**Project #2 - 음악 스트리밍 서비스 DBMS 개발**

1. **수정 사항**

**사용자:** 최근음악 attribute 대신 지금까지 재생했던 모든 음악에 대해 재생횟수를 기록하여 음악을 감상한 기록을 남기는 것으로 수정했다. 이는 재생이라는 relationship type으로 대체한다. 좋아요 attribute는 음악과 관계 있기 때문에 relationship type으로 변경했다. 플레이리스트는 이미 weak entity type으로 생성했기 때문에 따로 attribute가 필요 없다고 판단했다.

**음악:** 작사가와 작곡가는 여러명이 아닌 대표로 각각 한명씩만 기재되어도 충분하다고 판단해 수정했다. 음악의 장르도 하나만 저장될 수 있다. 총 스트리밍 횟수는 재생이라는 relationship type을 생성하여 해당 relationship의 attribute인 재생횟수에서 정보를 가져와 알아낼 수 있으므로 derived attribute로 변경했다.

**앨범:** 음악 attribute는 ‘수록되다’라는 relationship type으로 대체한다.

**플레이리스트:** weak entity type이므로 등록번호를 따로 두지 않기로 했다. 대신 제목을 partial key로 대체했다. 음악리스트는 ‘추가하다’라는 relationship type으로 대체한다.

1. **Entity-Relationship (ER) Model**

**그림, 도표, 라인, 스케치이(가) 표시된 사진

자동 생성된 설명**

**a. Entity & Attribute:**

**- 관리자:** 아이디가 key attribute이고, attributes로는 이름과 비밀번호가 있다.

**- 사용자:** 관리자처럼 아이디가 key attribute이고, attributes로는 이름과 비밀번호가 있다.

**- 음악:** 음악등록번호가 key attribute이고, 아티스트는 여러명일 수 있으니 multivalued attribute이며, 발매일, 작사가, 작곡가, 장르, 제목, 가사를 attribute로 갖는다. 총 재생 횟수는 재생 relationship type에서 알아낼 수 있기 때문에 derived attribute이다.

**- 앨범:** 앨범등록번호가 key attribute이고, 제목과 발매일을 attribute로 갖는다.

**- 플레이리스트:** 사용자에 의해 만들어지는 것으로, 같은 플레이리스트 제목이 존재하더라도 생성자에 따라 다른 플레이리스트로 구분한다. 사용자에 종속되는 것이기에 weak entity type으로 설정했다.

**b. Relationship:**

**- 사용자 관리:** 관리자와 사용자 사이의 relationship type으로, 관리자 1명이 여러 사용자를 관리한다. 그렇기에 1:N 관계이며, 모든 사용자는 관리하에 있기 때문에 total participation이고 관리자는 partial participation이다.

**- 음악 관리:** 관리자와 음악 사이의 관계로, 한명의 관리자가 여러 음악을 관리하므로 1:N 관계이다. 모든 음악은 관리자에 의해 관리되므로 total participation이고 관리자는 partial participation이다.

**- 좋아요:** 사용자와 음악 간의 관계로, 한명의 사용자가 좋아하는 여러 음악에 표시할 수 있고, 한 음악에 여러명의 사용자가 좋아요를 표시할 수 있으므로 M:N 관계이다. 두 entity 모두 partial participation이다.

**- 재생:** 사용자와 음악 간의 관계로, 한명의 사용자가 여러 음악을 감상할 수 있고, 한 음악에 여러명의 사용자가 재생할 수 있으므로 N:M 관계이다. 여기에 사용자가 해당 음악을 몇번 재생했는지 기록하기 위해 재생횟수 attribute를 갖는다. 두 entity 모두 partial participation이다.

**- 수록되다:** 앨범과 음악 사이의 관계로, 한 앨범에 여러 음악이 수록 될 수 있는 반면 한 음악은 한 앨범에만 수록 될 수 있으므로 1:N 관계를 갖는다. 이때 각 앨범에는 반드시 하나 이상의 음악이 수록되어 있어야 하므로 total participation이고, 앨범에 수록되지 않은 음악이 있을 수 있으니 음악은 partial participation이다.

**- 소유하다:** 사용자와 플레이리스트 간의 관계로, 한명의 사용자가 여러 플레이리스트를 생성할 수 있으며, 하나의 플레이리스트의 소유자는 한명이므로 1:N 관계를 갖는다. 플레이리스트가 weak entity type이므로 total participation constraint를 갖는다. 사용자는 partial participation이다. 해당 relationship은 플레이리스트가 사용자에 따라 같은 이름의 플레이리스트를 생성할 수 있으므로 identifying relationship이다. 이를 통해 사용자의 아이디로 identifying owner를 알아낼 수 있다.

**- 추가하다:** 플레이리스트와 음악 간의 관계로, 한 플레이리스트에 여러 음악을 추가할 수 있다. 여러 플레이리스트에서 같은 음악에 대해 추가할 수 있으므로 M:N 관계이다. 모든 플레이리스트는 반드시 음악을 하나 이상 포함하고 있어야 생성되기 때문에 total participation이다. 음악은 partial participation이다.

**III. Relation Model**

텍스트, 도표, 평면도, 직사각형이(가) 표시된 사진

자동 생성된 설명

**a. Mapping of Regular Entity Type**

**- 관리자:** new relation을 만든 후 attributes로 관리자아이디, 이름, 비밀번호를 갖는다. Primary key로 key attribute인 관리자아이디를 설정한다.

**- 사용자:** new relation을 만든 후 simple attributes로 사용자아이디, 이름, 비밀번호를 갖는다. Primary key로 key attribute인 사용자아이디를 설정한다.

**- 음악:** new relation을 만든 후 simple attributes로 음악등록번호, 제목, 작사가, 작곡가, 장르, 가사, 발매일인 음악발표일을 갖는다. Primary key로 key attribute인 음악등록번호를 설정한다. 아티스트는 multivalued attribute로 atomic한 값이 아니기 때문에 제외 시킨다.

**- 앨범:** new relation을 만든 후 simple attributes로 앨범등록번호, 제목, 앨범발매일을 갖는다. Primary key로 key attribute인 앨범등록번호를 설정한다.

**b. Mapping of Weak Entity Type**

**- 플레이리스트:** new relation을 만든 후 simple attributes로 제목, 공유여부를 갖는다. Identifying owner를 위해 사용자 relation의 primary key인 사용자아이디를 소유자아이디라는 foreign key로 가져와 추가한다. 제목을 partial key로 설정하여 소유자아이디와 함께 primary key로 설정한다. Foreign key인 소유자아이디는 사용자의 primary key인 사용자아이디에 화살표로 표시된다. (소유자아이디 🡪 사용자아이디)

**c. Mapping of Multivalued Attributes**

**- 아티스트:** new relation을 만든 후 attributes로 multivalued attribute인 아티스트이름과원래 속해 있던 entity의 primary key를 음악번호라는 foreign key로 갖는다. (음악번호 🡪 음악등록번호) Primary key로 모든 attributes를 합하여 설정한다.

**d. Mapping of 1:N Relationship Types**

**- 사용자관리:** 1쪽인 관리자 relation의 primary key인 관리자아이디를 N쪽인 사용자 relation에 담당자아이디라는 foreign key로 가져온다. (담당자아이디 🡪 관리자아이디)

**- 음악 관리:** 1쪽인 관리자 relation의 primary key인 관리자아이디를 N쪽인 음악 relation에 담당자아이디라는 foreign key로 가져온다. (담당자아이디 🡪 관리자아이디)

**- 수록되다:** 1쪽인 앨범 relation의 primary key인 앨범등록번호를 N쪽인 음악 relation에 수록앨범번호라는 foreign key로 가져온다. (수록앨범번호 🡪 앨범등록번호)

**e. Mapping of M:N Relationship Types**

**- 좋아요:** 좋아요라는 new relation을 만든 후 관계를 맺고 있는 모든 entity types, 사용자와 음악의 primary key인 사용자아이디와 음악등록번호를 사용자아이디와 음악번호인 foreign key로 가져온다. (사용자아이디(좋아요) 🡪 사용자아이디(사용자), 음악번호 🡪 음악등록번호) Primary key로 foreign key들을 모두 조합하여 설정한다.

**- 재생:** 재생이라는 new relation을 만든 후 관계를 맺고 있는 모든 entity types, 사용자와 음악의 primary key인 사용자아이디와 음악등록번호를 사용자아이디와 음악번호인 foreign key로 가져온다. (사용자아이디(좋아요) 🡪 사용자아이디(사용자), 음악번호 🡪 음악등록번호) 더불어 재생횟수를 simple attribute로 갖는다. Primary key로 foreign key들(사용자아이디와 음악번호)을 모두 조합하여 설정한다.

**- 추가하다:** 음악 추가라는 new relation을 만든 후 관계를 맺고 있는 모든 entity types, 음악과 플레이리스트의 primary key인 음악등록번호와 플레이리스트의 제목과 소유자아이디를 foreign key로 가져온다. (음악번호 🡪 음악등록번호, 플레이리스트제목 🡪 제목(플레이리스트), 플레이리스트 소유자아이디 🡪 소유자아이디(플레이리스트)) Primary key로 이 세가지 foreign key들을 모두 조합하여 설정한다.