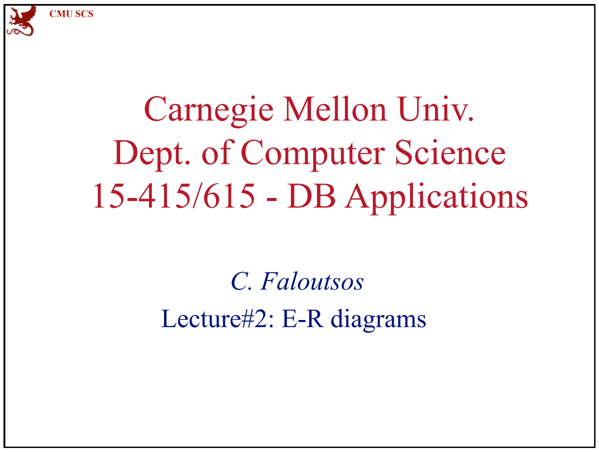
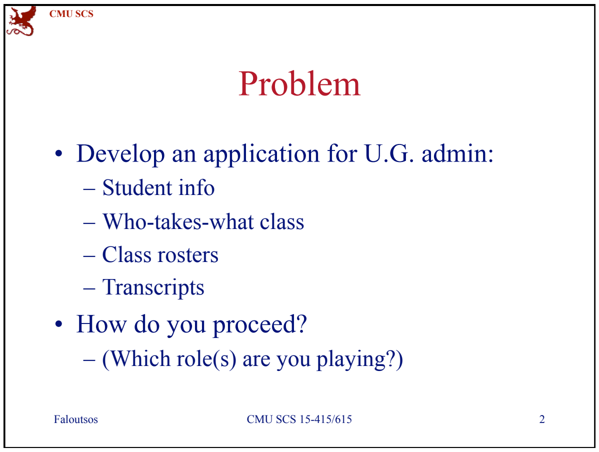


CMU SCS

Carnegie Mellon Univ.
Dept. of Computer Science
15-415/615 - DB Applications

C. Faloutsos

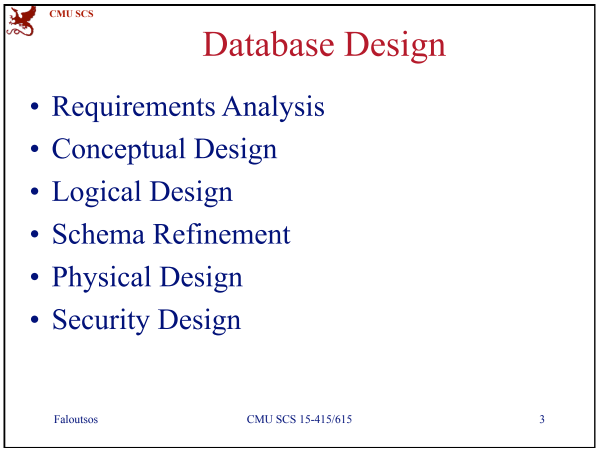
Lecture#2: E-R diagrams



CMU SCS

Problem

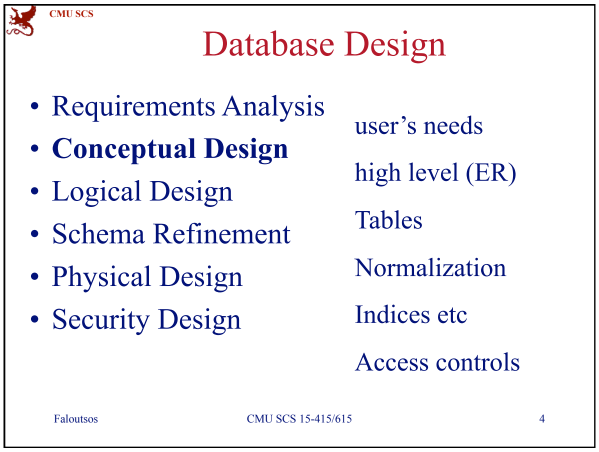
- Develop an application for U.G. admin:
  - Student info
  - Who-takes-what class
  - Class rosters
  - Transcripts
- How do you proceed?
  - (Which role(s) are you playing?)



CMU SCS

Database Design

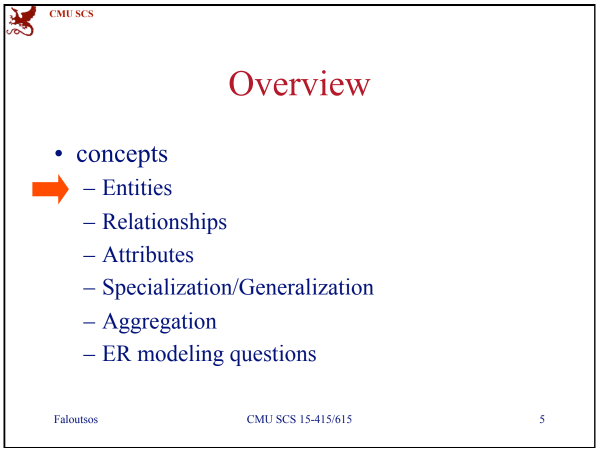
- Requirements Analysis
- Conceptual Design
- Logical Design
- Schema Refinement
- Physical Design
- Security Design



CMU SCS

Database Design

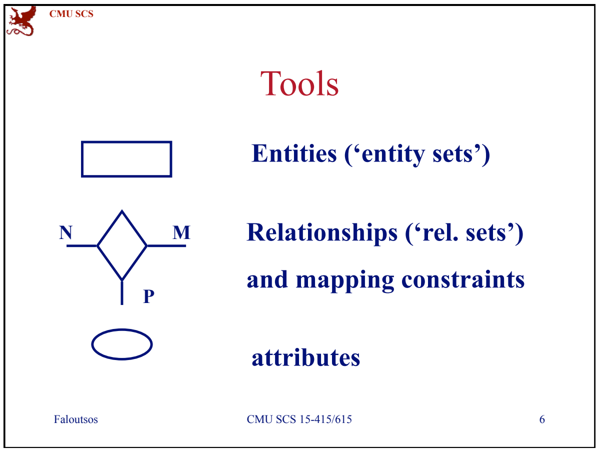
• Requirements Analysis	user's needs
• **Conceptual Design**	high level (ER)
• Logical Design	Tables
• Schema Refinement	Normalization
• Physical Design	Indices etc
• Security Design	Access controls



CMU SCS

Overview

- concepts
  - ➡ Entities
  - Relationships
  - Attributes
  - Specialization/Generalization
  - Aggregation
  - ER modeling questions



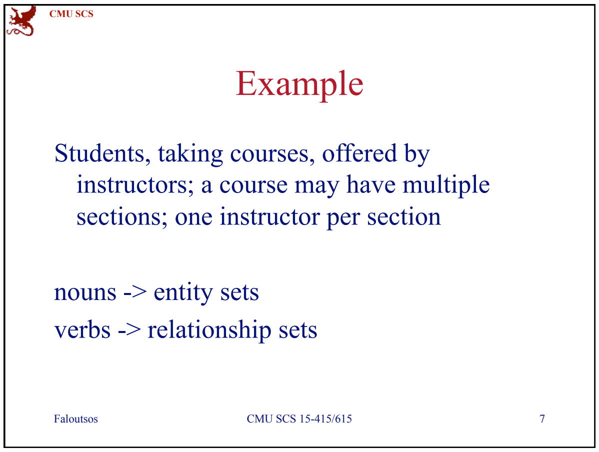
CMU SCS

Tools

Entities ('entity sets')

Relationships ('rel. sets') and mapping constraints

attributes



CMU SCS

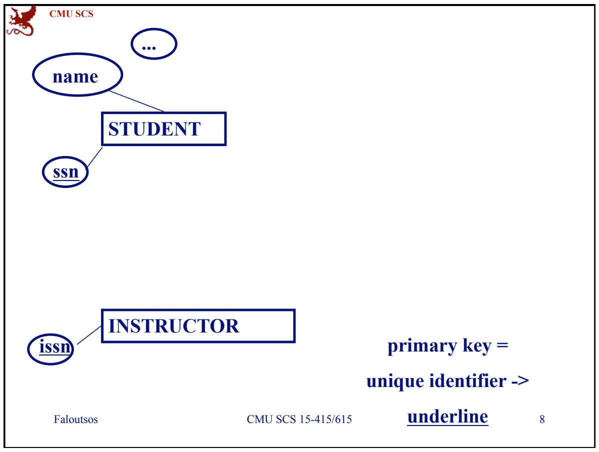
Example

Students, taking courses, offered by instructors; a course may have multiple sections; one instructor per section

nouns -> entity sets

verbs -> relationship sets

Faloutsos CMU SCS 15-415/615 7



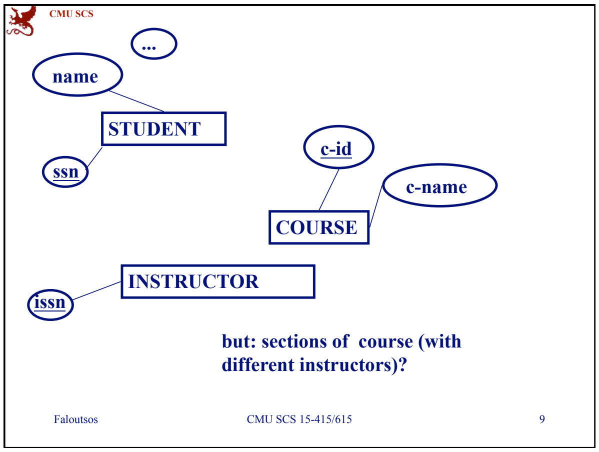
CMU SCS

primary key =

unique identifier ->

underline

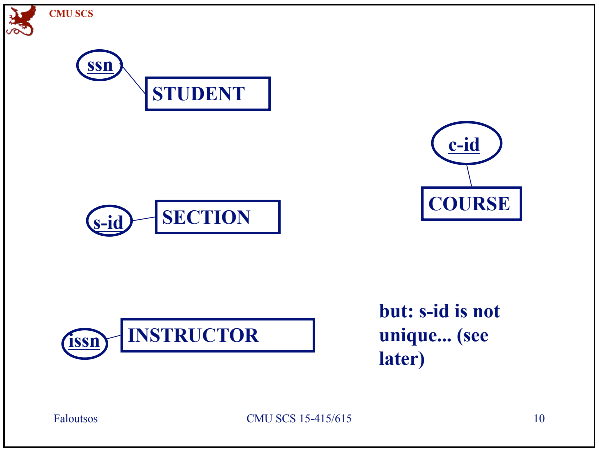
Faloutsos CMU SCS 15-415/615 8



CMU SCS

but: sections of course (with different instructors)?

Faloutsos CMU SCS 15-415/615 9



CMU SCS

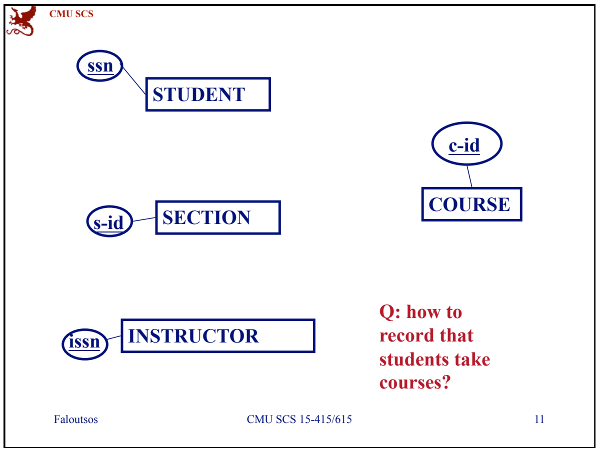
ssn
STUDENT

c-id
COURSE

s-id
SECTION

issn
INSTRUCTOR

but: s-id is not unique... (see later)



CMU SCS

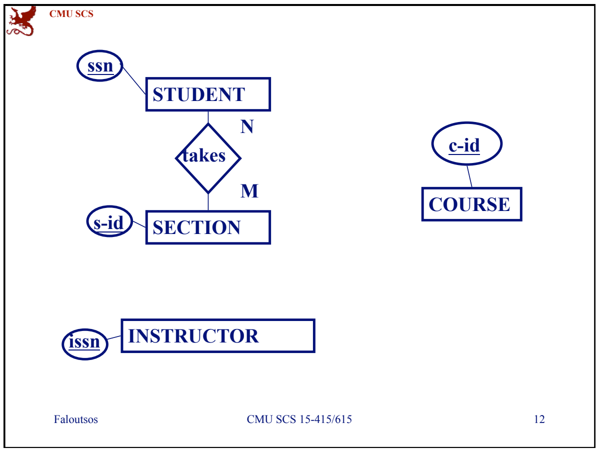
ssn
STUDENT

c-id
COURSE

s-id
SECTION

issn
INSTRUCTOR

Q: how to record that students take courses?

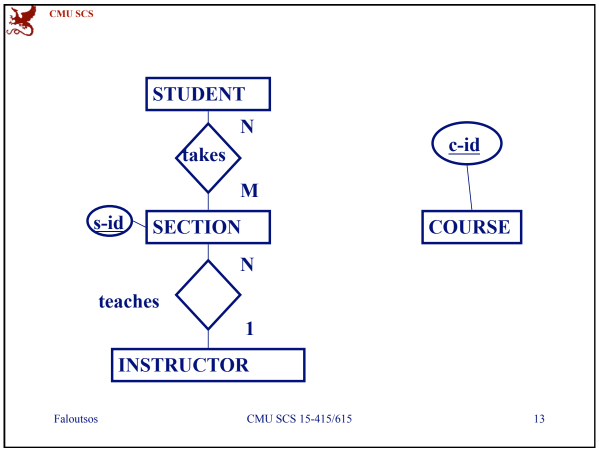


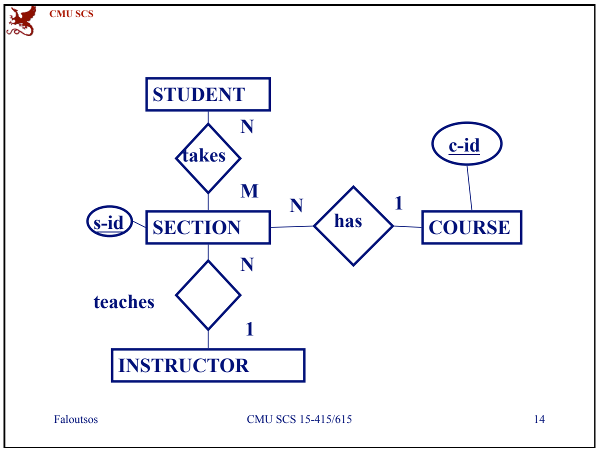
CMU SCS

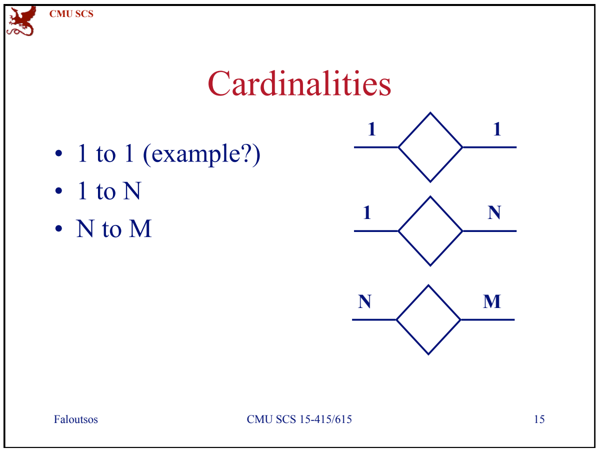
ssn
STUDENT
N
takes
M
s-id
SECTION

c-id
COURSE

issn
INSTRUCTOR





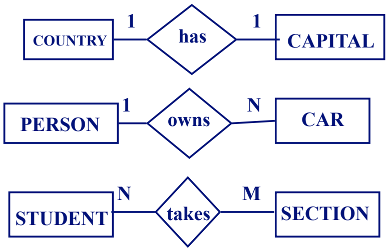


Cardinalities

- 1 to 1 (example?)
- 1 to N
- N to M

CMU SCS

Cardinalities

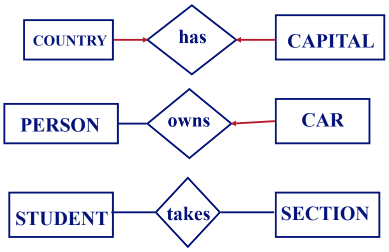


Faloutsos CMU SCS 15-415/615 16

CMU SCS

Cardinalities

Book's notation:



Faloutsos CMU SCS 15-415/615 17

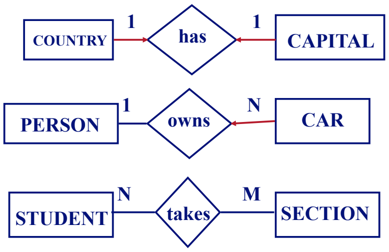
CMU SCS

Cardinalities

Book's notation

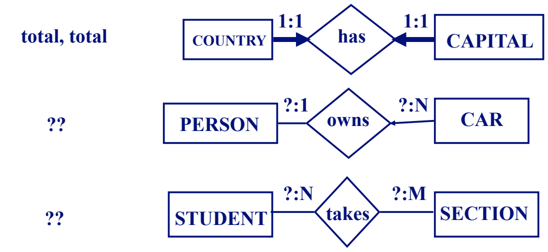
vs

***1 to N* notation**

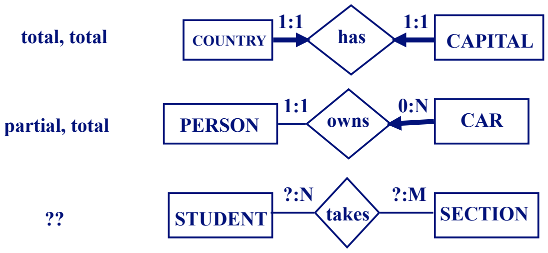


Faloutsos CMU SCS 15-415/615 18

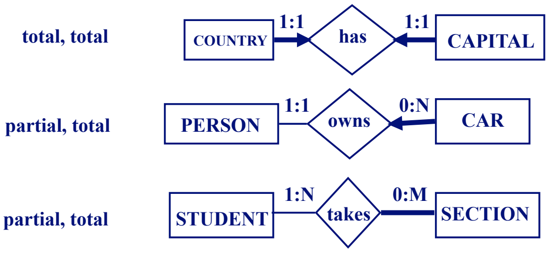
‘Total/partial’ participation

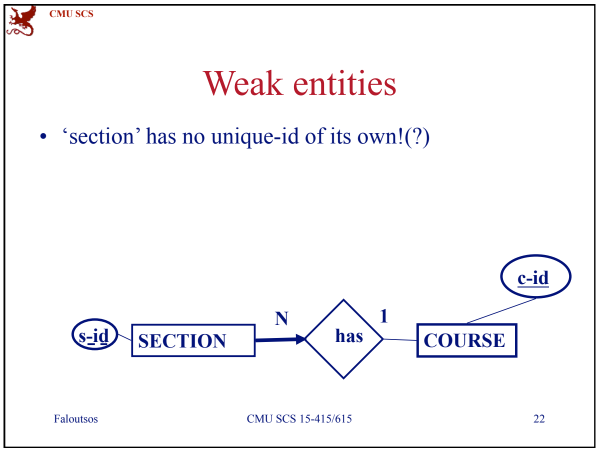


‘Total/partial’ participation



‘Total/partial’ participation



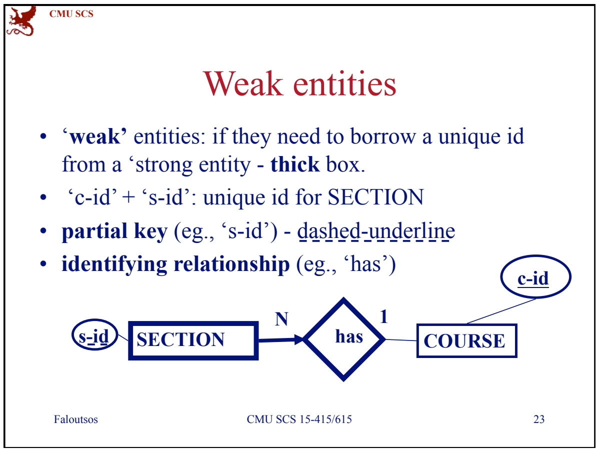


CMU SCS

Weak entities

- 'section' has no unique-id of its own!(?)

Faloutsos CMU SCS 15-415/615 22

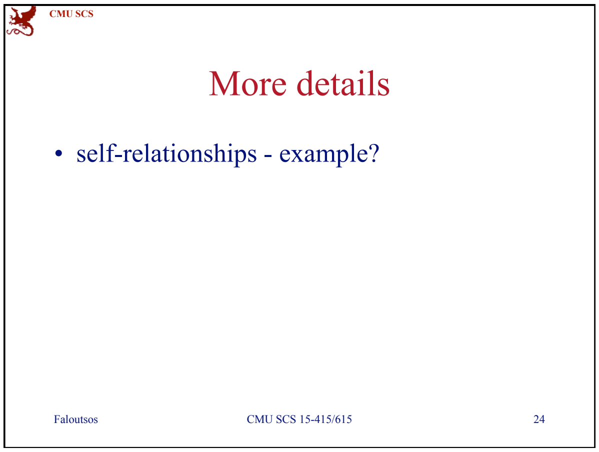


CMU SCS

Weak entities

- '**weak'** entities: if they need to borrow a unique id from a 'strong entity - **thick** box.
- 'c-id' + 's-id': unique id for SECTION
- **partial key** (eg., 's-id') - dashed-underline
- **identifying relationship** (eg., 'has')

Faloutsos CMU SCS 15-415/615 23

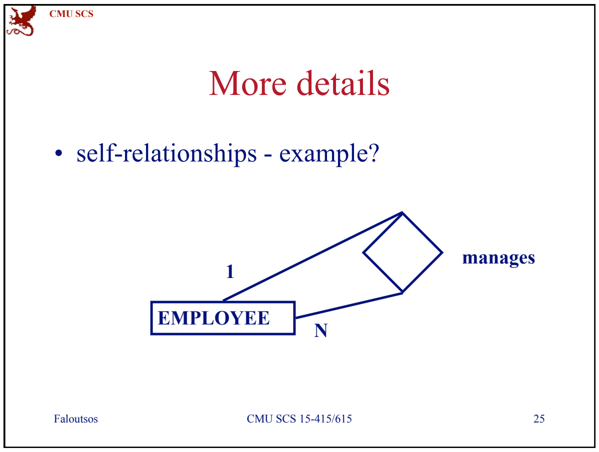


CMU SCS

More details

- self-relationships - example?

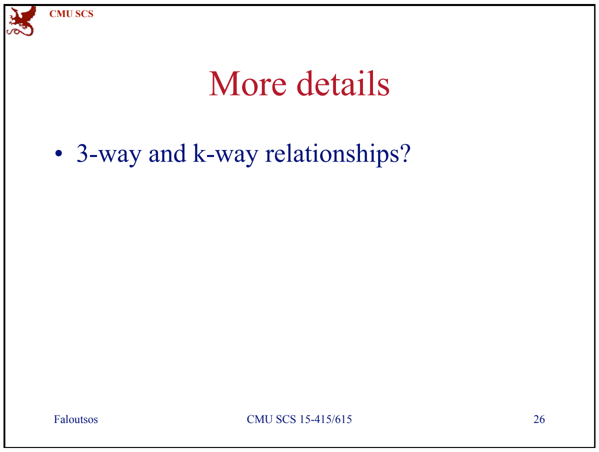
Faloutsos CMU SCS 15-415/615 24



CMU SCS

More details

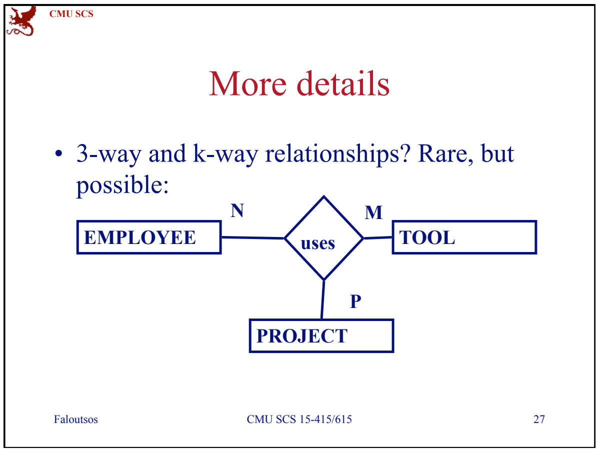
- self-relationships - example?



CMU SCS

More details

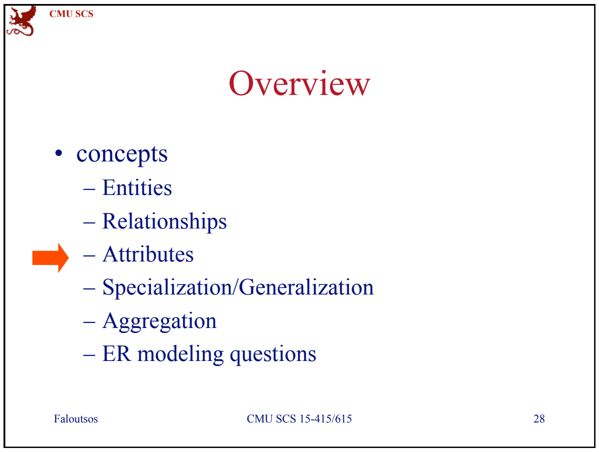
- 3-way and k-way relationships?



CMU SCS

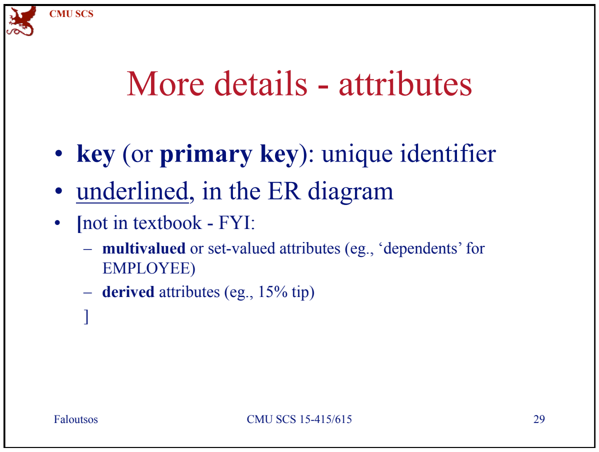
More details

- 3-way and k-way relationships? Rare, but possible:



Overview

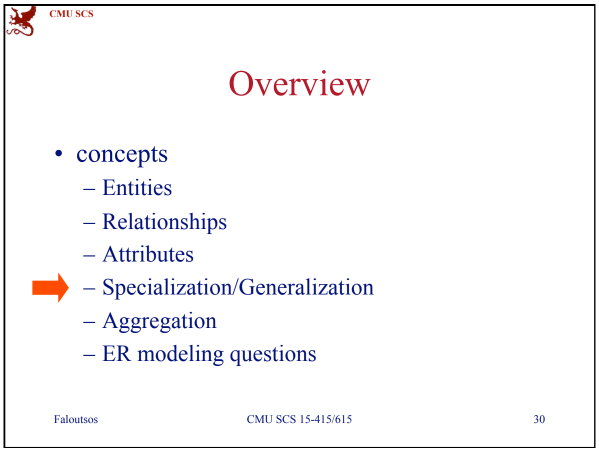
- concepts
  - Entities
  - Relationships
  - ➡ Attributes
  - Specialization/Generalization
  - Aggregation
  - ER modeling questions



More details - attributes

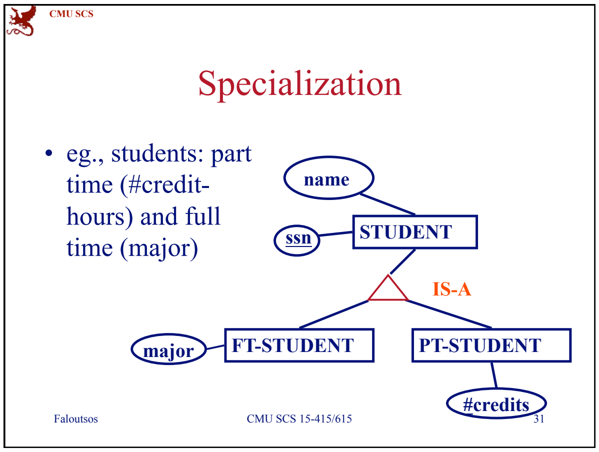
- **key** (or **primary key**): unique identifier
- underlined, in the ER diagram
- [not in textbook - FYI:
  - **multivalued** or set-valued attributes (eg., ‘dependents’ for EMPLOYEE)
  - **derived** attributes (eg., 15% tip)

  ]



Overview

- concepts
  - Entities
  - Relationships
  - Attributes
  - ➡ Specialization/Generalization
  - Aggregation
  - ER modeling questions



CMU SCS

Specialization

- eg., students: part time (#credit-hours) and full time (major)

name

ssn

STUDENT

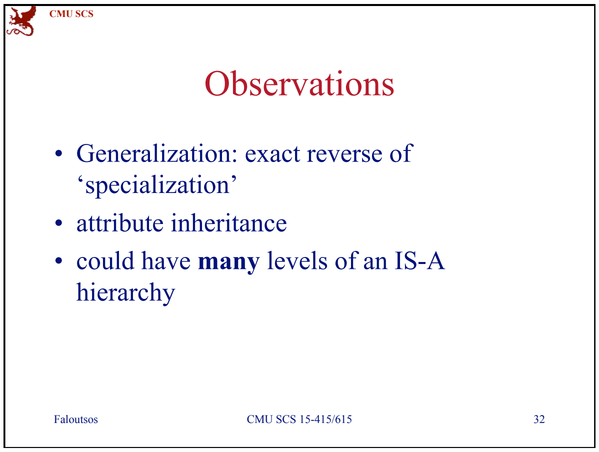
IS-A

major

FT-STUDENT

PT-STUDENT

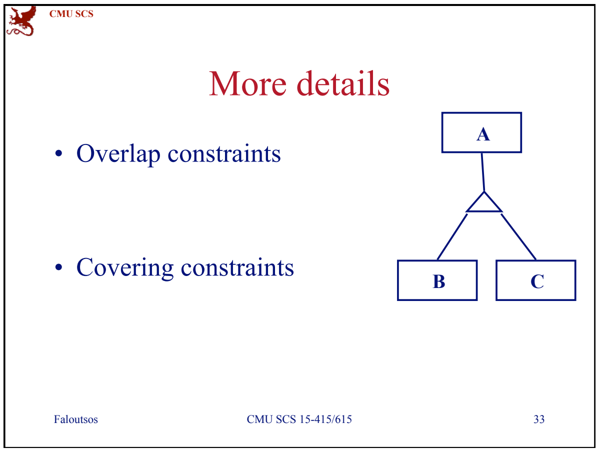
#credits



CMU SCS

Observations

- Generalization: exact reverse of ‘specialization’
- attribute inheritance
- could have **many** levels of an IS-A hierarchy



CMU SCS

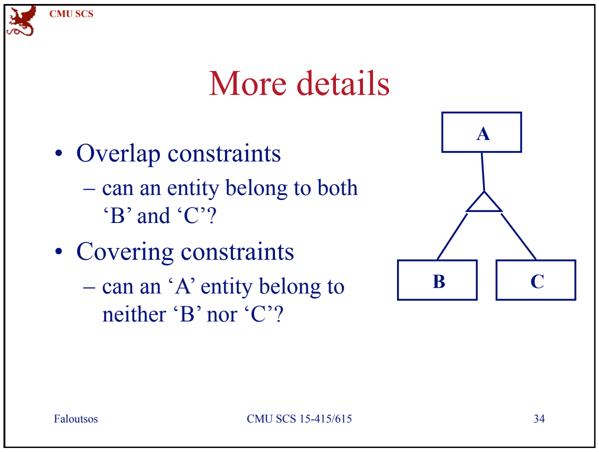
More details

- Overlap constraints
- Covering constraints

A

B

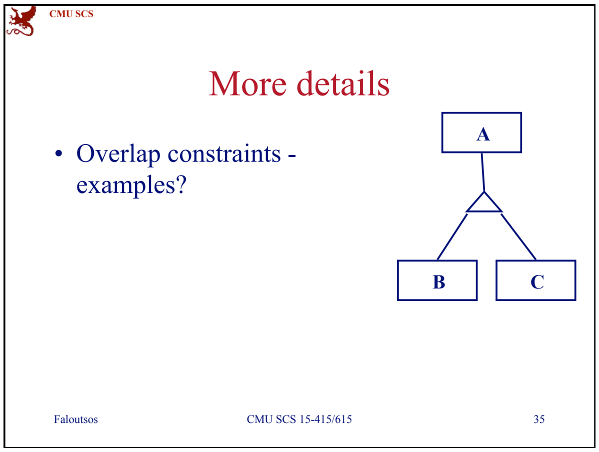
C



CMU SCS

More details

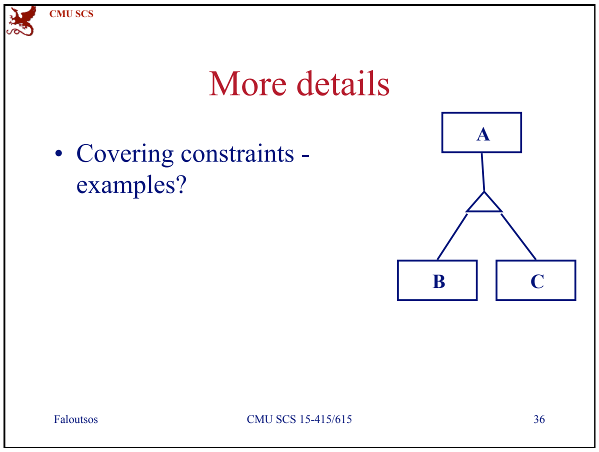
- Overlap constraints
  - can an entity belong to both ‘B’ and ‘C’?
- Covering constraints
  - can an ‘A’ entity belong to neither ‘B’ nor ‘C’?



CMU SCS

More details

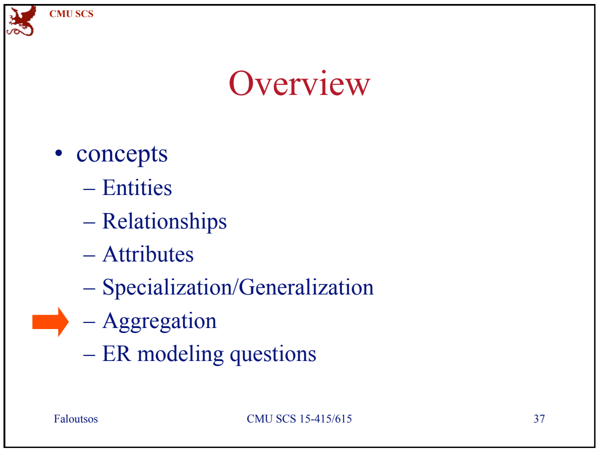
- Overlap constraints - examples?



CMU SCS

More details

- Covering constraints - examples?

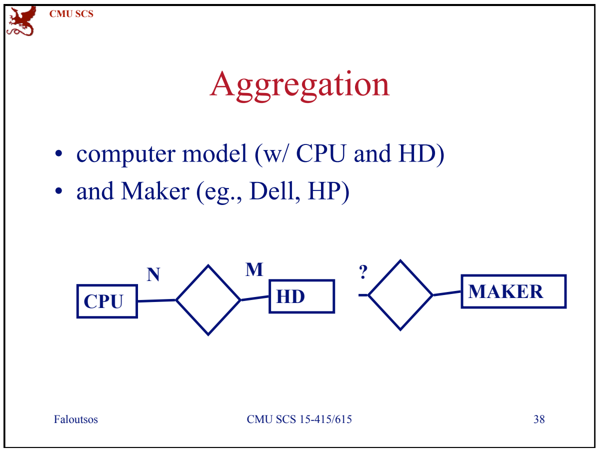


CMU SCS

Overview

- concepts
  - Entities
  - Relationships
  - Attributes
  - Specialization/Generalization
  - ➡ Aggregation
  - ER modeling questions

Faloutsos CMU SCS 15-415/615 37

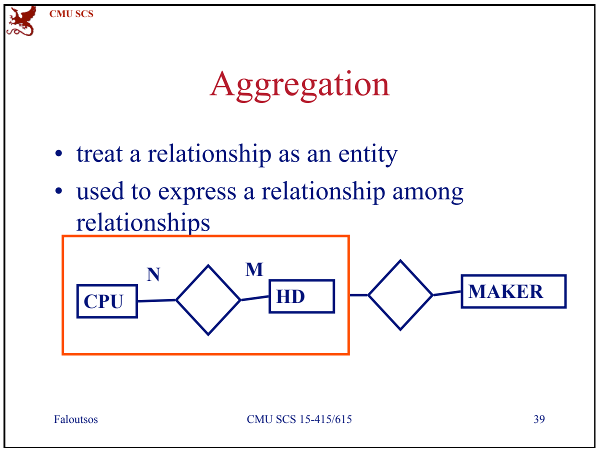


CMU SCS

Aggregation

- computer model (w/ CPU and HD)
- and Maker (eg., Dell, HP)

Faloutsos CMU SCS 15-415/615 38

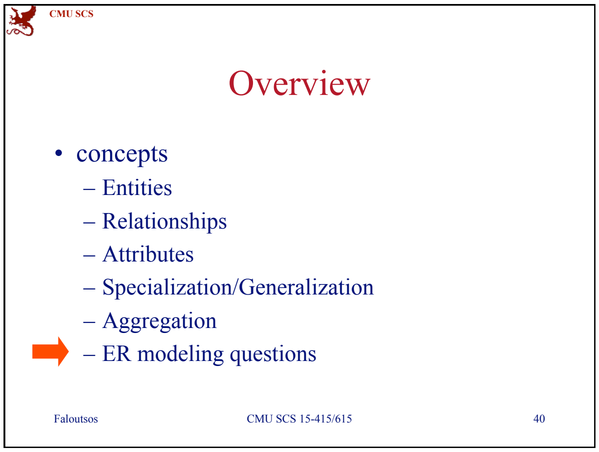


CMU SCS

Aggregation

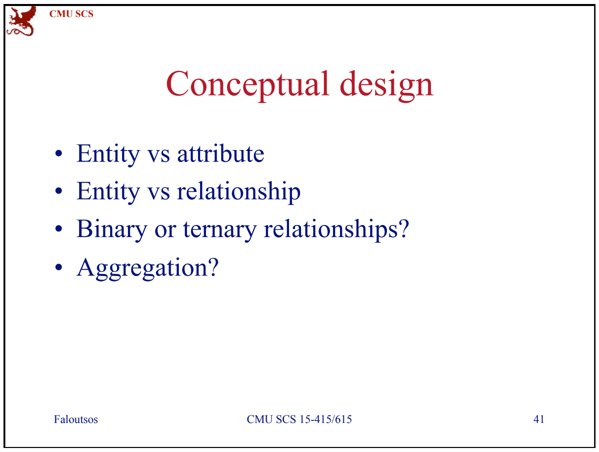
- treat a relationship as an entity
- used to express a relationship among relationships

Faloutsos CMU SCS 15-415/615 39



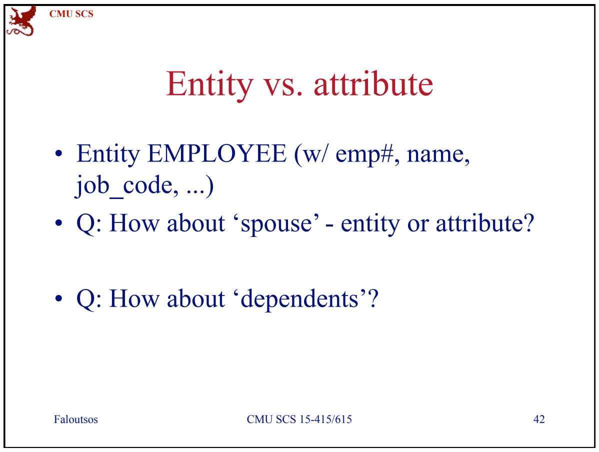
Overview

- concepts
  - Entities
  - Relationships
  - Attributes
  - Specialization/Generalization
  - Aggregation
  - ➡ ER modeling questions



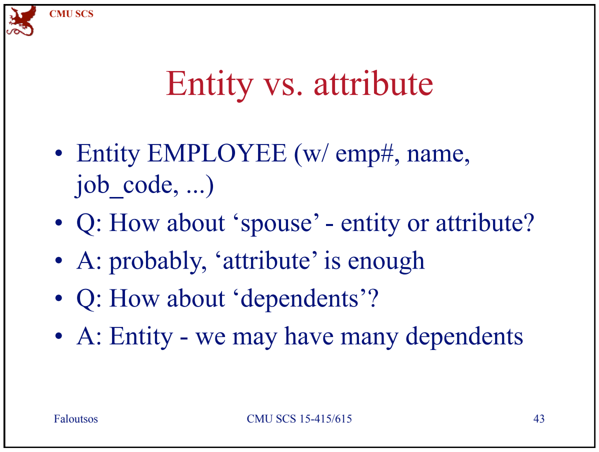
Conceptual design

- Entity vs attribute
- Entity vs relationship
- Binary or ternary relationships?
- Aggregation?



Entity vs. attribute

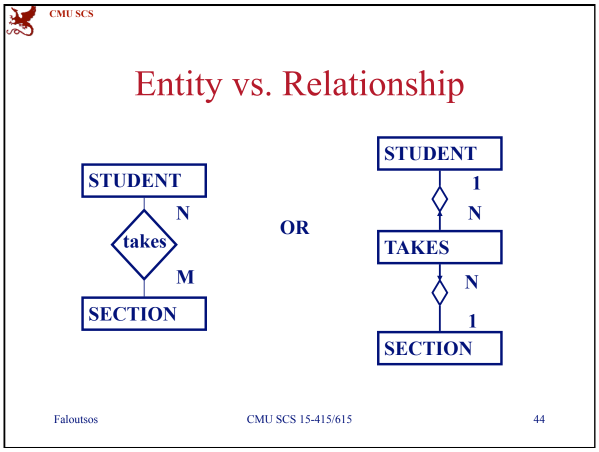
- Entity EMPLOYEE (w/ emp#, name, job_code, ...)
- Q: How about ‘spouse’ - entity or attribute?
- Q: How about ‘dependents’?



CMU SCS

Entity vs. attribute

- Entity EMPLOYEE (w/ emp#, name, job_code, ...)
- Q: How about 'spouse' - entity or attribute?
- A: probably, 'attribute' is enough
- Q: How about 'dependents'?
- A: Entity - we may have many dependents



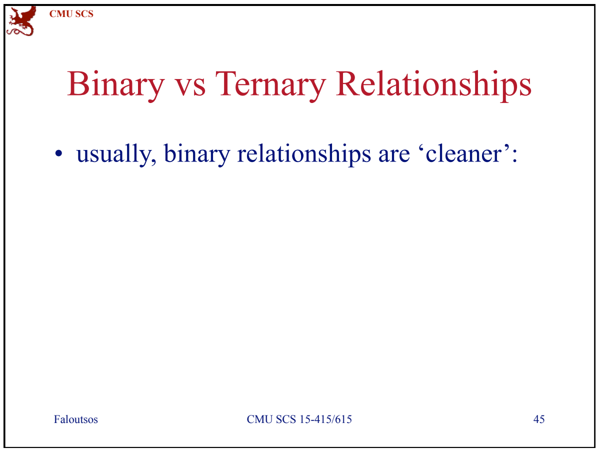
CMU SCS

Entity vs. Relationship

STUDENT
N
takes
M
SECTION

OR

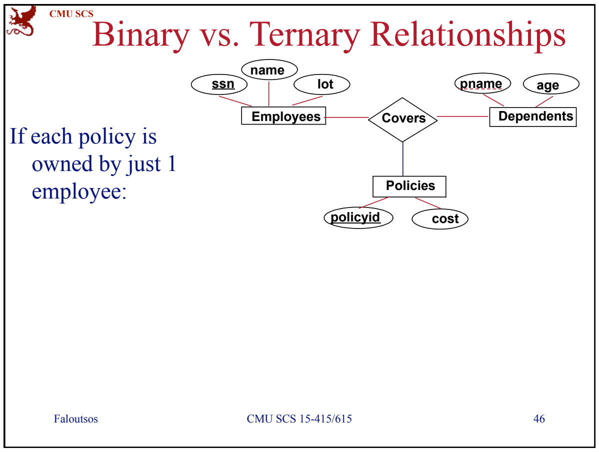
STUDENT
1
N
TAKES
N
1
SECTION



CMU SCS

Binary vs Ternary Relationships

- usually, binary relationships are 'cleaner':

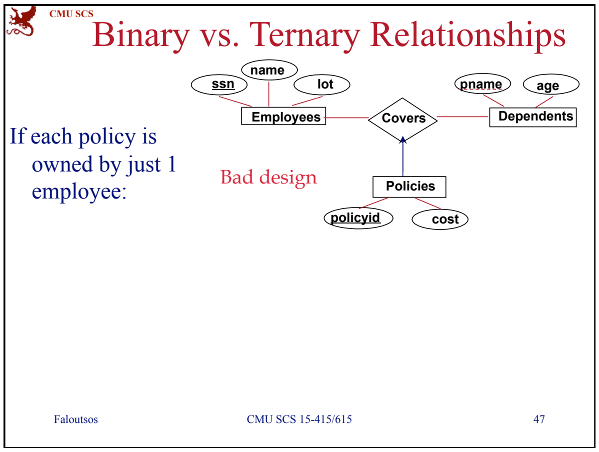


CMU SCS

Binary vs. Ternary Relationships

If each policy is owned by just 1 employee:

ssn
name
lot
Employees
Covers
pname
age
Dependents
Policies
policyid
cost



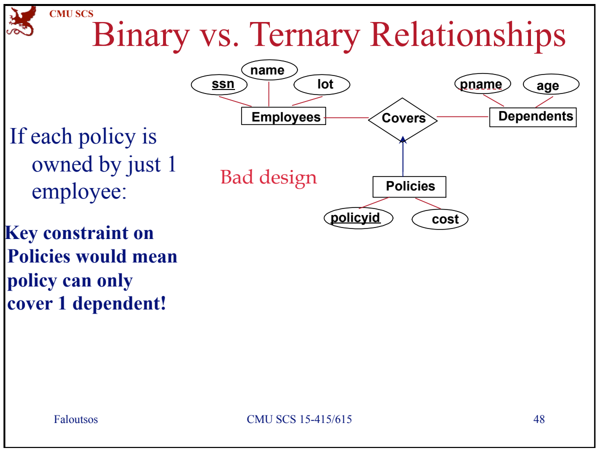
CMU SCS

Binary vs. Ternary Relationships

If each policy is owned by just 1 employee:

Bad design

ssn
name
lot
Employees
Covers
pname
age
Dependents
Policies
policyid
cost



CMU SCS

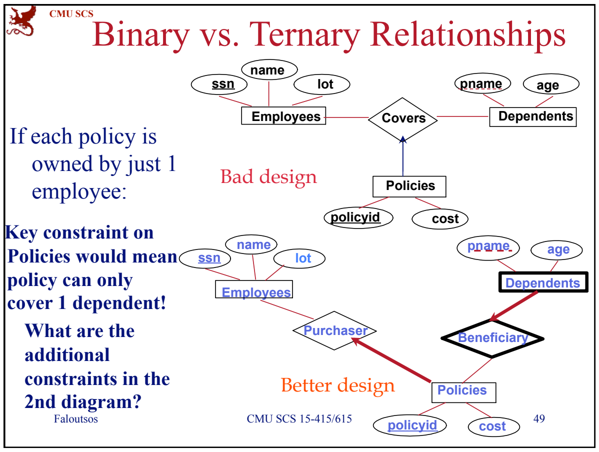
Binary vs. Ternary Relationships

If each policy is owned by just 1 employee:

Bad design

ssn
name
lot
Employees
Covers
pname
age
Dependents
Policies
policyid
cost

Key constraint on Policies would mean policy can only cover 1 dependent!



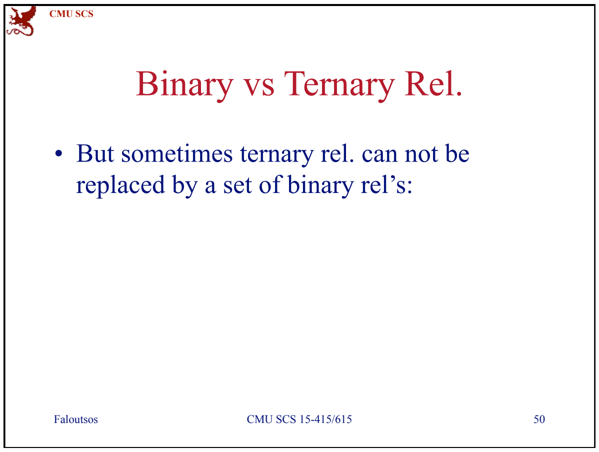
CMU SCS

Binary vs. Ternary Relationships

If each policy is owned by just 1 employee:

Key constraint on Policies would mean policy can only cover 1 dependent!

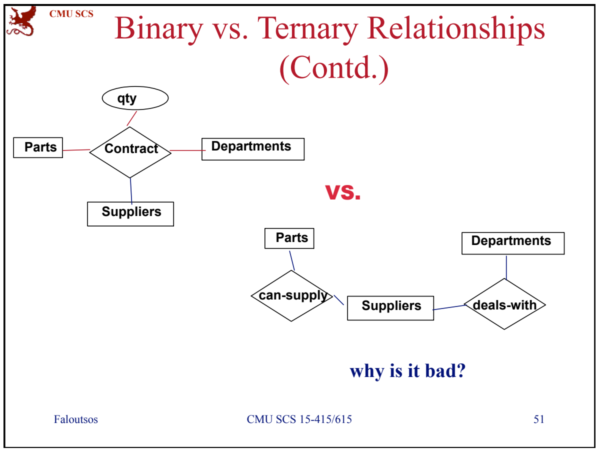
What are the additional constraints in the 2nd diagram?



CMU SCS

Binary vs Ternary Rel.

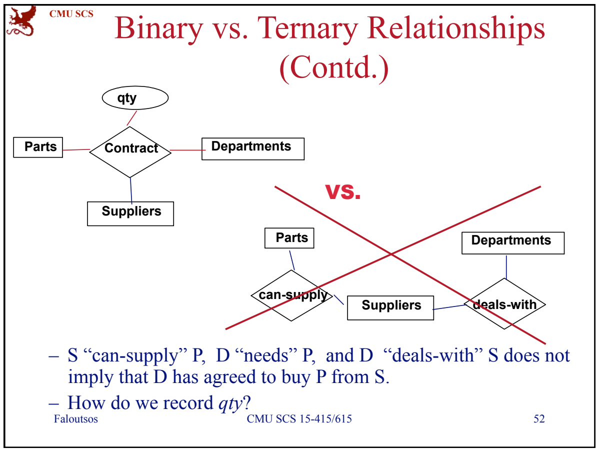
- But sometimes ternary rel. can not be replaced by a set of binary rel's:



CMU SCS

Binary vs. Ternary Relationships (Contd.)

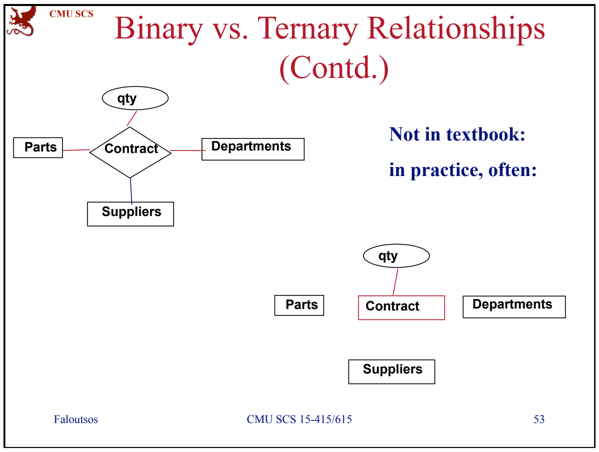
why is it bad?



CMU SCS

Binary vs. Ternary Relationships (Contd.)

- S “can-supply” P, D “needs” P, and D “deals-with” S does not imply that D has agreed to buy P from S.
- How do we record *qty*?

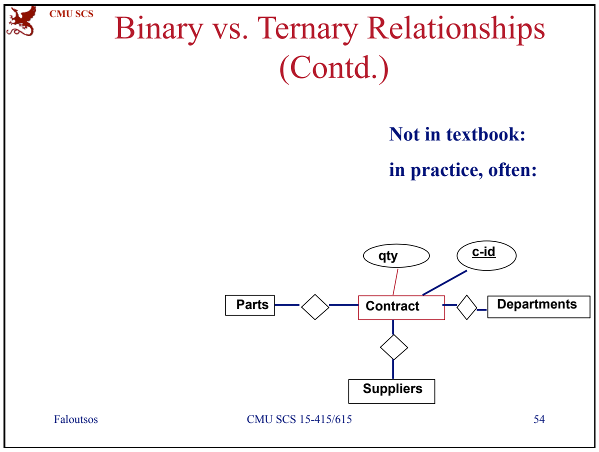


CMU SCS

Binary vs. Ternary Relationships (Contd.)

Not in textbook:

in practice, often:

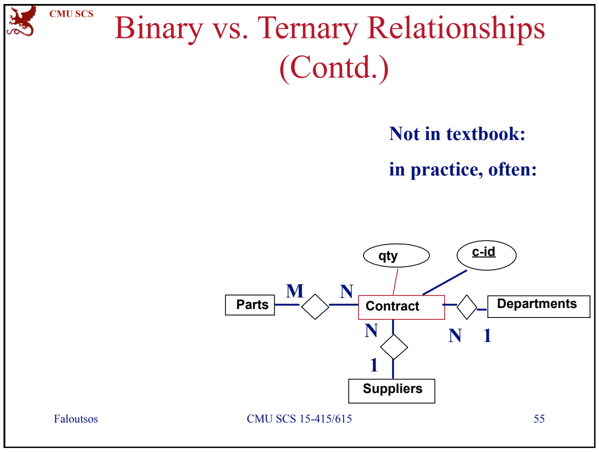


CMU SCS

Binary vs. Ternary Relationships (Contd.)

Not in textbook:

in practice, often:

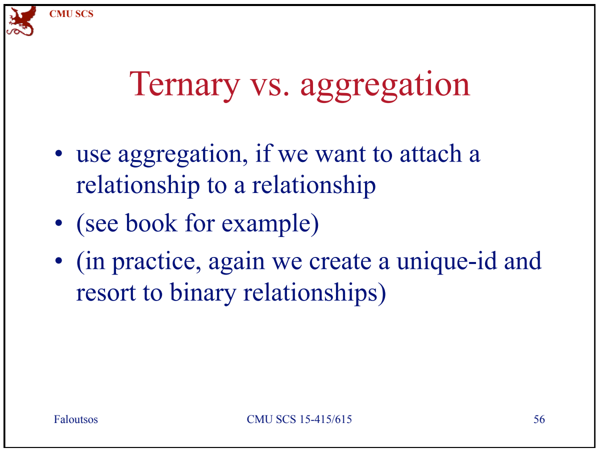


CMU SCS

Binary vs. Ternary Relationships (Contd.)

Not in textbook:

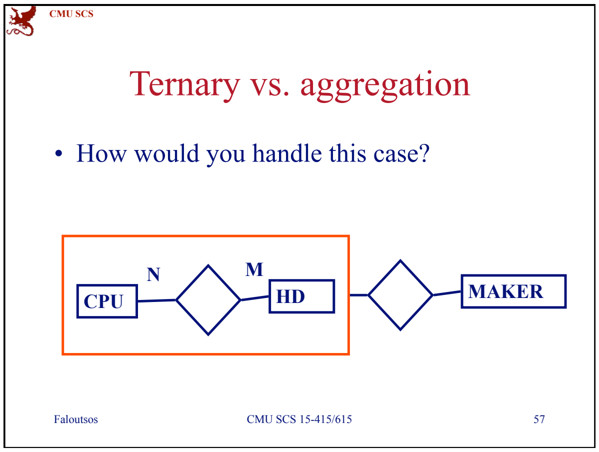
in practice, often:



CMU SCS

Ternary vs. aggregation

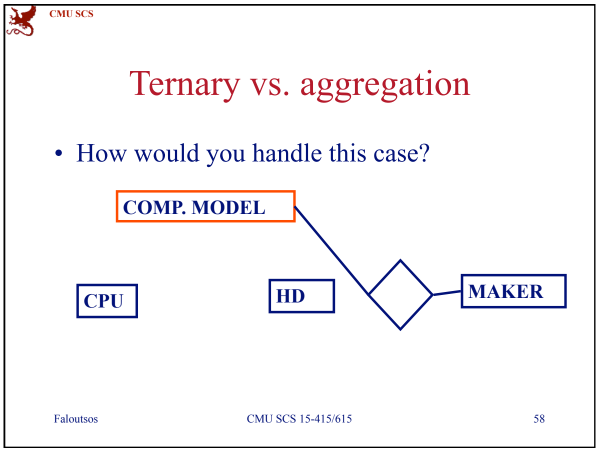
- use aggregation, if we want to attach a relationship to a relationship
- (see book for example)
- (in practice, again we create a unique-id and resort to binary relationships)



CMU SCS

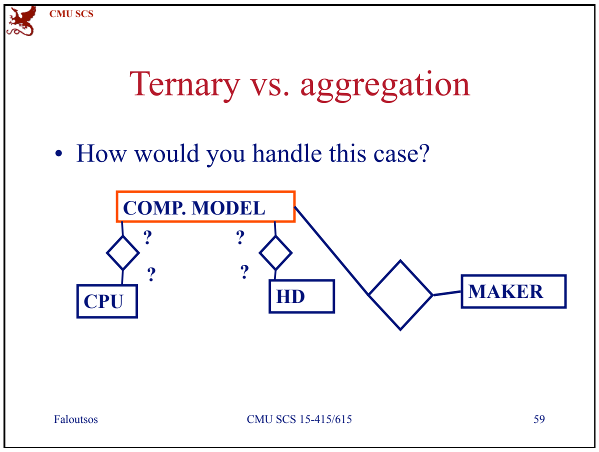
Ternary vs. aggregation

- How would you handle this case?



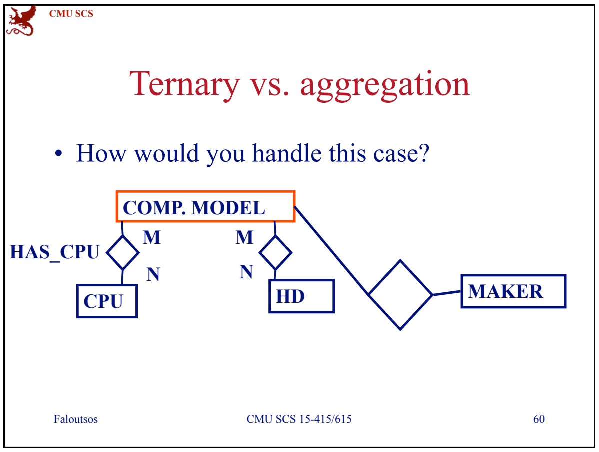
Ternary vs. aggregation

- How would you handle this case?



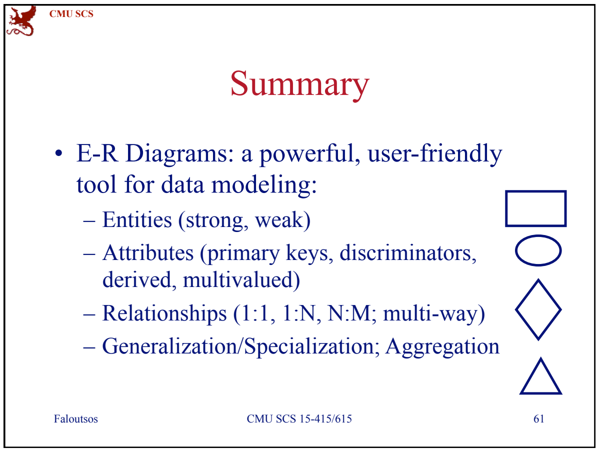
Ternary vs. aggregation

- How would you handle this case?



Ternary vs. aggregation

- How would you handle this case?

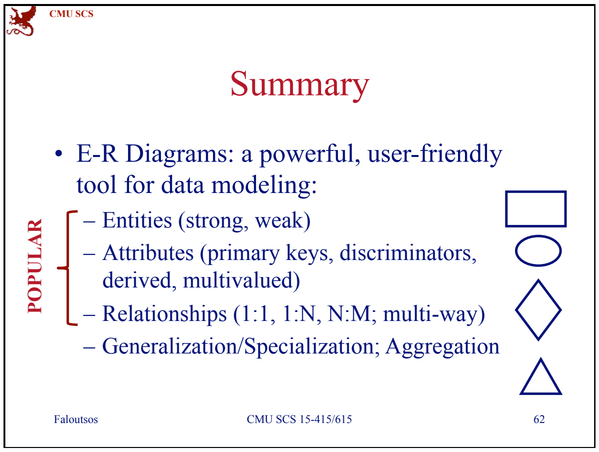


CMU SCS

Summary

- E-R Diagrams: a powerful, user-friendly tool for data modeling:
  - Entities (strong, weak)
  - Attributes (primary keys, discriminators, derived, multivalued)
  - Relationships (1:1, 1:N, N:M; multi-way)
  - Generalization/Specialization; Aggregation

Faloutsos CMU SCS 15-415/615 61

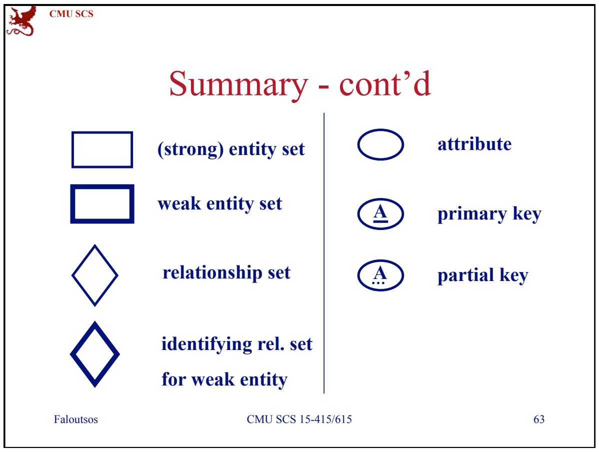


CMU SCS

Summary

- E-R Diagrams: a powerful, user-friendly tool for data modeling:
  - Entities (strong, weak)
  - Attributes (primary keys, discriminators, derived, multivalued)
  - Relationships (1:1, 1:N, N:M; multi-way)
  - Generalization/Specialization; Aggregation

Faloutsos CMU SCS 15-415/615 62



CMU SCS

Summary - cont’d

(strong) entity set

weak entity set

relationship set

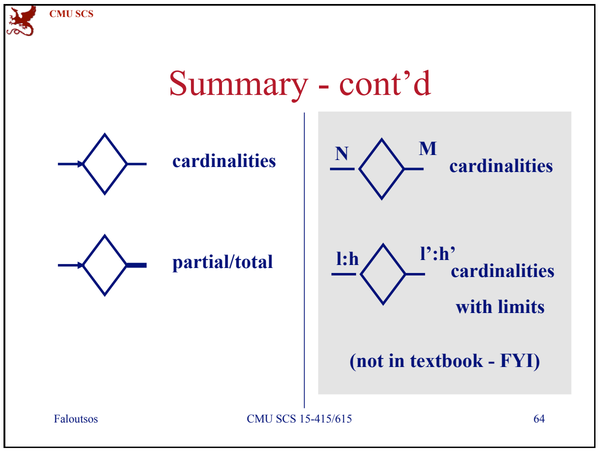
identifying rel. set for weak entity

attribute

primary key

partial key

Faloutsos CMU SCS 15-415/615 63



CMU SCS

Summary - cont'd

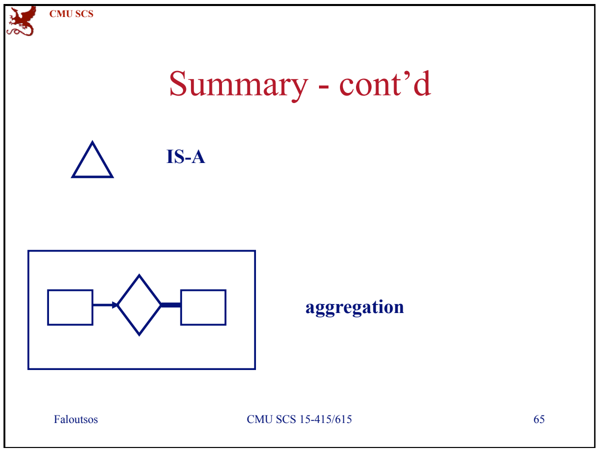
cardinalities

partial/total

N M cardinalities

l:h l':h' cardinalities with limits

(not in textbook - FYI)



CMU SCS

Summary - cont'd

IS-A

aggregation