

제 7 장 관계형 데이터베이스 설계 및 구현 예

2장과 6.2절의 개념적 설계와 관계형 데이터베이스 설계에서는 실세계 정의부터 각 릴레이션 정규화에 이르기까지의 각 설계 단계들을 소개하고 각 단계에서 무엇을 어떻게 해야 하는지를 소개했다.

이 장에서는 이러한 일련의 설계 과정들을 예를 가지고 6.2절에서 소개한 단계별로 보이게 한다. 이를 위해 그림 6.1에서 소개한 절차에 따라 강의 시스템을 대상으로 관계형 데이터베이스 스키마를 실제로 설계해 본다.

7.1 강의 실세계 정의

단계 1. 실세계를 시스템 측면에서 문장으로 자연스럽게 정의한다.

표 7.1은 강의시스템을 문장으로 표현한 것이다. 표 7-1은 실세계인 강의시스템의 구성요소들과 이들간의 관계를 자연스럽게 문장으로 표현한 것이다. 각 구성요소는 개체 차원이 아닌 개체 집합 차원에서 표현되었음을 주시하자. 또한 관계들은 개체 사이의 관계성이 아닌 개체 집합 사이의 관계성 집합 차원에서 표현되어야 한다. 강의시스템의 구성요소들 사이에 존재하는 규칙(제약조건)들도 문장으로 표현되었다.

표 7.1 강의 시스템 정의

강의시스템은 학과, 교수, 학생, 강좌, 교과목, 강의실, 교재 등으로 구성된다. 학과는 단과 대학에 소속되며 교수와 학생은 학과에 소속된다. 단, 교수와 학생은 단지 한 학과에만 소속되며 둘 이상의 소속관계는 허용되지 않는다. 교수는 한 강좌 이상을 담당하여 강의해야 한다. 한 강좌를 두 명이상의 교수가 강의할 수는 없다. 한 교과목에 대해 여러 개의 강좌가 개설될 수 있지만 한 강좌에 여러 개의 교과목이 대응할 수는 없다. 교수가 담당한 강좌에는 강의를 위해 이용되는 교재가 존재한다. 한 강좌에 대하여 여러 개의 교재가 이용될 수 있다. 또한 한 교재가 여러 강좌에 이용될 수도 있다. 각 강좌에는 강의실이 배정된다. 강좌에는 요일과 시간(교시)이 정해져야 한다.

표 7.1의 강의시스템 정의 문장을 강의시스템 구성요소, 구성요소 특성, 구성요소들 간의 관계, 관계 특성(관계의 속성을 의미함), 제약조건 등으로 문장들을 분류하면 다음 표 7.2와 같다.

표 7.2 강의시스템 정의 문장 분류

분류 항목	구분 번호	분류된 문장
구성요소	1	강의시스템은 학과, 교수, 학생, 강좌, 교과목, 강의실, 교재 등으로 구성 된다
구성요소 특성	2	강좌에는 요일과 시간(교시)이 정해져야 한다
구성요소들간의 관계	3	교수와 학생은 학과에 소속된다
	4	교수는 한 강좌 이상을 담당하여 강의해야 한다
	5	한 교과목에 대해 여러개의 강좌가 개설될 수 있다
	6	교수가 담당한 강좌에는 강의를 위해 이용되는 교재가 존재한다
	7	각 강좌에는 강의실이 배정된다
제약조건	8	수와 학생은 단지 한 학과에만 소속되며 둘 이상의 소속관계는 허용되지 않는다
	4	교수는 한 강좌 이상을 담당하여 강의해야 한다
	9	한 강좌를 두 명이상의 교수가 강의할 수는 없다
	5	한 교과목에 대해 여러 개의 강좌가 개설될 수 있다
	10	한 강좌에 여러 개의 교과목이 대응할 수는 없다
	11	한 강좌에 대하여 여러 개의 교재가 이용될 수 있다
	12	한 교재가 여러 강좌에 이용될 수도 있다

표 7.2에서 굵게 표시한 4번과 5번 문장들은 구성요소들 간의 관계를 나타냄과 동시에 제약 조건을 포함하고 있음을 주시하자.

7.2 강의 시스템의 개념적 설계

단계 2. 실세계를 구성하는 개체 집합을 식별하고 표현한다.

단계 3. 각 개체 집합이 강개체 집합인지 약개체 집합인지를 구분하고 표현한다.

강의시스템에서 실세계를 구성하는 개체 집합은 표 7.2에서 분류된 구성요소들이 된다. 따라서 학과, 교수, 학생, 강좌, 교과목, 강의실, 교재 등이 개체 집합으로 식별되며 2.5.6절에서 소개한 ER 모델 도형들 중 사각형으로 그림 7.1과 같이 표현된다.

개체 집합을 나타내는 ER 다이어그램

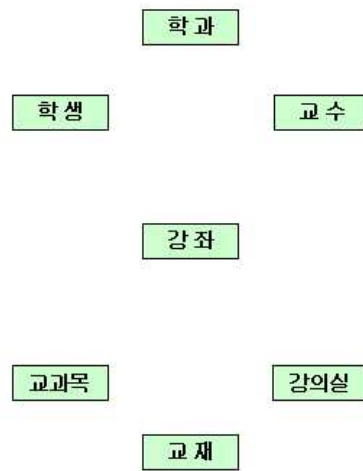


그림 7.2 개체집합들을 나타내는 ER 다이어그램

단계 4. 각 개체 집합의 속성을 식별하고 표현한다.

표 7.2에서 구분번호 2번 문장인 '강좌에는 요일과 시간(교시)이 정해져야 한다'에서 요일과 시간은 강좌 개체 집합이 갖는 속성이 된다. 데이터베이스 구축 이후의 사용(운용) 단계에서 각 속성 값들은 그 개체 집합에 대한 정보를 나타낸다. 그러므로 각 개체 집합의 속성을 식별할 때 표 7.1과 표 7.2의 강의시스템 정의 문장 중에 구성요소(즉, 개체 집합)의 특성(속성)이 언급되어 있지 않다 하더라도 구성요소(즉, 개체 집합)의 속성을 실세계 측면에서 식별하여 그 속성 정보를 참조할 필요성이 있는가를 판단하고 필요한 속성은 반드시 속성으로 나타내고 불필요한 속성들은 표현하지 않도록 한다. 이 단계에서 식별된 각 개체 집합의 속성들이 그림 7.2에서 표현된다.

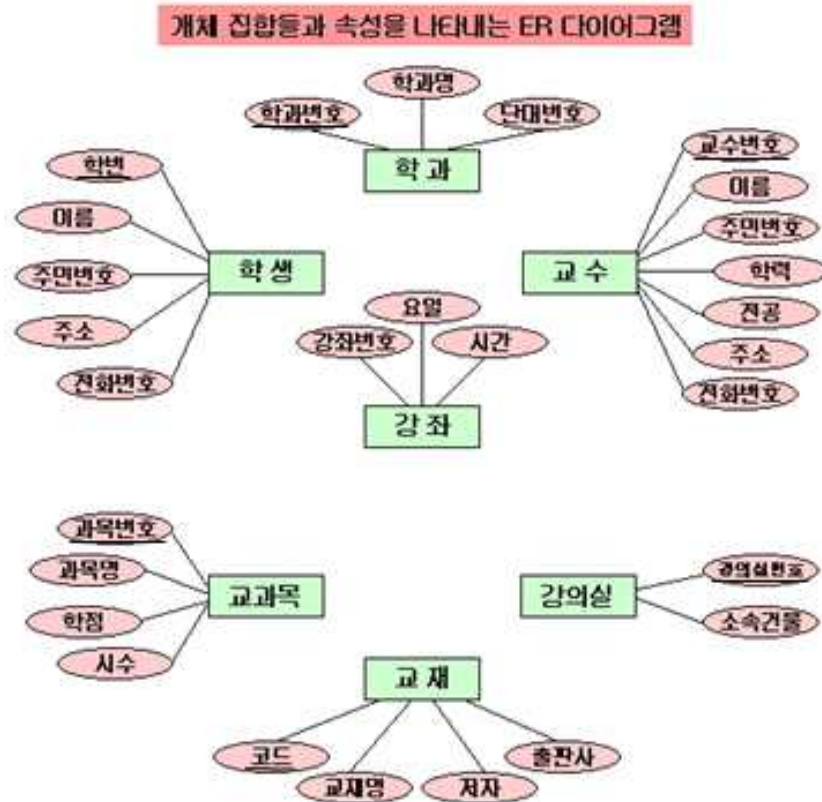


그림 7.2 개체집합들과 속성을 나타내는 ER 다이어그램

그림 7.2의 학생과 교수 개체 집합에서 소속학과에 대한 속성이 표현되지 않았음을 주시하자. 사실 상, 학생과 교수 개체 집합의 소속학과는 학과 개체 집합간의 관계에 의해 형성된다. 이 단계에서의 속성 식별은 ER 다이어그램 내에 존재하는 다른 개체 집합들과의 관계성을 고려하지 않고 개체 집합 고유의 속성만 고려한다. 즉, 한 개체 집합의 속성을 결정할 때 이 속성은 ER 다이어그램 내에 존재하는 다른 개체 집합의 고유 속성이 아닌 순수한 자기 자신의 고유 속성만 고려하면 된다.

단, 학과 개체 집합의 단대번호 속성(단과대학 번호를 의미함)의 경우 학과 개체집합 고유의 속성이 아님에도 불구하고 학과 개체 집합에 부여한 이유는 ER 다이어그램 내에 존재하지 않는 소속 단과대학 번호에 관심이 있기 때문이다. 그러나, 소속 단과대학 번호에 관심이 없거나, 혹은 만약 단과대학 구성요소가 이 ER 다이어그램 내에서 개체 집합으로 표현되어있다면 학과 개체 집합의 고유 속성이 아니고 대신 나중에 단과대학과 학과 개체 집합간의 관계성에 의해 표현되므로 학과 개체 집합의 속성으로 표현할 필요가 없다.

단계 5. 실세계로부터 개체 집합들 사이에 존재하는 관계성 집합을 식별하고 표현한다.

관계성 식별은 표 7.2에서 분류된 문장 중에서 구분번호 3, 4, 5, 6, 7번에 해당하는 구성요소들간의 관계성 문장들을 중심으로 명사(구성요소)와 서술어 또는 동사를 파악함으로써 식별할 수 있다. 예를 들어 구분번호 3번 문장인 '교수와 학생은 학과에 소속된다'는 '교수는 학과

에 소속된다' 문장과 '학생은 학과에 소속된다' 문장의 두 문장으로 분리 될 수 있으며 '교수는 학과에 소속된다' 문장에서 명사에 해당하는 교수와 학생은 이미 단계 2에서 개체 집합으로 식별되었으며, 서술어에 해당하는 '소속된다'가 바로 이 두 개체 집합 사이의 관계성 집합이 된다. 마찬가지로 '학생은 학과에 소속된다' 문장에서 '소속된다'는 학생 개체 집합과 학과 개체 집합 사이의 관계성 집합이 된다.

이러한 학생 개체 집합과 학과 개체 집합 사이의 소속 관계성 집합과 교수 개체 집합과 학과 개체 집합 사이의 소속 관계성 집합을 구별하기 위해 각각 학생소속 관계성 집합과 교수소속 관계성 집합으로 구분되어 그림 7.3의 ER 다이어그램에 반영된다.

표 7.2의 구분 번호 4번인 '교수는 한 강좌 이상을 담당하여 강의해야 한다' 문장에서 명사인 교수와 강좌는 각각 개체 집합으로 단계 2에서 ER 다이어그램으로 이미 식별되어 표현되었고, 따라서, 서술어인 '담당하여 강의해야 한다' 문장은 교수 개체 집합과 강좌 개체 집합 사이의 관계성 집합을 나타낸다. 이 관계성 집합은 강의라는 명칭을 갖는 관계성 집합으로 그림 7.3의 ER 다이어그램에 반영된다.

이와 같은 방법으로 표 7.2의 구분번호 5, 6, 7의 각 문장으로부터 명사와 서술어를 분리함으로써 관계성 집합을 쉽게 구별할 수 있으며, 이를 식별하여 그림 7.3의 ER 다이어그램에 표현하였다.

한편, 개체 집합들간의 관계성을 식별하는 과정에서 개체 집합이 강개체 집합인지 아니면 약개체 집합인지가 구분된다. 실세계에서 교과목이 있어야 이 교과목에 대한 강좌가 개설되는데, 이 때 개설 관계성 집합에서 강좌 개체 집합은 약개체 집합으로, 교과목 개체 집합은 약개체 집합이 된다.

개체 집합, 속성, 관계성을 나타내는 ER 다이어그램

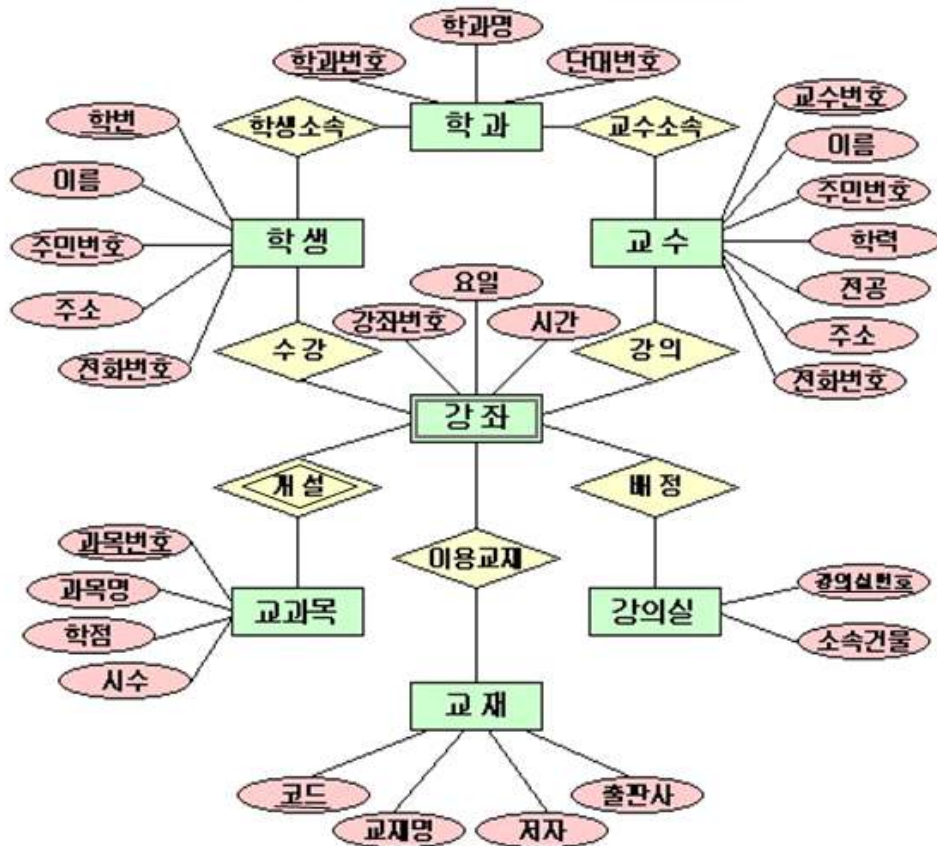


그림 7.3 개체집합, 속성, 관계성 집합을 나타내는 ER 다이어그램

단계 6. 실세계 정의의 제약사항으로부터 관계성 집합에 관련된 개체 집합들 사이의 대응 카디널리티(cardinality)를 식별하고 표현한다.

표 7.2에서 분류한 제약조건들인 구분번호 8, 4, 9, 5, 10, 11번들은 모두 대응 카디널리티(cardinality)와 관련된 문장들이다. 이 문장들을 중심으로 대응 카디널리티와 관련된 관계성 집합, 개체 집합들과 대응 카디널리티를 식별하면 표 7.3과 같다.

표 7.3 강의시스템 정의 문장 분류

관계성 집합	관련 개체 집합	대응 카디널리티 (cardinality)	제약조건 문장 구분번호
교수소속	학과 : 교수	1 : n	8
학생소속	학과 : 학생	1 : n	8
강의	교수 : 강좌	1 : n	4, 9
수강	학생 : 강좌	n : m	새로 식별됨
개설	교과목 : 강좌	1 : n	5, 10
이용교재	강좌 : 교재	n : m	11, 12
배정	강좌 : 강의실	n : m	새로 식별됨

표 7.3에서 수강 관계성 집합과 배정 관계성 집합에 대한 대응 카디널리티 제약조건은 표 7.1과 표 7.2에서 언급되지 않았음에도 불구하고 새로 식별된 것이다. 강의 시스템을 문장으로 표현할 때 제약조건이 완벽하게 식별되지 않음으로 해서 발생한 결과이다. 이 경우에는 ER 다이어그램을 표 7.1과 표 7.2를 중심으로 관계성 집합에 대한 대응 카디널리티를 식별하고 대응 카디널리티가 식별되지 않은 관계성 집합에 대해서는 실세계와 비교하여 추가적으로 대응수를 식별하도록 한다. 표 7.3의 새롭게 식별된 대응 카디널리티가 그림 7.4에 반영되었다.

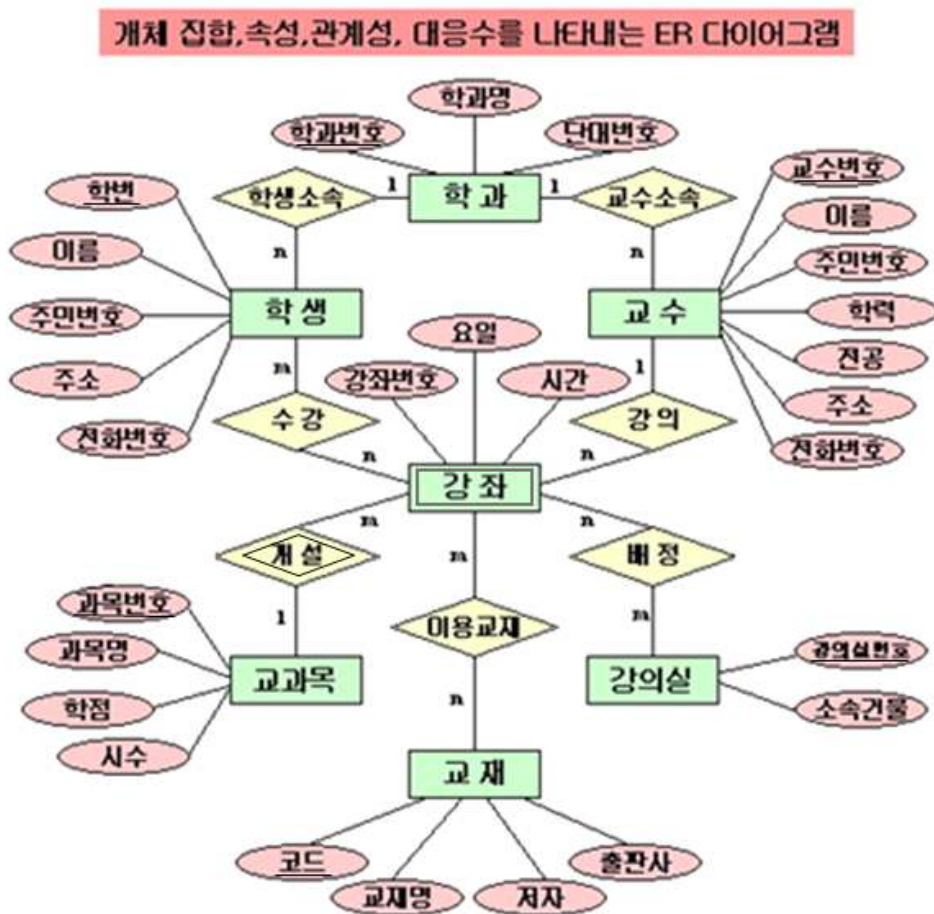


그림 7.4 개체집합, 속성, 관계성 집합, 대응수를 나타내는 ER 다이어그램

단계 7. 관계성 집합이 갖는 속성들을 식별하고 표현한다.

표 7.1과 표 7.2에서 관계성 집합에 대한 속성은 언급되지 않았다. 하지만 지금까지의 개념적 설계 결과인 ER 다이어그램의 관계성 집합을 중심으로 속성이 존재하는지 살펴보아야 한다. 그림 7.4의 개설 관계성 집합의 경우 교과목에 대한 강좌가 개설될 때 개설년도와 개설 학기가 주어지기 때문에 개설년도와 학기 속성이 개설 관계성 집합에 부여되었다. 또한, 강좌의 요일과 시간은 강의실 배정 여부에 따라 정해진다. 즉, 강좌의 요일과 시간이 정

해진다하더라도 그 시간에 해당 강의실이 다른 강좌에 배정되어 있으면 배정할 수 없다. 따라서 요일과 시간 속성은 강좌가 강의실에 배정될 때 결정됨을 알 수 있으며, 강좌의 요일과 시간 속성은 사실 상 배정 관계성의 속성들이다. 이에 따라 배정 관계성 집합은 그림 7.4에서와 같이 요일 속성과 시간 속성을 가지게 되며 강좌의 요일과 시간 속성들과 중복됨으로 강좌로부터 이 속성들을 배정 관계성 집합의 속성들로 이동해야 한다. 강좌 개체 집합의 요일과 시간 속성을 배정 관계성 집합으로 이동함을 표시하기 위해 그림 7.4에서 강좌 개체 집합의 요일과 시간 속성을 흐리게 표현했다.

개체 집합,속성,관계성, 대응수를 나타내는 ER 다이어그램

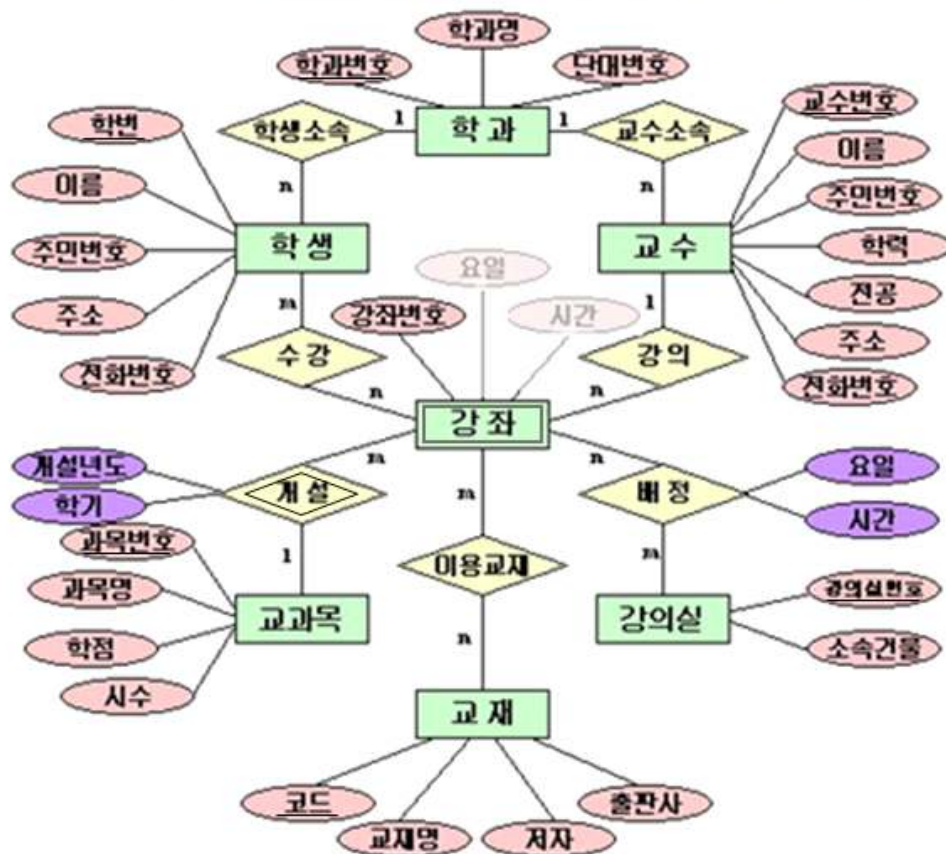


그림 7.4 관계성 집합의 속성을 표현한 최종 강의 시스템 ER 다이어그램

단계 8. 실세계 정의로부터 제약 사항을 식별하여 ER 모델로 반영하거나 ER 모델로 반영할 수 없는 제약 사항은 문장으로 정리하여 나열한다.

표 7.1과 표 7.2의 강의 시스템 문장 정의에서 식별된 대응 카디널리티에 관련한 제약 조건과 식별되지 못한 제약조건 등을 실세계를 기준으로 식별한다. 표 7.4에서 굵은 글자로 나타낸 13번과 14번 제약 조건은 새로 식별한 제약 조건을 나타낸다. 또한, 이 제약 조건이 ER 다이어그램에 반영되었는지 여부를 표시했으며 반영되지 않은 제약 조건은 관계형 데이터베이스를

구현할 때 제약 조건으로 반영하거나 아니면 트랜잭션나 응용 프로그램에 반영해야 한다.

표 7.4 새로 식별된 제약 조건을 반영한 강의시스템 제약조건

구분번호	제약 조건 문장	ER 다이어그램 반영 여부
8	교수와 학생은 단지 한 학과에만 소속되며 둘 이상의 소속관계는 허용되지 않는다	반영
4	교수는 한 강좌 이상을 담당하여 강의해야한다	반영
9	한 강좌를 두 명이상의 교수가 강의할 수는 없다	반영
5	한 교과목에 대해 여러 개의 강좌가 개설될 수 있다	반영
10	한 강좌에 여러 개의 교과목이 대응할 수는 없다	반영
11	한 강좌에 대하여 여러 개의 교재가 이용될 수 있다	반영
12	한 교재가 여러 강좌에 이용될 수도 있다	반영
13	교수의 학력은 박사 또는 석사이어야 한다	미반영
14	학생은 한학기에 7개 이상의 강좌를 들을 수 없다	미반영

표 7.4에서 구분번호 13번과 14번 제약조건은 강의 시스템 문장 정의에서 식별되었어야 한다. 그러나 처음부터 완벽하게 실세계를 정의할 수 없기 때문에 ER 다이어그램을 중심으로 검토하면서 제약조건들을 식별한다(그러나, 강의 시스템 문장 정의는 처음부터 완벽하게 작성하도록 노력해야 한다).

개념적 설계 단계에서의 결과는 그림 7.4의 ER 다이어그램과 표 7.4의 제약조건이다.

7.3 강의 시스템의 논리적 설계

7.1절에서 도출된 그림 7.4의 ER 다이어그램과 표 7.4의 제약 조건을 가지고 논리적 설계를 행한다. 특히, 그림 7.4의 강의 시스템 ER 다이어그램이 릴레이션을 도출하기 위해 사용된다. 관계형 데이터베이스 스키마 도출을 위한 논리적 설계는 6.2.3절에서 소개된 다음의 단계 9와 단계 10를 행함으로써 이루어진다.

단계 9. ER 다이어그램을 릴레이션으로 변환하고 변환된 릴레이션에 대한 주기를 설정한다.

단계 10. ER 다이어그램으로부터 변환된 각 릴레이션에 대하여 정규화를 행한다.

6.2.3절에서 소개했듯이 그림 7.4의 ER 다이어그램에서 변환해야 할 대상은 개체 집합과 관계성 집합이다. 이 ER 다이어그램에서 개체 집합과 관계성 집합을 릴레이션으로 변환하는 과정을 각각 7.3.1절과 7.3.2절에서 각각 소개하고 변환된 릴레이션에 대한 정규화 과정은 7.3.3절에서 소개한다.

7.3.1 개체 집합의 릴레이션 변환

6.2.3.1절의 규칙 1과 6.2.3.2절의 규칙 2는 각각 강개체 집합과 약개체 집합을 릴레이션으로 변환하는 방법이다. 그림 7.4의 강의 시스템 ER 다이어그램에서 강좌 개체 집합을 제외한 개체 집합들이 강개체 집합이므로 규칙 1에 따라 각 개체 집합을 같은 이름의 독립된 릴레이션으로 대응시키고 개체 집합의 속성들 또한 대응하는 릴레이션에서 동일한 이름으로 정한다. 또한, 개체 집합의 주키를 릴레이션의 주키로 한다.

한편, 강좌 개체 집합은 약개체 집합이므로 6.2.3.2절의 규칙 2를 적용시키면 강좌 개체 집합을 릴레이션으로 변환한 강좌(강좌번호) 스키마에 개설 관계성 집합에 참여하는 교과목 강개체 집합의 주키 속성 과목번호를 포함시켜 릴레이션 강좌(과목번호, 강좌번호)로 변환한다. 이때 주키는 강개체 집합의 주키 속성과 약개체 집합의 구별자 속성이므로 {과목번호, 강좌번호}가 된다.

이와 같이해서 도출한 릴레이션은 표 7.5와 같고 릴레이션의 주키는 밑줄로 표시되었다.

표 7.5 강의 시스템 ER 다이어그램에서 개체 집합을 변환한 릴레이션

릴레이션	설명
학과(<u>학과번호</u> , 학과명, 단대번호)	학과 개체 집합으로부터 변환
교수(<u>교수번호</u> , 이름, 주민번호, 학력, 전공, 주소, 전화번호)	교수 개체 집합으로부터 변환
학생(<u>학번</u> , 이름, 주민번호, 주소, 전화번호)	학생 개체 집합으로부터 변환
교과목(<u>과목번호</u> , 과목명, 학점, 시수)	교과목 개체 집합으로부터 변환
강좌(<u>과목번호</u> , <u>강좌번호</u>)	강좌 개체 집합으로부터 변환
교재(<u>코드</u> , 교재명, 저자, 출판사)	교재 개체 집합으로부터 변환
강의실(<u>강의실번호</u> , 소속건물)	강의실 개체 집합으로부터 변환

7.3.2 관계성 집합의 릴레이션 변환

표 7.3으로부터 강의 시스템에 대한 관계성 집합을 정리하면 표 7.6과 같다.

표 7.6 강의 시스템에 대한 관계성 집합

관계성 집합	관련 개체 집합	대응 카디널리티 (cardinality)
교수소속	학과 : 교수	1 : n
학생소속	학과 : 학생	1 : n
강의	교수 : 강좌	1 : n
수강	학생 : 강좌	n : m
개설	교과목 : 강좌	1 : n
이용교재	강좌 : 교재	n : m
배정	강좌 : 강의실	n : m

표 7.6의 각 관계성 집합에 6.2.3.3절의 규칙들을 적용하는 과정을 살펴보자.

(1) 교수소속 관계성 집합의 변환

표 7.6에서 교수소속 관계성 집합은 학과 개체 집합과 교수 개체 집합 사이에서 1:n 대응관계를 갖는다. 우선 모든 관계성 집합에 대해 규칙 3을 적용해야 하므로 규칙 3을 적용하면 다음의 릴레이션이 도출된다.

교수소속(학과번호, 교수번호)

게다가 실세계에서 모든 교수는 학과에 소속되므로 규칙 3.1.1의 조건에 해당하며 규칙 3.2 조건 또한 만족하므로 먼저 규칙 3.1.1을 적용하면 앞의 교수소속(학과번호, 교수번호) 릴레이션 스키마는 삭제되면서 학과번호 속성을 표 7.5의 교수 릴레이션에 포함시켜 교수 릴레이션의 스키마가 다음과 같이 변경된다.

교수(교수번호, 이름, 주민번호, 학력, 전공, 주소, 전화번호, 학과번호)

이 릴레이션에 규칙 3.2를 적용하면 교수소속 관계에서 n에 해당하는 개체 집합의 주기가 되므로 주기가 설정된 교수 릴레이션은 다음과 같다.

교수(교수번호, 이름, 주민번호, 학력, 전공, 주소, 전화번호, 학과번호)

이 교수 릴레이션에서 학과번호는 학과 릴레이션을 참조하는 외래키가 된다.

(2) 학생소속 관계성 집합의 변환

표 7.6에서 학생소속 관계성 집합 또한 학과 개체 집합과 학생 개체 집합 사이에서 1:n 대응관계를 갖는다. 또한, 모든 학생은 어느 한 학과에 속해 관계성을 가지므로 교수소속 관계성 집합과 마찬가지로 규칙 3, 규칙 3.1.1, 규칙 3.2 조건에 해당하며 이 규칙들을 차례로 적용하면 학생 릴레이션 스키마가 다음과 같이 변경된다.

학생(학번, 이름, 주민번호, 주소, 전화번호, 학과번호)

변경된 학생 릴레이션 스키마에서 학과번호는 학과 릴레이션을 참조하는 외래키이다.

(3) 강의 관계성 집합의 변환

표 7.6에서 강의 관계성 집합은 교수와 강좌 개체 집합 사이에서 1:n 대응관계를 갖는다. 또한, 표 7.4의 9번 문장에서 한 강좌를 두 명이상의 교수가 강의할 수는 없고 실세계에서 모든 강좌는 강의하는 교수가 존재하므로 교수소속과 학생소속 관계성 집합과 마찬가지로 규칙 3, 규칙 3.1.1, 규칙 3.2 조건에 해당한다. 이들 규칙들을 차례로 적용하면 강좌 릴레이션은 다음

과 같이 변경된다.

[강좌\(과목번호, 강좌번호, 교수번호\)](#)

이 강좌 릴레이션에서 교수번호 속성은 교수 릴레이션을 참조하는 외래키이다.

(4) 수강 관계성 집합의 변환

수강 관계성 집합은 표 7.6에서 학생과 강좌 개체 집합 사이에서 $m:n$ 대응 관계를 갖는다. 따라서, 조건에 맞는 규칙 3과 규칙 3.3을 적용하면 아래와 같은 새로운 수강 릴레이션이 도출되며, 규칙 3.3을 적용하면 수강 릴레이션의 주키는 학생 개체 집합의 주키인 {학번} 속성과 강좌 개체 집합의 주키인 {과목번호, 강좌번호}를 합친 {학번, 과목번호, 강좌번호}가 된다.

[수강\(학번, 과목번호, 강좌번호\)](#)

이 수강 릴레이션에서 학번 속성은 학생 릴레이션을 참조하는 외래키이고 과목번호, 강좌번호 속성은 강좌 릴레이션을 참조하는 외래키이다.

(5) 개설 관계성 집합의 변환

그림 7.4의 개설 관계성 집합에서 강개체 집합은 교과목과 약개체 집합인 강좌 사이에서 $1:n$ 대응 관계를 갖는다. 따라서, 규칙 3을 적용한 후 이 조건에 해당하는 규칙 3.2.1을 적용하면 새로운 릴레이션은 생성되지 않지만 개설 관계성 집합의 속성들인 개설년도와 학기는 6.2.3.2절의 규칙 2에 의해 이미 변환된 약개체 집합에 대응하는 릴레이션에 추가된다. 6.2.3.2절의 규칙 2를 적용하여 약개체 집합을 릴레이션으로 변환하는 과정에서 강개체 집합의 주키를 외래키로 포함시키고 이 외래키와 약개체 집합의 구별자를 합쳐서 약개체 집합의 주키가 됨을 6.2.3.2절과 7.3.1절을 통해 이미 소개되었다. 이렇게 해서 변경된 강좌 릴레이션은 다음과 같다.

[강좌\(과목번호, 강좌번호, 교수번호, 개설년도, 학기\)](#)

(6) 이용교재 관계성 집합의 변환

그림 7.4와 표 7.6에서 이용교재 관계성 집합은 강좌와 교재 개체 집합 사이에 $m:n$ 대응 관계를 갖는다. 이 대응 관계는 수강 관계성 집합과 마찬가지로 규칙 3과 규칙 3.3의 조건에 해당하므로 이 규칙들을 적용하여 새로운 릴레이션을 생성한다. 강좌 개체 집합은 약개체 집합이므로 주키는 {과목번호, 강좌번호}이고 강개체 집합인 교재의 주키는 {코드}이므로 새로운 릴레이션은 이용교재 명칭으로 아래와 같이 도출된다.

[이용교재\(과목번호, 강좌번호, 코드\)](#)

(7) 배정 관계성 집합의 변환

그림 7.4와 표 7.6에서 배정 관계성 집합도 역시 강좌와 강의실 개체 집합 사이에 n:m 대응 관계를 갖는다. 이 대응 관계는 수강과 이용교재 관계성 집합과 마찬가지로 규칙 3과 규칙 3.3의 조건에 해당하므로 이 규칙들을 적용하여 새로운 릴레이션을 생성한다. 이 규칙들을 적용하면 강좌 개체 집합은 약개체 집합이므로 주키는 {과목번호, 강좌번호}이고 강개체 집합인 강의실의 주키는 {강의실번호}이므로 새로운 릴레이션은 배정 명칭으로 아래와 같이 도출된다.

배정(과목번호, 강좌번호, 강의실번호)

7.3.3 개체 집합과 관계성 집합을 변환한 릴레이션

그림 7.4의 최종 강의 시스템 ER 다이어그램에서 7.3.1절에서 개체 집합의 릴레이션 변환과 7.3.2절에서 관계성 집합의 릴레이션 변환을 적용한 후에 도출된 릴레이션을 정리하면 표 7.7과 같다.

표 7.7 그림 7.4의 강의 시스템 ER 다이어그램을 변환한 릴레이션

릴레이션	외래키와 참조 릴레이션	
	외래키	참조 릴레이션
학과(학과번호, 학과명, 단대번호)	none	none
교수(교수번호, 이름, 주민번호, 학력, 전공, 주소, 전화번호, 학과번호)	학과번호	학과
학생(학번, 이름, 주민번호, 주소, 전화번호, 학과번호)	학과번호	학과
교과목(과목번호, 과목명, 학점, 시수)	none	none
강좌(과목번호, 강좌번호, 교수번호, 개설년도, 학기)	과목번호	교과목
교재(코드, 교재명, 저자, 출판사)	none	none
강의실(강의실번호, 소속건물)	none	none
수강(학번, 과목번호, 강좌번호)	학번	학생
	과목번호	교과목
	강좌번호	강좌
이용교재(과목번호, 강좌번호, 코드)	과목번호	교과목
	강좌번호	강좌
	코드	교재
배정(과목번호, 강좌번호, 강의실번호)	과목번호	교과목
	강좌번호	강좌
	강의실번호	강의실

7.3.4 각 릴레이션에 대한 정규화

관계형 데이터베이스의 논리적 설계에서 ER 모델을 이용하여 도출된 릴레이션은 일반적으로 제 3 정규형을 만족한다. 그럼에도 불구하고 도출된 각 릴레이션에 대해 정규형을 따져서 정규화를 행할 필요가 있는데 그 예를 들면 다음과 같다.

표 7.7에서 도출한 모든 릴레이션 각각은 모든 속성이 단일 값을 가지므로 제 1 정규형에 있고 각 속성이 주키인 학번 속성에 완전 함수적 종속이므로 제 2 정규형에 있으며 속성들 사이에 전이 종속 관계가 없으므로 제 3 정규형에 속한다. 또한, 각 속성의 결정자(determinant)가 주키인 학번 속성이므로 BCNF에 존재한다.

그러나, 만약 학생 릴레이션이 그 변환의 모체인 학생 개체 집합에 가족명 속성과 관계속성, 가족주민번호 속성을 포함하는 복합 속성인 가족관계 속성이 추가로 부여된다면 변환에 의해 도출된 릴레이션은 다음과 같이 된다.

학생 릴레이션

학번	이름	주민 번호	주소	전화 번호	가족관계			학과번호
					가족명	관계	가족주민번호	

이 릴레이션의 경우 가족관계 속성은 복합 속성이면서 가족이 두 명 이상일 경우에는 가족 수에 따라 여러 사람에 대한 속성 값들을 갖기 때문에 반복 그룹에 해당하므로 제 1 정규형에 해당하지 않는다. 따라서 ER 모델을 통해 도출된 릴레이션일지라도 정규형을 체크하고 그 정규형에 속하지 않을 경우 무손실 분해를 통한 정규화를 행할 필요가 있다.

7.3.5 관계형 데이터베이스 논리적 설계의 결과물

강의 시스템 ER 다이어그램을 이용한 관계형 데이터베이스의 논리적 설계의 결과물은 7.3.4절의 정규화를 거친 표 7.7의 릴레이션들과 표 7.4절의 ER 다이어그램에서 미반영된 제약 조건들이다.

표 7.7절의 최종 각 릴레이션들은 SQL 언어를 이용하여 이 스키마를 갖는 릴레이션을 생성함으로써 구현되는데 각 릴레이션을 생성할 때 표 7.4절의 미반영된 제약 조건들을 반영가능하면 된다. 릴레이션을 생성할 때 반영할 수 없는 제약 조건들은 트랜잭션(데이터베이스 상태를 변경하는 단위 프로그램)이나 응용 프로그램에 반영하여 무결성이 보장되도록 한다.

7.4 관계형 데이터베이스 구현

7.3.5절에서 언급한 결과물들인 최종 릴레이션 스키마들과 설계에서 아직 미반영된 제약 조건들을 가지고 최종 릴레이션들에 해당 제약 조건을 갖는 릴레이션을 생성함으로써 관계형 데이터베이스가 구현된다. 표 7.4의 제약 조건들 중에서 관계형 데이터베이스에 반영되지 못한 제약 조건들은 트랜잭션이나 응용 프로그램에 반영해야 한다.

7.4.1 릴레이션 생성

최종으로 도출된 표 7.7의 각 릴레이션과 반영가능한 표 7.4의 제약 조건을 갖춘 릴레이션을 SQL로 생성하면 다음과 같다.

- **학과 릴레이션 생성** : 표 7.7에 따라 학과 릴레이션은 다음과 같이 생성된다. 여기에서 학과 번호 속성이 주키이므로 not null로 선언된다.

```

create table 학과
(   학과번호      char(8) not null,
    학과명        char(14),
    단대번호      int(3),
    primary key(학과번호) )

```

- **교수 릴레이션 생성** : 교수 릴레이션 역시 표 7.7의 교수 릴레이션 스키마에 따라 생성된다. 교수번호 속성이 주키이므로 primary key 선언과 함께 not null로 선언되며 표 7.4의 교수의 학력은 '박사' 혹은 '석사'이어야 한다는 제약 조건을 check 도메인 제약 조건으로 반영하였다.

```

create table 교수
(   교수번호      char(5) not null,
    이름          char(10),
    주민번호      char(14),
    학력          char(8),
    전공          char(16),
    주소          char(30),
    전화번호      char(11),
    학과번호      char(8),
    primary key(교수번호),
    check(학력='박사' or 학력='석사') )

```

- **학생 릴레이션 생성** : 학생 릴레이션도 표 7.7의 학생 릴레이션 스키마에 따라 생성된다. 학번 속성이 주키이므로 primary key 선언과 함께 not null로 선언된다.

```

create table 학생
(   학번          char(5) not null,
    이름          char(10),
    주민번호      char(14),
    주소          char(30),
    전화번호      char(11),
    학과번호      char(8),
    primary key(학번) )

```

- **교과목 릴레이션 생성** : 표 7.7의 교과목 릴레이션 스키마를 갖는 교과목 릴레이션이 다음

과 같이 생성된다 주키는 과목번호이므로 not null로 선언되며 학점과 시수가 각각 3학점과 4시간 이하로 check 구문에 의해 도메인 제약 조건으로 새로 반영되었음을 주시하자.

```
create table 교과목
(   과목번호      char(5) not null,
    과목명        char(10),
    학점          int(1),
    시수          int(1),
    primary key(과목번호),
    check (학점>=1 and 학점<=3),
    check (시수>=1 and 시수<= 4) )
```

- **강좌 릴레이션 생성** : 강좌 릴레이션이 표 7.7의 강좌 릴레이션 스키마에 따라 생성된다. 주키는 {과목번호, 강좌번호} 속성들이므로 not null로 선언되며 과목번호 속성과 교수번호 속성은 각각 교과목 릴레이션과 교수 릴레이션의 주키이므로 이 릴레이션들을 참조하는 외래키로 선언된다.

```
create table 강좌
(   과목번호      char(5) not null,
    강좌번호      char(7), not null,
    교수번호      char(5),
    개설연도      int(4),
    학기          int(1),
    primary key(과목번호, 강좌번호),
    foreign key(과목번호) references 교과목,
    foreign key(교수번호) references 교수,
    check (학기 >= 1 and 학기 <= 2) )
```

- **교재 릴레이션 생성** : 교재 릴레이션 역시 표 7.7의 교재 릴레이션 스키마에 따라 선언되며 코드 속성이 주키로서 not null로 선언된다.

```
create table 교재
(   코드          char(6) not null,
    교재명        char(16),
    저자          char(10),
    출판사,       char(16),
    primary key(코드) )
```

- **강의실 릴레이션 생성** : 강의실 릴레이션도 표 7.7의 강의실 릴레이션 스키마에 따라 선언되며 강의실번호 속성이 주키로서 not null로 선언된다.

```
create table 강의실
(   강의실번호      char(5) not null,
    소속건물        char(14),
    primary key(강의실번호) )
```

- **수강 릴레이션 생성** : 표 7.7의 수강 릴레이션 스키마에 따라 수강 릴레이션이 선언된다. 주키는 {학번, 과목번호, 강좌번호} 속성들로 모두 not null로 선언되며 이들 속성은 외래키로서 학생 릴레이션과 강좌 릴레이션을 참조함을 주시하자.

```
create table 수강
(   학번            char(5) not null,
    과목번호        char(5) not null,
    강좌번호        char(7) not null,
    primary key(학번, 과목번호, 강좌번호),
    foreign key(학번) references 학생,
    foreign key(과목번호, 강좌번호) references 강좌 )
```

- **이용교재 릴레이션 생성** : 표 7.7의 이용교재 릴레이션 스키마에 따라 이용교재 릴레이션이 선언된다. 주키는 {과목번호, 강좌번호, 코드} 속성들로 모두 not null로 선언되며 이들 속성은 외래키로서 강좌 릴레이션과 교재 릴레이션을 참조하는 것으로 선언함으로써 데이터베이스 관리시스템에 의해 참조무결성을 자동으로 보호할 수 있도록 한다.

```
create table 이용교재
(   과목번호        char(5) not null,
    강좌번호        char(7) not null,
    코드            char(6) not null,
    primary key(과목번호, 강좌번호, 코드),
    foreign key(과목번호, 강좌번호) references 강좌,
    foreign key(코드) references 교재 )
```

- **배정 릴레이션 생성** : 표 7.7의 배정 릴레이션 스키마에 따라 배정 릴레이션이 선언된다. 주키는 {과목번호, 강좌번호, 강의실번호} 속성들로 모두 not null로 선언한다. 이들 속성은 외래키로서 강좌 릴레이션과 강의실 릴레이션을 참조하는 것으로 선언함으로써 데이터베이스관

리시스템에 의해 참조무결성을 자동으로 보호할 수 있도록 한다.

```
create table 배정
(
    과목번호      char(5) not null,
    강좌번호      char(7) not null,
    강의실번호    char(6) not null,
    primary key(과목번호, 강좌번호, 강의실번호),
    foreign key(과목번호, 강좌번호) references 강좌
    foreign key(강의실번호) references 강의실 )
```

7.4.2 제약 조건의 반영

표 7.4에서 마지막으로 남은 “학생은 한 학기에 7개 이상의 강좌를 들을 수 없다”는 14번 제약 조건을 데이터 무결성을 위해 반영해야 한다. 이에 대한 반영은 세 가지 방법으로 반영할 수 있다.

첫 번째 방법은 trigger를 선언하는 방법이다. trigger 선언의 내용은 수강 릴레이션에 수강 튜플이 삽입되기 전에 그 튜플의 학번 속성값을 이용하여 그 속성값을 갖는 튜플의 개수를 count 함수로 구하고 그 수가 6이면 삽입을 거부하는 것이다. 이렇게 하면 수강 릴레이션에 튜플이 삽입될 때 마다 DBMS에 의해 자동으로 검사되어 제약 조건이 유지된다.

두 번째 방법은 데이터베이스를 접근하는 트랜잭션에 반영하는 것이다. Trigger 자체를 트랜잭션 내부에서 선언하여 사용할 수도 있고 아니면 제약 조건을 체크하고 조건에 따라 처리하는 내용을 트랜잭션 프로그램에 반영하여 처리할 수도 있다.

세 번째 방법은 제약 조건을 응용 프로그램에 반영하는 방법이다. Trigger와 트랜잭션의 경우는 이들 모두 DBMS에 의해 수행되므로 빈번한 삽입으로 인해 매번 제약 조건을 체크하는 작업이 여러 트랜잭션들과 응용 프로그램들이 공유하는 DBMS에 부하를 증가시키는 요인이 될 수 있다. 하지만 응용 프로그램의 경우 조건 체크가 DBMS와는 독립적으로 응용 프로그램에서 직접 행해지기 때문에 DBMS에 부하를 줄일 수 있는 방법이기도 하다.

Trigger나 트랜잭션을 이용할 것인지 아니면 응용 프로그램을 이용할 것인지 결정하는 문제는 제약 조건의 복잡도와 체크 빈도수가 DBMS의 부하에 미치는 영향 등을 고려하여 판단해야 한다. 모든 제약 조건을 이들 방법 중의 하나로 선택하여 구현하는 것은 아니며 일부는 Trigger, 일부는 트랜잭션에, 나머지 일부는 응용 프로그램에 나누어 반영할 수도 있고 아니면 이들 세 가지 중 두 가지로 나누어 반영할 수도 있다.