5. Learning from Data Model

Spring 2019

Schedule*

0. Seminar Overview
1. Overview of the Relational Data Model
2. Why DB Design?
3. Development Process
4. Requirements
5. Conceptual Data Model
6. Generalization & Specialization
7. Relational Database Design
8. Normalization
9. Keys and Constraints

*subject to change without notification

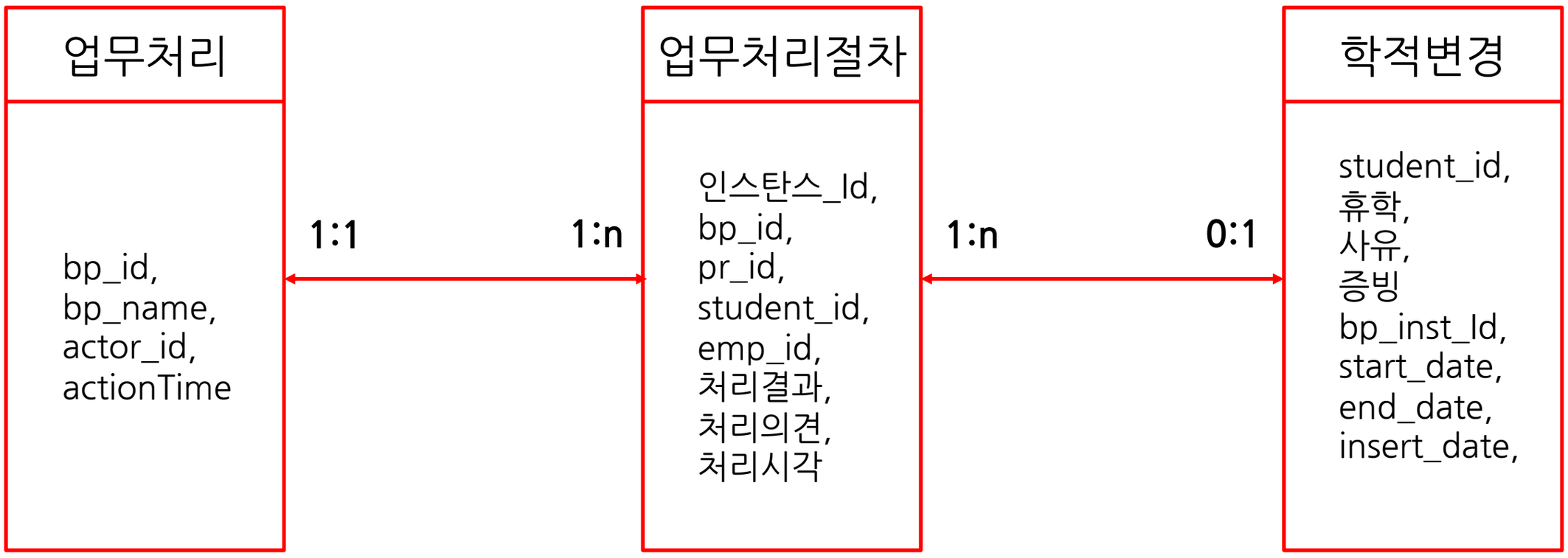
- 데이터 모델을 보다 면밀히 검토하여 데이터베이스 시스템에 대한 우리의 이해를 향상시킬 수 있는지 살펴 봅니다.
- 데이터 모델은 실제 문제에 대해 저장된 데이터의 정확한 설명입니다.
- 데이터 모델은 실제 상황에 대한 완전하거나 정확한 설명이 아닙니다. 항상 정의와 가정에 기초한 것이며, 지정된 범위내에서만 적용할 수 있습니다.
- 문제를 위하여 저장된 데이터 항목 간의 관계 모델이지만 실제 문제 자체의 완전한 모델은 아닙니다.
- 돈, 시간 및 전문 지식에 대한 제약은 필수 요소를 추출하기 위해 문제의 범위와 가정을 뜻합니다. 사용자와 분석가가 서로 다른 목표를 지향하지 않도록 정의와 가정을 명확하게 표현하는 것이 중요합니다.
- 초기 데이터 모델에서 정의와 범위를 보다 엄격하게 표현해야 할 부분을 파악하는 방법을 살펴봅니다.

Today's Lecture

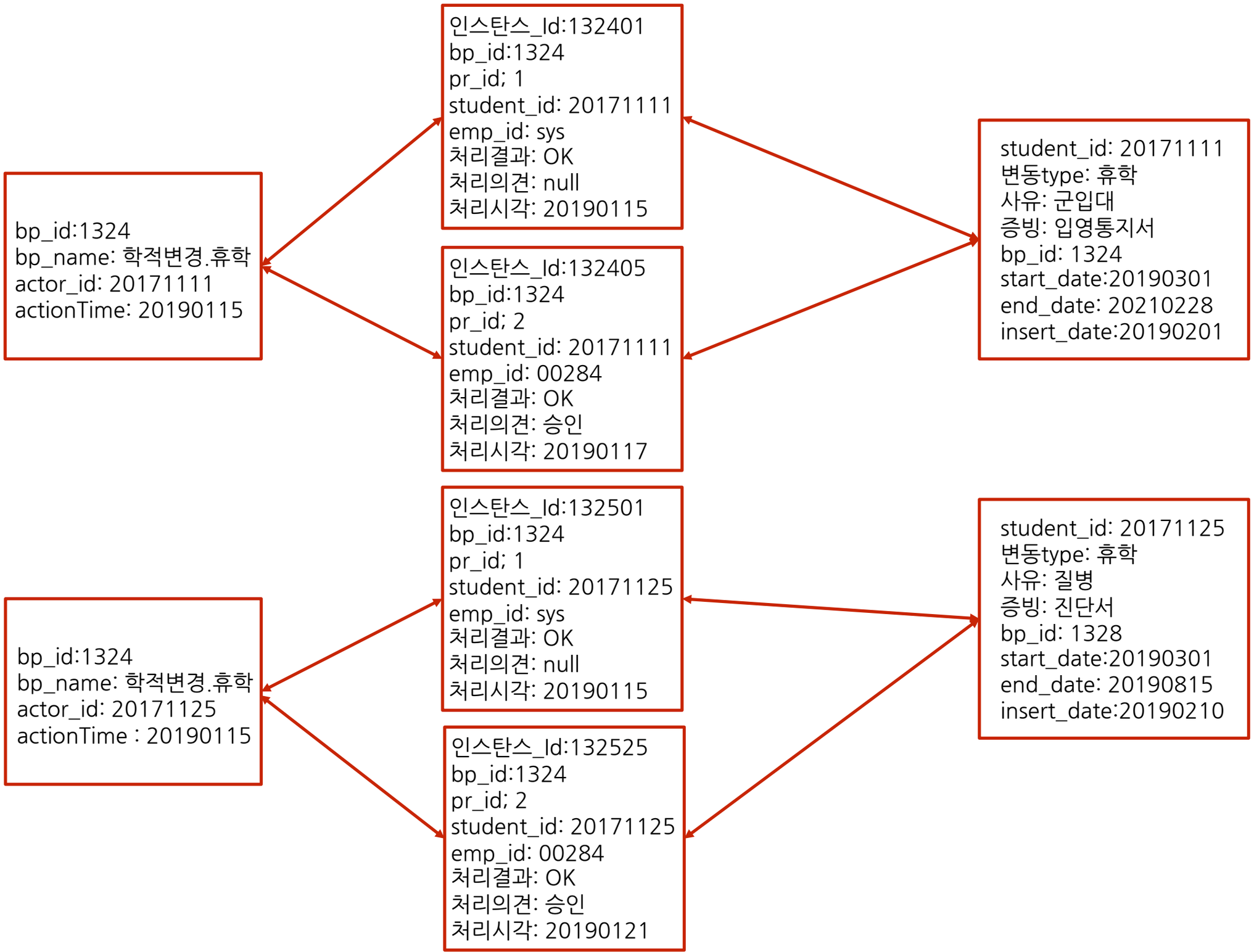
- Review of Data Model
- Optionality: Should It Be 0 or 1?
- A Cardinality of 1: Might It Occasionally Be Two?
- A Cardinality of 1: How About Historical Data?
- A Many-Many: Are We Missing Anything?
- Exercises
- Summary

Review of Data Models

- 몇 가지 추가 기능을 보일 수 있는 예제를 통해 데이터 모델의 필수 요소를 다시 검토합니다.



- 휴학레코드는 학적변경 업무처리들의 순서로 구성되며 각종 승인 업무처리절차는 승인 업무처리의 한 단계에 해당합니다. 예를 들면 휴학 업무처리절차는 휴학 승인 업무처리는 학적변경업무의 한 종류이며 한 학생의 휴학 기록은 학적변경 업무처리절차의 기록 순서로 구성되어 있습니다.



- *학적변경 업무처리들의 순서*의 정의?
- 업무처리는 단순한 업무들의 집합이 아니며, 업무처리의 순서를 정의하는 절차가 있습니다.
- 홍길동이 신청한 휴학 승인을 어떻게 기록할까요? 또한 각 승인 업무를 기록하여야 합니까?
- 모든 승인 업무를 처리하기 위하여는 어떻게 설계되어야 하나요?
- 이들에 대한 변경을 어떻게 수용할 수 있습니까?

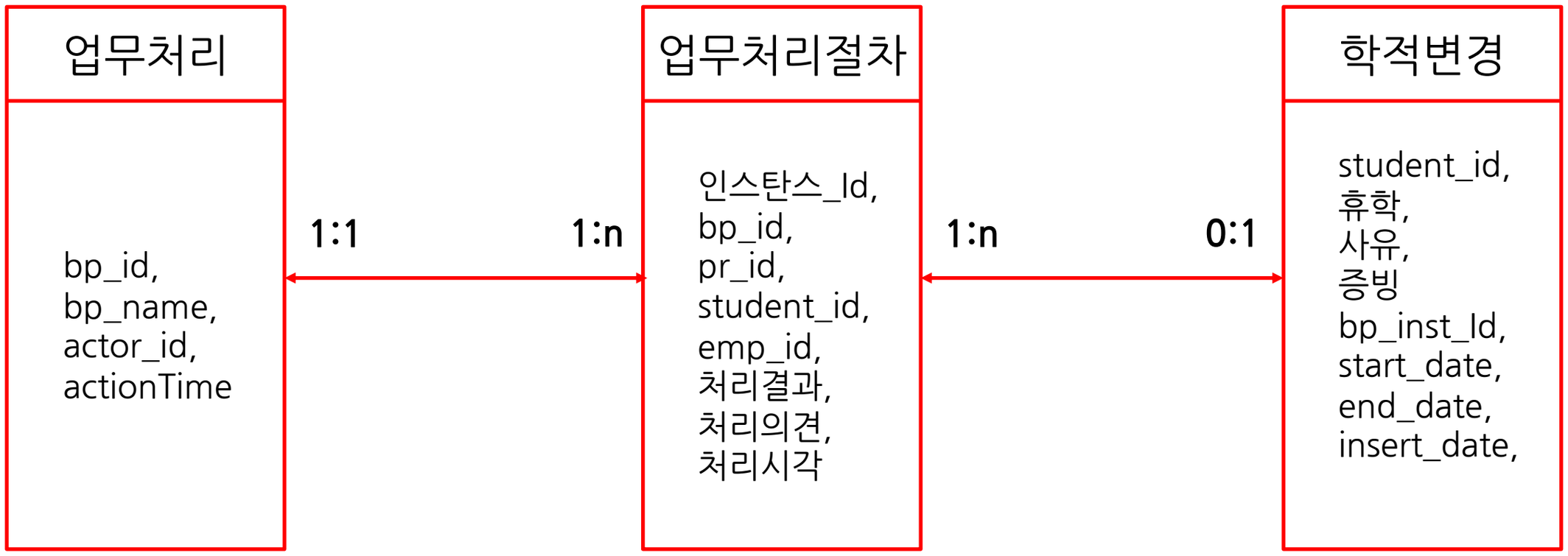
*학적변경 업무처리를 위하여, 먼저 학적변경에는 변경타입이 있으며 (예: 휴학, 복학 등), 신청자와 각 승인 또는 검토 업무들로 구성된다. 각 업무에는 정해진 **순서**가 있다. 승인 (또는 검토) 업무에는 업무에 따라 승인자를 지정하며, 처리결과, 사유, 처리시각을 기록한다.*

살펴볼 질문은 두 클래스 간의 관계에만 적용되지만 문제에 대한 많은 토론을 시작할 수 있습니다. 문제에 대해 더 많이 이해할수록 데이터 모델에서 배운 것을 use case에 반영 할 수 있습니다.

- **Optionality**: Should it be 0 or 1?
- **Cardinality of 1**: Might it occasionally be 2?
- **Cardinality of 1**: What about historical data?
- **Many-Many**: Are we missing anything?

Optionality: Should It Be 0 or 1?

관계의 한쪽 끝의 optionality는 다른 쪽 끝에있는 오브젝트와 연관 될 수있는 최소 오브젝트입니다. 이것은 일반적으로 0 또는 1입니다. 예를 들어, 업무처리절차와 학적변경의 관계를 왼쪽에서 오른쪽으로 읽는 경우, 특정 업무처리절차는 학적변경에 (optionality 0)해당되지 않을 수 있으며, 오른쪽에서 왼쪽으로 관계를 읽는 경우, 특정 학적 변경 기록은 관련 업무처리절차를 가져야 합니다. (optionality 1).

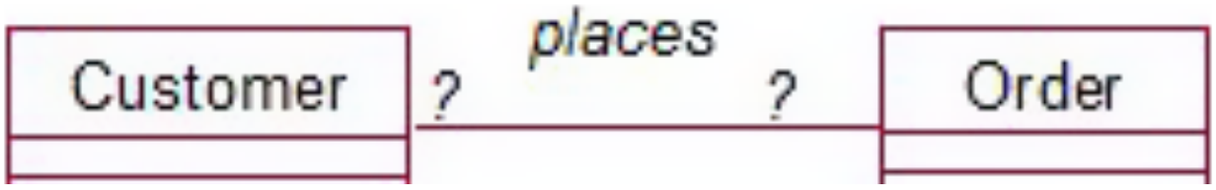


Student Course Example

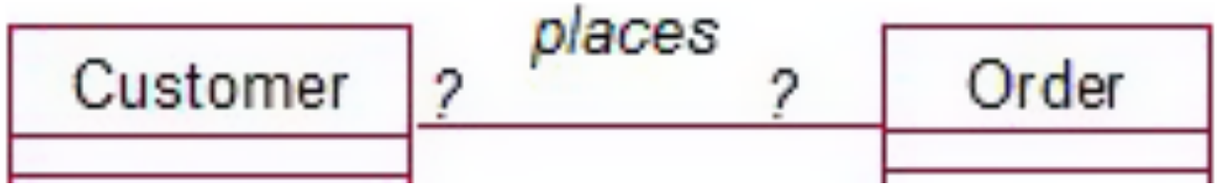
- 학생은 여러 과목 수강할 수 있으며, 한 과목에는 여러 수강생이 등록할 수 있습니다. optionality는 어떻습니까? 학생이 어떤 과목도 등록하지 않을 수 있나요? 재학생의 정상적인 대화식 정의는 무엇입니까? 공식적으로 등록한 사람입니까?
- 이 데이터에이스에서 학생의 정의는?
- 학생의 정의를 대학에서 인정하거나 대학에 등록한 사람들을 포함하도록 확대함으로써 이러한 상황을 명확히 할 수 있습니다.

- 대학에 연락하여 등록에 관한 정보를 요청한 사람은 어떠한 정보를 받을 수 있습니까?
- "학생이라고 부르는 어떠한 사람들 입니까? " 는 분석 과정의 시작 단계에서부터 바로 고려하여야 합니다.
- 가장 단순한 데이터 모델에서 라도 세부 사항을 신중하게 고려하면 문제의 더 넓은 측면에서 중요한 질문이 될 수 있습니다.
- 교과목을 어떻게 정의합니까? 교과목에 대해 어떤 데이터를 보관할 수 있습니까?

Customer Order Example

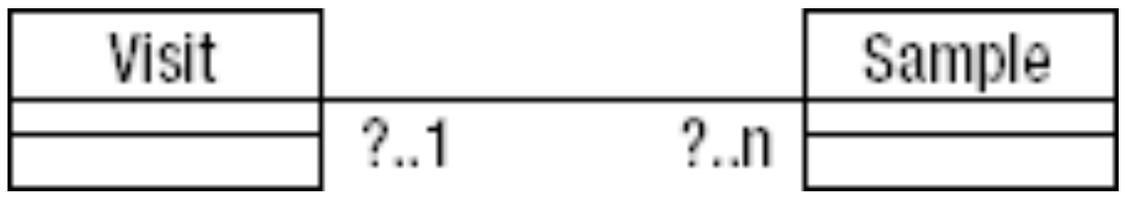


- 고객 및 주문한 주문에 대한 정보를 보관합니다. 첫 번째 의견은 "고객은 많은 주문을 할 수 있고, 각 주문은 오직 한 고객으로 부터 받았다." 입니다.
- Optionalities는?
- 주문을 하지 않은 고객도 있을 수 있습니까?
- *주문을 한 사람과 카탈로그를 받는 사람*
- "예전에 주문을 하였지만 지금은 카탈로그만 받는 사람을 식별 할 수 있기를 원하십니까?"



- 주문에는 해당 고객이 있어야 합니까?
- 이름이나 주소가 없이 메일로 주문이 들어 올 수 있습니까?
- "모든 주문에는 고객이 있어야 한다"고 주장 할 수 있습니다 (optionality 1).
- 지금 당장 이 문제들을 해결하려고 노력하지 않고 단순히 데이터 모델을 사용하여 이를 명확하게 하려고 합니다.

Insect Example



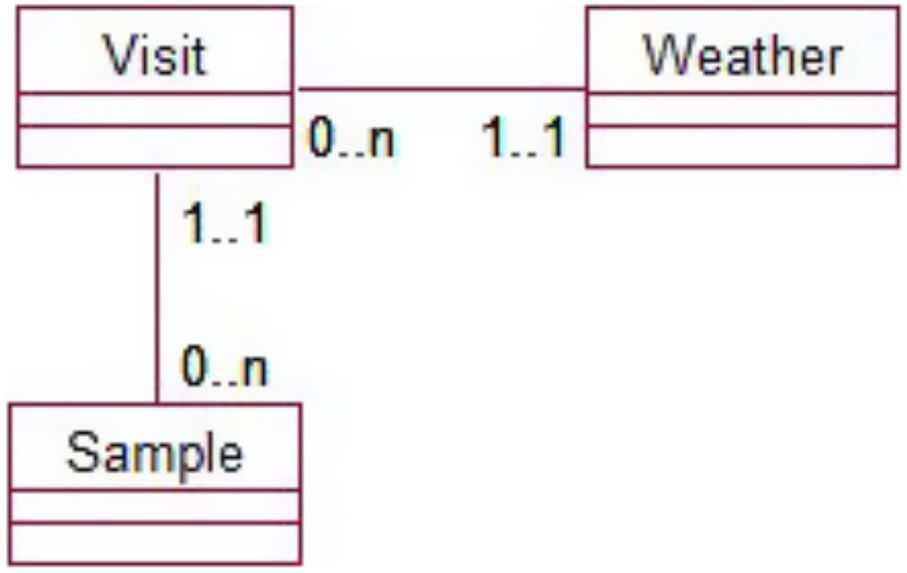
- 농장을 방문하여 곤충 표본을 수집했습니다. *Visit* 객체는 방문 날짜 및 방문시의 조건에 대한 정보를 포함하며 여러 Sample 객체와 관련됩니다. 각 *Sample* 객체는 채집한 곤충의 수에 대한 데이터를 포함하고 있습니다.
- 샘플에는 방문에 대한 정보가 필요합니까?
- 각 방문마다 이때 채취한 샘플이 있어야 하는지 여부

A Cardinality of 1: Might It Occasionally Be Two?

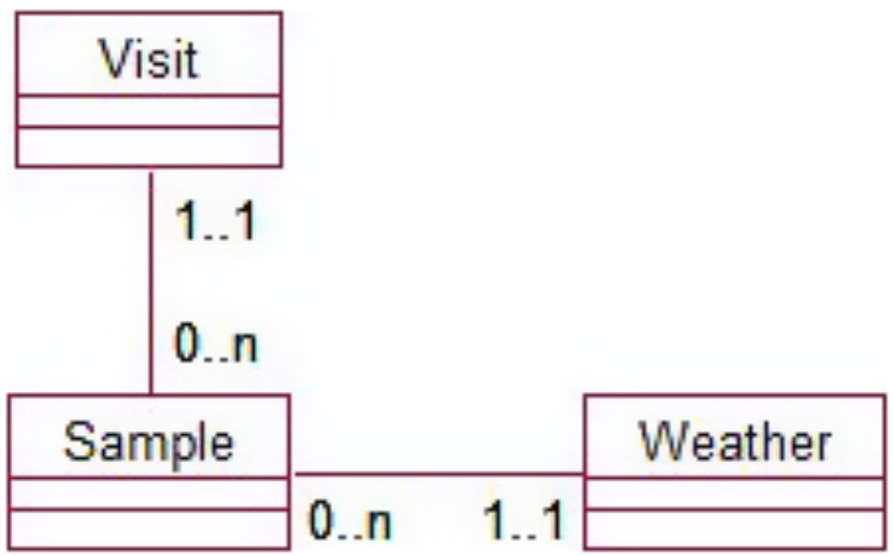
- 데이터베이스에서 발생할 수 있는 모든 데이터에 대한 적절한 처리 여부를 확인하기 위하여 다양한 시나리오를 신중하게 고려하십시오. 일부 "예외"는 실제로 간과한 문제입니다. 실생활과 실제 문제는 항상 복잡합니다.
- 그럼에도 불구하고 문제의 완전한 개혁을 보장하지는 않지만 그럼에도 불구하고 데이터베이스의 생명주기동안 나타날 수 있는 "예외"를 처리하는 방법에 대하여 생각하여야 합니다.

Insect Example

- 기상 조건을 일관되게 기록하기 위해 사용자는 여러 카테고리 중 하나를 선택할 수 있도록 합니다. 서로 다른 조건 (예 : 맑음, 갬, 흐림, 비 등)의 객체를 포함하는 날씨 클래스를 도입하면 이 정보를 일관되게 기록할 수 있습니다.



- 각 샘플을 수집하는 기상 조건이 중요할 수 있습니다. 이 경우 각 샘플을 기상 조건과 연관시키는 것이 더 바람직할 수 있습니다.
- This latter solution may be overkill when the majority of visits have stable weather conditions. 후자의 해결책은 대부분의 방문이 안정된 기상 조건일 때에는 과도하다 할 수 있습니다.
- 날씨가 크게 바뀌면 다른 방문 객체를 생성할 것입니다. 이 방법으로 모든 방문에는 하나의 연관된 날씨를 갖을 수 있습니다. 방문이 의미하는 바를 다시 정의함으로써 "예외적인" 사례에 대처할 수 있습니다.
- *방문은 하루 동안 일정한 기상 조건하에서 농장에서 지낸 시간입니다. 농장을 하루에 한 번이상 방문할 수 있습니다.*



Sports Club Example

- 지역 스포츠 클럽은 회원 목록과 각 회원이 현재 활동하고있는 팀 (SeniorB, JuniorA, Veteran 등)들을 관리하고자 합니다. 이 데이터를 모델링하는 한 가지 방법입니다.

- 회원이 소속될 수 있는 최대 팀 수에 대해 알아야 합니다.
- "한 회원이 둘 이상의 팀을 위해 경기를 할 수 있습니까? 그렇다면 어떻게 처리할까요?”
- "plays for" 관계는 플레이어가 플레이 할 수있는 팀이라기 보다 플레이어의 메인 팀을 의미합니다.
- 데이터 모델은 역사적인 기록을 유지하지 않습니다.
- 부상이나 질병으로 인해 특정 팀원이 다른 팀을 위해 어떤 경기를 치러야하는 상황이 데이터 모델에 어떤 영향을 미칩니까? 이것은 범위의 문제입니다.

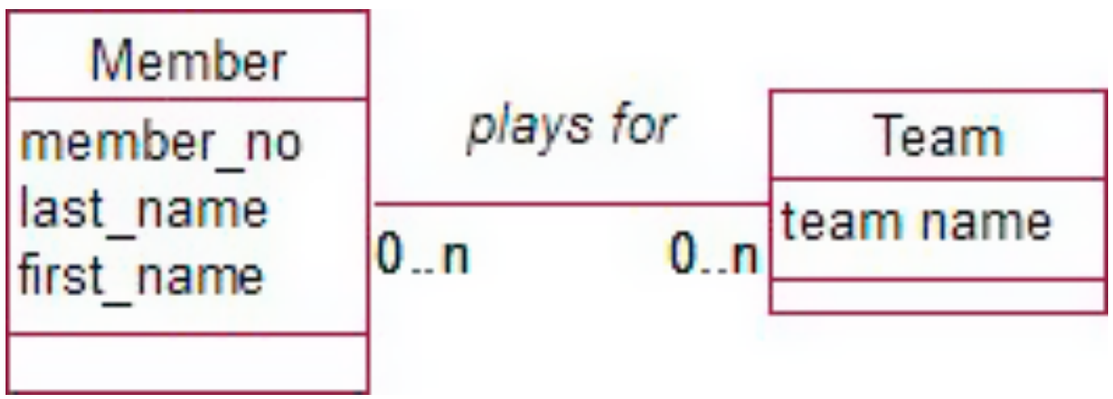
A Cardinality of 1: What About Historical Data?

- 한쪽의 카디널리티가 1 인 관계의 예는 여럿있습니다.
- 시간이 지남에 따라 한 방에는 많은 손님이 묵을 수 있고, 한 플레이어는 여러 팀에서 플레이할 수 있으므로 단어 추가에 신중했습니다.
- "시스템에서 이전 손님이나 이전 소속 팀을 추적하고 싶습니까?"
- 스포츠 클럽의 첫 시즌에 시스템이 잘 작동하지만, 다음 시즌 이전 팀이 교체된다면 그 기록이 삭제되어 놀라움을 금치 못할 것입니다.

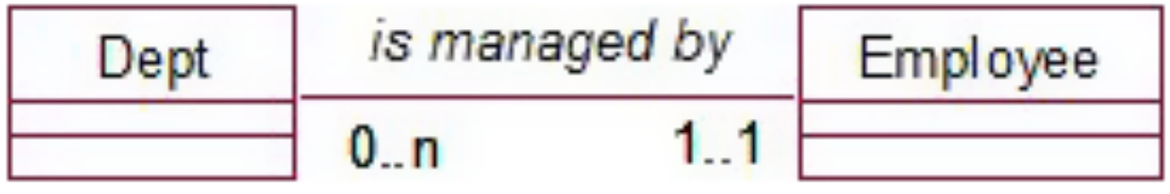
Sports Club Example

member_no	last_name	first_name	team
152	Abell	Walt	SeniorB
103	Anderson	James	JuniorA
276	Avery	Graeme	JuniorA
287	Brown	Bill	JuniorA
298	Burns	Lance	Veteran

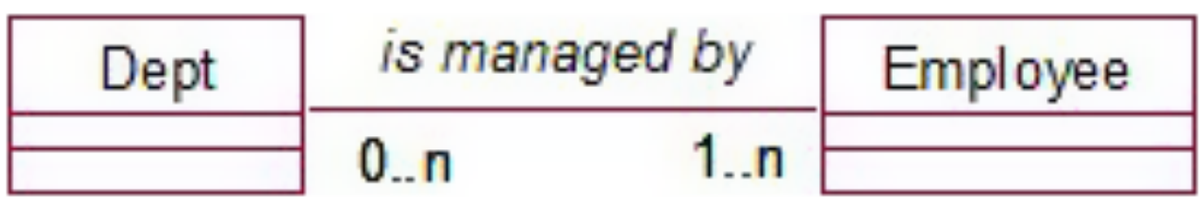
- 다음 시즌에 Bill Brown이 SeniorB 팀에 합류하면, JuniorA 팀과의 이전 관계는 사라집니다. 이력 데이터가 중요하다면, 회원들이 많은 팀과 관련될 것이라는 사실을 반영하도록 변경되어야 합니다.



Departments Example

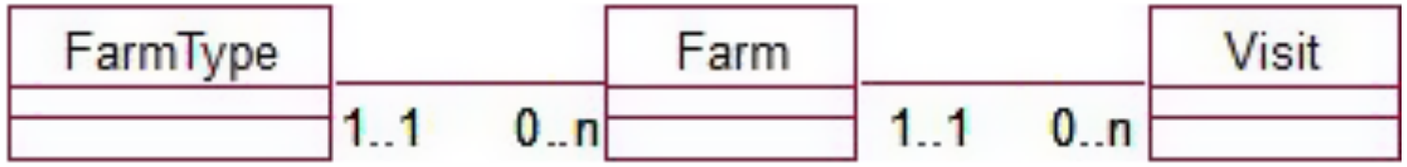


- "이전 관리자를 추적하고 싶습니까?"
- 작년에 어떤 일이 잘못되었을 때 담당 책임자를 알고 싶다면 이력을 유지하여야 합니다.



Insect Example

- 장기 프로젝트의 주요 목적은 농사법이 발전함에 따라 곤충의 수가 어떻게 변하는지를 확인하는 것이었습니다. 선정된 농장들은 다른 농사법 (유기농, 자르기 등) 을 채택하였습니다..
- 프로젝트 기간 동안 샘플을 수집하기 위해 각 농장을 여러 번 방문했습니다.



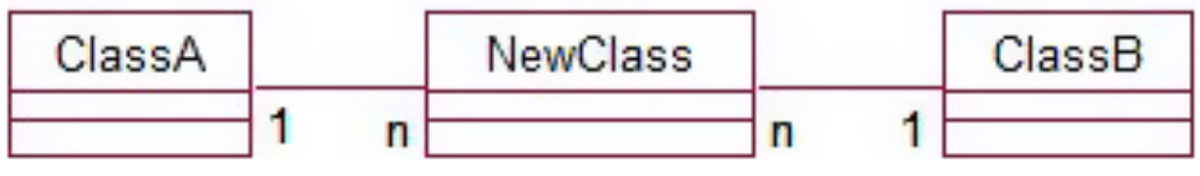
- 프로젝트가 진행되는 동안 농사법이 변경되지 않았기 때문에 프로젝트 기간 동안 샘플을 수집하기 위해 각 농장을 여러 번 방문했습니다.
- 농장이 결국 변경되면 이전 농사법이 유실됩니다.
- 농장은 한 번에 하나의 농사법만 채택할 수 있습니다.
- "시간이 지남에 따라 농사법이 변경될 수 있으며 시스템에 해당 이력데이터를 기록하는 것이 중요합니까?"

A Many–Many: Are We Missing Anything?

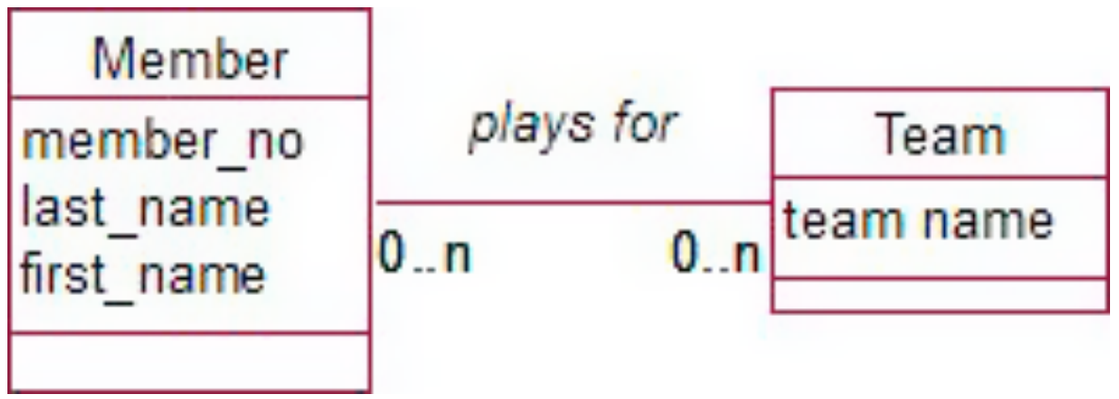
- 일부 사례에서 과거 데이터를 포함하도록 확대하면 1-Many 관계의 다수가 Many-Many 관계가 됩니다. (예 : 부서 관리자는 여러 명이며, 멤버는 플레이하는 팀이 많을 수 있으며, 농장에서는 오랜 기간 동안 여러가지 농사법을 사용하였습니다.)
- Many-Many 관계에 대한 몇 가지 추가 정보를 유지해야 합니다.
- 이력 데이터 어딘가에 첨부된 날짜가 없으면 별로 유용하지 않습니다. 그러나 날짜는 어디에 저장할 것인가?

다음과 같은 문제가 있습니다.

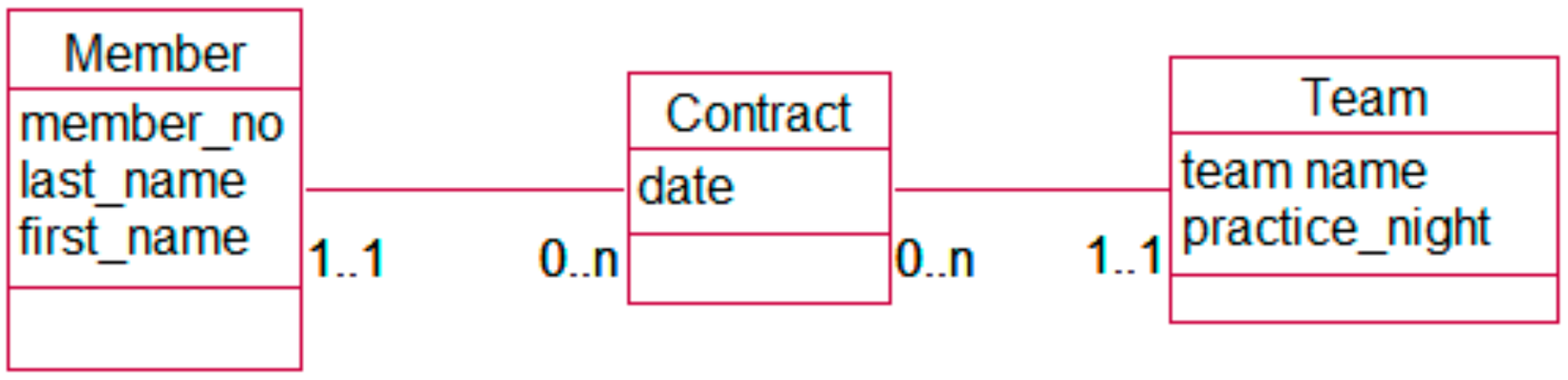
- *Many-Many 관계를 갖는 각 클래스의 특정 인스턴스에 따라 기록해야 하는 데이터가 있습니까?*
- *특정 플레이어와 특정 팀에 의존하는 데이터가 있습니까?*
- *플레이어가 해당 팀에서 플레이한 날짜입니다.*



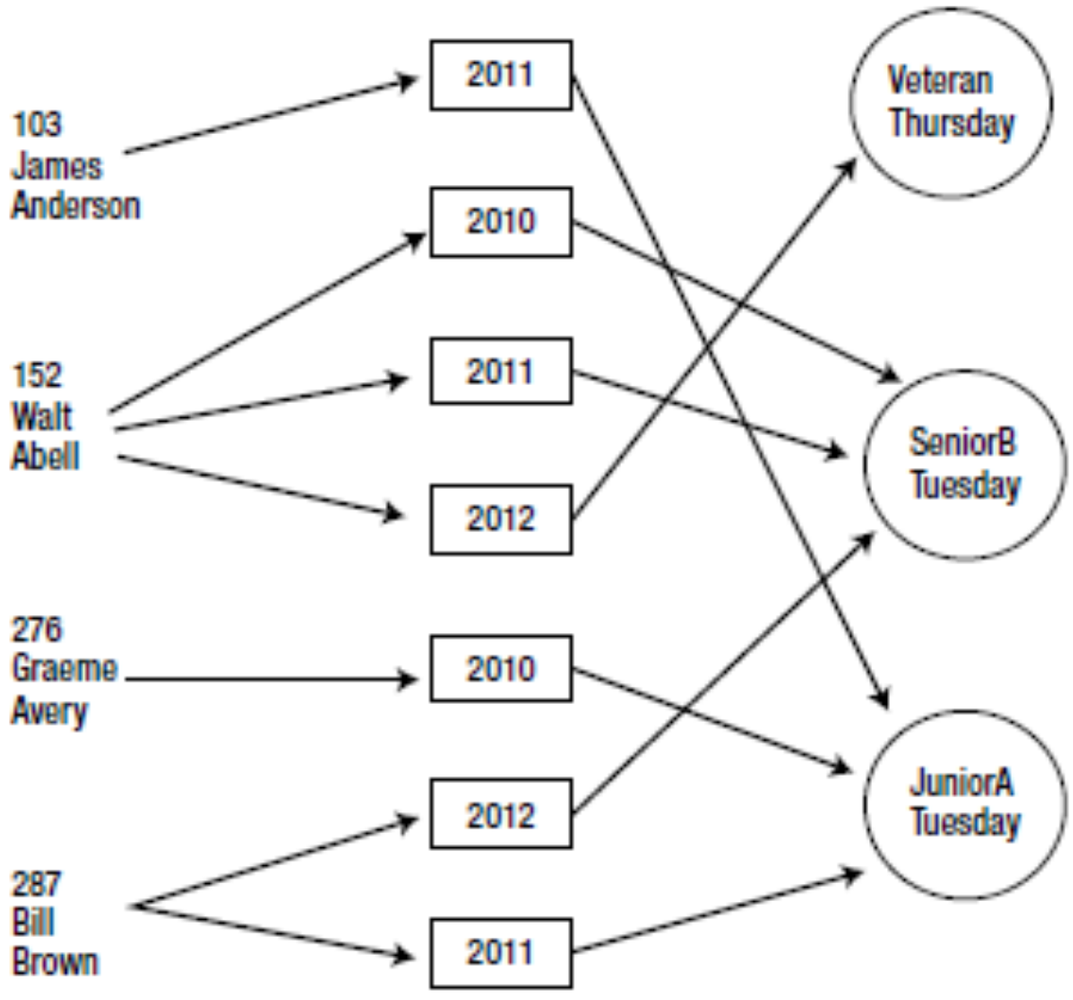
Sports Club Example



- 특정 팀에 대해 특정 멤버가 활동한 날짜는 **Member** 클래스 (멤버는 시간에 따라 여러 팀에서 플레이 할 수있기 때문에) 또는 **Team** 클래스에 선언할 수 없습니다.
- 새로운 중간 클래스 **Contract**를 정의합니다.



- 계약은 정확히 한 팀과 한 멤버를 위한 것입니다.
- 멥버들도 팀처럼 여러 계약을 체결할 수 있습니다.


103
James
Anderson
152
Walt
Abell
276
Graeme
Avery
287
Bill
Brown
2011
2010
2011
2012
2010
2012
2011
Veteran
Thursday
SeniorB
Tuesday
JuniorA
Tuesday

member_no	last_name	first_name
152	Abell	Walt
103	Anderson	James
276	Avery	Graeme
287	Brown	Bill
298	Burns	Lance

Member

team_name	practice_night
JuniorA	Tuesday
SeniorB	Tuesday
Veteran	Thursday
Under 18	Monday

Team

member	team	year
103	JuniorA	2011
276	JuniorA	2010
287	JuniorA	2011
287	SeniorB	2012
152	SeniorB	2010
152	Veteran	2012
152	SeniorB	2011

Contract

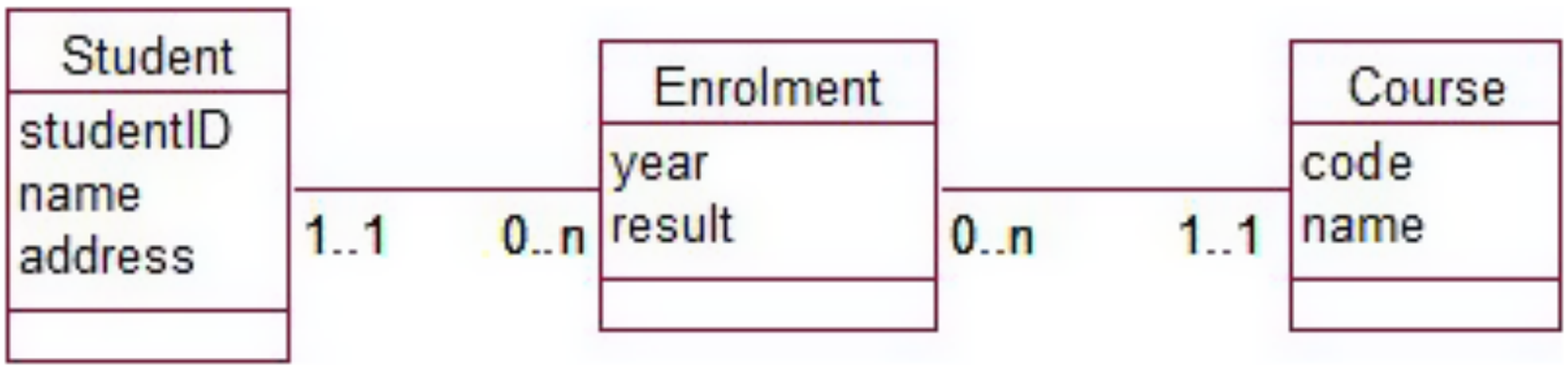
Student Course Example

- 과목을 수강하는 학생의 다대다 관계는 이력 데이터 문제일 뿐만 아니라 학생이 언제 과목을 이수했는 지도 알고 싶어합니다.

특정 학생과 특정 과목을 수강하는 데 필요한 특별한 데이터가 있습니까?

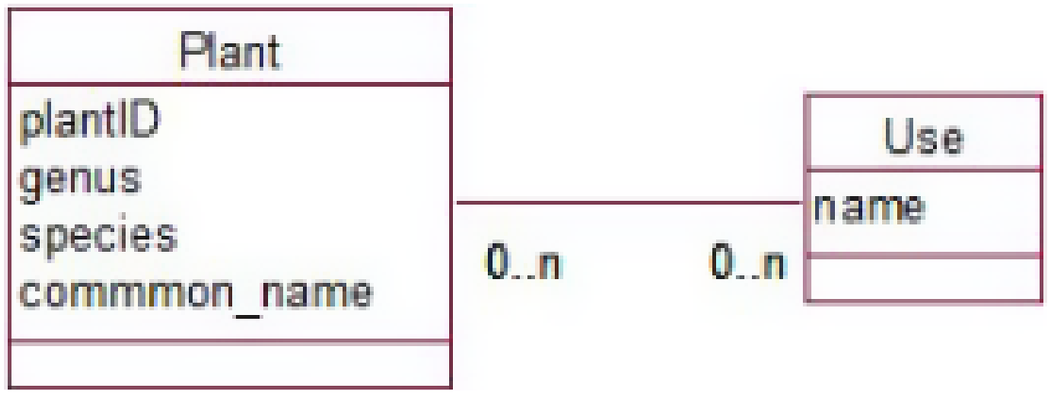
- 명백한 데이터는 **학점**입니다.

- 학생과 교과목은 각각 많은 수강기록을 포함할 수 있으나, 특정 하나의 수강은 정확히 한 학생과 한 코스에만 해당됩니다.



When a Many–Many Doesn't Need an Intermediate Class

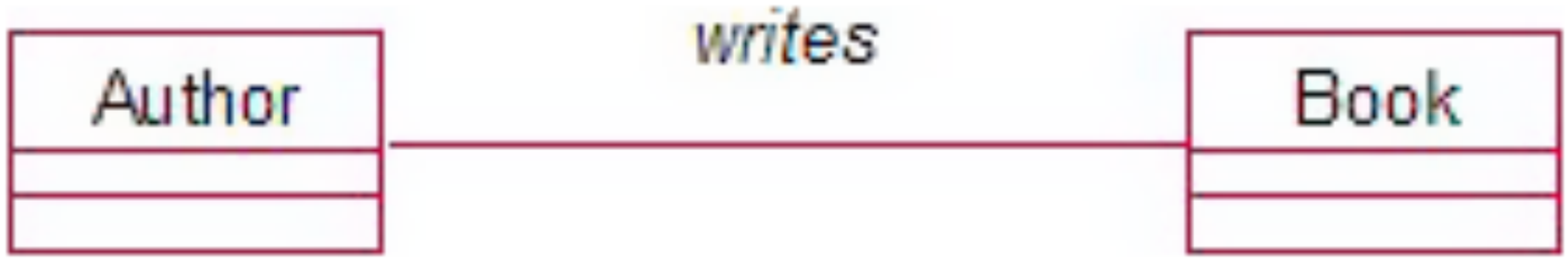
- 데이터 모델에서 몇몇 Many-Many 관계는 중간 클래스 없이도 문제에 대한 완전한 정보를 포함할 수 있습니다.
- 데이터로 카테고리를 표현하는 문제에 있어서는 종종 추가 클래스가 필요하지 않습니다.



- "특정 종과 특정 용도에 관해 유지하고 싶은 정보가 있습니까?"
  - 특정 식물이 헤지(hedging)에서 우수하거나 합리적인 지를 기록할 수 있습니다.
  - 벌을 유인할 수 있는 종이 어떤 것인지 알아야 합니다.

Exercise 5-1

아래 그림은 출판사가 저자와 책에 대한 정보를 유지하고자하는 상황을 모델링하는 첫 번째 초안입니다. write 관계의 양 끝에 있을 수 있는 optionality을 고려하여 책과 저자에 대한 몇 가지 가능한 정의를 시도하여 보십시오.



Summary

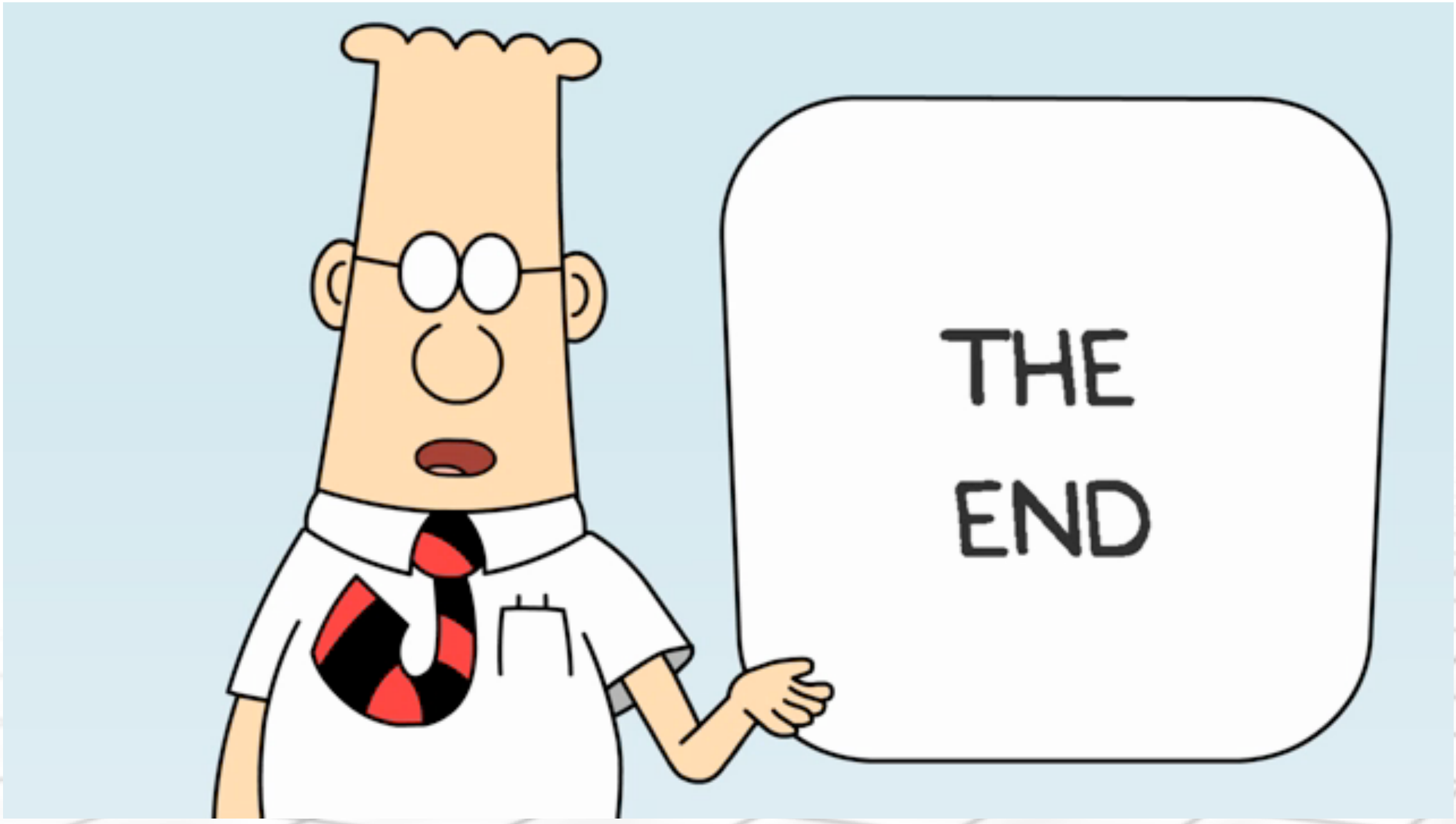
- 최종적으로 문제에 대한 설명은 use case에 반영되어야 하며 최종 모델 및 최종 구현에 영향을 미칠 수 있습니다.

  Optionality: 0 또는 1이어야 합니다. 0 또는 1 여부는 클래스 정의에 영향을 줄 수 있습니다. 예를 들어, "어떤 과목에도 수강하지 않는 학생을 여전히 데이터베이스에 학생으로 간주하여 저장하여야 합니까?"

  A cardinality of 1: 2일 수도 있을까요? 한 속성값으로 두 개의 숫자 나 카테고리를 저장하여야 하는 예외적인 경우를 고려해야 합니다. 예를 들어 "방문 중에 날씨가 바뀌면 어떻게 저장하여야 할까요?" 클래스를 재정의하면 예외적인 경우에 도움이 될 수 있습니다. "날씨가 바뀌면 두 번 방문한 것으로 취급합니다."

  A cardinality of 1: 이력 데이터는 어떻습니까? 관계에 있는 1은 실제로 "한 번에 하나씩"을 의미하는지 항상 고려하십시오. 예를 들어, "한 부서에는 관리자가 한 사람 있습니다. 우리 부서의 이전 관리자가 누군지 알고 싶습니까? "그렇다면, 관계는 다대다 입니다.

  Many-Many: 우리가 놓친 게 있니? 각 클래스의 특정 개체 쌍에 대해 기록해야하는 정보가 있는지 고려하십시오. 예를 들어, "한 학생과 특정 교과목에 대해 무엇을 알고 싶습니까?" 그러한 정보 (예 : 학년)가 있는 경우, 새로운 중간 클래스를 정의합니다.

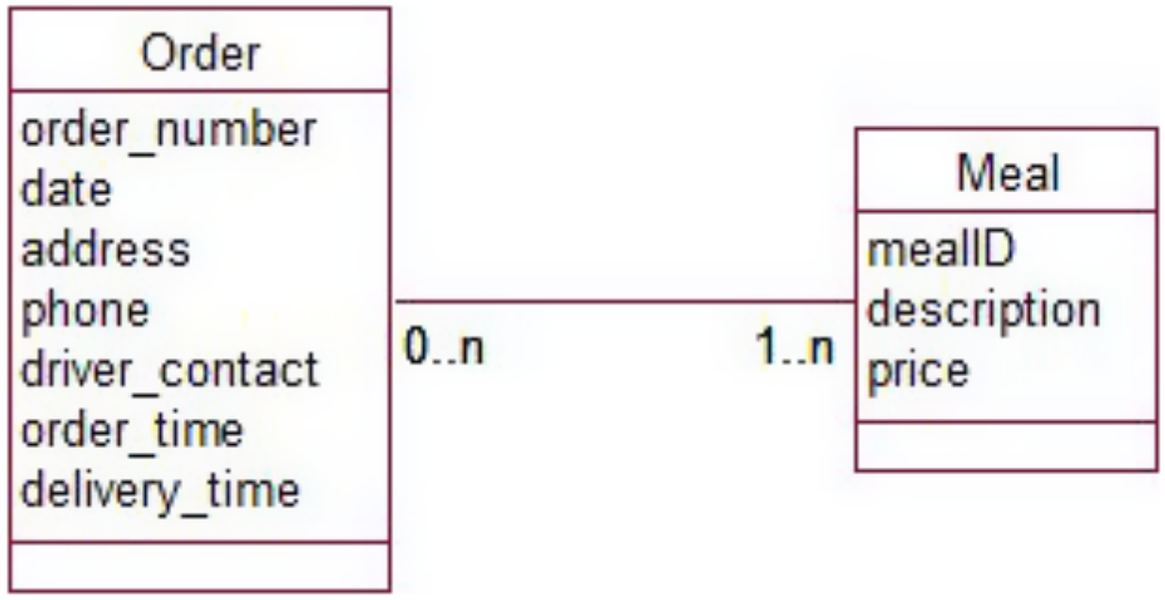


출처: metachannels.com

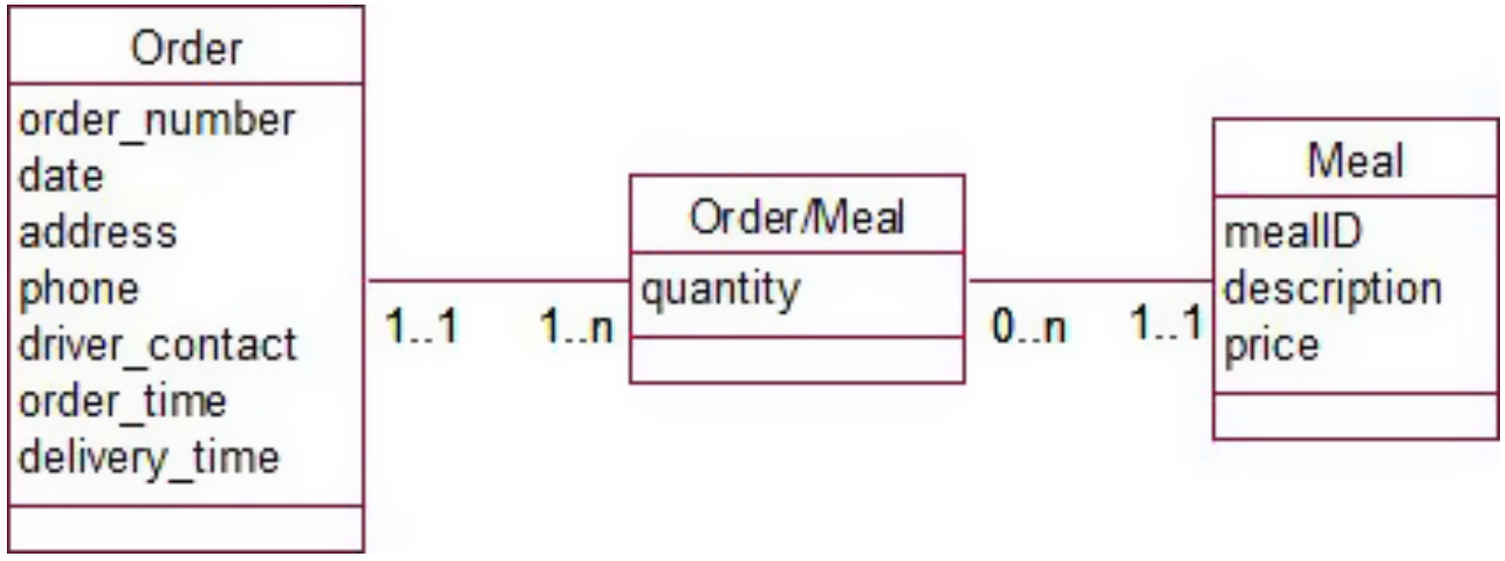
Thank you!

Meal Delivery Example

- The initial data model had a Many-Many relationship between types of meal and orders.



- If a family orders three chicken vindaloos, one hamburger, and one pork fried rice? Where do we put these quantities? The quantity cannot be an attribute in the **Order** class nor in the **Meal** class .



- It can be difficult to come up with a meaningful name for the intermediate class. We could maybe have called the class **Orderline**,
- We can also use this new intermediate class to solve one of the problem of coping with the price of a meal changing over time.
- This will be the price charged for a particular meal on a particular order and will not change when the current price changes in the **Meal** class. This way we have a complete history of the prices for each meal on each order. A price attribute in this intermediate class can allow us to keep historical data and also to deal with “unusual” situations such as specials or discounts..

Are there any data we need to store about a particular meal type on a particular order?

Yes, the quantity of that meal type ordered and the price being charged for that meal type on the order.

Exercise 5-2

The figure shows a possible data model for cocktail recipes. The Many-Many relationship uses can be navigated in either direction. To find out the ingredients in a Manhattan or to discover the possible uses for that bottle of Vermouth. What is missing?

Exercise 5-3

Part of the data model about guests at a hostel is shown in the figure. How could the model be amended to keep historical information about room occupancy?