
데이터베이스 설계

1. 데이터베이스 설계
 2. 개념적 설계
 3. 논리적 설계
 4. 물리적 설계
-

>> 학습목표 <<

- ❖ 데이터베이스 설계 과정을 이해한다.
- ❖ 데이터베이스 설계 예를 살펴본다.
- ❖ 요구사항 분석 과정을 이해하고 적용 방법을 알아본다.
- ❖ 개념적 설계 과정을 이해하고 적용 방법을 알아본다.
- ❖ 논리적 설계 과정을 이해하고 적용 방법을 알아본다.
- ❖ 물리적 설계 과정을 이해하고 적용 방법을 알아본다.

1. 데이터베이스 설계

● 데이터베이스의 역할

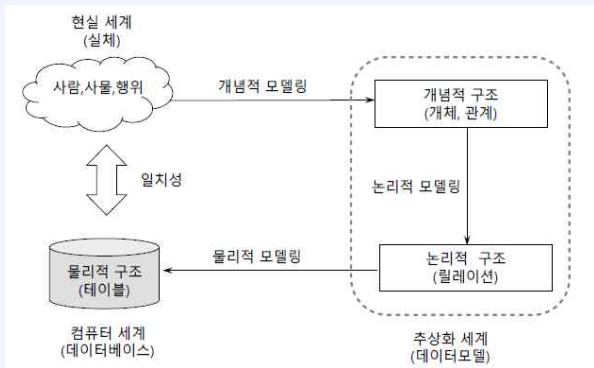


● 데이터 모델링(data modeling)

- 데이터베이스 구조를 생성하는 절차적 과정
- 실세계를 개념화하고 논리적 구조로 추상화한 다음 물리적 테이블 구조를 컴퓨터 안에 생성하는 일련의 단계를 거쳐 데이터베이스 구조를 완성

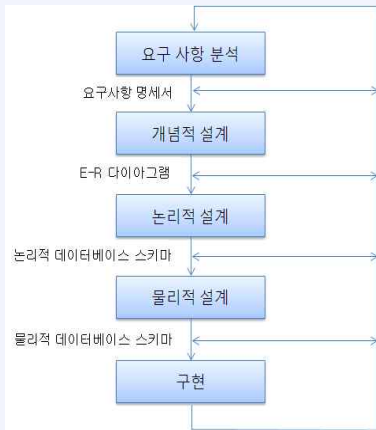
데이터베이스 모델링

● 데이터 모델링 단계



데이터베이스의 설계 과정

- 반복적인 설계 과정을 통해 정제된 최종 설계 결과가 생성



데이터베이스 설계 예(병원 DB)

● 요구사항 분석 적용 예

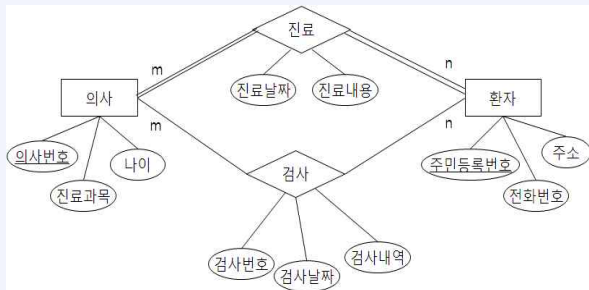
■ 병원 DB 요구사항 명세서

- 병원에는 많은 의사가 근무하고 있으며 환자는 여러 명의 의사로부터 진료를 받고 처방에 따라 필요할 경우 검사를 받는다. 의사 역시 여러 환자들을 진료하며 일부 의사는 검사도 수행한다.
- 각 환자에게는 진료날짜, 진료의사, 진료내용 등의 진료 정보와 실시된 검사에 대한 고유한 검사번호와 검사날짜, 검사내역, 검사의사번호 등의 검사기록들이 유지된다.
- 의사에 대해서는 고유한 의사번호와 진료과목, 나이 등의 정보가 유지되고 환자에 대해서는 고유한 주민등록번호와 전화번호, 주소 등의 신상 정보를 보관한다.

데이터베이스 설계 예(병원 DB)

● 개념적 설계 적용 예

■ 병원 DB E-R 다이어그램



데이터베이스 설계 예(병원 DB)

- 논리적 설계 적용 예

- 병원 DB 관계형 데이터베이스 스키마

의사(의사번호, 진료과목, 나이)

환자(주민등록번호, 전화번호, 주소)

진료(진료의사번호, 주민등록번호, 진료날짜, 진료내용)

검사(검사번호, 주민등록번호, 검사날짜, 검사내역, 검사의사번호)

2. 요구사항 분석

2.1 요구사항 분석

- 데이터베이스 설계 과정의 첫 번째 단계
 - 구축하고자 하는 데이터베이스의 구현 범위와 사용자의 범주를 결정
 - 예비 사용자들로부터 요구 사항을 수집하고 업무 처리를 위해 필요한 데이터를 분석
 - 사용자와의 인터뷰, 설문지 조사, 업무 관련 문서의 검토와 기존 시스템 또는 유사 시스템에 대한 분석 등을 포함
-
- 분석 결과는 요구사항 명세서로 문서화
 - 분석가는 담당자의 요구를 정확하게 파악하도록 노력하고 용어 등이 모호성이 없도록 최대한 명확하게 명세서를 정리
 - 명세서는 분석가의 자의적 해석이나 짐작이 아닌 사용자의 요구 사항만을 충실히 반영하여 구체적 문장 형태로 서술
 - 최종적인 요구사항 명세서는 사용자에게 최종 확인을 받는 과정이 필요

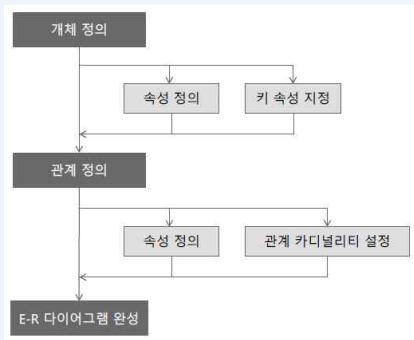
【단계1】 자동차극장 DB의 요구사항 명세서

- 1 자동차극장 관련하여 고유한 극장명과 주소, 전화번호, 홈페이지주소 정보를 유지해야 한다.
- 2 고객은 회원과 비회원으로 구분된다.
- 3 고객은 예약을 위해 극장 회원으로 가입해야 하며 가입 정보로 회원아이디, 비밀번호, 회원이름, 휴대폰번호, 생년월일, 나이, 성별 정보를 입력해야 한다.
- 4 회원은 가입 시 입력한 회원아이디에 의해 식별된다.
- 5 비회원은 미가입 고객으로 현장에서만 티켓 발급이 가능하며 현장에서 입력한 휴대폰번호에 의해 식별된다.
- 6 상영되는 영화 관련해서 영화번호, 영화제목, 관람시간, 등급, 제작날짜, 감독명, 출연배우명 정보를 유지해야 한다.
- 7 영화는 내부적으로 부여된 영화번호에 의해 식별된다.
- 8 극장이 소유한 상영관은 비어 있는 땅에 계속 추가 건설될 예정이며 상영관마다 상영관이름, 자동차 자리수 정보를 유지해야 한다.
- 9 상영관은 내부적으로 부여된 상영관번호에 의해 식별된다.
- 10 각 상영관은 매일 영화를 상영하는데 상영회차, 영화제목, 상영날짜, 상영시간 정보를 유지해야 하며 상영회차에 따라 관람료가 매번 다르다.
- 11 상영관은 여러 영화를 상영할 수 있고, 하나의 영화는 여러 상영관에서 상영될 수 있다.
- 12 회원은 원하는 영화를 미리 예약할 경우, 영화제목, 상영관이름, 상영날짜, 상영시간을 선택한 후 인원수를 입력하면 입력 내용에 회원이름, 영화제목, 상영관이름, 상영날짜, 상영시간, 결제금액과 예약날짜 정보가 추가로 출력된 예약확인증을 온라인으로 발급받는다.
- 13 회원은 여러 영화를 예약할 수 있고, 하나의 영화는 여러 회원이 예약할 수 있다.
- 14 회원 또는 비회원 모두 자동차극장에 도착한 고객은 차량의 상영관번호, 관람위치번호를 배정받고 배정 시 차량번호와 배정날짜를 등록하고 휴대폰번호를 확인받는다.
- 15 회원은 기본 20% 할인을 받고 회원가족도 10% 할인을 받을 수 있으므로 회원과 가족관계인 회원가족에 대해서도 가족이름, 나이, 성별 정보를 유지해야 한다.
- 16 회원만 영화에 대해 평가를 할 수 있고 평점을 부여한다.

3. 개념적 설계

- 개념적 설계(E-R 다이어그램 작성) 과정

- 데이터베이스 설계의 전체 골격을 결정하는 과정
- 요구사항 명세서의 내용을 기반으로 핵심적인 데이터 요소들을 추출하여 E-R 다이어그램을 작성



3.1 개체 정의

■ 개체 도출

【개체 식별 방법】

- 개체는 저장 가치가 있는 핵심 데이터를 가진 사람이나 사물, 장소 또는 무형적 개념을 의미한다. 명세서 문장 중에서 주어나 목적어로 표현되는 명사들을 찾는다.
- 명사들은 서로 의미를 명확히 표현하거나 꾸며주는 연관성을 가지고 있다. 속성에 해당하는 하위 개념의 명사들로부터 꾸밈을 받는 상위 개념의 명사가 주로 개체가 된다.
- 개체는 실세계에서 독립적 존재이므로 꾸밈을 받는 하위 개념 명사 중에 고유한 명칭(번호나 코드, 이름 등)을 가진다.
- 개체는 관계와 비교하면 상대적으로 오랜 시간 지속되는 특성이 있다.
- 개체는 보통 또 다른 여러 개체들이 공유하는 대상이다.

■ 개체 정의 예



3.2 관계 정의

■ 관계 도출

【관계 식별 방법】

- 관계는 저장 가치가 있는 데이터를 발생시키는 사건이나 행위와 같은 개체간의 연관성을 의미한다. 명세서 문장 중에서 서술어로 표현되는 동사들을 찾는다.
- 관계는 반드시 연관성을 갖는 둘 이상의 개체가 필요하다.
- 대부분의 관계는 의미를 명확히 표현하거나 꾸며주는 하위 개념의 속성을 갖는다.
- 관계는 실세계에서 종속적 존재이므로 보통 하위 개념의 속성 중에 고유한 명칭(번호, 코드, 이름 등)을 갖지 않는다.
- 관계는 개체와 비교하면 일시적이며 상대적으로 짧은 시간만 지속되는 특성이 있다.

■ 관계 정의 예



3.3 속성 정의

■ 속성 도출

【속성 식별 방법】

- 속성은 도출된 개체나 관계의 존재 혹은 상태를 나타내는 특성을 의미한다. 명세서 문장 중에서 주어 나 목적어, 서술어를 수식하거나 꾸며주는 명사들을 찾는다.
- 속성은 보통 또 다른 속성들과 함께 공통의 특정 대상을 명확히 표현하기 위해 연관성을 가진다. 개체에 해당하는 상위 개념의 명사나 관계에 해당하는 상위 개념의 동사를 꾸며주는 하위 개념의 명사가 주로 속성이 된다. 속성은 실제 값을 저장한다.
- 속성은 실세계에서 종속적 존재이므로 상위 개념이 반드시 필요하다.

■ 속성 정의 예



요구사항 명세서의 E-R 다이어그램 변환 규칙

● 개체, 관계, 속성 구별 방법

개체	관계	속성
독립적 존재	종속적 존재	종속적 존재
명사(주어, 목적어)로 표현	동사(서술어)로 표현	명사(수식어)로 표현
상위 개념	상위 개념	하위 개념
고유한 명칭(이름, 번호) 보유	인위적 명칭(이름, 번호) 부여	해당사항 없음
지속적	일시적	지속적/일시적

요구사항 명세서의 E-R 다이어그램 변환 규칙

●유의 사항

- 명세서 문장 안에서 개체를 찾아 사각형, 관계는 마름모, 속성은 타원형으로 표시
- 의미가 중복되거나 반복되는 대상은 하나의 표준 용어로 통일하여 표시
- 데이터와 직접적인 연관성이 낮거나 혹은 너무 일반적인 표현은 삭제(가운데줄) 표시
 - 특히, 데이터 관점에서 불필요하거나 기능 관련 표현들은 무시

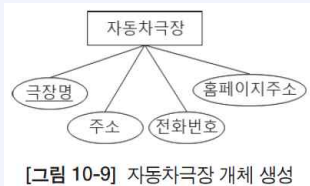
●E-R 다이어그램 작성 시 유의 사항

불필요한 표현	무시해도 좋은 이유
정보, 데이터	모든 특성이 갖는 공통 표현으로 속성으로 부적절함
유지, 저장, 관리, 기록	데이터베이스의 일반 기능으로 관계로 부적절함
입력, 수정, 삭제, 검색	데이터베이스 시스템의 일반 기능으로 모두에게 해당되어 관계로 부적절함
발급, 부여, 선택, 확인, 등록	관리자나 사용자의 소프트웨어 기능으로 관계로 부적절함

자동차극장 DB의 E-R 다이어그램

❶ 자동차극장 관련하여 고유한 극장명과 주소 전화번호 홈페이지주소 정보를 유지해야 한다.

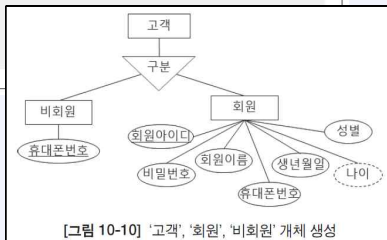
- 자동차극장 개체는 극장명, 주소, 전화번호, 홈페이지주소 속성을 갖는다.
- 자동차극장 개체의 주식별자는 극장명 속성이다.



자동차극장 DB의 E-R 다이어그램

- ② **고객**은 회원과 비회원으로 **구분**된다.
- ③ 고객은 예약을 위해 극장 **회원**으로 카입해야 하며 카입 정보로 **회원아이디**, **비밀번호**, **회원이름**, **휴대폰번호**, **생년월일**, **나이**, **성별** 정보를 입력해야 한다.
- ④ 회원은 카입 시 입력한 **회원아이디**에 의해 식별된다.
- ⑤ **비회원**은 미가입 고객으로 현장에서만 티켓 발급이 가능하며 현장에서 입력한 **휴대폰번호**에 의해 식별된다.

- 고객 상위 개체는 회원과 비회원 하위 개체로 구분된다.
- 회원 개체는 회원아이디, 비밀번호, 회원이름, 휴대폰번호, 생년월일, 나이, 성별 속성을 갖는다.
- 회원 개체의 주식별자는 회원아이디 속성이다.
- 비회원 개체의 주식별자는 휴대폰번호 속성이다.



자동차극장 DB의 E-R 다이어그램

- ⑥ 상영되는 **영화** 관련해서 (영화번호), (영화제목), (관람시간), (등급), (제작날짜), (감독명), (출연배우명) 정보를 유지해야 한다.
- ⑦ 영화는 내부적으로 부여된 영화번호에 의해 식별된다.

- 영화 개체는 영화번호, 영화제목, 관람시간, 등급, 제작날짜, 감독명, 출연배우명 속성을 갖는다.
- 영화 개체의 주식별자는 영화번호 속성이다.

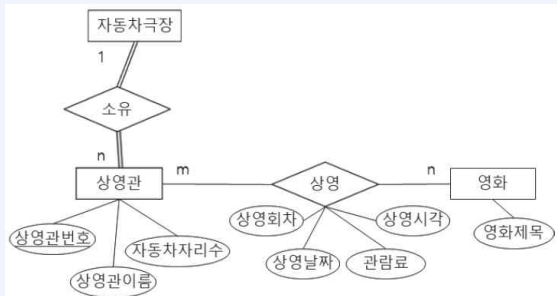


자동차극장 DB의 E-R 다이어그램

- ⑧ 극장이 소유한 상영관은 비어 있는 땅에 계속 추가 건설될 예정이며 상영관마다 상영관이름, 자동차자리수 정보를 유지해야 한다.
- ⑨ 상영관은 내부적으로 부여된 상영관번호에 의해 식별된다.
- ⑩ 각 상영관은 매일 영화를 상영하는데 상영회차, 영화제목, 상영날짜, 상영시간 정보를 유지해야 하며 상영회차에 따라 관람료가 매번 다르다.
- ⑪ 상영관은 여러 영화를 상영할 수 있고, 하나의 영화는 여러 상영관에서 상영될 수 있다.

- 자동차극장 개체와 상영관 개체 사이는 소유 관계이다.
- 자동차극장 개체는 상영관 개체와 일대다 소유 관계를 갖는다.
- 상영관 개체는 상영관이름, 자동차자리수 속성을 갖는다.
- 상영관 개체의 주식별자는 상영관번호 속성이다.
- 상영관 개체와 영화 개체 사이는 상영 관계이다.
- 상영관과 영화 사이에 상영 관계가 맺어질 때마다 상영회차, 상영날짜, 상영시간, 관람료 속성이 새롭게 결정된다.
- 상영관 개체는 영화 개체와 일대다 상영 관계를 갖고 영화 개체도 상영관 개체와 일대다 상영 관계를 갖는다.

자동차극장 DB의 E-R 다이어그램



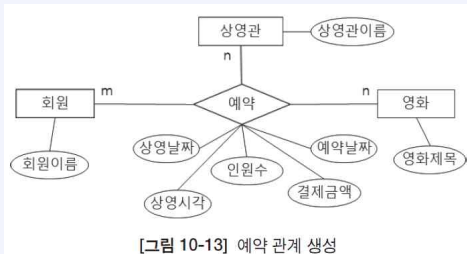
[그림 10-12] 상영관 개체, 상영 관계 생성

자동차극장 DB의 E-R 다이어그램

- ⑫ 회원은 원하는 영화를 미리 예약할 경우, 영화제목, 상영관이름, 상영날짜, 상영시각을 선택한 후 인원수를 입력하면 입력 내용에 회원이름, 영화제목, 상영관이름, 상영날짜, 상영시각, 결제금액과 예약날짜 정보가 추가로 출력된 예약확인증을 온라인으로 발급받는다.
- ⑬ 회원은 여러 영화를 예약할 수 있고, 하나의 영화는 여러 회원이 예약할 수 있다.

- 회원 개체와 영화 개체, 상영관 개체 사이는 예약 관계이다.
- 회원과 영화, 상영관 사이에 예약 관계가 맺어질 때마다 회원이름, 영화제목, 상영관이름, 상영날짜, 상영시각이 참조되어 인원수, 결제금액, 예약날짜 속성이 결정된다.
- 회원 개체는 영화 개체, 상영관 개체와 각각 일대다 예약 관계를 갖고 영화 개체도 회원 개체와 상영관 개체와 일대다 예약 관계를 갖으며 상영관 개체도 회원 개체, 영화 개체와 각각 일대다 예약 관계를 갖는다.

자동차극장 DB의 E-R 다이어그램

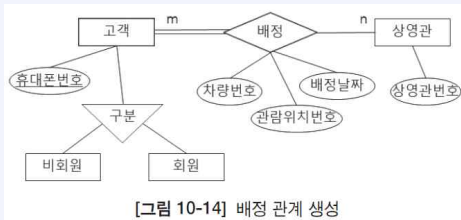


자동차극장 DB의 E-R 다이어그램

14 회원 또는 비회원 모두 자동차극장에 도착한 고객은 차량의 상영관번호, 관람위치번호를 배정받고
배정 시 차량번호와 배정날짜를 등록하고 휴대폰번호를 확인받는다.

- 고객 개체와 상영관 개체 사이는 배정 관계이다.
- 고객과 상영관 사이에 배정 관계가 맺어질 때마다 차량번호, 관람위치번호, 배정날짜 속성이 결정된다.
- 이때, 고객 개체의 주식별자는 하위 개체인 회원과 비회원의 공통 속성인 휴대폰번호 속성이다. 고객 개체의 속성은 상속되므로 비회원과 회원 개체의 휴대폰번호 속성은 각각 제거한다.
- 고객 개체는 상영관 개체와 일대다 배정 관계를 갖고 상영관 개체도 고객 개체와 일대다 배정 관계를 갖는다.

자동차극장 DB의 E-R 다이어그램

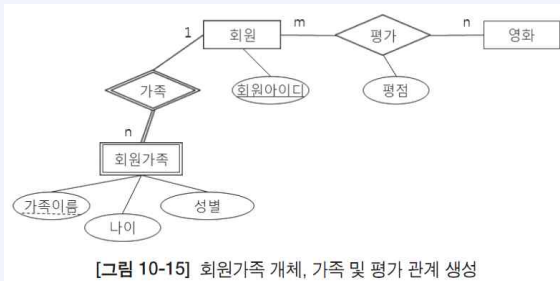


자동차극장 DB의 E-R 다이어그램

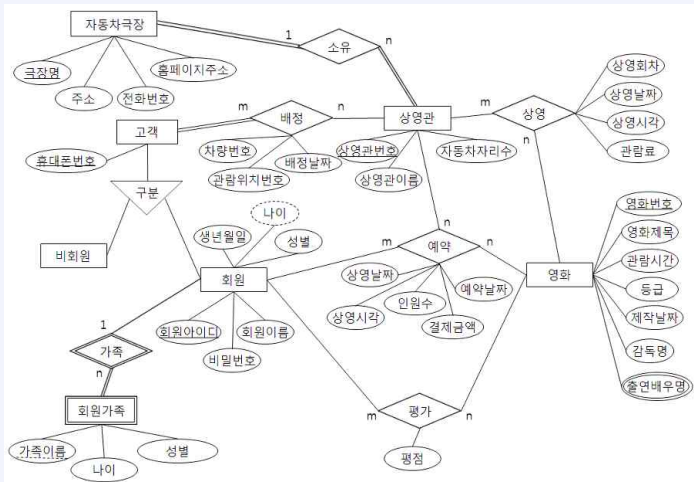
- 15 회원은 기본 20% 할인을 받고 회원가족도 10% 할인을 받을 수 있으므로 회원과 가족관계인 회원가족에 대해서도 가족이름, 나이, 성별 정보를 유지해야 한다.
- 16 회원만 영화에 대해 평가를 할 수 있고 평점을 부여한다.

- 회원 개체와 회원가족 개체 사이는 가족 관계이다.
- 회원 개체는 회원가족 개체와 일대다 가족 관계를 갖고 회원가족 개체는 회원 개체와 일대일 가족 관계를 갖는다.
- 회원가족 개체는 가족이름, 나이, 성별 속성을 갖는다.
- 이때, 회원가족 개체의 주식별자는 회원 개체의 주식별자 회원아이디와 가족이름 속성의 조합이다.
- 회원 개체와 영화 개체 사이는 평가 관계이다.
- 회원과 영화 사이에 평가 관계가 맺어질 때마다 평점 속성이 결정된다.
- 회원 개체는 영화 개체와 일대다 평가 관계를 갖고 영화 개체는 회원 개체와 일대다 평가를 갖는다.

자동차극장 DB의 E-R 다이어그램



【단계2】 자동차극장 DB의 전체 E-R 다이어그램



4. 논리적 설계

4.1 개체 변환

- 개체는 기본적으로 하나의 릴레이션으로 변환
 - ✓ 새로 생성된 릴레이션을 E-R 다이어그램의 개체를 표현한다고 해서 개체 릴레이션(entity relation)이라고 함.
 - ✓ 생성된 개체 릴레이션은 개체 이름이 릴레이션의 이름이 됨
- 개체의 키 속성은 릴레이션의 기본키 속성으로, 일반 속성은 릴레이션의 속성으로 변환
 - ✓ 개체의 속성 이름이 릴레이션의 속성 이름이 됨



[그림 10-17] 개체의 예(교수)

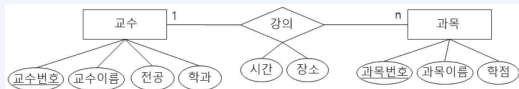
교수(교수번호, 교수이름, 전공, 학과)

4.2 관계 변환

● 일대다(1:n) 관계 변환

■ 1) 비식별 관계인 경우

- 일대다 관계 자체는 하나의 외래키 속성으로 변환
- 일대다 관계를 갖는 두 릴레이션에서 일측 개체 릴레이션의 기본키 속성을 가져와 다측 개체 릴레이션에 외래키 속성으로 추가하여 포함시킴
 - ✓ 추가하는 외래키 속성의 이름에 관계 이름을 포함하도록 변경('교수번호'를 '강의교수번호'처럼 변경)하면 그 의미를 명확히 할 수 있음
- 일대다 관계가 갖고 있던 모든 속성도 외래키를 추가한 다측 개체 릴레이션의 속성으로 함께 변환



[그림 10-18] 일대다 비식별 관계의 예(강의)

교수(교수번호, 교수이름, 전공, 학과)

과목(과목번호, 과목이름, 학점, 강의교수번호, 강의시간, 강의장소)

관계 변환

● 일대다(1:n) 관계 변환

■ 2) 식별 관계인 경우

- 비식별 관계와 동일하게 변환
- 추가하는 외래키를 다측 개체 릴레이션의 부분키와 조합하여 기본키로 지정
 - ✓ 관계를 맺는 개체가 기본키를 갖지 못하는 약 개체(weak entity)
 - ✓ 약 개체는 식별 관계를 갖는 강 개체의 존재 여부에 종속되기 때문



[그림 10-19] 일대다 식별 관계의 예(보유)

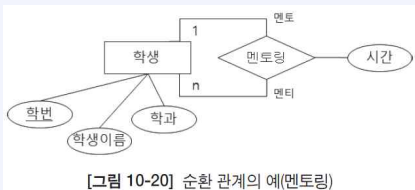
실습실(실습실번호, 실습실명, 전체좌석수, 관리자이름)
좌석(실습실번호, 좌석번호, 컴퓨터사양, 좌석유형)

관계 변환

● 일대다(1:n) 관계 변환

■ 3) 순환 관계인 경우

- 일대다 관계 자체는 하나의 외래키 속성으로 변환
- 일대다 관계를 갖는 두 릴레이션이 실제 물리적으로는 하나의 릴레이션이므로 기본키 속성을 같은 릴레이션의 외래키 속성으로 추가하여 포함시킴
 - ✓ 추가하는 외래키 속성의 이름은 기본키와 다르게 변경
- 순환 관계가 갖고 있던 모든 속성도 릴레이션의 속성으로 추가하여 포함시킴



학생(학번, 학생이름, 학과, 멘토학번, 멘토링시간)

관계 변환

● 일대일(1:1) 관계 변환

■ 1) 한쪽 개체만 전체참여하는 경우

- 가급적 관계에 전체참여하는 개체 릴레이션에 외래키를 추가함



[그림 10-21] 한쪽 전체참여 일대일 관계의 예(지도)

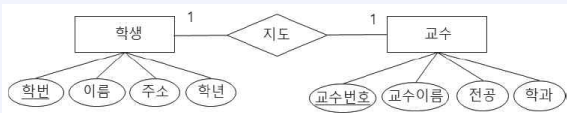
학생(학번, 이름, 주소, 학년, 지도교수번호)

교수(교수번호, 교수이름, 전공, 학과)

관계 변환

● 일대일(1:1) 관계 변환

■ 2) 양쪽 개체 모두가 부분참여하는 경우



[그림 10-22] 양쪽 부분참여 일대일 관계의 예(지도)

<방법1>

학생(학번, 이름, 주소, 학년, 지도교수번호)

교수(교수번호, 교수이름, 전공, 학과, 지도학생번호)

<방법2>

학생(학번, 이름, 주소, 학년)

교수(교수번호, 교수이름, 전공, 학과, 지도학생번호)

<방법3>

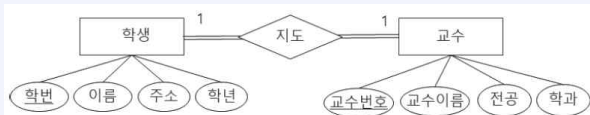
학생(학번, 이름, 주소, 학년, 지도교수번호)

교수(교수번호, 교수이름, 전공, 학과)

관계 변환

● 일대일(1:1) 관계 변환

■ 3) 양쪽 개체 모두가 전체참여하는 경우



[그림 10-23] 양쪽 전체참여 일대일 관계의 예(지도)

<방법1>

교수(교수번호, 교수이름, 전공, 학과, 지도학생번호)

학생(학번, 이름, 주소, 학년, 지도교수번호)

<방법2>

학생_교수_지도(학번, 교수번호, 이름, 주소, 학년, 교수이름, 전공, 학과)

관계 변환

● 다대다(m:n) 관계 변환

- 다대다 관계는 하나의 릴레이션으로 변환
 - ✓ 관계 이름이 새로운 릴레이션의 이름이 됨
- 관계의 속성은 릴레이션의 속성으로 변환
 - ✓ 관계 속성 이름이 새로운 릴레이션의 속성 이름이 됨
 - ✓ 새로 생성된 릴레이션을 E-R 다이어그램의 관계를 표현한다고 해서 관계 릴레이션 (relationship relation)이라고 함
- 관계 릴레이션은 보통 키 속성이 없으므로 양쪽 개체 릴레이션의 기본키 속성을 외래 키 속성으로 포함시키고 양쪽 기본키의 조합을 새로운 기본키로 지정함
 - ✓ 양쪽 기본키 속성의 이름이 같을 경우, 이름을 다르게 변경



[그림 10-24] 다대다 관계의 예(등록)

학생(학번, 이름, 주소, 학년)

등록(학번, 과목번호, 성적)

과목(과목번호, 과목이름, 학점)

4.3 속성 변환

- 개체의 일반 속성은 변환된 개체 릴레이션의 속성으로 변환
- 관계의 일반 속성은 관계가 외래키로 표현되는 개체 릴레이션 또는 관계 릴레이션의 속성으로 변환

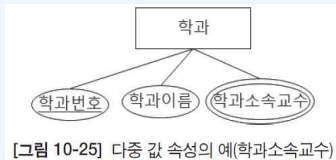
- 다중 값 속성인 경우

1) <규칙1>

- 다중 값 속성은 하나의 새로운 릴레이션으로 분리하여 변환
- 새로운 릴레이션은 다중 값 속성이 변환된 것으로 키 속성이 없으므로 분리 전 개체 릴레이션의 키 속성을 외래키로 포함시킴
- 다중 값 속성과 외래키를 조합하여 새로운 기본키로 지정
 - ✓ 다중 값 속성은 더 이상 다중 값을 갖지 않게 됨

2) <규칙2>

- 다중 값 속성이 갖는 최대 속성 값의 개수만큼 속성 이름만 다르게 여러 속성을 추가로 생성



<방법1>

학과(학과번호,학과이름)

학과소속교수(학과번호,학과소속교수)

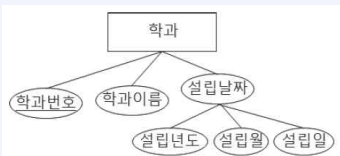
<방법2>

학과(학과번호,학과이름,학과소속교수1,학과소속교수2,학과소속교수3)

관계 변환

■ 복합 속성인 경우

- 기본적으로 복합 속성을 구성하는 가장 하위의 단순 속성들만 릴레이선의 속성으로 변환

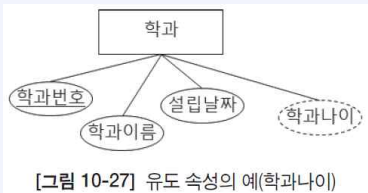


[그림 10-26] 복합 속성의 예(설립날짜)

학과(학과번호,학과이름,설립년도,설립월,설립일)

관계 변환

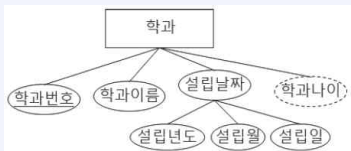
- 유도 속성인 경우
 - 기본적으로 중복을 최소화하기 위해 변환 과정에서 제외



학과(학과번호, 학과이름, 설립날짜)

4.4 기타 변환

■ 속성의 중복 허용 변환



[그림 10-28] 복합 및 유도 속성의 예

학과(학과번호, 학과이름, 설립년도, 설립월, 설립일, 설립날짜, 학과나이)

기타 변환

■ 일반화 관계 변환

1) <규칙1>

- 상위 개체와 하위 개체를 모두 각각 다른 릴레이션으로 변환
- 일반화 관계는 하위 개체 릴레이션에 각각 상위 개체의 기본키 속성을 가져와 외래키 속성으로 추가하여 변환
- 하위 개체인 릴레이션이 모두 키 속성이 없는 경우, 추가된 외래키 속성을 기본키로 지정
- 상위, 하위 개체의 속성도 각 개체 릴레이션의 속성으로 변환

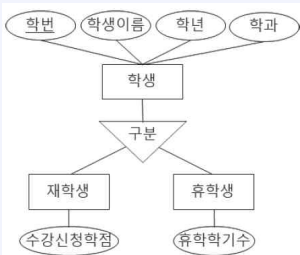
2) <규칙2>

- 3개 개체를 상위 개체에 해당하는 하나의 릴레이션으로 변환
- 일반화 관계는 상위 개체 릴레이션에 속성으로 추가하여 변환
- 하위 개체의 속성들도 상위 릴레이션의 속성으로 추가하여 포함

3) <규칙3>

- 3개 개체를 하위 개체에 해당하는 2개의 릴레이션으로 변환
- 상위 개체의 모든 속성들은 2개의 하위 개체 릴레이션의 속성으로 상속하여 모두 포함
- 하위 개체 릴레이션의 키 속성이 없는 경우, 상속된 상위 개체의 기본키 속성을 그대로 기본키로 지정
- 하위 개체의 속성들도 각 개체 릴레이션의 속성으로 변환

관계 변환



[그림 10-29] 일반화 관계의 예(구분)

<방법1>

학생(학번, 학생이름, 학년, 학과)

재학생(학번, 수강신청학점)

휴학생(학번, 휴학학기수)

<방법2>

학생(학번, 학생이름, 학년, 학과, 구분, 수강신청학점, 휴학학기수)

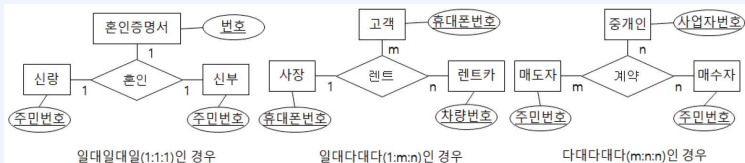
<방법3>

재학생(학번, 학생이름, 학년, 학과, 수강신청학점)

휴학생(학번, 학생이름, 학년, 학과, 휴학학기수)

관계 변환

다원 관계 변환



[그림 10-30] 다원 관계의 예(혼인,렌트,계약)

일대일대일인 경우

혼인 (신랑주민번호,신부주민번호,혼인증명서번호)

일대다대다인 경우

렌트 (고객휴대폰번호,렌트카차량번호,사장휴대폰번호)

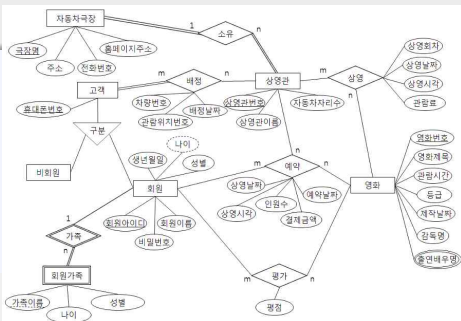
다대다대다인 경우

계약 (매도자주민번호,매수자주민번호,중개인사업자번호)

변환 결과 보완

자동차극장(극장명,주소,전화번호,홈페이지주소)
상영관(상영관번호,상영관이름,자동차자리수,자동차극장명)
고객(휴대폰번호,회원아이디,비밀번호,회원이름,생년월일,성별,나이)
회원()
비회원()
회원가족(회원아이디,가족이름,나이,성별)
영화(영화번호,영화제목,관람시간,등급,제작날짜,감독명)
영화출연배우(영화번호,출연배우명)
상영(상영관번호,영화번호,상영날짜,상영회차,관람료,상영시각)
예약(회원아이디,영화번호,상영날짜,상영시각,인원수,결제금액,예약날짜)
배정(휴대폰번호,상영관번호,관람위치번호,차량번호,배정날짜)
평가(회원아이디,영화번호,평점)

자동차극장(극장명,주소,전화번호,홈페이지주소)
상영관(상영관번호,상영관이름,자동차자리수,자동차극장명)
①고객(휴대폰번호,회원아이디,비밀번호,회원이름,생년월일,성별,나이)
②회원(회원아이디,비밀번호,회원이름,생년월일,성별,휴대폰번호,⑤나이)
②비회원(휴대폰번호)
회원가족(회원아이디,가족이름,나이,성별)
영화(영화번호,영화제목,관람시간,등급,제작날짜,감독명)
영화출연배우(영화번호,출연배우명)
⑥상영(상영관번호,영화번호,상영날짜,상영회차,관람료,상영시각)
⑦예약(회원아이디,영화번호,상영날짜,상영시각,인원수,결제금액,예약날짜)
③배정(휴대폰번호,상영관번호,관람위치번호,차량번호,배정날짜)
④비회원배정(휴대폰번호,상영관번호,관람위치번호,차량번호,배정날짜)
④회원배정(회원아이디,상영관번호,관람위치번호,차량번호,배정날짜)
평가(회원아이디,영화번호,평점)



자동차극장(극장명,주소,전화번호,홈페이지주소)

상영관(상영관번호,상영관이름,자동차자리수,자동차극장명)

고객(휴대폰번호,회원아이디,비밀번호,회원이름,생년월일,성별,나이)

회원()

비회원()

회원가족(회원아이디,가족이름,나이,성별)

영화(영화번호,영화제목,관람시간,등급,제작날짜,감독명)

영화출연배우(영화번호,출연배우명)

상영(상영관번호,영화번호,상영날짜,상영회차,관람료,상영시각)

예약(회원아이디,영화번호,상영날짜,상영시각,인원수,결제금액,예약날짜)

배정(휴대폰번호,상영관번호,관람위치번호,차량번호,배정날짜)

평가(회원아이디,영화번호,평점)

【단계3】 자동차극장 DB의 논리적 데이터베이스 스키마

자동차극장(극장명, 주소, 전화번호, 홈페이지주소)

상영관(상영관번호, 상영관이름, 자동차자리수, 자동차극장명)

회원(회원아이디, 비밀번호, 회원이름, 휴대폰번호, 생년월일, 성별)

비회원(휴대폰번호)

회원가족(회원아이디, 가족이름, 나이, 성별)

영화(영화번호, 영화제목, 관람시간, 등급, 제작날짜, 감독명)

영화출연배우(영화번호, 출연배우명)

상영(상영관번호, 영화번호, 상영날짜, 상영회차, 관람료, 상영시각)

예약(회원아이디, 영화번호, 상영날짜, 상영시각, 인원수, 결제금액, 예약날짜)

비회원배정(휴대폰번호, 상영관번호, 관람위치번호, 차량번호, 배정날짜)

회원배정(회원아이디, 상영관번호, 관람위치번호, 차량번호, 배정날짜)

평가(회원아이디, 영화번호, 평점)

5. 물리적 설계

5.1 영문 변환

```
cinema(c_name, c_address, c_phone, c_homepage)
theater(t_id, t_name, seats_cap, c_name)
member(m_id, password, m_name, m_mobile, m_birthdate, m_gender)
guest(g_mobile)
family(m_id, f_name, f_age, f_sex)
movie(movie_id, title, length, class, mdate, director_name)
actor(movie_id, actor_name)
play(t_id, movie_id, p_date, p_no, p_charge, p_time)
reservation(m_id, movie_id, p_date, p_time, persons_no, paid_amount, r_date)
g_assign(g_mobile, t_id, seats_no, car_no, assign_date)
m_assign(m_id, t_id, seats_no, car_no, assign_date)
evaluation(m_id, movie_id, grade)
```


자동차극장(극장명,주소,전화번호,홈페이지주소)
 상영관(상영관번호,상영관이름,자동차자리수,자동차극장명)
 회원(회원아이디,비밀번호,회원이름,휴대폰번호,생년월일,성별)
 비회원(휴대폰번호)
 회원가족(회원아이디,가족이름,나이,성별)
 영화(영화번호,영화제목,관람시간,등급,제작날짜,감독명)
 영화출연배우(영화번호,출연배우명)
 상영(상영관번호,영화번호,상영날짜,상영회차,관람료,상영시간)
 예약(회원아이디,영화번호,상영날짜,상영시간,인원수,결제금액,예약날짜)
 비회원배정(휴대폰번호,상영관번호,관람위치번호,차량번호,배정날짜)
 회원배정(회원아이디,상영관번호,관람위치번호,차량번호,배정날짜)
 평가(회원아이디,영화번호,평점)

```

cinema(c_name, c_address, c_phone, c_homepage)
theater(t_id, t_name, seats_cap, c_name)
member(m_id, password, m_name, m_mobile, m_birthdate, m_gender)
guest(g_mobile)
family(m_id, f_name, f_age, f_sex)
movie(movie_id, title, length, class, mdate, director_name)
actor(movie_id, actor_name)
play(t_id, movie_id, p_date, p_no, p_charge, p_time)
reservation(m_id, movie_id, p_date, p_time, persons_no, paid_amount, r_date)
g_assign(g_mobile, t_id, seats_no, car_no, assign_date)
m_assign(m_id, t_id, seats_no, car_no, assign_date)
evaluation(m_id, movie_id, grade)
    
```

5.2 뷰 생성

- 성능, 보안 그리고 편의성을 위해 필요할 경우 뷰를 생성
- 예) '회원'과 '비회원'을 포함한 전체 '고객'에 대한 차량좌석지정 정보를 갖는 뷰를 생성

```
v_customer_assign(mobile, t_id, seats_no, car_no, assign_date)
```

```
CREATE VIEW v_customer_assign  
SELECT g_mobile, t_id, seats_no, car_no, assign_date  
FROM g_assign  
UNION  
SELECT m_mobile, t_id, seats_no, car_no, assign_date  
FROM m_assign JOIN member ON m_assign.m_id = member.m_id
```

5.3 인덱스 생성

```
CREATE INDEX idx_member_m_name  
ON member (m_name) ;
```

5.4 내부 자동생성 열 추가

- 필요할 경우, 기본키를 대신하는 내부키를 추가
 - 내부키 값은 DBMS에 의해 자동 생성되는 값으로 각 튜플들을 식별
 - 보통 기본키가 많은 열로 구성된 복합키이거나 기본키 값의 크기가 커서 빈번한 비교 검색에 따른 성능 저하가 우려될 때 내부키 열을 고려

```
g_assign(grow_id, g_mobile, t_id, seats_no, car_no, assign_date)  
m_assign(mrow_id, m_id, t_id, seats_no, car_no, assign_date)
```

- MySQL DBMS가 테이블에 행이 추가될 때마다 자동으로 유일한 값을 생성하여 입력하도록 함

```
CREATE TABLE g_assign(grow_id int NOT NULL AUTO_INCREMENT PRIMARY KEY,  
...  
);
```

【4단계】 자동차극장 DB의 물리적 데이터베이스 스키마

• 테이블

```
cinema(c_name, c_address, c_phone, c_homepage)
theater(t_id, t_name, seats_cap, c_name)
member(m_id, password, m_name, m_mobile, m_birthdate, m_gender)
guest(g_mobile)
family(m_id(FK), f_name, f_age, f_sex)
movie(mid, title, length, class, mdate, director)
actor(mid(FK), a_name)
play(t_id(FK), mid(FK), p_no, p_date, p_charge, p_time)
reservation(m_id(FK), mid(FK), p_date, p_time, persons_no, paid_amount, r_date)
g_assign(grow_id, g_mobile(FK), t_id(FK), seats_no, car_no, assign_date)
m_assign(mrow_id, m_id(FK), t_id(FK), seats_no, car_no, assign_date)
evaluation(m_id(FK), mid(FK), grade)
```

• 뷰

```
v_customer_assign(mobile, t_id, seats_no, car_no, assign_date)
```

...

• 인덱스

```
idx_member_m_name
```

...

• SQL 명령문

```
CREATE TABLE cinema (
...
// 데이터 유형, NULL 허용유무, 도메인, 기본 값, 체크, 규칙, 각종 키 등 제약 조건 설정
...
)
```