Chapter 3: Entity Relationship Model

ER-Model
- conceptual schema design
- high-level data model used to design database
- independent of implementation – can map to several lower level data models (relational, object-oriented, etc)

- main units:
  - entity – basic object of ER model – represents real world thing – rectangle in diagram
  - attribute – properties of an entity – ovals in diagram
  - relationship – defines set of associations among entities – diamond in diagram

- ex: TOY entity – attributes – name, msrp, num_in_stock, etc.
- toy has relationship with manufacture – made-by (or makes)

- Attributes:
  - Simple – not decomposable – ex: toy’s price
  - composite – made of smaller, simple attributes – ex: address = street, city, state; contact = name, email
    - hierarchy of attributes in diagram
  - multivalued attribute – can have more than one value for same entity
    - ex: age_group – if made for more than one specified group
    - ex: location of manufacturer
    - double-lined oval in diagram
  - derived – computed from the values of other attributes
    - ex: age – computed from child’s birthdate
    - ex: # toys ordered computed from other data in the db
  - complex attribute – arbitrary nesting of multi-valued attributes
    - ex: location for manuf – multiple locations, each of which is composite
    - ex: customer may have multiple addresses/ phones
  - value sets of attribute (domain)
    - specifies allowable values for attributes
    - not in ER diagrams
  - null-valued attribute – no applicable value for the attribute for a specific entity
    - ex: ages for customer with no children
    - has three possible interpretations:
      - not applicable
      - unknown
      - missing

- entity type
  - describes the set of entities that have the same attributes
  - set of entity types defines part of database schema
  - key attribute:
    - entity type has a key attribute that defines each instance of the entity type uniquely (underlined in diagram)
    - single attribute – toy_num
    - combination – man_name+phone
    - key constraint – uniqueness on all extensions
- can have multiple keys – man_num and above combination
- can have not keys – weak entity type – more on this later
- entity set (extension) – of entity type = set of instances of entity type

- Relationships
- relationship type R among n entity types E1,…,En defines set of associations among entities of from these types (relationship set)
- instance r of a relationship type R associates n entities (e1,…,en) of types E1,…,En
  - e1,…,en participate in r
- ex: each instance of orders relationship associates one customer with one toy
- note: each relationship does not necessarily associate two unique entities

  (cust1) ---------------- OB1 ----------------------(toy1)
  (cust2) -----------------OB2 ----------------------(toy2)
  (cust3) -----------------OB3 ---------------------/

  - both cust2 and cust3 have ordered toy2
  - degree of this relationship type is 2 – associates two entity types (toy and cust)

- relationships with attributes
  - representing information about the relationship
  - ex: orders – date, quantity, etc.

- relationships as attributes:
  - alternative way to represent relationships
  - put attributes in one or both entities
  - ex: makes ➔ attribute in TOY – MAN_ID

- relationship roles
  - each entity type that participates in a relationship plays a role
  - ex: orders: cust – orderer; toy – thing ordered
  - when same entity participates more than once, role names are required
  - ex: [employee] ------manager------<supervises>--------emp----------[employee]
    - recursive relationship – relationship associates two entities of the same type
    - ex: [person]-------husband------<married>--------wife--------[person]

- Constraints on relationships
  - Cardinality
    - How many relationship instances an entity can participate in
      - 1-N: a single entity can relate to multiple other entities
        - ex: makes – each toy is made by a single manufacturer (1); each manufacturer makes multiple toys (N)
      - 1-1: single entity can relate to one other entity
        - ex: married (above)
        - ex: employee manages department
      - M-N: multiple entities can relate to multiple other entities
        - Ex: orders – each toy can be ordered by multiple customers; each customer can order multiple toys
    - No matter which cardinality, each instance of the relationship type
      - should only associate d entities, where d is the degree of the relationship
- Participation
  - specifies whether the existence of an entity depends on its relationship with another entity
  - total: every entity instance whose entity type participates in a relationship, must participate in a relationship instance
    - ex: every toy must be made by some manufacturer
    - represented by double line in diagram – on side of total part. Entity
  - partial: entity instance can stand alone – not in relationship
    - ex: not all toys must be ordered
    - represented by single line in diagram

- weak entity types – entity with no key attribute
  - must be associated with a strong entity type through a relationship – called its identifying relationship
  - must have total participation constraint – cannot exist without identifying relationship
  - ex: child attribute – no key attribute – must be related to a customer through parent relationship

- Design Guidelines
  - How to decide how to model a conceptual entity
  - Iterative refinement process
  - concept starts as attribute, but changes to relationship because of reference to another entity type
    - ex: put orders in customer entity – but find ref to toy
  - attribute in multiple entities may be refined into separate entity
    - ex: in university db – dept in student, course, prof., etc.
  - entity may be changed to attrib if it only has one attrib and related to one entity
    - ex: univ db – prof entity only has name and only related to dept. – make into attrib of course entity