

음원 스트리밍 서비스를 위한 DB

- 프로젝트 3: 관계형 스키마를 이용한 논리적 (logical) DB 설계 -

USER

<u>UserIndex</u>	UserName	Phone	SSN	Address	isAdmin
------------------	----------	-------	-----	---------	---------

ARTIST

<u>ArtistIndex</u>	ArtistName
--------------------	------------

MUSIC

<u>MusicIndex</u>	MusicTitle	Lyricist	Composer	HitsNumber	Arranger	AlbumIdx
-------------------	------------	----------	----------	------------	----------	----------

MUSIC_GENRE

<u>MIndex</u>	<u>MGenre</u>
---------------	---------------

ALBUM

<u>AlbumIndex</u>	AlbumTitle	Type	Agency	ReleaseCompany	ReleaseDate
-------------------	------------	------	--------	----------------	-------------

ALBUM_GENRE

<u>AIndex</u>	<u>AGenre</u>
---------------	---------------

PLAYLIST

<u>PlaylistIndex</u>	PlaylistTitle	NumOfMusic	CreationDate	UserIdx
----------------------	---------------	------------	--------------	---------

MUSIC_MAKE

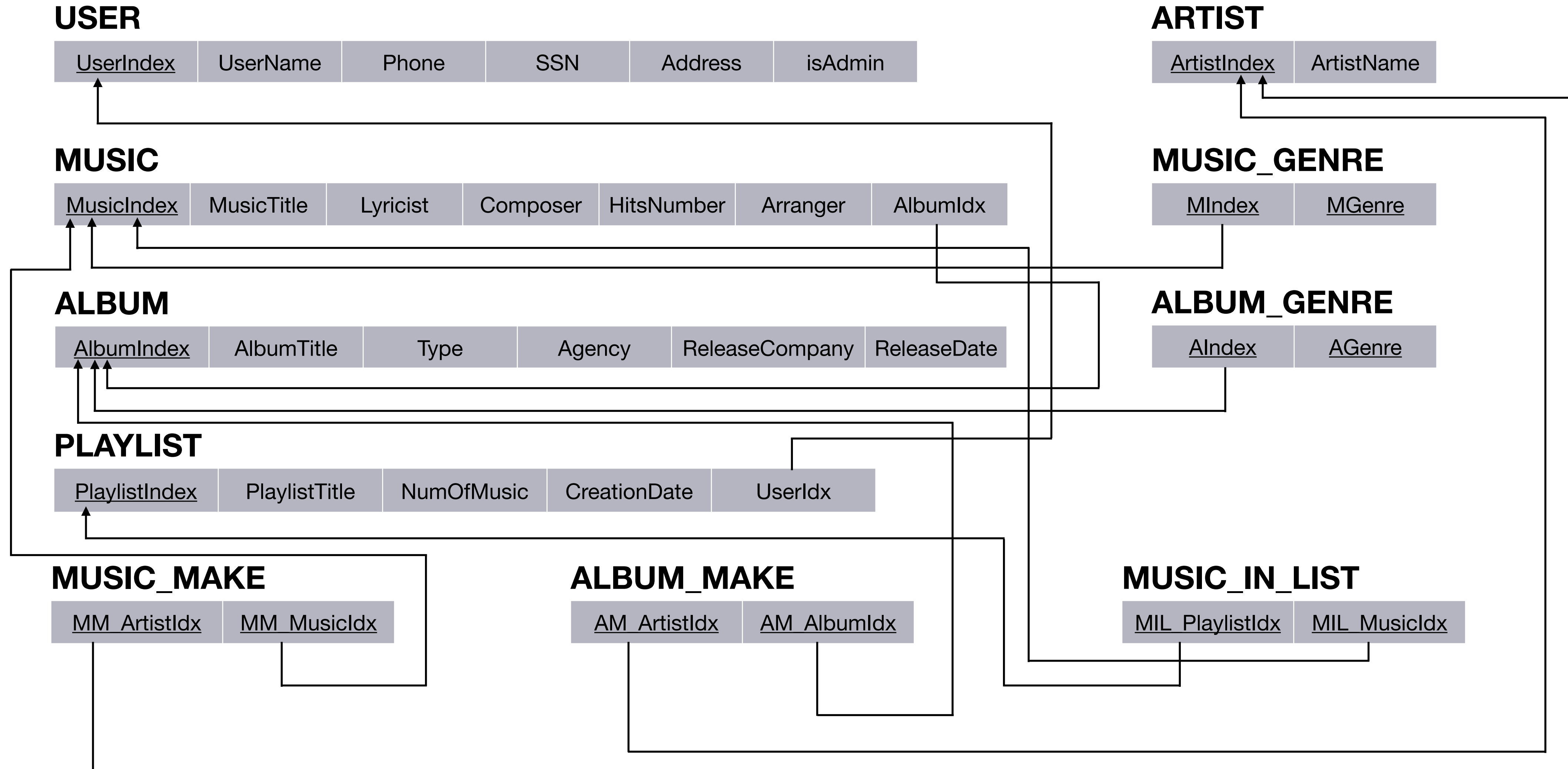
<u>MM_ArtistIdx</u>	<u>MM_MusicIdx</u>
---------------------	--------------------

ALBUM_MAKE

<u>AM_ArtistIdx</u>	<u>AM_AlbumIdx</u>
---------------------	--------------------

MUSIC_IN_LIST

<u>MIL_PlaylistIdx</u>	<u>MIL_MusicIdx</u>
------------------------	---------------------



1. Date 관련 Composite Attribute -> Simple Attribute로 교체

- ALBUM의 Composite Attribute였던 ReleaseDate와, PLAYLIST의 Composite Attribute였던 CreationDate를 Simple Attribute로 변경했다.
- 이유: 처음 DB를 디자인 할 때, Date Attribute의 Domain을 string 형식으로 밖에 표현할 수 없는 줄 알아서, 날짜를 integer 형식으로 저장하기 위해 Composite Attribute로 구상했다. 하지만 실제 DB에서, Date 형식으로 저장이 가능하기 때문에 하나의 Simple Attribute로 합쳤다.

ALBUM

<u>AlbumIndex</u>	AlbumTitle	Type	Agency	ReleaseCompany	ReleaseDate
-------------------	------------	------	--------	----------------	-------------

PLAYLIST

<u>PlaylistIndex</u>	PlaylistTitle	NumOfMusic	CreationDate	UserIdx
----------------------	---------------	------------	--------------	---------

2-1. USER

- USER Table 생성
- Simple Attribute: UserID, UserName, Phone, SSN, Address, isAdmin
- UserID는 PK이다.

USER

<u>UserID</u>	UserName	Phone	SSN	Address	isAdmin
---------------	----------	-------	-----	---------	---------

2-2-1. MUSIC

- MUSIC Table 생성
- Simple Attribute: MusicIndex, MusicTitle, Lyricist, Composer, HitsNumber, Arranger
- MusicIndex는 PK이다.

MUSIC

<u>MusicIndex</u>	MusicTitle	Lyricist	Composer	HitsNumber	Arranger
-------------------	------------	----------	----------	------------	----------

2-2-2. MUSIC_GENRE

- MUSIC_GENRE Table 생성
- MIndex: MUSIC Table의 PK인 MusicIndex를 FK로 include한다.
- MGenre: Multivalued Attribute인 Genre를 include한다.
- MIndex + MGenre가 PK이다.

MUSIC_GENRE

MIndex

MGenre

2-3-1. ALBUM

- ALBUM Table 생성
- Simple Attribute: AlbumIndex, AlbumTitle, Type, Agency, ReleaseCompany, ReleaseDate
- AlbumIndex는 PK이다.

ALBUM

<u>AlbumIndex</u>	AlbumTitle	Type	Agency	ReleaseCompany	ReleaseDate
-------------------	------------	------	--------	----------------	-------------

2-3-2. ALBUM_GENRE

- ALBUM_GENRE Table 생성
- AIndex: ALBUM Table의 PK인 AlbumIndex를 FK로 include한다.
- AGenre: Multivalued Attribute인 Genre를 include한다.
- AIndex + AGenre가 PK이다.

ALBUM_GENRE

AIndex

AGenre

2-4. ARTIST

- ARTIST Table 생성
- Simple Attribute: ArtistIndex, ArtistName
- ArtistIndex는 PK이다.

ARTIST

ArtistIndex

ArtistName

2-5. PLAYLIST

- PLAYLIST Table 생성
- Simple Attribute: PlaylistIndex, PlaylistName, NumOfMusic, CreationDate
- PlaylistIndex는 PK이다.

PLAYLIST

<u>PlaylistIndex</u>	PlaylistTitle	NumOfMusic	CreationDate
----------------------	---------------	------------	--------------

3-1. PLAYLIST_MAKE (1:N)

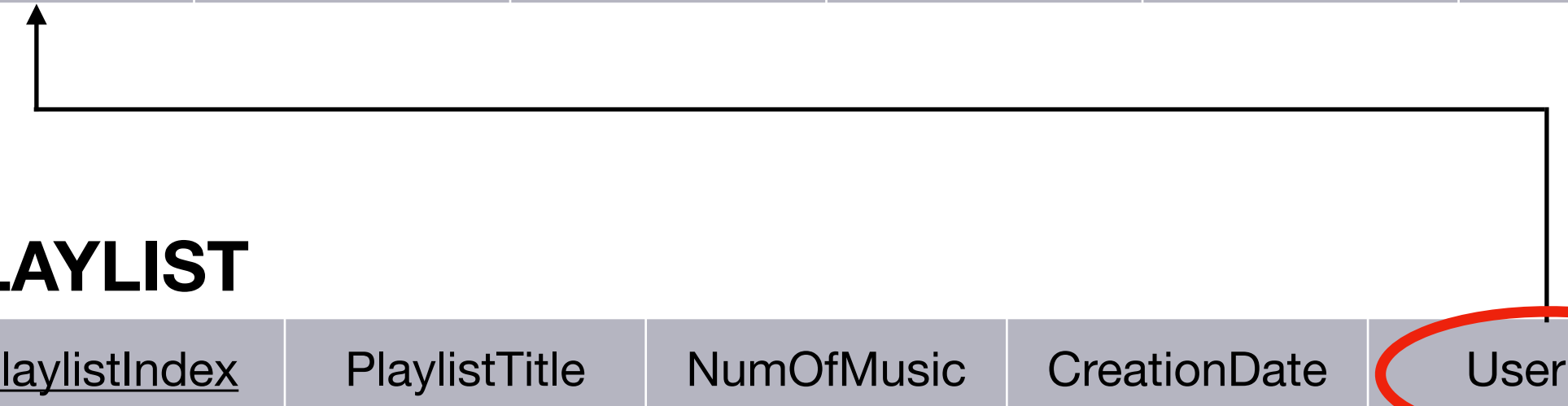
- PLAYLIST Table에 UserIdx 추가
- UserIdx: USER Table의 PK인 UserIndex를 PLAYLIST Table에 FK로 include한다.

USER

<u>UserIndex</u>	UserName	Phone	SSN	Address	isAdmin
------------------	----------	-------	-----	---------	---------

PLAYLIST

<u>PlaylistIndex</u>	PlaylistTitle	NumOfMusic	CreationDate	UserIdx
----------------------	---------------	------------	--------------	---------



3-2. MUSIC_CONTAIN (1:N)

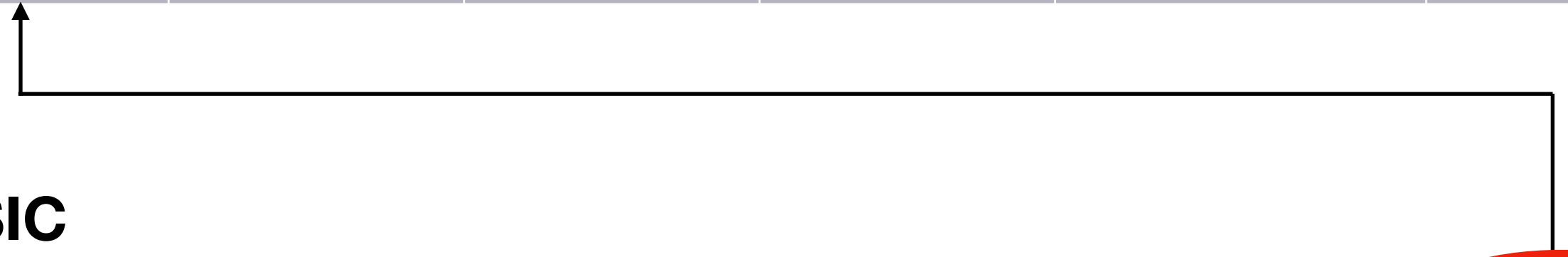
- MUSIC Table에 AlbumIdx 추가
- AlbumIdx: Album Table의 PK인 AlbumIndex를 Music Table에 FK로 include한다.

ALBUM

<u>AlbumIndex</u>	AlbumTitle	Type	Agency	ReleaseCompany	ReleaseDate
-------------------	------------	------	--------	----------------	-------------

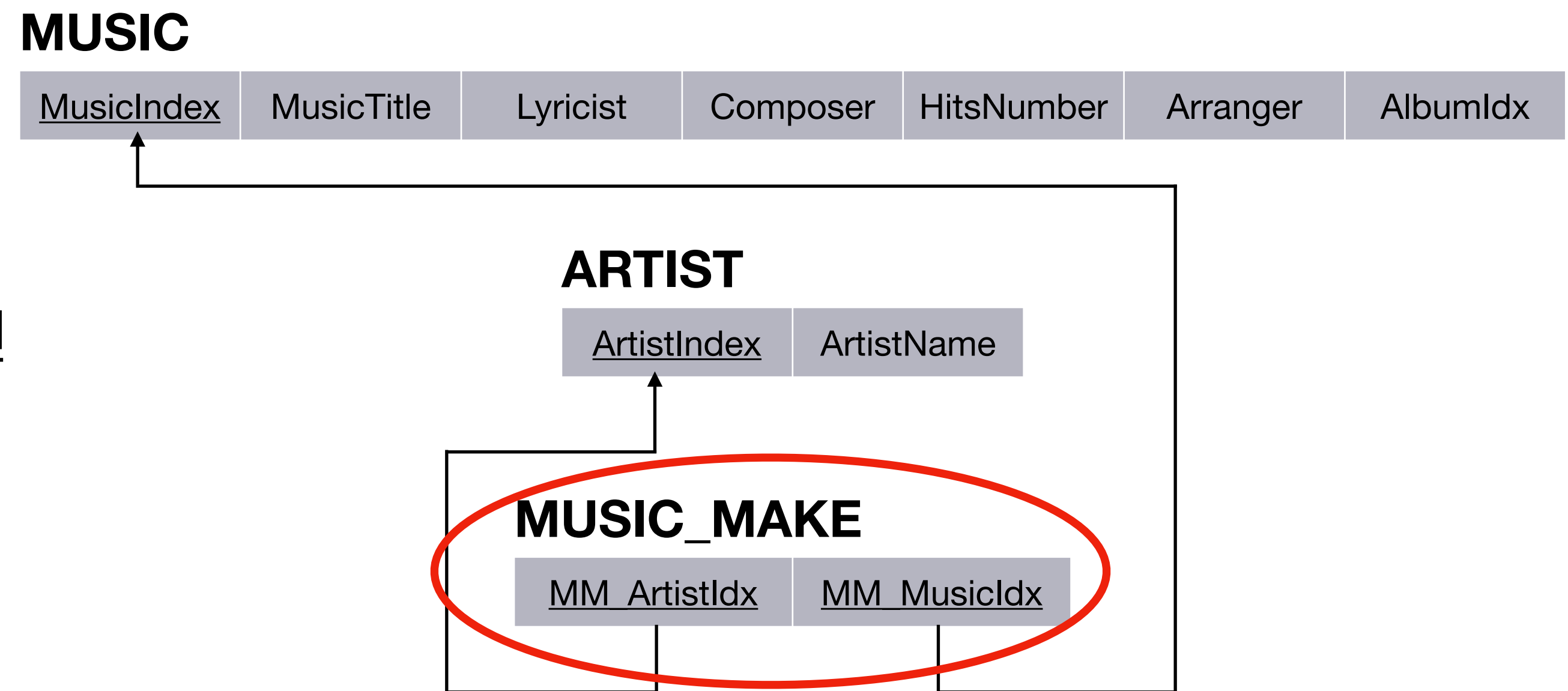
MUSIC

<u>MusicIndex</u>	MusicTitle	Lyricist	Composer	HitsNumber	Arranger	AlbumIdx
-------------------	------------	----------	----------	------------	----------	----------



3-3. MUSIC_MAKE (M:N)

- MUSIC_MAKE Table 생성
- MM_MusicIdx, MM_ArtistIdx: MUSIC Table의 PK인 MusicIndex와, ARTIST Table의 PK인 ArtistIndex를 include한다.
- MM_MusicIdx + MM_ArtistIdx를 PK로 한다.



3-4. ALBUM_MAKE (M:N)

- ALBUM_MAKE Table 생성
- AM_AlbumIdx, AM_ArtistIdx: ALBUM Table의 PK인 AlbumIndex와, ARTIST Table의 PK인 ArtistIndex를 include한다.
- AM_AlbumIdx + AM_ArtistIdx를 PK로 한다.

ALBUM

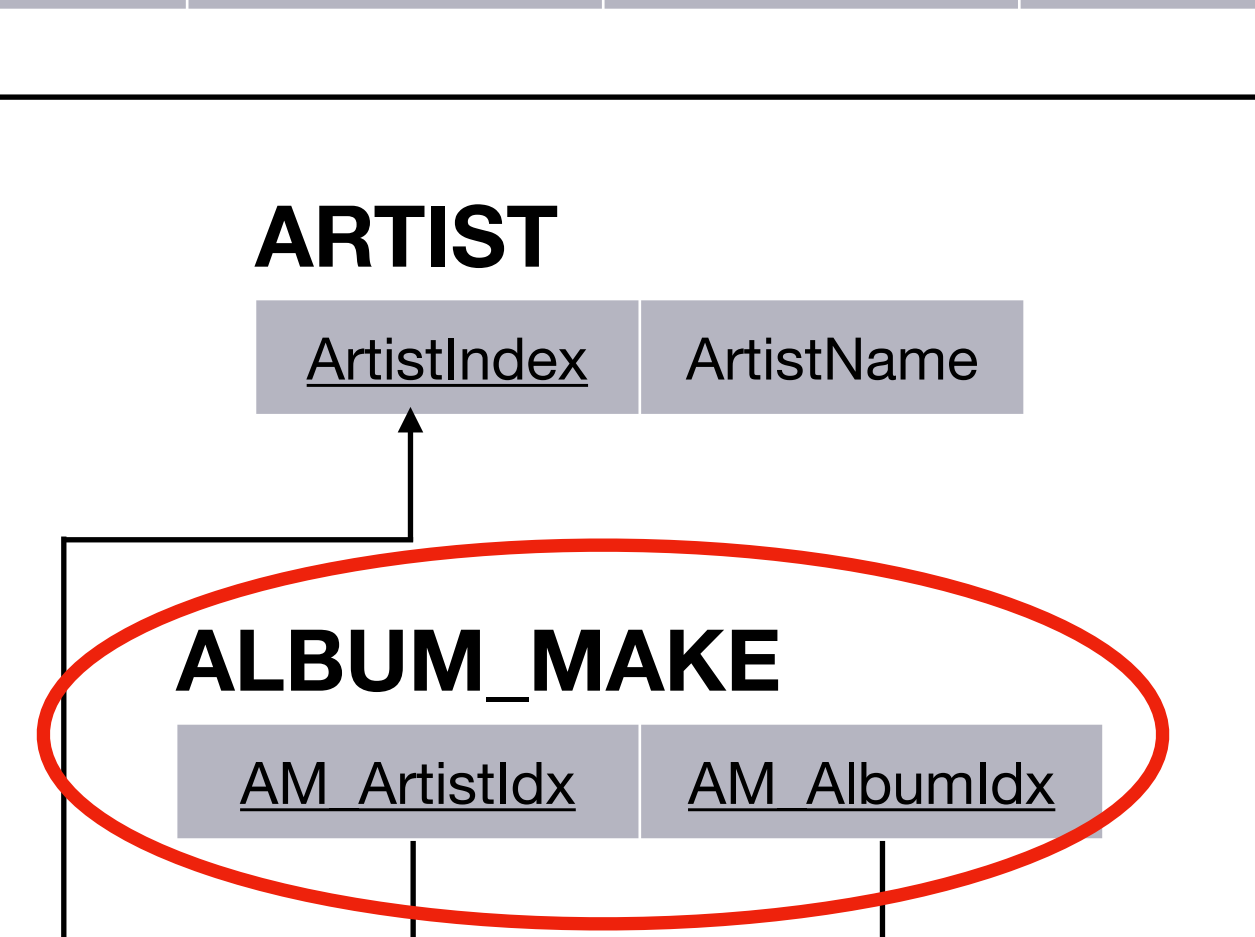
<u>AlbumIndex</u>	AlbumTitle	Type	Agency	ReleaseCompany	ReleaseDate
-------------------	------------	------	--------	----------------	-------------

ARTIST

<u>ArtistIndex</u>	ArtistName
--------------------	------------

ALBUM_MAKE

<u>AM_ArtistIdx</u>	<u>AM_AlbumIdx</u>
---------------------	--------------------



3-5. MUSIC_IN_LIST (M:N)

- MUSIC_IN_LIST Table 생성
- MIL_PlaylistIdx, MIL_MusicIdx: PLAYLIST Table의 PK인 PlaylistIndex와, MUSIC Table의 PK인 MusicIndex를 include한다.
- MIL_PlaylistIdx + MIL_MusicIdx를 PK로 한다.

MUSIC

<u>MusicIndex</u>	MusicTitle	Lyricist	Composer	HitsNumber	Arranger	AlbumIdx
-------------------	------------	----------	----------	------------	----------	----------

PLAYLIST

<u>PlaylistIndex</u>	PlaylistTitle	NumOfMusic	CreationDate	UserIdx
----------------------	---------------	------------	--------------	---------

MUSIC_IN_LIST

<u>MIL_PlaylistIdx</u>	<u>MIL_MusicIdx</u>
------------------------	---------------------

