음원 스트리밍 서비스를 위한 DB

- 프로젝트 1: 요구사항 분석 -

2017029916
양동해

1. 요구사항

- Entity로 음원 관리자, 스트리밍 구독자, 음원이 필요함
- 음원 관리자는 음원을 관리하여 등록 및 삭제가 가능함.
- 음원 관리자는 사용자 신상 또한 관리할 수 있음
- 사용자는 여러 플레이리스트를 만들 수 있으며, 각 음원들을 본인만의 플레이리스트에 등록할 수 있음
- 서비스는 각 음원이 플레이 된 횟수를 파악하여 각 음원의 통계 등을 제공할 수 있음

2. 요구사항 분석

<필요한 Entity 확인>
- 음원 관리자들이 관리되어야 함
- 스트리밍 구독자들이 관리되어야 함
- 사용자들이 들을 수 있는 음원들이 관리되어야 함
- 음원은 앨범 내에 존재하므로, 음원들이 들어가 있는 앨범들이 관리되어야 함
- 음원을 발매하는 아티스트들이 관리되어야 함
- 사용자들이 만든 플레이리스트들을 관리할 수 있어야 함

<추가적인 Constraints>
- 음원은 반드시 하나의 앨범에 속해야 함.
- 앨범은 싱글/EP, 정규앨범, 기타앨범 총 3가지로 분류됨.
- 아티스트 중 자신의 곡이 없는 아티스트가 존재할 수 있음.
- 앨범 내에는 음원이 존재하지 않는 경우를 허용함.
- 음원을 등록하기 위해선, 음원을 만든 아티스트와 그 음원이 포함되어 있는 앨범이 미리 만들어져 있어야 함.

<애플리케이션에서 구현>
- 음원 관리자는 음원을 등록 및 삭제할 수 있어야 함 => 음원을 등록하기 위해선 그 음원의 아티스트와 앨범도 존재해야 하므로, 각 관리자들은 아티스트와 앨범도 등록 및 삭제할 수 있음.
- 음원 관리자는 사용자의 신상을 관리할 수 있어야 함 => 각 관리자 마다 구독자 정보수정, 삭제를 진행할 수 있음. 구독자 정보수정을 통해 구독자의 전화번호 및 주소를 변경할 수 있고, 구독자 삭제를 통해 구독자를 삭제할 수 있음.

- 사용자는 여러 개의 플레이리스트를 만들 수 있으며, 음원들을 본인만의 플레이리스트에 등록할 수 있어야 함 => 각 구독자 마다 플레이리스트 생성, 수정, 삭제를 진행할 수 있음. 생성을 통해 처음에 음원들을 미리 추가할 수 있으며, 수정에서 음원을 낱개로 추가 및 삭제가 가능하고, 플레이리스트 삭제를 하면 삭제가 진행됨.
- 음원이 플레이 된 횟수를 파악하여, 음원의 속성 값인 전체재생수에 반영해야 함 => 애플리케이션에 재생 기능을 만들어, 재생을 한 번할 때마다 그 음원의 전체재생수에 1을 더함
- 음원의 전체재생수를 통해, 각 음원의 통계를 제공할 수 있어야 함 => 음원에 저장되어 있는 전체재생수로 정렬하여 내림차순으로 출력

3. Entity의 결정

Entity	Attribute
관리자	관리자번호(index)(1), 이름(1), 연락처(1), 주민번호(1), 주소(1)
구독자	구독자번호(index)(1), 이름(1), 연락처(1), 주민번호(1), 주소(1), 플레이리스트(PLAYLIST_MAKE)
음원	음원번호(index)(1), 곡명(1), 아티스트(MUSIC_MAKE), 장르(S>=1), 작사(1), 작곡(1), 편곡(1), 앨범(MUSIC_CONTAIN), 전체재생수(1), 플레이리스트(MUSIC_IN_LIST)
앨범	앨범번호(index)(1), 앨범명(1), 아티스트(ALBUM_MAKE), 앨범유형(1), 발매일(년도, 월, 일), 장르(S>=1), 발매사(1), 기획사(1), 음원(MUSIC_CONTAIN)
아티스트	아티스트번호(index)(1), 아티스트명(1), 앨범(ALBUM_MAKE), 음원(MUSIC_MAKE)
플레이리스트	플레이리스트번호(index)(1), 플레이리스트명(1), 음원(MUSIC_IN_LIST), 구독자(PLAYLIST_MAKE), 곡수(1), 최초생성(1)

4. 각 Entity의 Attribute 결정

① **관리자(음원 관리자)**

- 관리자번호는 관리자들을 구분하는 index이다. 이 속성은 Key Attribute로써, 모든 관리자들 마다 다르게 부여된다.
- 관리자의 기본 인적사항인 이름, 연락처, 주민번호, 주소를 기재한다. 여기서 나온 각각의 Attribute는 하나만 존재하기 때문에, 연락처와 주소의 경우 대표 번호, 대표 주소만을 기재한다.
- 관리자는 음원 Entity를 등록 및 삭제할 수 있다.
- 관리자는 구독자 Entity의 신상을 관리할 수 있다.

② **구독자(스트리밍 구독자)**

- 구독자번호는 구독자들을 구분하는 index이다. 이 속성은 Key Attribute로써, 모든 구독자들 마다 다르게 부여된다.
- 구독자의 기본 인적사항인 이름, 연락처, 주민번호, 주소를 기재한다. 여기서 나온 각각

의 Attribute는 하나만 존재하기 때문에, 연락처와 주소의 경우 대표 번호, 대표 주소만을 기재한다.
- 구독자는 여러 개의 플레이리스트 Entity를 만들 수 있으며, 음원 Entity들을 본인만의 플레이리스트에 등록할 수 있다.
- 구독자와 플레이리스트는 1:N의 관계를 갖는다. 하나의 구독자는 여러 플레이리스트를 만들 수 있지만, 하나의 플레이리스트는 한 명의 구독자와만 관계를 맺는다. Relationship Type은 PLAYLIST_MAKE이다. 구독자는 플레이리스트를 만들 수도 있고 안 만들 수도 있으므로 partial participation이지만, 플레이리스트는 한 명의 구독자가 반드시 존재 해야하므로 total participation이다.

③ **음원**
- 음원번호는 음원들을 구분하는 index이다. 이 속성은 Key Attribute로써, 모든 음원들마다 다르게 부여된다.
- 각 음원은 기본 분류 요소인 곡명을 필수로 가지고 있어야 한다. 곡명은 하나만 가질 수 있다.
- 음원과 아티스트는 M:N의 관계를 갖는다. 하나의 음원은 한 명 이상의 아티스트에 의해 만들어지고, 하나의 아티스트는 여러 음원을 만들 수 있다. Relationship Type은 MUSIC_MAKE이다. 음원은 반드시 한 명 이상의 아티스트에 의해 만들어지므로 total participation이지만, 아티스트는 자신의 음원이 없는 경우를 허용하므로 partial participation이다.
- 음원은 장르에 의해서 구분되며, 최소 1개 이상의 장르를 가지지만 여러 개의 장르를 가질 수 있으므로 Multivalued Attribute이다.
- 음원을 제작한 작사가와 작곡가, 편곡자는 각각 작사, 작곡, 편곡에 기재되며, 여러 명일 경우 대표자 한 명만을 기재한다.
- 앨범과 음원은 1:N의 관계를 갖는다. 하나의 앨범엔 여러 개의 음원이 존재할 수 있지만, 하나의 음원은 하나의 앨범에만 포함된다. Relationship Type은 MUSIC_CONTAIN이다. 앨범은 음원이 없는 경우를 허용하므로 partial participation이지만, 음원은 반드시 하나의 앨범에 포함되어야 하므로 total participation이다.
- 전체재생수는 음원을 재생한 전체 횟수를 의미한다. 값을 양의 정수 형태로 나타내기 때문에 전체재생수는 1개만 가질 수 있다. 초기에 음원이 생성될 땐 값을 0으로 초기화 한다.
- 음원이 플레이 된 횟수를 파악하여, 음원의 Attribute인 전체재생수에 반영한다. 이를 통해 각 음원의 통계를 제공할 수 있다.
- 음원과 플레이리스트는 M:N의 관계이다. 하나의 음원은 여러 플레이리스트에 들어갈 수 있으며, 하나의 플레이리스트도 여러 음원이 들어갈 수 있다. Relationship Type은 MUSIC_IN_LIST이다. 플레이리스트는 비어있는 경우가 존재할 수 있으며, 음원 역시 반드시 하나 이상의 플레이리스트에 들어갈 필요가 없으므로 둘 다 partial participation이다.

④ **앨범**
- 앨범번호는 앨범들을 구분하는 index이다. 이 속성은 Key Attribute로써, 모든 앨범들

마다 다르게 부여된다.
- 각 앨범은 기본 분류 요소인 앨범명을 필수로 가지고 있어야 한다. 앨범명은 하나만 가질 수 있다.
- 앨범과 아티스트는 M:N의 관계를 갖는다. 하나의 음원은 여러 아티스트에 의해 만들어질 수 있으며, 하나의 아티스트는 여러 음원을 만들 수 있기 때문이다. Relationship Type은 ALBUM_MAKE이다. 앨범은 반드시 한 명 이상의 아티스트에 의해 만들어지므로 total participation이지만, 아티스트는 자신의 앨범이 없는 경우를 허용하므로 partial participation이다.
- 앨범은 싱글/EP, 정규앨범, 기타앨범으로만 구분되며, 이는 앨범유형에 기재된다. 각 앨범은 반드시 하나의 앨범유형이 존재 해야한다.
- 발매일은 앨범을 발매한 날짜를 의미하며 년도, 월, 일을 각각 저장하므로, Composite Attribute이다.
- 앨범은 장르에 의해서 구분되며, 최소 1개 이상의 장르를 가지지만 여러 개의 장르를 가질 수 있으므로 Multivalued Attribute이다.
- 앨범은 반드시 하나의 발매사와 기획사를 가지며, 기획사가 존재하지 않을 경우 앨범의 Attribute인 아티스트명으로 대체한다.
- 앨범과 음원은 1:N의 관계를 갖는다. (음원에서 관계 설명)

⑤ **아티스트**
- 아티스트번호는 아티스트들을 구분하는 index이다. 이 속성은 Key Attribute로써, 모든 아티스트들 마다 다르게 부여된다.
- 각 아티스트의 기본 분류 요소인 아티스트명을 필수로 가지고 있어야 한다. 아티스트명은 하나만 가질 수 있다.
- 앨범과 아티스트는 M:N의 관계를 갖는다. (앨범에서 관계 설명)
- 음원과 아티스트는 M:N의 관계를 갖는다. (음원에서 관계 설명)

⑥ **플레이리스트**
- 플레이리스트번호는 플레이리스트들을 구분하는 index이다. 이 속성은 Key Attribute로써, 모든 플레이리스트들 마다 다르게 부여된다.
- 각 플레이리스트의 기본 분류 요소인 플레이리스트명을 필수로 가지고 있어야 한다. 플레이리스트명은 하나만 가질 수 있다.
- 음원과 플레이리스트는 M:N의 관계이다. (음원에서 관계 설명)
- 구독자와 플레이리스트는 1:N의 관계를 갖는다. (구독자에서 관계 설명)
- 곡수는 플레이리스트에 있는 음원의 수를 나타낸다. 즉, 음원의 개수에 따라 0 이상의 값을 갖는다. 값을 양의 정수 형태로 나타내기 때문에 곡수는 1개만 가질 수 있다.
- 최초생성은 플레이리스트를 생성한 날짜를 기재하며, 년도, 월, 일을 각각 저장하므로, Composite Attribute이다.

4. 최종 Entity Types, Attribute Types 및 Relationship Types

Entity	Attribute
ADMIN	AdminIndex, Name, Phone, SSN, Address
USER	UserIndex, Name, Phone, SSN, Address, PLAYLIST_MAKE
MUSIC	MusicIndex, Title, MUSIC_MAKE, {Genre}, Lyricist, Composer, Arranger, MUSIC_CONTAIN, HitsNumber, MUSIC_IN_LIST
ALBUM	AlbumIndex, Title, ALBUM_MAKE, Type, ReleaseDate(Year, Month, Day), {Genre}, ReleaseCompany, Agency, MUSIC_CONTAIN
ARTIST	ArtistIndex, Name, ALBUM_MAKE, MUSIC_MAKE
PLAYLIST	PlaylistIndex, Title, MUSIC_IN_LIST, PLAYLIST_MAKE, NumberOfMusic, CreationDate(Year, Month, Day)