

Fundamentals/ICY: Databases 2013/14

WEEK 6 - Friday

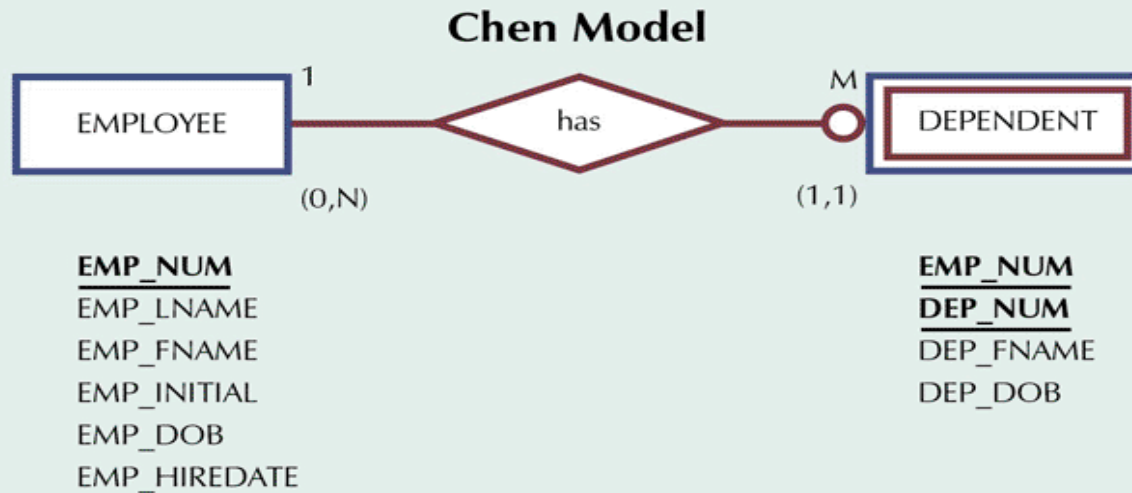
John Barnden
Professor of Artificial Intelligence

School of Computer Science
University of Birmingham, UK

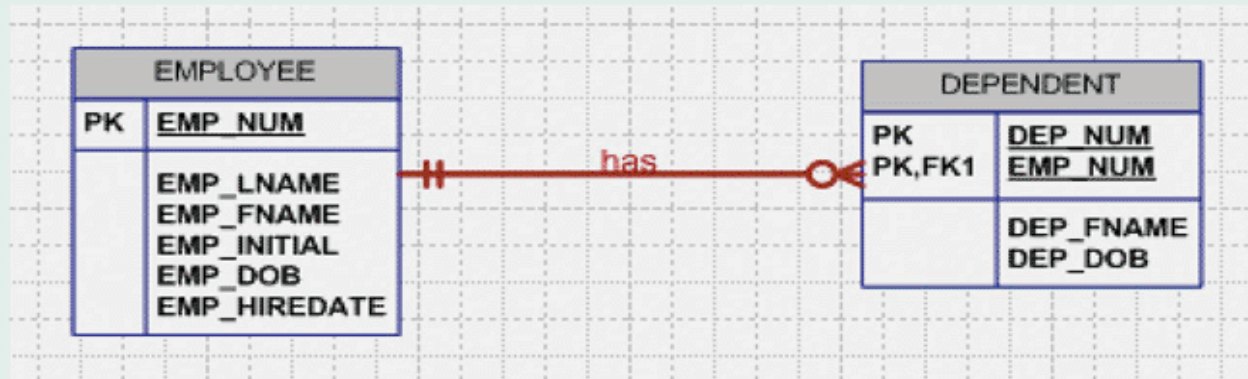
Reminder of Monday

A Weak Entity in an ERD

FIGURE 4.14 A WEAK ENTITY IN AN ERD



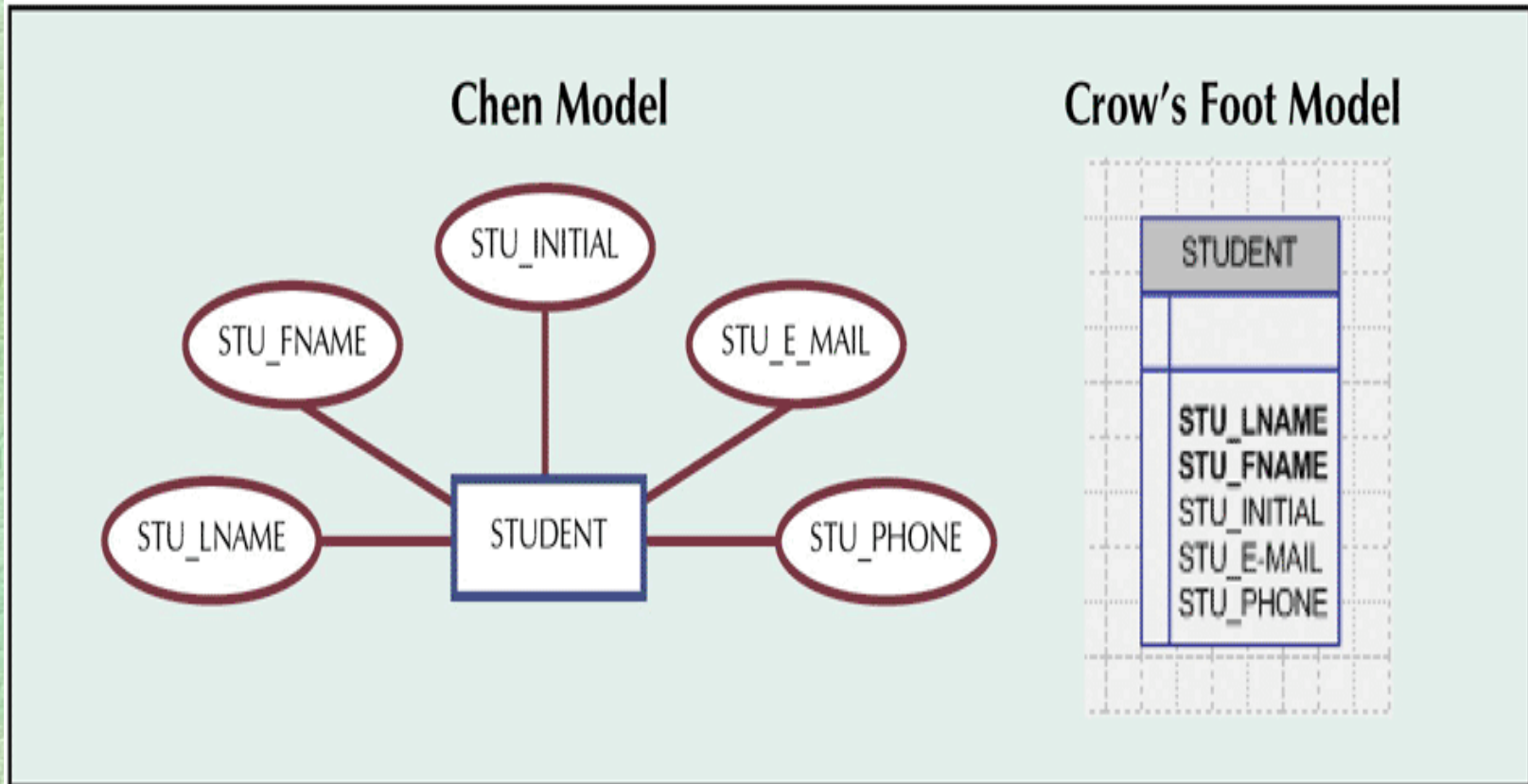
Crow's Foot Model



New

The Attributes of the STUDENT Entity

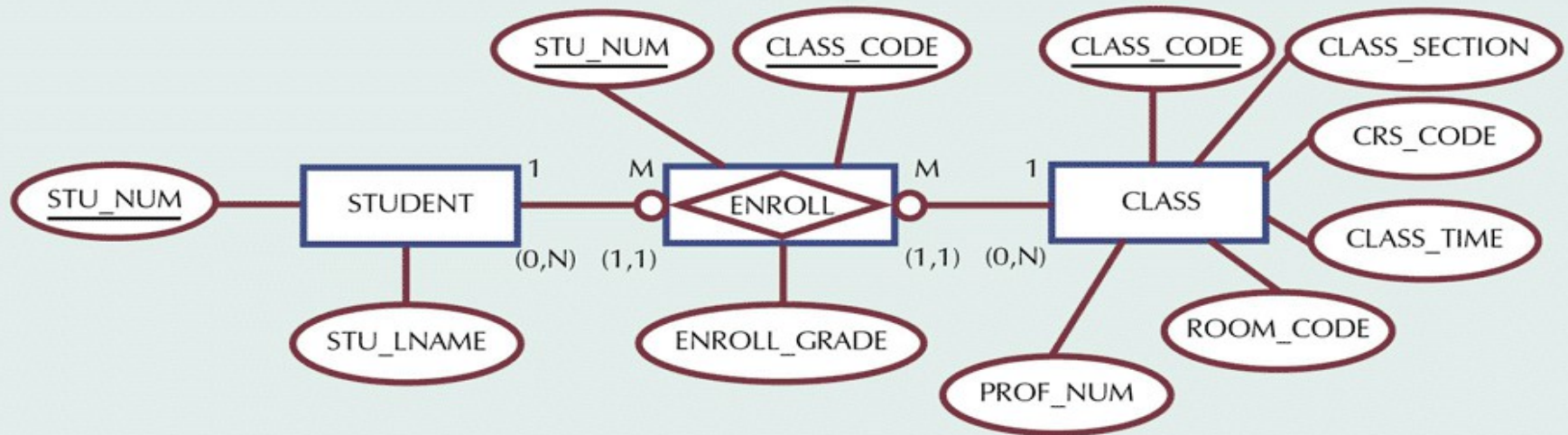
FIGURE 4.1 THE ATTRIBUTES OF THE STUDENT ENTITY



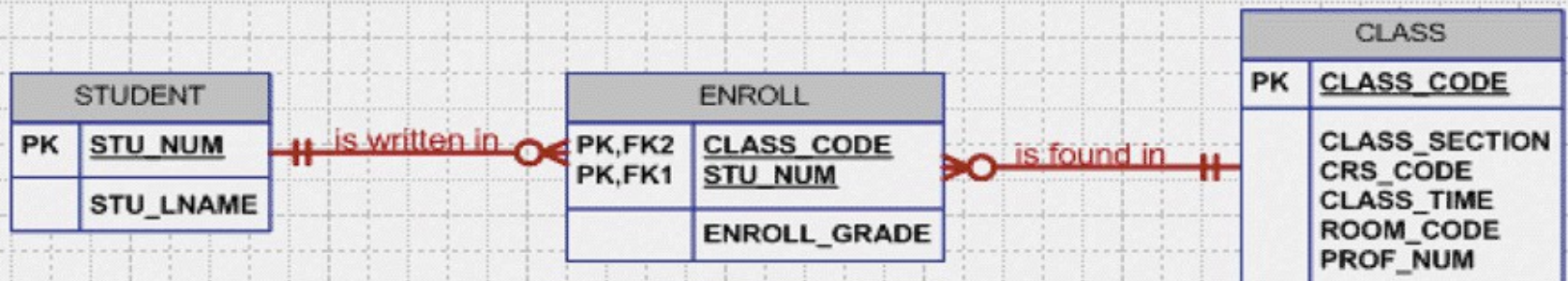
Attributes

FIGURE 4.26 A COMPOSITE ENTITY IN AN ERD

Chen Model



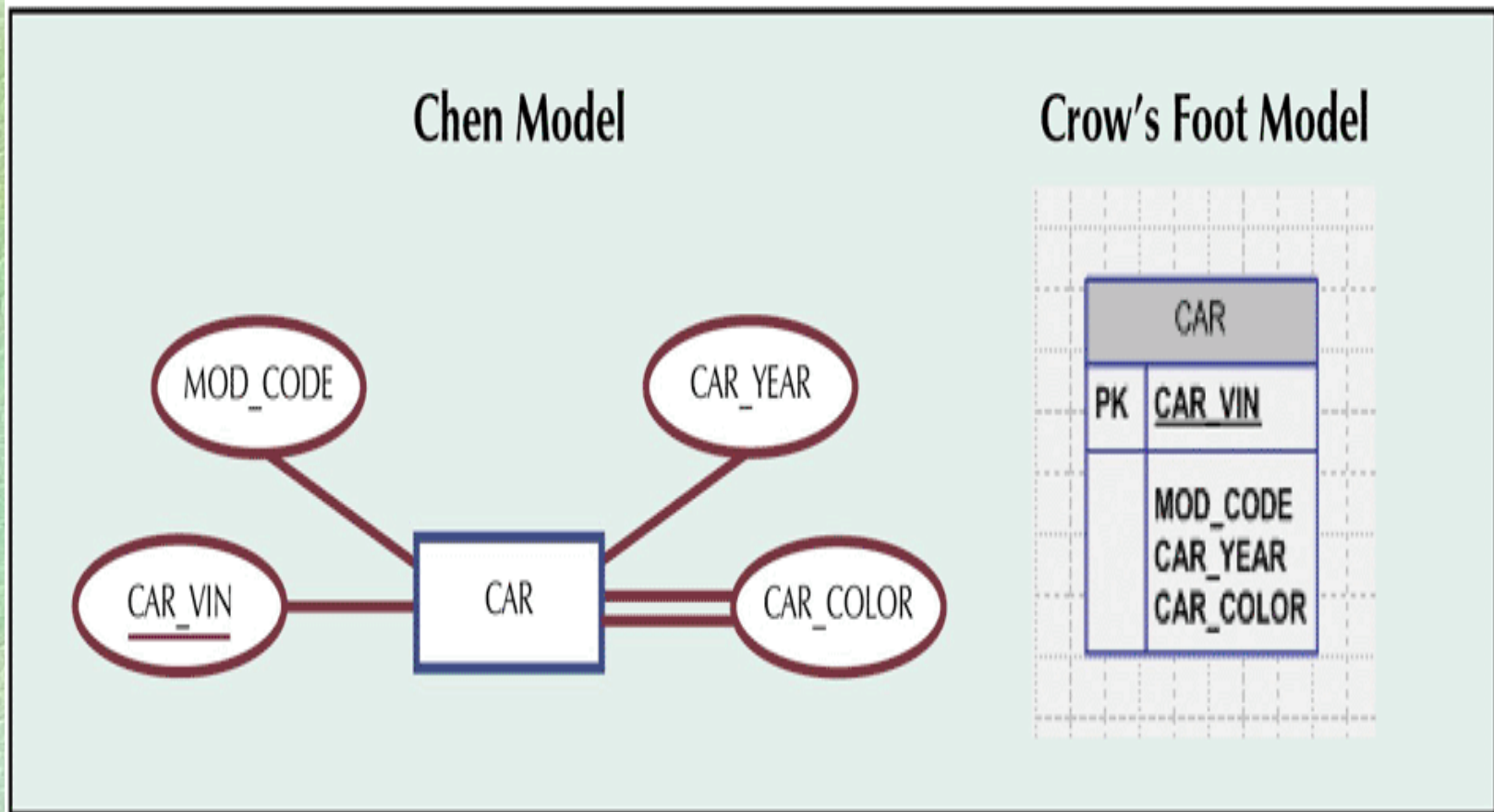
Crow's Foot Model



Multivalued Attributes in ERMs and ERDs

A Multivalued Attribute in an Entity: CAR_COLOR involves *multiple* colours

FIGURE 4.3 A MULTIVALUED ATTRIBUTE IN AN ENTITY



Multivalued Attributes

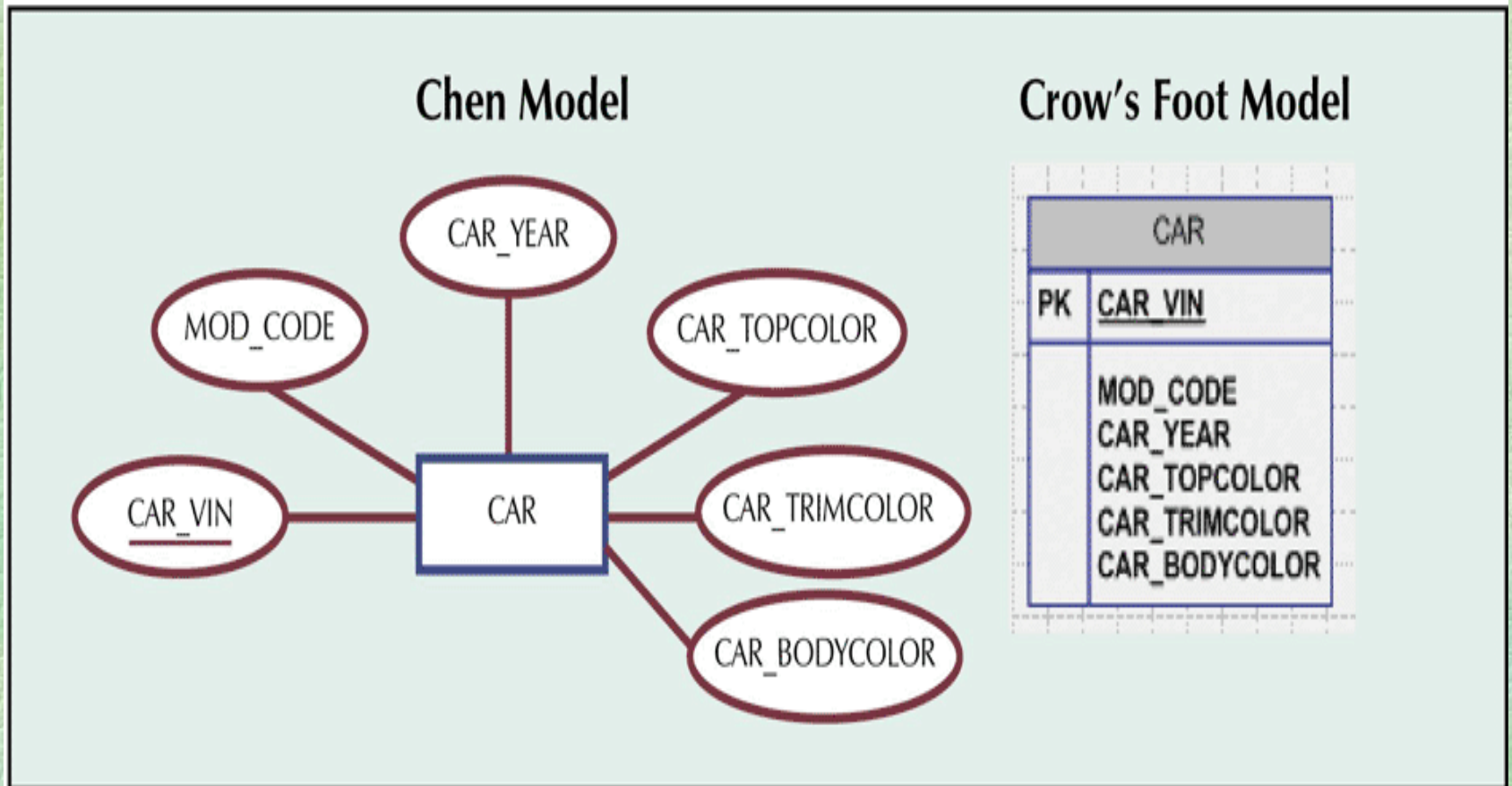
- ◆ “You should not implement them in the relational DBMS” [*rather, you should re-represent them in a special way* – J.A.B.]
- ◆ ***One (usually poor) possibility:*** Use a variable-length string for the attribute, and list all the values within the string.
Disadvantage: little support supplied by the DBMS – insertions and deletion require special extra programming. Similarly if calculations are needed on the individual values.

Multivalued Attributes, contd

- ◆ *Another possibility*: Within original entity type, split the attribute into *several different attributes* corresponding to different natural components of the entity. (See next slide.)

Splitting the Multivalued Attribute into New Naturally Namable Attributes

FIGURE 4.4 SPLITTING THE MULTIVALUED ATTRIBUTE INTO NEW ATTRIBUTES



Multivalued Attributes, contd

◆ *Disadvantages:*

- The attribute may in reality need to be *split differently* for different entities in the entity type (e.g. different cars).
- The attribute *may not have naturally namable aspects* at all.
E.g., imagine blotches of colour in random places on a car.

Multivalued Attribute Problems, contd

- ◆ *Another possibility*: Within original entity type, split the attribute into *several different attributes not* corresponding to specific components of the entity.

E.g., have attributes called **Colour1, Colour2, ... , Colour6.**

- *Advantage*: copes with the no-identifiable-components problem and the different-split problems. NB: also allows repetition of colours.

- *Disadvantages*:

Have to *set aside enough columns* to accommodate the conceivable max, but if this max is large and not often approached then have a lot of wasted space.

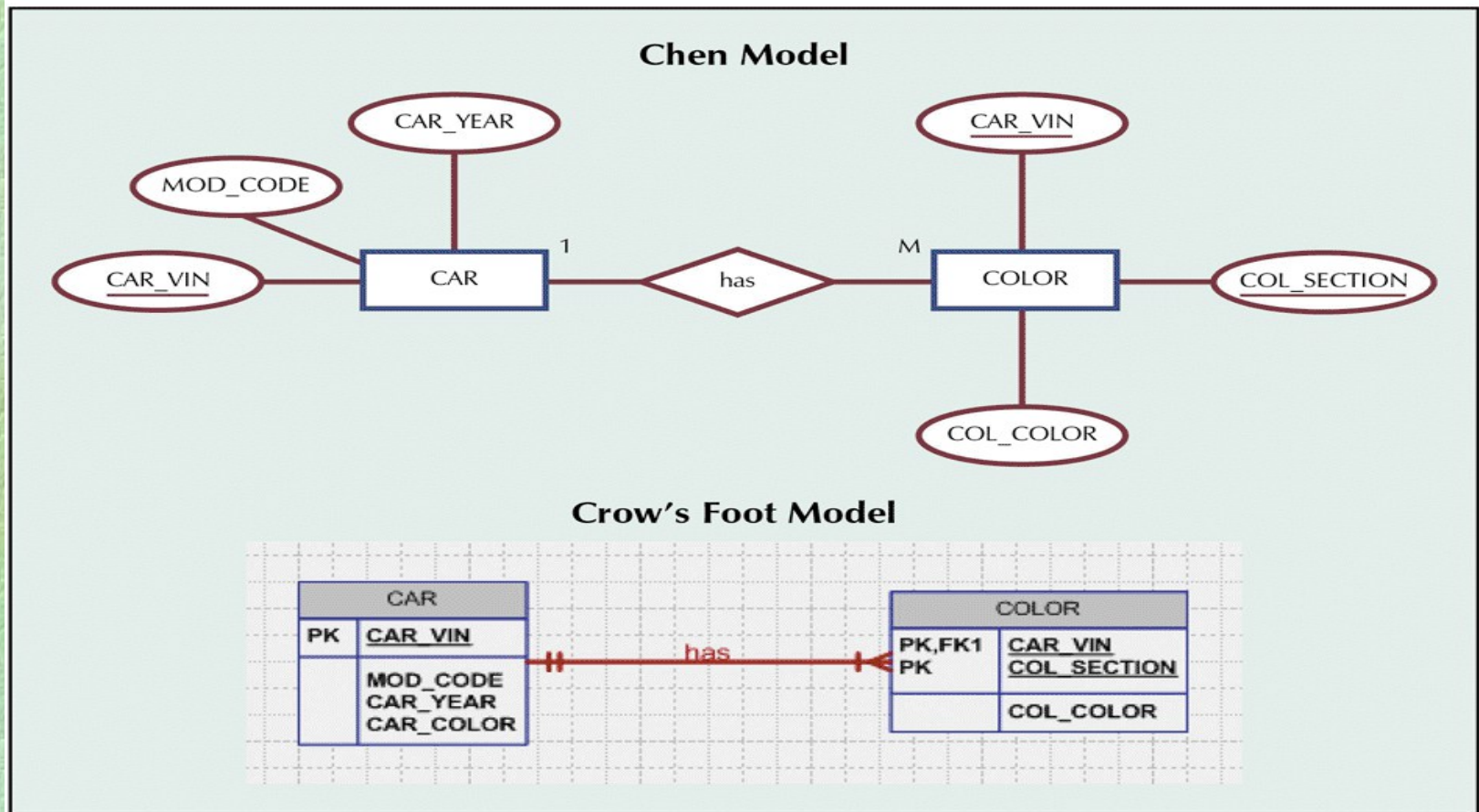
Searching for a colour, or doing insertions and deletions, can be *very cumbersome*.

Multivalued Attributes, contd

- ◆ *Often Better*: Replace the attribute by a new 1:M relationship to **a new entity type** holding the original attribute's data.

If the components of the original attribute are conceptually distinguishable in a natural way, the new entity can have an attribute whose values identify those components.

FIGURE 4.5 A NEW ENTITY SET COMPOSED OF A MULTIVALUED ATTRIBUTE'S COMPONENTS



There's a slight deficiency in the Chen diagram. What?

Multivalued Attributes, contd

- ◆ If the original multivalued attribute does not have naturally namable components, could perhaps use values like section13 in COL_SECTION.
- Or could perhaps leave out COL_SECTION. But NB: the PK would then need to include the colour. So we can't have easily repetitions of colours.
- What might we do about this? (Exercise.)

Generalization Hierarchies in ERMs and ERDs

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]

topic continued in following week