Reading: Elmasri & Navathe, Fundamentals of Database Systems, Chapter 3

THREE

THREE

TWO

key

top-down

bottom-up

inside-out

!

slim

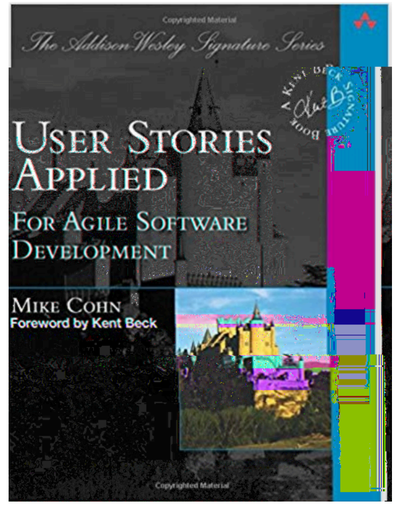
in

a collaborative manner

Agile

- *(we will have a go at evolutionary, agile (scrum based) ER modelling in Assignment 2, though this approach will not be mandatory - more in Tutorial 3)*

Each of these is ONE SINGLE STORY



- *As a (role) I want (something) so that (benefit).*

  - *As a… [which type of user has this need?]*
  - *I need/want/expect to… [what does the user want to do?]*
  - *So that… [why does the user want to do this?]*
  - *When… [what triggers the user's need?]*
  - *Because… [is the user constrained by any circumstances?]*

entity

attribute

relationship

We wish to create a system for a company that runs training courses. For each course participant, identified by a code, we want to store the national insurance number, surname, age, sex, place of birth, employer's name, address and telephone number, previous employers (and period employed), the course attended and the final assessment of each course. We need also to represent the seminars that each participant is attending at present and, for each day, the places and times the classes are held. Each course has a code and a title and any course can be given any number of times. Each time a course is given, we call it an "edition" of the course. For each edition, we represent the start and end dates and the number of participants. If a trainee is a self employed professional, we need to know his or her area of expertise, and, if appropriate, his or her title. For somebody who works for a company we store the level and position held. For each instructor we will show surname, age, place of birth, the edition the course is taught, those taught in the past and the courses the tutor is qualified to teach. All the instructor's telephone numbers are also stored. An instructor can be permanently employed or freelance.

1. Trainees attend Courses

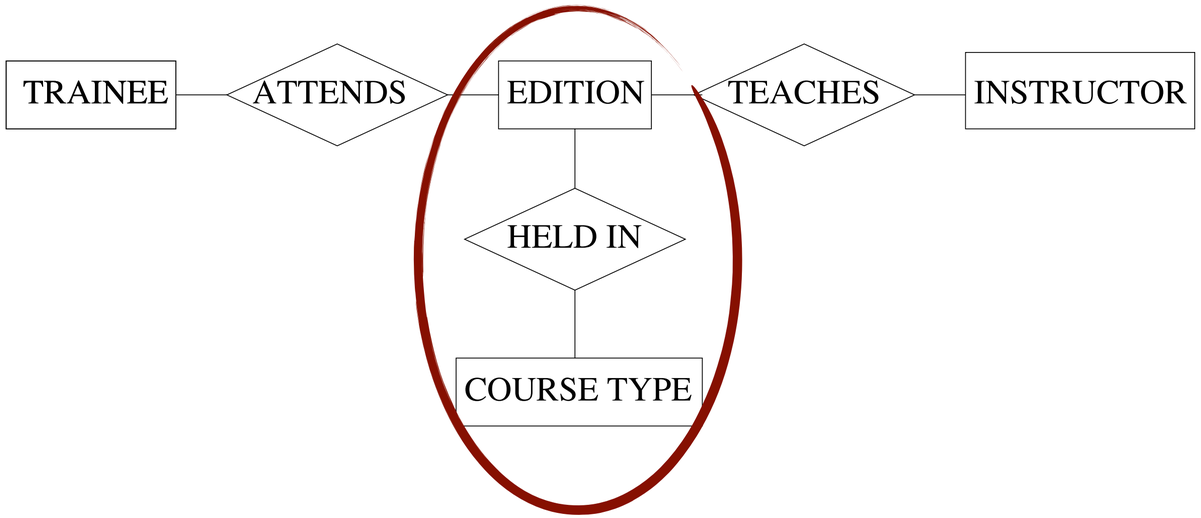
2. Instructors teach Courses

3. Courses are held in “editions”

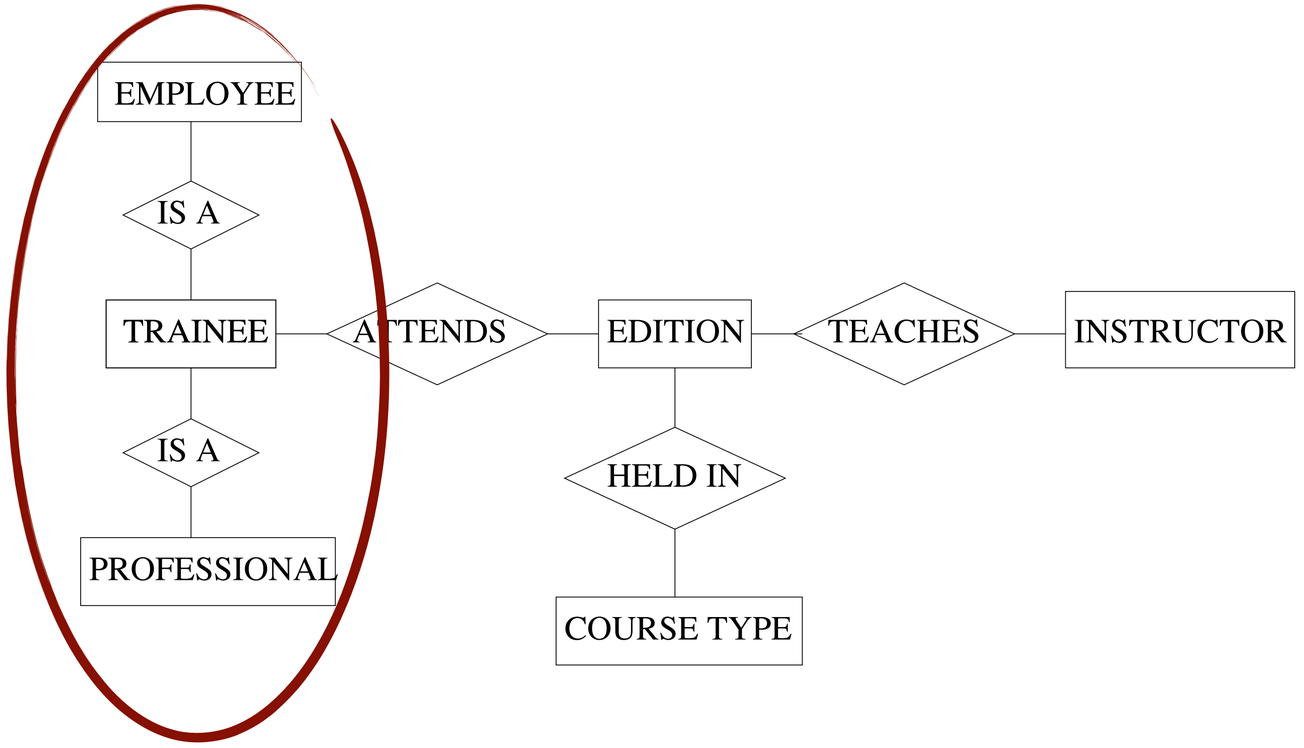
4. Trainees can be self employed professionals or work for a company

5. We distinguish between current and past editions

- *Story 3: “Courses can be held in Editions”*
  - *from 1 entity to 2 entities+relationship*
    - *identify cases in which an entity describes two different concepts logically linked to each other:*



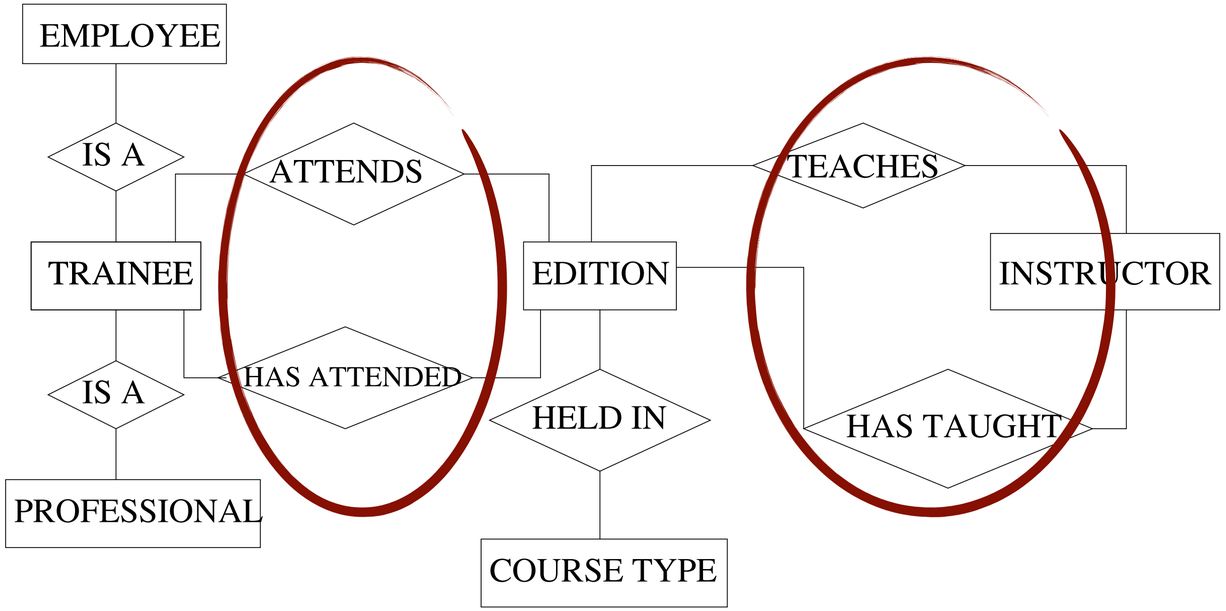
- *Story 4: Trainees can be self employed professionals or work for a company*
- *from 1 entity to 1 entity+N entities+N relationships*
  - *identify cases in which an entity is made up of distinct sub-entities:*



- *Story 5: we distinguish between current and past edition of a course*

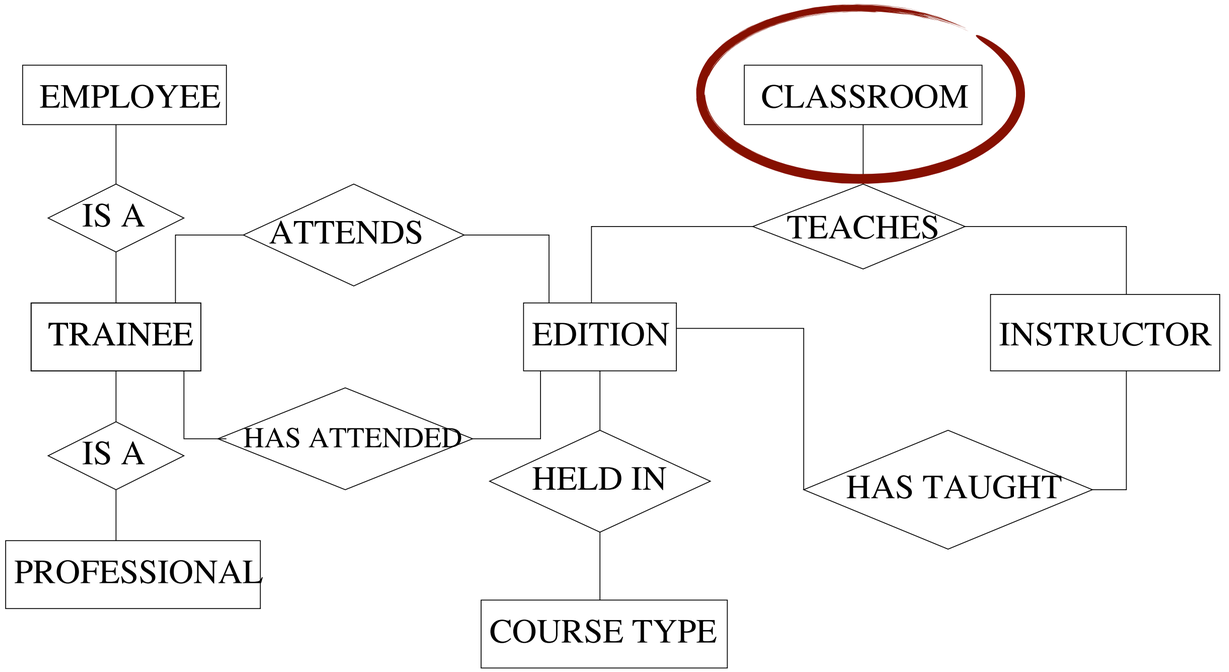
- *from 1 relationship to multiple relationships*

  - *identify cases in which a relationship describes two or more different concepts linking the same entities:*

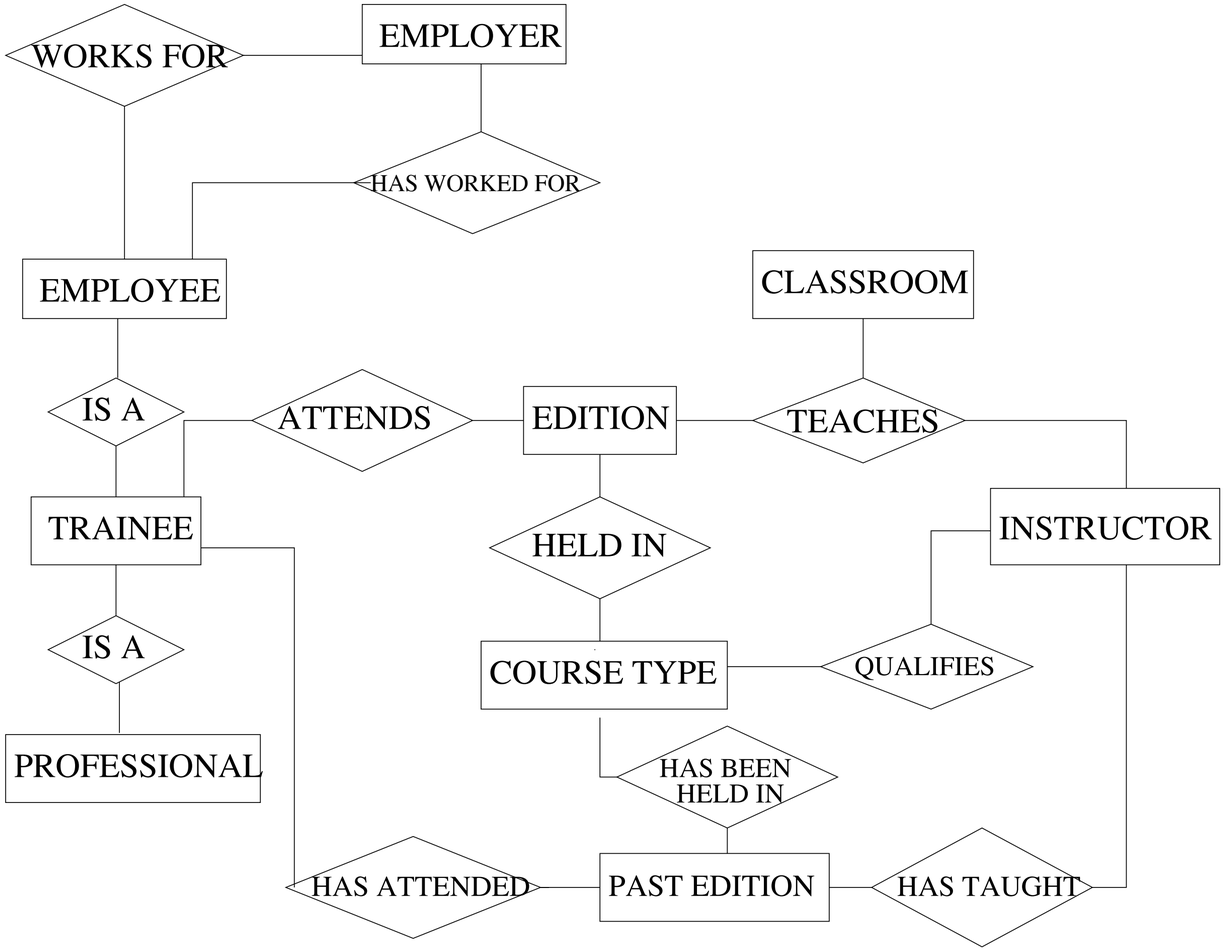


6. Courses are held in classrooms
7. Instructors only teach Courses for which they are qualified
8. We archive past editions of courses keeping summary data
9. We maintain data of trainees' employers

- *Story 6: Courses are held in classrooms*
  - *from 1 relationship to 1 entity + relationships*
    - *identify cases in which a relationship describes a concept having an autonomous existence:*



- *Story 7, 8 and 9: spot the differences!*

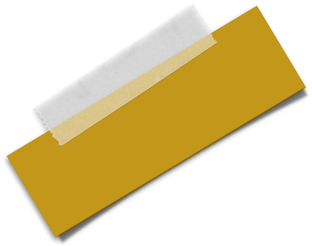


dodgy

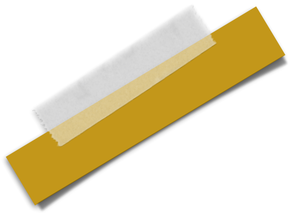
weak entity type

identifying owner

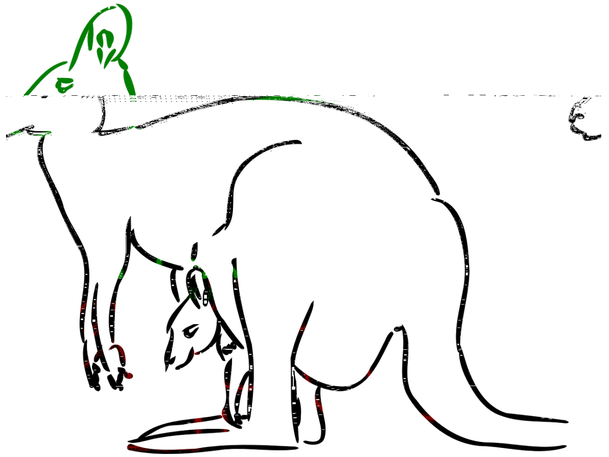
identifying relationship

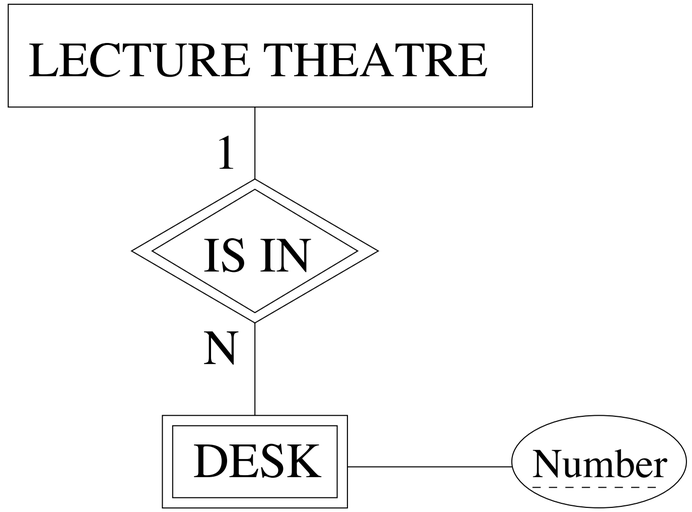


must

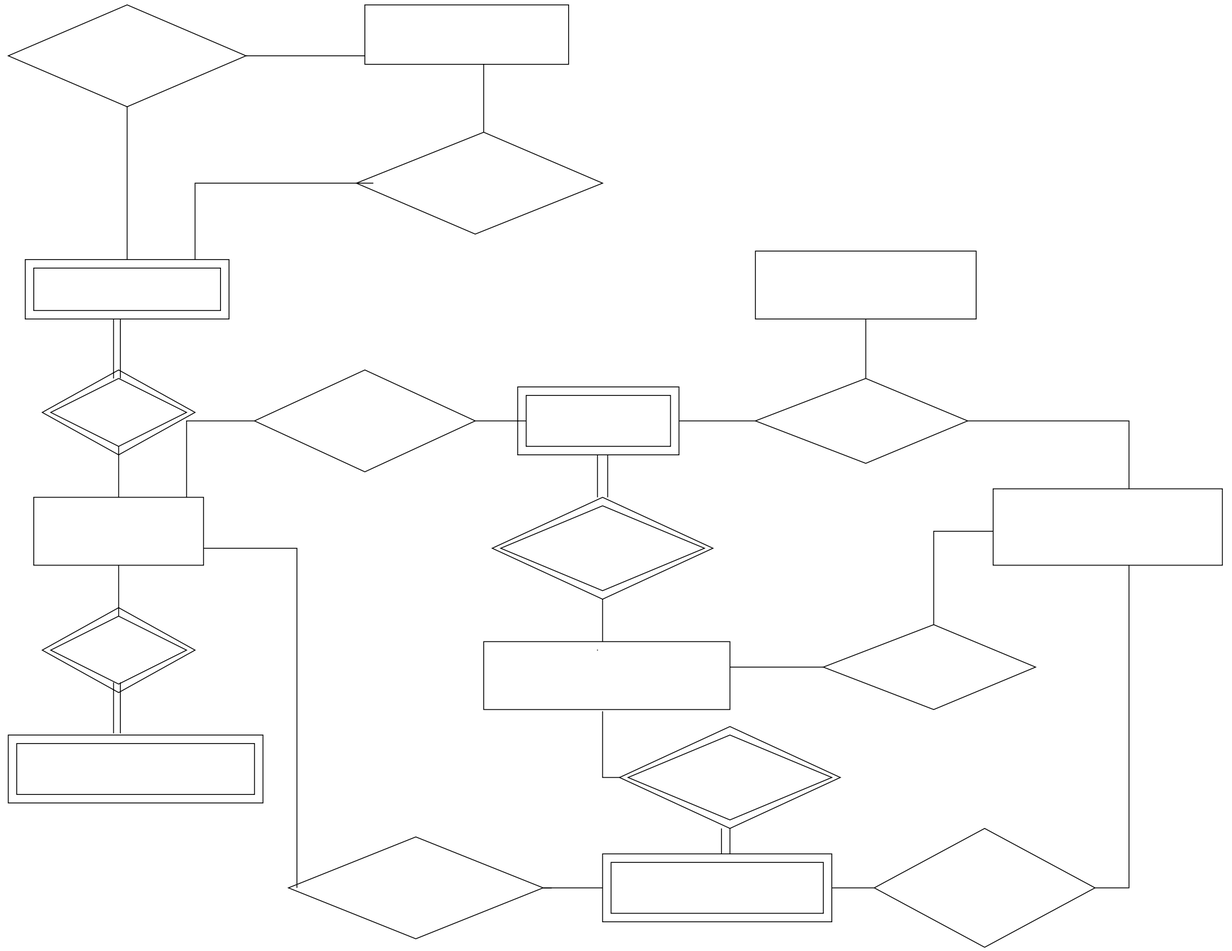


must

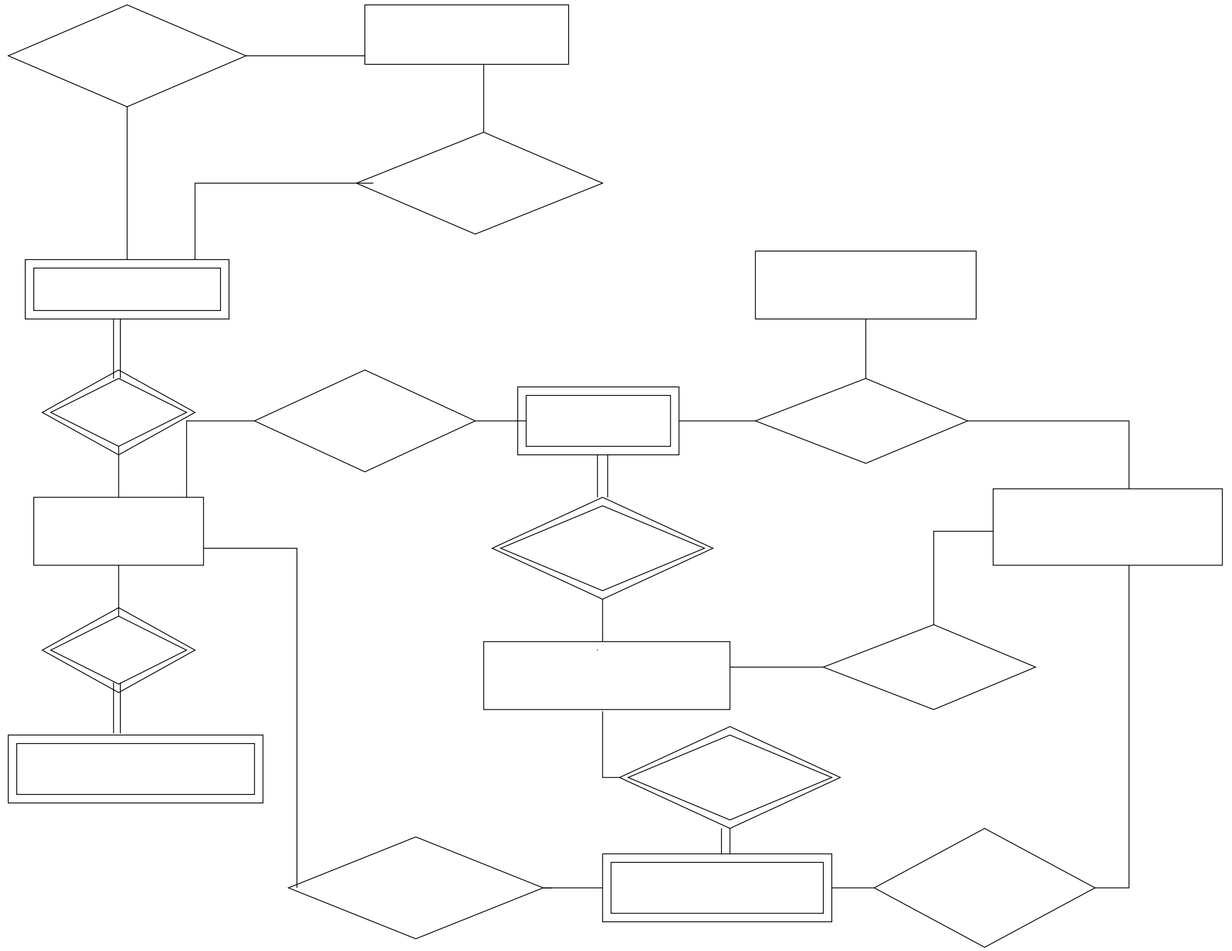



LECTURE THEATRE
1
IS IN
N
DESK
Number

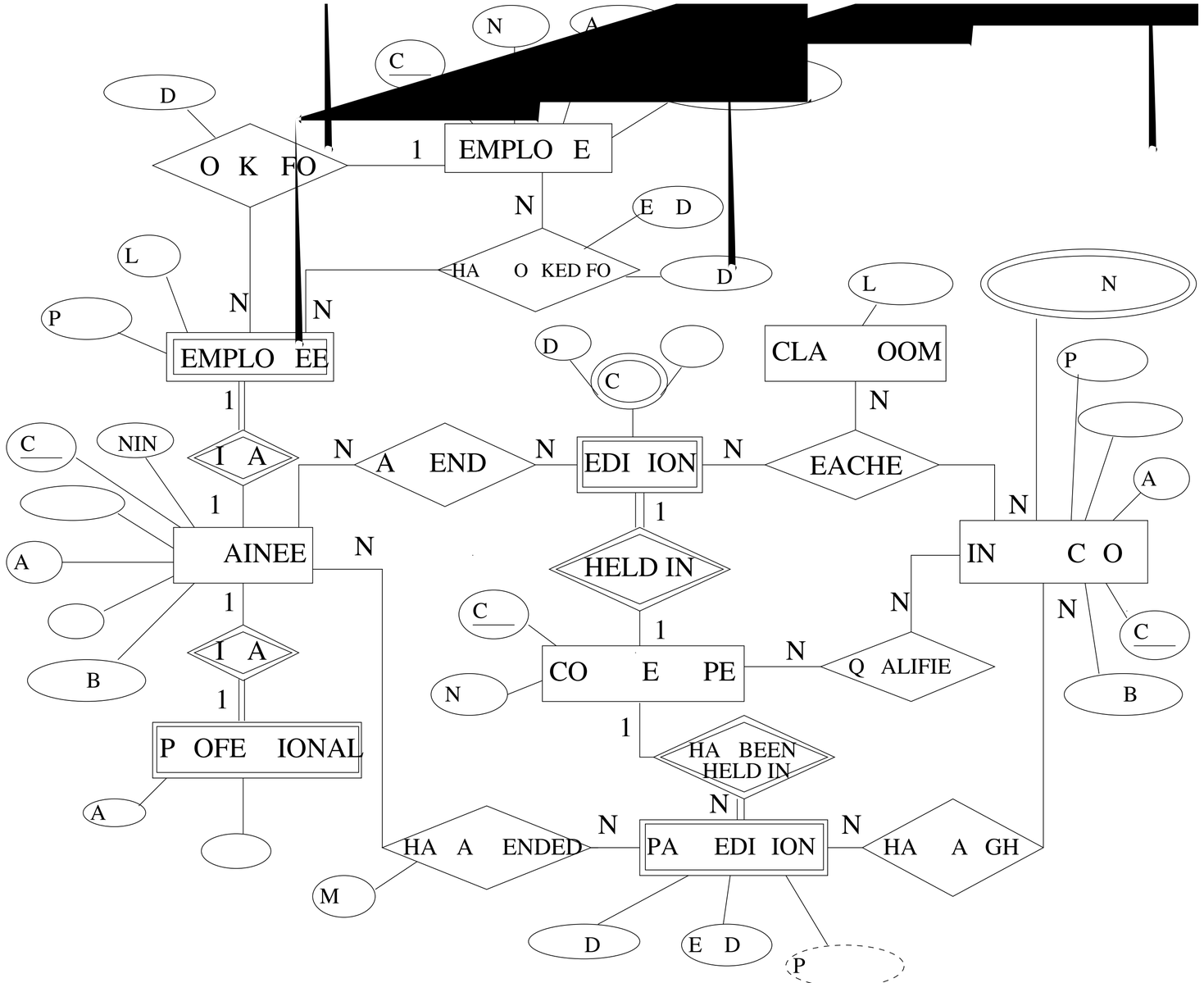
- *establish cardinalities and participation constraints for relationship, and **strong/weak** entity types*



- *and now, finish off with the attributes (you could add attributes at each iteration - or you could add just the main attributes, then refine, …)*



- *Final ER schema*



one

one

one

many

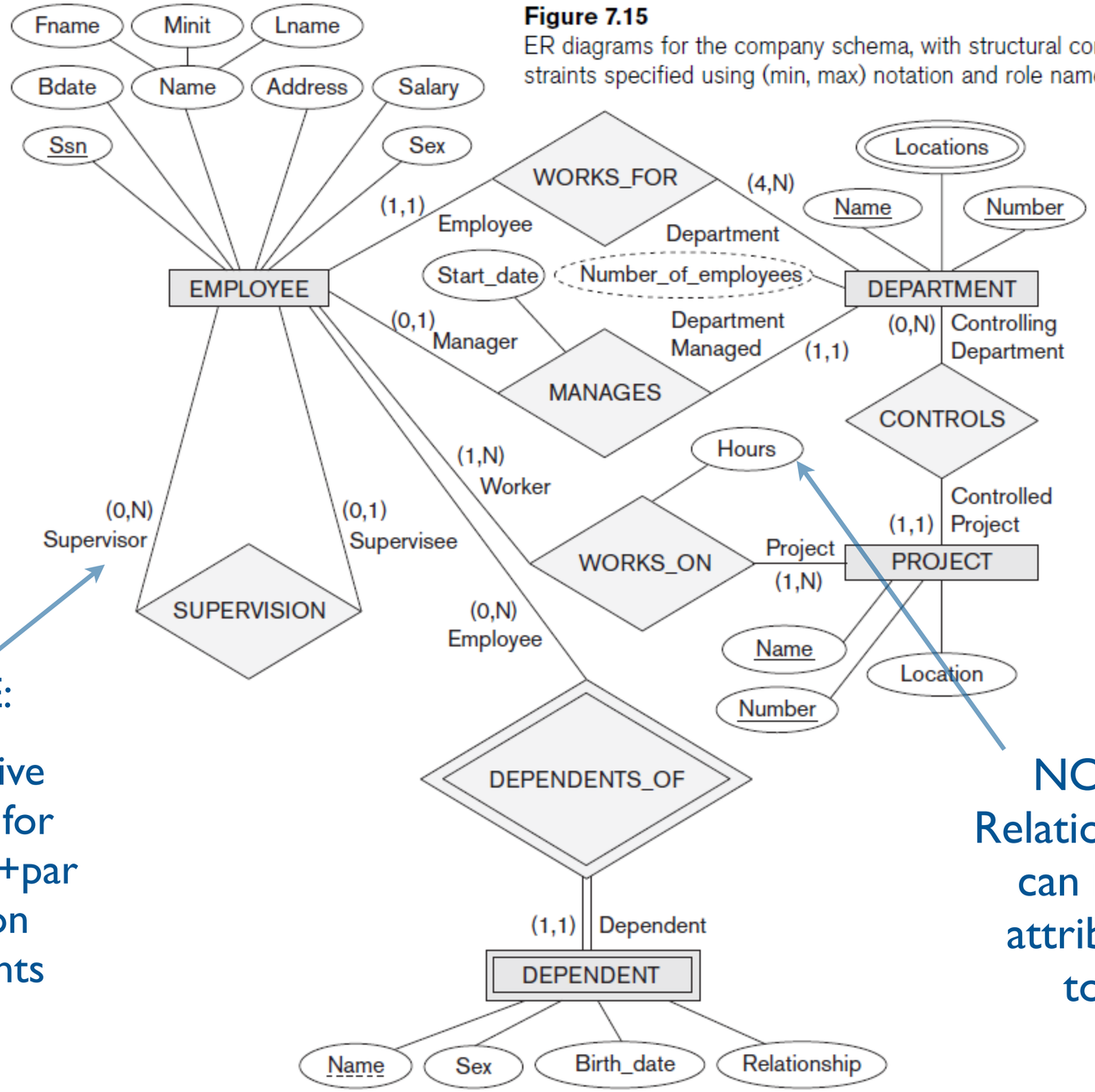


Figure 7.15
ER diagrams for the company schema, with structural constraints specified using (min, max) notation and role names.

NOTE:

Alternative notation for cardinality+participation constraints

NOTE: Relationships can have attributes too