



4주차(03/31)

개념적 데이터 모델링(개념적 설계)

- 데이터 모델링 3단계 :
 1. 개념적 설계 : ERD
 2. 논리적 설계 : 스키마
 3. 물리적 설계
 - 개념적 설계 - ERD(Entity Relationship Diagram : 개체 관계 다이어그램)
 - 1976년 P.Chen 제안
 - 개체(엔티티 : entity) ⇒ 테이블
 - 업무와 관여하는 어떤 것(thing)
 - 실세계에 존재하는 유, 무형의 객체
 - 정규 엔티티(strong entity / regular entity) ⇒ 한 겹 사각형
 - 자신의 키 속성을 사용하여 고유하게 엔티티들을 식별할 수 있는 엔티티
 - 독립적으로 존재하면서 고유하게 식별(사람, 사물, 사건, 장소)
 - 약 엔티티(weak entity) : 약한 엔티티 ⇒ 두 겹 사각형(특수 케이스)
 - 자신의 키 속성을 갖기에 충분한 속성을 갖지 못한 엔티티
 - 자체적으로 키를 보유하지 못하는 엔티티
- ⇒ 부분키 사용(기본키 == 소유엔티티의 기본키 + 부분키)



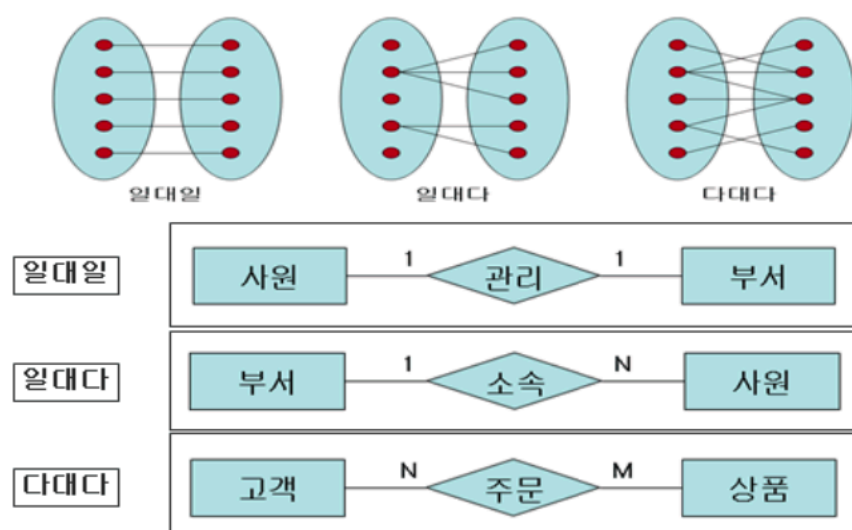
부분키(Partial key)

- 키와 비슷하지만 완벽하게 키라고 할 수 없음
- 약 엔티티에서 사용되는 키(점선 밑줄)

- 식별 관계로 연결(실선 연결) : 주인의 기본키와 연결
 - 다른 엔티티에 종속 : 소유 엔티티 ⇒ 소유 엔티티 타입이 있어야 함
 - 의존 종속성 : 한 개체의 존재가 다른 개체에 의해 영향 받음
 - 존재 종속 : 해당 엔티티가 없다면 존재하지 않는 종속성
- 속성(애트리뷰트 : attribute) ⇒ 필드
 - 어떤 것이 가지는 속성(attribute)
 - 엔티티 또는 관계가 갖는 성질이나 특성
 - 단순 속성 : 더 이상 다른 속성으로 나눌 수 없는 속성
 - 키 속성(주 식별자) : 엔티티들을 식별할 수 있는 유일한 제약 조건을 갖는 속성(밑줄 표시)
 - 복합 속성 : 두 개 이상의 속성으로 이루어진 속성
 - ex) 주소(시, 구, 동), 이름(성, 이름)
 - 방법 1 : 한 칸에 그대로 씀
 - 방법 2 : 칸칸이 씀
 - 다중 값(다치) 속성
 - 속성 하나에 여러 개의 값을 가질 수 있는 속성(별도의 엔티티로 분리)
 - 식별 관계로 연결
 - 콤마로 분리 가능
 - 유도 속성
 - 실제 값이 저장된 것이 아니라 저장된 값으로부터 계산해서 얻은 값을 사용하는 속성(점선 표시)
 - 대부분 논리적 설계에서는 제거

기호	의미
	개체 타입
	약한 개체 타입
	관계 타입
	식별 관계 타입
	전체 참여 개체 타입
	속성
	식별자 속성
	부분키 속성
	다중 값 속성
	복합 속성
	유도 속성

- 관계(relationship) : 업무가 관여하는 어떤 것 간의 관계(relationships) ⇒ 차수
 - 관계 차수 : 관계와 이어진 엔티티 수 $N \Rightarrow N$ 진 관계
 - 관계 카디널리티
 - 일대일 관계(1 : 1)
 - 일대다 관계(1 : N / N : 1) ⇒ N 쪽으로 외교관이 감
 - 다대다 관계(N : M)



- 관계 종속성

- 필수(Mandatory) ⇒ 전체 참여
- 선택(Optional) ⇒ 부분 참여



스키마

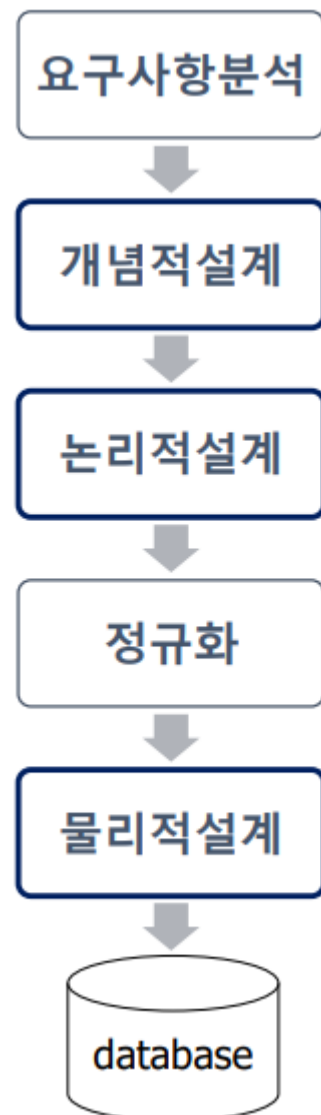
- 데이터베이스를 구성하는 데이터구조와 제약조건에 대한 명세를 구체적으로 기술한 것
 - 개체(Entity)
 - 속성(Attribute)
 - 관계(Relationship)
 - 제약조건

데이터 모델의 구성요소($D = \langle S, O, C \rangle$)

- 구조(Structure)
- 연산(Operation)
- 제약조건(Constraint)

데이터베이스 설계 과정

1. 요구사항 수집과 분석 : 인터뷰 설문조사 등
2. 개념적 설계 : ER 모델(P.Chen 1976년 제안)
 - 엔티티 타입, 관계 타입, 애트리뷰트 식별
 - 도메인, 후보키, 기본키 설정
3. 논리적 설계 : 관계스키마로 사상
 - 관계성 : 일대일(1 : 1), 일대다(1 : N), 다대다(M : N) 관계
 - 선택성 : 전체참여(필수), 부분참여(선택)
4. 정규화
5. 물리적 설계
6. 구현



[실습]

```
create schema W04 default character set utf8mb4;  
use W04;
```

```
-- 테이블 목록 조회  
show tables;
```

```
-- 테이블 구조 확인  
-- describe tableName; 도 가능
```

```

desc dept;
desc emp;
desc 학생_수강;
desc 학생;

-- 테이블 삭제
drop table if exists dept;
drop table if exists emp;
drop table if exists 학생_수강;
drop table if exists 학생;

```

```

-- ***** dept, emp table *****
-- dept(부서코드, 부서명, 위치)
create table dept (
    부서코드 char(2),
    부서명 varchar(20),
    위치 varchar(45)
);

alter table dept
    add constraint pk_dept
        primary key(부서코드);

insert into dept values('AA','총무부','서울');
insert into dept values('BB','영업부','대전');
insert into dept values('CC','기획부','서울');

select * from dept;

-- emp(사원번호, 사원명, 부서코드)
create table emp (
    사원번호 int,
    사원명 varchar(10),
    부서코드 char(2)
);

alter table emp
    add constraint pk_emp

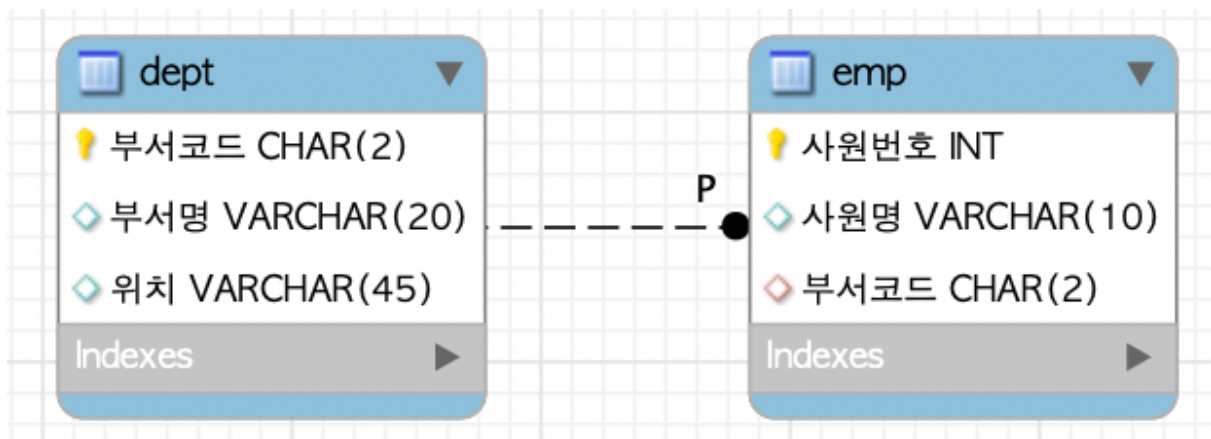
```

primary key(사원번호);

```
alter table emp
add constraint fk_dept_TO_emp
foreign key(부서코드)
references dept(부서코드);
```

```
insert into emp values(100, '신재영', 'BB');
insert into emp values(101, '오주원', 'AA');
insert into emp values(102, '이대은', 'BB');
```

```
select * from emp;
```



비식별관계

```
-- ***** 학생, 학생_수강 table *****
-- 학생(학번, 이름, 전공)
create table 학생 (
  학번 int,
  이름 varchar(10),
  전공 varchar(20)
);

alter table 학생
add constraint pk_학생
primary key(학번);

insert into 학생 values(213068, '홍길동', '컴퓨터공학');
```

```

insert into 학생 values(20230880, '유승민', '컴퓨터소프트웨어공학과');

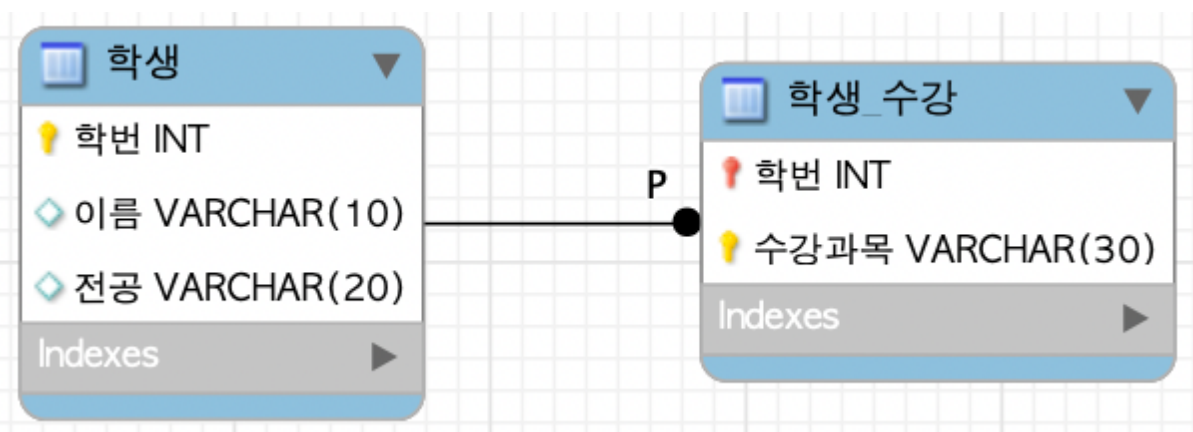
select * from 학생;

-- 학생_수강(학번, 수강과목)
create table 학생_수강 (
    학번 int,
    수강과목 varchar(30),
    primary key(학번, 수강과목), # 복합키
    key fk_학생_TO_학생_수강_idx (학번), # 빠른 검색을 위한 인덱스 설정(index fk_학
    constraint fk_학생_TO_학생_수강
        foreign key(학번)
        references 학생(학번)
);

insert into 학생_수강 values(213068, 'C언어');
insert into 학생_수강 values(213068, '자료구조');

select * from 학생_수강;

```



식별관계