

2.3 릴레이션의 키

□ 릴레이션의 키

- ✓ 각 튜플을 고유하게 식별할 수 있는 하나 이상의 애트리뷰트들의 모임
- ✓ 수퍼 키(superkey), 후보 키(candidate key), 기본 키(primary key), 대체 키(alternate key), 외래 키(foreign key)

□ 수퍼 키

- ✓ 한 릴레이션 내의 특정 튜플을 고유하게 식별하는 하나의 애트리뷰트 또는 애트리뷰트들의 집합
- ✓ 예: 신용카드 회사의 고객 릴레이션에서 (신용카드번호, 주소) 또는 (주민등록번호, 이름) 또는 (주민등록번호)
- ✓ 튜플들을 고유하게 식별하는데 꼭 필요하지 않은 애트리뷰트들을 포함할 수 있음

2.3 릴레이션의 키(계속)

□ 후보 키

- ✓ 각 튜플을 고유하게 식별하는 최소한의 애트리뷰트들의 모임

예: (신용카드번호, 주소)는 신용카드 회사의 고객 릴레이션의
후보 키가 아니지만 (신용카드번호)는 후보 키

- ✓ 모든 릴레이션에는 최소한 한 개 이상의 후보 키가 있음
- ✓ 후보 키도 두 개 이상의 애트리뷰트로 이루어질 수 있으며 이런 경우에

복합 키(composite key)라고 부름

예: (학번, 과목번호)가 후보 키

수강

학번	과목번호	학점
11002	CS310	A0
11002	CS313	B+
24036	CS345	B0
24036	CS310	A+

[그림 2.9] 수강 릴레이션

2.3 릴레이션의 키(계속)

그림 2.10의 학생 릴레이션에서 이름이 후보 키가 될 수 있는가?

그림 2.10의 학생 릴레이션에서 이메일이 후보 키가 될 수 있는가?

학생	학번	이름	이메일
	11002	이홍근	sea@hanmail.net
	24036	김순미	smkim@venus.uos.ac.kr
	13427	박상웅	blue@hanmir.com

[그림 2.10] 학생 릴레이션

2.3 릴레이션의 키(계속)

□ 기본 키

- ✓ 한 릴레이션에 후보 키가 두 개 이상 있으면 설계자 또는 데이터베이스 관리자가 이들 중에서 하나를 기본 키로 선정함

예: 신용카드 회사의 고객 릴레이션에서 신용카드번호와 주민등록번호가

후보 키가 될 수 있음. 이 중에서 신용카드 번호를 기본 키로 선정

- ✓ 자연스러운 기본 키를 찾을 수 없는 경우에는 레코드 번호와 같이 종종 인위적인 키 애트리뷰트를 릴레이션에 추가할 수 있음

2.3 릴레이션의 키(계속)

□ 대체 키

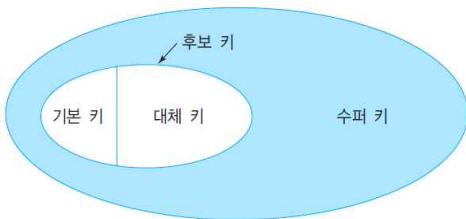
- ✓ 기본 키가 아닌 후보 키

예: 신용카드 회사의 고객 릴레이션에서 신용카드번호를 기본 키로
선정하면 주민등록번호는 대체 키

□ 외래 키

- ✓ 어떤 릴레이션의 기본 키를 참조하는 애트리뷰트
- ✓ 관계 데이터베이스에서 릴레이션들 간의 관계를 나타내기 위해서 사용됨
- ✓ 외래 키 애트리뷰트는 참조되는 릴레이션의 기본 키와 동일한 도메인을 가져야 함
- ✓ 자신이 속한 릴레이션의 기본 키의 구성요소가 되거나 되지 않을 수 있음

2.3 릴레이션의 키(계속)

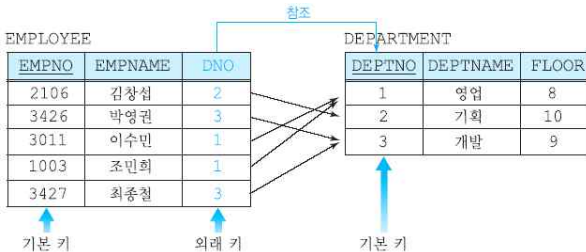


[그림 2.11] 키들의 포함 관계

2.3 릴레이션의 키(계속)

□ 외래 키의 유형

- ✓ 다른 릴레이션의 기본 키를 참조하는 외래 키




[그림 2.12] 다른 릴레이션을 참조하는 외래 키

2.3 릴레이션의 키(계속)

□ 외래 키의 유형(계속)

- ✓ 자체 릴레이션의 기본 키를 참조하는 외래 키

참조



<u>EMPNO</u>	EMPNAME	MANAGER	DNO
2106	김창섭	3426	2
3426	박영권	3011	3
3011	이수민	∧	1
1003	조민희	3011	1
3427	최종철	2106	3

[그림 2.13] 자체 릴레이션을 참조하는 외래 키

2.3 릴레이션의 키(계속)

□ 외래 키의 유형(계속)

✓ 기본 키의 구성요소가 되는 외래 키



[그림 2.14] 기본 키의 구성요소가 되는 외래 키

2.4 무결성 제약조건

❑ 데이터 무결성(data integrity)

- ✓ 데이터의 정확성 또는 유효성을 의미
- ✓ 일관된 데이터베이스 상태를 정의하는 규칙들을 묵시적으로 또는 명시적으로 정의함
- ✓ 데이터베이스가 갱신될 때 DBMS가 자동적으로 일관성 조건을 검사하므로 응용 프로그램들은 일관성 조건을 검사할 필요가 없음

2.4 무결성 제약조건(계속)

❑ 도메인 제약조건(domain constraint)

- ✓ 각 애트리뷰트 값이 반드시 원자값이어야 함
- ✓ 애트리뷰트 값의 디폴트 값, 가능한 값들의 범위 등을 지정할 수 있음
- ✓ 데이터 형식을 통해 값들의 유형을 제한하고, CHECK 제약 조건을 통해 값들의 범위를 제한할 수 있음
- ✓ SQL2는 도메인을 명시적으로 정의하는 것을 허용하지만, 오라클은 지원하지 않음

❑ 키 제약조건(key constraint)

- ✓ 키 애트리뷰트에 중복된 값이 존재해서는 안됨

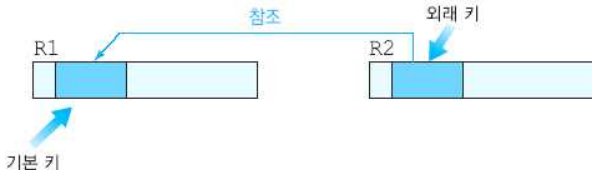
2.4 무결성 제약조건(계속)

- 기본 키와 엔티티 무결성 제약조건(entity integrity constraint)
 - ✓ 릴레이션의 기본 키를 구성하는 어떤 애트리뷰트도 널값을 가질 수 없음
 - ✓ 대체 키에는 적용되지 않음
 - ✓ 사용자는 릴레이션을 생성하는 데이터 정의문에서 어떤 애트리뷰트가 릴레이션의 기본 키의 구성요소인가를 DBMS에게 알려줌

2.4 무결성 제약조건(계속)

- 외래 키와 **참조 무결성 제약조건**(referential integrity constraint)
 - ✓ 참조 무결성 제약조건은 두 릴레이션의 연관된 튜플들 사이의 일관성을 유지하는데 사용됨
 - ✓ 관계 데이터베이스가 릴레이션들로만 이루어지고, 릴레이션 사이의 관계들이 다른 릴레이션의 기본 키를 참조하는 것을 기반으로 하여 묵시적으로 표현되기 때문에 외래 키의 개념이 중요
 - ✓ 릴레이션 R2의 외래 키가 릴레이션 R1의 기본 키를 참조할 때 참조 무결성 제약조건은 아래의 두 조건 중 하나가 성립되면 만족됨
 - 외래 키의 값은 R1의 어떤 튜플의 기본 키 값과 같다
 - 외래 키가 자신을 포함하고 있는 릴레이션의 기본 키를 구성하고 있지 않으면 널값을 가진다

2.4 무결성 제약조건(계속)



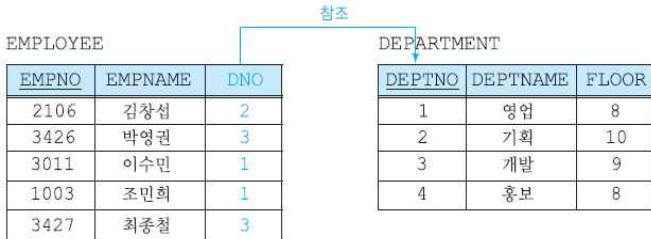
[그림 2.16] 참조 무결성 제약조건

2.4 무결성 제약조건(계속)

□ 무결성 제약조건의 유지

- ✓ 데이터베이스에 대한 갱신 연산은 삽입 연산, 삭제 연산, 수정 연산으로 구분함
- ✓ DBMS는 각각의 갱신 연산에 대하여 데이터베이스가 무결성 제약조건들을 만족하도록 필요한 조치를 취함
- ✓ DBMS는 외래 키가 갱신되거나, 참조된 기본 키가 갱신되었을 때 참조 무결성 제약조건이 위배되지 않도록 해야 함
- ✓ EMPLOYEE 릴레이션의 DNO 애트리뷰트가 DEPARTMENT 릴레이션의 기본 키인 DEPTNO를 참조하는 외래 키이므로, DEPARTMENT를 **참조된 릴레이션**, EMPLOYEE를 **참조하는 릴레이션**으로 부르기로 함

2.4 무결성 제약조건(계속)



[그림 2.17] 관계 데이터베이스 인스턴스

2.4 무결성 제약조건(계속)

□ 삽입

- ✓ 참조되는 릴레이션에 새로운 튜플이 삽입되면 참조 무결성 제약조건은 위배되지 않음
- ✓ DEPARTMENT에 새로 삽입되는 튜플의 기본 키 애트리뷰트의 값에 따라서는 도메인 제약조건, 키 제약조건, 엔티티 무결성 제약조건 등을 위배할 수 있음
- ✓ 참조하는 릴레이션에 새로운 튜플을 삽입할 때는 도메인 제약조건, 키 제약조건, 엔티티 무결성 제약조건 외에 참조 무결성 제약조건도 위배할 수 있음
 - 예: EMPLOYEE 릴레이션에 (4325, 오혜원, 6)이라는 튜플을 삽입하면 참조 무결성 제약조건을 위배하게 됨

2.4 무결성 제약조건(계속)

❑ 삭제

- ✓ 참조하는 릴레이션에서 튜플이 삭제되면 도메인 제약조건, 키 제약조건, 엔티티 무결성 제약조건, 참조 무결성 제약조건 등 모든 제약조건을 위배하지 않음
- ✓ 참조되는 릴레이션에서 튜플이 삭제되면 참조 무결성 제약조건을 위배하는 경우가 생기거나 생기지 않을 수 있음
 - 예1: DEPARTMENT 릴레이션에서 네 번째 튜플인 (4, 홍보, 8)을 삭제하더라도 참조 무결성 제약조건을 위배하지 않음
 - 예2: DEPARTMENT 릴레이션에서 세 번째 튜플인 (3, 개발, 9)를 삭제하면 참조 무결성 제약조건을 위배하게 됨

2.4 무결성 제약조건(계속)

- ❑ 참조 무결성 제약조건을 만족시키기 위해서 DBMS가 제공하는 옵션
 - ✓ 제한(restricted)
 - 위배를 야기한 연산을 단순히 거절
 - 예: DEPARTMENT 릴레이션에서 (3, 개발, 9)를 삭제하면 참조 무결성 제약조건을 위배하게 되므로 삭제 연산을 거절
 - ✓ 연쇄(cascade)
 - 참조되는 릴레이션에서 투표를 삭제하고, 참조하는 릴레이션에서 이 투표를 참조하는 투표들도 함께 삭제
 - 예: DEPARTMENT 릴레이션에서 (3, 개발, 9)를 삭제하면 EMPLOYEE 릴레이션에서 부서번호 3을 참조하는 두 번째 투표와 다섯 번째 투표도 함께 삭제

2.4 무결성 제약조건(계속)



[그림 2.18] 연쇄 삭제

2.4 무결성 제약조건(계속)

❑ 참조 무결성 제약조건을 만족시키기 위해서 DBMS가 제공하는 옵션(계속)

✓ 널값(nullify)

- 참조되는 릴레이션에서 튜플을 삭제하고, 참조하는 릴레이션에서 이 튜플을 참조하는 튜플들의 외래 키에 널값을 삽입
- 예: DEPARTMENT 릴레이션에서 (3, 개발, 9)를 삭제하면 EMPLOYEE 릴레이션에서 부서번호 3을 참조하는 두 번째 튜플과 다섯 번째 튜플의 부서번호에 널값을 삽입

✓ 디폴트값

- 널값을 넣는 대신에 디폴트값을 넣는다는 것을 제외하고는 바로 위의 옵션과 비슷함

2.4 무결성 제약조건(계속)

□ 수정

- ✓ DBMS는 수정하는 애트리뷰트가 기본 키인지 외래 키인지 검사함
- ✓ 수정하려는 애트리뷰트가 기본 키도 아니고 외래 키도 아니면 수정 연산이 참조 무결성 제약조건을 위배하지 않음
- ✓ 기본 키나 외래 키를 수정하는 것은 하나의 튜플을 삭제하고 새로운 튜플을 그 자리에 삽입하는 것과 유사하므로, 삽입 및 삭제에서 설명한 제한, 연쇄, 널값, 디폴트값 규칙이 수정 연산에도 적용됨
- ✓ 오라클에서는 수정 연산에 대해 제한적으로 참조 무결성 제약조건을 유지