Mapping E/R to Relational

Textbook Ch. 6.8

Big Picture

• E/R is better for design than relational
• Better semantics, closer to user view
• Relational is better for implementation
  – RDBMS’s are widely available
• Mapping E/R to relational is pretty mechanical
  – Roughly: entities map to entities; relationships show up as foreign keys or new relations; all attributes end up somewhere

Entities

• Strong entities (i.e., having a key)
  – map unchanged
• Weak entities
  – add to the entity the (foreign) key of its owner

As we’ll soon see, additional attributes may get added...

Relationships

• Entities S and T are 1:1 in the relationship
  – Add T’s key as foreign key to S
  – Add any relationship attributes to S
  – If one of the entities is total, use it as S
• Entities S and T are N:1 (S is the N side)
  – Add T’s key as foreign key to S
  – Add any relationship attributes to S

N:M relationships

• Create a new relation
  – contains the keys of both S and T as attributes
  – contains a row for each pair of related S and T entities
  – contains the relationship attributes
• Ternary and higher relationships
  – proceed as for binary N:M relationships: create a new relation with as many foreign keys as there are entities in the relationship, etc.

Attributes

• Simple attributes map unchanged
  – As noted, relationship attributes migrate to an entity (usually the “weaker” or “smaller” one)
• Compound attributes
  – Break into individual items
    • “address” -> “street”, “city”, “state”, “zip”
• Multivalued attributes
  – Create a relation which joins primary key with each occurring value of the attribute