

DB 설계

김근형 강사

데이터베이스 설계 개요

▶ 데이터베이스 설계 정의

- ▶ 사용자 요구조건 에서부터 데이터베이스 구조를 도출해 내는 과정
- ▶ 데이터들을 효과적으로 관리하기 위하여 데이터베이스의 구조를 조직화하는 작업

▶ 데이터베이스 설계 목적

- ▶ 이해관계자의 데이터 관점 요구사항에 대한 정확한 이해 및 추상화
- ▶ 데이터를 중심으로 한 이해관계자 간의 원활한 의사소통 수단
- ▶ 데이터 중심의 분석방법

데이터베이스 설계 개요

▶ 데이터베이스 설계 시 고려사항

항목	설명
제약조건	저장된 데이터 값이 만족해야 될 주어진 조건
데이터베이스 무결성	갱신, 삽입, 삭제 등의 연산이 수행된 뒤에도 데이터 값은 제약조건을 만족해야하는 조건
일관성	저장된 두 데이터 값 또는 특정 질의에 대한 응답들에 모순성 없이 일치하는 특성
회복	시스템에 장애가 발생했을 때 장애 발생 직전의 일관된 데이터 상태로 돌아가는 기법
보안	불법적인 데이터의 변경이나 손실 또는 노출에 대한 보호
효율성	응답 시간의 단축, 저장공간의 최적화, 시스템 생산성이 포함
데이터베이스 확장성	시스템 운영에 영향을 주지 않으면서 새로운 데이터를 계속적으로 추가 가능한 기법

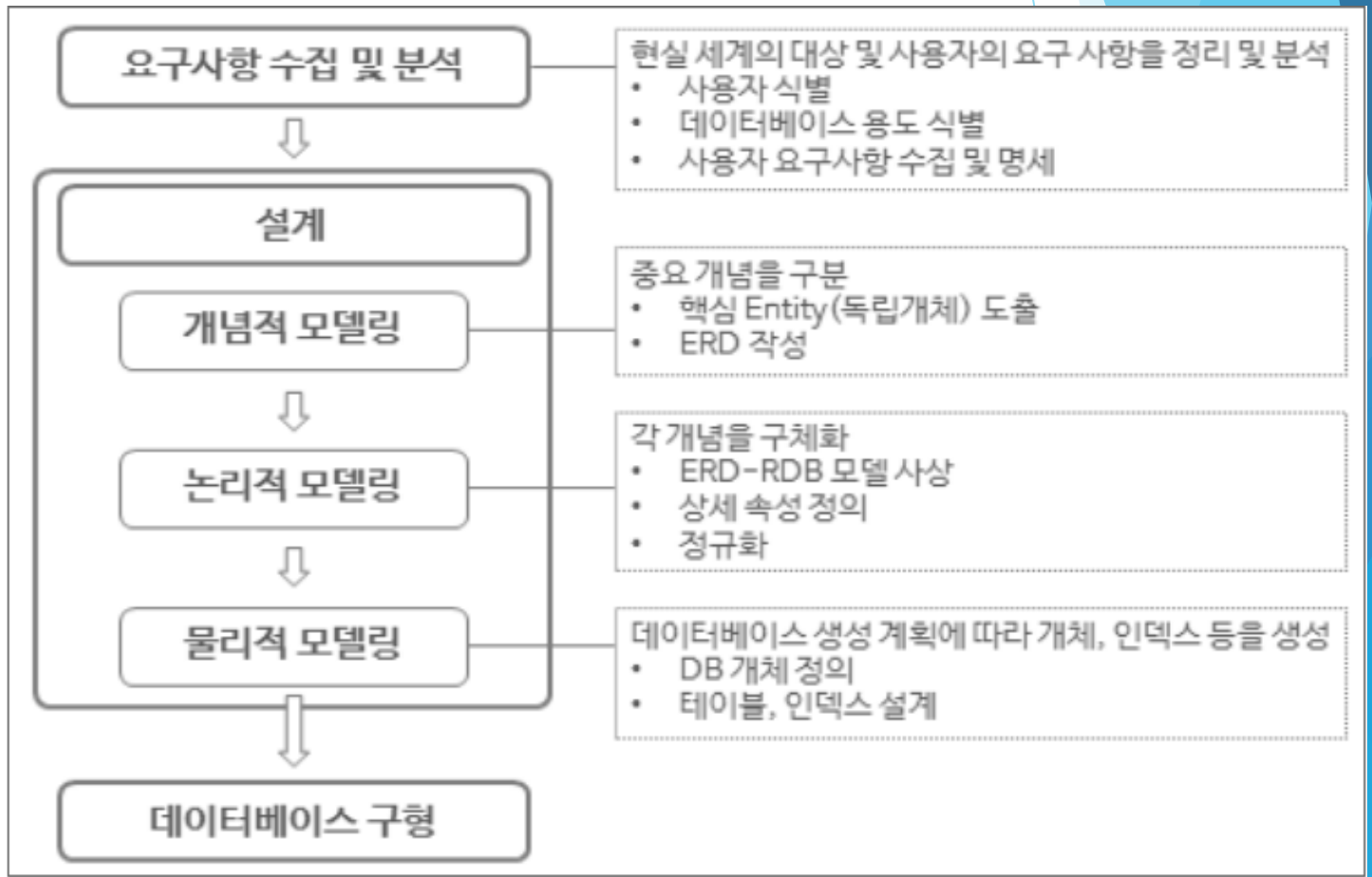
데이터베이스 설계 개요

▶ 데이터베이스 설계 시 고려사항

항목	설명
제약조건	저장된 데이터 값이 만족해야 될 주어진 조건
데이터베이스 무결성	갱신, 삽입, 삭제 등의 연산이 수행된 뒤에도 데이터 값은 제약조건을 만족해야하는 조건
일관성	저장된 두 데이터 값 또는 특정 질의에 대한 응답들에 모순성 없이 일치하는 특성
회복	시스템에 장애가 발생했을 때 장애 발생 직전의 일관된 데이터 상태로 돌아가는 기법
보안	불법적인 데이터의 변경이나 손실 또는 노출에 대한 보호
효율성	응답 시간의 단축, 저장공간의 최적화, 시스템 생산성이 포함
데이터베이스 확장성	시스템 운영에 영향을 주지 않으면서 새로운 데이터를 계속적으로 추가 가능한 기법

데이터베이스 설계 개념도 및 설계 단계

▶ 데이터베이스 설계 개념도



데이터베이스 설계 개념도 및 설계 단계

▶ 데이터베이스 설계 단계별 특성

단계	주요관점	설명
요구조건 분석	문서화	<ul style="list-style-type: none">- 개체, 속성, 관계, 제약조건과 같은 정적정보 구조- 트랜잭션 유형, 트랜잭션 실행빈도와 같은 동적 DB 처리 요구조건- 기관의 경영목표, 정책 및 규정과 같은 기관적 제약조건
개념적 설계	현실세계 추상화	<ul style="list-style-type: none">- 현실세계를 데이터관점으로 추상화 단계- DBMS를 고려하지 않는 독립적 설계- 데이터베이스의 개념적 스키마 (E-R 다이어그램) 구성
논리적 설계	데이터모형기반 설계	<ul style="list-style-type: none">- 특정 데이터모델(계층형, 관계형, 객체지향형 등)을 적용한 설계- 사용 할 DBMS 특성을 고려한 설계- 데이터베이스의 논리적 스키마 (릴레이션 스키마) 생성
물리적 설계	DBMS 기반 설계	<ul style="list-style-type: none">- 특정 DBMS의 물리적 구조와 내부적인 저장구조, 분산형태, 데이터타입의 특징, 인덱스의 특징 등을 구체화하는 설계단계- 오브젝트, 접근방법, 트랜잭션분석, 인덱스, 뷰, 데이터베이스 용량설계 등을 수행- 데이터베이스의 물리적 스키마 생성 (하드웨어/운영체제 특성 고려)

데이터베이스 요구사항 수집 및 분석

▶ 데이터베이스 요구사항 수집 및 분석 항목

항목	설명	
요구조건 수집	<ul style="list-style-type: none"> - 데이터베이스를 사용하는 주요 분야와 사용자 그룹을 식별 - 식별된 그룹들의 업무, 데이터 종류, 용도, 처리형태, 흐름 및 제약/요구조건 수집 - 트랜잭션에 대한 입력과 출력 데이터 식별 - 설문지/인터뷰 및 회의를 통해 도출하고 문서화 진행 	
제약조건 식별	<ul style="list-style-type: none"> - 경영정책, 내/외부적 환경, 조직의 정보전략을 연구/분석하여 요구조건 명세에 반영 - 반영된 요구조건 명세는 데이터베이스 설계 범위를 결정짓는 중요 요소로 활용 	
명세 작성	<ul style="list-style-type: none"> - 관련 데이터 요소, 트랜잭션, 처리 특성에 대한 명세 포함 - 빈번한 변경을 방지하여 안정적인 개발 환경 조성 - 작업-데이터 관계, 데이터 요소간 제약조건, 유일성 및 함수 종속성 명세 포함 	
명세 검토	<ul style="list-style-type: none"> - 정보 처리 요구조건을 조직하여 표현하기 위한 소프트웨어 공학기법 사용 - 사용자 그룹과 다시 검토하고 확인 한 뒤 최종 시스템 명세로 확정 	
명세검토 활용 기법	소프트웨어 공학	<ul style="list-style-type: none"> - HIPO, SADT, DFD, Orr-Warnier, Nassi-Schneiderman - 정보 처리 요구조건을 조직하여 표현하기 위한 다이어그램 방식
	프로그램 활용 (PSL/PSA)	<ul style="list-style-type: none"> - 명세에 대한 일관성과 완전성을 검사하기 위한 자동화 도구 활용 - 요구조건을 설계 데이터베이스에 저장하여 설계 과정동안 활용

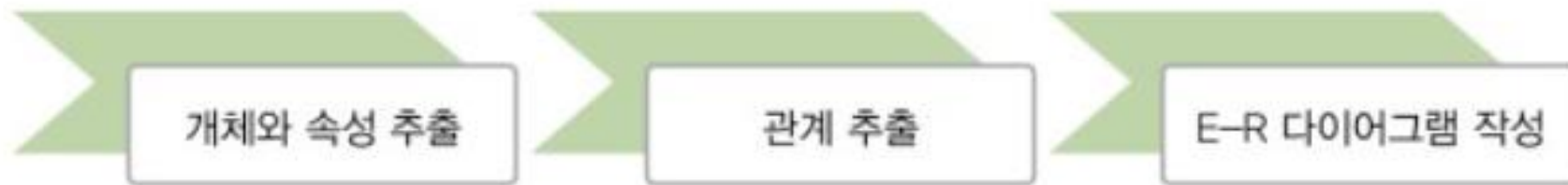
데이터베이스 요구사항 수집 및 분석

▶ 요구사항 분석설계 예제

- ▶ 한빛 항공사에 회원으로 가입하려면 회원아이디, 비밀번호, 성명, 신용카드 정보를 입력해야 한다
- ▶ 회원의 신용카드 정보는 여러 개를 저장할 수 있는데, 세부적으로는 신용카드번호, 유효기간을 저장할 수 있다
- ▶ 한빛 항공사에서는 보유한 비행기에 대해 비행기번호, 출발날짜, 출발시간 정보를 저장하고 있다
- ▶ 한빛 항공사에서는 좌석에 대한 좌석번호, 등급 정보를 저장하고 있다
- ▶ 회원은 좌석을 예약하는데, 회원 한 명은 좌석을 하나만 예약할 수 있고, 한 좌석은 회원 한명만 예약할 수 있다
- ▶ 비행기에는 좌석이 존재하는데, 비행기 하나에는 좌석이 여러 개 존재할 수 있고 한 좌석은 반드시 하나의 비행기에만 존재해야 한다.
- ▶ 그리고 좌석은 비행기가 없으면 의미가 없다.

데이터베이스 개념적 설계

- ▶ 개념적 설계 정의 및 특징
 - ▶ 요구사항 분석 결과를 개념적인 데이터 모델(E-R 모델)을 사용하여 고차원적인 표현 기법으로 기술하는 방식
 - ▶ DBMS를 고려하지 않는 독립적 설계
 - ▶ 데이터베이스의 개념적 스키마 (E-R 다이어그램) 구성
- ▶ 개념적 스키마 설계 과정



데이터베이스 개념적 설계

▶ 개념적 스키마 설계 과정

개체와 속성 추출	- 요구사항 분석단계에서 개체와 속성을 추출하는 일
관계 추출	- 개체간의 의미있는 연관성이며, 추출한 관계의 매핑 카디널리티를 기준으로 하여 관계(1:1, 1:N, N:M) 결정 - 개체가 관계에 필수적 참여인지 또는 선택적 참여인지를 구분짓는 참여특성을 결정
E-R 다이어그램 작성	- 요구 사항 명세서에서 추출한 개체, 속성 및 관계를 E-R 다이어그램으로 작성

데이터베이스 개념적 설계

- ▶ 개체(Entity)와 속성(attribute) 추출
 - ▶ 개체는 현실에서의 사물이나 사람이 생각하는 개념
 - ▶ 개체를 나타내는 속성이 있고 여러 관련 속성이 모여 하나의 정보 단위를 이루는 것이 개체가 된다.
 - ▶ 요구사항에서 개체는 대부분 명사로 이루어져 있지만, 속성과 구별할 필요가 있다.

데이터베이스 개념적 설계

▶ 개체(Entity)와 속성(attribute) 추출 예제

- ▶ 한빛 항공사에 **회원**으로 가입하려면 **회원 아이디**, **비밀번호**, **성명**, **신용카드** 정보를 입력해야 한다
- ▶ 회원의 **신용카드** 정보는 여러 개를 저장할 수 있는데, 세부적으로는 **신용카드번호**, **유효기간**을 저장할 수 있다
- ▶ 한빛 항공사에서는 보유한 **비행기**에 대해 **비행기번호**, **출발날짜**, **출발시간** 정보를 저장하고 있다
- ▶ 한빛 항공사에서는 **좌석**에 대한 **좌석번호**, **등급** 정보를 저장하고 있다
- ▶ 회원은 좌석을 예약하는데, 회원 한 명은 좌석을 하나만 예약할 수 있고, 한 좌석은 회원 한명만 예약할 수 있다
- ▶ 비행기에는 좌석이 존재하는데, 비행기 하나에는 좌석이 여러 개 존재할 수 있고 한 좌석은 반드시 하나의 비행기에만 존재해야 한다.
- ▶ 그리고 좌석은 비행기가 없으면 의미가 없다.

개체	속성
회원	<u>회원아이디</u> , 비밀번호, 성명, 신용카드
신용카드	<u>신용카드번호</u> , 유효기간
비행기	<u>비행기번호</u> , 출발날짜, 출발시간
좌석	<u>좌석번호</u> , 등급

데이터베이스 개념적 설계

▶ 개체 간의 관계 추출

- ▶ 개체와 속성이 구별되었다면 개체 간의 관계를 추출한다.
- ▶ 개체 간의 관계도 여러 가지로 분류해서 정의된다.
 - ▶ 일대일(1:1), 일대다(1:N), 다대다(N:M)
 - ▶ 관계 : 선택적인 관계, 필수적인 관계
- ▶ 요구사항에서 개체 간의 관계는 동사로 묘사되기 때문에 동사부터 찾으면 된다.

데이터베이스 개념적 설계

▶ 개체 간의 관계 추출

▶ 필수적으로 관계에 참여 (전체 참여 : Total Participation)

- ▶ $[A : B]$ 관계에서 개체 집합 B 의 모든 개체가 $[A : B]$ 관계에 참여한다
- ▶ 개체 A 에 대해 개체 조건을 만족하는 개체가 반드시 존재할 경우에 필수적인 관계라고 본다.

▶ 선택적으로 관계에 참여 (부분 참여 : Partial Participation)

- ▶ $[A : B]$ 관계에서 개체 집합 B 의 일부 개체만 $[A : B]$ 관계에 참여한다
- ▶ 객체 A 에 대해 개체 조건을 만족하는 개체가 존재할수도, 존재하지 않을 수도 있는 경우에 선택적인 관계라고 본다.

데이터베이스 개념적 설계

▶ 개체 간의 관계 추출

▶ 예1. [학과 : 교수]

- ▶ 학과(필수): 한 학과에는 여러 교수가 소속된다
- ▶ 교수(필수): 한 교수는 한 학과에만 소속된다

▶ 예2. [회원 : 주문]

- ▶ 회원(선택): 한 회원은 여러번 주문할 수 있다
- ▶ 주문(필수): 한 주문은 한 회원에 의해 주문된다

▶ 예3. [교수 : 과목]

- ▶ 교수(선택): 한 교수는 여러 과목을 강의할 수 있다
- ▶ 과목(필수): 한 과목은 한 교수에 의해서 강의되어야 한다

▶ 예4: [주문 : 주문목록]

- ▶ 주문(필수): 한 주문은 여러 개의 주문목록을 포함한다
- ▶ 주문목록(필수): 한 주문목록에 하나의 주문내용에 포함된다

▶ 예5 : [고객 : 책] - 구매 관계

- ▶ 고객(필수): 모든 고객이 책을 반드시 구매해야 한다
- ▶ 책(선택): 고객이 구매하지 않은 책이 존재할 수 있다.

데이터베이스 개념적 설계

▶ 개체 간의 관계 추출 예제

- ▶ 한빛 항공사에 회원으로 가입하려면 회원 아이디, 비밀번호, 성명, 신용카드 정보를 입력해야 한다
- ▶ 회원의 신용카드 정보는 여러 개를 저장할 수 있는데, 세부적으로는 신용카드번호, 유효기간을 저장할 수 있다
- ▶ 한빛 항공사에서는 보유한 비행기에 대해 비행기번호, 출발날짜, 출발시간 정보를 저장하고 있다
- ▶ 한빛 항공사에서는 좌석에 대한 좌석번호, 등급 정보를 저장하고 있다
- ▶ 회원은 좌석을 예약하는데, 회원 한 명은 좌석을 하나만 예약할 수 있고, 한 좌석은 회원 한명만 예약할 수 있다
- ▶ 비행기에는 좌석이 존재하는데, 비행기 하나에는 좌석이 여러 개 존재할 수 있고 한 좌석은 반드시 하나의 비행기에만 존재해야 한다.
- ▶ 그리고 좌석은 비행기가 없으면 의미가 없다.

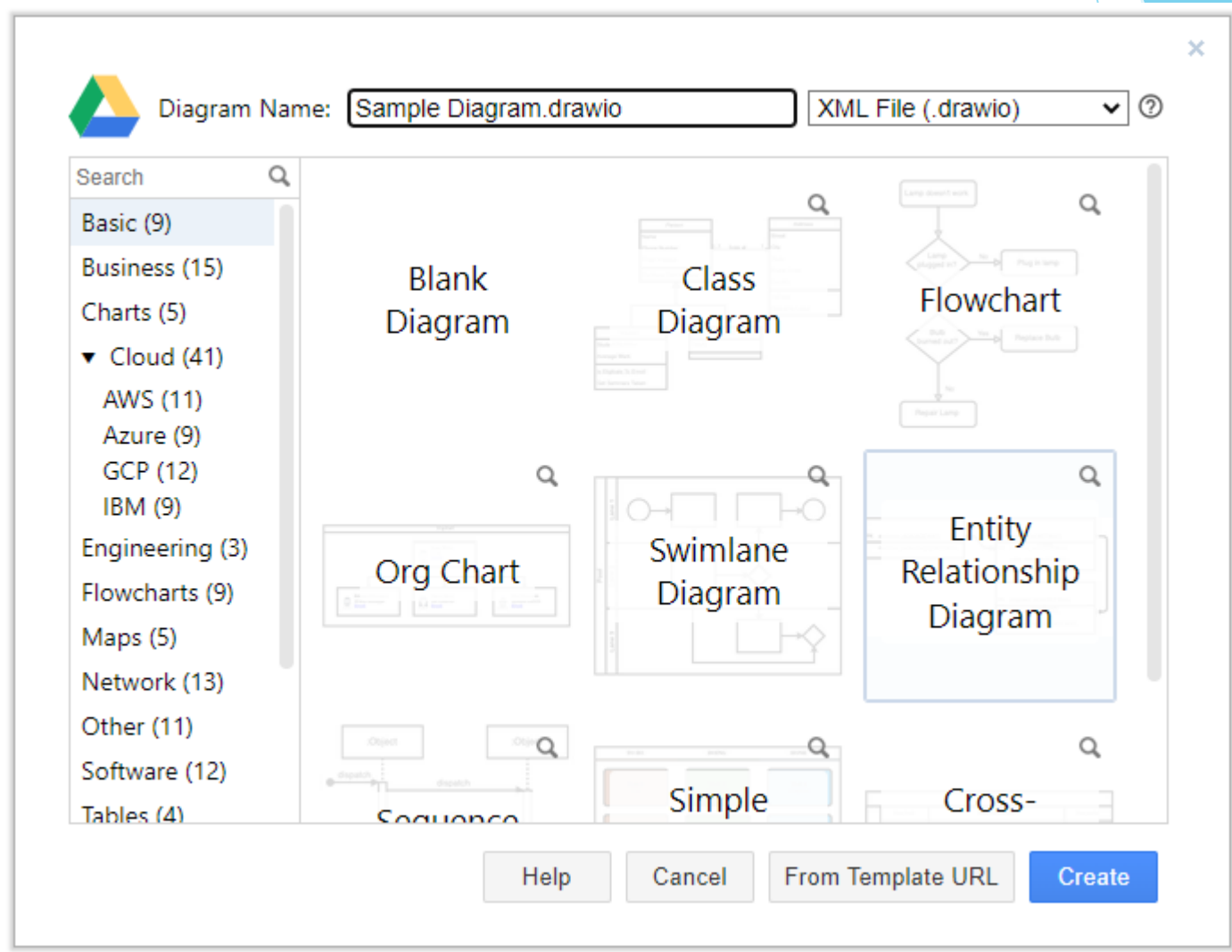
데이터베이스 개념적 설계

▶ 개체 간의 관계 추출 예제

관계	관계에 참여하는 개체	관계유형	관계 속성
보유	회원(선택): 한 회원은 여러 신용카드를 가질 수 있다 신용카드(필수): 한 신용카드는 한 회원이 보유한다	1:N	
예약	회원(선택): 한 회원은 여러 비행기를 예약할 수 있다 비행기(필수): 비행기 한 좌석은 한 회원이 예약한다	1:N	요구사항에서는 없음 ex. 예약번호, 예약일자
존재	비행기(필수): 비행기 하나에는 좌석이 여러개 존재한다 좌석(필수): 한 좌석은 하나의 비행기에만 존재해야 한다	1:N	비행기 없이 좌석이 존재할 수 없음

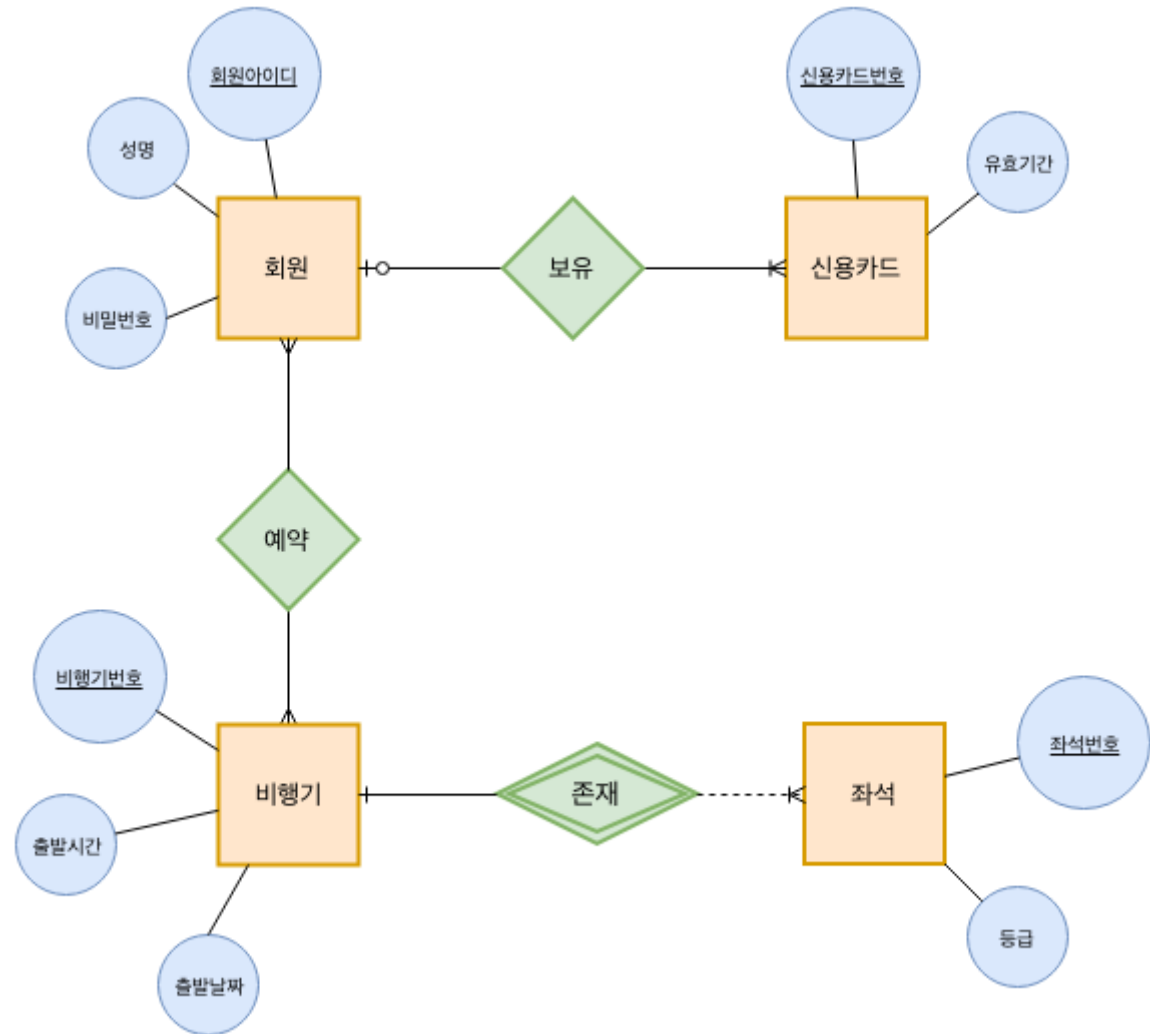
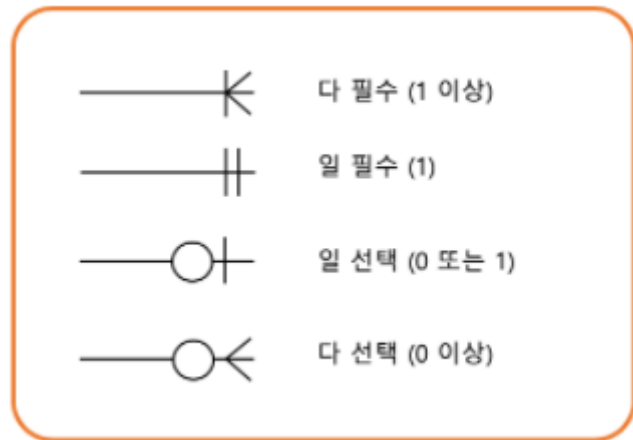
데이터베이스 개념적 설계

- ▶ E-R 다이어그램 작성
 - ▶ draw.io



데이터베이스 개념적 설계

- ▶ E-R 다이어그램 작성
 - ▶ draw.io를 사용한 ER Diagram

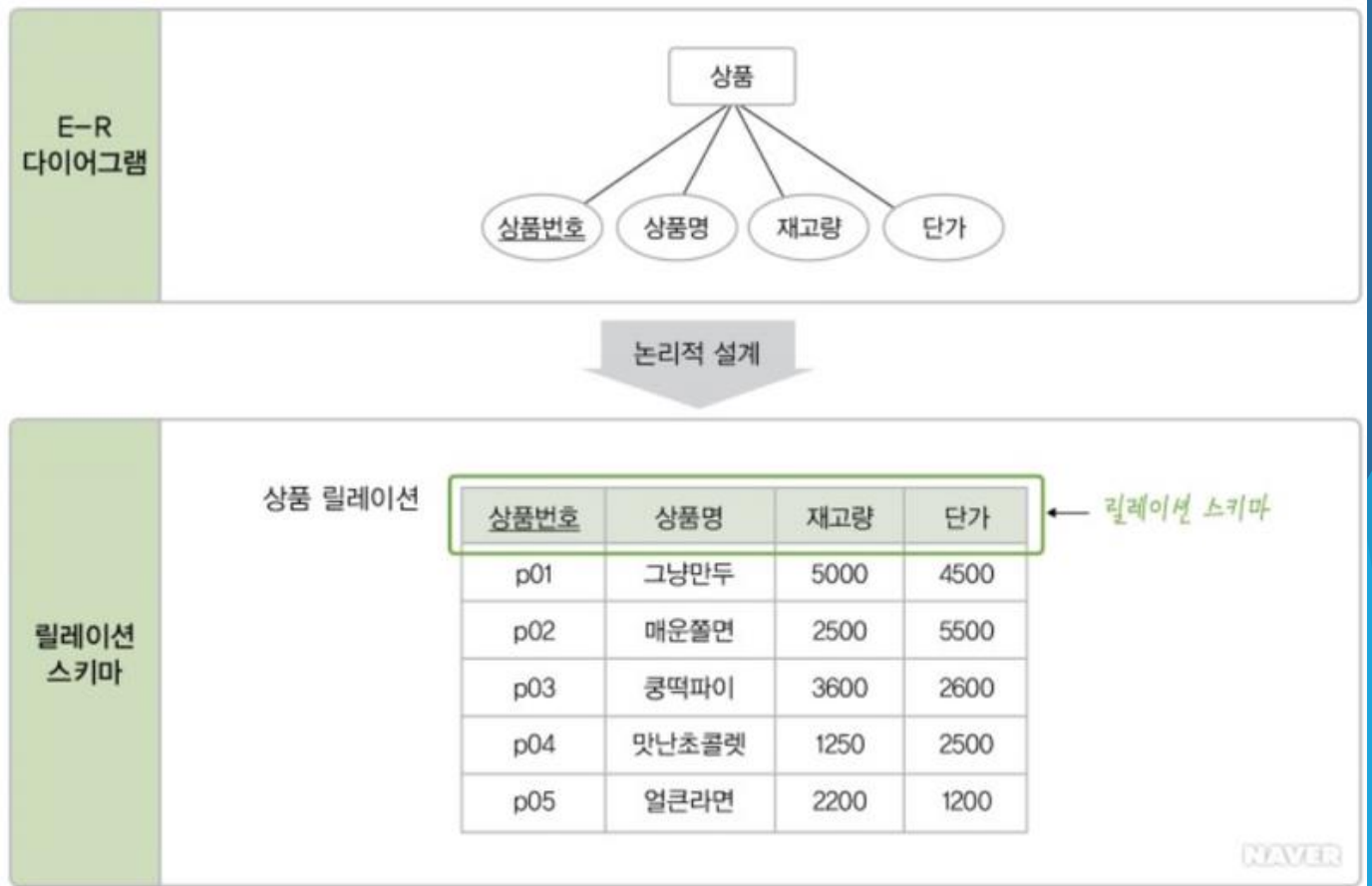


데이터베이스 논리적 설계

- ▶ ERD에서 릴레이션 스키마를 만들려면 다음 5가지 릴레이션 변환 규칙에 따라서 릴레이션 스키마로 변환해주면 된다.
 - ▶ 규칙1 : 모든 개체는 릴레이션으로 변환한다
 - ▶ 규칙2 : N:M 관계는 릴레이션으로 변환한다
 - ▶ 관계의 이름을 릴레이션 이름으로 하고 관계의 속성도 릴레이션의 속성으로 변환한다
 - ▶ 규칙3 : 1:N 관계는 외래키로 표현한다
 - ▶ 규칙3-1: 일반적인 1:N 관계는 외래키로 표현한다
 - ▶ 규칙3-2: 약한 개체가 참여하는 1:N 관계는 외래키로 포함해서 기본키로 지정한다
 - ▶ 규칙4 : 1:1 관계는 외래키로 표현한다
 - ▶ 규칙4-1: 일반적인 1:1 관계는 외래키를 서로 주고 받는다
 - ▶ 규칙4-2: 1:1 관계에 필수적으로 참여하는 개체의 릴레이션만 외래키를 받는다
 - ▶ 규칙4-3: 모든 개체가 1:1 관계에 필수적으로 참여하면 릴레이션 하나로 합친다
 - ▶ 규칙5 : 다중 값 속성은 독립 릴레이션으로 변환한다

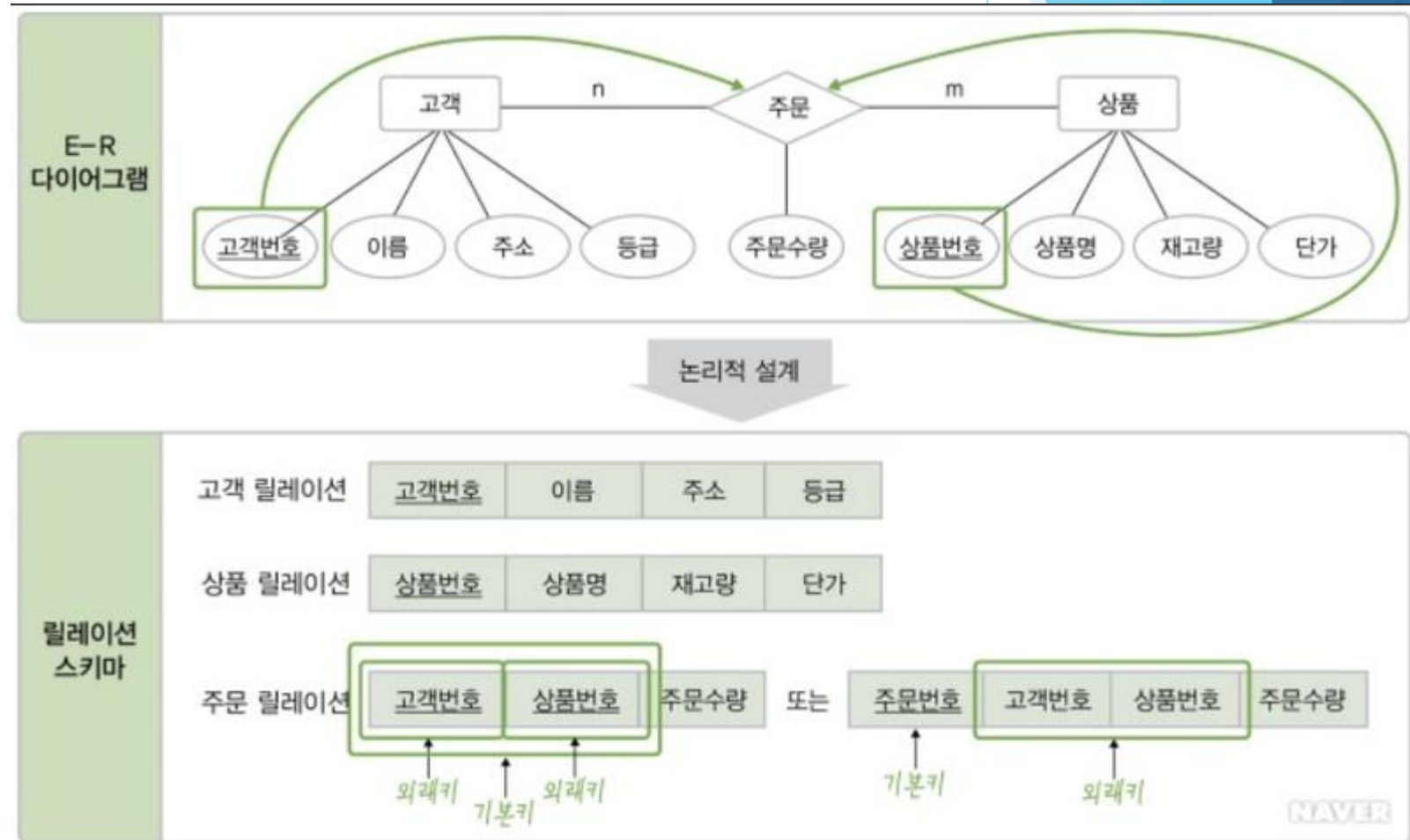
데이터베이스 논리적 설계

- ▶ 규칙1 : 모든 개체는 릴레이션으로 변환한다
 - ▶ E-R 다이어그램의 각 개체를 하나의 릴레이션으로 변환. 개체의 이름을 릴레이션 이름으로, 속성은 릴레이션 속성으로 변환



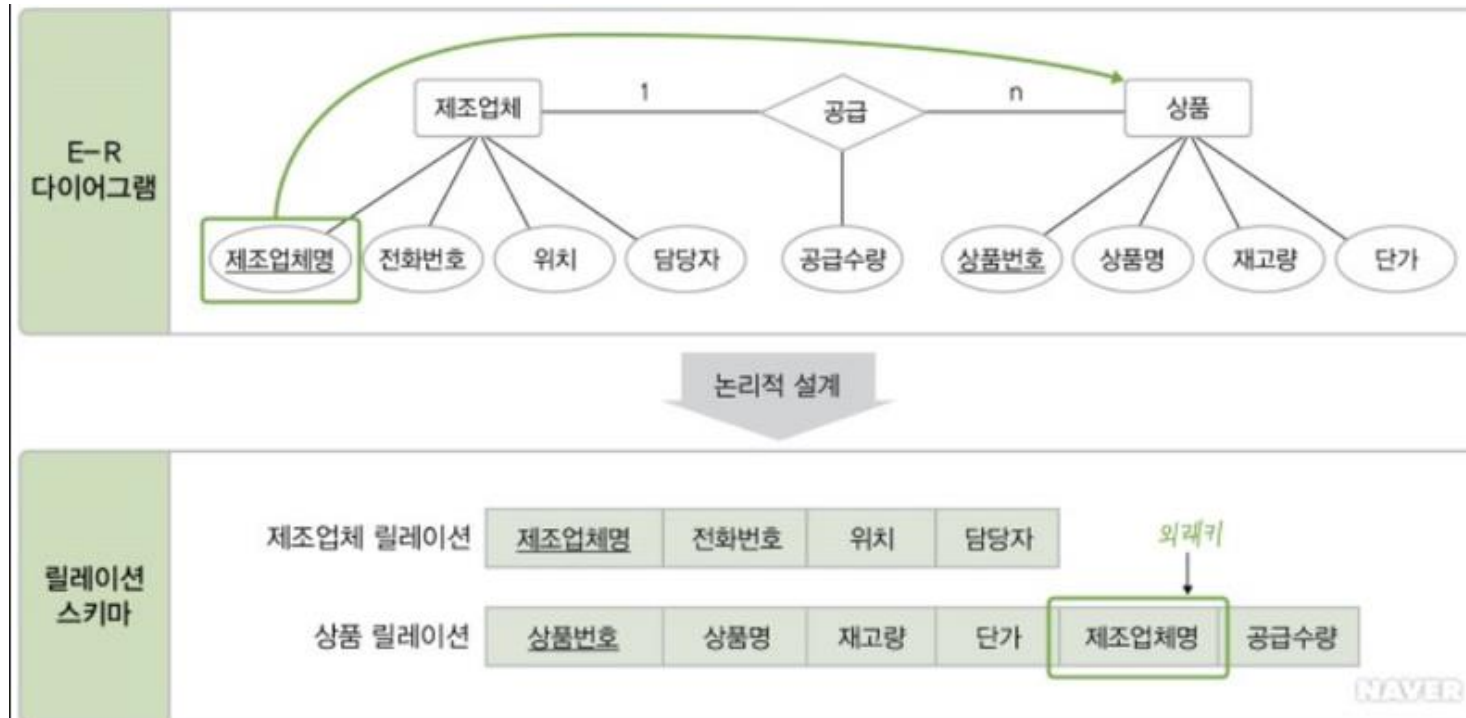
데이터베이스 논리적 설계

- ▶ 규칙2 : N:M 관계는 릴레이션으로 변환한다
 - ▶ E-R 다이어그램의 관계는 릴레이션 이름, 관계의 속성은 릴레이션 속성으로 변환
 - ▶ 관계를 맺고 있는 개체는 제1 규칙에 따라 변환한 후 이 릴레이션의 기본키를 관계 릴레이션에 포함시키고 외래키로 지정 (외래키 지정 시 이름이 같을 경우 변경)



데이터베이스 논리적 설계

- ▶ 규칙3 : 1:N 관계는 외래키로 표현한다
 - ▶ 일반적인 1:N 관계는 릴레이션으로 변환하지 않고 외래키로만 표현
 - ▶ 제1규칙에 따라 개체를 릴레이션으로 변환, 1에 해당하는 개체의 기본키를 N에 해당하는 개체의 외래키로 지정

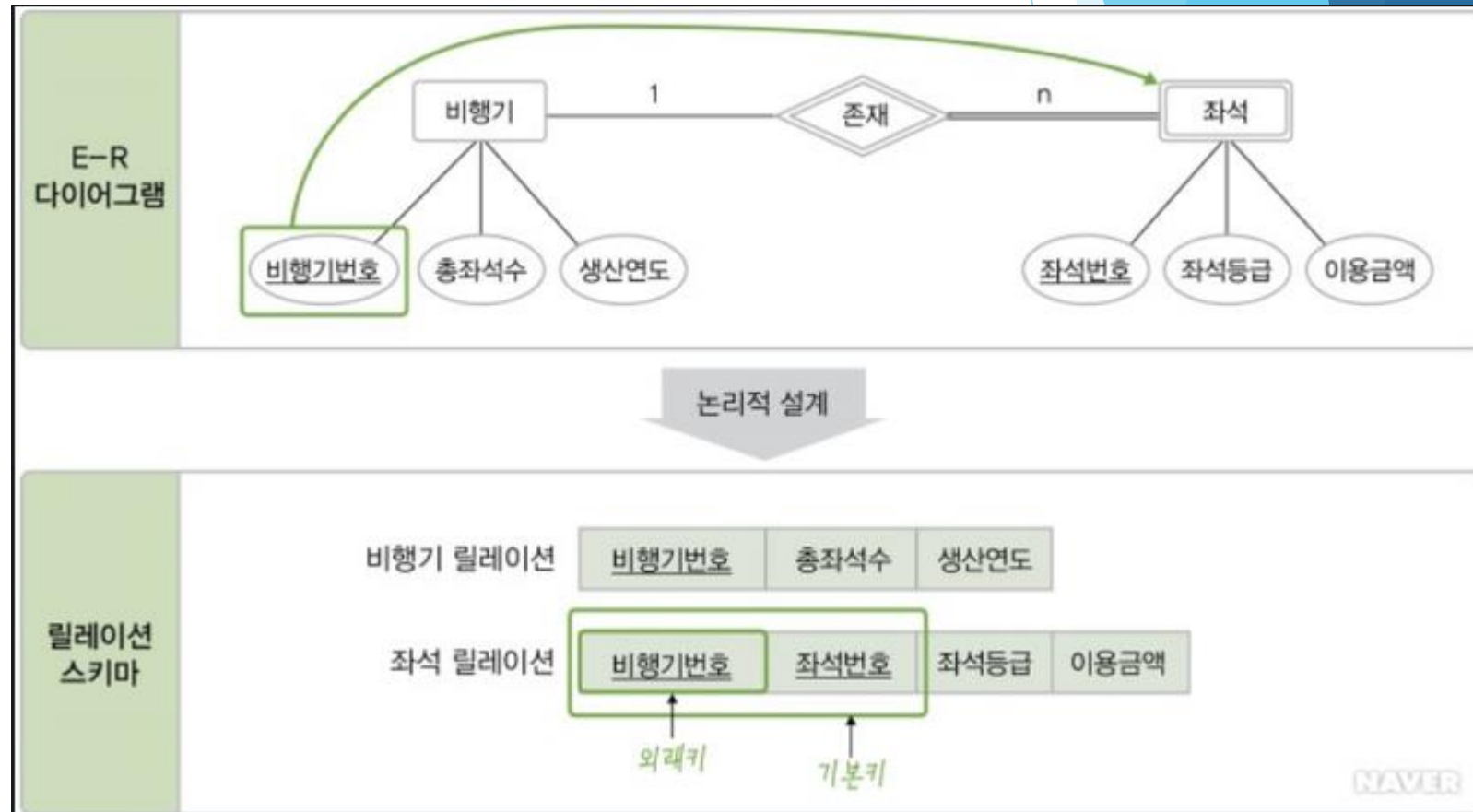


데이터베이스 논리적 설계

▶ 규칙3 : 1:N 관계는 외래키로 표현한다

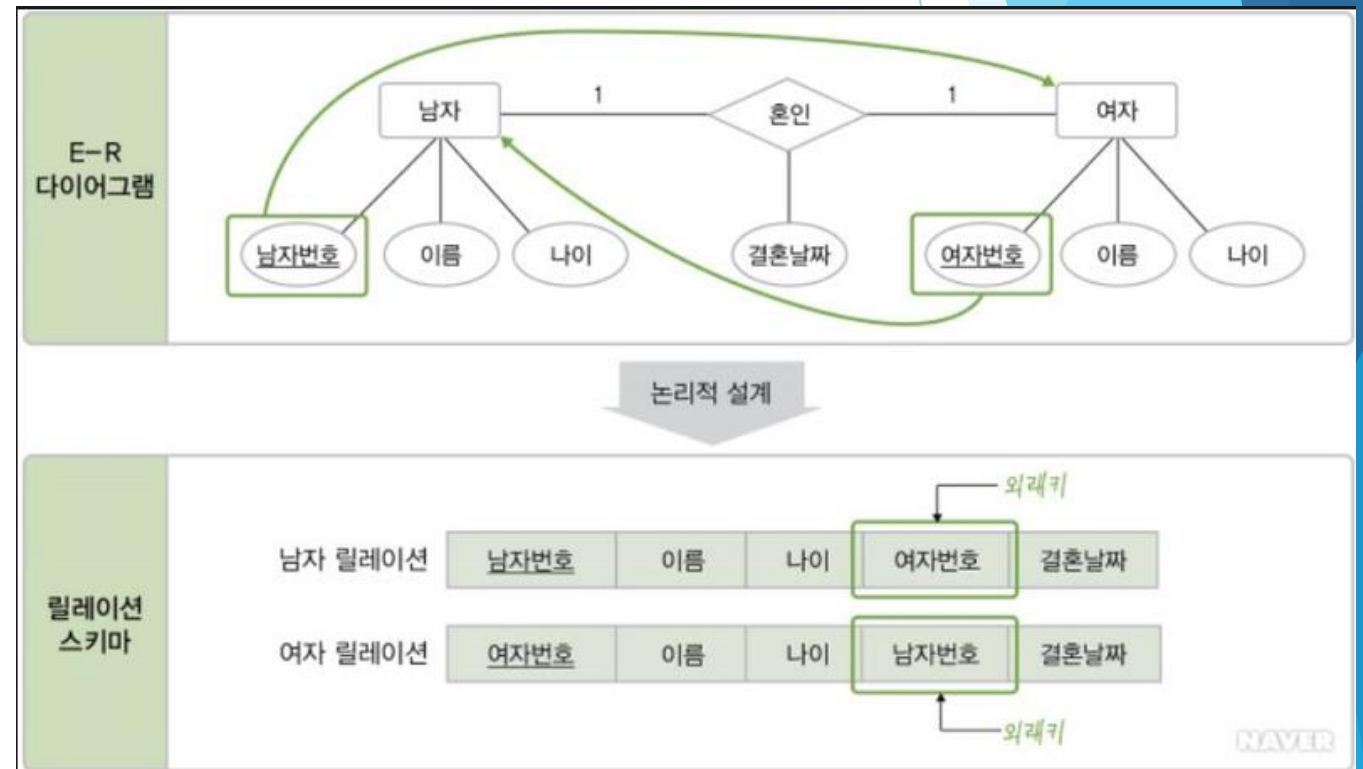
▶ 3.2 약개체가 참여하는 1:N 관계의 경우 외래키가 포함된 릴레이션에서는 해당 외래키를 포함하여 기본키를 지정

- ▶ 약개체의 기본키는 외래키(오너개체의 기본키)와 약개체의 기본키의 혼합으로 구성
- ▶ 승객이 좌석번호만을 가지고 자신이 예매한 좌석을 정확하게 찾을 수 없다. 예매한 비행기까지 알아야만 정확한 확인이 가능하기에 약개체 좌석 릴레이션에서 비행기번호와 좌석번호를 기본키로 설정하는 것이 타당하다



데이터베이스 논리적 설계

- ▶ 규칙4 : 1:1 관계는 외래키로 표현한다
 - ▶ 4.1 일반적인 1:1 관계는 외래키를 서로 주고 받음
 - ▶ 관계는 릴레이션으로 변환하지 않고 개체 릴레이션 기본키를 상호간 외래키로만 표현
 - ▶ 관계가 가지고 있는 속성들은 관계에 참여하는 개체를 변환한 릴레이션에 포함

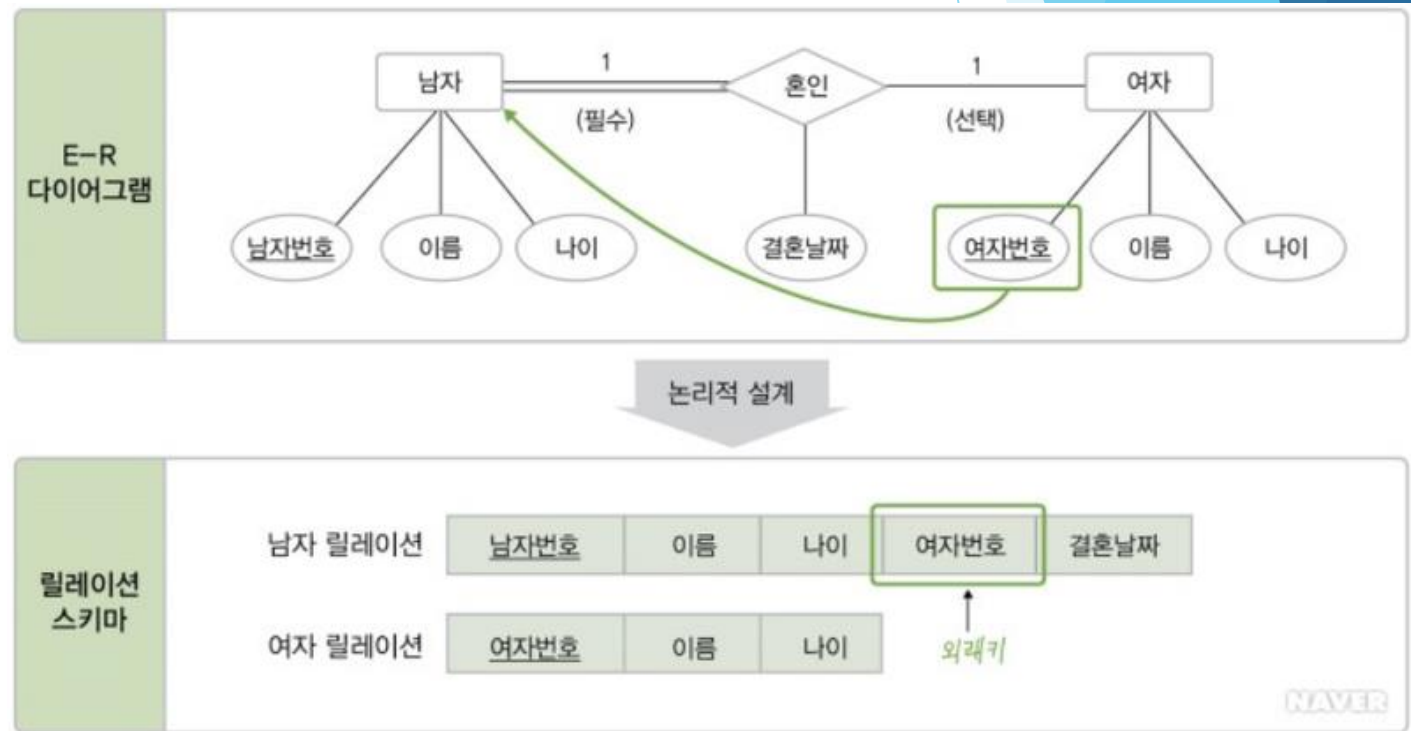


데이터베이스 논리적 설계

▶ 규칙4 : 1:1 관계는 외래키로 표현한다

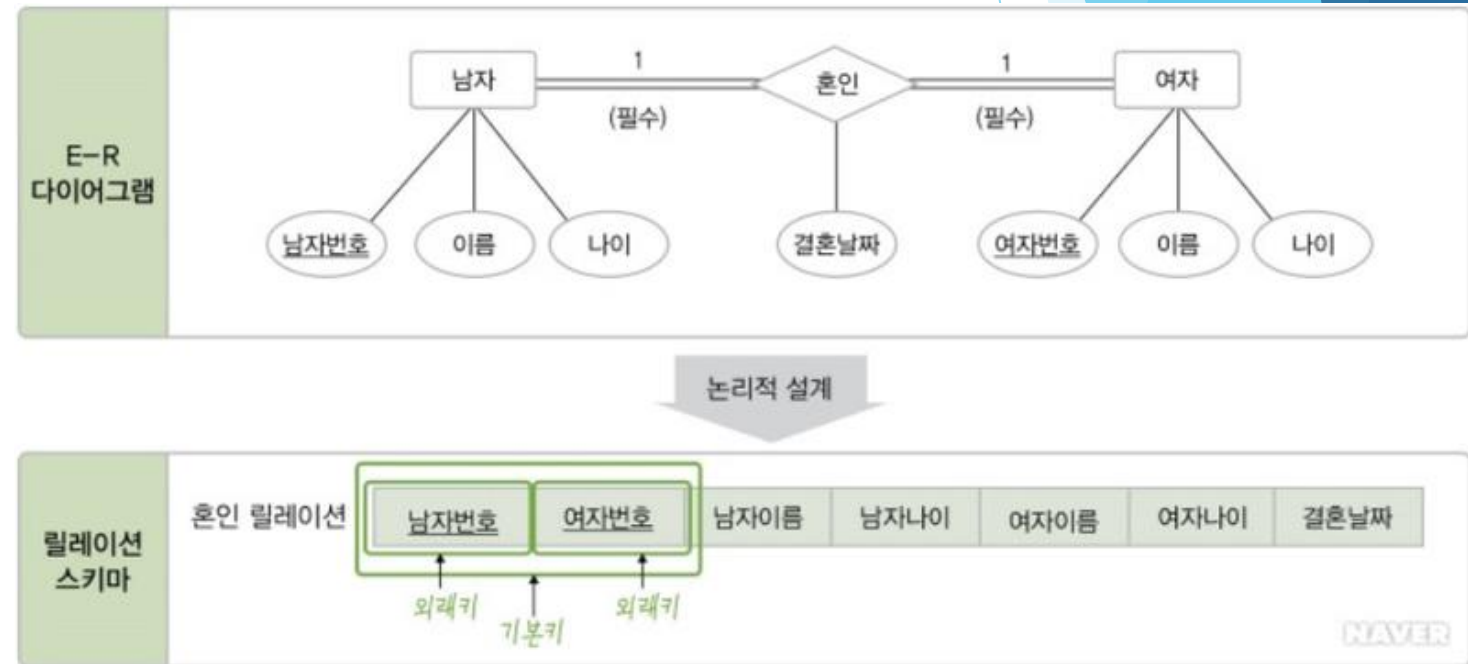
▶ 4.2 1:1관계에 필수적으로 참여하는 개체의 릴레이션만 외래키를 수용

- ▶ 관계를 맺는 두 개체 중 관계에 필수적으로 참여하는 개체 릴레이션에만 외래키 포함
- ▶ 관계에 필수적으로 참여하는 개체 릴레이션이 선택적으로 참여하는 개체 릴레이션의 기본키를 받아 외래키로 지정, 관계의 속성들은 관계에 필수적인 개체 릴레이션에 포함
- ▶ 선택적 참여 릴레이션이 외래키를 가져도 되지만 외래키로 지정된 속성에는 Null 값이 저장되는 경우가 많이 발생



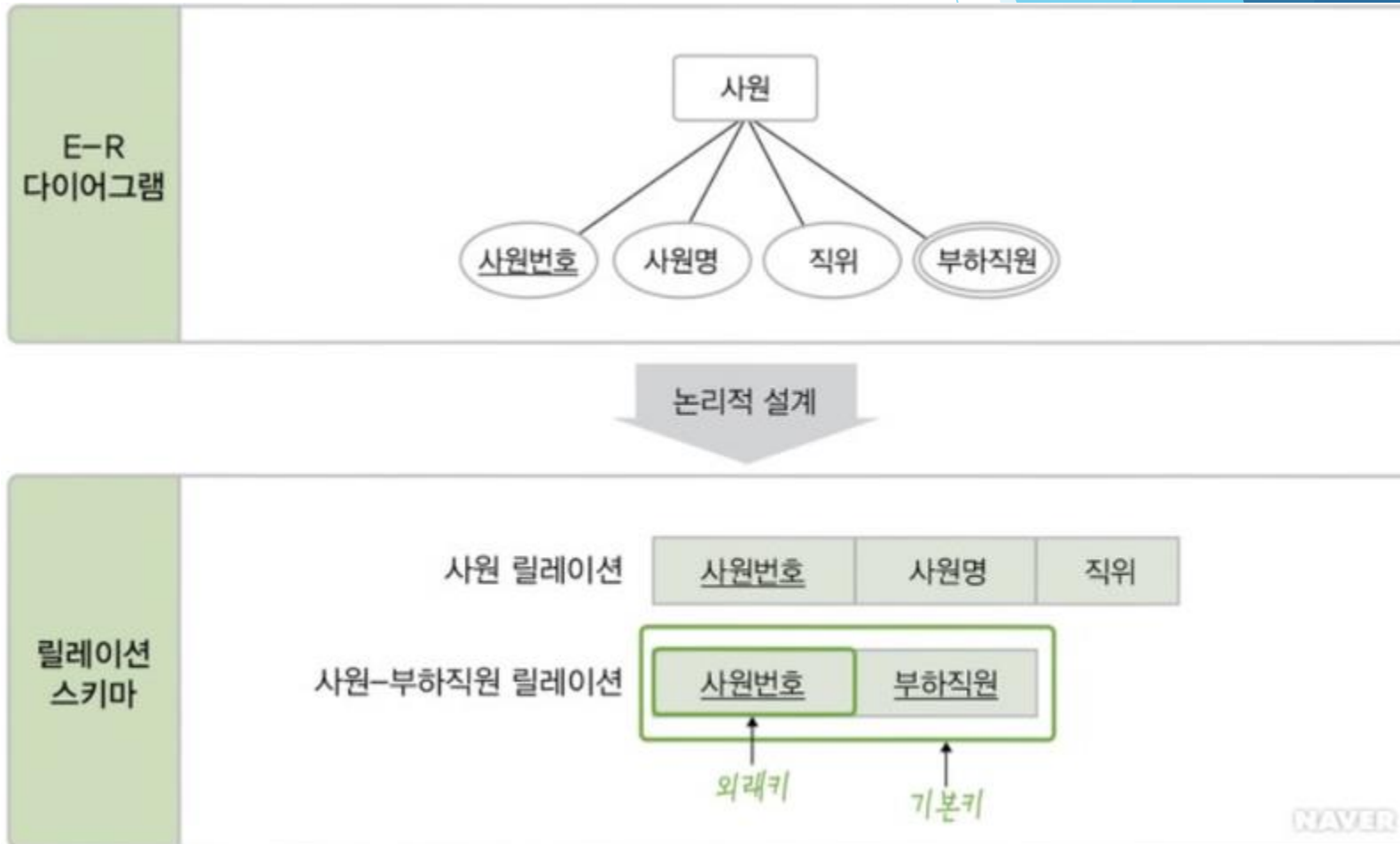
데이터베이스 논리적 설계

- ▶ 규칙4 : 1:1 관계는 외래키로 표현한다
 - ▶ 4.3 모든 개체가 1:1 관계에 필수적으로 참여 시 릴레이션 하나로 표현
 - ▶ 두 개체가 모두 필수적으로 참여한다면 관련성이 높기에 릴레이션을 합쳐 하나로 표현
 - ▶ 관계의 이름을 릴레이션으로 사용, 관계에 참여하는 개체의 속성을 관계 릴레이션에 포함
 - ▶ 두 개체 릴레이션의 키 속성을 조합하여 관계 릴레이션의 기본키로 지정



데이터베이스 논리적 설계

- ▶ 규칙5 : 다중 값 속성은 독립 릴레이션으로 변환한다
 - ▶ 관계 데이터 모델 릴레이션은 다중 값 속성을 허용하지 않아 E-R 다이어그램에 있는 다중 값 속성은 별도의 릴레이션을 만들어 포함
 - ▶ 신규 릴레이션의 속성 구성은 다중 값 속성으로 구성되고 기본키 구성은 다중값 속성과 외래키를 조합하여 지정



데이터베이스 물리적 설계

- ▶ 데이터베이스 물리적 설계 개요
 - ▶ 물리적 설계 정의 및 특징

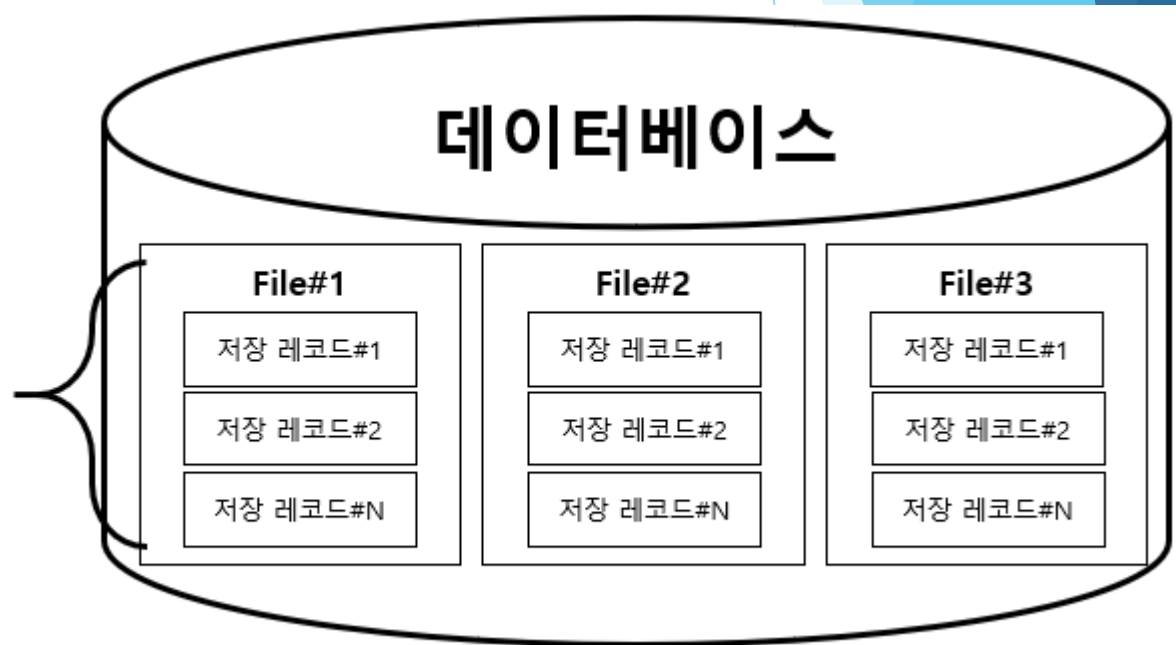
정의	- 논리적 스키마로부터 효율적이고 구현 가능한 물리적 데이터베이스 구조를 설계하는 행위
특징	- 구현단계에서 실행할 수 있는 트랜잭션을 구현하는데 필요한 내부 구조를 결정 (병행 수행) - 데이터베이스의 물리적 구조는 데이터베이스 시스템 성능에 중대한 영향을 미침 - 물리적 설계는 하드웨어와 운영 체제의 특성을 고려하여 설계

데이터베이스 물리적 설계

▶ 물리적 데이터베이스 기본 저장 구조

- ▶ 물리적 데이터베이스의 기본적인 데이터 단위는 저장 레코드(= 논리적 레코드 + 물리적 저장 정보)
- ▶ 동일 파일(한 타입의 저장 레코드들의 집합) 내에 있는 저장 레코드라고 해서 반드시 크기 미일치
- ▶ 물리적 설계는 "저장 레코드 양식", "저장 장치에서의 레코드 집중화", "접근 경로 설계"가 포함

파일구조는
레코드 양식,
순서, 인덱스
(접근경로),
물리적 저장
공간 할당



데이터베이스는 파일의 형태로 저장되고 접근

데이터베이스 물리적 설계

▶ 물리 데이터베이스 (내부 스키마) 설계

분 류	설 명
저장 레코드의 양식 설계	<ul style="list-style-type: none">- 저장 레코드 양식은 데이터 타입, 데이터 값의 분포, 프로그램, 접근빈도 고려- 저장 레코드에 대한 데이터 표현과 압축에 대한 양식- 물리적 단계의 성능평가에 따라 앞의 결정 사항이 변경 가능성 존재
레코드 집중의 분석 및 설계	<ul style="list-style-type: none">- 저장 공간에 레코드들이 물리적으로 집중 저장되도록 할당 (물리적 순차성)- 레코드들을 연속된 저장 공간에 할당, 블록 크기 선정(효율적 검색 목적)- 데이터 레코드의 순차 처리시 "큰 블록" 유리, 임의 접근 처리시 "작은 블록" 유리
접근 경로 설계	<ul style="list-style-type: none">- 접근 경로는 물리적 저장 장치 위에 저장된 데이터의 접근(저장구조)과 처리(탐색 기법)를 가능하게 하는 절차- 저장 구조는 주로 인덱스를 통한 접근 방법과 저장 레코드를 정의- 탐색 기법은 주어진 응용을 위해 취해야 될 적절한 접근 경로 정의- DBMS가 정해지면 물리적 설계는 DBMS가 제공하는 DB파일 조직 방법 중 선택

데이터베이스 물리적 설계

▶ 물리 데이터베이스 설계 시 고려사항

항목	설명
응답 시간	- 트랜잭션을 실행시키기 위해 시스템에 입력 시킨 때부터 다시 결과를 받을 때 까지 시간 - 트랜잭션이 참조하는 데이터에 대한 데이터베이스 접근시간은 DBMS에 영향을 받음
저장 공간 효율화	- 데이터베이스 파일, 접근 경로 구조들을 저장하기 위해 최소한의 저장공간을 사용
트랜잭션 처리도	- 단위 시간에 데이터베이스 시스템이 처리할 수 있는 평균 트랜잭션 처리도 - 시스템에 부하가 절정을 이루는 시간대를 고려

- ▶ 성능 요건을 만족하는 여부를 평가는 시뮬레이션이나 프로토타입과 같은 분석적 기법이나 실험적 기법 사용

데이터베이스 물리적 설계

- ▶ 물리 데이터 베이스 설계 실습
 - ▶ ERDCloud

팀원들과 함께 ERD를 그려보세요.

