UNIVERSITY OF ILLINOIS
AT URBANA-CHAMPAIGN

CS411 - E/R Modeling

illinois.edu

Announcements

- MP1 Due Feb 15 (Friday!)
- Project Stage 1 due Feb 17
- Midterm 1 March 1st

Review

- Why did we learn about functional dependencies?
- What kinds of anomalies do we try to avoid in relational database design?

Review

- What are some examples of normal forms?
- Which normal form did we learn?
- Informally, what does it require?
- How can we recover the original relation?
- What is its disadvantage?

Normal Forms

- ~~First Normal Form (1NF)~~ trivial
- ~~Second Normal Form (2NF)~~ obsolete
- Third Normal Form (3NF)
- Boyce Codd Normal Form (BCNF)
- Fourth Normal Form (4NF)

BCNF (formally)

- A relation R is in BCNF if:
  - For any nontrivial FD $\{A_1,A_2,\ldots,A_n\} \rightarrow B$, $\{A_1,A_2,\ldots,A_n\}$ is a superkey

BCNF (informally)

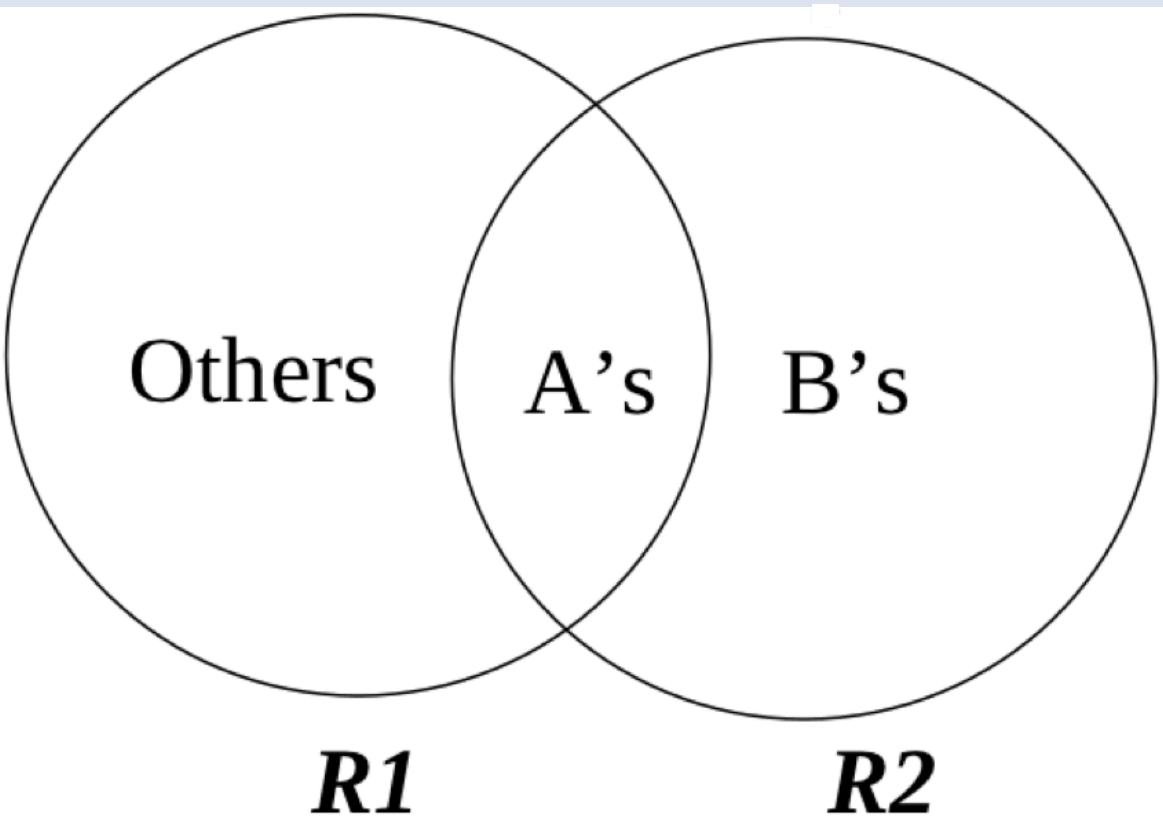
- If a set of attributes determines ***some*** of the other attributes, then it determines ***all*** of the attributes

BCNF Decomposition Algorithm

- INPUT:
  - an input relation R with attributes α
  - a set of FDs S
- OUTPUT:
  - a set of relations in BCNF

BCNF Decomposition

Decompose:



BCNF Properties

- Eliminates anomalies ☺
- Not unique 😐
- Always exists ☺
- Information is recoverable ☺
- Does *not* preserve dependencies ☹

Dependency Preservation

- Are FDs preserved by decomposition?
  - NOT in BCNF

Example

Concert(band,venue,city)

venue→city

band,city→venue

Example

Concert(band,venue,city)

venue→city

band,city→venue

Keys: {band,city},{venue,band}

Example

Concert(band,venue,city)

venue→city ← VIOLATION OF BCNF

band,city→venue

Keys: {band,city},{venue,band}

Example

Using BCNF Decomposition:

Concert1(venue,city)

Concert2(venue,band)

Example

Using BCNF Decomposition:

Concert1(venue,city)
Concert2(venue,band)

band,city→venue ← Uh oh…

Example

venue	city
Assembly Hall	Champaign
High Dive	Champaign

venue	band
High Dive	Death Cab
Assembly Hall	Death Cab

venue	city	band
Assembly Hall	Champaign	Death Cab
High Dive	Champaign	Death Cab

band,city→venue

Normal Forms

- ~~First Normal Form (1NF)~~ trivial
- ~~Second Normal Form (2NF)~~ obsolete
- Third Normal Form (3NF)
- Boyce Codd Normal Form (BCNF)
- Fourth Normal Form (4NF)

3NF

- Third normal form relaxes BCNF to preserve FDs

3NF (formally)

- A relation R is in BCNF if:
  - For any nontrivial FD $\{A_1,A_2,\ldots,A_n\} \rightarrow B$, $\{A_1,A_2,\ldots,A_n\}$ is a superkey **or B is part of a key**

3NF (informally)

- Everything must depend on the key ***or be part of a key***

3NF Decomposition Algorithm

- INPUT:
  - an input relation R
  - a set of FDs S
- OUTPUT:
  - a set of relations in 3NF

3NF Decomposition Algorithm

1. Find a minimal basis T for S
2. For each FD X→A in T, create a new relation with all the attributes of X and A
3. If none of the relations in the previous step is a superkey for R, add one more relation for a key

Example

R(A,B,C,D,E)

S={AB→C,C→B, A→D}

Example

R(A,B,C,D,E)

S={AB→C,C→B, A→D}

Is S a basis?

1. All the FDs have singleton right sides
2. If any FD is removed, it is no longer a basis
3. If we remove one or more attributes from the left side of a FD, it is no longer a basis

Example

R(A,B,C,D,E)

S={AB→C,C→B, A→D}

Create relations for each FD

R_1(A,B,C), R_2(C,D), R_3(A,D)

Example

R(A,B,C,D,E)

S={AB→C,C→B, A→D}

Keys: {A,B,E} and {A,C,E}

R_1(A,B,C), R_2(C,D), R_3(A,D)

Example

R(A,B,C,D,E)

S={AB→C,C→B, A→D}

Keys: {A,B,E} and {A,C,E}

$R_1(A,B,C)$, $R_2(C,D)$, $R_3(A,D)$, $R_4(A,C,E)$

3NF Properties

- Doesn't eliminate all anomalies ☹
- Not unique 😐
- Always exists ☺
- Information is recoverable ☺
- Preserve dependencies ☺

Example

Concert(band,venue,city)

venue→city

band,city→venue

keys: {band,city}, {band,venue}

Already in 3NF!

Design principle

- Aim for BCNF, settle for 3NF

Normal Forms

- ~~First Normal Form (1NF)~~ trivial
- ~~Second Normal Form (2NF)~~ obsolete
- Third Normal Form (3NF)
- Boyce Codd Normal Form (BCNF)
- Fourth Normal Form (4NF)

Multivalued Dependencies

- If certain values are fixed, other attributes become independent of the rest

$$A_1, A_2, \ldots, A_n \twoheadrightarrow B_1, B_2, \ldots, B_m$$

Example

album	track	song	band	year
Contra	1	Horchata	Vampire Weekend	2010
Contra	2	White Sky	Vampire Weekend	2010
Contra	3	Holiday	Vampire Weekend	2010
Contra	4	California English	Vampire Weekend	2010
Contra	5	Taxi Cab	Vampire Weekend	2010
Contra	6	Run	Vampire Weekend	2010
Contra	7	Cousins	Vampire Weekend	2010
Contra	8	Giving up the Gun	Vampire Weekend	2010
Contra	9	Diplomat's Son	Vampire Weekend	2010

Example

album	track	song	band	year
Contra	1	Horchata	Vampire Weekend	2010
Contra	2	White Sky	Vampire Weekend	2010
Contra	3	Holiday	Vampire Weekend	2010
Contra	4	California English	Vampire Weekend	2010
Contra	5	Taxi Cab	Vampire Weekend	2010

$$album, band \twoheadrightarrow track, song$$

4NF (formally)

- A relation R is in 4NF if:
  - For any nontrivial MVD $\{A_1,A_2,\ldots,A_n\} \twoheadrightarrow B$, $\{A_1,A_2,\ldots,A_n\}$ is a superkey

4NF and MVDs

- MVDs have essentially the same inference rules as FDs
- 4NF is essentially BCNF with MVDs instead of FDs

Normal Forms

- ~~First Normal Form (1NF)~~ trivial
- ~~Second Normal Form (2NF)~~ obsolete
- Third Normal Form (3NF)
- Boyce Codd Normal Form (BCNF)
- Fourth Normal Form (4NF)
- Fifth Normal Form (5NF)
- Sixth Normal Form (6NF)

Normal Forms

- ~~First Normal Form (1NF)~~ trivial
- ~~Second Normal Form (2NF)~~ obsolete
- Third Normal Form (3NF)
- Boyce Codd Normal Form (BCNF)
- Fourth Normal Form (4NF)
- ~~Fifth Normal Form (5NF)~~ too esoteric
- ~~Sixth Normal Form (6NF)~~ too esoteric

Why did we learn all this?

- Normal forms allow us to eliminate certain kinds of redundancy
  - Prevent anomalies
- But you know the saying…

In theory, theory and practice are the same.

In practice, they are not.

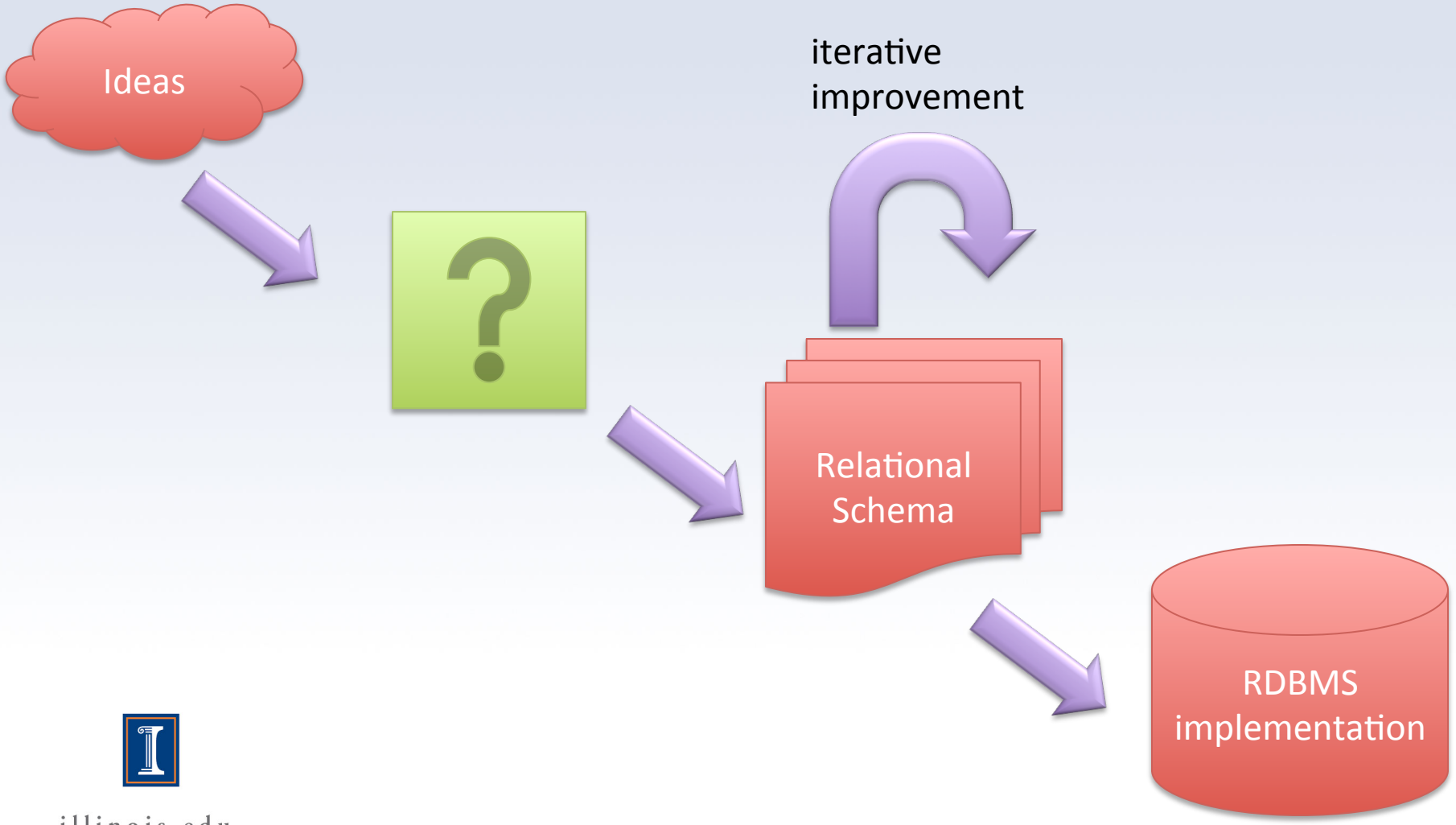
Denormalization

- Joining tables has a cost
- Denormalizing can be more efficient
- Best schema ***always*** depends on the ***application***

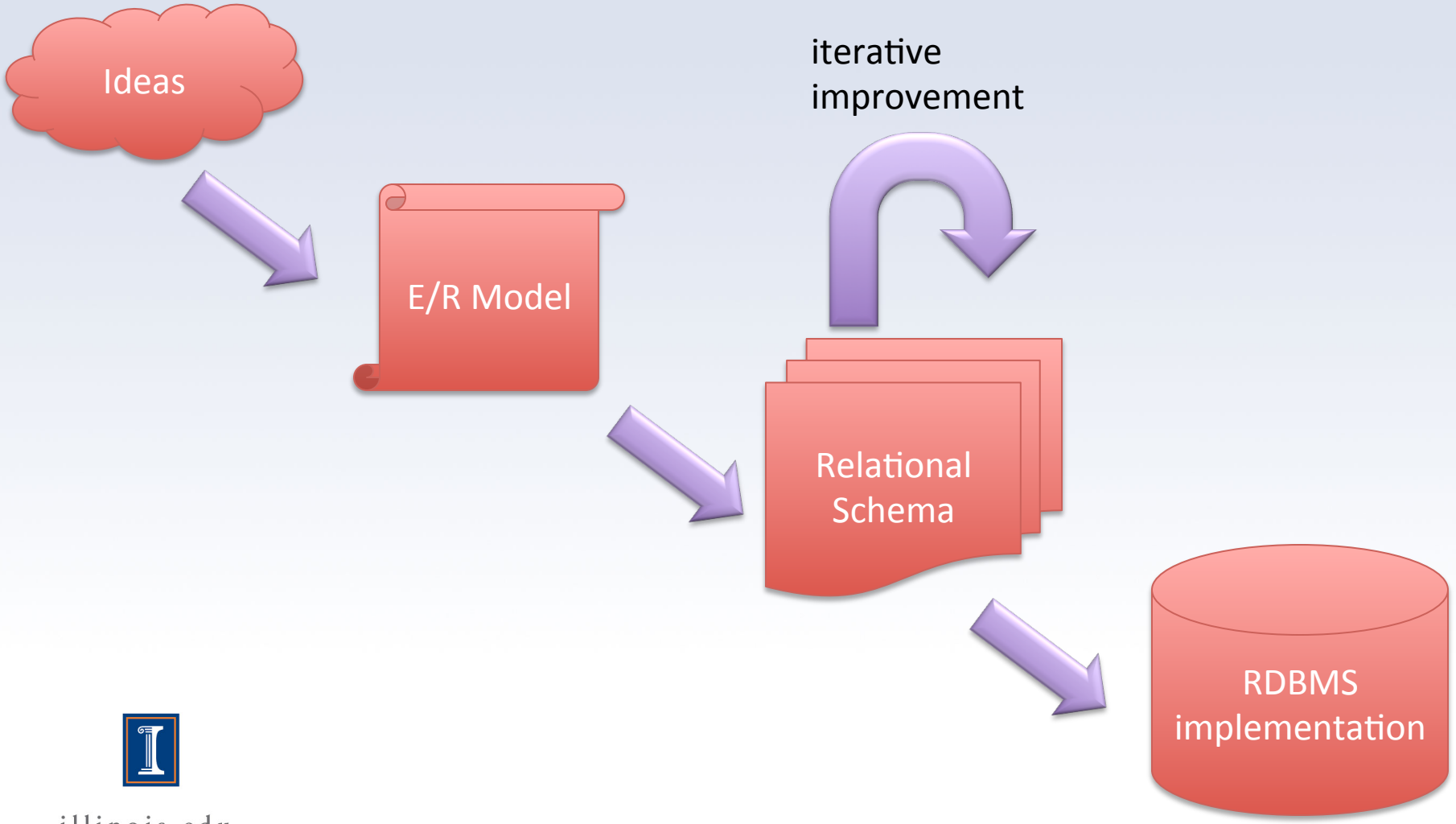
Where are we?

- We can query a good database schema.
- We can iteratively improve existing database schema through normalization.
- How do we go from an idea to a schema?

Design Process



Design Process



Chapter 4.2

- I'm not going to go over this chapter
- Elaborates on this process at a higher level
- No technical content
- Might be very helpful for course project

E/R Modeling

- ER stands for Entity/Relationship
- Graphical model of data
- Models three element types:
  1. Entities 
  2. Attributes 
  3. Relationships 

E/R Modeling

- ER stands for Entity/Relationship
- Graphical model of data
- Models three element types:
  1. Entities 
     - the real-world objects we want to model
     - similar to a "class" in programming languages
     - examples: person, country, movie

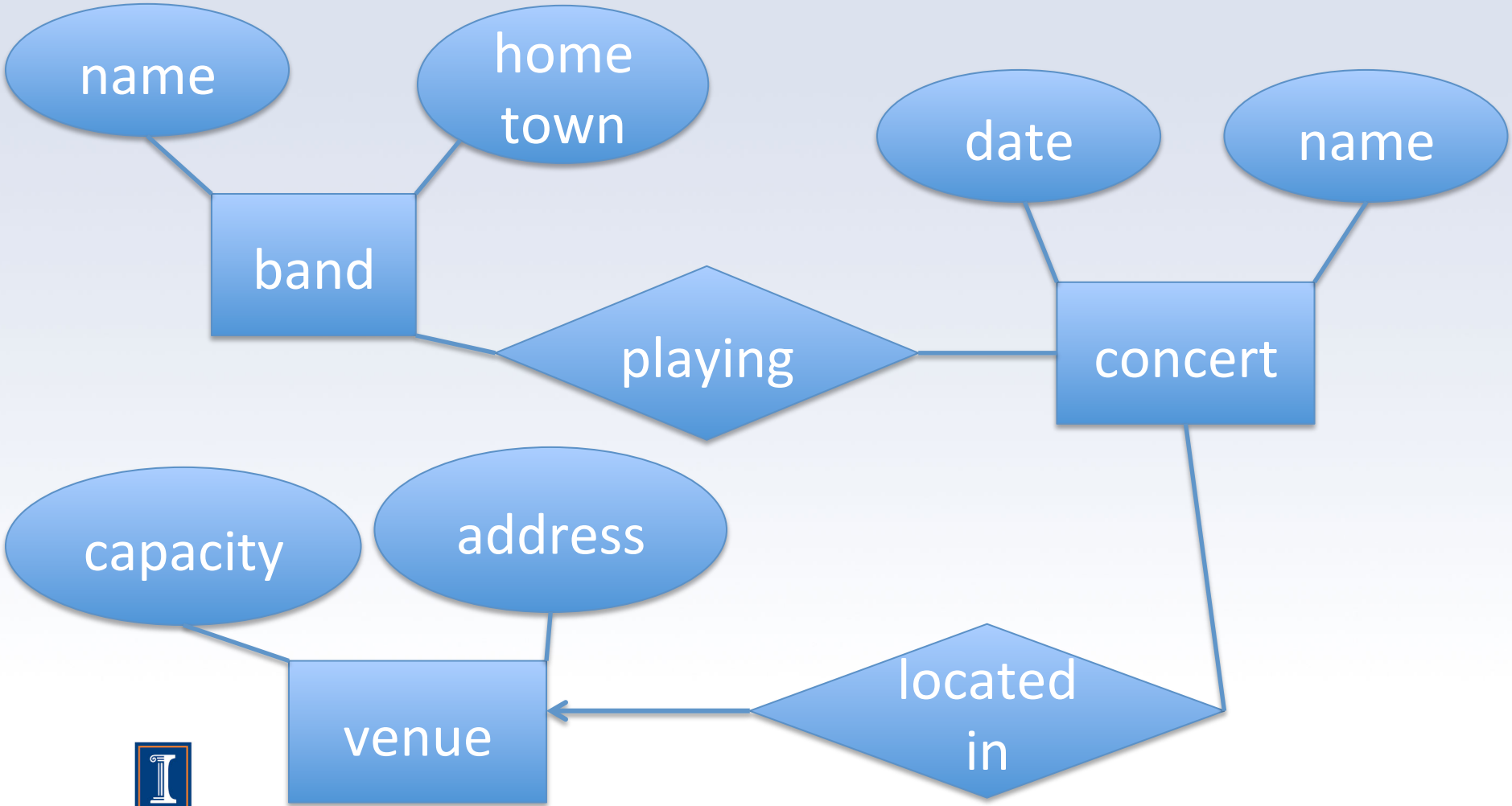
E/R Modeling

- ER stands for Entity/Relationship
- Graphical model of data
- Models three element types:
  1. Entities 
  2. Attributes 
     - describe the entity
     - atomic types: string, int, real, boolean, etc.

E/R Modeling

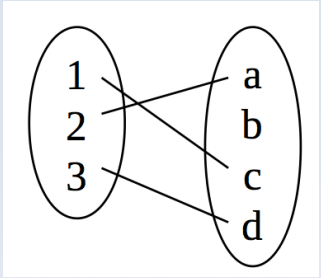
- ER stands for Entity/Relationship
- Graphical model of data
- Models three element types:
  1. Entities 
  2. Attributes 
  3. Relationships 
     - describe the relationships among entities

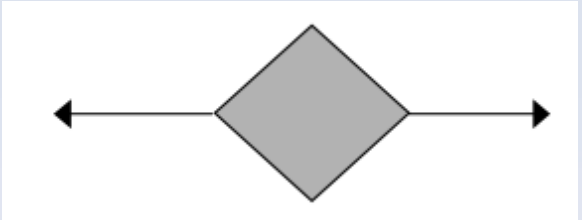
E/R Diagram


name
home town
band
playing
date
name
concert
capacity
address
venue
located in

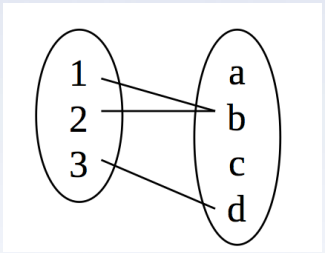
Multiplicity of Relationships

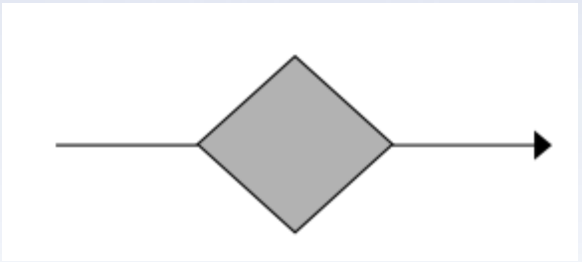
one-one



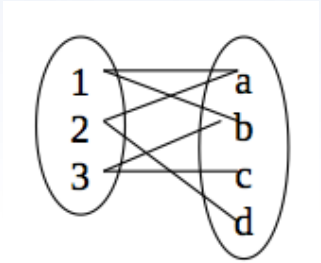


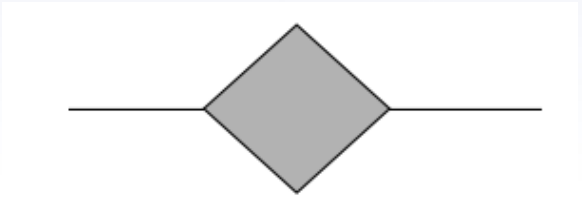
many-one



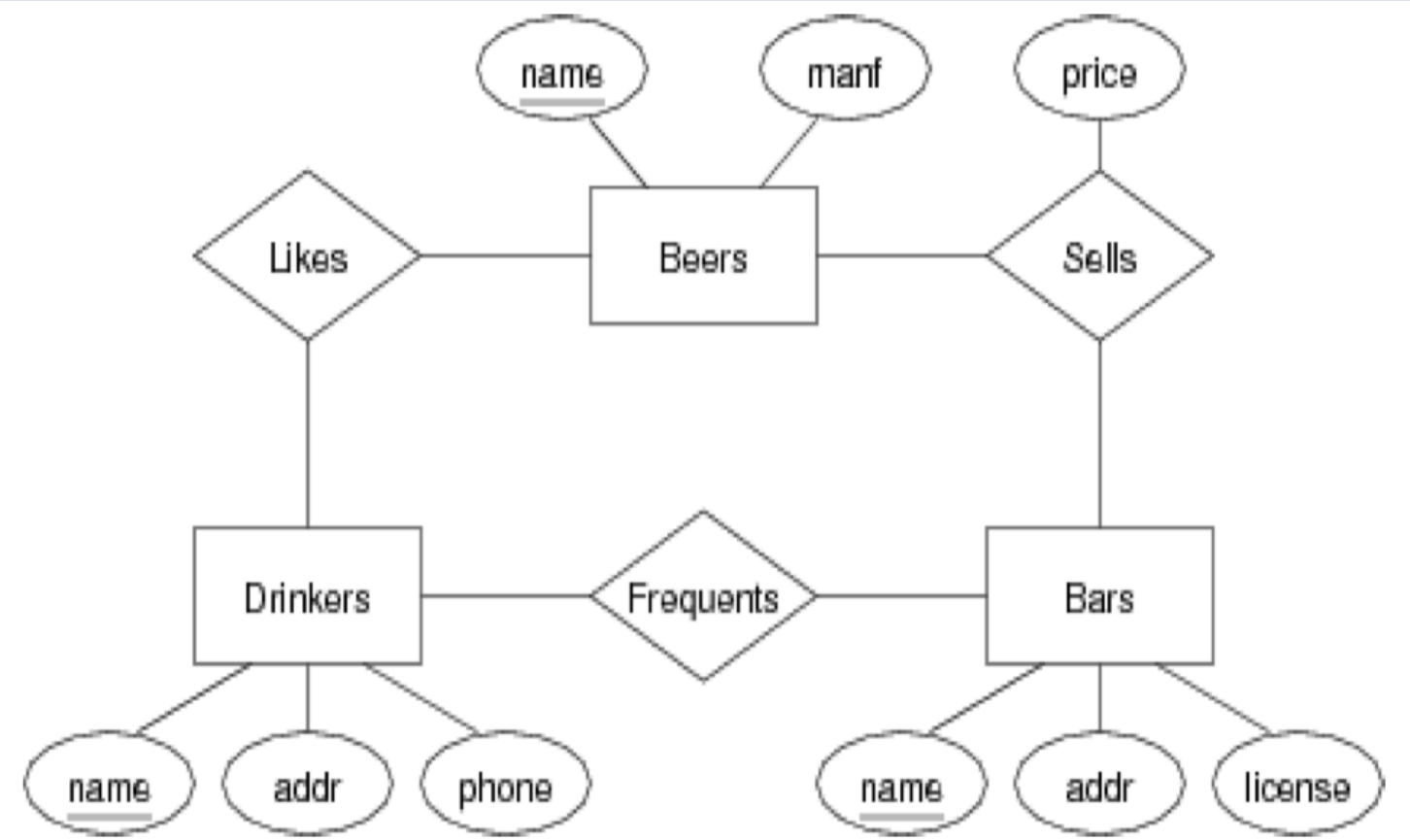


many-many





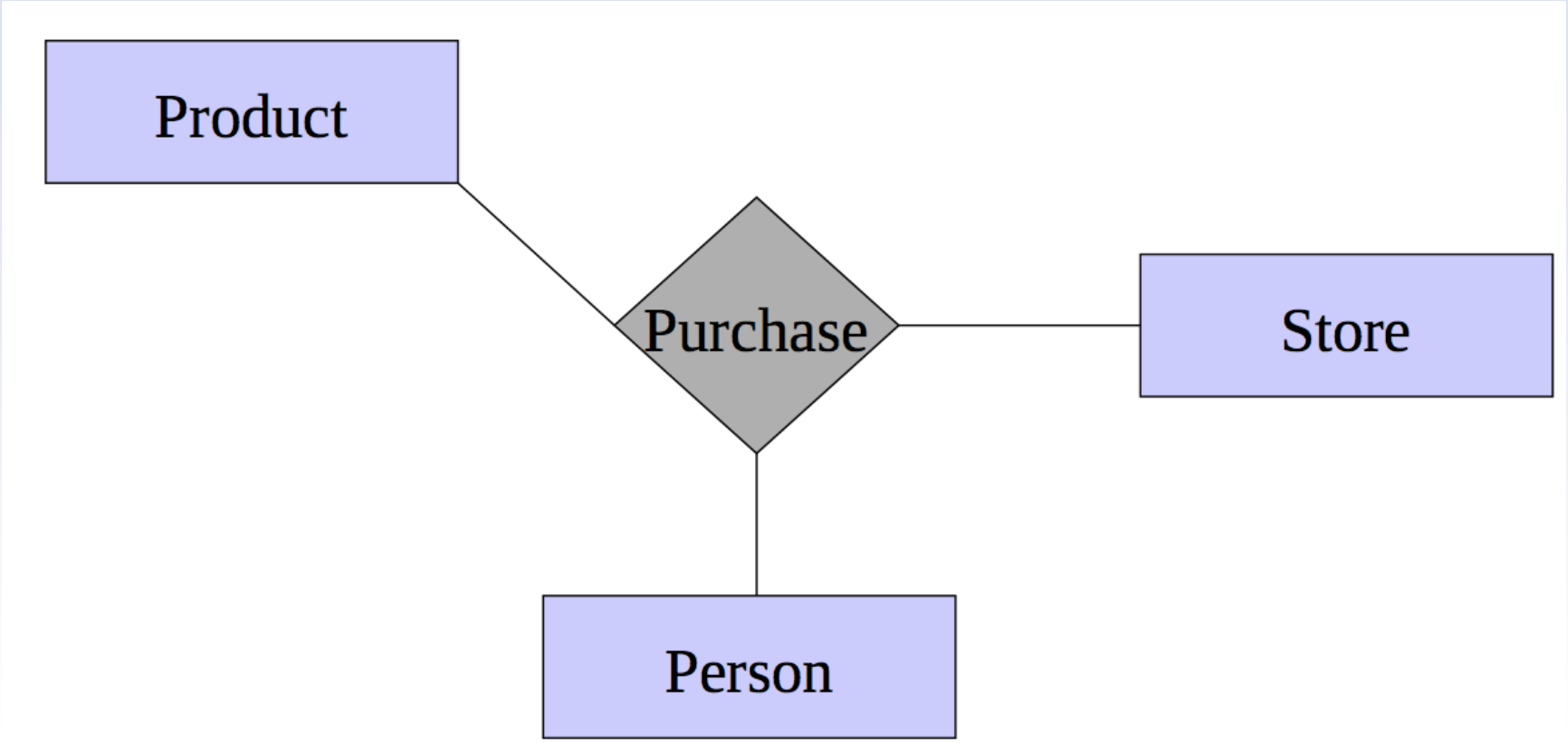
Example


name
manf
price
Likes
Beers
Sells
Drinkers
Frequents
Bars
name
addr
phone
name
addr
license

Multiway relationships

- Sometimes a relation involves 3 or more entities
  - Musician plays instrument in band
  - Person buys product from store
- Simply add more entities to the relationship

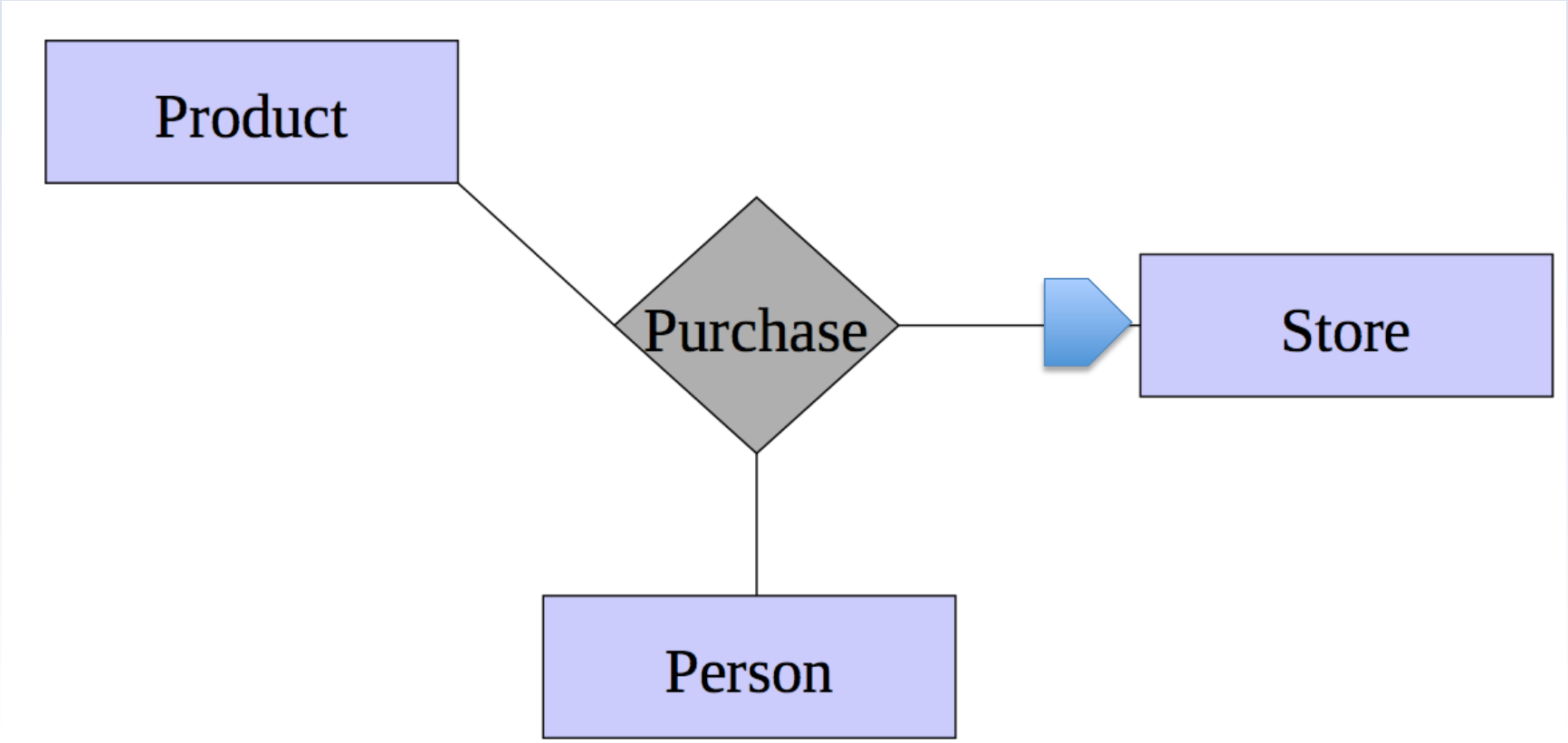
Example



Multiplicity of Multiway

- We can add arrows to multiway relationships
  - Fundamentally limited, though

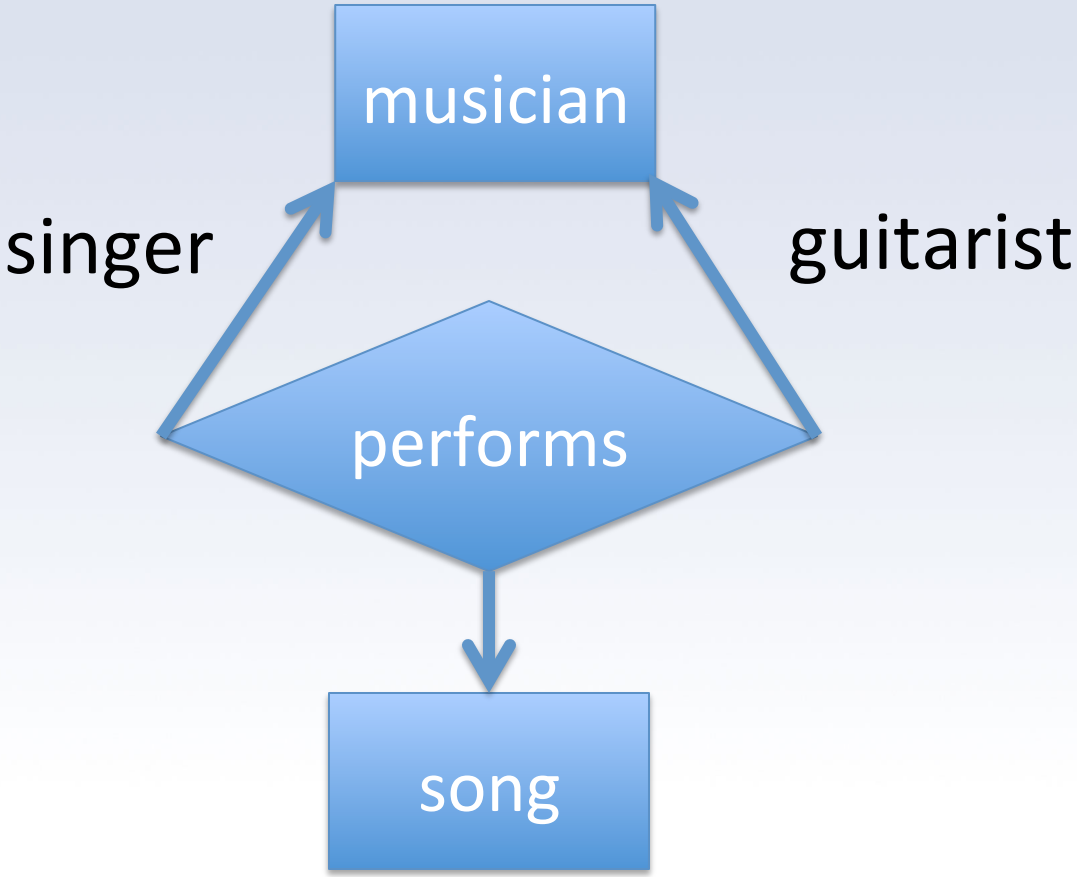
Example



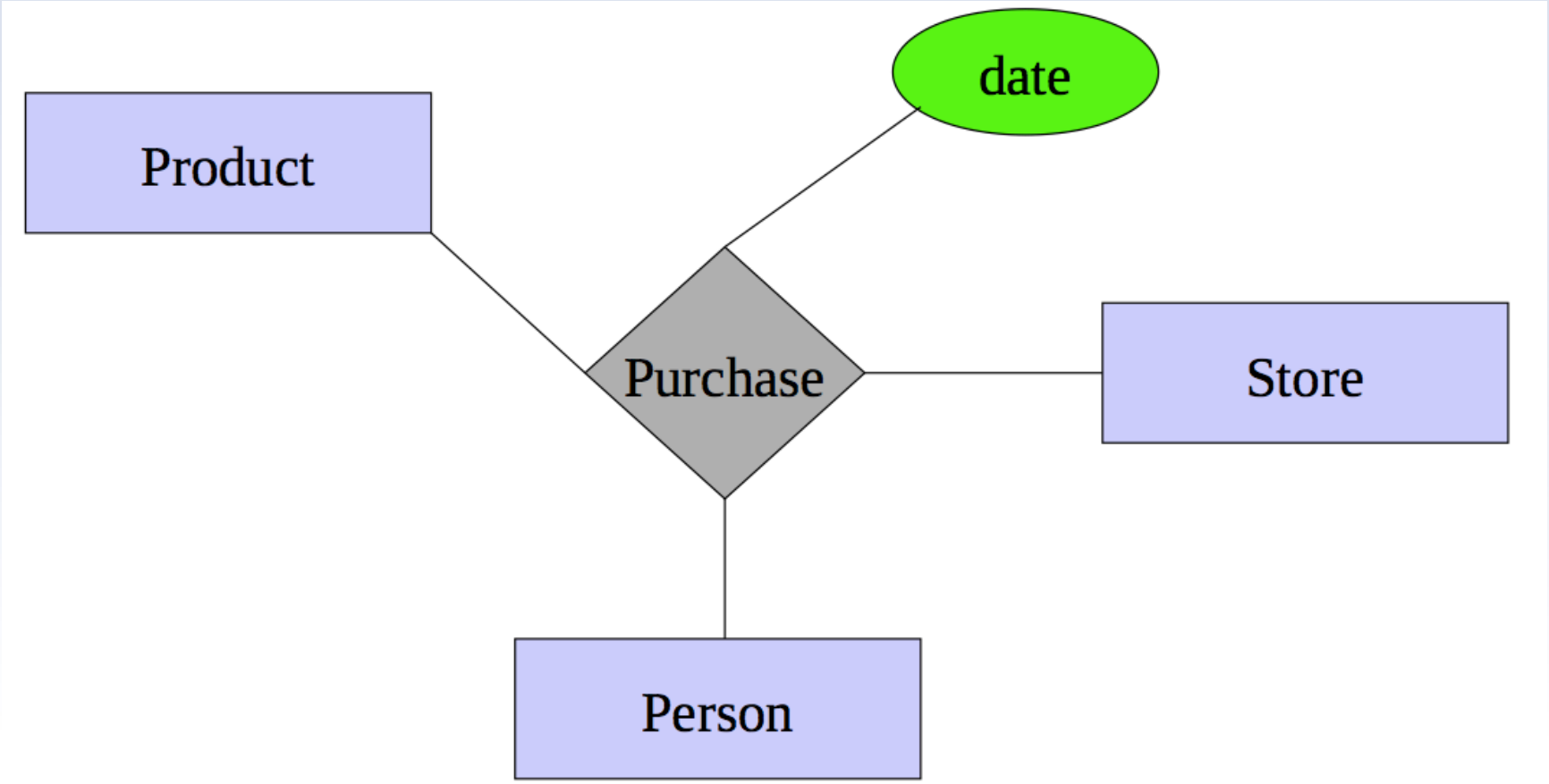
Roles

- Sometimes one entity set can be involved in more than one kind of relationship with another
- We establish “roles” by labeling the arrows of a relationship

Example



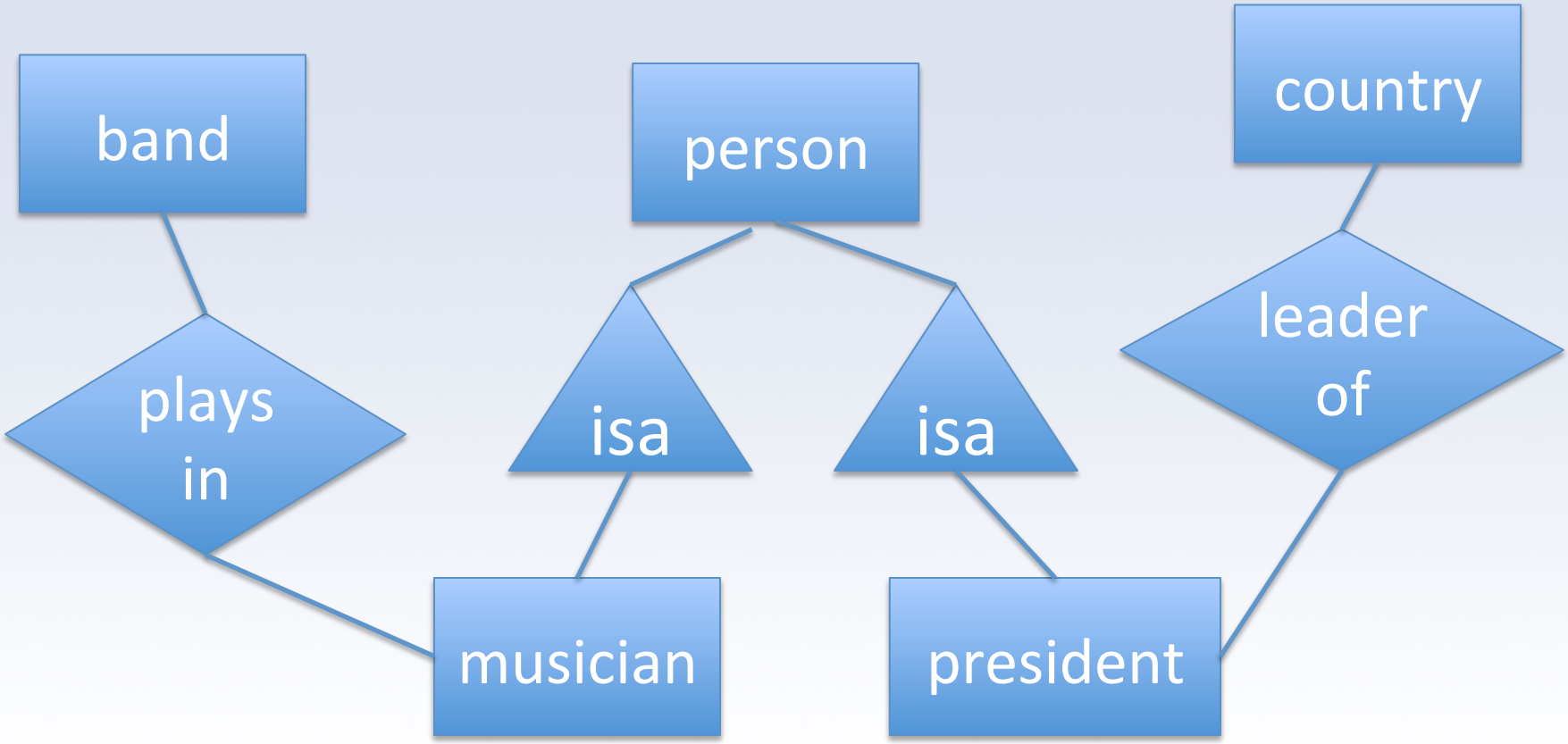
Relationship Attributes



Subclasses

- Sometimes, one entity set is a “kind” of a broader entity set
- Examples: a musician is a person, a customer is a person, a president is a person
- Use special “isa” (is a) relationship

Example

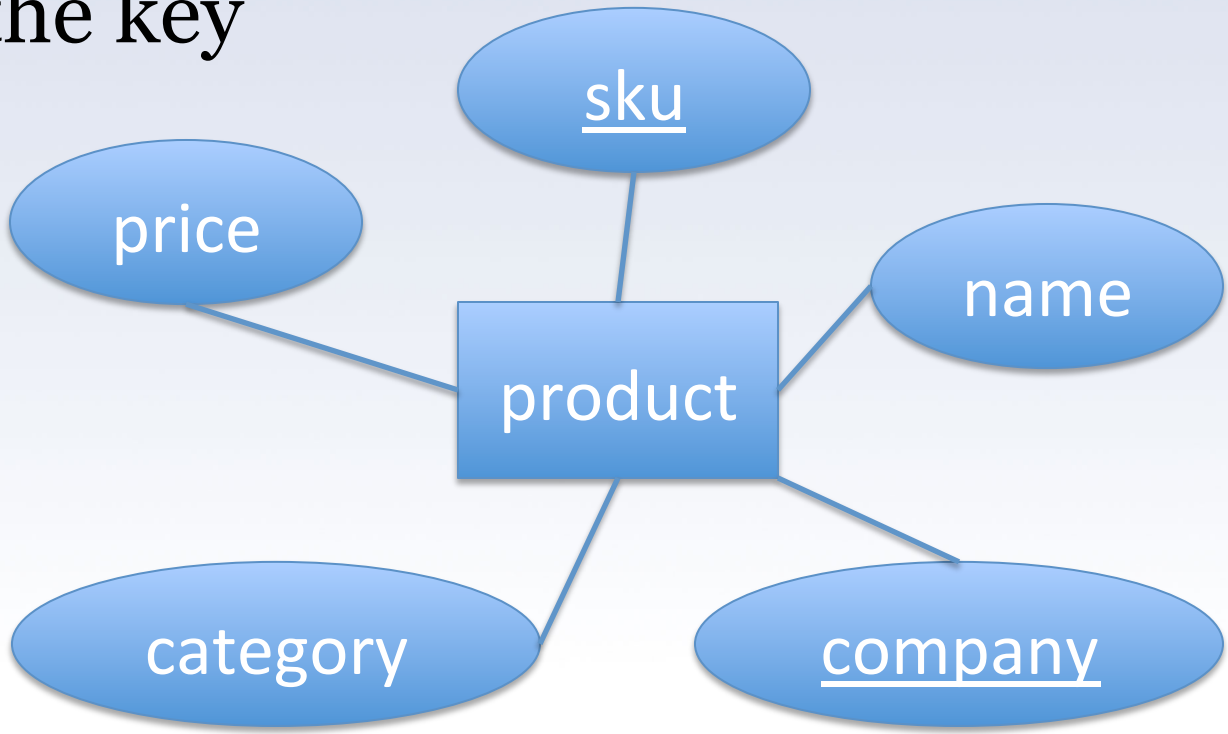

band
person
country
plays in
isa
isa
leader of
musician
president

Constraints in E/R Model

- We can represent 3 kinds of constraints
  1. Keys
  2. Referential integrity
  3. Degree

Key Constraints in E/R

- Simply underline the attributes which are in the key



Referential integrity

- Represent with a curved arrow

- Each product MUST be made by one company

Degree constraint

- Represented a condition on the edge

- A product must be made by less than 10 companies

Degree constraint

- Represented a condition on the edge

- A product must be made by exactly one company

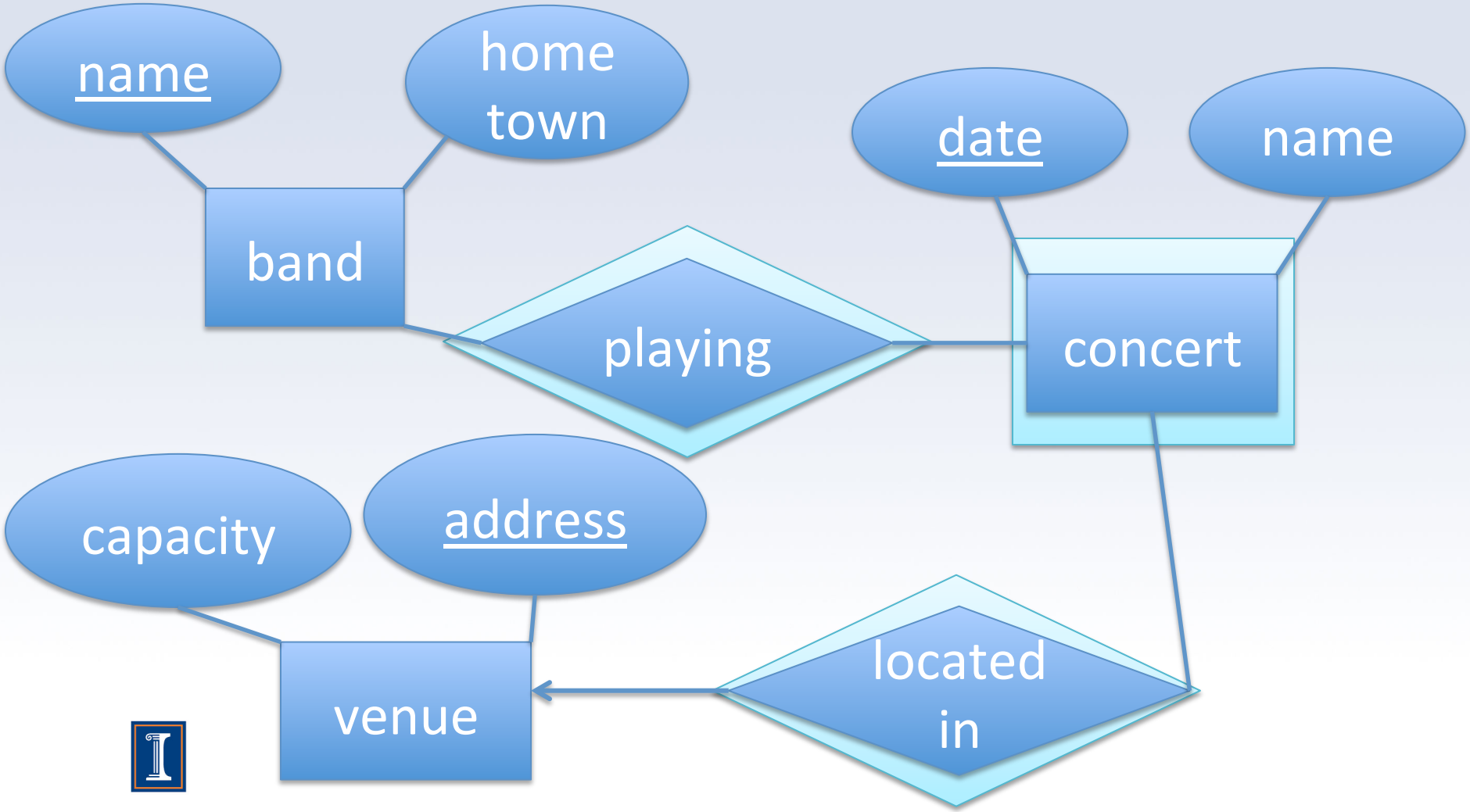
Weak Entity Sets

- Some entities can only be identified in terms of others
  - A student is identified by their school
  - A product is identified by its manufacturer
  - A street is associated with a city
- Need "help" getting its key from some other relation

Weak Entity Sets

- Indicate “weakness” of entity with double borders around it
- Indicate relationships that define weak entity with double borders around them

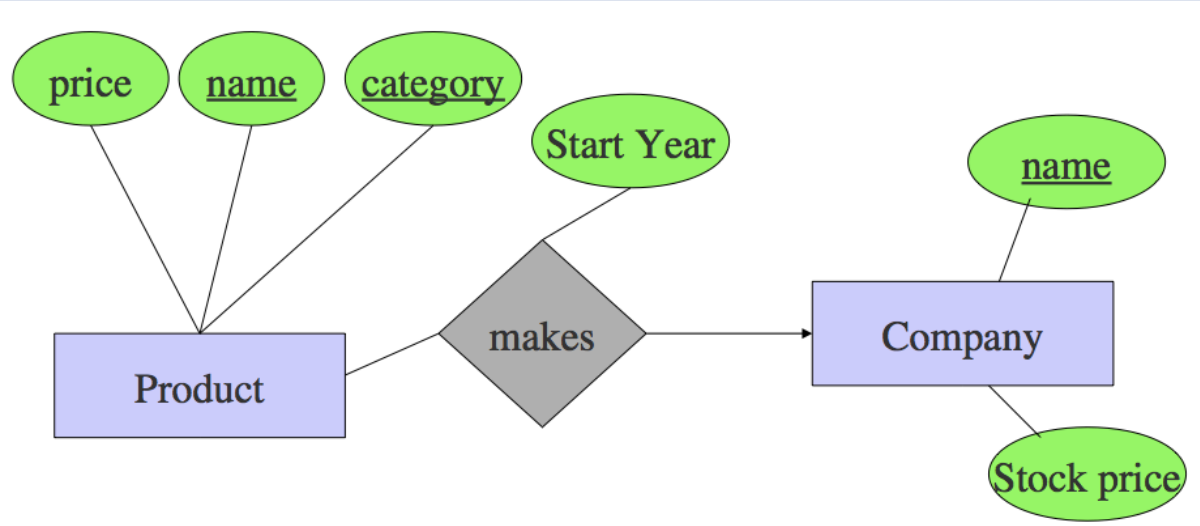
Example


name
home town
band
playing
date
name
concert
capacity
address
venue
located in

E/R → Relational Schema

- Turn each entity set into a relation
  - use attributes of entity as attributes of relation
- Turn each relationship into a relation whose attributes are the keys of connected entity sets

Example



Product(productName,category,price)

Company(companyName,stock)

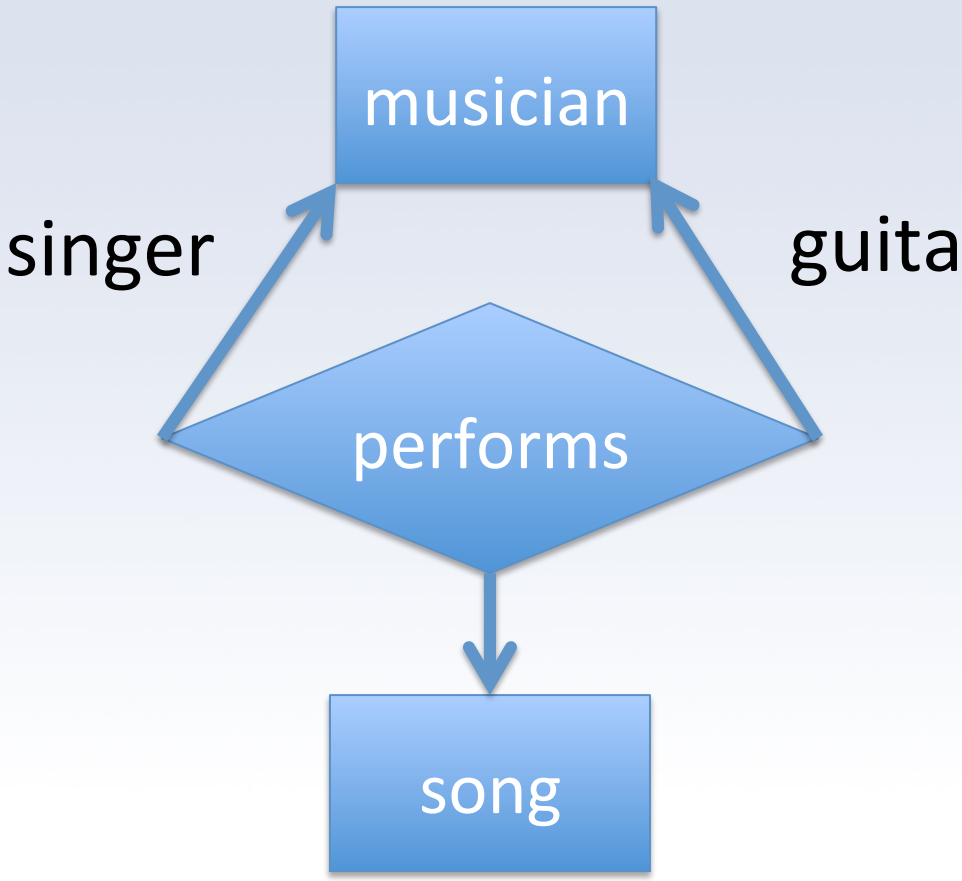
Makes(companyName,productName,startYear)

Example

Musician(name,…)

Song(name,…)

Performs(singerName,
guitaristName,
songName)



Some complications

- Weak entity sets
- “isa” relationships
- Combining relations