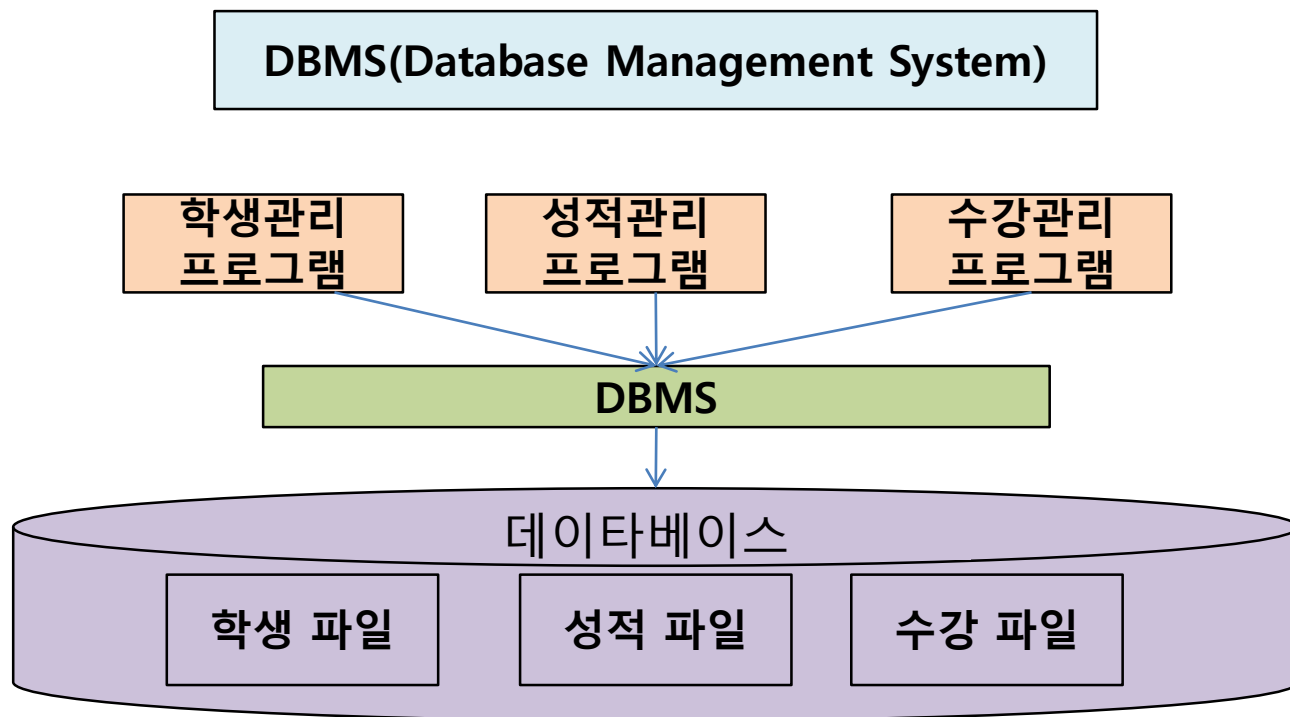


Integrated Data(통합)
Stored Data(저장)
Shared Data(공용)
Operational Data(운영)

한 조직에서 다수의 사용자를 위해 통합, 저장, 공용, 운영되는 데이터



데이터베이스의 생성과 관리를 담당하는 소프트웨어

- 정의기능 (DDL)
- 조작기능 (DML)
- 제어기능 (DCL)

데이터 베이스

데이터 베이스란?

어느 한 조직에서 업무처리를 위해 다수의 응용 시스템 혹은 다수의 사용자들이 공용으로 사용하기 위해 통합·저장된 운영데이터의 집합을 말한다.

| | |
|------------------------------|--|
| 통합 데이터 (Integrated Data) | 하나의 주제에 따라 중복을 최소화한 데이터의 집합 |
| 저장된 데이터 (Stored Data) | 사용자나 응용시스템이 필요시 언제든지 이용할 수 있도록 저장된 데이터의 집합 |
| 공용 데이터 (Shared Data) | 여러 사용자와 다수의 응용 시스템이 공유할 수 있도록 만든 데이터의 집합 |
| 운영 데이터 (Operational Data) | 중복을 최소화하고 여러 사람이 공유함에 있어 문제가 발생하지 않도록 관리를 필요로 하는 데이터로 이용가치가 있는 데이터의 집합 |

데이터 베이스의 특징

| | |
|--|--|
| 계속적인 변화(진화) Continuous Evolution | 항상 최신 정보를 유지 할 수 있도록 삽입, 삭제, 갱신이 이루어짐 |
| 동시공유 (Concurrent Sharing) | 여러 사용자가 동시에 접근하여 이용 |
| 실시간 접근성 (Real time Accessibility) | 질의(query)에 대해 실시간 처리 및 응답 |
| 내용에 의한 참조 (contents Reference) | 데이터의 물리적 주소나 위치에 의지하지 않고 사용자가 요구하는 데이터 내용으로 검색(이용) |
| 데이터의 중복의 최소화 (Redundancy Minimize) | 동일 데이터의 중복성을 최소화해야 함 |

데이터베이스 설계

요구조건 분석



설계

개념적 설계
논리적 설계
물리적 설계



구현



운영 및 개선(유지보수)

데이터베이스 설계
[database architecture](#)

데이터베이스 설계자
[database architect](#)

데이터베이스 관리자
[database administrator](#)

관계 데이터 모델의 용어

| | |
|---|-------------------------------|
| 릴레이션(Relation) | 자료저장의 형태가 2차원 구조의 테이블(표)로 표현 |
| 속성(Attribute) | 릴레이션을 구성하는 각 열 (Column =항목) |
| 튜플(Tuple) | 릴레이션의 한 행을 구성하는 속성들의 집합 |
| 도메인(Domain) | 하나의 속성이 가질 수 있는 값들의 범위 |
| 릴레이션 스키마 (Relation Schema) | 릴레이션의 이름과 속성이름의 집합 (릴레이션의 구조) |
| 릴레이션 인스턴스 (Relation Instance) | 릴레이션에서 어느 시점까지 입력된 튜플들의 집합 |
| 차수(Degree) | 릴레이션을 구성하는 속성(항목)의 수 |
| 카디널리티(Cardinality) | 릴레이션에 입력된 튜플(행)의 수 |

릴레이션

기본키 (primary key)

속성 (Attribute)

스키마

| 학생 | | | |
|------|-----|----|-----|
| 학번 | 이름 | 학년 | 학과 |
| 0001 | 이순신 | 1 | 정보과 |
| 0002 | 홍길동 | 2 | 무역과 |
| 0003 | 이영희 | 3 | 경영과 |
| 0004 | 김철수 | 2 | 정보과 |

튜플 (Tuple)

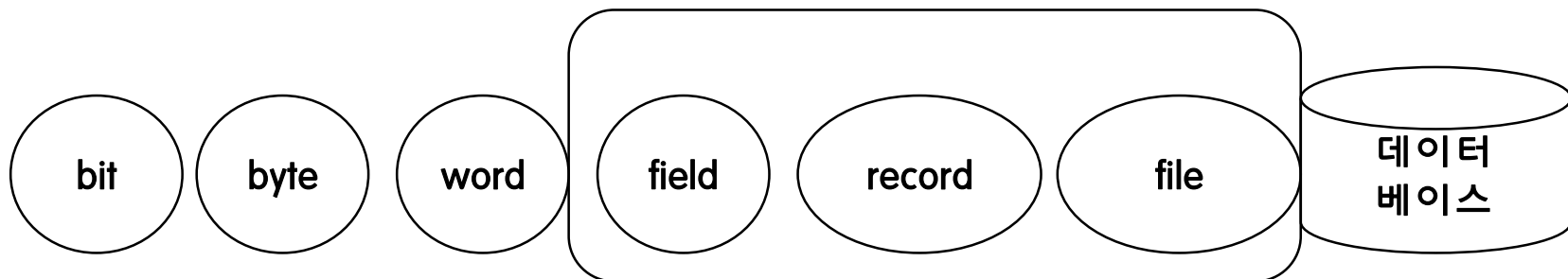
-차수 (degree)

-카디널리티 (cardinality)

도메인 (Domain) : 한 속성에 나타날 수 있는 값의 범위

릴레이션과 키
무결성과 무결성 제약 조건

정보표현의 단위



| 학번 | 이름 | 전화 | 주소 |
|------|----|----|----|
| 0801 | | | |
| | | | |
| | | | |

릴레이션 (표, 테이블 구조)

기본키(primary key)

유일하게 식별할 수
있는 값

속성 (Attribute)

스키마

학생

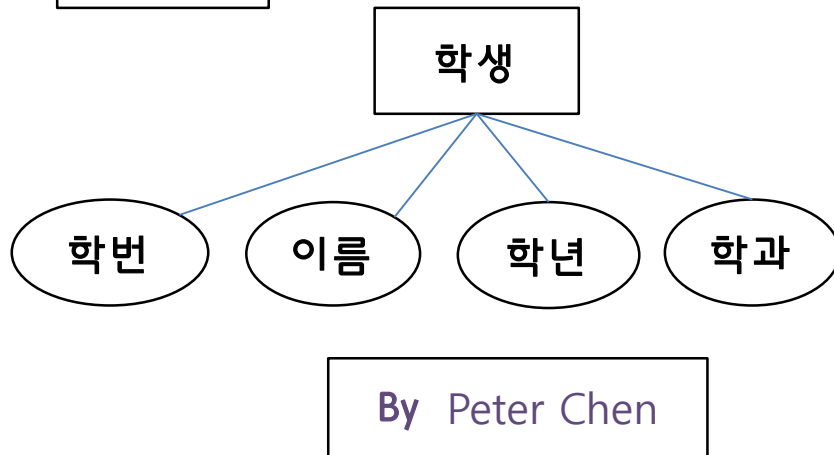
튜플 (Tuple)

| 학번 | 이름 | 학년 | 학과 |
|------|-----|----|-----|
| 0001 | 이순신 | 1 | 정보과 |
| 0002 | 홍길동 | 2 | 무역과 |
| 0003 | 이영희 | 3 | 경영과 |
| 0004 | 김철수 | 2 | 정보과 |

- 차수 (degree) : 4 : 속성의 개수
- 카디날리티 (cardinality) : 4 : 튜플의 개수

도메인 (Domain) : 한 속성에 나타날 수 있는 값의 범위

ERD



| 학번 | 이름 | 학년 | 학과 |
|------|-----|----|-----|
| 0001 | 이순신 | 1 | 정보과 |
| 0002 | 홍길동 | 2 | 무역과 |
| 0003 | 이영희 | 3 | 경영과 |
| 0004 | 김철수 | 2 | 정보과 |

[학생]

학번

이름
학년
학과

By James Martin

키(key)

- ❖ 튜플을 유일하게 식별할 수 있는 단일 속성 또는 속성그룹
- ❖ 식별자(Identifier)라고도 함

키의 특징

- ❖ 유일성
- ❖ 최소성

키의 종류

- ❖ 후보키 (Candidate key)
- ❖ 기본키 (Primary key)
- ❖ 대체키 (Alternate key)
- ❖ 외래키 (Foreign key)

학생

| 학번 | 이름 | 주민번호 | 학년 | 학과 코드 |
|------|-----|------|----|-------|
| 0001 | 이순신 | 8001 | 1 | A |
| 0002 | 홍길동 | 8512 | 2 | B |
| 0003 | 이영희 | 8910 | 3 | C |
| 0004 | 김철수 | 9001 | 2 | A |

학과

| 학과 코드 | 학과 |
|-------|----|
| 0001 | A |
| 0002 | B |
| 0003 | C |
| 0004 | D |

무결성(integrity)

권한이 없는 사용자가 데이터베이스를 접근하여 검색하거나 갱신하지 못하도록 데이터베이스를 보호

권한을 가진 사용자들로부터 데이터베이스의 정확성을 지키는 것

장점]

1)스키마를 정의할 때 일관성 조건을 오직 한 번 만 명시할 수 있다.

2) DBMS가 자동적으로 일관성 조건을 검사하므로 응용프로그램들은 일관성 조건을 검사할 필요가 없다는 것이다.

무결성의 종류





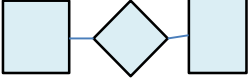
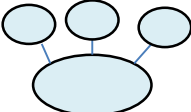
1. NULL무결성: 릴레이션의 속성값이 null이 될 수 없도록 하는 규정
2. 고유 무결성 : 각 튜플들이 갖는 속성값들이 서로 달라야 한다는 규정
3. 도메인 무결성: 속성의 값이 그 속성이 정의된 도메인에 속한 값이어야 한다는 규정
4. 키 무결성: 하나의 릴레이션에는 적어도 하나의 키가 존재해야 한다는 규정
5. 관계무결성 : 한 릴레이션의 튜플의 삽입여부 또는 다른 릴레이션의 튜플과의 관계에 대한 적절성 여부를 지정한 규정
6. 참조무결성: 외래키값이 Null이거나 참조 릴레이션의 기본키 값과 동일해야 한다는 규정
7. 개체무결성: 기본 릴레이션의 기본 키를 구성하는 어떤 속성도 Null일 수 없다는 규정

모델링
〈설계〉

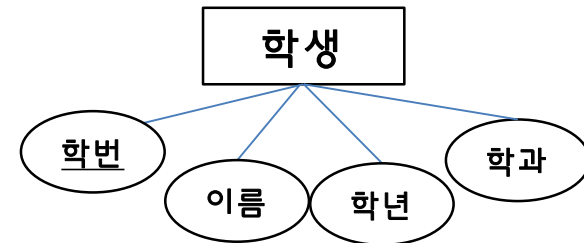
모델링이란?
ERD의 이해

개념적 모델링
논리적 모델링
물리적 모델링

ERD <Entity Relationship Diagram>

| 기호 | 의미 |
|---|-------------------|
|  | 개체 (Entity) |
|  | 속성 (Attribute) |
|  | 관계 (Relationship) |
|  | 키속성, 기본키속성 |
|  | 개체와 개체 간의 관계 구조 |
|  | 복합 속성 |

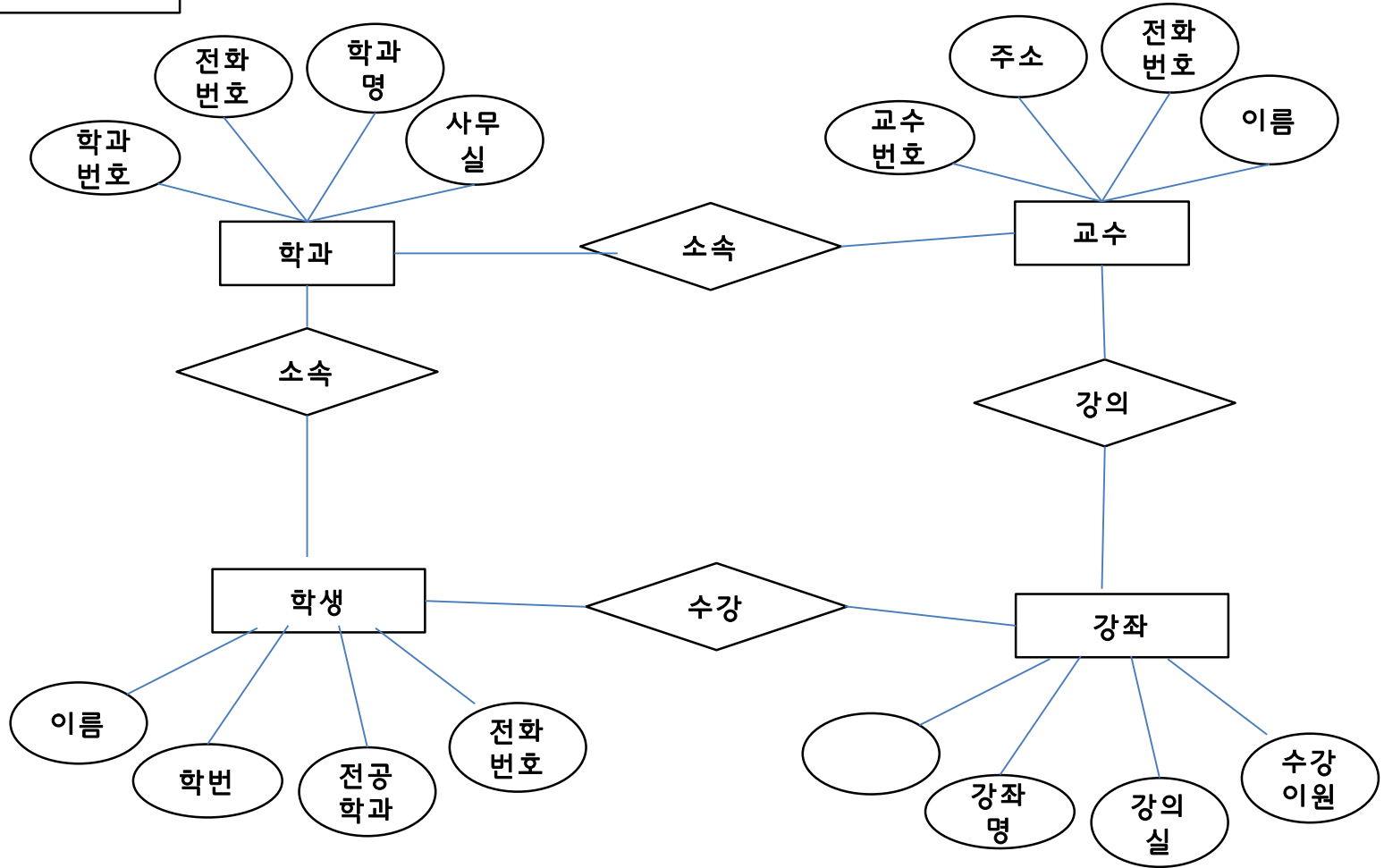
| 학번 | 이름 | 학년 | 학과 |
|------|----|----|----|
| 0901 | 홍가 | 1 | A |
| 0902 | 이가 | 1 | B |
| 1001 | 김가 | 2 | C |



By peter chen

릴레이션과 테이블
테이블과 개체(entity)의 차이

ERD



**개념적 설계
(개념적 모델링)**

ERD –피터첸

현실세계에 대한 인식을 추상적 개념으로 표현하는 과정

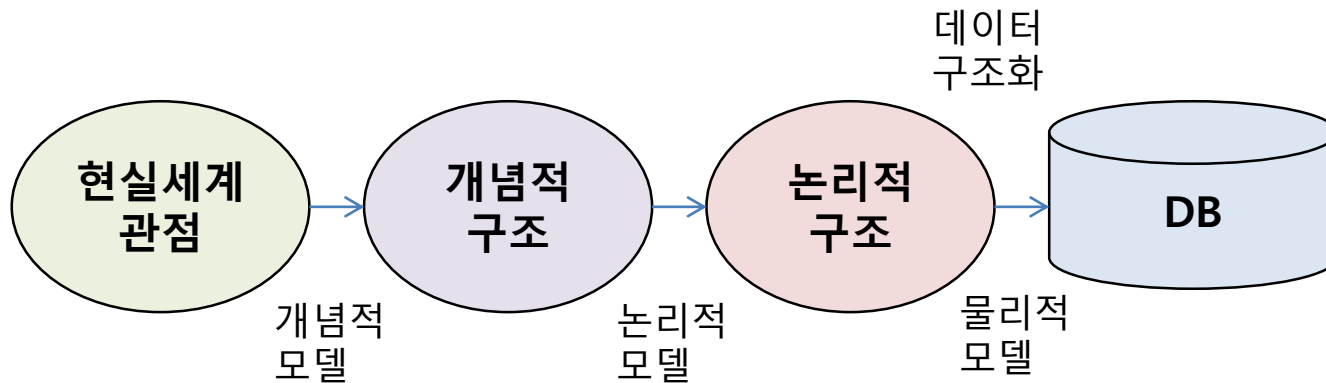
**논리적 설계
(논리적 모델링)**

현실세계에서 발생하는 데이터 형태를 컴퓨터가 이해하고 처리할 수 있는 논리적 데이터 구조로 변환시키는 과정

**물리적 설계
(물리적 모델링)**

데이터베이스 파일에 대한 특정한 저장구조와 접근 경로를 결정하는 과정

데이터모델링 과정



개념적 설계

현실세계에 대한 인식을 추상적 개념으로 표현하는 과정

논리적 설계

현실세계에서 발생하는 데이터 형태를 컴퓨터가 이해하고 처리할 수 있는 논리적 데이터 구조로 변환시키는 과정

물리적 설계

데이터베이스 파일에 대한 특정한 저장구조와 접근 경로를 결정하는 과정

ERD 구성요소

1. 개체 (Entity)

사람이 생각하는 개념이나 정보 단위와 같은 현실세계의 대상체로
실 세계에 존재하는 유.무형의 정보의 대상이며 서로 구별이 되는 하나 하나의 대상을 말한다

개체는 하나 이상의 속성으로 구성된다.

2. 속성 (Attribute)

데이터베이스를 구축하는 가장 작은 논리적 단위로 개체의 특성이나 혹은 상태를 기술하는 것을 말한다.

3. 관계 (Relationship)

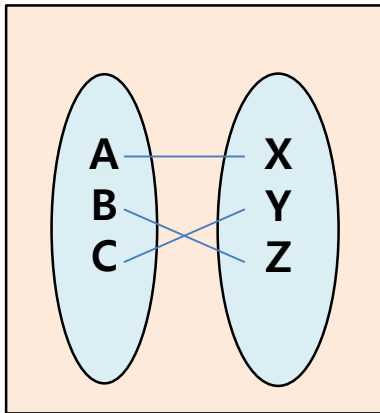
:두 개 이상의 개체 사이 또는 속성 간의 상호 작용을 말한다.

관계의 종류

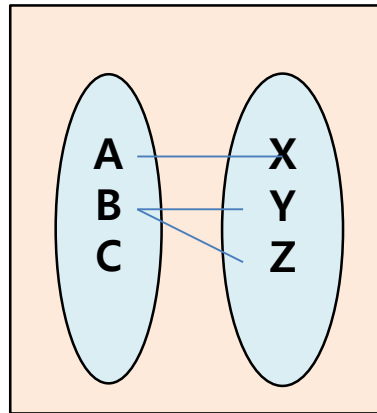
1: 1 (일 대 일)

1: N (일 대 다)

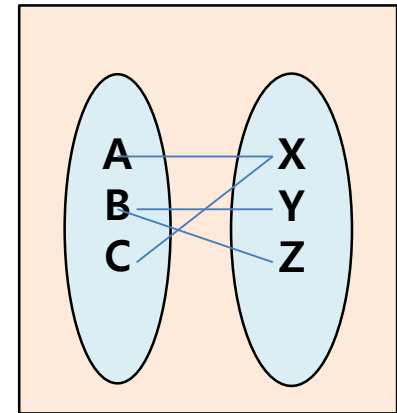
N:M (다 대 다)



1:1 단계



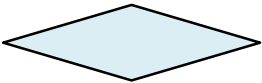

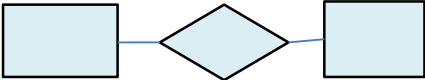
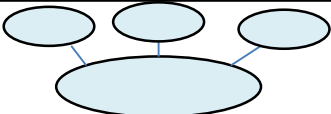


1:N 단계

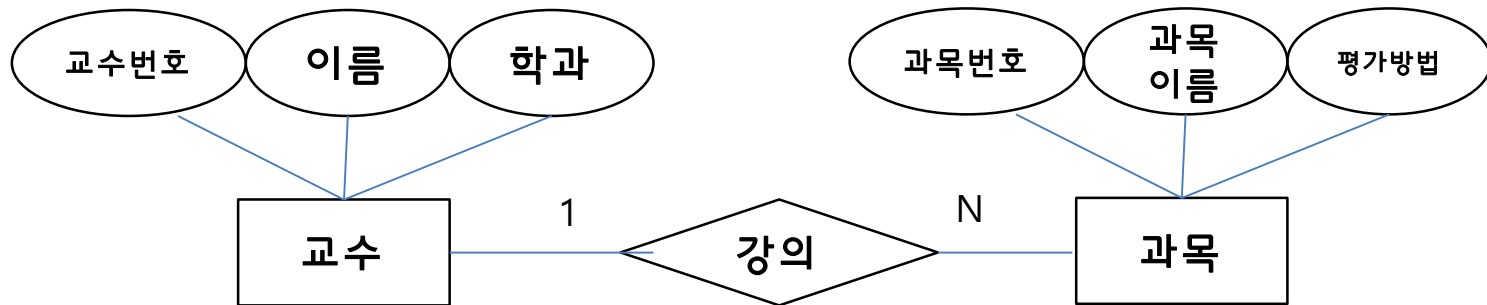


N:M 관계

개체 관계모델(E-R model)

| 기호 | 의미 |
|---|-------------------|
|  | 개체 (Entity) |
|  | 속성 (Attribute) |
|  | 관계 (Relationship) |
|  | 키속성, 기본키속성 |
|  | 개체와 개체 간의 관계 구조 |
|  | 복합 속성 |

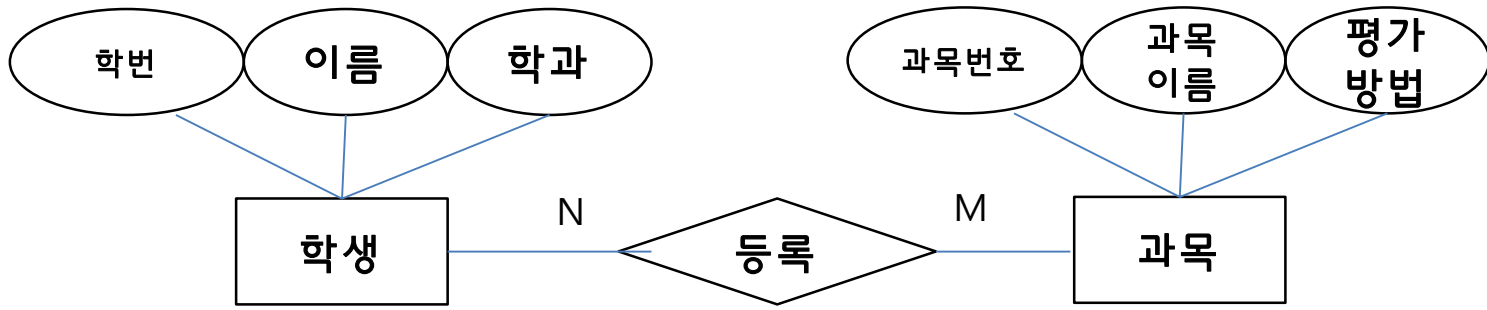
1:N관계



| 번호 | 이름 | 학과 |
|----|-----|-------|
| 1 | 김교수 | 컴퓨터공학 |
| 2 | 박교수 | 국어국문 |

| 과목번호 | 과목이름 | 평가방법 | 교수번호 |
|------|-------|------|------|
| A10 | 컴퓨터개론 | 서술형 | 1 |
| b10 | c언어 | 작업형 | 1 |

N:M관계



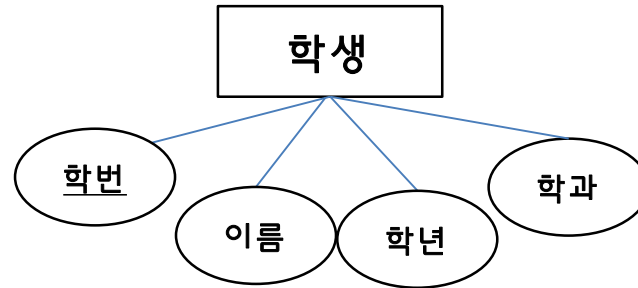
ERD를 릴레이션 스키마로 변환



매핑룰 (Mapping Rule)

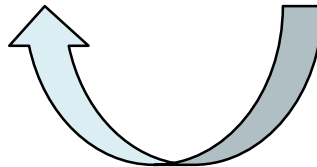
학생

| | | | |
|-----------|----|----|----|
| <u>학번</u> | 이름 | 학년 | 학과 |
|-----------|----|----|----|



By peter chen

논리적모델

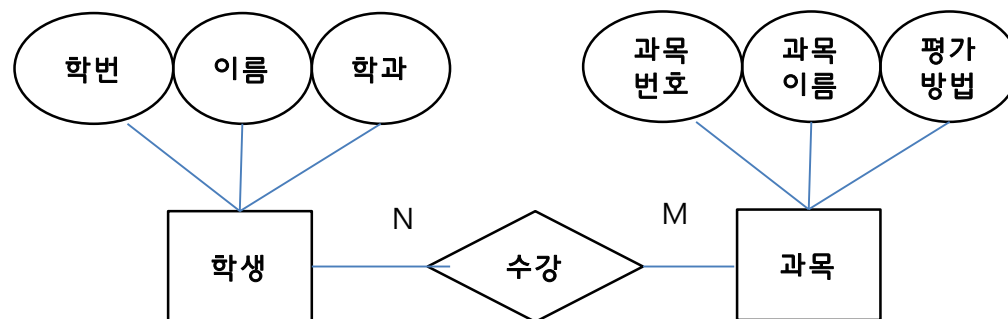


개념적모델

| 학생 | | | |
|-----------|----|----|----|
| <u>학번</u> | 이름 | 학년 | 학과 |

| 수강 | |
|-----------|-------------|
| <u>학번</u> | <u>과목코드</u> |

| 과목 | |
|-------------|-----|
| <u>과목코드</u> | 코드명 |



물리ERD

학생

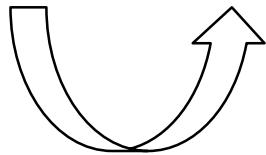
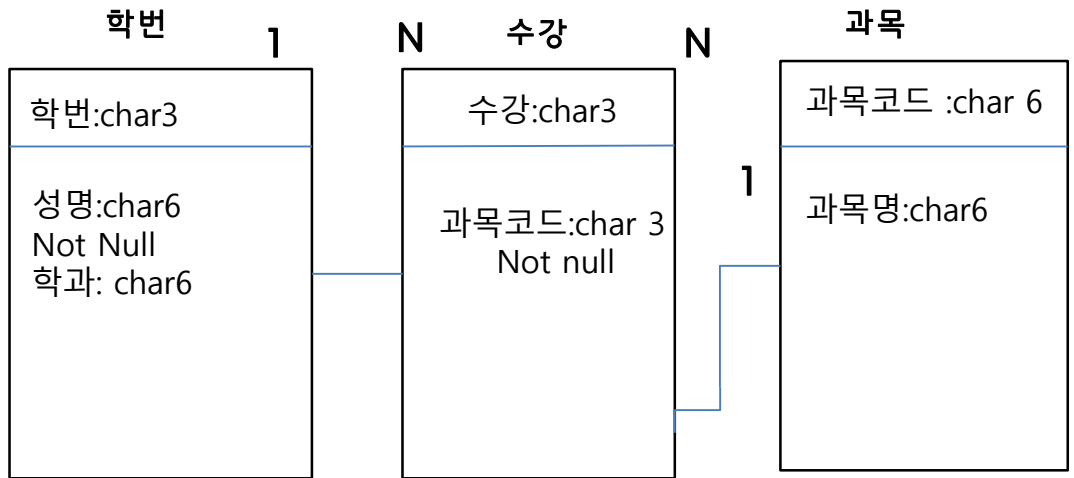
| | | | |
|-----------|----|----|----|
| <u>학번</u> | 이름 | 학년 | 학과 |
|-----------|----|----|----|

수강

| | |
|-----------|------|
| <u>학번</u> | 과목코드 |
|-----------|------|

과목

| | |
|-------------|-----|
| <u>과목코드</u> | 코드명 |
|-------------|-----|



Structured

Query

Language

주문

| <u>id</u> | goods | qty | amt | name | addr |
|-----------|-------|-----|-------|------|------|
| 1 | 공룡책 | 1 | 2000 | 이선원 | 노원구 |
| 2 | 곡물이야기 | 2 | 1500 | 이선원 | 노원구 |
| 3 | 왕딱지 | 10 | 10000 | 이준원 | 마포구 |
| 4 | 공주치마 | 1 | 5000 | 이수연 | 성북구 |

| 고객 | <u>id</u> | name | addr |
|----|-----------|------|------|
| | 1 | 이선원 | 노원구 |
| | 2 | 이준원 | 마포구 |
| 주문 | 3 | 이수연 | 성북구 |

| <u>id</u> | goods | qty | amt | id |
|-----------|-------|-----|-------|----|
| 1 | 공룡책 | 1 | 2000 | 1 |
| 2 | 곡물이야기 | 2 | 1500 | 1 |
| 3 | 왕딱지 | 10 | 10000 | 2 |
| 4 | 공주치마 | 1 | 5000 | 3 |

관계형 데이터베이스의 필요성

테이블 분리하기 - 정규화

테이블 join 하기

1. 데이터 중복 발생
 2. 데이터 중복으로 인한 문제점
 - 중복으로 인한 불필요한 공간 낭비
 - 중복으로 인한 불필요한 작업
- => 데이터 무결성 유지가 힘들어진다

하나의 테이블처럼 결과보기

테이블 join 하기

```
Select * from order_detail left join customer on order_detail.id = customer.id
```

C :INSERT
R : SELECT
U :UPDATE
D :DELETE

사원테이블

```
create table employee  
(  
  name varchar(15) not null,  
  dept varchar(10) not null,  
  birthday datetime not null,  
  addr varchar(20),  
  salary int not null,  
  primary key (name)  
);
```

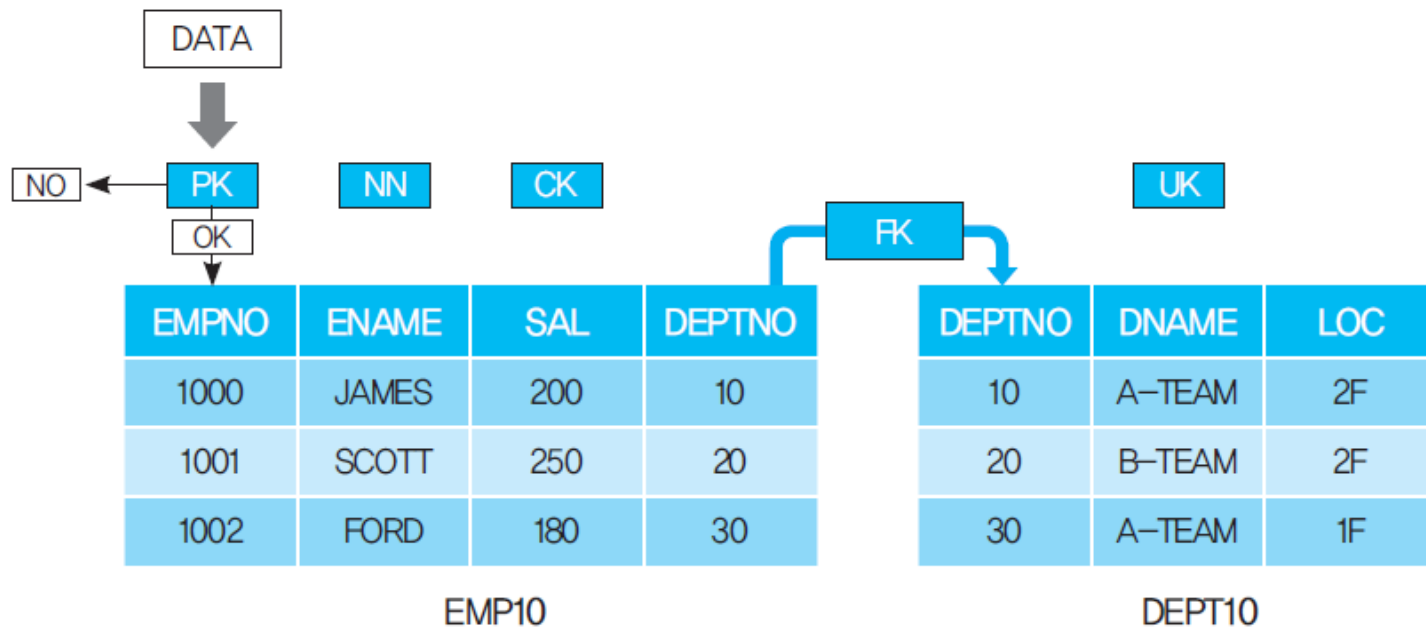
```
create table leisure  
(  
  name varchar(15),  
  experience int,  
  id varchar(15) not null,  
  primary key (name),  
  foreign key(id) references employee (name)  
);
```



```
insert into employee(name, dept, birthday, addr, salary)
values('홍길동' , '기획', '1961-05-04', '효자동', 120);
insert into employee(name, dept, birthday, addr, salary)
values('' , '기획', '1961-05-04', '효자동', 120);
insert into employee(name, dept, birthday, addr, salary)
values('홍길동' , '기획', '1961-05-04', '효자동', 120);
```

제약 조건(Constraint)이란?

테이블에 올바른 데이터만 입력 받고 잘못된 데이터는 들어오지 못하도록 컬럼마다 정하는 규칙을 의미함

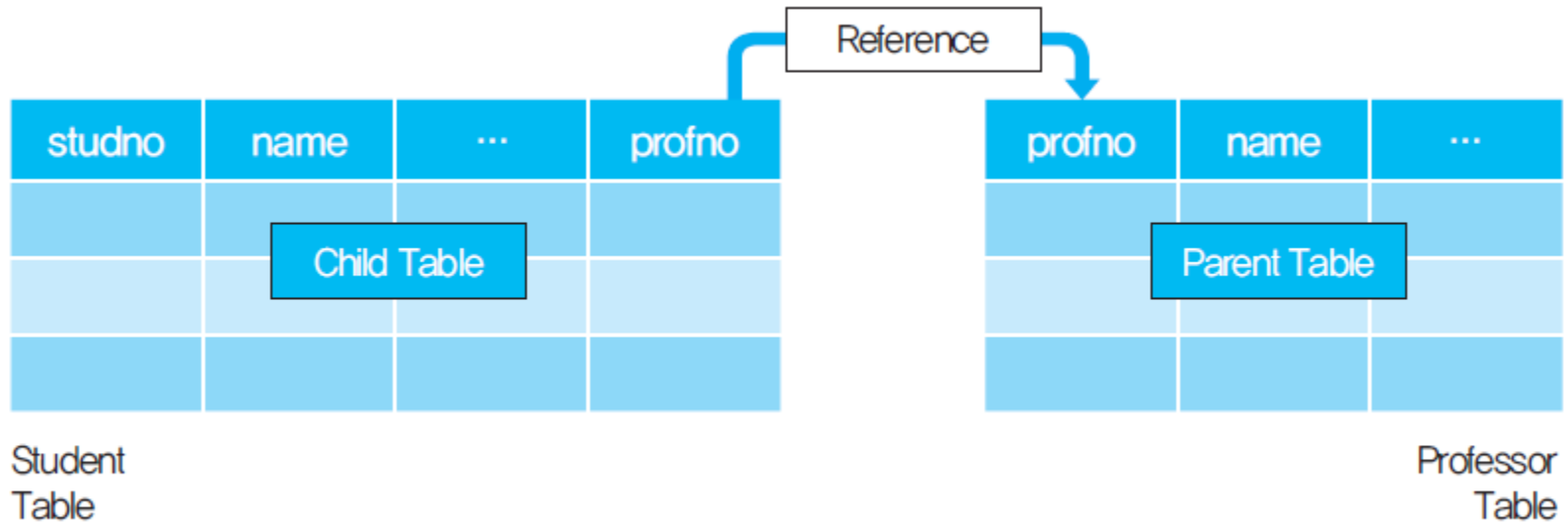


제약조건의 종류

| NOT NULL | 컬럼에 NULL 허용안함 |
|-------------|----------------------|
| UNIQUE | 컬럼에 중복된 값 허용안함 |
| PRIMARY KEY | NOT NULL + UNIQUE |
| FOREIGN KEY | 다른테이블의 컬럼을 참조해서 검사한다 |
| CHECK | 조건에 설정된 값만 입력을 허용 |

FOREIGN KEY 란?

두 개의 테이블을 서로 참조하도록 설정됨



2. 제약 조건 사용하기

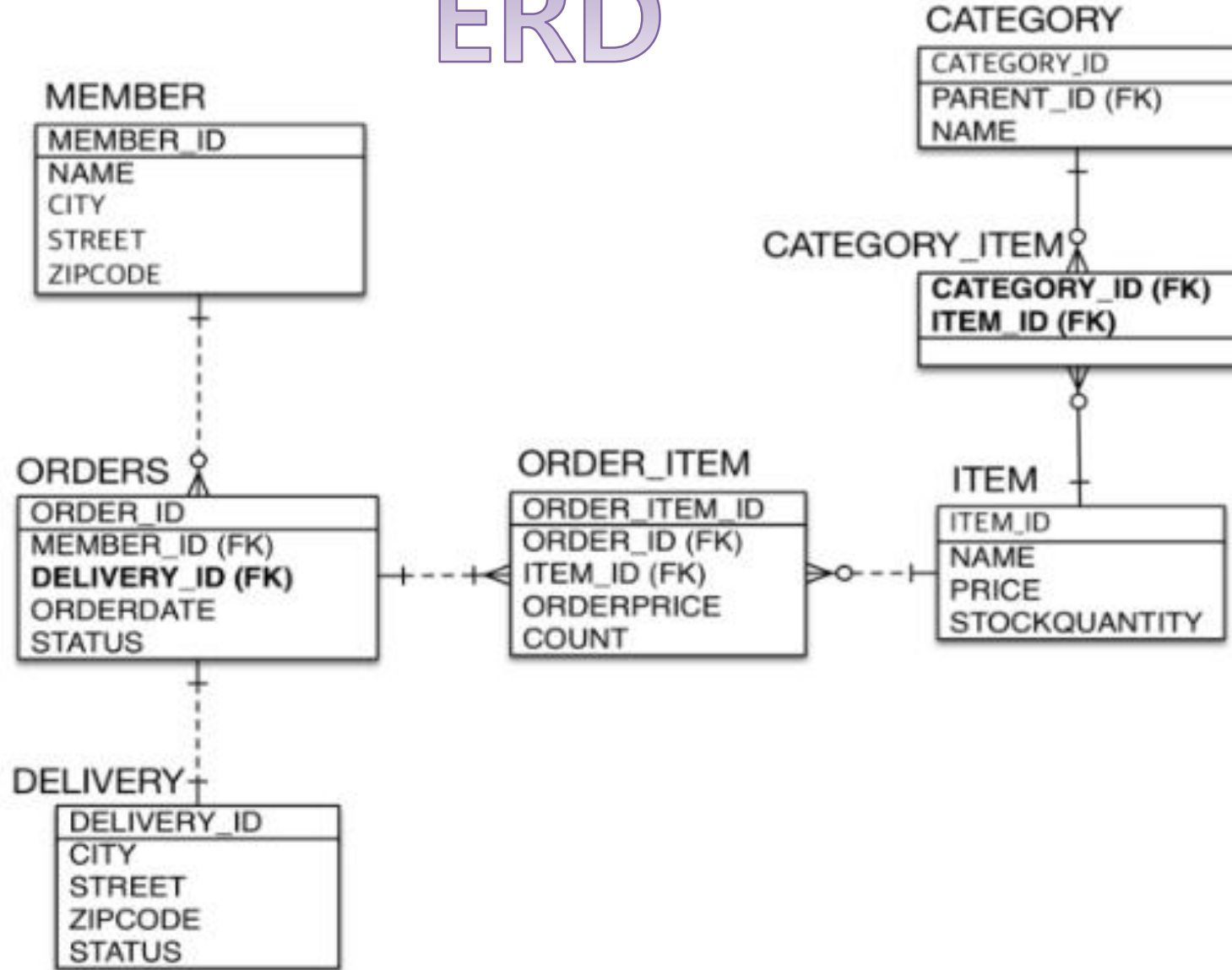
1) 테이블 생성시에 지정하기

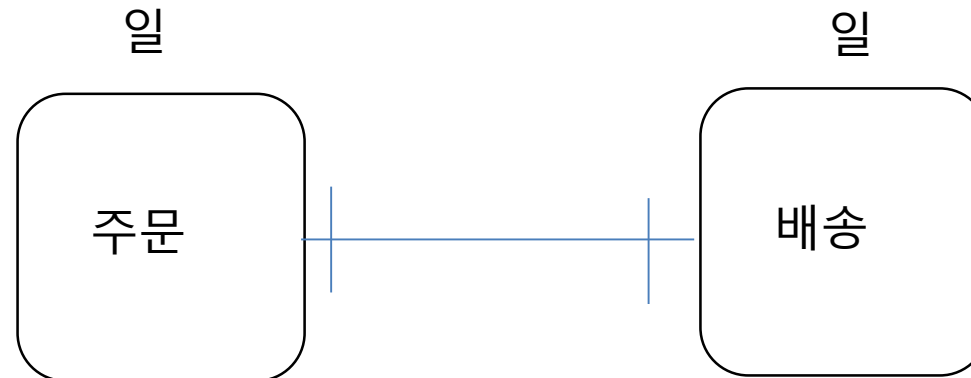
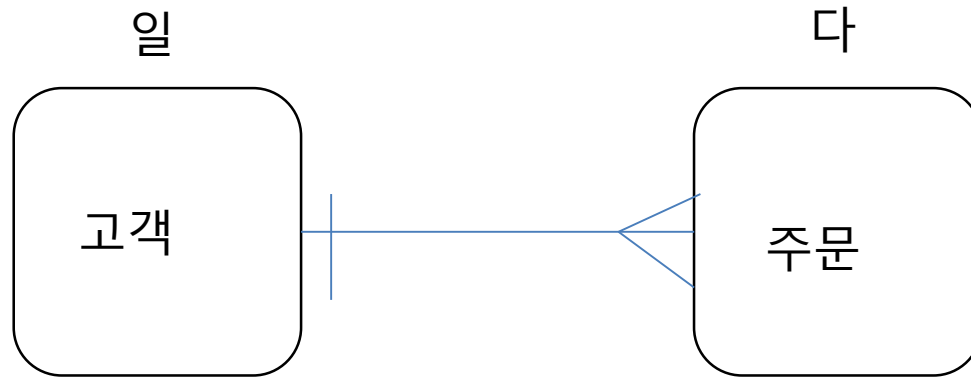
```
SCOTT> CREATE TABLE new_emp1
 2 ( no NUMBER(4)
 3     CONSTRAINT emp1_no_pk PRIMARY KEY,
 4   name VARCHAR2(20)
 5     CONSTRAINT emp1_name_nn NOT NULL,
 6   jumin VARCHAR2(13)
 7     CONSTRAINT emp1_jumin_nn NOT NULL
 8     CONSTRAINT emp1_jumin_uk UNIQUE,
 9   loc_code NUMBER(1)
10     CONSTRAINT emp1_area_ck CHECK ( loc_code < 5 ),
11   deptno VARCHAR2(6)
12     CONSTRAINT emp1_deptno_fk REFERENCES dept2(dcode)
13 ) ;
```

요약(약식)방법

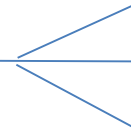
```
CREATE TABLE TEST_EMP(  
    NO NUMBER(4) PRIMARY KEY ,  
    NAME VARCHAR2(20) NOT NULL,  
    JUMIN VARCHAR2(13) NOT NULL UNIQUE,  
    LOC_CODE NUMBER(1) CHECK ( LOC_CODE < 5),  
    GRADE CHAR(1) CHECK ( GRADE IN ('A', 'B', 'C')) ,  
    DEPTNO NUMBER(5) REFERENCES T_DEPT(DEPTNO)  
);
```

ERD

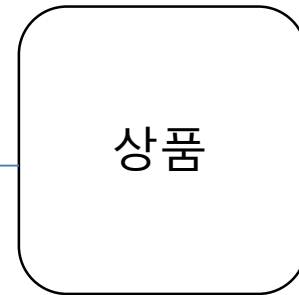
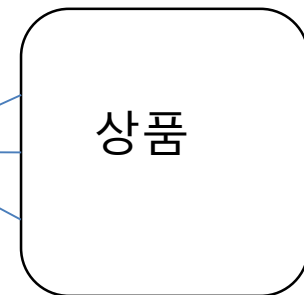




다



다



상품

Reader

| |
|----------------------------|
| ReaderNum: NUMERIC(8) |
| ReaderName: CHAR(10) |
| ReaderAge: TIMESTAMP() |
| ReaderSex: SMALLINT |
| ReaderAdd_do: CHAR(10) |
| ReaderAdd_city: CHAR(10) |
| ReaderAdd_gu: CHAR(10) |
| ReaderAdd_dong: CHAR(10) |
| ReaderAdd_detail: CHAR(30) |
| ReaderTel: NUMERIC(12) |
| ReaderTerm: INTEGER |
| Reader3Term: TIMESTAMP |

Publisher

| |
|------------------------|
| ReaderNum: NUMERIC(8) |
| idPublisher: INTEGER |
| sendDate: DATE |
| passwordOwner: INTEGER |



이상(Anomaly) 과 함수적 종속

-이상(Anomaly)

- 삭제이상 (Deletion Anomaly)
: 원하지 않은 자료까지 함께 삭제가 이루어져 발생하는 문제
- 삽입이상(Insertion Anomaly)
: 원하지 않는 자료가 삽입
- 갱신이상(Update Anomaly)
: 일관성이 없어져 정확한 정보 파악이 안되는 상황

함수적 종속(Functional Dependency)

완전함수 종속:

릴레이션에서 한 속성이 오직 기본키에만 종속이 되는 경우

부분 함수 종속

복합키인 경우 키의 일부에 의해 종속되는 경우

함수적 종속

릴레이션에서 속성 A,B가 있을 때, 임의 튜플에서 A의 값이 B의 값을 함수적으로 결정한다면

A의 값을 알면 B의 값을 알 수 있거나 A의 값에 따라 B의 값이 달라진다면
B는 A에 함수적으로 종속되었다고 한다

'A→B'

B가 A에 종속되어 A 값을 알 수 있을 때
A(결정자) → B(종속자)

'학번' , '성명' , '수강과목' , '학년'

| 학번 | 성명 | 수강과목 | 학년 |
|--------|-----|--------|----|
| 190111 | 김철수 | 정보통신 | 1 |
| 181010 | 이철준 | 컴퓨터 | 3 |
| 190223 | 박태인 | 데이터베이스 | 1 |
| 172020 | 김길동 | 운영체제 | 2 |
| 181533 | 이영진 | 산업공학 | 3 |
| 161017 | 최길동 | 컴퓨터 | 4 |
| 162111 | 이철준 | 데이터베이스 | 4 |

학번->성명
학번->수강과목
학번->학년

'성명' , '수강과목' , '학년' 은
학번에 종속되었다고 한다.

주문

| 고객번호 | 제품번호 | 제품명 | 단가 | 주문량 |
|------|-------|-------|--------|-----|
| A012 | S-321 | SD메모리 | 25000 | 2 |
| A012 | M-789 | 메모리 | 28000 | 1 |
| A023 | K-002 | 키보드 | 5000 | 1 |
| A123 | K-012 | 헤드셋 | 10000 | 2 |
| A134 | M-123 | 마우스 | 6000 | 4 |
| A134 | S-321 | SD메모리 | 25000 | 2 |
| A789 | S-567 | 스캐너 | 100000 | 1 |

(고객번호, 제품번호) -> 주문량

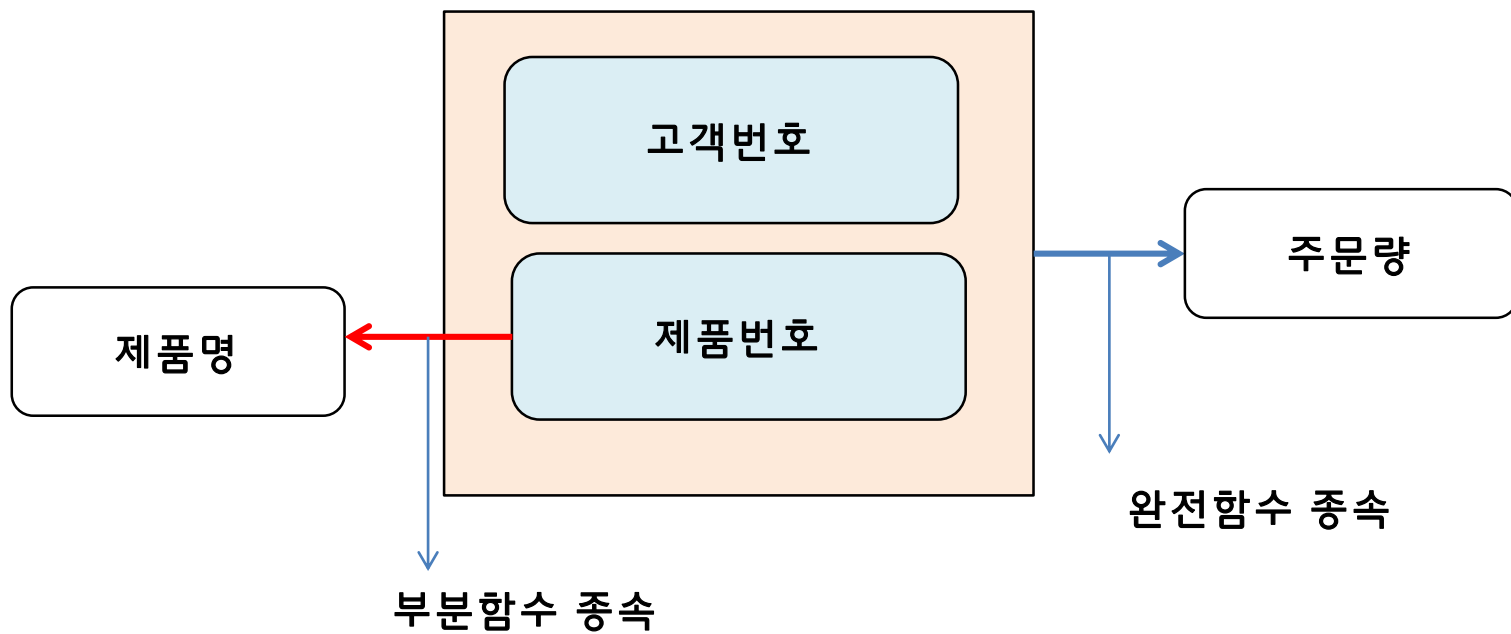
완전함수 종속

제품번호->제품명

부분함수 종속

복합키의 일부키가 결정자 되는 경우

<복합키경우 발생할 수 있음>



정규화(Normalization)

- ❖ 정규화는 논리적 설계 단계에서 발생할 수 있는 종속으로 인한 이상 (Anomaly)현상의 문제점을 해결하기 위해, 속성들간의 종속 관계를 분석하여 여러 개의 릴레이션으로 분해하는 과정을 말한다.
- ❖ 정규화된 결과를 정규형이라고 하며, 정규형의 종류로는 제1정규형, 제2정규형, 제3정규형, 제4정규형, 제5정규형등이 있다.

제1 정규화

제1 정규형은 릴레이션을 구성하는 모든 도메인이 원자값만으로 구성 되도록 하는 정규형이다.

회원

| 회원번호 | 성명 | 연락처 | 수강과목 | 수강료 |
|-------|-----|----------|-------|-------|
| 10010 | 박순신 | 123-4567 | POP글씨 | 40000 |
| | | | 지점토공예 | 40000 |
| 20020 | 이감찬 | 234-1122 | 팬글씨 | 30000 |
| 20030 | 김길동 | 321-4321 | 지점토공예 | 40000 |
| | | | 기타 | 50000 |

회원 (회원번호, 성명, 연락처, 수강과목, 수강료)

회원 (회원번호, 성명)
강좌 (수강과목, 수강료)

| <u>회원번호</u> | 성명 | 연락처 |
|-------------|-----|----------|
| 10010 | 이순신 | 123-4567 |
| 20020 | 이감찬 | 234-1122 |
| 20030 | 김길동 | 321-4321 |

| <u>수강과목</u> | 수강료 |
|-------------|-------|
| Pop글씨 | 40000 |
| 지점토공예 | 40000 |
| 팬글씨 | 30000 |
| 기타 | 50000 |

제2 정규화

제1 정규형을 만족하면서 릴레이션을 구성하는 모든 속성이 기본키에 완전 함수 종속이 되도록 분해하는 과정을 말한다
부분함수 종속을 제거하고 모든 속성이 **기본키에 완전 함수 종속**이 되도록 한다.

고객주문

| <u>고객번호</u> | <u>제품번호</u> | 제품명 | 주문량 |
|-------------|-------------|-------|-----|
| A012 | S-321 | SD메모리 | 2 |
| A012 | M-789 | 메모리 | 1 |
| A023 | K-002 | 키보드 | 1 |
| A123 | K-012 | 헤드셋 | 2 |
| A134 | M-123 | 마우스 | 4 |
| A134 | S-321 | SD메모리 | 2 |

(고객번호, 제품번호) ->주문량
제품번호->제품명

부분함수 종속 관계가 있는 테이블을 기본키에 완전 함수 종속이 되도록 분해

주문량(고객번호, 제품번호, 주문량)
제품(제품번호, 제품명)

주문량

| <u>고객번호</u> | <u>제품번호</u> | 주문량 |
|-------------|-------------|-----|
| A012 | S-321 | 2 |
| A012 | M-789 | 1 |
| A023 | K-002 | 1 |
| A123 | K-012 | 2 |
| A134 | M-123 | 4 |
| A134 | S-321 | 2 |

제품

| <u>제품번호</u> | 제품명 |
|-------------|-------|
| S-321 | SD메모리 |
| M-789 | 메모리 |
| K-002 | 키보드 |
| K-012 | 헤드셋 |
| M-123 | 마우스 |
| S-567 | 스캐너 |