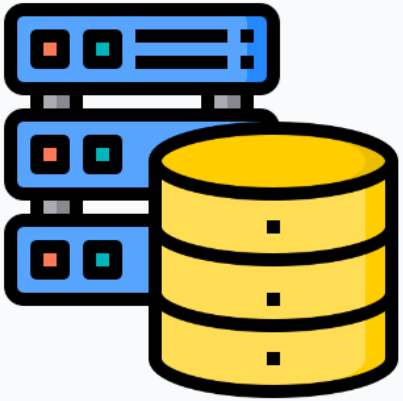
2장. 데이터 모델링

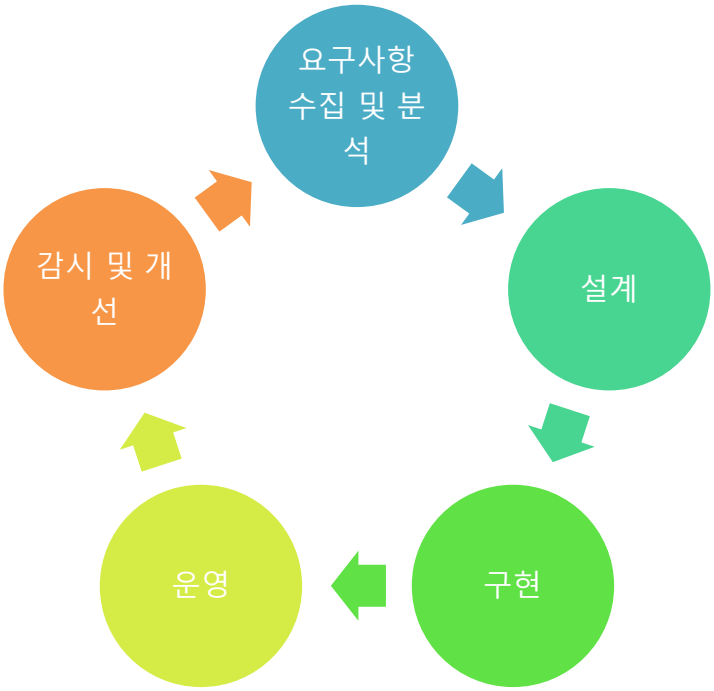
ER 모델



데이터 모델링

데이터 베이스 생명주기

데이터 베이스는 최초 사용자의 요구에 의해 구축되어 사용되다가 필요에 따라 개선 또는 다시 구축되어 사용된다.



데이터 모델링

① 요구사항 수집 및 분석
- 사용자들의 요구사항을 듣고 분석하여 데이터베이스 구축의 범위를 정하는 단계
- 마당서점의 경우 고객, 운영자, 경영자 등 사용자의 범위와 서비스 수준을 정하는 것

② 설계
- 분석된 요구사항을 기초로 주요 개념과 업무 프로세스 등을 식별하고(개념적 설계), 사용하는 DBMS의 종류에 따라 맞게 변환(논리적 설계)한 후, 데이터 베이스 스키마(물리적 설계)한다.

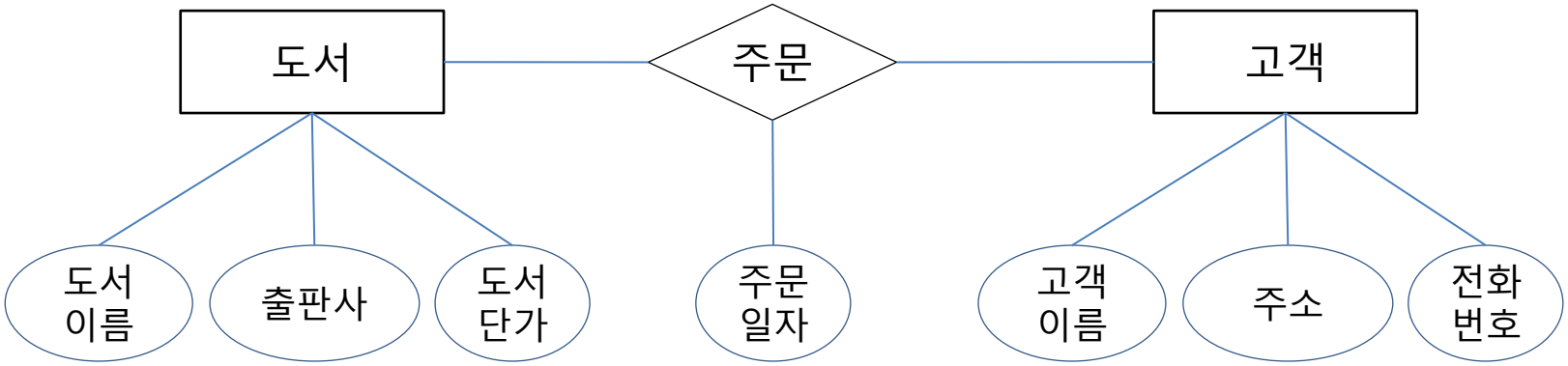
③ 구현
- 설계 단계에서 생성한 스키마를 실제 DBMS에 적용하여 데이블 및 관련 객체(뷰, 인덱스등)를 만든다.

④ 운영
- 구현된 데이터베이스를 기반으로 소프트웨어를 구축하여 서비스를 제공한다.

⑤ 감시 및 개선
- 데이터 베이스가 지속적으로 운영될 수 있도록 변경 및 유지 보수를 한다.

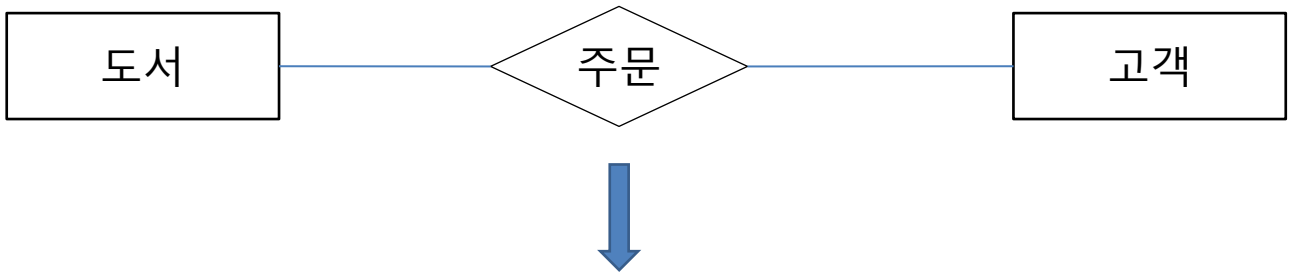
데이터 모델링

개념적 모델링



데이터 모델링

논리적 모델링



도서(<u>도서번호</u>, 도서이름, 출판사, 도서단가)

고객(<u>고객번호</u>, 고객이름, 주소, 전화번호)

주문(<u>주문번호</u>, 고객번호(FK), 도서번호(FK), 주문금액, 주문일자)

데이터 모델링

물리적 모델링

DBMS

도서(도서번호, 도서이름, 출판사, 도서단가)

```
CREATE TABLE book(
    bookid      NUMBER PRIMARY KEY,
    bookname    VARCHAR2(40),
    publisher   VARCHAR2(40),
    price       NUMBER
);
```

고객(고객번호, 고객이름, 주소, 전화번호)

```
CREATE TABLE customer(
    custid      NUMBER PRIMARY KEY,
    name        VARCHAR2(40),
    address     VARCHAR2(50),
    phone       VARCHAR2(20)
);
```

주문(주문번호, 고객번호(FK), 도서번호(FK), 주문금액, 주문일자)

```
CREATE TABLE orders(
    orderid     NUMBER PRIMARY KEY,
    custid      NUMBER,
    bookid      NUMBER,
    saleprice   NUMBER,
    orderdate   DATE,
    FOREIGN KEY(custid) REFERENCES customer(custid),
    FOREIGN KEY(bookid) REFERENCES book(bookid)
);
```

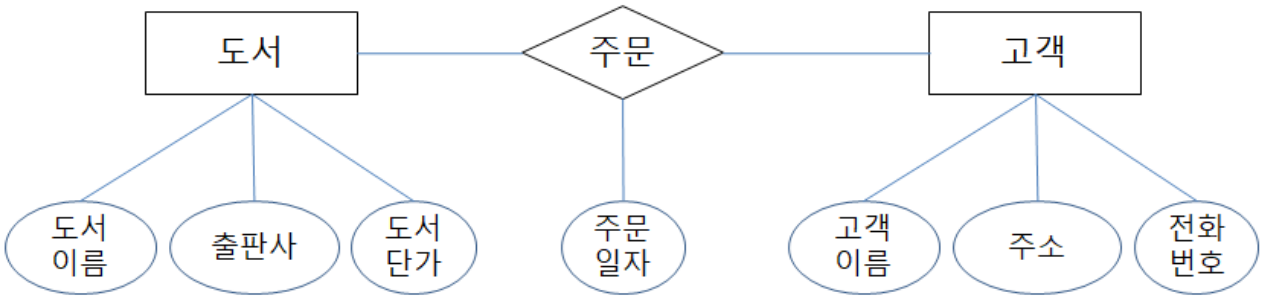
ER 모델

ER(Entity Relationship) 모델은 데이터 모델링 과정 중 개념적 모델링에서 사용하는 모델로 1976년 피터 첸이 제안하였다.

ER 모델은 세상의 사물을 개체와 개체 간의 관계로 표현한다.

개체는 독립적인 의미를 지니고 있는 유무형의 사람 또는 사물을 말하며, 개체의 특성을 나타내는 속성(attribute)으로 식별한다. 또한 개체끼리는 서로 관계를 가진다.

ER 모델은 개체와 개체간의 관계를 ER 다이어그램이라는 표준화된 그림으로 표현한다.



ER 모델

관계 대응 수에 따른 유형

- **일대일(1:1) 관계**

회사에서 사원이 개인별로 한 대의 컴퓨터만 사용한다면 사원과 컴퓨터는 일대일 관계이다.

- **일대다(1:N) 관계**

하나의 학과에는 여러 명의 학생이 소속되어 있어 일대다 관계로 표현할 수 있다.
N은 0이상의 자연수를 말한다.

ER 모델

관계 대응 수에 따른 유형

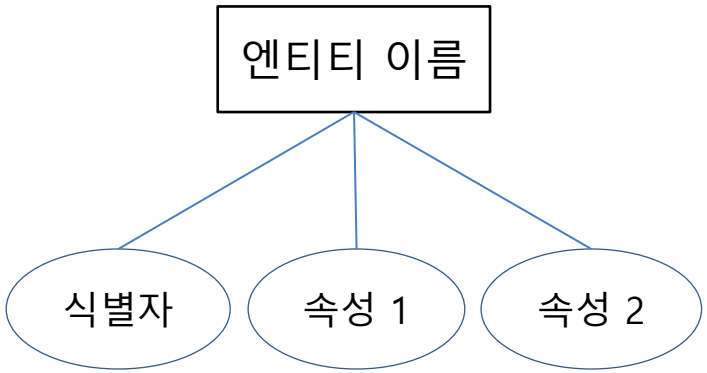
- **다대다(N:N) 관계**

한 학생은 여러 강좌를 수강할 수 있고, 한 강좌 역시 여러 학생이 들을 수 있다.

ER 모델

IE(Information Engineering Notation) 표기법

관계 대응 수를 새발 모양의 기호로 표현하여 새발 표기법(crow-feet)이라고도 부른다.



피터 첸 표기법

IE 표기법

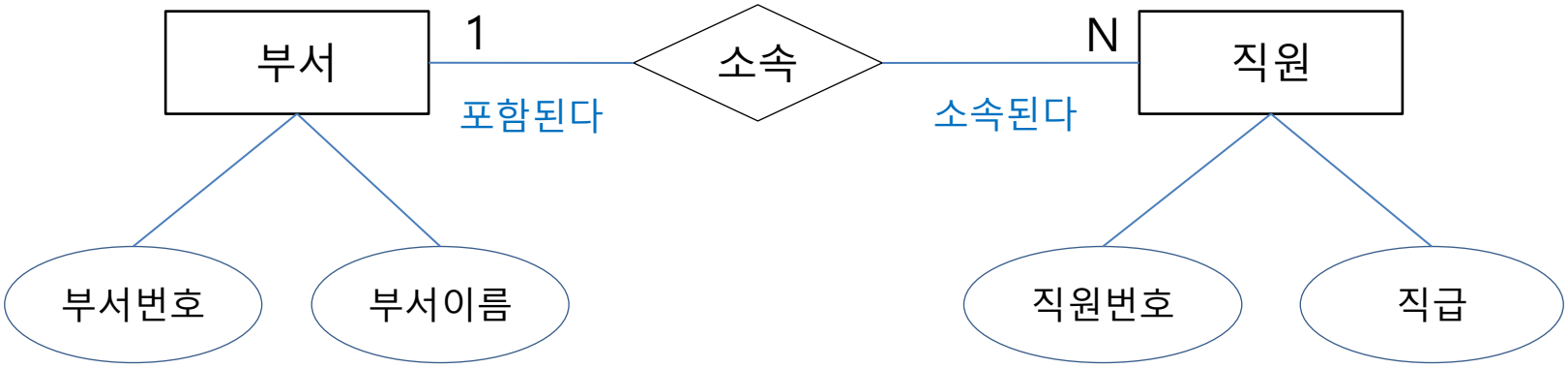
ER 모델

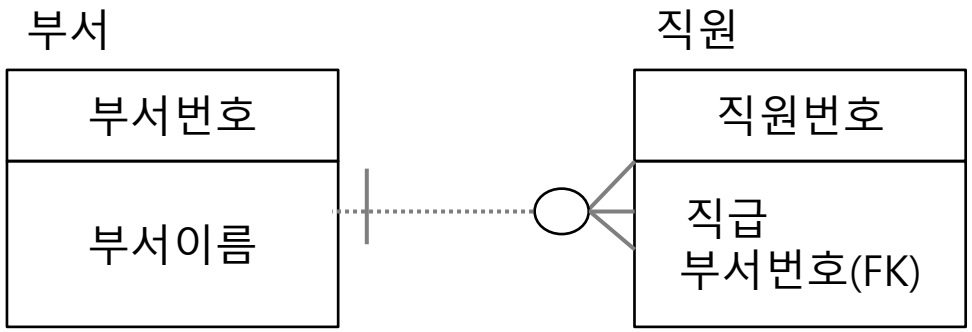
IE(Information Engineering Notation) 표기법

기호	전화번호
----------	- 비 식별자 관계: 강한 개체 타입 - 부모 개체의 키가 일반 속성으로 포함되는 관계
————	- 식별자 관계: 약한 개체 타입 - 부모 개체의 키가 주식별자로 포함되는 관계
	- 일대다(1:N)의 관계: N쪽에 새발을 표시
	- 0(선택 참여), 최소 참여가 0일 경우
	- 1(필수 참여), 최소 참여가 1일 경우

ER 모델

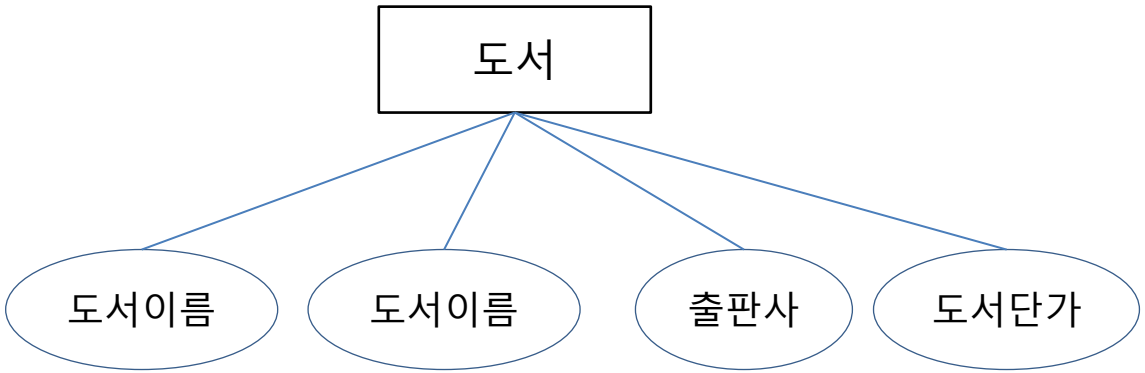
IE(Information Engineering Notation) 표기법





ER 모델

ER 모델을 **관계 데이터 모델**로 사상(Mapping)



ER 모델

사상(Mapping)

도서(<u>도서번호</u>, 도서이름, 출판사, 도서단가)

관계 데이터 모델

마당 서점 설계 실습

마당 서점의 논리적 모델링

- 도서 목록에는 도서번호, 도서이름, 출판사이름, 도서단가를 기록한다(개체)
- 출판사 목록에는 출판사이름, 담당자이름, 전화번호를 기록한다(개체)
- 고객 목록에는 고객번호, 고객이름, 주소, 전화번호를 기록한다(개체)
- 마당서점은 출판사에서 공급한 도서만 등록하여 관리한다(출판사와 도서의 관계는 1:N)
- 고객은 여러 번에 걸쳐 여러 권의 도서를 구입할 수 있다(고객과 도서의 관계 M:N)
- 마당서점은 고객이 도서를 구입한 날(주문일자)과 구매한 가격(주문금액)을 따로 저장한다(고객과 도서의 관계에 속성이 존재함)

데이터 베이스 모델링

마당 서점의 개체와 속성

개체	키속성(PK)	속성
출판사	출판사이름	담당자이름, 전화번호
도서	도서번호	도서이름, 도서단가, 출판사이름(FK)
고객	고객번호	고객이름, 주소, 전화번호
주문	주문번호	고객번호(FK), 도서번호(FK), 주문일자, 주문금액

마당서점 – 도서 테이블

bookid	bookname	publisher	price
1	축구의 역사	굿스포츠	7000
2	축구아는 여자	나무수	13000
3	축구의 이해	대한미디어	22000
4	골프바이블	대한미디어	35000
5	피겨교본	굿스포츠	8000
6	양궁의 실제	굿스포츠	6000
7	야구의 추억	이상미디어	20000
8	야구를 부탁해	이상미디어	13000
9	올림픽 이야기	삼성당	7500
10	Olympic Champions	Pearson	13000

마당서점 – 고객 테이블

custid	name	address	phone
1	박지성	영국 맨체스터	000-5000-0001
2	김연아	대한민국 서울	000-6000-0001
3	안산	대한민국 광주광역시	000-7000-0001
4	류현진	미국 토론토	NULL
5	손흥민	영국 토트넘	000-8000-0001

마당서점 – 주문 테이블

orderid	custid	bookid	saleprice	orderdate
1	1	1	6000	2018-07-01
2	1	3	21000	2018-07-03
3	2	5	8000	2018-07-03
4	3	6	6000	2018-07-04
5	4	7	20000	2018-07-05
6	1	2	12000	2018-07-07
7	4	8	13000	2018-07-07
8	3	10	12000	2018-07-08
9	2	10	7000	2018-07-09
10	3	8	13000	2018-07-10

도서 테이블 만들기

```
-- book 테이블 생성
CREATE TABLE book(
    bookid      NUMBER PRIMARY KEY,
    bookname    VARCHAR2(40) NOT NULL,
    publisher   VARCHAR2(40) NOT NULL,
    price       NUMBER NOT NULL
);
```

```
INSERT INTO book VALUES (1, '축구의 역사', '굿스포츠', 7000);
INSERT INTO book VALUES (2, '축구아는 여자', '나무수', 13000);
INSERT INTO book VALUES (3, '축구의 이해', '대한미디어', 22000);
INSERT INTO book VALUES (4, '골프 바이블', '대한미디어', 35000);
INSERT INTO book VALUES (5, '피겨 교본', '굿스포츠', 8000);
INSERT INTO book VALUES (6, '양궁의 실제', '굿스포츠', 6000);
INSERT INTO book VALUES (7, '야구의 추억', '이상미디어', 20000);
INSERT INTO book VALUES (8, '야구를 부탁해', '이상미디어', 13000);
INSERT INTO book VALUES (9, '올림픽 이야기', '삼성당', 7500);
INSERT INTO book VALUES (10, 'Olympic Champions', 'Pearson', 13000);

COMMIT;
```

고객 및 주문 테이블 만들기

```
-- 고객 테이블 생성
CREATE TABLE customer(
    custid      NUMBER PRIMARY KEY,
    name        VARCHAR2(40) NOT NULL,
    address     VARCHAR2(50),
    phone       VARCHAR2(20)
);

-- customer 자료 삽입
INSERT INTO customer VALUES (1, '박지성', '영국 맨체스타', '000-5000-0001');
INSERT INTO customer VALUES (2, '김연아', '대한민국 서울', '000-6000-0001');
INSERT INTO customer VALUES (3, '안산', '대한민국 광주광역시', '000-7000-0001');
INSERT INTO customer VALUES (4, '류현진', '미국 토론토', NULL);
INSERT INTO customer VALUES (5, '손흥민', '영국 토트넘', '000-8000-0001');

COMMIT;
```

고객 및 주문 테이블 만들기

```
-- 주문 테이블 생성
CREATE TABLE orders(
    orderid     NUMBER PRIMARY KEY,
    custid      NUMBER NOT NULL,
    bookid      NUMBER NOT NULL,
    saleprice   NUMBER NOT NULL,
    orderdate   VARCHAR2(20) NOT NULL,
    FOREIGN KEY(custid) REFERENCES customer(custid),   --외래키(custid)
    FOREIGN KEY(bookid) REFERENCES book(bookid)        --외래키(bookid)
);
```

고객 및 주문 테이블 만들기

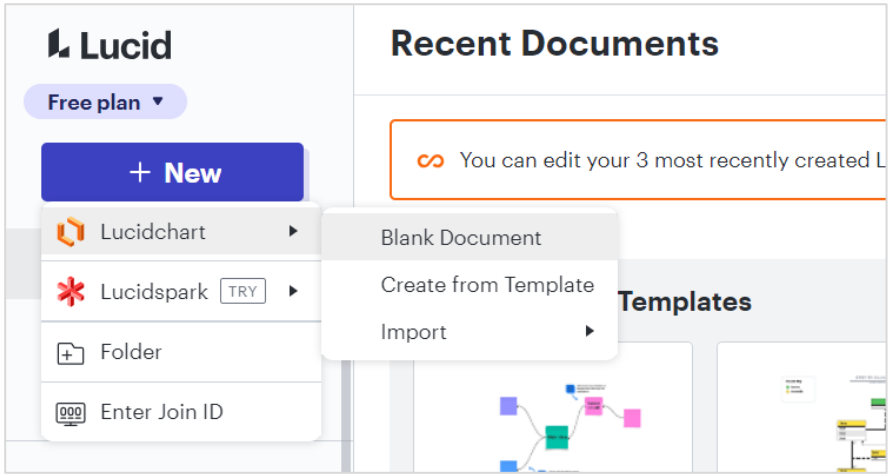
```
-- orders 자료 삽입
INSERT INTO orders VALUES (1, 1, 1, 6000, '2018-07-01');
INSERT INTO orders VALUES (2, 1, 3, 21000, '2018-07-03');
INSERT INTO orders VALUES (3, 2, 5, 8000, '2018-07-03');
INSERT INTO orders VALUES (4, 3, 6, 6000, '2018-07-04');
INSERT INTO orders VALUES (5, 4, 7, 20000, '2018-07-05');
INSERT INTO orders VALUES (6, 1, 2, 12000, '2018-07-07');
INSERT INTO orders VALUES (7, 4, 8, 13000, '2018-07-07');
INSERT INTO orders VALUES (8, 3, 10, 12000, '2018-07-08');
INSERT INTO orders VALUES (9, 2, 10, 7000, '2018-07-09');
INSERT INTO orders VALUES (10, 3, 8, 13000, '2018-07-10');

COMMIT;
```

Lucid Chart

➢ 루시드 차트

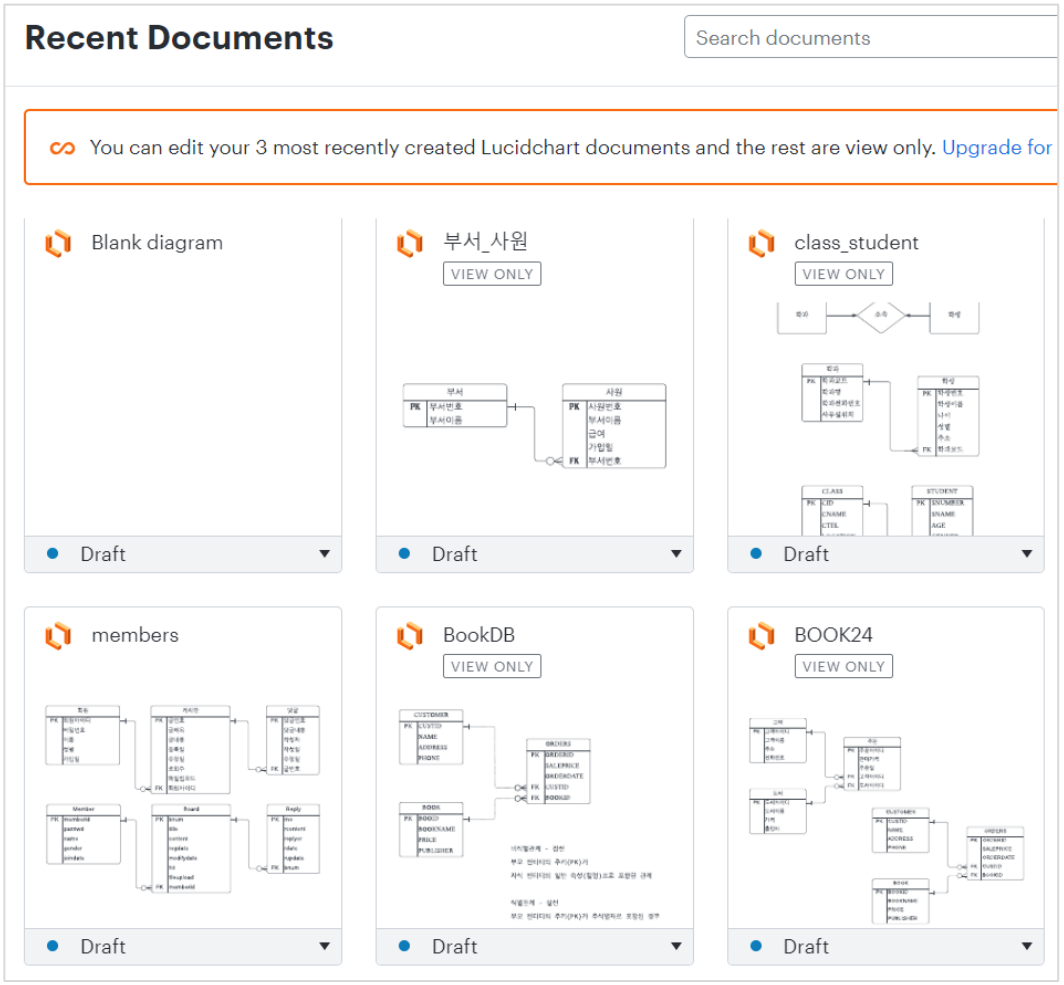
사용자가 차트 및 다이어그램을 그릴 수 있는 웹 기반 다이어그램 작성 응용 프로그램이다. 수정 및 공유에 대해 시각적으로 협업하고 프로세스, 시스템 및 조직 구조를 개선할 수 있다.

- ▶ 회원 가입 -> 로그인
- ▶ Recent Documents > +New > Lucidchart > Blank Document



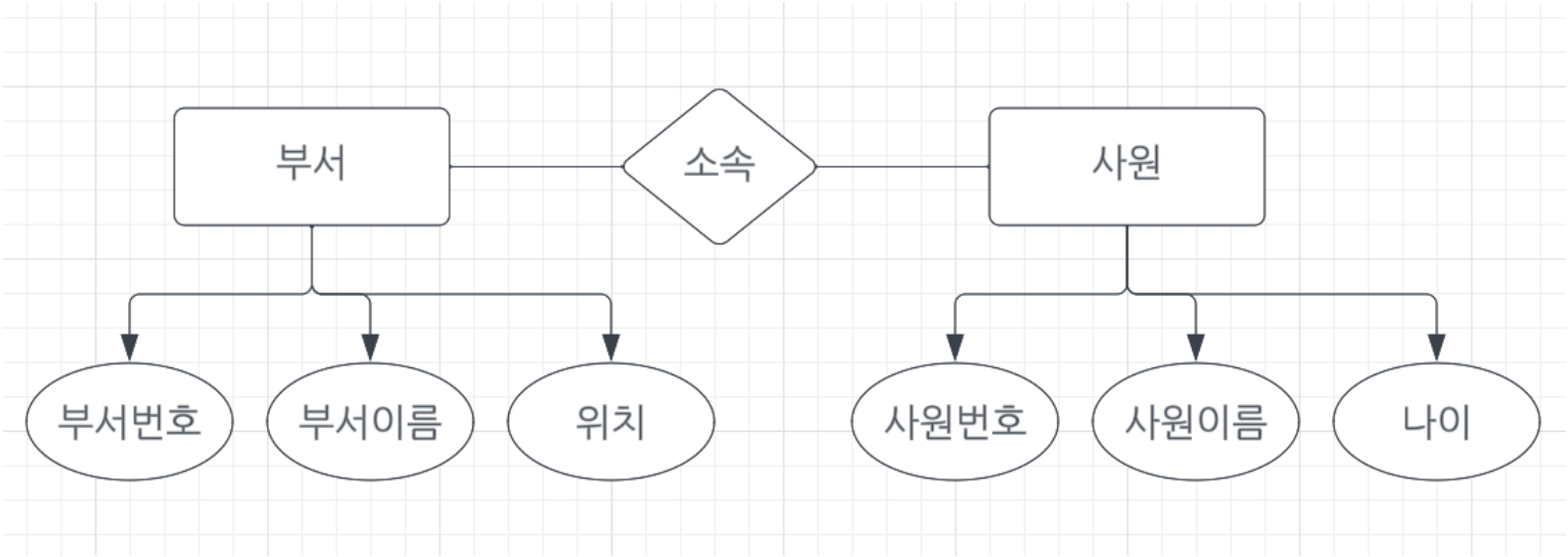
Lucid Chart

- 루시드 차트



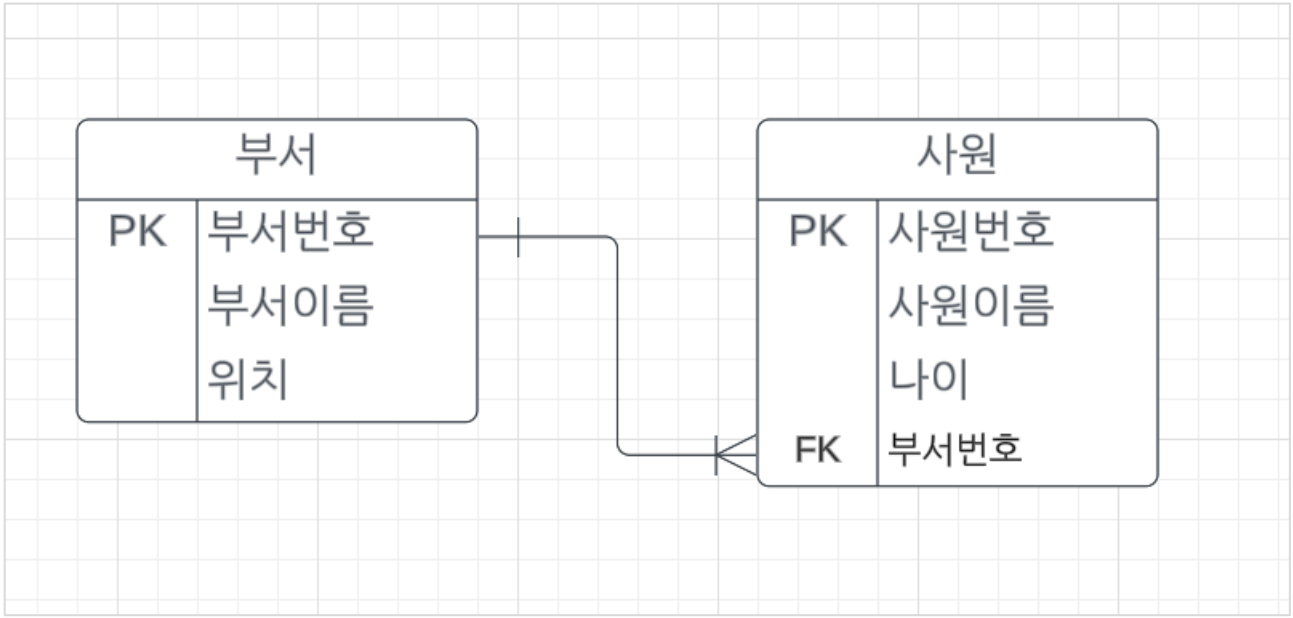
ER 다이어그램

- 부서와 사원 개체 관계도



Lucid Chart

- 부서와 사원 개체 관계도



관계(Releation)

외래키(FK : Foreign Key)

- 특정 테이블에 포함되어 있으면서 다른 테이블의 기본키로 지정된 키

기본키

부서 코드	부서 이름	위치
10	전산팀	서울
20	총무팀	인천

기본키

외래키

사원 번호	사원 이름	나이	부서코드
501	이강	23	10
502	김산	24	20
503	남한강	31	10
504	북한강	35	20

관계(Releation) – 외래키

부서와 직원 테이블 생성

- 사원(employee) 테이블에 Foreign Key 설정

```
-- 부서 테이블
CREATE TABLE department(
    deptid      NUMBER,
    deptname    VARCHAR2(20) NOT NULL,
    location    VARCHAR2(20) NOT NULL,
    PRIMARY KEY(deptid)
);

CREATE TABLE employee(
    empid       NUMBER,
    empname     VARCHAR2(20) NOT NULL,
    age         NUMBER,
    deptid      NUMBER,
    CONSTRAINT EMP_FK FOREIGN KEY(deptid) REFERENCES department(deptid)
);
```

관계(Releation) – 외래키 제약

부서와 직원 자료 추가

```
-- 부서 자료 추가
INSERT INTO department VALUES (10, '전산팀', '서울');
INSERT INTO department VALUES (20, '총무팀', '인천');

-- 사원 자료 추가
INSERT INTO employee VALUES (101, '이강', 27, 10);
INSERT INTO employee VALUES (102, '김산', 28, 20);
INSERT INTO employee VALUES (103, '정들', 35, 30); -- 부서코드 없음
```

```
명령의 24 행에서 시작하는 중 오류 발생 -
INSERT INTO employee VALUES (103, '정들', 35, 30)
오류 보고 -
ORA-02291: 무결성 제약조건(SYSTEM.SYS_C008344)이 위배되었습니다- 부모 키가 없습니다
```

관계(Releation) – 외래키 제약

부서 자료 삭제

```
-- 부서 삭제
DELETE FROM department WHERE deptid = 20; -- employee 테이블에서 참조하고 있어 오류.
```

```
명령의 27 행에서 시작하는 중 오류 발생 -
DELETE FROM department WHERE deptid = 20
오류 보고 -
ORA-02292: 무결성 제약조건(SYSTEM.SYS_C008344)이 위배되었습니다- 자식 레코드가 발견되었습니다
```

외래키 제약 조건 삭제

```
ALTER TABLE employee DROP CONSTRAINT EMP_FK;
```

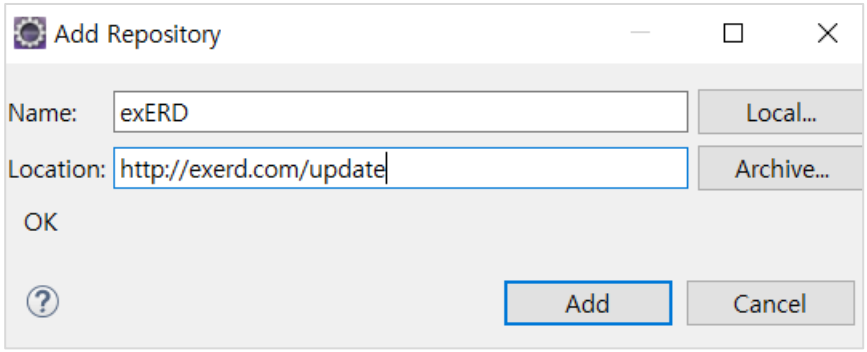
테이블 삭제

```
-- 테이블 삭제 - CONSTRAINT가 설정되어 있는 경우
DROP TABLE department CASCADE CONSTRAINTS;
```

데이터베이스 모델링

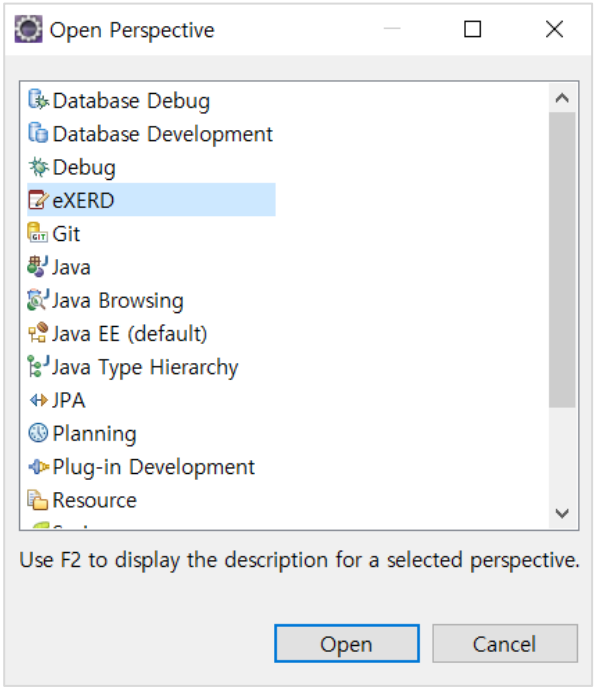
exERD – 데이터베이스 모델링 소프트웨어

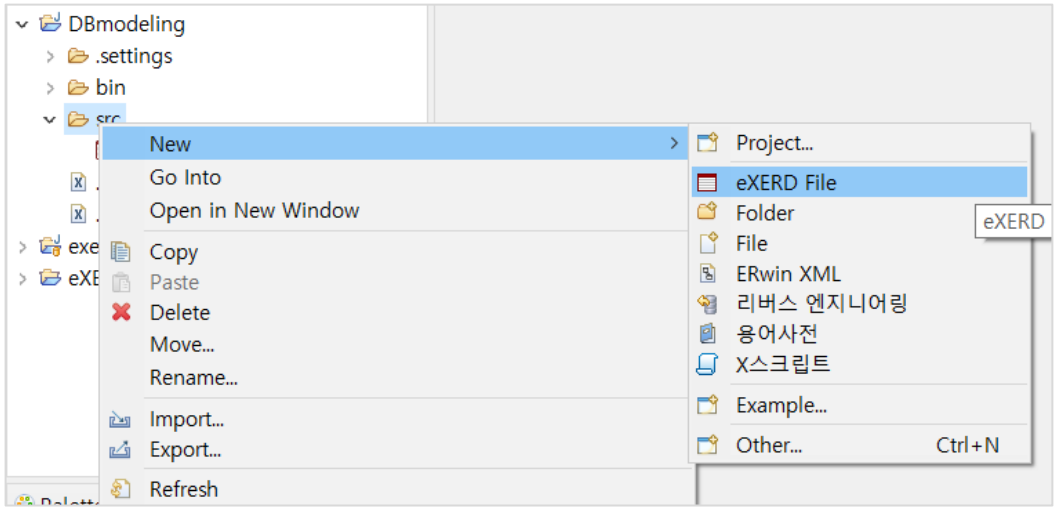
이클립스 – help – install NewSoftware – add 버튼 클릭



데이터베이스 모델링

exERD – 새 파일 만들기

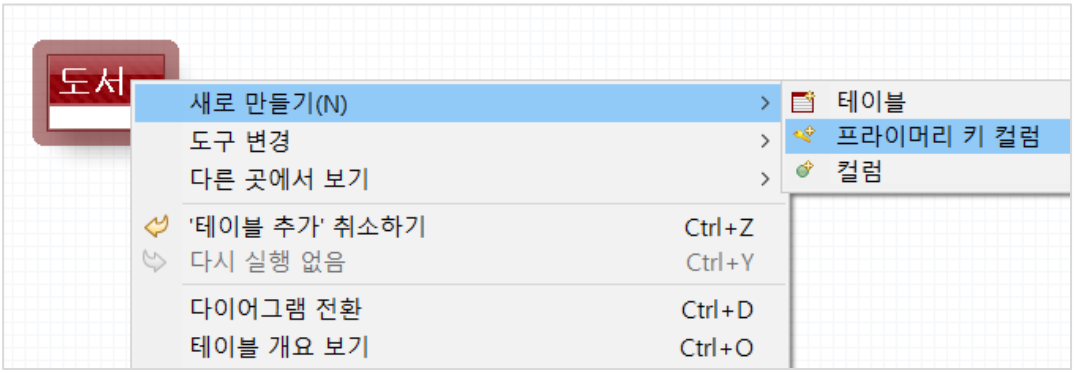




데이터베이스 모델링

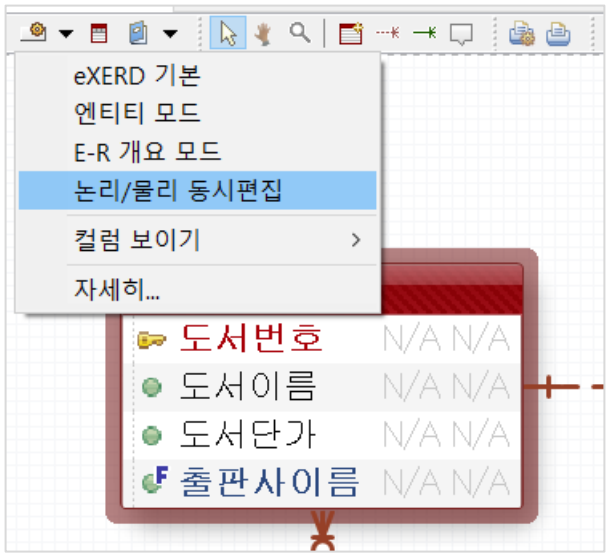
exERD – 새 파일 만들기

기본키(primary key)

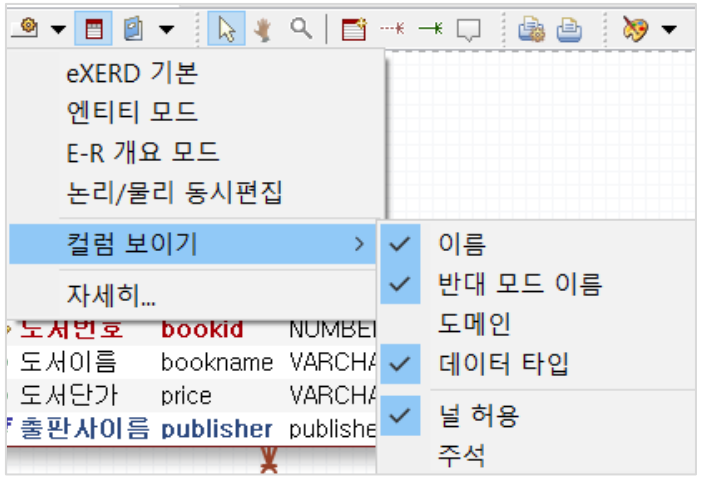


데이터베이스 모델링

논리/물리 동시편집

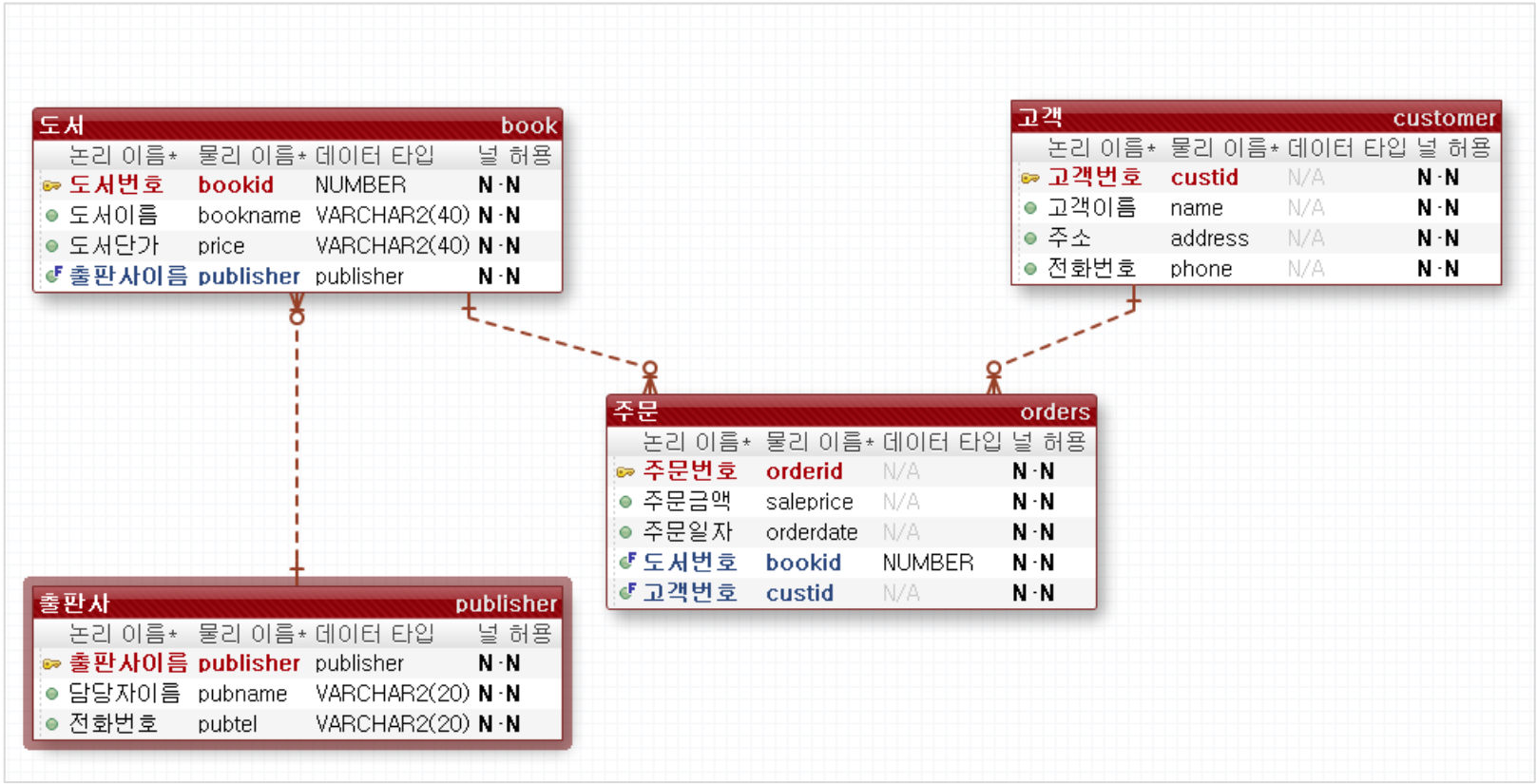


칼럼 보이기



데이터베이스 모델링

개체 관계도(ERD-Entity Releation Diagram)



데이터 베이스 모델링

학과

학과명	전화번호	사무실 위치
소프트웨어학과	02-1234-1234	B동 3층
전기전자공학과	02-1234-4567	B동 4층
화학공학과	02-1234-5678	B동 5층

학생

학번	학생 이름	나이	성별	주소
20211234	이강	22	여	서울시 구로구
20211235	박대양	25	남	서울시 성동구
20211236	한비야	23	여	서울시 강남구
20211237	정산들	27	남	경기도 수원시

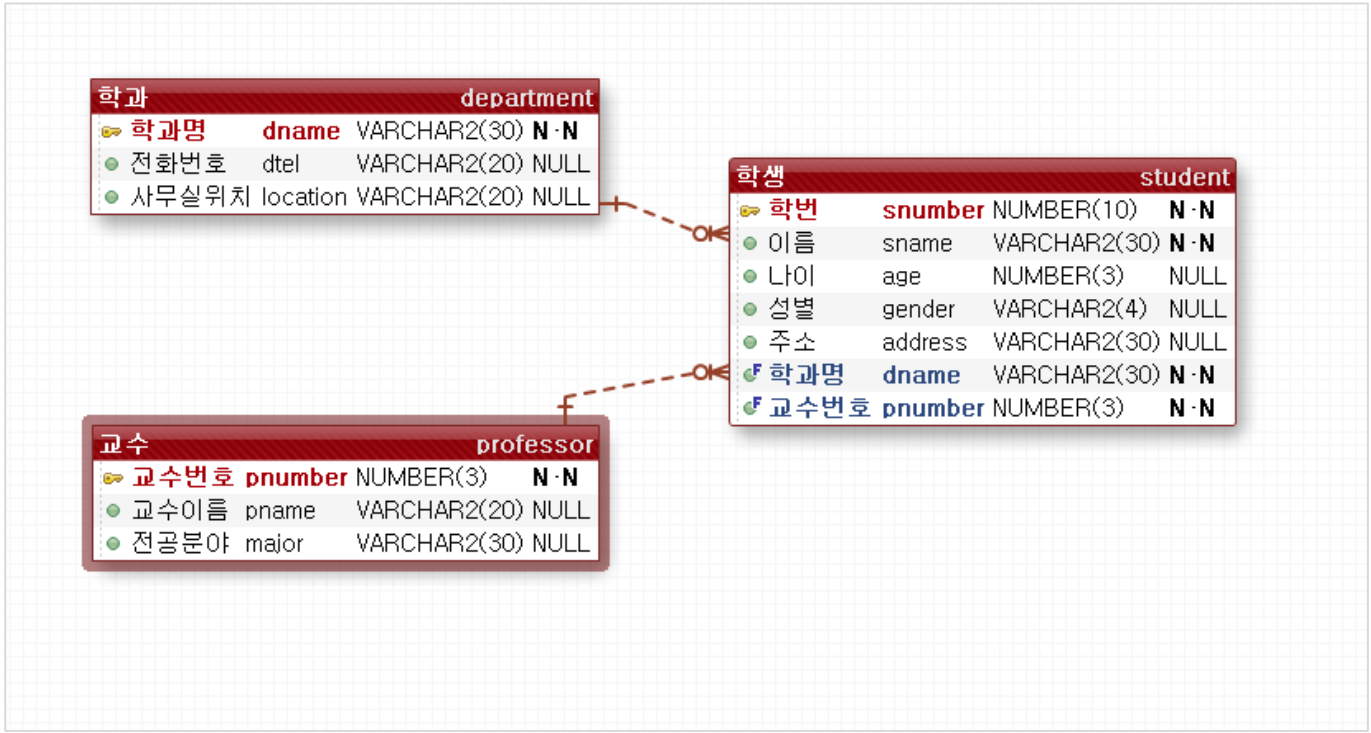
데이터 베이스 모델링

교수

학번	학생 이름	나이	성별	주소
20211234	이강	22	여	서울시 구로구
20211235	박대양	25	남	서울시 성동구
20211236	한비야	23	여	서울시 강남구
20211237	정산들	27	남	경기도 수원시

데이터베이스 모델링

개체 관계도(ERD-Entity Releation Diagram)



데이터베이스 구축하기

대학교 업무 관계

```
-- 학과 테이블
CREATE TABLE department(
    dname VARCHAR2(30),
    dtel VARCHAR2(20),
    location VARCHAR2(20),
    PRIMARY KEY(dname)
);
```

```
-- 교수 테이블
CREATE TABLE professor(
    pnumber NUMBER(3),
    pname VARCHAR2(20),
    major VARCHAR2(30),
    PRIMARY KEY(pnumber)
);
```

데이터베이스 구축하기

```
-- 학생 테이블
CREATE TABLE student(
    snumber NUMBER(10),
    sname VARCHAR2(20),
    age NUMBER(3),
    gender VARCHAR2(4),
    address VARCHAR2(30),
    dname VARCHAR2(30),
    pnumber NUMBER(3),
    PRIMARY KEY(snumber),
    CONSTRAINT FK_dept_std FOREIGN KEY(dname)
    REFERENCES department(dname),
    CONSTRAINT FK_prof_std FOREIGN KEY(pnumber)
    REFERENCES professor(pnumber)
);
```

데이터베이스 구축하기

```
-- 학과 추가 --
INSERT INTO department VALUES ('소프트웨어학과', '02-1234-1234', 'B동 3층');
INSERT INTO department VALUES ('화학공학과', '02-1234-4567', 'B동 4층');
INSERT INTO department VALUES ('전기전자공학과', '02-1234-5678', 'B동 5층');
```

```
-- 교수 추가
INSERT INTO professor VALUES (301, '박은종', 'JAVA 프로그래밍');
INSERT INTO professor VALUES (402, '송미영', 'JSP 웹프로그래밍');
INSERT INTO professor VALUES (501, '오용철', '데이터베이스');

-- 학생 추가 --
INSERT INTO student VALUES (20211234, '이강', 22, '여', '서울시 구로구', '소프트웨어학과', 301);
INSERT INTO student VALUES (20211235, '박대양', 25, '남', '서울시 성동구', '전기전자공학과', 501);
INSERT INTO student VALUES (20211236, '한비야', 23, '여', '서울시 강남구', '소프트웨어학과', 402);
INSERT INTO student VALUES (20211237, '정산들', 27, '남', '경기도 수원시', '화학공학과', 501);
```