CSC 261/461 – Database Systems
Lecture 8

Spring 2018

Announcement

- Quiz
  - No New Problem Set
  - Study Chapter 5, 6, and 7
  - Go through the problem set

Announcement

- Project 2 Part 1
  - Already out.
  - Workshop covered the basics
  - Objective: Applying SQL queries on ‘real’ data
  - Secondary Objective: Get proficient with Database Design and ER diagram

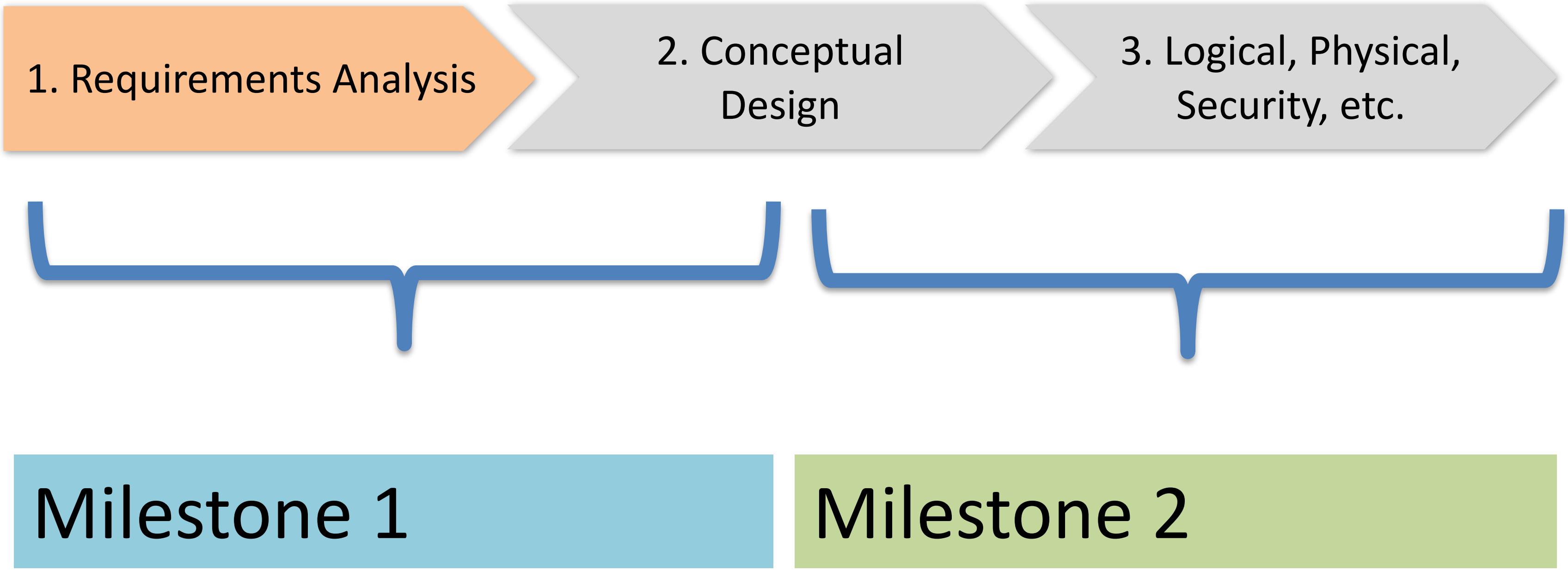
- Project 1 Milestone 2
  - Structure is closely related to Project 2 Part 1 (though, No SQL coding involved)
  - You should start from scratch (though, keep in mind what you have done in Milestone 1)

Start from Scratch!

What does it mean?

- This time, start with the ER diagram
- Feel free to add more entity sets or relations, add more attributes or remove attributes as required.
- Generate the resultant tables/relations from the ER diagram
  - (NOTE: This tables/relations may be significantly different from what you had come up with in Project 1 Milestone 1. And that is absolutely fine!!!
  - And, that is, in fact, the purpose.

Database Design Process



1. Requirements Analysis

2. Conceptual Design

3. Logical, Physical, Security, etc.

Milestone 1

Milestone 2

Q & A

- How are Project 2 part 1 and Project 1 Milestone 1 related?

- Project 2 Part 1 and Project 1 Milestone 2 both deals with ER diagram

- After this week, we will not talk about ER diagram a lot.
- You need to study ER diagram for the next week's quiz too.
- Both this projects give you a chance to apply ER diagrams in real world scenarios.

Q & A

- Piazza:
  - Not many questions are asked!
  - Please avoid marking questions as private

Q & A

Reminder: To share any concern you have
Please use Feedback form on course website:
http://www.cs.rochester.edu/courses/261/spring2018/
-> Forms -> Feedback Form

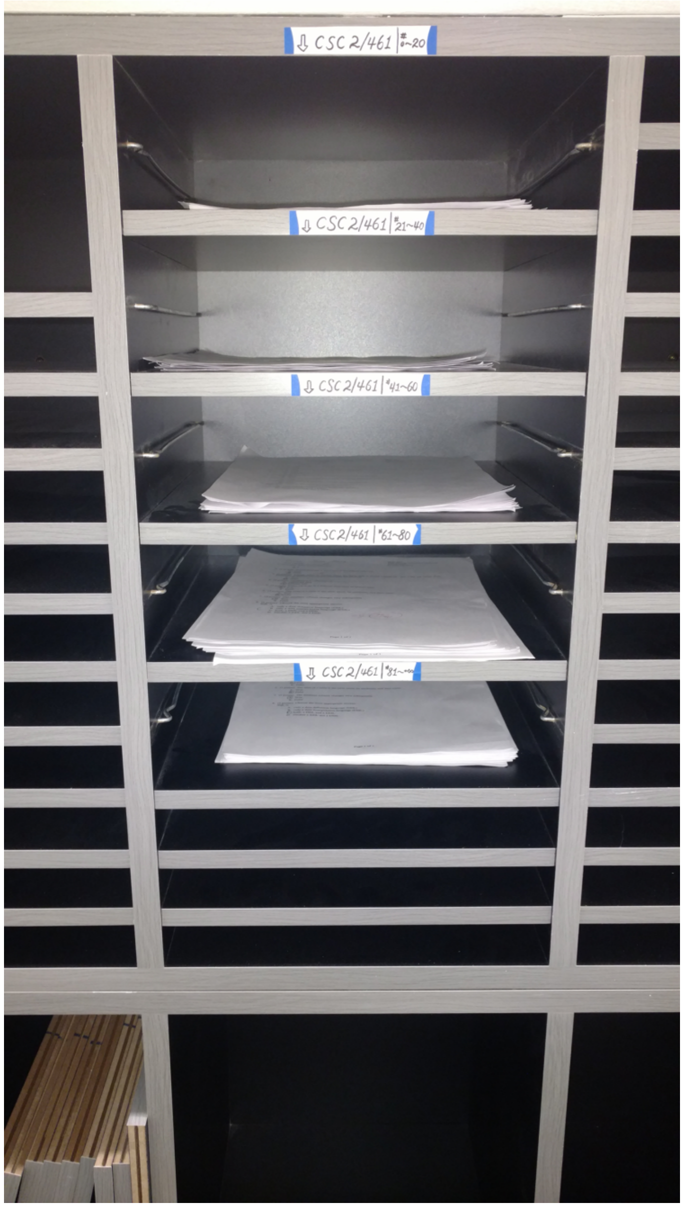
I personally will look into each of your concerns and try my best to resolve any issue

Quiz Collection

From where can I collect my quizzes?

Mailbox next to my office

Quiz Collection



Are they sorted

Yes.
Sorted by ClassID

The ER (Entity-Relationship) Model

Agenda

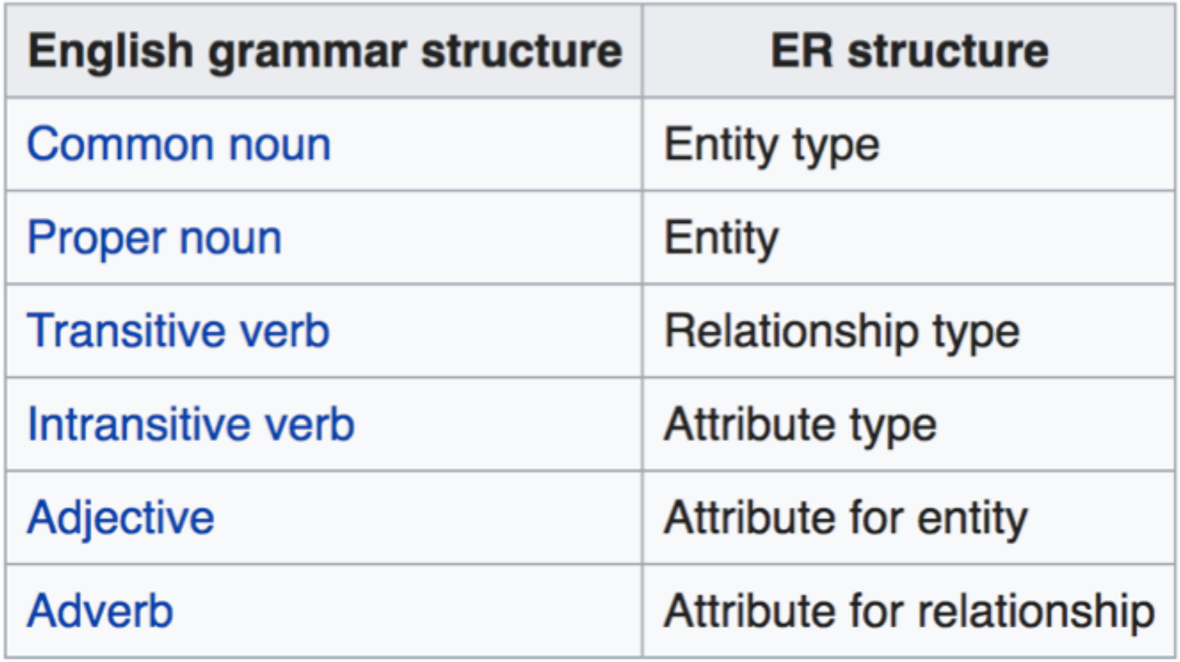
1. High-level motivation for the E/R model

2. Entities

3. Relations

4. E/R Model

Mapping natural language

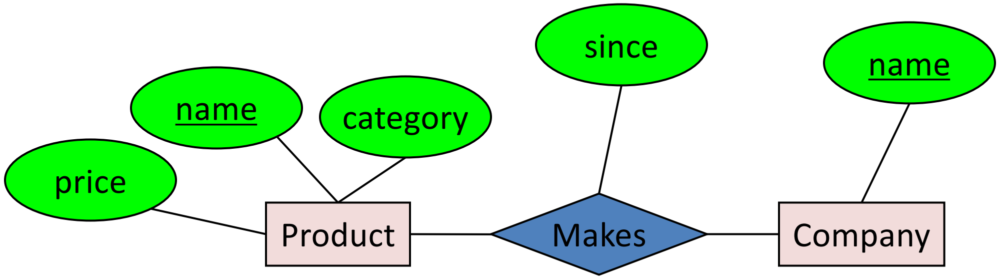
- Chen proposed the following "rules of thumb" for mapping natural language descriptions into ER diagrams: "English, Chinese and ER diagrams" by Peter Chen.
- 

English grammar structure	ER structure
Common noun	Entity type
Proper noun	Entity
Transitive verb	Relationship type
Intransitive verb	Attribute type
Adjective	Attribute for entity
Adverb	Attribute for relationship

Source: https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model

Relationships and Attributes

- Relationships may have attributes as well.

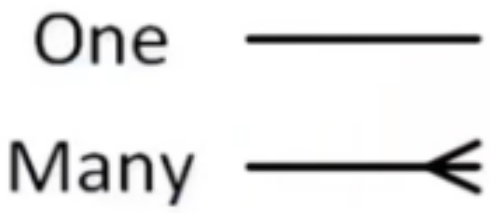


For example: "since" records when company started making a product

Note: "*since*" is implicitly unique per pair here! Why?

TYPES OF RELATIONSHIP

Conceptual Crow's Foot Relationship Symbols



One

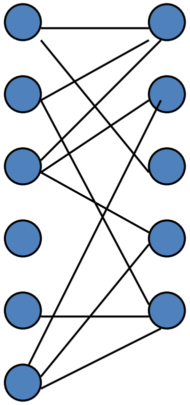
Many

Many-Many Relationships

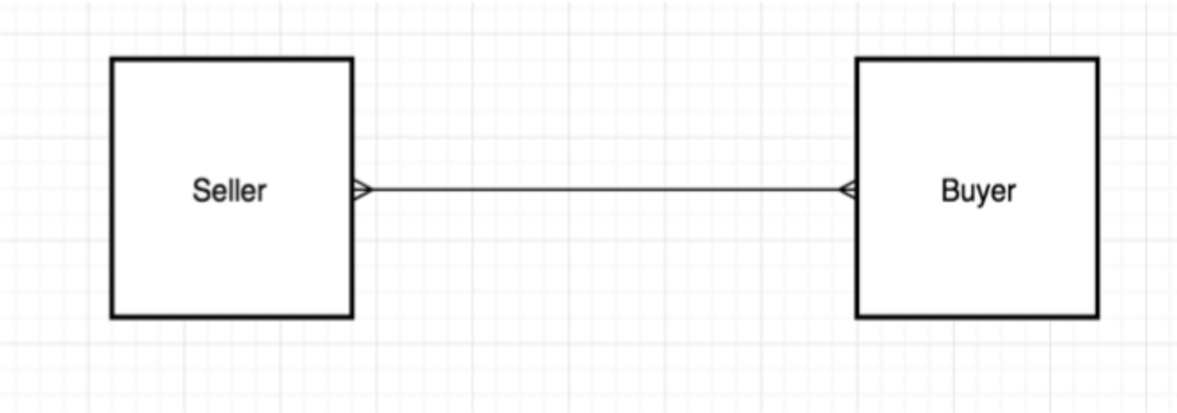
- Focus: binary relationships, such as Sells between Seller and Buyer.

- In a *many-many* relationship, an entity of either set can be connected to many entities of the other set.
  - E.g., a seller sells many items; a buyer can buy many items too.

In Pictures:



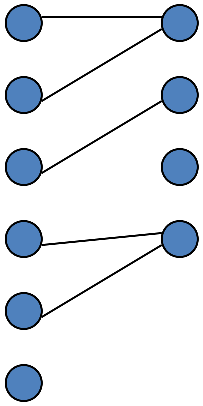
many-many



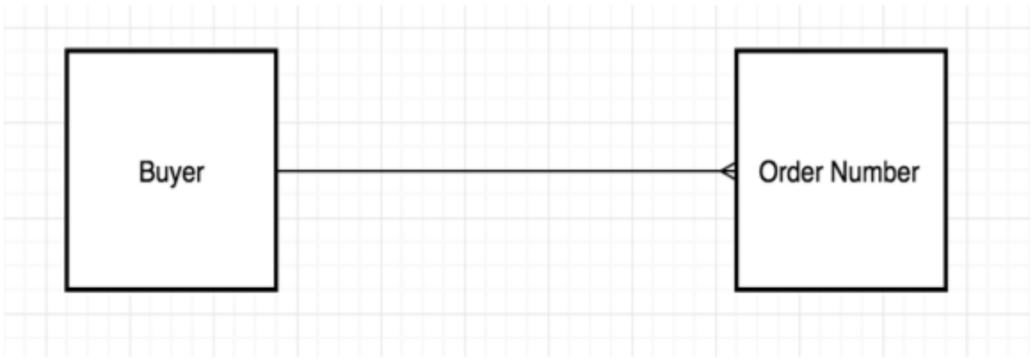
Many-One Relationships

- Some binary relationships are *many -one* from one entity set to another.
- Each entity of the first set is connected to at most one entity of the second set.
- But an entity of the second set can be connected to zero, one, or many entities of the first set.
- E.g.: One buyer can have multiple order number, but one order can be bought by only one buyer.

In Pictures:



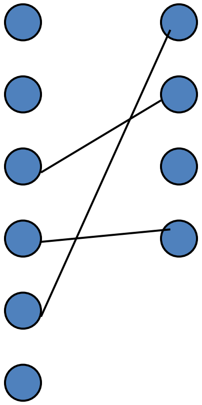
many-one

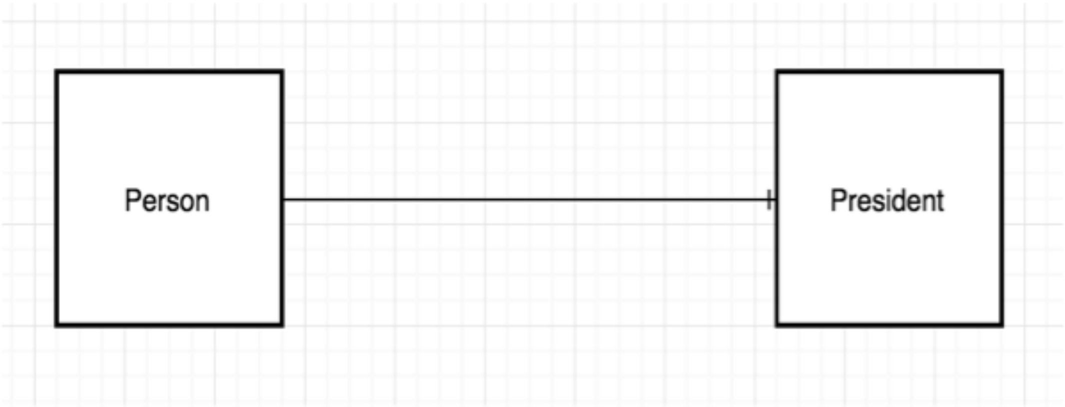


One-One Relationships

- In a *one-one* relationship, each entity of either entity set is related to at most one entity of the other set.
- Example: Relationship president between entity country and person.
  - A person can be the president of only one country. One country can have only one president.

In Pictures





Maximum Cardinality

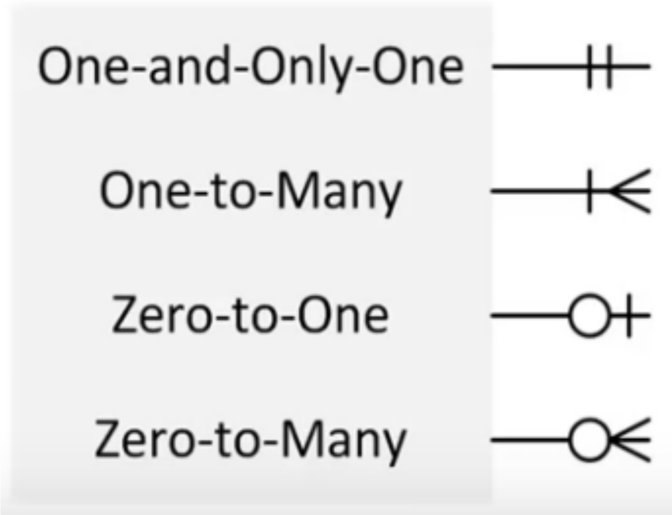
- Relationships are named and classified by their cardinalities, which is a word that means *count* (as in the number of items in a set)
- Each of the three types of binary relationship shown previously has a different *maximum cardinality*
- Maximum cardinality is the maximum number of entity instances that can participate in a relationship instance
  - One, many, or some other positive fixed number

Minimum Cardinality

- Minimum cardinality is the minimum number of entity instances that must participate in a relationship instance

- These values typically assume a value of zero (optional) or one (mandatory)

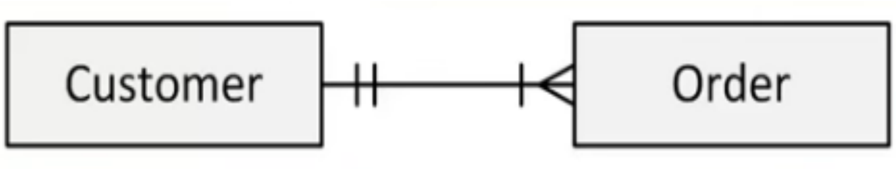
Crow’s Foot Symbols with Cardinalities



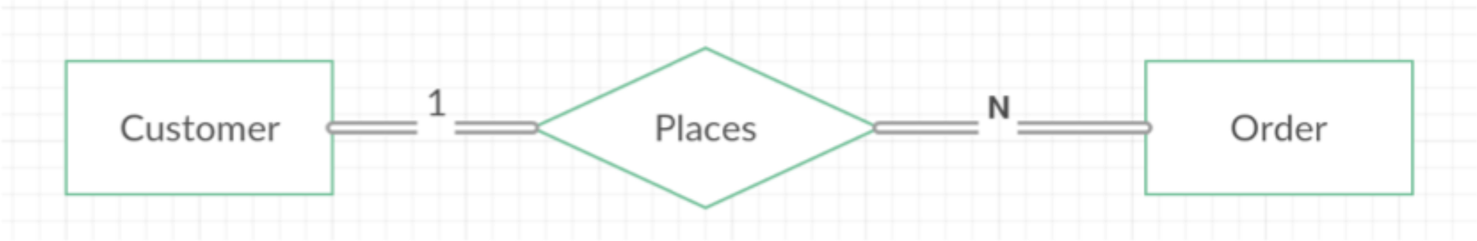
Cardinality Example

- Maximum cardinality is many for Order and one for Customer
- Minimum cardinality is one for both Customer and Order
  - Each customer can place one or more orders
  - Each order is associated with one and only one customer

Crow's foot Notation



Chen Notation



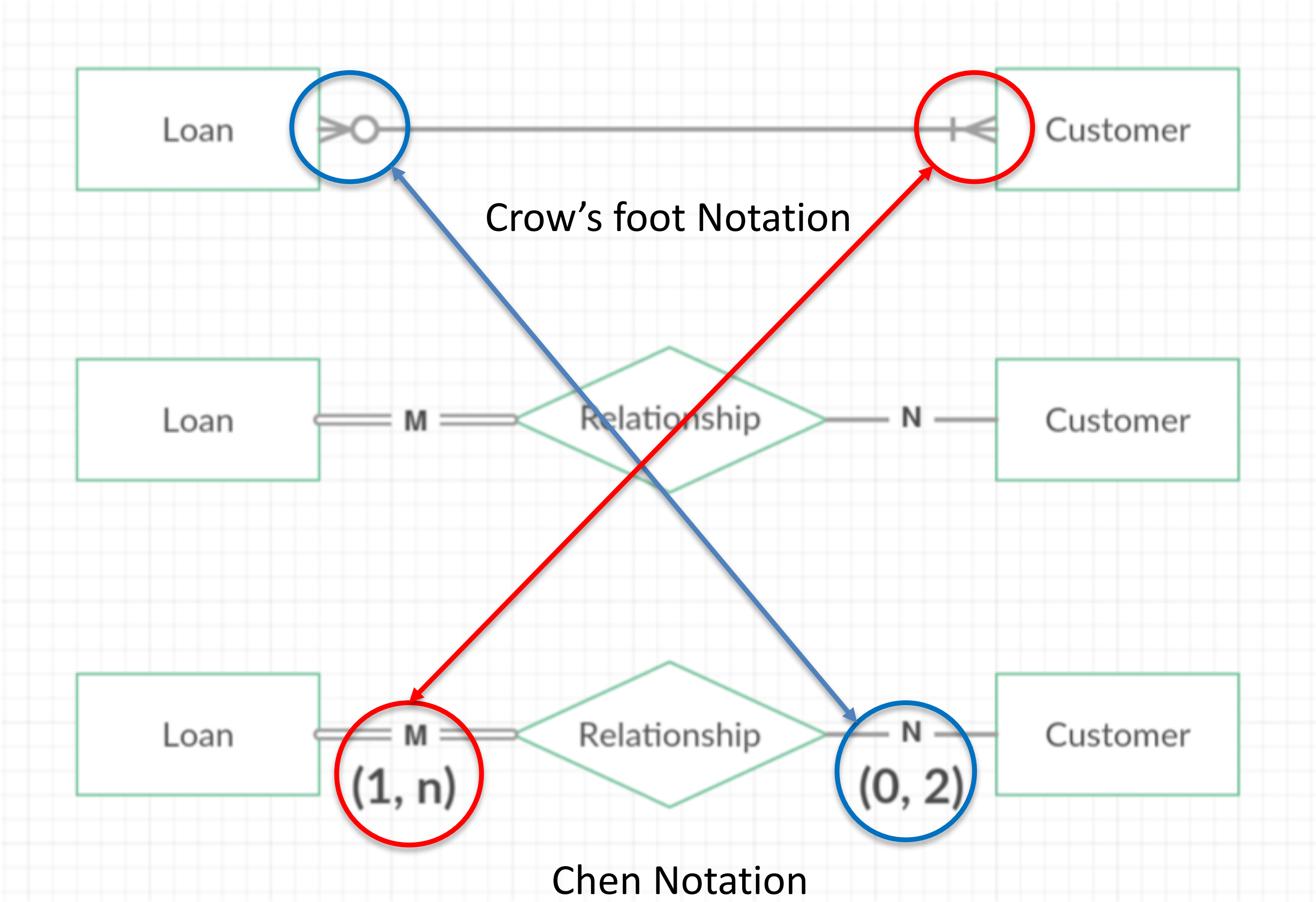
Entity-Relationship Diagrams

- The diagrams in previous slides are called entity-relationship diagrams
  - Entities represented by rectangles
  - Relationships represented by lines
  - Cardinalities represented by Crow’s Foot symbols

HAS-A Relationships

- The relationships in the previous slides are called HAS-A relationships
- The term is used because each entity instance *has a* relationship to a second entity instance
  - An employee has a locker
  - A locker has an employee
- There are also IS-A relationships

Different Representations



Crow’s foot Notation

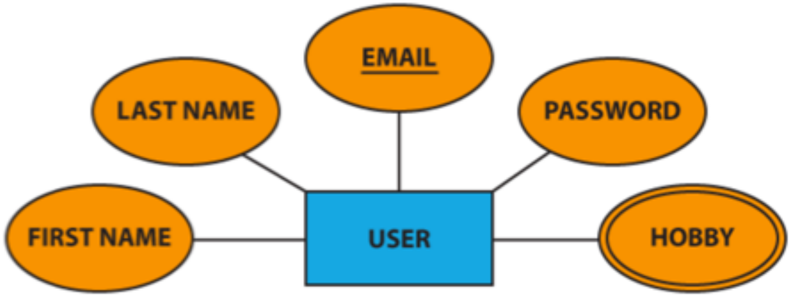
Chen Notation

(min,max) constraint

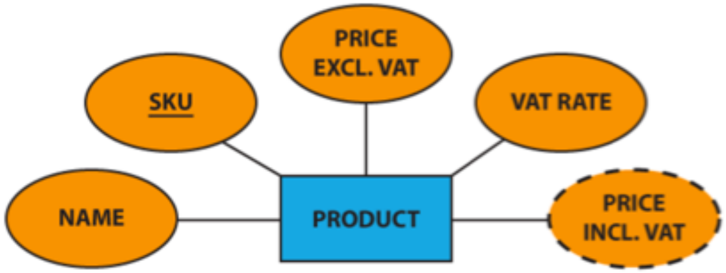
Agenda

- More about ER model
- ER model to Relation (Table)

Special Symbols Used



Multivalued Attribute



Derived Attribute

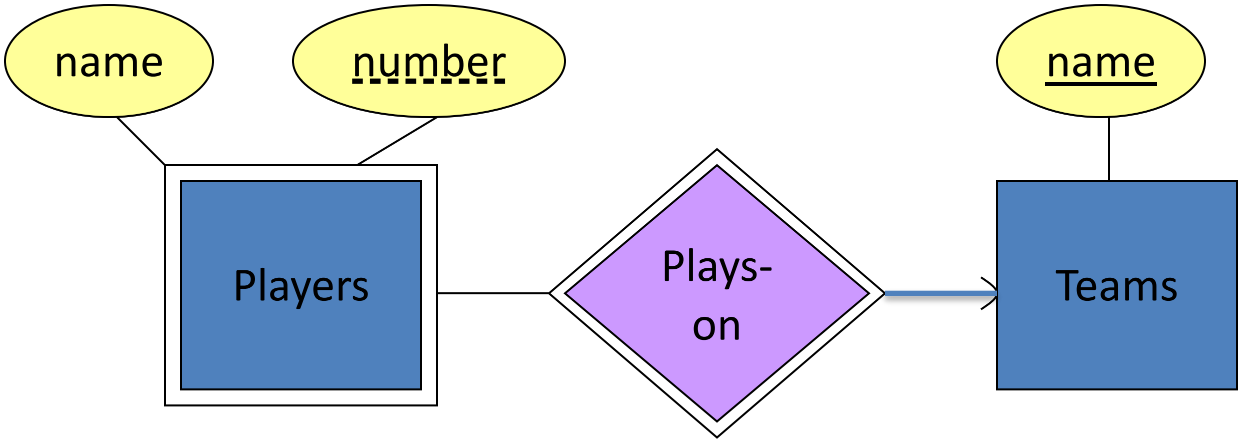
Strong and Weak Entities

- A weak entity is an entity whose instances **cannot** exist in the database without the existence of an instance of another entity

- Any entity that is not weak entity is called a strong entity
  - Instances of a strong entity can exist in the database independently

- The weak entity's identifier is a combination of the identifier of the owner entity and the partial key of the weak entity.

In E/R Diagrams



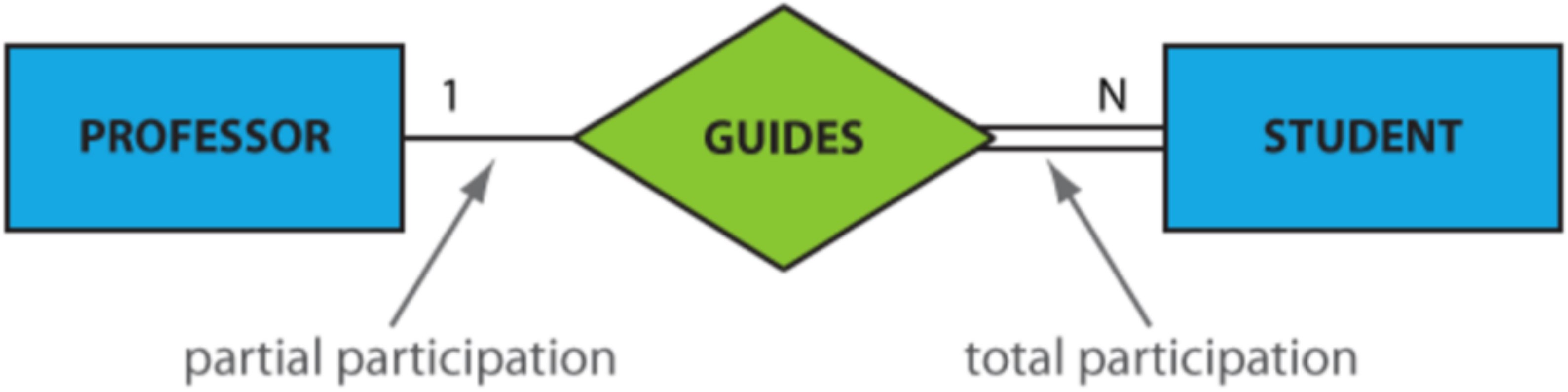
- Double diamond for *supporting* many-one relationship with Weak Entity.
- Double rectangle for the weak entity set.

Example: Weak Entity Set

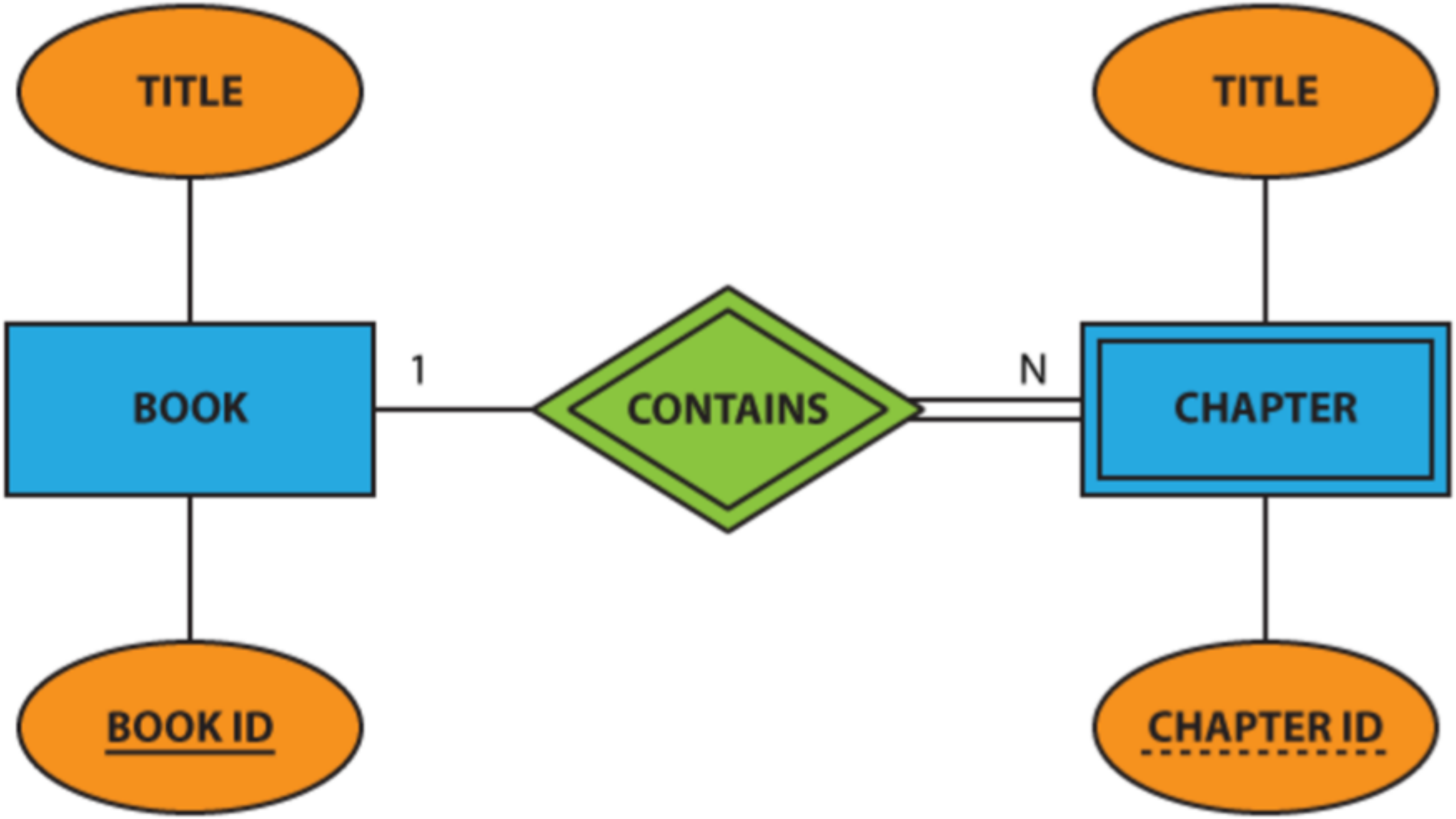
- name is almost a key for football players, but there might be two with the same name.
- number is certainly not a key, since players on two teams could have the same number.
- But number, together with the team name related to the player should be unique.

Partial vs Total Participation

- An entity set may participate in a relation either totally or partially.
- **Total participation** means that every entity in the set is involved in the relationship
  - depicted as a **double line.**
- **Partial participation** means that not all entities in the set are involved in the relationship, e.g., not every professor guides a student
  - depicted by a **single line**.



Weak Entity , Total Participation and Partial Key

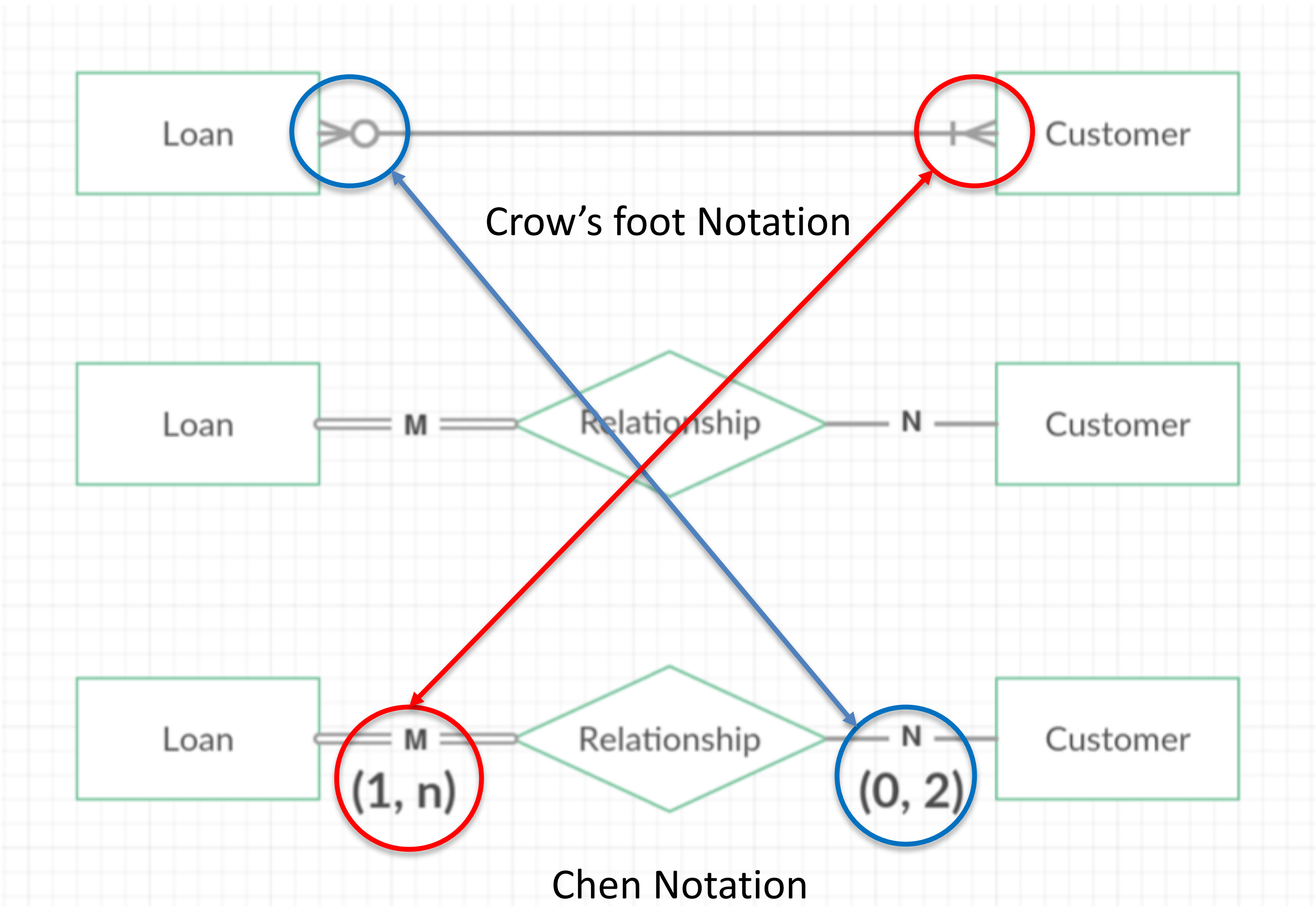


(min, max) constraint

- Min = 0 implies partial participation

- Min > 0 implies total participation

Different Representations



Crow's foot Notation

Chen Notation

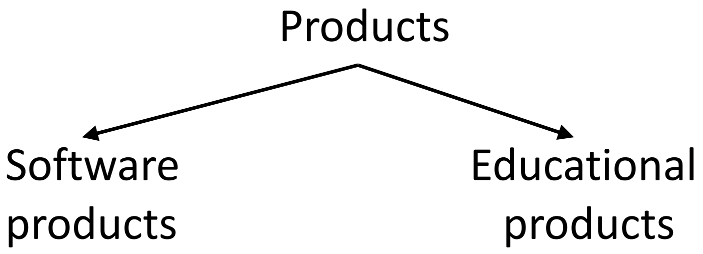
(min,max) constraint

HAS-A Relationships

- The relationships in the previous slides are called HAS-A relationships
- The term is used because each entity instance *has a* relationship to a second entity instance
  - An employee has a locker
  - A locker has an employee
- There are also IS-A relationships

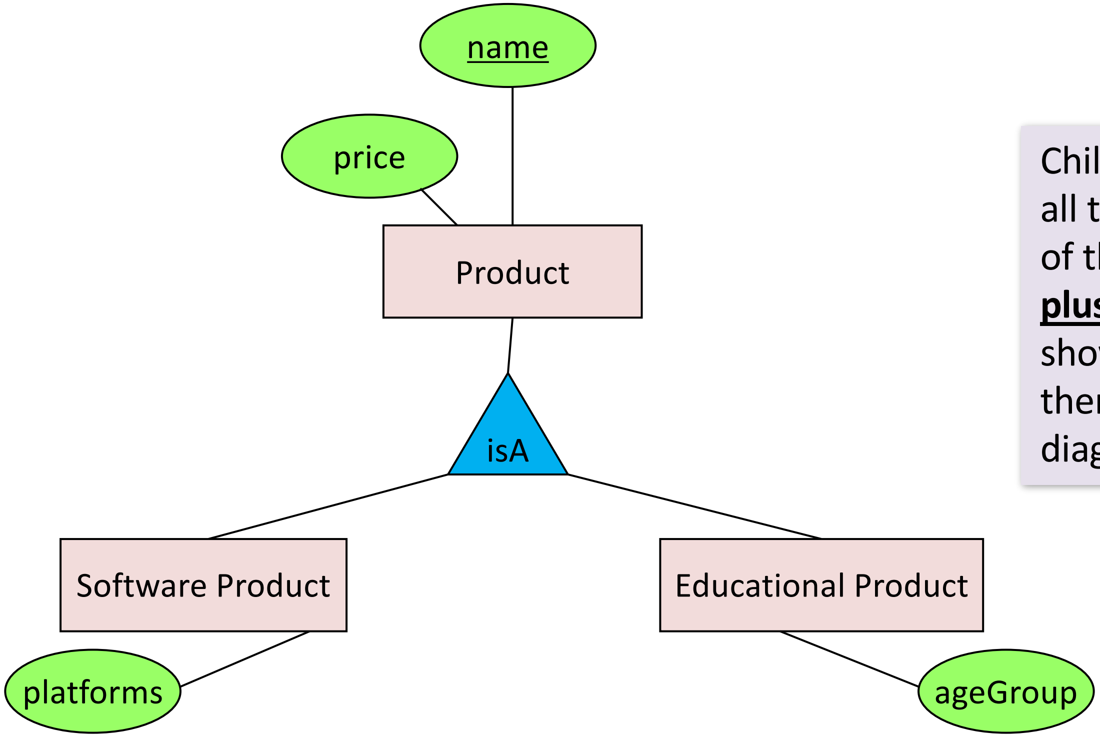
Modeling Subclasses

- Some objects in a class may be special, i.e. worthy of their own class
  - Define a new class?
    - *But what if we want to maintain connection to current class?*
  - Better: define a *subclass*
    - *Ex:*



We can define **subclasses** in E/R!

Modeling Subclasses



Child subclasses contain all the attributes of *all* of their parent classes **plus** the new attributes shown attached to them in the E/R diagram

Understanding Subclasses

- Think in terms of records; ex:
  - Product

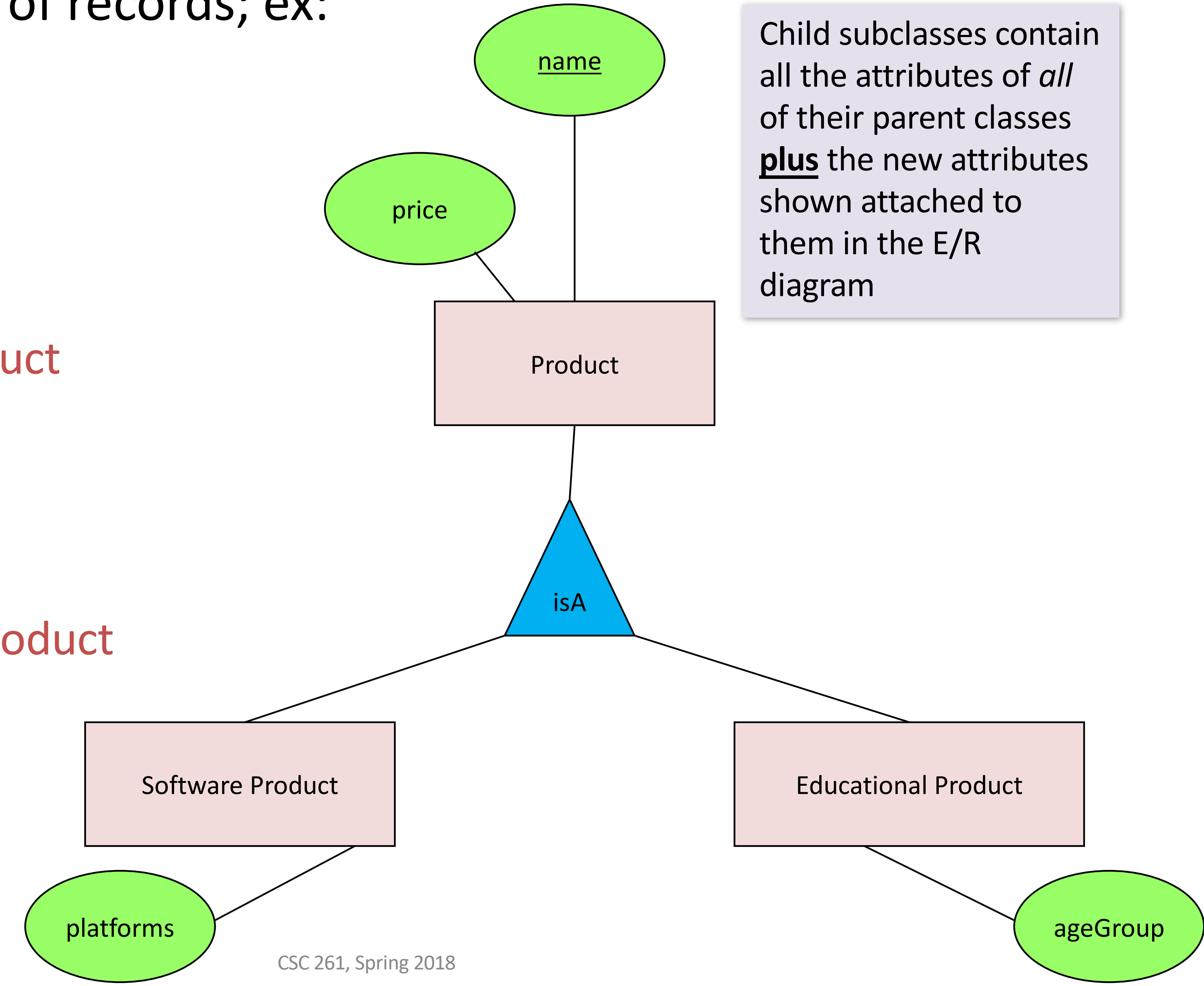
    name
    price

  - SoftwareProduct

    name
    price
    platforms

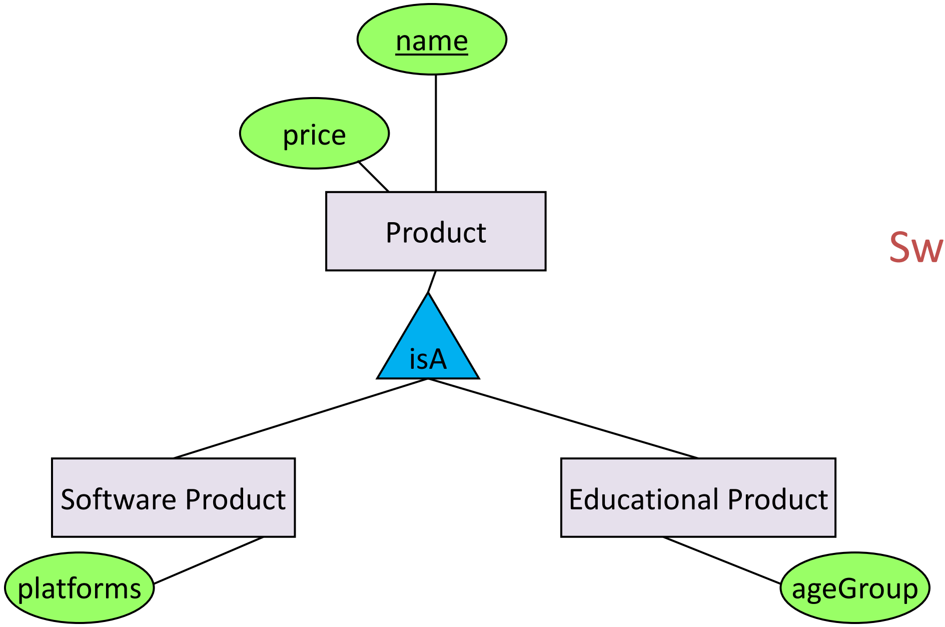
  - EducationalProduct

    name
    price
    ageGroup



Child subclasses contain all the attributes of *all* of their parent classes **plus** the new attributes shown attached to them in the E/R diagram

Think like tables…



Product

name	price	category
Gizmo	99	gadget
Camera	49	photo
Toy	39	gadget

Sw.Product

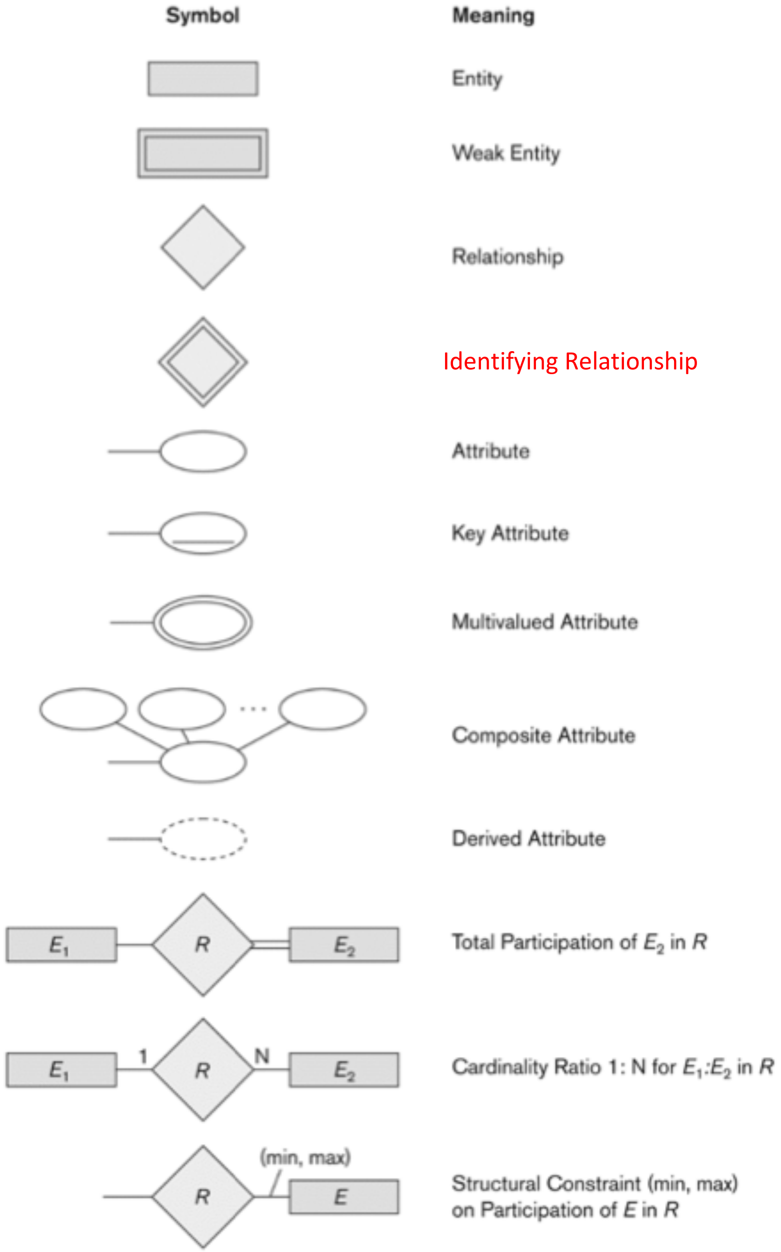
name	platforms
Gizmo	unix

Ed.Product

name	ageGroup
Gizmo	todler
Toy	retired

- Summary

Figure 3.14
Summary of the notation for ER diagrams.



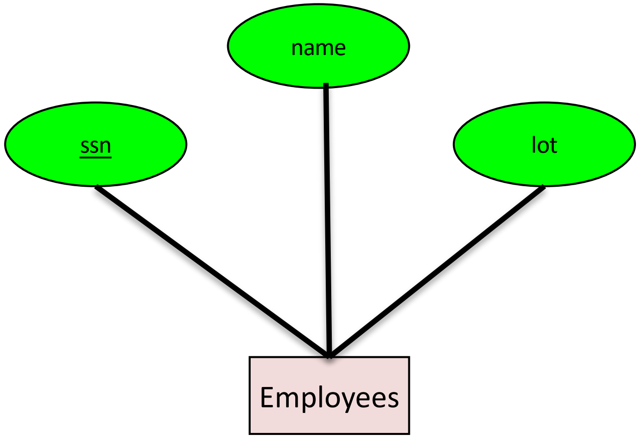
Symbol	Meaning
	Entity
	Weak Entity
	Relationship
	Identifying Relationship
	Attribute
	Key Attribute
	Multivalued Attribute
	Composite Attribute
	Derived Attribute
	Total Participation of E_2 in R
	Cardinality Ratio 1: N for E_1:E_2 in R
	Structural Constraint (min, max) on Participation of E in R

Design Theory (ER model to Relations)

ER Models to Relations

Please go through Chapter 9

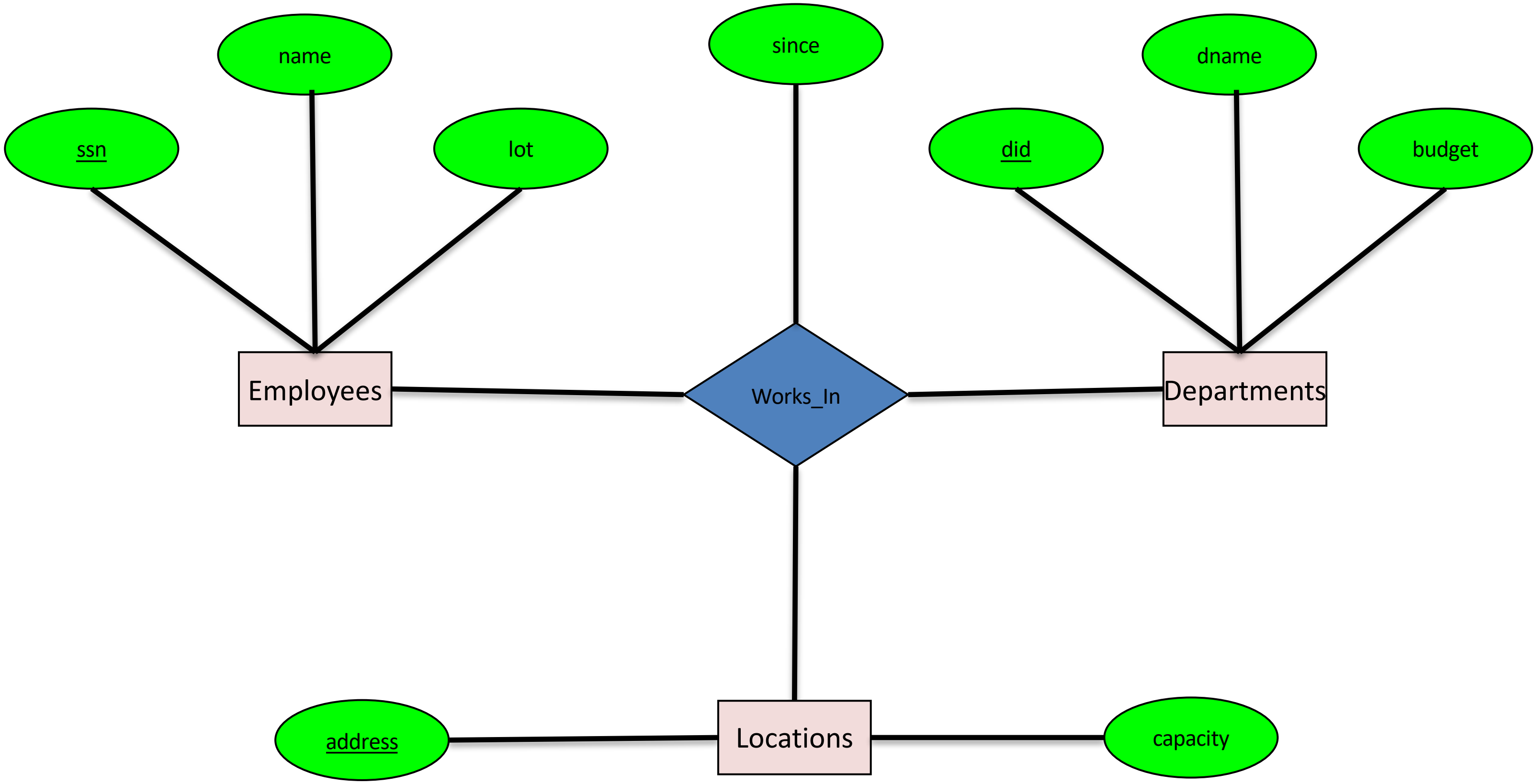
Entity Sets to Tables



ssn	name	lot
123-22-3333	Alex	23
234-44-6666	Bob	44
567-88-9787	John	12

```
CREATE TABLE Employees (   ssn char(11),
                           name varchar(30),
                           lot Integer,
                           PRIMARY KEY (ssn))
```

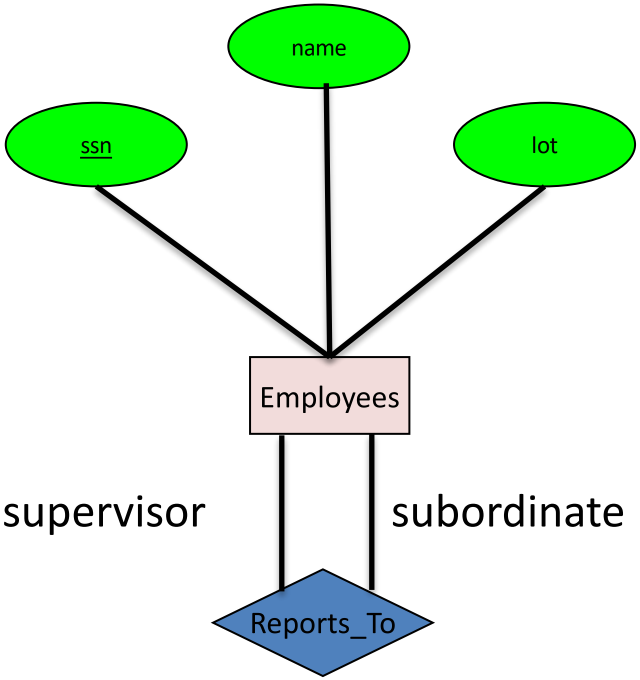
Relationship Sets (without Constraints) to Tables



Relationship Sets (without Constraints) to Tables

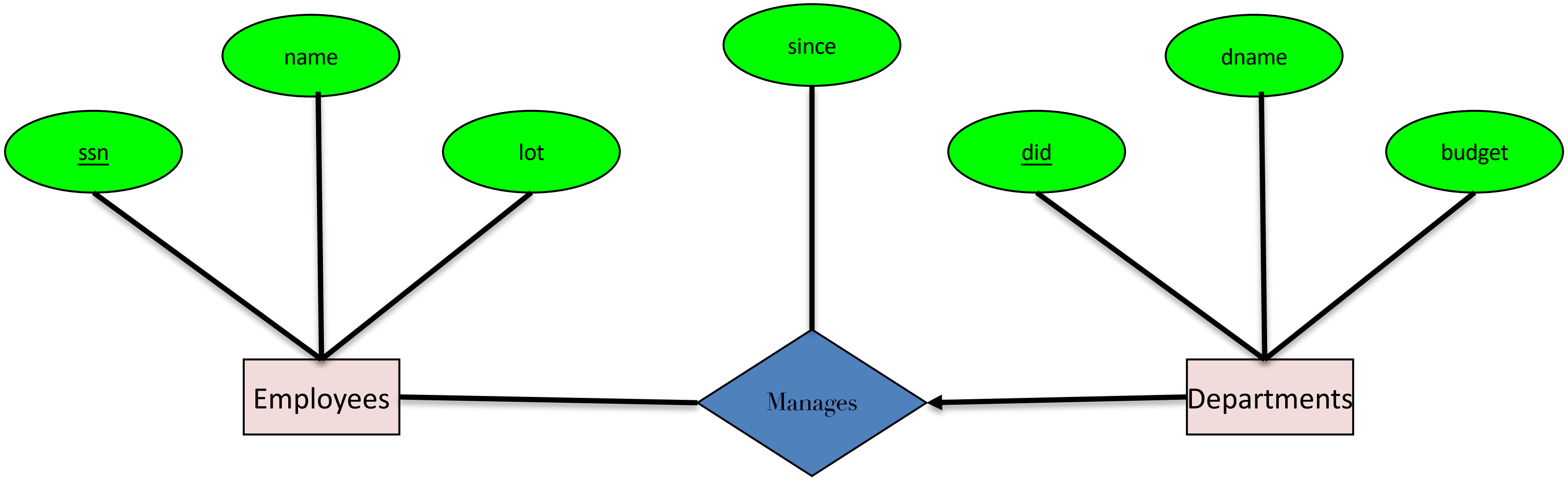
```
CREATE TABLE Works_in(     ssn char(11),
                           did integer (30),
address varchar(30),
since date,
PRIMARY KEY (ssn, did, address),
FOREIGN KEY (ssn) REFERENCES Employees,
FOREIGN KEY (address) REFERENCES Locations,
FOREIGN KEY (did) REFERENCES Departments,
)
```

Relationship Sets (without Constraints) to Tables



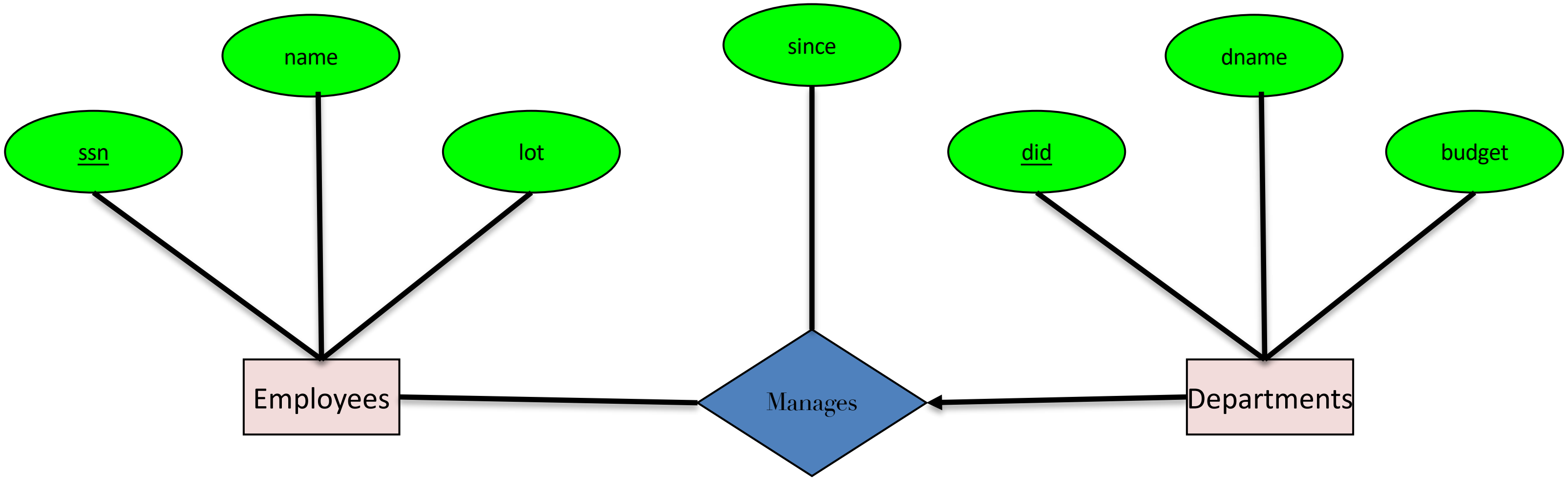
```
CREATE TABLE Reports_To(
             supervisor_ssn char(11),
             subordinate_ssn char(11),
             PRIMARY KEY (supervisor_ssn,
                     subordinate_ssn),
FOREIGN KEY (supervisor_ssn )
       REFERENCES Employees(ssn),
FOREIGN KEY (subordinate_ssn )
       REFERENCES Employees(ssn)
)
```

Relationship Sets (with key Constraints) to Tables



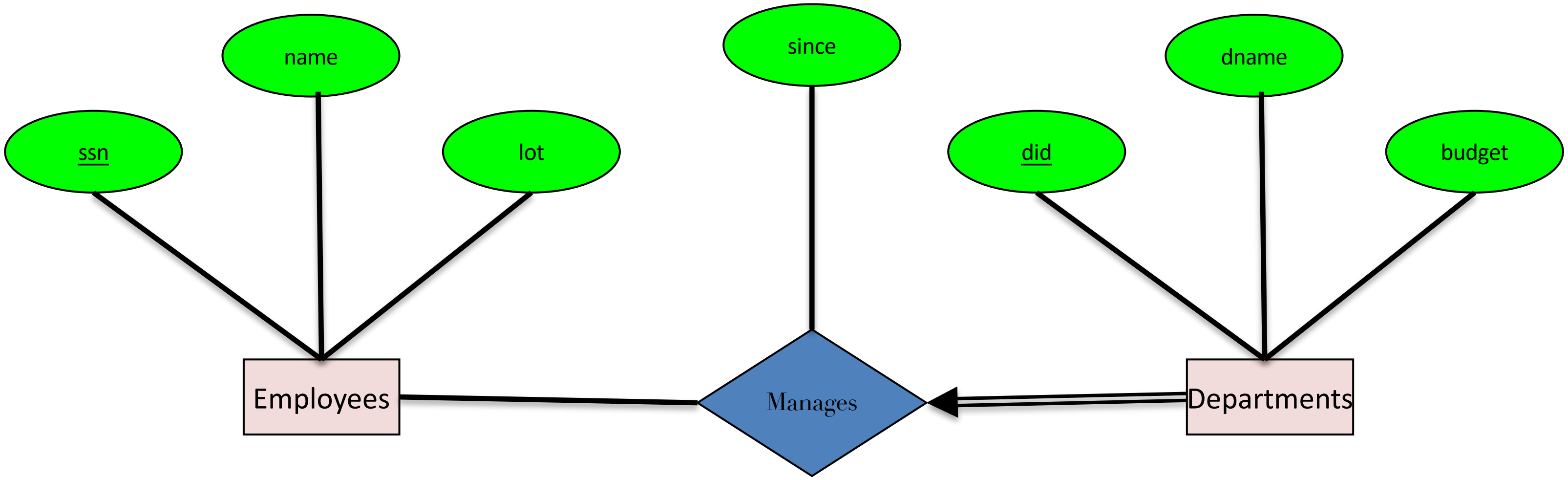
```
CREATE TABLE Manages(      ssn char(11),
                           did integer (30),
                           since date,
PRIMARY KEY (did),
FOREIGN KEY (ssn) REFERENCES Employees,
FOREIGN KEY (did) REFERENCES Departments,
)
```

Better way of doing it



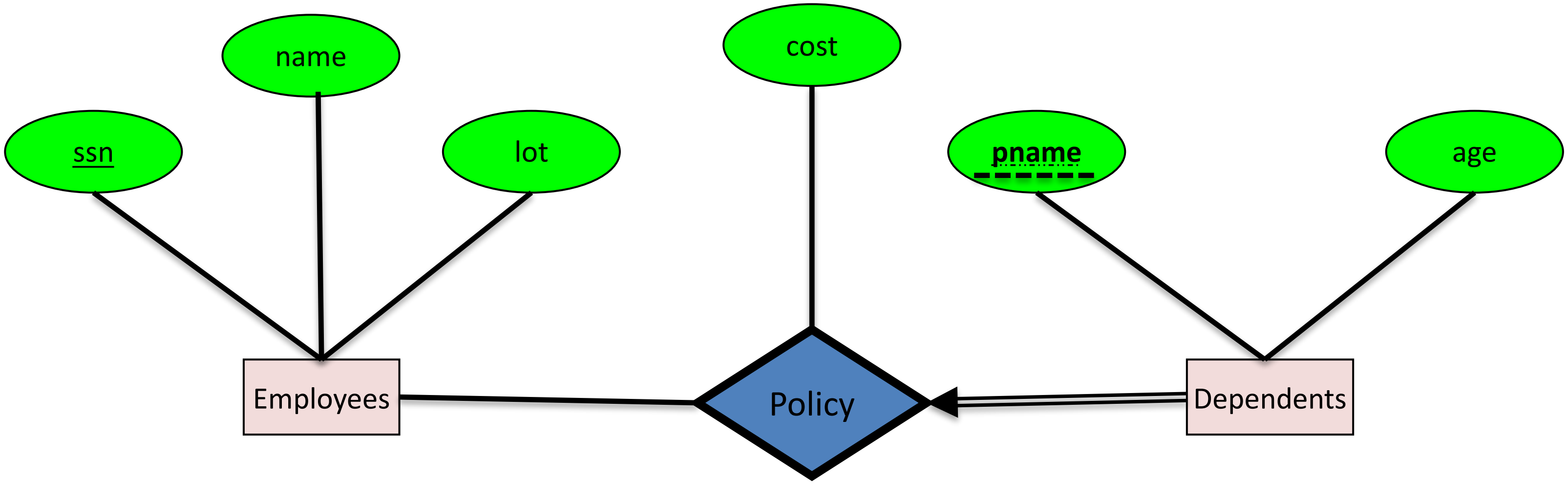
```
CREATE TABLE Dept_Mgr(     did integer (30),
                           dname varchar(30),
                           budget float(30),
                           ssn char(11),
                           since date,
PRIMARY KEY (did),
FOREIGN KEY (ssn) REFERENCES Employees,
)
```

Relationship Sets (with Participation Constraints) to Tables



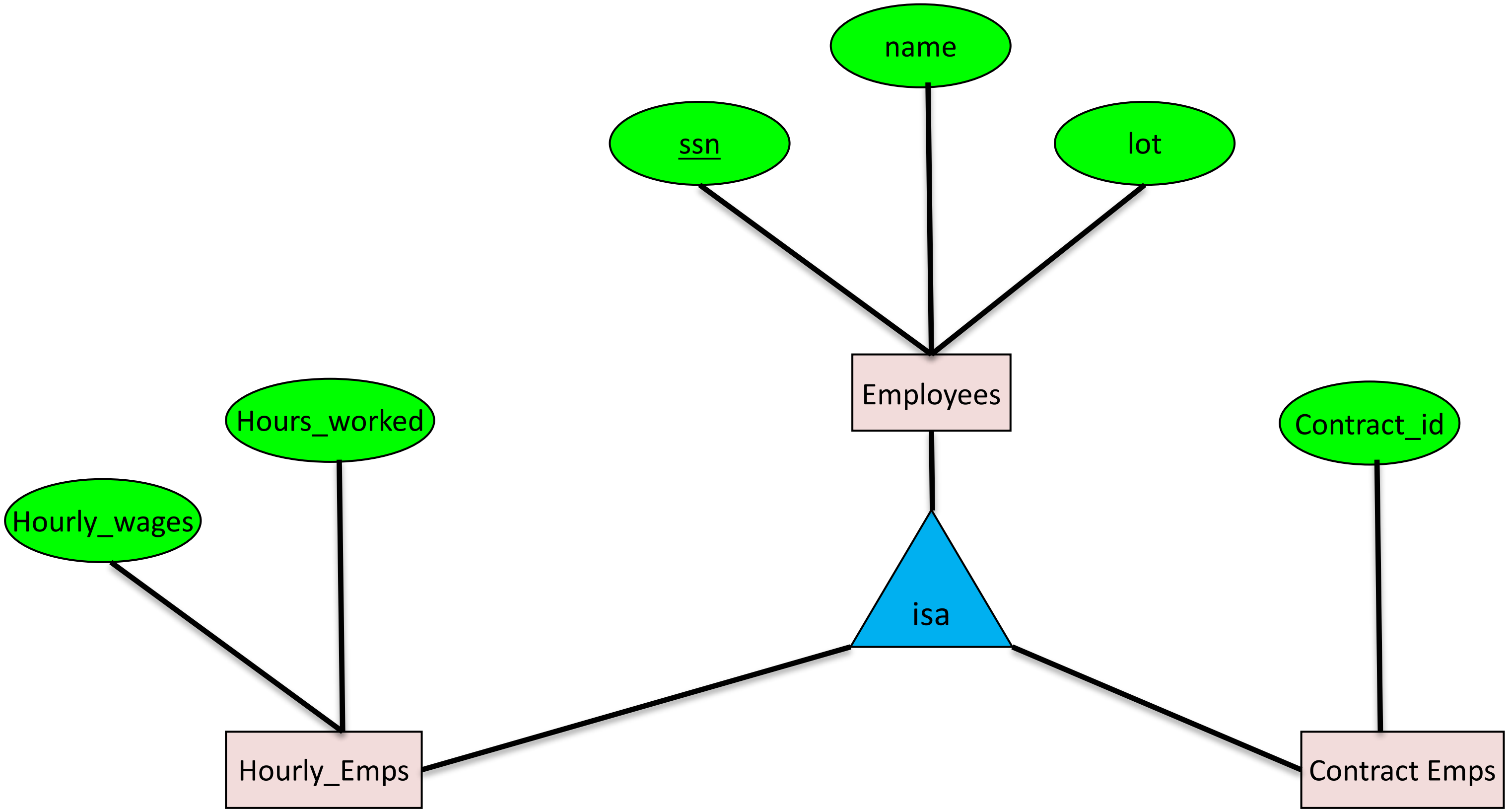
```
CREATE TABLE Dept_Mgr(       did integer (30),
                             dname varchar(30),
                             budget float(30),
                             ssn char(11),
                             since date,
PRIMARY KEY (did),
FOREIGN KEY (ssn) REFERENCES Employees ON
DELETE NO ACTION
)
```

Translating Weak Entity Sets



```
CREATE TABLE Dept_Policy(  pname varchar(30),
                           age integer,
                           cost float,
                           ssn char(11),
      PRIMARY KEY (pname, ssn),
      FOREIGN KEY (ssn) REFERENCES Employees
ON DELETE CASCADE
)
```

Translating Class Hierarchies



Two options

1. We can map each of the entity sets Employees, Hourly_Emps, and Contract_Emps to a distinct relation.
2. We can create just two relations, corresponding to Hourly_Emps and Contract_Emps

Both have their pros and cons

- Redundancy
- Performance

E/R Summary

- E/R diagrams are a visual syntax that allows technical and non-technical people to talk
  - For conceptual design

- Basic constructs: **entity**, **relationship**, and **attributes**

- A good design is faithful to the constraints of the application, but not overzealous

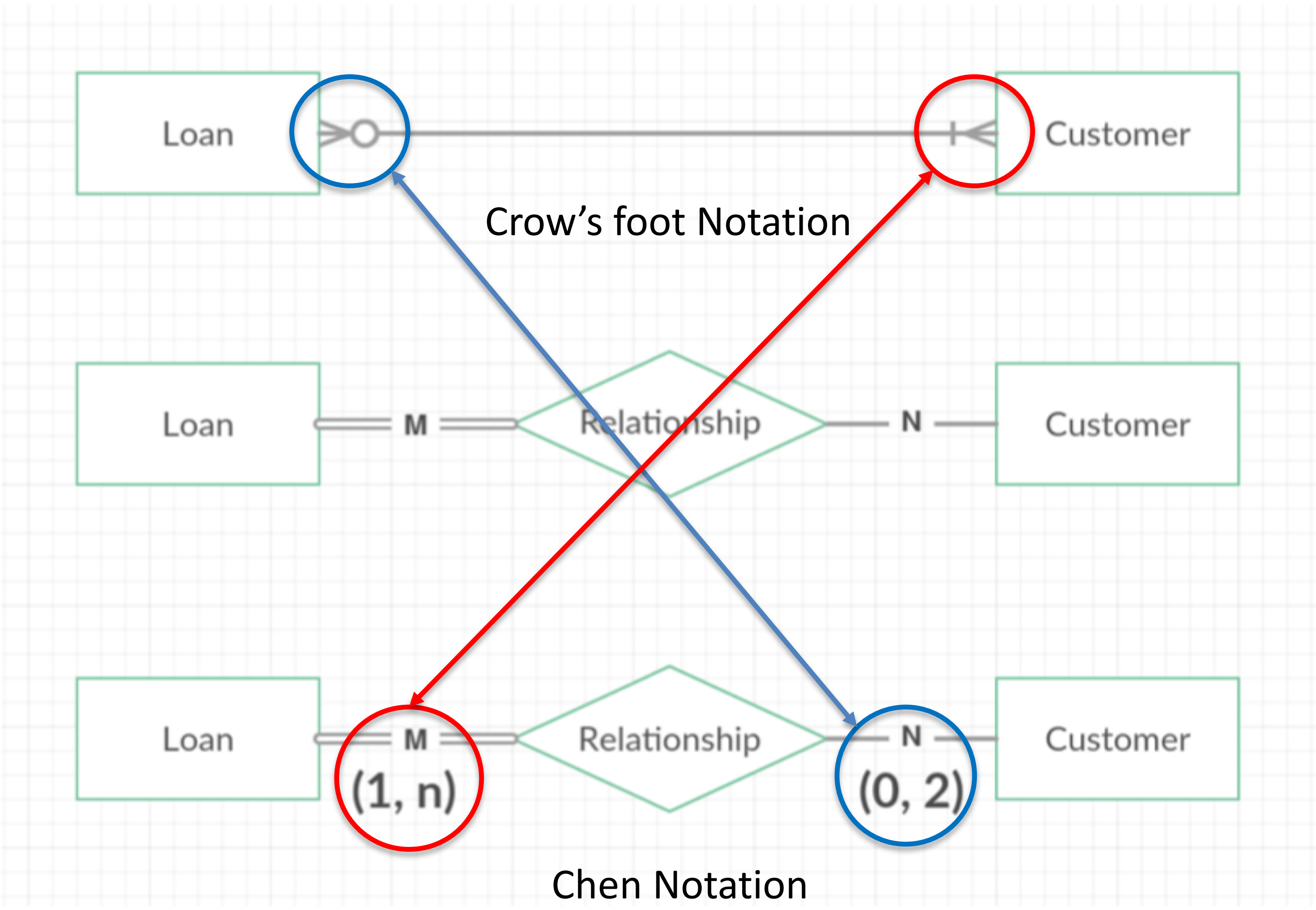
Scenario

- One customer can have at max 2 loans. One loan can be given to multiple customers.

What it really means:

- One customer can have (0,2) loans
- One loan can be given to (1,n) customer
- This is a many to many scenario

Different Representations



Crow's foot Notation

Chen Notation

Acknowledgement

- Some of the slides in this presentation are taken from the slides provided by the authors.

- Many of these slides are taken from cs145 course offered by Stanford University.

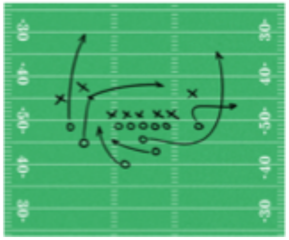
ACTIVITIES

DRAW AN E/R DIAGRAM FOR FOOTBALL

Use the following simplified model of a football season
(concepts to include are underlined):

Teams play each other in **Games**. Each pair of teams can play each other multiple times

Players belong to Teams



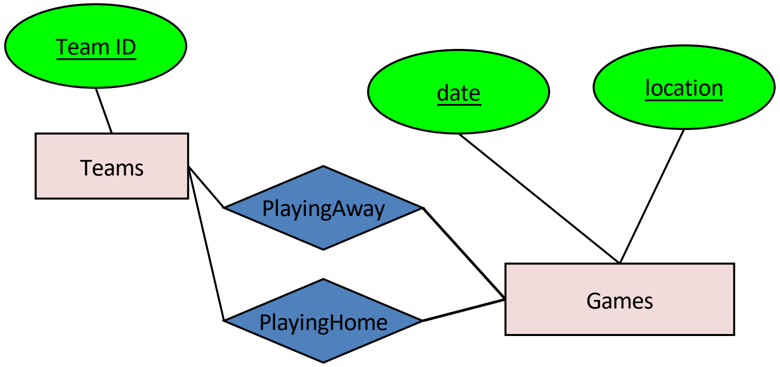
A Game is made up of **Plays** that result in a yardage gain/loss, and potentially a touchdown

A Play will contain either a **Pass** from one player to another, or a **Run** by one player

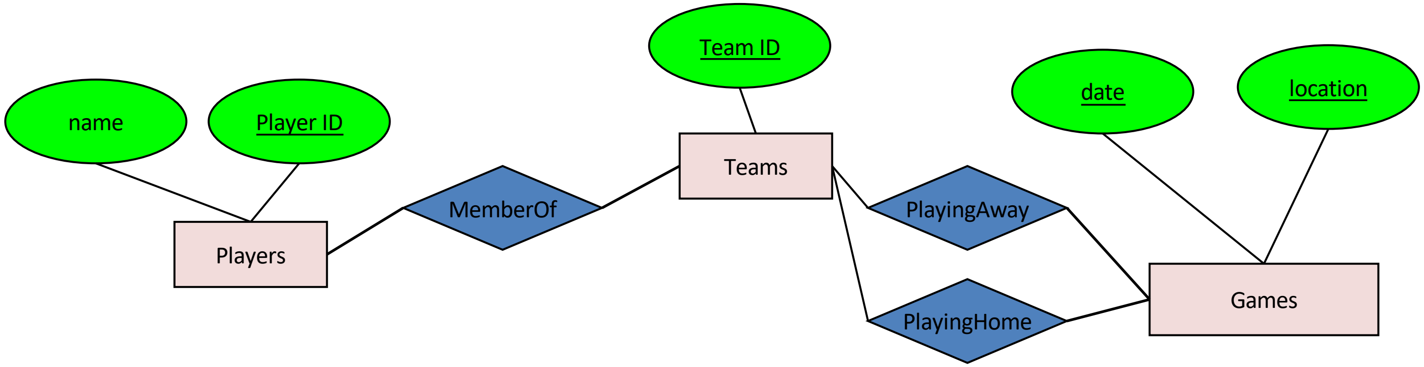
Note that various ER diagrams could work, not just the following one!

ACTIVITY

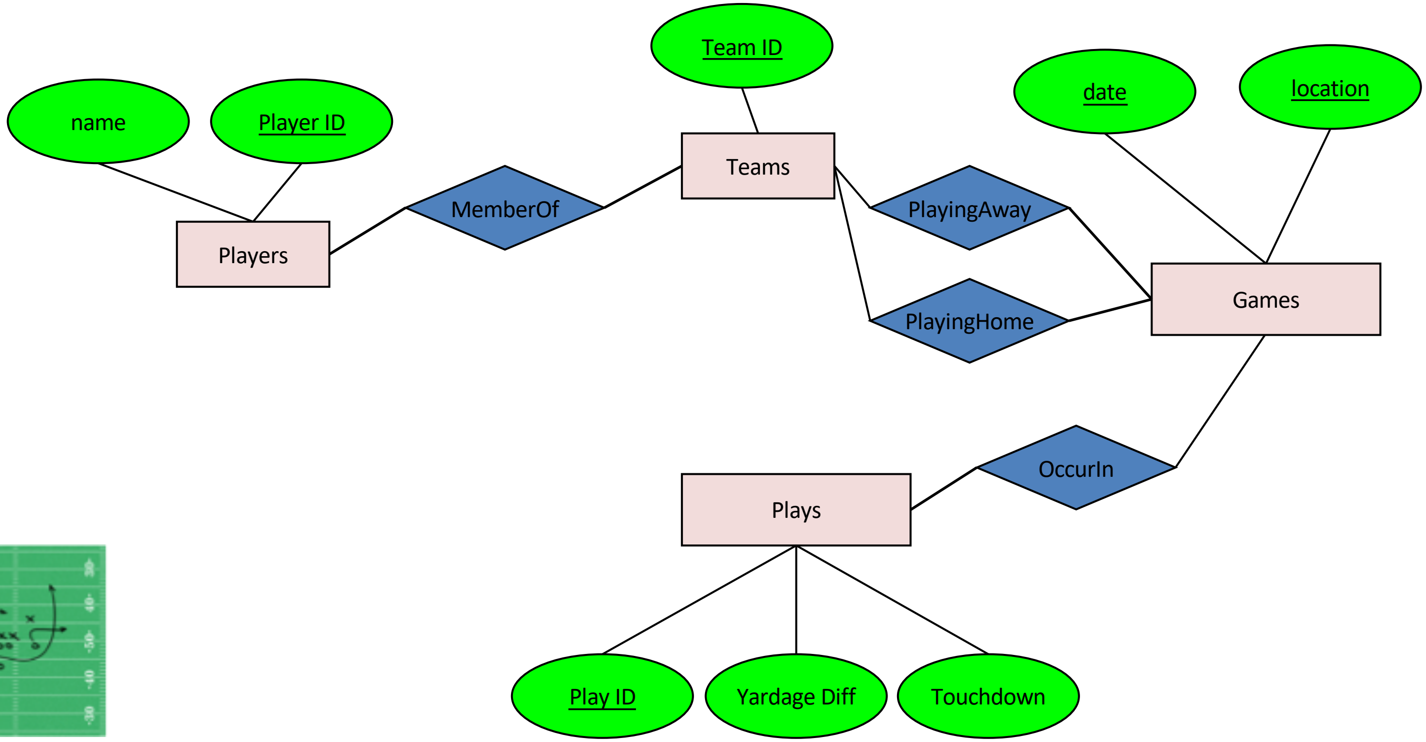
Note two copies of the Teams entity here!

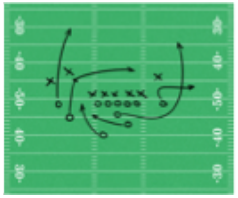


Teams play each other in **Games**. Each pair of teams can play each other multiple times

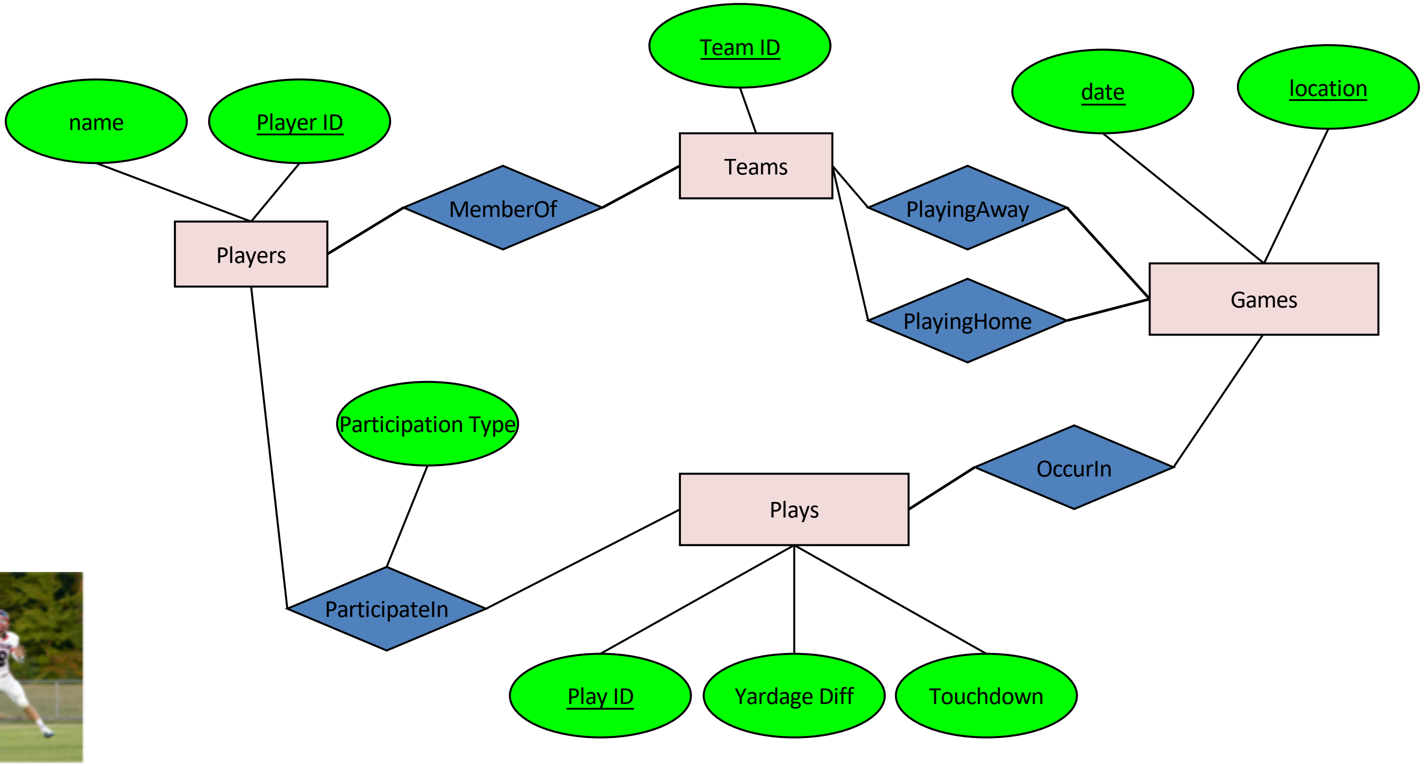


Players belong to Teams (assume no trades / changes)





A Game is made up of **Plays** that result in a yardage gain/loss, and potentially a touchdown



A Play will contain either a **Pass** from one player to another, or a **Run** by one player

Note that various ER diagrams could work, not just the following one!

ACTIVITY 2

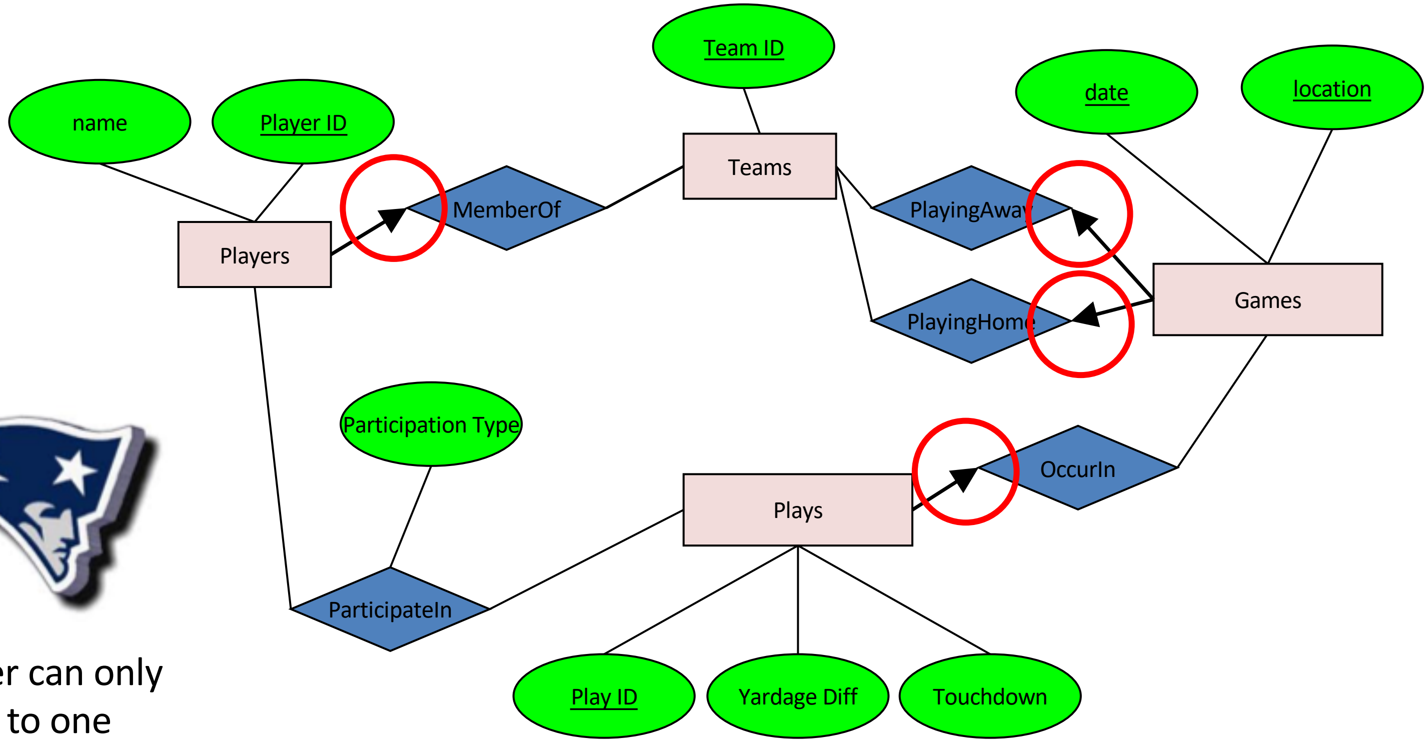
ENHANCE YOUR E/R DIAGRAM!

Also make sure to add **(new concepts underlined):**

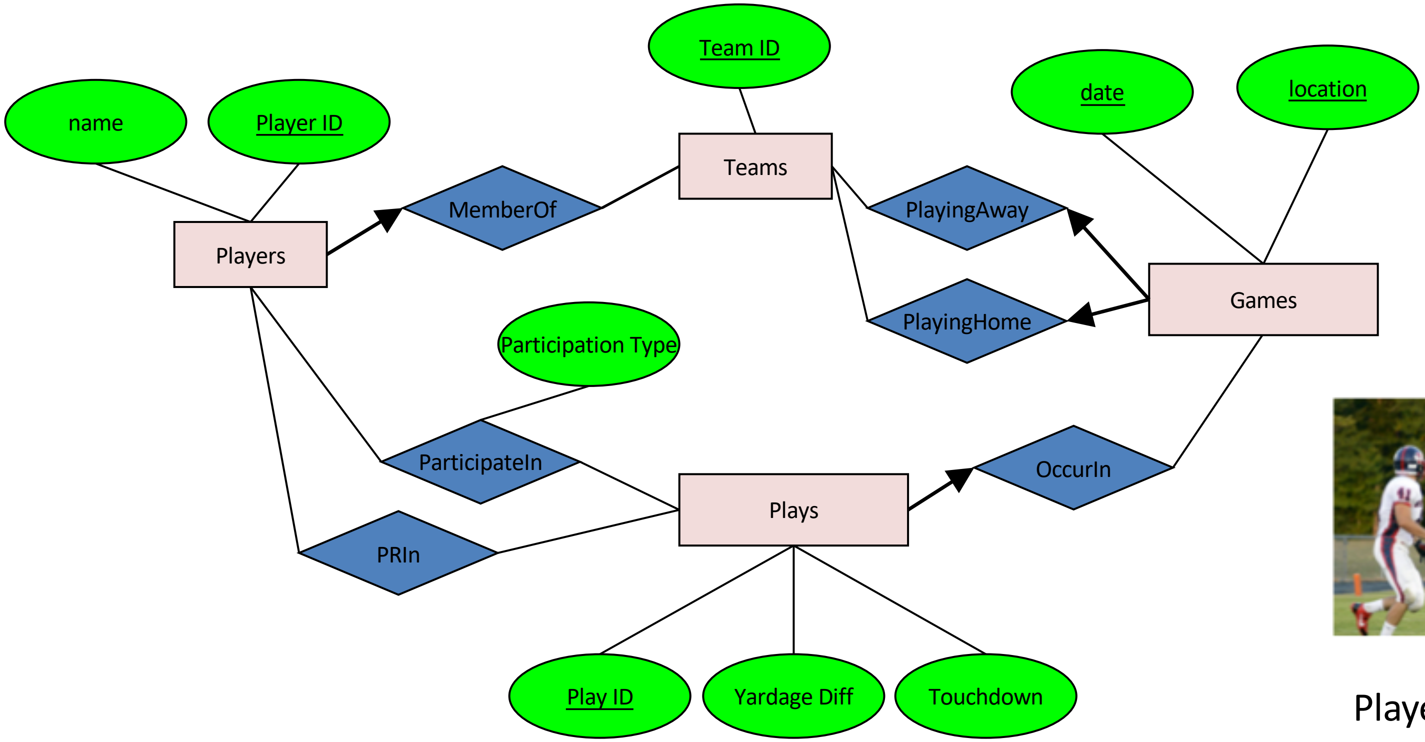
A player can only belong to one team, a play can only be in one game, a pass/run..?

Players can achieve a **Personal Record** linked to a specific Game and Play

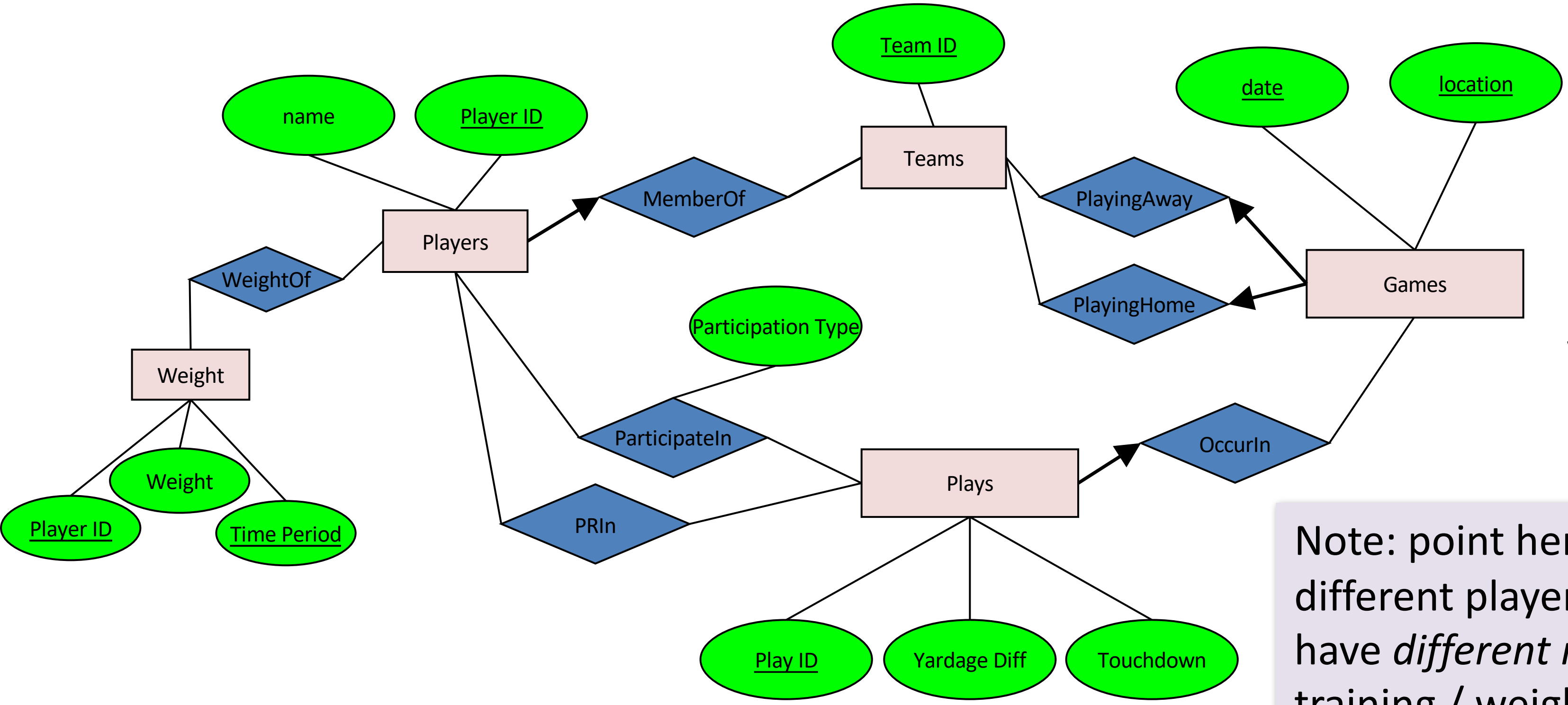
Players have a **weight** which changes in on vs. off-season



A player can only belong to one team, a play can only be in one game, a pass/run..?



Players can achieve a **Personal Record** linked to a specific Game and Play



Players might have different weights at different times

Note: point here is that different players might have *different numbers* of training / weight phases- hence should represent as new entity!

Note that various ER diagrams could work, not just the following one!

ACTIVITY 3

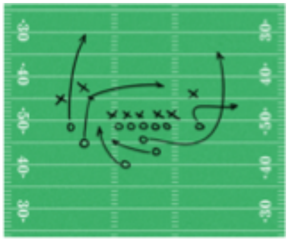
ADD IN: SUBCLASSES, CONSTRAINTS, AND WEAK ENTITY SETS

Concepts to include / model:

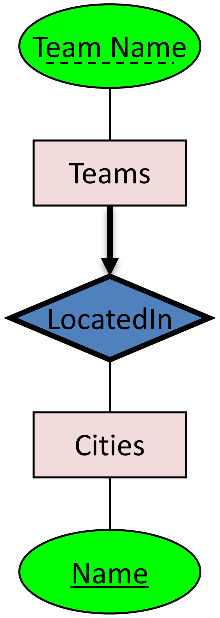
Teams belong to cities- model as ***weak entity sets***

Players are either on Offense or Defense, and are of types (QB, RB, WR, TE, K)

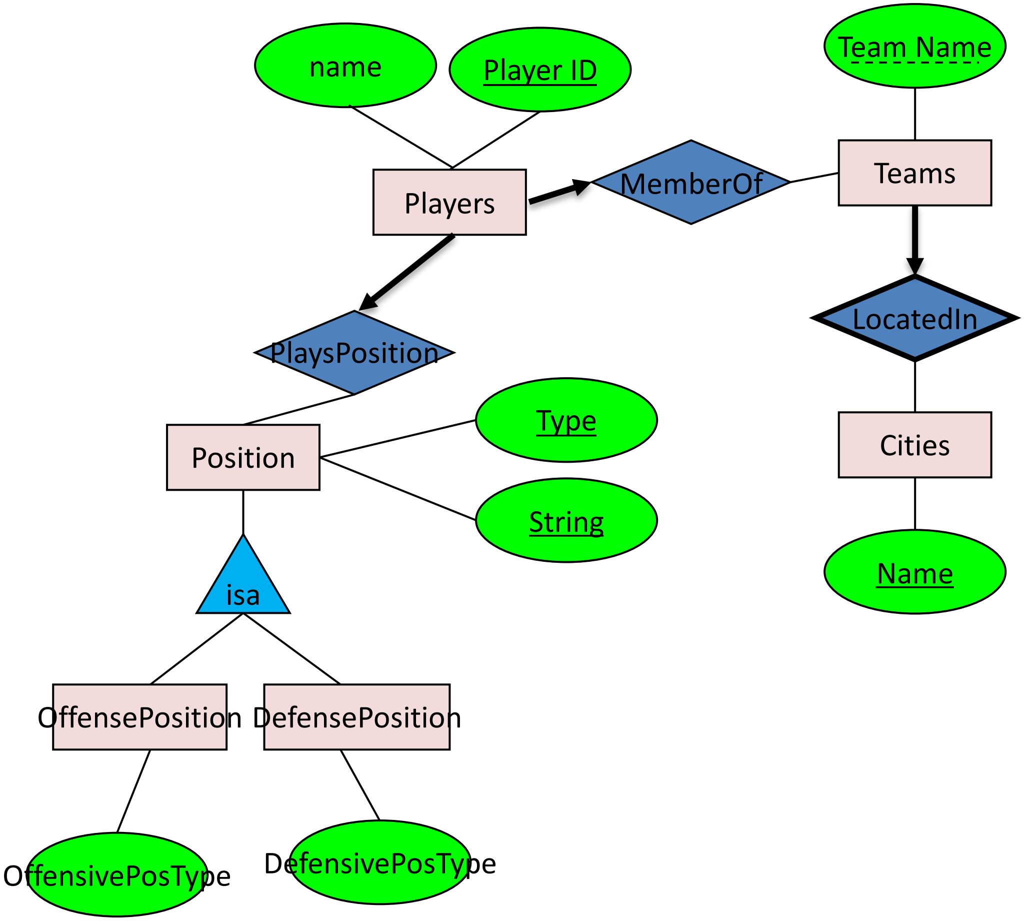
All passes are to exactly one player; all runs include a player



Make sure you have designated keys for all our concepts!



Teams belong to cities- model as ***weak entity sets***



Players are either on Offense or Defense, and are of types (QB, RB, WR, TE, etc.)