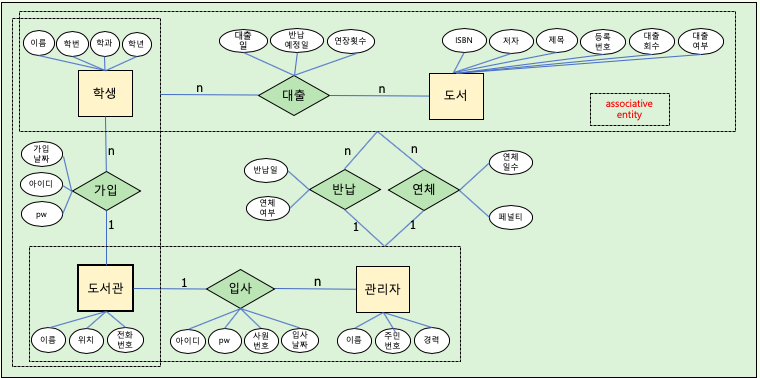
**DB설계문서 – LDB팀**

1. ERD



1. ERD에서 테이블 변환

학생(학번, 이름, 학과, 학년, 아이디, pw)

도서(ISBN, 저자, 제목, 출판사, 등록번호, 대출회수, 대출여부)

관리자(주민번호, 이름, 경력,, 아이디, pw, 사번, 입사일)

연체(학번, ISBN, 대출일, 연체일수, 페널티)

대출(학번, ISBN, 대출일, 반납예정일, 연장회수, 반납자사번, 반납일, 연체여부)

1. 데이터베이스, 테이블(5개), 쿼리문(26개), insert(22개)

drop database if exists library;  
create database library;  
  
use library;  
  
drop table if exists student;  
create table student(  
 sno int not null ,  
 sname varchar(20),  
 year int,  
 dept varchar(20),  
 id varchar(20) not null,  
 pw varchar(20) not null,  
 primary key(sno)  
);  
insert into student values(100, '최우연', 4, '컴퓨터공학부','wy0723','1234');  
insert into student values(200, '김수교', 4, '컴퓨터공학부','sk0293', '1234');  
insert into student values(300, '이현재', 4, '컴퓨터공학부','hj0123','1234');  
insert into student values(400, '김준수', 4, '컴퓨터공학부','js1942','1234');  
insert into student values(500, '박신혜', 3, '응용통계학과','sh9302','1234');  
insert into student values(600, '고준희', 3, '국어국문학과','jh9573','1234');  
insert into student values(700, '이진수', 3, '컴퓨터공학부', 'lee1048','1234');  
insert into student values(800, '김석현', 2, '컴퓨터','ssh4927','1234');  
insert into student values(900, '성민제', 2, '수학과','mj1027','1234');  
insert into student values(1000, '오승배', 2, '컴퓨터공학부','sb4058','1234');  
insert into student values(1100, '이찬혁', 1, '독어독문학과','ch4856','1234');  
insert into student values(1200, '방정식', 1, '독어독문학과','bombe12','1234');  
  
drop table if exists manager;  
create table manager(  
 rrn varchar(20) not null , #rrn = resident registration number  
 mname varchar(20),  
 career int,  
 id varchar(20) not null,  
 pw varchar(20) not null,  
 mno int not null,  
 dateOfEntry date,  
 primary key(mno)  
);  
insert into manager values('701225-1234567', '일반관리자', 20,'manager1','1415',1,'1999-04-16');  
insert into manager values('701225-1234561', 'root관리자', 20,'root','1415',0,'1999-04-16');  
  
drop table if exists book;  
create table book(  
 isbn varchar(20) ,  
 author varchar(20),  
 bname varchar(40),  
 publisher varchar(20) ,  
 bno varchar(20),  
 cnt int,  
 avail tinyint(1), # 대출여부 : 있으면 1 없으면 0  
 primary key(isbn)  
);  
insert into book values('9788994492032', '남궁','자바의정석','도우출판','8-00001',0,1);  
insert into book values('9788970509419', '최영규','두근두근 자료구조','생능출판사','8-00003',0,0);  
insert into book values('9791185459547', '페드로 도밍고스','마스터 알고리즘','비즈니스북스','8-00007',0,0);  
insert into book values('9791163030072', '보요 시바','Do it! 자료구조와 함께 배우는 알고리즘 입문: 자바 편','이지스퍼블리싱','8-00004',0,0);  
insert into book values('9788996094067', '윤성우','윤성우의 열혈자료구조','오렌지미디어','8-00005',0,1);  
  
  
drop table if exists borrow;  
create table borrow(  
 sno int not null ,  
 isbn varchar(20) ,  
 first date, #대출일  
 deadline date, #반납예정일  
 extraCnt int,  
 mno int,  
 back date, #반납일  
 late tinyint(1), #연체여부  
 primary key(sno, isbn, first),  
 foreign key(sno) references student(sno),  
 foreign key (isbn) references book(isbn)  
);  
  
insert into borrow values (1200, 9791163030072, '2019-04-01','2019-04-15',0,1,null,1);  
insert into borrow values (400, 9788970509419, '2019-04-15',null,0,1,null,0);  
insert into borrow values (100, 9791185459547, '2019-04-15','2019-04-29',0,1,null,0);  
  
drop table if exists overdue;  
create table overdue(  
 sno int not null ,  
 isbn varchar(20) ,  
 first date, #대출일  
 howlong int, #연체일  
 penalty long, #연체금  
 primary key(sno, isbn, first),  
 foreign key(sno) references student(sno),  
 foreign key (isbn) references book(isbn)  
);  
insert into overdue values (1200, 9791163030072, '2019-04-01',null,null);  
  
#리스트를 보여주는 쿼리들  
  
 /\*01사용자 리스트\*/  
 select sno as 학번, sname as 이름, year as 학년, dept as 학과 from student;  
  
 /\*02관리자 리스트\*/  
 select mno as 관리자번호, mname as 관리자, career as 경력, dateOfEntry as 입사일 from manager;  
  
 /\*03총 도서 리스트\*/  
 select bno as 책번호, bname as 책, author as 저자, publisher as 출판사, isbn  
 from book order by bno asc;  
  
 /\*04보유 도서 리스트\*/  
 select bno as 책번호, bname as 책, author as 저자, publisher as 출판사, isbn  
 from book where avail = 1 order by bno asc;  
  
 /\*05대출중인 도서 리스트\*/  
 select bno as 책번호, bname as 책, author as 저자, publisher as 출판사, borrow.isbn  
 from book, borrow  
 where book.isbn = borrow.isbn order by bno asc;  
  
 /\*06연체중인 도서 리스트\*/  
 select bno as 책번호, bname as 책, author as 저자, publisher as 출판사, overdue.isbn  
 from book, overdue  
 where book.isbn = overdue.isbn  
 order by bno asc;  
  
 /\*07연체자 명단\*/  
 select overdue.sno /\*as 학번\*/, sname as 연체자 from overdue, student  
 where overdue.sno = student.sno  
 order by sno asc;  
  
 /\*08 학년별 대출현황\*/  
 select year as 학년, *count*(borrow.sno) as 대출횟수  
 from borrow, student  
 where borrow.sno = student.sno  
 group by year;  
  
 /\*09 학과별 대출현황\*/  
 select dept as 학과, *count*(borrow.sno) as 대출횟수  
 from borrow, student  
 where borrow.sno = student.sno  
 group by dept;  
  
 /\*10 인기순위 도서리스트(전체기간)\*/  
 select bno as 책번호, bname as 책, author as 저자, publisher as 출판사, cnt as 대여횟수  
 from book  
 order by cnt desc;  
  
 /\*11 이번달 베스트 도서 리스트\*/  
 select bno as 책번호, bname as 책, author as 저자, publisher as 출판사, *count*(borrow.isbn) as 대여횟수  
 from borrow, book  
 where borrow.isbn = book.isbn and ( first > *LAST\_DAY*(*NOW*() - interval 1 month) AND first <= *LAST\_DAY*(*NOW*()) )  
 group by borrow.isbn;  
  
 /\*12 독서왕 리스트\*/  
 select borrow.sno as 학번, sname as 이름, *count*(borrow.sno) as 대여횟수  
 from student, borrow  
 where student.sno = borrow.sno and year <= 4  
 group by borrow.sno;  
  
 /\*13 졸업자 미반납 도서 리스트\*/  
 select overdue.sno as 학번, sname as 이름, bname as 책  
 from student, overdue, book  
 where overdue.isbn = book.isbn and student.sno = overdue.sno and year > 4;  
  
# 대여  
  
 /\*14 borrow 테이블에 value추가 (이때 반납일,반납예정일 다 자동설정),  
 필!, 자바에서 먼저 book의 avail이 1인지 확인하고 사용해야함  
 사용방법 : 학번과 isbn만 가져다 넣으면됨  
  
 insert into borrow values(<sno>,<isbn>,curdate(),date\_add(curdate(),interval 14 day),0,null,null,0);  
  
 \*/  
  
  
 # 필) 책 대여 후 꼭 처리해줘야하는 쿼리!!!  
 /\*15 도서의 이용여부(avail)를 업데이트 해주는 쿼리 변수 1개  
 update book  
 set avail = 0  
 where isbn = <대여한 책의 isbn>;  
 \*/  
  
 /\*16 대출 횟수 올리는 쿼리 변수 1개  
 update book  
 set cnt = cnt+1  
 where bno = <빌린책의 bno>;  
 \*/  
  
 /\*17 대출한 책의 반납예정일을 업데이트 하는 쿼리\*/  
 update borrow  
 set deadline = *date\_add*(first, interval 14 day)  
 where deadline is null;  
  
# 반납  
 /\*18 borrow 테이블의 반납일 업데이트 변수 2개  
 update borrow  
 set back = curdate(), mno = <현재 로그인되어있는 관리자의 mno>  
 where isbn = <반납하는 책의 isbn>;  
 \*/  
  
 /\*19 도서의 이용여부(avail)를 업데이트 해주는 쿼리  
 update book  
 set avail = 0  
 where isbn = <대여한 책의 isbn>;  
 \*/  
  
 /\*20 연체자의 경우 overdue 테이블에서 제거해야함  
  
 delete  
 FROM overdue  
 where sno = <반납자의 sno> and isbn = <반납된 책의 isbn> and first = <대출한 날짜>;  
 \*/  
  
# 도서 대출 연장 처리  
 /\*21 반납예정일 조정쿼리 (연체자 연장 제외처리, 2번 이상 연장 방지 처리) <isbn> 넣어주면됨  
 update borrow  
 set deadline = date\_add(deadline, interval 14 day), extraCnt = extraCnt +1  
 where isbn = <isbn> and extraCnt < 1 and isbn not in (select isbn  
 from overdue);  
  
 \*/  
  
#연체  
 /\*22 반납이 안된 도서 중 반납예정일이 지난 것을 overdue테이블에 삽입 \*/  
 replace  
 into overdue(sno, isbn, first,howlong, penalty)  
 select borrow.sno, isbn, first, *to\_days*(*now*())-*to\_days*(first), (*to\_days*(*now*())-*to\_days*(first)) \*100  
 from borrow, student  
 where borrow.sno = student.sno and borrow.deadline < *current\_date* and back is null and year < 5;  
  
  
  
# 학생들 학년 조정,졸업 처리 및 관리자 경력 조정 (대출중인 졸업자들 정보유지 처리완료)  
 /\*23 학년 조정\*/  
 update student  
 set year = year + 1;  
  
 /\*24 대출중인 사람의 정보를 잃지 않고 졸업자들의 정보는 삭제\*/  
 delete  
 from student  
 where year =5 and not sno in  
 (select sno  
 from borrow);  
  
 /\*25 관리자 경력조정\*/  
 update manager  
 set career = career +1;