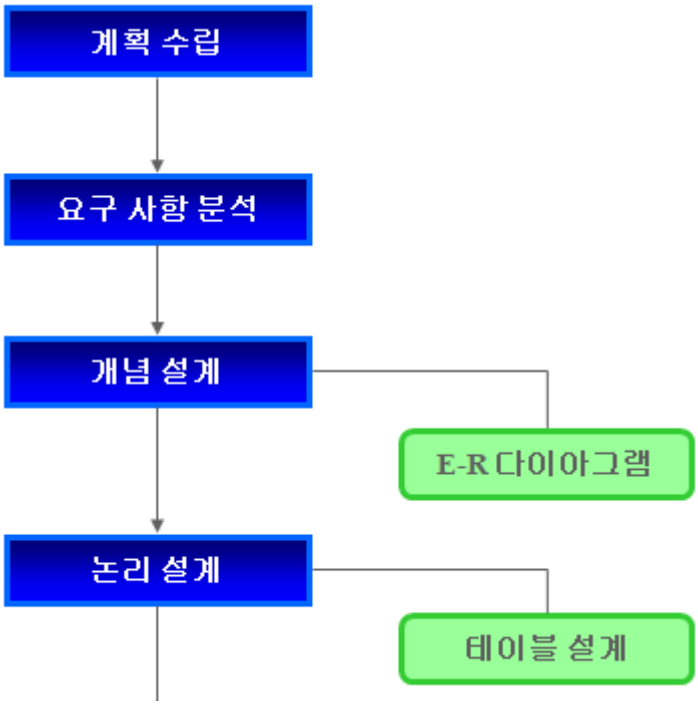
조민지 프로젝트: DBMS 프로그램 개발

데이터 베이스 구축 절차

1. 구축 목적을 정의한다.
2. DB 구축에 필요한 요구사항을 수집한다.
3. 수집된 요구사항을 정리하여, 요구사항 명세서를 작성한다.
4. 요구사항 명세서를 기반으로, DB의 개념적 설계를 한다.
   Entity-Relation 다이어그램을 나타낸다.
5. 논리적 설계를 한다.
   A. 구축할 DB의 데이터 모델에 맞춘다. ->관계형 데이터 모델을 사용한다.
   B. 속성의 특성을 정의한다 -> 속성들의 형과 크기
   C. 설계된 테이블에 맞게 예제를 작성한다.
6. DBMS 제품을 이용하여 구축한다.

1-5번까지는 문서화 작업이다.



소개

◆ 제목: 음악 플레이어 데이터 베이스
   ✓ 사용자와 음악에 관한 정보를 관리하는 음악 어플리케이션을 고려

◆ 목적

- ✓ 다수의 어플리케이션 프로그램을 가지는 음악 플레이어의 효율성과 성능향상을 도모하기 위해
- ✓ 각 기능이 요구하는 데이터들의 개별적인 파일 시스템의 관리보다는, 데이터베이스 시스템을 사용하여 파일 시스템이 가지는 단점을 극복하고 플레이리스트 관리에 편리하고 효율적인 환경을 제공하고자 한다.

◆ 음악 플레이어의 기능

- ✓ 필수 요소
  - ① 사용자: 음악 플레이어 서비스를 이용할 사람이 존재한다.
  - ② 관리자: 음악 플레이어 서비스를 제공해줄 사람이 존재한다.
  - ③ 서비스: 관리자가 사용자에게 제공할 수 있는 서비스가 존재한다.
    음악업데이트, 회원가입 ,플레이리스트 만들기 등

◆ 요구 조건 분석

- ✓ 요구사항의 명세
  - ① 사용자 (user)
    각각의 사용자는 유일한 user-id 와 user-ssn를 가진다. (중복을 허용하지 않는다)
    회원가입을 해야 한다.
    회원가입을 해지할 수 있다.
    회원등급을 선택 할 수 있다. (B 브론즈, S 실버 , G 골드 , P 플레)
    회원등급을 변경을 할 수 있다.
    사용자 한 명당 하나의 플레이리스트를 가진다.
    플레이리스트에서 음악을 추가할 수 있다.
    플레이리스트에서 음악을 삭제할 수 있다.
  - ② 관리자 (admin)
    admin 이라는 단 하나의 id 와 ssn를 가짐
    음악을 등록한다.
    음악을 삭제한다.
    중복된 음악은 등록하지 않는다.
    사용자의 회원가입 정보를 관리한다.
  - ③ 음악 (music)
    사용자의 플레이리스트에 음악이 추가된다.
    관리자에 의해 음악이 추가된다.
    관리자에 의해 음악이 삭제된다.
    사용자가 플레이리스트에 담은 수를 체크한다. (퍼가요 개수)
  - ④ 플레이리스트(Playlist)

사용자가 음악을 추가하거나 삭제한다.
탈퇴한 회원의 플레이리스트는 삭제된다. (퍼가요 개수는 수정되지 않음)

⑤ 회원등급 (level)
사용자의 등급을 나타낸다.
등급은 변경 가능하다
등급별로 플레이리스트에 추가할 수 있는 음악의 개수가 다르다. (플레이 리스트의 크기가 다르다)
플레이리스트의 곡을 추가할 수 있는지 없는지 user 에게 상태를 알려준다. (질의했을 경우)

개념적 설계

◆ 개체 (엔티티, entity)

✓ 사용자 (user)
사용자 이름 (user-name)
//사용자 성별 (user-sex)
사용자 주민번호 (user-ssn): 주 키
사용자 아이디 (user-id):중복허용 x
사용자 비밀번호(user-passwd)
사용자 플레이리스트 id (user-Plistid)
사용자 레벨 id (user-levelid)
전제조건: 같은 아이디를 가진 사용자는 존재하지 않는다. 유닉크

✓ 관리자 (admin)
관리자 아이디 (admin-id) 중복 허용 x
관리자 비밀번호(admin-pw)
관리자 주민번호 (admin-ssn) 주 키
전제조건: 관리자는 한명이다.

✓ 사용자들의 플레이리스트 (Plist)
플레이리스트 아이디 (Plist-id) 주 키
음악 아이디 (music-id)

✓ 음악 (music)
음악 아이디 (music-id) 주 키
곡 이름 (music-name)
곡 가수 (music-singer)

장르 (music-genre)
가사 (music-lyric)
퍼가요 수 (music-like)

- ✓ 등급 (level)
  등급 (level)
  등급 아이디 (level-id) 주키

◆ 관계타입 (릴레이션, relation)

- ✓ Plist-make
  User : 전체참여
  Plist : 전체참여
  사용자가 Plist 를 소유하고 있다는 뜻
  카디날리티 비율 : 1:1

- ✓ Music-contain
  Plist : 부분참여
  Music: 부분참여
  Plist 에 music 을 담는다는 뜻
  카디날리티 비율 : M:N

- ✓ Level-make
  User:전체참여
  Level: 전체참여
  사용자마다 등급이 있다는 뜻
  카디날리티 비율 : N:1

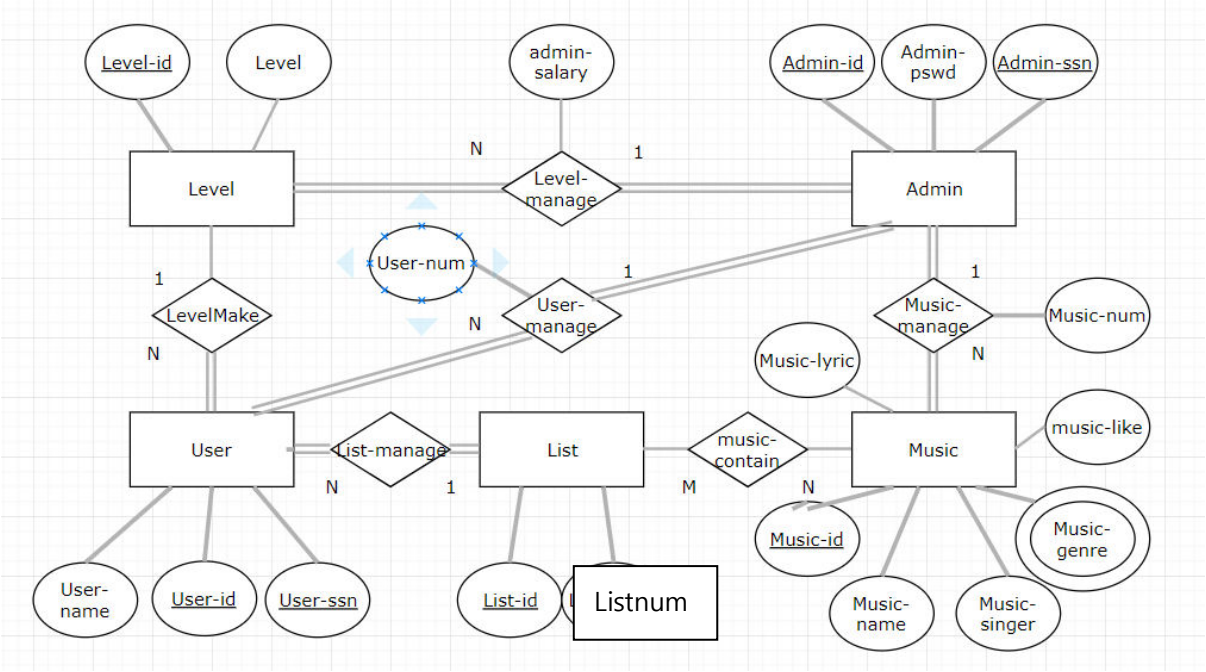
- ✓ User-manage
  User:전체참여
  Admin: 전체참여
  관리자가 user 를 관리한다는 뜻
  카디날리티 비율 : 1:N
  사용자의 개수 관계타입의 애트리뷰트가 있다.

- ✓ Music-manage
  Music: 전체참여
  Admin: 전체참여
  관리자가 music 을 관리한다는 뜻
  카디날리티 비율 : 1:N

Music 개수 관계차입의 애트리뷰트가 있다.

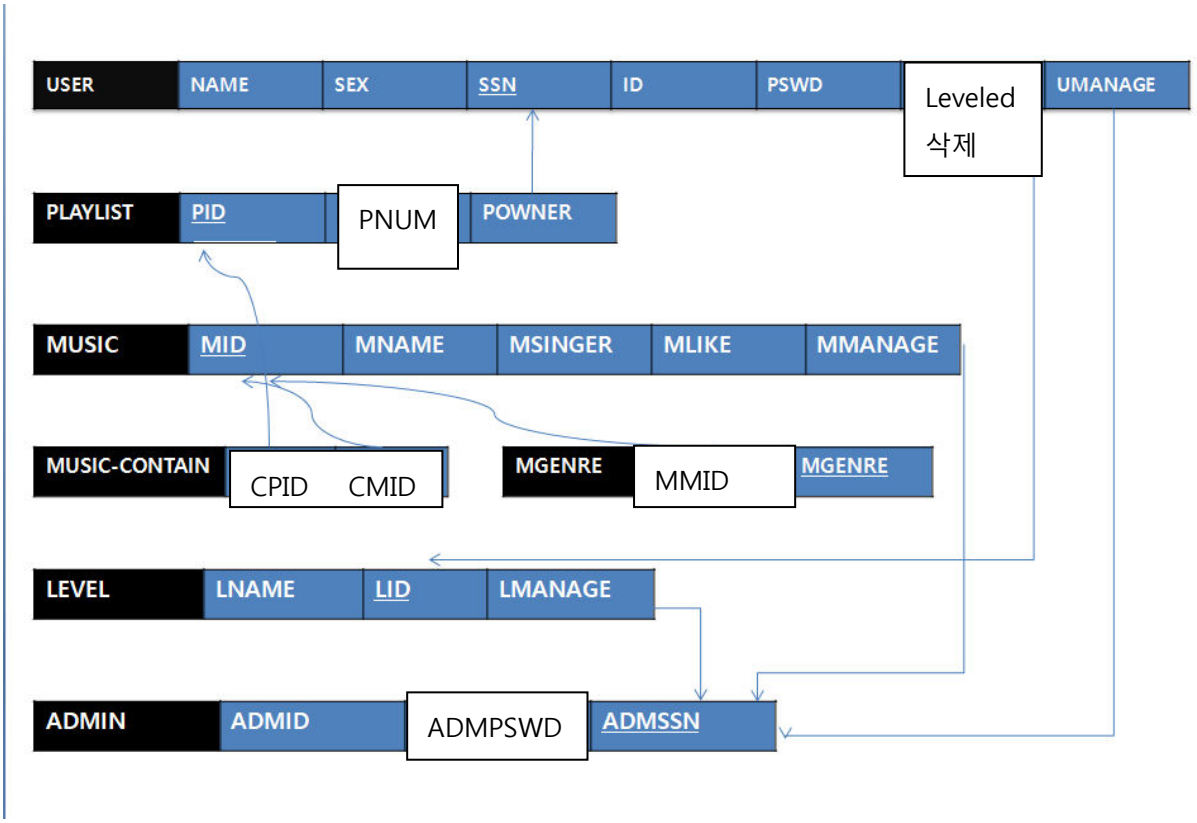
✓ Level-manage
Level : 전체참여
Admin: 전체참여
관리자가 level 을 관리한다는 뜻
카디날리티 비율 : 1:N
Level 의 각 개수를 구해 관리자의 월급을 구할 수 있다.

◆ E-R 다이어 그램



User-num: admin이 관리하는 사용자의 개수를 나타냅니다.
Music-num: admin이 관리하는 음악의 개수를 나타냅니다.
Music-like: 음악이 사용자의 Plist 에 추가될 때 증가하는 값입니다.
{Music-genre}: 장르는 여러 개의 애트리뷰트를 갖습니다. (ex 락, 발라드) 다치 애트리뷰트

◆ 개체관계로부터 관계 모델로의 전환



*검은 부분은 릴레이션의 이름입니다.

- ✓ USER의 기본키 PK 는 SSN 입니다.
- ✓ PLAYLIST의 기본키는 PID 입니다.
- ✓ MUSIC의 기본키는 MID 입니다.
- ✓ MUSIC-CONTAINT 의 기본키는 PID와 MID 를 결합한 키입니다.
- ✓ MGENRE

  집합 애트리뷰트인 GENRE를 새로운 테이블 MGENRE 로 지정하고 이 테이블에 MUSIC 의 기본키인 MID 와 MGENRE를 애트리뷰트로 포함합니다. 이 두 애트리뷰트를 기본기로 지정합니다.
- ✓ LEVEL의 기본키는 LID 입니다.
- ✓ ADMIN의 기본키는 ADMSSN 입니다.

◆ 관계 타입의 변환

- ✓ List-make

차수 2인 1:1의 관계타입
USER 의 기본키인 SSN 을 PLAYLIST에 외래키 POWNER 로 추가

- ✓ Music-contain
  차수 2인 N:M의 관계 타입
  PLAYLIST 의 기본키인 PID와 MUSIC 의 기본키인 MID를 애트리뷰트로 가지는 새로운 테이블을 생성 , 이 두 애트리뷰트들이 테이블의 기본키를 구성한다.

- ✓ Level-make
  차수 2인 N:1의 관계타입
  1인부분인 LEVEL 의 기본키인 LID를 N인 부분인 USER내에 외래키 LEVELID 로 추가

- ✓ User-manage
  차수 2인 M:N의 관계타입
  ~~1인 부분인 ADMIN의 기본키인 ADMSSN을 N인 부분인 USER내에 외래키 UMANAGE로 추가~~

- ✓ Music-manage
  차수 2인 M:N의 관계타입
  ~~1인 부분인 ADMIN의 기본키인 ADMSSN을 N인 부분인 MUSIC내에 외래키 MMANAGE로 추가~~

- ✓ Level-manage
  차수 2인 M:N의 관계타입
  ~~1인 부분인 ADMIN의 기본키인 ADMSSN을 N인 부분인 LEVEL 내에 외래키 LMANAGE로 추가~~