별행동 | 이우 | 발라드 | 이우 | <https://www.youtube.com/watch?v=D1PvIWdJ124> | 2 |  
1 : Blueming | IU | 락 | Love Poem | <https://www.youtube.com/watch?v=D1PvIWdJ8xo> | 3 |  
-----

입력을 취소하시려면 exit을 눌러주세요.  
삭제할 음악의 제목을 선택하세요: 포장마차  
음악이 존재하지 않습니다.

플레이리스트에 담긴 음악 목록:

-----  
정보 : 제목 / 아티스트 / 장르 / 플레이시간 / 발매일자 / 앨범제목 / 뮤직비디오 / 순위

0 : 이별행동 | 이우 | 발라드 | 이우 | <https://www.youtube.com/watch?v=D1PvIWdJ124> | 2 |  
1 : Blueming | IU | 락 | Love Poem | <https://www.youtube.com/watch?v=D1PvIWdJ8xo> | 3 |  
-----

입력을 취소하시려면 exit을 눌러주세요.  
삭제할 음악의 제목을 선택하세요: 이별행동  
음악을 플레이리스트에서 삭제하였습니다.

-----  
입력을 취소하시려면 exit을 눌러주세요.  
플레이리스트 이름을 입력하세요: exit

존재하지 않는 플레이리스트는 삭제할 수 없습니다.

플레이리스트에 담긴 음악이 아니면 존재하지 않는다고 출력합니다.

존재하는 음악일시 삭제합니다.

결과:

-----  
플레이리스트에 담긴 음악 목록:

정보 : 제목 / 아티스트 / 장르 / 플레이시간 / 발매일자 / 앨범제목 / 뮤직비디오 / 순위

0 : Blueming | IU | 락 | 3분28초 | 2019-11-30 | Love Poem | <https://www.youtube.com/watch?v=D1PvIWdJ8xo> | 3 |  
-----

## ● 플레이리스트 열기

```
Input: 4
입력을 취소하시려면 exit을 눌러주세요.
플레이리스트 이름을 입력하세요: 팝
잘못된 값입니다.
입력을 취소하시려면 exit을 눌러주세요.
플레이리스트 이름을 입력하세요: 락
```

플레이리스트에 담긴 음악 목록:

정보 : 제목 / 아티스트 / 장르 / 플레이시간 / 발매일자 / 앨범제목 / 뮤직비디오 / 순위

0 : Blueming | IU | 락 | 3분28초 | 2019-11-30 | Love Poem | <https://www.youtube.com/watch?v=D1PvIWdJ8xo> | 3 |

없는 이름을 입력하면 잘못된 값이라고 출력합니다.

플레이리스트에 담긴 음악을 출력해줍니다.

- 플레이리스트 공유하기

```
Input: 5
공유할 사용자의 ID를 입력하세요: fefefqw
존재하지 않는 ID입니다.
공유할 사용자의 ID를 입력하세요: lqeq2328
공유를 완료하였습니다.
```

데이터베이스 내에 존재하는 아이디에 한해서 자신의 플레이리스트를 공유 할 수 있습니다.

lqeq2328 아이디로 접속해보겠습니다.

```
Input: 1
아이디: lqeq2328
비밀번호: 1
```

-----  
플레이리스트 목록:

-----  
0. 이전화면으로  
1. 전체 음악 순위 보기  
2. 플레이리스트에 음악 넣기  
3. 플레이리스트에서 음악 삭제  
4. 플레이리스트 열기  
5. 플레이리스트 공유하기  
6. 공유받은 플레이리스트 보기  
7. 이용권 구매하기  
8. 구매한 이용권 보기  
-----

```
Input: 6
```

-----  
플레이리스트 목록:

0 : EDM |

1 : 발라드 |

2 : 락 |

입력을 취소하시려면 exit을 눌러주세요.  
플레이리스트 이름을 입력하세요: |

Id 아이디의 플레이리스트를 공유 받았음을 알 수 있습니다.

Input: 6

-----  
플레이리스트 목록:

0 : EDM |

1 : 발라드 |

2 : 락 |

입력을 취소하시려면 exit을 눌러주세요.

플레이리스트 이름을 입력하세요: df

잘못된 값입니다.

입력을 취소하시려면 exit을 눌러주세요.

플레이리스트 이름을 입력하세요: df

잘못된 값입니다.

입력을 취소하시려면 exit을 눌러주세요.

플레이리스트 이름을 입력하세요: gea

잘못된 값입니다.

입력을 취소하시려면 exit을 눌러주세요.

플레이리스트 이름을 입력하세요: 발라드

-----  
플레이리스트에 담긴 음악 목록:

-----  
정보 : 제목 / 아티스트 / 장르 / 플레이시간 / 발매일자 / 앨범제목 / 뮤직비디오 / 순위

0 : 포장마차 | 황인욱 | 발라드 | 4분10초 | 2019-04-22 | 포장마차 | <https://www.youtube.com/watch?v=Sfsf3WdJ8xo> | 1 |

1 : 이별행동 | 이우 | 발라드 | 3분22초 | 2019-08-21 | 이우 | <https://www.youtube.com/watch?v=D1PvIWdJ124> | 2 |

공유받은 플레이리스트 내에 존재하는 이름을 입력하면 그 플레이리스트에 들어있는 음악을 볼 수 있습니다.

### ● 이용권 구매하기

-----  
이용권:

0 : 스트리밍 | 1m | streaming | 30만원 |

1 : 정기권 | 3m | all | 44322232 |

-----  
입력을 취소하시려면 exit을 눌러주세요.

이용권 이름을 입력하세요: 다운로드

잘못된 값입니다.

입력을 취소하시려면 exit을 눌러주세요.

이용권 이름을 입력하세요: 스트리밍

구매를 완료했습니다.

### ● 구매한 이용권 보기

8. 구매한 이용권 보기

-----  
Input: 8

-----  
이용권 목록:

-----  
정보 : 이용권이름 / 기간 / 허용서비스 / 가격

-----  
0 : 스트리밍 | 1m | streaming | 30만원 |

## 2. 관리자 로그인

```
Input: 0
-----
0. 종료
1. 사용자 로그인
2. 관리자 로그인
3. 사용자 회원가입
4. 사용자 회원탈퇴
5. 관리자 등록
-----

Input: 2
관리자 식별 ID를 입력하세요: 1
-----

0. 이전 화면으로
1. 음악 추가하기
2. 음악 삭제하기
3. 이용권 추가하기
4. 이용권 삭제하기
5. 부서 추가하기
6. 부서 정보 출력
7. 사원 정보 출력
-----

Input:
```

관리자로 로그인 했을 때의 화면입니다.

### ● 음악 추가하기

```
-----

Input: 1
입력을 취소하시려면 exit을 입력하세요.
추가할 음악의 정보를 입력합니다.
제목: 어떻게 이별까지 사랑하겠어, 널 사랑하는거지
입력을 취소하시려면 exit을 입력하세요.
음악 ID: 5
입력을 취소하시려면 exit을 입력하세요.
순위: 5
아티스트 이름: 악동뮤지션
장르: 발라드
가사: 어떻게...
플레이 시간 양식: m분ss초
플레이시간: 3분10초
발매일자 양식: yyyy-mm-dd
발매일자: 2019-09-15
앨범 제목: 항해
뮤직비디오 링크: https://www.youtube.com/results?search_query=%EC%96%B4%EB%96%BB%EA%B2%8C+%
입력에 성공하였습니다.
입력을 취소하시려면 exit을 입력하세요.
추가할 음악의 정보를 입력합니다.
제목: 안녕
입력을 취소하시려면 exit을 입력하세요.
음악 ID: 5
존재하는 음악 ID 입니다. 다시 입력하세요.
입력을 취소하시려면 exit을 입력하세요.
음악 ID: 6
입력을 취소하시려면 exit을 입력하세요.
순위: 5
존재하는 순위 입니다. 다시 입력하세요.
입력을 취소하시려면 exit을 입력하세요.
순위: exit
입력을 취소하시려면 exit을 입력하세요.
추가할 음악의 정보를 입력합니다.
제목: exit
```

음악ID나 순위가 겹치면 다시 입력해야 합니다.

music.음악: 5 행 (총) (대략적) 

» 다음 

모두 보기 정렬 열 (11/1)

제목	아티스트	장르	가사	플레이시간	등록일자	발매일자	앨범제목	유튜브비디오	음악ID	순위
포장마차	황인욱	발라드	그대와 자주 가던 그 술집에 혼자 널 생각하며 소주 한...	4분10초	2019-11-27	2019-04-22	포장마차	https://www.youtube.com/watch?v=Sfs3Wd1...	1	1
이별행동	이우	발라드	다시금 날 봐줘요	3분22초	2019-11-29	2019-08-21	이우	https://www.youtube.com/watch?v=D1PvIWd1...	2	2
Blueming	IU	락	Love~	3분28초	2019-12-02	2019-11-30	Love Poem	https://www.youtube.com/watch?v=D1PvIWd1...	3	3
늦은 밤 너의 집 앞 골목길에서	노을	발라드	늦은밤~	4분20초	2019-12-02	2019-11-07	디지털싱글	https://www.youtube.com/watch?v=D1PvIWd1...	4	4
어떻게 이별까지 사랑하겠어, 널 사랑하는거지	악동뮤지션	발라드	일부러 몇 발자국 물러나 내가 혼자 걷는 널 바라...	3분20초	2019-12-02	2019-09-25	항해	https://www.youtube.com/watch?v=SI368DNYZFo	5	5

데이터베이스에 입력되었습니다.

● 음악 삭제하기

전체 음악 목록:

정보 : 제목 / 음악 ID

0 : 포장마차 | 1 |  
1 : 이별행동 | 2 |  
2 : Blueming | 3 |  
3 : 늦은 밤 너의 집 앞 골목길에서 | 4 |  
4 : 어떻게 이별까지 사랑하겠어, 널 사랑하는거지 | 5 |

입력을 취소하시려면 exit을 눌러주세요.  
삭제할 음악의 음악 ID를 입력하세요: 5  
음악을 삭제하였습니다.

전체 음악 목록:

정보 : 제목 / 음악 ID

0 : 포장마차 | 1 |  
1 : 이별행동 | 2 |  
2 : Blueming | 3 |  
3 : 늦은 밤 너의 집 앞 골목길에서 | 4 |

입력을 취소하시려면 exit을 눌러주세요.  
삭제할 음악의 음악 ID를 입력하세요: exit

music.음악: 4 행 (총) (대략적) 

»

제목	아티스트	장르	가사	플레이시간	등록일자	발매일자	앨범제목	유튜브비디오
포장마차	황인욱	발라드	그대와 자주 가던 그 술집에 혼자 널 생각하며 소주 한...	4분10초	2019-11-27	2019-04-22	포장마차	https://www.youtul
이별행동	이우	발라드	다시금 날 봐줘요	3분22초	2019-11-29	2019-08-21	이우	https://www.youtul
Blueming	IU	락	Love~	3분28초	2019-12-02	2019-11-30	Love Poem	https://www.youtul
늦은 밤 너의 집 앞 골목길에서	노을	발라드	늦은밤~	4분20초	2019-12-02	2019-11-07	디지털싱글	https://www.youtul

음악ID를 입력하면 음악을 삭제합니다.



전체 음악 목록:

-----  
정보 : 제목 / 음악 ID  
-----

0 : 포장마차 | 1 |

1 : 이별행동 | 2 |

2 : Blueming | 3 |

3 : 늦은 밤 너의 집 앞 골목길에서 | 4 |

-----  
입력을 취소하려면 exit을 눌러주세요.  
삭제할 음악의 음악 ID를 입력하세요: 66  
존재하지 않는 음악 ID입니다.

없는 ID를 입력 시 삭제하지 않습니다.

### ● 이용권 추가하기

```
Input: 3
입력을 취소하려면 exit을 입력하세요.
이용권 이름을 입력하세요:다운로드
이벤트 유무를 0또는 1로 입력하세요: 0
기간을 입력하세요. ex: 2달이면 2m, 1주면 1w입력: 1w
허용서비스를 입력하세요.
(streaming, download, all) 중 하나: download
가격을 숫자만 입력하세요. 단위: won: 3만원
입력을 완료하였습니다.
```

### ● 이용권 삭제하기

```
Input: 4
입력을 취소하려면 exit을 입력하세요.
이용권 이름을 입력하세요:a a a a
존재하지 않는 이용권 이름 입니다.
입력을 취소하려면 exit을 입력하세요.
이용권 이름을 입력하세요:다운로드
삭제를 완료하였습니다.
```

### ● 부서 추가

```
Input: 5
등록을 취소하려면 exit을 입력하세요.
부서 번호를 입력하세요: 4
부장으로 임명할 사원의 관리자 ID를 입력하세요: 1
부서 이름을 입력하세요: 고객관리

입력을 그만하려면 exit을 입력하세요. 최대 10개까지 입력 가능합니다.
부서의 위치를 입력하세요: 왕십리

입력을 그만하려면 exit을 입력하세요. 최대 10개까지 입력 가능합니다.
부서의 위치를 입력하세요: 판교

입력을 그만하려면 exit을 입력하세요. 최대 10개까지 입력 가능합니다.
부서의 위치를 입력하세요: 세종

입력을 그만하려면 exit을 입력하세요. 최대 10개까지 입력 가능합니다.
부서의 위치를 입력하세요: exit

입력을 그만하려면 exit을 입력하세요. 최대 10개까지 입력 가능합니다.
관리 영역을 입력하세요: 고객응대

입력을 그만하려면 exit을 입력하세요. 최대 10개까지 입력 가능합니다.
관리 영역을 입력하세요: A/S

입력을 그만하려면 exit을 입력하세요. 최대 10개까지 입력 가능합니다.
관리 영역을 입력하세요: exit
입력을 완료하였습니다.
```

위치와 관리영역은 multivalued attribute입니다.

```
Input: 5
등록을 취소하시려면 exit을 입력하세요.
부서 번호를 입력하세요: 4
이미 존재하는 부서 번호 입니다.
```

존재하는 번호는 저장하지 않습니다

부서

호스트: 127.0.0.1    데이터베이스: music

music.부서: 4 행 (중) (내)    >> 다음    << 모두 보

식별번호	부서이름	부서_관리자ID
1	인사	1
2	백엔드	2
3	정훈	3
4	고객관리	4

부서위치

호스트: 127.0.0.1    데이터베이스: m

music.부서위치: 7 행 (중)    >> 다음    << 모두 보

부서_식별번호	부서위치
1	왕십리
2	이천
2	판교
3	왕십리
3	이천
4	세종
4	판교

관리영역

호스트: 127.0.0.1    데이터베이스: music

music.관리영역: 7 행 (중)    >> 다음    << 모두 보

부서_식별번호	관리영역
1	채용
2	데이터관리
2	데이터유지보수
3	관리
3	교육
4	A/S
4	고객응대

● 부서 정보 출력

```

5. 부서 추가하기
6. 부서 정보 출력
7. 사원 정보 출력
-----
Input: 6
정보 : 식별번호 / 부서이름 / 부장이름 / 부서위치 / 관리영역
-----
0 : 1 | 인사 | 이영석 | 왕십리 | 채용
1 : 2 | 백엔드 | 서건식 | 이천 판교 | 데이터관리 데이터유지보수
2 : 3 | 정훈 | 조동훈 | 왕십리 이천 | 관리 교육
3 : 4 | 고객관리 | 이미자 | 세종 판교 | A/S 고객응대
  
```

● 사원 정보 출력

```

5. 부서 추가하기
6. 부서 정보 출력
7. 사원 정보 출력
-----
Input: 7
정보 : 이름 / 성별 / 주민등록번호 / 식별 ID / 부서이름 / 도로명 / 건물번호 / 핸드폰번호
-----
0 : 이영석 | 남자 | 939884-3342332 | 1 | 인사 | 도선동 | 30 | 010-2234-1422 010-5522-1323
1 : 서건식 | 남자 | 929392-3232323 | 2 | 백엔드 | 무학 | 21 | 010-2512-3244 010-3235-1423
2 : 조동훈 | 남자 | 938828-3323332 | 3 | 정훈 | 사동로 | 138 | 010-4232-1323
3 : 이미자 | 여자 | 685343-1324242 | 4 | 고객관리 | 이천로 | 443 | 010-3232-1111
  
```

3. 사용자 회원가입

```

Input: 3
아이디: grunaru
비밀번호: 3322
이름: 그루나루
남자는 1, 여자는 2를 입력해주세요.
성별: 2
하이픈을 제외하고 11자리의 숫자를 입력하세요. EX:01022223333
핸드폰 번호: 010-3232-1131
회원가입이 완료되었습니다.
  
```

music.사용자: 6 행 (승) ( >> 다음 << 모두 보기 | ▼ 정렬 ▼ 열 (9/9) ▼ 필터

ID	Password	이름	성별	핸드폰번호	플레이리스트갯수	최대음악재
123	123	건식	남자	010-3232-3232	(NULL)	(NULL)
f	f	f	여자	010-1242-3434	(NULL)	(NULL)
grunaru	3322	그루나루	여자	010-3232-1131	(NULL)	(NULL)
id	pw	이영석	남자	010-3241-1324	(NULL)	(NULL)
iqeq2328	1	건식이	여자	01023223143	(NULL)	(NULL)
whchdgns	whchdgns	조동훈	남자	01023020030	(NULL)	(NULL)

#### 4. 사용자 회원 탈퇴

```
Input: 4
아이디: grunar
비밀번호: 33
존재하지 않는 아이디와 비밀번호입니다.
```

```
-----
0. 종료
1. 사용자 로그인
2. 관리자 로그인
3. 사용자 회원가입
4. 사용자 회원탈퇴
5. 관리자 등록
-----
```

```
Input: 4
아이디: grunaru
비밀번호: 3322
계정을 삭제할까요?(y or n): y
이용해주셔서 감사합니다.
```

ID	Password	이름	성별	핸드폰번호	플레이리스트갯수	최대음악자
123	123	건설	남자	010-3232-3232	(NULL)	(NULL)
f	f	f	여자	010-1242-3434	(NULL)	(NULL)
id	pw	이영석	남자	010-3241-1324	(NULL)	(NULL)
iqeq2328	1	건설이	여자	01023223143	(NULL)	(NULL)
whehdgns	whehdgns	조동훈	남자	01023020030	(NULL)	(NULL)

#### 5. 관리자 등록

```
Input: 5
등록을 취소하려면 exit을 입력하세요.
```

```
이름: 이건희
식별번호: 4
이미 존재하는 사원입니다.
```

```
등록을 취소하려면 exit을 입력하세요.
```

```
이름: 이건희
식별번호: 5
주민등록번호: 988838-1132332
남자는 1, 여자는 2를 입력해주세요.
성별: 1
도로명을 입력하세요: 왕십리로 3길
건물번호를 입력하세요: 32
```

```
입력을 그만하려면 exit을 입력하세요. 최대 10개까지 입력 가능합니다.
핸드폰 번호를 입력하세요: 010-3232-1313
```

```
입력을 그만하려면 exit을 입력하세요. 최대 10개까지 입력 가능합니다.
핸드폰 번호를 입력하세요: 010-4242-1313
```

```
입력을 그만하려면 exit을 입력하세요. 최대 10개까지 입력 가능합니다.
핸드폰 번호를 입력하세요: exit
부서번호를 입력하세요: 5
존재하지 않는 부서번호입니다.
```

```
부서번호를 입력하세요: 4
사원등록을 완료하였습니다
```

music.관리자: 5 행 (중) (대략적)

건물번호	도로명	식별_ID	주민등록번호	이름	성별	부서_식별번호
30	도선동	1	939884-3342332	이영석	남자	1
21	무학	2	929392-3232323	서건식	남자	2
138	사동로	3	938828-3323332	조동호	남자	3
443	이천로	4	685343-1324242	이미자	여자	4
32	왕십리로 3길	5	988838-1132332	이건희	남자	4

## ● 스키마

music	304.0 KiE
관리영역	32.0 KiE
관리자	32.0 KiE
부서	32.0 KiE
부서위치	32.0 KiE
사용자	48.0 KiE
음악	16.0 KiE
이용권	16.0 KiE
플레이리스트	64.0 KiE
핸드폰번호	32.0 KiE

테이블은 총 9개입니다.

## 1. 관리영역

기본
옵션
인덱스
외래 키
분할
CREATE 코드
ALTER 코드

이름: 관리영역
  
코멘트:

열:
+ 추가
x 제거
▲ 위로
▼ 아래로

#	이름	데이터 유형	길이/설정	부호 없음	NULL 허용	0으로 채움	기본값
1	부서_식별번호	VARCHAR	50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음
2	관리영역	VARCHAR	50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음

관리영역은 multivalued attribute이기 때문에 새로 테이블을 생성했습니다.

## 2. 관리자

호스트: 127.0.0.1
데이터베이스: music
테이블: 관리자
데이터
쿼리

기본
옵션
인덱스
외래 키
분할
CREATE 코드
ALTER 코드

이름: 관리자
  
코멘트:

열:
+ 추가
x 제거
▲ 위로
▼ 아래로

#	이름	데이터 유형	길이/설정	부호 없음	NULL 허용	0으로 채움	기본값
1	건물번호	VARCHAR	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
2	도로명	VARCHAR	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
3	식별_ID	VARCHAR	50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	''
4	주민등록번호	VARCHAR	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
5	이름	VARCHAR	20	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
6	성별	CHAR	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
7	부서_식별번호	VARCHAR	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL

식별ID가 primary key입니다.

부서\_식별번호는 관리자가 속한 부서를 의미하며 부서의 식별번호를 foreign key로 가져왔습니다.

### 3. 부서

호스트: 127.0.0.1

데이터베이스: music

테이블: 부서

데이터

쿼리

기본

옵션

인덱스

외래 키

분할

CREATE 코드

ALTER 코드

이름:

부서

코멘트:

열:

+

 추가

×

 제거

▲

 위로

▼

 아래로

#	이름	데이터 유형	길이/설정	부
1	식별번호	VARCHAR	50	
2	부서이름	VARCHAR	50	
3	부서_관리자ID	VARCHAR	50	

식별번호가 primary key입니다.

부서\_관리자ID는 부장을 의미하며 관리자의 식별\_ID를 foreign key로 가져왔습니다.

### 4. 부서위치

호스트: 127.0.0.1

데이터베이스: music

테이블: 부서위치

데이터

쿼리

기본

옵션

인덱스

외래 키

분할

CREATE 코드

ALTER 코드

이름:

부서위치

코멘트:

열:

+

 추가

×

 제거

▲

 위로

▼

 아래로

#	이름	데이터 유형	길이/설정
1	부서_식별번호	VARCHAR	50
2	부서위치	VARCHAR	50

부서의 위치는 multivalued attribute이기 때문에 새로 테이블을 생성했습니다.

부서\_식별번호와 부서위치 두개가 primary key입니다.

## 5. 사용자

호스트: 127.0.0.1
데이터베이스: music
테이블: 사용자
데이터
쿼리

기본
옵션
인덱스
외래 키
분할
CREATE 코드
ALTER 코드

이름: 사용자

코멘트:

열: + 추가 x 제거 ▲ 위로 ▼ 아래로

#	이름	데이터 유형	길이/설
1	ID	VARCHAR	50
2	Password	VARCHAR	50
3	이름	VARCHAR	50
4	성별	CHAR	50
5	핸드폰번호	VARCHAR	50
6	공유받은ID	VARCHAR	50
7	소유한이용권	VARCHAR	50

ID가 primary key입니다.

공유받은ID는 recursive relationship 이므로 ID를 foreign key로 가져왔습니다.

소유한이용권은 이용권 이름 attribute를 foreign key로 가져왔습니다.

## 6. 음악

호스트: 127.0.0.1
데이터베이스: music
테이블: 음악
데이터
쿼리

기본
옵션
인덱스
외래 키
분할
CREATE 코드
ALTER 코드

이름: 음악

코멘트:

열: + 추가 x 제거 ▲ 위로 ▼ 아래로

#	이름	데이터 유형	길이/설정
1	제목	VARCHAR	50
2	아티스트	VARCHAR	50
3	장르	VARCHAR	50
4	가사	VARCHAR	10000
5	플레이시간	VARCHAR	50
6	등록일자	DATE	
7	발매일자	DATE	
8	앨범제목	VARCHAR	50
9	뮤직비디오	VARCHAR	100
10	음악ID	INT	11
11	순위	VARCHAR	50

음악ID와 순위가 primary key입니다. 순위는 실시간 업데이트 되는 기능은 구현 안 했기 때문에 중복을 허용하지 않습니다.

## 7. 이용권

호스트: 127.0.0.1 데이터베이스: music 테이블: 이용권 데이터 쿼리

기본 옵션 인덱스 외래 키 분할 CREATE 코드 ALTER 코드

이름: 이용권

코멘트:

열: + 추가 x 제거 ▲ 위로 ▼ 아래로

#	이름	데이터 유형	길이/설정
1	이용권이름	VARCHAR	50
2	이벤트유무	VARCHAR	50
3	기간	VARCHAR	50
4	허용서비스	VARCHAR	50
5	가격	VARCHAR	50

이용권이름이 primary key입니다.

## 8. 플레이리스트

호스트: 127.0.0.1 데이터베이스: music 테이블: 플레이리스트 데이터 쿼리

기본 옵션 인덱스 외래 키 분할 CREATE 코드 ALTER 코드

이름: 플레이리스트

코멘트:

열: + 추가 x 제거 ▲ 위로 ▼ 아래로

#	이름	데이터 유형	길이/설정
1	소유한사용자	VARCHAR	50
2	음악ID	INT	11
3	제목	VARCHAR	50

소유한사용자,음악ID,제목이 모두 key이며 중복을 허용하지 않습니다.

(3개의 결합이 key를 이루지 않으면 필요없는 tuple이 join시 검색됩니다.)



## 9. 핸드폰번호

기본 옵션 인덱스 외래 키 분할 CREATE 코드 ALTER 코드

이름:

핸드폰번호

코멘트:

열:

+ 추가

✕ 제거

▲ 위로

▼ 아래로

#	이름	데이터 유형	길이/설정
1	관리자_식별ID	VARCHAR	50
2	핸드폰번호	VARCHAR	20

핸드폰번호는 multivalued attribute입니다.(관리자)

두 attribute의 조합이 key입니다.

- Code description

```
import pymysql as pms
from datetime import date

connection = pms.connect(host = 'localhost',
                        port=3306,
                        user='root',
                        password='ad',
                        db='music',
                        charset='utf8')

cursor = connection.cursor()
```

첫 연결입니다.

현재 저의 컴퓨터에서 root 계정의 비밀번호를 변경할 수 없는 오류가 생겨

```
start mysqld.exe --skip-grant-tables
```

를 통해 비밀번호를 치지 않고 접근할 수 있도록 일시적으로 설정하였습니다.

```

def print_list(sql):
    cursor.execute(sql)
    result=cursor.fetchall()
    for i in range(len(result)):
        print(i,":",end = ' ')

        for j in range(len(result[i])):
            print(result[i][j],end = ' | ')

        print('\n')

    return 0

def get_attribute(sql):
    cursor.execute(sql)
    result=cursor.fetchone()
    return result[0]

def insert_into(sql):
    cursor.execute(sql)
    connection.commit()
    return 0

def check_exist(sql):
    cursor.execute(sql)
    success = cursor.fetchone()
    return success[0]

def delete_from(sql):
    cursor.execute(sql)
    connection.commit()

def print_playlist(sql):
    print("\n플레이리스트에 담긴 음악 목록:")
    print("-----")
    print("정보 : 제목 / 아티스트 / 장르 / 플레이시간 / 발매일자 / 앨범제")
    print("-----")
    print_list(sql)
    print("-----")

```

함수 목록입니다.

함수는 간단하며, 중복을 피하기 위해 작성했습니다.

```

print(
while 1:
    print("""
0. 종료\n1. 사용자 로그인\n2. 관리자 로그인\n3. 사용자 회원가입\n4. 사용자 회원탈퇴\n5. 관리자 등록

num1 = input("Input: ")

if num1 == '0':
    break
elif num1 == '1':
    Id = input("아이디: ")
    Pw = input("비밀번호: ")

    sql = "select EXISTS (select * from 사용자 where ID = '{0}' and Password = '{0}')) as success".format(Id,Pw)

    if check_exist(sql):
        while 1:
            print('\n\n\n\n-----플레이리스트 목록\n')
            sql = "select distinct `제목` from 플레이리스트 where `소유한사용자` = '{0}'".format(Id)
            cursor.execute(sql)
            result=cursor.fetchall()
            for i in range(len(result)):
                print(i,":",end = ' ')

                for j in range(len(result[i])):
                    print(result[i][j],end = ' ')

            print('\n')

            print("""
0. 이전화면으로\n1. 전체 음악 순위 보기\n2. 플레이리스트에 음악 넣기\n3. 플레이리스트에서 음악 삭제\n4. 플레이리스트 열기\n5. 플레이리스트 공유하기\n6. 공유받은 플레이리스트 보기\n

num2 = input("Input: ")

```

가장 바깥에 있는 while문입니다. 메뉴별로 코드를 설명하겠습니다.

## 0. 종료

0을 입력하면 종료합니다.

## 1. 사용자 로그인

Id와 Pw를 입력받고

```
sql = "select EXISTS (select * from 사용자 where ID = '{0}' and Password = '{0}')) as success".format(Id,Pw)
```

이와 같은 쿼리문을 작성합니다. ID와 Password가 존재하는지 확인하는 sql문입니다.

이를 check\_exist 함수에 집어넣어서, 있다면 메뉴로 진입하고 안되었다면 실패했다고 출력합니다.

```

if check_exist(sql):
    while 1:
        print('\n\n\n\n-----플레이리스트 목록\n')
        sql = "select distinct `제목` from 플레이리스트 where `소유한사용자` = '{0}'".format(Id)
        cursor.execute(sql)
        result=cursor.fetchall()
        for i in range(len(result)):
            print(i,":",end = ' ')

            for j in range(len(result[i])):
                print(result[i][j],end = ' ')

        print('\n')

        print("""
0. 이전화면으로\n1. 전체 음악 순위 보기\n2. 플레이리스트에 음악 넣기\n3. 플레이리스트에서 음악 삭제\n4. 플레이리스트 열기\n5. 플레이리스트 공유하기\n6. 공유받은 플레이리스트 보기\n

num2 = input("Input: ")

if num2 == '0':
    break
elif num2 == '1':

```

진입했다면 사용자 메뉴를 출력합니다.

처음에 사용자가 가지고 있는 플레이리스트를

```
sql = "select distinct `제목` from 플레이리스트 where `소유한사용자` = '{0}'".format(Id)
```

를 통해서 출력해줍니다.

#### A. 종료

0을 입력하면 이전화면으로 돌아갑니다.

#### B. 전체음악순위보기

```
elif num2 == '1':
    print("-----")
    print("정보 : 순위 / 제목 / 아티스트 / 장르 / 플레이시간 / 발매일자 / 앨범제목 / 뮤직비디오")
    print("-----")
    sql = "select 순위,제목,아티스트,장르,플레이시간,발매일자,앨범제목,뮤직비디오 from `음악` order by 순위"
    print_list(sql)
    print("-----")
```

화면의 sql에서 보이듯, 순위순으로 정렬하며 출력합니다.

#### C. 플레이리스트에 음악 넣기

```
if(name == 'exit'):
    break
```

exit 입력시 빠져나옵니다.

```
else:
    print("\n전체 음악 목록:")
    print("-----")

    print("정보 : 제목 / 아티스트 / 장르 / 플레이시간 / 발매일자 / 앨범제목 / 뮤직비디오 / 순위")
    print("-----")
    sql = "select 제목,아티스트,장르,플레이시간,발매일자,앨범제목,뮤직비디오,순위 from `음악`"
    print_list(sql)
    print("-----")
    sql = "select 음악,제목,아티스트,장르,플레이시간,발매일자,앨범제목,뮤직비디오,순위 from `플레이리스트`,`음악`,사용자 where 플레이리스트.소유한사용자 = '{}' and 음악.제목 = '{}'"
    print_playlist(sql)
    while 1:
        print("입력을 취소하시려면 exit을 눌러주세요.")
        music_name = input("날을 음악의 제목을 선택하세요: ")
        sql = "select EXISTS (select * from 음악 where 제목 = '{}') as success".format(music_name)

        if music_name == 'exit':
            break
        elif (check_exist(sql) != 1):
            print("음악이 존재하지 않습니다.")
            print("-----")
        else:
            sql = "select `음악ID` from `음악` where `제목` = '{}'.format(music_name)
            music_id = get_attribute(sql)
            sql = "select EXISTS (select * from 플레이리스트 where 소유한사용자 = '{}' and 음악ID = {} and 제목 = '{}') as success".format(Id,music_id,name)
            if(check_exist(sql)):
                print("음악이 이미 존재합니다.")
                print("-----")
            else:
                sql = "insert into `플레이리스트` values ('{}',{},{})".format(Id,music_id,name)
                insert_into(sql)
                print("입력을 완료했습니다.")
                break
```

플레이리스트가 존재하지 않는다면 새로 생성합니다.

전체 음악목록을 select문을 통해 뽑아서 출력한 후에, name으로 받은 플레이리스트 안에 있는 음악도 출력해줍니다. 다시 while문을 돌면서, 예외를 처리하며 음악의 제목을 받습니다.

플레이리스트를 소유한사용자의 플레이리스트이며, 다른사용자와 이름이 겹치더라도 해당 플레이리스트에 속한 음악만을 출력하기 위해서

```
sql = "select 음악.제목,아티스트,장르,플레이시간,발매일자,앨범제목,뮤직비디오,순위 from `플레이리스트`,`음악`,사용자 where 플레이리스트.소유한사용자 = '{}' and 플레이리스트.소유한사용자 = 사용자.ID and `플레이리스트`.음악ID = `음악`.음악ID and `플레이리스트`.제목 = '{}".format(Id,name)
```

로 작성했습니다.

Music\_id를 sql문으로 받은 후에 이 id가 음악 테이블에 존재하는지 검사 후, 존재한다면 플레이리스트안에 넣어줍니다.

Insert 쿼리문은 다음과 같습니다.

```
sql = "insert into `플레이리스트` values ('{}',{},{})".format(Id,music_id,name)
```

플레이리스트의 모든 attribute의 조합이 key이기 때문에 중복은 허용되지 않습니다.

#### D. 플레이리스트에서 음악 삭제

3을 입력하면 해당 메뉴로 진입합니다.

sql문을 통해서 이름이 존재하는지 확인합니다.

```
break
elif(check_exist(sql)):
    while 1:
        done = 0
        sql = "select 음악,제목,아티스트,장르,플레이시간,발매일자,앨범제목,뮤직비디오,순위 from `플레이리스트`,`음악`,`사용자`"
        print_playlist(sql)
        print("입력을 취소하시려면 exit을 눌러주세요.")
        music_name = input("삭제할 음악의 제목을 선택하세요: ")

        sql = "select EXISTS (select * from 음악 where 제목 = '{}') as success".format(music_name)
        while 1:
            if music_name == 'exit':
                done = 1
                break
            elif (check_exist(sql) != 1):
                print("음악이 존재하지 않습니다.\n")
                break
            else:
                sql = "select `음악ID` from `음악` where `제목` = '{}'"
                music_id = get_attribute(sql)
                sql = "select EXISTS (select * from 플레이리스트 where 소유한사용자 = '{}' and 음악ID = {} and 제목 = '{}') as success".format(Id,music_id,music_name)
                if(check_exist(sql)):
                    sql = "delete from `플레이리스트` where 소유한사용자 = '{}' and 음악ID = {} and 제목 = '{}'"
                    delete_from(sql)
                    print("음악을 플레이리스트에서 삭제하였습니다.")
                    print("-----")
                    done = 1
                    break
            if done == 1:
                break
        done = 1
    else:
        print("존재하지 않는 플레이리스트 입니다.")
        print("-----")
```

존재한다면 음악을 삭제하는데, 음악의 이름이 음악 테이블의 목록에 있을 때 지워줍니다.

else문에서 한번 더 소유자의 플레이리스트에 해당음악이 존재하는지 검사후 있다면 지워줍니다.

Delete sql 문은 여기에 명시하겠습니다.

```
sql = "delete from `플레이리스트` where 소유한사용자 = '{}' and 음악ID = {} and 제목 = '{}'"
format(Id,music_id,name)
```

#### E. 플레이리스트 열기

```
elif num2 == '4':
    while 1:
        name = input("입력을 취소하시려면 exit을 눌러주세요.\n플레이리스트 이름을 입력하세요: ")
        sql = "select EXISTS (select * from 플레이리스트 where 제목 = '{}' and 소유한사용자 = '{}') as success".format(name,Id)
        if name == 'exit':
            break

        elif(check_exist(sql)):
            sql = "select 음악,제목,아티스트,장르,플레이시간,발매일자,앨범제목,뮤직비디오,순위 from `플레이리스트`,`음악`,`사용자`"
            print_playlist(sql)
            break
        else:
            print("장문된 값 입니다.")
```

소유한 플레이리스트를 출력합니다.

```
sql = "select 음악.제목,아티스트,장르,플레이시간,발매일자,앨범제목,뮤직비디오,순위 from `플레이리스트`,`음악`,`사용자` where 플레이리스트.소유한사용자 = '{}' and 플레이리스트.소유한사용자 = 사용자.ID and `플레이리스트`.음악ID = `음악`.음악ID and `플레이리스트`.제목 = '{}".format(lid,name)
```

이 sql문은 다른 곳에서도 쓰였으며, 플레이리스트안에 있는 음악을 select하기 위함입니다. 사용자마다 같은 플레이리스트 이름이 있을 수 있으니 소유한사용자도 where절에서 구분해줘야 합니다.

#### F. 플레이리스트 공유

```
elif num2 == '5':
    while 1:
        share_id = input("공유할 사용자의 ID를 입력하세요: ")
        sql = "select EXISTS (select * from 사용자 where ID = '{}') as success".format(share_id)
        if share_id == 'exit':
            break
        elif(check_exist(sql)):
            sql = "update 사용자 set 공유받은ID = '{}' where ID = '{}'".format(lid,share_id)
            insert_into(sql)
            print("공유를 완료하였습니다.")
            break
        else:
            print("존재하지 않는 ID입니다.")
```

플레이리스트를 다른사용자 에게 공유합니다.

이렇게 하면 공유받은 사용자의 공유받은ID를 update하게 됩니다.

sql문은 위 사진에 포함되어있습니다.

#### G. 공유받은 플레이리스트 보기

```
elif num2 == '6':
    while 1:
        sql = "select 공유받은ID from 사용자 where ID = '{}'".format(lid)
        share_id = get_attribute(sql)
        print("~~~~~플레이리스트 목록:~~~~~")
        sql = "select distinct `제목` from 사용자,플레이리스트 where `소유한사용자` = 사용자.공유받은ID and 소유한사용자 = '{}'".format(share_id)
        print_list(sql)

        while 1:
            name = input("입력을 취소하시려면 exit을 눌러주세요.~~~~~플레이리스트 이름을 입력하세요: ")
            sql = "select EXISTS (select * from 플레이리스트 where 제목 = '{}' and 소유한사용자 = '{}') as success".format(name,share_id)
            if name == 'exit':
                break
            elif(check_exist(sql)):
                sql = "select 음악.제목,아티스트,장르,플레이시간,발매일자,앨범제목,뮤직비디오,순위 from `플레이리스트`,`음악`,`사용자` where 플레이리스트.소유한사용자 = '{}' and 플레이리스트.공유받은ID = '{}'".format(share_id,name)
                print_playlist(sql)
                break
            else:
                print("잘못된 값 입니다.")
        break
```

공유받은 사용자의 플레이리스트를 출력하는 과정입니다.

다른 곳에서 쓰인 sql문과 거의 동일합니다.

#### H. 이용권 구매하기

```

        break
    elif num2 == '7':
        while 1:
            sql = "select 이용권 이름,기간,허용서비스,가격 from 이용권"
            print('\n\n\n이용권 목록:\n-----')
            print("정보 : 이용권 이름 / 기간 / 허용서비스 / 가격 ")
            print('-----')
            print_list(sql)
            print('-----')
            while 1:
                name = input("입력을 취소하시려면 exit을 눌러주세요.\n이용권 이름을 입력하세요: ")
                sql = "select EXISTS (select * from 이용권 where 이용권 이름 = '{}') as success".format(name)
                if name == 'exit':
                    break

                elif(check_exist(sql)):
                    sql = "update 사용자 set 소유한이용권 = '{}' where ID = {}".format(name,Id)
                    insert_into(sql)
                    print("구매를 완료했습니다.")
                    break
                else:
                    print("잘못된 값 입니다.")
            break
        break
    elif num2 == '8':

```

이용권의 목록을 출력 후, sql문을 통해 입력한 이용권 이름이 존재한다면 update를 해줍니다.

#### I. 구매한 이용권 보기

```

        break
    elif num2 == '8':
        sql = "select 소유한이용권 from 사용자 where ID = {}".format(Id)
        name = get_attribute(sql)
        sql = "select 이용권 이름,기간,허용서비스,가격 from 이용권 where 이용권.이용권 이름 = {}".format(name)
        print('\n\n\n이용권 목록:\n-----')
        print("정보 : 이용권 이름 / 기간 / 허용서비스 / 가격 ")
        print('-----')
        print_list(sql)
        print('-----')

```

select문에 해당하는 ID를 넣어서, name에 소유한 이용권 이름을 받아온 후 select합니다.

## 2. 관리자 로그인

첫 메뉴에서 2를 입력하면 관리자 모드로 들어갑니다.

```

elif num1 == '2':
    Id = input("관리자 식별 ID를 입력하세요: ")
    sql = "select exists (select * from 관리자 where 식별_ID={}) as success".format(Id)
    if(check_exist(sql)):
        while 1:

```

관리자모드는 자신의 식별 ID를 입력하면 들어갈 수 있습니다.

## A. 음악 추가하기

```
break
elif num2 == '1':
    while 1:
        name = input("입력을 취소하시려면 exit을 입력하세요.\n추가할 음악의 정보를 입력합니다.\n제목: ")
        if name == 'exit':
            break
        else:
            while 1:
                music_id = input("입력을 취소하시려면 exit을 입력하세요.\n음악 ID: ")
                sql = "select exists (select * from 음악 where 음악ID='{0}') as success".format(music_id)
                if music_id == 'exit':
                    break
                elif(check_exist(sql)):
                    print("존재하는 음악 ID 입니다. 다시 입력하세요.")
            else:
                while 1:
                    rank = input("입력을 취소하시려면 exit을 입력하세요.\n순위: ")
                    sql = "select exists (select * from 음악 where 순위='{0}') as success".format(rank)
                    if rank == 'exit':
                        break
                    elif(check_exist(sql)):
                        print("존재하는 순위 입니다. 다시 입력하세요.")
                else:
                    artist = input("아티스트 이름: ")
                    genre = input("장르: ")
                    lyrics = input("가사: ")
                    play_time = input("플레이 시간 양식: m분ss초\n플레이시간: ")
                    enroll_date = today.isoformat()
                    release_date = input("발매일자 양식 : yyyy-mm-dd\n발매일자: ")
                    album_name = input("앨범 제목: ")
                    mv = input("뮤직비디오 링크: ")
                    sql = "insert into 음악 values ('{0}','{0}','{0}','{0}','{0}','{0}','{0}','{0}','{0}','{0}','{0}')".format(name,artist,genre,lyrics,play_time,enroll_date,release_date,album_name,mv)
                    insert_into(sql)
                    print("입력에 성공하였습니다.")
                    print("-----")
                    break
```

중복체크를 해가면서 음악의 정보를 입력하고, 마지막에 sql문을 통해서 최종적으로 commit합니다.

Music\_id와 rank는 중복될 수 없습니다.

## B. 음악 삭제하기

```
elif num2 == '2':
    while 1:
        print("\n\n\n전체 음악 목록:")
        print("-----")
        print("정보 : 제목 / 음악ID")
        print("-----")

        sql = "select 제목,음악ID from `음악`"
        print_list(sql)
        print("-----")

        music_id=input("입력을 취소하시려면 exit을 눌러주세요.\n삭제할 음악의 음악 ID를 입력하세요: ")
        sql = "select EXISTS (select * from 음악 where 음악ID = '{0}') as success".format(music_id)
        if music_id == 'exit':
            break
        elif(check_exist(sql)):
            sql = "delete from 플레이리스트 where 음악ID = {}".format(music_id)
            delete_from(sql)
            sql = "delete from `음악` where 음악ID = {}".format(music_id)
            delete_from(sql)
            print("음악을 삭제하였습니다.")
        else:
            print("존재하지 않는 음악 ID입니다.")
```

전체 음악 목록을 출력한 후에 삭제할 음악 ID만 입력후에 삭제 해줍니다.



### C. 이용권 추가하기

```
elif num2 == '3':
    print("존재하지 않는 이용권 이름입니다.")
    while 1:
        name = input("입력을 취소하시려면 exit을 입력하세요. \n이용권 이름을 입력하세요:")
        sql = "select EXISTS (select * from 이용권 where 이용권이름 = '{}') as success".format(name)

        if name == 'exit':
            break
        elif(check_exist(sql)):
            print("이미 존재하는 이용권 이름 입니다.")
        else:
            event = input("이벤트 유무를 0또는 1로 입력하세요: ")
            period = input("기간을 입력하세요. ex: 2달이면 2m, 1주면 1w입력: ")
            service = input("허용서비스를 입력하세요. \n(streaming, download , all) 중 하나: ")
            price = input("가격을 숫자만 입력하세요. 단위: won: ")

            sql = "insert into 이용권 values('{}','{}','{}','{}','{}')".format(name,event,period,service,price)
            insert_into(sql)
            print("입력을 완료하였습니다.")
            break
```

이용권의 정보를 입력하고 추가해줍니다.

### D. 이용권 삭제하기

```
elif num2 == '4':
    while 1:
        name = input("입력을 취소하시려면 exit을 입력하세요. \n이용권 이름을 입력하세요:")
        sql = "select EXISTS (select * from 이용권 where 이용권이름 = '{}') as success".format(name)

        if name == 'exit':
            break
        elif(check_exist(sql)):

            sql = "delete from 이용권 where 이용권이름 = '{}'.format(name)
            delete_from(sql)
            print("삭제를 완료하였습니다.")
            break

        else:
            print("존재하지 않는 이용권 이름 입니다.")
```

이용권의 정보를 입력하고 삭제해줍니다.

## E. 부서 추가하기

```
elif num2 == '5':
    while 1:
        Dno = input("등록을 취소하려면 exit을 입력하세요. \n부서 번호를 입력하세요: ")
        if(Dno == 'exit'):
            break
        sql = "select EXISTS (select * from 부서 where 식별번호 = '{}') as success".format(Dno)

        if(check_exist(sql)):
            print("이미 존재하는 부서 번호 입니다.\n\n")

        else:
            ssn = input("부장으로 임명할 사원의 관리자 ID를 입력하세요: ")
            sql = "select EXISTS (select * from 관리자 where 식별_ID = '{}') as success".format(ssn)
            if(check_exist(sql)):
                name = input("부서 이름을 입력하세요: ")

                location = []
                num = 0

                while 1:
                    temp = input("\n입력을 그만하려면 exit을 입력하세요. 최대 10개까지 입력 가능합니다. \n부서의 위치를 입력하세요: ")
                    if temp == 'exit':
                        break
                    elif num == 9:
                        break
                    else:
                        location.append(temp)
                        num = num+1

                area = []
                num = 0
                while 1:
                    temp = input("\n입력을 그만하려면 exit을 입력하세요. 최대 10개까지 입력 가능합니다. \n관리 영역을 입력하세요: ")
                    if temp == 'exit':
                        break
                    elif num == 9:
                        break
                    else:
                        area.append(temp)
                        num = num+1

            sql = "insert into 부서 values ('{}','{}','{}')".format(Dno,name,ssn)
            insert_into(sql)
            for i in range(len(location)):
                sql = "insert into 부서위치 values ('{}','{}')".format(Dno,location[i])
                insert_into(sql)
            for i in range(len(area)):
                sql = "insert into 관리영역 values ('{}','{}')".format(Dno,area[i])
                insert_into(sql)
            print("입력을 완료하였습니다.")
            break
        else:
            print("존재하지 않는 관리자입니다.\n\n")
```

Key는 중복을 체크하면서 부서를 추가합니다.

특히 부장으로 임명할 관리자 ID는, 관리자에 존재하는 식별ID일때만 넣을 수 있도록 구분을 해주었습니다.

Location과 area(관리영역)은 multivalued attribute입니다.

각각은, temp라는 임시 변수에 입력을 받으면서 최대 10개(num)까지 입력을 받습니다.

이후에 insert문에 Dno,name,ssn까지만 넣어준 후 for문을통해 각 list의 값을 하나씩 넣어주면서

Multivalued attribute는 따로 처리해줍니다.

## F. 부서 정보 출력

```
elif num2 == '6':
    print("정보 : 식별번호 / 부서이름 / 부장이름 / 부서위치 / 관리영역 ")
    print("-----")
    sql = "select distinct 식별번호,부서이름,관리자.이름 from 부서,관리자 where 부서.부서_관리자 ID = 관리자.식별_ID"

    cursor.execute(sql)
    result=cursor.fetchall()

    for i in range(len(result)):
        print(i,":",end = ' ')

        for j in range(len(result[i])):
            print(result[i][j],end = ' | ')

        Dno = result[i][0]
        sql = "select 부서위치,부서위치 from 부서,부서위치 where 부서.식별번호 = 부서위치,부서_식별번호 and 부서위치.부서_식별번호 = '{} '".format(Dno)
        cursor.execute(sql)
        location = cursor.fetchall()
        for k in range(len(location)):
            print(location[k][0],end = ' ')
        print('\n',end=' ')
        sql = "select 관리영역.관리영역 from 부서,관리영역 where 부서.식별번호 = 관리영역,부서_식별번호 and 관리영역.부서_식별번호 = '{} '".format(Dno)
        cursor.execute(sql)
        area = cursor.fetchall()
        for k in range(len(area)):
            print(area[k][0],end = ' ')
        print('\n\n')
```

부서의 정보를 출력합니다.

Multivalue attribute의 출력은, 이중 for문을 통해 처리합니다.

부장 이름까지는 j를 통해서 모두 출력합니다.

각각의 부서위치, 관리영역은 cursor에서 sql문으로 각 테이블의 값을 select한 후에 출력해줍니다.

최종적으로 tuple별로 attribute들을 출력해줍니다.

## G. 사원 정보 출력

```
elif num2 == '7':
    print("정보 : 이름 / 성별 / 주민등록번호 / 식별ID / 부서이름 / 도로명 / 건물번호 / 핸드폰번호 ")
    print("-----")
    sql = "select distinct 이름,성별,주민등록번호,식별_ID,부서이름,도로명,건물번호 from 부서,관리자 where 부서.부서_관리자 ID = 관리자.식별_ID"

    cursor.execute(sql)
    result=cursor.fetchall()

    for i in range(len(result)):
        print(i,":",end = ' ')

        for j in range(len(result[i])):
            print(result[i][j],end = ' | ')

        ssn = result[i][3]
        sql = "select 핸드폰번호,핸드폰번호 from 관리자,핸드폰번호 where 관리자.식별_ID = 핸드폰번호.관리자_식별ID and 핸드폰번호.관리자_식별ID = '{} '".format(ssn)
        cursor.execute(sql)
        area = cursor.fetchall()
        for k in range(len(area)):
            print(area[k][0],end = ' ')
        print('\n\n')

    else:
        print("잘못된 값 입니다.")
```

부서 정보의 출력과 굉장히 유사합니다.

핸드폰번호는 multivalued attribute이기 때문에 위와 같이 처리합니다.

### 3. 사용자 회원가입

```
elif num1 == '3':
    while 1:
        Id = input("아이디: ")
        Pw = input("비밀번호: ")
        sql = "select EXISTS (select * from 사용자 where ID = '{}' and Password = '{}') as success".format(Id,Pw)
        if(check_exist(sql)):
            print("이미 존재하는 아이디 입니다.\n")
        else:
            name = input("이름: ")
            sex = '00'
            while 1:
                sex = input("남자는 1, 여자는 2를 입력해주세요.\n성별: ")
                if sex=='1':
                    sex = '남자'
                    break
                elif sex == '2':
                    sex = '여자'
                    break
                else:
                    print("잘못된 값입니다.")
            phone_number = input("하이픈을 제외하고 11자리의 숫자를 입력하세요. EX: 01022223333\n핸드폰 번호: ")
            sql = "insert into 사용자(ID>Password>이름>성별>핸드폰번호) values ('{}','{}','{}','{}','{}')".format(Id,Pw,name,sex,phone_number)
            insert_into(sql)
            print("회원가입이 완료되었습니다.\n")
            break
```

사용자의 회원가입입니다.

아이디와 비밀번호를 입력후 사용자 테이블에 존재하는지 검사합니다.

존재한다면 입력을 취소하고, 없다면 정보를 입력합니다.

마지막에 insert문으로 commit합니다.

### 4. 사용자 회원탈퇴

```
elif num1 == '4':
    while 1:
        Id = input("아이디: ")
        Pw = input("비밀번호: ")
        sql = "select EXISTS (select * from 사용자 where ID = '{}' and Password = '{}') as success".format(Id,Pw)
        if(check_exist(sql)):
            done = input("계정을 삭제합니까 ?(y or n): ")
            if done == 'y':
                sql = "delete from `사용자` where ID = '{}' ".format(Id)
                delete_from(sql)
                print("이름 해주셔서 감사합니다.\n")
                break
            elif done == 'n':
                break
            else:
                print("잘못된 값입니다.")
        else:
            print("존재하지 않는 아이디와 비밀번호 입니다.\n")
            break
```

사용자의 회원 탈퇴입니다.

회원가입과 반대로 존재한다면 지워주고, 존재하지 않는다면 입력을 취소합니다.

delete문은 ID만 맞으면 지워줍니다. ID가 Key 이기 때문입니다.

## 5. 관리자 등록

```
elif num1 == '5':
    while 1:
        print("등록을 취소하려면 exit을 입력하세요.\n")
        name = input("이름: ")
        if(name == 'exit'):
            break
        ssn = input("식별번호: ")
        sql = "select EXISTS (select * from 관리자 where 식별_ID = '{}') as success".format(ssn)

        if(check_exist(sql)):
            print("이미 존재하는 사원입니다.\n")
        else:
            number_region = input("주민등록번호: ")
            sex = '00'
            while 1:
                sex = input("남자는 1, 여자는 2를 입력해주세요.\n성별: ")
                if sex=='1':
                    sex = '남자'
                    break
                elif sex == '2':
                    sex = '여자'
                    break
                else:
                    print("잘못된 값입니다.")

            street = input("도로명을 입력하세요: ")
            building = input("건물번호를 입력하세요: ")

            phone_number = []
            num = 0

            while 1:
                temp = input("\n입력을 그만하려면 exit을 입력하세요. 최대 10개까지 입력 가능합니다.\n핸드폰 번호를 입력하세요: ")
                if temp == 'exit':
                    break
                elif num == 9:
                    break
                else:
                    phone_number.append(temp)
                    num = num+1
```

관리자 등록입니다.

식별번호가 이미 존재하면 다시 입력합니다.

성별은 1또는 2로만 입력이 가능합니다.

핸드폰번호는 multivalued attribute이므로 부서의 관리영역,부서위치를 추가할 때와 동일한 코드입니다.

```
Dno = 'default'
while 1:
    Dno = input('부서번호를 입력하세요: ')
    sql = "select EXISTS (select * from 부서 where 식별번호 = '{}') as success".format(Dno)
    if(check_exist(sql)):
        sql = "insert into 관리자(건물번호, 도로명, 식별_ID, 주민등록번호, 이름, 성별, 부서_식별번호) values ('{}','{}','{}','{}','{}','{}','{}')".format(building,street)
        insert_into(sql)
        for i in range(len(phone_number)):
            sql = "insert into 핸드폰번호 values ('{}','{}')".format(ssn,phone_number[i])
            insert_into(sql)

        print("사원등록을 완료하였습니다.\n")
        break
    else:
        print("존재하지 않는 부서번호 입니다.\n")
        break
```

부서번호가 실제로 존재하는 부서번호인지 체크 후, 값을 commit 합니다.