**Database Design & Administration - CST8250**

**Week 2 - January 18th**

Class Intro

* Labs are there for your benefit so you can go if you need help or you’re free to just stop by
* This we are going to talk about drawing diagrams

Objectives

* Define terms related to entity relationship modeling, including entity, entity instance, attribute, relationship and cardinality, and primary key (covered in week 1 slides).
* Describe the entity modeling process.
* Review relationships and cardinality.
* Discuss how to draw an entity relationship diagram.
* Describe how to recognize entities, attributes, relationships, and cardinalities.

Database Model

* A database can be modeled as:
  + a collection of entities,
  + relationship among entities.
  + Depending on what you’re looking at, it will be some kind of combination of the two
* Database systems are often modeled using an Entity Relationship (ER) diagram as the "blueprint" from which the actual data is stored — the output of the design phase.

Entity Relationship Diagram (ERD)

These diagrams help communicate the concepts (the data, how it is stored etc.) before the final stage.

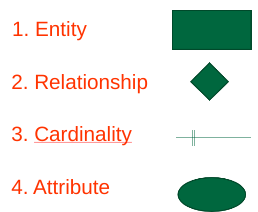
* ER model allows us to sketch database designs
* ERD is a graphical tool for modeling data.
* ERD is widely used in database design
* ERD is a graphical representation of the logical structure of a database
* ERD is a model that identifies the concepts or entities that exist in a system and the relationships between those entities

Purposes of ERD

* An ERD serves several purposes
  + An ERD allows you to question and evaluate what you are trying to do.
* The database analyst/designer gains a b**etter understanding of the information to be contained in the database** through the process of constructing the ERD.
* The ERD serves as a documentation tool.
* Finally, the ERD is used to **communicate the logical structure of the database to users.** In particular, the ERD effectively communicates the logic of the database to users.
  + As you go through the different stages of diagramming you will show these to the stakeholders. This way you can both communicate your understanding so that by the end of the proces you are pretty close to what you need

Components of an ERD

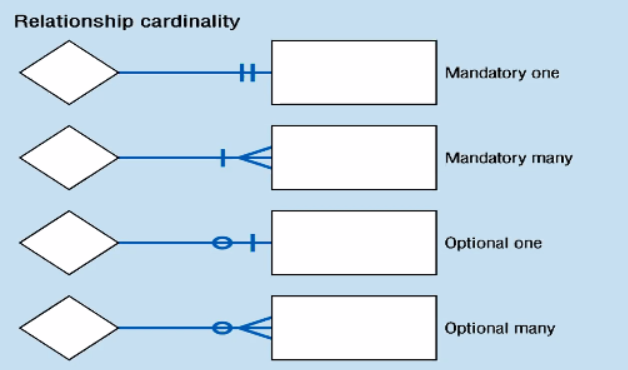
* An ERD typically consists of four different graphical components:

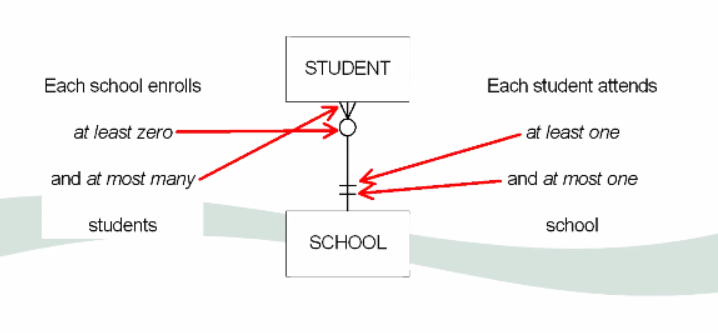


Classification of Relationship

* Optional Relationship
  + An Employee may or may not be assigned to a Department
  + A Patient may or may not be assigned to a Room
* Mandatory Relationship
  + Every Course must be taught by at least one Teacher
  + Every Mother has at least one Child

Cardinality Constraints

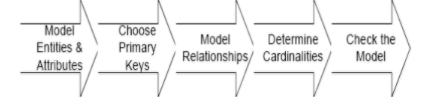
* Express the number of entities to which another entity can be associated via a relationship set.
* Cardinality Constraints - the number of instances of one entity that can or must be associated with each instance of another entity.
* Minimum Cardinality
* If zero, then optional
* If one or more, then mandatory
* Maximum Cardinality
* The maximum number
* For a binary relationship set the mapping cardinality must be one of the following types:
* One to one
  + A Manager Heads one Department and vice versa
* One to many ( or many to one)
* An Employee Works in one Department or one Department has many Employees
* Many to many
  + A Teacher Teaches many Students and a Student is taught by many Teachers
* 
* Constraint example:
  + In our model, we wish to indicate that each school may enroll many students, or may not enroll any students at all.
  + We also wish to indicate that each student attends exactly one school. The following diagram on the next slide indicates this optionality and cardinality.



General Steps to create an ERD

* Identify the entity
* Identify the entity's attributes
* Identify the primary keys
* Identify the relationship between entities
* Identify the cardinality constraint
* Draw the ERD
* Check the ERD

Steps in building an ERD



In Depth Steps to Develop an ERD

* With practice, you may find it simpler to follow the steps outlined in the previous slides
  + Dan often does not do the conceptual diagram and goes straight to the actual diagram because he has a lot of experience
  + For example if you put together PCs a lot, you’ll eventually start skipping steps because you have enough experience. This is similar to diagramming.
* We will follow this detailed process in the next example:
* 1. Identify Entities
* 2. Find Relationships
* 3. Draw Rough ERD
* 4. Fill in Cardinality
* 5. Define Primary Keys
* 6. Draw Key-Based ERD
* 7. Identify Attributes
* 8. Map Attributes
* 9. Draw fully attributed ERD
* 10. Check Results

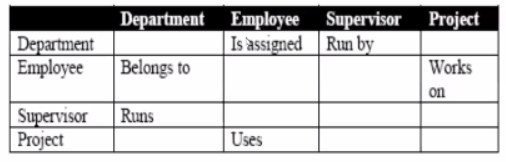
**A Simple Example**

* **A company has several departments. Each department has a supervisor and at least one employee. Employees must be assigned to at least one, but possibly more departments. At least one employee is assigned to a project, but an employee may be on vacation and not assigned to any projects. The important data fields are the names of the departments, projects, supervisors and employees, as well as the supervisor and employee number and a unique project number.**

Step 1: Identify entities

* One approach to this is to work through the information and **highlight** those words which you think correspond to entities.
* A **company** has several **departments**. Each department has a supervisor and at least one **employee**. Employees must be assigned to at least one, but possibly more departments. At least one employee is assigned to a **project**, but an employee may be on vacation and not assigned to any projects. The important data fields are the names of the departments, projects, supervisors and employees, as well as the supervisor and employee number and a unique project number.
  + A true entity should have more than one instance
* A true entity should have more than one instance

Step 2: Find Relationships

* Aim is to identify the associations, the connections between pairs of entities.
* A simple approach to do this is using a relationship matrix (table) that has rows and columns for each of the identified entities.
* Go through each cell and decide whether or not there is an association. For example, the first cell on the second row is used to indicate if there is a relationship between the entity "Employee" and the entity "Department".
* Go through each cell and decide whether or not there is an association. For example, the first cell on the second row is used to indicate if there is a relationship between the entity "Employee" and the entity "Department".
* 

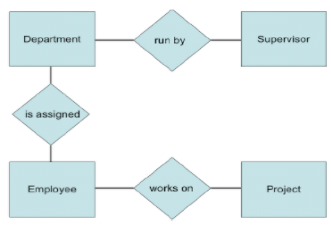
Step 2: Identified Relationships

These are sometimes termed business rules but as we are modeling more than business nowadays it is not always termed this way.

* Names placed in the cells are meant to capture/describe the relationships.
* A Department is *assigned* an employee
* A Department is *run* by a supervisor
* An employee *belongs* to a department
* An employee *works* on a project
* A supervisor *runs* a department
* A project *uses* an employee

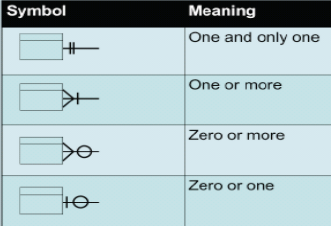
Step 3: Draw Rough ERD

* Draw a diagram and:
  + Place all the entities in rectangles
  + Use diamonds and lines to represent the relationships between entities.

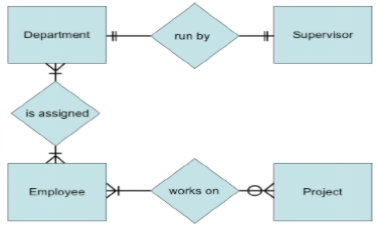


Step 4: Fill in Cardinality

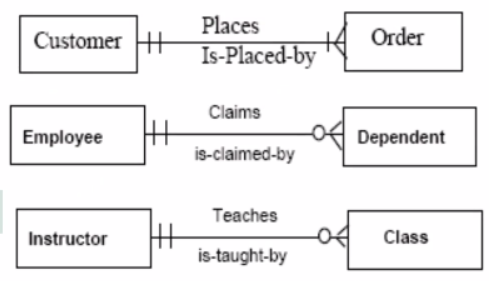
* Supervisor
  + Each department has **one** supervisor.
* Department
  + Each supervisor has **one** department.
  + Each employee can belong to **one or more** departments
* Employee
  + Each department must have **one or more** employees
  + Each project must have **one or more** employees
* Project
  + Each employee can have **0 or more** projects.
* The cardinality of a relationship can only have the following values
  + One and only one
  + One or many
  + Zero or many
  + Zero or one

Cardinality Notation

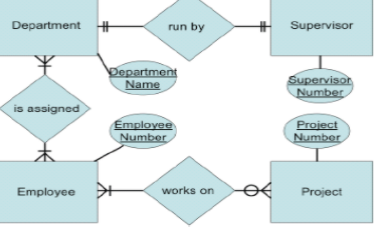
Step 4: ERD with cardinality



* These diagrams are taken from a textbook
* This can also be drawn without diamonds:



Step 5, 6: Rough ERD Plus Primary Keys

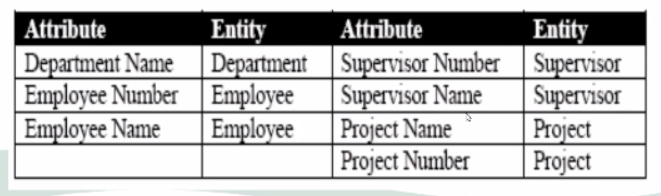


* Attributes are added - supervisor number, project number, department name employee number
* With very little learning, you can get the stakeholders to understand the diagram

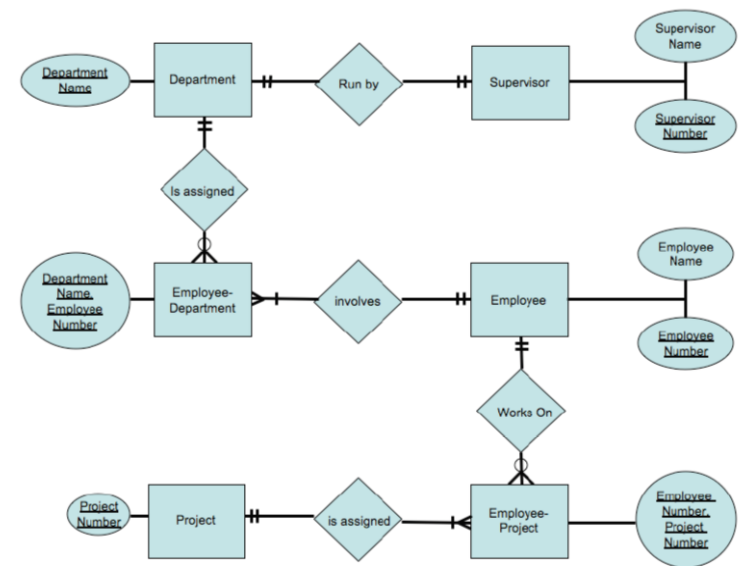
Step 7: Identify Attributes

* In this step we try to identify and name all the attributes essential to the system we are studying without trying to match them to particular entities.
* The best way to do this is to study the forms, files and reports currently kept by the users of the system and circle each data item on the paper copy.
* Cross out those which will not be transferred to the new system, extraneous items such as signatures, and constant information which is the same for all instances of the form (e.g. your company name and address). The remaining circled items should represent the attributes you need. You should always verify these with your system users. (Sometimes forms or reports are out of date.)
* The only attributes indicated are the names of the departments, projects, supervisors and employees, as well as the supervisor and employee NUMBER and a unique project number.

Step 8: Map Attributes

* **For each attribute we need to match it with exactly one entity**. Often it seems like an attribute should go with more than one entity (e.g. Name). In this case you need to add a modifier to the attribute name to make it unique (e.g. Customer Name, Employee Name, etc.) or determine which entity an attribute "best' describes.
* If you have attributes left over without corresponding entities, you may have missed an entity and its corresponding relationships. Identify these missed entities and add them to the relationship matrix now
* This chart should be 2 wide, 7 long:
  + 

Step 9: Draw Fully Attributed ERD



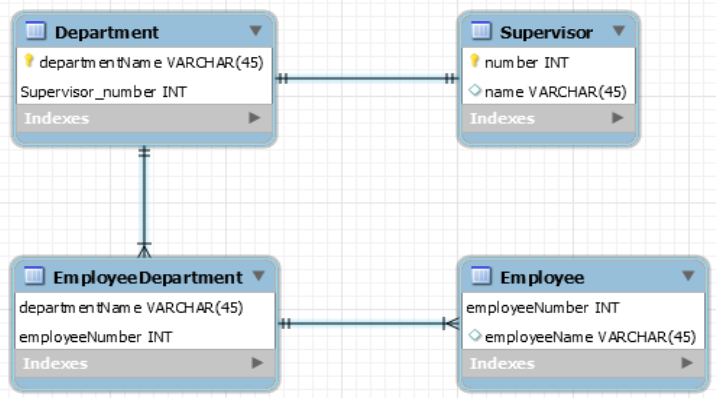
* Normally you want to try and build these in alphabetical order

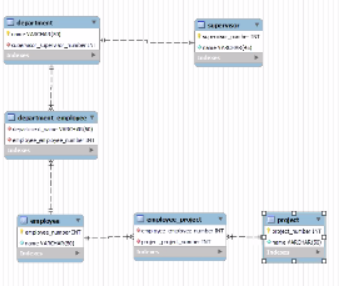
Step 10: Check ERD Results

* Look at your diagram from the point of view of a system owner or user. **Is everything clear?**
* Check through the Cardinality pairs.
* Also, look over the list of attributes associated with each entity to see if anything has been omitted.

Converting to Physical Diagram

* Convert all entities into tables
* All single valued attributes of an entity become columns
* Key attributes in the diagram become primary keys
* Multi-valued attributes become tables
* Composite attributes become separate columns
* Ignore derived attributes
* Assign data types



* ERStudio and Toad are two designers
* Remember to use the proper naming conventions, only using lower case in this course
* 
* Any “number” becomes “id” in the database because programmers will not have to guess what the primary key is

**You have 2 weeks to finish your lab :)**