

CSE231

데이터베이스 설계

(Database Design)

Lecture 10: 데이터베이스 설계II

담당교수: 전강욱(컴퓨터공학부)

kw.chon@koreatech.ac.kr

NCS 정보

- 능력 단위명 : 논리 데이터베이스 설계
- 능력 단위요소 : 개체 상세화하기, 관계 상세화하기
- 학습목표(수행 준거) :
 - 1.1 주제영역 별 업무에 필요한 정보를 관리하기 위한 논리적인 데이터 집합 인 개체를 도출하고 상세화할 수 있다
 - 2.1 도출된 개체 간 상호 연관성을 기반으로 개체 간 모든 관계를 도출하고, 각 관계별 관계명, 기수성(Cardinality) 및 선택성(Optionality)을 정의할 수 있다

NCS 정보 (계속)

■ 지식

- 릴레이션 스키마 표현 방법
- 개체 릴레이션 변환 방법
- 관계 릴레이션 변환 방법

■ 기술

- 개체를 릴레이션 스키마로 변환하는 능력
- 약한 개체를 릴레이션 스키마로 변환하는 능력
- 관계를 릴레이션 스키마로 변환하는 능력
- 관계 릴레이션 스키마의 식별자를 결정하는 능

■ 태도

- 개체의 특성을 파악해서 릴레이션 스키마로 정확히 변환하려는 태도
- 개체 릴레이션이 정확히 표현되었는지 재검토하는 습관
- 관계의 특성을 파악해서 릴레이션 스키마로 정확히 변환하려는 태도
- 관계 릴레이션이 정확히 표현되었는지 재검토하는 습관

세부 학습목표

- 1. 논리적 모델링이 무엇인지 설명할 수 있다.
- 2. ER 다이어그램을 릴레이션 스키마로 변환하는 방법을 설명할 수 있다.
- 3. 개체를 릴레이션 스키마로 변환할 수 있다.
- 4. 약한 개체를 릴레이션 스키마로 변환할 수 있다.
- 5. 관계를 릴레이션 스키마로 변환할 수 있다.
- 6. 다대다 관계를 릴레이션 스키마로 변환할 수 있다.
- 7. 순환 관계를 릴레이션 스키마로 변환할 수 있다.
- 8. 다중 관계를 릴레이션 스키마로 변환할 수 있다.

데이터베이스 설계 단계

- ER모델과 릴레이션 변환 규칙을 이용한 설계는 5단계
 - 1단계: 요구사항분석
 - 정확하게 하는 것이 중요
 - 2단계: 개념적 설계
 - 현실 세계 문제를 개념화
 - 3단계: 논리적 설계
 - 4단계: 물리적 설계
 - 저장장치 입장에서 어떻게 구현할 것인가를 고려
 - 5단계: 구현

논리적 설계

■ 목적

- DBMS에 적합한 논리적 스키마 설계
- 개념적 스키마를 논리적 데이터 모델을 이용해 논리적 구조로 표현
 - 일반적으로 관계 데이터 모델을 주로 활용

■ 결과물

- 논리적 스키마: 릴레이션 스키마

■ 주요작업

- 개념적 설계 단계의 결과물인 ER 다이어그램을 릴레이션 스키마로 변환
- 릴레이션 스키마 변환 후 속성의 데이터 타입, 길이, 널 값 허용여부, 기본 값, 제약 조건 등을 세부적으로 결정하고 결과를 문서화 시킴

논리적 설계 (계속)

- ER 다이어그램을 릴레이션 스키마로 변환하는 규칙
 - 규칙 1: 모든 개체는 릴레이션으로 변환
 - 하나의 개체는 테이블로 변환
 - 규칙 2: 다 대 다 (n:m) 관계는 릴레이션으로 변환
 - 관계 자체를 테이블로 변환
 - 규칙 3: 일 대 다 (1:n) 관계는 외래키로 표현
 - 1의 기본 키를 n의 외래키로 표현하여 1쪽 테이블을 참조
 - 규칙 4: 일 대 일 (1:1) 관계는 외래키로 표현
 - 규칙 5: 다중 값 속성은 릴레이션으로 변환
- 변환 규칙을 순서대로 적용하되, 해당되지 않는 규칙은 제외

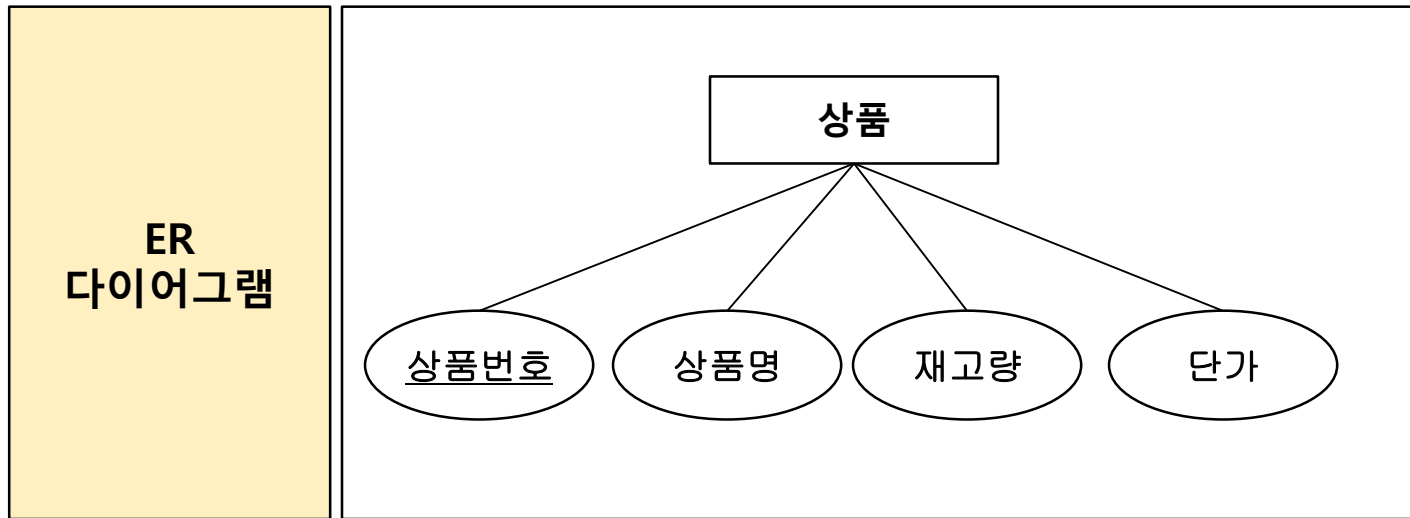
논리적 설계 (계속)

- (규칙 1) 모든 개체는 릴레이션으로 변환
 - ER 다이어그램의 각 개체를 하나의 릴레이션으로 변환
 - 개체의 이름 → 릴레이션 이름
 - 개체의 속성 → 릴레이션의 속성
 - 개체의 키 속성 → 릴레이션의 기본키
 - 개체의 속성이 복합 속성인 경우, 복합 속성을 구성하고 있는 단순 속성만 속성으로

논리적 설계 (계속)

- (규칙1) 모든 개체는 릴레이션으로 변환

- 개체 → 릴레이션



논리적 설계

상품 릴레이션

상품번호

상품명

재고량

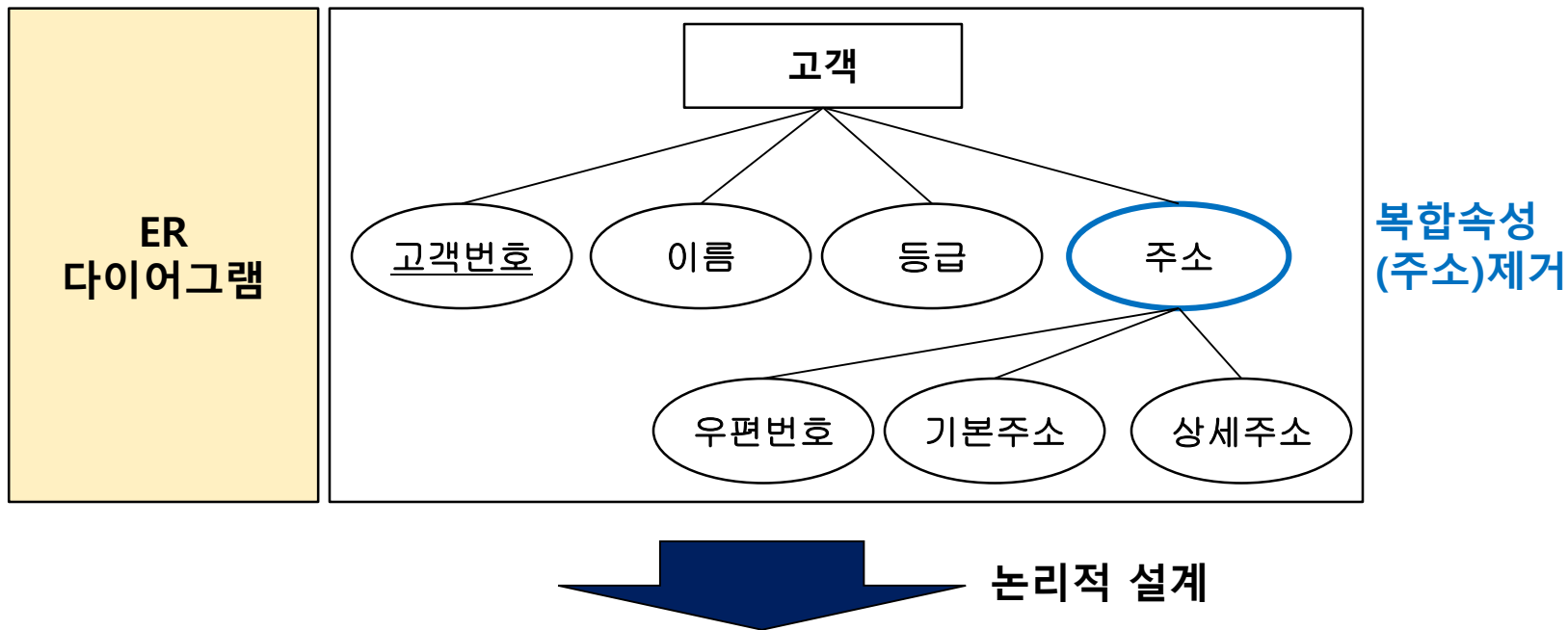
단가

상품(상품번호, 상품명, 재고량, 단가)

논리적 설계 (계속)

■ (규칙1) 모든 개체는 릴레이션으로 변환

- 복합 속성을 가지는 개체 → 릴레이션



고객 릴레이션

<u>고객번호</u>	이름	등급	우편번호	기본주소	상세주소
-------------	----	----	------	------	------

고객(고객번호, 이름, 등급, 우편번호, 기본주소, 상세주소)

논리적 설계 (계속)

- (규칙 2) 다 대 다 관계는 릴레이션으로 변환
 - ER다이어그램의 다 대 다 관계를 하나의 릴레이션으로 변환
 - 관계의 이름 → 릴레이션 이름
 - 관계의 속성 → 릴레이션 속성
 - 관계에 참여하는 개체를 규칙1에 따라 릴레이션으로 변환
 - 이 릴레이션의 기본키를 관계 릴레이션에 포함시켜 외래키로 지정
 - 외래키들을 조합하여 관계 릴레이션의 기본키로 지정
 - 또는, 새로운 속성을 하나 추가하여 기본키로 지정

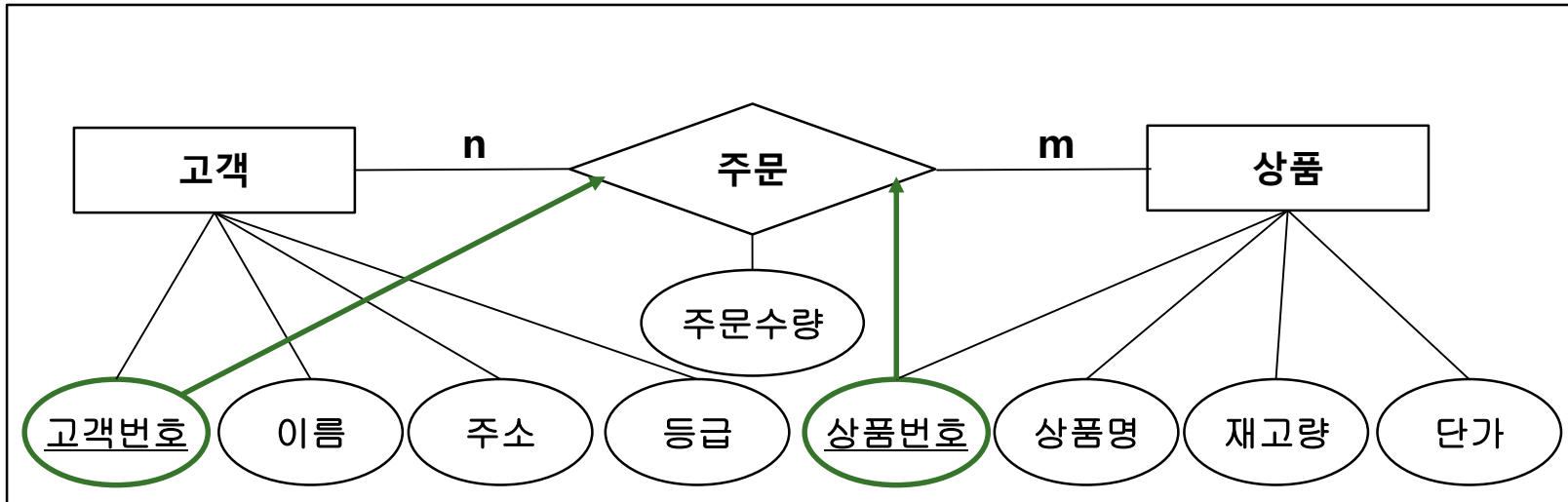
논리적 설계 (계속)

■ (규칙 2) 다 대 다 관계는 릴레이션으로 변환

□ 다대다 관계 → 릴레이션

- 주문 릴레이션에서 고객번호, 상품번호가 기본키가 됨
- 또는, 주문번호라는 새로운 기본키를 설정할 수 있음

ER
다이어그램



논리적 설계

고객 릴레이션

고객번호

이름

등급

우편번호

기본주소

상세주소

상품 릴레이션

상품번호

상품명

재고량

단가

주문 릴레이션

고객번호

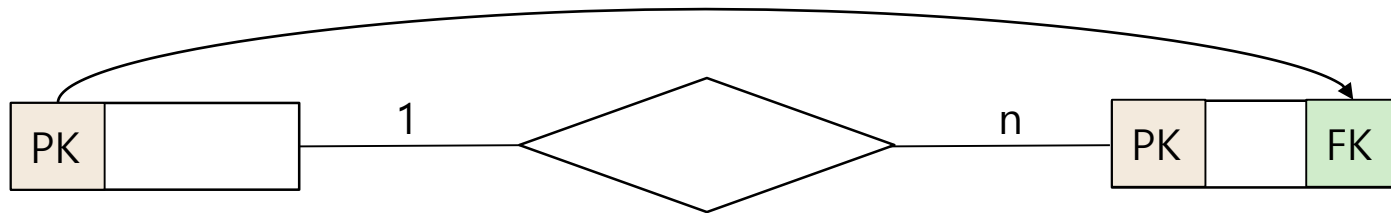
상품번호

주문수량

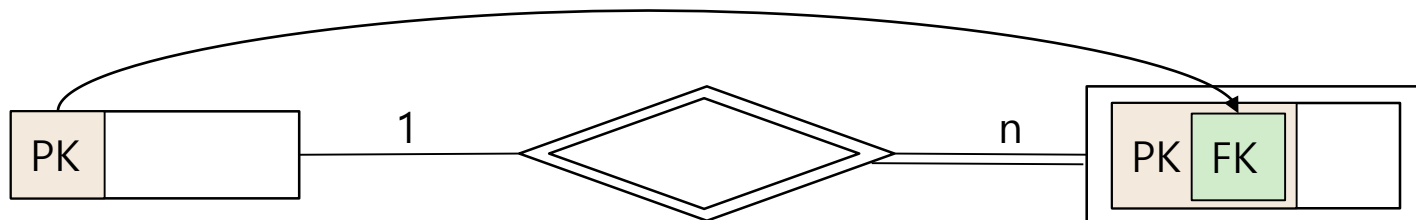
논리적 설계 (계속)

■ (규칙 3) 일 대 다 관계는 외래키로 표현

- ER 다이어그램의 **일 대 다 관계**는 **외래키만**로 표현
 - (규칙 3-1) 일반적인 일 대 다 관계는 외래키로 표현



- (규칙 3-2) **약한 개체**가 참여하는 **일 대 다 관계**는 외래키를 포함해서 기본키로 지정
 - 강한 개체인 1의 기본키를 n으로 가지고 와서, 1의 기본키를 포함하여 약한 개체의 기본키로 지정

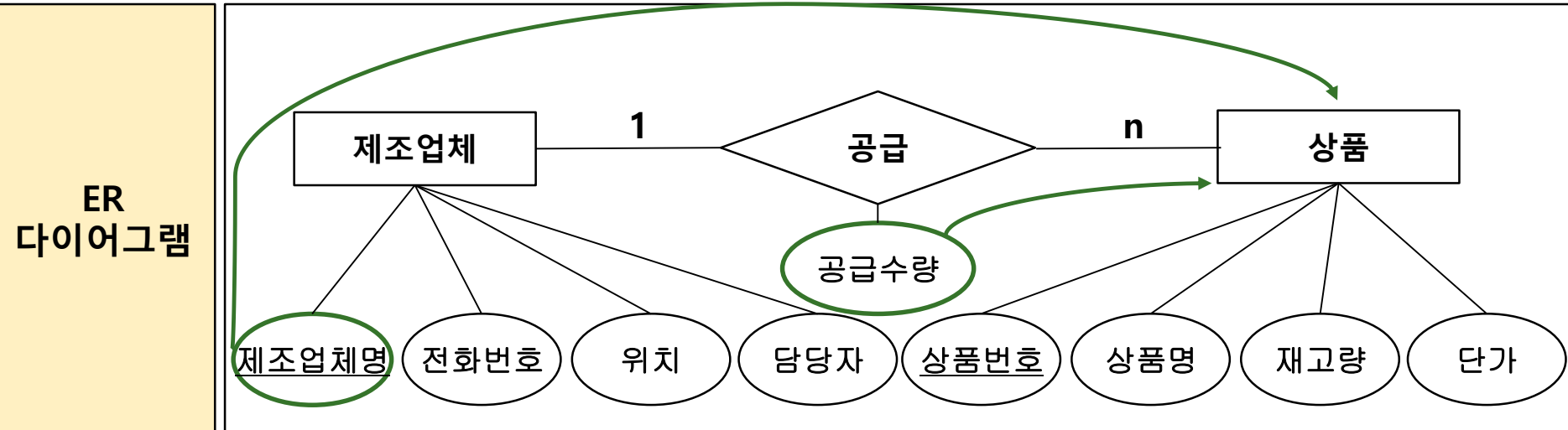


논리적 설계 (계속)

■ (규칙 3-1) 일반적인 일대다 관계는 외래키로 표현

□ 일 대 다 관계는 외래키로 표현

- 일 대 다 관계의 표현을 위해서, 제조업체명과 공급수량 속성이 추가



논리적 설계

제조업체 릴레이션

<u>제조업체명</u>	전화번호	위치	담당자
--------------	------	----	-----

상품 릴레이션

<u>상품번호</u>	상품명	재고량	단가	제조업체명	공급수량
-------------	-----	-----	----	-------	------

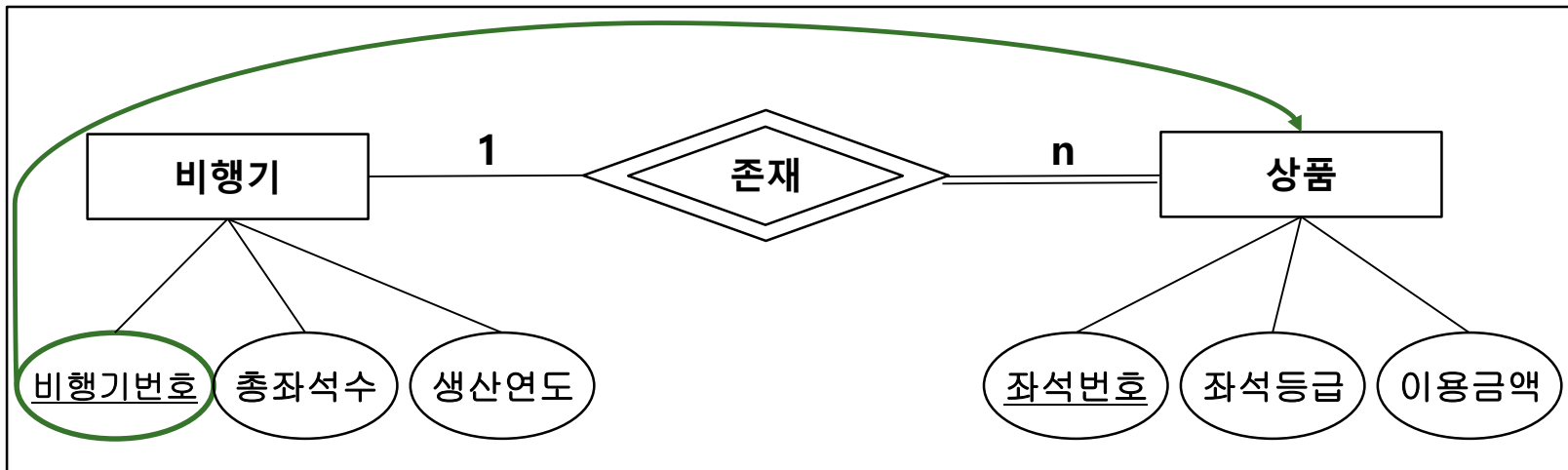
논리적 설계 (계속)

- (규칙 3) 일대다 관계는 외래키로 표현
 - (규칙 3-2) 약한 개체가 참여하는 일대다 관계는 외래키를 포함해서 기본키로 지정
 - 일 대 다(1:n) 관계에서 1측 개체 릴레이션의 기본키를 n측 개체 릴레이션에 포함시켜 외래키로 지정
 - 규칙 3-1과 동일
 - 관계의 속성들도 n측 개체 릴레이션에 포함시킴
 - n측 개체 릴레이션은 외래키를 포함하여 기본키를 지정
 - 약한 개체는 오너 개체에 따라 존재 여부가 결정되므로 오너 개체의 기본키를 이용해 식별해야 함

논리적 설계 (계속)

- (규칙 3-2) 약한 개체가 참여하는 일대다 관계는 외래키를 포함해서 기본키로 지정
 - 약한개체의 일대다 관계 → 외래키 포함 기본키 지정
 - 예제에서 좌석은 비행기에 속하는 약한 개체
 - 약한개체여서 (비행기번호, 좌석번호)가 기본키가 됨

ER
다이어그램



논리적 설계

비행기 릴레이션

<u>비행기번호</u>	총좌석수	생산연도
--------------	------	------

좌석 릴레이션

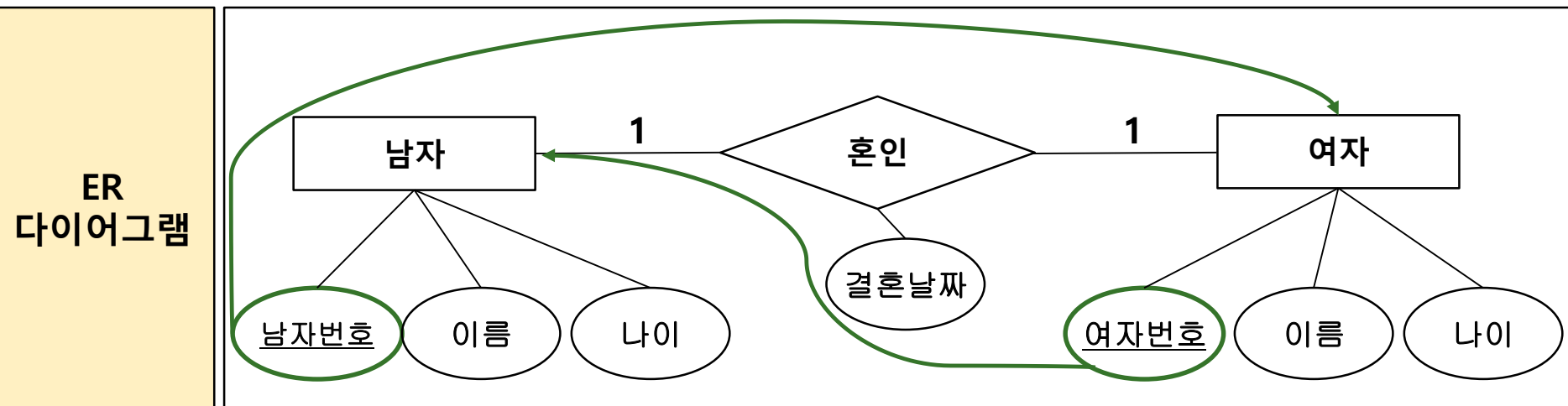
<u>비행기번호</u>	<u>좌석번호</u>	좌석등급	이용금액
--------------	-------------	------	------

논리적 설계 (계속)

- (규칙 4) 일 대 일 관계는 외래키로 표현
 - ER 다이어그램의 일 대 일 관계는 외래키로만 표현
 - (규칙 4-1) 일반적인 일 대 일 관계는 외래키를 서로 주고 받음
 - (규칙 4-2) 일 대 일 관계에 필수적으로 참여하는 개체의 릴레이션만 외래키를 받음
 - (규칙 4-3) **모든 개체**가 일 대 일 관계에 **필수적으로 참여**하면 **릴레이션을 하나로 합침**
 - 굳이 외래키를 주고 받을 필요가 없음

논리적 설계 (계속)

- (규칙 4-1) 일반적인 일대일 관계는 외래키를 서로 주고 받음
 - 상대 테이블에 외래키로 포함



논리적 설계

여자 릴레이션

<u>여자번호</u>	이름	나이	남자번호	결혼날짜
-------------	----	----	------	------

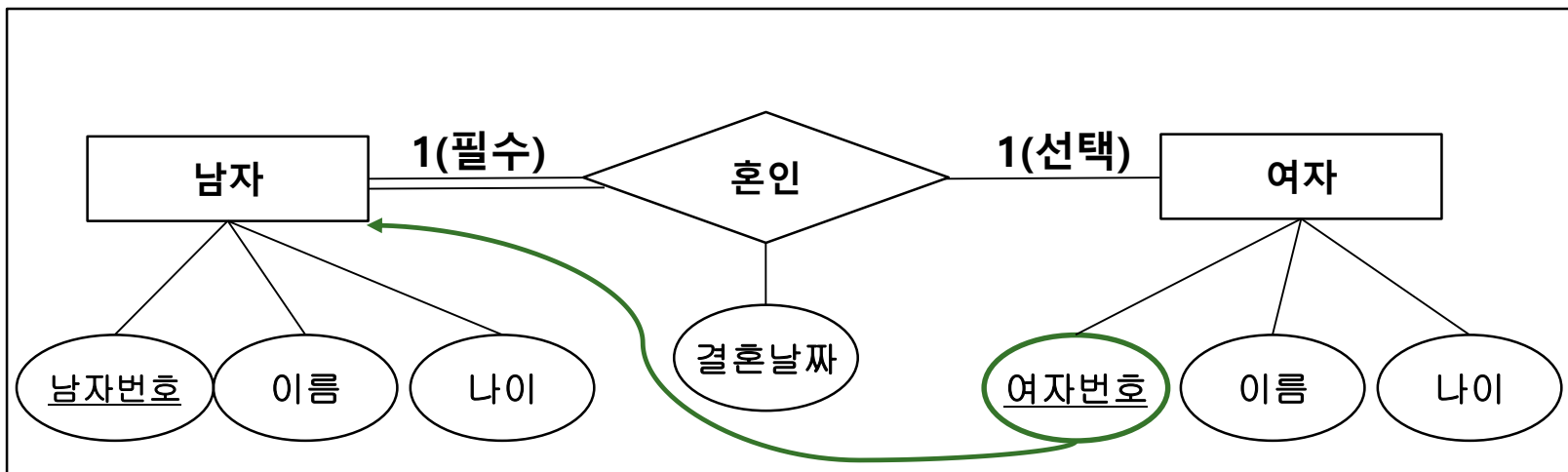
남자 릴레이션

<u>남자번호</u>	이름	나이	여자번호	결혼날짜
-------------	----	----	------	------

논리적 설계 (계속)

- (규칙 4-2) 필수적으로 참여하는 개체 릴레이션만 외래 키를 받는다
 - 관계에 필수적으로 참여하는 개체 릴레이션에만 외래키를 포함
 - 관계의 속성들은 관계에 필수적으로 참여하는 개체 릴레이션만 외래키를 받음

ER
다이아그램



논리적 설계

여자 릴레이션 여자번호 이름 나이

FK

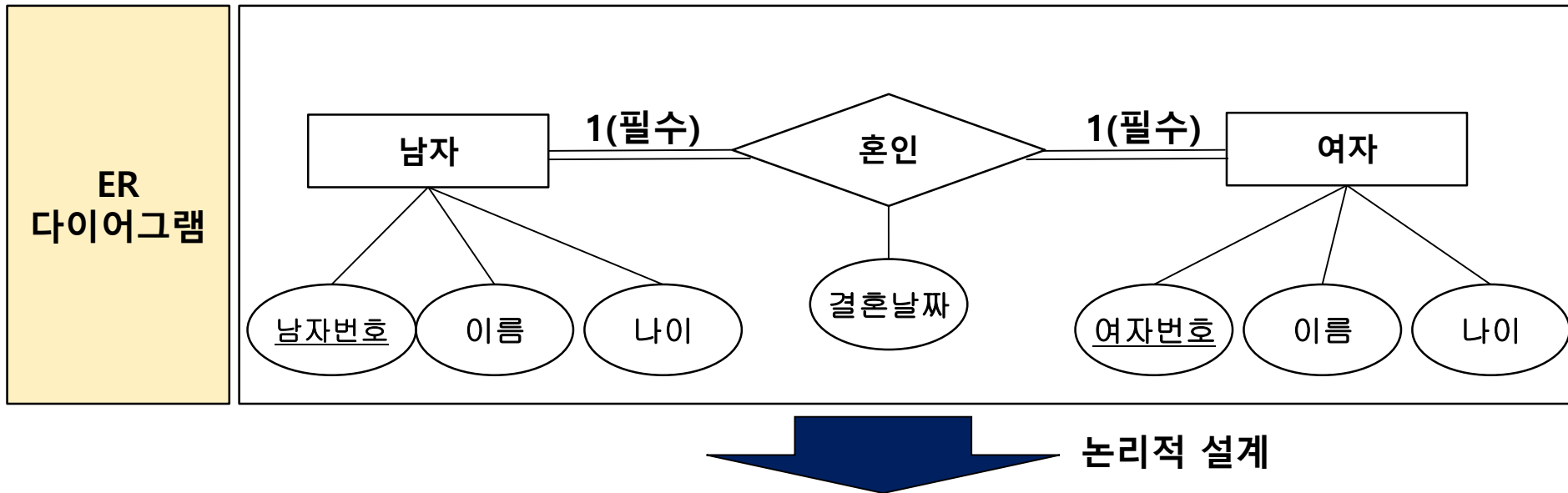
남자 릴레이션 남자번호 이름 나이 여자번호 결혼날짜

논리적 설계 (계속)

- (규칙 4-3) 모든 개체가 필수적으로 참여하면 릴레이션을 하나로 합침
 - 관계에 참여하는 개체 릴레이션들을 하나의 릴레이션으로 합쳐서 포함
 - 관계의 이름을 릴레이션 이름으로 사용하고 관계에 참여하는 두 개체의 속성들을 관계 릴레이션에 모두 포함시킴
 - 두 개체 릴레이션의 키 속성을 조합하여 관계 릴레이션의 기본키로 지정

논리적 설계 (계속)

- (규칙 4-3) 모든 개체가 필수적으로 참여하면 릴레이션을 하나로 합침



혼인 릴레이션

남자번호

여자번호

남자이름

남자나이

여자이름

여자나이

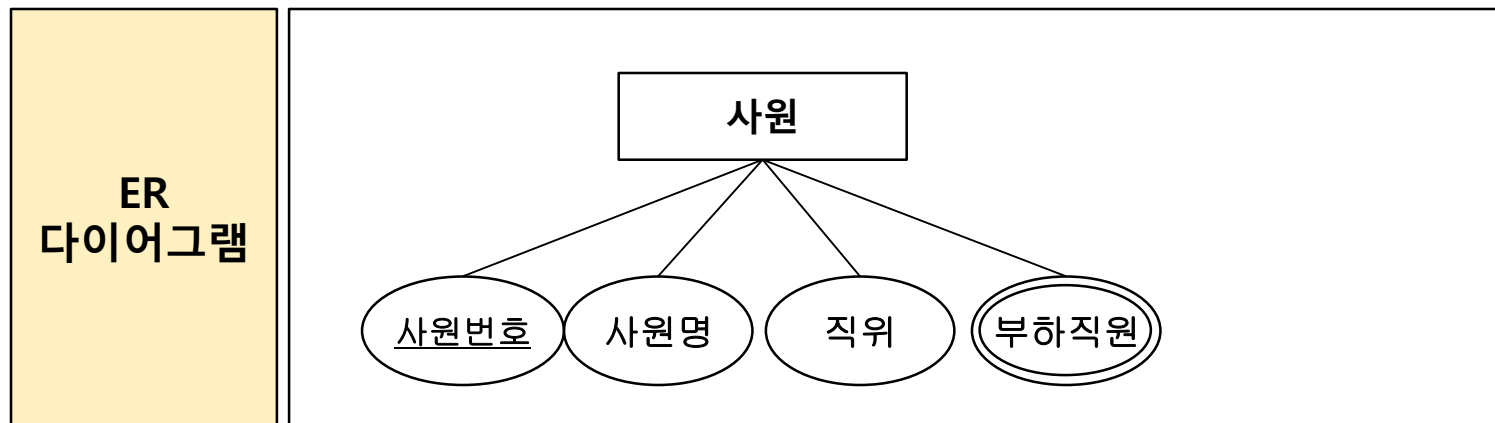
결혼날짜

논리적 설계 (계속)

- (규칙 5) 다중 값 속성은 릴레이션으로 변환
 - ER 다이어그램의 다중 값 속성은 독립적인 릴레이션으로 변환
 - 다중 값 속성과 함께 그 속성을 가지고 있던 개체 릴레이션의 기본키를 외래키로 가져와 새로운 릴레이션에 포함
 - 새로운 릴레이션의 기본키는 다중 값 속성과 외래키를 조합해서 생성

논리적 설계 (계속)

- (규칙 5) 다중 값 속성은 릴레이션으로 변환
 - 부하직원이 다중 값 속성 포함
 - 사원-부하직원 릴레이션 생성
 - 사원번호를 외래키로 가져오고, (사원번호,부하직원)을 기본키로 설정



↓ 논리적 설계

사원 릴레이션

사원번호

사원명

직위

사원_부하직원 릴레이션

사원번호

부하직원

논리적 설계 (계속)

■ (규칙 5) 다중 값 속성은 릴레이션으로 변환

- (규칙 1)을 이용하여도 릴레이션 특성에 부합하지만, 데이터가 중복되는 경향이 있음
 - 예: 부하직원이 10명인 경우 10번 중복
- (규칙 2)를 이용하면, 중복 문제가 완화

사원번호	사원명	직위	부하직원
E001	홍정화	부장	{김정수, 이수연}
E002	김주창	과장	{박영길}
E003	최종민	차장	{이수영, 배길수}

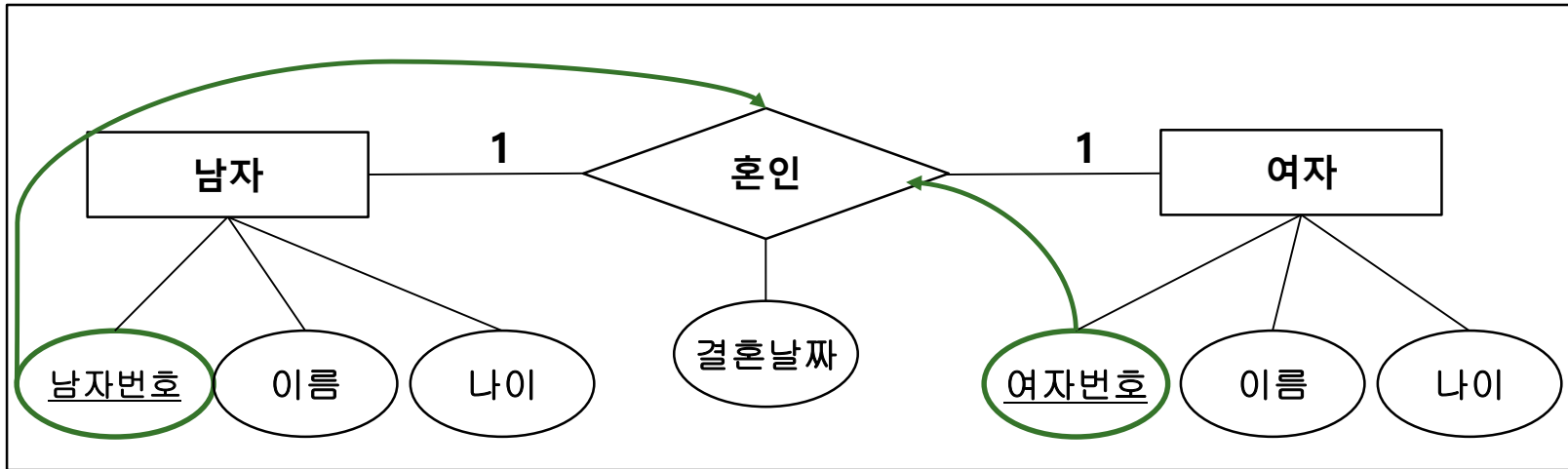
사원번호	사원명	직위	부하직원
E001	홍정화	부장	김정수
E001	홍정화	부장	이수연
E002	김주창	과장	박영길
E003	최종민	차장	이수영
E003	최종민	차장	배길수

사원번호	사원명	직위	사원번호	부하직원
E001	홍정화	부장	E001	김정수
E002	김주창	과장	E001	이수연
E003	최종민	차장	E002	박영길
			E003	이수영
			E003	배길수

논리적 설계 (계속)

- 모든 관계는 독립적인 릴레이션으로 변환 가능
 - 속성이 많은 관계는 유형에 관계없이 릴레이션으로의 변환을 고려할 수 있음

ER
다이어그램



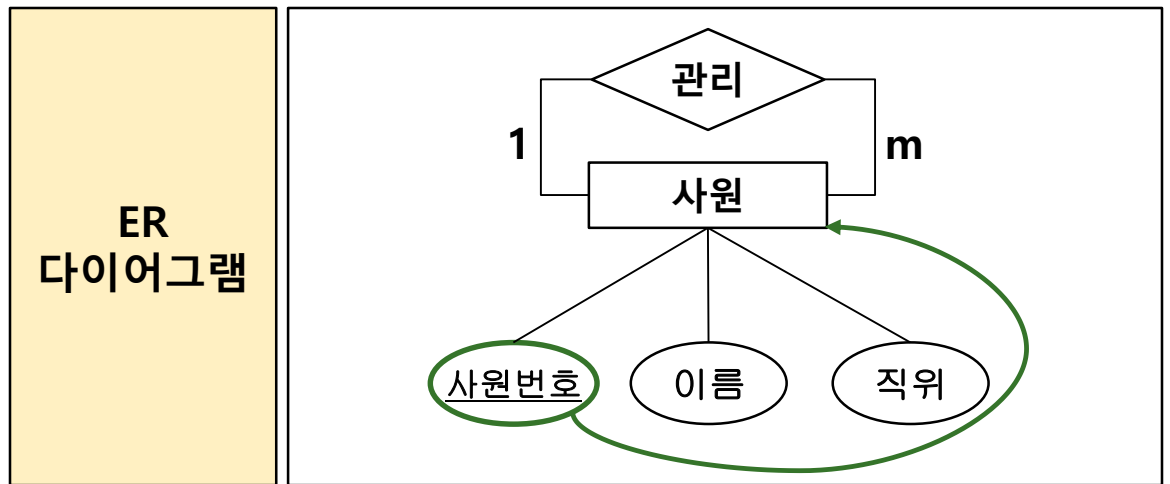
논리적 설계

여자 릴레이션	<u>여자번호</u>	이름	나이	남자 릴레이션	<u>남자번호</u>	이름	나이
혼인 릴레이션	<u>남자번호</u>	<u>여자번호</u>	결혼날짜				

논리적 설계 (계속)

■ 순환 관계

- 개체가 자기 자신과 관계를 맺는 순환 관계도 기본 규칙 적용
- 예: 사원과 사원이 관계를 갖는 것으로 생각
 - 관리의 관계면 1:m, 사원번호를 외래키로 붙이면 됨
 - 상사번호 외래키(1의 기본키)

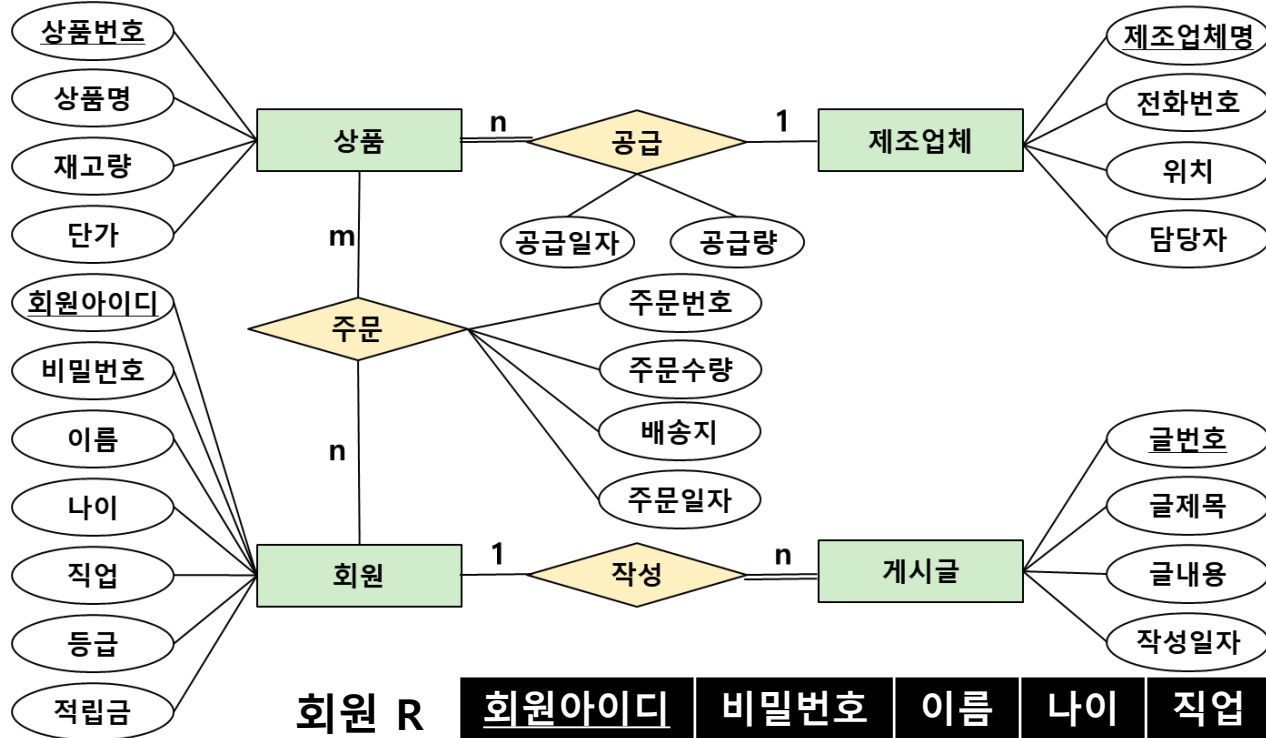


논리적 설계

사원 릴레이션	<u>사원번호</u>	이름	직위	FK 상사번호

논리적 설계 (계속)

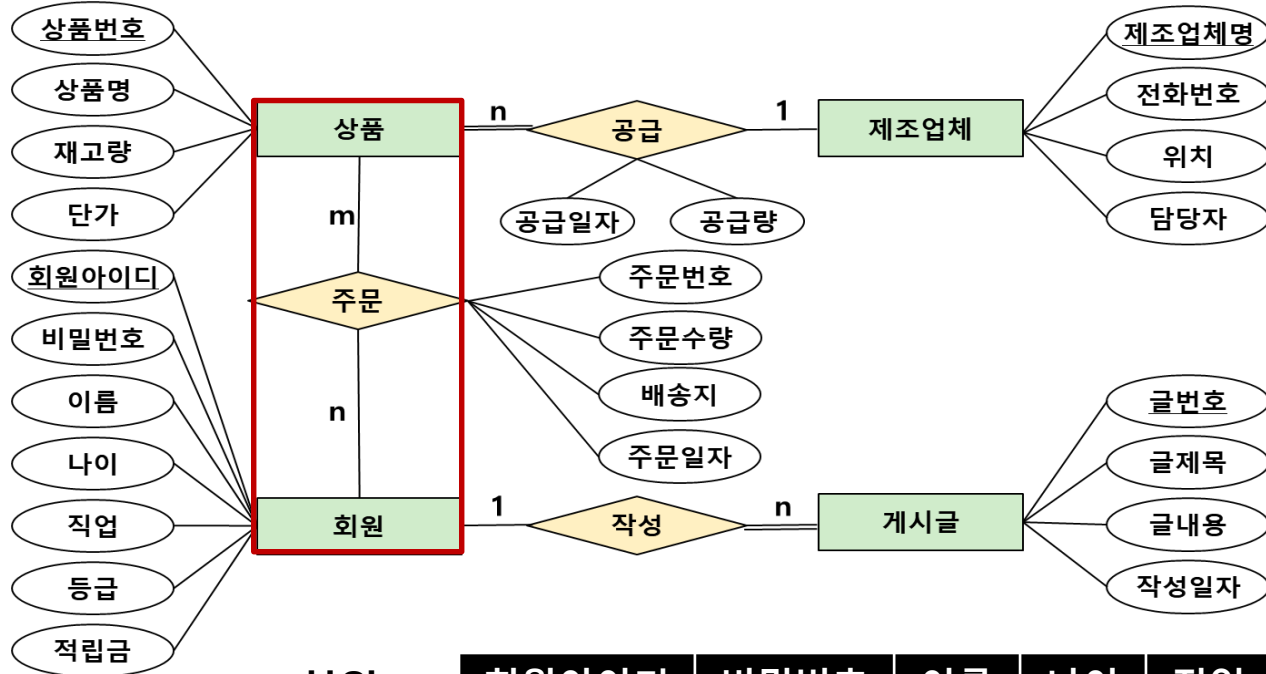
■ (규칙1) 모든 개체는 릴레이션으로 변환



회원 R	<u>회원아이디</u>	비밀번호	이름	나이	직업	등급	적립금
상품 R	<u>상품번호</u>	상품명	재고량	단가			
제조업체 R	<u>제조업체명</u>	전화번호	위치	담당자			
게시글 R	<u>글번호</u>	글제목	글내용	작성일자			

논리적 설계 (계속)

■ (규칙2) 다대다 관계는 릴레이션으로 변환



회원 R	<u>회원아이디</u>	비밀번호	이름	나이	직업	등급	적립금
상품 R	<u>상품번호</u>	상품명	재고량	단가			
제조업체 R	<u>제조업체명</u>	전화번호	위치	담당자			
게시글 R	<u>글번호</u>	글제목	글내용	작성일자			
주문 R	<u>회원아이디</u>	<u>상품번호</u>	주문번호	주문수량	배송지	주문일자	

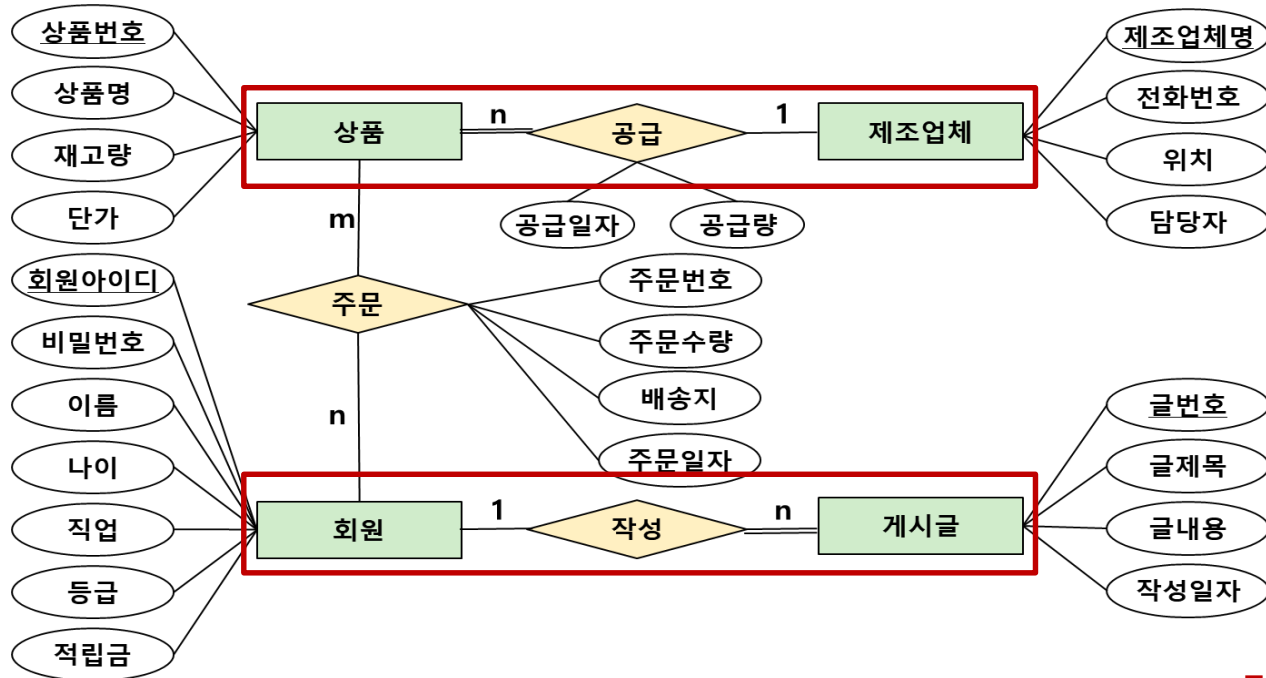
FK

FK

PK

논리적 설계 (계속)

- (규칙3) 일 대 다 관계인 경우 1측의 기본키를 n측의 외래키로 붙이고, 관계의 속성을 n측으로 이동



FK

상품 R

<u>상품번호</u>	상품명	재고량	단가	제조업체명	공급일자	공급량
-------------	-----	-----	----	-------	------	-----

FK

게시글 R

<u>글번호</u>	글제목	글내용	작성일자	회원아이디
------------	-----	-----	------	-------

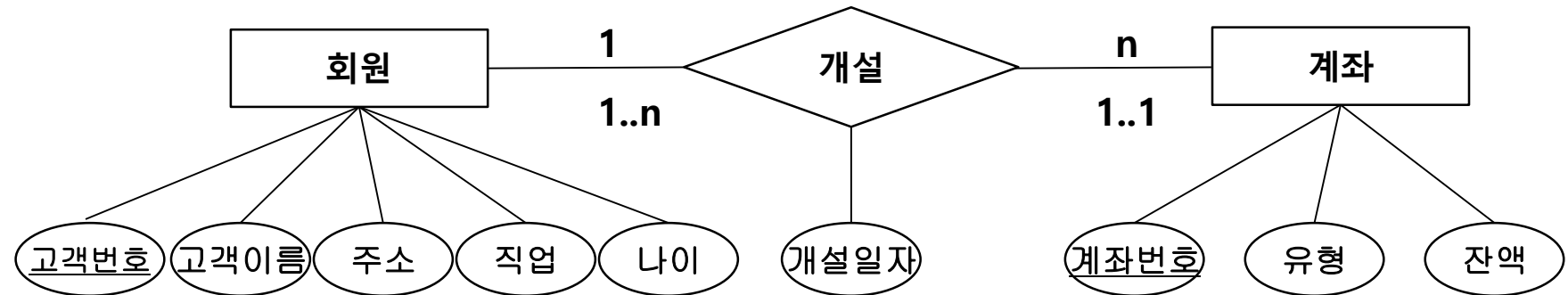
논리적 설계 (계속)

- 최종적인 ER 스키마는 아래와 같음

회원 R	<u>회원아이디</u>	비밀번호	이름	나이	직업	등급	적립금
상품 R	<u>상품번호</u>	상품명	재고량	단가	제조업체명	공급일자	공급량
제조업체 R	<u>제조업체명</u>	전화번호	위치	담당자			
게시글 R	<u>글번호</u>	글제목	글내용	작성일자	회원아이디		
주문 R	<u>회원아이디</u>	<u>상품번호</u>	주문번호	주문수량	배송지	주문일자	

예제

- 다음 은행 데이터베이스 ER 다이어그램을 릴레이션 스키마로 변환



예제

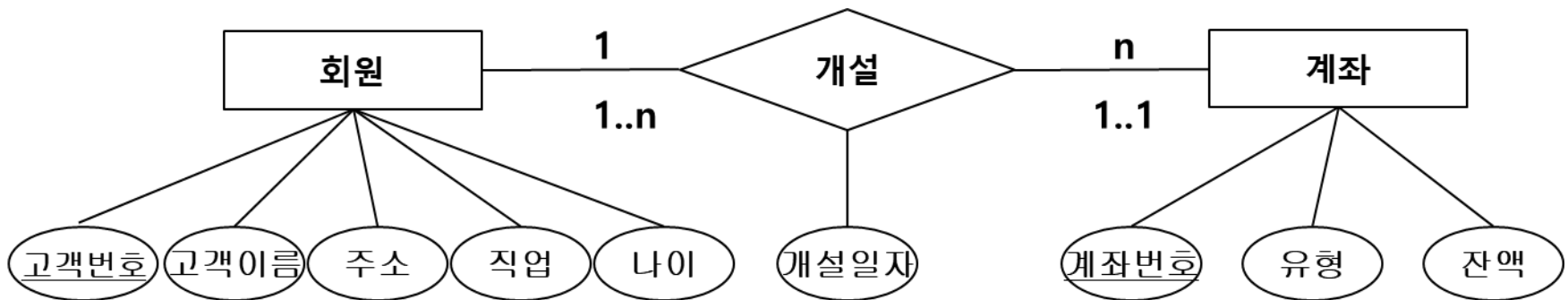
■ 릴레이션 스키마

- 규칙 1: **개체와 속성**은 **릴레이션**으로 변환
- 회원 R:

<u>회원번호</u>	고객이름	주소	직업	나이
-------------	------	----	----	----
- 계좌 R:

<u>계좌번호</u>	유형	잔액
-------------	----	----
- 규칙 2: 다 대 다의 관계가 아니므로 생략
- 규칙 3 적용: **일 대 다 관계**는 **1측 릴레이션의 기본키를 다른 릴레이션의 외래키로 삽입**
 - 약한 개체가 아니므로 위 동작만 수행
 - 계좌 R:

<u>계좌번호</u>	유형	잔액	FK 회원번호	개설일자
-------------	----	----	-------------------	------



물리적 설계

- 하드웨어나 운영체제 특성을 고려하여 필요한 **인덱스 구조나 내부저장구조에 대한 물리적 구조를 설계**
 - 특정 DBMS에 구현 가능한 형태의 데이터 타입, 오프셋, 인덱스 등의 정보를 포함
 - 결과물: 물리적 스키마, 테이블 명세서

속성이름	데이터타입	널허용	기본값	기본키	외래키	제약조건
회원아이디	VARCHAR(20)	N		PK		
비밀번호	VARCHAR(20)	N				
이름	VARCHAR(20)	N				
나이	INT	Y				0이상
직업	VARCHAR(20)	Y				
등급	VARCHAR(10)	N	Silver			silver, gold, vip만 허용
적립금	INT	N	0			

구현

- DBMS에서 SQL 언어를 사용하여 실제 데이터베이스를 생성하는 단계
 - **CREATE TABLE** 제품(
제품번호 **CHAR(3) NOT NULL PRIMARY KEY**,
제품명 **VARCHAR2(20)**,
재고량 **NUMBER(5) CHECK (재고량 >=0 AND 재고량 <= 10000)**,
단가 **INT**,
제조업체 **VARCHAR2(20)**);

감사합니다!

담당교수: 전강욱(컴퓨터공학부)

kw.chon@koreatech.ac.kr

예제2: 항공사 DB 설계

■ 요구사항

- 항공사에 회원으로 가입하려면 회원아이디, 비밀번호, 성명, 신용카드정보를 입력해야 한다.
- 회원의 신용카드 정보는 여러 개를 저장할 수 있는데, 세부적으로 신용카드번호, 유효기간을 저장할 수 있다.
- 항공사에서는 보유한 비행기에 대해 비행기번호, 출발날짜, 출발 시간 정보를 저장하고 있다.
- 항공사에서는 좌석에 대해 좌석번호, 등급 정보를 저장하고 있다.
- 회원은 좌석을 예약하는데, 회원 한 명은 좌석을 하나만 예약할 수 있고, 한 좌석은 회원 한 명만 예약 할 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.

예제2: 항공사 DB 설계

■ 개념설계

□ 개체, 속성, 관계 추출

- 개체: 회원, 비행기, 좌석 (아래에서 밑줄은 속성)
- 관계: 예약, 좌석

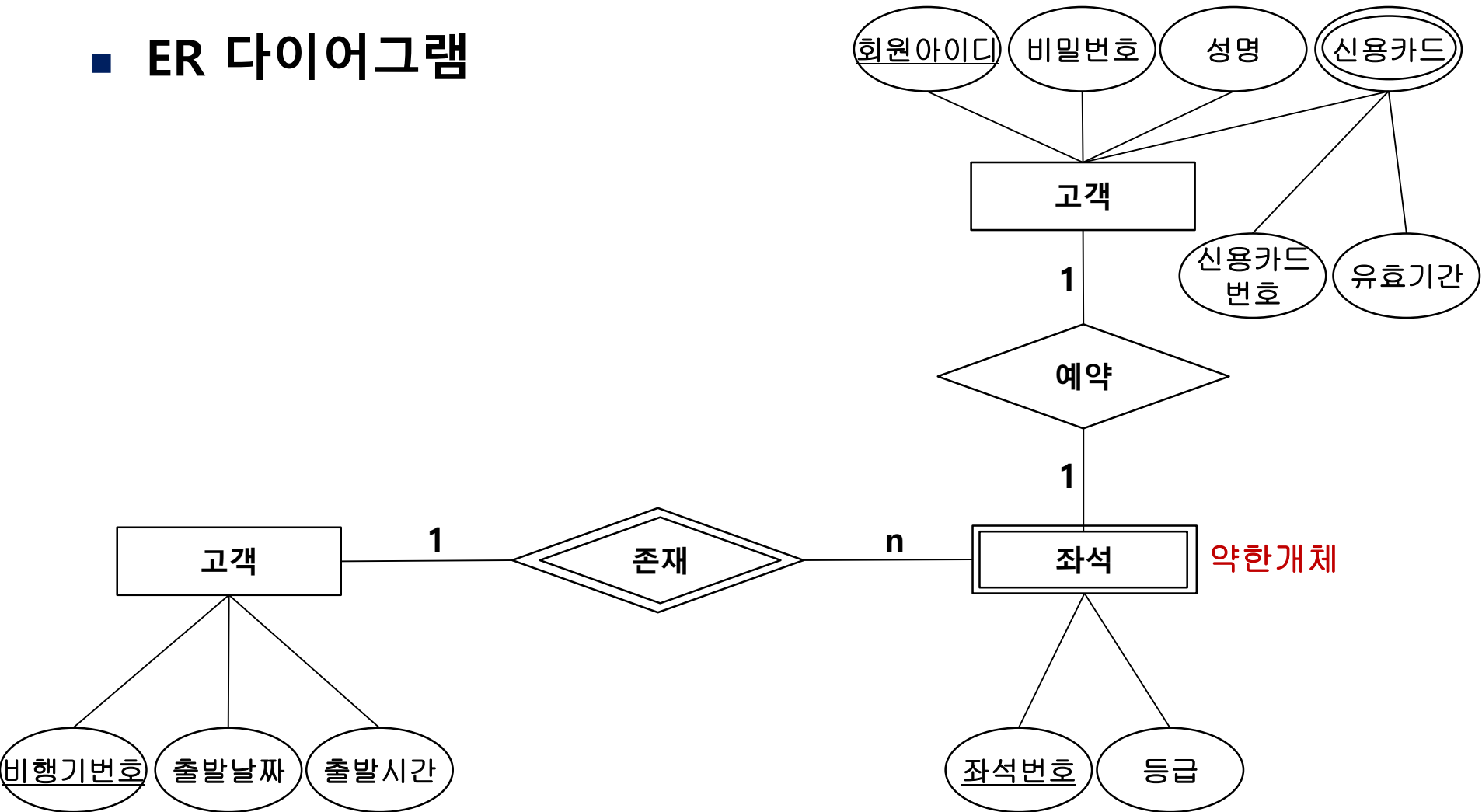
□ 관계 추출

- 항공사에 **회원**으로 가입하려면 회원아이디, 비밀번호, 성명, 신용카드정보를 입력해야 한다.
- 회원의 신용카드 정보는 여러 개를 저장할 수 있는데, 세부적으로 신용카드번호, 유효기간을 저장할 수 있다.
 - 여러 개: 다중 값 속성
 - 신용카드번호, 유효기간: 복합속성
- 항공사에서는 보유한 **비행기**에 대해 비행기번호, 출발날짜, 출발시간 정보를 저장하고 있다.
- 항공사에서는 **좌석**에 대해 좌석번호, 등급 정보를 저장하고 있다.
- 회원은 좌석을 **예약**하는데, 회원 한 명은 좌석을 하나만 예약할 수 있고, 한 좌석은 회원 한 명만 예약 할 수 있다.
 - 1:1의 관계
- 비행기에는 **좌석**이 존재하는데, 비행기 하나에는 좌석이 여러 개 존재할 수 있고, 한 좌석은 반드시 하나의 비행기에만 존재해야 한다. 그리고 좌석은 비행기가 없으면 의미가 없다.
 - 1:n 관계, 좌석은 약한개체

예제2: 항공사 DB 설계

다중값, 복합속성

■ ER 다이어그램



예제2: 항공사 DB 설계

■ 릴레이션 스키마

□ 규칙1: **개체와 속성**은 **릴레이션으로 변환**

- 비행기(비행기번호, 출발날짜, 출발시간)
- 좌석(좌석번호, 등급)
- 회원(회원아이디, 비밀번호, 성명, **신용카드번호**, 유효기간) **복합속성**

□ 규칙 3-2적용: **약한 개체의 일 대 다** 관계는 **외래키포함 기본키 적용**

- 좌석(좌석번호, 등급, 비행기번호)

□ 규칙 4적용: **일 대 일** 관계는 **서로 외래키로 삽입**

- 좌석(좌석번호, 등급, 비행기번호, **회원아이디**) **FK**
- 회원(회원아이디, 비밀번호, 성명, **좌석번호**, **비행기번호**, 신용카드번호, 유효기간)

□ 규칙 5 적용: **다중 값 속성**은 **독립적인 릴레이션으로 변환**

- 회원_신용카드(회원아이디, 신용카드, 유효기간) **FK**

예제2: 항공사 DB 설계

■ 최종 결과 릴레이션 스키마

- 비행기(비행기번호, 출발날짜, 출발시간)
- 좌석(좌석번호, 등급, 비행기번호, 회원아이디)
- 회원(회원아이디, 비밀번호, 성명, 좌석번호, 비행기번호)
- 회원_신용카드(회원아이디, 신용카드정보, 유효기간)