CS143: E/R model

Book Chapters

(5th) Chapters 6.1-6.10
(6th) Chapters 7.1-7.10
(7th) Chapters 6.1-6.10

E/R model

Why E/R model?

- The first step of database construction is to figure out what data needs to be stored
  - Relations are not given
  - Talk to domain experts to learn the information that needs to be handled by the database
- E/R model: graphical, intuitive and informal representation of data
  - E/R model is often used to "document" what we learned about the domain
  - Entities, relations, attributes, ...
- Start with E/R model, convert it into table
  - E/R model is not directly implemented by DBMS
  - Some E/R tools perform this conversion semi-automatically
- Unfortunately, many variations of E/R models exist
  - We learn the model described in the textbook.
  - Due to Oracle's adoption, crow's foot notation is also popular

Entity-Relationship (E/R) Model

- Entity: "thing" or "object" in real world

  eg) I, this book, UCLA,
- Entity set: a set of entities (object). Like a class is OOPL.
  - Rectangle in ER
    eg) Students, Schools, Classes

- Consists of “name” and “attributes”
  ⟨ex: Students, Classes, Faculty⟩

- Can informally think of entities as records (tuples)
  ⟨show records for example⟩

- Key: a set of attributes that uniquely identifies an entity in an entity set, underline in E/R
  - Important: all entity sets need a key
    ⟨add keys to example⟩

  - No good way to notate multiple keys
- Relationship: connection between entities.
- Relationship set: a set of relations of the same kind
  - Diamond in ER
    ⟨ex: add Take, Teach⟩

  - Think of relationships as connections between entities (or as records)
    ⟨examples of each⟩

    * Not all entities need to participate in relationships.
    * Relationships can also have ATTRIBUTES

    ⟨add grade to Take, quarter to Teach⟩

CARDINALITY of relationships

1. ONE-TO-ONE: Each entity in E1 is related to at most one entity in E2 and vice-versa

   ⟨abstract dot diagram for entity sets E1 and E2⟩

   - Notation: arrow at the "one" end

   - **Q:** Meaning of one-to-one in Teach?

2. MANY-TO-ONE: Each entity in E1 is related to at most one entity in E2 (converse is ONE-TO-MANY)

   ⟨abstract picture⟩

   - Notation: arrow at the "one" end

   - **Q:** Meaning of many-to-one in Teach?

3. MANY-TO-MANY: Each entity in E1 may be related to 0 or more entities in E2 and vice-versa

⟨abstract picture⟩

  - Notation: no arrow.

  - **Q:** Meaning in Teach? Take?

- TOTAL PARTICIPATION: an entity participates in the relationship AT LEAST ONCE.
  - double lines in E/R

⟨eg: double line between Class and Teach. meaning?⟩

⟨eg: double line between Teach and Faculty. meaning?⟩

⟨eg: double line and arrow between Teach and Faculty. meaning?⟩

⟨eg: double lines at both sides of Teach vs one-to-one of Teach. The same?⟩

- GENERAL CARDINALITY NOTATION: l..h on an edge.
  - The object participate in a relationship l to h times
  - "*" means unlimited
    ⟨abstract diagram⟩

    ⟨eg: 1..1 on Class and Teach. 1..1 on Teach and Faculty. meaning?⟩

    * **Q:** For this example, is it one-to-one, many-to-many? What is the equivalent notation using arrows?

    ⟨eg: 0..* on Class and Teach. 0..1 on Teach and Faculty. meaning?⟩

    * again, "*" means unlimited

    * **Q:** For this example, equivalent notation?

    * Comments: don't get confused. It is one-to-many. "0..*" corresponds to one, and "0..1" to many

N-ARY RELATIONSHIPS

- Sometimes we need more than binary relationship

  ⟨ex: Students, TA, Class⟩

  - ⟨All TAs for all students⟩

  - ⟨Each student assigned to a particular TA⟩

      * Arrow in a N-ary relationship: pick one entity from every other set without arrow. Together, these entities must be related to at most one entity with arrow
        - ⟨eg: Arrow to TA. Meaning?⟩

        - Do not put multiple arrows for non-binary relationships. Very confusing. No standard interpretation. (Case tools do not allow anyway)

ROLES

- useful if an entity set participates more than once in a relationship
  - labels on the edge in E/R

⟨eg: Partner relation between students. Coder and Tester⟩

SUBCLASSES

- Similar to class inheritance in OOPL. ISA relationship in E/R
  ⟨eg: Student, ForeignStudent, DomesticStudent⟩

  - Generalization: Subclass $\rightarrow$ Superclass
  - Specialization: Superclass $\rightarrow$ Subclass
  - Subclass inherits all attributes of its superclass

  - Subclass participates in the relationships of its superclass
  - Subclass may participate in its own relationship
    ⟨eg: Student, HonorStudent, HonorClass⟩

- Disjoint specialization vs overlapping specialization
  - Either-or vs multiple specialization
  - single hollow arrow vs multiple hallow arrows

WEAK ENTITY SET

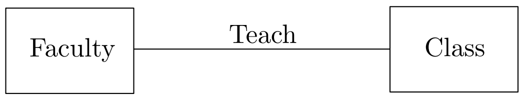
- Entity sets without unique keys
  - Notation: Double rectangle and double diamond in E/R
    ⟨eg: ProjectReport⟩
  - A part of its key comes from one or more entity set it is linked to.
- Discriminator: a set of attributes in W.E.S. that are part of the key.
  - Dashed underline in E/R
- OWNER ENTITY SET: entity set providing a part of the key
- IDENTIFYING RELATIONSHIP: relationship between a weak entity set and owner entity set
- **Q:** Can a weak entity not participate in the identifying relationship?
  - Always double edge between a weak entity and identifying relationship

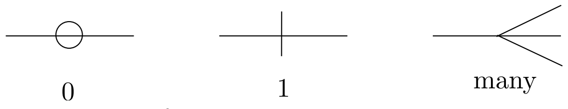
Drawing ER Example

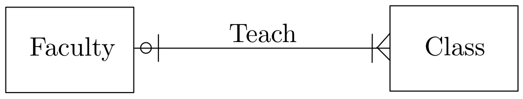
⟨eg. Draw ER diagram for the following example⟩

- Inventory management for chain stores (like Costco, Target, Wallmart, etc.)
- Products are either
  - a “store-brand product” (like Kirkland shoes at Costco) or
  - a “manufacturer-brand product” (like Kleenex Tissue, etc)
- Each product is manufactured by exactly one manufacturer
  - like Sprite by Coke company, etc.
- Each store-brand product is carried by exactly one chain store
  - eg, Kirkland shoes by Costco
- Some manufacturer-brand product product may not be carried by any store

Crow's Foot Notation

- Relationship set is represented just as a line, not a diamond.
  - 

  - Relationship set cannot have its own attribute
- Notation for participation and cardinality constraints
  - 

  - Use a pair of the above symbols to represent the cardinality constraints
- Q: What will be the equivalent ER diagram under the Chen's notation?
  - 

Design Principles

- Often it is not clear what choices to make.
- A general rule of thumb for good design: avoid redundancy
  - Saying the same thing more than once
  - Space waste and potential inconsistency

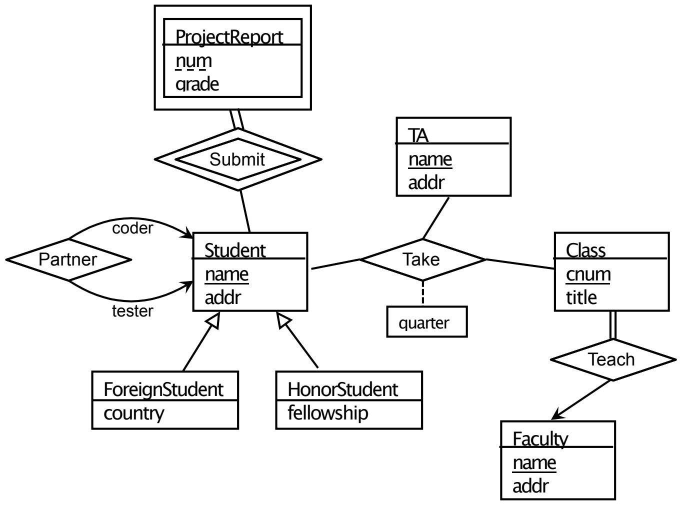
  ⟨eg: Faculty (id, name, address), and class (dept, cnum, title)⟩

  - ⟨eg1: All as attributes of class ES⟩
    1. repeats the faculty name and address for every class. potential inconsistency
    2. What if a faculty does not teach any class?
  - ⟨eg2: Faculty ES, Class ES, Teach RS⟩
- Things to consider for Entity set vs. Attribute
  - Do we need more attributes than keys? eg, Faculty name, address
  - Is it "one-to-one" relationship?
    Separate entities for many-to-one or many-to-many relationship

E/R to Relation

- translation from ER diagram to tables is mostly straightforward
- Database design tools do this automatically from ER diagram

⟨ER example slide⟩



- (STRONG) ENTITY SET: one table with all attributes

⟨eg: Faculty, Class, Student⟩

- RELATIONSHIP SET: one table with keys from the linked ES and its own attributes

⟨eg: Teach⟩

  - **Q:** What is the key for the relations?

⟨eg: Take⟩

  - Rename attributes when names conflict, like TA.name and Student.name

⟨eg: Partner⟩

  - Use role label as attribute names

- WEAK ENTITY SET: one table with its own attributes and keys from owner ES

  ⟨eg, ProjectReport⟩

  - **Q:** What is the key?

  - **Q:** Need to convert Submit to a relation?

    * Separate submit is redundant (already captured by ProjectReport)
    * No need to translate identifying relationship set

- SUBCLASS: two approaches

  1. one table for each subclass with all its attributes plus key from its superclass
  2. one big relation with all attributes with null values for missing attributes

  ⟨eg, Student, ForeignStudent, HonorStudent⟩

  1. Student, ForeignStudent, HonorStudent

  2. Student