Lecture 17:
  • ER Diagrams (cont)

Announcements:
  • Project Step 1 due
  • PS-3 due Tues, Nov 8
  • HW-5 out, due Thurs, Nov 10

---

ER Basics

![Entity Set vs Entity Diagram]

**Entity Set vs Entity**

  • An entity set is a collection of similar entities
  • An entity is an instance (or member) of an entity set
  • Each entity is uniquely identified and has attribute values
  • Entity sets denoted using rectangular boxes (with attributes as ovals)

... also called Entity Type
ER Basics (cont)

Relationship Set vs Relationship  ... also called Relationship Type
- A relationship set is a collection of similar relationships
- A relationship is an instance (or member) of a relationship set
- A relationship is an association among 2 or more entities
- Relationship sets denoted using diamonds ... note: no directionality!

ER model vs the Relational Model

ER is a different data model than the relational model
- different constructs for modeling schemas and instances

The Relational Model has ...
- tables (relations) with attributes, keys and foreign keys, rows, values

The ER model has ...
- entities with attributes and identifiers (like keys)
- relationships that associate entities (potentially with roles and attributes)

Usually start with ER models, then map them to relational schemas
ER Cardinality Constraints (Example)

Examples: 
- An Employee can have 0 or 1 home Departments \((0..1)\)
- A Department can have 0 to many Employees \((0..*)\)
- A Department must have exactly one Manager \((1..1)\)

Note on notation: using “UML” style versus “crow’s foot”

Check In: How do these differ? Which is the correct one?

Note Terminology: one-to-one, one-to-many, many-to-one, many-to-many
- where “one” implies \(0..1\) or \(1..1\)
- and “many” implies \(0..*\), \(1..*\), etc.
ER Cardinality Constraints (Cont)

Constraints are expressed over entity and relationship sets
• constrain members participation in the corresponding sets

Check In: Do these instances satisfy the given cardinality constraints?

S. Bowers CPSC 321, Fall 2022 7
ER Cardinality Constraints (Cont)

Constraints are expressed over entity and relationship sets

- constrain members participation in the corresponding sets

![Diagram]

Check In: Do these instances satisfy the given cardinality constraints?

Relationship Attributes

Similar to Entity Sets, attributes can be defined for Relationship Sets

- Each instance of the relationship set has a value for the attribute

![Diagram]

Note: In this case, for any particular pair of employee and department entities:

- each such pair can participate in the home relationship at most once
- thus, each pair can have at most one startdate value
- here, an employee has at most one startdate from the 0..1 constraint
Check In: Trying all three locations ... where does the attribute make sense?

Role Names

Role names provide labels to help define entity participation

- Here, manages and managed by define different roles in the relationship
- Help to “read” the relationship (e.g., an employee manages a department)
Role Names

Role names can help with more complex relationships

- Here, roles disambiguate participation of employees in the relationship
- Role are especially needed for these types of “recursive” relationships

Weak Entity Sets

In this model assume we ...

(1) need to record the insurance policies of employees
(2) need to track employee dependents for policies
(3) only need to store the name and date-of-birth of dependents
(4) no longer track their policies or dependents when an employee leaves
(5) dependents of an employee have unique names, but
(6) different employees could have dependents with the same name

Note: A common pattern!
In this case ...

- Employee is the **strong** entity set
- Policy is the **identifying** relationship set ... double lines
- Dependent is the **weak** entity set ... double lines
- Dependent name is a **partial** key ... dashed underline

Note:

- “weak” in that it wouldn’t be in the DB if strong entity were not present
- partial key must be combined with strong entity key to identify dependent