Fundamentals/ICY: Databases 2013/14

WEEK 7 – Monday

John Barnden
Professor of Artificial Intelligence

School of Computer Science
University of Birmingham, UK

Reminder of Friday

Entity Supertypes and Subtypes

- Generalization (or: specialization) hierarchy
  - A group of relationships each of which is between a higher-level "supertype" entity (e.g. EMPLOYEE) and a lower-level "subtype" entity (e.g., PROFESSOR)
- Supertype
  - Contains attributes shared by all its subtypes
- Subtype
  - Contains special attributes: ones that not all sister subtypes have.
- Primary key of a subtype = that of the supertype (normally)

Disjoint (or: Non-Overlapping) Subtypes

- Each entity in the supertype can appear in **at most** one of the subtypes
- Overlapping = a given entity can be in more than one subtype.

Exhaustive Subtypes

- Each entity in the supertype must appear in **at least** one of the subtypes
- Other terminology:
  - exhaustiveness = total completeness (!!)
    = mandatoriness [of being in some subtype]
  - non-exhaustiveness = partial completeness (!!)
    = optionality [of being in som subtype]

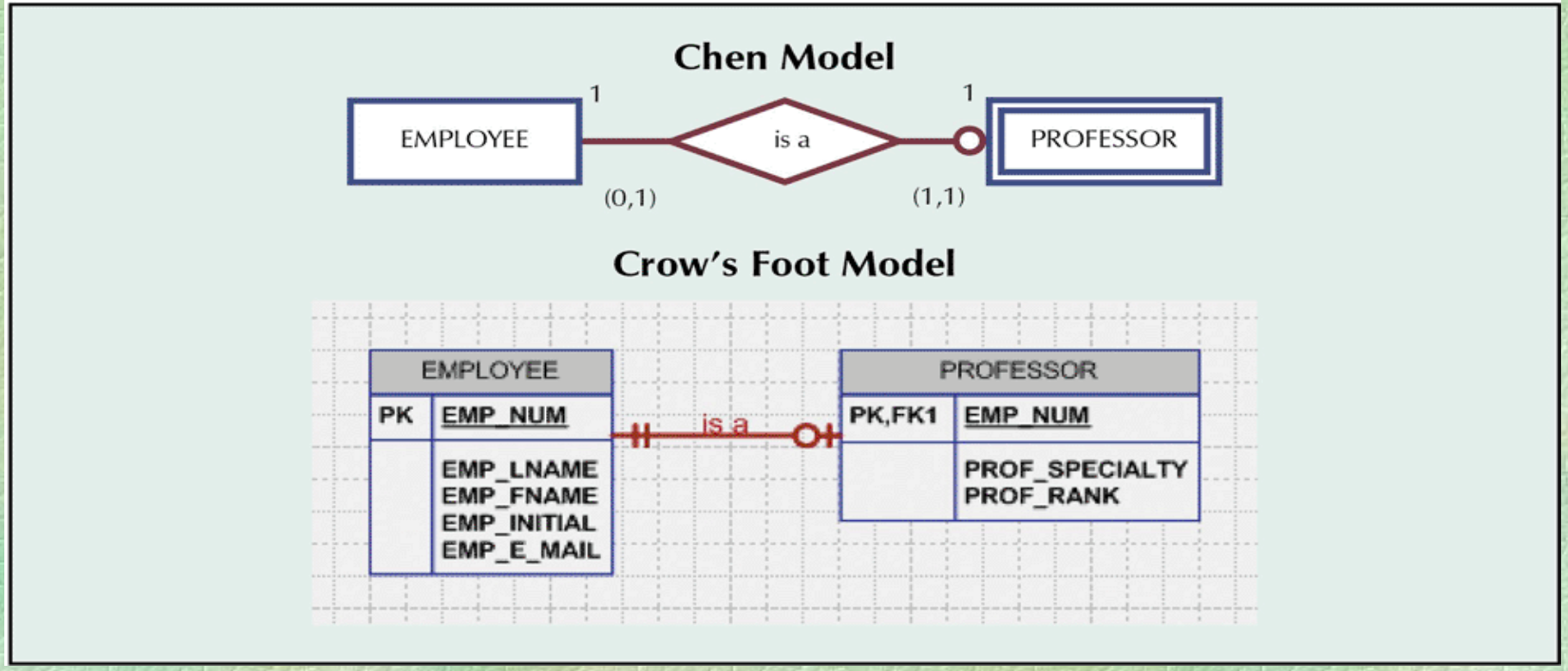
New

Why Consider Supertypes and Subtypes?

- We would not just want to have a table for the supertype and none for the subtypes, because of the resulting poor table structure.
- And we would not ***just*** want to use separate entity types for the subtypes, because of
  - Attributes (etc.) in common and relationships in common
  - Need an extra entity type anyway when the subtypes are not exhaustive.

Actual Realization of Subtyping

FIGURE 4.37 A SUPERTYPE/SUBTYPE RELATIONSHIP IN AN ERD



A supertype has a ***1:1 relationship*** with each subtype.
Mandatory in the ***sub-to-super*** direction.
Optional in the ***super-to-sub*** direction, ***even in*** exhaustive case.

The EMPLOYEE/PILOT Supertype/Subtype Relationship

FIGURE 4.29 THE EMPLOYEE/PILOT SUPERTYPE/SUBTYPE RELATIONSHIP

Table name: EMPLOYEE (the supertype) **Database name: Ch04_AirCo**

EMP_NUM	EMP_LNAME	EMP_HIRE_DATE
100	Kolmycz	15-Mar-88
101	Lewis	25-Apr-89
102	Vandam	20-Dec-93
103	Jones	28-Aug-03
104	Lange	20-Oct-97
105	Williams	08-Nov-97
106	Duzak	05-Jan-04
107	Diante	02-Jul-97
108	Wiesenbach	18-Nov-95
109	Travis	14-Apr-01
110	Genkazi	01-Dec-03

Table name: PILOT (the subtype)

EMP_NUM	PIL_LICENSE	PIL_RATINGS	PIL_MED_TYPE
101	ATP	SEL/MEL/Instr/CFII	1
104	ATP	SEL/MEL/Instr	1
105	COM	SEL/MEL/Instr/CFI	2
106	COM	SEL/MEL/Instr	2
109	COM	SEL/MEL/SES/Instr/CFII	1

But Special ERD Notation is Desirable

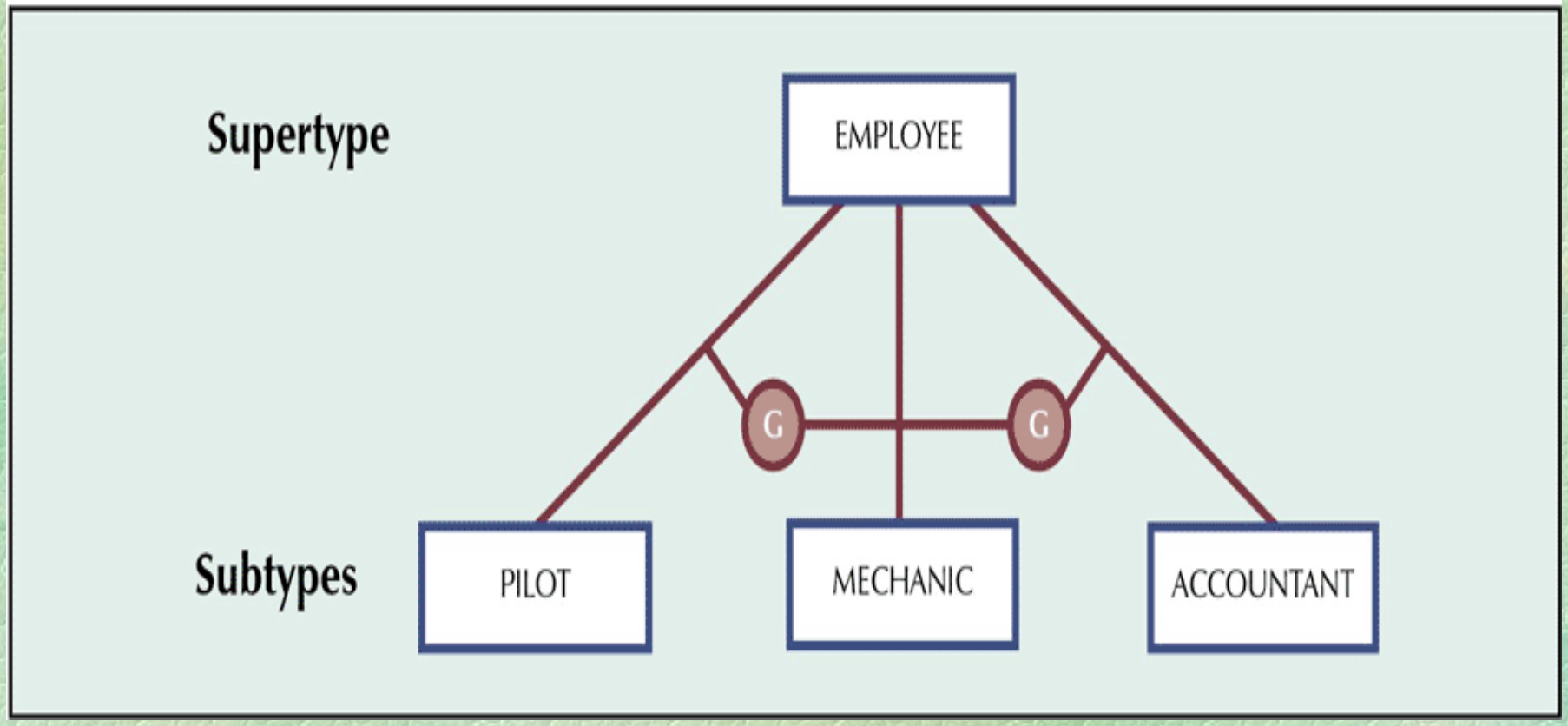
- To clearly bring out the structure in a conceptual ERD.
- Can refine later to get rid of the special notation and instead use the basic ERD facilities as in the "Actual Realization" slide above.
- Those basic facilities don't clearly show that there is a specialization/generalization at all, and also don't show (non-)disjointness or (non-)exhaustiveness.

Special Notation in following slides is from a previous textbook.

See other, *better*, notations in current textbook and in Additional Notes

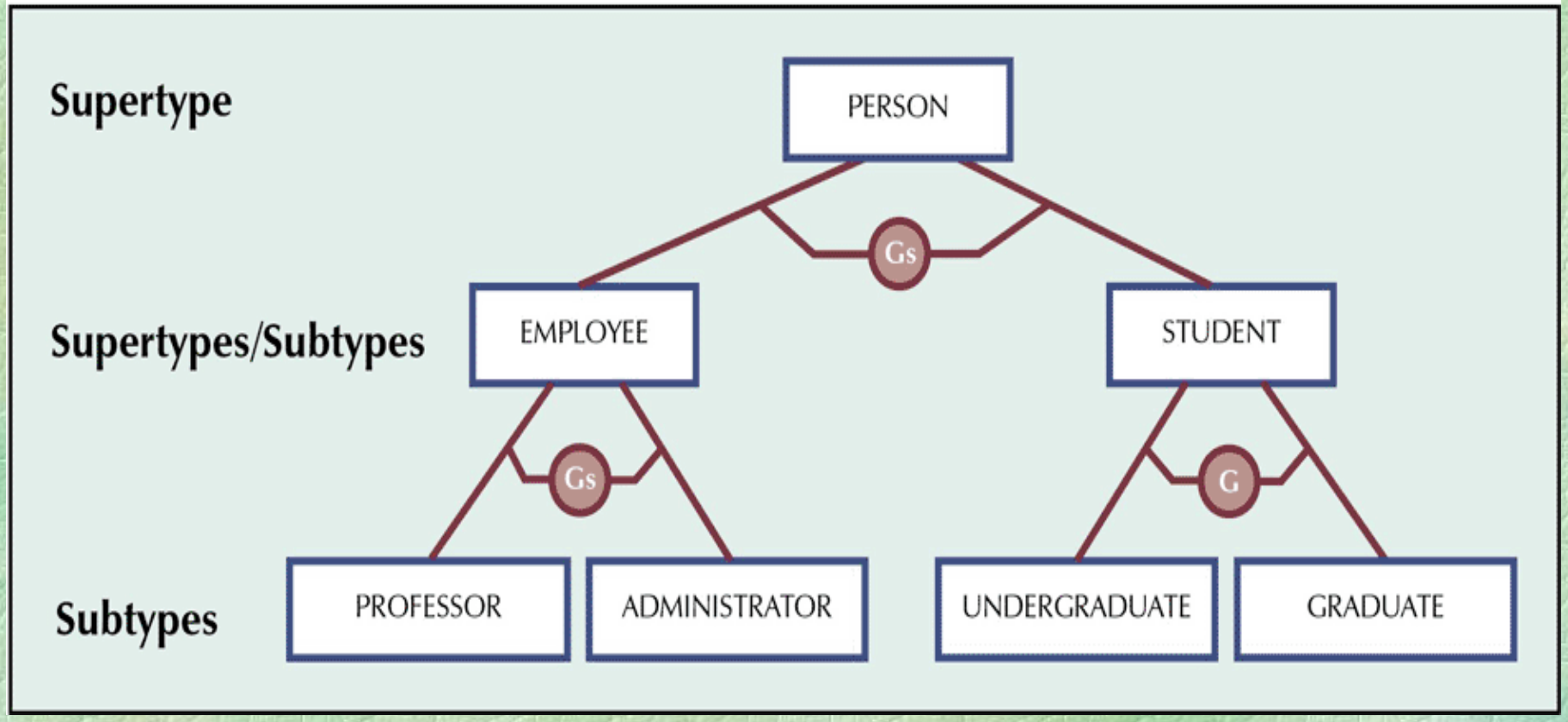
Disjoint Subtypes

FIGURE 4.28 A GENERALIZATION HIERARCHY



A Generalization Hierarchy with Overlapping Subtypes

FIGURE 4.30 A GENERALIZATION HIERARCHY WITH OVERLAPPING SUBTYPES



G = disjoint, GS = overlapping.

NB: No representation of (non-)exhaustiveness.

More-Than-Binary Relationships in ERMs and ERDs

Relationship Degree

- **Binary** relationship [***my*** definition]
  - **Two** entities ("entity occurrences") are associated by each instance of the relationship, as in all previous examples in lecture slides.
- **Ternary** relationship [***my*** definition]
  - **Three** entities are associated by each relationship instance.
- Etc.
- NB: the entities associated with each other need not be of different ***types.***
  - Indeed, could have an entity (occurrence) associated with itself: e.g. an "employs" relationship where someone can employ him/herself. Still binary.

Relationship Degree: Terminology Problems

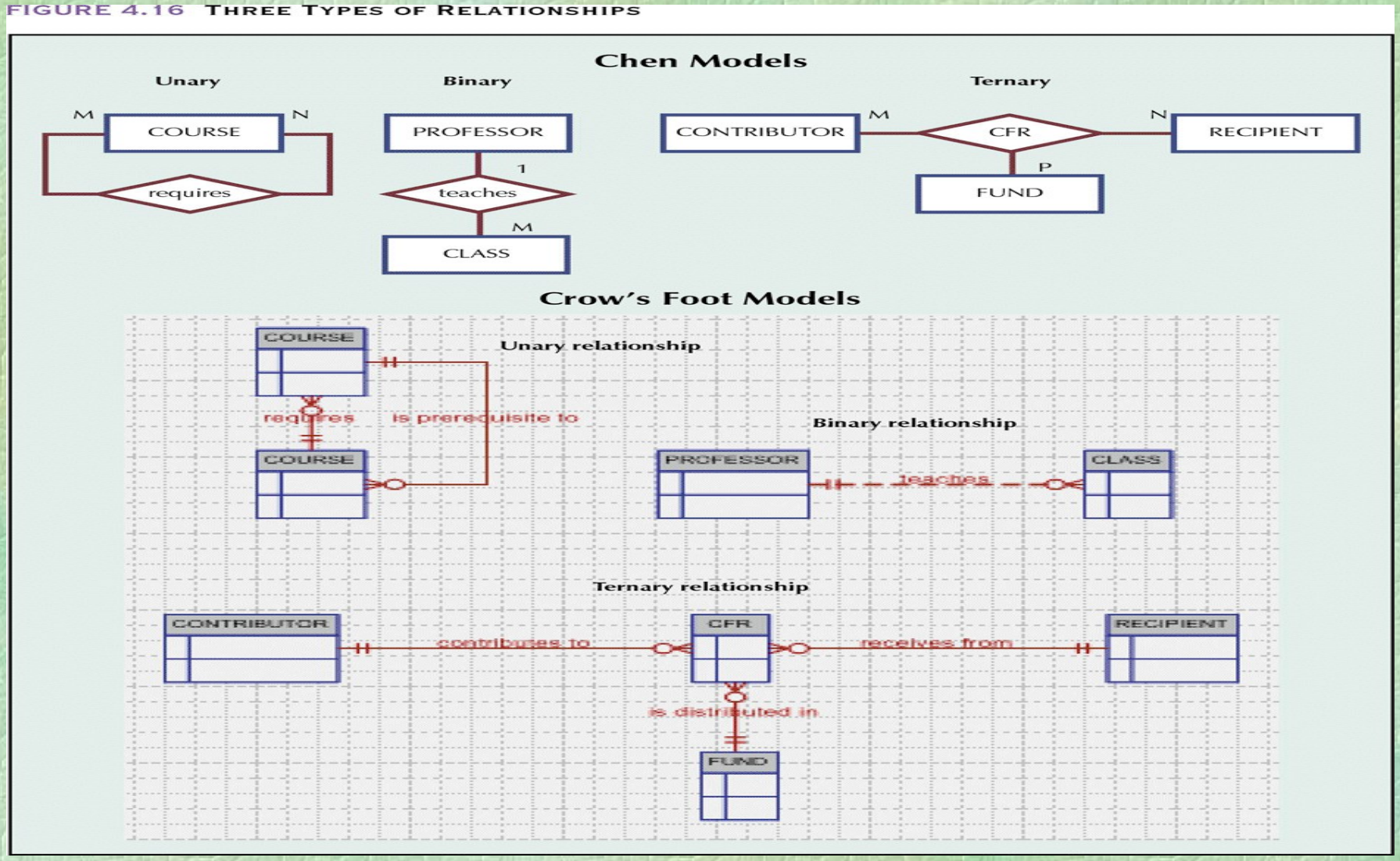
- The standard terminology & definitions relating to relationship degree (see the textbook) are *mathematically anomalous*.
- I believe the degree of a relationship should be the number of ***entities*** ("entity ***occurrences***") that are associated by each instance of the relationship, no matter how many entity types are involved.
- But the definitions used in the books count the number of entity ***types*** ("entities" in the abbreviated language used), departing from normal mathematical practice.

Terminology Problems contd.

- A **"unary"** relationship is standardly defined as being a relationship where the entity occurrences are all within the same entity type ("entity").
  - E.g., a "manages" relationship between employees.
- A better name is "recursive" (also used in the books).
  - Above example is binary recursive, under **my** definition.
- The books define a **binary** relationship as one relating ***two different types***, even when there are more than two entity occurrences per relationship instance.
- Similarly **ternary** and **three different** types.

Different Degrees (using Book Terminology)

FIGURE 4.16 THREE TYPES OF RELATIONSHIPS



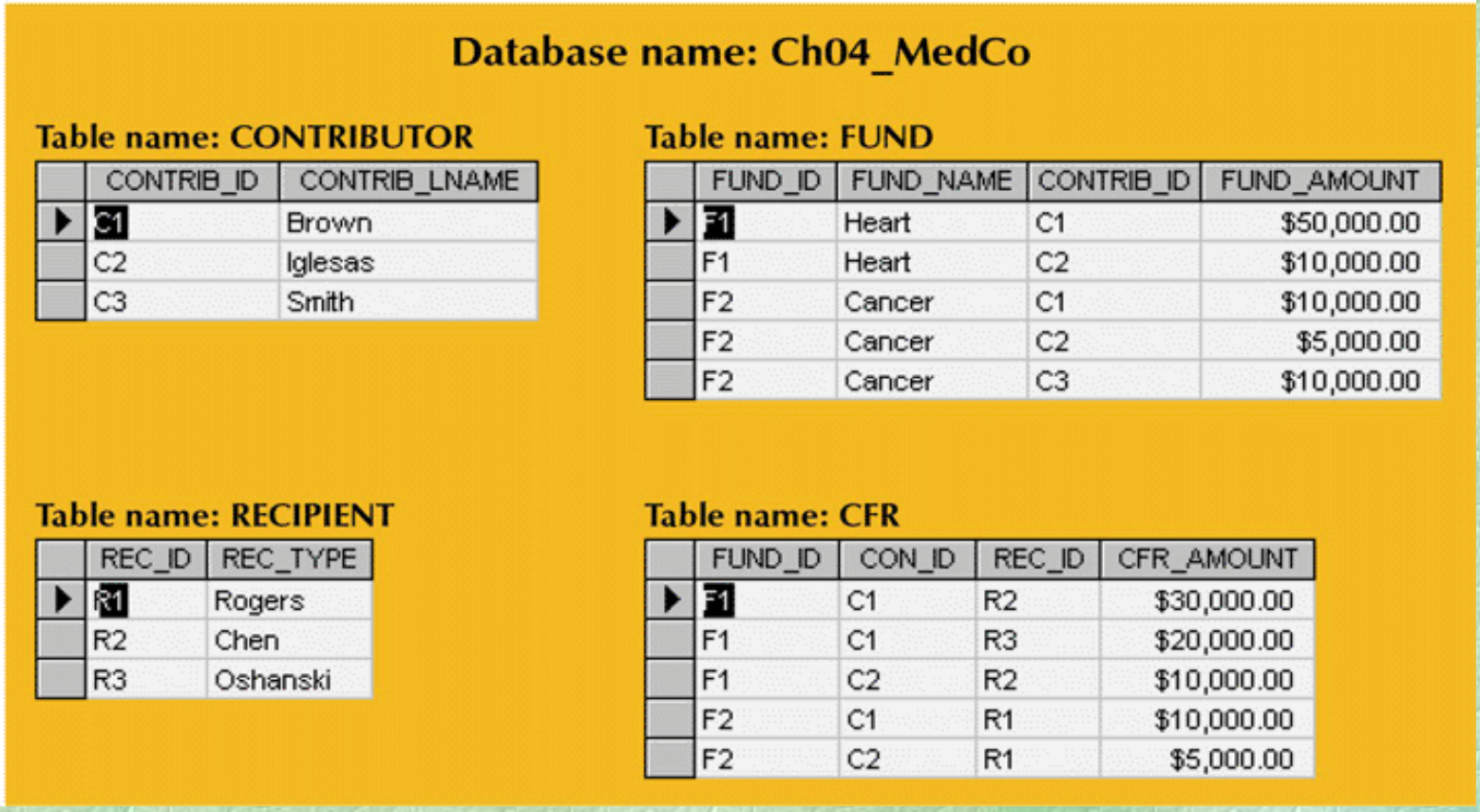
The "unary" case is badly named. It's really a type of binary.
The word "recursive" [later] is better.

Caution about Next Slide

- There is an error in the following slide.
- The FUND table should just be descriptions of different funds, such as Heart fund, a Cancer fund, and so on. FUND_ID should be the PK.
- It shouldn't have the CONTRIB_ID and FUND_AMOUNT columns.

Tables for a Ternary Relationship

FIGURE 4.17 THE IMPLEMENTATION OF A TERNARY RELATIONSHIP



Database name: Ch04_MedCo

Table name: CONTRIBUTOR

CONTRIB_ID	CONTRIB_LNAME
C1	Brown
C2	Iglesas
C3	Smith

Table name: FUND

FUND_ID	FUND_NAME	CONTRIB_ID	FUND_AMOUNT
F1	Heart	C1	$50,000.00
F1	Heart	C2	$10,000.00
F2	Cancer	C1	$10,000.00
F2	Cancer	C2	$5,000.00
F2	Cancer	C3	$10,000.00

Table name: RECIPIENT

REC_ID	REC_TYPE
R1	Rogers
R2	Chen
R3	Oshanski

Table name: CFR

FUND_ID	CON_ID	REC_ID	CFR_AMOUNT
F1	C1	R2	$30,000.00
F1	C1	R3	$20,000.00
F1	C2	R2	$10,000.00
F2	C1	R1	$10,000.00
F2	C2	R1	$5,000.00

CFR is just like the bridging entity types you've seen before, but has 3 links to other types instead of 2

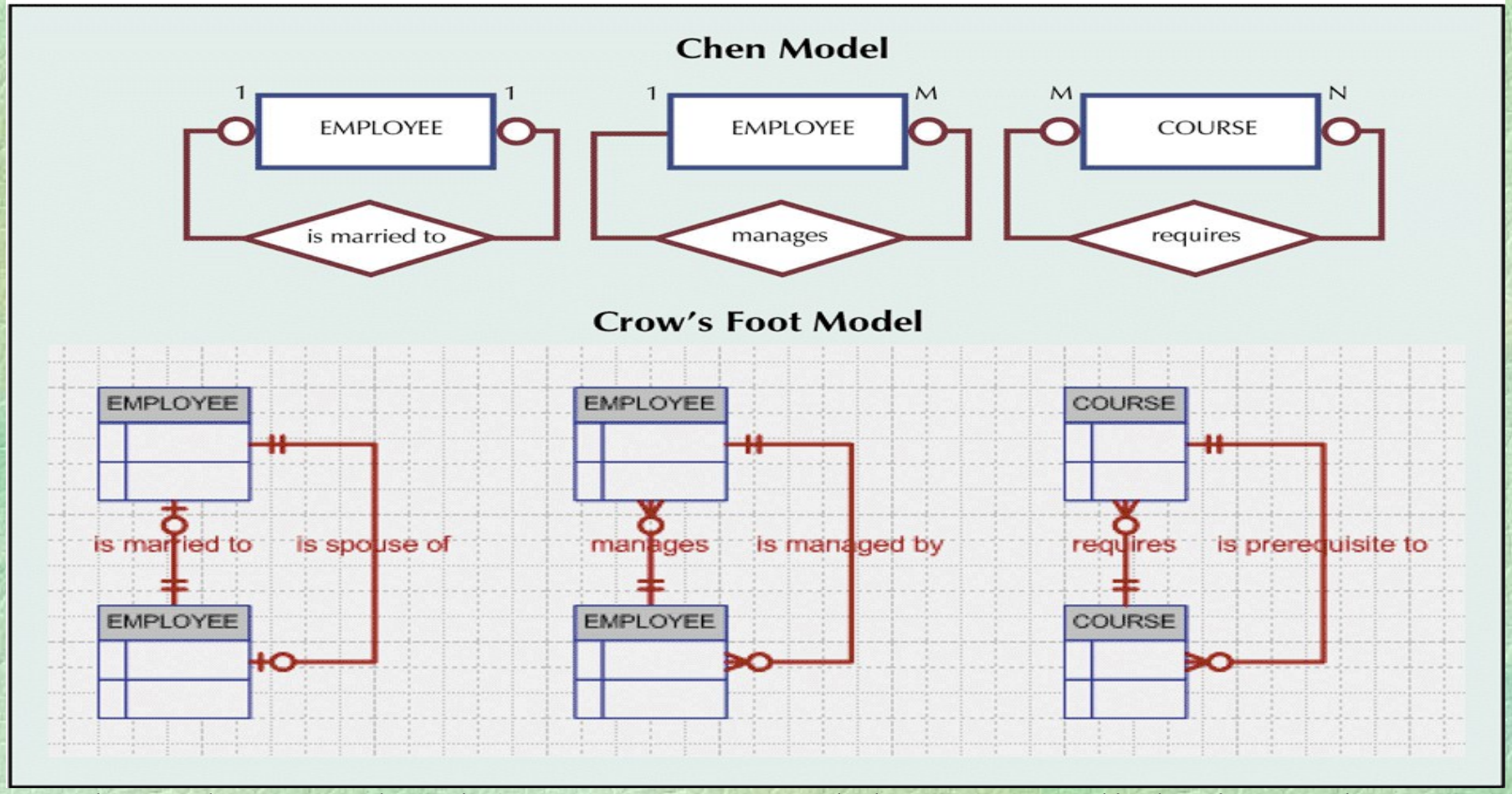
Recursive Relationships in ERMs and ERDs

Recursive Relationships

- A ***recursive*** relationship links entities of the same type.
  - *E.g.:* marriage, management, parthood, …
- Can have partial recursion: just ***some*** of the entity types involved in a relationship (that is ternary or of higher degree in my sense) could be the same.

An ER Representation of Recursive Relationships

FIGURE 4.18 AN ER REPRESENTATION OF RECURSIVE RELATIONSHIPS



I don't know why the Crow's Ft model uses two links in each case

Recursive Relationships: Symmetry

- A relationship R between entity types E,F (possibly the same) is ***symmetric*** iff:

  if ***eRf*** then ***fRe*** (i.e., IF R relates entity ***e*** of type E to entity ***f*** of type F, then it must ALSO relate ***f*** to ***e***.)

  E.g.: marriage, being-sibling-of.

- ***Recursive relationships cause major redundancy problems when ALSO symmetric.***

- ***Symmetry only makes sense in the 1:1 and M:N cases.***

- ((Can generalize the points to partly-recursive cases.))

(necessarily **non**-symmetric) **1:M** recursive: "EMPLOYEE Manages EMPLOYEE"

FIGURE 4.23 IMPLEMENTATION OF THE 1:M "EMPLOYEE MANAGES EMPLOYEE" RECURSIVE RELATIONSHIP

Table name: EMPLOYEE_V2 Database name: Ch04_PartCo

EMP_CODE	EMP_LNAME	EMP_MANAGER
101	Waddell	102
102	Orincona	
103	Jones	102
104	Reballoh	102
105	Robertson	102
106	Deltona	102

Just a standard 1:M implementation except linking a table to itself. No redundancy problem.

non-symmetric M:N recursive: "PART Contains PART"

FIGURE 4.21 IMPLEMENTATION OF THE M:N RECURSIVE "PART CONTAINS PART" RELATIONSHIP

Table name: COMPONENT **Database name: Ch04_PartCo**

COMP_CODE	PART_CODE	COMP_PARTS_NEEDED
C-130	AA21-6	4
C-130	AB-121	2
C-130	E129	1
C-131A2	E129	1
C-130	X10	4
C-131A2	X10	1
C-130	X34AW	2
C-131A2	X34AW	2

Table name: PART

PART_CODE	PART_DESCRIPTION	PART_IN_STOCK
AA21-6	2.5 cm. washer, 1.0 mm. rim	432
AB-121	Cotter pin, copper	1,034
C-130	Rotor assembly	36
E129	2.5 cm. steel shank	128
X10	10.25 cm. rotor blade	345
X34AW	2.5 cm. hex nut	879

The COMPONENT entity type is just a bridging type, linking PART to itself. NB: its first two columns both refer to PART's PK but must be differently named. No redundancy problem.

symmetric (1:1) recursive relationship: "EMPLOYEE Married to EMPLOYEE"

Suppose you tried the following:

FIGURE 4.19 THE 1:1 RECURSIVE RELATIONSHIP "EMPLOYEE IS MARRIED TO EMPLOYEE"

Table name: EMPLOYEE_V1 Database name: Ch04_PartCo

EMP_NUM	EMP_LNAME	EMP_FNAME	EMP_SPOUSE
345	Ramirez	James	347
346	Jones	Anne	349
347	Ramirez	Louise	345
348	Delaney	Robert	
349	Shapiro	Anton	346

Redundancy problem!!