# CS 445
# Introduction to Database Design
# E-R Diagrams

## Chapter 2

## September 2, 2009

# Design Steps

- Read Chapter 2
  - homework: page 52: 2.2 (1–5) (Due Sept 14)

- How do we model the data?
  - what do we need to identify?

# Design Steps

- Requirement Analysis

  - talk to the user!

- Conceptual Database Design

  - E-R Diagram

- Logical Database Design

  - logical schema

- Schema Refinement

  - normalization

- Physical Database Design

  - performance tuning

- Application and Security Design

  - GUI / end user software

# Bits of Data

- Entity
  - some particular object in the real world

- Entity Set

- Attribute
  - domain
  - key
  - candidate key
  - primary key

# Doing interesting things with data

- Relationship
  - association among two or more entities

- Relationship Set

- Descriptive attribute

- Roles

# Constraints

- What limits are placed on how entities are involved in a relationship
    - Key Constraints
        - One to many

        - Many to many

        - One to one

    - Participation Constraints

# Weak Entities

- Entities without keys!

- Identifying owner



- Identifying relationship

# Class Hierarchy

- Some entities may be related
    - similar to Object Oriented class hierarchy
    - C++/Java
    - superclass
    - specialized subclasses

- Inheritance
    - ISA

- Overlap constraints

# Aggregation

- View a set of entities/relationships as one big entity
    - meta-entity

# How do we use all this?

- When do we use an entity vs an attribute to represent data?

    - it all depends on how you want to use the data

    - how many other bits of data will reference it?

    - how will they reference it?

        - will our model allow that?

- Example: Name and Address

CS445
Pacific University

# How do we use all this?

- When do we use an entity vs a relationship?

# Tool Support

- E–R diagram builders
  - Microsoft Visio
  - MySQL Workbench (alpha, buggy, promising)

- Unified Modeling Language (UML)
  - used to model all kinds of data interactions
  - Object Oriented code design
  - database design
    - think of entities and relationships as classes
  - Use cases (process flow)
  - http://argouml.tigris.org/

# Key Constraints

- emp MANAGES dept

- each emp can manage more than one dept

- each dept is managed by only one emp

    - Each dept key appears in ONE MANAGES relationship

    - ONE TO MANY

    - one employee can be associated with MANY depts

    - each dept associated with ONE emp

    - what if each emp ONLY managed ONE dept? (ONE TO ONE)

- emp WORKSIN dept

    - each emp can work in several depts

    - each dept has several emp

    - MANY TO MANY

    - what is each emp worked in only one DEPT?