

# KpopDictionary

## DB design

Lee jeong mo

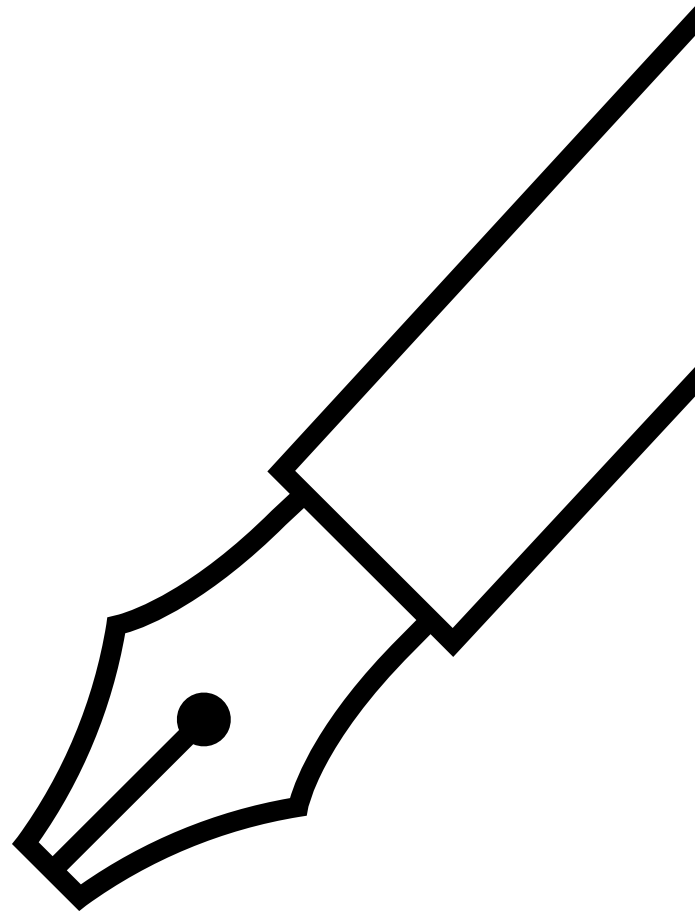
# Contents.

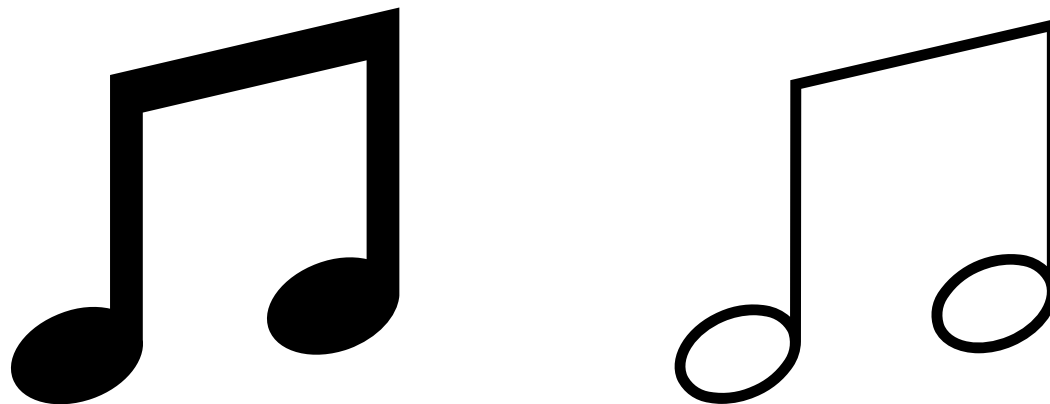
- 01 필요성
- 02 요구사항 분석
- 03 개념적 설계 - ERD
- 04 논리적 설계
- 05 물리적 설계



Part 1

# 필요성





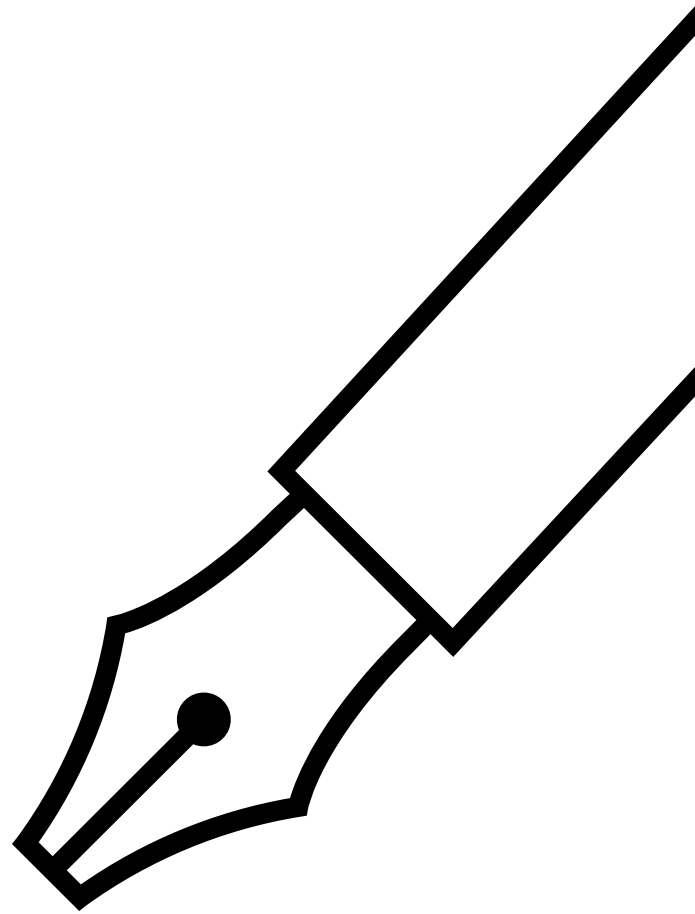
음원사이트에는 수많은 가수와 곡들의 정보가 저장되어 있다.

나는 내가 평소에 듣는 가수들의 음악만 따로 관리하고 싶었다.

MFC를 이용한 가수 정보시스템을 만들자.

Part 2

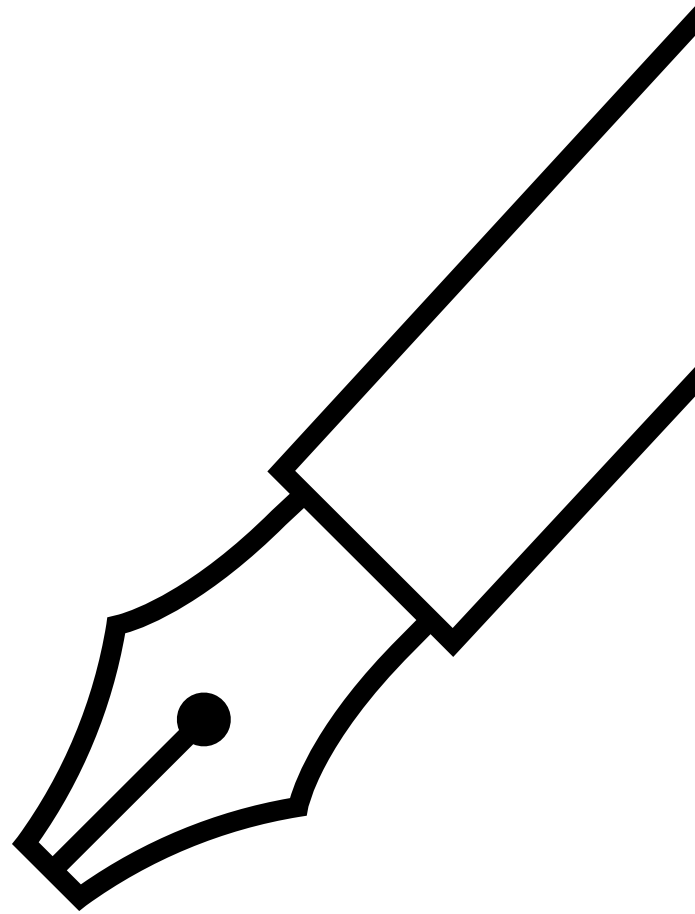
# 요구사항 분석



가수

Part 3

# 개념적 설계 - ERD



## Part 3 개념적 설계 - ERD

### 가수 정보

이름, 나이, 출생, 소속사, 데뷔 년도

가수 1

케이시, 26, 1995-10-06, 넥스타 엔터테인먼트, 2015

가수 2

전지윤, 31, 1990-10-15, 아츠로이엔티, 2009

가수 3

매드클라운, 36, 1985-03-25, Beautiful Noise, 2008



# singer\_info

+ singer_info							
가수정보							
가수 이름 + 짧은 가수 설명							
PK	AI	FK	Null	Logical Name	Name	Type	Comment
✓	✓	+		가수 아이디	singer_id	INT	
		+	✓	가수 이름	singer_name	VARCHAR(45)	
		+	✓	가수 나이	singer_age	INT	
		+	✓	출생	singer_birth	VARCHAR(45)	
		+	✓	소속사	singer_agency	VARCHAR(45)	
		+	✓	데뷔년도	singer_debut	INT	

## Part 3 개념적 설계 - ERD

가수 음악

이름, 곡명, 작사, 작곡, 장르, 발매일

노래 1

전지윤, 내가 해, 전지윤, 다비, 랩/힙합, 2016-11-02

노래 2

전지윤, Because, 전지윤, 다비, 알앤비/어반, 2018-01-13

노래 3

전지윤, 바꾸지마, 전지윤, 서재우, 랩/힙합, 2015-11-07

# singer\_song

singer_song							
가수 곡 정보							
모든 가수의 노래정보는 singer_song 테이블에 저장							
PK	AI	FK	Null	Logical Name	Name	Type	Comment
✓	✓	+		곡 아이디	song_id	INT	
		✓	✓	가수 이름	singer_name	VARCHAR(45)	
		+	✓	가수 곡명	song_name	VARCHAR(45)	
		+	✓	작사	song_lyric	VARCHAR(45)	
		+	✓	작곡	song_compose	VARCHAR(45)	
		+	✓	장르	song_genre	VARCHAR(45)	
		+	✓	발매일	song_release	VARCHAR(45)	

## Part 3 개념적 설계 - ERD

가수  
팬사이트

이름, 유튜브, 페이스북, 트위터, 인스타그램, 브이라이브

가수 1

케이시, kassy\_youtube, kassy\_instagram, kassy\_vlive

가수 2

전지윤, jenyer\_youtube, jenyer\_instagram, jenyer\_vlive

가수 3

매드클라운, madclown\_youtube, madclown\_vlive

# singer\_site

+ singer_site							
가수 팬사이트							
PK	AI	FK	Null	Logical Name	Name	Type	+
✓	✓	✚		사이트 아이디	site_id	INT	-
		✓	✓	가수 이름	singer_name	VARCHAR(45)	-
		✚	✓	유튜브	site_youtube	VARCHAR(45)	-
		✚	✓	페이스북	site_facebook	VARCHAR(45)	-
		✚	✓	트위터	site_twitter	VARCHAR(45)	-
		✚	✓	인스타그램	site_instagram	VARCHAR(45)	-
		✚	✓	브이라이브	site_vlive	VARCHAR(45)	-

SQL / Menu

## Part 3 개념적 설계 - ERD

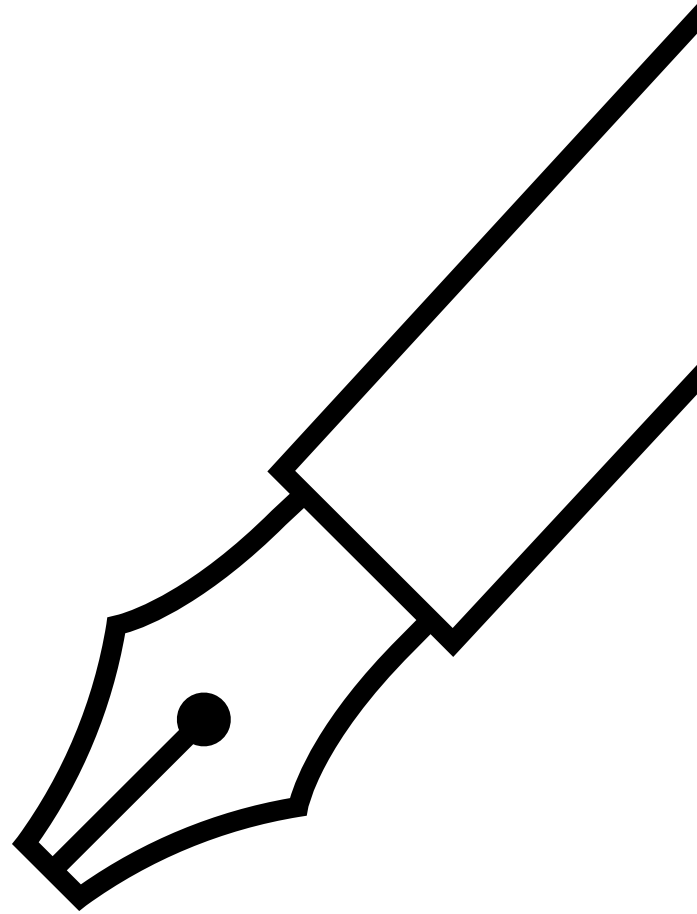
+ singer_info						
가수정보						
가수 이름 + 짧은 가수 설명						
PK	AI	FK	Null	Logical Name	Name	Type
✓	✓	+		가수 아이디	singer_id	INT
		+	✓	가수 이름	singer_name	VARCHAR(45)
		+	✓	가수 나이	singer_age	INT
		+	✓	출생	singer_birth	VARCHAR(45)
		+	✓	소속사	singer_agency	VARCHAR(45)
		+	✓	데뷔년도	singer_debut	INT
SQL / Menu						

+ singer_song						
가수 곡 정보						
모든 가수의 노래정보는 singer_song 테이블에 저장						
PK	AI	FK	Null	Logical Name	Name	Type
✓	✓	+		곡 아이디	song_id	INT
		✓	✓	가수 이름	singer_name	VARCHAR(45)
		+	✓	가수 곡명	song_name	VARCHAR(45)
		+	✓	작사	song_lyric	VARCHAR(45)
		+	✓	작곡	song_compose	VARCHAR(45)
		+	✓	장르	song_genre	VARCHAR(45)
		+	✓	발매일	song_release	VARCHAR(45)
SQL / Menu						

+ singer_site					
가수 팬사이트					
PK	AI	FK	Null	Logical Name	Name
✓	✓	+		사이트 아이디	site_id
		✓	✓	가수 이름	singer_name
		+	✓	유튜브	site_youtube
		+	✓	페이스북	site_facebook
		+	✓	트위터	site_twitter
		+	✓	인스타그램	site_instagram
		+	✓	브이라이브	site_vlive
SQL / Menu					

Part 4

# 논리적 설계



## singer\_info

Singer_id	Singer_name	Singer_age	Singer_birth	Singer_agency	Singer_debut
1	케이시	26	1995-10-06	넥스타 엔터테인먼트	2015
2	전지윤	31	1990-10-15	아츠로이엔티	2009
3	조동림(매드클라운)	36	1985-03-25	세임사이드 컴퍼니	2008



## singer\_song

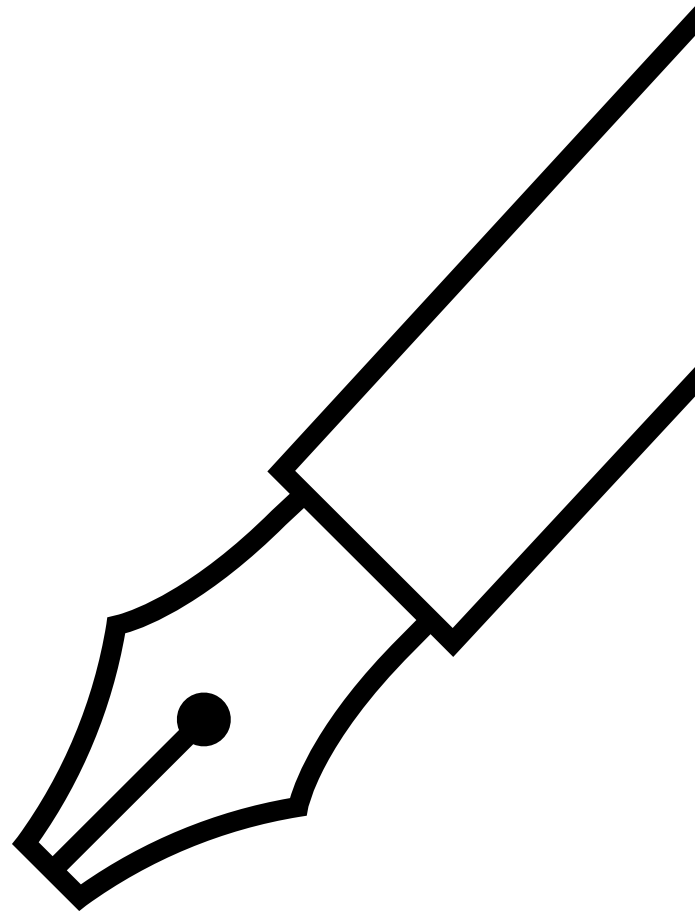
Song_id	Singer_name	Song_name	Song_lyric	Song_compose	Song_genre	Song_release
1	전지윤	내가 해	전지윤	다비	랩/힙합	2016-11-02
2	전지윤	Because	전지윤	다비	알앤비/어반	2018-01-13
3	전지윤	바꾸지마	전지윤	서재우	랩/힙합	2015-11-07

## singer\_site

Site_id	Singer_name	Site_youtube	Site_facebook	Site_twitter	Site_instgram	Site_vlive
1	케이시	Kassy_Official	Kassy_Official	Kassy_Official	Kassy_Official	Kassy_Official
2	전지윤	Jenyer_Official	Jenyer_Official	Jenyer_Official	Jenyer_Official	Jenyer_Official
3	매드클라운	Madclown_Official	Madclown_Official	Madclown_Official	Madclown_Official	Madclown_Official

Part 5

# 물리적 설계



## singer\_info

```
CREATE TABLE singer_info
(
    `singer_id`      INT          NOT NULL      AUTO_INCREMENT COMMENT '가수 아이디',
    `singer_name`    VARCHAR(45)  NULL          COMMENT '가수 이름',
    `singer_age`     INT          NULL          COMMENT '가수 나이',
    `singer_birth`   VARCHAR(45)  NULL          COMMENT '출생',
    `singer_agency`  VARCHAR(45)  NULL          COMMENT '소속사',
    `singer_debut`   INT          NULL          COMMENT '데뷔년도',
    PRIMARY KEY (singer_id)
);
```

```
ALTER TABLE singer_info COMMENT '가수 이름 + 짧은 가수 설명';
```

## singer\_song

```
CREATE TABLE singer_song
(
    `song_id`          INT          NOT NULL      AUTO_INCREMENT COMMENT '곡 아이디',
    `singer_name`      VARCHAR(45)  NULL         COMMENT '가수 이름',
    `song_name`        VARCHAR(45)  NULL         COMMENT '가수 곡명',
    `song_lyric`       VARCHAR(45)  NULL         COMMENT '작사',
    `song_compose`     VARCHAR(45)  NULL         COMMENT '작곡',
    `song_genre`       VARCHAR(45)  NULL         COMMENT '장르',
    `song_release`     VARCHAR(45)  NULL         COMMENT '발매일',
    PRIMARY KEY (song_id)
);

ALTER TABLE singer_song COMMENT '모든 가수의 노래정보는 singer_song 테이블에 저장';
```

## singer\_site

```
CREATE TABLE singer_site
(
    `site_id`          INT          NOT NULL    AUTO_INCREMENT COMMENT '사이트 아이디',
    `singer_name`      VARCHAR(45)  NULL       COMMENT '가수 이름',
    `site_youtube`     VARCHAR(45)  NULL       COMMENT '유튜브',
    `site_facebook`    VARCHAR(45)  NULL       COMMENT '페이스북',
    `site_twitter`     VARCHAR(45)  NULL       COMMENT '트위터',
    `site_instagram`   VARCHAR(45)  NULL       COMMENT '인스타그램',
    `site_vlive`       VARCHAR(45)  NULL       COMMENT '브이라이브',
    PRIMARY KEY (site_id)
);

ALTER TABLE singer_site COMMENT '가수 팬사이트';
```