

The background features a series of flowing, wavy green lines that create a sense of movement and depth. These lines are layered, with some appearing more prominent than others, and they curve across the frame. A solid dark green horizontal bar is positioned at the very bottom of the image.

Introduction to DBMS

데이터베이스 시스템의 구성

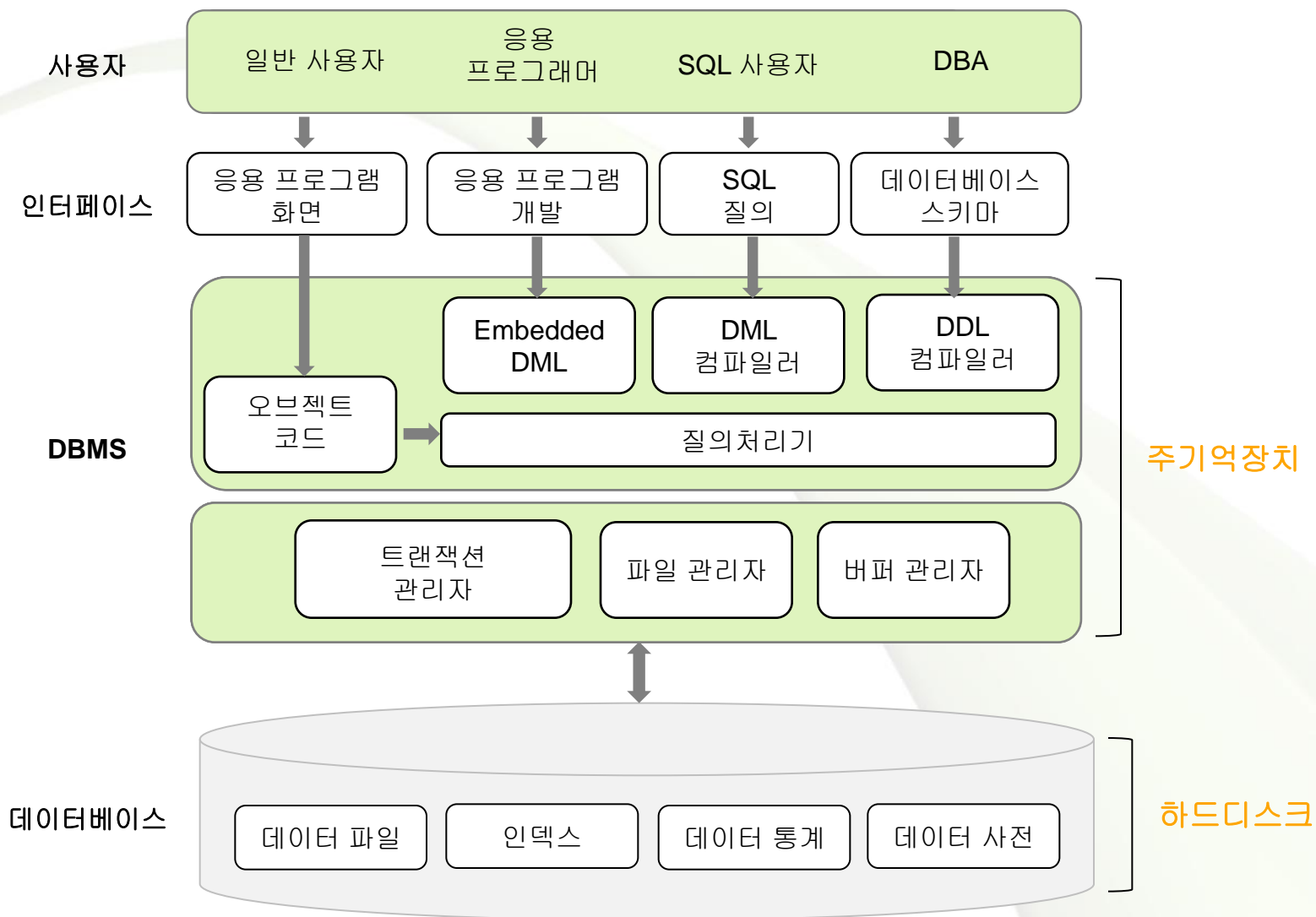


그림 1-22 데이터베이스 시스템의 구성

데이터베이스 언어

▪ SQL

- 데이터 정의를어(DDL, Data Definition Language)
- 데이터 조작어(DML, Data Manipulation Language)
- 데이터 제어어(DCL, Data Control Language)

질의 1-1 Book 테이블에서 모든 도서이름(bookname)과 출판사(publisher)를 검색하시오.

```
SELECT bookname, publisher  
FROM Book;
```

Book 테이블

bookid	bookname	publisher	price
1	축구의 역사	굿스포츠	7000
2	축구아는 여자	나무수	13000
3	축구의 이해	대한미디어	22000
4	골프 바이블	대한미디어	35000
5	피겨 교본	굿스포츠	8000

bookname	publisher
축구의 역사	굿스포츠
축구아는 여자	나무수
축구의 이해	대한미디어
골프 바이블	대한미디어
피겨 교본	굿스포츠

데이터베이스 언어

질의 1-2 가격(price)이 10,000원 이상인 도서이름(bookname)과 출판사(publisher)를 검색하시오.

```
SELECT bookname, publisher
FROM Book
Where price >= 10000;
```

Book 테이블

bookid	bookname	publisher	price
1	축구의 역사	굿스포츠	7000
2	축구아는 여자	나무수	13000
3	축구의 이해	대한미디어	22000
4	골프 바이블	대한미디어	35000
5	피겨 교본	굿스포츠	8000

bookname	publisher
축구아는 여자	나무수
축구의 이해	대한미디어
골프 바이블	대한미디어

데이터베이스 사용자

■ 일반사용자

- 은행의 창구 혹은 관공서의 민원 접수처 등에서 데이터를 다루는 업무를 하는 사람
- 프로그래머가 개발한 프로그램을 이용하여 데이터베이스에 접근 일반인

■ 응용프로그래머

- 일반 사용자가 사용할 수 있도록 프로그램을 만드는 사람
- 자바, C, JSP 등의 프로그래밍 언어와 SQL을 사용하여 일반 사용자를 위한 사용자 인터페이스와 데이터를 관리하는 응용 로직을 개발

■ SQL 사용자

- SQL을 사용하여 업무를 처리하는 IT 부서의 담당자
- 응용 프로그램으로 구현되어 있지 않은 업무를 SQL을 사용하여 처리

■ 데이터베이스 관리자(DBA, Database Administrator)

- 데이터베이스 운영 조직의 데이터베이스 시스템을 총괄하는 사람
- 데이터 설계, 구현, 유지보수의 전 과정을 담당
- 데이터베이스 사용자 통제, 보안, 성능 모니터링, 데이터 전체 파악 및 관리, 데이터 이동 및 복사 등 제반 업무를 함

데이터베이스 사용자

표 1-7 데이터베이스 사용자 별로 갖추어야 할 지식 수준(× : 없음, ○ : 보통, ◎ : 높음)

	SQL 언어	프로그래밍 능력	DBMS 지식	데이터 구성
일반 사용자	×	×	×	×
SQL 사용자	◎	×	○	○
응용 프로그래머	◎	◎	○	○
데이터베이스 관리자	◎	○	◎	◎

표 1-8 DBMS의 기능

데이터 정의(Definition)	데이터의 구조를 정의하고 데이터 구조에 대한 삭제 및 변경 기능을 수행함
데이터 조작(manipulation)	데이터를 조작하는 소프트웨어(응용 프로그램)가 요청하는 데이터의 삽입, 수정, 삭제 작업을 지원함
데이터 추출(Retrieval)	사용자가 조회하는 데이터 혹은 응용 프로그램의 데이터를 추출함
데이터 제어(Control)	데이터베이스 사용자를 생성하고 모니터링하며 접근을 제어함. 백업과 회복, 동시성 제어 등의 기능을 지원함



릴레이션

▪ 릴레이션(relation) : 행과 열로 구성된 테이블

표 2-1 릴레이션과 관련된 한글 용어

용어	한글 용어	비고
relation	릴레이션, 테이블	“관계”라고 하지 않음
relational data model	관계 데이터 모델	
relational database	관계 데이터베이스	
relational algebra	관계대수	
relationship	관계	

릴레이션

▪ 릴레이션이란?

도서 1, 축구의 역사, 굿스포츠, 7000

도서 2, 축구아는 여자, 나무수, 13000

도서 3, 축구의 이해, 대한미디어, 22000

도서 4, 골프 바이블, 대한미디어, 35000

도서 5, 피겨 교본, 굿스포츠, 8000

도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000
2	축구아는 여자	나무수	13000
3	축구의 이해	대한미디어	22000
4	골프 바이블	대한미디어	35000
5	피겨 교본	굿스포츠	8000

그림 2-1 데이터와 테이블(릴레이션)

도서번호 = {1,2,3,4,5}

도서이름 = {축구의 역사, 축구아는 여자, 축구의 이해, 골프 바이블, 피겨 교본}

출판사 = {굿스포츠, 나무수, 대한미디어}

가격 = {7000, 13000, 22000, 35000, 8000}

첫 번째 행(1, 축구의 역사, 굿스포츠, 7000)의 경우 네 개의 집합에서 각각 원소 한 개씩 선택하여 만들어진 것으로 이 원소들이 관계(relationship)를 맺고 있다.

릴레이션

▪ 관계(relationship)

- ① 릴레이션 내에서 생성되는 관계 : 릴레이션 내 데이터들의 관계
- ② 릴레이션 간에 생성되는 관계 : 릴레이션 간의 관계

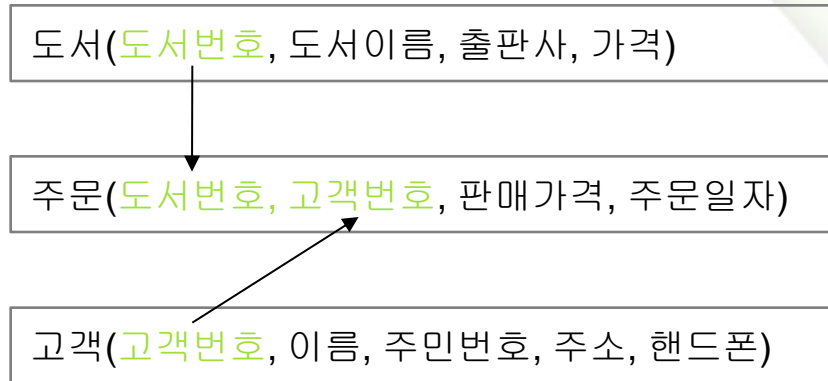


그림 2-2 릴레이션 간의 관계

릴레이션 스키마와 인스턴스

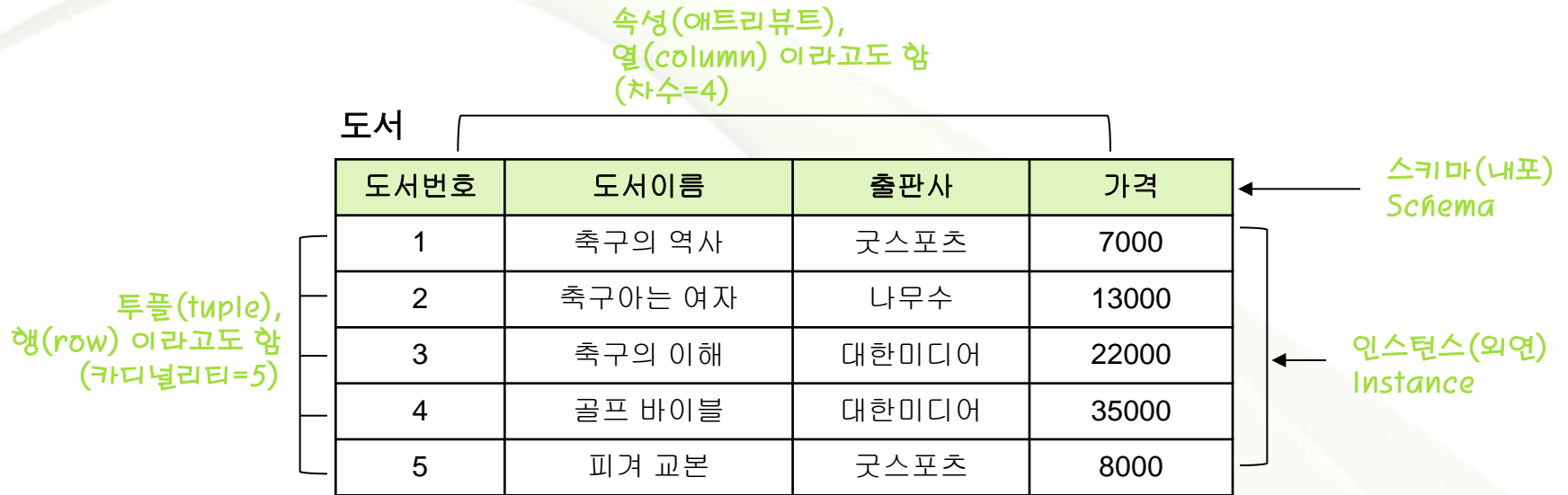


그림 2-3 도서 릴레이션

릴레이션 스키마

■ 스키마의 요소

- 속성(attribute) : 릴레이션 스키마의 열
- 도메인(domain) : 속성이 가질 수 있는 값의 집합
- 차수(degree) : 속성의 개수

■ 스키마의 표현

- 릴레이션 이름(속성1 : 도메인1, 속성2 : 도메인2, 속성3 : 도메인3 ...)
EX) 도서(도서번호, 도서이름, 출판사, 가격)

릴레이션 인스턴스

■ 인스턴스 요소

- 튜플(tuple) : 릴레이션의 행 → 튜플이 가지는 속성의 개수는 릴레이션 스키마의 차수와 동일하고, 릴레이션 내의 모든 튜플들은 서로 중복되지 않아야 함
- 카디널리티(cardinality) : 튜플의 수

표 2-2 릴레이션 구조와 관련된 용어

릴레이션 용어	같은 의미로 통용되는 용어	파일 시스템 용어
릴레이션(relation)	테이블(table)	파일(file)
스키마(schema)	내 포(intension)	헤더(header)
인스턴스(instance)	외 연(extension)	데이터(data)
튜플(tuple)	행(row)	레코드(record)
속성(attribute)	열(column)	필드(field)

릴레이션의 특징

- 속성은 단일 값을 가진다

각 속성의 값은 도메인에 정의된 값만을 가지며 그 값은 모두 단일 값이어야 함.

- 속성은 서로 다른 이름을 가진다

속성은 한 릴레이션에서 서로 다른 이름을 가져야만 함.

- 한 속성의 값은 모두 같은 도메인 값을 가진다

한 속성에 속한 열은 모두 그 속성에서 정의한 도메인 값만 가질 수 있음.

- 속성의 순서는 상관없다

속성의 순서가 달라도 릴레이션 스키마는 같음.

예) 릴레이션 스키마에서 (이름, 주소) 순으로 속성을 표시하거나 (주소, 이름) 순으로 표시하여도 상관 없음.

릴레이션의 특징

- 릴레이션 내의 중복된 튜플은 허용하지 않는다

하나의 릴레이션 인스턴스 내에서는 서로 중복된 값을 가질 수 없음. 즉 모든 튜플은 서로 값이 달라야 함.

- 튜플의 순서는 상관없다

튜플의 순서가 달라도 같은 릴레이션임. 관계 데이터 모델의 튜플은 실제적인 값을 가지고 있으며 이 값은 시간이 지남에 따라 데이터의 삭제, 수정, 삽입에 따라 순서가 바뀔 수 있음.

릴레이션의 특징

도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000
2	축구 아는 여자	나무수	13000
3	축구의 이해	대한미디어	22000
4	골프 바이블	대한미디어	35000
5	피겨 교본	굿스포츠	8000
5	피겨 교본	굿스포츠	8000
6	피겨 교본, 피겨 기초	굿스포츠	8000

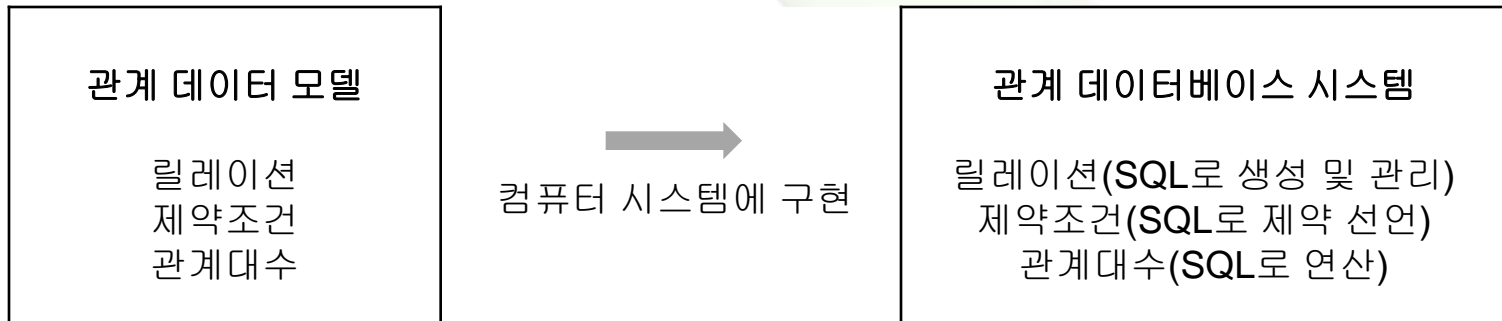
동일한 튜플이 중복되면 안 됨

속성의 값은 단일 값이어야 함

그림 2-4 릴레이션의 특징에 위배된 경우

관계 데이터 모델

- 관계 데이터 모델은 데이터를 2차원 테이블 형태인 릴레이션으로 표현함
- 릴레이션에 대한 제약조건(constraints)과 관계 연산을 위한 관계대수(relational algebra)를 정의함



키

- 특정 튜플을 식별할 때 사용하는 속성 혹은 속성의 집합임
- 릴레이션은 중복된 튜플을 허용하지 않기 때문에 각각의 튜플에 포함된 속성들 중 어느 하나(혹은 하나 이상)는 값이 달라야 함. 즉 키가 되는 속성(혹은 속성의 집합)은 반드시 값이 달라서 튜플들을 서로 구별할 수 있어야 함
- 키는 릴레이션 간의 관계를 맺는 데도 사용됨

고객

고객번호	이름	주민번호	주소	핸드폰
1	박지성	810101-1111111	영국 맨체스터	000-5000-0001
2	김연아	900101-2222222	대한민국 서울	000-6000-0001
3	장미란	830101-2333333	대한민국 강원도	000-7000-0001
4	추신수	820101-1444444	미국 클리블랜드	000-8000-0001

도서

도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000
2	축구아는 여자	나무수	13000
3	축구의 이해	대한미디어	22000
4	골프 바이블	대한미디어	35000
5	피겨 교본	굿스포츠	8000

주문

고객번호	도서번호	판매가격	주문일자
1	1	7000	2014-07-01
1	2	13000	2014-07-03
2	5	8000	2014-07-03
3	2	13000	2014-07-04
4	4	35000	2014-07-05
1	3	22000	2014-07-07
4	3	22000	2014-07-07

후보키

- 튜플을 유일하게 식별할 수 있는 속성의 최소 집합
(주문 릴레이션 예)
 - 고객번호 : 한 명의 고객이 여러 권의 도서를 구입할 수 있으므로 후보키가 될 수 없음. 고객번호가 1인 박지성 고객은 세 번의 주문 기록이 있으므로 튜플을 유일하게 식별할 수 없음
 - 도서번호 : 도서번호가 2인 '죽어가는 여자' 는 두 번의 주문 기록이 있으므로 튜플을 유일하게 식별할 수 없음
- 주문 릴레이션의 후보키는 2개의 속성을 합한 (고객번호, 도서번호)가 됨.
참고로 이렇게 2개 이상의 속성으로 이루어진 키를 복합키(composite key)라고 함

기본키

- 여러 후보키 중 하나를 선정하여 대표로 삼는 키
- 후보키가 하나뿐이라면 그 후보키를 기본키로 사용하면 되고 여러 개라면 릴레이션의 특성을 반영하여 하나를 선택하면 됨
- 기본키 선정 시 고려사항
 - 릴레이션 내 튜플을 식별할 수 있는 고유한 값을 가져야 함
 - NULL 값은 허용하지 않음
 - 키 값의 변동이 일어나지 않아야 함
 - 최대한 적은 수의 속성을 가진 것이어야 함
 - 향후 키를 사용하는 데 있어서 문제 발생 소지가 없어야 함
- 릴레이션 스키마를 표현할 때 기본키는 밑줄을 그어 표시함
릴레이션 이름(속성1, 속성2, ..., 속성N)
EX) 고객(고객번호, 이름, 주민번호, 주소, 핸드폰)
도서(도서번호, 도서이름, 출판사, 가격)

대리키

- 기본키가 보안을 요하거나, 여러 개의 속성으로 구성되어 복잡하거나, 마땅한 기본키가 없을 때는 일련번호 같은 가상의 속성을 만들어 기본키로 삼는 경우가 있음. 이러한 키를 대리키(surrogate key) 혹은 인조키(artificial key)라고 함
- 대리키는 DBMS나 관련 소프트웨어에서 임의로 생성하는 값으로 사용자가 직관적으로 그 값의 의미를 알 수 없음

주문

주문번호	고객번호	도서번호	판매가격	주문일자
1	1	1	7000	2014-07-01
2	1	2	13000	2014-07-03
3	2	5	8000	2014-07-03
4	3	2	13000	2014-07-04
5	4	4	35000	2014-07-05
6	1	3	22000	2014-07-07
7	4	3	22000	2014-07-07

그림 2-8 대리키를 사용하도록 변경된 주문 릴레이션

대체키

- 대체키(alternate key)는 기본키로 선정되지 않은 후보키를 말함
- 고객 릴레이션의 경우 고객번호와 주민번호 중 고객번호를 기본키로 정하면 주민번호가 대체키가 됨

외래키

- 다른 릴레이션의 기본키를 참조하는 속성을 말함. 다른 릴레이션의 기본키를 참조하여 관계 데이터 모델의 특징인 릴레이션 간의 관계(relationship)를 표현함
- 외래키의 특징
 - 관계 데이터 모델의 릴레이션 간의 관계를 표현함
 - 다른 릴레이션의 기본키를 참조하는 속성임
 - 참조하고(외래키) 참조되는(기본키) 양쪽 릴레이션의 도메인은 서로 같아야 함
 - 참조되는(기본키) 값이 변경되면 참조하는(외래키) 값도 변경됨
 - NULL 값과 중복 값 등이 허용됨
 - 자기 자신의 기본키를 참조하는 외래키도 가능함
 - 외래키가 기본키의 일부가 될 수 있음

외래키

고객

고객번호	이름	주민번호	주소	핸드폰
1	박지성	810101-1111111	영국 맨체스타	000-5000-0001
2	김연아	900101-2222222	대한민국 서울	000-6000-0001
3	장미란	830101-2333333	대한민국 강원도	000-7000-0001
4	추신수	820101-1444444	미국 클리블랜드	000-8000-0001

기본키

도서

도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000
2	축구아는 여자	나무수	13000
3	축구의 이해	대한미디어	22000
4	골프 바이블	대한미디어	35000
5	피겨 교본	굿스포츠	8000

기본키

참조

주문

주문번호	고객번호	도서번호	판매가격	주문일자
1	1	1	7000	2014-07-01
2	1	2	13000	2014-07-03
3	2	5	8000	2014-07-03
4	3	2	13000	2014-07-04
5	4	4	35000	2014-07-05
6	1	3	22000	2014-07-07
7	4	3	22000	2014-07-07

기본키

외래키

참조

그림 2-9 릴레이션 간의 참조 관계

외래키

- 외래키 사용 시 참조하는 릴레이션과 참조되는 릴레이션이 꼭 다른 릴레이션일 필요는 없음. 즉 자기 자신의 기본키를 참조할 수도 있음

선수번호	이름	주소	멘토번호
1	박지성	영국 맨체스타	NULL
2	김연아	대한민국 서울	3
3	장미란	대한민국 강원도	4
4	추신수	미국 클리블랜드	NULL

그림 2-10 멘토 릴레이션

키 - 내용 요약

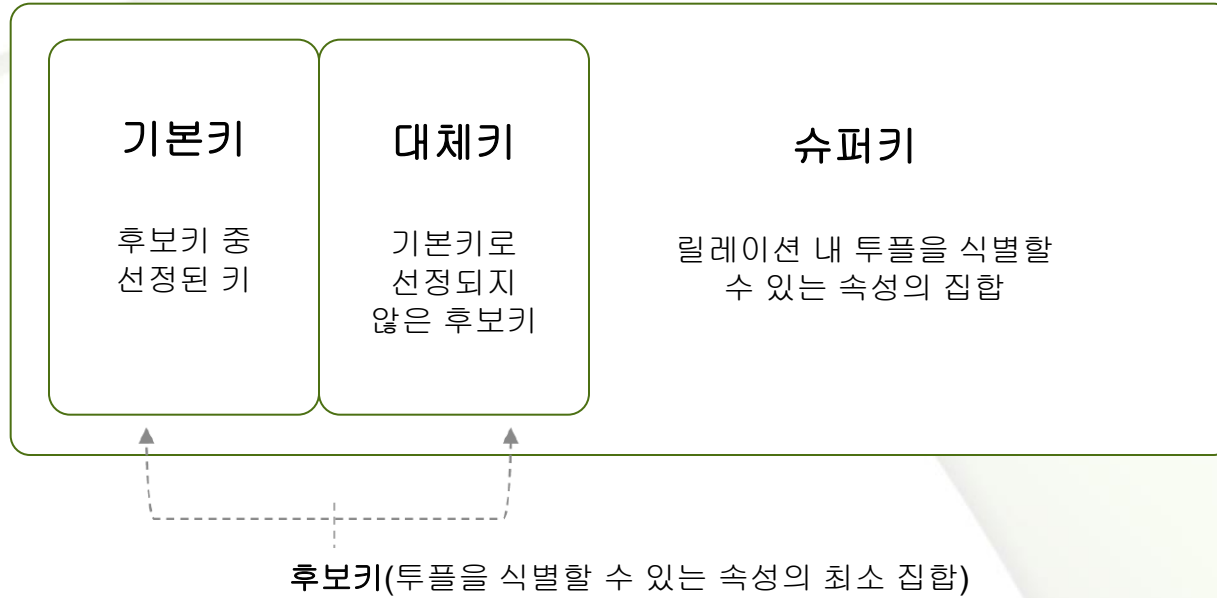


그림 2-11 키의 포함 관계

무결성 제약조건

- 데이터 무결성(integrity, 無缺性)은 데이터베이스에 저장된 데이터의 일관성과 정확성을 지키는 것을 말함

- 도메인 무결성 제약조건

도메인 제약(domain constraint)이라고도 하며, 릴레이션 내의 튜플들이 각 속성의 도메인에 지정된 값만을 가져야 한다는 조건임. SQL 문에서 데이터 형식(type), 널(null/not null), 기본 값(default), 체크(check) 등을 사용하여 지정할 수 있음

- 개체 무결성 제약조건

기본키 제약(primary key constraint)이라고도 함. 릴레이션은 기본키를 지정하고 그에 따른 무결성 원칙 즉, 기본키는 NULL 값을 가져서는 안 되며 릴레이션 내에 오직 하나의 값만 존재해야 한다는 조건임

- 참조 무결성 제약조건

외래키 제약(foreign key constraint)이라고도 하며, 릴레이션 간의 참조 관계를 선언하는 제약조건임. 자식 릴레이션의 외래키는 부모 릴레이션의 기본키와 도메인이 동일해야 하며, 자식 릴레이션의 값이 변경될 때 부모 릴레이션의 제약을 받는다는 것임

무결성 제약조건

표 2-3 제약조건의 정리

구분	도메인	키	
	도메인 무결성 제약조건	개체 무결성 제약조건	참조 무결성 제약조건
제약 대상	속성	투플	속성과 투플
같은 용어	도메인 제약 (Domain Constraint)	기본키 제약 (Primary Key Constraint)	외래키 제약 (Foreign Key Constraint)
해당되는 키	-	기본키	외래키
NULL 값 허용 여부	허용	불가	허용
릴레이션 내 제약조건의 개수	속성의 개수와 동일	1개	0~여러 개
기타	<ul style="list-style-type: none"> 투플 삽입, 수정 시 제약 사항 우선 확인 	<ul style="list-style-type: none"> 투플 삽입/수정 시 제약 사항 우선 확인 	<ul style="list-style-type: none"> 투플 삽입/수정 시 제약사항 우선 확인 부모 릴레이션의 투플 수정/삭제 시 제약사항 우선 확인

개체 무결성 제약조건

- 삽입 : 기본키 값이 같으면 삽입이 금지됨
- 수정 : 기본키 값이 같거나 NULL로도 수정이 금지됨
- 삭제 : 특별한 확인이 필요하지 않으며 즉시 수행함

학번	이름	학과코드
501	박지성	1001
401	김연아	2001
402	장미란	2001
502	추신수	1001

그림 2-12 학생 릴레이션

(501, 남슬찬, 1001)



삽입 거부

학번	이름	학과코드
501	박지성	1001
401	김연아	2001
402	장미란	2001
502	추신수	1001

(NULL, 남슬찬, 1001)



삽입 거부

학번	이름	학과코드
501	박지성	1001
401	김연아	2001
402	장미란	2001
502	추신수	1001

그림 2-13 개체 무결성 제약조건 수행 예(기본키 충돌 및 NULL 값 삽입)

참조 무결성 제약조건

▪ 삽입

- 학과(부모 릴레이션) : 튜플 삽입한 후 수행하면 정상적으로 진행된다.
- 학생(자식 릴레이션) : 참조받는 테이블에 외래키 값이 없으므로 삽입이 금지된다.

학생(자식 릴레이션)

학번	이름	학과코드
501	박지성	1001
401	김연아	2001
402	장미란	2001
502	추신수	1001

학과(부모 릴레이션)

학과코드	학과명
1001	컴퓨터학과
2001	체육학과

참조

그림 2-14 학생 관리 데이터베이스

참조 무결성 제약조건

■ 삭제

- 학과(부모 릴레이션) : 참조하는 테이블을 같이 삭제할 수 있어서 금지하거나 다른 추가 작업이 필요함
- 학생(자식 릴레이션) : 바로 삭제 가능함

※ 부모 릴레이션에서 튜플을 삭제할 경우 참조 무결성 조건을 수행하기 위한 고려사항

- ① 즉시 작업을 중지
- ② 자식 릴레이션의 관련 튜플을 삭제
- ③ 초기에 설정된 다른 어떤 값으로 변경
- ④ NULL 값으로 설정

■ 수정

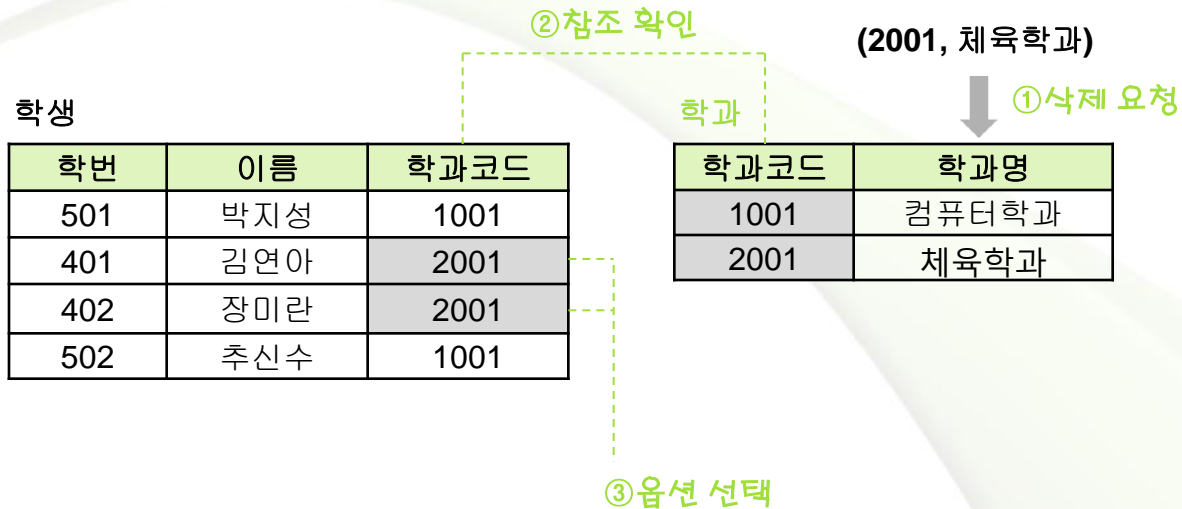
- 삭제와 삽입 명령이 연속해서 수행됨
- 부모 릴레이션의 수정이 일어날 경우 삭제 옵션에 따라 처리된 후 문제가 없으면 다시 삽입 제약조건에 따라 처리됨

참조 무결성 제약조건

표 2-4 참조 무결성 제약조건의 옵션(부모 릴레이션에서 튜플을 삭제할 경우)

명령어	의미	예
RESTRICTED	자식 릴레이션에서 참조하고 있을 경우 부모 릴레이션의 삭제 작업을 거부함	학과 릴레이션의 튜플 삭제 거부
CASCADE	자식 릴레이션의 관련 튜플을 같이 삭제 처리함	학생 릴레이션의 관련 튜플을 삭제
DEFAULT	자식 릴레이션의 관련 튜플을 미리 설정해둔 값으로 변경함	학생 릴레이션의 학과가 다른 학과로 자동 배정
NULL	자식 릴레이션의 관련 튜플을 NULL 값으로 설정함(NULL 값을 허가한 경우)	학과 릴레이션의 학과가 NULL 값으로 변경

참조 무결성 제약조건



- ① RESTRICTED : 요청한 삭제 작업중지(에러 처리)
- ② CASCADE : 학생 릴레이션의 해당 튜플을 같이 연쇄적으로 삭제(CASCADE)
- ③ 기본값으로 변경(미리 설정한 값, DEFAULT)
- ④ NULL 값으로 설정

그림 2-15 참조 무결성 제약조건에서 부모 릴레이션의 튜플을 삭제할 경우