

## 제3장 정규화 이론

관계형 데이터 모델의 높은 인기는 그 개념의 단순성과 사용의 용이성 이외에 확고한 이론의 뒷받침에서 기인한다. 수학의 집합 이론과 더불어 구조화된 관계를 유도해 내기 위한 정규화(normalization) 이론이 관계형 데이터 모델을 보다 유용한 데이터 모델로 이끌고 있다. 본 장에서는 정규화 이론과 관련된 함수적 종속성, 다중값 종속성, 그리고 결합 종속성 등의 개념과 다양한 유형의 정규형을 통하여 비구조적인 관계를 점차 구조적인 관계로 전환시켜 가는 과정을 살펴보기로 한다. 정규화 이론은 관계형 데이터 모델의 관계에 근거하여 생성된 이론이나, 상당부분은 데이터 모델의 유형에 상관없이, 데이터베이스의 개념적 설계에 활용될 수 있다.

### 3.1 정규화의 개념

코드(Codd, 1972)에 의해 처음으로 제안된 정규화 이론은 다양한 유형의 검사를 통하여, 관계형 데이터 모델의 관계를 보다 구조화된 것으로 개선시켜 나가는 절차에 관련된 이론이다. 구조화된 관계란 데이터베이스상에서 삽입, 삭제, 또는 갱신이 이루어질 때 문제점이 발생하지 않는 관계이다. 코드는 처음에 관계의 속성 간의 함수적 종속을 분석함으로써 이룩할 수 있는 세 가지 유형의 정규형(normal form), 즉, 1차, 2차, 그리고 3차 정규형을 제시하였다.

이 후, 보다 강력한 유형의 3차 정규형을 발견하여, 이를 3차 정규형과 구분하여 제안한 사람의 이름을 따서 보이스-코드(Boyce-Codd) 정규형이라 명하였다. 이와 같이 함수적 종속과 관련 있는 네 가지 유형의 정규형을 3.2절에서 공부하기로 한다. 그 외의 정규형으로는 다중값 종속성과 관련된 4차 정규형, 결합(join) 종속성과 관련된 5차 정규형, 그리고 이론상으로 가장 완벽한 관계를 제시하는 영역-키(domain-key) 정규형 등이 있는데, 3.3절에서 이 세 가지 유형의 정규형을 살펴보기로 한다.

### 3.1.1 구조화된 관계

데이터의 정규화는 하나의 비구조적인 관계 스키마를 두 개 이상의 보다 구조적인 관계 스키마로 분리해 내는 과정으로, 데이터베이스 설계자가 개념적인 데이터베이스를 구축하는 데 있어서 필요한 분석의 틀을 제공한다. 구조화된 관계(well-structured relations)란 데이터베이스에 저장되는 데이터의 중복을 최소화시킴으로써, 삽입, 삭제, 또는 갱신으로 인한 오류나 데이터의 불일치를 발생시키지 않는 관계이다. 예를 들어, <그림 3-1>에 표시된 수강과목 관계를 생각해 보자. 이 관계는 학생이 현재 수강하고 있는 과목을 나타내고 있으며, 주키는 학번과 과목번호가 결합된 합성키이다.

<그림 3-1> 수강과목 관계

수강과목

| 학번      | 과목번호  | 과목이름      |
|---------|-------|-----------|
| 0619046 | BB203 | 컴퓨터 프로그래밍 |
| 0619143 | BB201 | 경영의 이해    |
| 0619143 | BB353 | 정보시스템     |
| 0619272 | BB203 | 컴퓨터 프로그래밍 |
| 0619272 | BB353 | 정보시스템     |
| ⋮       | ⋮     | ⋮         |

위의 관계는 세 가지 속성으로 구성된 매우 간단한 관계이지만, 자세히 살펴보면 데이터를 갱신할 때 다음과 같은 세 가지 유형의 이상異狀(anomaly)이 발생한다.

1. 삽입 이상: 새로운 과목을 개설하였으나 아직 수강생이 없는 경우에 이 정보를 <그림 3-1>의 테이블에 포함시킬 수 없다. 이 관계의 주키는 학번과 과목번호가 합쳐진 합성키이므로, 주키의 일부인 학번이 결정되지 않은 상태에서 과목에 관한 정보만 수록할 수 없기 때문이다. 예를 들어, 과목번호 BB208, 과목이름 '전자상거래'라는 새로운 과목이 개설되더라도 수강생이 없는 한 이 정보는 데이터베이스에 기록될 수 없다.
2. 삭제 이상: 만약 0619143 학생이 BB201 과목의 수강을 철회하였는데, 이 학생이 이 과목을 수강하는 유일한 학생이었다면, 이 타플의 삭제로 말미암아 BB201 과목의 이름이 '경영의 이해'라는 중요한 정보가 사라지게 된다. 이처럼 특정 타플의 삭제로 인하여 필요한 정보의 손실이라는 원치 않는 결과를 낳는 것은 바람직하지 못하다.
3. 갱신 이상: 과목이름은 그 과목을 수강하고 있는 학생 수만큼 반복되어 나타나고 있는데, 이로 인하여 저장 공간의 낭비와 데이터 불일치의 문제를 야기할 수 있다. 예를 들어, BB203의 과목이름을 과목 내용의 변경에 따라 '경영과 컴퓨터'라고 갱신하였다면, 이 과목을 수강하고 있는 학생의 수만큼 해당 타플의 과목이름 속성값을 수정해 주어야 한다. 만약 모든 해당 타플을 동시에 수정해 주지 않을 경우에는 데이터의 불일치가 발생한다. 즉, 일부는 BB203의 과목이름이 '경영과 컴퓨터'로 수정되어 있고, 일부는 '컴퓨터 프로그래밍'이란 이름으로 남아 있다면, 어느 것이 올바른 과목이름인지 판단할 수 없게 된다.

이러한 문제점으로 미루어 볼 때, <그림 3-1>의 관계는 구조화된 관계가 아니다. 이 관계를 보다 구조화된 관계로 전환시키기 위해서는 <그림 3-2>와 같이 두 개의 관계로 분리시켜야 한다. 자세한 분리 기준에 관하여는 3.2절에서 설명하기로 한다. 새로 분리된 두 관계는 앞에서 설명한 세 가지 이상이 발생하지 않는 구조화된 관계이다.

〈그림 3-2〉 분리되어 구조화된 관계

| 수강 | 수강      |       | 과목 | 과목    |           |
|----|---------|-------|----|-------|-----------|
|    | 학번      | 과목번호  |    | 과목번호  | 과목이름      |
|    | 0619046 | BB203 |    | BB201 | 경영의 이해    |
|    | 0619143 | BB201 |    | BB203 | 컴퓨터 프로그래밍 |
|    | 0619143 | BB353 |    | BB353 | 정보시스템     |
|    | 0619272 | BB203 |    | ⋮     | ⋮         |
|    | 0619272 | BB353 |    | ⋮     | ⋮         |
|    | ⋮       | ⋮     |    | ⋮     | ⋮         |

### 3.1.2 구조화된 관계의 문제점

정규화를 통하여 구조화된 관계는 삽입, 삭제, 그리고 갱신 이상 등의 문제점은 해결할 수 있으나, 데이터베이스의 실행속도와 참조 무결성과 관련된 다른 문제점을 야기시킨다. 예를 들어, 각 학생이 수강하고 있는 과목의 이름을 파악하고자 할 때, 구조화되기 이전의 〈그림 3-1〉의 관계에서는 다음과 같은 SQL 명령으로 간단히 해결될 수 있다.<sup>1</sup>

```
SELECT 학번, 과목이름
FROM 수강과목;
```

그러나 정규화된 〈그림 3-2〉의 관계에서는 학번과 과목이름이 한 테이블에 존재하지 않기 때문에 두 테이블을 결합시켜야 한다. 이를 위하여 다음과 같이 더 복잡한 SQL 명령을 사용해야 한다.

```
SELECT 학번, 과목이름
FROM 수강, 과목
WHERE 수강.과목번호=과목.과목번호;
```

정규화된 경우의 명령어가 더 복잡해졌을 뿐만 아니라, 보다 중요한 사실은 두 테이블을 결합시키는 이와 같은 명령의 실행시간이 한 테이블에만 관련된 명령의 실행시간보다 더 많이 소요된다는 점이다.

<sup>1</sup> SQL에 관한 자세한 내용은 다음 장에 설명되므로 이 명령문에 관한 사항은 무시하여도 좋다.

정규화를 통해 구조화된 관계가 갖는 또 다른 문제점은 분리된 두 관계의 참조 무결성을 유지하기 위한 노력이 필요하다는 점이다. 즉, <그림 3-2>의 수강 관계의 과목번호의 속성값은 반드시 과목 관계의 어느 한 타플의 과목번호의 속성값과 일치하여야 한다. 이를 위하여 데이터베이스 관리시스템이나 응용 프로그램에서 이를 통제할 수 있는 장치를 마련하여야 한다.

✱ 이러한 문제점에도 불구하고, 일반적으로 정규화의 장점은 단점보다 크다. 정규화로 인하여 발생하는 문제점은 보다 우수한 성능의 컴퓨터나 정교한 데이터베이스 관리시스템을 도입함으로써 해결될 수 있기 때문이다. 실행 속도의 문제점은 빠른 속도로 향상되고 있는 컴퓨터의 성능으로 해결할 수 있고, 참조 무결성은 데이터베이스 관리시스템에서 자동적으로 통제하기 때문이다. 따라서 시간이 흐를수록 정규화의 장점이 더욱 부각될 것이다.

정규화는 단계적인 검사과정을 통하여 점차 높은 수준까지 진행될 수 있다. 그러나 앞에서 살펴본 문제점으로 인하여, 실현가능한 최고 수준의 정규형이 반드시 최상의 해결책은 아니다. 일반적으로 정규형의 수준을 높일수록 데이터의 삽입, 삭제, 그리고 갱신 이상이 감소되나, 데이터베이스의 실행속도나 참조 무결성 등의 문제는 보다 심각해진다. 따라서 모든 관계는 데이터의 삽입, 삭제, 그리고 갱신으로 발생할 수 있는 문제의 심각성과 요구되는 데이터베이스의 실행속도를 고려하여 적절한 수준까지의 정규형을 추구하는 것이 바람직하다. 실무에서는 대략 3차 정규형 내지 보이스-코드 정규형을 적절한 수준으로 보고 있다. 이 수준의 정규형 관계가 너무 세분화되어 요구하는 실행속도를 만족시키지 못할 때, 정규화된 관계 중 일부 관계들을 다시 결합시키는 탈정규화(denormalization) 방법도 실무에서 제시되고 있다(Inmon, 1987). 탈정규화에 대한 자세한 내용은 9.3.4절에서 공부하기로 한다.

### 3.2 함수적 종속과 정규화

관계형 데이터 모델의 개념적 데이터베이스 설계에 있어서 가장 중요한 개념은 함수적 종속과 그에 근거한 정규화 이론이다. 본 절에서는 함수적 종속과

함수적 종속에 근거한 네 가지 유형의 정규형에 관하여 살펴보기로 한다.

### 3.2.1 함수적 종속

정규화 이론은 속성간의 함수적 종속관계에 기초하고 있다. 속성 Y가 속성 X에 함수적으로 종속되어 있다는 것은 X의 속성값이 Y의 속성값을 결정짓는다는 것을 의미한다. 바꾸어 말하면, X의 속성값이 주어지면 Y의 속성값을 알아낼 수 있다는 것을 의미한다. 그러나 역으로 Y의 속성값을 알고 있다고 하더라도 X의 속성값을 안다고 할 수는 없다. 이러한 함수적 종속 관계를 그림으로 표현하면 다음과 같다.

$$X \rightarrow Y$$

〈그림 3-1〉의 수강과목 관계를 살펴보면 속성간에 다음과 같은 함수적 종속을 발견할 수 있다.

과목번호  $\rightarrow$  과목이름

과목이름  $\rightarrow$  과목번호

단, 이러한 함수적 종속은 과목이름이 유일한 값을 가진다는 가정이 있을 때만 성립된다. 만약 경영학과와 전산학과에서 공히 ‘컴퓨터 프로그래밍’이라는 동일한 이름으로 과목을 개설하고 있으나, 과목번호는 각각 BB203과 CH201로 서로 다르다면, 두 번째 표시된 ‘과목이름  $\rightarrow$  과목번호’의 함수적 종속은 성립되지 않고, 첫번째의 ‘과목번호  $\rightarrow$  과목이름’의 함수적 종속만 존재한다. 그 이유는 ‘컴퓨터 프로그래밍’이라는 과목이름의 속성값을 알더라도, 과목번호의 속성값이 BB203인지 CH201인지 결정지을 수 없기 때문이다. 이처럼 함수적 종속은 동일한 속성으로 구성된 관계라 하더라도 속성이 갖는 의미(semantics)에 따라 달라진다. 일반적으로 속성 Y가 속성 X에 함수적으로 종속되었다면, X로부터 Y에 대한 대응비는 다:1이며, X를 Y의 결정자(determinant)라고 부른다. 위의 예에서 과목번호로부터 과목이름에 대한 대응비는 2:1이며, 과목번호는 과목이름 속성의 결정자이다.

〈그림 3-3〉 고객 주문서

| 주 문 서                 |           |                    |         |         |
|-----------------------|-----------|--------------------|---------|---------|
| 주문번호: 2157            |           | 주문일: 2006년 10월 20일 |         |         |
| 고객이름: 이 명 훈           |           | 고객번호: c1560        |         |         |
| 주 소: 서울 마포구 창전동 7-120 |           |                    |         |         |
| 전 화: 362-2342         |           |                    |         |         |
| 제품번호                  | 제 품 명     | 수 량                | 단 가     | 금 액     |
| BD307                 | 로얄 침대 킴   | 1                  | 350,000 | 350,000 |
| TB115                 | 로얄 식탁 6인용 | 1                  | 300,000 | 300,000 |
| CH115                 | 로얄 식탁 의자  | 6                  | 100,000 | 600,000 |

## 3.2.2 1차 정규형

〈그림 3-3〉은 명문가구의 고객 주문서의 일부를 발췌한 것인데, 다음과 같은 데이터의 의미를 내포하고 있다. 한 주문서는 반드시 한 고객에 속하며, 한 주문서에는 여러 종류의 제품이 포함될 수 있다. 따라서 주문번호 속성이 주 키가 되며, 제품에 관한 속성은 한 주문서 내에서 주문된 제품 수만큼 반복되는 반복집단(repeating groups)이다. 이 주문서에 나타난 데이터를 관계 스키마로 표현하면 아래와 같다.

주문(주문번호, 주문일, 고객이름, 고객번호, 주소, 전화, {제품번호, 제품명, 수량, 단가})

제품번호에서 단가까지는 다중값을 갖는 반복집단으로 { } 속에 표시하였다. 그리고 주문서에 나타난 금액은 수량과 단가를 곱하여 계산될 수 있는 유도 속성이므로 데이터베이스에 저장하지 않기로 한다. 이를 테이블의 형태로 전환시키면 〈그림 3-4〉와 같은 형태가 된다.

1차 정규형(first normal form(1NF))은 “테이블의 각 셀(cell)은 단일값을 갖는다”는 관계형 데이터 모델의 본질적 제약과 관련된 개념이다. 1차 정규형의 관계는 반복집단을 포함하지 않는 원자값(atomic value)으로만 구성된 타플을 갖는 관계이다. 바꾸어 말하면, 1차 정규형은 ‘타플의 속성으로서의 관계’ 또는 ‘관계 속의 관계’를 허용하지 않는다.

〈그림 3-4〉 비정규형의 관계

| 주문<br>번호 | 주문일             | 고객<br>이름 | 고객<br>번호 | 주소                  | 전화       | 제품<br>번호 | 제품명       | 수량 | 단가      |
|----------|-----------------|----------|----------|---------------------|----------|----------|-----------|----|---------|
| 2157     | 2006.<br>10. 20 | 이명훈      | c1560    | 서울 마포구<br>창전동 7-120 | 362-2342 | BD307    | 로얄 침대 퀸   | 1  | 350,000 |
|          |                 |          |          |                     |          | TB115    | 로얄 식탁 6인용 | 1  | 300,000 |
|          |                 |          |          |                     |          | CH115    | 로얄 식탁 의자  | 6  | 100,000 |
| 2158     | 2006.<br>10. 20 | 김철규      | c0714    | 서울 서대문구<br>연희동 125  | 332-6114 | SP120    | 듀크 쇼파     | 1  | 400,000 |
|          |                 |          |          |                     |          | TT120    | 듀크 차 탁자   | 1  | 150,000 |
| 2159     | 2006.<br>10. 20 | 이명훈      | c1560    | 서울 마포구<br>창전동 7-120 | 362-2342 | SP120    | 듀크 쇼파     | 1  | 400,000 |
| ⋮        | ⋮               | ⋮        | ⋮        | ⋮                   | ⋮        | ⋮        | ⋮         | ⋮  | ⋮       |

〈그림 3-4〉에 나타난 주문 관계를 살펴보면, 하나의 주키값(주문번호 2157)에 대응하는 제품번호의 속성값이 다수(BD307, TB115, CH115)이므로, 이 관계는 비정규형의 관계이다. 사실 주문이라는 관계 속에는 제품이라는 다른 관계가 내재되어 있는데, 이러한 비정규형의 관계는 관계형 데이터 모델의 본질적 제약을 위배한 것이므로 데이터베이스에 저장될 수 없다. 이를 데이터베이스에 저장할 수 있는 1차 정규형의 관계로 전환하기 위해서는 반복집단을 제거하여야 한다. 모든 타플이 원자값을 갖는 1차 정규형을 이루기 위하여, 역설적이지만 각 제품에 대하여 주문과 고객에 대한 속성값을 〈그림 3-5〉와

〈그림 3-5〉 1차 정규형의 관계

| 주문<br>번호 | 주문일             | 고객<br>이름 | 고객<br>번호 | 주소                  | 전화       | 제품<br>번호 | 제품명       | 수량 | 단가      |
|----------|-----------------|----------|----------|---------------------|----------|----------|-----------|----|---------|
| 2157     | 2006.<br>10. 20 | 이명훈      | c1560    | 서울 마포구<br>창전동 7-120 | 362-2342 | BD307    | 로얄 침대 퀸   | 1  | 350,000 |
| 2157     | 2006.<br>10. 20 | 이명훈      | c1560    | 서울 마포구<br>창전동 7-120 | 362-2342 | TB115    | 로얄 식탁 6인용 | 1  | 300,000 |
| 2157     | 2006.<br>10. 20 | 이명훈      | c1560    | 서울 마포구<br>창전동 7-120 | 362-2342 | CH115    | 로얄 식탁 의자  | 6  | 100,000 |
| 2158     | 2006.<br>10. 20 | 김철규      | c0714    | 서울 서대문구<br>연희동 125  | 332-6114 | SP120    | 듀크 쇼파     | 1  | 400,000 |
| 2158     | 2006.<br>10. 20 | 김철규      | c0714    | 서울 서대문구<br>연희동 125  | 332-6114 | TT120    | 듀크 차 탁자   | 1  | 150,000 |
| 2159     | 2006.<br>10. 20 | 이명훈      | c1560    | 서울 마포구<br>창전동 7-120 | 362-2342 | SP120    | 듀크 쇼파     | 1  | 400,000 |
| ⋮        | ⋮               | ⋮        | ⋮        | ⋮                   | ⋮        | ⋮        | ⋮         | ⋮  | ⋮       |



같이 반복하여야 한다. 따라서 비정규형 관계의 한 타플(주문 번호 2157)이 1차 정규형의 관계로 전환된 후, 주문제품 수에 따라 3개로 늘어났다. 그리고 정규화된 관계에서는 주문번호 속성이 더 이상 각 타플에서 유일한 값을 갖지 못하므로 주키의 역할을 수행할 수 없다. 각 타플에서 유일성과 최소성을 만족시키는 조합은 (주문번호와 제품번호를 연결시킨 값)이므로, 이 두 속성을 연결시킨 합성키가 1차 정규형 관계의 새로운 주키가 된다.

### 3.2.3 2차 정규형

1차 정규형의 관계는 그 자체로서 데이터베이스 내에 테이블의 형태로 저장될 수 있다. 그러나 <그림 3-5>의 1차 정규형의 관계는 다음과 같은 문제점을 내포하고 있다.

삽입 이상: 아직 주문이 없는 제품에 대한 정보는 데이터베이스에 수록할 수 없다. 주문번호가 주키의 일부를 형성하고 있으므로, 주문번호 속성 값을 공소값으로 갖는 타플을 삽입할 수 없기 때문이다.

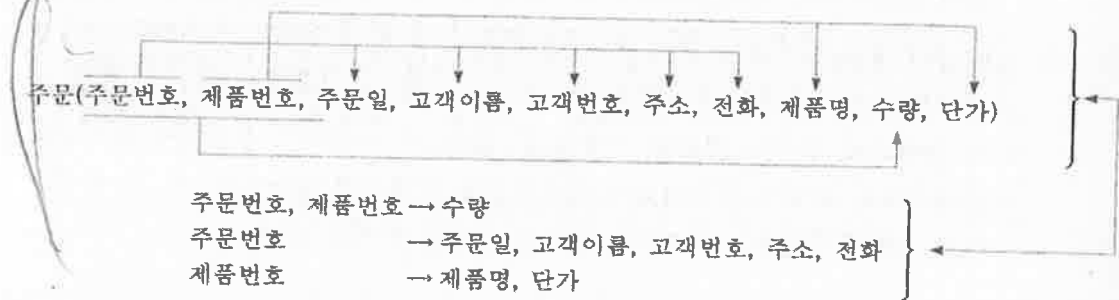
삭제 이상: 만약 이명훈 고객이 제품 BD307을 주문한 유일한 고객인데, 이 주문을 취소하였다면, <그림 3-5>의 첫번째 타플을 삭제하여야 한다. 이 경우 BD307 제품의 정보도 또한 같이 사라지게 된다.

갱신 이상: 예를 들어, 이명훈 고객의 주소가 변경되었다면, 이 주소를 포함하고 있는 모든 타플을 반복하여 갱신해 주어야 한다.

이러한 문제점을 제거하기 위하여 속성간의 함수적 종속을 분석하여, 하나의 관계를 두 개 이상의 관계로 분리시켜야 한다. 사용자와 면담한 결과, <그림 3-5>의 1차 정규형의 관계에서 <그림 3-6>과 같은 함수적 종속을 발견하였다.

어느 관계가 1차 정규형의 조건을 만족하면서, 주키를 구성하지 않는 나머지 속성(비주키)이 주키에 완전 함수적 종속(full functional dependency)을 이룰 때, 이를 2차 정규형(second normal form(2NF))의 관계라고 한다. 비주키 속성이 주키에 완전히 함수적으로 종속되어 있다는 것은 주키의 일부만으로는

〈그림 3-6〉 주문 관계의 함수적 종속



비주키 속성의 값을 결정지을 수 없다는 것을 의미한다. 예를 들어, 〈그림 3-6〉의 관계에서 비주키 속성인 제품명은 주키의 일부인 제품번호만으로 그 값을 결정지을 수 있으므로, 비주키 속성은 주키에 함수적으로 완전 종속되어 있지 않다. 다시 말하면, 제품명은 주문번호에 상관없이 제품번호에만 종속되어 있으며, 이런 경우 부분 종속(partial dependency)이 존재한다고 말한다. 따라서 〈그림 3-6〉의 관계는 1차 정규형의 조건은 만족시키고 있으나, 2차 정규형의 관계는 아니다.

〈그림 3-6〉의 관계를 2차 정규형으로 전환하기 위해서는 부분 종속을 제거하여야 한다. 비주키 속성이 주키에 완전 종속되도록 관계를 분리하면, 주문번호와 제품번호를 합성키로 하며 수량을 비주키 속성으로 갖는 주문-제품 관계, 주문번호를 주키로 하며 주문일, 고객이름, 고객번호, 주소, 전화 등으로 구성된 주문-고객 관계, 그리고 제품번호를 주키로 하고 제품명과 단가로 구성된 제품 관계로 나누어지게 된다. 나누어진 결과는 〈그림 3-7〉과 같다.

다시 한번 정리하면, 2차 정규형은 1차 정규형에서 부분 종속을 제거한 것이다. 부분 종속은 주키가 합성키인 경우에만 발생하게 되므로, 어느 관계의 주키가 단일 속성으로 되어 있으면 1차 정규형의 조건을 만족시키면, 그 관계는 당연히 2차 정규형의 조건도 만족시킨다.

그림 3-7) 2차 정규형으로 전환된 주문 관계

| 주문-제품 | 주문번호 | 제품번호  | 수량 |
|-------|------|-------|----|
|       | 2157 | BD307 | 1  |
|       | 2157 | TB115 | 1  |
|       | 2157 | CH115 | 6  |
|       | 2158 | SP120 | 1  |
|       | 2158 | TT120 | 1  |
|       | 2159 | SP120 | 1  |
|       | ⋮    | ⋮     | ⋮  |

| 주문-고객 | 주문번호 | 주문일          | 고객이름 | 고객번호  | 주소               | 전화       |
|-------|------|--------------|------|-------|------------------|----------|
|       | 2157 | 2006. 10. 20 | 이명훈  | c1560 | 서울 마포구 창전동 7-120 | 362-2342 |
|       | 2158 | 2006. 10. 20 | 김철규  | c0714 | 서울 서대문구 연희동 125  | 332-6114 |
|       | 2159 | 2006. 10. 20 | 이명훈  | c1560 | 서울 마포구 창전동 7-120 | 362-2342 |
|       | ⋮    | ⋮            | ⋮    | ⋮     | ⋮                | ⋮        |

| 제품 | 제품번호  | 제품명       | 단가      |
|----|-------|-----------|---------|
|    | BD307 | 로얄 침대 퀸   | 350,000 |
|    | TB115 | 로얄 식탁 6인용 | 300,000 |
|    | CH115 | 로얄 식탁 의자  | 100,000 |
|    | SP120 | 듀크 쇼파     | 400,000 |
|    | TT120 | 듀크 차 탁자   | 150,000 |
|    | ⋮     | ⋮         | ⋮       |

### 3.2.4 3차 정규형

<그림 3-7>에 나타난 관계는 2차 정규형의 조건을 만족시키고 있으나, 그중 두 번째의 주문-고객 관계를 자세히 살펴보면 다음과 같은 문제점을 내포하고 있다.

삽입 이상: 주문을 내지 않은 고객에 대한 정보는 데이터베이스에 수록할 수 없다. 이 관계의 주키는 주문번호이므로, 주문번호 속성이 공값인 타폴을 삽입할 수 없기 때문이다.

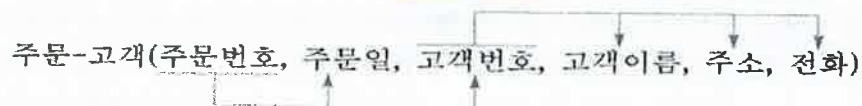
삭제 이상: 만약 김철규 고객이 2158 주문을 취소하였다면, 이 고객에 관한 정보가 사라지게 된다.

갱신 이상: 한 고객이 여러 개의 주문을 낸 경우, 고객과 관련된 정보가 중복되어 저장된다. 따라서 고객관련 정보가 변경될 경우, 모든 해당 타  
플을 반복하여 갱신해 주어야 한다.

이와 같은 문제점은 속성간의 이전 종속(transitive dependence) 때문에 발생한다. 이전 종속이란 비주키 속성간에 발생하는 함수적 종속을 의미한다. 3차 정규형(third normal form(3NF))의 관계는 2차 정규형의 조건을 만족하면서, 이전 종속이 존재하지 않는 관계이다. <그림 3-7>의 주문-고객 관계에는 다음과 같은 함수적 종속이 존재한다.

주문번호 → 주문일, 고객이름, 고객번호, 주소, 전화  
고객번호 → 고객이름, 주소, 전화

이 관계에서 주키인 주문번호가 비주키 속성값을 모두 결정할 수 있으나, 위의 함수적 종속을 자세히 살펴보면 다음과 같은 경로를 통하여 주키가 비주키 속성값을 결정한다는 것을 알 수 있다.



먼저 주키인 주문번호가 고객번호를 결정하고, 그리고 그 고객번호가 다시 고객이름, 주소, 그리고 전화를 결정한다. 즉, 이 관계에는 비주키 속성인 고객번호가 다른 비주키 속성들(고객이름, 주소, 전화)의 값을 결정하는 이전 종속이 존재하고 있다.

이전 종속을 제거하기 위해서는 이전 종속과 관련된 비주키 속성들을 다른 관계로 분리시켜야 한다. 이때 결정자 역할을 하는 비주키 속성(위의 경우에 고객번호)은 분리되는 관계와 원래 관계에 중복되어 존재하며, 두 관계를 연결시키는 외부키 역할을 하게 된다. 3차 정규형의 관계로 전환된 주문-고객 관계는 <그림 3-8>과 같다.

이로써 3차 정규형까지의 과정이 완료되었다. <그림 3-3>의 고객 주문서 사용자 관점은 일련의 정규화 절차를 거쳐 4개의 3차 정규형 관계로 전환되었다. 네 개의 관계를 관계형 데이터베이스 스키마로 나타내면 다음과 같

〈그림 3-8〉 3차 정규형으로 전환된 주문-고객 관계

| 주문-고객 | 주문번호 | 주문일          | 고객번호  |
|-------|------|--------------|-------|
|       | 2157 | 2006. 10. 20 | c1560 |
|       | 2158 | 2006. 10. 20 | c0714 |
|       | 2159 | 2006. 10. 20 | c1560 |
|       | ⋮    | ⋮            | ⋮     |

| 고객 | 고객번호  | 고객이름 | 주소               | 전화       |
|----|-------|------|------------------|----------|
|    | c1560 | 이명훈  | 서울 마포구 창전동 7-120 | 362-2342 |
|    | c0714 | 김철규  | 서울 서대문구 연희동 125  | 332-6114 |
|    | ⋮     | ⋮    | ⋮                | ⋮        |

〈그림 3-9〉 3차 정규형의 관계로 전환된 고객 주문서 관점

| 주문-제품 | 주문번호 | 제품번호  | 수량 |
|-------|------|-------|----|
|       | 2157 | BD307 | 1  |
|       | 2157 | TB115 | 1  |
|       | 2157 | CH115 | 6  |
|       | 2158 | SP120 | 1  |
|       | 2158 | TT120 | 1  |
|       | 2159 | SP120 | 1  |
|       | ⋮    | ⋮     | ⋮  |

| 제품 | 제품번호  | 제품명       | 단가      |
|----|-------|-----------|---------|
|    | BD307 | 로얄 침대 퀸   | 350,000 |
|    | TB115 | 로얄 식탁 6인용 | 300,000 |
|    | CH115 | 로얄 식탁 의자  | 100,000 |
|    | SP120 | 듀크 쇼파     | 400,000 |
|    | TT120 | 듀크 차 탁자   | 150,000 |
|    | ⋮     | ⋮         | ⋮       |

| 주문-고객 | 주문번호 | 주문일          | 고객번호  |
|-------|------|--------------|-------|
|       | 2157 | 2006. 10. 20 | c1560 |
|       | 2158 | 2006. 10. 20 | c0714 |
|       | 2159 | 2006. 10. 20 | c1560 |
|       | ⋮    | ⋮            | ⋮     |

| 고객 | 고객번호  | 고객이름 | 주소               | 전화       |
|----|-------|------|------------------|----------|
|    | c1560 | 이명훈  | 서울 마포구 창전동 7-120 | 362-2342 |
|    | c0714 | 김철규  | 서울 서대문구 연희동 125  | 332-6114 |
|    | ⋮     | ⋮    | ⋮                | ⋮        |

다. 그리고 이 스키마에 의해 형성되는 최종 테이블의 형태는 <그림 3-9>와 같다.

주문-제품(주문번호, 제품번호, 수량)

제품(제품번호, 제품명, 단가)

주문-고객(주문번호, 주문일, 고객번호)

고객(고객번호, 고객이름, 주소, 전화)

### 3.2.5 보이스-코드 정규형

보이스-코드 정규형(Boyce-Codd normal form(BCNF))은 보이스(R.F. Boyce)와 코드(E.F. Codd)가 기존의 3차 정규형에서 함수적 종속으로 인하여 추가적으로 발생할 수 있는 문제점을 발견하고, 이를 보완하기 위하여 제시한 보다 엄격한 형태의 3차 정규형이다. 따라서 보이스-코드 정규형의 조건을 만족시키는 관계는 항상 3차 정규형의 조건을 만족시키나, 그 역은 사실이 아니다.

어떤 관계에 다수의 후보키가 존재하면, 그 관계가 3차 정규형을 만족시키더라도 삽입, 삭제, 그리고 갱신 이상이 발생할 수 있다. 예를 들어, <그림 3-10>의 수강과목-교수 관계를 살펴보자. 이 관계는 아래와 같은 상황을 전제로 하여, 학생이 수강하고 있는 과목과 그 과목을 강의하는 교수를 나타내고 있다.

- 각 학생은 여러 과목을 수강할 수 있다.
- 각 과목은 여러 교수가 강의할 수 있다.
- 각 과목 내에서, 한 학생은 오직 한 교수의 강의만 수강한다. 다시 말

<그림 3-10> 수강과목-교수 관계

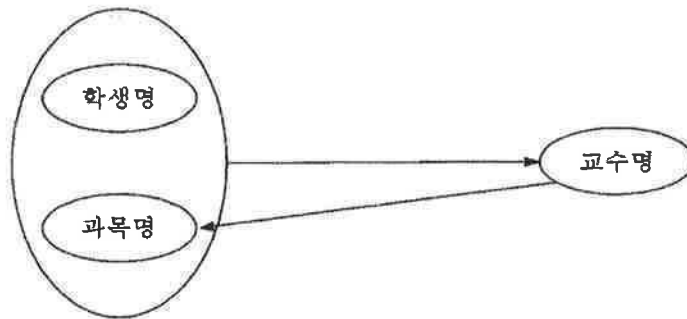
| 수강과목-교수 | 학생명 | 과목명    | 교수명 |
|---------|-----|--------|-----|
|         | 홍기호 | 데이터베이스 | 김준식 |
|         | 홍기호 | 정보시스템  | 안수희 |
|         | 이지신 | 데이터베이스 | 김준식 |
|         | 장한중 | 경영의 이해 | 최중일 |
|         | 김성한 | 데이터베이스 | 이미영 |

하면, 한 학생이 한 과목을 여러 교수로부터 수강할 수는 없다.

- 각 교수는 한 과목만 강의한다.
- 학생명, 과목명, 교수명은 각각 유일한 값을 갖는다.

이러한 데이터의 의미를 토대로 파악한 속성간의 함수적 종속을 그림으로 표현하면 <그림 3-11>과 같다.

<그림 3-11> 학생, 교수, 과목간의 함수적 종속



수강과목-교수 관계는 비주키 속성(교수명)이 주키 속성(학생명, 과목명)에 완전 종속되어 있고, 비주키 속성이 하나밖에 없으므로 비주키 속성간의 이전 종속도 존재하지 않는다. 따라서 이 관계는 3차 정규형의 관계이다. 그럼에도 불구하고 다음과 같은 이상이 발생한다.

**삽입 이상:** 수강 학생이 없는 경우에는 어느 교수가 어느 과목을 강의하는지에 관한 정보를 데이터베이스에 수록할 수 없다. 이 관계의 주키로 학생명이 포함되기 때문이다.

**삭제 이상:** 만약 장한중 학생이 '경영의 이해' 과목의 수강을 철회한다면, 최충일 교수가 '경영의 이해' 과목을 강의한다는 정보가 사라지게 된다.

**갱신 이상:** 김준식 교수가 '데이터베이스' 과목을 강의한다는 정보가 데이터베이스에 중복되어 저장되어 있으므로, 이 정보가 변경될 경우 모든 해당 타플을 반복하여 갱신해 주어야 한다.

이러한 이상은 비주키 항목인 교수명이 합성키의 일부인 과목명을 결정

하기 때문이다. 3차 정규형의 문제점을 인식한 보이스와 코드는 3차 정규형보다 더 엄격한 형태의 보이스-코드 정규형을 제시하였다. 어느 관계의 모든 결정자(determinant)가 후보키의 역할을 수행할 수 있다면, 그 관계는 보이스-코드 정규형의 관계이다. 이 조건을 <그림 3-10>의 관계에 적용시켜 보면, 이 관계가 비록 3차 정규형을 만족시키지만, 보이스-코드 정규형의 관계가 아님을 알 수 있다. 이 관계의 결정자는 학생명과 과목명 속성의 결합체와 교수명 속성 두 가지이다. 학생명과 과목명 속성값이 결합하여 교수명 속성값을 결정짓고, 교수명 속성값은 과목명 속성값을 결정짓기 때문이다. 두 결정자 중 학생명과 과목명 속성의 결합체는 후보키이지만, 교수명 속성은 후보키가 아니다. 따라서 이 관계는 보이스-코드 정규형의 관계가 아니다.

수강과목-교수 관계를 보이스-코드 정규형으로 전환시키기 위하여 결정자이지만 후보키가 아닌 교수명 속성을 다음과 같이 분리시켜야 한다.

학생-교수(학생명, 교수명)

교수-과목(교수명, 과목명)

<그림 3-10>의 관계는 다음과 같이 분리하여도 역시 보이스-코드 정규형을 만족시킨다.

학생-과목(학생명, 과목명)

교수-과목(교수명, 과목명)

그러나, 이 경우에는 두 관계를 결합하여 원래의 수강과목-교수 관계를 복원할 수 없다. 그 이유는 과목명이 교수명을 결정지을 수 없으므로 어느 학생이 어느 과목을 어떤 교수로부터 수강하는지 알 수 없기 때문이다. 따라서 두번째 방법은 보이스-코드 정규형은 만족하지만, 원래 관계의 의미를 나타내지 못하므로 잘못 분리한 경우이다.

<그림 3-10>의 관계에서 후보키가 될 수 있는 것은 학생명과 과목명 속성의 결합체와 학생명과 교수명 속성의 결합체이다. 만약 두 후보키 중에서 학생명과 과목명 대신 학생명과 교수명을 주기로 선정하였다고 가정해 보자.



그러면 이 관계에는 주키의 일부인 교수명이 비주키 속성인 과목명을 결정하므로 부분 종속이 존재하고, 따라서 이 관계는 2차 정규형을 만족시킬 수 없다. 부분 종속을 제거하기 위하여 과목명 속성을 분리시키면 위의 첫번째 방법과 같은 두 관계로 분리된다. 이처럼 다수의 후보키가 존재할 경우, 어느 후보키를 주키로 선정하느냐에 따라 최종 정규형으로 가는 과정이 달라질 수 있으나 결과는 동일하다.

보이스-코드 정규형은 3차 정규형에서 발생할 수 있는 문제점을 해결할 뿐만 아니라, 개념적으로도 3차 정규형보다 간결하다. 보이스-코드 정규형은 모든 결정자가 후보키의 역할을 수행한다는 조건만 요구될 뿐, 1차 정규형, 2차 정규형, 부분 종속, 이전 종속 등의 개념을 필요로 하지 않는다. 그러나, 복잡한 비정규형 관계에서 모든 결정자를 파악하여 관계를 분리시킨다는 것은 실질적으로 불가능하므로, 1차 정규형에서 3차 정규형까지 단계별로 진행한 후 보이스-코드 정규형의 조건을 적용하는 방법이 현실적이다.

### \*3.3 타유형의 종속과 정규화

지금까지 함수적 종속과 관련된 정규형에 관하여 공부하였다. 일반적으로 실무에서 데이터베이스를 설계할 때 요구되는 정규화의 적절한 수준은 보이스-코드 정규형이며, 현실적으로 3차 정규형도 상관없다. 이처럼 실무에서 최적의 수준으로 인식되는 보이스-코드 정규형을 만족하는 관계라 하더라도 완전한 관계가 될 수 없는 경우가 존재한다. 본 절에서는 보이스-코드 정규형 이후의 정규화 과정에 대하여 살펴보기로 한다.

#### 3.3.1 다중값 종속과 4차 정규형

보이스-코드 정규형을 만족시키는 관계는 더 이상 함수적 종속으로 인한 이상은 발생하지 않는다. 그러나 다중값 종속(multivalued dependency)으로 인한 문제가 발생할 수 있다. 한 관계에 둘 이상의 독립적인 다중값 속성이 존재하

〈그림 3-12〉 직원-특기-언어 관계

| 직원  | 특기       | 언어       |
|-----|----------|----------|
| 이상현 | 야구       | 영어       |
|     | 요리       | 일어<br>불어 |
| 이귀정 | 검도<br>성악 | 영어       |

는 경우, 이를 1차 정규형으로 전환하면서 다중값 종속이 발생한다. 켄트(Kent, 1983)는 이 문제에 관하여 기술적인 용어의 사용을 자제하면서 비교적 쉽게 설명하고 있다. 예를 들어, 직원이 보유한 특기와 구사할 수 있는 언어를 나타내는 〈그림 3-12〉의 직원-특기-언어 관계를 살펴보자. 단, 이 관계에는 다음과 같은 가정을 포함하고 있다.

- 한 직원은 여러 가지 특기를 보유할 수 있다.
- 한 직원은 여러 가지 언어를 구사할 수 있다.
- 한 직원의 특기와 언어는 상호 독립적이다.

이 관계는 한 직원 속성값에 대하여 특기와 언어 속성이 상호 독립적으로 다중값을 갖는 비정규형의 관계이다. 이 관계를 직원 속성에 대하여 나머지 두 속성이 원자값을 갖는 1차 정규형으로 전환하기 위해서는 〈그림 3-13〉

〈그림 3-13〉 1차 정규형으로 전환된 직원-특기-언어 관계

| 직원  | 특기 | 언어 |
|-----|----|----|
| 이상현 | 야구 | 영어 |
| 이상현 | 야구 | 일어 |
| 이상현 | 야구 | 불어 |
| 이상현 | 요리 | 영어 |
| 이상현 | 요리 | 일어 |
| 이상현 | 요리 | 불어 |
| 이귀정 | 검도 | 영어 |
| 이귀정 | 성악 | 영어 |

(a) 모든 가능한 값의 조합

| 직원  | 특기 | 언어 |
|-----|----|----|
| 이상현 | 야구 | 영어 |
| 이상현 | 요리 | 일어 |
| 이상현 | 요리 | 불어 |
| 이귀정 | 검도 | 영어 |
| 이귀정 | 성악 | 영어 |

(b) 무작위 배정을 통한 중복의 최소화

과 같은 두 가지 방법이 가능하다.

〈그림 3-13〉의 (a)는 한 직원에 속한 특기와 언어의 모든 속성값의 조합으로 구성되어 있는 관계이며, (b)는 한 직원에 속한 특기와 언어의 속성값을 무작위로 짝을 지어 중복을 최소화시킨 관계이다. 두 관계 모두 직원, 특기, 언어 세 속성의 합성키가 주키가 되며, 1차 정규형인 동시에 보이스-코드 정규형까지 만족시킨다. 그러나 이 관계들은 다음과 같은 문제점을 가지고 있다.

삽입 이상: 예를 들어, 직원이 새로운 언어를 구사하게 되었을 때, 현재 지니고 있는 모든 또는 일부 특기와 조합함으로써, 다수의 타플을 삽입하여야 한다.

삭제 이상: 예를 들어, 직원이 구사할 수 있는 언어를 삭제하고자 할 때, 역시 동일한 이유로 다수의 타플을 삭제하여야 한다.

갱신 이상: 속성값의 중복으로 인하여 중복적인 갱신이 요구되고, 그렇지 않을 경우 데이터의 불일치가 발생한다.

이와 같은 문제점은 다중값 종속으로 인하여 발생한다. 한 관계가 가, 나, 다의 세 속성을 가지고 있을 때,

- ① 가의 어떤 속성값에 대응하는 나,의 속성값이 여러 개 있고,
- ② 가의 어떤 속성값에 대응하는 다의 속성값이 여러 개 있으며,
- ③ 나,의 속성값과 다의 속성값 사이에 아무런 관계가 없는 경우,

이 관계에는 다중값 종속이 존재한다고 말한다.

다중값 종속으로 인한 문제점을 해결하기 위해서는 두 개의 상호 독립적인 다중값 속성을 서로 다른 두 관계로 분리시켜야 한다. 〈그림 3-14〉는 〈그

〈그림 3-14〉 4차 정규형으로 전환된 직원-특기-언어 관계

직원-특기

| 직원  | 특기 |
|-----|----|
| 이상현 | 야구 |
| 이상현 | 요리 |
| 이귀정 | 검도 |
| 이귀정 | 성악 |

직원-언어

| 직원  | 언어 |
|-----|----|
| 이상현 | 영어 |
| 이상현 | 일어 |
| 이상현 | 불어 |
| 이귀정 | 영어 |

림 3-13>의 관계를 4차 정규형의 관계로 분리시킨 그림이다. 4차 정규형(fourth normal form(4NF))은 보이스-코드 정규형을 만족시키면서 다중값 종속을 포함하지 않는 관계이다.

<그림 3-13>의 직원-특기-언어 관계에서 만약 한 직원의 특기와 언어가 상호 독립적이 아니라면, 이 관계는 그 자체로서 4차 정규형을 만족시키며, 따라서 <그림 3-14>와 같이 분리할 필요가 없다. 예를 들어, 언어와 특기가 독립적인 것이 아니라 특기와 관련하여 직원이 구사할 수 있는 언어를 나타낸 관계라면, 각 속성이 원자값을 갖도록 전환하는 방법이 명확하다. 만약 <그림 3-12>의 비정규화 관계가 <그림 3-13>의 (b)와 같이 정규화되었다면, 이 관계의 의미는 이상현은 야구용어를 영어로 구사할 수 있지만 일어나 불어로는 구사할 수 없으며, 요리용어를 일어와 불어로 구사할 수 있음을 나타낸다. 따라서 이 관계는 1차 정규형을 만족시키는 동시에 4차 정규형까지 만족시키며, 직원, 특기, 언어 세 속성의 합성키가 주기가 된다. 이처럼 동일한 속성으로 구성된 관계라 하더라도 데이터의 의미에 따라 정규형의 수준이 달라진다.

### 3.3.2 결합 종속과 5차 정규형

지금까지 공부한 비정규형의 관계에서 4차 정규형의 관계로 전환하는 과정을 살펴보면, 이상을 제거하기 위하여 복잡한 관계를 보다 단순한 관계로 나누는 작업을 수행하였다. 이 과정의 목적인 가정은 둘로 나누어진 관계를 다시 결합(join)시킴으로써, 원래의 관계로 회복시킬 수 있다는 것이었다. 그러나 관계 중에는 둘로 나눌 때는 원래의 관계로 회복할 수 없으나, 셋 또는 그 이상으로 분리시킬 때 원래의 관계를 복원할 수 있는 특수한 경우가 있다. 이러한 경우에 그 관계에는 결합 종속(join dependency)이 존재한다고 한다. 5차 정규형(fifth normal form(5NF))의 관계는 4차 정규형의 조건을 만족시키며, 결합 종속을 포함하지 않는 관계이다. 켄트(Kent, 1983)는 5차 정규형의 관계를 '동일한 주기를 갖지 않는 작은 관계들로 더 이상 분리할 수 없는 관계'로 정의하였다. 따라서 이 정의에 따르면 5차 정규형은 최종 정규형이 된다.

추출-결합 정규형(projection-join normal form(PJNF))이라는 별명으로 불리

기도 하는 5차 정규형의 사례는 매우 드물게 발생하며, 현실적으로 찾아내기도 쉽지 않다. 5차 정규형에 관한 보다 자세한 내용은 데이트(Date, 1990), 엘마스리와 나바드(Elmasri and Navathe, 1989), 또는 겐트(Kent, 1983)를 참조하기 바란다.

### 3.3.3 영역-키 정규형

영역-키 정규형(domain-key normal form(DKNF))은 모든 종속성과 제약을 고려한 이론적으로 가장 완벽한 형태의 정규형이다. 패긴(Fagin, 1981)은 그의 논문에서 영역-키 정규형의 관계는 어떠한 유형의 이상도 갖지 않음을 증명하였다. 어느 관계의 모든 종속성과 제약이 그 관계의 주키와 영역의 제약으로 인해서만 발생한다면 그 관계는 영역-키 정규형의 관계이다.

패긴의 이 논문은 이상적으로 완벽한 정규형을 발견하였다는 점에서 높이 평가받을 만하다. 그러나 영역-키 정규형이 비록 이론적으로 완벽한 정규형이라고 하더라도, 현실적으로 어떤 관계를 영역-키 정규형으로 전환시키는 방법론이 존재하지 않기 때문에 실용적인 면이 부족하다. 영역-키 정규형에 관한 보다 자세한 내용은 패긴(Fagin, 1981)이나 크룅케와 도란(Kroenke and Dolan, 1988)을 참조하기 바란다.

## 3.4 정규화된 관계의 통합

지금까지 비정규형의 관계를 정규형으로 전환하는 과정을 공부하였다. 본 절에서는 정규화된 여러 관계를 통합하는 과정을 사례를 통하여 살펴보고, 이 과정에서 발생할 수 있는 문제점의 해결 방법을 공부한다.

### 3.4.1 사 례

명문가구의 데이터베이스를 설계하기 위하여 현재 사용하고 있는 여러 가지

서류를 수집하고 사용자와 면담한 결과, 다음과 같은 정보가 요구된다는 것을 파악하였다.

#### 사 례

명문가구는 현재 10,000여 명의 고객을 보유하고 있으며, 고객이 제품을 주문하면 <그림 3-3>과 같은 고객 주문서를 작성하여 고객과 명문가구가 각각 나누어 보관한다. <그림 3-3>의 관계를 정규화하면 <그림 3-9>와 같고, 이를 관계 스키마로 표현하면 다음과 같다.

- 주문-제품(주문번호, 제품번호, 수량) ..... (1)
- 제품(제품번호, 제품명, 단가)..... (2)
- 주문-고객(주문번호, 주문일, 고객번호) ..... (3)
- 고객(고객번호, 고객이름, 주소, 전화) ..... (4)

한편 명문가구는 고객 관리를 위하여 <그림 3-15>와 같은 고객 정보를 보관하고 있다.

<그림 3-15> 고객 명단

|                      |
|----------------------|
| 고객번호: c1560          |
| 성명: 이명훈              |
| 주소: 서울 마포구 상수동 1-123 |
| 신용 한도액: 2,000,000    |

고객 명단에 나타난 주소는 <그림 3-3>의 고객 주문서의 주소와 일치하는 경우도 있고, 그렇지 않은 경우도 있다. 그 이유는 고객 주문서의 주소는 제품이 배달될 주소를 나타내고, 고객 명단의 주소는 고객의 거주지 주소를 나타내기 때문이다. 신용 한도액은 고객이 신용 거래를 할 수 있는 최대액수를 나타낸다. 이 관계는 반복집단을 포함하고 있지 않으며, 부분 종속이나 이전 종속의 문제점도 가지고 있지 않다. 따라서 이 관계는 고객번호를 주키로 하는 3차 정규형의 관계이며, 다음과 같이 표현할 수 있다.

고객(고객번호, 성명, 주소, 신용한도액) ..... (5)

명문가구는 위와 같은 고객 정보 외에 고객의 구매 상황을 보다 자세히 파악하여 마케팅 전략에 반영하기 위하여 <그림 3-16>과 같은 구매 현황 정보를 필요로 한다.

<그림 3-16> 고객별 구매 현황

| 고객별 구매 현황<br>(2006년 10월 31일) |      |          |            |            |       |
|------------------------------|------|----------|------------|------------|-------|
| 고객번호                         | 고객이름 | 최초 거래일   | 최근 거래일     | 누적 구입액     | 담당 직원 |
| c1560                        | 이명훈  | 1995/3/5 | 2006/10/20 | 4,526,000  | 박혜영   |
| c1561                        | 김한승  | 1995/3/5 | 2006/10/ 5 | 25,337,000 | 김명수   |
| c1562                        | 진철규  | 1995/3/6 | 1995/ 3/ 6 | 750,000    | 박혜영   |
| ⋮                            | ⋮    | ⋮        | ⋮          | ⋮          | ⋮     |

최초 거래일은 고객이 명문가구와 처음 거래를 시작한 날을 의미하며, 최근 거래일은 가장 최근에 구매한 날을 표시한다. 이 속성들은 주문서를 작성할 때, 자동적으로 생성 또는 갱신될 수 있도록 프로그램을 개발할 경우 구매여 데이터베이스에 저장할 필요가 없는 유도 속성이다. 누적 구매액은 고객이 명문가구와 거래를 개시한 이래 지금까지 구매한 총액을 나타낸다. 이 속성 역시 매번 주문이 발생할 때마다 주문액을 이전 누적액에 더하여 계산될 수 있는 유도 속성이다. 각 고객에게는 자신의 주문을 도와주는 담당 직원이 배정되는데, 이에 대한 정보가 <그림 3-16>의 마지막 열에 나타나 있다. 이 관계 역시 반복집단을 포함하고 있지 않으며, 부분 종속이나 이전 종속의 문제점도 가지고 있지 않다. 따라서 이 관계는 고객번호를 주키로 하는 3차 정규형의 관계이며, 다음과 같이 표현할 수 있다. 단, 유도 속성은 데이터베이스에 저장하지 않는다고 가정한다.

고객구매현황(고객번호, 고객이름, 담당직원)..... (6)

〈그림 3-17〉 제품 재고 장부

제품번호: BD307  
 제품명: 로얄 침대 퀸  
 재고량: 10  
 재주문 시점: 5  
 원가: 657,000  
 판매가: 890,000

〈그림 3-18〉 직원 명단

| 직원번호 | 성명  | 주소              | 생년월일       | 연봉         |
|------|-----|-----------------|------------|------------|
| e01  | 김준규 | 서울 서대문구 창천동 36  | 1956/03/15 | 55,000,000 |
| e02  | 이승휘 | 서울 마포구 도화동 1-21 | 1960/06/10 | 53,000,000 |
| ⋮    | ⋮   | ⋮               | ⋮          | ⋮          |

〈그림 3-19〉 직원 매출 현황

| 직원번호 | 성명  | 할당액 | 직원 매출 현황<br>(2006년 10월 31일) |        | (단위: 100만원) |
|------|-----|-----|-----------------------------|--------|-------------|
|      |     |     | 연간 누적 판매액                   | 달성도(%) | 최대 할인율(%)   |
| e05  | 박혜영 | 180 | 193                         | 107    | 10          |
| e15  | 최근식 | 150 | 105                         | 70     | 8           |
| e17  | 김명수 | 125 | 111                         | 89     | 8           |
| ⋮    | ⋮   | ⋮   | ⋮                           | ⋮      | ⋮           |

고객 정보 외에 명문가구는 〈그림 3-17〉, 〈그림 3-18〉, 그리고 〈그림 3-19〉에 나타난 것과 같은 제품과 직원에 관한 정보를 요구한다.

〈그림 3-17〉의 제품 재고 장부에 나타난 재고량은 현재 제품의 재고량을 나타내며, 재주문 시점은 제품 생산 주문을 발주해야 하는 시점을 나타낸다. 예를 들어, 제품 BD307은 재고량이 5개 이하로 떨어지면 생산 주문을 발주해야 한다. 이 관계 역시 반복집단을 포함하고 있지 않으며, 부분 종속이나 이전 종속의 문제점도 가지고 있지 않다. 따라서 이 관계는 제품번호를 주키로 하는 3차 정규형의 관



개이며, 다음과 같이 표현할 수 있다.

제품(제품번호, 제품명, 재고량, 재주문시점, 원가, 판매가) ..... (7)

한편 <그림 3-18>의 직원 명단에 속한 모든 속성은 설명 없이도 쉽게 이해될 수 있다. 이 관계 역시 반복집단을 포함하고 있지 않으며, 부분 종속이나 이전 종속의 문제점도 가지고 있지 않다. 따라서 이 관계는 직원번호를 주키로 하는 3차 정규형의 관계이며, 다음과 같이 표현할 수 있다.

직원(직원번호, 성명, 주소, 생년월일, 연봉) ..... (8)

마지막으로 <그림 3-19>의 직원 매출 현황은 영업 직원에 한하여 작성된다. 할당액은 연초에 각 영업 직원에게 배정한 한해의 판매 목표액이며, 연간 누적 판매액은 1월 1일 부터 현재까지 판매한 액수를 나타낸다. 달성도는 연간 누적 판매액을 할당액으로 나눈 값이다. 연간 누적 판매액과 달성도는 프로그램을 통해 계산될 수 있는 유도 속성이다. 마지막으로 최대 할인율은 각 영업 사원이 자신의 권한으로 제품의 판매가에서 할인해 줄 수 있는 폭을 나타낸다. 이 관계 역시 반복집단을 포함하고 있지 않으며, 부분 종속이나 이전 종속의 문제점도 가지고 있지 않다. 따라서 이 관계는 직원번호를 주키로 하는 3차 정규형의 관계이며, 다음과 같이 표현할 수 있다. 단, 유도 속성은 데이터베이스에 저장하지 않는다고 가정한다.

직원매출현황(직원번호, 성명, 할당액, 최대할인율) ..... (9)

### 3.4.2 관계의 통합

앞 절의 명문가구 사례에서 나온 9개의 정규화된 관계를 자세히 살펴보면 중복되거나 상호 보완적인 내용이 포함되어 있다. 그 이유는 한 개체를 여러 사용자의 관점에서 조명한 결과 서로 다른 속성을 지닌 관계로 표현되었기 때문이다. 본 절에서는 동일한 개체를 표현한 여러 관계를 하나의 관계로 통합하는 과정을 공부한다. 관계의 통합은 여러 사용자의 관점을 통합하는 것이므로 **관점 통합(view integration)**이라고도 불린다. 관계를 통합할 때 주의할 점

으로 이음동의어, 동음이의어, 이전 종속, 그리고 일반화의 문제점 등이 있다.

### 고객 관계

위의 9개의 관계 중 고객 개체를 표현하고 있는 것은 (4), (5), 그리고 (6)번 관계이다. (6)번 관계는 비록 고객구매현황이라고 명명되었지만, 실질적으로는 고객의 속성을 나타내는 고객 관계이다. 일반적으로 주키가 동일한 관계는 한 관계로 통합될 수 있다.<sup>2</sup> 관계 (4), (5), (6)번은 주키(고객번호)가 동일하며, 한 관계로 통합될 수 있다.

|   |     |
|---|-----|
| 고객( <u>고객번호</u> , 고객이름, 주소, 전화) .....   | (4) |
| 고객( <u>고객번호</u> , 성명, 주소, 신용한도액) .....  | (5) |
| 고객구매현황( <u>고객번호</u> , 고객이름, 담당직원) ..... | (6) |

위의 세 관계를 관찰해 보면 고객이름과 성명은 고객의 이름이라는 동일한 속성을 나타내는데, 서로 다른 이름으로 표현된 것을 알 수 있다. 이처럼 동일한 속성이 서로 다른 이름으로 불리는 경우를 이음동의어(synonym)라고 한다. 이음동의어는 하나의 명칭으로 통일하여 한 번만 저장한다. 그러나 사용자는 각자 자신이 친숙한 명칭으로 부르기 원할 수 있으므로, 데이터베이스 내에 속성의 별명(alias)을 저장해 두는 것이 바람직하다. 예를 들어, 성명과 고객이름을 고객이름으로 통일하고 성명은 별명으로 지정하여, 사용자가 원하는 경우 성명이라는 속성 이름으로도 접근이 가능하도록 하는 것이다. 대부분의 데이터베이스 관리시스템은 속성의 별명을 지정할 수 있는 기능을 제공한다. 그리고 담당직원 항목은 실질적으로 담당직원의 번호를 통하여 접근하므로 담당직원번호라는 이름으로 대체하고 외부키임을 알리는 점선으로 표시한다.

(4)번 관계와 (5)번 관계는 각각 주소라는 속성을 가지고 있다. 그러나 이 두 속성을 자세히 살펴보면, (4)번 관계의 주소는 배달 주소를 나타내는 반면, (5)번 관계의 주소는 고객의 거주지 주소를 나타낸다. 이처럼 동일한 이름으로 불리나 실제 그 내용이 다른 것을 동음이의어(homonym)라고 한다.

<sup>2</sup> 일반화 관계성 집합의 경우는 예외인데, 아래의 직원 관계에서 살펴보기로 한다.

동음이의어의 문제를 해결하기 위해서는 새로운 속성 이름을 부여하여 서로 다른 속성임을 나타내 주어야 한다. 이처럼 데이터베이스를 설계할 때에는 관계나 속성이 가지고 있는 이름이 중요한 것이 아니라 실질적인 의미를 파악하는 것이 중요하다.

이음동의어와 동음이의어의 문제가 해결되면, 나머지 중복되지 않은 속성을 모두 합쳐 관계를 통합하면 된다. (4), (5), (6)번의 관계를 하나의 관계로 통합하면 다음과 같다.

고객(고객번호, 고객이름, 거주지주소, 배달주소, 전화, 신용한도액, 담당직원번호)

고객이름과 성명은 고객이름으로 통일하였고, 주소는 각각 거주지주소와 배달주소로 구분되었다. 그리고 나머지 속성들은 중복 없이 새로운 관계로 통합되었다. 관계를 통합한 후에는 새로 생성된 관계에 이전 종속이 발생하지 않는지 확인해야 한다. 위의 통합된 관계에서는 이전 종속이 발생하지 않으므로, 고객 관계의 통합은 완료되었다.

관계 통합시 이전 종속의 문제가 발생하는 경우도 있는데, 예를 들어 다음과 같은 두 관계를 살펴보자.

학생(학번, 이름, 전공, 학년)

학생(학번, 지도교수)

두 관계 모두 학번이 주키이므로 다음과 같이 통합할 수 있다.

학생(학번, 이름, 전공, 학년, 지도교수)

그러나 만약 각 전공마다 한 명의 지도교수만 있다고 가정하면, 위의 관계에는 이전 종속이 발생한다. 즉, 전공을 알면 지도교수를 알 수 있기 때문에 이 관계는 다음과 같이 분리되어야 한다.

학생(학번, 이름, 전공, 학년)

전공지도교수(전공, 지도교수)

### 제품 관계

위의 9개의 관계 중 제품 개체를 표현하고 있는 것은 (2)번과 (7)번 관계이다. 이 두 관계의 주키는 제품번호로 동일하며, 한 관계로 통합될 수 있다.

제품(제품번호, 제품명, 단가) .....(2)

제품(제품번호, 제품명, 재고량, 재주문시점, 원가, 판매가) .....(7)

위의 두 관계의 단가와 판매가는 이음동의어이다. 두 속성의 이름을 판매가로 통일하면 실질적으로 (2)번 관계는 (7)번 관계에 완전히 포함된다. 따라서 두 관계를 통합하면 (7)번 관계가 된다.

### 직원 관계

위의 9개의 관계 중 직원 개체를 표현하고 있는 것은 (8)번과 (9)번 관계이다.

직원(직원번호, 성명, 주소, 생년월일, 연봉) .....(8)

직원매출현황(직원번호, 성명, 할당액, 최대할인율) .....(9)

이 두 관계의 주키는 직원번호로 동일하기 때문에 아래와 같이 한 관계로 통합될 수 있다.

직원(직원번호, 성명, 주소, 생년월일, 연봉, 할당액, 최대할인율)

그러나 자세히 살펴보면 앞의 사례에서 밝혔듯이 ‘할당액’과 ‘최대할인율’은 영업직원에서만 발생하는 속성이다. 따라서 위와 같이 한 관계로 통합하게 되면, 영업직원 외의 다른 직원의 경우에는 이 두 속성값이 공값이 된다. 특정 관계에 구조적으로 공값이 발생하는 것은 바람직하지 않으므로 이 관계를 보다 세분화할 수 있는데 이를 상세화(specialization)라고 한다.<sup>3</sup> 즉, 직원에서 보다 세분화된 영업직원을 분리해 내는 것인데, 모든 직원에 공통되는 속성은 직원 관계에 남겨두고, 영업직원에만 해당되는 속성을 분리하여 영업직원이란 관계를 만든다. 이때 두 관계를 연결할 수 있도록 주키는 양쪽 관계에

<sup>3</sup> 상세화의 반대 과정, 즉 세분화된 관계에서 공통 속성을 찾아 일반화된 관계를 도출하는 것을 일반화라고 한다. 상세화와 일반화에 관한 내용은 6장에서 자세히 다루기로 한다.

모두 두어야 하는데, 이를 통해 영업직원의 성명, 주소 등을 검색할 수 있다. 따라서 위의 관계를 분리하면 아래와 같다.

직원(직원번호, 성명, 주소, 생년월일, 연봉)

영업직원(직원번호, 할당액, 최대할인율)

이제 9개 관계 중 통합될 수 있는 관계는 모두 통합되었고, 최종 결과는 다음과 같은 6개의 관계로 축소되었다.

주문-제품(주문번호, 제품번호, 수량).....(1)

주문-고객(주문번호, 주문일, 고객번호).....(3)

고객(고객번호, 고객이름, 거주지주소, 배달주소, 전화, 신용한도액,

담당직원번호) .....(4), (5), (6)의 통합

제품(제품번호, 제품명, 재고량, 재주문시점, 원가,

판매가) .....(2)와 (7)의 통합

직원(직원번호, 성명, 주소, 생년월일, 연봉).....(8)

영업직원(직원번호, 할당액, 최대할인율).....(9)의 변형

### 3.5 요약

정규화 이론은 다양한 유형의 검사를 통하여, 관계형 데이터 모델의 관계를 보다 구조화된 것으로 개선시켜 나가는 절차에 관련된 이론이다. 구조화된 관계란 데이터베이스상에서 삽입, 삭제, 또는 갱신이 이루어질 때 문제점이 발생하지 않는 관계이다. 반면 정규화를 통하여 구조화된 관계는 데이터베이스의 실행속도를 떨어뜨리고, 참조 무결성과 관련된 문제를 발생시킨다. 그러나 정규화로 인하여 발생하는 문제점은 보다 우수한 성능의 컴퓨터나 정교한 데이터베이스 관리시스템을 도입함으로써 해결될 수 있기 때문에 시간이 흐를수록 정규화의 장점이 더욱 부각될 것이다.

정규화 이론은 속성간의 함수적 종속관계에 기초하고 있다. 1차 정규형

의 관계는 반복집단을 포함하지 않는 원자값으로만 구성된 타플을 갖는 관계이다. 2차 정규형의 관계는 1차 정규형의 조건을 만족하면서, 주키를 구성하지 않는 나머지 속성이 주키에 완전 함수적 종속을 이루는 관계이다. 3차 정규형의 관계는 2차 정규형의 조건을 만족하면서, 이전 종속이 존재하지 않는 관계이다. 한편 보이스-코드 정규형은 보이스와 코드가 기존의 3차 정규형에서 함수적 종속으로 인하여 추가적으로 발생할 수 있는 문제점을 발견하고, 이를 보완하기 위하여 제시한 보다 엄격한 형태의 3차 정규형이다. 보이스-코드 정규형의 관계는 어느 관계의 모든 결정자가 후보키의 역할을 수행할 수 있는 관계이다. 일반적으로 실무에서 데이터베이스를 설계할 때 요구되는 정규화의 적절한 수준은 3차 정규형 또는 보이스-코드 정규형이다. 4차 정규형의 관계는 보이스-코드 정규형을 만족시키면서, 다중값 종속을 포함하지 않는 관계이다. 마지막으로 5차 정규형의 관계는 4차 정규형의 조건을 만족시키며, 결합 종속을 포함하지 않는 관계이다.

정규화된 여러 관계가 동일한 개체를 나타내는 경우에는 서로 통합되어야 한다. 관계를 통합할 때 주의해야 할 점으로는 이음동의어, 동음이의어, 이전 종속, 그리고 일반화의 문제점 등이 있다. 이음동의어는 동일한 속성이 서로 다른 이름으로 불리는 경우로, 하나의 명칭으로 통일하여야 한다. 동음이의어는 동일한 이름으로 불리나 실제 그 내용이 다른 경우이며, 새로운 속성 이름을 부여하여 서로 다른 속성임을 나타내 주어야 한다. 이음동의어와 동음이의어의 문제가 해결되면, 나머지 중복되지 않은 속성을 모두 합쳐 관계를 통합하면 된다. 관계를 통합한 후에는 새로 생성된 관계에 이전 종속이 발생하지 않는지 확인해야 한다. 마지막으로 일반화 관계성 집합은 서로 통합될 수 없다.