5. 관계 데이터 모델

관계 데이터 모델의 개념

- 관계 데이터 모델의 기본 용어
  - 일반적으로 관계 데이터 모델에서는 하나의 개체에 관한 데이터를 릴레이션(relation)하나에 담아 데이터베이스에 저장한다
  - 릴레이션은 SQL의 테이블과 같다

그림 5-1 릴레이션 예 : 고객 릴레이션

  - 인터넷 쇼핑몰을 위한 데이터베이스에서 고객 개체를 표현한 고객 릴레이션의 예
- 릴레이션과 관련된 용어
  - 속성
    - 릴레이션의 열을 속성 또는 애트리뷰트(attribute)라고 부른다
    - 릴레이션은 파일 관리 시스템의 파일, 속성은 해당 파일의 필드에 대응하는 개념
  - 튜플
    - 릴레이션의 행을 튜플(tuple)이라 부른다
    - 개체의 인스턴스
      - 고객 4명의 데이터를 저장하고 있는 고객 릴레이션에는 4개의 튜플 또는 4개의 고객 개체 인스턴스가 존재
    - 파일 관리시스템에서 해당 파일의 레코드에 대응하는 개념
  - 도메인
    - 속성 하나가 가질 수 있는 모든 값의 집합을 해당 속성의 도메인(domain)이라한다
    - 관계 데이터 모델에서는 더는 분해할 수 없는 원자 값만 속성 값으로 사용할 수 있다
      - 그래서 도메인을 특정 속성이 가질 수 있는 모든 원자 값의 모임이라고도 정의한다

        - 고객 릴레이션에서 등급 속성의 값으로 vip, gold, silver, bronze 중 하나만 허용된다면, 네 가지 값을 모아놓은 것이 등급 속성의 도메인
            - 등급 속성의 도메인을 미리 정의해두면 사용자가 속성 값을 입력하거나 수정할 때 데이터베이스 시스템이 적합성을 판단하여 네 가지 이외의 값은 허용하지 않음으로써 항상 올바른 값만 유지할 수 있다는 장점이 있다
        - 하지만 나이, 이름 처럼 도메인을 정확히 정의하기 어려운 경우가 있다
            - 따라서 일반족으로 속성의 특성을 고려한 데이터 타입으로 도메인을 정의한다
        - 즉 프로그래밍 언어에서 도메인 -> 타입, 속성 -> 변수 로 이해하면 쉽다
    - 널 값
        - 릴레이션에 있는 특정 튜플의 속성 값을 모르거나, 적합한 값이 없는 경우에는 널이라는 특별한 값을 사용할 수 있다
        - 널 값은 값 자체가 없다는 뜻으로 0이나 공백문자와는 다르다
    - 차수
        - 하나의 릴레이션에서 속성의 전체 개수를 릴레이션의 차수(degree)라고 한다
            - 고객 릴레이션의 차수는 6이다
        - 모든 릴레이션은 최소 1이상의 차수를 유지해야 한다
        - 릴레이션의 차수는 일반적으로 자주 변하지 않는다는 정적인 특징이 있다
    - 카디널리티
        - 하나의 릴레이션에서 튜플의 전체 개수를 릴레이션의 카디널리티(cardinality)라고 한다
            - 고객 릴레이션의 카디널리티는 4이다
        - 튜플이 없는 릴레이션이 존재할 수도 있다
        - 새로운 튜플이 계속 삽입되거나 삭제될 수 있으므로 릴레이션의 카디널리티는 일반적으로 자주 변하는 동적인 특성이 있다
- 릴레이션과 데이터베이스의 구성
    - 관계 데이터 모델에서 릴레이션은 릴레이션 스키마와 릴레이션 인스턴스로 구성되어 있다

고객아이디	고객이름	나이	등급	직업	적립금
apple	김현준	20	gold	학생	1000
banana	정소화	25	vip	간호사	2500
carrot	원유선	28	gold	교사	4500
orange	정지영	22	silver	학생	0

릴레이션 스키마

릴레이션 인스턴스

그림 5-2 릴레이션 구성 예 : 고객 릴레이션

    - 릴레이션 스키마
        - 릴레이션 스키마(relation schema)는 릴레이션의 이름과 릴레이션에 포함된 모든 속성의 이름으로 정의하는 릴레이션의 논리적 구조다

        - 릴레이션 스키마는 데이터베이스 관리 시스템이 내부적으로 데이터 정의어를 이용해 정으히 ㅏ지만, 일반적으로는 다음과 같은 형태로 쉽게 표현한다
            - 릴레이션이름(속성이름1, 속성이름2, ... , 속성이름n)
        - 고객 릴레이션에서 릴레이션 스키마는 고객(고객아이디, 고객이름, 나이, 등급, 직업, 적립금)이다
        - 릴레이션 스키마를 보면 릴레이션의 이름이 무엇이고, 어떤 속성들로 구성되어 있는지 전체 구조를 쉽게 파악할 수 있다
        - 릴레이션 스키마는 릴레이션 내포(relation intension)라고도 부른다
    - 릴레이션 인스턴스
        - 릴레이션 인스턴스(relation instance)는 어느 한 시점에 릴레이션에 존재하는 튜플들의 집합이다
        - 릴레이션 인스턴스에 포함된 튜플은 릴레이션 스키마에서 정의하는 각 속성에 대응하는 실제 값으로 구성되어 있다
            - 고객 릴레이션에서는 4개의 튜플로 구성된 릴레이션 인스턴스가 있다
        - 데이터베이스 관리 시스템이 내부적으로는 데이터 조작어를 이용해 릴레이션 인스턴스의 튜플을 검색하거나, 새로운 튜플 삽입과 기존 튜플 삭제 및 수정을 수행한다
        - 릴레이션 인스턴스는 간단히 릴레이션이라 부르리도 하고 릴레이션 외연(relation extension)이라고도 부른다
        - 논리적 구조를 정의하는 릴레이션 스키마는 자주 변하지 않는다는 정적인 특징이 있는 반면, 릴레이션 인스턴스는 튜플의 삽입, 삭제, 수정이 자주 발생한다는 동적인 특징이 있다
    - 데이터베이스 스키마와 데이터베이스 인스턴스
        - 일반적으로 데이터베이스는 릴레이션 여러 개로 구성된다
        - 데이터베이스의 전체 구조를 의미하는 데이터베이스 스키마는 데이터베이스를 구성하는 릴레이션들의 스키마를 모아놓은 것이다
            - 즉, 특정 데이터베이스 스키마를 설계한다는 것은 필요한 모든 릴레이션의 스키마를 모두 정의한다는 뜻이다
        - 데이터베이스 인스턴스는 어느 한 시점에서 데이터베이스에 저장된 데이터 내용의 전체 집합을 의미한다
          - 즉, 데이터베이스를 구성하는 모든 릴레이션의 인스턴스를 모아놓은 것이다

그림 5-3 데이터베이스 구성 예 : 인터넷 쇼핑몰 데이터베이스

- 릴레이션의 특성

- 튜플의 유일성 : 하나의 릴레이션에는 동일한 튜플이 존재할 수 없다
  - 하나의 릴레이션에 똑같은 튜플이 있으면 안되고, 모든 튜플에는 다른 튜플과 구별되는 유일한 특성이 있어야 한다
    - 관계 데이터 모델의 릴레이션에서는 하나 또는 여러개의 속성을 미리 선정해두고 이 속성 값을 튜플마다 다르게 지정하여 튜플의 유일성을 판단한다
    - 이처럼 튜플을 유일하게 구별하기위해 선정하는 속성(또는 속성들의 모임)을 키(key)라고 부른다
    - 키를 이용해 튜플의 유일성이 만족되면 릴레이션에서 원하는 튜플에 쉽게 접근할 수 있다
  - 집합(Set)을 생각해보자
- 튜플의 무순서 : 하나의 릴레이션에서 튜플 사이의 순서는 무의미하다
  - 튜플의 순서가 바뀐다고 다른 릴레이션이 될 수 없고, 순서와 상관없이 튜플 내용이 같아야 같은 릴레이션이다
  - 데이터베이스는 위치가 아닌 내용으로 검색되므로 튜플의 순서는 중요하지 않다
  - 집합(Set)을 생각해보자
- 속성의 무순서 : 하나의 릴레이션에서 속성 사이의 순서는 무의미하다
  - 속성은 순서가 바뀌어도 다른 릴레이션이 될 수 없고, 순서와 상관없이 같은 속성들로 구성되어 있어야 같은 릴레이션
    - 고객(나이, 이름) == 고객(이름, 나이)
      - 두 스키마는 같은 스키마
  - 속성 값은 릴레이션에서 위치가 아닌 속성의 이름으로 접근하므로 하나의 릴레이션에는 이름이 같은 속성이 존재할 수 없고, 이름도 속성의 의미가 명확히 드러나는 것으로 사용하는 것이 좋다
- 속성의 원자성 : 속성값으로 원자 값만 사용할 수 있다
  - 모든 속성 값은 더는 분해할 수 없는 하나의 값, 즉 원자 값만 가질 수 있다
    - 하나의 속성은 여러 개의 값, 즉 다중 값을 가질 수 없다

고객아이디	고객이름	나이	등급	직업	적립금
apple	김현준	20	gold	학생	1000
banana	정소화	25	vip	간호사	2500
carrot	원유선	28	gold	교사	4500
orange	정지영	22	silver	회사원, 학생	0

그림 5-4 다중 값 속성을 포함하는 릴레이션 예 : 고객 릴레이션

  - 위 고객 릴레이션은 직업 속성이 값이 여러개를 가질 수 있으므로 관계 데이터 모델의 릴레이션으로 적합하지 않다
- 키의 종류

- 튜플을 유일하게 구별하기 위해 모든 속성을 이용하는 것보다 일부 속성만을 이용하는 것이 효율성을 높일 수 있다
- 릴레이션에 포함된 튜플들을 유일하게 구별해주는 역할은 속성 또는 속성들의 집합인 키가 담당한다
- 키는 관계 데이터 모델에서 중요한 제약조건을 정의한다
- 관계 데이터 모델에서 키는 슈퍼키, 후보키, 기본키, 대체키, 왜래키의 다섯가지 종류가 있다

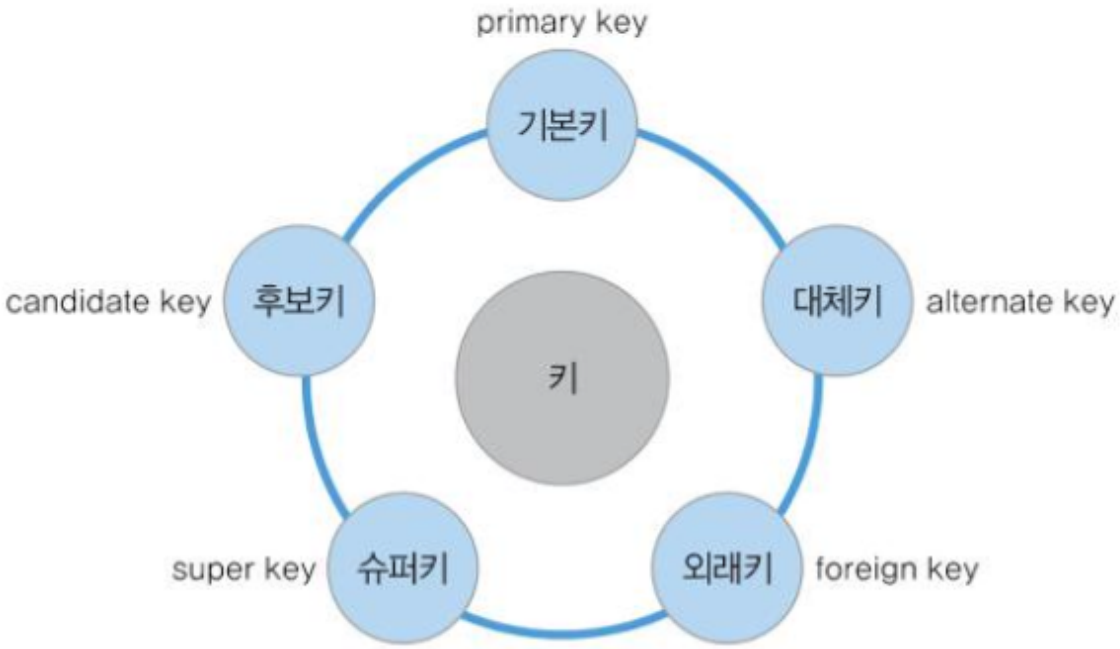

그림 5-5 키의 종류

- 슈퍼키
  - 슈퍼키는 유일성의 특성을 만족하는 속성 또는 속성들의 집합이다
  - 유일성(uniqueness)은 키가 갖추어야하는 기본 특성으로, 하나의 릴레이션에서 키로 지정된 속성 값은 튜플마다 달라야 한다는 의미
    - 즉, 키 값이 같은 튜플은 존재할 수 없다
  - 슈퍼키는 유일성을 만족하는 모든 속성들의 집합이므로 매우 많을 수 있다
    - 예) 고객 아이디, (고객 아이디, 고객 이름), (고객 아이디, 등급), (고객 아이디, 직업)...
- 후보키
  - 후보키(candidate key)는 유일성과 최소성을 만족하는 속성 또는 속성들의 집합이다
  - 최소성은 꼭 필요한 최소한의 속성들로만 키를 구성하는 특성이다
    - 즉, 슈퍼키 중에서 최소성을 만족하는 것이 후보키가 된다
    - 고객 아이디는 그 자체만으로 유일성을 만족하므로 고객 아이디와 다른 속성을 묶어서 키를 구성하는 것은 최소성에 어긋난다
    - 하지만 주소 속성이 추가되어 (고객이름, 주소)로 묶을때 유일성을 만족한다면, 이는 후보키가 될 수 있다
      - 고객 이름 속성만으로는 유일성을 만족하지 않아서 가능하다

    - 후보키가 되기 위해 만족해야 하는 유일성과 최소성의 특성은 새로운 튜플이 삽입되거나 기존 튜플의 속성 값이 바뀌어도 유지되어야 한다
    - 후보키를 선정할 때는 현재의 릴레이션 내용, 즉 릴레이션 인스턴스만 보고 유일성과 최소성을 판단해서는 안된다
        - 데이터베이스가 사용될 현실 세계의 환경까지 염두에 두고 속성의 본래 의미를 정확히 이해한 후 슈퍼키와 후보키를 선별해야한다
- 기본키
    - 릴레이션에서 튜플을 구별하기 위해 여러 개의 후보키를 모두 사용할 필요는 엇ㅂ다
    - 여러 후보키 중에서 기본적으로 사용할 키를 선택한것이 기본키(primary key)다
    - 후보키가 1개만 존재하면 당연히 해당 후보키를 기본키로 선택해야 하겠지만 여러 개일 경웨는 데이터베이스 사용 환경을 고려하여 적합한 것을 기본키로 선택하면 된다
    - 기본 키를 선택할 때 고려하면 도움이 되는 기준
        - 널 값을 가질 수 있는 속성이 포함된 후보키는 기본키로 부적합하다
        - 값이 자주 변경될 수 있는 속성이 포함된 후보키는 기본키로 부적합하다
        - 단순한 후보키를 기본키로 선택한다
- 대체키
    - 대체키(alternate key)는 기본키로 선택되지 못한 후보키들이다
    - 대체키는 기본키를 대신할 수 있지만 기본키가 되지 못하고 탈락한 이유가 있을 수 있다

그림 5-8 키의 관계

- 외래키
    - 외래키(foreign key)는 어떤 릴레이션에 소속된 속성 또는 속성 집합이 다른 릴레이션의 기본키가 되는 키다
        - 다른 릴레이션의 기본 키를 그대로 참조하는 속성의 집합이 외래키다

- 외래키는 릴레이션들 사이의 관계를 올바르게 표현하기 위해 필요하다

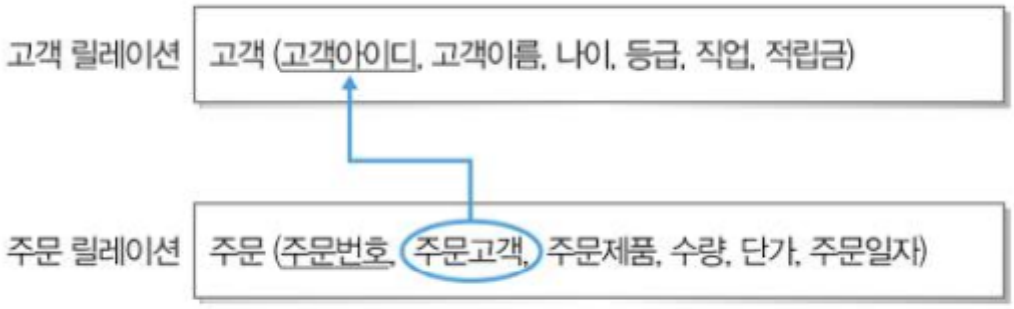

그림 5-9 고객 릴레이션과 주문 릴레이션의 스키마

- 위와 같은 예제에서 주문 릴레이션의 주문고객 속성이 고객 릴레이션의 기본키인 고객 아이디 속성을 참조하면 외래키다
- 외래키를 통해 고객 릴레이션과 주문 릴레이션이 관계를 맺어, 주문 릴레이션의 튜플과 연관성 있는 고객 릴레이션 튜플을 연결시킬 수 있다
- 일반적으로 주문 릴레이션과 같이 외래키를 가진 릴레이션을 '참조하는 릴레이션'이라 하고, 고객 릴레이션과 같이 기본키를 가진 릴레이션을 '참조되는 릴레이션'이라한다
- 외래키가 되는 속성과 기본키가 되는 속성의 이름은 서로 달라도 된다
- 하지만 외래키 속성의 도메인과 참조되는 기본키 속성의 도메인은 반드시 같아야한다
  - 도메인이 같아야 연관성 있는 튜플을 찾기 위한 비교연산이 가능하기 때문이다
- 만약 외래키가 기본키가 아닌 다른 속성을 참조한게 된다면, 기본키가 아니면 튜플을 유일하게 구별하기 어렵기 때문에 참조되는 릴레이션에서 관련 있는 튜플을 검색하지 못할 수 있다
- 그러므로 외래키는 반드시 다른 릴레이션의 기본키를 참조해야 하며 외래키의 도메인은 참조되는 기본키과 같게 정의되어야 한다

그림 5-10 외래키 예 : 고객 릴레이션과 주문 릴레이션

그림 5-11 학생 상담 데이터베이스 스키마

- 위의 예제처럼 하나의 릴레이션에는 외래키가 여러개 존재할 수 있고, 외래키 또는 외래키를 사용하여 기본키를 구성할 수 있다

그림 5-12 기본키와 외래키의 관계가 함께 정의된 릴레이션 예 : 고객 릴레이션

- 위의 예제처럼 외래키가 자기 자신의 기본키를 참조할 수 있다
  - 즉, 참조하는 릴레이션과 참조되는 릴레이션이 같을 수도 있다
- 또한, 외래키는 기본키를 참조하지만 기본키가 아니므로 널 값을 가질 수 있다
- 마찬가지로, 외래키는 기본키가 아니기 때문에 서로 다른 튜플이 같은 값을 가질 수 있다

관계 데이터 모델의 제약

- 관계 데이터 모델에서 정의하고 있는 기본 제약 사항은 키와 관련한 무결성 제약조건(integrity constraint)이다
  - 무결성은 데이터에 결함이 없는 상태, 즉 데이터가 정확하고 유효하게 유지된 상태를 말한다
- 무결성 제약조건의 주요 목적은 데이터베이스에 저장된 데이터의 무결성을 보장하고, 데이터베이스의 상태를 일관되게 유지하는 것이다
  - 이를 위해 필요한 세부 규칙도 정의하고 있다
  - 무결성 제약조건은 어느 시점에 데이터베이스에 저장된 데이터를 의미하는 데이터베이스 상태 또는 데이터베이스 인스턴스가 항상 지켜야 하는 중요한 규칙이다
- 데이터베이스가 삽입, 삭제, 수정 연산으로 상태가 변하더라도 무결성 제약조건은 반드시 지켜져야한다
- 보안과 무결성의 차이점은 보안은 권한이 없는 사용자로부터 데이터를 보호하는 것이라면, 무결성은 권한이 있는 사용자의 잘못된 요구에 대해 데이터가 부정확해지지 않도록 보호하는 것
- 관계 데이터 모델이 기본으로 포함하고 있는 무결성 제약조건에는 개체 무결성 제약조건과 참조 무결성 제약조건이 있다

- 데이터베이스의 상태를 일관성 있게 유지하기 위해서는 두 가지를 모두 만족시켜야한다

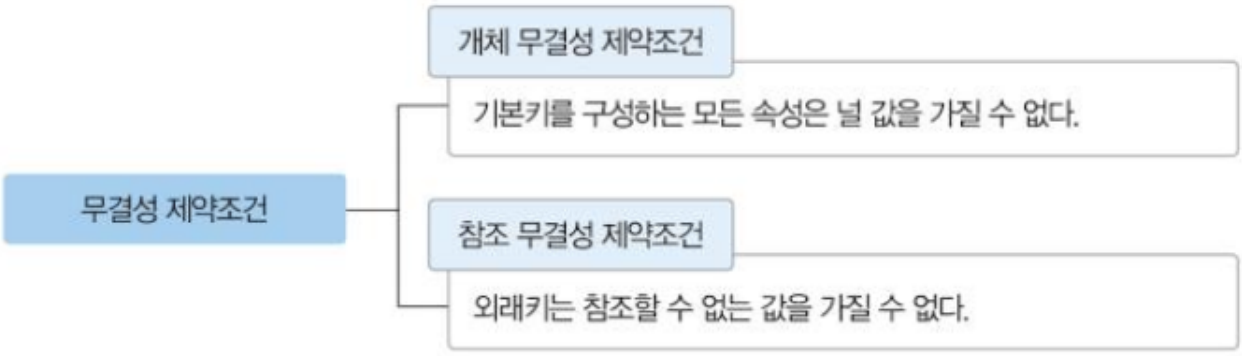

그림 5-13 관계 데이터 모델의 무결성 제약조건

- 개체 무결성 제약조건
  - 개체 무결성 제약조건(entity integrity constraint)은 기본키를 구성하는 모든 속성은 널 값을 가지면 안된다는 규칙이다
  - 기본키를 구성하는 속성 전체나 일부가 널 값이 되면 튜플의 유일성을 판단할 수 없어 기본키의 원래 목적을 상실하게 된다

고객아이디	고객이름	나이	등급	직업	적립금
apple	김현준	20	gold	학생	1000
NULL	정소화	25	vip	간호사	2500
carrot	원유선	28	gold	교사	4500
NULL	정지영	22	silver	학생	0

그림 5-14 개체 무결성 제약조건을 위반한 릴레이션 예 : 고객 릴레이션

  - 개체 무결성 제약 조건을 만족시키려면 새로운 튜플이 삽입되는 연산과 기본 튜플의 기본키 속성 값이 변경되는 연산이 발생할 때 기본키에 널 값이 포함되는 상황에서는 연산의 수행을 거부하면 된다
    - 이는 데이터베이스 관리 시스템이 자동으로 수행하므로 새로운 릴레이션을 생성할 때마다 기본키를 어떤 속성들로 구성할 것인지 데이터베이스 관리 시스템에 알려주면 된다
- 참조 무결성 제약조건
  - 참조 무결성 제약조건은 외래키에 대한 규칙으로 연관된 릴레이션들에 적용된다
  - 참조 무결성 제약조건(referential integrity constraint)이란 외래키는 참조할 수 없는 값을 가질 수 없다는 규칙이다
  - 외래키가 자신이 참조하는 릴레이션의 기본키가 상관이 없는 값을 가지게 되면 두 릴레이션을 연관시킬 수 없으므로 외래키 본래의 의미가 없어진다
    - 그러므로, 외래키는 자신이 참조하는 릴레이션에 기본키 값으로 존재하는 값, 즉 참조 가능한 값만

가져와야 한다

그림 5-15 참조 무결성 제약조건을 위반한 릴레이션 예 : 주문 릴레이션

- cherry라는 기본키가 고객 릴레이션에 존재하지 않으므로 참조 무결성 제약조건을 위반한 것

그림 5-16 외래키가 널 값인 릴레이션 예 : 주문 릴레이션

- 하지만 널 값을 가진다고해서 참조 무결성 제약조건을 위반한 것은 아니다

    - 주문고객 속성이 널이라는 것은 주문한 고객이 누구인지 모를 뿐, 고객 릴레이션에 존재하지 않는 고객이 주문한 것으로 판단하기는 어렵다
- 데이터베이스의 상태가 변해도 참조 무결성 제약조건을 만족시켜야한다
    - 만약 고객 릴레이션의 기본키인 고객아이디 속성 값으로 존재하지 않는 값이 주문 릴레이션의 주문 고객(외래키) 값으로 선택되면 주문 릴레이션에 새로운 튜플을 삽입하는 연산을 거부하면 된다
    - 또한, 고객 릴레이션(참조되는 릴레이션)에 존재하는 튜플을 삭제하는 연산은 참조 무결성 제약조건을 위반하지 않는 경우에만 수행한다
    - 고객 릴레이션(참조되는 릴레이션)의 기본키인 고객아이디 속성의 값을 변경하는 연산을 수행할 때는, 참조하는 외래키가 있으면 변경 연산을 수행하지 않고나, 주문 릴레이션(참조하는 릴레이션)에 남아있는 관련 튜플에서 주문고객 속성의 값을 함께 변경해야 참조 무결성 제약조건을 만족시킨다