

데이터베이스 기말과제

경제학부 응용통계전공 201911275 박예림

주제

- 음악 스트리밍 어플을 관리하는 데이터베이스
 - 음악 스트리밍 어플 : 멜론, 지니, 벅스 등



요구사항 분석 및 개념 모델링

▪ 요구사항 명세서

- 앨범은 각각의 고유번호를 가지며, 앨범명, 장르, 소개글, 재생시간, 수록곡 수 정보를 가진다. 앨범은 여러 장르를 포함할 수 있다.
- 앨범에 수록되는 음원은 트랙번호, 제목, 장르, 재생시간, 가사, 음질, 실시간순위, 가격 정보를 가진다. 앨범과 마찬가지로 장르는 여러 개일 수 있다.
- 한 앨범에는 1개 이상의 여러 음원이 수록되어 있고, 음원은 앨범에 수록된 형태로만 나온다. 즉, 음원이 존재하기 위해서는 앨범이 존재해야 한다.
- 앨범은 유통사에 의해 발매된다. 발매 시 발매일 정보를 저장하며, 유통사는 회사명(고유함), 전화번호 정보를 가진다. 유통사는 여러 앨범을 발매할 수 있으며, 각 앨범은 한 유통사에서만 발매될 수 있다.
- 아티스트는 이름(고유함), 데뷔일, 소속사, 장르, 유형 정보를 가진다. 장르는 여러 개일 수 있고, 각 아티스트의 유형은 다양하게 존재한다.
- 아티스트는 1명 이상의 아티스트로 구성된다. 즉, 아티스트는 다른 여러 아티스트에 소속될 수 있고, 여러 아티스트를 소속시킬 수도 있다.

요구사항 분석 및 개념 모델링

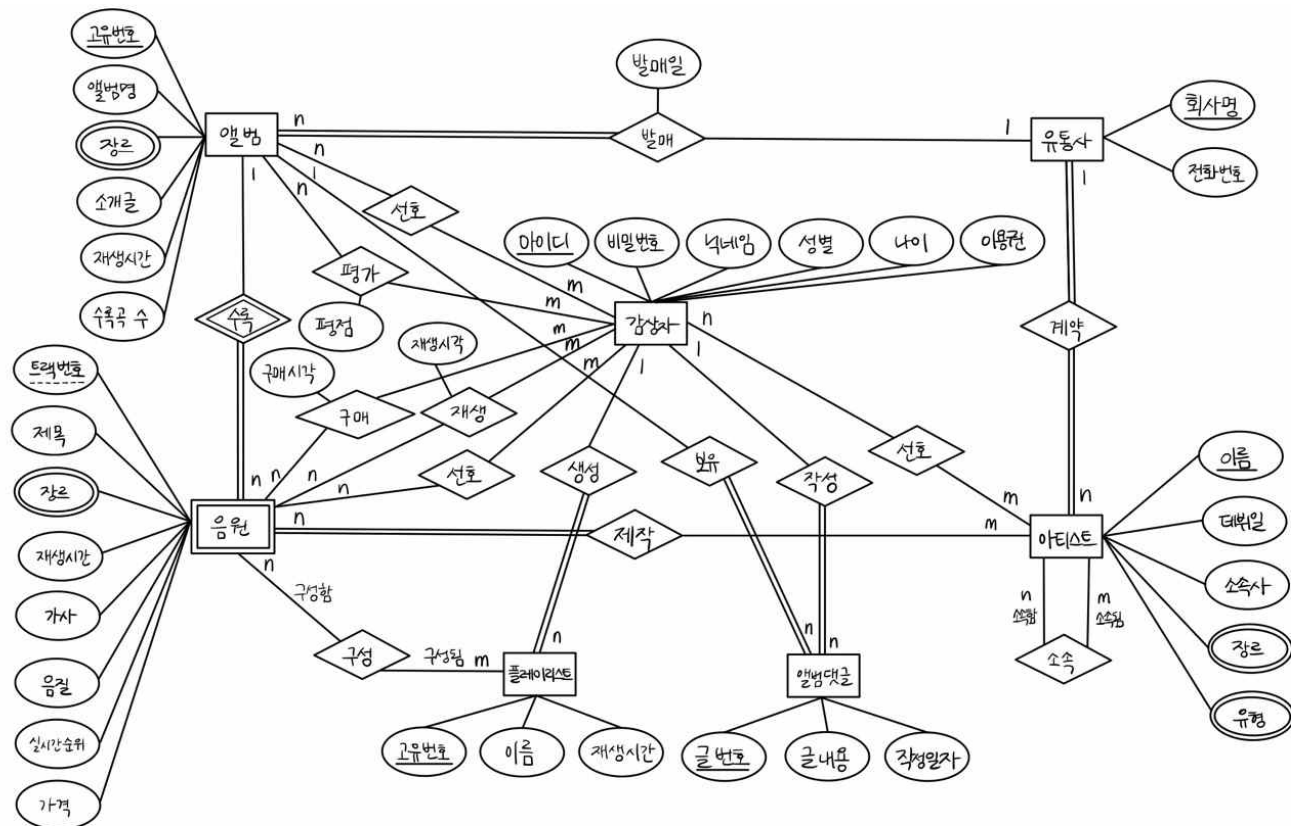
▪ 요구사항 명세서

- 아티스트는 음원을 제작하고, 유통사와의 계약을 통해 앨범을 발매한다. 음원을 제작한다는 것은 노래를 부르거나, 작곡/작사에 참여하는 것을 의미한다. 각 아티스트는 여러 음원을 제작할 수 있으며, 음원은 1명 이상의 아티스트에 의해 제작된다. 아티스트가 앨범을 발매하기 위해서는 유통사와 아티스트의 계약이 필수적이며, 한 유통사는 여러 아티스트와 계약할 수 있다. 각 아티스트는 한 유통사와만 계약 가능하다.
- 감상자는 아이디, 비밀번호, 닉네임, 성별, 나이, 이용권 정보를 가지고, 음원을 재생하거나 구매한다. 감상자가 음원 재생 시 재생 시각에 대한 정보를 저장하며, 구매 시에는 구매 시각 정보를 저장한다. 또한 감상자는 앨범을 평가하여 평점을 남기기도 한다.
- 감상자는 여러 앨범, 음원, 아티스트를 선호할 수 있으며, 각 앨범, 음원, 아티스트도 여러 감상자에 의해 선호될 수 있다.
- 앨범은 여러 개의 댓글을 보유하는데, 이러한 댓글은 감상자에 의해 작성된다. 앨범댓글은 글 번호 (고유함), 글 내용, 작성일자 정보를 포함하며, 감상자는 한 앨범에 대해 여러 댓글을 작성할 수 있다.
- 감상자는 자신의 플레이리스트를 생성할 수 있다. 플레이리스트는 고유번호, 이름, 재생시간 정보를 가지며, 여러 음원들로 구성된다. 각 음원은 여러 플레이리스트에 포함될 수도 있다.

요구사항 분석 및 개념 모델링

▪ ER 다이어그램

✓ 앨범, 음원, 아티스트, 유통사, 감상자 등의 개체 존재



기말과제 제안서 대비 차이점, 발전성

- **식별 관계 보완**

- 약한 개체와 식별 개체 간의 관계를 식별 관계로 설정하지 않았던 것을 올바르게 수정

- **데이터의 구체적 구상을 통한 선택과 집중**

- 앨범, 음원, 아티스트 각 개체가 모두 댓글을 보유하는 관계가 존재했지만, 앨범이 나머지를 포함하는 넓은 개념이므로 앨범만 관계를 가지도록 수정하여 복잡도를 줄임. 이에 따라 '댓글' 개체의 이름을 '앨범댓글'로 구체적으로 명시
- 아티스트의 'SNS', '소개글' 속성, 음원의 '뮤직비디오' 속성은 구체적인 구상이 없었고, 중요도가 낮아 굳이 사용할 이유가 없어 삭제

- **이름을 알아보기 쉽게 수정**

- 음원 속성 중 '순위'가 어떤 순위인지 구체적이지 않아 '실시간순위'로 정정
- 음원을 구매한다는 것은 다운로드 한다는 것이 내재되어 있는 것이므로, 이름이 길었던 '구매및다운로드' 속성 이름을 '구매'로 변경

논리 모델링 (관계형 모델)

- 테이블 정의 (DDL)

- 유통사(회사명, 전화번호)

```
CREATE TABLE COMPANY(  
    C_NAME VARCHAR2(100) PRIMARY KEY,  
    C_CONTACT VARCHAR2(20) NOT NULL  
);
```

- 앨범(고유번호, 앨범명, 유통사명(FK), 발매일, 소개글, 재생시간, 수록곡 수)

```
CREATE TABLE ALBUM(  
    A_ID NUMBER(20) PRIMARY KEY,  
    A_NAME VARCHAR2(200) NOT NULL,  
    A_COMPANYNAME VARCHAR2(100) NOT NULL,  
    A_RELEASEDATE VARCHAR2(20) NOT NULL,  
    A_DESCRIPTION VARCHAR2(1000),  
    A_PLAYTIME VARCHAR2(10) CHECK (A_PLAYTIME LIKE '%:%%'),  
    A_NUMOFSONGS NUMBER(10),  
    FOREIGN KEY(A_COMPANYNAME) REFERENCES COMPANY(C_NAME) ON DELETE CASCADE  
);
```

■ 테이블 정의 (DDL)

- 앨범장르(앨범고유번호(FK), 장르)

```
CREATE TABLE ALBUMGENRE (  
    AG_AID NUMBER(20),  
    AG_GENRE VARCHAR2(20),  
    FOREIGN KEY(AG_AID) REFERENCES ALBUM(A_ID) ON DELETE CASCADE,  
    PRIMARY KEY(AG_AID, AG_GENRE)  
);
```

- 음원(앨범고유번호(FK), 트랙번호, 제목, 재생시간, 가사, 음질, 실시간순위, 가격)

```
CREATE TABLE MUSIC(  
    M_AID NUMBER(20),  
    M_TRACKID NUMBER(20),  
    M_NAME VARCHAR(200) NOT NULL,  
    M_PLAYTIME VARCHAR2(10) NOT NULL CHECK (M_PLAYTIME LIKE '%:%%'),  
    M_LYRICS VARCHAR(3000),  
    M_SOUNDQUALITY VARCHAR2(20),  
    M_REALTIMERANKING NUMBER(10),  
    M_PRICE NUMBER(10) NOT NULL,  
    FOREIGN KEY(M_AID) REFERENCES ALBUM(A_ID) ON DELETE CASCADE,  
    PRIMARY KEY(M_AID, M_TRACKID)  
);
```

- 음원장르(앨범고유번호(FK), 음원트랙번호(FK), 장르)

```
CREATE TABLE MUSICGENRE(  
    MG_AID NUMBER(20),  
    MG_MTRACKID NUMBER(20),  
    MG_GENRE VARCHAR2(20),  
    FOREIGN KEY(MG_AID, MG_MTRACKID) REFERENCES MUSIC(M_AID, M_TRACKID) ON DELETE CASCADE,  
    PRIMARY KEY(MG_AID, MG_MTRACKID, MG_GENRE)  
);
```


■ 테이블 정의 (DDL)

- 아티스트(이름, 계약유통사명(FK), 데뷔일, 소속사)

```
CREATE TABLE ARTIST(  
    AT_NAME VARCHAR2(50) PRIMARY KEY,  
    AT_COMPANY VARCHAR2(100) NOT NULL,  
    AT_DEBUIDATE VARCHAR2(10),  
    AT_ENTERTAINMENT VARCHAR(100) NOT NULL,  
    FOREIGN KEY(AT_COMPANY) REFERENCES COMPANY(C_NAME) ON DELETE CASCADE  
);
```

- 아티스트소속(아티스트멤버이름(FK), 아티스트그룹이름(FK))

```
CREATE TABLE ARTISTBELONG(  
    AT_MEMBERNAME VARCHAR2(50),  
    AT_GROUPNAME VARCHAR2(50),  
    FOREIGN KEY(AT_MEMBERNAME) REFERENCES ARTIST(AT_NAME) ON DELETE CASCADE,  
    FOREIGN KEY(AT_GROUPNAME) REFERENCES ARTIST(AT_NAME) ON DELETE CASCADE,  
    PRIMARY KEY(AT_MEMBERNAME, AT_GROUPNAME)  
);
```

- 아티스트장르(아티스트이름(FK), 장르)

```
CREATE TABLE ARTISTGENRE(  
    ATG_ATNAME VARCHAR2(50),  
    ATG_GENRE VARCHAR2(20),  
    FOREIGN KEY(ATG_ATNAME) REFERENCES ARTIST(AT_NAME) ON DELETE CASCADE,  
    PRIMARY KEY(ATG_ATNAME, ATG_GENRE)  
);
```

- 아티스트유형(아티스트이름(FK), 유형)

```
CREATE TABLE ARTISTTYPE(  
    ATT_ATNAME VARCHAR2(50),  
    ATT_TYPE VARCHAR2(30),  
    FOREIGN KEY(ATT_ATNAME) REFERENCES ARTIST(AT_NAME) ON DELETE CASCADE,  
    PRIMARY KEY(ATT_ATNAME, ATT_TYPE)  
);
```

■ 테이블 정의 (DDL)

- 음원제작(아티스트이름(FK), 앨범고유번호(FK), 음원트랙번호(FK))

```
CREATE TABLE MUSICPRODUCT(  
    MP_ATNAME VARCHAR2(50),  
    MP_AID NUMBER(20),  
    MP_MTRACKID NUMBER(20),  
    FOREIGN KEY(MP_ATNAME) REFERENCES ARTIST(AI_NAME) ON DELETE CASCADE,  
    FOREIGN KEY(MP_AID, MP_MTRACKID) REFERENCES MUSIC(M_AID, M_TRACKID) ON DELETE CASCADE,  
    PRIMARY KEY(MP_ATNAME, MP_AID, MP_MTRACKID)  
);
```

- 감상자(아이디, 비밀번호, 닉네임, 성별, 나이, 이용권)

```
CREATE TABLE LISTENER(  
    L_ID VARCHAR2(20) PRIMARY KEY,  
    L_PW VARCHAR2(20) NOT NULL,  
    L_NICKNAME VARCHAR2(50) NOT NULL,  
    L_GENDER CHAR(2),  
    L_AGE NUMBER(3),  
    L_TICKET VARCHAR2(20)  
);
```

- 선호앨범(감상자아이디(FK), 앨범고유번호(FK))

```
CREATE TABLE PREFERALBUM(  
    PA_LID VARCHAR2(20),  
    PA_AID NUMBER(20),  
    FOREIGN KEY(PA_LID) REFERENCES LISTENER(L_ID) ON DELETE CASCADE,  
    FOREIGN KEY(PA_AID) REFERENCES ALBUM(A_ID) ON DELETE CASCADE,  
    PRIMARY KEY(PA_LID, PA_AID)  
);
```

- 앨범평가(감상자아이디(FK), 앨범고유번호(FK), 평점)

```
CREATE TABLE EVALALBUM(  
    EA_LID VARCHAR2(20),  
    EA_AID NUMBER(20),  
    EA_SCORE NUMBER(1) CHECK (EA_SCORE<=5),  
    FOREIGN KEY(EA_LID) REFERENCES LISTENER(L_ID) ON DELETE CASCADE,  
    FOREIGN KEY(EA_AID) REFERENCES ALBUM(A_ID) ON DELETE CASCADE,  
    PRIMARY KEY(EA_LID, EA_AID)  
);
```

■ 테이블 정의 (DDL)

- 음원구매(감상자아이디(FK), 앨범고유번호(FK), 음원트랙번호(FK), 구매시각)

```
CREATE TABLE BUYMUSIC(  
    BM_LID VARCHAR2(20),  
    BM_AID NUMBER(20),  
    BM_MTRACKID NUMBER(20),  
    BM_BUYTIME VARCHAR2(10) NOT NULL CHECK (BM_BUYTIME LIKE '%%:%%'),  
    FOREIGN KEY(BM_LID) REFERENCES LISTENER(L_ID) ON DELETE CASCADE,  
    FOREIGN KEY(BM_AID, BM_MTRACKID) REFERENCES MUSIC(M_AID, M_TRACKID) ON DELETE CASCADE,  
    PRIMARY KEY(BM_LID, BM_AID, BM_MTRACKID)  
);
```

- 음원재생(감상자아이디(FK), 앨범고유번호(FK), 음원트랙번호(FK), 재생시각)

```
CREATE TABLE PLAYMUSIC(  
    PM_LID VARCHAR2(20),  
    PM_AID NUMBER(20),  
    PM_MTRACKID NUMBER(20),  
    PM_PLAYTIME VARCHAR2(10) NOT NULL CHECK (PM_PLAYTIME LIKE '%%:%%'),  
    FOREIGN KEY(PM_LID) REFERENCES LISTENER(L_ID) ON DELETE CASCADE,  
    FOREIGN KEY(PM_AID, PM_MTRACKID) REFERENCES MUSIC(M_AID, M_TRACKID) ON DELETE CASCADE,  
    PRIMARY KEY(PM_LID, PM_AID, PM_MTRACKID)  
);
```

- 선호음원(감상자아이디(FK), 앨범고유번호(FK), 음원트랙번호(FK))

```
CREATE TABLE PFERMUSIC(  
    PFM_LID VARCHAR2(20),  
    PFM_AID NUMBER(20),  
    PFM_MTRACKID NUMBER(20),  
    FOREIGN KEY(PFM_LID) REFERENCES LISTENER(L_ID) ON DELETE CASCADE,  
    FOREIGN KEY(PFM_AID, PFM_MTRACKID) REFERENCES MUSIC(M_AID, M_TRACKID) ON DELETE CASCADE,  
    PRIMARY KEY(PFM_LID, PFM_AID, PFM_MTRACKID)  
);
```

■ 테이블 정의 (DDL)

- 선호아티스트(감상자아이디(FK), 아티스트이름(FK))

```
CREATE TABLE PREFERARTIST(  
    PAT_LID VARCHAR2(20),  
    PAT_ATNAME VARCHAR2(50),  
    FOREIGN KEY(PAT_LID) REFERENCES LISTENER(L_ID) ON DELETE CASCADE,  
    FOREIGN KEY(PAT_ATNAME) REFERENCES ARTIST(AT_NAME) ON DELETE CASCADE,  
    PRIMARY KEY(PAT_LID, PAT_ATNAME)  
);
```

- 플레이리스트(고유번호, 이름, 재생시간, 생성자아이디(FK))

```
CREATE TABLE PLAYLIST(  
    PL_ID NUMBER(20) PRIMARY KEY,  
    PL_NAME VARCHAR2(200) NOT NULL,  
    PL_PLAYTIME VARCHAR2(10) NOT NULL CHECK (PL_PLAYTIME LIKE '%:%%'),  
    PL_LID VARCHAR2(20),  
    FOREIGN KEY(PL_LID) REFERENCES LISTENER(L_ID) ON DELETE CASCADE  
);
```

- 플레이리스트구성(플레이리스트고유번호(FK), 앨범고유번호(FK), 음원트랙번호(FK))

```
CREATE TABLE PLAYLISTCONFIG(  
    PLC_PLID NUMBER(20),  
    PLC_AID NUMBER(20),  
    PLC_MTRACKID NUMBER(20),  
    FOREIGN KEY(PLC_PLID) REFERENCES PLAYLIST(PL_ID) ON DELETE CASCADE,  
    FOREIGN KEY(PLC_AID, PLC_MTRACKID) REFERENCES MUSIC(M_AID, M_TRACKID) ON DELETE CASCADE,  
    PRIMARY KEY(PLC_PLID, PLC_AID, PLC_MTRACKID)  
);
```

- 앨범댓글(글번호, 앨범고유번호(FK), 감상자아이디(FK), 글내용, 작성일자)

```
CREATE TABLE ALBUMCOMMENT(  
    AC_ID NUMBER(30) PRIMARY KEY,  
    AC_AID NUMBER(20),  
    AC_LID VARCHAR2(20),  
    AC_CONTENT VARCHAR2(1000) NOT NULL,  
    AC_WRITEDATE VARCHAR2(20) NOT NULL,  
    FOREIGN KEY(AC_AID) REFERENCES ALBUM(A_ID) ON DELETE CASCADE,  
    FOREIGN KEY(AC_LID) REFERENCES LISTENER(L_ID) ON DELETE CASCADE  
);
```

논리 모델링 (관계형 모델)

▪ 구현 (DML) - 데이터 삽입/수정/검색/삭제

- 실제 데이터를 참고하여 삽입

앨범 정보



[싱글]
strawberry moon
아이유

발매일 2021.10.19
장르 록/메탈
발매사 카카오엔터테인먼트
기획사 EDAM엔터테인먼트

♥ 35,437 > 앨범다운 >

앨범 정보



[EP]
Spring to Spring
호피폴라

앨범 정보



[싱글]
입만 열면 거짓말
I! (아일)

발매일 2021.11.23

★★★★★ 5.0 294명 >

앨범 정보



[싱글]
영웅 수집가
이승윤

발매일 2020.09.25
장르 인디음악, 록/메탈
발매사 비스킷 사운드
기획사 알라리강송

♥ 11,797 > 앨범다운 > > FLAC앨범다운 > > 선물하기 >

수록곡 (1)

> 듣기 + 담기 > 다운 > FLAC > 선물

☐ 번호 곡정보 좋아요 듣기 담기 다운

☐ 1 > TITLE strawberry moon ♥ 163,411 > + > FLAC

아이유

논리 모델링 (관계형 모델)

- 구현 (DML) - 데이터 삽입/수정/검색/삭제

- 유통사(COMPANY) 테이블

```
INSERT INTO COMPANY VALUES('카카오엔터테인먼트', '02-2280-7700');  
INSERT INTO COMPANY VALUES('Dreamus', '1577-5557');  
INSERT INTO COMPANY VALUES('비스킷 사운드', '070-4457-3085');  
SELECT * FROM COMPANY;
```

	C_NAME	C_CONTACT
1	카카오엔터테인먼트	02-2280-7700
2	Dreamus	1577-5557
3	비스킷 사운드	070-4457-3085

논리 모델링 (관계형 모델)

▪ 구현 (DML) - 데이터 삽입/수정/검색/삭제

- 앨범을 기준으로 데이터 삽입 : 한 앨범마다 앨범/음원/아티스트 데이터 삽입 반복
 - ✓ 총 4개의 앨범과 그에 관련하여 음원/아티스트 데이터 존재

```
INSERT INTO ALBUM VALUES(10743453, 'strawberry moon', '카카오엔터테인먼트', '2021.10.19',
'포토샵으로 만든 것 같은 6월 밤하늘의 달기 색깔 달보다, 사랑에 빠졌을 때 내 안에서 일어나는 일들이 더 믿기 힘든 판타지에 가깝다.
자주 오지 않더라도, 다시 오지 않더라도 누구나 한 번쯤 경험한 적 있을 그 선비한 순간을, 이 곡을 들으면서 떠올렸으면 좋겠다.', '03:25', 1);
INSERT INTO ALBUMGENRE VALUES(10743453, '록/메탈');
INSERT INTO MUSIC VALUES(10743453, 1, 'strawberry moon', '03:25',
'달이 익어가니 서둘러 절은 피야
민들레 한 송이 들고
사랑이 어지러이 떠다니는 밤이야
날아가 사뭇히 이렇듯
```

(생략)

```
바람을 세로질러
날아오르는 기분 so cool
삶이 어떻게 더 완벽해 ooh',
'Flac 16/24bit', 6, 1800);
INSERT INTO MUSICGENRE VALUES(10743453, 1, '록/메탈');
INSERT INTO ARTIST VALUES('아이유', '카카오엔터테인먼트', '2018.09.18', 'EDAM엔터테인먼트');
INSERT INTO ARTISTGENRE VALUES('아이유', '발라드');
INSERT INTO ARTISTGENRE VALUES('아이유', '댄스');
INSERT INTO ARTISTGENRE VALUES('아이유', '록/메탈');
INSERT INTO ARTISTGENRE VALUES('아이유', '국내드라마');
INSERT INTO ARTISTGENRE VALUES('아이유', '알앤비/소울');
INSERT INTO ARTISTGENRE VALUES('아이유', '포크/블루스');
INSERT INTO ARTISTGENRE VALUES('아이유', '국내영화');
INSERT INTO ARTISTGENRE VALUES('아이유', '일렉트로니카');
INSERT INTO ARTISTTYPE VALUES('아이유', '솔로');
INSERT INTO ARTISTTYPE VALUES('아이유', '여성');
INSERT INTO MUSICPRODUCT VALUES('아이유', '10743453', '1');
```

논리 모델링 (관계형 모델)

▪ 구현 (DML) - 데이터 삽입/수정/검색/삭제

- 앨범/음원/아티스트 등의 음악 정보

SELECT * FROM ALBUM;

⚡ A_ID	⚡ A_NAME	⚡ A_COMPANYNAME	⚡ A_RELEASEDATE	⚡ A_DESCRIPTION
1 10743453	strawberry moon	카카오엔터테인먼트	2021.10.19	포토샵으로 만든 것 같은 6월 밤하늘의 달기 색깔 달보다, 사랑에 빠졌을 때 내 안에서 일어나는 일들이
2 10421113	Spring to Spring	Dreamus	2020.04.22	자연의 온기에 더해진 따뜻한 위로격정을 다스리는 사랑의 메시지새롭게 수록된 4곡의 공통된 특징은 '특
3 10495943	영웅 수집가	비스킷 사운드	2020.09.25	함부로 누군가를 상점으로 만들고 영웅으로 둔갑시키고 것발에 이름을 새겨 흔들다가 원치 않는 모습을
4 10786723	입만 열면 거짓말	Dreamus	2021.11.23	오는 12월 15일 발매되는 아일의 첫 번째 EP 'Kiwi Mixtape'의 선공개 곡 '입만 열면 거짓말' 그리고

SELECT * FROM MUSIC;

⚡ M_AID	⚡ M_TRACKID	⚡ M_NAME	⚡ M_PLAYTIME	⚡ M_LYRICS
1 10743453	1	strawberry moon	03:25	달이 익어가니 서둘러 젊은 피아민들레 한 송이 들고사랑이 어지러이 떠다니는 밤이야날아
2 10421113	1	Opfern	02:18	No matter how farI'll be there where you goI'm always by your sideSo woul
3 10421113	2	About Time	05:29	She said when you are by my side불안한 새벽도다시 밝아온 댔조마지막 날 그날엔비
4 10421113	3	동화 (Märchen)	05:44	(null)
5 10421113	4	그거면 돼요	04:06	다른 누군갈 사랑하란 말내겐 너무 어렵조내일 보잔 말 이젠 불안할 뿐인 걸요내겐 어쨌보
6 10421113	5	Our Song	04:40	I'm losing my way나만 이런 건지아닐 거라고 생각했조그런 줄 알고 있던 거죠Maybe I
7 10421113	6	소량	03:09	차가웠던 공기 그리고 저 강물 소리그때나 지금이나 내겐 버거운 당신을 참 향기롭군요비
8 10786723	1	입만 열면 거짓말	03:31	Red lips 난 너 밖에 없지다 새빨간 거짓말 거짓말말린 your eyes your eyesOh plea
9 10495943	1	영웅 수집가	04:13	그토록 찾아 해맨 사람을 만난 것 같아아마도 나의 영웅이야아쩌면 저렇게도 올곧고 위대
10 10495943	2	시적 허용	05:03	고요를 깨지 않는 것보다적절한 말을 몰라그냥 입술을 뚫고만 있었던 거죠 그땐시적 허용

SELECT * FROM ALBUMGENRE;

⚡ AG_AID	⚡ AG_GENRE
1 10421113	발라드
2 10495943	록/메탈
3 10495943	인디음악
4 10743453	록/메탈
5 10786723	알앤비/소울

SELECT * FROM MUSICGENRE;

⚡ MG_AID	⚡ MG_MTRACKID	⚡ MG_GENRE
1 10421113	1	발라드
2 10421113	2	발라드
3 10421113	3	발라드
4 10421113	4	발라드
5 10421113	5	발라드
6 10421113	6	발라드
7 10495943	1	록/메탈
8 10495943	1	인디음악
9 10495943	2	록/메탈
10 10495943	2	인디음악
11 10743453	1	록/메탈
12 10786723	1	알앤비/소울

논리 모델링 (관계형 모델)

▪ 구현 (DML) - 데이터 삽입/수정/검색/삭제

- 앨범/음원/아티스트 등의 음악 정보

SELECT * FROM ARTIST;

AT_NAME	AT_COMPANY	AT_DEBUTDATE	AT_ENTERTAINMENT
1 아이유	카카오엔터테인먼트	2018.09.18	EDAM엔터테인먼트
2 호피콜라	Dreamus	2019.11.16	모스뮤직
3 I'11 (아일)	Dreamus	2017.11.01	엠제이드림시스
4 이승윤	비스킷 사운드	2016.06.16	쇼플레이엔터테인먼트

SELECT * FROM ARTISTBELONG;

AT_MEMBERNAME	AT_GROUPNAME
1 I'11 (아일)	호피콜라

SELECT * FROM ARTISTTYPE;

ATT_ATNAME	ATT_TYPE
1 I'11 (아일)	남성
2 I'11 (아일)	솔로
3 아이유	솔로
4 아이유	여성
5 이승윤	남성
6 이승윤	솔로
7 호피콜라	그룹
8 호피콜라	남성

SELECT * FROM ARTISTGENRE;

ATG_ATNAME	ATG_GENRE
1 I'11 (아일)	국내드라마
2 I'11 (아일)	록/메탈
3 I'11 (아일)	발라드
4 I'11 (아일)	알앤비/소울
5 I'11 (아일)	포크/블루스
6 아이유	국내드라마
7 아이유	국내영화
8 아이유	댄스
9 아이유	록/메탈
10 아이유	발라드
11 아이유	알앤비/소울
12 아이유	일렉트로니카
13 아이유	포크/블루스
14 이승윤	국내드라마
15 이승윤	록/메탈
16 이승윤	발라드
17 이승윤	알앤비/소울
18 이승윤	인디음악
19 이승윤	포크/블루스
20 호피콜라	뉴에이지
21 호피콜라	댄스
22 호피콜라	록/메탈
23 호피콜라	발라드
24 호피콜라	포크/블루스

SELECT * FROM MUSICPRODUCT;

MP_ATNAME	MP_AID	MP_MTRACKID
1 I'11 (아일)	10786723	1
2 아이유	10743453	1
3 이승윤	10495943	1
4 이승윤	10495943	2
5 호피콜라	10421113	1
6 호피콜라	10421113	2
7 호피콜라	10421113	3
8 호피콜라	10421113	4
9 호피콜라	10421113	5
10 호피콜라	10421113	6

논리 모델링 (관계형 모델)

▪ 구현 (DML) - 데이터 삽입/수정/검색/삭제

• 감상자 관련 정보

```
INSERT INTO LISTENER VALUES('abc123', '1234', '가나다', 'W', 21, '스트리밍클럽');
INSERT INTO LISTENER VALUES('qwe520', '1207', '뮤직라이프', 'M', 32, 'MP3 30 정기결제');
INSERT INTO LISTENER VALUES('good7', '0102', 'Ilovemusic', 'W', 18, NULL);
```

```
SELECT * FROM LISTENER;
```

	L_ID	L_PW	L_NICKNAME	L_GENDER	L_AGE	L_TICKET
1	abc123	1234	가나다	W	21	스트리밍클럽
2	qwe520	1207	뮤직라이프	M	32	MP3 30 정기결제
3	good7	0102	Ilovemusic	W	18	(null)

```
INSERT INTO PREFERALBUM VALUES('abc123', 10421113);
INSERT INTO PREFERALBUM VALUES('qwe520', 10743453);
INSERT INTO PREFERALBUM VALUES('good7', 10495943);
```

```
SELECT * FROM PREFERALBUM;
```

	PA_LID	PA_AID
1	abc123	10421113
2	good7	10495943
3	qwe520	10743453

```
INSERT INTO EVALALBUM VALUES('abc123', 10743453, 4);
INSERT INTO EVALALBUM VALUES('abc123', 10421113, 5);
INSERT INTO EVALALBUM VALUES('abc123', 10786723, 4);
INSERT INTO EVALALBUM VALUES('abc123', 10495943, 3);
INSERT INTO EVALALBUM VALUES('qwe520', 10743453, 5);
INSERT INTO EVALALBUM VALUES('qwe520', 10421113, 4);
INSERT INTO EVALALBUM VALUES('qwe520', 10786723, 3);
INSERT INTO EVALALBUM VALUES('qwe520', 10495943, 2);
INSERT INTO EVALALBUM VALUES('good7', 10743453, 4);
INSERT INTO EVALALBUM VALUES('good7', 10421113, 4);
INSERT INTO EVALALBUM VALUES('good7', 10786723, 3);
INSERT INTO EVALALBUM VALUES('good7', 10495943, 5);
```

```
SELECT * FROM EVALALBUM;
```

	EA_LID	EA_AID	EA_SCORE
1	abc123	10743453	4
2	abc123	10421113	5
3	abc123	10786723	4
4	abc123	10495943	3
5	qwe520	10743453	5
6	qwe520	10421113	4
7	qwe520	10786723	3
8	qwe520	10495943	2
9	good7	10743453	4
10	good7	10421113	4
11	good7	10786723	3
12	good7	10495943	5

논리 모델링 (관계형 모델)

▪ 구현 (DML) - 데이터 삽입/수정/검색/삭제

• 감상자 관련 정보

```
ALTER TABLE BUYMUSIC MODIFY BM_BUYTIME VARCHAR2(20);
INSERT INTO BUYMUSIC VALUES('abc123', 10421113, 4, '2021.12.01 13:22');
INSERT INTO BUYMUSIC VALUES('abc123', 10421113, 5, '2021.11.03 15:01');
INSERT INTO BUYMUSIC VALUES('qwe520', 10743453, 1, '2021.10.28 01:24');
INSERT INTO BUYMUSIC VALUES('qwe520', 10421113, 4, '2021.11.25 11:33');
INSERT INTO BUYMUSIC VALUES('qwe520', 10786723, 1, '2021.12.06 23:55');
```

```
SELECT * FROM BUYMUSIC;
```

	BM_LID	BM_AID	BM_MTRACKID	BM_BUYTIME
1	abc123	10421113	4	2021.12.01 13:22
2	abc123	10421113	5	2021.11.03 15:01
3	qwe520	10743453	1	2021.10.28 01:24
4	qwe520	10421113	4	2021.11.25 11:33
5	qwe520	10786723	1	2021.12.06 23:55

```
ALTER TABLE PLAYMUSIC RENAME COLUMN PM_PLAYTIME TO PM_PLAYTIME; --오타 수정
ALTER TABLE PLAYMUSIC MODIFY PM_PLAYTIME VARCHAR2(20);
INSERT INTO PLAYMUSIC VALUES('abc123', 10743453, 1, '2021.11.01 00:01');
INSERT INTO PLAYMUSIC VALUES('abc123', 10421113, 5, '2021.11.02 18:01');
INSERT INTO PLAYMUSIC VALUES('abc123', 10421113, 4, '2021.11.31 18:05');
INSERT INTO PLAYMUSIC VALUES('qwe520', 10743453, 1, '2021.10.28 00:20');
INSERT INTO PLAYMUSIC VALUES('qwe520', 10421113, 4, '2021.11.21 12:34');
INSERT INTO PLAYMUSIC VALUES('qwe520', 10786723, 1, '2021.12.01 18:52');
```

```
SELECT * FROM PLAYMUSIC;
```

	PM_LID	PM_AID	PM_MTRACKID	PM_PLAYTIME
1	abc123	10743453	1	2021.11.01 00:01
2	abc123	10421113	5	2021.11.02 18:01
3	abc123	10421113	4	2021.11.31 18:05
4	qwe520	10743453	1	2021.10.28 00:20
5	qwe520	10421113	4	2021.11.21 12:34
6	qwe520	10786723	1	2021.12.01 18:52

```
INSERT INTO PREFERMUSIC VALUES('abc123', 10743453, 1);
INSERT INTO PREFERMUSIC VALUES('abc123', 10421113, 1);
INSERT INTO PREFERMUSIC VALUES('abc123', 10421113, 2);
INSERT INTO PREFERMUSIC VALUES('abc123', 10421113, 3);
INSERT INTO PREFERMUSIC VALUES('abc123', 10421113, 4);
INSERT INTO PREFERMUSIC VALUES('abc123', 10421113, 5);
INSERT INTO PREFERMUSIC VALUES('abc123', 10421113, 6);
INSERT INTO PREFERMUSIC VALUES('qwe520', 10743453, 1);
INSERT INTO PREFERMUSIC VALUES('qwe520', 10421113, 4);
INSERT INTO PREFERMUSIC VALUES('qwe520', 10421113, 5);
INSERT INTO PREFERMUSIC VALUES('good7', 10743453, 1);
INSERT INTO PREFERMUSIC VALUES('good7', 10495943, 1);
```

```
SELECT * FROM PREFERMUSIC;
```

	PFM_LID	PFM_AID	PFM_MTRACKID
1	abc123	10421113	1
2	abc123	10421113	2
3	abc123	10421113	3
4	abc123	10421113	4
5	abc123	10421113	5
6	abc123	10421113	6
7	abc123	10743453	1
8	good7	10495943	1
9	good7	10743453	1
10	qwe520	10421113	4
11	qwe520	10421113	5
12	qwe520	10743453	1

논리 모델링 (관계형 모델)

▪ 구현 (DML) - 데이터 삽입/수정/검색/삭제

• 감상자 관련 정보

```
INSERT INTO PREFERARTIST VALUES('abc123', '호피콜라');
INSERT INTO PREFERARTIST VALUES('abc123', 'I'11 (아일)');
INSERT INTO PREFERARTIST VALUES('qwe520', '아이유');
INSERT INTO PREFERARTIST VALUES('good7', '미승윤');
```

```
SELECT * FROM PREFERARTIST;
```

	PAT_LID	PAT_ATNAME
1	abc123	I'11 (아일)
2	abc123	호피콜라
3	good7	미승윤
4	qwe520	아이유

```
INSERT INTO ALBUMCOMMENT VALUES(111, 10421113, 'abc123', '노래 너무 좋아요 명반ㅠㅠ 호피콜라 최고!!', '2021.12.05');
INSERT INTO ALBUMCOMMENT VALUES(21, 10495943, 'qwe520', '앨범이 전달하고자 하는 의미가 인상 깊네요. 잘 듣고 갑니다.', '2021.12.03');
INSERT INTO ALBUMCOMMENT VALUES(2198, 10743453, 'good7', '역시 아이유! 이번 노래도 너무 좋네요~', '2021.12.07');
```

```
SELECT * FROM ALBUMCOMMENT;
```

	AC_ID	AC_AID	AC_LID	AC_CONTENT	AC_WRITEDATE
1	111	10421113	abc123	노래 너무 좋아요 명반ㅠㅠ 호피콜라 최고!!	2021.12.05
2	21	10495943	qwe520	앨범이 전달하고자 하는 의미가 인상 깊네요. 잘 듣고 갑니다.	2021.12.03
3	2198	10743453	good7	역시 아이유! 이번 노래도 너무 좋네요~	2021.12.07

```
INSERT INTO PLAYLIST VALUES(12346, '호피콜라 노래 모음', '28:51', 'abc123');
INSERT INTO PLAYLISTCONFIG VALUES(12346, 10421113, 1);
INSERT INTO PLAYLISTCONFIG VALUES(12346, 10421113, 2);
INSERT INTO PLAYLISTCONFIG VALUES(12346, 10421113, 3);
INSERT INTO PLAYLISTCONFIG VALUES(12346, 10421113, 4);
INSERT INTO PLAYLISTCONFIG VALUES(12346, 10421113, 5);
INSERT INTO PLAYLISTCONFIG VALUES(12346, 10421113, 6);
INSERT INTO PLAYLISTCONFIG VALUES(12346, 10786723, 1);
```

```
INSERT INTO PLAYLIST VALUES(15342, '밴드 음악', '23:31', 'good7');
INSERT INTO PLAYLISTCONFIG VALUES(15342, 10421113, 2);
INSERT INTO PLAYLISTCONFIG VALUES(15342, 10421113, 4);
INSERT INTO PLAYLISTCONFIG VALUES(15342, 10421113, 5);
INSERT INTO PLAYLISTCONFIG VALUES(15342, 10495943, 1);
INSERT INTO PLAYLISTCONFIG VALUES(15342, 10495943, 2);
```

```
SELECT * FROM PLAYLIST;
```

	PL_ID	PL_NAME	PL_PLAYTIME	PLL_ID
1	12346	호피콜라 노래 모음	28:51	abc123
2	15342	밴드 음악	23:31	good7

```
SELECT * FROM PLAYLISTCONFIG;
```

	PLC_PLID	PLC_AID	PLC_MTRACKID
1	12346	10421113	1
2	12346	10421113	2
3	12346	10421113	3
4	12346	10421113	4
5	12346	10421113	5
6	12346	10421113	6
7	12346	10786723	1
8	15342	10421113	2
9	15342	10421113	4
10	15342	10421113	5
11	15342	10495943	1
12	15342	10495943	2

논리 모델링 (관계형 모델)

▪ 구현 (DML) - 데이터 삽입/수정/검색/삭제

- 수록곡이 1개인 앨범의 이름과 유통사명, 재생시간을 검색

```
SELECT A_NAME, A_COMPANYNAME, A_PLAYTIME FROM ALBUM WHERE A_NUMOFSONGS = 1;
```

	A_NAME	A_COMPANYNAME	A_PLAYTIME
1	strawberry moon	카카오엔터테인먼트	03:25
2	입만 열면 거짓말	Dreamus	03:31

- 수록곡이 2개 이상인 앨범의 이름과 유통사명, 재생시간을 검색

```
SELECT A_NAME, A_COMPANYNAME, A_PLAYTIME FROM ALBUM WHERE A_NUMOFSONGS >= 2;
```

	A_NAME	A_COMPANYNAME	A_PLAYTIME
1	Spring to Spring	Dreamus	25:20
2	영웅 수집가	비스킷 사운드	09:16

- 모든 앨범 장르를 중복 없이 검색

```
SELECT DISTINCT AG_GENRE FROM ALBUMGENRE;
```

AG_GENRE
1 알앤비/소울
2 발라드
3 록/메탈
4 인디음악

- 모든 음원 장르를 중복 없이 검색

```
SELECT DISTINCT MG_GENRE FROM MUSICGENRE;
```

MG_GENRE
1 알앤비/소울
2 발라드
3 록/메탈
4 인디음악

논리 모델링 (관계형 모델)

▪ 구현 (DML) - 데이터 삽입/수정/검색/삭제

- 모든 아티스트를 아티스트이름순(오름차순)으로 검색

```
SELECT AT_NAME FROM ARTIST ORDER BY AT_NAME;
```

AT_NAME
1 I'11 (아일)
2 아이유
3 이승윤
4 호피폴라

- 아티스트 '아이유'의 아티스트 장르를 모두 검색

```
SELECT * FROM ARTISTGENRE  
WHERE ATG_ATNAME = '아이유';
```

ATG_ATNAME	ATG_GENRE
1 아이유	국내드라마
2 아이유	국내영화
3 아이유	댄스
4 아이유	록/메탈
5 아이유	발라드
6 아이유	알앤비/소울
7 아이유	일렉트로니카
8 아이유	포크/블루스

- 아티스트 유형이 '그룹'이 아닌 모든 아티스트의 이름을 검색

```
SELECT ATT_ATNAME FROM ARTISTTYPE  
MINUS  
SELECT ATT_ATNAME FROM ARTISTTYPE WHERE ATT_TYPE = '그룹';
```

ATT_ATNAME
1 I'11 (아일)
2 아이유
3 이승윤

- 아티스트 '호피폴라'가 직접 제작한 음원의 제목을 검색

```
SELECT M_NAME FROM MUSICPRODUCT MP, MUSIC M  
WHERE MP.MP_ATNAME = '호피폴라' AND MP.MP_AID = M.M_AID AND MP.MP_MTRACKID = M.M_TRACKID;
```

M_NAME
1 Opfern
2 About Time
3 동화 (Märchen)
4 그거면 돼요
5 Our Song
6 소담

- **구현 (DML) - 데이터 삽입/수정/검색/삭제**

- ```
SELECT * FROM LISTENER ORDER BY L NICKNAME DESC;
```

```
SELECT EA_AID, AVG(EA_SCORE) FROM EVALALBUM GROUP BY EA_AID;
```

```
SELECT EA_AID, A_NAME, AG_GENRE, EA.AVGSCR FROM
(SELECT A_ID, A_NAME, AG_GENRE
FROM ALBUM, ALBUMGENRE
WHERE A_ID = AG_AID) AF
INNER JOIN
(SELECT EA_AID, AVG(EA_SCORE) AS AVGSCR FROM EVALALBUM GROUP BY EA_AID HAVING AVG(EA_SCORE) >= 4) EA
ON AF.A ID = EA.EA AID;
```

[illegible]

# 논리 모델링 (관계형 모델)

## ▪ 구현 (DML) - 데이터 삽입/수정/검색/삭제

- 2회 이상 구매한 음원의 이름, 가격을 검색

```
SELECT M_NAME, M_PRICE FROM
(SELECT BM_AID, BM_MTRACKID, COUNT(BM_MTRACKID) FROM BUYMUSIC GROUP BY BM_AID, BM_MTRACKID
HAVING COUNT(BM_AID) >= 2 AND COUNT(BM_MTRACKID) >= 2) BM, MUSIC M
WHERE BM.BM_AID = M.M_AID AND BM.BM_MTRACKID = M.M_TRACKID;
```

| M_NAME   | M_PRICE |
|----------|---------|
| 1 그거면 돼요 | 1800    |

- 감상자가 재생한 각 음원의 제목과, 제작에 참여한 아티스트, 그리고 재생된 횟수를 검색

```
SELECT M_NAME, MP_ATNAME, FPM.M_CNT FROM
(SELECT M_AID, M_TRACKID, M_NAME, MP_ATNAME
FROM MUSICPRODUCT MP INNER JOIN MUSIC M
ON MP.MP_AID = M.M_AID AND MP.MP_MTRACKID = M.M_TRACKID) FM,
(SELECT PM_AID, PM_MTRACKID, COUNT(PM_AID), COUNT(PM_MTRACKID) AS M_CNT FROM PLAYMUSIC
GROUP BY PM_AID, PM_MTRACKID) FPM
WHERE FM.M_AID = FPM.PM_AID AND FM.M_TRACKID = FPM.PM_MTRACKID;
```

| M_NAME            | MP_ATNAME | M_CNT |
|-------------------|-----------|-------|
| 1 Our Song        | 호피콜라      | 1     |
| 2 그거면 돼요          | 호피콜라      | 2     |
| 3 strawberry moon | 아이유       | 2     |
| 4 입만 열면 거짓말       | I'11 (아일) | 1     |

- 모든 플레이리스트의 이름과, 각 플레이리스트에 수록된 곡의 개수를 검색

```
SELECT PL_NAME, PLC.CNT FROM PLAYLIST,
(SELECT PLC_PLID, COUNT(PLC_PLID) AS CNT FROM PLAYLISTCONFIG GROUP BY PLC_PLID) PLC
WHERE PL_ID = PLC_PLID;
```

| PL_NAME      | CNT |
|--------------|-----|
| 1 밴드 음악      | 5   |
| 2 호피콜라 노래 모음 | 7   |

- 아이디가 'good7'인 감상자의 앨범댓글 삭제 후 확인

```
DELETE FROM ALBUMCOMMENT WHERE AC_LID = 'good7';
SELECT * FROM ALBUMCOMMENT;
```

| AC_ID | AC_AID | AC_LID   | AC_CONTENT                                | AC_WRITEDATE |
|-------|--------|----------|-------------------------------------------|--------------|
| 1     | 111    | 10421113 | abc123 노래 너무 좋아요 명반ㅠㅠ 호피콜라 최고!!           | 2021.12.05   |
| 2     | 21     | 10495943 | qwe520 앨범이 전달하고자 하는 의미가 인상 깊네요. 잘 듣고 갑니다. | 2021.12.03   |



# 장점/단점

## ▪ 장점

- 데이터가 많고 다양하다.
- 실제 음악 스트리밍 어플 데이터와 최대한 비슷하게 구현하였다.
- 따라서 데이터가 구체적이고 다양한 값을 가진다.
- 수업 중 배운 내용을 최대한 다양하게 활용하기 위해 노력했다.

## ▪ 단점

- 풍부하고 다양한 데이터를 더 폭넓게 활용할 수 있었을 것 같다.
- '아티스트'의 범위가 조금 넓은 것 같다. 가수, 작곡가, 작사가를 나눠 구체적으로 작성했어도 좋을 것 같다.
- SQL 코드 상에서 변수 이름이 복잡해 가독성이 떨어진다.
- 각 음원의 음질이나 가격 속성의 값은 모든 음원에서 차이가 거의 없기 때문에 속성에서 제외해도 괜찮을 것 같다. 사용되는 개체나 속성에 대한 중요도 판단을 더욱 신중히 해 볼 필요가 있을 것 같다.

감사합니다

---