

DATA MODEL의 정규화

- 정규화 작업은 관계형 데이터베이스의 개념이지만 그 원칙은 개념 데이터 모델링에 적용된다
- 정규화 단계별 규칙을 사용하여 ATTRIBUTE의 위치를 적절히 하는 것
- 정규화 작업이 완료된 E-R Model은 바로 정규화된 관계형 데이터베이스 설계로 옮겨진다.
- 제3정규형은 데이터중복을 제거하기 위한 데이터베이스 설계의 일반적인 목표 그 이상의 정규형은 별로 폭 넓게 사용되지 않는다.

제 1 정규형

모든 **ATTRIBUTE**는 반드시 하나의 값을 가져야 한다
(반복형태가 있어서는 안됨)

제 2 정규형

모든 **ATTRIBUTE**는 반드시 **UID** 전부에 종속되어야 한다
(**UID** 일부에만 종속되어서는 안됨, **Unique Identifier**, **Primary Key**)

제 3 정규형

UID 가 아닌 모든 **ATTRIBUTE** 간에는 서로 종속될 수 없다
(**ATTRIBUTE** 간 종속성 배제)

*선행작업: <전국 대학생 회원> 테이블을 설계하시오.→
정규화가 필요한 이유를 찾아 설명한 후 정규화를 설명할 것

제 1 정규형

제 1 정규형

모든 **ATTRIBUTE**는 반드시 하나의 값을 가져야 한다
(반복형태가 있어서는 안됨)

고객

고객번호
고객명
계약일

← 고객이 여러번 계약을 하게 된다면 ?

1NF ?

No

계약하여

계약

계약일
장소
결과



주체가되어

고객

#* 고객번호
* 고객명

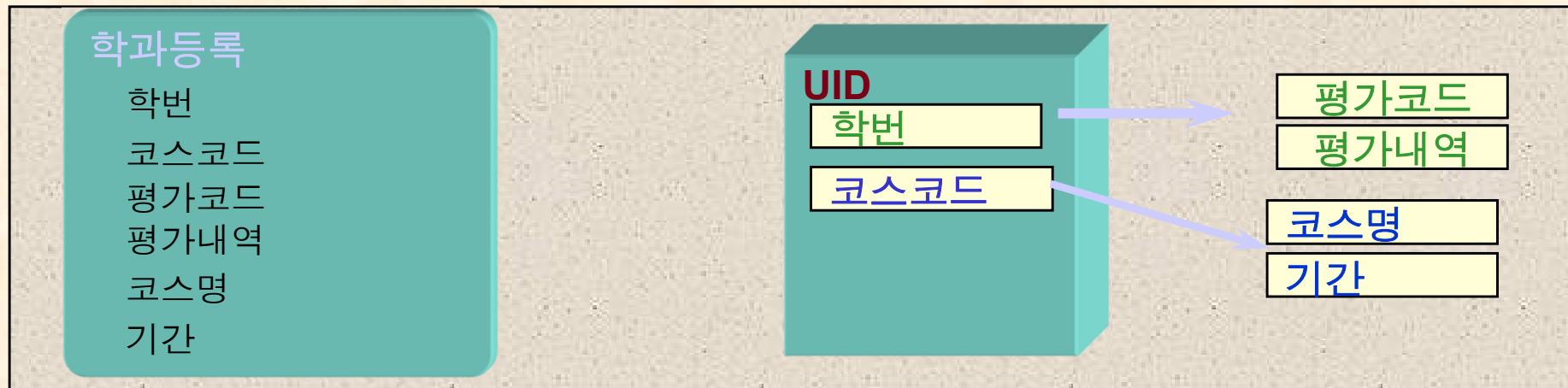
어떤 **ATTRIBUTE**가 다수의 값을 가지고 있다면

M : 1 관계의 새로운 **ENTITY**를 추가한다

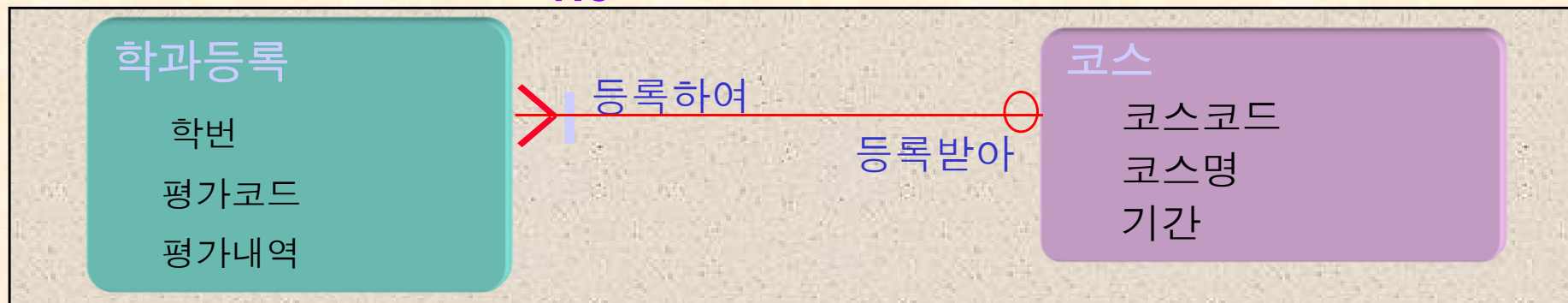
제 2 정규형

제 2 정규형

모든 **ATTRIBUTE**는 반드시 **UID** 전체에 종속되어야 한다 (**UID** 일부에만 종속되어서는 안 됨)



2NF ?
No

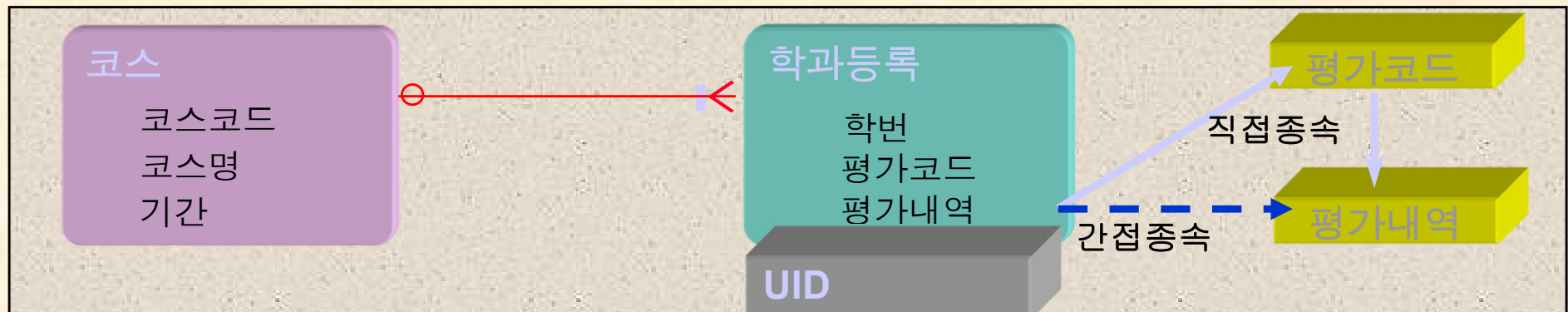


어떤 **ATTRIBUTE**가 **UID** 전체에 종속되어 있지 않으면
잘못된 위치이며 새로운 **ENTITY**를 만들거나 추가한다

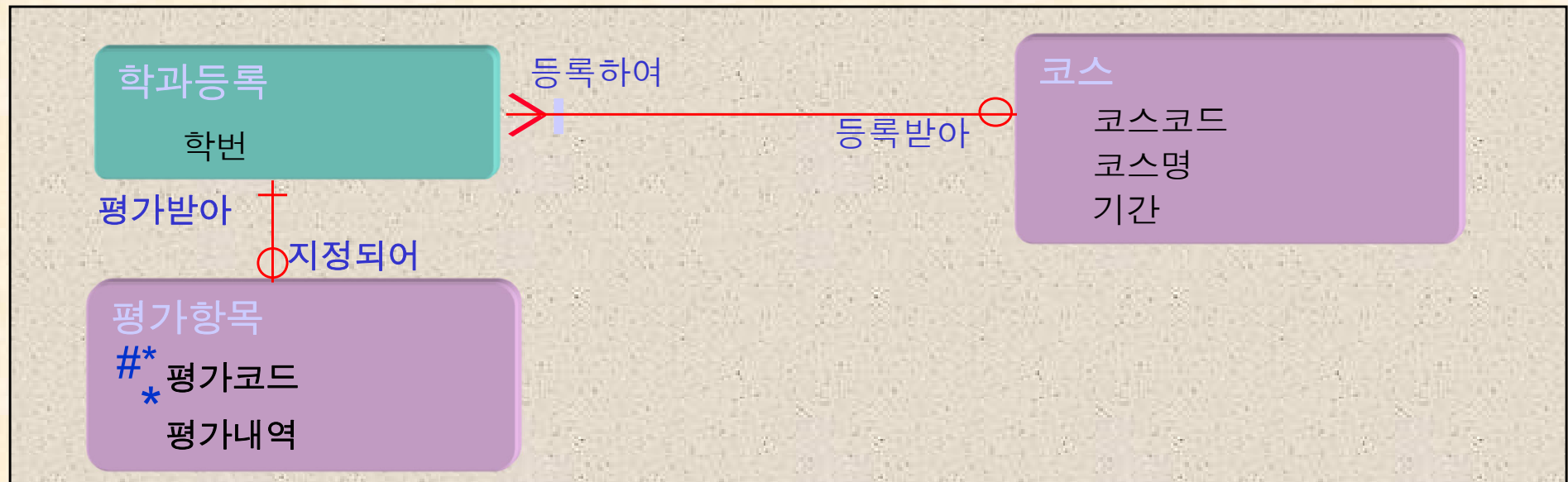
제 3 정규형

제 3 정규형

UID 가 아닌 모든 **ATTRIBUTE** 간에는 서로 종속될 수 없다
(**ATTRIBUTE** 간 종속성 배제)

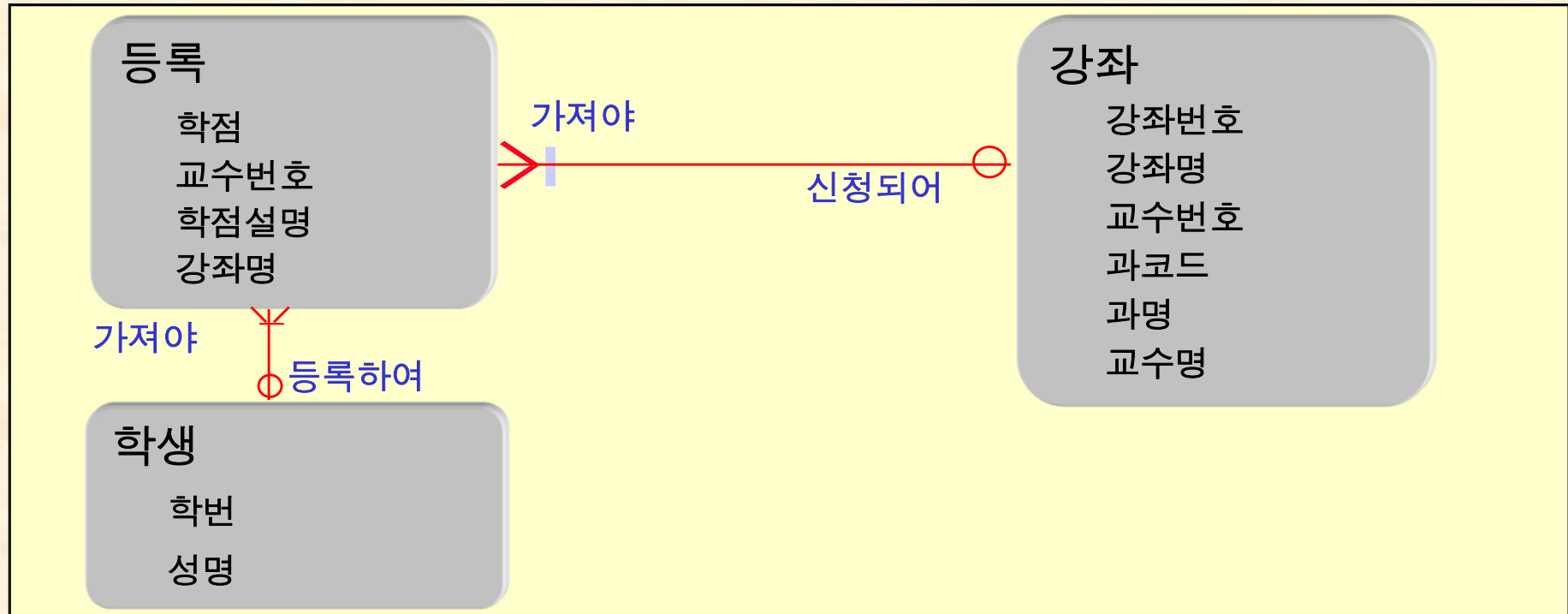


3NF ?
No



연습문제

정규화 작업



- 위의 **E-R** 모델을 보고 제3 정규형에 맞도록 다시 작성하시오.
- **CRUD**: 데이터베이스에서 데이터 처리 기능. **Create, Read, Update, Delete**
- **ACID**: 관계형 데이터베이스에서 **Transaction**이 안전하게 처리되기 위한 속성
- **Atomicity, Consistency, Isolation, Durability**: 트랜잭션 직렬성(**Serialization**) 보장