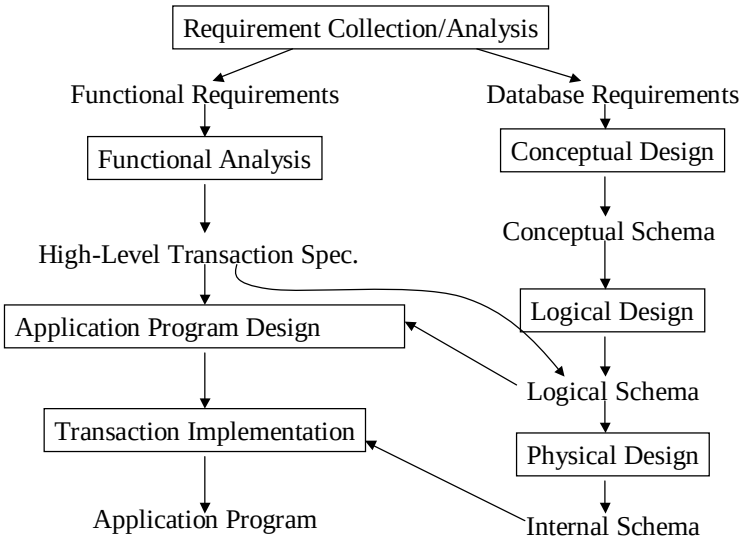
CSC 742
Database Management Systems

Topic #4: Data Modeling

Phases of Database Design



Part A: Entity-Relationship Model

What is ER Model About?

- Structure of the data
  - Entities and relationships between (among) entities
- Constraints
  - Conditions that the entities and relationships must satisfy.
  - Key constraint
  - Domain constraint
  - Structural constraint

ER Concepts

- Entities
- Relationships
- Attributes

Attributes

- Atomic vs. composite
- Single- vs. multivalued
- Stored vs. derived
- Complex Attributes

Null Values

- Need
- Meanings
  - not applicable
  - unknown: missing or questionable existence

Entities

- Entity type (intension): e.g., Employee or Dept
  - A *collection* of entities that have the same attributes
- Entity instance: e.g., Fred or Payroll
- Entity set (extension): e.g., {Fred, Bob, ...}

Keys

- An intension corresponds to all possible extensions
- *Superkey*: a set of attributes that are unique for an entity type (i.e., for all possible extensions)
- *Key*: a minimal superkey—fewer attributes won't be unique
- An entity type may have multiple keys

Relationships

- Relationship types: e.g., works-in
- Relationship instances: e.g., Fred works-in Payroll

Relationship Properties

A relationship type

- associates entity types
- typically binary or ternary
- recursive
- may have attributes

Entity Types

- Participate in relationship types
- Have roles in those relationship types

Cardinality

Cardinality constraints: number of relationship instances in which an entity instance may feature

- 1:1
- 1:N
- M:N

Achtung!

- Don't confuse 1:N with N:1
- Some notations, especially for O-O modeling, write the cardinalities differently

Inferring Cardinalities

- We can construct paths between entity types
- These paths represent relationships composed from series of the existing relationships
- Their cardinalities can be inferred

Fan Traps

- Situations where the inferred, i.e., implied, cardinality is weaker than the actual cardinality

Participation

Participation constraints: whether each entity instance must feature in some relationship instance

- *total*: yes
- *partial*: no

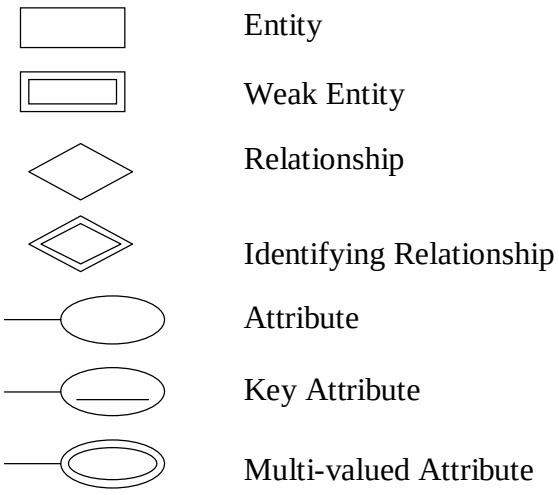
Chasm Traps

- When the composed relationship, i.e., path, has a weaker participation constraint than is actual

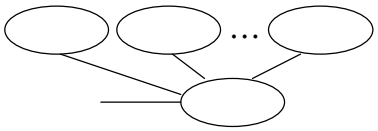
Weak Entity Types

- No key of its own attributes
- Must participate in a total relationship
- Another participant of the relationship becomes the *owner*
- Key = owner's key + partial key

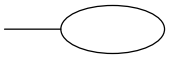
Summary of ER Diagram Notations



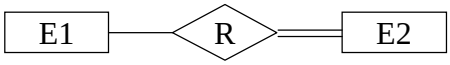
Summary of ER Diagram Notations



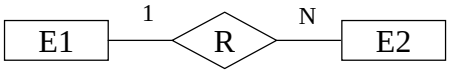
Composite Attribute



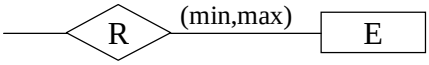
Derived Attribute



Total Participation of E2 in R



Cardinality Ratio 1:N for E1:E2 In R



Structural Constraint (min,max) on Participation of E in R

Part B: Enhanced ER Model

Why Do We Need EER

- ER modeling is sufficient for representing many database schemas for “traditional” database applications.
- Recent applications require additional *semantic data modeling* concepts
  - Class/subclass relationship
  - Type inheritance
  - Specialization and generalization.

Subclass-Superclass

- Subclasses:
  - Further refinement (grouping) of a (super)class
  - Attributes are inherited
  - Class/subclass relationship is different from the relationship in ER modeling.

Specialization

- *Specialization*: The process of defining a set of subclasses of an entity type
  - Top-down conceptual refinement
  - Allows us to
    - Define a set of subclasses of an entity type
    - Establish additional specific attributes with each subclass
    - Establish additional specific relationship types between each subclass and other entity types or other subclasses.

Generalization

- *Generalization*: creating a superclass by combining classes
  - bottom-up conceptual synthesis
  - Can be viewed as the inverse of the specialization process.

Classification

- Predicate-based: when a defining predicate determines the subclass of which a given instance is member
- Attributed-based: when the predicate applies only on an attribute
- User-defined: when the user decide the subclass membership
- Disjoint vs. overlapping
- Total vs. partial

Constraints on Specialization/Generalization

- Disjointness constraint
  - The subclasses of the specialization must be disjoint.
    - Specified by (d)
  - Otherwise, the subclasses may overlap.
    - Specified by (o)

Constraints on Specialization/Generalization

- Completeness constraint
  - Total Specialization
  - Partial Specialization
- Disjointness and completeness constraints are *independent*.
- Superclass identified from generalization is usually *total*.

Rules

- Delete from superclass ⇒ delete from all subclasses
- Insert into predicate-based superclass ⇒ insert where predicate holds
- Insert into total superclass ⇒ insert into a subclass
  - can't reasonably be done unless a predicate is specified

Structure

- *Hierarchy*: single inheritance
- *Lattice*: multiple inheritance
  - Shared subclass
- Attribute inheritance
  - Single inheritance: trivial
  - Multiple inheritance

Union Types

- Category
  - A subclass of the Union of some entity types
  - A category has two or more super-classes
  - Different from generalization.

Aggregation

- Combining objects to form a composite object.
- Three types of aggregations
  - Aggregate attribute values of an object to form the object
  - Represent an aggregation relationship as an ordinary relationship
  - Combine objects that are related by a relationship into a higher-level aggregate object.

Association

- Associate objects from several *independent* classes.
- Not quite aggregation because deleting an entity instance doesn't destroy the instances it is composed of