The Entity Relationship Model

Data Modeling

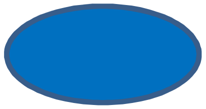
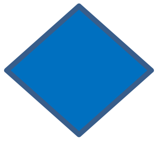
Data modeling – a technique for organizing and documenting a system's data. Sometimes called *database modeling*.

Entity relationship diagram (ERD) – a data model utilizing several notations to depict data in terms of the entities and relationships described by that data.

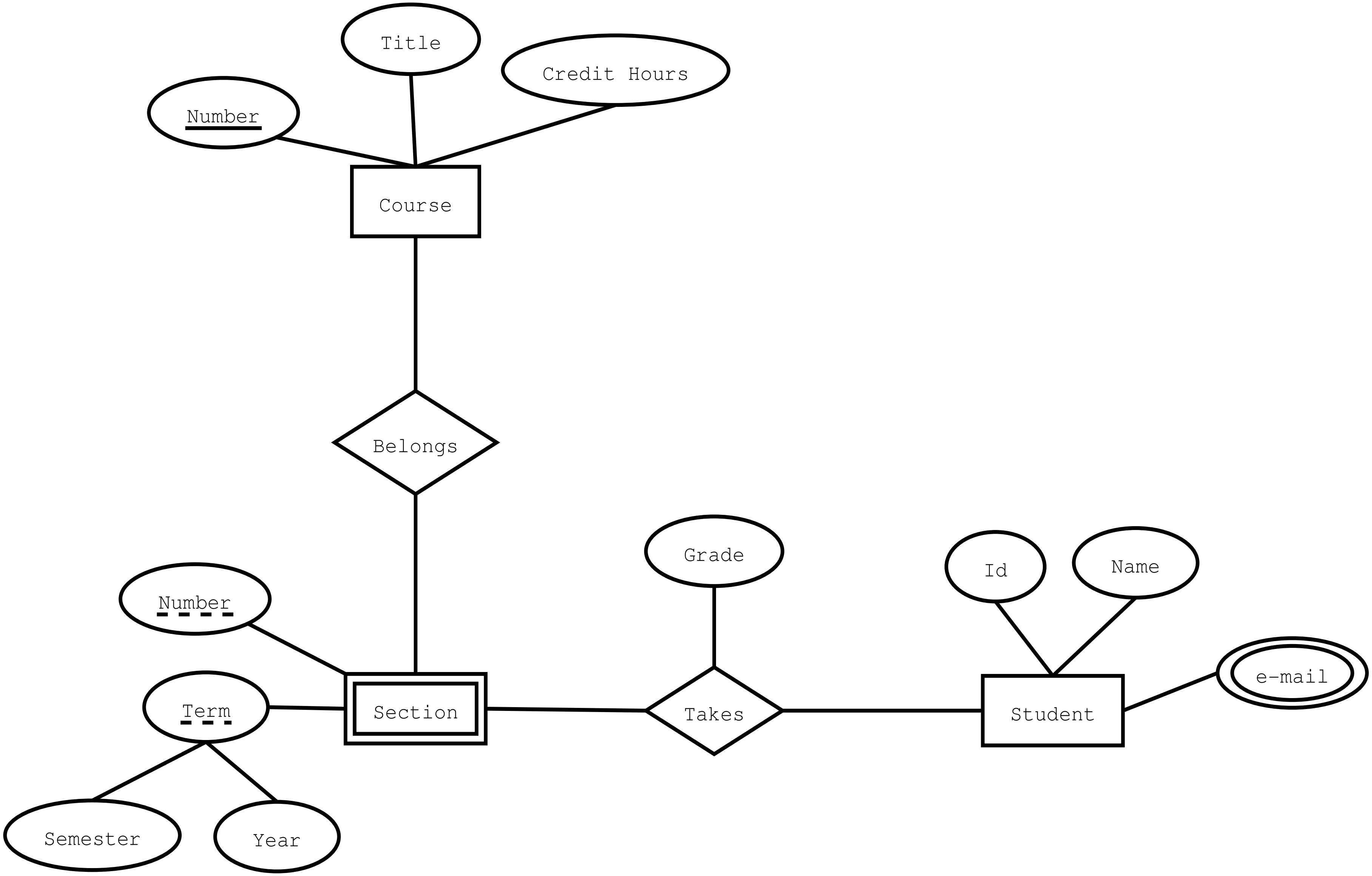
Purpose of E-R Model

- Facilitates database design
- Express logical properties of mini-world of interest within enterprise - Universe of Discourse
- Conceptual level model
- Not limited to any particular DBMS
- E-R diagrams used as design tools
- A semantic model – captures meanings

Symbols used in E-R Diagram

- Entity – rectangle
- Attribute – oval 
- Relationship – diamond 
- Link - line

ER diagram

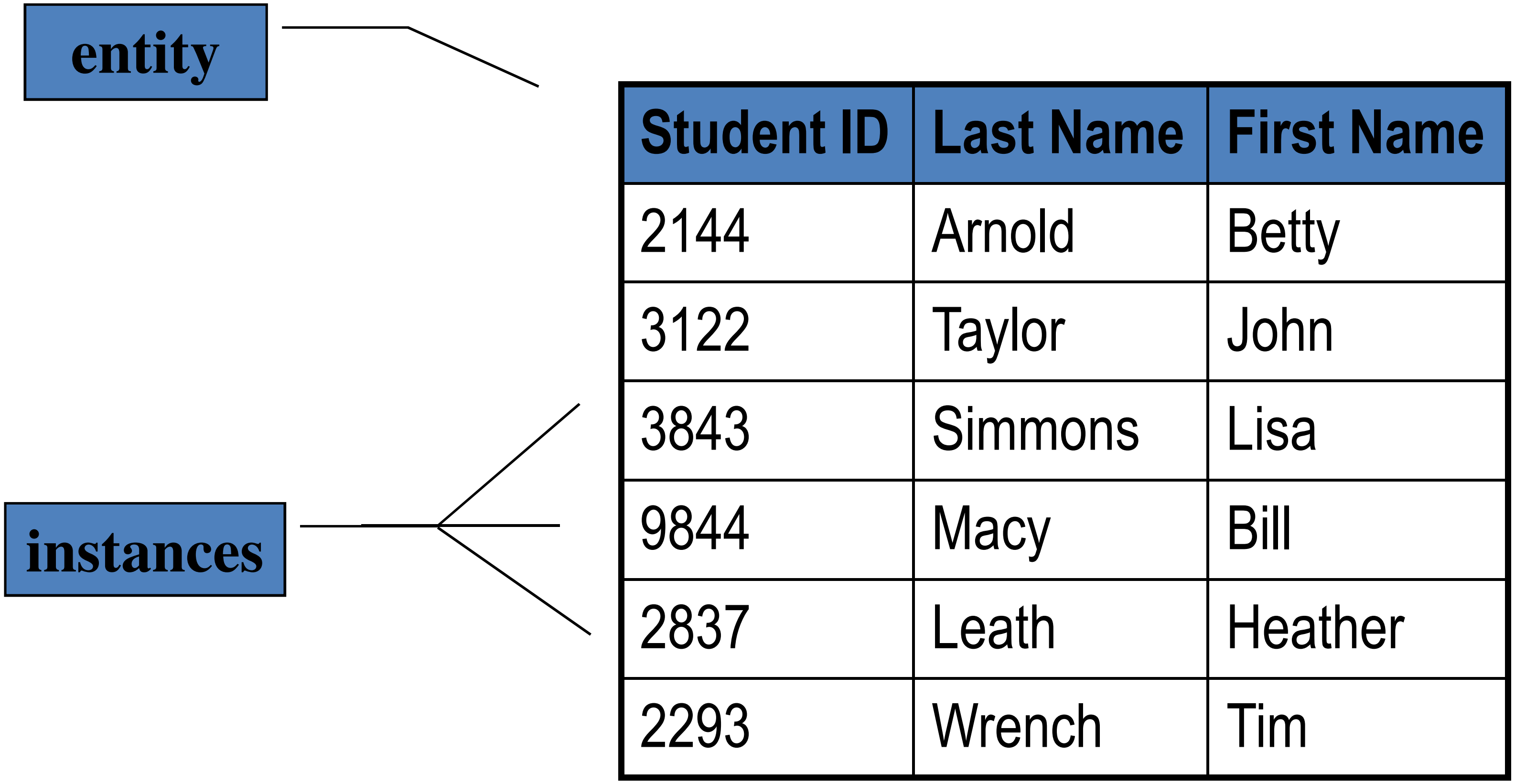

Title
Credit Hours
Number
Course
Belongs
Grade
Id
Name
Number
Term
Section
Takes
Student
e-mail
Semester
Year

Entity

- Object that exists and that can be distinguished from other objects
- Can be person, place, event, object, concept in the real world
- Can be physical object or abstraction
- Entity **instance** is a particular person, place, etc.
- Entity **type** is a category of entities
- Entity **set** is a collection of entities of same type-must be well-defined
- In E-R diagram, rectangle represents entity set

Data Modeling Concepts: Entity

Entity instance – a single occurrence of an entity.



entity

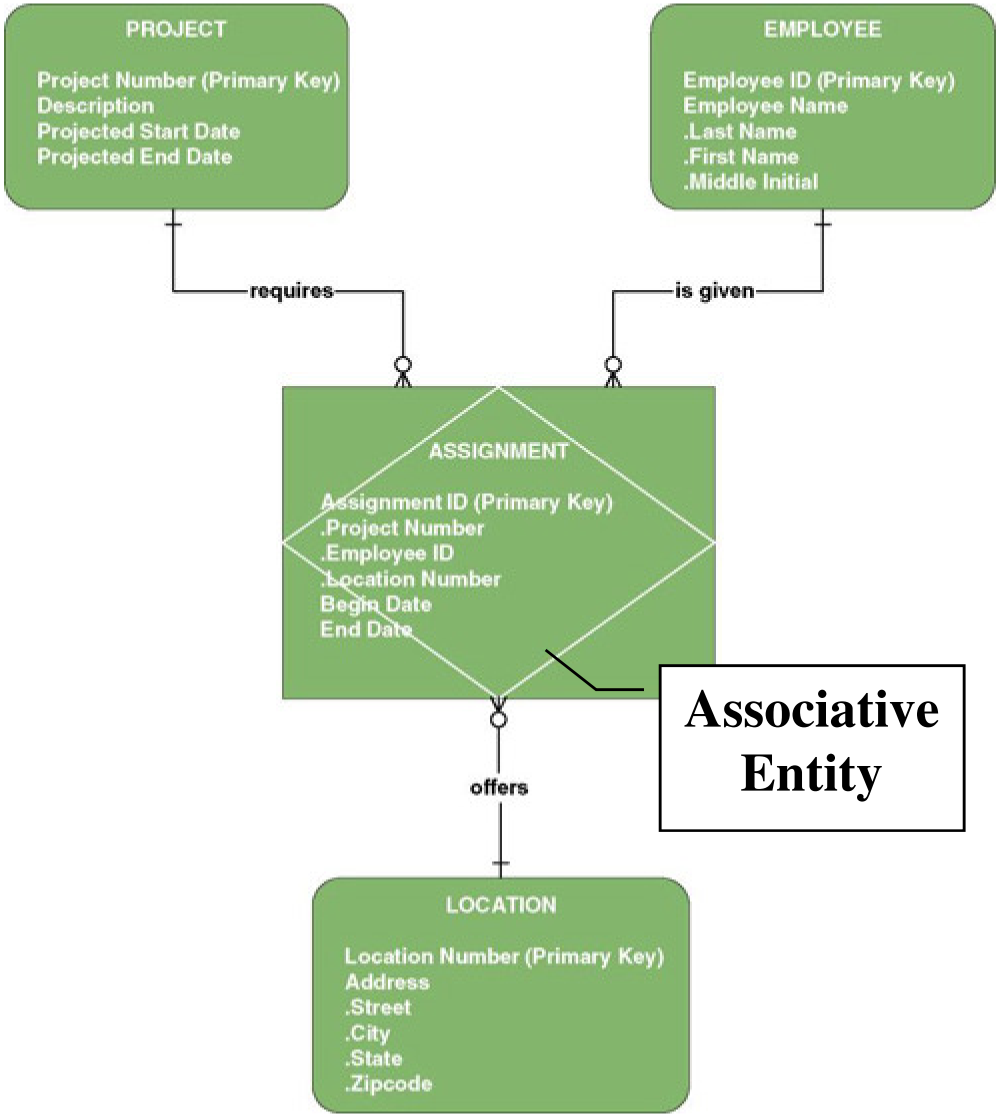
instances

Student ID	Last Name	First Name
2144	Arnold	Betty
3122	Taylor	John
3843	Simmons	Lisa
9844	Macy	Bill
2837	Leath	Heather
2293	Wrench	Tim

Data Modeling Concepts: Degree

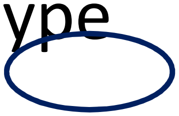
Associative entity – an entity that inherits its primary key from more than one other entity (called parents).

Each part of that concatenated key points to one and only one instance of each of the connecting entities.



Attributes

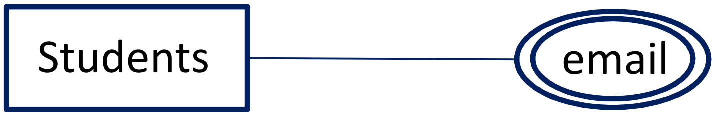
- Defining properties or qualities of entity type
- Represented by oval on E-R diagram



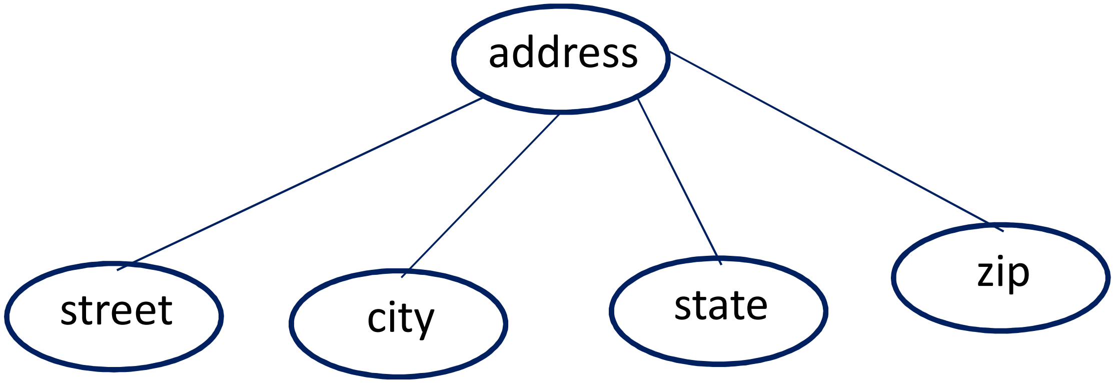
- **Domain** – set of allowable values for attribute
  - Credit hours might be integer values between 0 and 150
  - lastName might be all legal last names – a string that might also include apostrophes, blanks, hyphens, or other special characters
- Attribute maps entity set to domain
- May have **null values** for some entity instances – no mapping to domain for those instances

Attributes (cont.)

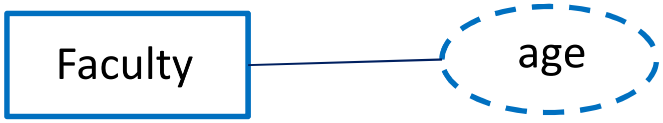
- May be **multi-valued** – use double oval on E-R diagram (e.g., student may have more than one email address)



- May be **composite** – use oval for composite attribute, with ovals for components connected to it by lines



- May be **derived** – use dashed oval



Keys

- **Superkey**: attribute or set of attributes that uniquely identifies an entity (can always tell one entity instance from another)
  - stuId is superkey for Student entity
  - stuId, credits together form a superkey, because stuId is a superkey
- **Composite key**: key with more than one attribute
  - courseNumber, sectionNumber, semester make up a composite key
- **Candidate key**: superkey such that no proper subset of its attributes is also a superkey (minimal superkey –no unnecessary attributes)
  - Although stuId, credits is a superkey, only stuId is a candidate key
  - courseNumber, sectionNumber, semester is a candidate (no one attribute is a superkey)
  - If Student entity contains stuId and ssan as atributes, both stuId and ssan are candidate keys

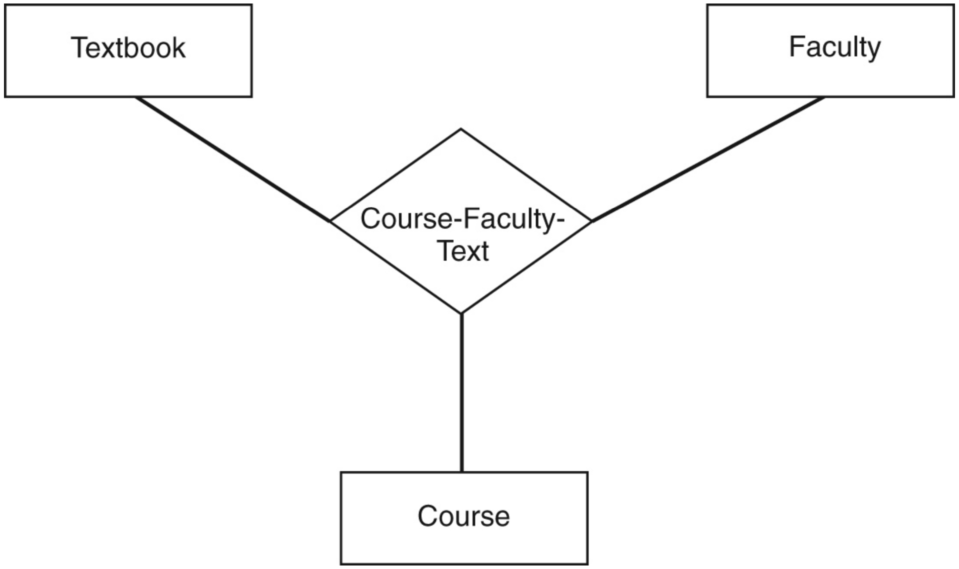
Keys (cont.)

- **Primary key**: the candidate key actually used for identifying entities and accessing records
- **Alternate key**: candidate key not used for primary key
- **Secondary key**: attribute or set of attributes used for accessing records, but not necessarily unique
  - lastName might be used to find instances in Student, to help narrow down the results
- **Foreign key**: term used in relational model (but not in the E-R model) for an attribute that is primary key of a table and is used to establish a relationship, usually with another table, where it appears as an attribute also
  - stuId in Enroll entity

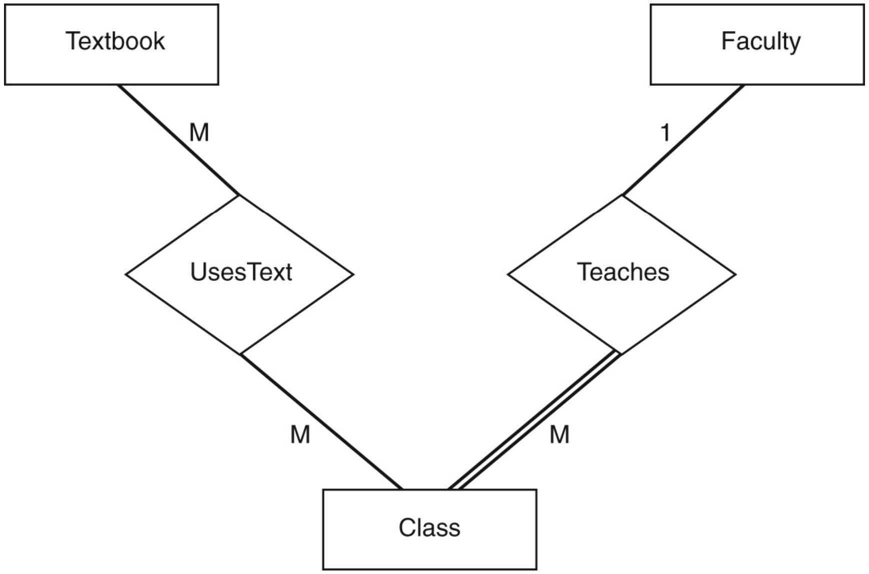
Relationships

- Connections or interactions between entity instances
- Represented by diamond on E-R diagram
- Relationship **type** – category of relationships
- Relationship **set** – collection of relationships of same type, consists of relationship instances – relationships that exist at a given moment
- Type forms intension; set forms extension of relationship
- Relationship can have descriptive attributes
- **Degree** of relationship
  - **Binary** – links two entity sets; set of ordered pairs
  - **Ternary** – links three entity sets; ordered triples
  - **N-ary** – links n entity sets; ordered n-tuples
  - Note: ternary relationships may sometimes be replaced by two binary relationships

Replace Ternary Relationship with Two Binary Relationships



Ternary Relationship



Binary Relationship

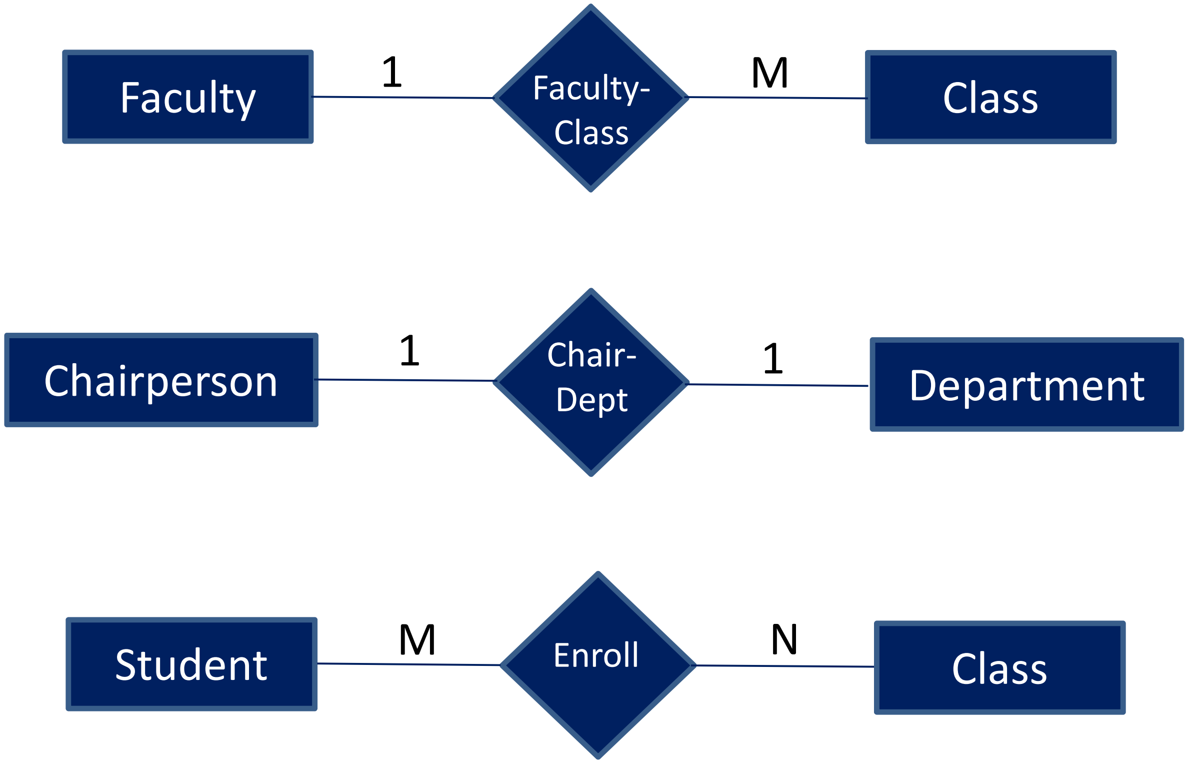
Cardinality of Relationships

- Number of entity instances to which another entity can map under the relationship
- **One-to-one**: X:Y is 1:1 is each entity in X is associated with at most one entity in Y and each entity in Y with at most one entity in X.
- **One-to-many**: X:Y is 1:M is each entity in X can be associated with many entities in Y, but each entity in Y with at most one entity in X.
- **Many-to-many**: X:Y is M:M if each entity in X can be associated with many entities in Y, and each entity in Y with many entities in X (many=more than one)

Showing Cardinalities on ER Diagram

One: 1

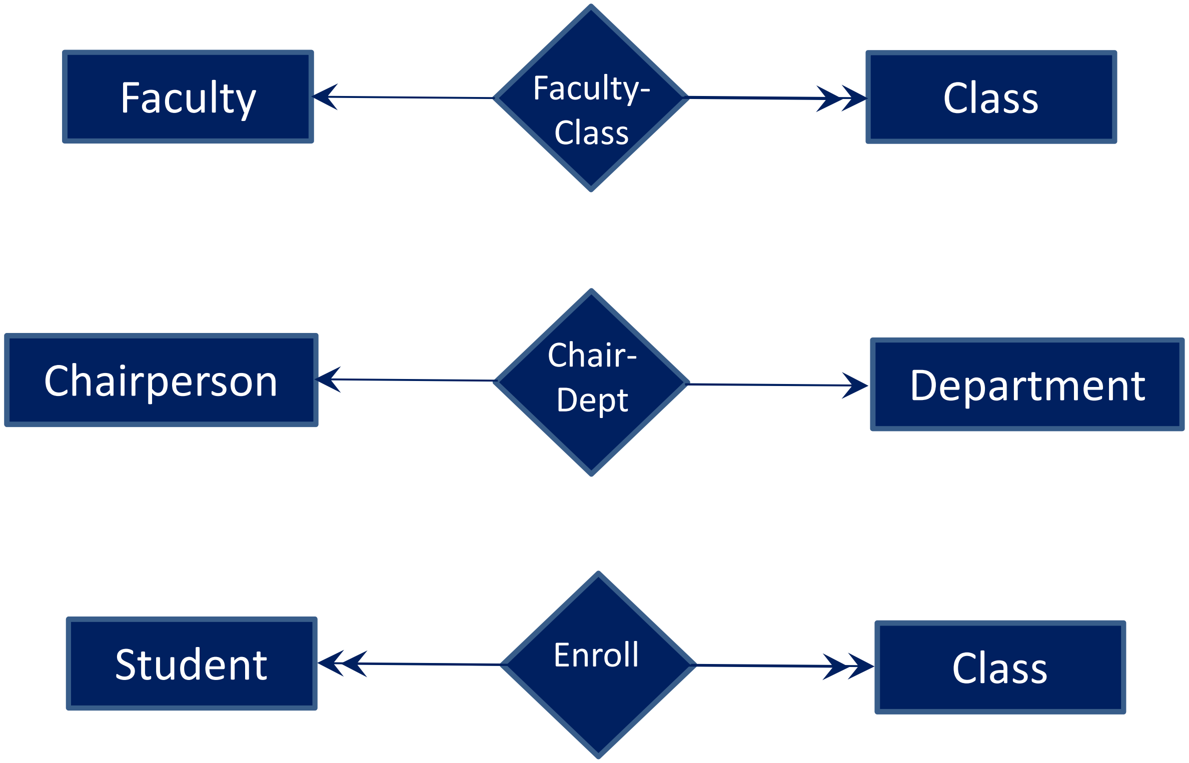
Many: M, n



Showing Cardinalities on ER Diagram

One: Single Arrow

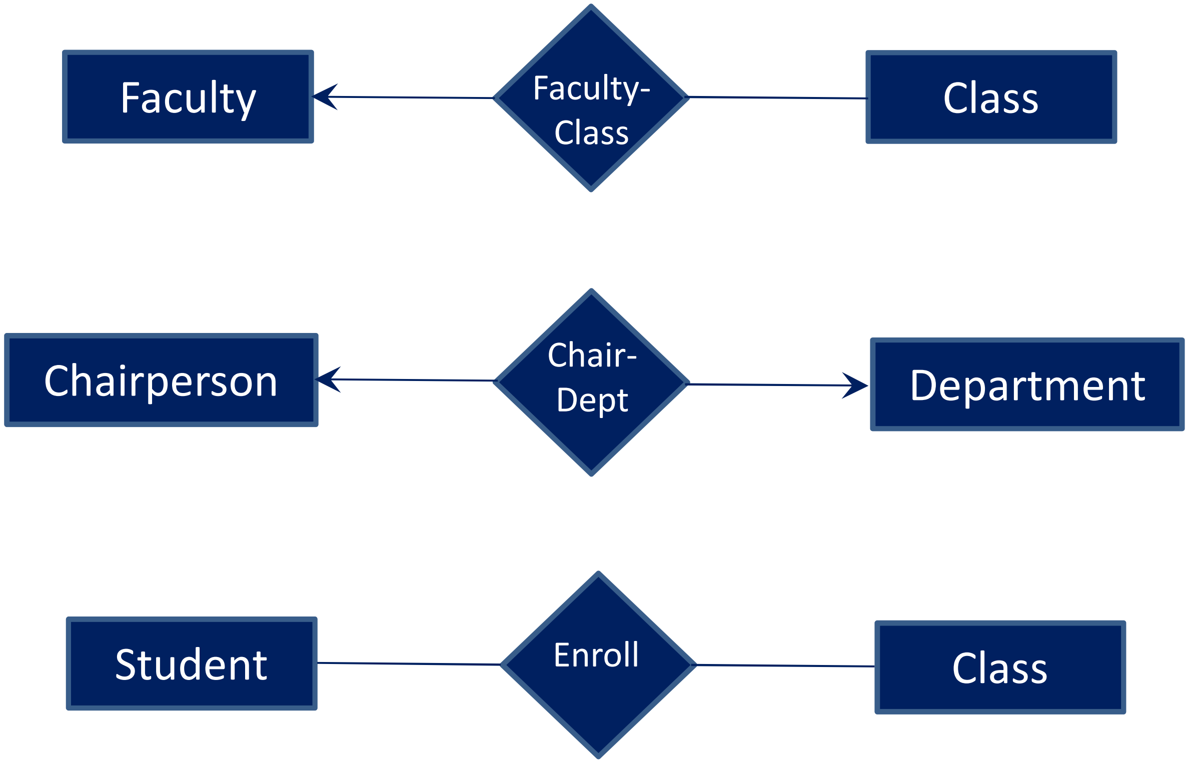
Many: Double Arrow



Showing Cardinalities on ER Diagram

One: Single Arrow

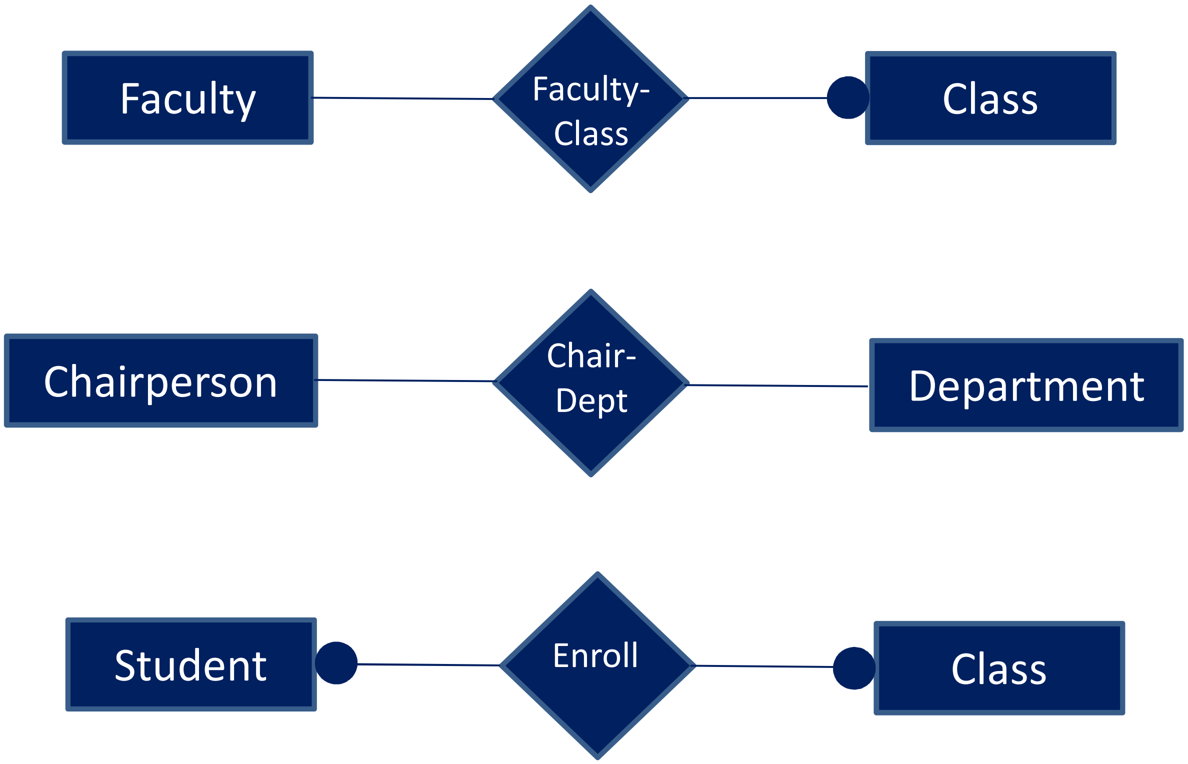
Many: No Arrow



Showing Cardinalities on ER Diagram

One: No Arrow

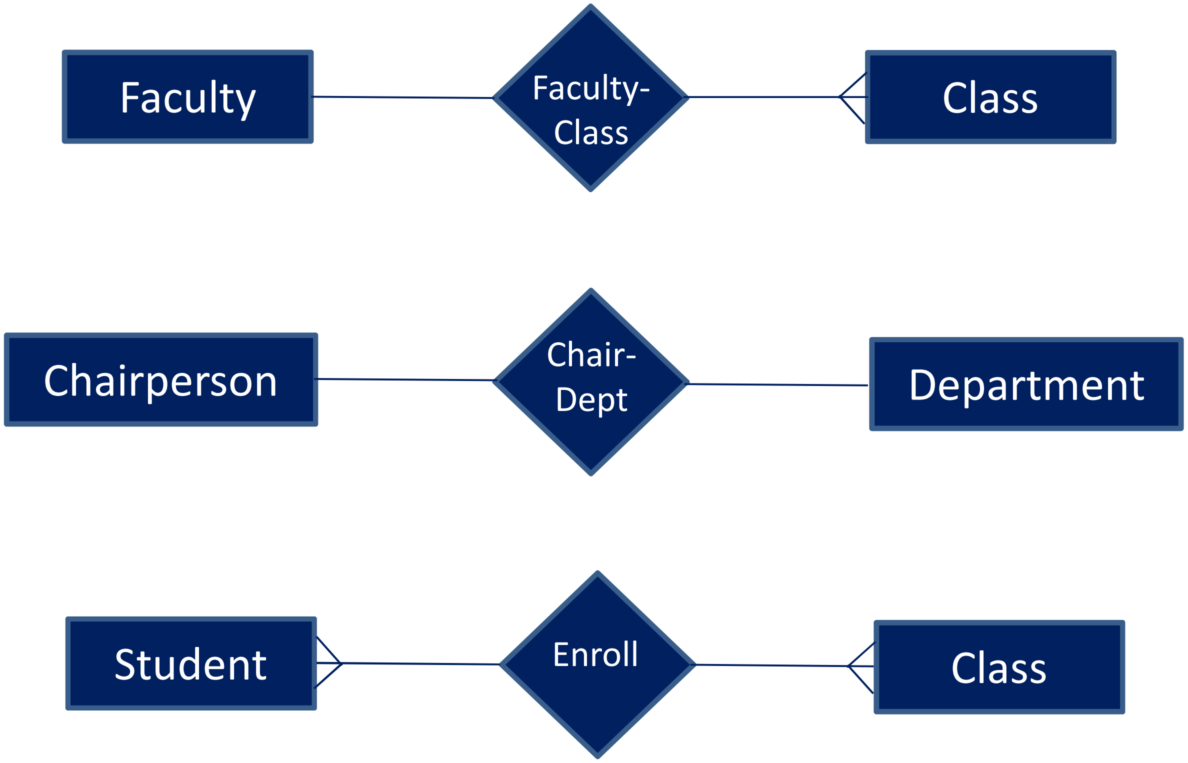
Many: Big Dot



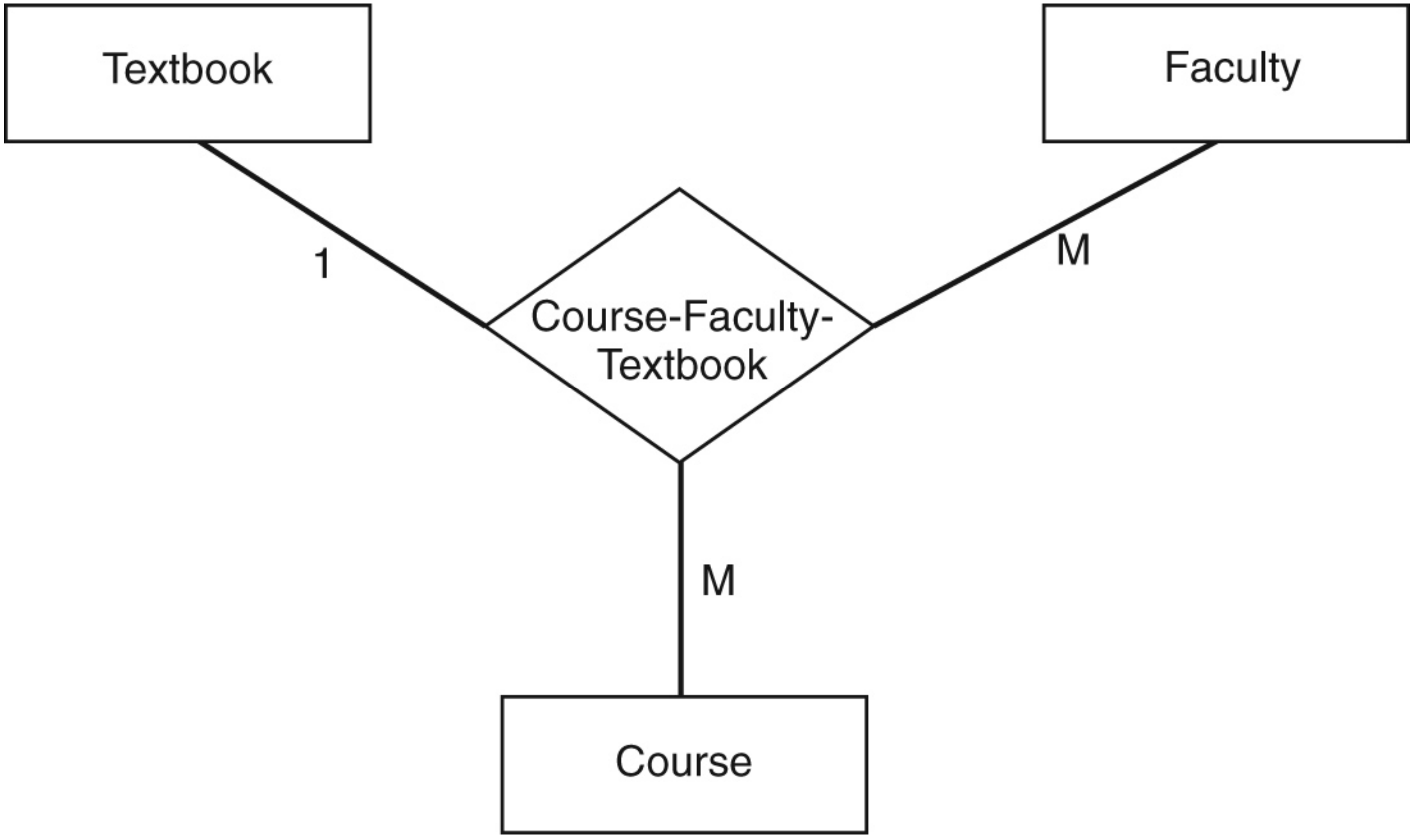
Showing Cardinalities on ER Diagram

One: No Arrow

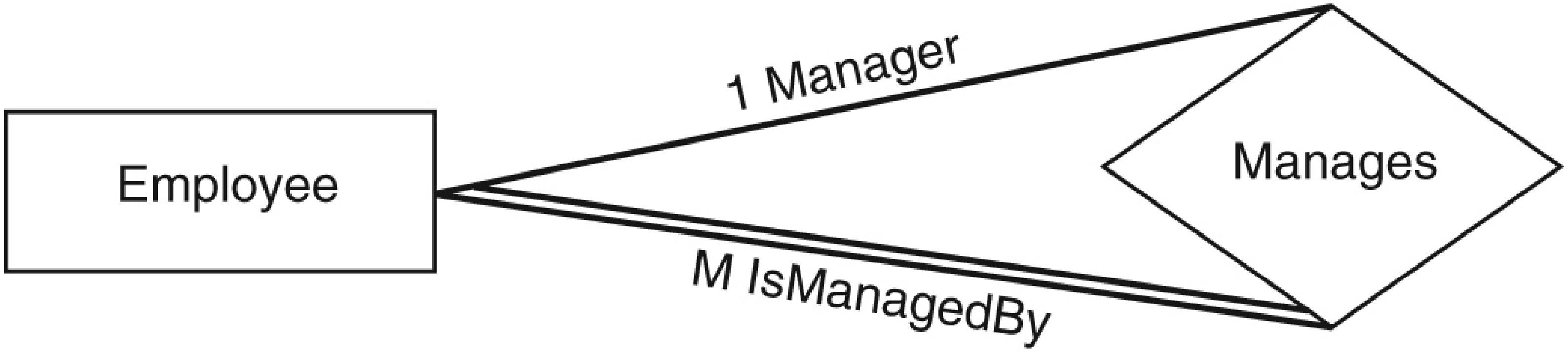
Many: Crow’s Feet



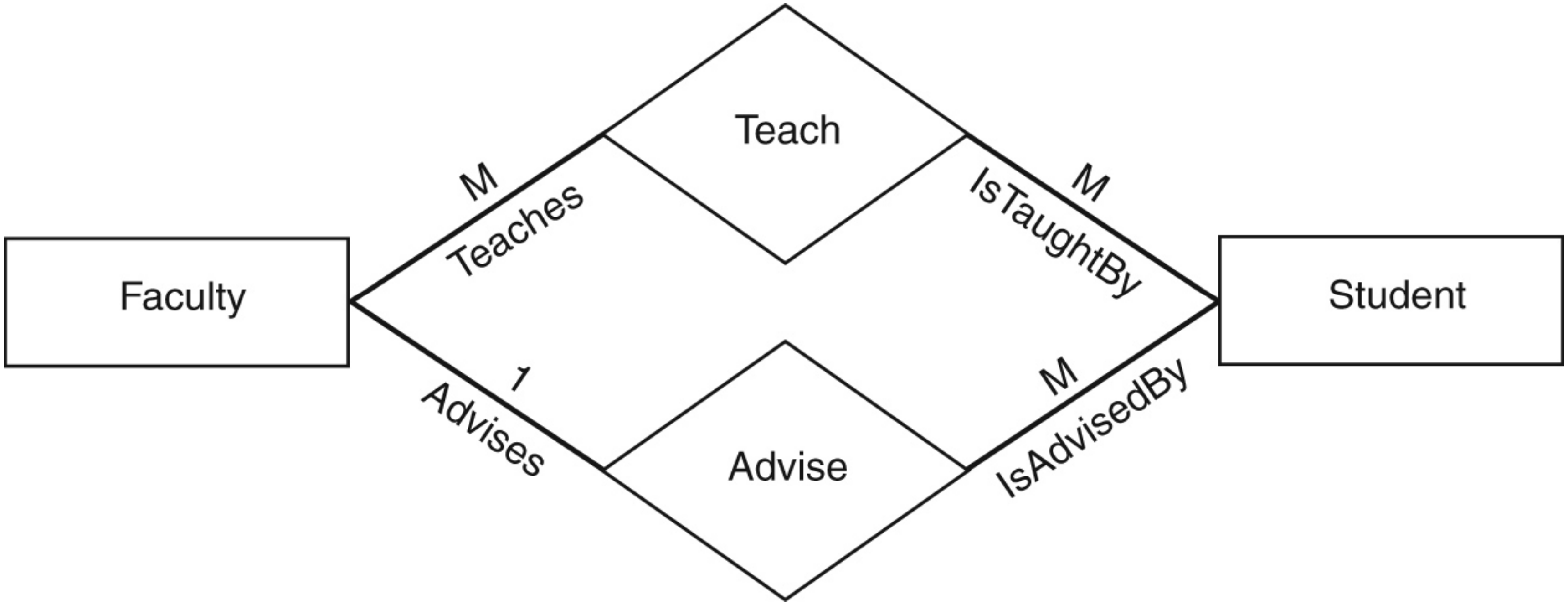
Cardinality Example



Recursive Relationship



Entity Sets with Two Relationships



Existence Dependency and Weak Entities

- Entity Y is **existence dependent** on entity X is each instance of Y must have a corresponding instance of X
- Y must have **total participation** in its relationship with X
- **If Y does not have its own candidate key**, Y is called a **weak entity**, and X is **strong entity**
- Weak entity may have a partial key, a **discriminator**, that distinguishes instances of the weak entity that are related to the same strong entity
- Use double rectangle for weak entity, with double diamond for relationship connecting it to its strong entity

Data Modeling Concepts: Parent and Child Entities

Parent entity - a data entity that contributes one or more attributes to another entity, called the child. In a one-to-many relationship the parent is the entity on the "one" side.

Child entity - a data entity that derives one or more attributes from another entity, called the parent. In a one-to-many relationship the child is the entity on the "many" side.

Data Modeling Concepts: Generalization

Generalization – a concept wherein the attributes that are common to several types of an entity are grouped into their own entity.

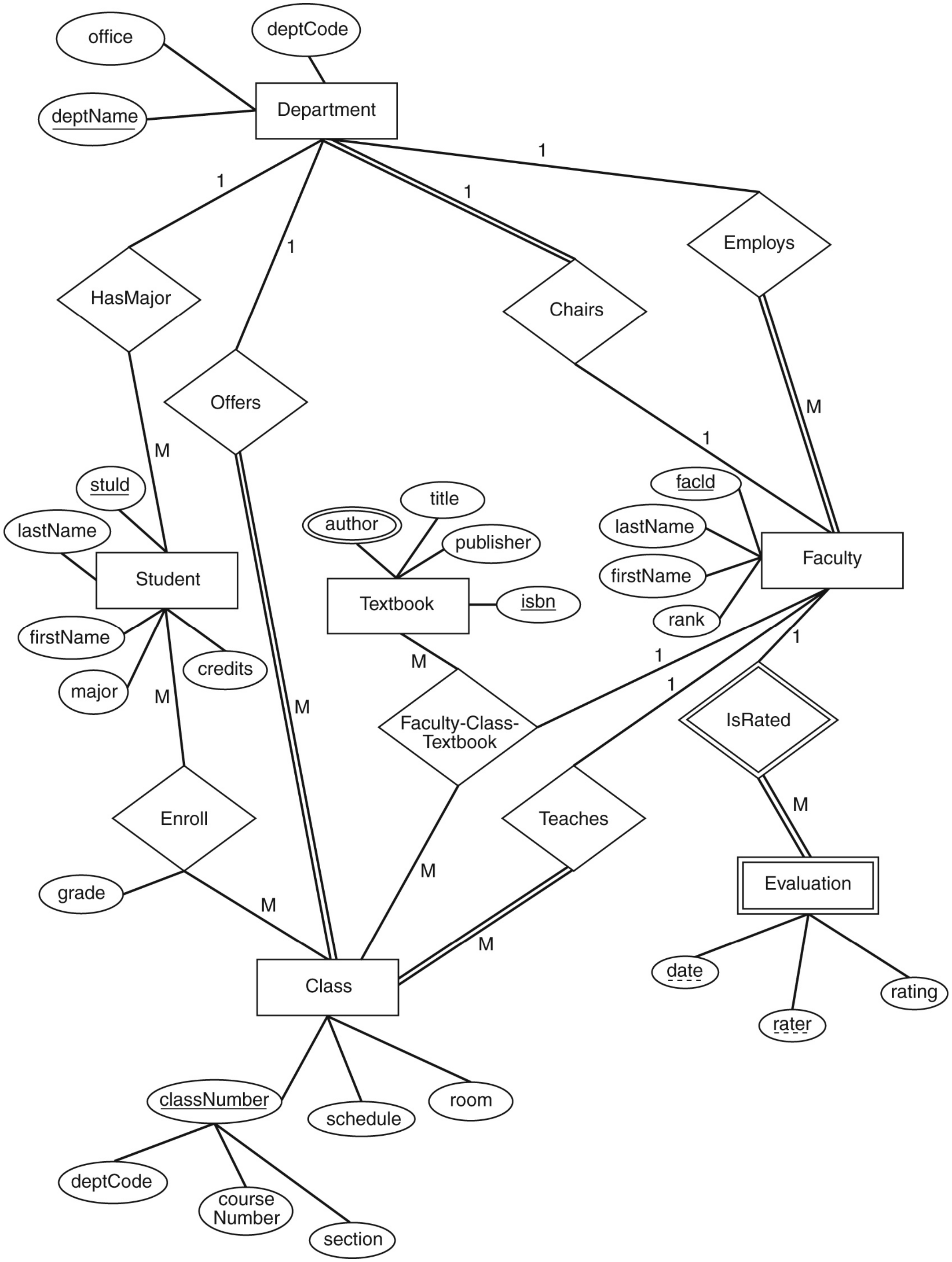
Supertype – an entity whose instances store attributes that are common to one or more entity subtypes.

Subtype – an entity whose instances may inherit common attributes from its entity supertype

And then add other attributes unique to the subtype.

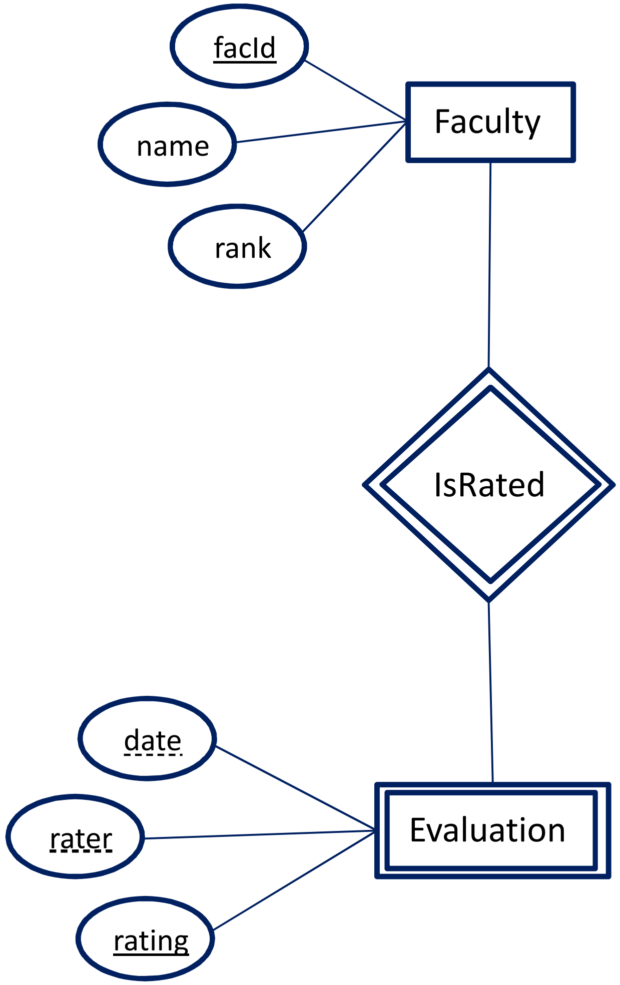
Entity Discovery

- In interviews or JRP sessions, pay attention to key words (i.e. "we need to keep track of ...").
- In interviews or JRP sessions, ask users to identify things about which they would like to capture, store, and produce information.
- Study existing forms, files, and reports.
- Scan use case narratives for nouns.
- Some CASE tools can reverse engineer existing files and databases.

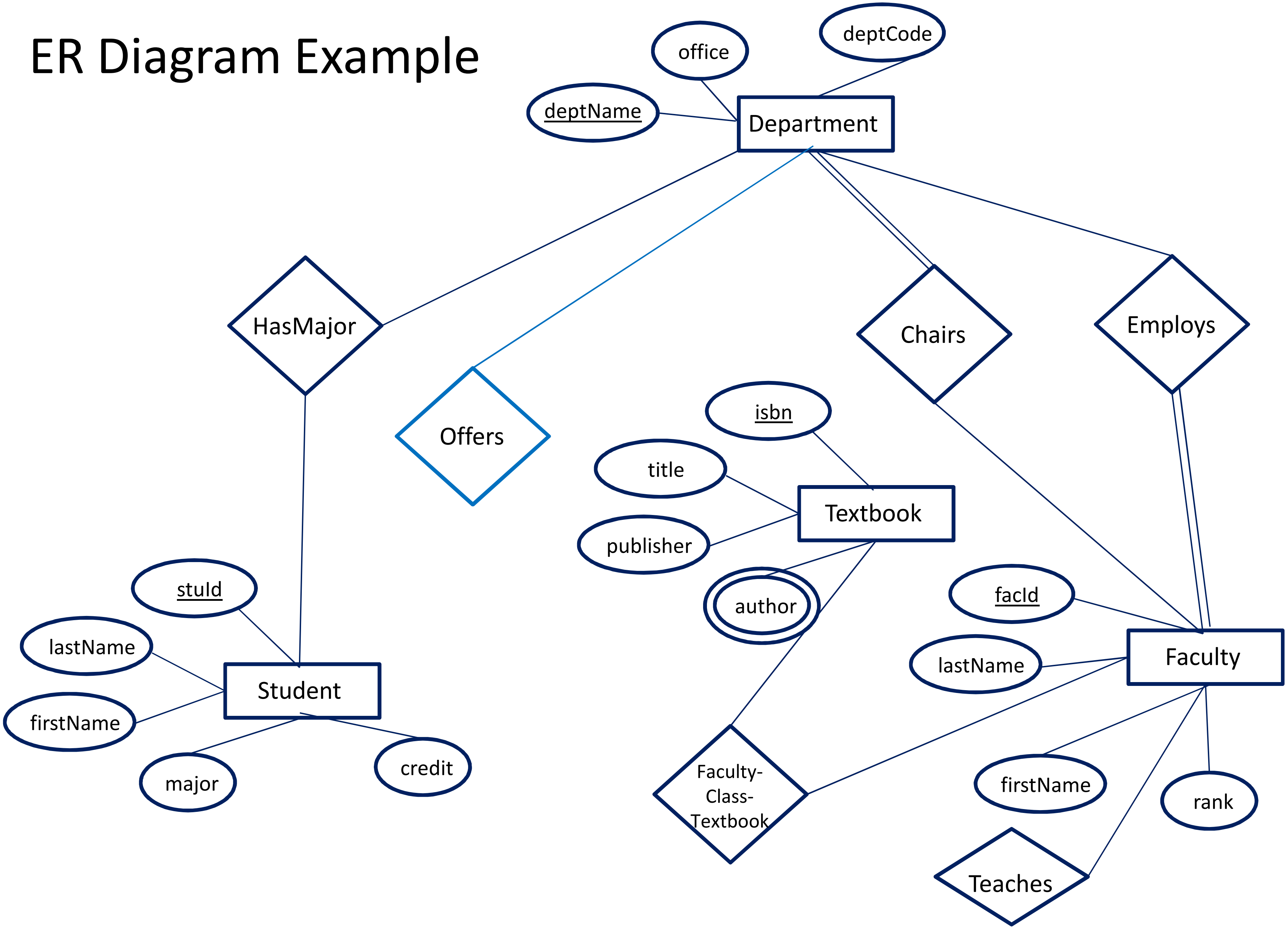


ER Diagram Example

Existence Dependency and Weak Entities Example



ER Diagram Example



E-R Diagram description

- Student: stuId, lastName, firstName, major, credits
  - Each student has a unique id and has at most one major
- Department: deptCode, deptName, office
  - Each department has a unique code and a unique name, and that each department has one office designated as the departmental office
- Faculty: facId, lastName, firstName, rank
  - facId is unique and that every faculty member must belong to department. One faculty member in each department is the chairperson.
- Class: classNumber, sched, room
  - classNumber consists of deptCode, courseNumber, section
- Textbook: isbn, author, title, publisher
  - A book can have multiple authors
- Evaluation: date, rater, rating
  - Evaluation is a weak entity, dependent on Faculty