

데이터베이스 모델링

세부목차

1 데이터베이스 모델링

2 개념적 모델링

3 논리적 모델링

4 물리적 모델링

5 데이터 베이스 모델링 실습

관계형 데이터베이스

➤ 정의

- 정보구조를 체계적으로 기술하는 방법으로 데이터를 사용자 관점에서 인식 및 분석하고 표준화된 심볼을 표현하는 기법으로 현실세계의 업무 프로세스를 물리적으로 데이터베이스화 시키는 작업

➤ 필요성

- 사용자 관점에서 정보요건을 분석
- 비즈니스 지식의 정확한 mapping
- 통합적이고, 안정적인 데이터베이스 설계

관계형 데이터베이스

- 모든 데이터들을 테이블과 같은 형태로 나타내어 저장하는 데이터베이스
- 일상 생활에서 데이터를 정리하여 표현할 때 흔히 표와 같은 방법을 사용하게 되는데, 관계형 데이터베이스는 이 '표'의 개념을 사용해서 데이터를 구성하는 방법을 사용

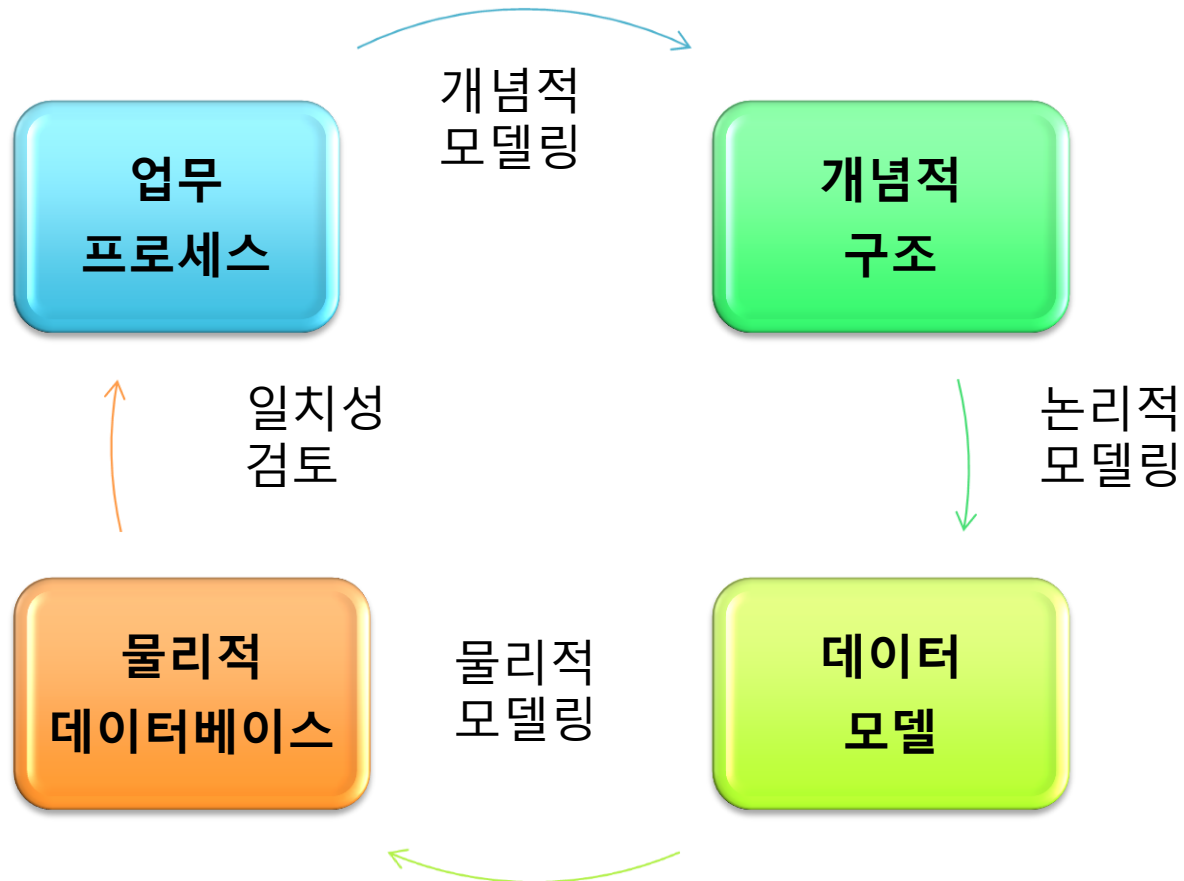
관계형 데이터베이스 – 테이블의 정의

열(속성, Attribute)

행(Tuple)

책제목	출판사	저자명	가격	
...	...	3.78	...	
SQLite3	Wikibooks	Derick, DSP	30000	
HTML5	jaen	이세우	26000	
JSP guide	Apress	7000	
software	Wikibooks	22000	
Android...	jaen	신동진	25000	
...	
...	

데이터베이스 모델링 개요



세부목차

1 데이터 베이스 모델링

2 개념적 데이터베이스 모델링

3 논리적 데이터베이스 모델링

4 물리적 데이터베이스 모델링

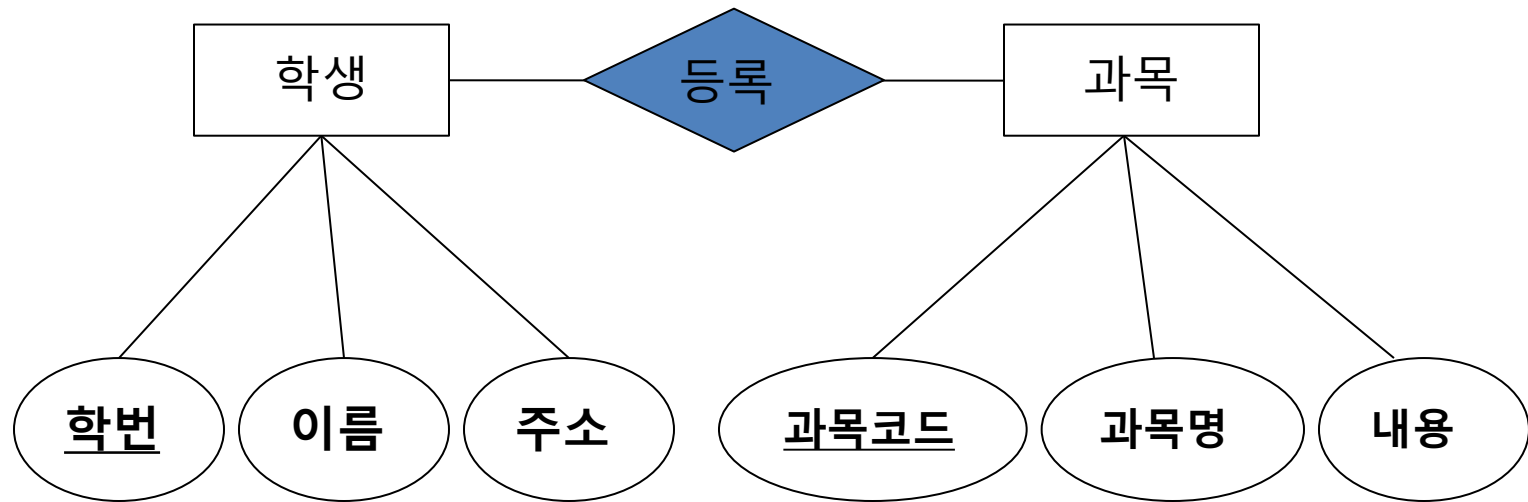
5 데이터 베이스 모델링 실습

개념적 데이터베이스 모델링

➤ 개념적인 데이터 모델링

- 업무분석 단계에서 얻어진 내용을 토대로 우선 엔티티(Entity)를 추출하고 엔티티내에 속성(Attribute)을 구성하며 엔티티간의 관계를 정의해서 ER-Diagram 을 정의하는 단계이다.

➤ [ER-Diagram 작성 예]



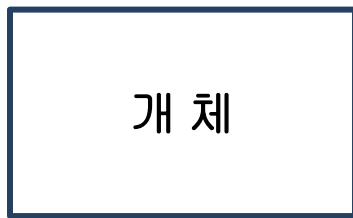
개념적 데이터베이스 모델링

➤ 개념적 데이터 모델링의 절차를 살펴보면 다음과 같다.

1. 사용자 부문의 처리현상을 분석한다.
2. 중요 실체와 관계를 파악하여 ERD를 작성한다.
3. 실체에 대한 상세 정의를 한다.
4. 식별자를 정의하고, 식별자 업무규칙을 정한다.
5. 실체별로 속성을 상세화 한다.
6. 필요한 속성 및 영역을 상세 정의한다.
7. 속성에 대한 업무규칙을 정의한다.
8. 각 단계를 마친 후 사용자와 함께 모델을 검토한다.

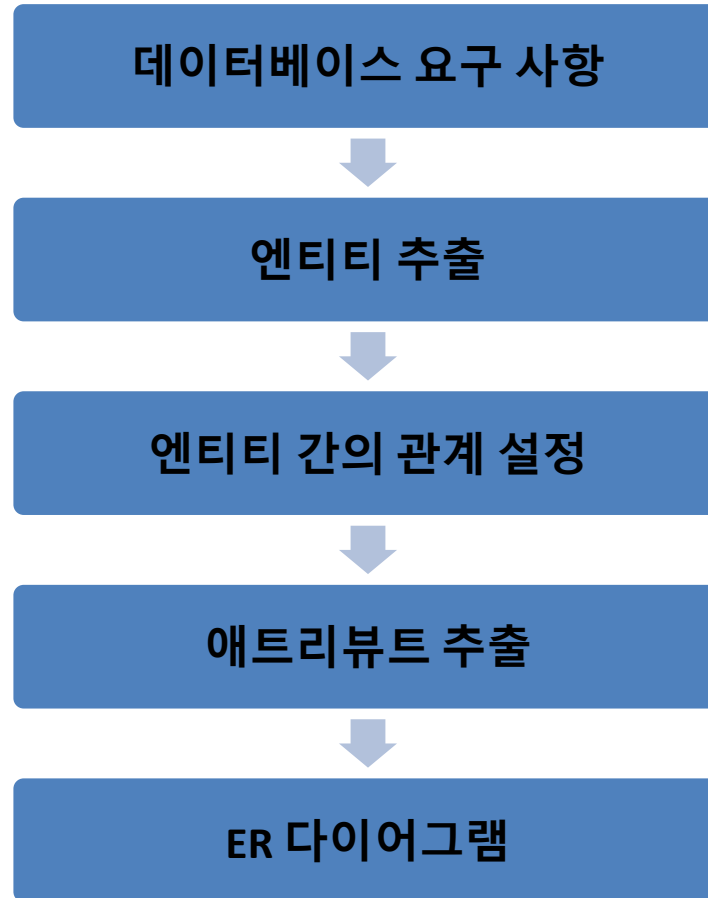
개념적 데이터베이스 모델링

➤ Entity Relationship (E-R) Diagram 기호



개념적 데이터베이스 모델링

▶ 개념적 설계 단계



개념적 데이터베이스 모델링

➤ Entity 정의

- 사용자와 관계가 있는 주요 객체로 업무 수행을 위해 데이터로 관리되어야 하는 사람, 사물, 장소, 사건 등을 실체(Entity) 라고 한다.

➤ Entity 찾는 법

- 영속적으로 존재하는 것
- 새로 식별이 가능한 데이터 요소를 가짐
- 명사적 표현
- Entity는 반드시 Attribute를 가져야 함.

개념적 데이터베이스 모델링

▶ Entity 예제

- ① 사람(사원(직원, 행원, 공원,...), 계약자(가입자, 회원,...), 이용자(학생, 환자,...))
- ② 물건(재료(부품, 원자재, 연료, ...), 상품(제품,...), 시설(건물, 창고, 운송센터,...),
지점(영업소, 소매점,...))
- ③ 사건(계약(수주, 발주,...), 작업(공정, 보관, 선전, 광고,...), 사고(재해, 고장,...))
- ④ 장소(구획(창고, 선반, 진열케이스, 생산라인, ...), 지역(판매구역, 관할구, 선거구,...),
하천, 항만(부두, 선창,...))
- ⑤ 개념(목표, 계획(지침, 방침, 지표, 판매목표, 생산계획, 판매계획, 인원계획,...),
시간(월, 일, 년, 시각, 시각분할,...), 평가(기준, 지표))
- ⑥ 금전(예입금(구좌,...), 예산(년간예산, 수정예산, 실행예산,...), 차입(단기, 장기,...),
융자(단기, 장기,...))

개념적 데이터베이스 모델링

➤ Entity 작성 법

- Softbox로 표현
- Entity명은 단수명사이고, 현업용어를 사용하며 유일하게 부여
- Entity명은 대문자, 크게 표시
- Attribute명은 소문자, 작게 표시
- Entity명을 Attribute명과 같게 사용하지 말 것
- 보통 Entity는 다수의 사용형태(instance)를 가져야 함.
- 모든 Entity의 사용형태는 Attribute에 특정한 값을 가져야 함.
- 유일한 식별자가 없다면 Entity가 아님.

개념적 데이터베이스 모델링

➤ 속성(Attribute)

- 저장할 필요가 있는 실체에 관한 정보
- 실체(Entity)의 성질, 분류, 수량, 상태, 특성 등을 나타내는 세부사항
- 실체에 포함되는 속성의 숫자는 10개 내외로 하는 것이 바람직함
- 최종 DB 모델링 단계를 통해 테이블의 컬럼으로 활용
- Ex)
 - 학생 : 학번, 이름, 주민번호, 전화번호, 주소
 - 직원 : 직원ID, 이름, 주민번호, 주소, 입사일자, 소속부서

개념적 데이터베이스 모델링

➤ 속성(Attribute)의 유형

- 기초속성
 - 원래 갖고 있는 속성으로 현업에서 기본적으로 사용되는 속성
- 추출 속성
 - 기초 속성으로부터 계산(가공)에 의해 얻어질 수 있는 속성
- 설계 속성
 - 실제로 존재하지 않으나 시스템의 효율성을 도모하기 위해 설계자가 임의로 부여하는 속성

개념적 데이터베이스 모델링

➤ 식별자

- 한 실체(Entity) 내에서 인스턴스를 유일하게 구분할 수 있는 단일 속성 또는 속성 그룹
- 후보키(Candidate Key)
 - 실체내에서 각각의 인스턴스를 구분할 수 있는 속성으로 기본키가 될 수 있는 후보 속성
- 기본키(Primary Key)
 - 실체(Entity)에서 각 인스턴스를 유일하게 식별하는데 가장 적합한 Key
 - 기본키 설정시 고려할 사항으로 해당 실체를 대표할수 있을 것, 업무적으로 활용도가 높을 것, 길이가 짧을 것 등이 있다

개념적 데이터베이스 모델링

➤ 식별자

- 대체키(Alternate Key)
 - 후보키중에서 기본키로 선정되지 않은 속성
- 복합키(Composite Key)
 - 하나의 속성으로 기본키가 될 수 없는 경우 둘 이상의 컬럼을 묶어서 식별자로 정의하는 경우
- 대리키(Surrogate Key)
 - 식별자가 너무 길거나 여러 개의 속성으로 구성되어 있는 경우에 인위적으로 추가한 식별자

개념적 데이터베이스 모델링

➤ 관계(Relationship)

- 정의

두 Entity간의 업무적인 연관성 또는 관련 사실

- relationship 분석

entity간에 특정한 관계의 존재여부 결정

- 각 Entity간에 특정한 존재여부 결정

- 현재의 관계뿐만 아니라 장래에 사용될 경우도 고려

- Relationship Matrix 활용

개념적 데이터베이스 모델링

➤ 관계(Relationship)

- ER-Diagram 으로 관계를 설정하는 순서
 - 1) 관계가 있는 두 실체를 실선으로 연결하고 관계를 부여한다.
 - 2) 관계차수를 표현한다.
 - 3) 선택성을 표시한다.



두 실체를 실선으로 연결하고 등록이라는 관계를 마름모로 표현한다.

개념적 데이터베이스 모델링

▶ 관계(Relationship) 표현

- 차수성(cardinality)

차수성이란 한 실체의 하나의 인스턴스가 다른 실체의 몇 개의 인스턴스와 관련될 수 있는가를 정의하는 것이다.

- 경우에 따라 발생횟수를 조사
- 양쪽 방향 모두 조사



- 선택성(optionality)

선택적(optional)이나 반드시(mandatory)를 나타낸다.

- 일반적이고 상식적인 선에서 먼저 판단
- 항상 그 관계를 만족해야만 하는지 파악
- 관계가 만족되지 않는 경우를 찾아보고 하나라도 만족되지 않는 경우가 있으면 **optional** 로 표시
- 양쪽 방향 모두 조사

개념적 데이터베이스 모델링

➤ 차수성의 종류

- 1 : 1(일대일)

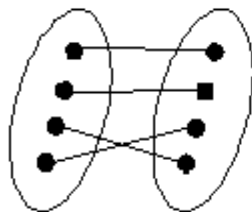
두 실체의 레코드가 서로 하나씩 대응

- 1 : n(일대다)

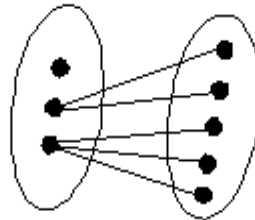
부모 실체의 하나의 레코드가 자식 실체의 여러 레코드에 대응

- n : n(다대다)

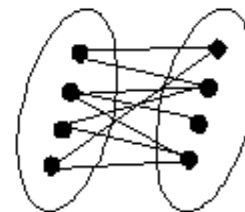
양 쪽 실체간에 여러 개의 레코드와 관계를 맺을 수 있는 경우



1 : 1 관계



1 : 다 관계

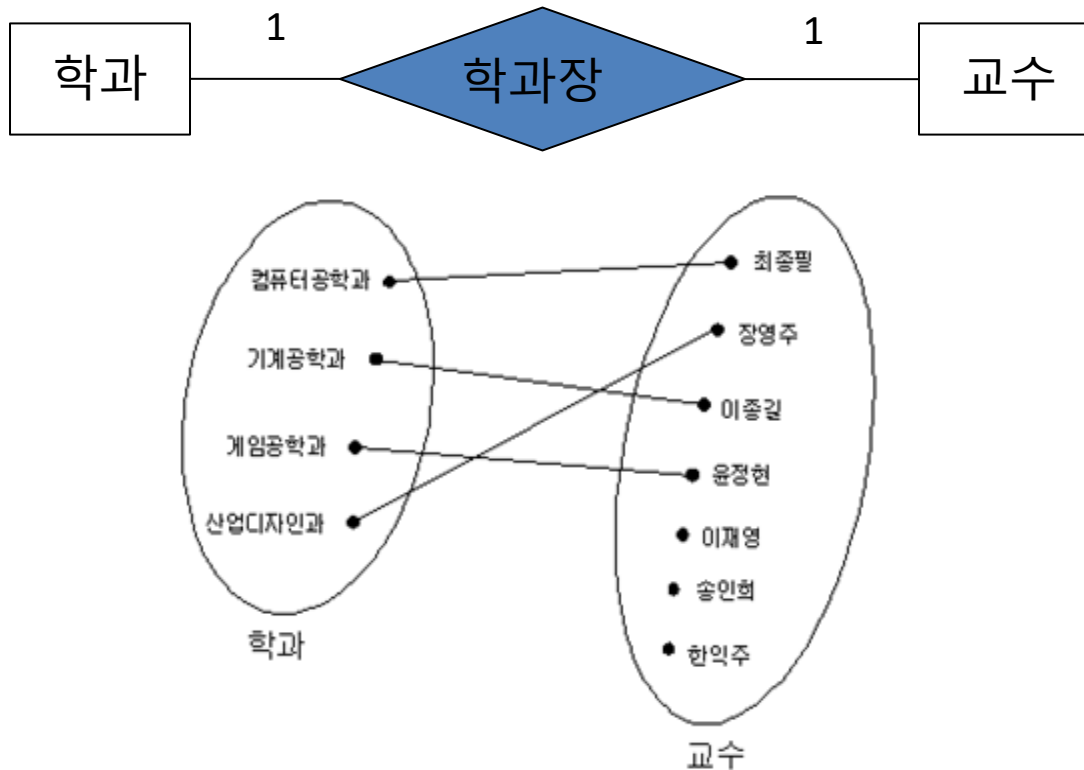


다 : 다 관계

개념적 데이터베이스 모델링

▶ 차수성의 종류

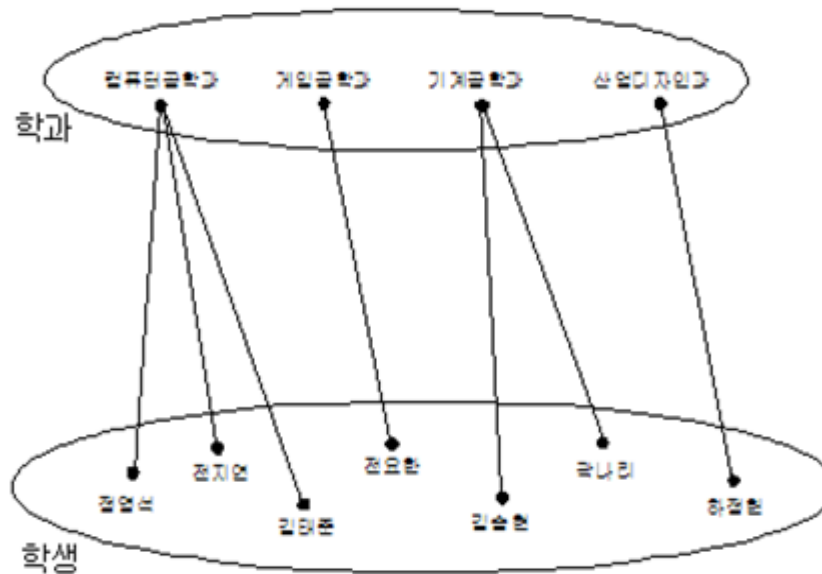
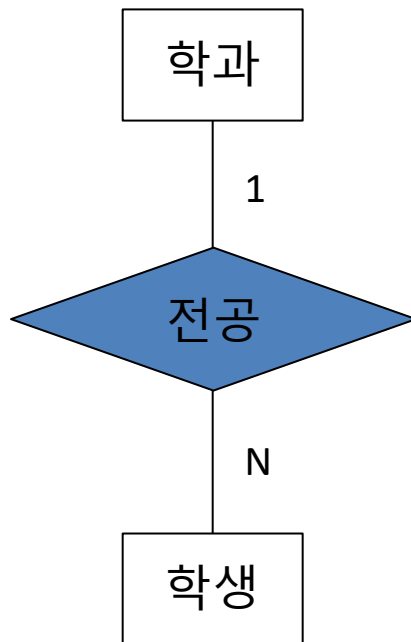
- 1:1 학과와 학과장 사이



개념적 데이터베이스 모델링

▶ 차수성의 종류

- 1 : n 학과와 학생 사이



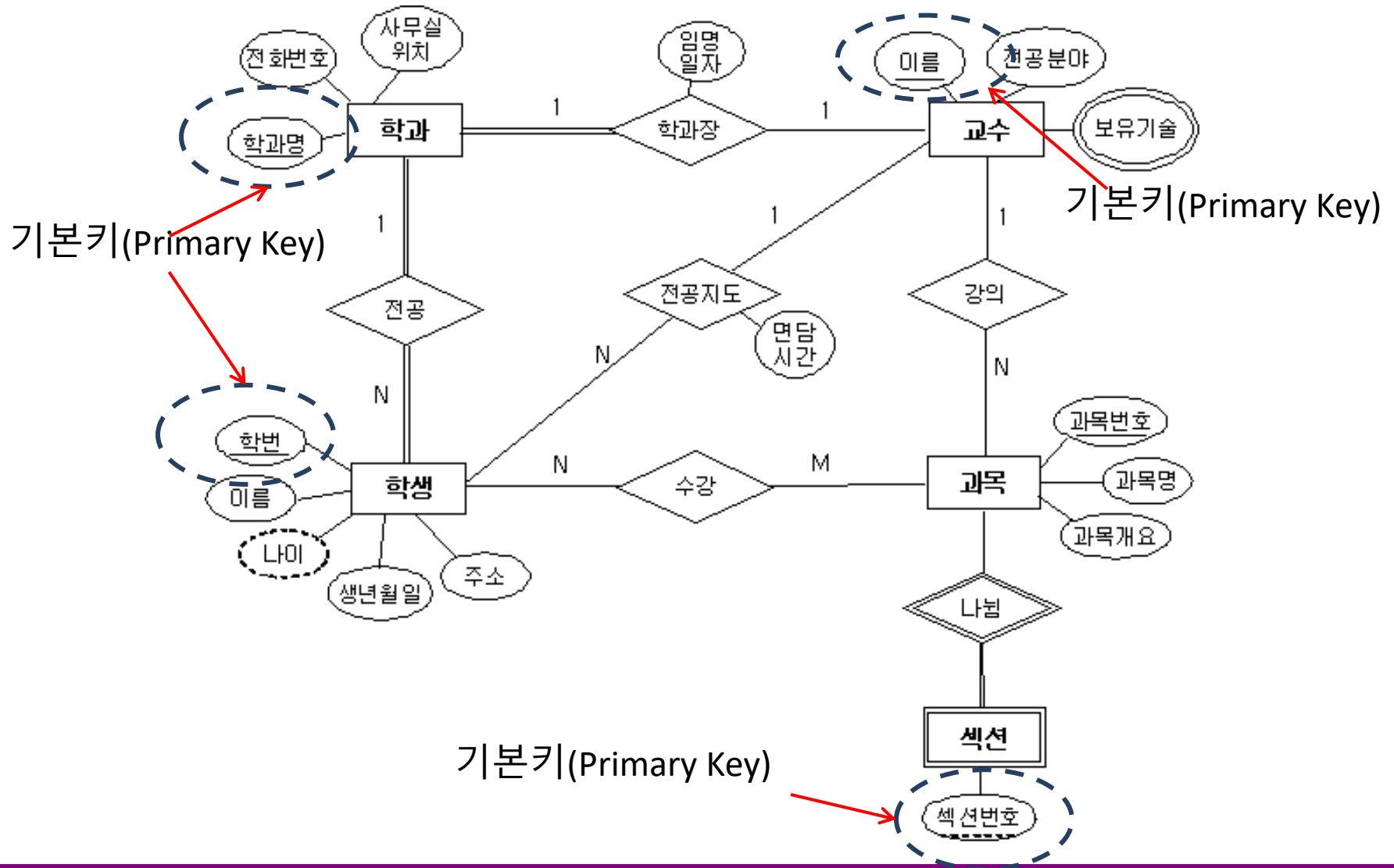
개념적 데이터베이스 모델링

▶ 차수성의 종류

- $n:n$ 학생과 과목 사이



개념적 데이터베이스 모델링



세부목차

- 1 데이터 베이스 모델링 개요
- 2 개념적 데이터베이스 모델링
- 3 논리적 데이터베이스 모델링**
- 4 물리적 데이터베이스 모델링
- 5 데이터 베이스 모델링 실습

논리적 데이터베이스 모델링

➤ 논리적 데이터베이스 모델링 과정

- 개념적 데이터베이스 모델링 단계에서 정의된 ER-Diagram을 Mapping Rule 을 적용하여 관계형 데이터베이스 이론에 입각한 스키마를 설계하는 단계와 이를 정규화 하는 단계로 구성

- 스키마(Schema)

데이터베이스의 구조로써 데이터베이스 내의 개체와 개체들의 관계, 제약조건 등에 대한 명세

논리적 데이터베이스 모델링

➤ 기본키(Primary Key) 와 포린키(Foreign Key)

- 기본키

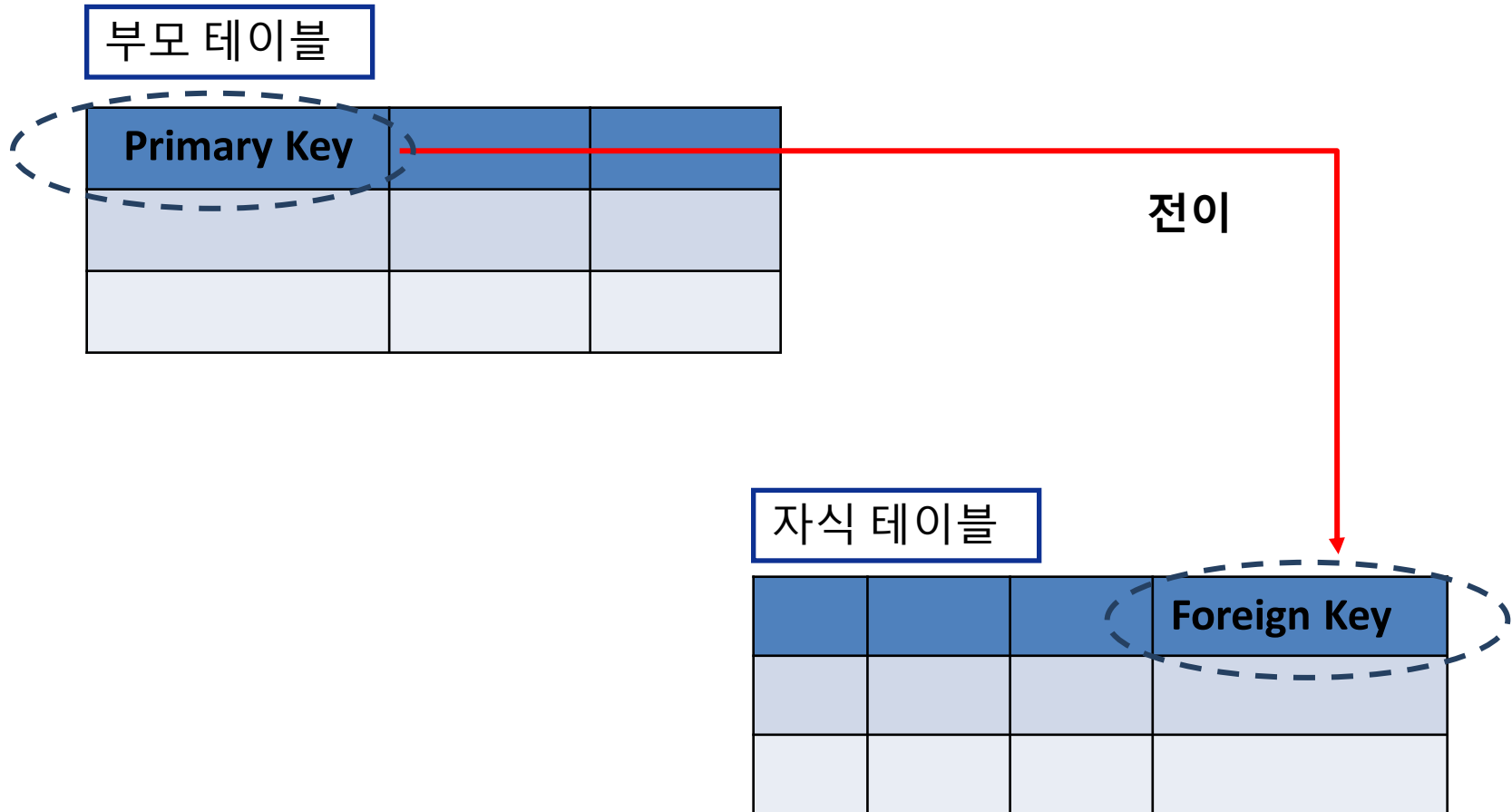
- 후보키 중에서 선택한 주 키
- 널(Null)을 값으로 가질 수 없다
- 동일한 값이 중복해서 저장될 수 없다

- 포린키

관계를 맺는 두 엔티티에서 서로 참조하는 릴레이션의 애트리뷰트로 지정되는 키 값을 말한다

논리적 데이터베이스 모델링

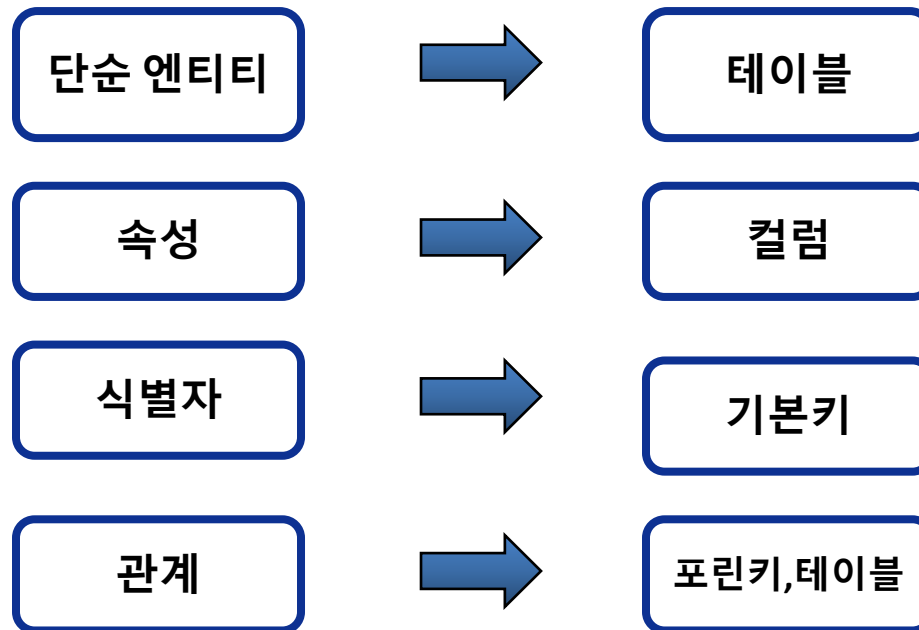
➤ 기본키(Primary Key)와 포린키(Foreign Key) 관계



논리적 데이터베이스 모델링

➤ Mapping Rule

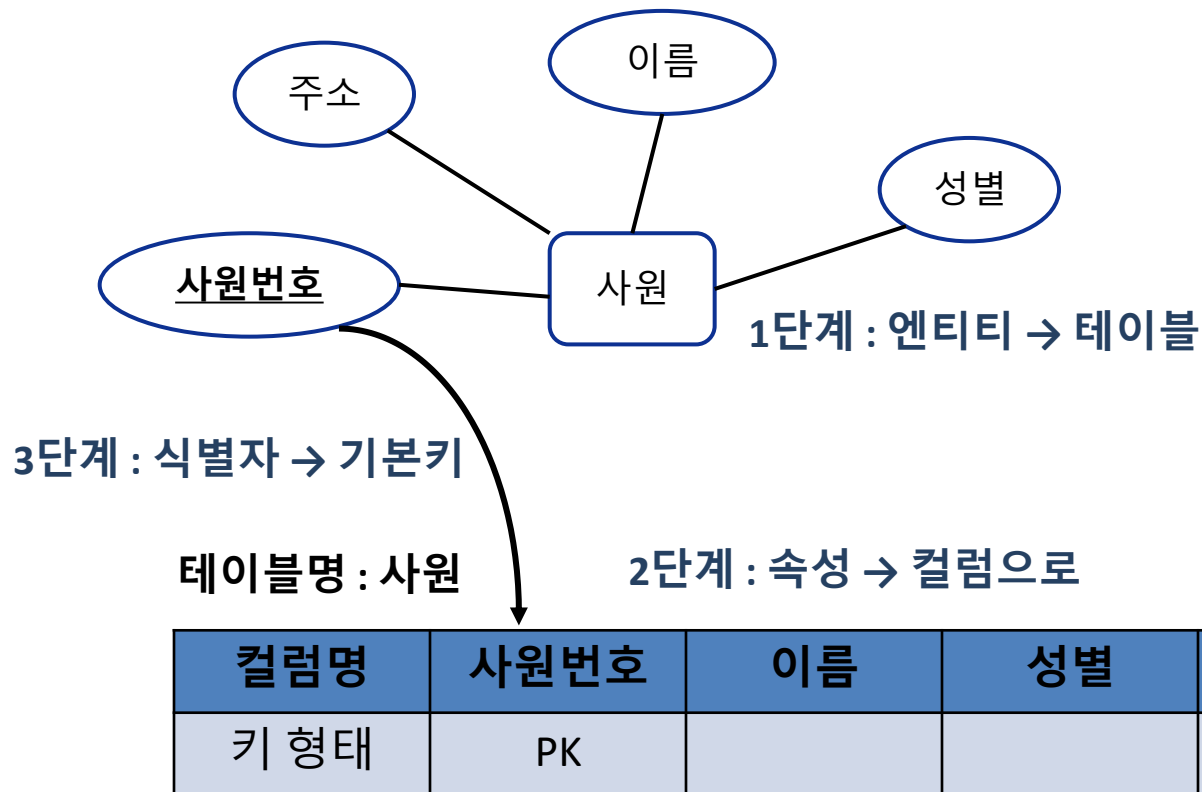
- 개념적 데이터베이스 모델링에서 도출된 개체 타입과 관계타입의 테이블 정의



논리적 데이터베이스 모델링

➤ Mapping Rule

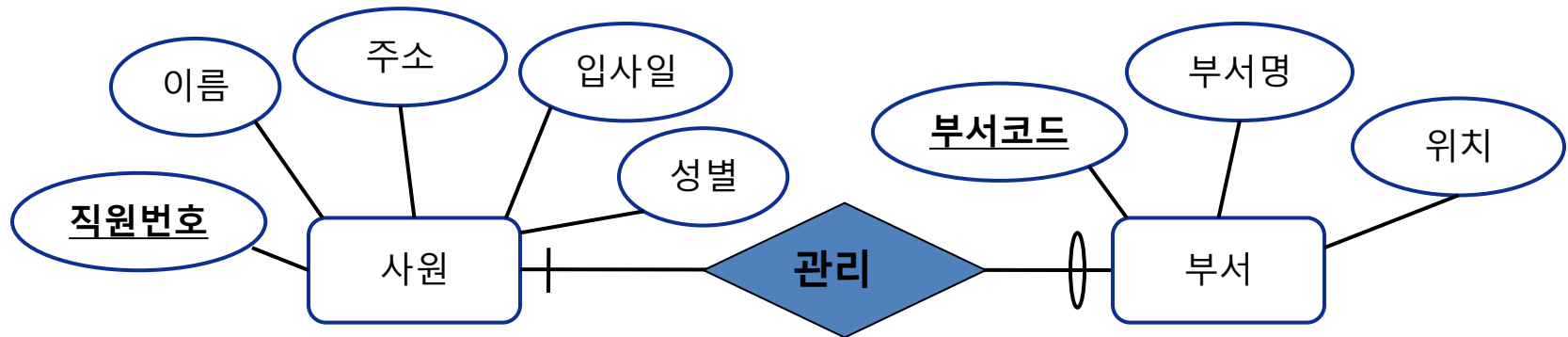
- 사원 엔티티 → Mapping Rule 을 적용하여 관계 스키마로 변환



논리적 데이터베이스 모델링

➤ Mapping Rule

- 1 : 1 관계(직원과 부서의 ER-Diagram → 관계 스키마)



좌 => 우 (Optional)

우 => 좌 (mandatory)

사원번호(PK)	이름	주소	성별	입사일
1	이동국	전주	남	20120101
2	아이유	나고야	여	20120101
3	차범근	서울	남	20120101

테이블명 : 사원

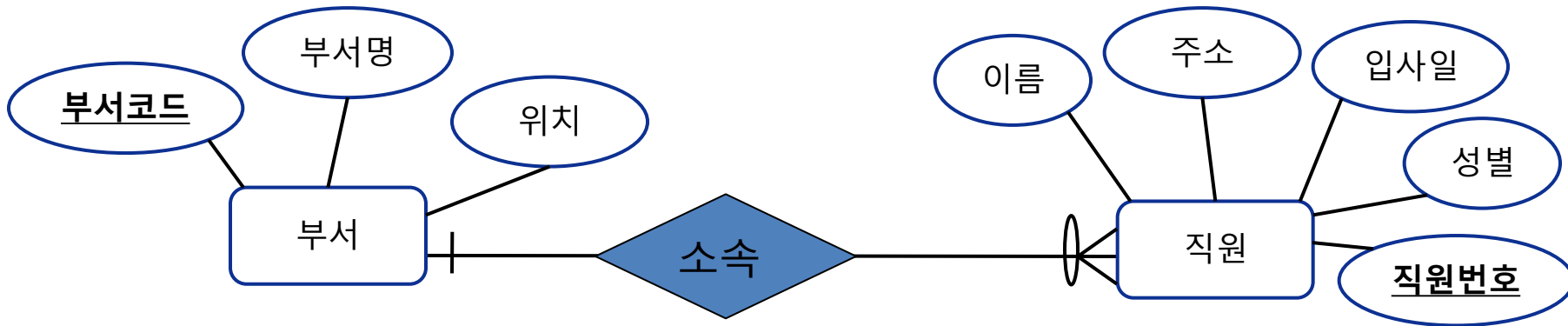
부서코드(PK)	부서명	위치	사원번호 (FK)
01	총무부	서울	1
02	연구소	판교	2

테이블명 : 부서

논리적 데이터베이스 모델링

➤ Mapping Rule

- 1 : N 관계(부서과 직원의 ER-Diagram → 관계 스키마)



테이블명 : 직원

부서코드(PK)	부서명	위치
01	총무부	서울
02	연구소	판교

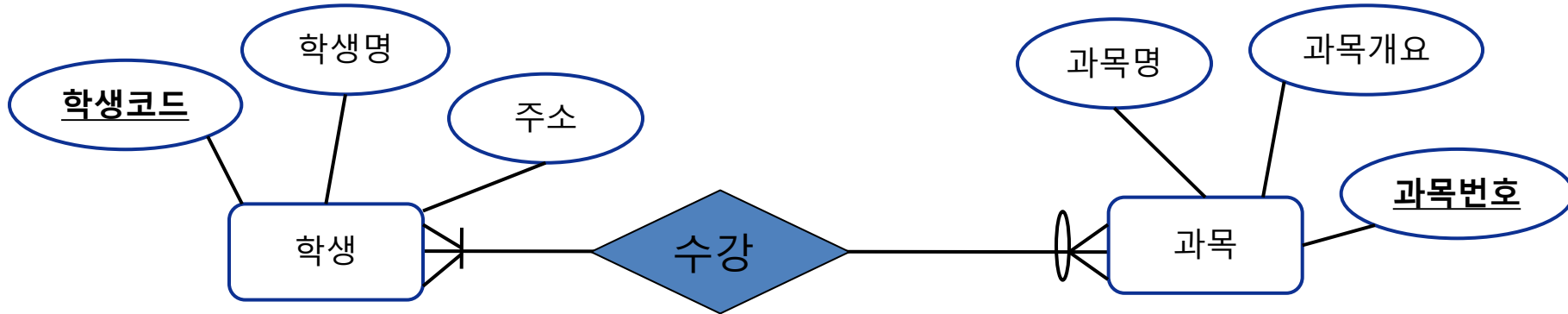
테이블명 : 부서

직원번호 (PK)	이름	주소	성별	입사일	부서코드 (FK)
1	이동국	전주	남	20120101	01
2	아이윙	나고야	여	20120101	02
3	차범균	서울	남	20120101	01

논리적 데이터베이스 모델링

➤ 다 대 다 (M :N) 해결

- M : N 관계는 Primary key 제약조건에 위배
- 관계를 이용해 테이블로 만들어 1 : N : 1 의 관계로 변환한다



테이블명 : 수강

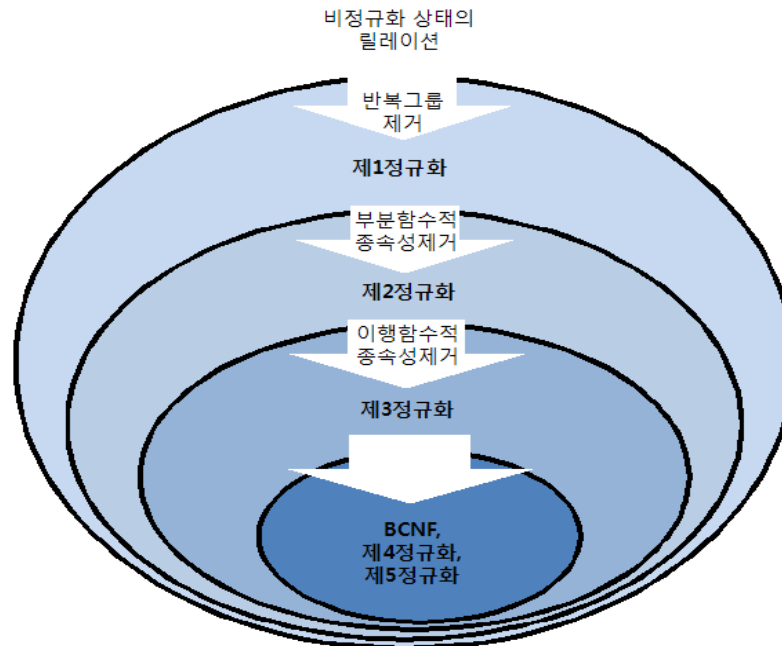
수강일시(PK)	학생코드(PK,FK)	과목번호(PK,FK)
20121201	Student1	Subject1
20121202	Student2	Subject1
20121203	Student2	Subject2

논리적 데이터베이스 모델링

➤ 정규화

- 애트리뷰트 간에 존재하는 함수적 종속성을 분석해서 관계형 스키마를 더 좋은 구조로 정제해 나가는 일련의 과정
- 데이터의 중복의 제거하고 속성들을 본래의 제자리에 위치시키는 것

- 제1 정규화
- 제2 정규화
- 제3 정규화



논리적 데이터베이스 모델링

➤ 정규화

1) 제1 정규화

- 반복되는 그룹 속성을 제거한 뒤 기본 테이블의 기본키를 추가해 새로운 테이블을 생성하고 기존의 테이블과 1:N 의 관계를 형성하는 과정
- 반복되는 그룹속성이란 같은 성격과 내용의 컬럼이 연속적으로 나타나는 컬럼을 말함

논리적 데이터베이스 모델링

➤ 정규화

- 정규화 적용 전 테이블

- 문제점 : 부수적인 데이터를 저장하기 위해 기본 데이터가 중복됨

컬럼명	주문 ID	주문일	회원 ID	회원명	회원등급	상품 ID	상품명	단위	수량	단가
키형태	PK									
견본 테이블	1	01월 13일	아이유	아이유	A	A1	커피	1	1	1000
						B1	도서	1	1	2000
						C1	음료	1	1	3000
	2	01월 15일	김범수	김범수	B	A1	커피	1	1	1000
						C1	음료	1	1	3000
	3	01월 16일	김혜수	김혜수	C	B1	도서	1	1	2000

Data 중복

논리적 데이터베이스 모델링

➤ 정규화

- 제 1 정규화에 의한 테이블 분리

테이블명 : 주문

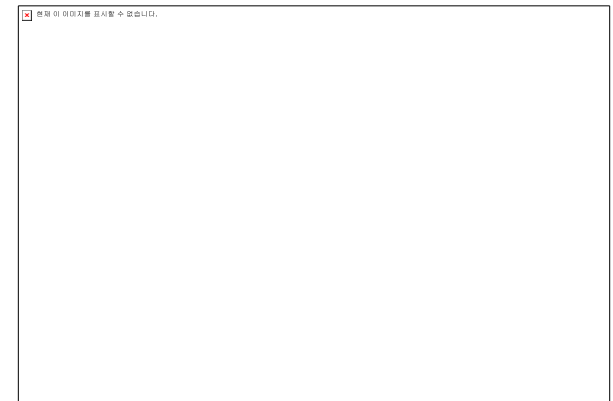
컬럼명	주문ID	주문일	회원 ID	회원명	회원등급
키형태	PK				
견본 데이터	1	01월 13일	아이유	아이유	A
	2	01월 15일	김범수	김범수	B
	3	01월 16일	아이유	아이유	C



테이블명 : 주문상세

컬럼명	주문 ID	상품 ID	상품명	단위	수량	단가
키형태	PK	PK				
견본 데이터	1	A1	커피	1	1	1000
	1	B1	도서	1	1	2000
	1	C1	음료	1	1	3000
	2	A1	커피	1	1	1000
	2	C1	음료	1	1	3000
	3	B1	도서	1	1	2000

제 1 정규화 적용 ER-Diagram



논리적 데이터베이스 모델링

➤ 정규화

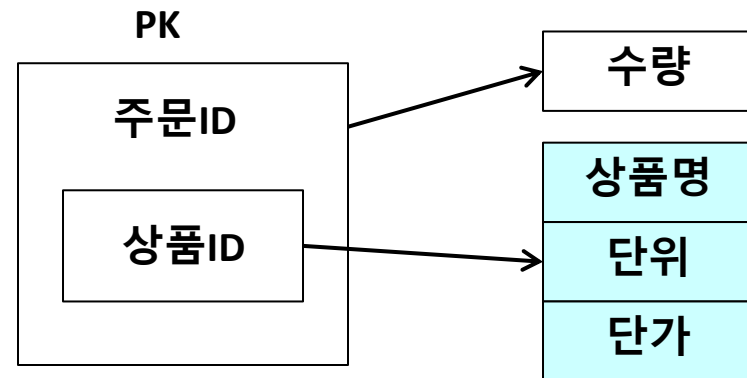
2) 제 2 정규화

- 복합키(Composite Primary Key)에 전체적으로 의존하지 않는 속성 제거
- 제 2 정규화의 대상이 되는 테이블은 키가 여러 컬럼으로 구성된 경우
- 복합키의 일부분에 의해 종속되는 것을 부분적 함수 종속관계라 하며
- 이를 제거하는 작업

복합키로 구성된 주문상세 테이블이
제 2정규화의 대상임

* 상품명, 단위, 단가 컬럼은 복합키
전체에 의존하지 않는 관계로
제 2 정규화의 대상이 됨

주문상세 테이블의 컬럼 의존관계



논리적 데이터베이스 모델링

정규화

- 제 2 정규화에 의한 테이블 분리

테이블명 상품

컬럼명	상품 ID	상품명	단위	단가
키형태	PK			
견본 데이터	A1	커피	1	1000
	B1	도서	1	2000
	C1	음료	1	3000

테이블명 주문상세

컬럼명	주문 ID	상품 ID	수량
키형태	PK	PK,FK	
견본 데이터	1	A1	1
	1	B1	1
	1	C1	1
	2	A1	1
	2	C1	1
	3	B1	1

제 2 정규화 적용 ER-Diagram



논리적 데이터베이스 모델링

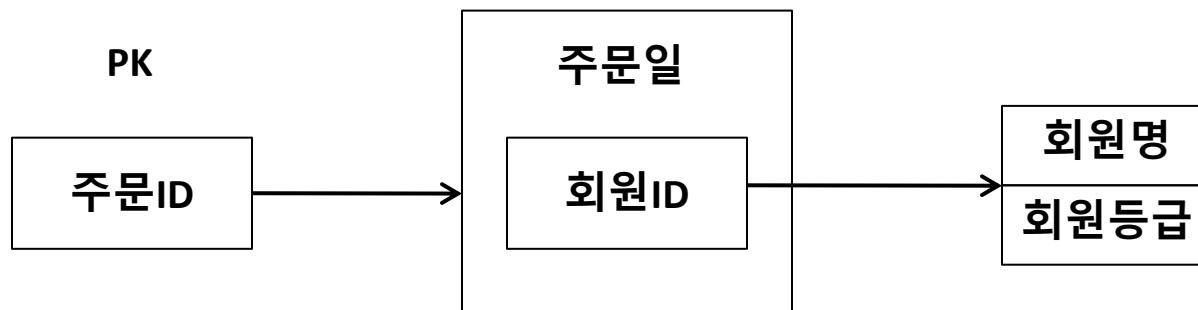
➤ 정규화

● 제 3 정규화

- 기본키에 의존하지 않고 일반 컬럼에 의존하는 컬럼들을 제거한다
- 이행적 함수 종속관계를 갖는 컬럼을 제거하는 과정

주문테이블에서 키가 아닌 모든 컬럼은 기본키인 주문 ID에 종속적인지 검증한다.

주문 테이블의 컬럼 의존관계



논리적 데이터베이스 모델링

- 정규화
 - 제 3 정규화에 의한 테이블 분리

테이블명 회원

컬럼명	회원 ID	회원명	회원등급
키형태	PK		
견본 데이터	아이유	아이유	A
	김범수	김범수	B
	아이유	아이유	C

테이블명 주문

컬럼명	주문ID	주문일	회원 ID
키형태	PK		FK
견본 데이터	1	01월 13일	아이유
	2	01월 15일	김범수
	3	01월 16일	아이유

제 3 정규화 적용 ER-Diagram



논리적 데이터베이스 모델링

- 정규화
 - 최종 정규화 결과 관계 스키마

테이블명 회원

컬럼명	회원 ID	회원명	회원등급
키형태	PK		

테이블명 주문

컬럼명	주문ID	주문일	회원 ID
키형태	PK		FK

테이블명 주문상세

컬럼명	주문 ID	상품 ID	수량
키형태	PK, FK	PK, FK	

테이블명 상품

컬럼명	상품 ID	상품명	단위	단가
키형태	PK			



세부목차

- 1 데이터 베이스 모델링 개요
- 2 개념적 데이터베이스 모델링
- 3 논리적 데이터베이스 모델링
- 4 물리적 데이터베이스 모델링**
- 5 데이터 베이스 모델링 실습

물리적 데이터베이스 모델링

➤ 물리적 데이터베이스 모델링 과정

- 논리적 데이터베이스 모델링 단계에서 얻어진 데이터베이스 스키마를 좀더 효율적으로 구현하기 위한 작업과, DBMS 특성에 맞게 실제 데이터베이스 내의 개체들을 정의하는 단계
 - Column의 domain 설정 (text, integer, real, BLOB, null)
- 데이터 사용량 분석과 업무 프로세스 분석을 통해서 보다 효율적인 데이터베이스가 될 수 있도록 효과적인 인덱스를 정의하고 상황에 따른 역정규화 작업을 수행함.
 - Index, Trigger, 역정규화

물리적 데이터베이스 모델링

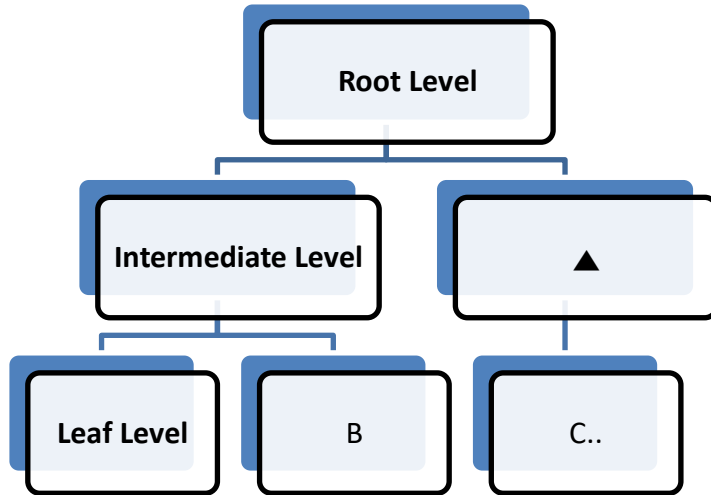
➤ 인덱스(Index)

➤ 개요

- 데이터베이스 내의 테이블에서 원하는 데이터를 좀더 빨리 찾아줄 수 있게끔 데이터의 위치정보를 모아 놓은 데이터베이스 내의 개체
- 정렬된 데이터의 키 값과 해당 키의 주소 정보를 이용

물리적 데이터베이스 모델링

➤ 인덱스(Index)



- 루트 레벨(Root Level)
인덱스 구조의 가장 상위 레벨로 하위 레벨의 인덱스에 대한 정보를 저장
- 중간 레벨(Intermediate Level)
루트레벨과 리프 레벨을 제외한 중간 영역으로 하위 레벨 인덱스에 대한 정보를 저장
- 리프 레벨(Leaf Level)
인덱스 구조의 가장 하위 부분으로
인덱스 키(인덱스가 설정된 컬럼) 값에 대한 데이터의 위치 정보를 저장
일반적으로 인덱스의 리프 레벨 밑에는 실제 데이터를 저장하고 있는 데이터 페이지가 위치한다.

물리적 데이터베이스 모델링

➤ 인덱스 유형

- 클러스터드 인덱스(Clustered Index)
 - 키가 아닌 특정 필드 값을 정렬해서 그 값을 탐색 키로 이용
 - 관련된 데이터가 일반적으로 동시에 함께 사용된다는 특징을 이용
 - 클러스터드 인덱스를 만들게 되면 기본적으로 그 열(Index Key)을
 - 기준으로 물리적으로 데이터를 정렬시킨다.
- 논 클러스터드 인덱스(Non Clustered Index)
 - 물리적으로 데이터베이스를 정렬시키지 않고 데이터 페이지에 있는 데이터들의 위치정보를 인덱스로 구성한다.
 - 물리적으로 데이터를 정렬시키지 않기 때문에 한 테이블에 여러 개의 인덱스 정의가 가능함

물리적 데이터베이스 모델링

➤ 데이터 검색 방법

- Table Scan (Full Scan)
- Index Scan

➤ 데이터 검색 유형

- 포인터 쿼리(Point Query)
 - 단일 건 조회
- 범위 조회(Range Query)
 - N 건 조회
- 커버드 조회(Covered Query)
 - 조회의 조건과 대상이 되는 컬럼이 모두 인덱스로 구성된 경우

물리적 데이터베이스 모델링

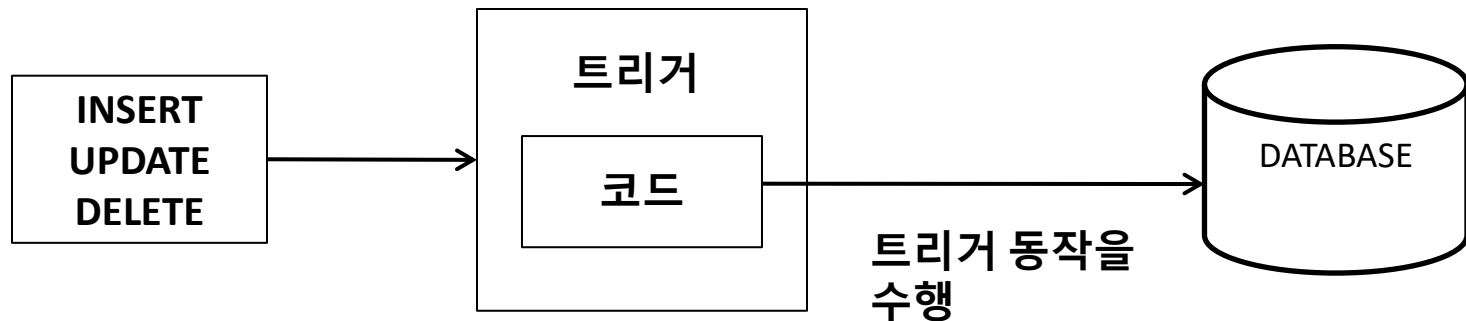
➤ 인덱스 적용 시 고려 사항

- 포린키(Foreign Key)에 인덱스를 부여하는 것을 적극 고려하자
- 두 개 이상의 컬럼으로 구성된 복합인덱스의 경우 컬럼의 순서를 고려하자
- 데이터의 사용량을 분석한다
- 한 테이블에 들어가는 컬럼의 수는 10개를 넘지 않도록 노력한다
 - 컬럼의 수가 많아 한 row가 차지하는 page 수가 늘어나면 index level이 높아져 좋지 않게 된다

트리거(Trigger)

➤ 정의

- ✓ 업무 규칙을 정의 하기 위한 데이터베이스 내의 개체이며 데이터의 무결성과 일관성을 정의하는 기능을 함
- ✓ 테이블의 데이터가 입력, 수정, 삭제되는 경우 자동 실행되어 짐



트리거(Trigger)

➤ 데이터의 입력, 수정, 삭제 방식

✓ 입력

- Insert data -> Inserted table-> 실제 테이블 -> Inserted table 삭제

✓ 삭제

- Delete data -> Deleted table -> 실제 테이블 -> Deleted table 삭제

✓ 수정

- 수정할 데이터 -> Inserted table -> 수정될 데이터 -> deleted table -> 실제 테이블

➤ Inserted Table 을 new, Deleted Table 을 old 라 한다

역정규화

➤ 역정규화(Denormalization) 정의

- 시스템 성능을 고려하여 기존 설계를 재구성하는 것
- 정규화에 위배되는 행위
- 테이블의 재구성

➤ 역정규화 방법

- 데이터 중복(컬럼 역정규화)
 - 조인 프로세스가 많아 시스템의 성능저하를 가져오는 경우, 조인 프로세스를 줄이기 위해 해당 컬럼을 중복함으로써 성능을 향상시키기 위한 방법
- 파생 컬럼의 생성
 - 기본적으로 테이블에 없는 컬럼을 숫자 연산이나 데이터 조작 등을 통해 조회할 때 새로운 정보를 보여주는 경우
 - ex) 판매테이블의 판매금액 컬럼이나 성적 테이블의 총점과 평균

역정규화

- 역정규화 방법
 - 테이블 분리
 - 컬럼 기준으로 분리(컬럼수)
 - 레코드 기준으로 분리(레코드 양)
 - 요약 테이블 생성(Summary Table)
 - 조회의 프로세스를 줄이기 위해 요약된 정보만을 저장하는 테이블을 만드는 것
 - 테이블 통합
 - 분리된 두 테이블이 시스템 성능에 영향을 끼칠 경우 고려

세부목차

- 1 데이터 베이스 모델링 개요
- 2 개념적 데이터베이스 모델링
- 3 논리적 데이터베이스 모델링
- 4 물리적 데이터베이스 모델링
- 5 데이터 베이스 모델링 실습**

데이터베이스 모델링 실습

➤ 학사관리 모델링 실습

1. 업무 분석표
2. 개념적 데이터베이스 모델링
3. 논리적 데이터베이스 모델링
4. 물리적 데이터베이스 모델링

데이터베이스 모델링 실습

➤ “사내 대학교 학사관리” 모델링 실습

1. 업무 분석표

- 자앤 주식회사 인재개발팀 얼씨구 사원은 매년 반복되는 서류 처리를 DB화 시키려고 한다.
- 얼씨구 사원이 전산화 하려고 하는 업무는 매년 사내교육이 끝날 때 어느 반의 어떤 과목이 규정된 수업시간과 맞는지를 비교하려는 것이다.
- 사내 대학 학년별로 해당 과목의 규정 수업시간은 정해져 있으며 연말에 과목별 실제 수업일수가 규정된 수업시간과 맞는지를 체크해서 부족한 경우에는 보강 등을 통해서 규정된 수업시간을 채워야 하므로 이 데이터는 정확해야 한다.
- 현재 사내 대학 학년별 각 반의 수업일수를 계산하기 위해서 각 반의 수업일지를 과목별로 일일이 수작업으로 확인하고 있다.

데이터베이스 모델링 실습

➤ “사내 대학교 학사관리” 모델링 실습

2. 개념적 데이터베이스 모델링

- 업무 분석
- 엔티티 도출
- 속성(Attribute) 도출
- 식별자 도출
- 관계 도출

[실 습]

개념적 데이터 베이스 모델링 단계에서 정의한 내용을 바탕으로
학사관리 ER-Diagram을 작성해 보자

데이터베이스 모델링 실습

➤ “사내 대학교 학사관리” 모델링 실습

3. 논리적 데이터베이스 모델링

- Mapping Rule을 이용한 데이터베이스 스키마 생성
 - 1단계 : 관계설정
 - 2단계 : 다 대 다(N:M) 관계 해소
- 정규화

[실 습]

논리적 데이터 베이스 모델링 단계에서 정의한 내용을 바탕으로
학사 관리 ER-Diagram을 작성해 보자

.

데이터베이스 모델링 실습

➤ “사내 대학교 학사관리” 모델링 실습

4. 물리적 데이터베이스 모델링

- 데이터 타입과 사이즈 정의
 - 컬럼의 데이터 타입 정의시 참고사항
 - . 데이터를 받아들일 수 있는 가장 작은 크기로 컬럼 사이즈 정의
 - . 계산을 필요로 하지 않는 숫자는 문자로 처리
- 역정규화
 - 데이터 사용량 및 프로세스 분석을 통한 역정규화 작업 수행

[실 습]

물리적 데이터 베이스 모델링 단계에서 정의한 내용을 바탕으로
학사 관리 ER-Diagram을 작성해 보자