Entity-relationship (ER) model

Database Design Phases
1. characterize user needs
2. conceptual design
   - structure of data
   - nature of functionality
     - questions
     - modifications
3. implement in database system
   - logical design
   - physical design

Entity-relationship model
- Goal: Capture semantics of information objects
- Goal: Capture complexity of relationships between objects
  - Used first for database modeling but now expanded use

ER model basics
- An entity is a distinct object in the "real" world
  - person, book, movie character, disease, ...
  - conceptual
- Attributes are basic properties of entities
  - some defs. don’t allow substructure for attributes
    - name = first name, last name
- An entity is described/defined by its attributes
  - entity is tuple (or set) of attributes
  - attribute is function: entity set → domain of attribute values

History
- Developed 1976 by Peter Chen after relational model
- Chen felt relational model not rich enough
  - relational model: everything a (mathematical) relation on collection of domains $D_i$
    - e.g., name from domain of strings
    - Relation subset of $D_1 \times D_2 \times \ldots \times D_k$ (k-ary)
  - ER model differentiate between objects described by attributes and relationships among objects
ER model basics II

- A relationship is a tuple of entities
  - entities are thus related
  - relationship has some meaning
    - (PU store, "cute tiger" baby shirt)
- A relationship set is a set of relationships of the same type
  - "same type" = same component types
    - (stores) X (items for sale)
- A relationship can have its own attributes
  - different from entity attributes
  - descriptive only
    - cannot use to distinguish two tuples of a relationship set
    - (PU store, "cute tiger" baby shirt), "in stock?"

Example

- Entity course with attributes:
  - department, number, semester
- Entity student with attributes:
  - first name, last name, ID number
- Relationship “take” relating:
  - A student to a course

Types

- Entity type:
  - Defined by A₁ x A₂ x ... x Aₖ where A₁, ..., Aₖ are attribute types (for entity with k attributes)
  - Defines kind of object (e.g. student)
  - Set of entities of same type – entity set
- Relationship type:
  - Defined by E₁ x E₂ x ... x Eₘ where E₁, ..., Eₘ are entity types (for relationship between m entities)
  - Defines kind of relationship (e.g. "take")
  - Set of relationships of same type – relationship set
- Then have instances of entity type and relationship type (e.g. (fred, smith, 123456))