Updated 10/25/2022

Topic 2: Database design
L13: ER modeling

Wolfgang Gatterbauer

CS3200 Database design (fa22)

https://northeastern-datalab.github.io/cs3200/fa22s3/

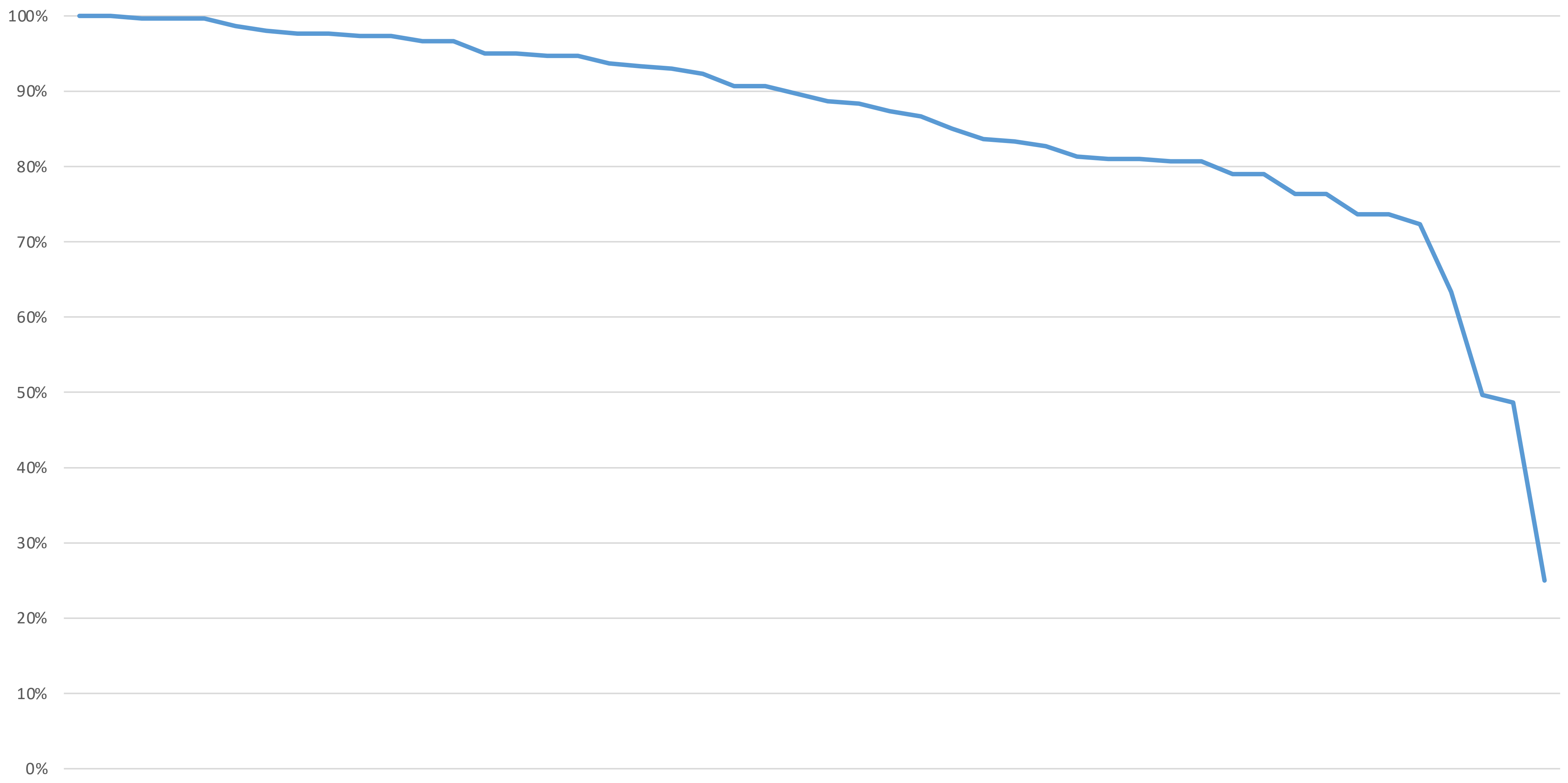
10/24/2022

Class warm-up

- Last class summary
- Any questions on project?
- Next week WED online class: Zoom vs Teams?
- Please have paper and pen ready (or touchscreen, whatever works) and keep ask questions: many aspects of "database design" are not set in stone and in the "eye of the designer"
- Exam1 discussion: FMs (frequent mistakes)
  - closed book; Please use our various options for feedback

- We continue with Database Design, hands-on

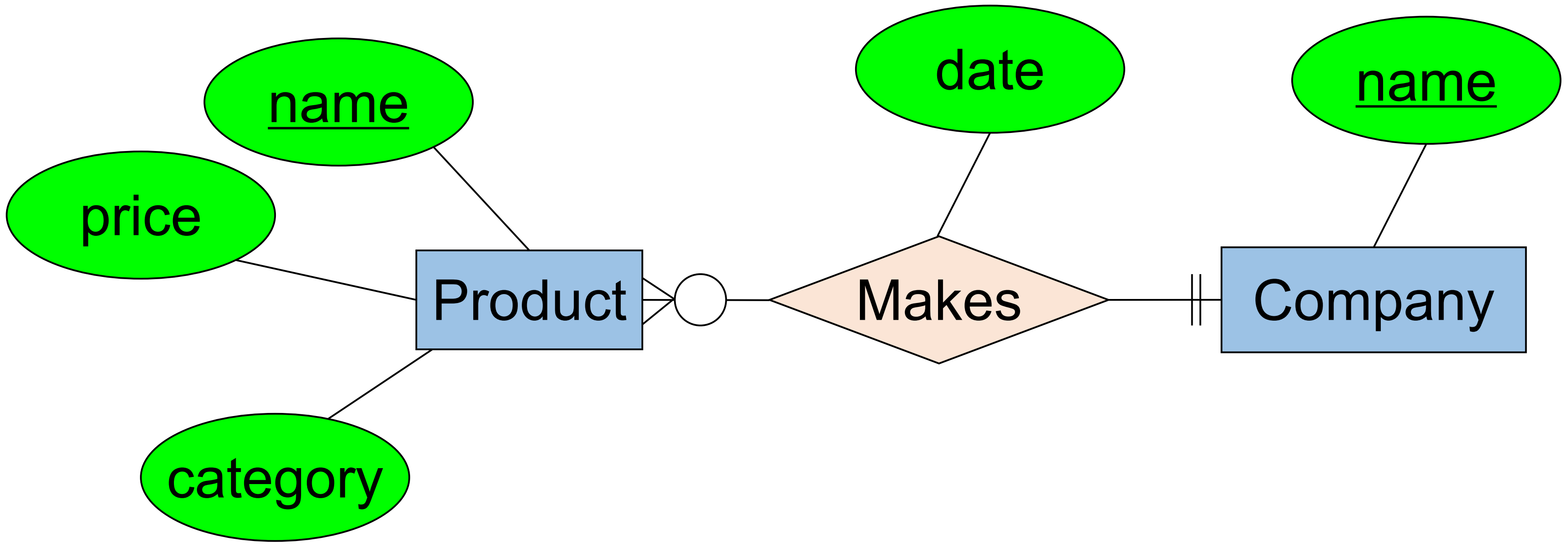
Exam 1 distribution



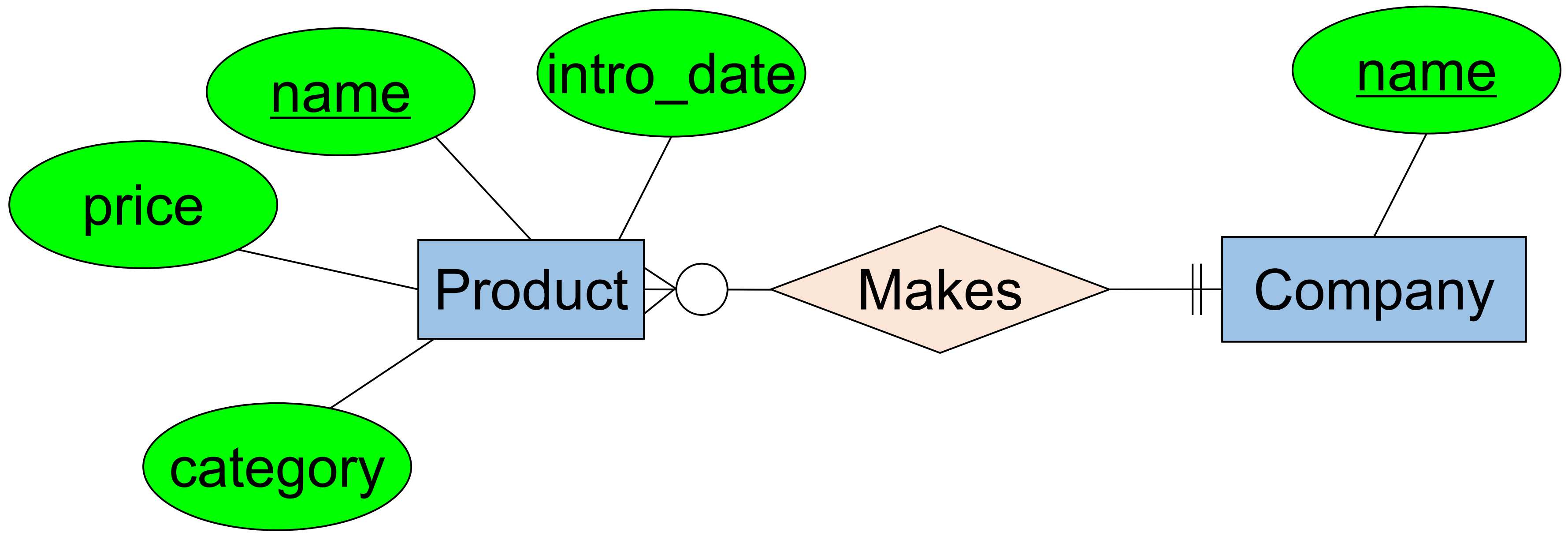
Exam 1 FM (Frequent Mistakes)

- Sort results if needed
- Include the number of output tuples as comment below query
- Start a fresh database before you start

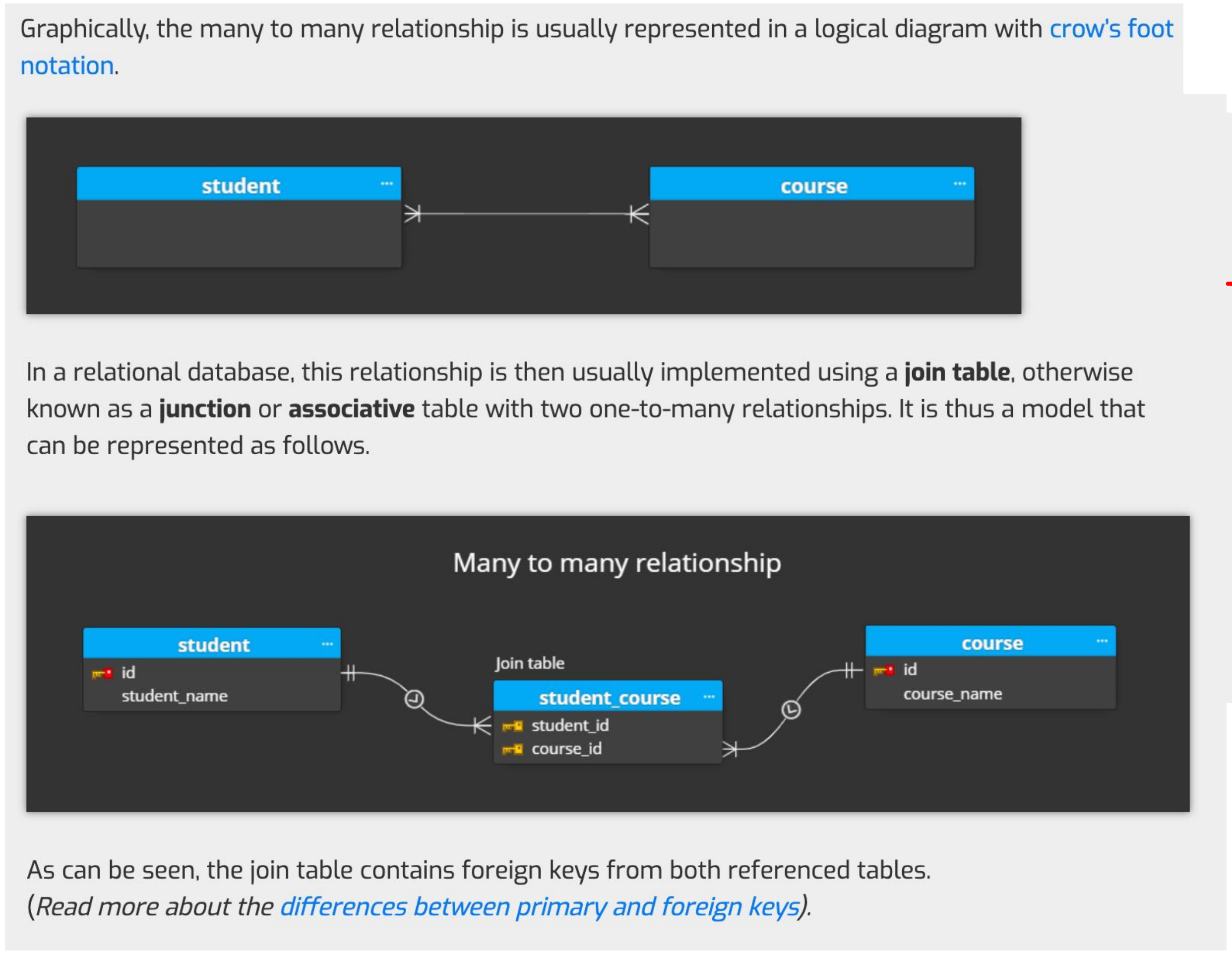
Practice: anything you would change?



Practice: now slightly better



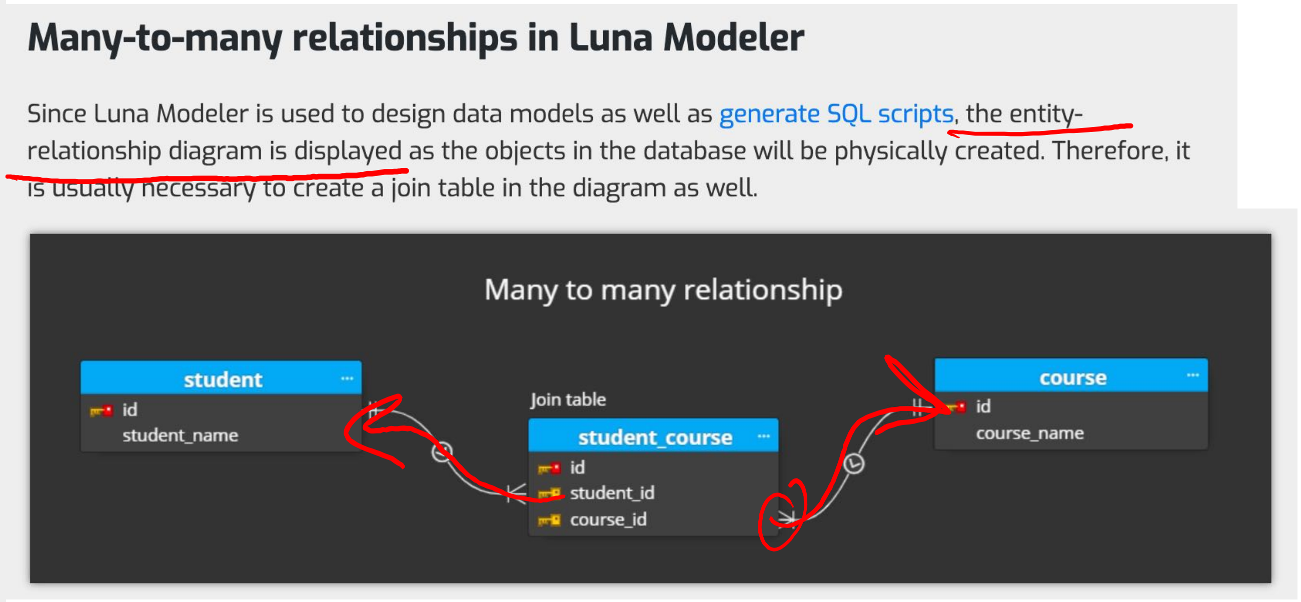
Don't mix ER notation with relational schemas!



Graphically, the many to many relationship is usually represented in a logical diagram with crow's foot notation.

In a relational database, this relationship is then usually implemented using a **join table**, otherwise known as a **junction** or **associative** table with two one-to-many relationships. It is thus a model that can be represented as follows.

As can be seen, the join table contains foreign keys from both referenced tables.
(*Read more about the differences between primary and foreign keys*).



Many-to-many relationships in Luna Modeler

Since Luna Modeler is used to design data models as well as generate SQL scripts, the entity-relationship diagram is displayed as the objects in the database will be physically created. Therefore, it is usually necessary to create a join table in the diagram as well.

That is imprecise notation:
There is not constraint in the database that every student needs to have at least one entry in student_course...

Source: https://www.datensen.com/blog/er-diagram/many-to-many-relationships/

Referential Integrity

Product

<u>PName</u>	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi

Company

<u>CName</u>	StockPrice	Country
GizmoWorks	25	USA
Canon	65	Japan
Hitachi	15	Japan

```
Insert into Product values ('NewToy', 12.99, 'Gadgets', null);
```

Will this work?

?

Referential Integrity

Product

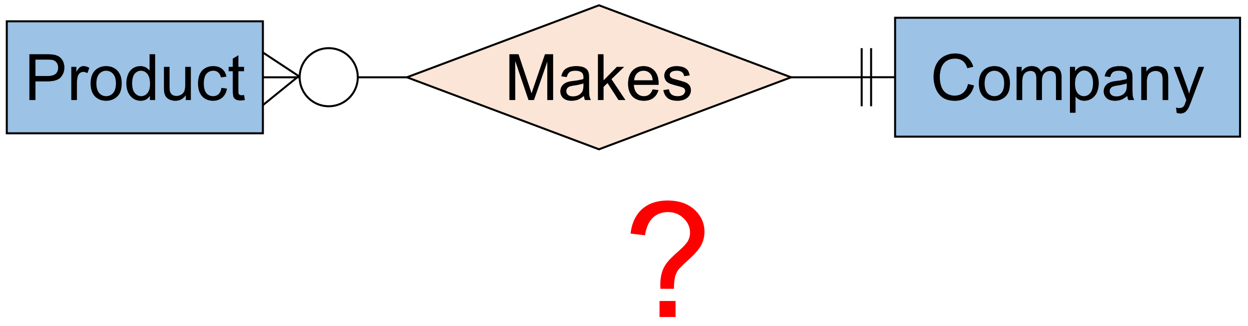
<u>PName</u>	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi

Company

<u>CName</u>	StockPrice	Country
GizmoWorks	25	USA
Canon	65	Japan
Hitachi	15	Japan

Insert into Product values ('NewToy', 12.99, 'Gadgets', null);

```
create table Product (
    PName char(20),
    Price decimal(9, 2),
    Category char(20),
    Manufacturer char(20),
PRIMARY KEY (PName),
FOREIGN KEY (Manufacturer)
    REFERENCES Company(CName) );
```



Referential Integrity

Product

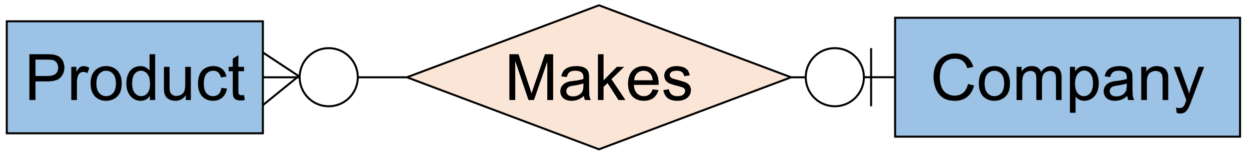
<u>PName</u>	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi

Company

<u>CName</u>	StockPrice	Country
GizmoWorks	25	USA
Canon	65	Japan
Hitachi	15	Japan

```
Insert into Product values ('NewToy', 12.99, 'Gadgets', null);
```

✓ ☺



```
create table Product (
    PName char(20),
    Price decimal(9, 2),
    Category char(20),
    Manufacturer char(20),
PRIMARY KEY (PName),
FOREIGN KEY (Manufacturer)
    REFERENCES Company(CName) );
```

Referential Integrity

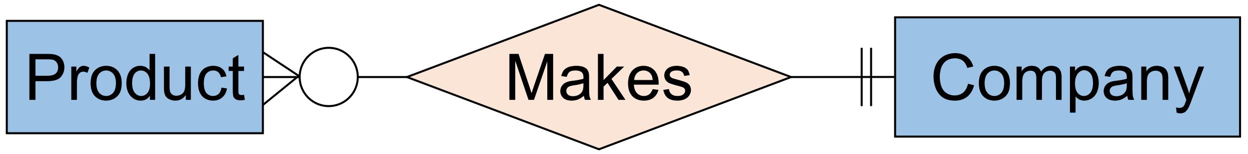
Product

<u>PName</u>	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi

Company

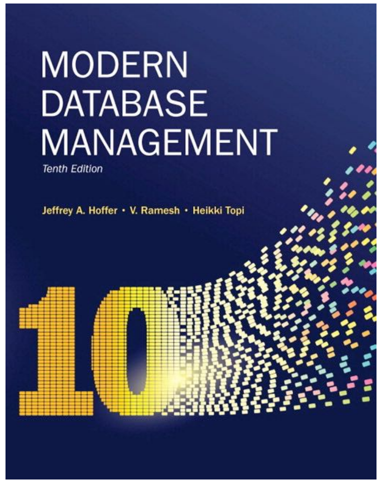
<u>CName</u>	StockPrice	Country
GizmoWorks	25	USA
Canon	65	Japan
Hitachi	15	Japan

~~Insert into Product values ('NewToy', 12.99, 'Gadgets', null);~~ ☹



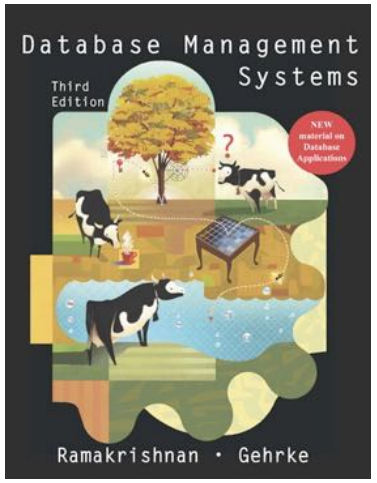
```
create table Product (
    PName char(20),
    Price decimal(9, 2),
    Category char(20),
    Manufacturer char(20) not null,
PRIMARY KEY (PName),
FOREIGN KEY (Manufacturer)
    REFERENCES Company(CName) );
```

Different sources, different notations

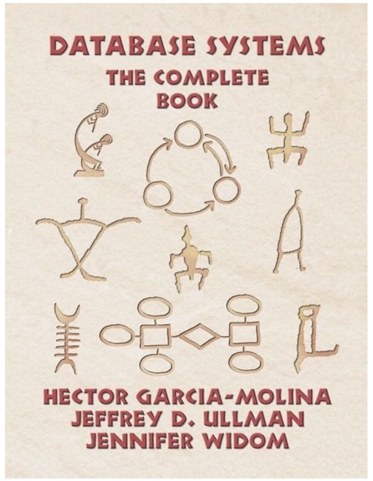

MODERN DATABASE MANAGEMENT
Tenth Edition
Jeffrey A. Hoffer • V. Ramesh • Heikki Topi
10

[Hoffer+'10]

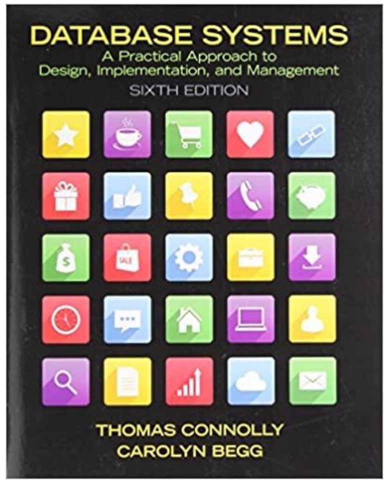
Crow foot



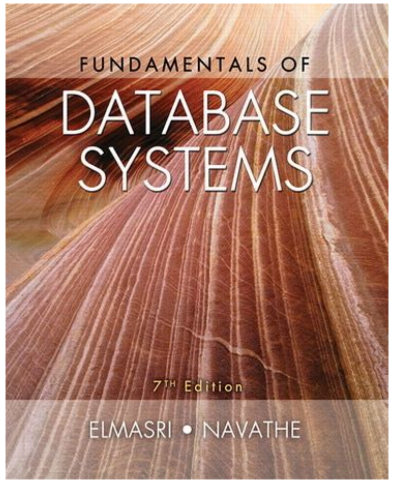
[Cow book'03]



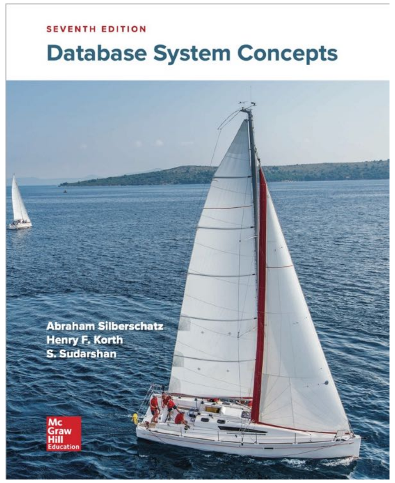
[Stanford book'08]


DATABASE SYSTEMS
A Practical Approach to Design, Implementation, and Management
SIXTH EDITION
THOMAS CONNOLLY
CAROLYN BEGG

[Connolly+'15]



[Elmasri+'15]



[Silberschatz+'20]

SDK arrows

[Hoffer+'10]: Hoffer, Ramesh, Topi. Modern Database Management, 10th ed, 2010.
https://www.pearson.com/us/higher-education/product/Hoffer-Modern-Database-Management-10th-Edition/9780136088394.html

[Cow book'03]: Ramakrishnan, Gehrke, Database Management Systems, 3rd ed, 2003. http://pages.cs.wisc.edu/~dbbook/

[Stanford book'08]: Garcia-Molina, Ullman, Widom. Database Systems: The Complete Book, 2nd ed, 2008. http://infolab.stanford.edu/~ullman/dscb.html

[Connolly+'15]: Connolly, Begg. Database systems: A practical approach to design, implementation, and management, 6th ed, 2015.
https://www.pearson.com/us/higher-education/program/Connolly-Database-Systems-A-Practical-Approach-to-Design-Implementation-and-Management-6th-Edition/PGM116956.html

[Elmasri+'15]: Elmasri, Navathe. Fundamentals of Database Systems, 7th ed, 2015.
https://www.pearson.com/us/higher-education/program/Elmasri-Fundamentals-of-Database-Systems-7th-Edition/PGM189052.html

[Silberschatz+'20]: Silberschatz, Korth, Sudarshan. Database system concepts, 7th ed, 2020. https://www.db-book.com/db7

Notations for binary one-to-many relationships

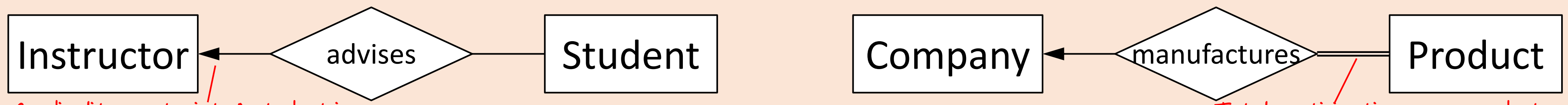
10/24/2022

if you are overwhelmed by the different notations, just use the one from our textbook

Every student is advised by maximum 1 instructor.
An instructor may advise 0, 1 or more students.

Every product is manufactured by exactly 1 company.
A company may manufacture 0, 1 or more products.

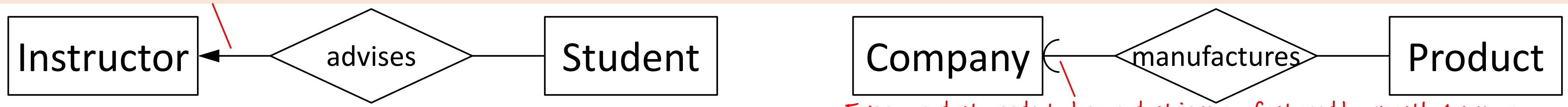
SDK [Silberschatz+'20]



Cardinality constraint: A student is advised by max. 1 instructor (injective)

Total participation: every product needs to be manufactured

[Stanford book'03]
also used by Gradiance

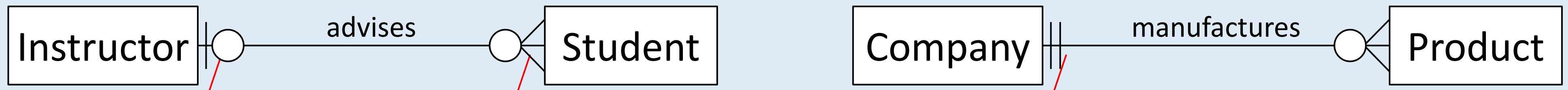


Every product needs to be ~~product~~ is manufactured by exactly 1 company. (referential integrity constraint).

Crow's foot
[Hoffer+'10]

Most often used in practice

look across notation

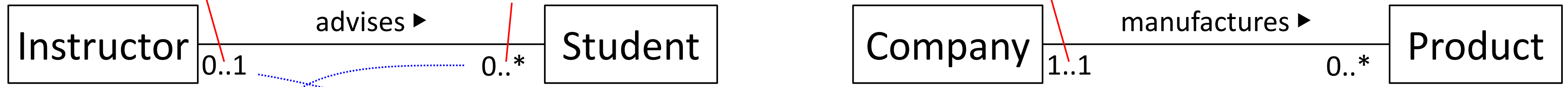


Students are advised by min 0 and max 1 instructors

Instructors advise an unrestricted number of students

Products are mapped to min 1 and max 1 companies

UML
[Connolly+'15]



also (0,1)

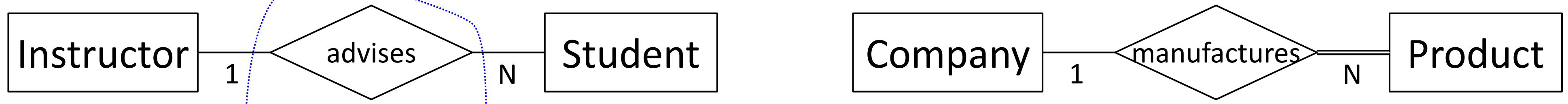
also (0,N)

also (1,1)

also: sometimes participation not shown

[Elmasri+'15]

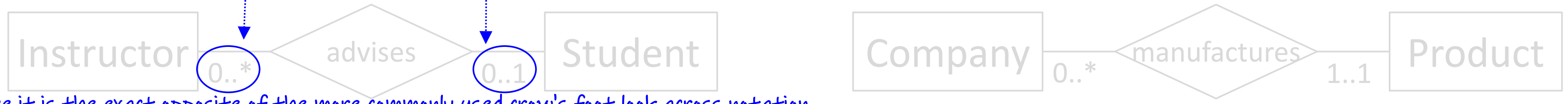
look across for cardinality, same-side for participation



Avoid (min-max) !!!

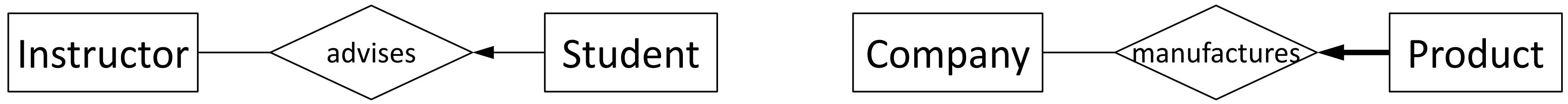
(min-max)
[Elmasri+'15]

same-side notation



strongly discouraged since it is the exact opposite of the more commonly used crow's foot look across notation

[Cow book'03]



Notations for entities and their attributes

10/30/2022

Chen "bubble" variants

"Boxed" variants

[Silberschatz+'20]

[Stanford book'03]
also used by Gradiance

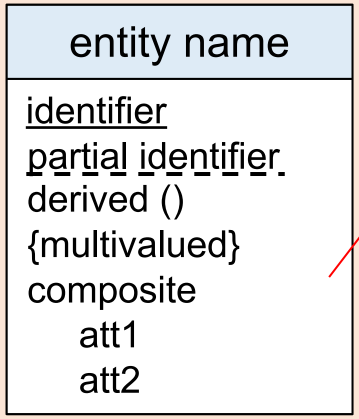
Crow's foot
[Hoffer+'10]

UML
[Connolly+'15]

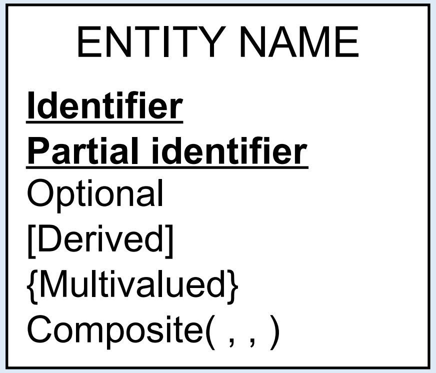
[Elmasri+'15]

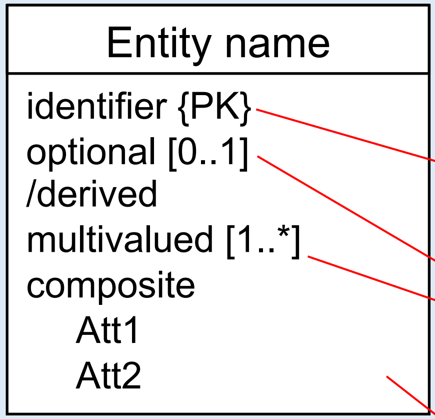
[Cow book'03]

if you are overwhelmed by the different notations, just use the one from our textbook



no particular notation for optional attributes (vs. mandatory)



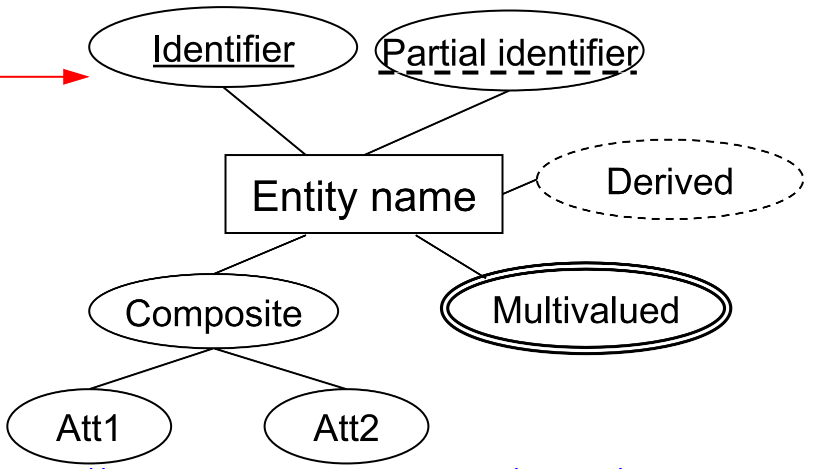


sometimes {id}

unified concept for mandatory, optional, multivalued

no notion of weak or strong entity in UML notation

In practice you will likely see some variant of all these three variants



Notations for attributes on binary relationships

10/24/2022

"Bubble variants"

"Box variants"

[Silberschatz+'20]

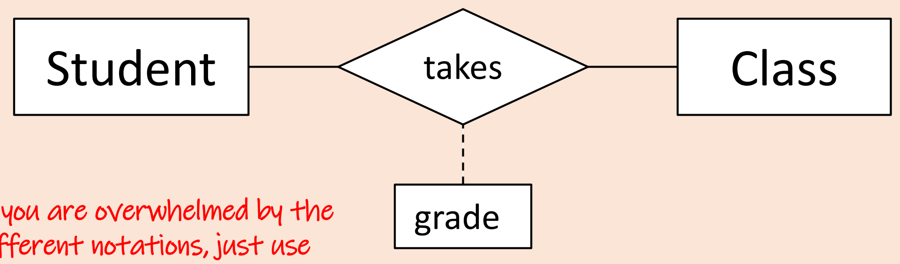
[Stanford book'03]
also used by Gradiance

Crow's foot
[Hoffer+'10]

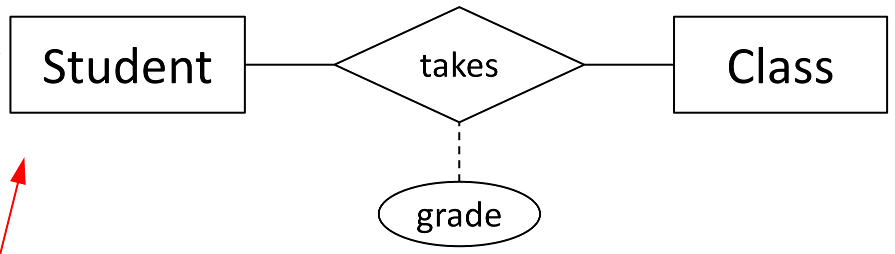
UML
[Connolly+'15]

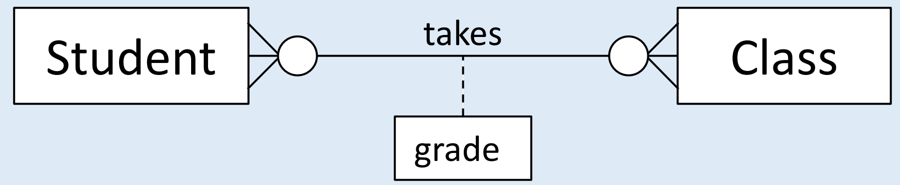
[Elmasri+'15]

[Cow book'03]

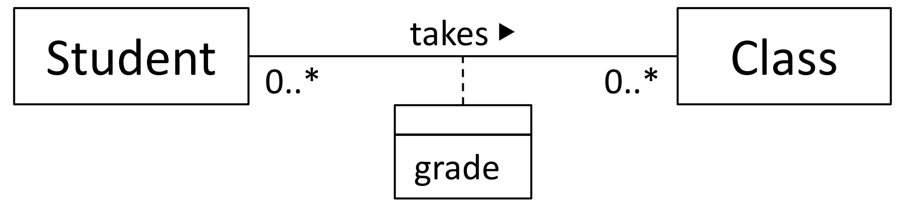


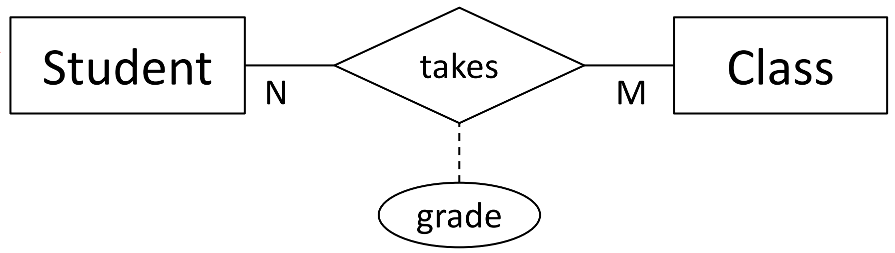
if you are overwhelmed by the different notations, just use the one from our textbook



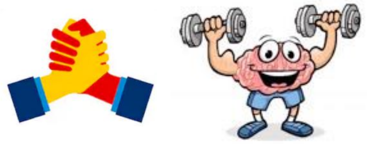


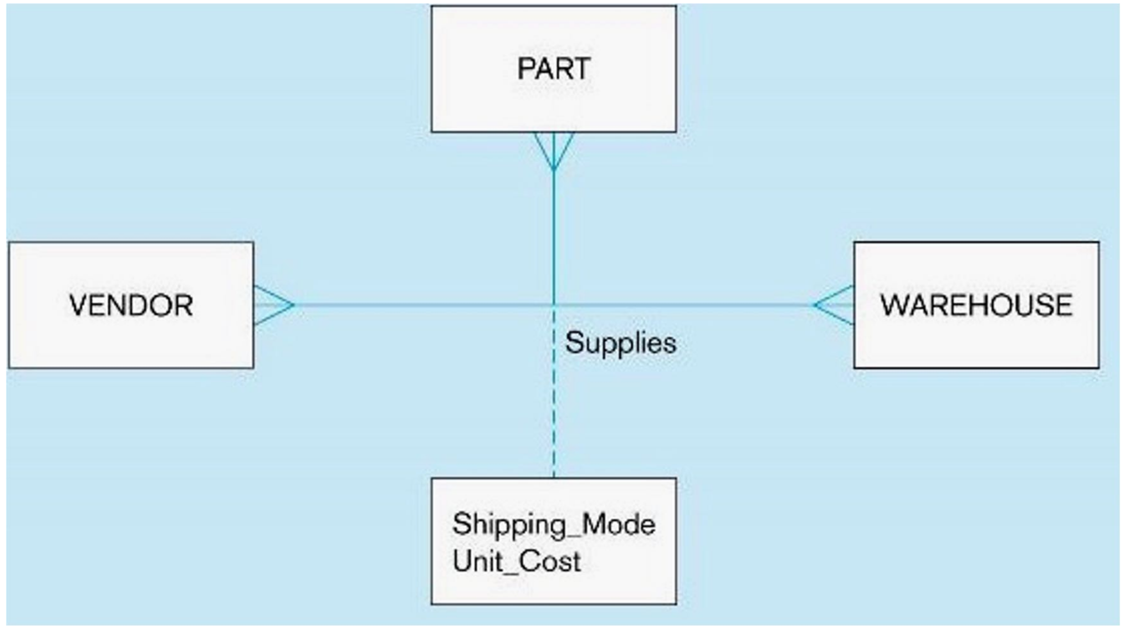
In practice you will likely see some variant of all these three variants





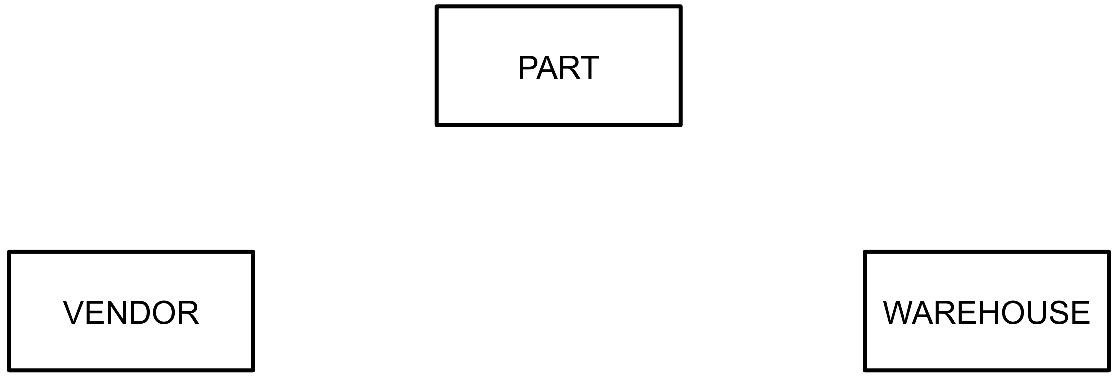
Ternary vs. binary relationships





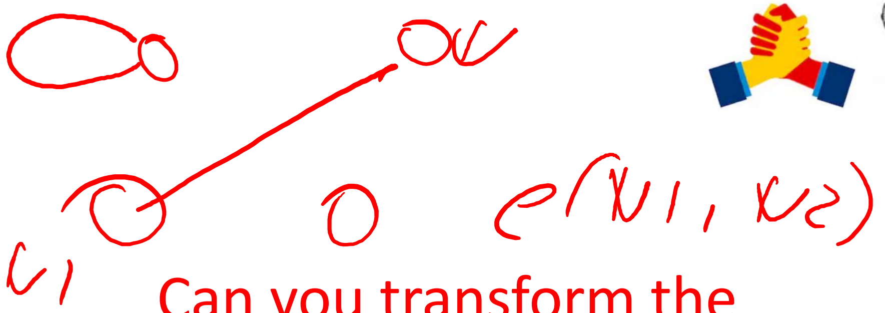
Can you transform the ternary relationship into a number of binary relationships?

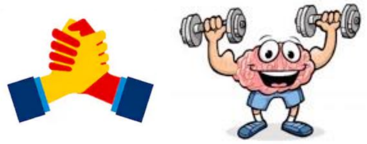
?

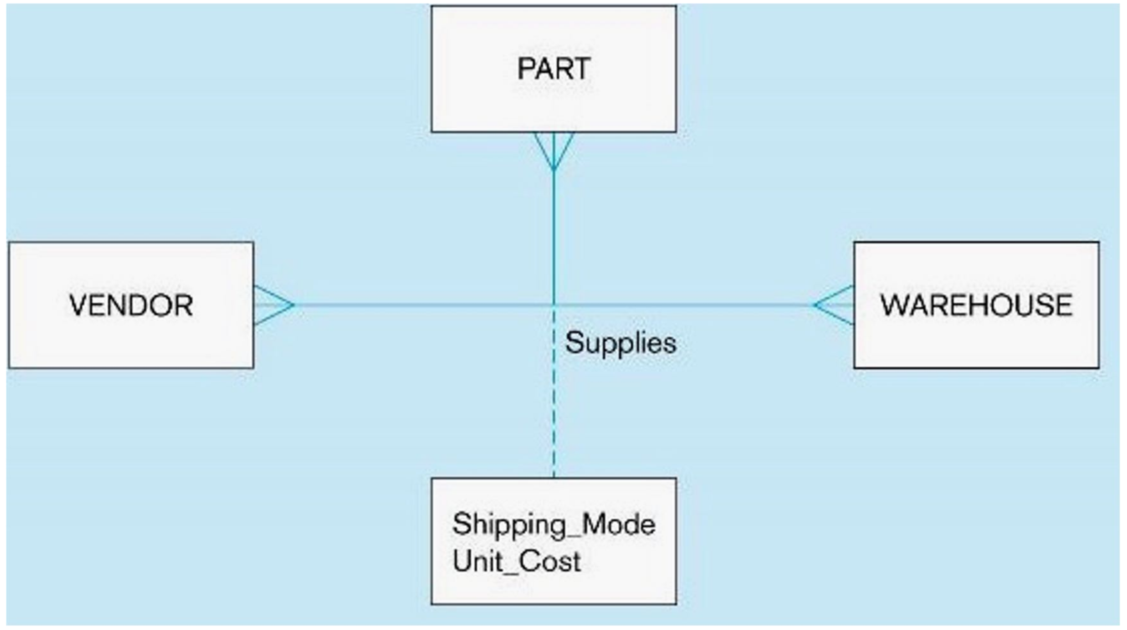


Source: Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010. Fig 2.18

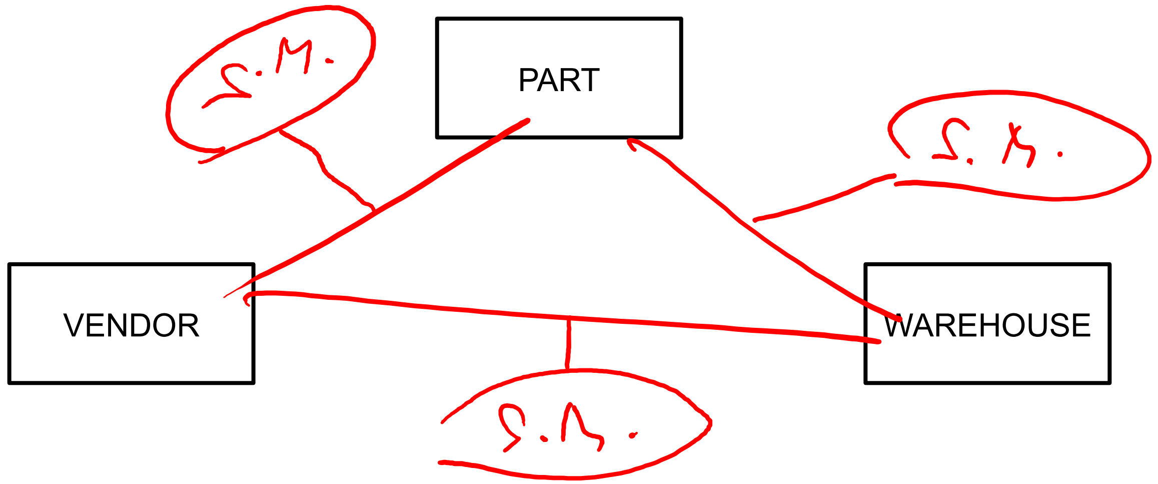
Ternary vs. binary relationships







Can you transform the ternary relationship into a number of binary relationships?

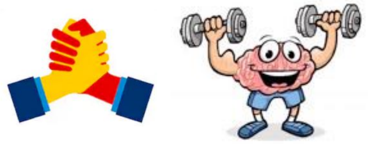


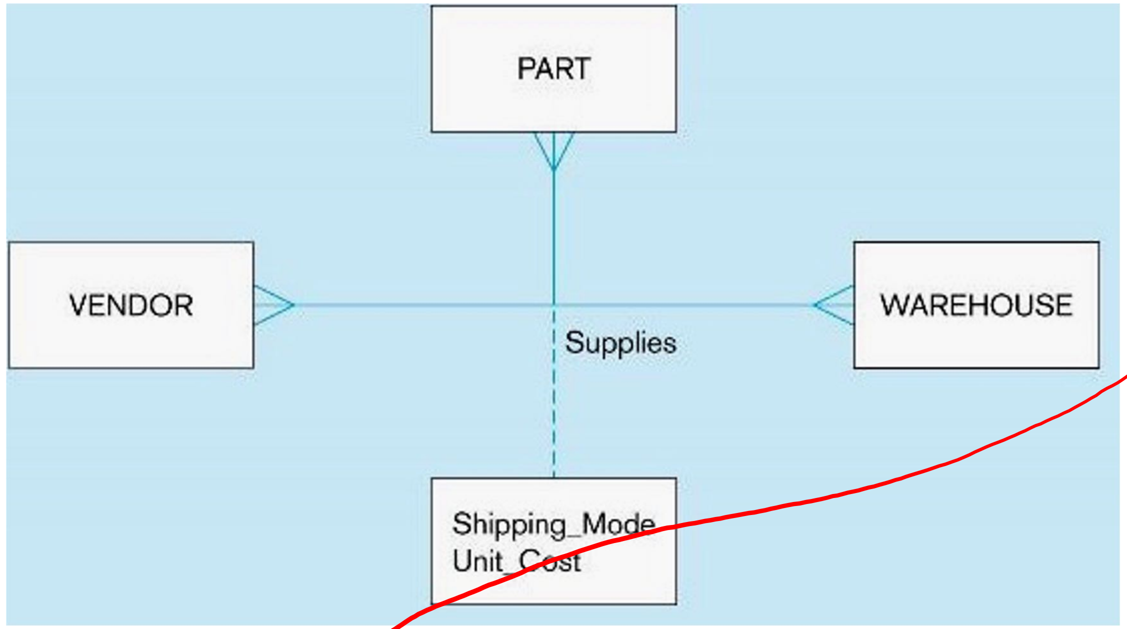
Part 1: No! E.g., where to put the attribute shipping_mode?

!

Source: Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010. Fig 2.18

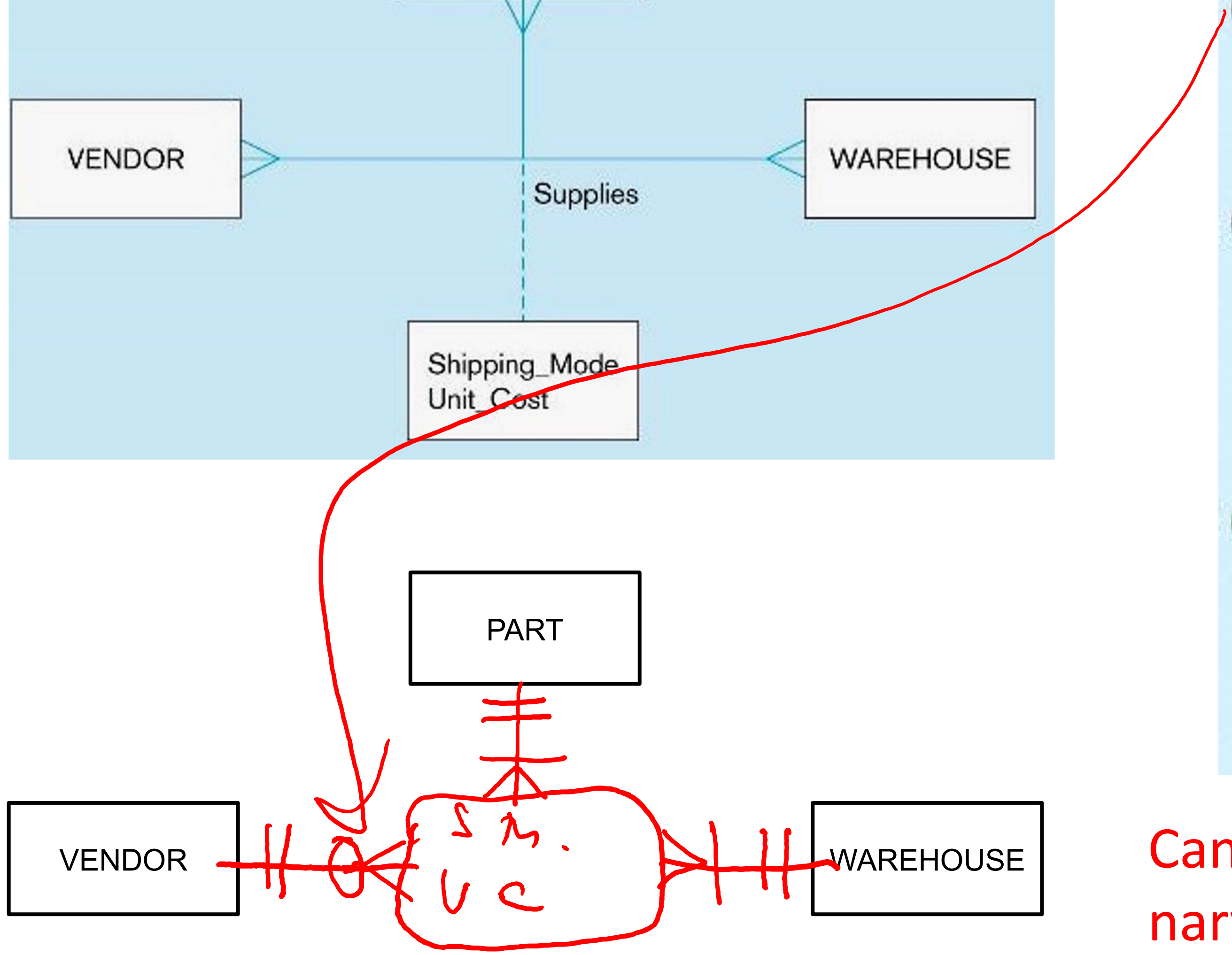
Ternary vs. binary relationships





Business Rules

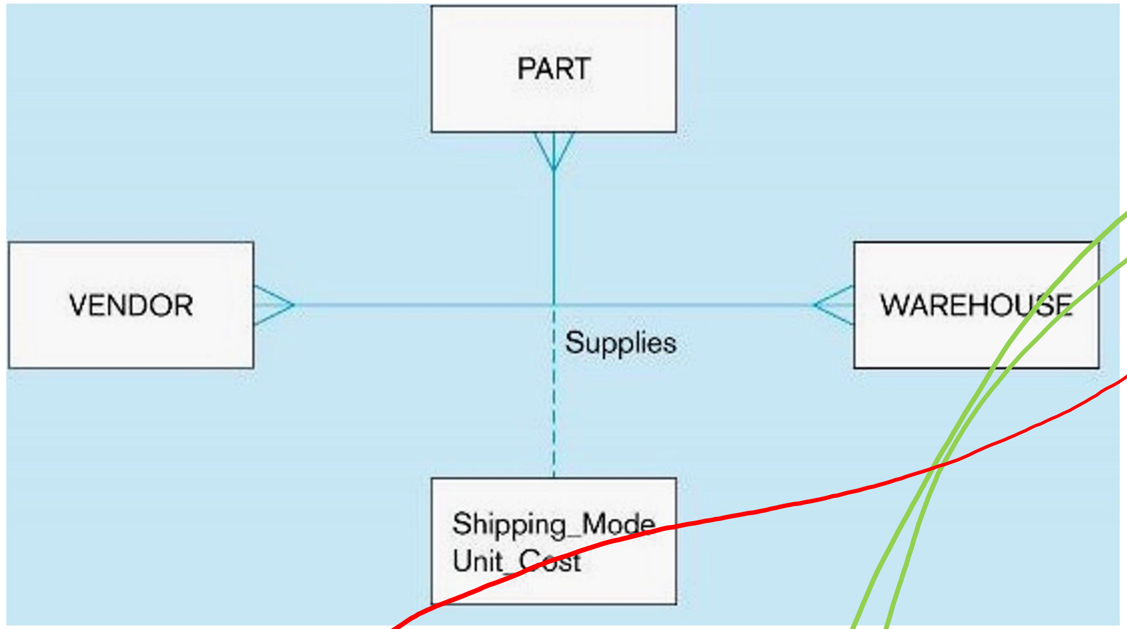
1. Each vendor can supply many parts to any number of warehouses but need not supply any parts.
2. Each part can be supplied by any number of vendors to more than one warehouse, but each part must be supplied by at least one vendor to a warehouse.
3. Each warehouse can be supplied with any number of parts from more than one vendor, but each warehouse must be supplied with at least one part.



Can you transform the ternary relationship to an associative entity?

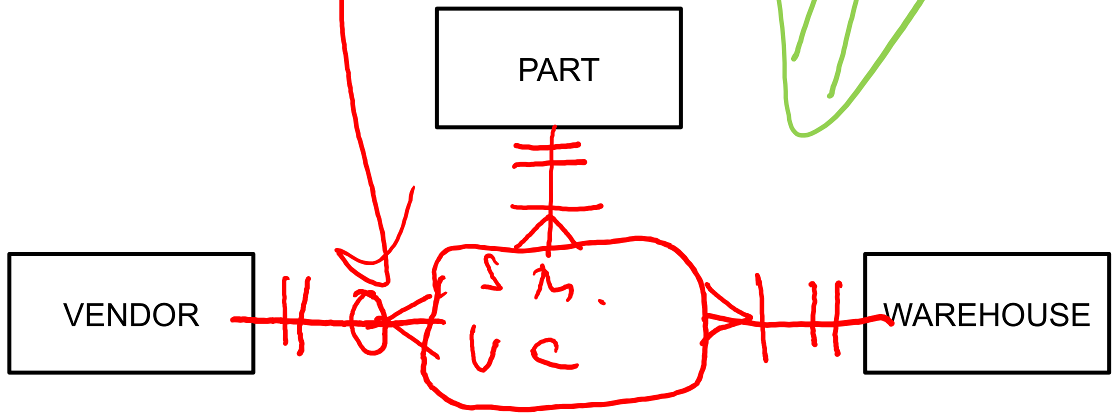
?

Ternary vs. binary relationships



Business Rules

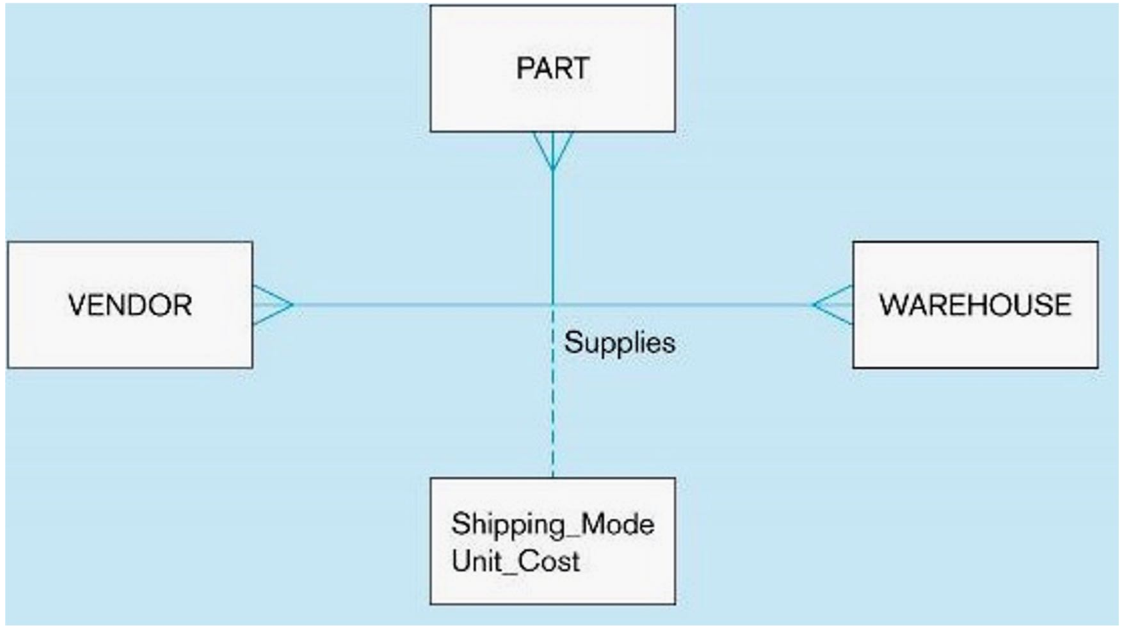
1. Each vendor can supply many parts to any number of warehouses but need not supply any parts.
2. Each part can be supplied by any number of vendors to more than one warehouse, but each part must be supplied by at least one vendor to a warehouse.
3. Each warehouse can be supplied with any number of parts from more than one vendor, but each warehouse must be supplied with at least one part.

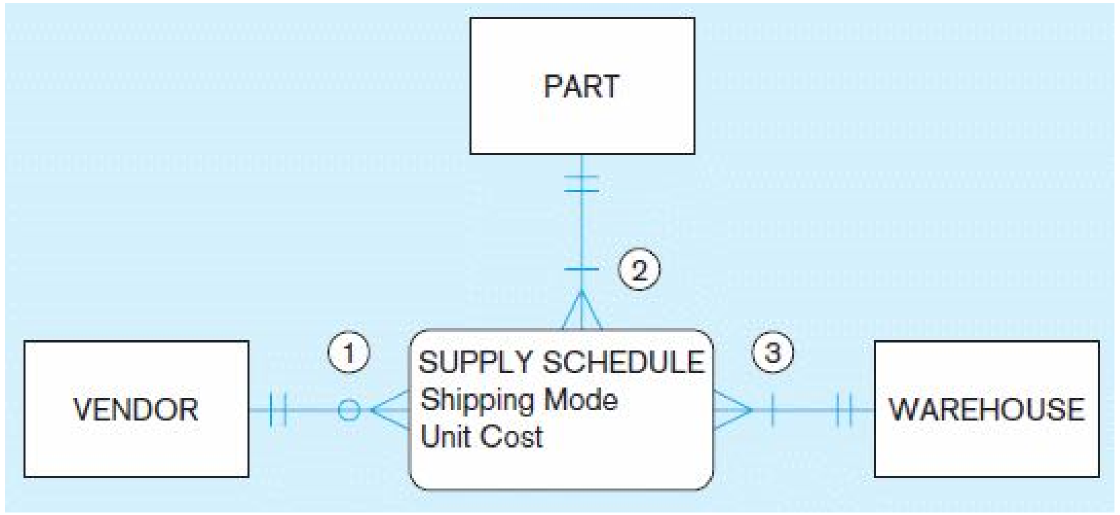


Can you transform the ternary relationship to an associative entity?

Source: Hoffer, Ramesh, Topi, "Modern database management," 10th ed, 2010. Fig 2.18
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

Cardinality + participation in a ternary relationship





Business Rules

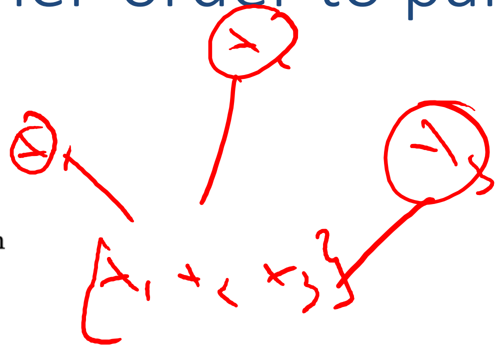
1. Each vendor can supply many parts to any number of warehouses but need not supply any parts.
2. Each part can be supplied by any number of vendors to more than one warehouse, but each part must be supplied by at least one vendor to a warehouse.
3. Each warehouse can be supplied with any number of parts from more than one vendor, but each warehouse must be supplied with at least one part.

Can you transform the ternary relationship to an associative entity?

Transforming higher-order to pairwise relationships beyond databases

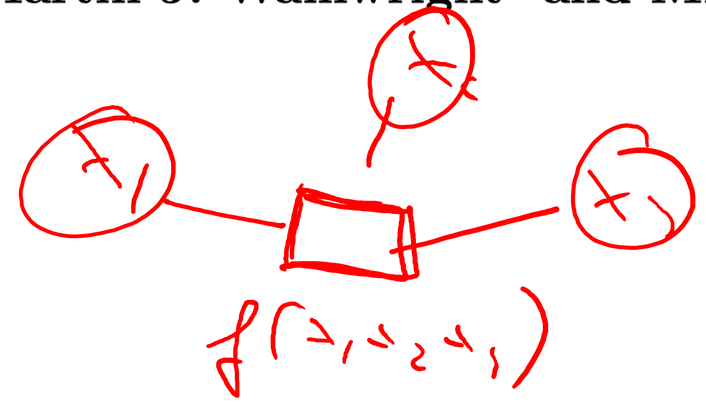
Beyond class topic

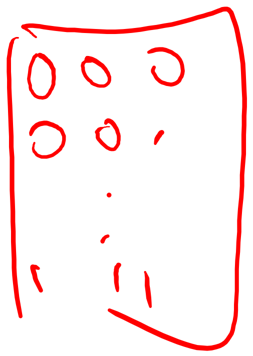
Foundations and Trends® in
Machine Learning
Vol. 1, Nos. 1–2 (2008) 1–305
© 2008 M. J. Wainwright and M. I. Jordan
DOI: 10.1561/2200000001



Graphical Models, Exponential Families, and Variational Inference

Martin J. Wainwright[1] and Michael I. Jordan[2]





E.3 Conversion to a Pairwise Markov Random Field

In this appendix, we describe how any Markov random field with discrete random variables can be converted to an equivalent pairwise form (i.e., with interactions only between pairs of variables). To illustrate the general principle, it suffices to show how to convert a compatibility function ψ_{123} defined on a triplet $\{x_1, x_2, x_3\}$ of random variables into a pairwise form. To do so, we introduce an auxiliary node A, and associate with it random variable z that takes values in the Cartesian product space $\mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3$. In this way, each configuration of z can be identified with a triplet (z_1, z_2, z_3). For each $s \in \{1,2,3\}$, we define a pairwise compatibility function ψ_{As}, corresponding to the interaction between z and x_s, by $\psi_{As}(z, x_s) := [\psi_{123}(z_1, z_2, z_3)]^{1/3}\mathbb{I}[z_s = x_s]$. (The purpose of the $1/3$ power is to incorporate ψ_{123} with the correct exponent.) With this definition, it is straightforward to verify that the equivalence

$$\psi_{123}(x_1, x_2, x_3) = \sum_z \prod_{s=1}^{3} \psi_{As}(z, x_s)$$

holds, so that our augmented model faithfully captures the interaction defined on the triplet $\{x_1, x_2, x_3\}$.

Source: Wainwright, Jordan, "Graphical models, exponential families, and variational inference," Foundations and Trends in ML, 2008. http://dx.doi.org/10.1561/2200000001

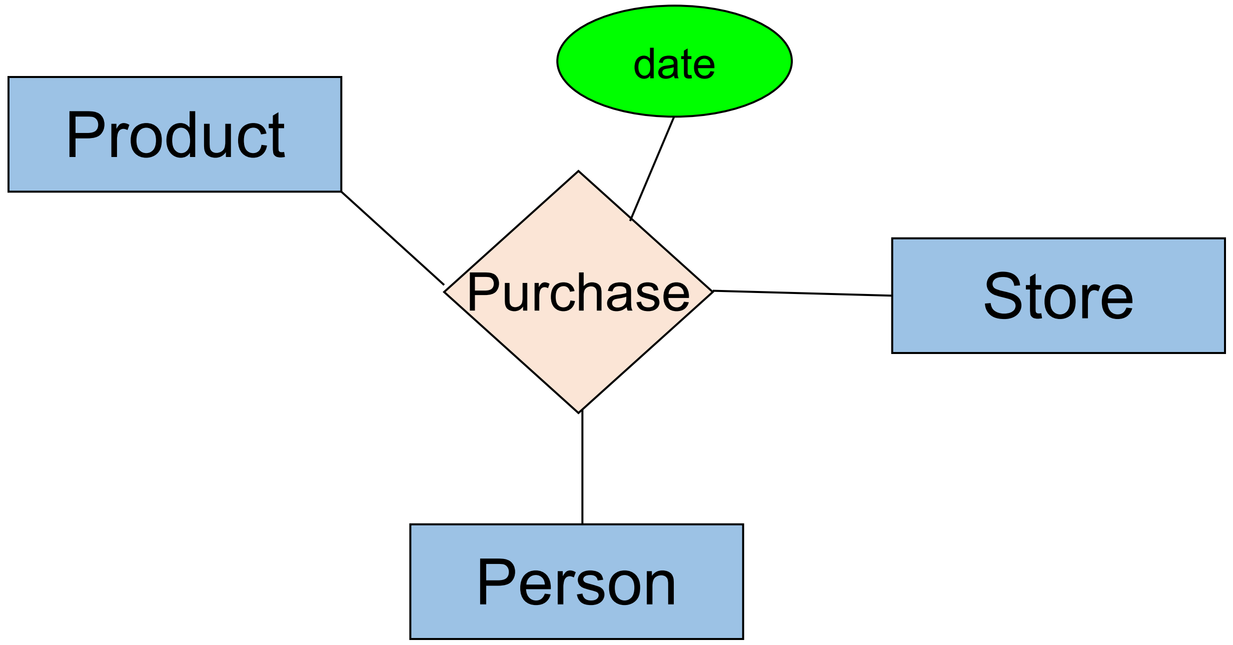
Multi-way Relationships

How do we model a purchase relationship between buyers, products and stores (assume 1 product per purchase)?

?

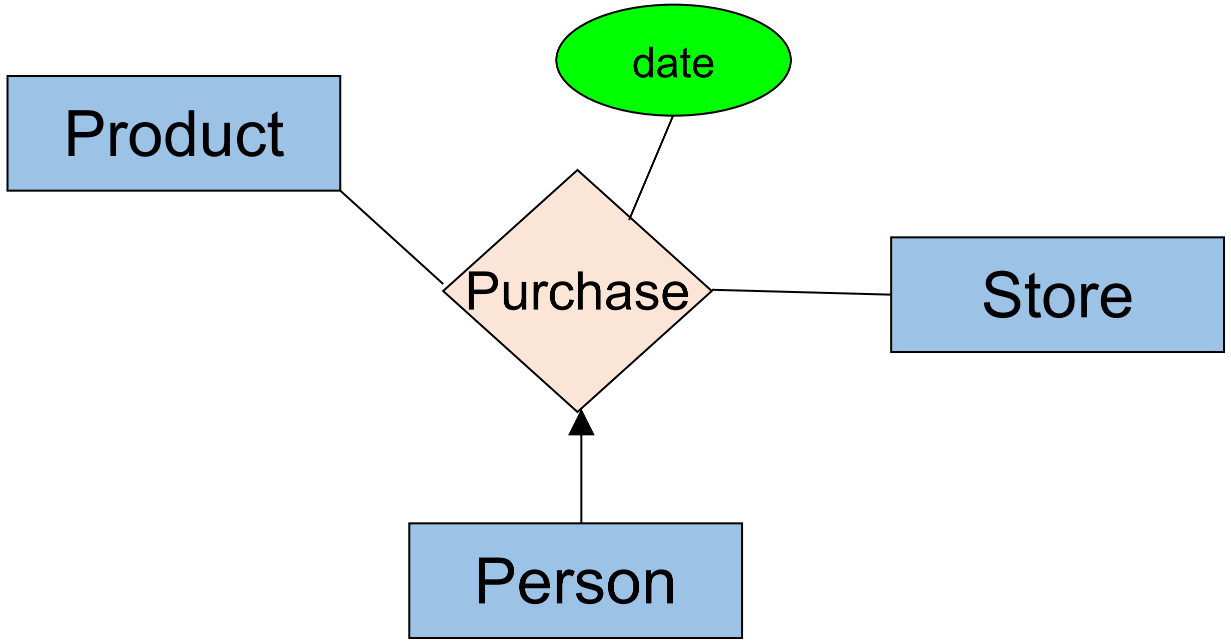
Multi-way Relationships

How do we model a purchase relationship between buyers, products and stores (assume 1 product per purchase)?



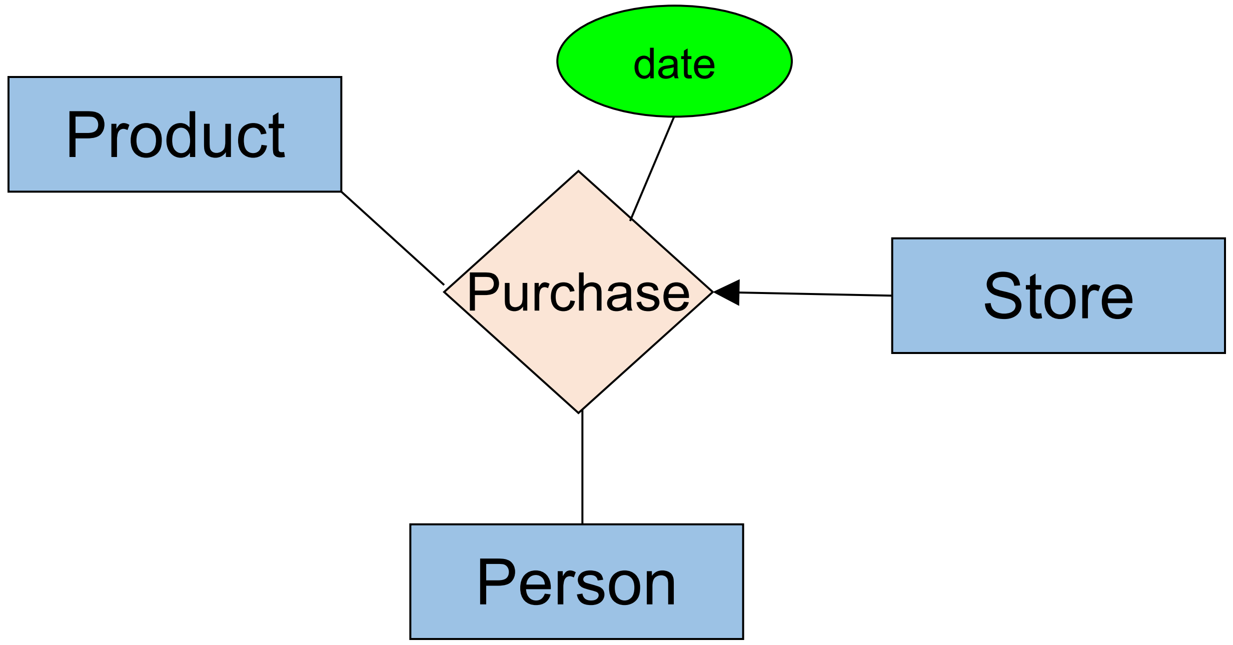
Arrows in Multiway Relationships (Cow book variant)

Q: What does the arrow mean ?



Arrows in Multiway Relationships

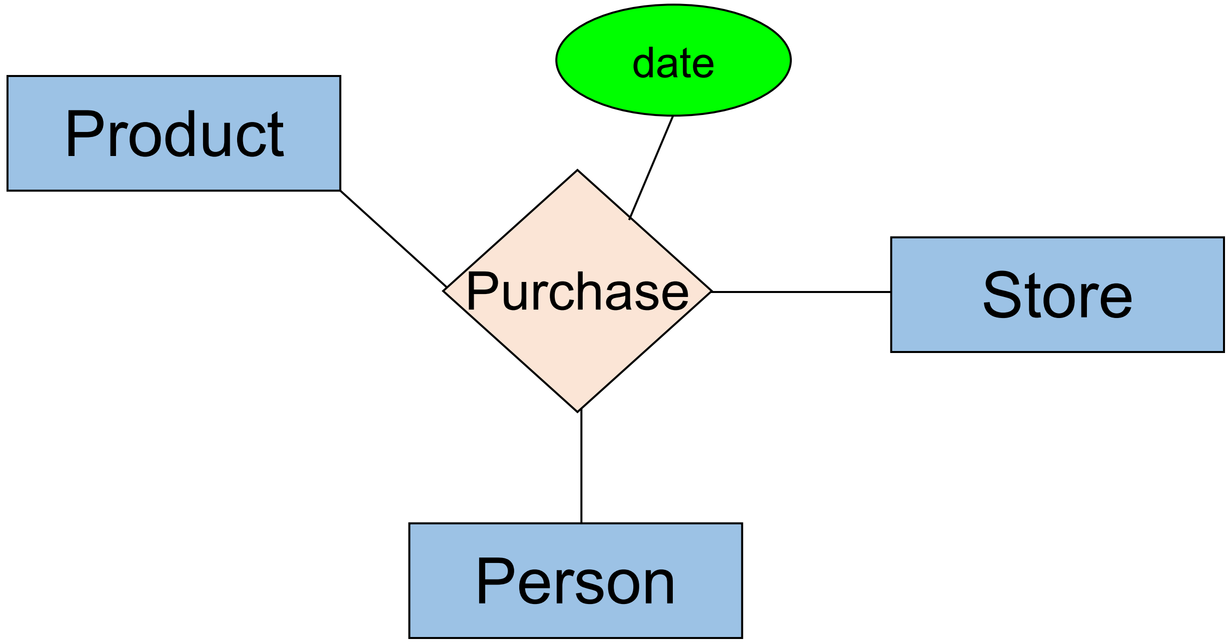
Q: What does the arrow mean ?



Arrows in Multiway Relationships

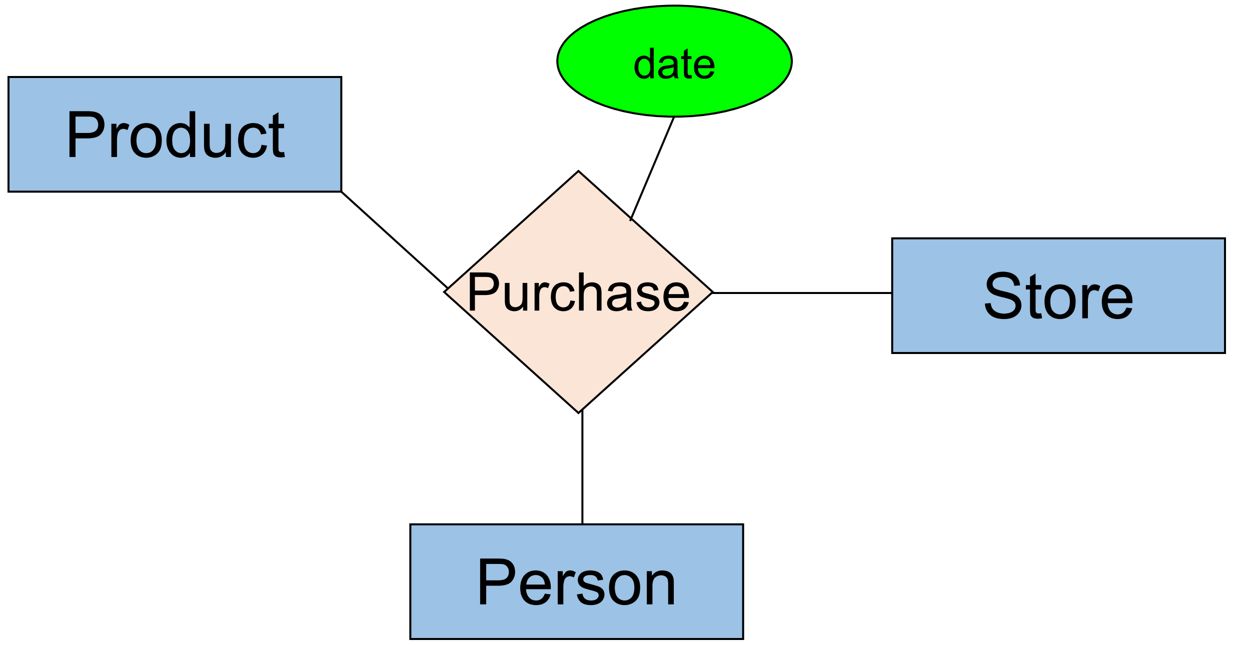
Q: How do we say that every person shops in at most one store ?

?



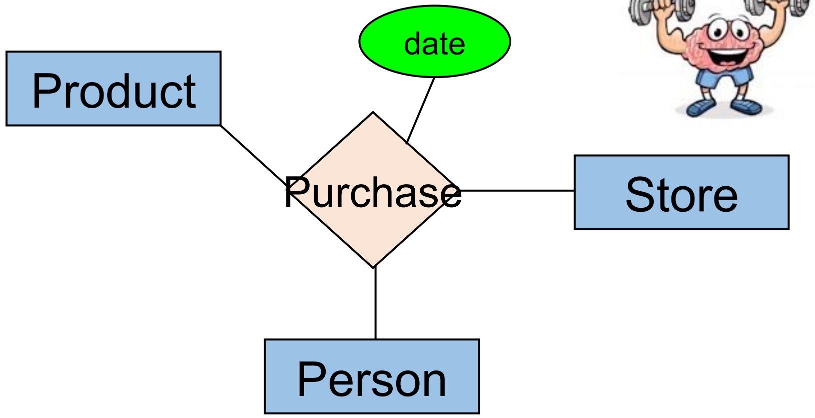
Arrows in Multiway Relationships

Q: How do we say that every person shops in at most one store ?



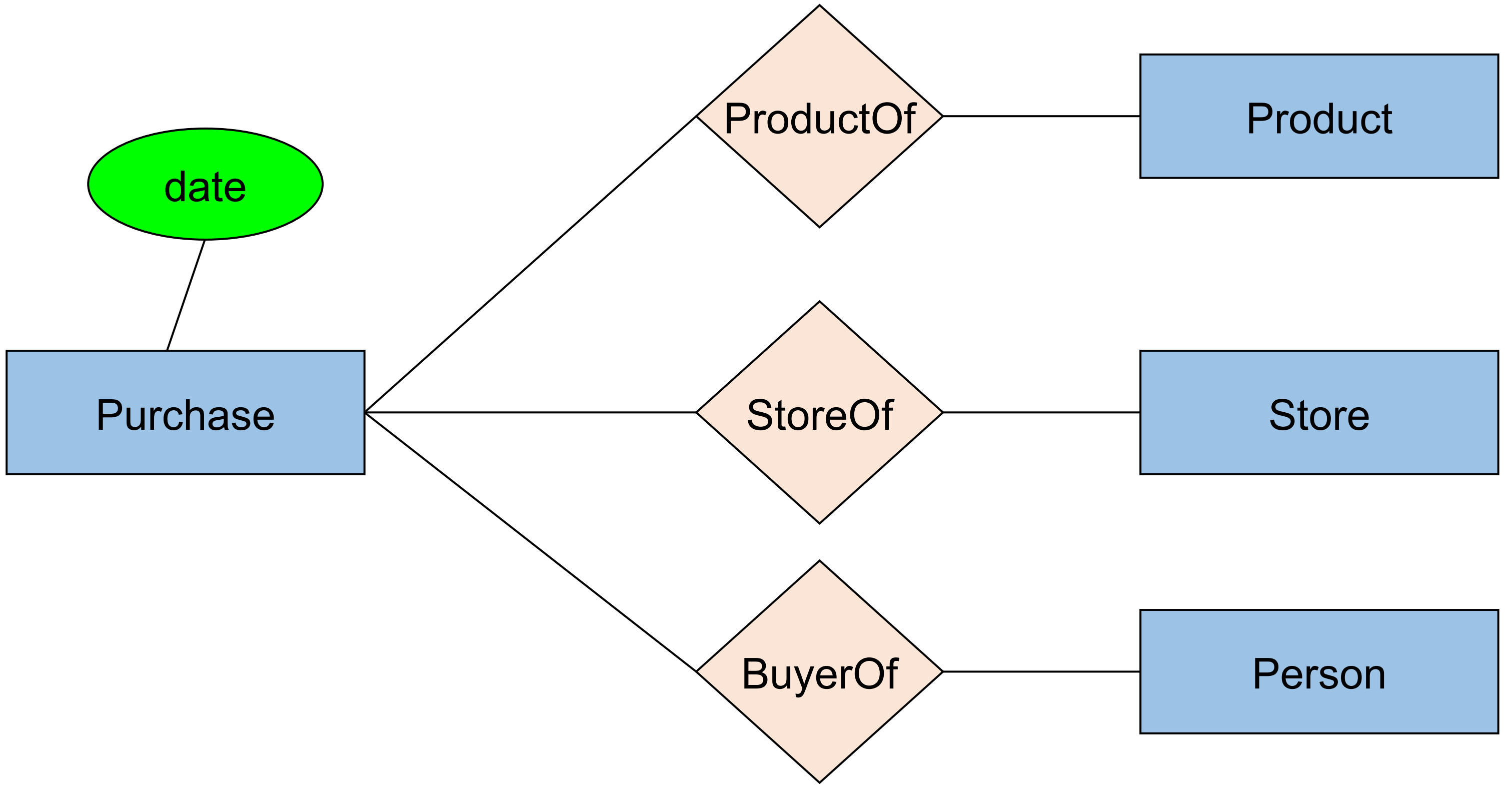
Cannot be done ☹ This is the best approximation.

Converting Multi-way Relationships

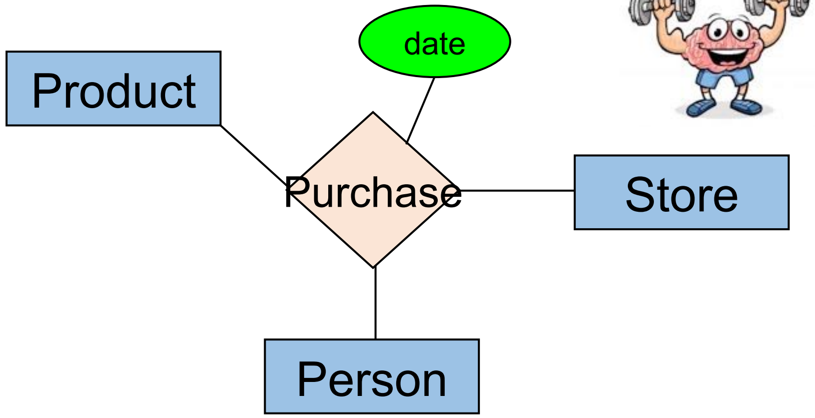


From what we had previously to this – what did we do?

?

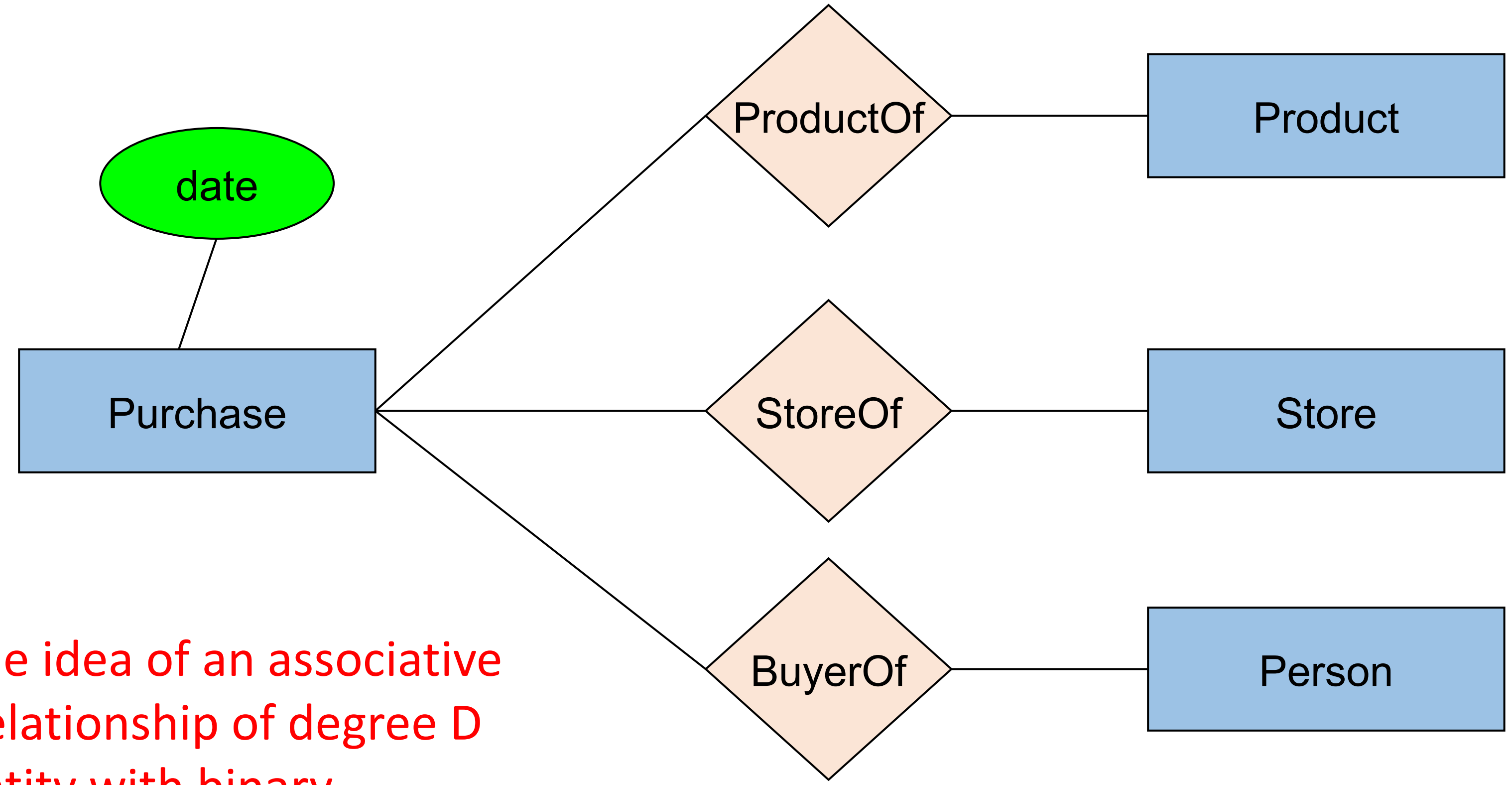


Converting Multi-way Relationships



From what we had previously to this – what did we do?

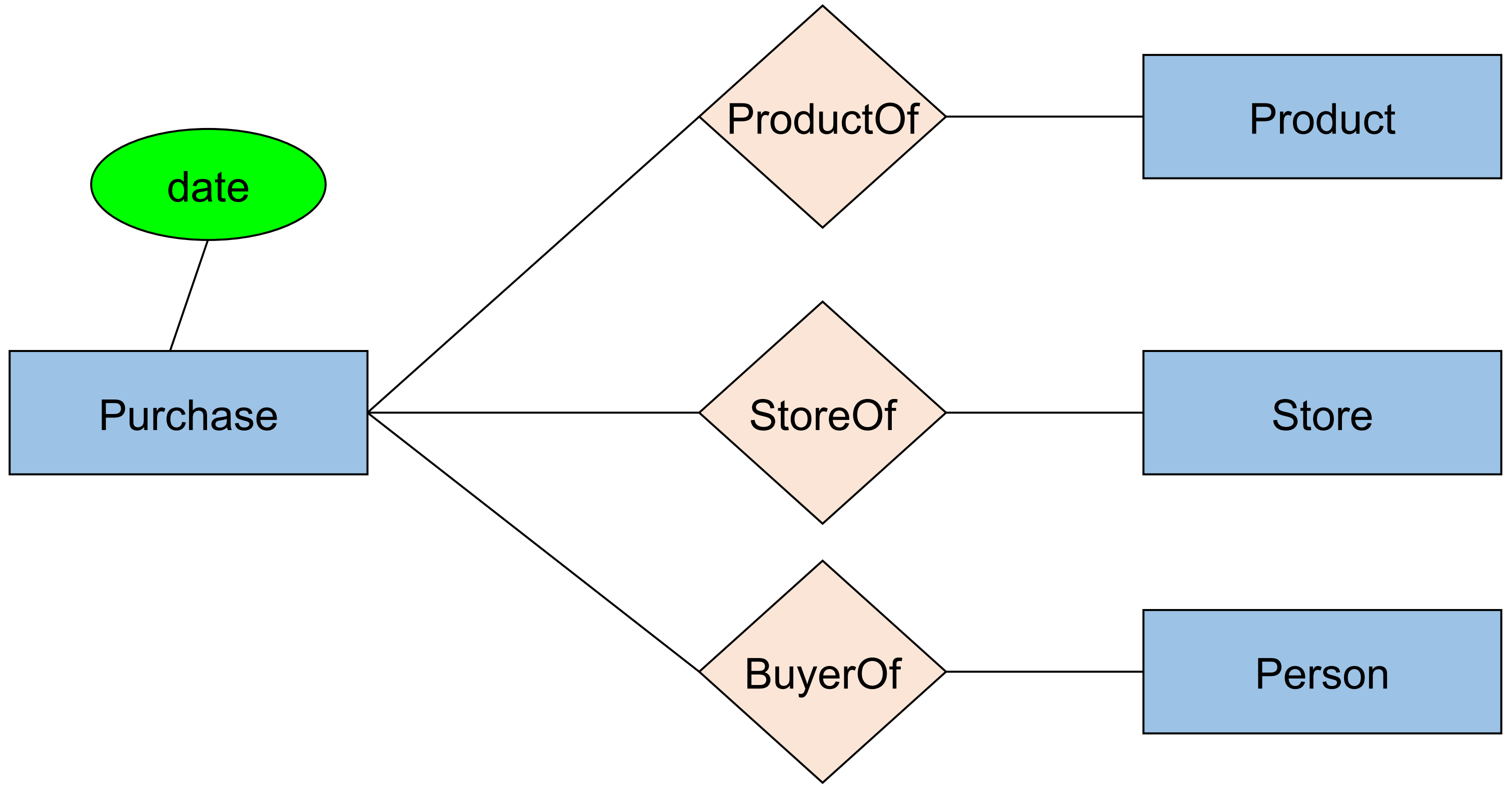
?



Notice that this is the same idea of an associative entity: you can take any relationship of degree D and create a new D+1th entity with binary relationships to all others .

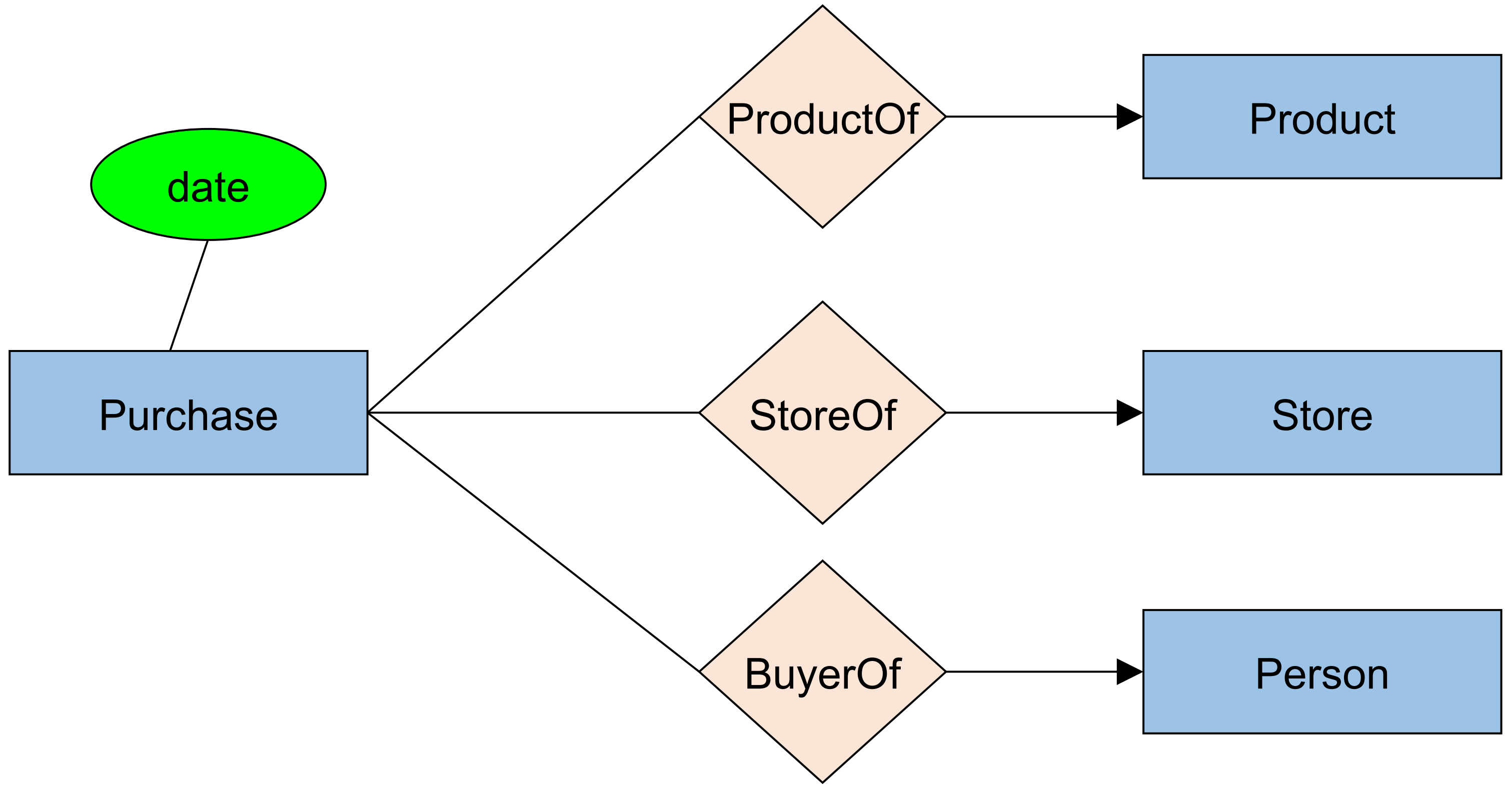
Converting Multi-way Relationships to New Entity + Binary Relationships

What cardinality constraints should we add here ?



Converting Multi-way Relationships to New Entity + Binary Relationships

What cardinality constraints should we add here

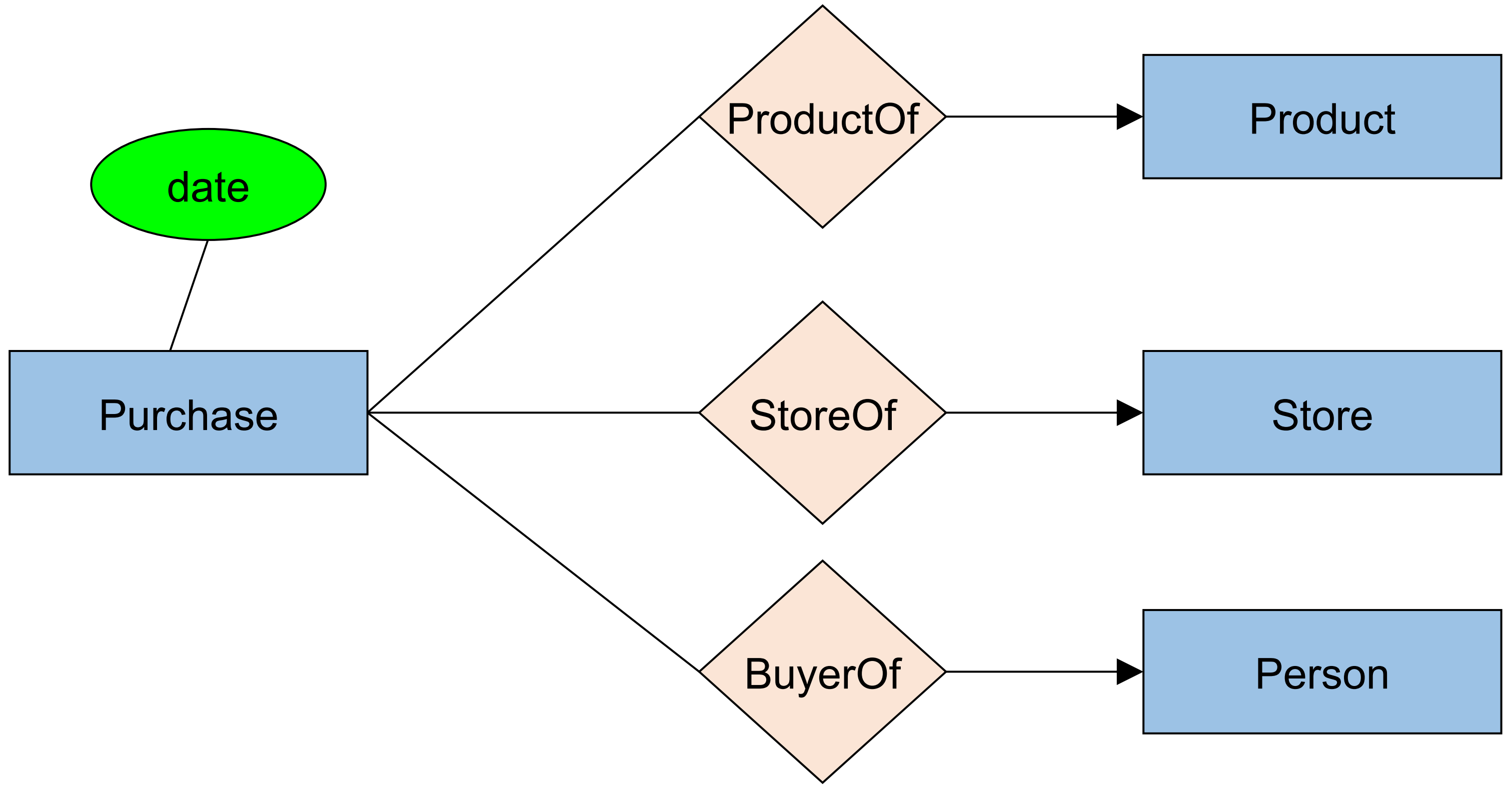


Converting Multi-way Relationships to New Entity + Binary Relationships

What cardinality constraints should we add here

What participation constraints should we add here

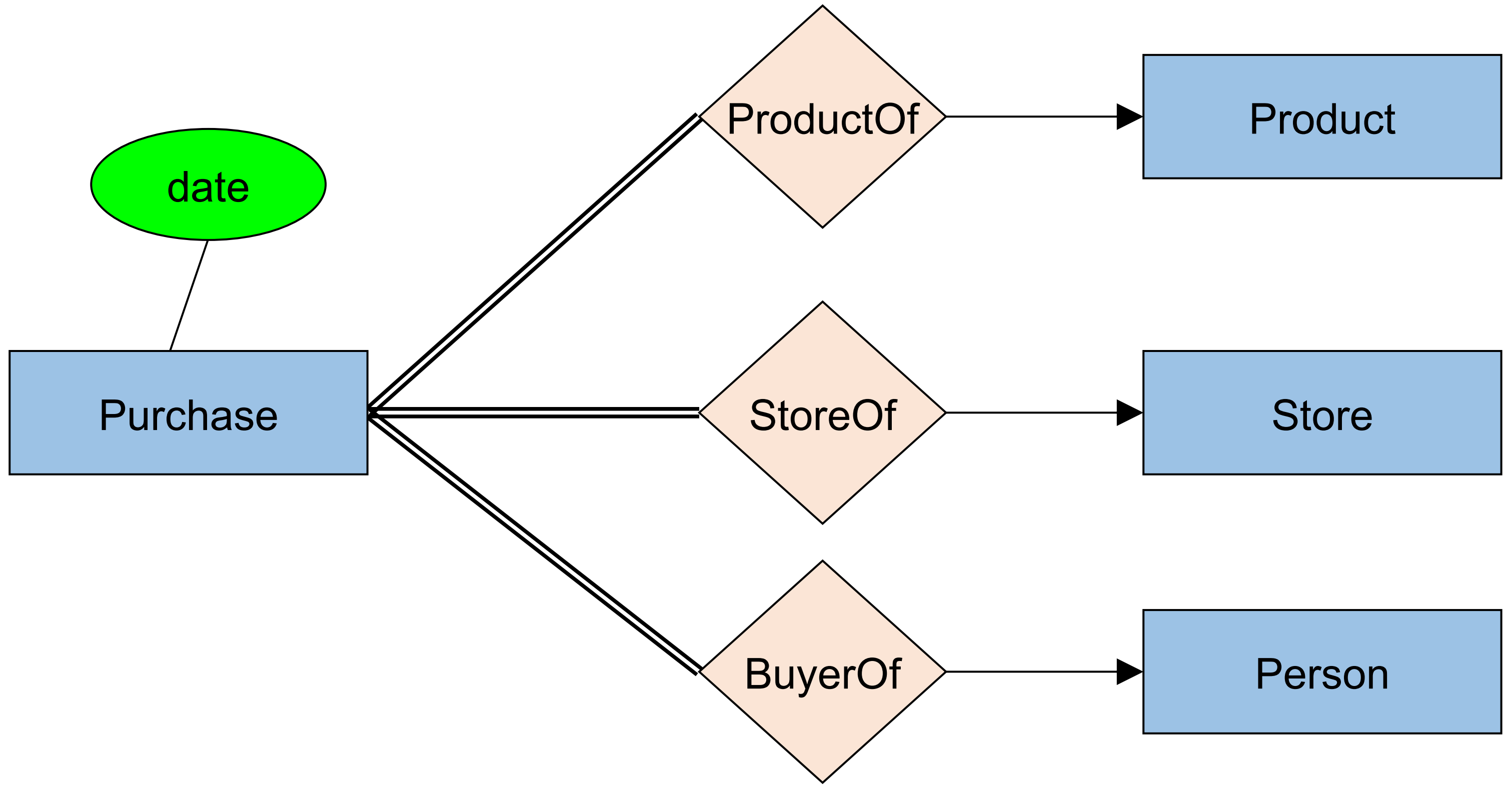
?



Converting Multi-way Relationships to New Entity + Binary Relationships

What cardinality constraints should we add here

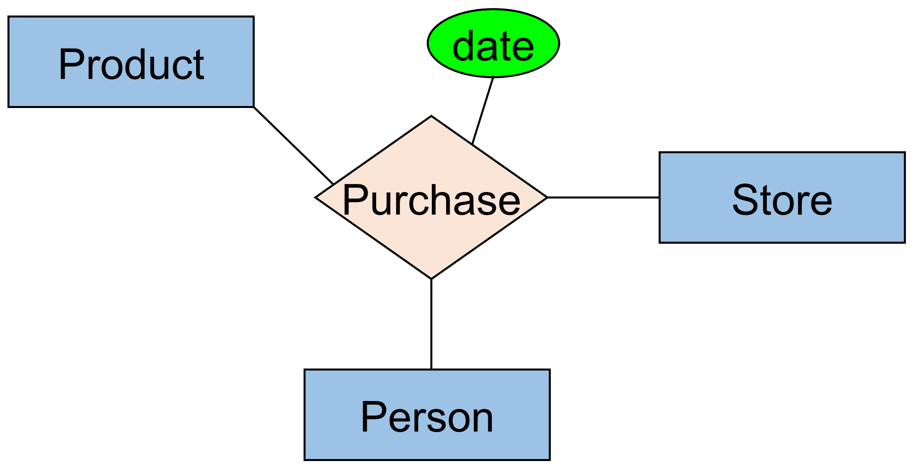
What participation constraints should we add here



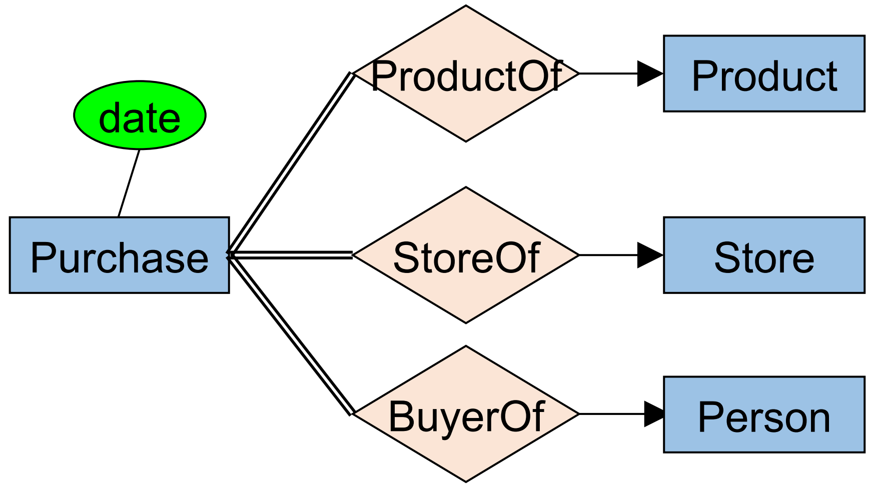
Decision: Multi-way or New Entity + Binary?

Should we use a single multi-way relationship or a new entity with binary relations?

(A) Multi-way Relationship



(B) Entity + Binary

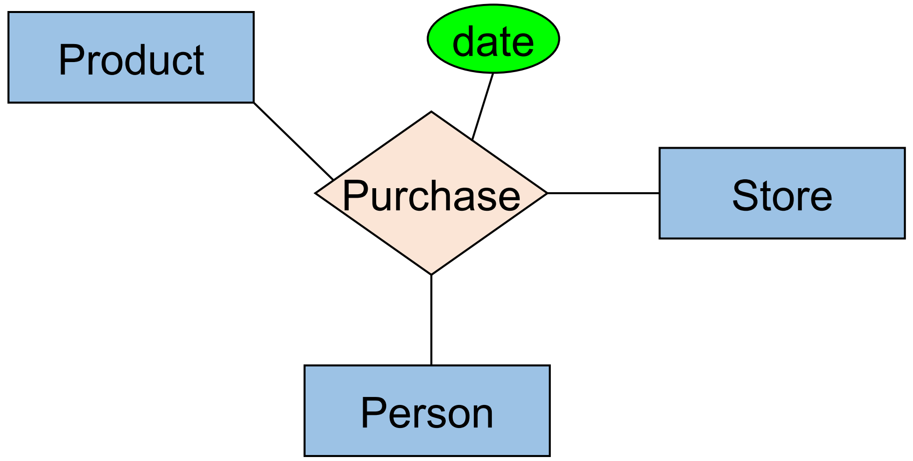


?

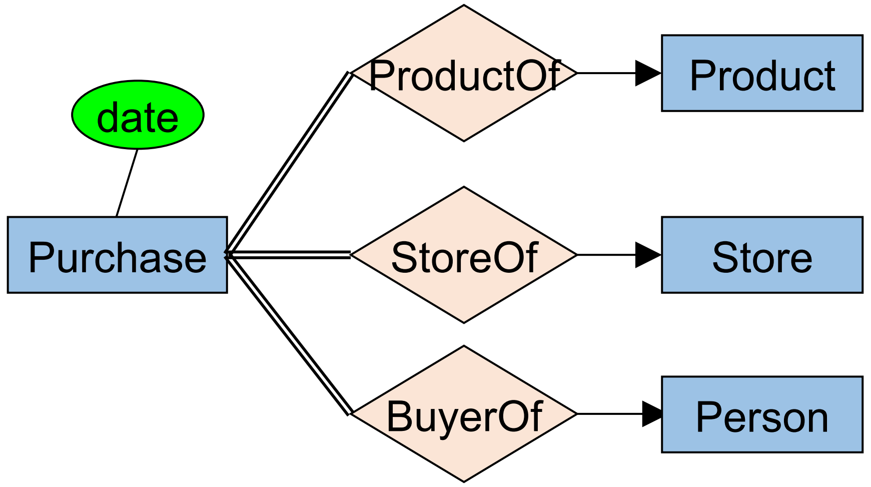
Decision: Multi-way or New Entity + Binary?

Should we use a single multi-way relationship or a new entity with binary relations?

(A) Multi-way Relationship



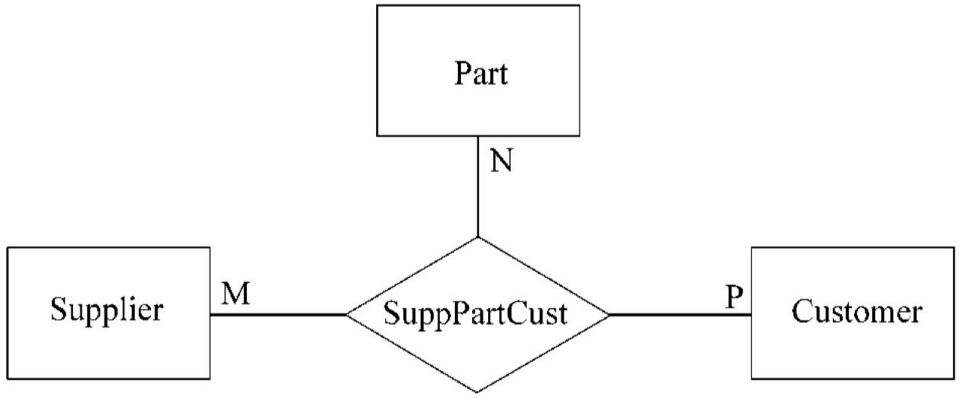
(B) Entity + Binary

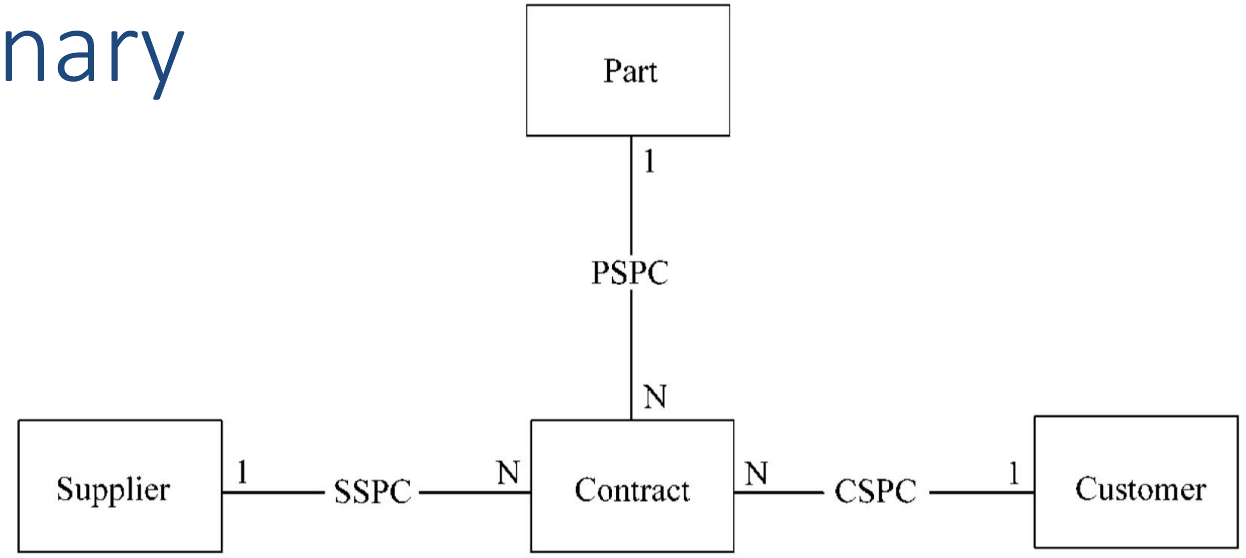


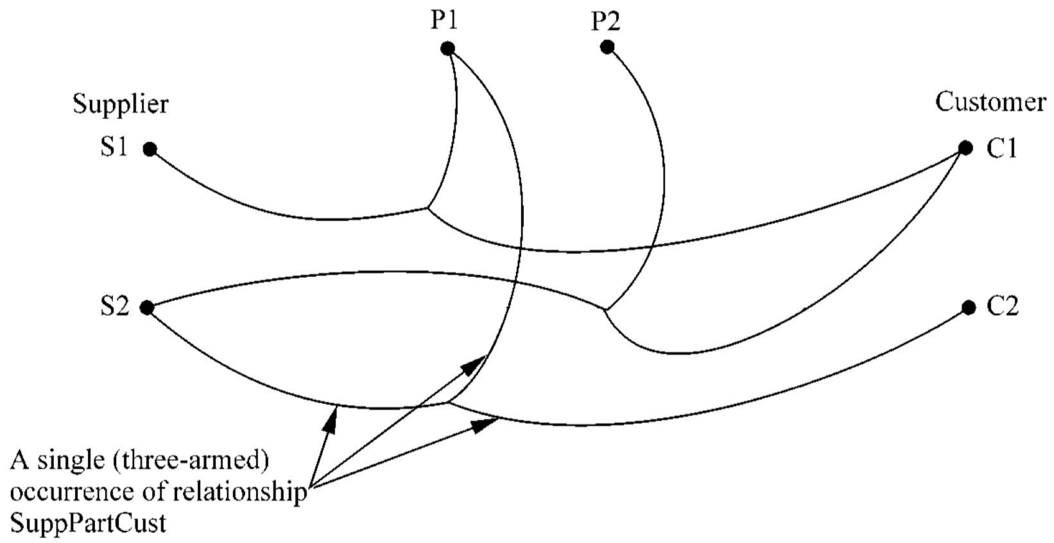
- (A) is useful when a relationship really is between multiple entities
  - Ex: A three-party legal contract

- (B) is useful if we want to have multiple instances of the "relationship" per entity combination
  - Multiple purchases per (product, store, person) combo possible here!
- *Also the way we will represent in relational tables later*

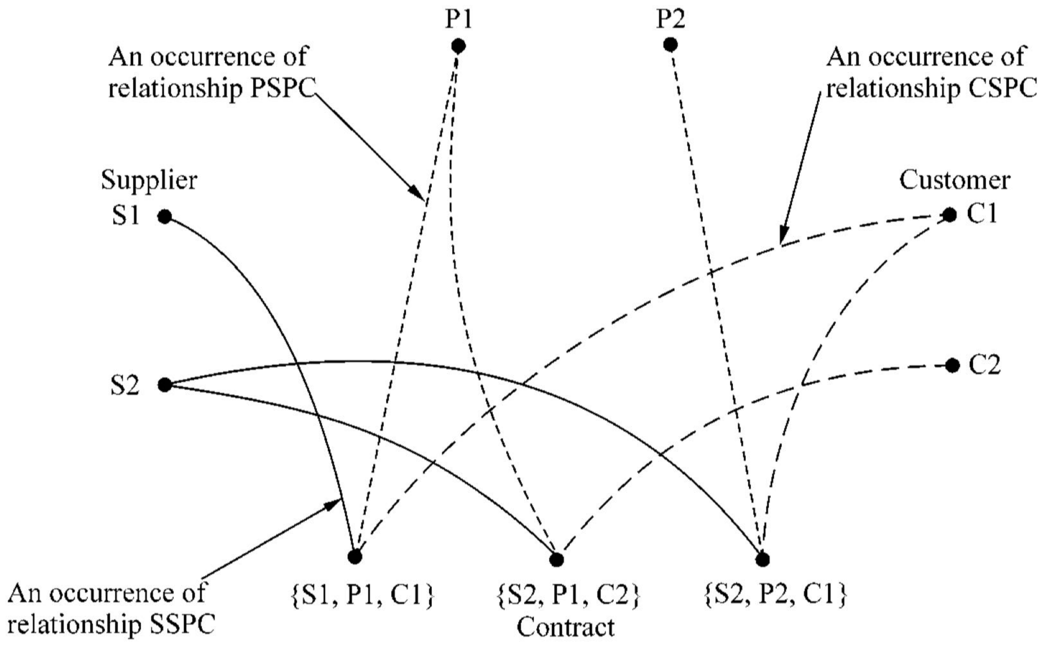
Complex relationships as binary







P1
P2
Supplier
S1
Customer
C1
S2
C2
A single (three-armed) occurrence of relationship SuppPartCust

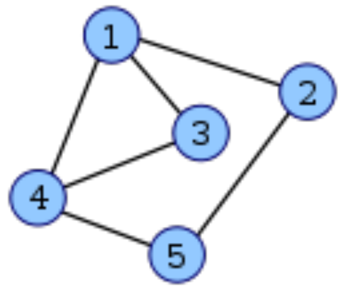


P1
P2
An occurrence of relationship PSPC
An occurrence of relationship CSPC
Supplier
S1
Customer
C1
S2
C2
An occurrence of relationship SSPC
{S1, P1, C1}
{S2, P1, C2}
Contract
{S2, P2, C1}

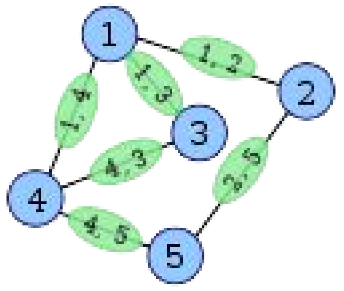
Source: David Howe. Data Analysis for Database Design, 3rd ed, 2001. https://www.amazon.com/Data-Analysis-Database-Design-David/dp/0750650869

Flipping edges and nodes: Line graphs

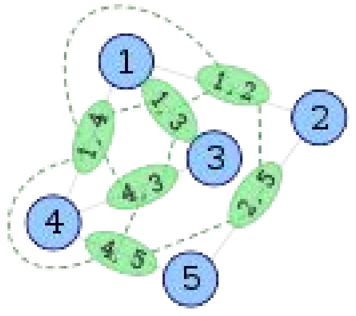
Beyond class topic



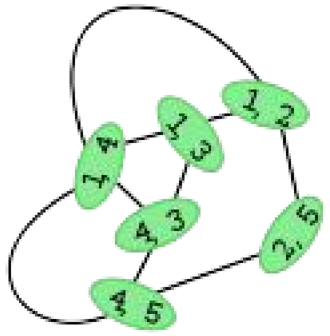
Graph G



Vertices in L(G) constructed from edges in G



Added edges in L(G)

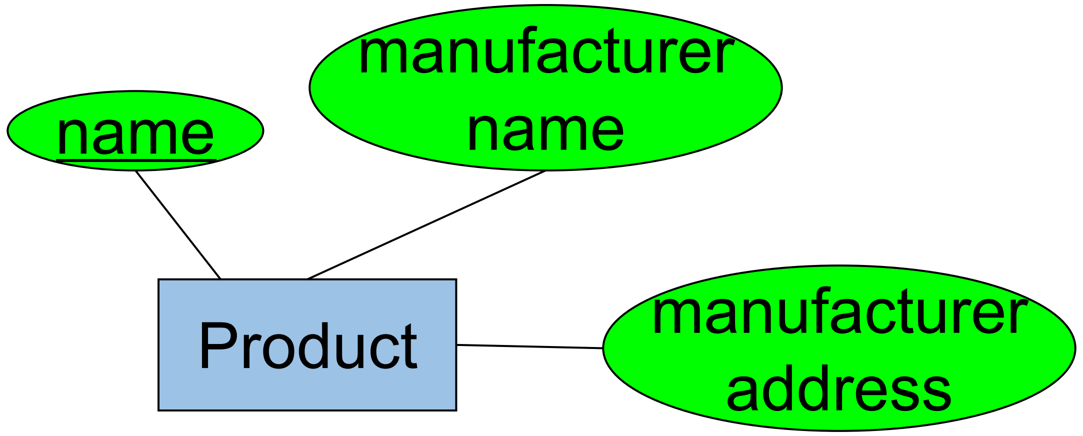


The line graph L(G)

Duality between nodes and vertices omnipresent throughout CS, e.g.:

- duality in linear programming
- binary vs higher-order probabilistic graphical models
- min edges cover vs. max independent (vertex) set
- worst-case optimal join algorithms in databases
- ...

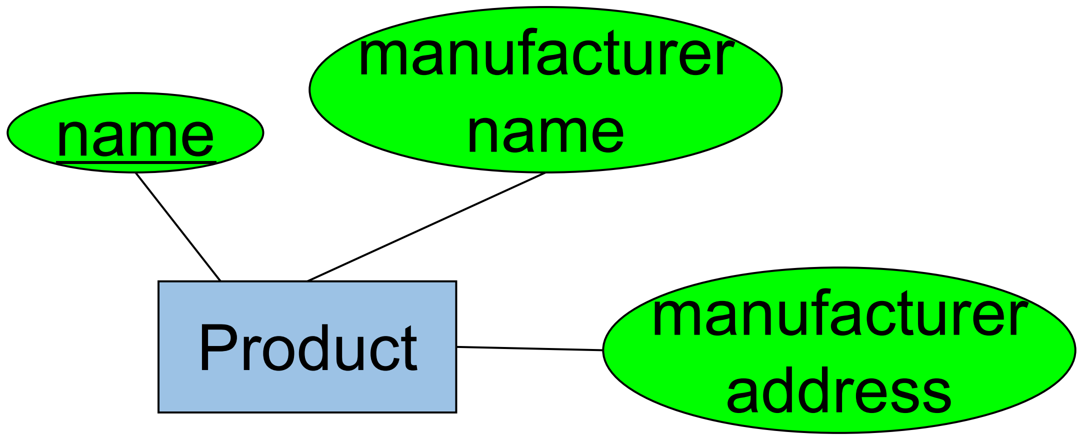
Source: https://en.wikipedia.org/wiki/Line_graph



What do you think about this ERD?

?

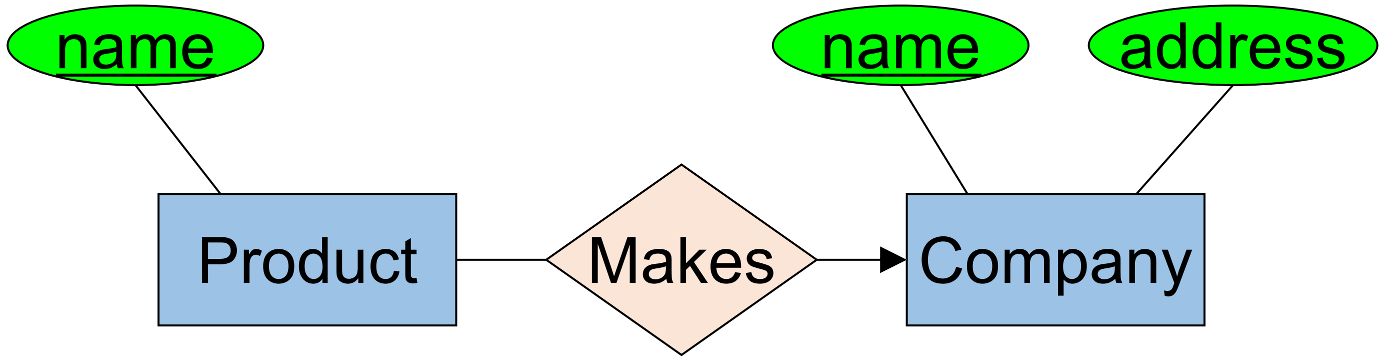
Entities vs attributes



☹

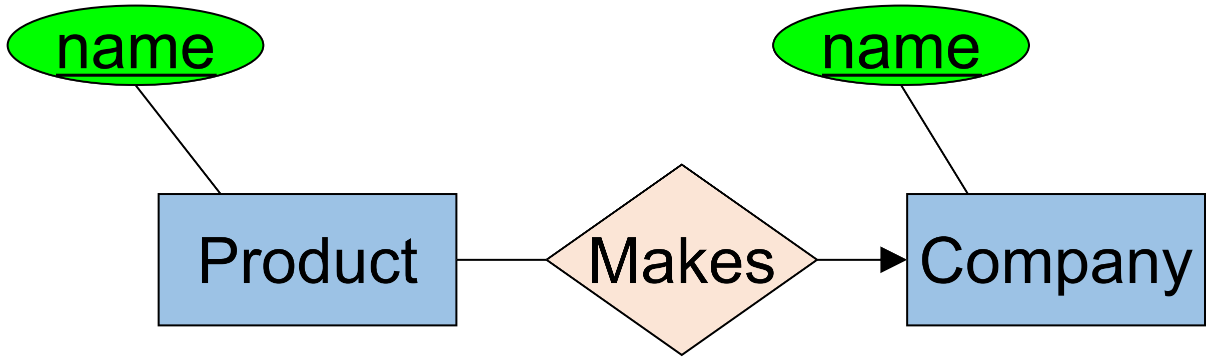
- This ERD repeats the producer's address for each product.
- Cannot store the address if a producer has no product.

Entities vs attributes



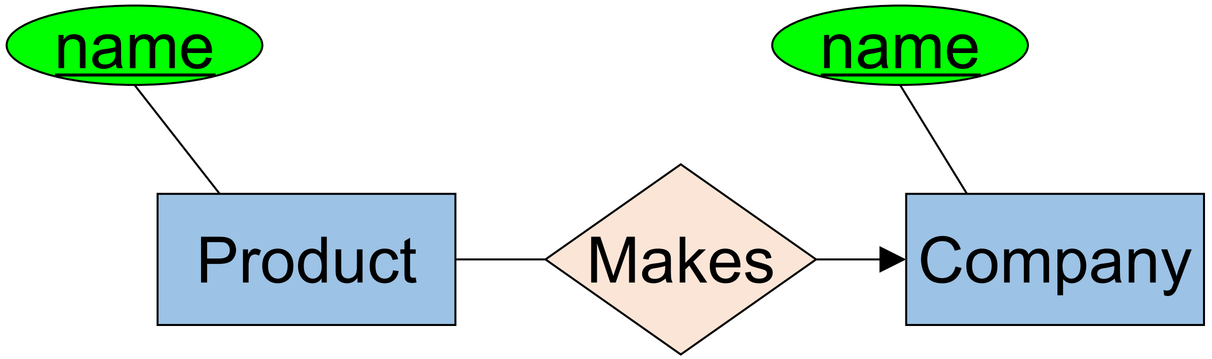
- The company deserves to be its own entity set because it has at least one nonkey attribut: "address"
- Product deserves to be an entity set because it has likely additional attributes, and can possibly participate in additional relationships
- According to some design philosophies: product should be an entity because it is at the many end of the many-to-one relationship (cp. with genre in IMDB)

Entities vs attributes



?

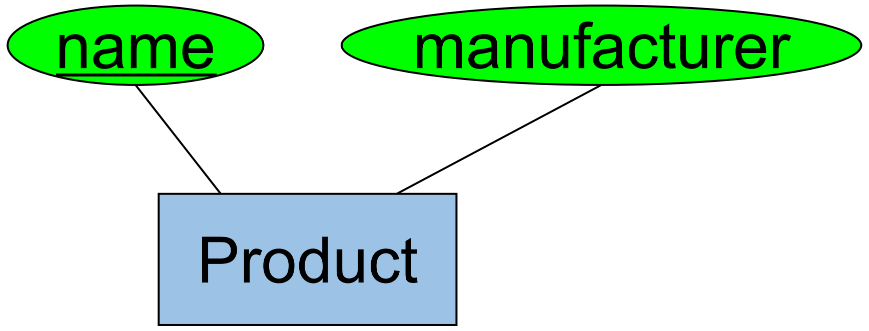
Entities vs attributes



☹

- The company is just a name, it should not be an entity set.

Entities vs attributes



☺

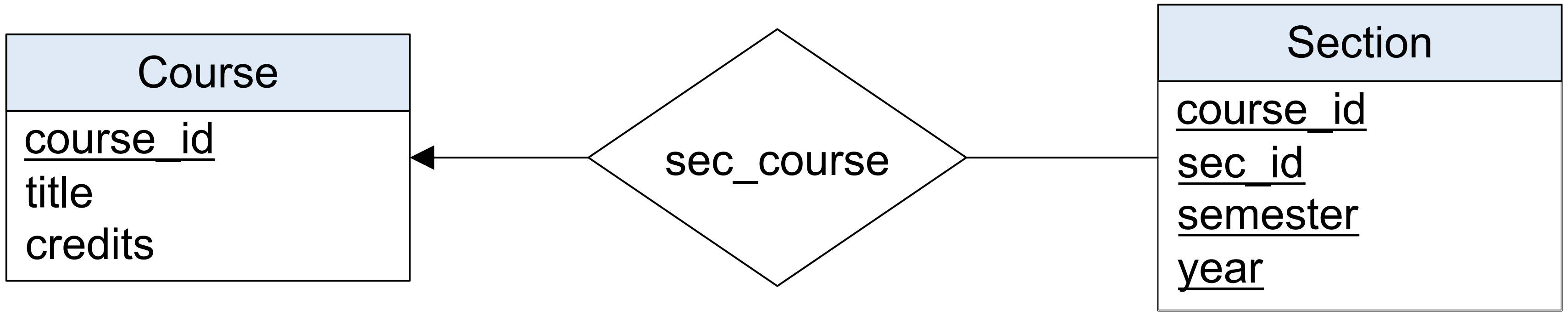
- There is no need to make the company an entity set, because we record nothing about the company besides their name

Entity set should:
- be more than the name of something; it has at least one nonkey attribute; or
- be at the "many" end in a many-to-one or many-to-many relationship

Weak (or dependent) Entities

What do we do here?

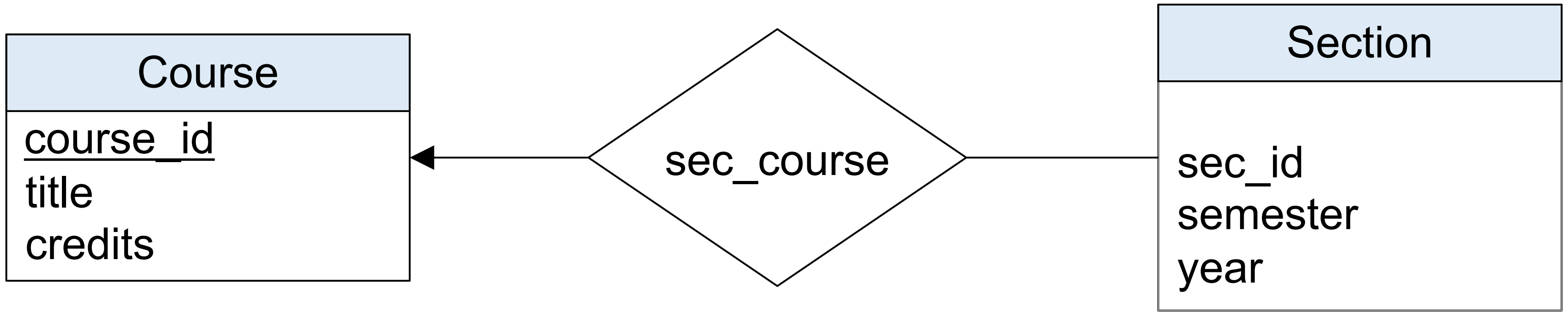
Assume a section is uniquely identified by course_id, section_id, semester, year



?

What do we do here?

Assume a section is uniquely identified by course_id, section_id, semester, year

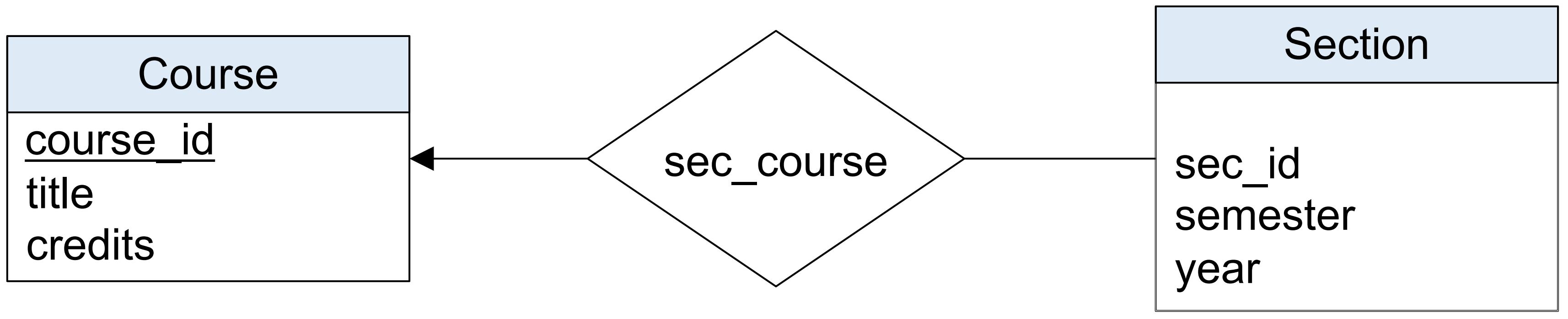


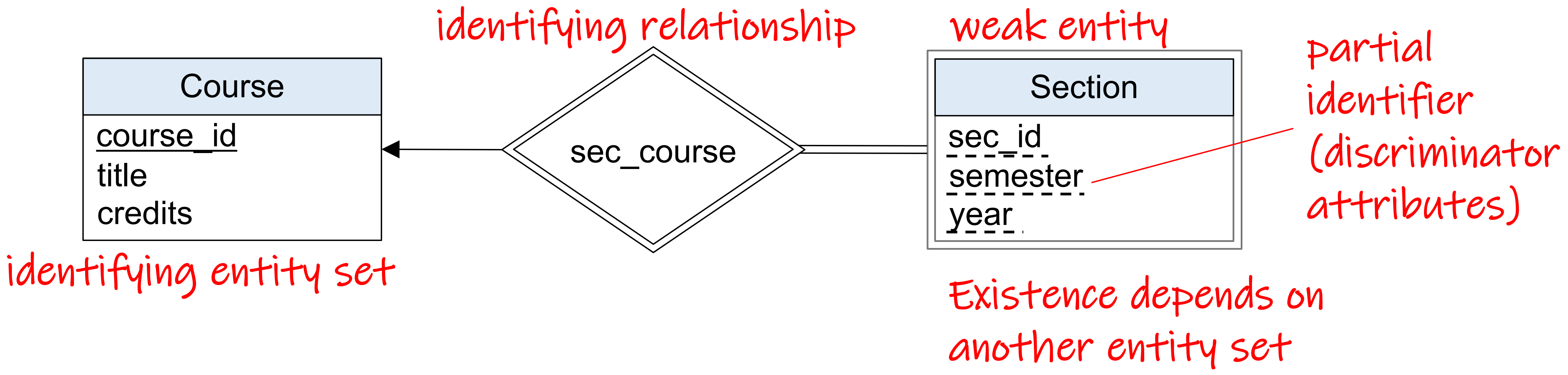
?

Now does not have enough attributes to identify a particular section entity uniquely ☹

What do we do here?

Assume a section is uniquely identified by course_id, section_id, semester, year





Strong vs. Weak (Dependent) Entities

- **Strong entities**
  - Can be identified ("exist") independently of other types of entities
  - Have their own unique identifier
- **Weak entities**
  - Dependent on a strong entity, cannot exist on their own (better: cannot be identified independently)
  - Do not have unique identifiers: PK overlaps with parent's PK
  - (represented with double-line rectangle)
- **Identifying relationship**
  - Links strong entities to weak entities
  - Represented with double line relationship

Entity sets are weak when part of their identifier comes from classes to which they are related

In ER diagrams, make sure to mark partial identifier different from primary keys (notations vary)

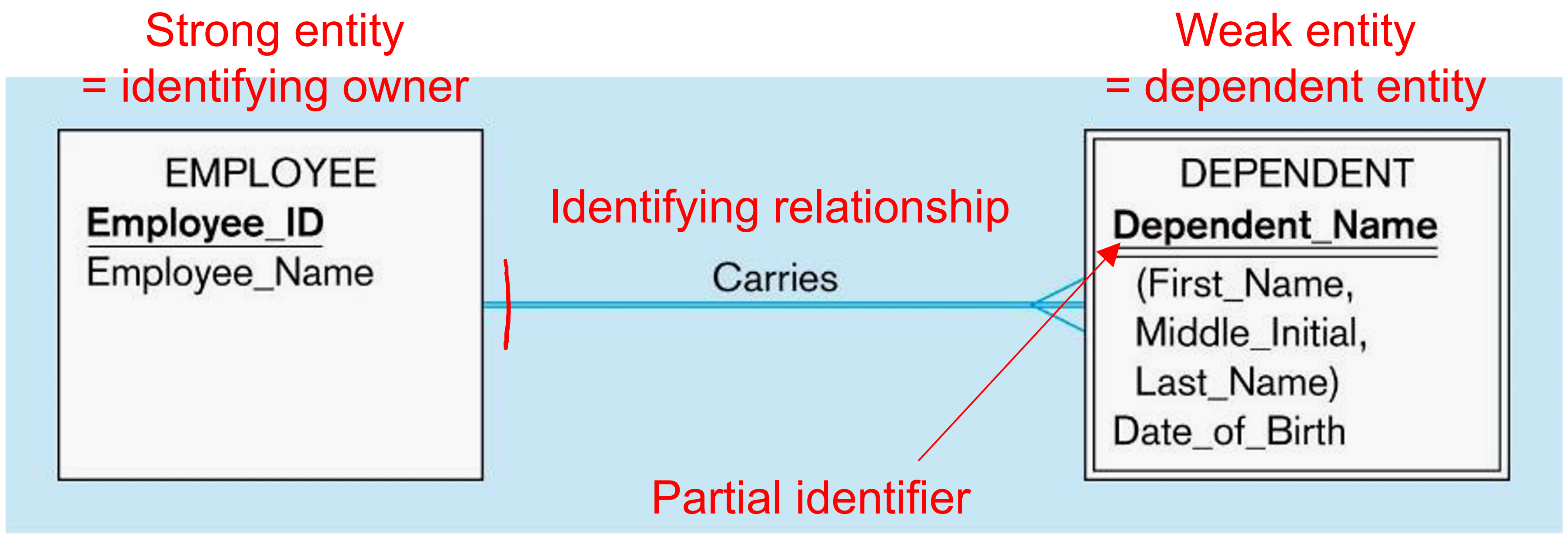
Example: Strong and Weak Entities in crow foot

- Employee caries one dependent
  - Employee: ID, name
  - Dependent: name, Date_of_Birth

?

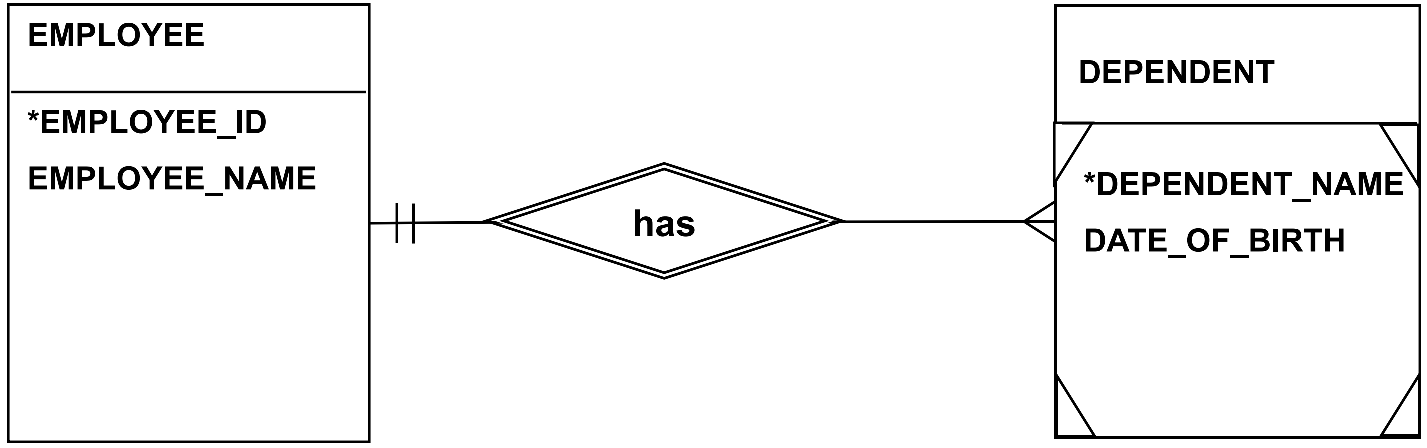
Example: Strong and Weak Entities in crow foot

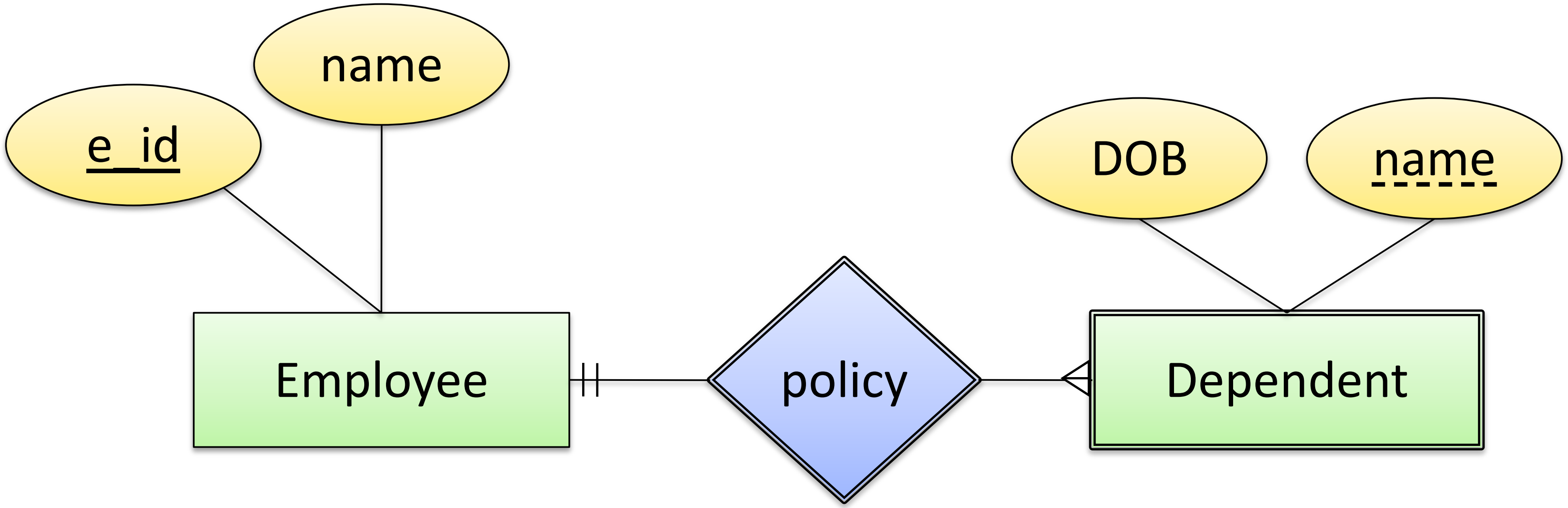
- Employee caries one dependent
  - Employee: ID, name
  - Dependent: name, Date_of_Birth



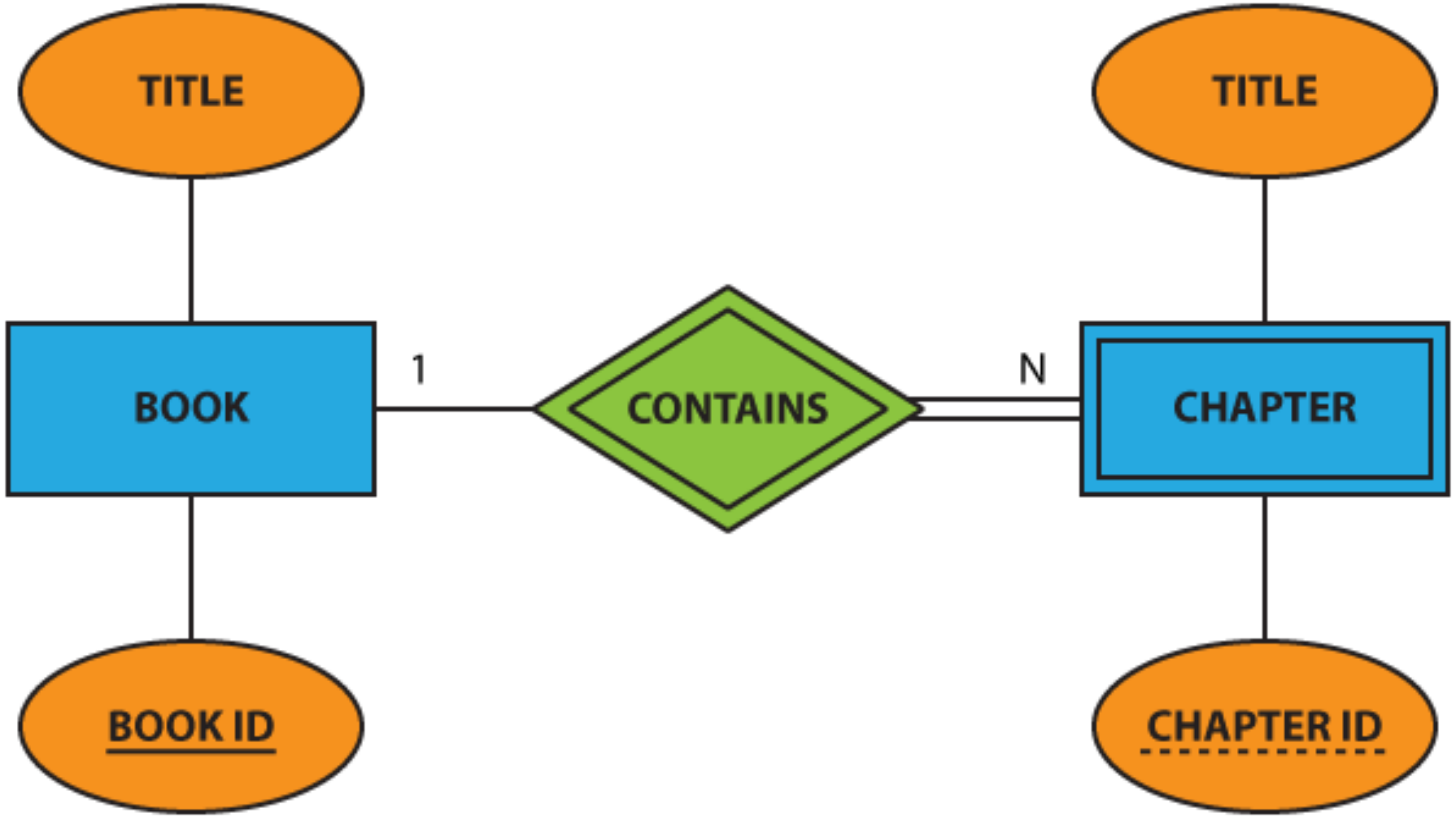
Note we need both EMPLOYEE_ID and DEPENDENT_NAME to uniquely identify a dependent

Alternative notations for same scenario



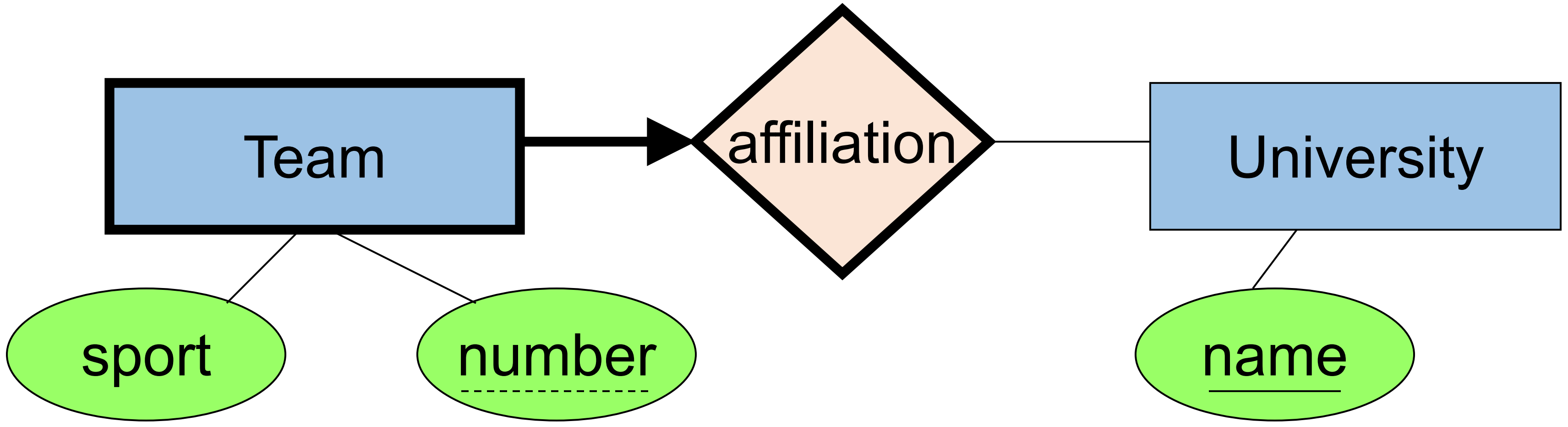


Participation constraints and weak entities in Elmasri notat



Source: http://www.vertabelo.com/blog/technical-articles/chen-erd-notation

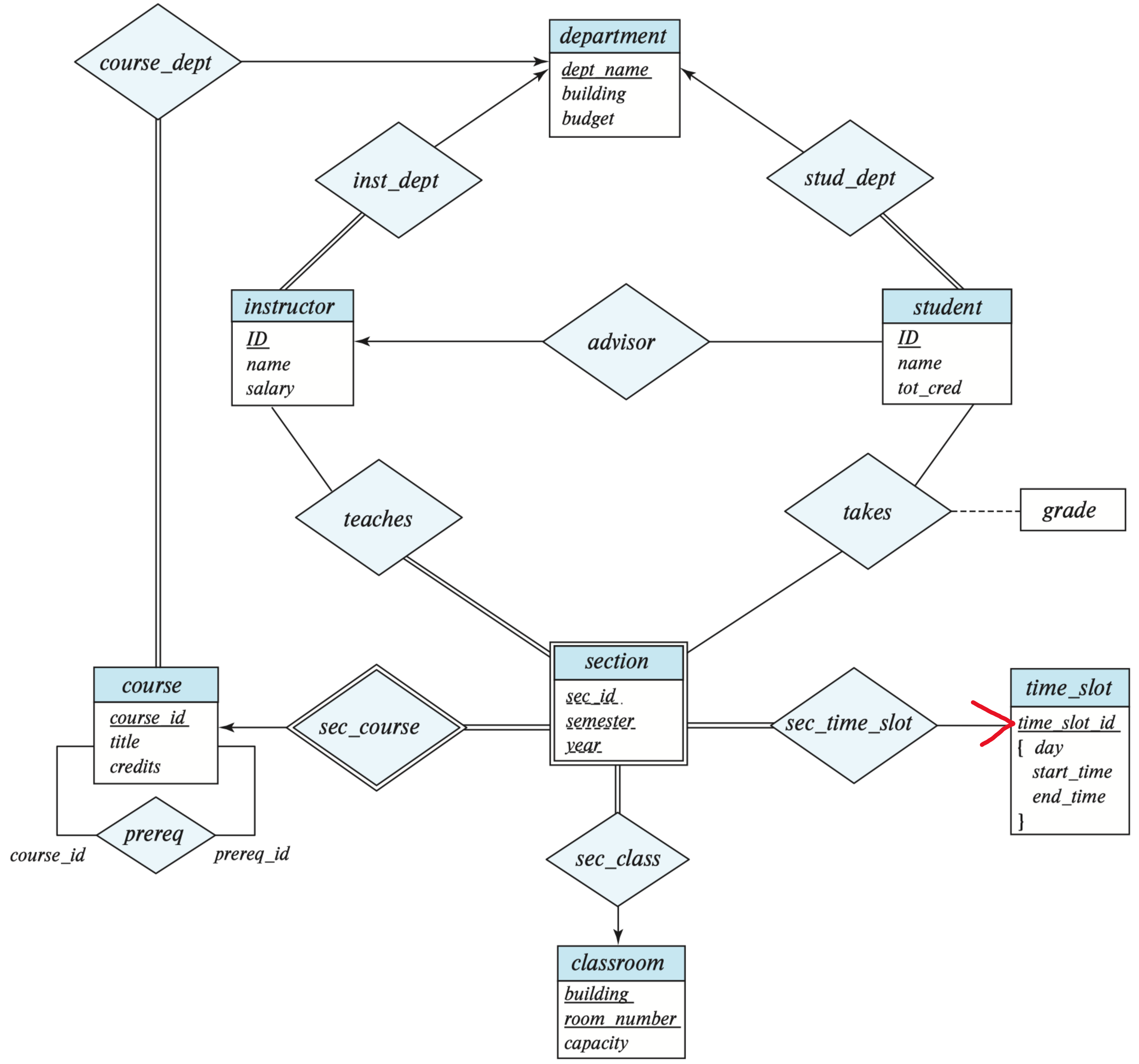
Weak Entity Sets (cow book)



"Football team" v. "***The Northeastern*** Football team" *(E.g., BU has a football team too, sort of)*

- number is a *partial key*. (denote with dashed underline).
- University is called the *identifying owner*.
- Participation in affiliation must be total. Why?

University ERD



Source: "SDK", Silberschatz, Korth, Sudarshan, Database system concepts, 7th ed, 2020. Fig 6.15
Wolfgang Gatterbauer. Database design: https://northeastern-datalab.github.io/cs3200/

University ERD

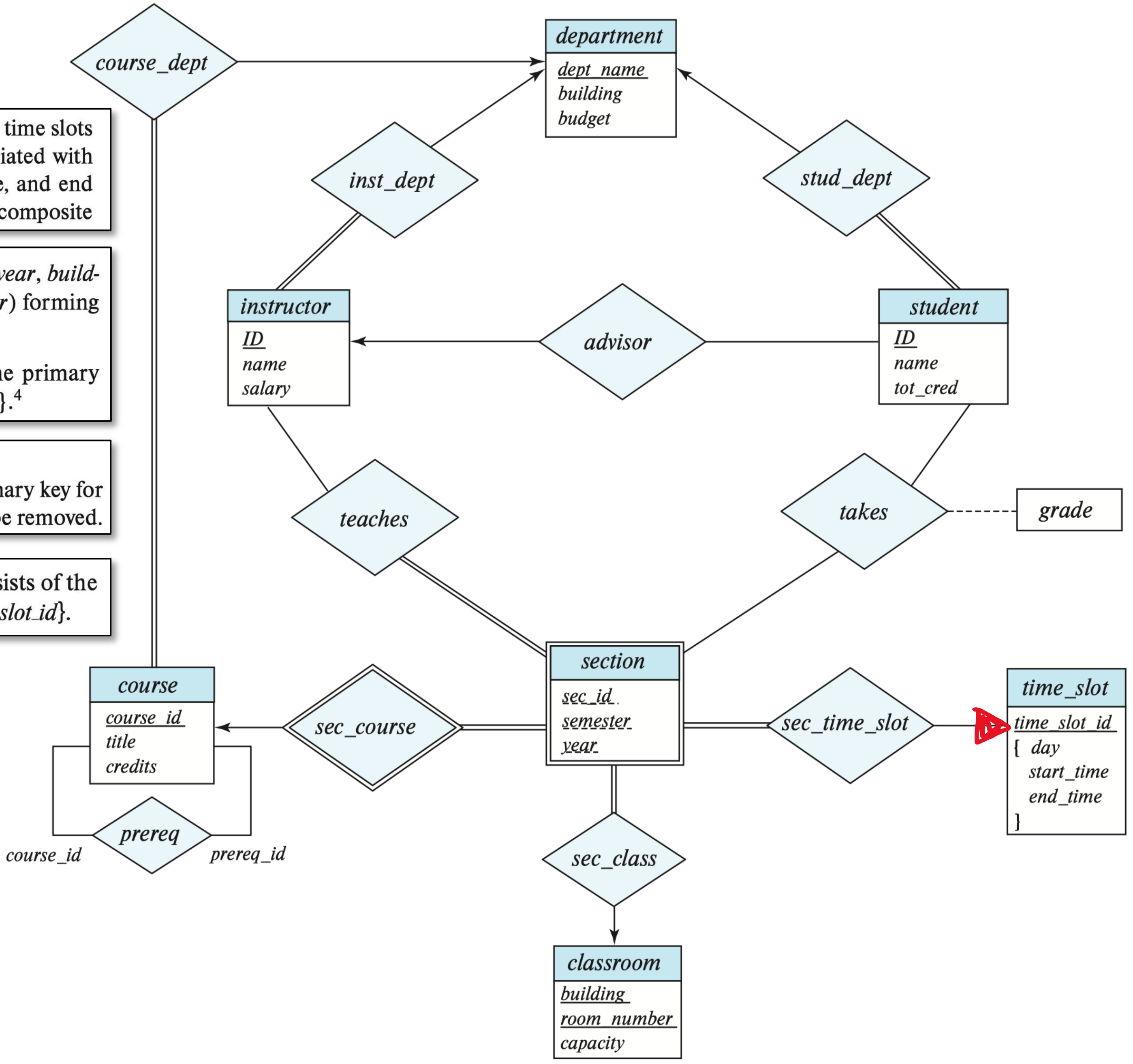
As another example, consider course offerings (sections) along with the time slots of the offerings. Each time slot is identified by a *time_slot_id*, and has associated with it a set of weekly meetings, each identified by a day of the week, start time, and end time. We decide to model the set of weekly meeting times as a multivalued composite

- The entity set *section* includes the attributes *course_id*, *sec_id*, *semester*, *year*, *building*, *room_number*, and *time_slot_id*, with (*course_id*, *sec_id*, *year*, *semester*) forming the primary key.
- The entity set *time_slot* includes the attributes *time_slot_id*, which is the primary key,[3] and a multivalued composite attribute {(*day*, *start_time*, *end_time*)}.[4]

These entities are related through the relationship set *sec_time_slot*.
The attribute *time_slot_id* appears in both entity sets. Since it is the primary key for the entity set *time_slot*, it is redundant in the entity set *section* and needs to be removed.

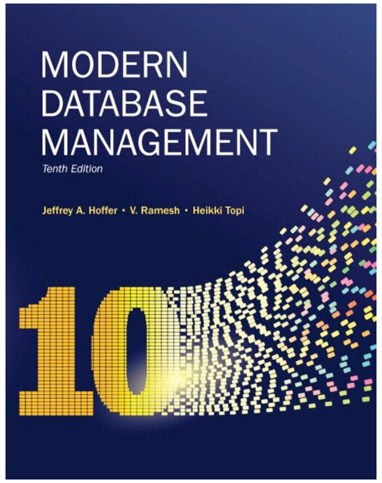
schema obtained in the previous step. The resulting *section* schema consists of the attributes {*course_id*, *sec_id*, *semester*, *year*, *building*, *room_number*, *time_slot_id*}.

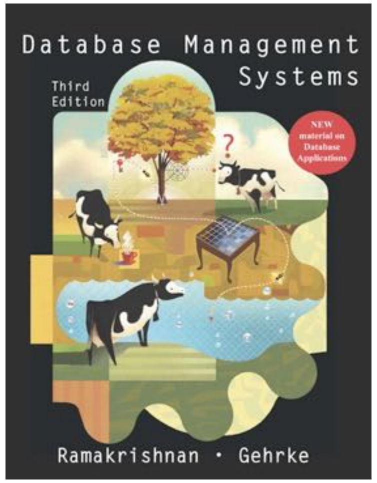
sec_time_slot (*course_id*, *sec_id*, *semester*, *year*, *time_slot_id*)

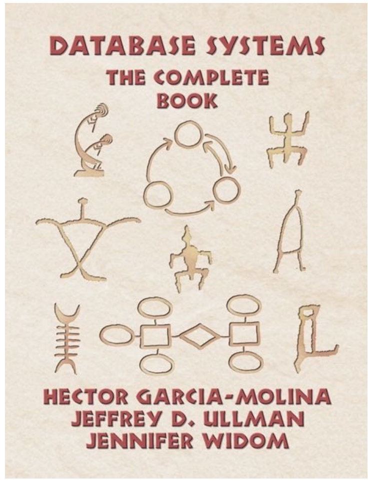


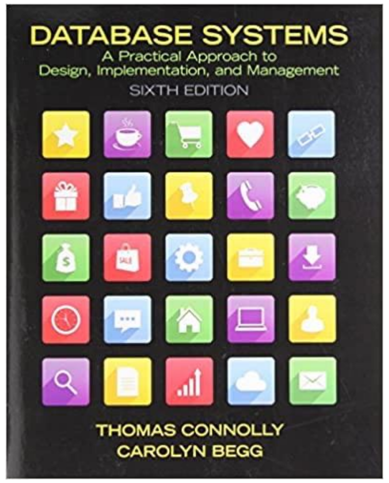
Source: "SDK", Silberschatz, Korth, Sudarshan, Database system concepts, 7th ed, 2020. Fig 6.15

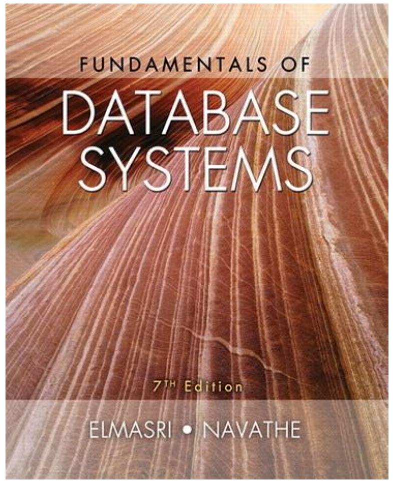
Different sources, different notations

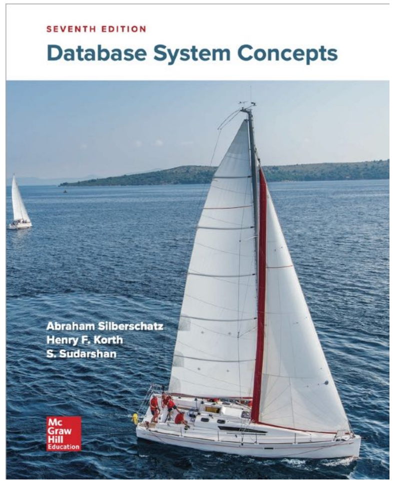

MODERN DATABASE MANAGEMENT
Tenth Edition
Jeffrey A. Hoffer • V. Ramesh • Heikki Topi
10






DATABASE SYSTEMS
A Practical Approach to Design, Implementation, and Management
SIXTH EDITION
THOMAS CONNOLLY
CAROLYN BEGG





[Hoffer+'10] [Cow book'03] [Stanford book'08] [Connolly+'15] [Elmasri+'15] [Silberschatz+'20]

Crow foot SDK arrows

[Hoffer+'10]: Hoffer, Ramesh, Topi. Modern Database Management, 10th ed, 2010.
https://www.pearson.com/us/higher-education/product/Hoffer-Modern-Database-Management-10th-Edition/9780136088394.html

[Cow book'03]: Ramakrishnan, Gehrke, Database Management Systems, 3rd ed, 2003. http://pages.cs.wisc.edu/~dbbook/

[Stanford book'08]: Garcia-Molina, Ullman, Widom. Database Systems: The Complete Book, 2nd ed, 2008. http://infolab.stanford.edu/~ullman/dscb.html

[Connolly+'15]: Connolly, Begg. Database systems: A practical approach to design, implementation, and management, 6th ed, 2015.
https://www.pearson.com/us/higher-education/program/Connolly-Database-Systems-A-Practical-Approach-to-Design-Implementation-and-Management-6th-Edition/PGM116956.html

[Elmasri+'15]: Elmasri, Navathe. Fundamentals of Database Systems, 7th ed, 2015.
https://www.pearson.com/us/higher-education/program/Elmasri-Fundamentals-of-Database-Systems-7th-Edition/PGM189052.html

[Silberschatz+'20]: Silberschatz, Korth, Sudarshan. Database system concepts, 7th ed, 2020. https://www.db-book.com/db7

Notations for binary one-to-many relationships

10/24/2022

if you are overwhelmed by the different notations, just use the one from our textbook

Every student is advised by maximum 1 instructor.
An instructor may advise 0, 1 or more students.

Every product is manufactured by exactly 1 company.
A company may manufacture 0, 1 or more products.

SDK [Silberschatz+'20]

[Stanford book'03]
also used by Gradiance

Crow's foot
[Hoffer+'10]

Most often used in practice

look across notation

UML
[Connolly+'15]

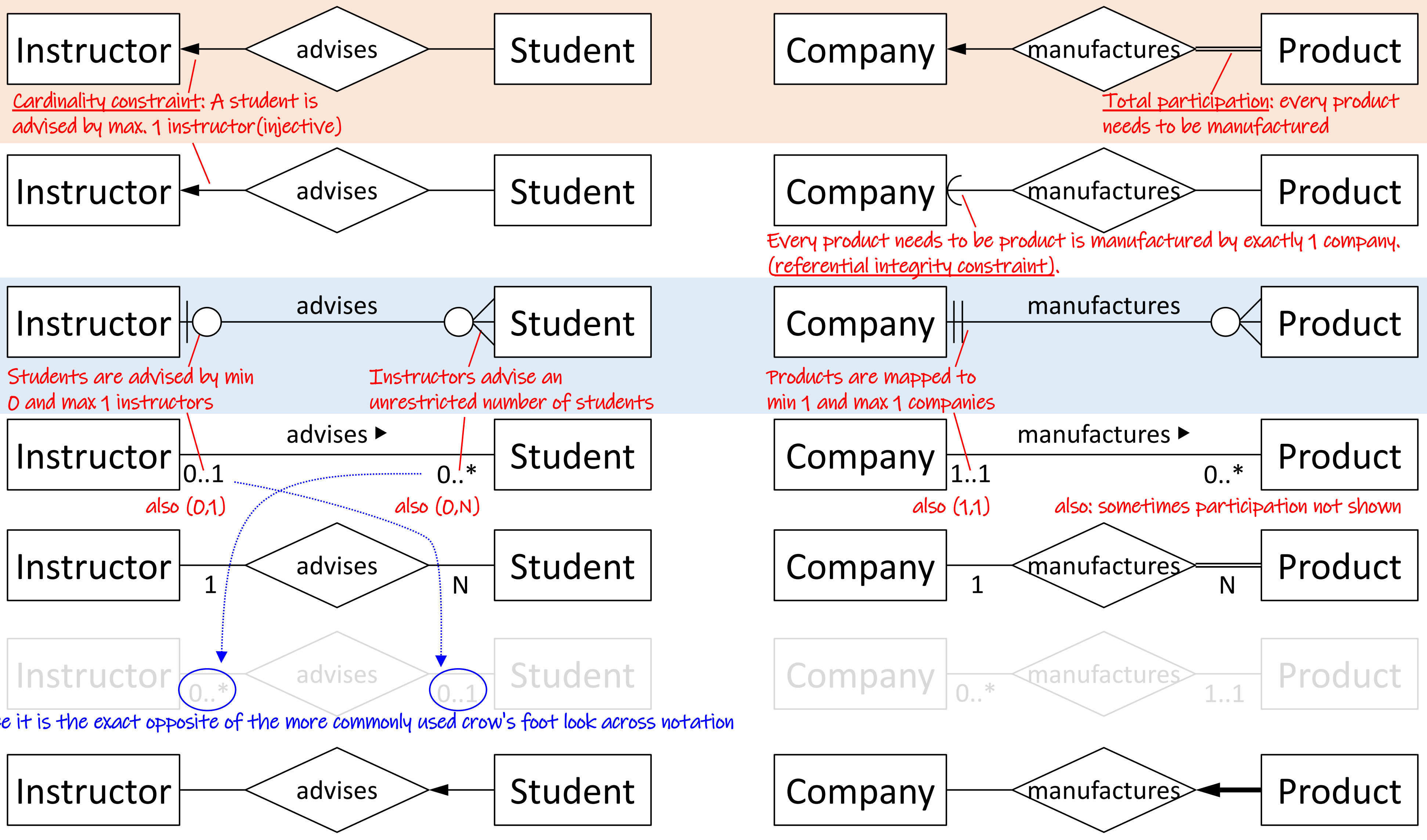
[Elmasri+'15]

look across for cardinality, same-side for participation

Avoid (min-max) !!!
(min-max)
[Elmasri+'15]

same-side notation

[Cow book'03]



Cardinality constraint: A student is advised by max. 1 instructor (injective)

Total participation: every product needs to be manufactured

Every product needs to be product is manufactured by exactly 1 company. (referential integrity constraint).

Students are advised by min 0 and max 1 instructors

Instructors advise an unrestricted number of students

Products are mapped to min 1 and max 1 companies

also (0,1)

also (0,N)

also (1,1)

also: sometimes participation not shown

strongly discouraged since it is the exact opposite of the more commonly used crow's foot look across notation

Notations for weak entities

10/24/2022

if you are overwhelmed by the different notations, just use the one from our textbook

A course may have 0, 1 or more sections

Every product is manufactured by exactly 1 company.
A company may manufacture 0, 1 or more products.

[Silberschatz+'20]

[Stanford book'03]
also by Gradiance

Crow's foot
[Hoffer+'10]

Most often used in practice

look across notation

UML
[Connolly+'15]

[Elmasri+'15]

look across for cardinality, same-side for participation

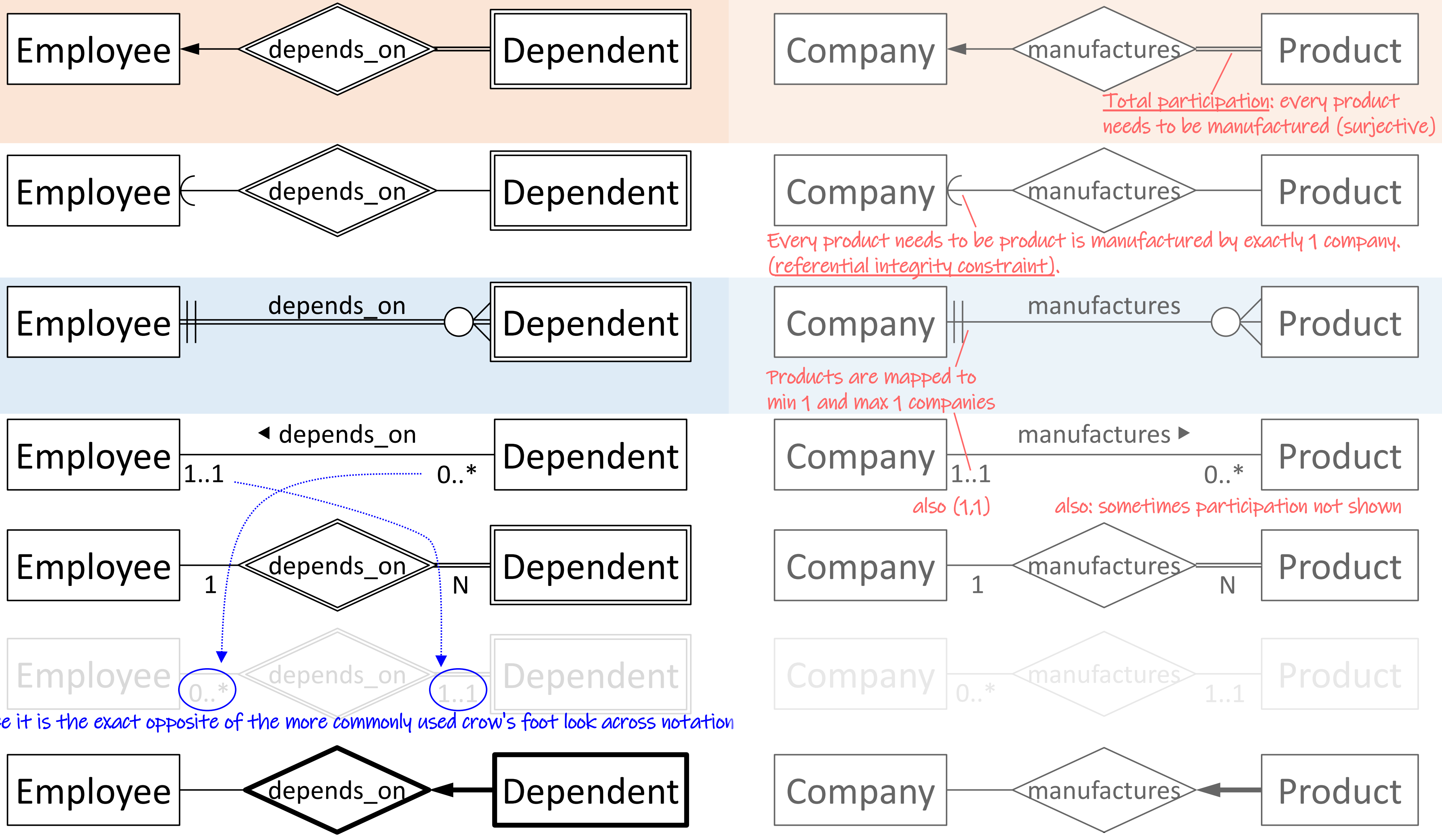
Avoid (min-max) !!!

(min-max)
[Elmasri+'15]

same-side notation

strongly discouraged since it is the exact opposite of the more commonly used crow's foot look across notation

[Cow book'03]



Employee
depends_on
Dependent
Company
manufactures
Product
Total participation: every product needs to be manufactured (surjective)
Every product needs to be product is manufactured by exactly 1 company. (referential integrity constraint).
Products are mapped to min 1 and max 1 companies
◂ depends_on
1..1
0..*
manufactures ▸
also (1,1)
also: sometimes participation not shown
1
N
0..*
1..1