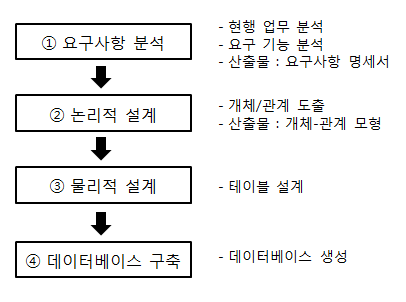
* **데이터 모델링**

데이터베이스를 새롭게 구축하기 위해서는 사용자의 요구 사항을 듣고 사용자의 관점을 분석하여 이를 추상화하여 문서화하는 과정인 데이터 모델링 과정을 거쳐야 한다.

데이터 모델은 현실 세계를 데이터베이스에 표현하는 중간 과정, 즉 데이터베이스 설계과정에서 데이터의 구조를 이해하기 위해 사용하는 도구이다.

현실 세계를 데이터베이스로 표현하기 위해서는 적어도 개념적인 구조와 논리적인 구조를 거쳐 실제 데이터를 저장할 수 있는 물리적인 구조로 변환 되어야 하는데 이 모든 과정을 총칭하여 ‘데이터베이스 설계’라 하고 이 과정에서 중요한 핵심은 ‘모델’이며 이 모델은 그 구성 요소를 추상적 개념으로 하느냐 또는 논리적인 개념으로 하느냐에 따라 개념적 데이터 모델과 논리적 데이터 모델로 구분하다.



* **요구사항 분석**

데이터 모델링을 하기 위해서는 사용자와의 면담을 통해서 얻은 내용으로 사용자의 요구 사항을 간단하게 기술해 놓아야 한다. 요구 사항 명세서나 업무 흐름도가 최종 산출물이 된다.

서점 관리를 예를 들어 요구 사항 분석을 해 보도록 하자.

1. 회원들이 도서를 주문하는 작업이 주 업무이다.
2. 업무 처리에 있어 주된 자료는 회원 정보와 도서 정보이며 주문에 따라 발생되는 정보이다.
3. 서점 관리 시스템 개발의 주된 목적은 기존 서점 관리 업무를 데이터베이스로 구축하여 전산 처리가 가능하도록 하는 것이다.

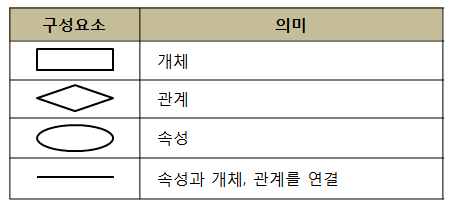
* **개체 관계 모델(논리적 설계 단계)**

논리적 설계 단계는 얻어진 산출물을 근거로 데이터 모델링을 해야 하는데 데이터 모델링은 업무 처리에 필요한 자료와 속성을 기술하고, 자료 간의 관계를 정의하고 특정 데이터베이스에 구애되지 않고 골격을 구축하는 작업 단계이다. 이 단계에서 모델링 기법이 제공되어야 하는데 가장 일반적으로 사용되는 모델링 기법으로 개체 관계 모델(Entity Relationship Diagram)이 있다. ERD는 데이터베이스의 전체 구조를 쉽게 나타낼 수 있으며, 사용자와 함께 업무를 분석하고 검토할 때도 이용한다. 개체(Entity)와 개체 간의 관계를 이해하고 쉽게 정해진 표기법에 따라 그림으로 표시한다.

데이터 모델링을 하는 과정에서 결정되어야 할 사항은 구조, 연산, 제약 조건이다.

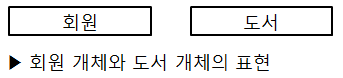
|  |  |
| --- | --- |
| 구조 | 데이터베이스에 표현될 대상으로서 개체 타입과 이들간의 관계를 명세한 것이다. |
| 연산 | 데이터베이스에 표현된 개체 인스턴스를 처리하는 작업에 대하여 명세한 것이다. |
| 제약조건 | 데이터베이스에 허용될 수 있는 개체 인스턴스에 대한 논리적 제약을 명세한 것이다. |

객체 관계 모델은 객체(Entity), 속성(Attribute), 관계(RelationShip)와 같은 개념을 사용하여 실생활을 데이터베이스화하는 것이다. 다음은 개체 관계 모델을 구성하는 요소이다.



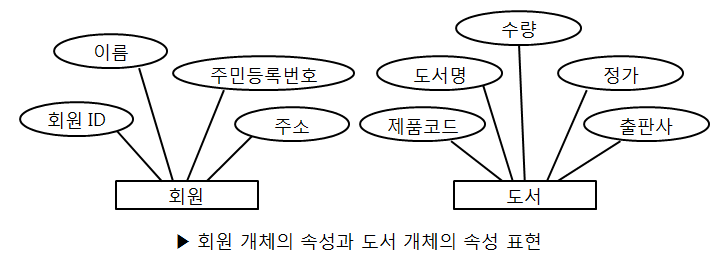
* **개체**

개체(Entity)는 실 세계에서 기본적으로 표현할 수 있는 항목으로 레코드와 같은 의미이다. 관리 대상이 되는 독립적 특성을 갖는 사람, 사물, 사건, 개념 등 요구사항 명세서에서 명사형으로 표현되는데, ERD에서 사각형으로 표시한다.



* **속성**

속성(Attribute)은 성질이나 상태 등 개체가 가질 수 있는 세부적인 특성으로 타원으로 표현되며, 개체에 실선으로 연결하여 나타낸다. 데이터의 가장 작은 논리적인 단위로서 필드와 같은 의미이다.



* **인스턴스**

인스턴스(Instance)는 개체의 구체적인 하나의 예, 즉 단일 사례로서 개체는 인스턴스의 집합이다.

<회원 인스턴스>

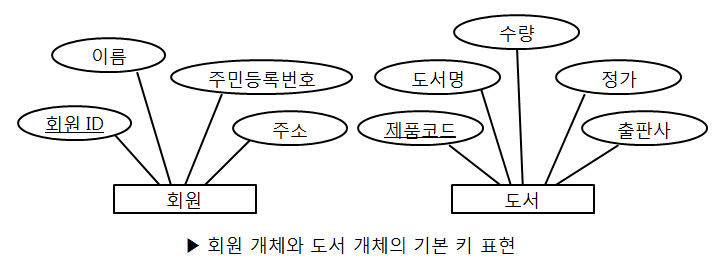
|  |  |  |  |
| --- | --- | --- | --- |
| 회원ID | 이름 | 주민등록번호 | 주소 |
| eunhee | 조은희 | 7302251346581 | 대구 |
| ginsoon | 장길순 | 8508153214567 | 충남 |
| gildong | 홍길동 | 0001112354894 | 경기 |

<도서 인스턴스>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 제품코드 | 도서명 | 수량 | 가격 | 출판사 |
| 1 | 베이비 토크 | 25 | 13,000 | 마고북스 |
| 2 | 우리만의 커플여행 | 32 | 9,000 | 중앙 |
| 3 | 선물 | 50 | 8,500 | 중앙M&B |

* **기본 키**

하나의 개체 내에 똑같은 인스턴스가 존재하면 서로 구별하여 검색할 수 없고, 데이터 중복 문제가 발생한다. 인스턴스들을 서로 구별할 수 있는 유일한 값을 갖는 속성이 필요하다. 이 속성이 바로 기본 키(Primary Key)이다. 한 개 이상의 속성 조합으로도 구성된다. 기본 키는 속성에 밑줄을 표시한다.



* **관계**

관계(Relationship)는 개체간의 연관성을 말하며, 요구사항 명세서에서 동사 형으로 표현된다. 예를 들면 고객과 상품 사이에는 고객은 상품을 주문한다. 상품은 고객이 주문 한다.

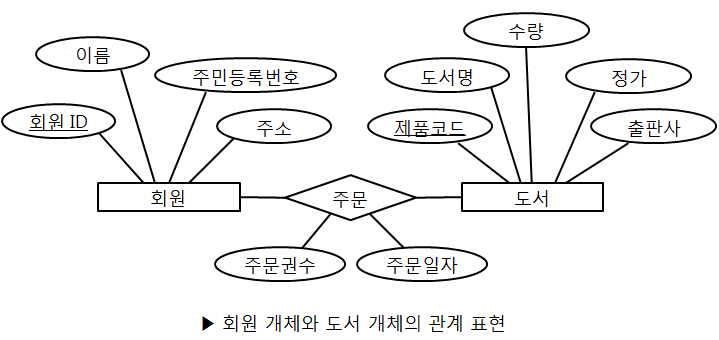
고객과 상품 사이에 ‘주문 관계’가 있다.



사원과 부서 사이에는 ‘소속 관계’가 있다

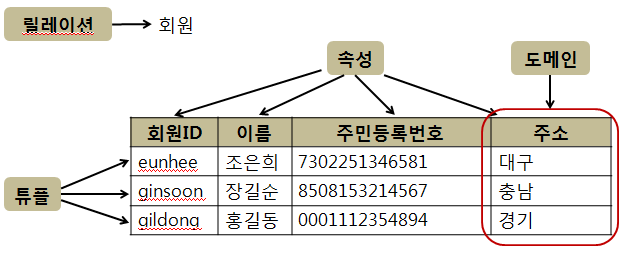


관계가 있는 개체를 실선으로 연결하고 마름모 사각형 안에 관계를 표시한다.

****

**릴레이션**

모델링 과정에서 테이블을 릴레이션 이라고 한다. 릴레이션은 튜플의 집합으로 표현한다.



속성은 릴레이션에서 열의 이름을 말한다. 도메인은 하나의 속성이 취할 수 있는 같은 타입의 모든 원자 값들의 집합이다. 튜플은 릴레이션에서 한 행을 말한다.

* **사례연구**

다음과 같은 조건에 따라 ERD를 그려보도록 하자.

|  |
| --- |
| 1. A택배회사의 직원은 입사하면서 직원 한 명당 한대의 자동차를 할당 받게 된다. 직원은 직원 번호, 이름, 입사일을 기록하며, 할당 받은 자동차는 자동차 번호, 색상, 모델명을 기록하게 된다. |

<풀이과정(1:1 관계)>

1. 우선 위 글에서 테이블에 해당되는 것은 네모, 관계에 해당되는 것은 마름모, 속성에 해당되는 것은 타원으로 표현한다.
2. 이를 ERD로 그린다.

|  |
| --- |
| 2. B유통회사는 한 사람의 고객이 여러 가지 주문에 대한 관계를 기록하는데, 고객에는 고객번호, 이름, 주소를 관리하고 주문에는 주문 번호, 주문일, 주문량을 기록한다. |

<풀이과정(1:m 관계)>

1. 우선 위 글에서 테이블에 해당되는 것은 네모, 관계에 해당되는 것은 마름모, 속성에 해당되는 것은 타원으로 표현한다.
2. 이를 ERD로 그린다.

|  |
| --- |
| 3. 어느 회사에서 프로젝트 배정 현황에 관한 데이터베이스를 구축하고자 한다. 아래와 같은 상황을 ERD로 표현하시오.  1) 각 직원은 하나 또는 그 이상의 프로젝트에 배정될 수 있으며 프로젝트가 없는 직원도 있을 수 있다. 그러나 각 프로젝트는 반드시 한 명 이상의 직원이 배정되어야 한다.  2) 직원은 이름, 호봉, 특기, 생년월일을 속성으로 가지며, 프로젝트는 프로젝트 번호, 프로젝트 내역, 시작일, 예상 완료일을 속성으로 갖는다. |

<풀이과정(n:m 관계)>

1. 우선 위 글에서 테이블에 해당되는 것은 네모, 관계에 해당되는 것은 마름모, 속성에 해당되는 것은 타원으로 표현한다.
2. 이를 ERD로 그린다.

|  |
| --- |
| 4. 다음은 무지개슈퍼에서 관리해야 할 데이터를 기술한 것이다. 무지개 슈퍼를 위한 ERD를 그리시오.  1) 무지개 슈퍼에서는 고객 명단을 관리하는데, 각 고객마다 고객번호, 고객명, 주소 및 전화 번호를 기록한다.  2) 또한 매장에 있는 물품들의 명세를 관리하는데, 각 물품마다 물품명, 단가, 수량 및 각 물품에 고유한 물품 번호를 기록한다.  3) 어느 한 고객이 여러 물품들을 구입하며, 또한 어느 한 물품을 여러 고객들이 구입한다. 고객이 물품을 구입할 때마다 날짜와 구입 수량 및 총 구입 가격을 기록한다.  4) 물품들은 제조회사에서 제조되므로, 제조회사들에 대한 데이터도 관리하며, 각 제조회사마다 제조회사명, 주소, 전화번호 및 각 제조회사에 대한 제조회사번호를 기록한다.  5) 물품이 제조될 때마다 날짜와 수량을 기록한다. |

<풀이과정(n:m 관계)>

1. 우선 위 글에서 테이블에 해당되는 것은 네모, 관계에 해당되는 것은 마름모, 속성에 해당되는 것은 타원으로 표현한다.
2. 이를 ERD로 그린다.

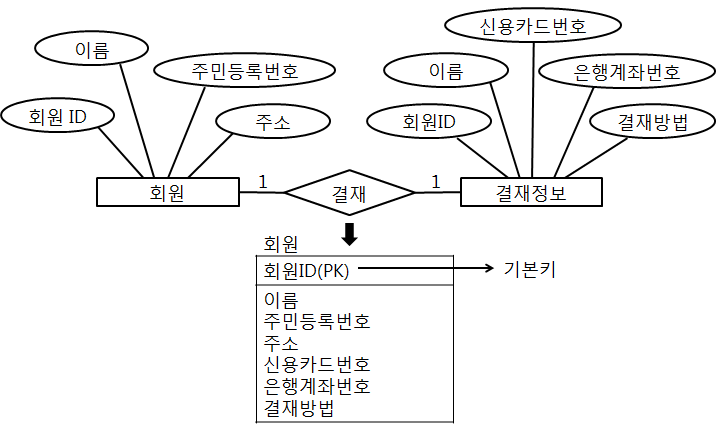
* **물리적 설계**

물리적 설계의 목적은 개발에 사용할 데이터베이스를 선정하여 특정 데이터베이스로 구현될 수 있도록 구체적으로 설계하는 과정이다.

산출물은 테이블 명세서이다. 개체를 테이블로 변환한다. ERD의 개체를 하나의 테이블로 나타내고, 각 개체의 속성을 테이블의 필드로 변환한다. ERD에서의 개체 간의 관계는 테이블의 외래키로 변환 되거나 또 다른 테이블로 변환된다. 각 속성에 대해서는 데이터의 형식과 각종 제약 조건, 인덱스 등을 설정해야 한다.

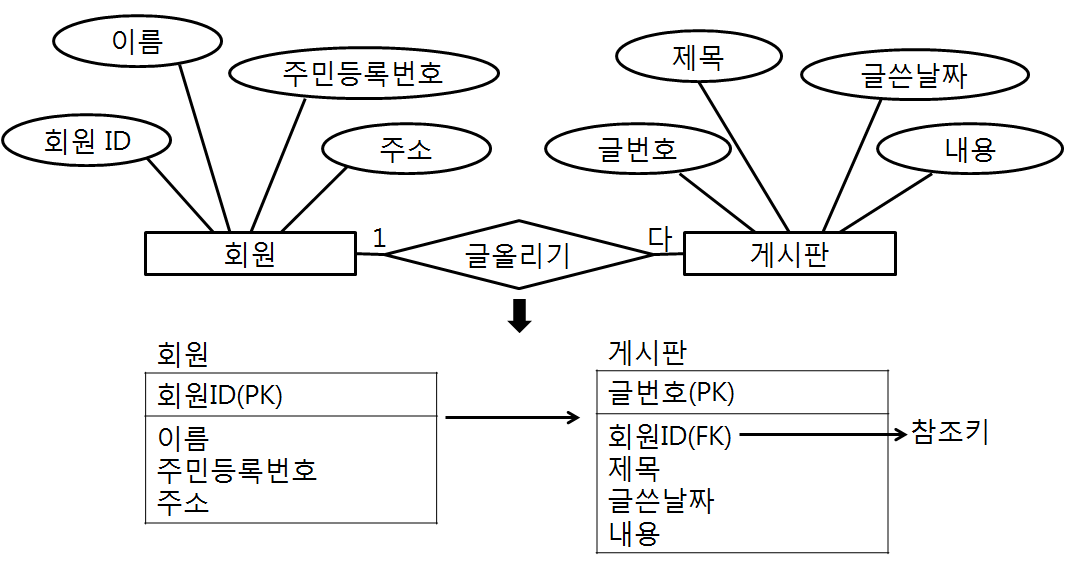
**<일대일 관계>**

ERD를 보면 회원과 결제정보 사이에는 결제 라는 관계가 일대일로 매핑 되어 있다. 이럴 경우에는 두 테이블을 하나의 테이블로 통합한다. 회원 개체가 주체이므로 회원 테이블을 설계하는데, 회원 테이블이 갖고 있는 속성인 회원ID, 이름, 주민등록번호, 주소와 결제 정보에 있는 이름, 신용카드번호, 은행계좌번호, 결재 방법을 칼럼으로 대응시킨다. 회원ID는 기본 키로 설정한다.



**<일대다 관계>**

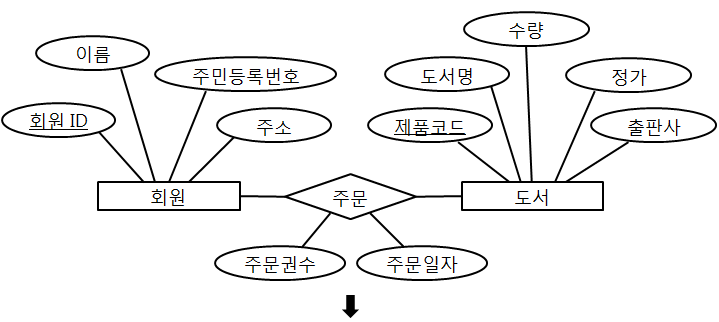
일대다 관계일 경우 우선 주종 관계를 따져서 부모 테이블과 자식 테이블을 구분한다. 회원이 게시판에 올리는 것이므로 회원이 주체가 되므로 회원 테이블이 부모 테이블이고 게시판은 자식 테이블이 된다. 일반적으로 일에 해당하는 테이블이 부모 테이블이고 다에 해당되는 테이블이 자식 테이블이 된다. 부모 테이블의 기본 키를 자식 테이블에 포함시키고 외래키로 설정한다.

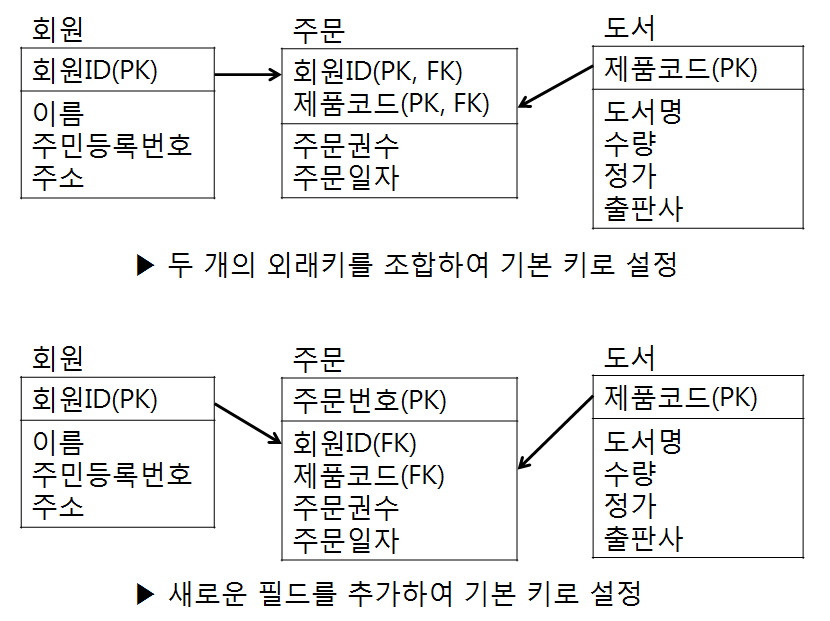


**<다대다 관계>**

새 테이블을 만들어 다대다 관계에 있는 두 테이블의 기본 키를 포함시키고 테이블의 외래키로 설정한다. 새 테이블의 기본 키를 만드는 방법은 다음과 같이 두 가지 방법이 있다.

1. 두 개의 외래 키를 조합하여 기본 키로 설정
2. 새로운 필드를 추가하여 기본 키로 설정





* **테이블 명세**

개체를 테이블로 변환하고 외래키를 설정한 후에 테이블 필드의 데이터 형식과 제약 조건을 정의한다.

회원

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| NO | 컬럼명 | 자료형 | 크기 | 유일키 | NULL허용 | 인덱스 | 키 |
| 1 | 회원ID | VARCHAR | 20 | Y | N | Y(중복불가능) | PK |
| 2 | 이름 | VARCHAR | 20 | N | N | Y(중복가능) |  |
| 3 | 주민등록번호 | VARCHAR | 13 | Y | N | N |  |
| 4 | 주소 | VARCHAR | 100 | N | Y | N |  |

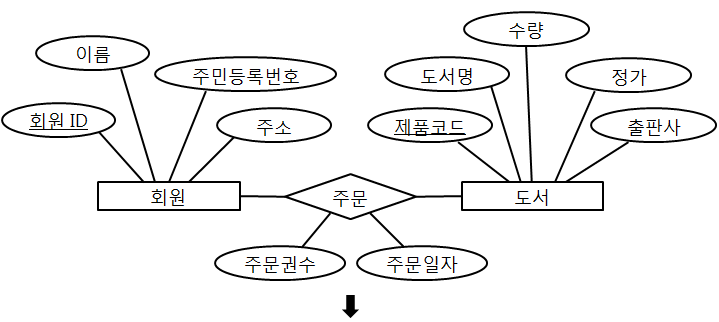
도서

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| NO | 컬럼명 | 자료형 | 크기 | 유일키 | NULL허용 | 인덱스 | 키 |
| 1 | 제품코드 | 시퀀스 |  |  |  | Y(중복불가능) | PK |
| 2 | 도서명 | VARCHAR2 | 50 | Y | N | Y(중복가능) |  |
| 3 | 수량 | NUMBER |  | N |  | N |  |
| 4 | 정가 | NUMBER |  | N | N | N |  |
| 5 | 출판사 | VARCHAR2 | 50 | N | Y | N |  |

주문

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| NO | 컬럼명 | 자료형 | 크기 | 유일키 | NULL허용 | 인덱스 | 키 |
| 1 | 주문번호 | VARCHAR | 9 | Y | N | Y(중복불가능) | PK |
| 2 | 회원ID | VARCHAR | 20 | Y | N | N | FK |
| 3 | 제품코드 | NUMBER |  | Y |  | N | FK |
| 4 | 주문권수 | NUMBER |  | Y |  | N |  |
| 5 | 주문일자 | DATE |  | Y |  | N |  |

* **관계 모델의 릴레이션 표현**



회원(회원ID, 이름, 주민등록번호, 주소)

도서(제품코드, 도서명, 수량, 정가, 출판사)

주문(주문번호, 회원ID, 제품코드, 주문권수, 주문일자)

* 회원, 도서, 주문은 릴레이션 이름이고 소괄호 안에 기술한 것이 릴레이션에 속한 속성이다. 속성들 중에서 기본 키는 밑줄로 표시한다.
* **정규화**

정규화란 자료의 손실이나 불필요한 정보의 도입 없이 데이터의 일관성, 데이터 중복을 최소화하고 최대의 데이터 안정성 확보를 위한 안정적 자료구조로 변화하기 위해서 하나의 테이블을 둘 이상으로 분리하는 작업이다.

**정규화의 필요성**

-자료 저장에 필요한 저장 공간을 최소화

-자료의 불일치성 최소화

-자료의 삽입, 갱신 및 삭제에 따른 이상 현상 제거

-자료구조의 안전성 최대화

**이상현장**

학생의 과목 수강 테이블

|  |  |  |  |
| --- | --- | --- | --- |
| 학번 | 과목번호 | 성적 | 학교 |
| 100 | C413 | A | 4 |
| 100 | E412 | B | 3 |
| 200 | C123 | A | 1 |
| 300 | C324 | C | 1 |
| 300 | C413 | A | 1 |
| 400 | C312 | A | 4 |
| 400 | C324 | A | 4 |
| 400 | C413 | B | 4 |
| 400 | C412 | C | 4 |
| 500 | C321 | B | 2 |

-삽입이상

600학번 학생이 2학년으로 편입하여 삽입하려 할 경우 이 학생이 어떤 과목도 등록하지 않았다면 삽입이 불가능하다. 왜냐하면 학번과 과목번호가 기본 키인데 과목번호가 널이기 때문이다.

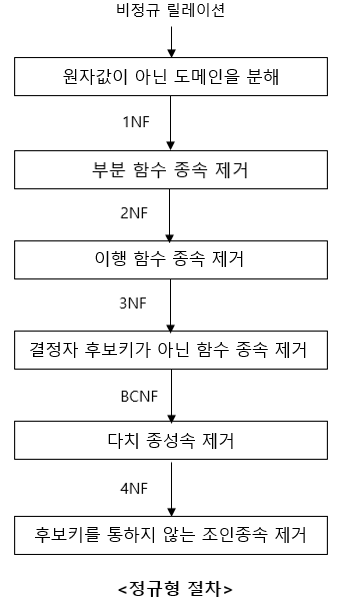
-삭제이상

200번 학생이 C123과목의 등록을 취소할 경우 이 학생이 1학년이라는 정보도 함께 사라진다.

-갱신이상

학번이 400인 학생이 4학년에서 3학년으로 변경될 경우 일부에 대해서 변경이 누락될 경우 데이터의 일관성이 없어진다.

이러한 이상 현상은 속성 간의 종속 관례를 분석하여 여러 개의 릴레이션으로 분해함으로써 해결 할 수 있는데 이렇게 이상현상을 제거하는 과정을 **정규화 과정**이라고 한다.



* **제1정규화(1NF)**

비 정규형 개체에서 반복 또는 복수 속성 값을 가진 속성을 분리시킴으로써 모든 도메인이 원자 값으로 구성하면 제 1 정규형에 속하는 개체가 된다.

[자격증 보유 사항 테이블] 기본키:{회원 번호}

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 회원 번호 | 이름 | 주소 | 핸드폰번호 | 주민등록번호 | 자격증 | 취득일 |
| 1 | 홍길동 | 서울시 | 111-111-1111 | 111111-1111111 | 운전면허증 | 19880203 |
| 1 | 홍길동 | 서울시 | 111-111-1111 | 111111-1111111 | MCSE | 19990603 |
| 1 | 홍길동 | 서울시 | 111-111-1111 | 111111-1111111 | 정보처리기사 | 20000502 |

**문제점**

1. 홍길동이 갖고 있는 자격증 정보를 관리하기 위해 홍길동에 대한 정보(회원 번호, 이름, 주소, 핸드폰번호, 주민등록번호)가 중복해서 저장되어 공간의 낭비가 발생한다.
2. 홍길동의 주소나 핸드폰번호가 바뀌면 3개의 레코드를 모두 변경해야 한다.(갱신 이상)

**해결방법**

반복되는 그룹 속성을 제거 한 뒤 새로운 테이블을 생성하고 기존의 테이블과 1:N 관계를 형성한다.

[회원 테이블] 기본키:{회원 번호}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 회원 번호 | 이름 | 주소 | 핸드폰번호 | 주민등록번호 |
| 1 | 홍길동 | 서울시 | 111-111-1111 | 111111-1111111 |

[자격증 보유 테이블] 기본키:{자격증 번호,회원 번호}

외래키:{회원 번호} 참조할 테이블:[회원]

|  |  |  |  |
| --- | --- | --- | --- |
| 자격증 번호 | 회원 번호 | 자격증 | 취득일 |
| 1 | 1 | 운전면허증 | 19980203 |
| 2 | 1 | MCSE | 19990603 |
| 3 | 1 | 정보처리기사 | 20000502 |

* **제2정규형(2NF)**

제1 정규형에 속하는 개체이지만, 부분 함수의 종속이 존재한다면 이로 인해 이상 현상이 발생한다.

**함수의 종속이란?**

어떤 릴레이션에 있는 속성들의 부분 집합을 X,Y 라고 할 때 임의 튜플에 X의 값이 Y의 값을 함수적으로 결정한다면 X에서 Y로의 함수적 종속성이 존재한다고 말하고 Y가 X에 함수적으로 종속되었다고 말한다.

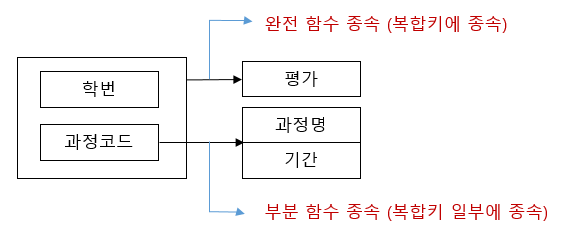
표현 : R.X🡪R.Y (R.X를 결정자, R.Y를 종속자라 한다.)

**부분 함수 종속이란?**

복합 키를 구성하는 일부 속성에 대해서만 부분적으로 함수 종속적인 것을 의미한다.

[교육 과정 등록 테이블] 기본키:{학번, 과정코드}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 학번 | 과정코드 | 평가 | 과정명 | 기간 |
| 100 | A01 | A | JAVA프로그래밍 | 4개월 |
| 101 | B01 | D+ | 웹 마스터 | 6개월 |
| 101 | B03 | A | DBMS전문가 | 5개월 |
| 100 | B01 | B | 웹 마스터 | 6개월 |
| 102 | A01 | B | JAVA프로그래밍 | 4개월 |
| 102 | B03 | A | DBMS전문가 | 5개월 |



**문제점**

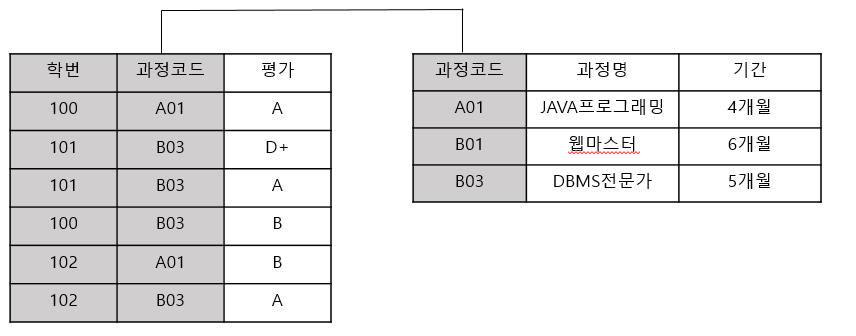
새로운 데이터가 등록될 때마다 과정명과 기간에 중복되는 데이터가 계속 쌓이게 되어 데이터의 무결성(정확성,일관성,유효성)을 유지하는데 어려움이 있다.(갱신 이상)

**해결방법 🡪 함수 종속성 제거**

[교육 과정 등록 테이블] [교육과정 테이블]

기본키 : {학번,과정코드} 기본키 : {과정코드}

외래키 : {과정코드}, 참조 테이블 : [교육과정]



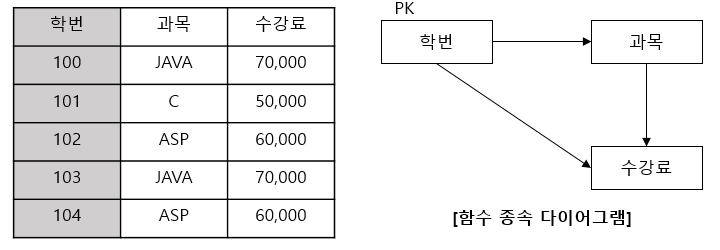
* **제3정규형(3NF)**

제2 정규형에 속하는 개체 여도 이행 함수의 종속이 존재하면 이로 인한 이상 현상이 발생한다.

**이행적 함수 종속이란?**

한 테이블 안의 키가 아닌 모든 컬럼들이 기본 키에만 의존적이어야 하는데 그렇지 못하고 키가 아닌 컬럼에 종속되는 속성이 존재하는 것을 의미한다.

[과목 수강 테이블] 기본키:{학번}



키인 학번이 과목을 결정하고, 과목이 수강료를 결정하면 학번이 수강료를 결정할 수 있게 된다. 이를 이행적 함수 종속성이라고 한다.

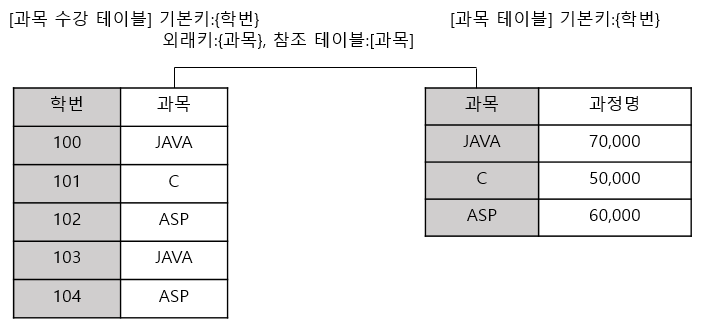
**문제점**

. C++과목이 수강료가 65,000원인 내용을 삽입하려고 할 때 키인 학번은 임의의 값으로 삽입해야 한다.(삽입 이상)

. 101번 학생이 수강을 취소하면 C 과목의 수강료도 함께 삭제된다.(삭제 이상)

. JAVA의 수강료를 80,000월으로 변경하려고 할 때 데이터의 불일치가 발생할 수 있다.(갱신 이상)

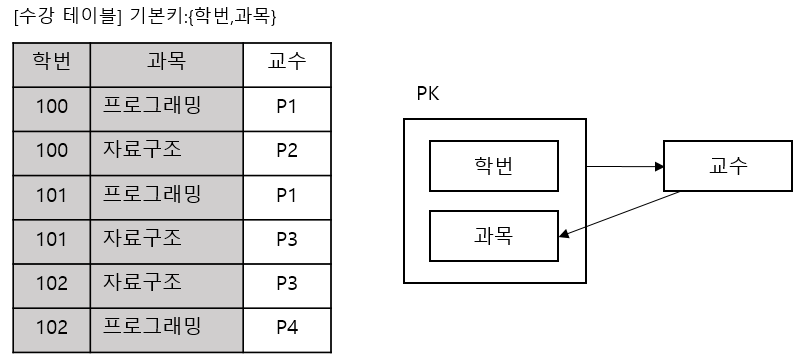
**해결방법 🡪 테이블을 분리한다.**



* **BCNF 정규형**

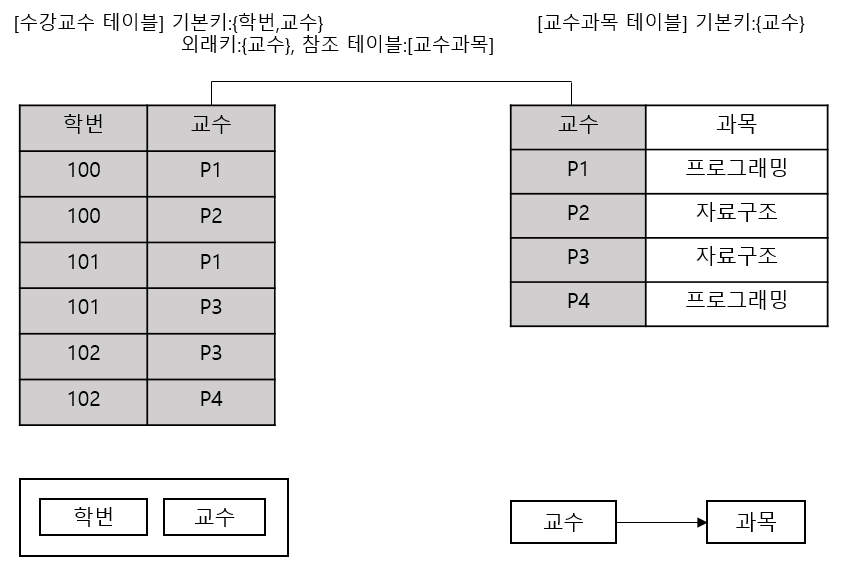
BCNF는 (Boyce and Codd Normal Form) 3차 정규형을 조금 더 강화한 버전이다. BCNF란 3차정규형을 만족하면서 **모든 결정자가 후보 키 집합에 속한 정규형** 이다. 아래와 같은 경우를 생각해보면, 후보 키는 수퍼키중에서 최소성을 만족하는 건데, 이 경우 (학번, 과목)이다. (학생, 과목)은 그 로우를 유일하게 구분할 수 있다. 근데 이 테이블의 경우 교수가 **결정자**이다. 즉, 교수가 정해지면 과목이 결정된다. 근데 교수는 후보키가 아니다. 따라서 이 경우에 BCNF를 만족하지 못한다고 한다. 3차 정규형을 만족하면서 BCNF는 만족하지 않는 경우는 언제일까요? 바로 **일반 컬럼이 후보키를 결정**하는 경우이다.

\*\* 교수가 한 과목만 강의할 수 있다고 가정



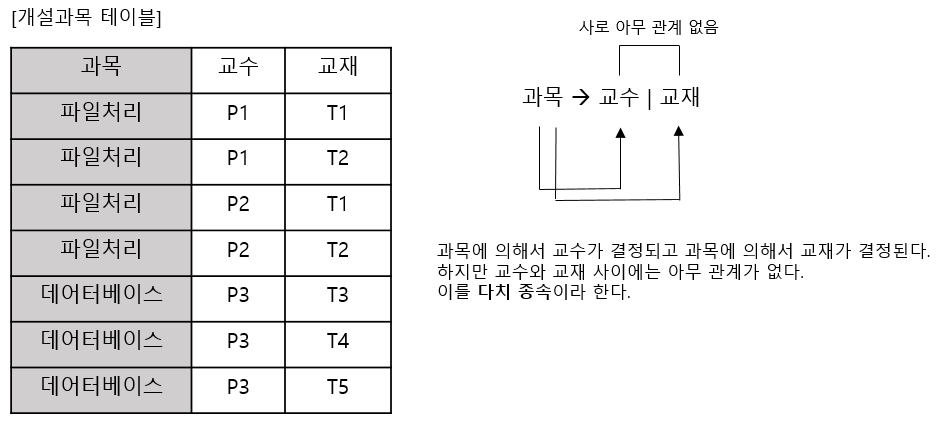
문제점 : 삽입 이상, 삭제 이상, 갱신 이상

해결방법 : 테이블 분리(과목은 교수에 의해 결정)



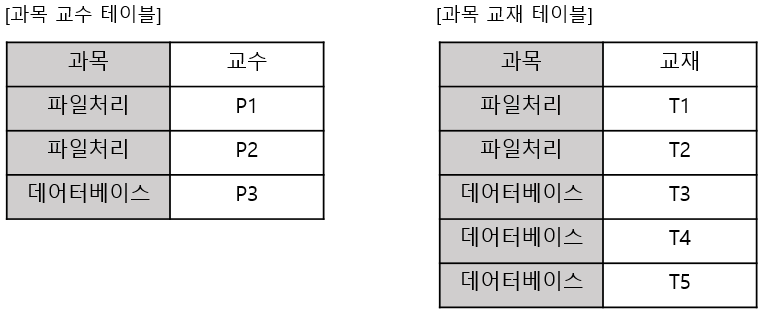
* **제4정규형(4NF)**

BCNF 정규형을 만족하면서 다치 종속이 존재한다면 제4정규형에 속하지 않는다. 다치 종속을 제거하면 제 4NF라고 할 수 있다.



문제점 : 삽입 이상, 삭제 이상, 갱신 이상

해결방법 : 테이블 분리



* **제5정규형(5NF)**

지금까지 이상현상이 발생했을 때 두 개의 릴레이션으로 분해해서 문제를 해결하였고 이 때 정보가 무 손실 분해 되었다. 4NF까지는 이러한 방법이 성공적이었다. 그러나 분해한 것을 다시 조인했을 경우 원래 상태로 돌아가지 못하게 되는 경우가 있는데 이를 조인 종속성이라고 한다. 조인 종속성까지 제고하면 제5 정규형에 속하게 된다.

예를 들어 (S,P,C) 테이블을 (S,P), (P,C) 두 개의 테이블로 분리했다가 다시 조인했을 때 원래의 상태로 돌아오지 못하게 되면 조인 종속성을 제거하기 위해 (S,P), (P,C), (C,S) 세 개의 테이블로 분리한다. 그러면 이 세 개의 테이블을 다시 조인했을 때 원래의 데이터로 돌아온다.