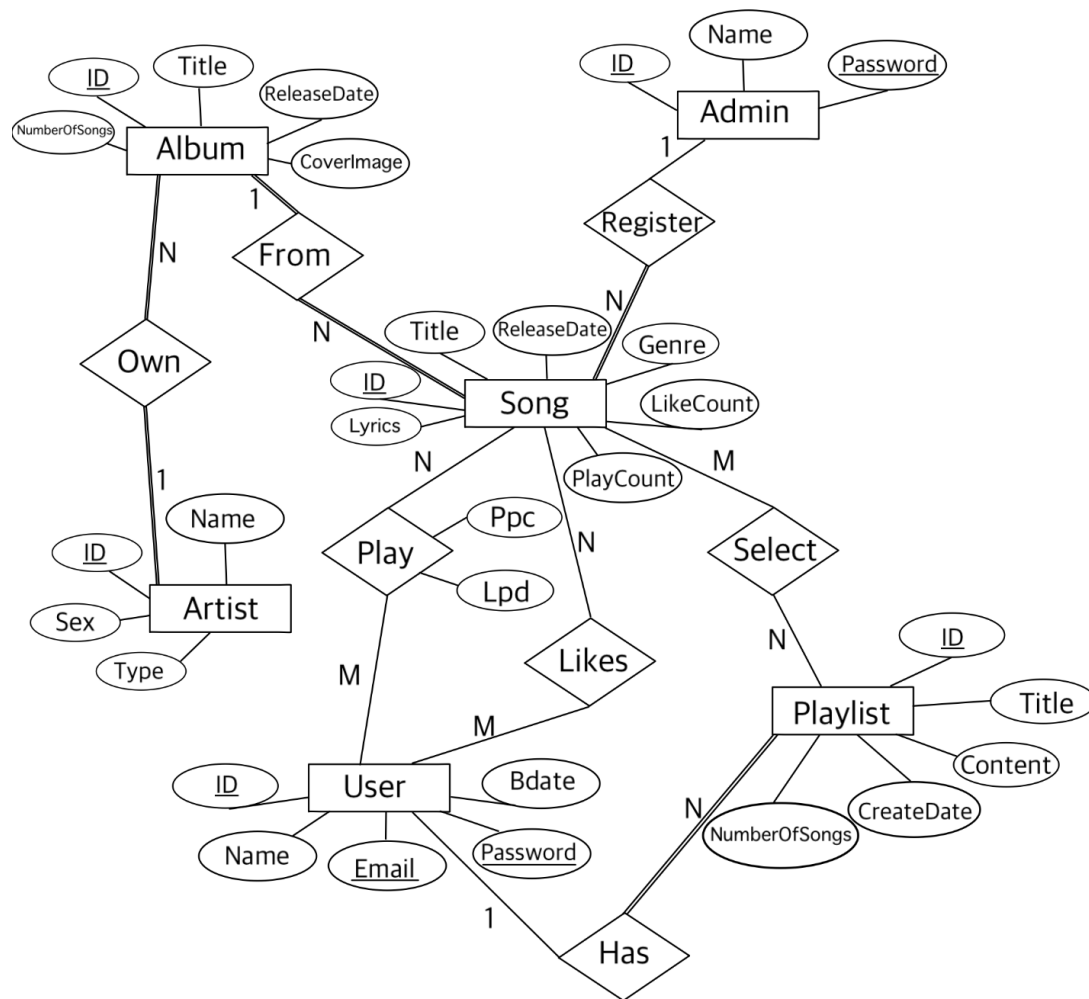


## **Project 2:** Conceptual modeling, Relational modeling

컴퓨터소프트웨어학부

2022085069 손주은

## 1. Entity-Relationship (ER) Schema



### Entity Type

#### 1) User

ID	user들을 식별할 수 있는 key attribute
Name	user의 이름
Email, Password	본인 확인 용도로 쓰이며, key attribute에 해당한다
Bdate	생년월일 정보이며, 나이제한 등을 위해 필요하다

#### 2) Song

ID	각 곡을 식별할 수 있는 key attribute
Title / ReleaseDate / Genre / Lyrics	곡 제목 / 발매일 / 장르 / 노래가사
LikeCount, PlayCount	해당 곡의 총 좋아요 개수, 해당 곡의 총 재생 횟수 인기곡 추천 등에 사용될 정보이다.

### 3) Playlist

ID	플레이리스트를 식별하는 key attribute
Title / Content / CreateDate / NumberOfSongs	플레이리스트 제목 / 설명 내용 / 생성일 / 수록된 곡 수

### 4) Album

ID	앨범을 식별하는 key attribute
Title / ReleaseDate / CoverImage / NumberOfSongs	앨범명 / 발매일 / 커버 이미지 / 수록된 곡 수

### 5) Artist

ID	아티스트를 식별하는 key attribute
Name / Sex	아티스트 이름 / 성별
Type	솔로, 듀오, 그룹 중 하나

### 6) Admin

ID	관리자를 식별하는 key attribute
Password	ID와 함께 관리자 확인 용도로 사용되며, key attribute이다
Name	관리자 이름

## Relationship Type

#### 1) Own

Album, Artist간의 관계로, 앨범의 아티스트가 누구인지를 나타낸다.

앨범은 artist를 한 명만 가질 수 있고, 여러 명이라면 대표자 한 명만 기입한다.

아티스트는 여러 개의 앨범을 가질 수 있다.

앨범과 아티스트 모두 total participation constraint을 가진다.

#### 2) From

Album, Song간의 관계로, 해당 곡이 어떤 앨범에 속하는지를 나타낸다.

앨범 하나에 여러 곡이 수록될 수 있으며, 곡은 해당 앨범 하나에만 수록될 수 있다.

앨범과 곡 모두 total participation constraint을 가진다.

### 3) Register

Song, Admin간의 관계로, 곡을 누가 등록했는지를 나타낸다.

한 명의 관리자는 여러 곡을 등록할 수 있다.

Admin은 partial, Song은 total participation constraint을 가진다.

### 4) Play

Song, User간의 관계 중 하나로, 사용자가 곡을 재생했음을 나타낸다.

다수의 사용자가 하나의 곡을 재생할 수도 있고, 한 명의 사용자가 여러 곡을 재생할 수도 있다.

이 Relationship type의 attribute으로 Ppc, Lpd가 있다.

- Ppc: Personal play count로, Song entity의 PlayCount와 구별할 수 있게 Ppc로 이름 지었다.
- Lpd: Last play date을 의미하며, 재생 순서대로 default playlist를 관리하기 위해 추가한 attribute이다.

Song, User 모두 partial participation constraint이다.

### 5) Likes

Song, User간의 관계 중 하나로, 사용자가 곡에 좋아요 표시를 했음을 나타낸다.

다수의 사용자가 한 곡에 좋아요를 표시할 수 있고, 한 사용자가 여러 곡에 좋아요 표시를 할 수 있다.

Song, User 모두 partial participation constraint을 가진다.

### 6) Select

Song, Playlist간의 관계로, playlist에 해당 곡이 수록되어 있는지를 나타낸다.

하나의 곡이 여러 플레이리스트에 수록될 수 있으며, 한 플레이리스트는 여러 곡을 수록할 수 있다.

Song, Playlist 모두 partial participation constraint을 가진다.

## 7) Has

User, Playlist 간의 관계로, 플레이리스트의 소유자가 누구인지를 나타낸다.

한 명의 사용자는 여러 플레이리스트를 만들 수 있지만, 플레이리스트의 소유자는 한 명이어야 한다.

User는 partial, Playlist는 total participation constraint을 가진다.

## 2. Relational Schema

### User

<u>User_id</u>	UName	Email	Password	Bdate	
----------------	-------	-------	----------	-------	--

### Song

<u>Song_id</u>	Title	ReleaseDate	Genre	LikeCount	PlayCount	Register	From	Lyrics
----------------	-------	-------------	-------	-----------	-----------	----------	------	--------

### Playlist

<u>Playlist_id</u>	PITitle	Content	CreateDate	Owner	PnumOfSongs			
--------------------	---------	---------	------------	-------	-------------	--	--	--

### Album

<u>Album_id</u>	AbTitle	AbReleaseDate	Artist	NumOfSongs				
-----------------	---------	---------------	--------	------------	--	--	--	--

### Artist

<u>Artist_id</u>	AtName	Sex	Type					
------------------	--------	-----	------	--	--	--	--	--

### Admin

<u>Admin_id</u>	AdName	AdPassword						
-----------------	--------	------------	--	--	--	--	--	--

### Play

<u>Who</u>	<u>PSong</u>	Ppc	Lpd					
------------	--------------	-----	-----	--	--	--	--	--

### Likes

<u>Who</u>	<u>LSong</u>							
------------	--------------	--	--	--	--	--	--	--

### Select

<u>WhichSong</u>	<u>WhatPI</u>							
------------------	---------------	--	--	--	--	--	--	--

ER schema로 relational modeling 한 모습이다.

Attribute의 name이 달라진 경우는 중복을 방지하기 위해서 바꾼 것이다.

여기서는 Foreign key위주로 설명하겠다.

(Song의 Lyrics, Playlist와 Album의 NumOfSongs는 검토하면서 누락된 것을 발견하여 추가된 것으로, 선들과 겹쳐 보이는 점 양해 부탁드립니다.)

### 1) User

Rename: ID → User\_id, Name → UName

### 2) Song

Rename: ID → Song\_id

Foreign Key

- Register: 1:N의 관계에서 N-side인 Song에 attribute로 추가된 모습이다.
- From: 1:N의 관계에서 N-side인 Song에 attribute로 추가된 모습이다.

### 3) Playlist

Rename: ID → Playlist\_id, Title → PlTitle, NumberOfSongs → PnumOfSongs

Foreign Key

- Owner: 1:N의 관계에서 N-side인 Playlist에 attribute로 추가된 모습이다.

### 4) Album

Rename: ID → Album\_id, Title → AbTitle, ReleaseDate → AbReleaseDate, NumberOfSongs → NumOfSongs

Foreign Key

- Artist: 1:N의 관계에서 N-side인 Album에 attribute로 추가된 모습이다.

### 5) Artist

Rename: ID → Artist\_id, Name → AtName

### 6) Admin

Rename: ID → Admin\_id, Name → AdName, Password → AdPassword

여기서부터는 M:N 의 relationship type을 하나의 relation으로 맵핑한 것이다.

### 7) Play

Who: User relation의 PK를 가리키는 FK, PSong: Song relation의 PK를 가리키는 FK

Who, PSong을 합쳐서 이 테이블의 PK로 가진다.

### 8) Likes

Who: User relation의 PK를 가리키는 FK, LSong: Song relation의 PK를 가리키는 FK

Who, LSong을 합쳐서 이 테이블의 PK로 가진다.

### 9) Select

WhichSong: Song relation의 PK를 가리키는 FK, WhatPl: Playlist relation의 PK를 가리키는 FK

WhichSong, WhatPl을 합쳐서 이 테이블의 PK로 가진다.

## 3. 프로젝트1의 요구사항과 달라진 점

Entity	Attributes	Added attributes
User	User_id, 이름, 구독유형	Email, Password, Bdate
Admin	Admin_id, 이름	Password
Song	Song_id, 곡명, 아티스트, 장르, 수록앨범, 가사	ReleaseDate, LikeCount, PlayCount,
Album	Album_id, 앨범명, 아티스트, 커버 이미지, 발매일자, 장르, 수록된 곡 수	-
Artist	Artist_id, 이름, 규모 및 성별 정보, 장르	-
Playlist	Playlist_id, 제목, 소유자, 생성 날짜, 수록된 곡 수	content
Genre	Genre_id, 장르명	
Song_Tags	Tag_id, 태그명	

### User

- 구독유형을 삭제
- 로그인을 위한 email, password와 나이제한 등을 위한 Bdate을 추가.

### Admin

- password추가
- user와 admin구분은 id로 구분한다.
- Admin 로그인은 id와 password로 한다.

### Song

- 아티스트는 Song에서 관리하지 않는다. (앨범에서만 관리)
- ReleaseDate, LikeCount, PlayCount 추가.

Album, Artist: 장르만 삭제

Playlist: 설명 attribute(content) 추가

Genre, song\_tags entity는 삭제하였다.

Genre는 entity로 관리하지 않고 song의 attribute로 관리하기로 했다.

Song\_tags는 인기곡이나 최신곡을 관리하기 위해서 도입했던 entity인데, 곡의 재생횟수, 발매일 attribute로 관리할 수 있을 것 같아서 삭제하였다.