

## < 데이터베이스 개념 >

### 1. 데이터베이스의 정의

데이터베이스 : 어느 한 조직에서 업무 처리를 위해 다수의 응용 시스템 혹은 다수의 사용자들이 공용으로 사용하기 위해 통합·저장된 운영 데이터의 집합

통합된 데이터(Integrated Data)	하나의 주제에 따라 중복을 최소화한 데이터의 집합
저장된 데이터(Stored Data)	사용자나 응용 시스템이 필요시 언제든지 이용할 수 있도록 저장된 데이터의 집합
공용 데이터(Shared Data)	여러 사용자와 다수의 응용 시스템이 공유할 수 있도록 만든 데이터의 집합
운영 데이터(Operational Data)	중복을 최소화하고 여러 사람이 공유함에 있어 문제가 발생하지 않도록 관리를 필요로 하는 데이터로 이용가치가 있는 데이터의 집합

### 2. 데이터베이스의 특징

계속적인 변화(진화)	항상 최신 정보를 유지할 수 있도록 삽입, 삭제, 갱신이 이루어짐
동시 공유	여러 사용자가 동시에 접근하여 이용
실시간 접근성	질의에 대해 실시간 처리 및 응답
내용에 의한 참조	데이터의 물리적 주소나 위치에 의하지 않고 사용자가 요구하는 데이터 내용으로 검색(이용)

### 3. 데이터베이스의 구성요소

#### ① 개체(Entity)

- 사람이 생각하는 개념이나 정보 단위와 같은 현실 세계의 대상으로 실세계에 존재하는 유형 혹은 무형 정보의 대상이며 서로 구별이 되는 하나하나의 대상
- 개체는 하나 이상의 속성(정보)으로 구성됨.

#### ② 속성(Attribute)

- 개체의 특성이나 혹은 상태를 기술하는 것
- 속성만으로는 개체를 구분하기 어려움.

#### ③ 관계(Relationship)

- 두 개 이상의 개체 사이 또는 속성 간의 상호 연관성을 말함.
- 관계의 종류(사상 대응수)

1 : 1(일 대 일)	두 개체 간의 구성 원소가 각각 하나씩 대응되는 경우
1 : n(일 대 다)	두 개체 간의 구성 원소 중 하나의 원소와 여러 개의 원소가 대응되는 경우
n : m(다 대 다)	두 개체 간의 구성 원소들이 상호 여러 개의 원소들과 대응되는 경우

## < 데이터베이스 관리 시스템 >

### 1. DBMS (DataBase Management System)의 정의

파일 시스템의 문제점인 데이터의 중복성과 종속성 등의 문제를 최소화하기 위해 등장하였으며, 사용자와 데이터베이스 간의 중계 역할을 함. 데이터베이스의 내용을 정의하고, 조작, 제어(관리)할 수 있도록 함으로써 모든 사용자나 응용 프로그램들이 데이터베이스를 공유할 수 있도록 관리·운영해 주는 소프트웨어 시스템을 말함.

## 2. DBMS의 필수 기능

정의 기능	저장될 데이터의 형태, 구조 등 데이터베이스의 저장에 관한 여러 가지 사항을 정의(생성)하는 기능
조작 기능	데이터베이스의 자료를 사용자가 이용할 수 있도록 요구에 따라 검색, 갱신, 삽입, 삭제 등을 지원하는 기능
제어 기능	데이터의 정확성과 안전성 유지를 위한 관리 기능으로 데이터의 무결성 유지, 보안, 병행 수행 제어 등을 제공

※ 무결성 : 데이터베이스의 자료가 오류 없이 정확성과 안정성을 유지하기 위한 제약 조건이나 성질

※ 병행 수행 제어 : 동시에 여러 가지 작업을 하는 경우 무결성 유지를 위해 동시에 제어하는 것.

## 3. DBMS의 장·단점

장점	<ul style="list-style-type: none"> <li>• 데이터의 중복성과 종속성을 최소화함</li> <li>• 데이터의 일관성을 유지함</li> <li>• 데이터의 무결성을 유지함</li> <li>• 사용자 간의 데이터 공유가 가능함</li> <li>• 데이터의 보안 유지가 가능함</li> <li>• 데이터의 표준화 구현이 가능함</li> </ul>
단점	<ul style="list-style-type: none"> <li>• 많은 운영비가 소요됨</li> <li>• 자료 처리가 복잡함</li> <li>• Backup(백업)과 Recovery(회복)의 어려움이 있음</li> </ul>

## 4. 데이터베이스 언어

데이터베이스 언어 : 데이터베이스의 전체 구조와 구성요소 및 제약조건 등을 정의(생성)하고, 데이터베이스를 이용하며, 관리·운영을 위해 사용되는 언어. 데이터베이스에서 사용되는 언어는 크게 **정의어**, **조작어**, **제어어**로 나뉨.

정의어 (DDL : Data Definition Language)	데이터베이스 구조를 정의 및 수정 등을 위해 사용되는 언어. 예) CREATE, DROP, ALTER
조작어 (DML : Data Manipulation Language)	데이터베이스 내의 자료를 검색, 삽입, 수정, 삭제하기 위해 사용되는 언어. 예) SELECT, INSERT, UPDATE, DELETE
제어어 (DCL : Data Control Language)	데이터베이스의 데이터에 대해 무결성 유지, 병행 수행 제어, 보호와 관리를 위한 언어 예) COMMIT, ROLLBACK, GRANT, REVOKE

### < 데이터베이스 구조(스키마) >

#### 1. 데이터베이스의 표현

- ① 논리적 구조 : 사용자 관점에서 본 구조를 나타내며, 사용자가 이해하고 생각하는 것을 나타내는 형태
- ② 물리적 구조 : 저장 장치 관점에서 본 구조를 나타내며, 기계 처리에 맞는 형태

#### 2. 스키마(Schema)

스키마 : 데이터베이스의 전체적인 구조와 제약조건에 대한 명세를 기술·정의한 것. 스킴이라고도 함.

내부 스키마 (Internal Schema)	물리적 저장 장치 관점(기계 관점)에서 본 데이터베이스의 물리적 구조
개념 스키마 (Conceptual Schema)	논리적 관점(사용자 관점)에서 본 전체적인 데이터 구조
외부 스키마 (External Schema)	전체 데이터 중 사용자가 사용하는 한 부분에서 본 논리적 구조를 말하며, 서브 스키마라고도 함.

### 3. 데이터베이스 관리자(DBA : DataBase Administrator)

데이터베이스 시스템과 관련된 모든 자원에 대해 기획·통제를 하며 데이터베이스 언어를 이용해 DBMS를 거쳐 전체적인 관리 운영에 책임을 지는 사람이나 집단을 말함.

#### 데이터베이스 사용자

- 응용 프로그래머(Application Programmer)  
데이터베이스의 내용을 일반 사용자가 사용할 수 있도록 프로그램을 개발하는 사람
- 사용자(User) : 데이터베이스의 내용을 실제 사용하는 사람이나 집단을 말함.
- 데이터 관리자(DA : Data Administrator)  
정보 관리의 책임을 지는 경영 분야의 고위직으로 특정한 데이터베이스의 유지나 보안에 대해서는 책임을 지지 않으나 정보의 가치, 무결성과 질을 관리함.

#### < 데이터베이스 설계 >

- 데이터베이스 설계 : 데이터베이스의 스키마를 정의하고, 이에 따라 데이터베이스를 구현하기 위한 전반적인 과정을 말함.
- 데이터베이스 설계 과정 : 요구 조건 분석, 설계, 구현, 운영 및 유지보수 등의 과정을 통해 이루어짐.



#### 1. 요구 조건 분석

- 사용자가 무엇에 대한 정보를 필요로 하는지 문제가 무엇인지 등을 분석하는 과정으로 실제로 만들어야 할 데이터베이스를 정의하고, 사용자의 요구 조건에 따라 명세서를 작성하게 됨.

#### 2. 설계

- 요구 조건 분석을 통해 얻는 정보를 토대로 실제 데이터베이스를 만들기 위한 이전 단계로서, 개념적 설계 → 논리적 설계 → 물리적 설계 과정을 통해 이루어짐.

##### ① 개념적 설계

- 구축하고자 하는 데이터베이스를 개념적으로 표현함으로써 구현할 데이터베이스를 정하고, 데이터베이스를 구성할 구성요소를 결정한 후 수행할 작업과 관계를 설계하는 과정
  - ※ 회사에서 사원들 간에 비상 연락망을 만들 경우 비상 연락망을 구성할 구성요소, 즉 항목을 결정해야 함. 가령 사원번호, 사원명, 부서명, 연락처 등의 항목으로 만들어야 겠다면 그 항목을 결정하는

단계를 개념적 설계 단계라고 함.

- 구성요소를 정하고 수행할 작업을 설계하기 위해 ‘E-R 모델’을 대표적으로 이용함.

※ E-R 모델 (Entity-Relationship Model)

데이터베이스에서 사용되는 개체, 속성, 개체와 개체 간의 관계 등을 약속된 기호를 이용하여 표현함으로써 데이터베이스의 전반적인 구조를 이해하기 쉽도록 표현한 모델

## ② 논리적 설계

- 개념적 설계에서 만들어진 구조를 논리적으로 구현 가능한 데이터 모델로 변환하는 단계로 사용자가 알아볼 수 있는 형태로 변환하고, 스키마를 정의하는 과정

※ 사원번호, 사원명, 부서명, 연락처 등의 항목으로 구성된 비상 연락망을 만들 경우 아래와 같이 우리가 알아 볼 수 있는 테이블(표)과 같은 형태(구조)로 표현하여 사용자가 이해할 수 있도록 하는데, 이와 같은 단계를 논리적 설계 단계라고 함.

비상 연락망

사원번호	사원명	부서명	연락처

- 위와 같이 테이블(표)의 형태로 표현된 모델을 ‘관계 데이터 모델’이라고 함.

## ③ 물리적 설계

- 논리적 데이터베이스 구조를 실제 기계가 처리하기에 알맞도록 내부 저장 장치 구조와 접근 경로 등을 설계하는 과정으로, 효율적인 기계 처리에 맞도록 설계하는 과정

## ④ 구현

- 설계 과정에서 얻어진 것을 토대로 실제 데이터베이스를 만드는 과정
- 데이터베이스 언어를 이용하여 간결·명료하면서도 분석·계획한 내용과 일치하고 유지·보수가 용이하도록 작성

## ⑤ 운영 및 유지보수

- 구현된 데이터베이스를 실제로 운영하는 단계로, 실제 사용해봄으로써 문제점과 개선점 등을 파악하게 됨.

## < 개체-관계 모델(E-R Model) >

### 1. 개체-관계 모델(E-R Model)

- 개념적 설계 단계에서 사용되는 설계 기법으로 데이터베이스를 구성하는 개체 타입과 관계 타입 간의 구조 또는 개체를 구성하는 속성 등을 약속된 기호를 이용하여 표현함으로써 데이터베이스의 전반적인 구조를 이해하기 쉽도록 표현한 모델
- P.Chen 박사에 의해 최초로 제안됨.

### 2. E-R Model의 기호

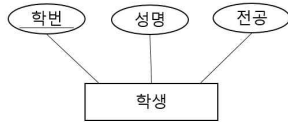
기호	의미
	개체
	속성
	관계
	키 속성(기본키 속성)
	개체와 개체 간의 관계 구조
	복합 속성

※ 키 속성 : 개체의 속성들 중 모두 다른 값을 가져 개체를 식별할 수 있는 속성

※ 복합 속성 : 하나의 속성 값이 세부적으로 나누어질 수 있는 속성

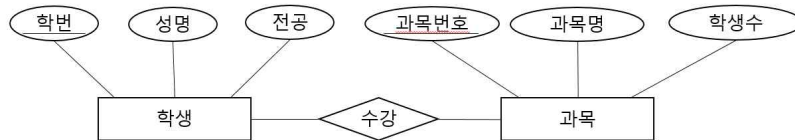
### 3. E-R Model의 표현

#### ① 학번, 성명, 전공 속성으로 구성된 '학생' 개체의 경우



- '학생'이라는 이름의 데이터베이스를 구축하기 위한 개념적 설계 단계에서 E-R Model을 사용.
- 학번, 성명, 전공의 세 가지 속성 중 성명과 전공은 같은 학생이 있을 수 있으나 학번은 모두 다른 값을 가지고, 학생들을 유일하게 구분할 수 있으므로 키 속성(기본키 속성)으로 표현됨.

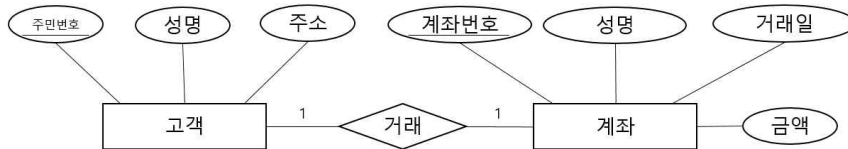
#### ② 개체와 개체 간의 관계 타입을 표현한 경우



- 학번, 성명, 전공으로 구성된 학생 개체와 과목번호, 과목명, 학생수 속성으로 구성된 과목 개체 간에 '수강' 관계가 있음을 표현하고 있음.

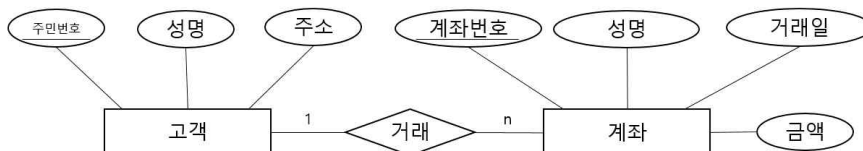
#### ③ 관계의 종류에 따른 표현 방법

##### • 1:1(일 대 일) 관계의 경우



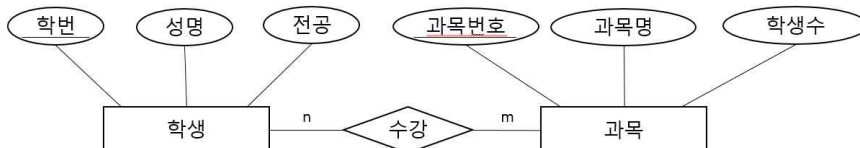
주민번호, 성명, 주소 속성으로 구성된 고객 개체와 계좌번호, 성명, 거래일, 금액 속성으로 구성된 계좌 개체 간에 '거래' 관계가 있으며, 고객과 계좌는 1:1의 거래 관계가 있음을 나타냄. 즉, 한 명의 고객은 하나의 계좌와 거래가 이루어지고, 하나의 계좌는 한명의 고객과 거래가 이루어짐을 의미함.

##### • 1:n(일 대 다) 관계의 경우



고객 개체와 계좌 개체 간에 '거래' 관계가 있으며, 고객과 계좌는 1:n의 거래 관계가 있음을 나타냄. 즉, 한 명의 고객은 여러 개의 계좌와 거래할 수 있고, 하나의 계좌는 한명의 고객과 거래가 이루어짐을 의미함.

##### • n:m(다 대 다) 관계의 경우

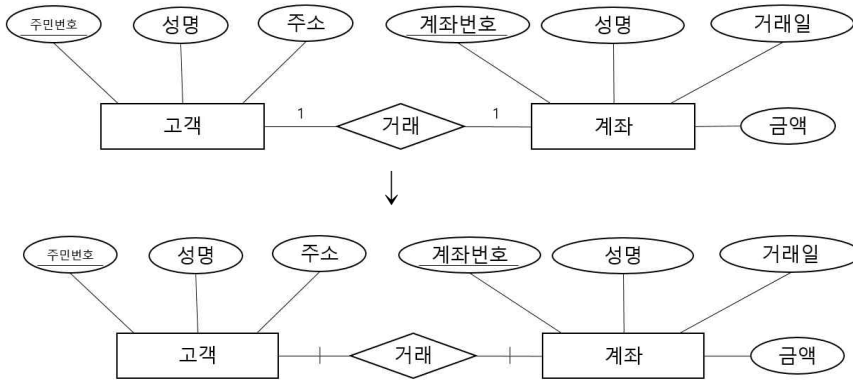


학생 개체와 과목 개체 간에 '수강' 관계가 있으며, 학생과 과목은 n:m의 수강 관계가 있음을 나타냄. 즉, 한 명의 학생은 여러 과목을 수강할 수 있고, 한 과목은 여러 학생이 수강할 수 있음을 의미함.

### 4. 다양한 관계 표현법(정보 공학적 표현법)

기호	의미
—+—+—	1:1 관계
—+—<—	1:n 관계
>—+—<—	n:m 관계
—+—○—	관계가 있을 수도 있고, 없을 수도 있음.

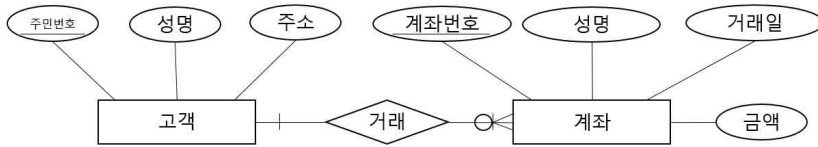
① 1:1 관계의 정보 공학적 표현



② 1:n 관계의 정보 공학적 표현

③ n:m 관계의 정보 과학적 표현

④ 'O'의 선택적 관계 표현



'O'는 선택의 의미로 고객 개체와 계좌 간에 관계가 없을 수도 있고, 관계가 있다면 1:n의 거래 관계가 있음을 의미함.

5. 데이터베이스 모델

- 개념적 설계 이후 논리적 설계 단계에서 사용되는 모델로 논리적으로 구현 가능한 데이터 모델로 변환하기 위해 사용되며, 관계 데이터 모델, 네트워크 데이터 모델, 계층 데이터 모델 등이 있음.

① 관계 데이터 모델

- 표 데이터 모델이라고도 함. 2차원 구조의 표(테이블) 형태로 표현하는 방법으로 구조가 단순하며 사용이 편리하여 가장 많이 사용하고 있는 형태
- n:m 표현이 가능
- '학번', '성명', '전공', '학년'을 속성으로 갖는 '학생' 개체를 관계 데이터 모델로 표현하면 다음과 같음.

학번	성명	전공	학년
083577	강희영	컴퓨터	3
072719	홍길동	토목	4
093505	김정미	컴퓨터	2

② 네트워크 데이터 모델

- 그래프 형태로 표현하며 망 데이터 모델이라고도 하며 레코드 타입 간 관계를 도형으로 표현함.

③ 계층 데이터 모델

- 트리 구조로 표현하여 트리 데이터 모델이라고도 하며 부모-자식 관계, 즉 일 대 다(1:n) 관계를 나타냄.

## < 관계 데이터 모델 >

### 1. 관계 데이터 모델의 용어

릴레이션(Relation)	자료 저장의 형태가 2차원 구조의 테이블(표)로 표현
속성(Attribute)	릴레이션을 구성하는 각 열(Attribute = Column = Field)
튜플(Tuple)	릴레이션의 한 행을 구성하는 속성들의 집합 (Tuple = Row = Record)
도메인(Domain)	하나의 속성이 가질 수 있는 값들의 범위
릴레이션 스키마(Relation Schema)	릴레이션의 이름과 속성 이름의 집합(릴레이션의 구조)
릴레이션 인스턴스(Relation Instance)	릴레이션에서 어느 시점까지 입력된 튜플들의 집합
디그리, 차수(Degree)	릴레이션을 구성하는 속성(항목)의 수
카디널리티(Cardinality)	릴레이션에 입력된 튜플(레코드)의 수

• 다음은 학번, 이름, 주민번호, 학과, 학년 항목으로 구성된 ‘학생’이라는 이름을 가진 릴레이션

학번	이름	주민번호	학과	학년
083577	강희영	850502-1234567	컴퓨터	3
093505	김정미	840127-2345678	컴퓨터	2
072719	홍길동	811022-1345678	토목	4
100325	오승윤	890628-1456789	법학	1

릴레이션 인스턴스

도메인

- 릴레이션을 구성하는 각각의 항목 ‘학번’, ‘이름’, ‘주민번호’, ‘학과’, ‘학년’은 속성이 됨.
- 릴레이션에서 한명 한명에 대한 ‘학번’, ‘이름’, ‘주민번호’, ‘학과’, ‘학년’ 속성 값, 즉 하나의 행이 튜플이 됨.
- 하나의 속성이 가질 수 있는 값의 범위를 도메인이라고 하며, 도메인은 릴레이션을 만들 때 속성의 값으로 올 수 있는 범위를 제한함으로써 범위 외의 값은 올 수 없도록 해야 함. ‘학년’ 속성의 경우 속성 값의 범위를 1, 2, 3, 4로 제한해 도메인을 만들면 ‘학년’ 속성에는 그 외의 값은 입력될 수 없음.
- ‘학생’ 릴레이션에서 항목은 ‘학번’, ‘이름’, ‘주민번호’, ‘학과’, ‘학년’ 속성으로 구성되어 속성의 수가 5개이므로 차수(Degree)는 5가 됨.
- ‘학생’ 릴레이션에서 현재까지 강희영, 김정미, 홍길동, 오승윤 학생의 자료가 입력되었다면 이 4명에 대한 ‘학번’, ‘이름’, ‘주민번호’, ‘학과’, ‘학년’ 모든 값 자체, 즉 튜플(자료)의 집합을 릴레이션 인스턴스라고 함.
- ‘학생’ 릴레이션에서 현재까지 강희영, 김정미, 홍길동, 오승윤 학생의 자료가 입력되었다면 입력된 튜플의 수는 4이므로 카디널리티는 4가 됨.
- ‘학생’ 릴레이션에서 릴레이션 이름 ‘학생’과 속성명 ‘학번’, ‘이름’, ‘주민번호’, ‘학과’, ‘학년’ 즉, 릴레이션의 구조를 릴레이션 스키마라고 함.

※ 릴레이션의 특징

- ① 릴레이션의 튜플들은 모두 상이함.
- ② 릴레이션의 튜플들은 유일하며 순서에는 의미가 없음.
- ③ 릴레이션의 속성들 간의 순서는 의미가 없음.
- ④ 릴레이션의 속성은 원자값으로 구성되며 분해가 불가능함.

원자값 : 더 이상 분해되지 않는 최소 구성의 단위. 널 값(Null value)도 원자값에 속함.

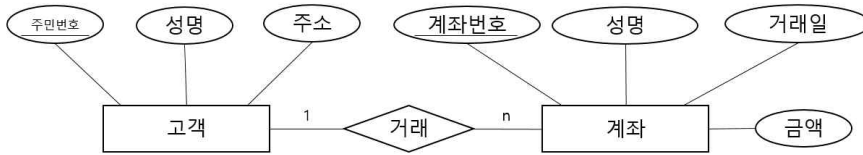
### 2. E-R Model과 관계 데이터 모델과의 관계

① E-R Model로 표현된 단순한 개체와 속성을 릴레이션으로 표현한 경우

- 개념적 설계 단계에서 E-R Model로 표현된 ‘학번’, ‘성명’, ‘학과’, ‘연락처’ 속성으로 구성된 ‘학생’ 개체를 논리적 설계 단계에서 ‘학번’, ‘성명’, ‘학과’, ‘연락처’ 속성으로 구성된 ‘학생’ 릴레이션으로 구현
- E-R Model에서 개체는 릴레이션으로, E-R Model에서 각각의 속성은 릴레이션의 각 속성(항목)으로 변환됨.
- E-R Model에서 키 속성인 ‘학번’은 릴레이션에서 기본키(Primary Key) 속성으로 표현됨.

※ 기본키(Primary Key) : 키 속성과 동일한 개념으로 릴레이션에서 개체를 식별하기 위해 선정된 속성 즉, ‘학생’ 릴레이션에서 한명 한명의 학생을 구별하기 위해 선정된 속성

② 개체와 개체 간의 관계를 나타낸 E-R Model을 릴레이션으로 표현한 경우



- E-R Model에서 두 개체 간의 관계를 나타낸 경우 위와 같이 두 릴레이션으로 표현하며, 두 릴레이션의 관계를 연결하기 위해서는 두 릴레이션의 기본키 속성을 필요한 릴레이션에 추가해서 관계가 성립이 되도록 함.
- 위 릴레이션 중 '고객' 릴레이션에서 홍길동 고객이 있다면, 홍길동 고객의 금액을 알고 싶은 경우 '고객' 릴레이션의 자료만으로는 금액을 알 수 없다. 이런 경우 '계좌' 릴레이션의 기본키인 '계좌번호'를 필요에 의해 '고객' 릴레이션에 추가해줌으로써 '계좌' 릴레이션을 이용해 금액을 알 수 있다. 즉, '고객' 릴레이션에서 홍길동 고객의 계좌번호를 알면 '계좌' 릴레이션에서 홍길동 고객의 계좌번호와 같은 계좌번호의 금액이 홍길동 고객의 금액이 됨.
- ※ Mapping Rule : 개념적 데이터베이스 모델링 결과를 관계형 데이터베이스 이론에 근거하여 데이터베이스 구조로 변환하는 과정을 말하며, 개체는 릴레이션(테이블)으로, 속성은 릴레이션의 항목으로, 키 속성(식별자)은 기본키로, 관계는 외래키로 변환됨.
- ※ 교차 엔티티(Intersection Entity) - n:m 관계의 E-R Model을 릴레이션으로 표현하는 경우 보다 정확한 상호 참조를 위해 하나의 릴레이션을 더 만들게 되는데, 이를 '교차 엔티티'라고 함.

< 키와 무결성 제약조건 >

1. 키의 개념

- 관계 데이터베이스에서 튜플을 식별하기 위해 사용하는 속성이나 속성의 집합으로 데이터베이스의 참조 또는 검색 시에 사용됨.

2. 키의 종류

학생

인명

학번	이름	주민번호	학과	학년	이름	주소
070222	강희영	810503-1234567	컴퓨터	2	강희영	서울특별시 1가 12번지
090203	김정미	830225-2345678	컴퓨터	1	박태인	경기 안산시 2가 3번지
020525	이나라	761207-2456789	신문방송	4	강희영	인천광역시 3가 10번지
020723	강희영	741002-1345678	체육	4	김정애	경기 안산시 2가 3번지

① 후보키(Candidate Key)

- 릴레이션에서 각 튜플을 유일하게 식별할 수 있는 속성이나 속성의 집합
- 후보키가 될 수 있는 조건은 유일성과 최소성을 모두 만족해야 함.

- 위 [학생] 테이블에서 학생 개개인은 학번이나 주민번호를 알면 구분할 수 있다. 따라서 '학번'과 '주민번호'는 후보키가 됨.
- 위 [인명] 테이블에서는 이름에도 중복되는 값이 있을 수 있고, 주소에도 중복되는 값이 있을 수 있다. 따라서 이런 경우 이름과 주소 모두 알아야 개개인을 구분할 수 있으므로 이름, 주소를 조합한 (이름, 주소)가 후보키가 됨.

※ 유일성 : 각 튜플을 유일하게 식별하는 성질

위 [학생] 테이블에서 학번과 이름 두 가지를 알아도 학생 개개인을 구분할 수 있으나 이름 없이 학번 하나만 가지고도 학생 개개인을 구분할 수 있다. 이와 같이 하나하나의 자료를 식별하는데 필요한 최소의 속성으로만 구성되는 성질을 최소성이라고 함.

② 기본키(Primary Key)

- 후보키 중에서 튜플을 식별하기 위해 특별히 선택된 키
- 중복될 수 없으며, NULL 값이 올 수 없음.



- 유일성과 최소성을 만족해야 함.
- 위 [학생] 테이블에서 후보키인 '학번', '주민번호' 중에서 '학번'을 기본키로 설정할 수 있음. (주민번호를 기본키로 설정해도 됨.)

### ③ 대체키(Alternate Key)

- 후보키 중에서 기본키를 제외한 속성
- 위 [학생] 테이블에서 '학번'을 기본키로 했다면 대체키는 '주민번호'가 됨. '주민번호'를 기본키로 했다면 대체키는 '학번'이 됨.

### ④ 외래키(Foreign Key)

- 하나의 테이블에서 원하는 자료를 얻지 못하는 경우 다른 테이블의 참조(이용)하기 위해 사용되는 속성을 말하며, 외래 식별자라고도 함.
- 참조 릴레이션(테이블)의 기본키와 같아야 함.
- NULL이 올 수 있음.
- 외래키의 속성명과 참조 릴레이션의 기본키 속성명은 서로 달라도 무방함.

교수				학과			
번호	교수 이름	학과 번호	직급	학과 번호	학과 이름	교수 번호	학생수
1001	오준석	A1	주임	A1	컴퓨터	1001	30
1002	이순신	A2	부주임	A2	정보통신	1002	20
1003	홍길동	B1	교수	B1	토목	1003	50

- 위 [교수] 테이블에서는 홍길동 교수가 담당하는 학생 수를 알 수 없다. 이때 [교수] 테이블에서 '학과번호'를 가지고 [학과] 테이블의 '학과번호'를 참조해서 홍길동 교수의 학과번호 'B1'에 해당하는 학생 수를 알면 된다. 이와 같이 [교수] 테이블의 '학과번호'를 외래키로 선정해서 [학과] 테이블을 참조하면 된다. 이때 [학과] 테이블과 같이 참조되는 테이블을 '참조 테이블'이라고 함.

#### 식별 관계와 비식별 관계

위와 같이 [교수] 테이블의 외래키로 '학과번호'를 선정한 경우, 즉 외래키가 일반 속성인 경우는 참조 테이블과 '비식별 관계'라고 함. 비식별 관계에서는 테이블을 참조하여 원하는 정보만 얻을 수 있는 관계임. 반면 기본키를 외래키로 선정하여 다른 테이블을 참조하는 경우는 참조 테이블의 튜플을 식별할 수 있기 때문에 '식별 관계'라고 함.

- 식별 관계 : 외래키가 기본키인 경우
- 비식별 관계 : 외래키가 일반 속성인 경우

### ⑤ 슈퍼키(Super Key)

- 한 릴레이션(테이블) 내의 튜플들을 식별할 수 있는 후보키와 다른 속성들과의 모든 조합
- 유일성은 만족하지만, 최소성은 만족하지 않음.
- 위 [학생] 테이블에서 학생 개개인을 구분하기 위한 모든 경우를 후보키와 조합해서 나열하면 다음과 같음.

(학번, 이름), (학번, 주민번호), (학번, 학과), (학번, 학년), (학번, 이름, 학과),  
(학번, 이름, 학과, 학년), (주민번호, 이름), (주민번호, 학과) . . .

## 3. 무결성(Integrity) 제약조건

무결성은 데이터베이스 자료의 오류 없는 정확성과 안정성을 나타내는 것으로, 무결성 제약조건은 정확성과 안정성을 유지하기 위한 제약 조건

### ① 개체 무결성

- 기본키는 NULL 값이 올 수 없으며, 중복될 수 없음을 나타내는 제약조건
- 개체를 식별하기 위해서 오류가 없도록 하기 위한 제약조건
- 아래 '학생' 릴레이션(테이블)에서 '학번'은 기본키로 학생들을 서로 구분하기 위해 선정된 속성이므로 중복되거나 NULL 값이 올 수 없음.

## 학생

학번	이름	학년	학과	성적
400	박태인	1	수학과	90
200	홍길동	2	국문과	70
100	오유빈	4	컴퓨터과	95
300	이순신	3	토목과	80

### ② 참조 무결성

- 외래키는 NULL 값이 올 수 있으며, 참조 릴레이션(테이블)의 기본키와 같아야 하는 제약조건으로 테이블 참조 시 오류가 없도록 하기 위한 제약조건
- 아래 [교수] 테이블의 '학과번호'에서 참조하는 [학과] 테이블의 '학과번호'의 값은 반드시 존재해야 함.

교수				학과			
번호	교수 이름	학과 번호	직급	학과 번호	학과 이름	교수 번호	학생수
1001	오준석	A1	주임	A1	컴퓨터	1001	30
1002	이순신	A2	부주임	A2	정보통신	1002	20
1003	홍길동	B1	교수	B1	토목	1003	50

### ③ 도메인 무결성

- 릴레이션(테이블)에서 속성값의 범위가 정의된 경우 그 속성값은 정해진 범위 이내의 값으로 구성해야 하는 제약조건이며, 동일한 속성에 대해 데이터 타입과 데이터 길이가 동일해야 함.
- 다음 [학생] 릴레이션(테이블)에서 학년 속성의 값으로 1, 2, 3, 4만 올 수 있도록 도메인을 정의했다면 그 외 값이 올 수 없으며, 학년 속성은 누구든지 동일한 숫자형 데이터 타입과 데이터 길이가 적용됨.

## 학생

학번	이름	학년	학과	성적
400	박태인	1	수학과	90
200	홍길동	2	국문과	70
100	오유빈	4	컴퓨터과	95
300	이순신	3	토목과	80

### ④ 고유(Unique) 무결성

- 특정 속성에 대해 고유한 값을 가지도록 조건이 주어진 경우, 그 속성값은 모두 달라야 하는 제약조건
- [학생] 릴레이션에서 릴레이션 정의 시 '이름' 속성에는 중복된 값이 없도록 제한했다면 '이름' 속성에는 중복된 이름이 있어서는 안 됨.

### ⑤ NULL 무결성

- 특정 속성값에 NULL이 올 수 없다는 조건이 주어진 경우, 그 속성값은 NULL 값이 올 수 없다는 제약조건
- [학생] 릴레이션에서 릴레이션 정의 시 '학과' 속성에는 NULL 값이 올 수 없도록 제한했다면 '학과' 속성에는 NULL이 있어서는 안 됨.

### ⑥ 키 무결성

- 한 릴레이션(테이블)에는 최소한 하나의 키가 존재해야 하는 제약조건