

UNIVERSITY OF ILLINOIS  
AT URBANA-CHAMPAIGN

# CS411 - E/R Models to Relational Schema



[illinois.edu](http://illinois.edu)

# Announcements

- Project Stage 1 due Feb 17
- HW2 Released tonight
- Midterm 1 March 1st



# Review

- What is the difference between 3NF and BCNF?
- What is a MVD?
- What is 4NF?
- What is denormalization?



# Multivalued Dependencies

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

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



# MVD (informally)

- Presence of one row in the table implies many other rows in the table
- Every time we see album “Contra” in a row, there were rows for all of its songs
- $R(\text{name, phone, songsLiked})$ 
  - Someone could have multiple phone numbers
  - Every time we see their name, we see all of their phone numbers



# Example

R(course,instructor,textbook)

If a certain course always uses the same set of textbooks...



# Example

course	Semester	instructor	textbook
CS373	SP13	Viswanathan	Hopcroft, Ullman
CS373	SP13	Viswanathan	Sipser
CS373	SP12	LaValle	Hopcroft, Ullman
CS373	SP12	LaValle	Sipser
CS373	SU11	Cook	Hopcroft, Ullman
CS373	SU11	Cook	Sipser
CS373	SU12	Cunningham	Hopcroft, Ullman
CS373	SU12	Cunningham	Sipser





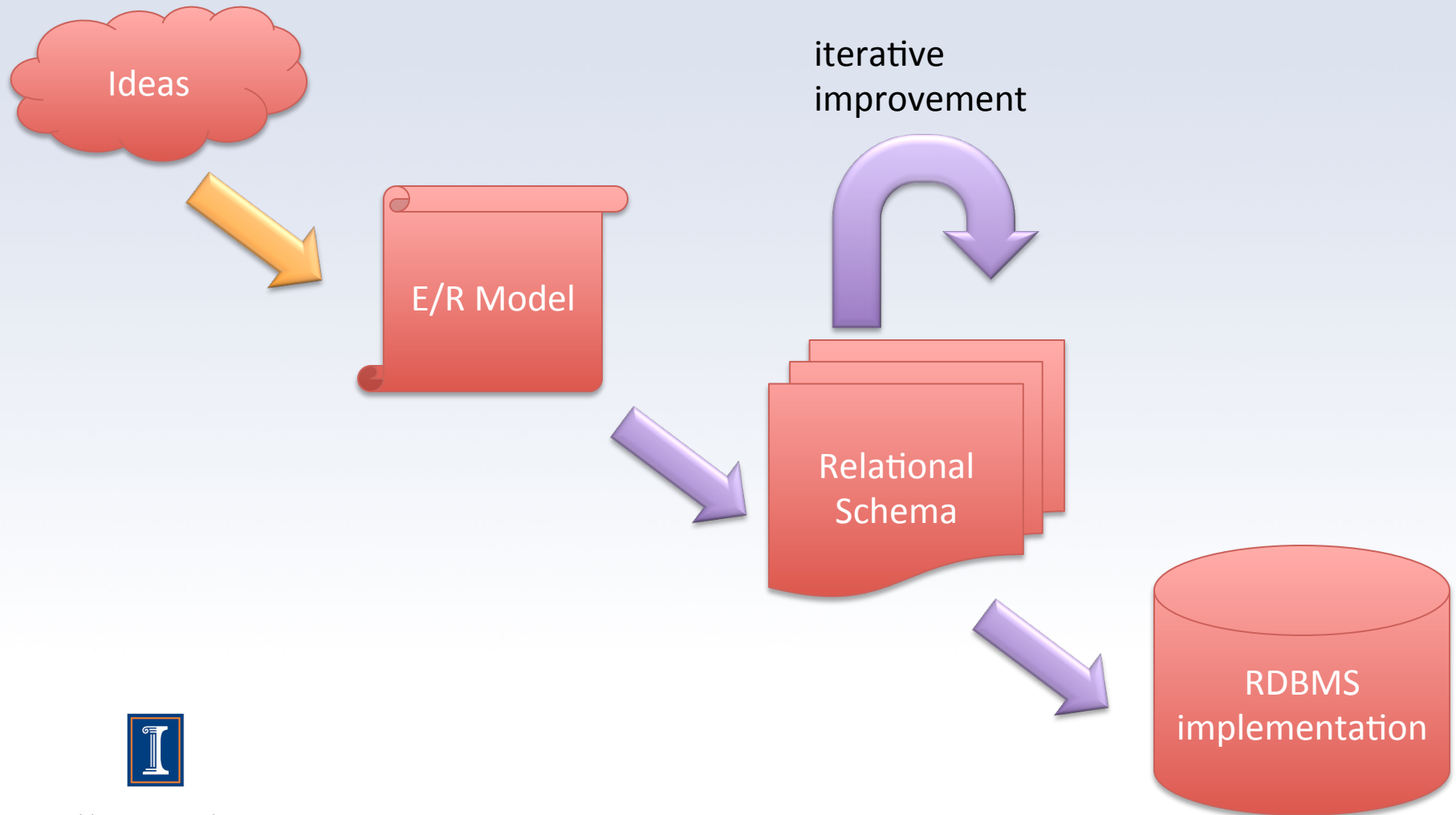
# Review

- What is E/R modeling?
- What are the components of an E/R model?
- What do arrows mean in an E/R diagram?

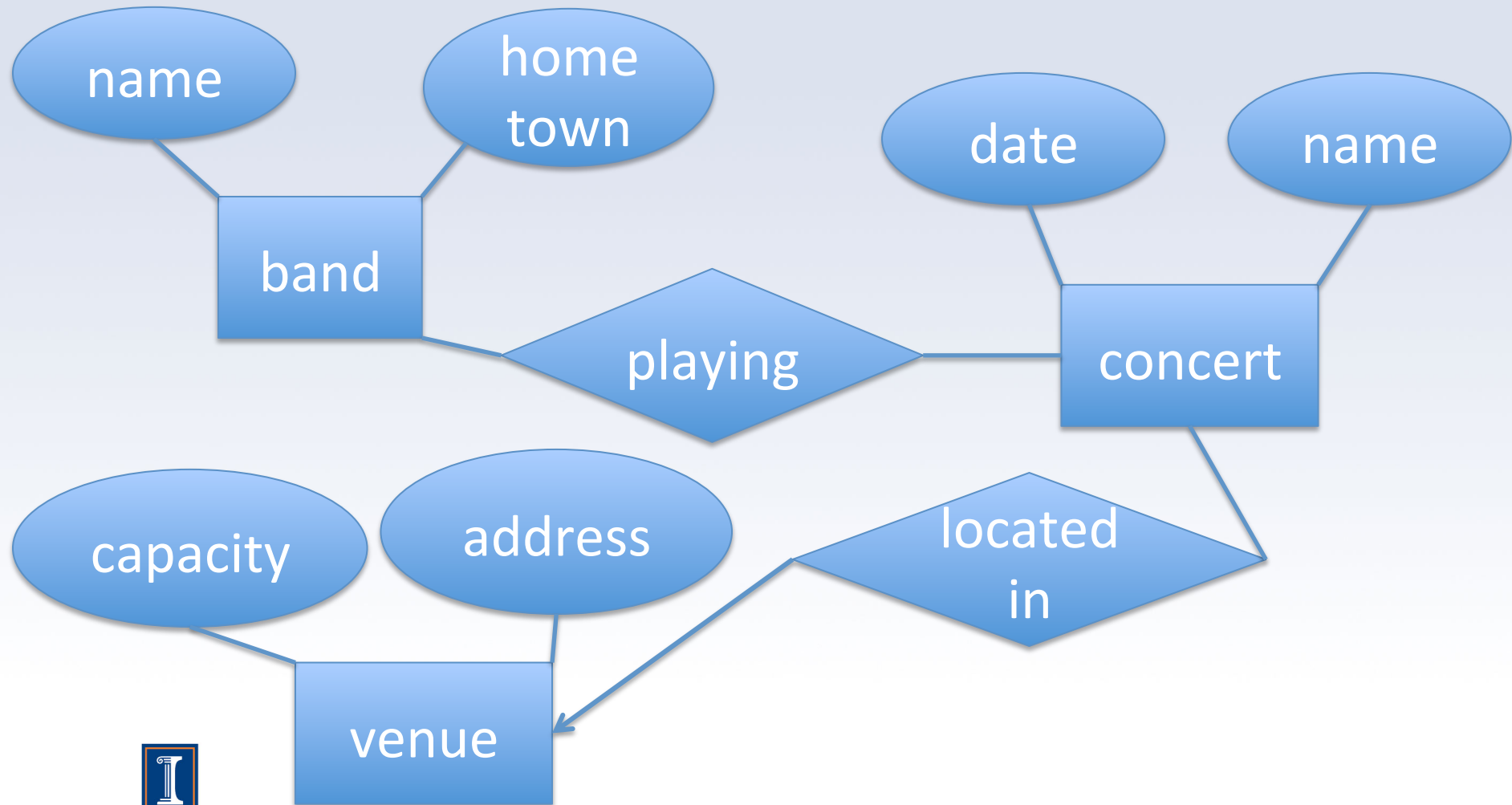




# Design Process



# E/R Diagram



# 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

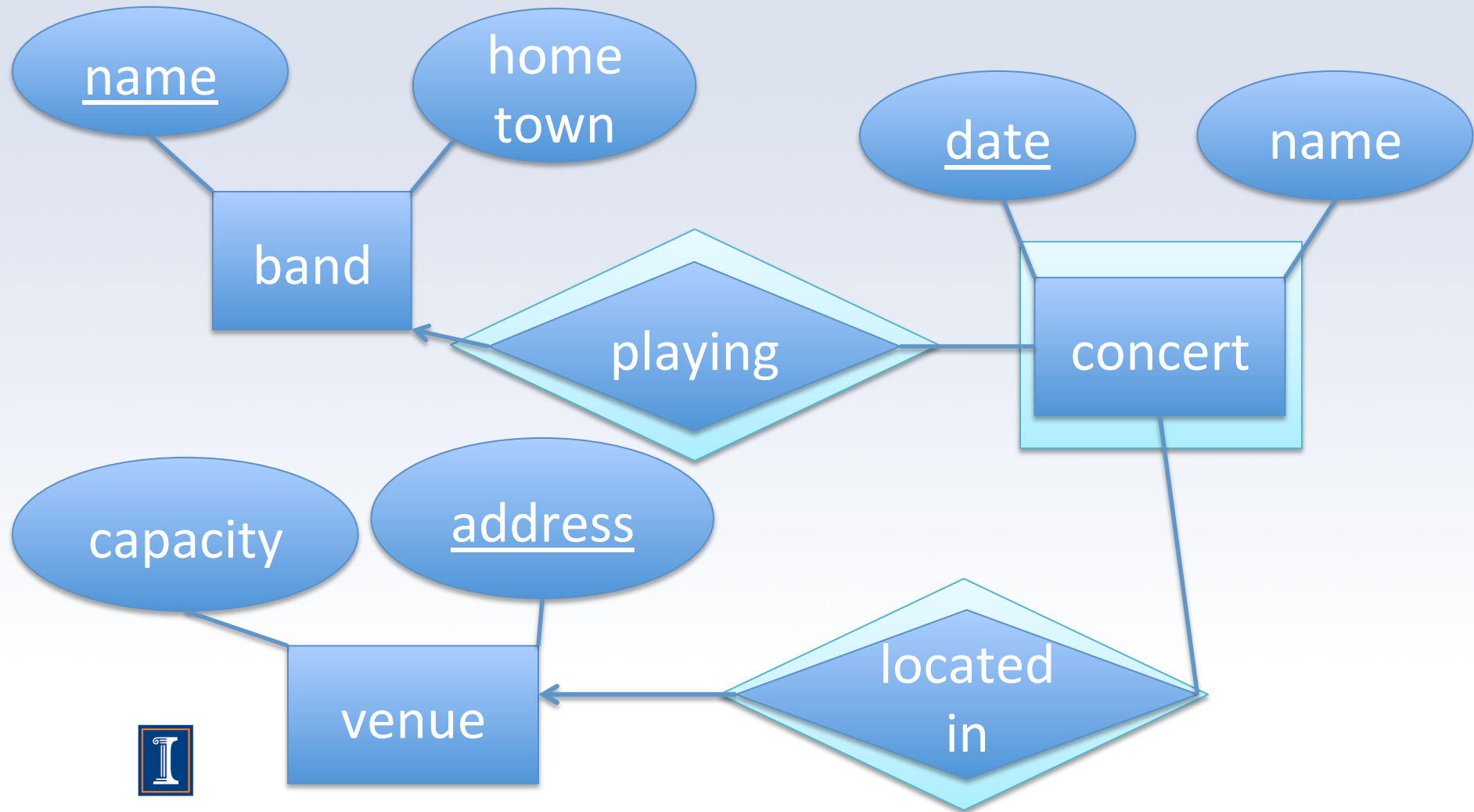


# Weak Entity Sets

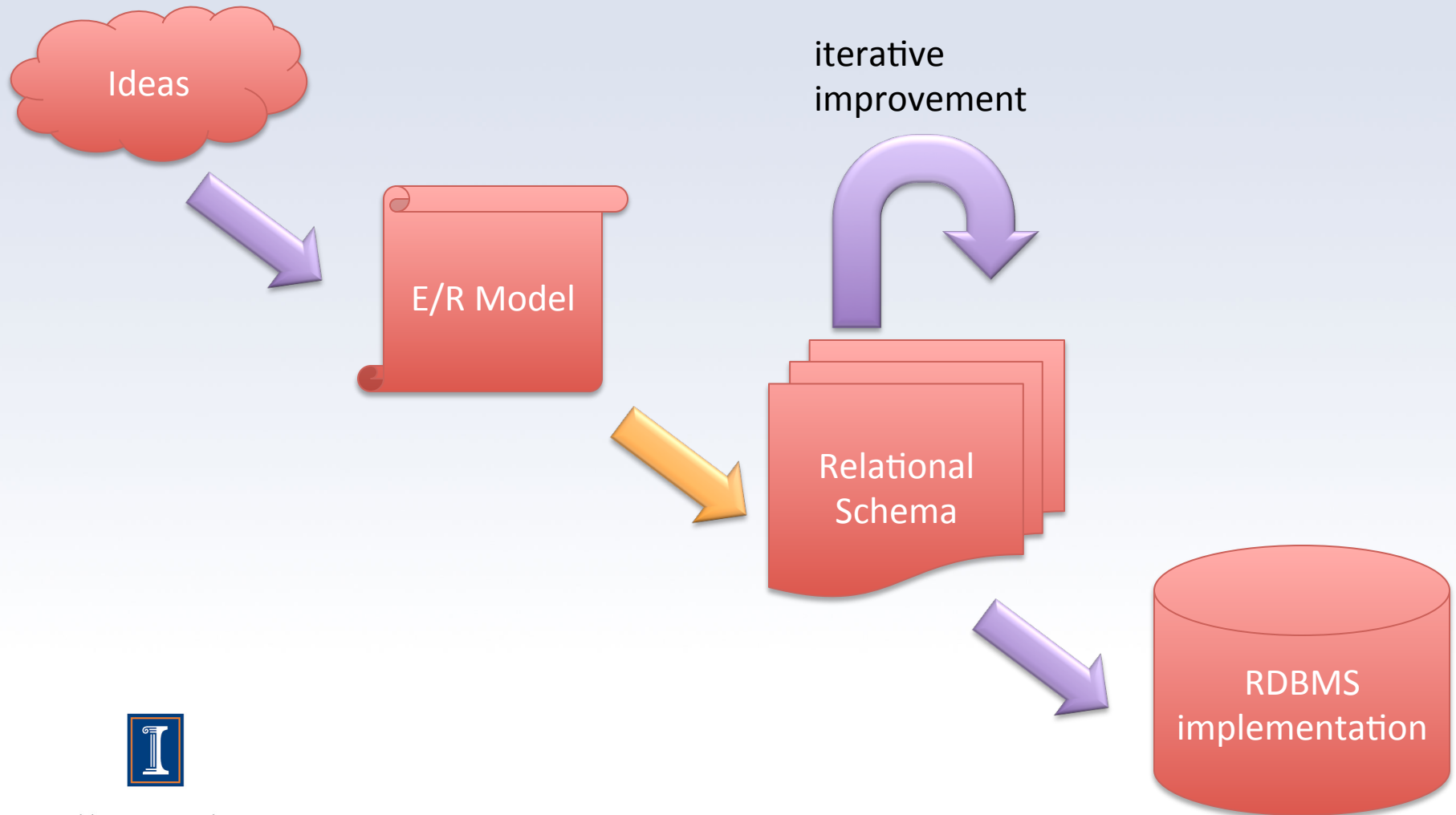
- Indicate “weakness” of entity with double borders around it
- Indicate relationships that define weak entity with double borders around them



# Example



# Design Process



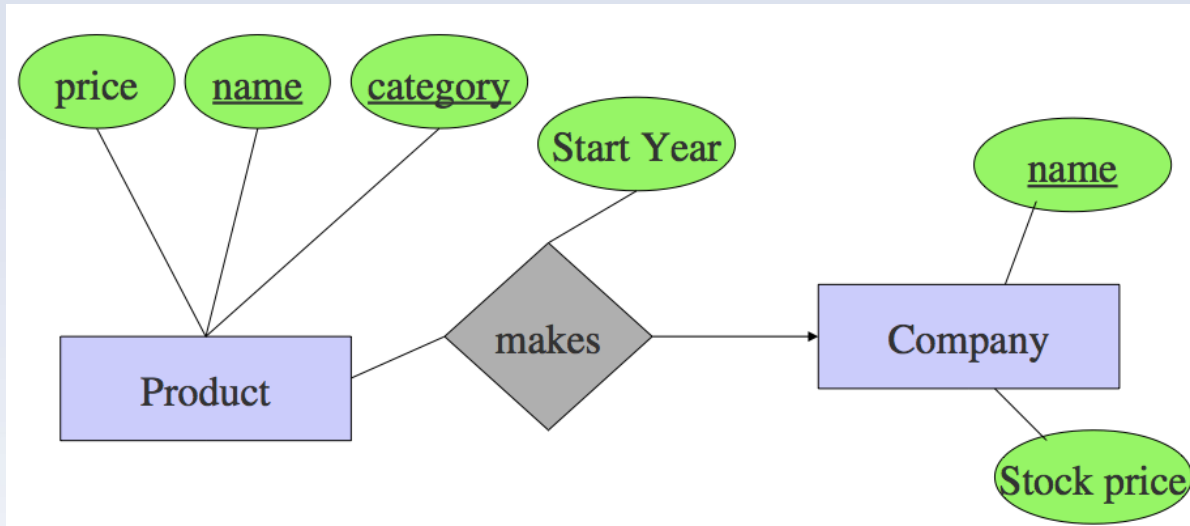
# E/R $\rightarrow$ Relational Schema

- 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)

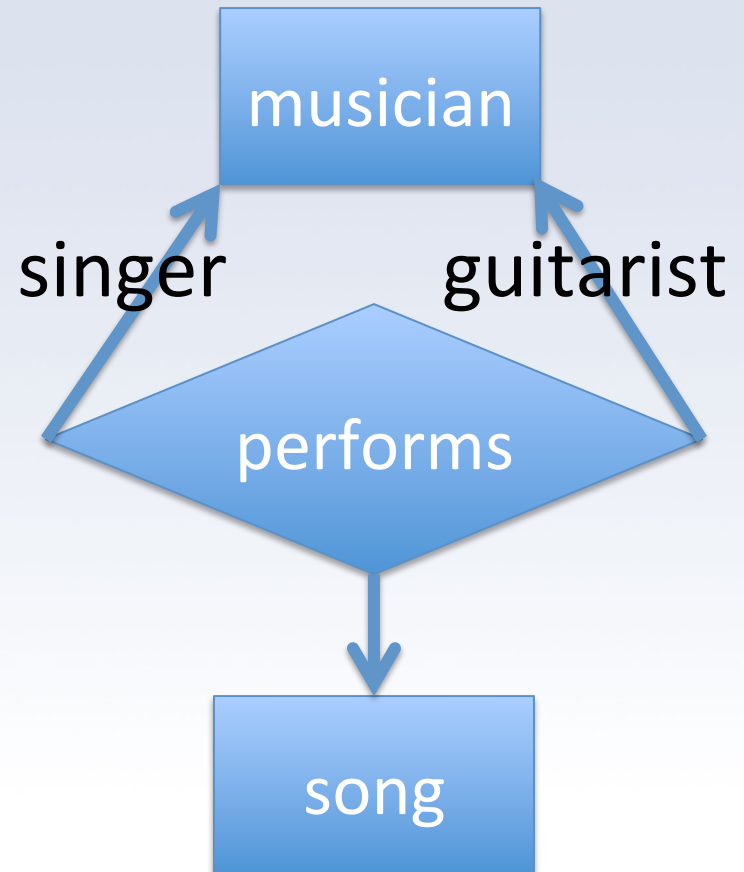


# Example

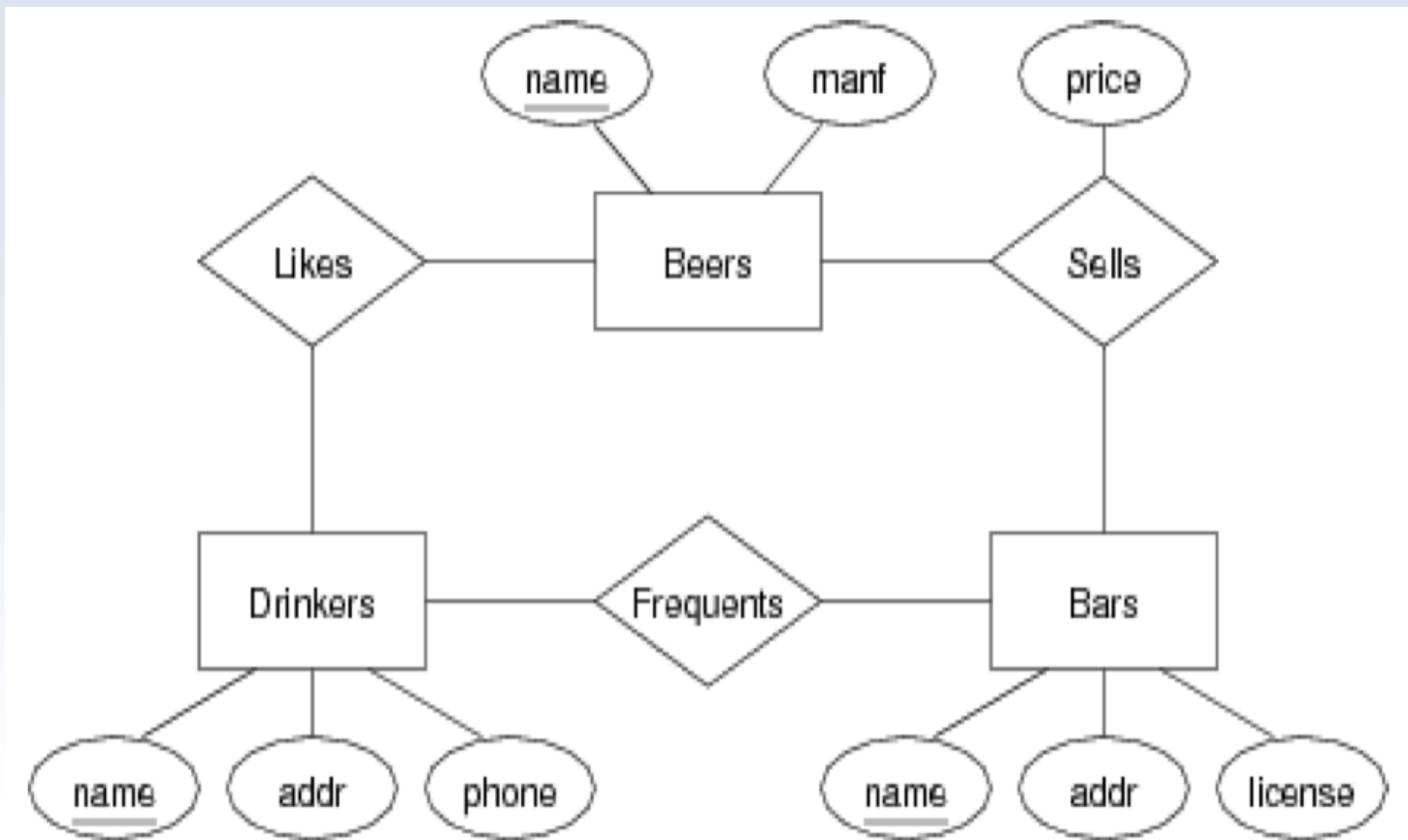
Musician(name,...)

Song(name,...)

Performs(singerName,  
guitaristName,  
songName)



# Example



# Example

Drinkers(name,addr,phone)

Bars(name,addr,license)

Beers(name,manf)

Likes(drName,beName)

Frequents(drName,baName)

Sells(baName,beName,price)



# Some complications

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

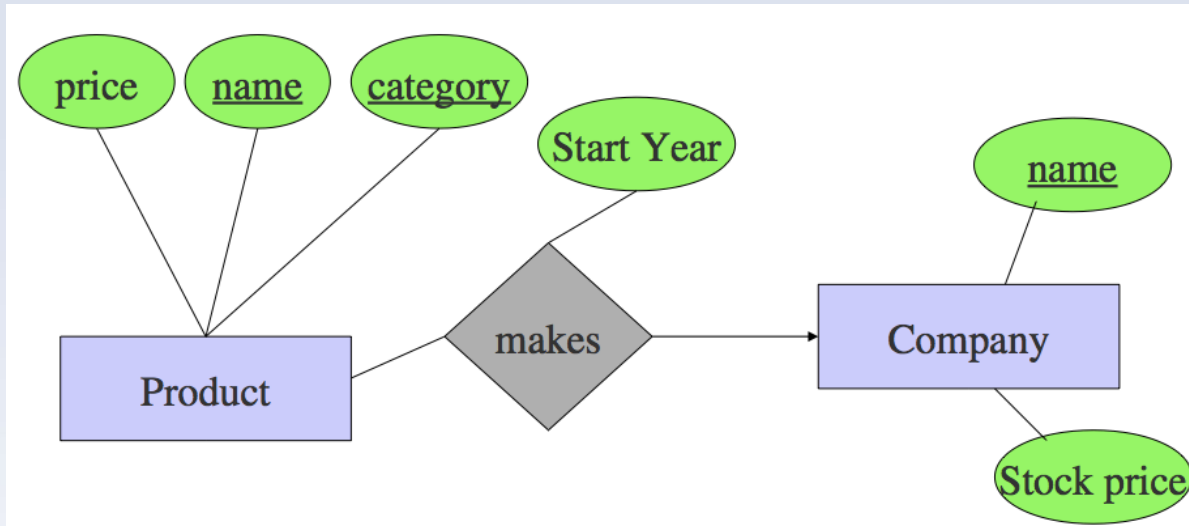


# Combining many-one relations

- Many-one relationships sometimes produce unnecessary relations
- We can move the attributes of these relations into the “many” entity of the relationship



# Example



Product(productName,category,price)

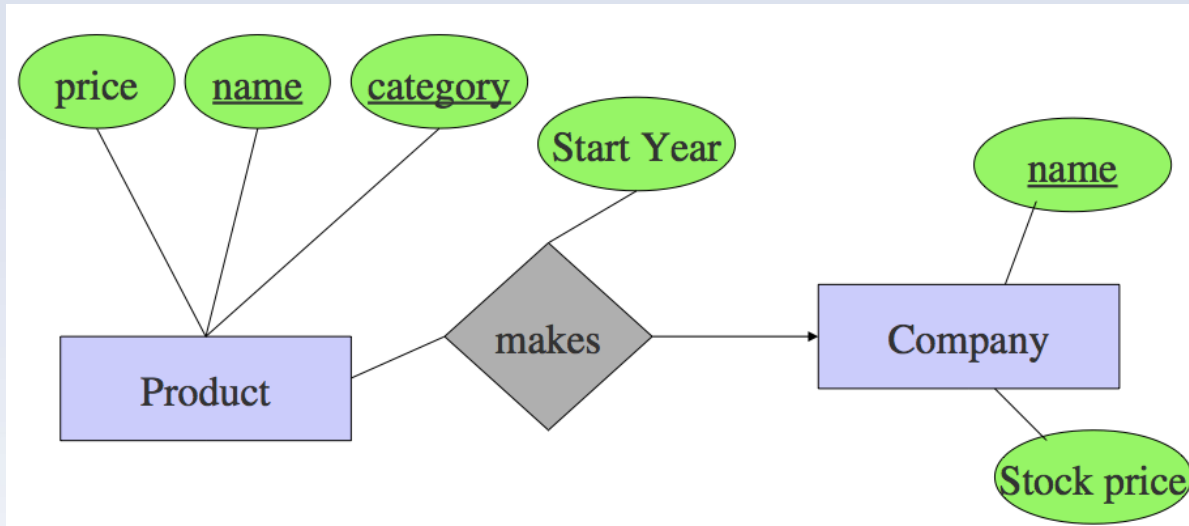
Company(companyName,stock)

Makes(companyName,productName,startYear)





# Example



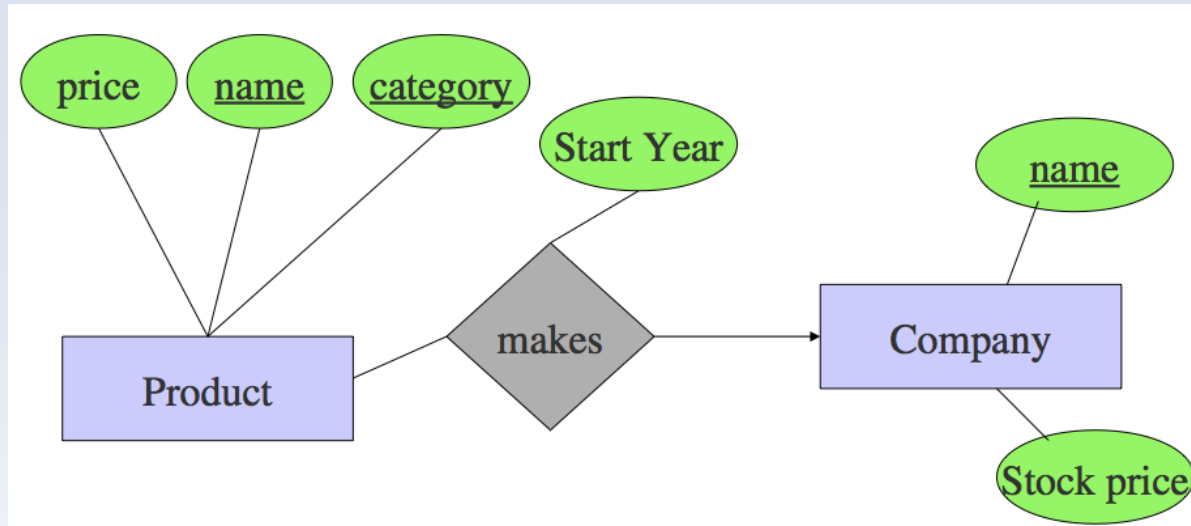
Product(productName,category,price)

Company(companyName,stock)

~~Makes(companyName,productName,startYear)~~



# Example



Product(productName,category,price,  
companyName,startYear)

Company(companyName,stock)



# Example

Drinkers(name,addr,phone)

Beers(name,manf)

Favorite(drName,beName)



# Example

Drinkers(name,addr,phone,favBeer)

Beers(name,manf)

~~Favorite(drName,beName)~~

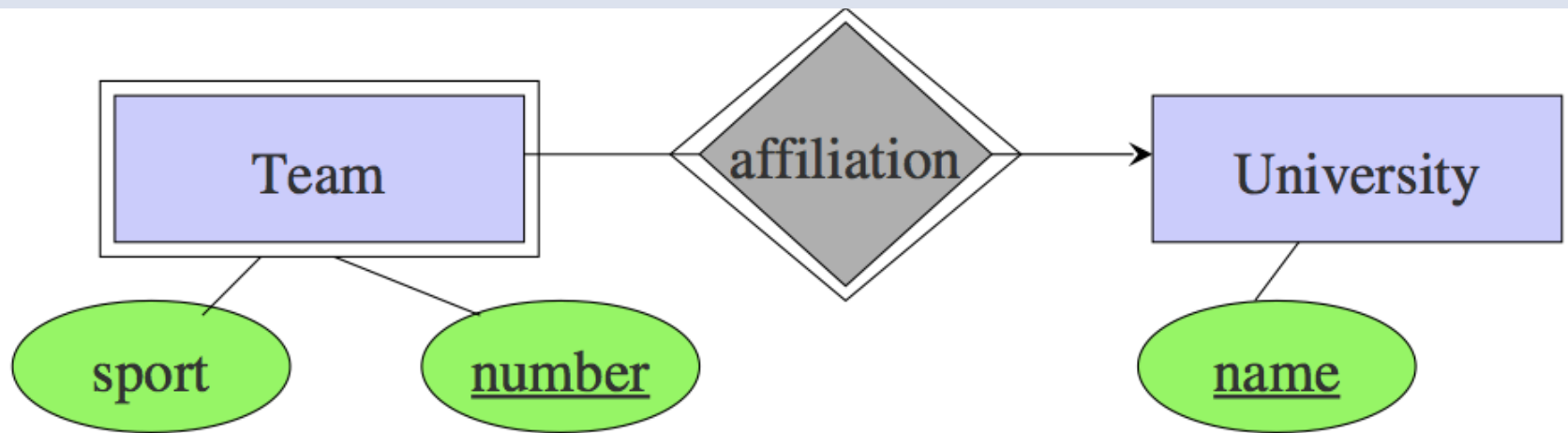


# Weak Entity Sets

- Relation for weak entity set contains the keys of the supporting (non-weak) entities
- Supporting (double diamond) relationship doesn't need a relation



# Example

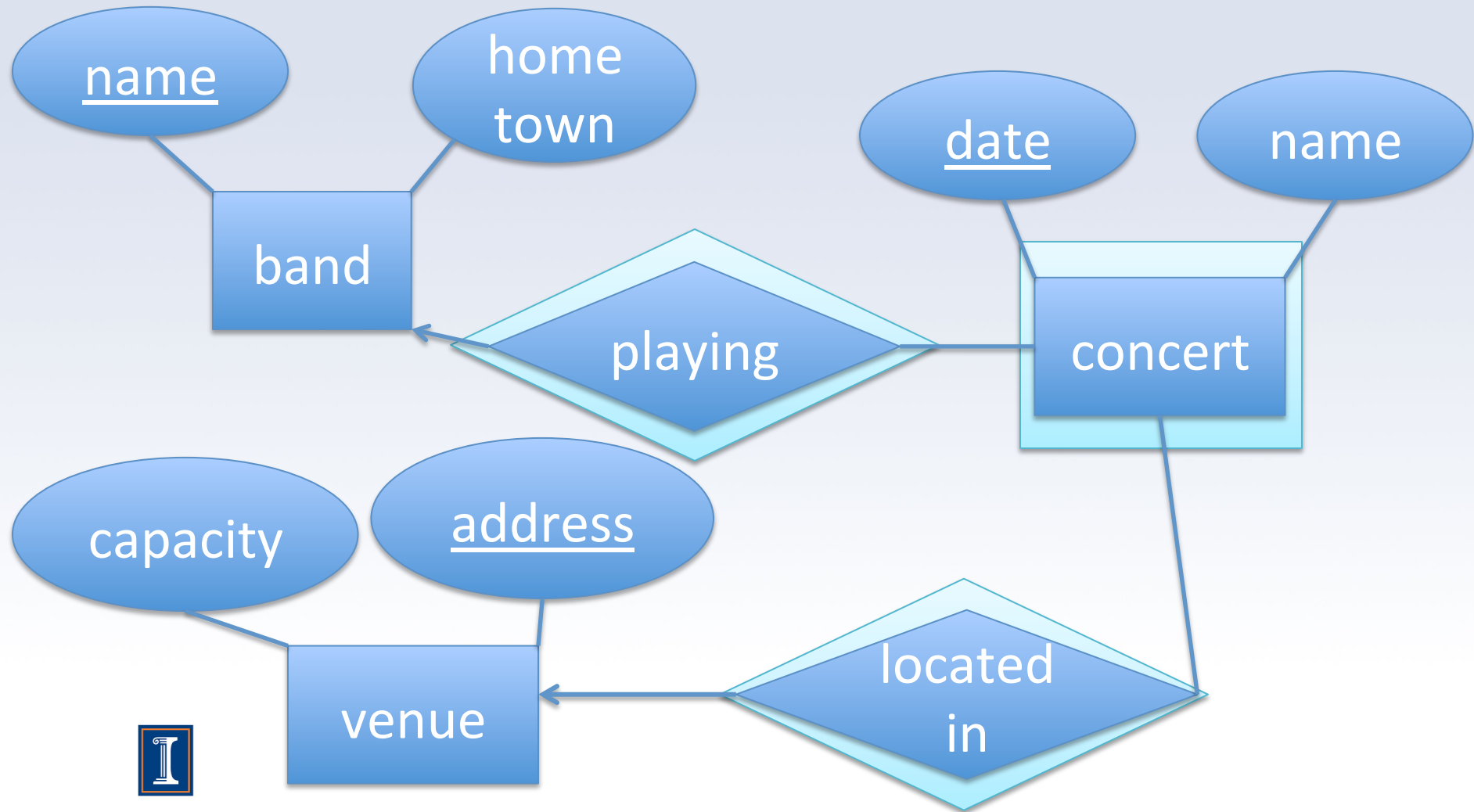


Relation **Team**:

Sport	Number	Affiliated University
mud wrestling	15	Montezuma State U.



# Example





# Example

Band(name,homeTown)

Venue(address,capacity)

Concert(bandName,address,date,name)



# “is a” relationships

- We have three different options:
  1. E/R Approach
  2. Object-oriented Approach
  3. Null values Approach

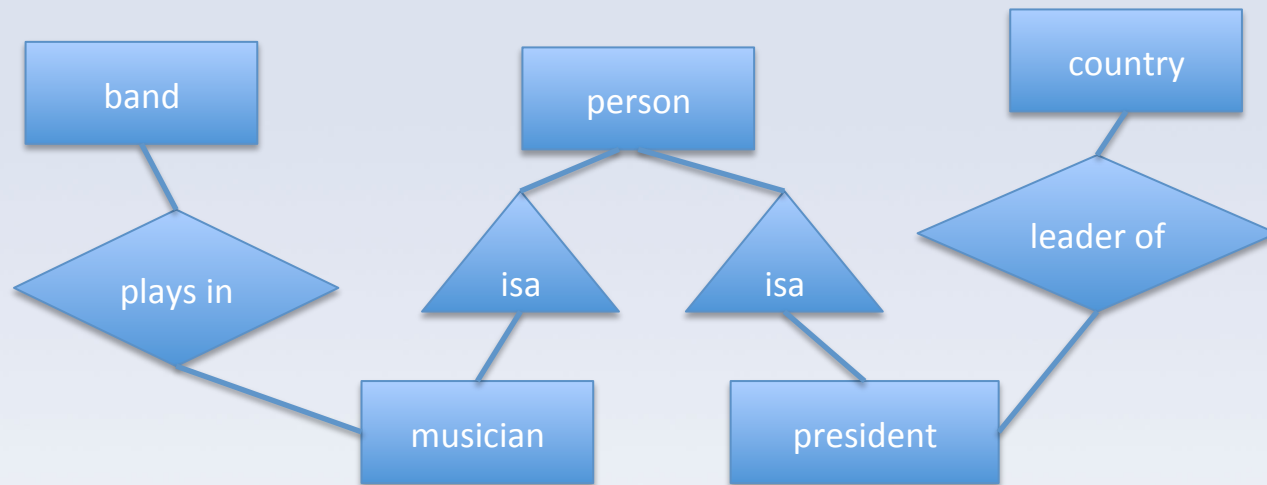


# 1) E/R Approach

- Create relations for each of the entities, just like we usually do
  - For “subclass” entities, include key of superclass



# Example



Person(ssn,name,birthday,...)

Musician(ssn,instrument,bandName,...)

President(ssn,yearElected,countryName,...)

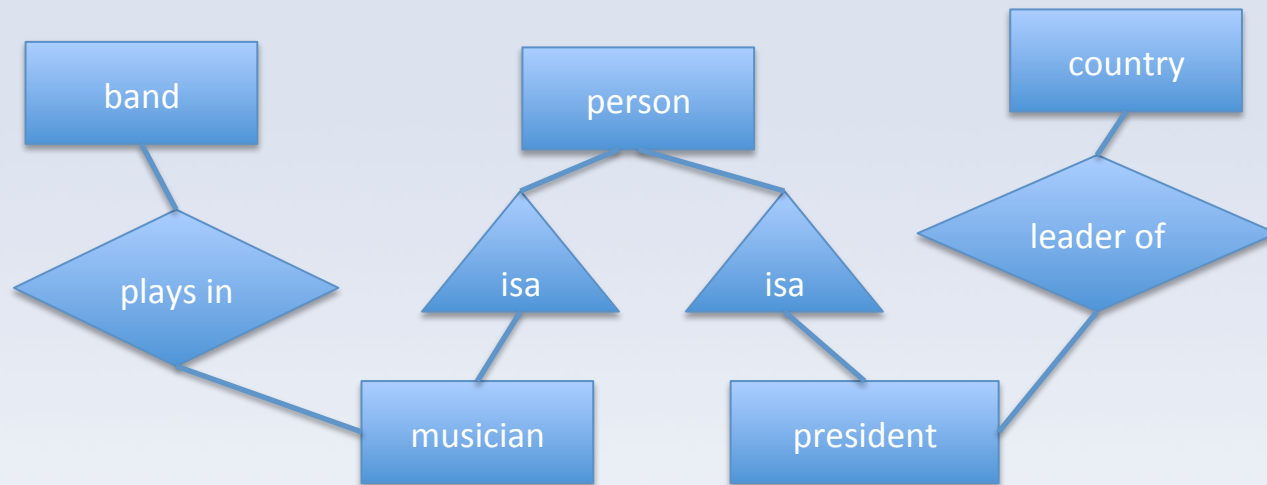


## 2) Object-oriented Approach

- Create a relation for all possible subtrees
- The attributes of the relation correspond to *all possible attributes* of entities on that subtree



# Example



Person(ssn,name,birthday,...)

Musician(ssn,name,birthday,instrument,bandName,...)

President(ssn,name,birthday,yearElected,countryName,...)



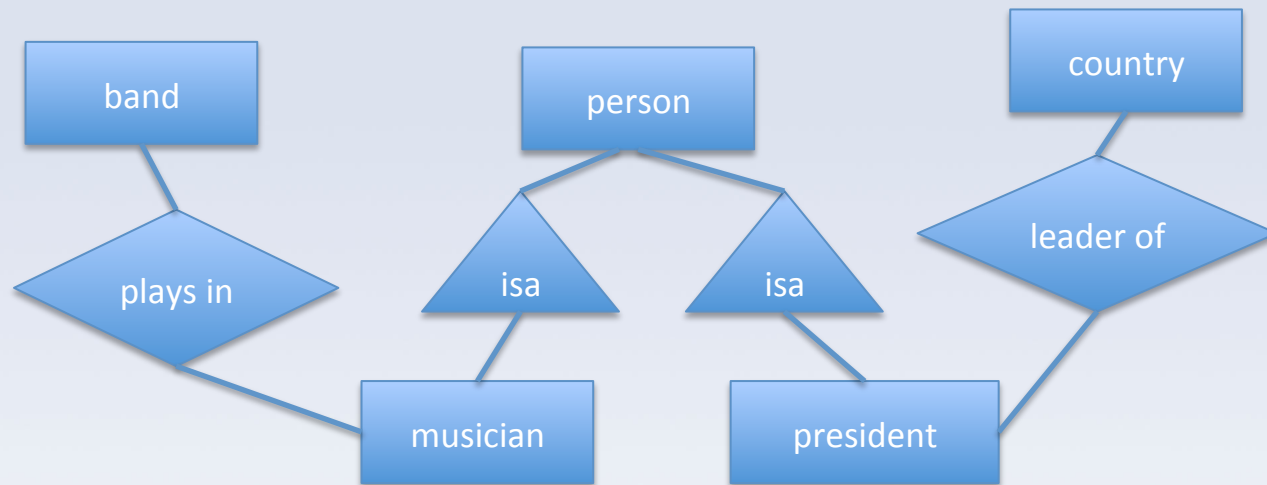
# NULL Value Approach

- Create one relation for all the subtrees
- If an attribute doesn't apply to a subclass, tuples in that subclass will just leave it NULL





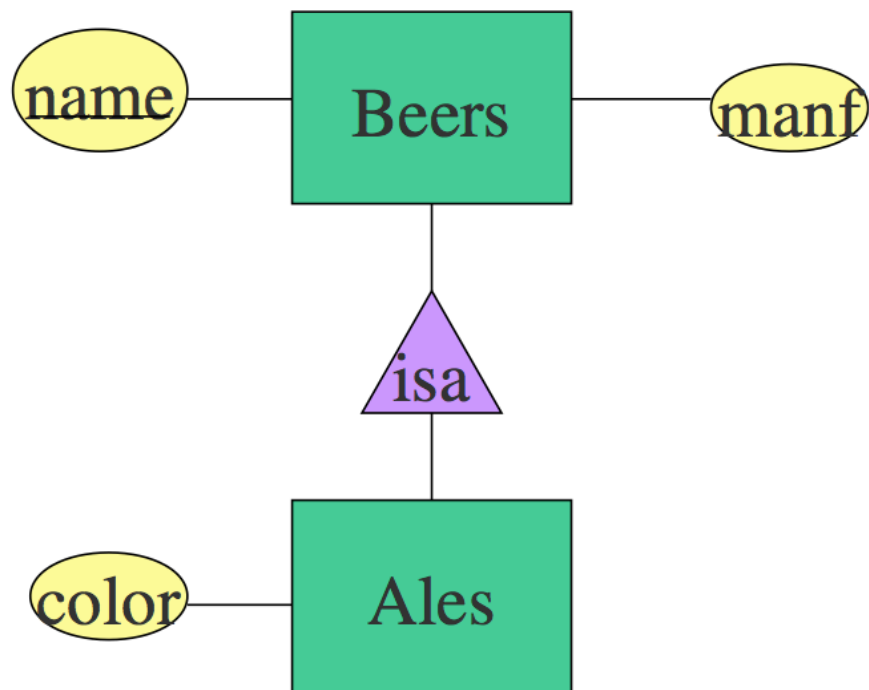
# Example



Person(ssn,name,birthday,instrument,  
bandName,yearElected,countryName...)



# Example



# Example

## E/R Approach

Beers(name,manf)

Ales(name,color)

## OO Approach

Beers(name,manf)

Ales(name,manf,color)

## NULL Approach

Beers(name,manf,color)



# Example

E/R Approach

Beers(name,manf)

Ales(name,color)

OO Approach

Beers(name,manf)

Ales(name,manf,color)

NULL Approach

Beers(name,manf,color)

name	manf
Bud	Busch
Summerbrew	Pete's

name	color
Summberbrew	dark



# Example

E/R Approach

Beers(name,manf)

Ales(name,color)

OO Approach

Beers(name,manf)

Ales(name,manf,color)

NULL Approach

Beers(name,manf,color)

name	manf
Bud	Busch

name	brewer	color
Summberbrew	Pete's	dark



# Example

E/R Approach

Beers(name,manf)

Ales(name,color)

OO Approach

Beers(name,manf)

Ales(name,manf,color)

NULL Approach

Beers(name,manf,color)

name	manf	color
Bud	Bush	NULL
Summerbrew	Pete's	dark



# Comparison

- E/R approach
  - Pros:
    - Potentially fewer relations than OO approach
    - Only key attributes are repeated across relations
  - Cons:
    - More relations than NULL approach
    - Requires a joining to recover information



# Comparison

- Object oriented approach
  - Pros:
    - Only one tuple per entity (maximal space usage)
  - Cons:
    - Potentially many relations for deeply nested hierarchies
    - Have to search all relations for desired information





# Comparison

- NULL:
  - Pros:
    - All attributes available in one relation
  - Cons:
    - Possibly many useless “NULL” values in each tuple
    - Meaning of “NULL” overloaded
      - NULL because I don’t know musician’s instrument
      - Or NULL because they’re not a musician

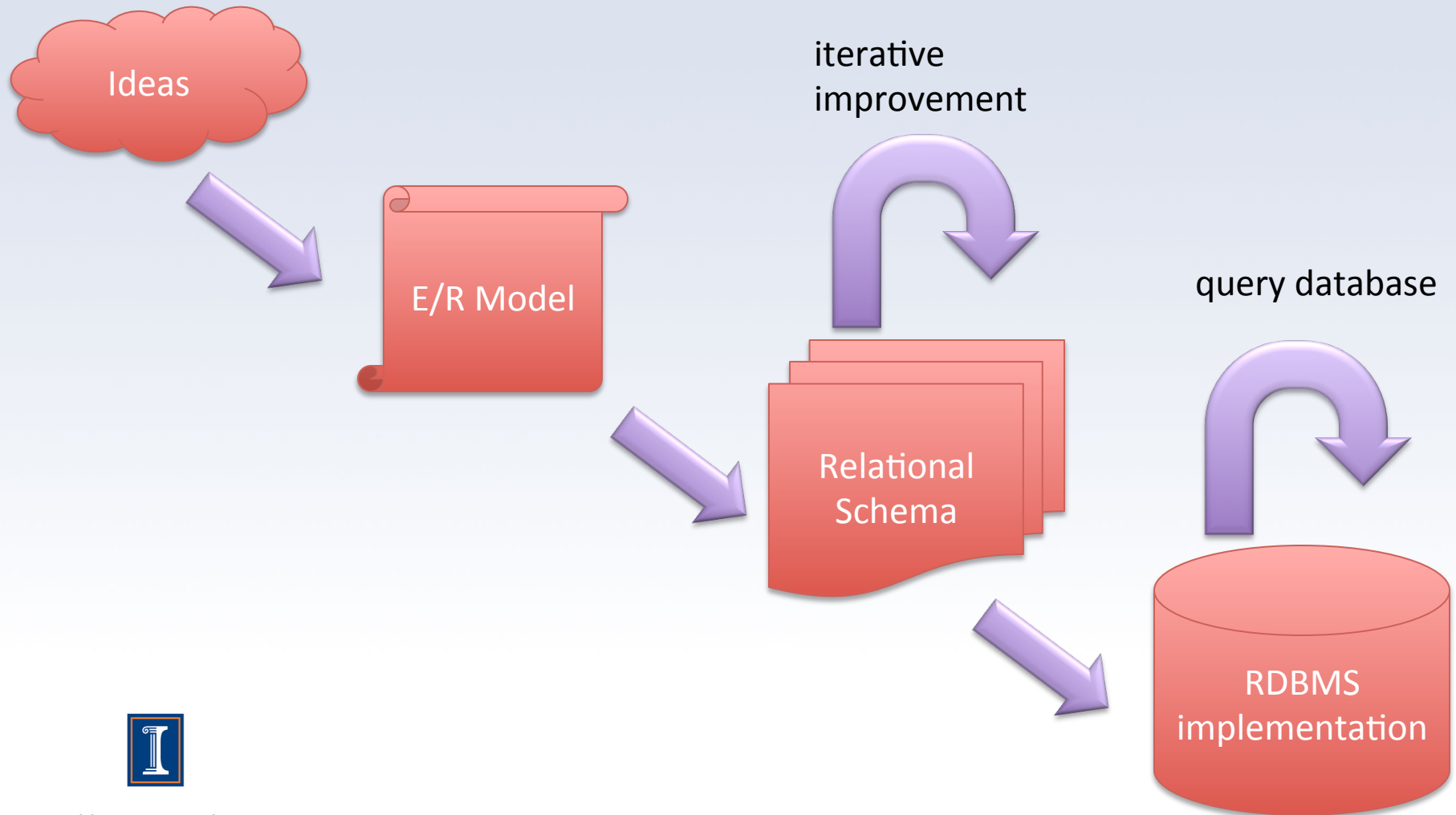


# Now what?

- We can start from an idea
- Design an relational database
- Answer questions using queries



# Design Process



# Now what?

- We can start from an idea
- Build an relational database
- Answer questions using queries
- ***Aren't we done now?!***



# RDBMSs can do much more

- Can model dynamic data
  - we can insert and deleted tuples
  - we can update values
  - transaction interaction can be controlled
  - automatic triggers can update database when a condition is met



# RDBMSs can do much more

- Can model many constraints
  - Referential integrity, value constraints, assertions
- Can create relations that don't exist on disk - views
- Can optimize structure of data to enable faster queries - indexing



# RDBMSs can do much more

- We'll start learning how to do these things in SQL next week
- Let's start with a taste...



# Database Modification

- Insert tuples into a relation
- Delete tuples from a relation
- Update values of certain tuples





# Insertion

- Add tuples to an existing relation R
- Written like this:

INSERT INTO R( $A_1, A_2, \dots, A_n$ )

VALUES ( $v_1, v_2, \dots, v_n$ )

- Attribute list helps us know which values correspond to which attributes



# Example

INSERT INTO Album(title,released,band)  
VALUES

(“Nevermind”, 1991, “Nirvana”),

(“Bleach”, 1989, “Nirvana”),

(“In Utero”, 1993, “Nirvana”);



# Subqueries are allowed

```
INSERT INTO Band(name)
  SELECT DISTINCT bandName
  FROM Album
  WHERE bandName NOT IN
    (SELECT name from Band);
```



# Deletion

- Delete tuples from a relation R
- Written like this:

DELETE FROM R WHERE <condition>;

- If <condition> is true, we delete the tuple



# Example

```
DELETE FROM Album  
WHERE bandName = 'Nickelback' OR  
bandName = 'Creed';
```



# Subqueries allowed

```
DELETE FROM Album  
WHERE bandName IN  
  (SELECT DISTINCT bandName  
   FROM Band NATURAL JOIN ON Musician  
   WHERE instrument="Keytar");
```



# Subqueries allowed

```
DELETE FROM Album  
WHERE bandName IN  
  (SELECT DISTINCT bandName  
   FROM Band NATURAL JOIN ON Musician  
   WHERE instrument="Keytar");
```



# Updates

- We can change the values of attributes for tuples in relation R
- Written like this:  
UPDATE R SET <new-values>  
WHERE <condition>
- <new-values> is a list like this:  
“ $A_1=v_1, A_2=v_2, \dots, A_n=v_n$ ”





# Example

UPDATE Album

SET bandName= 'The ' || bandName

WHERE bandName IN

('Beatles', 'Police', 'Verve', 'Antlers', 'Who',  
'Doors', 'Shins', 'Smiths', 'Strokes');

