DATABASE SYSTEMS

Dr. Raj Sunderraman

Department of Mathematics and Computer Science
Georgia State University

January 29, 1999

Ch 3. Entity-Relationship Model

Database Design Process:

use a high-level conceptual data model (ER Model).

Figure 3.1

An Example: Company database

- company organized into DEPARTMENTs. each department has unique name and a particular employee who manages the department. Start date for the manager is recorded. Department may have several locations.

- A department controls a number of PROJECTs projects have a unique name, number and a single location

- We store each EMPLOYEEs name, ssno, address, salary, sex and birthdate. An employee is assigned to one department, but may work for several projects (not necessarily controlled by her dept.) We record number of hours/week worked on each project; We also record the immediate supervisor for the employee.

- We want to keep track of DEPENDENTs of employees for health insurance purposes (dependent name, birthdate, relationship to employee)

ER Model Concepts

- Entities and Attributes (Figure 3.3)
- Attribute types:
  Simple vs Composite (Figure 3.4)
  Single-valued vs Multi-valued (e.g. Locations for DEPARTMENT)
  Stored vs Derived (e.g. NumberOfEmployees for DEPARTMENT)
- Null values for attributes: Not applicable, Unknown (Missing; not known if applicable)

- Entity Types, Value Sets, Key Attributes

  An entity type defines a set of entities that have the same attributes. Figure 3.5

  Rectangular box in ER Diagram denotes Entity Types

  Ovals denote Attributes (double-ovals: multi-valued attribute; tree structured ovals: composite attribute)

  Entity Set = Set of all entities of the same type.

  Key Attributes: uniquely identify each entity within an entity set (these are underlined in the ER Diagram)

  Value Sets: or Domains for attributes.

  A: E -> P(V)   A:attribute, E:Entity set, V: Value set
  V = P(V1) x ... x P(Vn) for composite attributes

  A(e) denotes the value of attribute A for entity e

  Figure 3.7: Example of an attribute with specific notation

Figure 3.8 gives an initial design of the COMPANY DATABASE

4 entity types:

DEPARTMENT
EMPLOYEE
PROJECT
DEPENDENT

Not a perfect design! because it captures relationships between entities as attributes (in general not a good idea)

Relationships, Roles, Structural Constraints

- Relationship type: R among n Entity types E1, ..., En
  defines a set of associations among entities from these types.
  each association will be denoted as:
  (e1, ..., en) where ei belongs to Ei, 1 <= i <= n.

  ex. Works_FOR relationship in Figure 3.9

- Degree of relationship = n (usually n = 2, binary relationship)

- Ternary relationship: Figure 3.10

- Relationships as attributes (e.g. Dept -- Empl relationship
  can be viewed as two attributes one in Dept. and the other in
  Empl)

- Role names

  Each entity participating in a relationship has a ROLE.
  e.g. Employee plays the role of worker and Department
  plays the role of employer in the WORKS_FOR relationship type

- Role names are more important in recursive relationships
  ex. Figure 3.11

Structural Constraints on Relationships

2 types:

- Cardinality Ratio Constraint (1-1, 1-N, M-N)
- Participation Constraint

    * total participation (existence dependency)
    * partial participation

Figure 3.12

In ER Diagram total participation is denoted by double line
and partial participation by single line
Cardinality ratios are mentioned as labels of edges

Attributes of relationships:

Hours attribute for WORKS_ON relationship

if relationship is 1-N or 1-1, these attributes can be be migrated to the entity sets involved in the relationship.

1-N: migrate to N side

1-1: migrate to either side

Weak Entity Types:

Entity Types that do not have key attributes.

Such entities are identified by being related to other entity sets called identifying owner. This relationship is called identifying relationship.

Partial key: attributes that uniquely identify entities within the identifying relationship.

A weak entity type always has TOTAL participation.

Ex. DEPENDENT is a weak entity type.

Notation for ER Diagrams:

Fig 3.14, 3.15

(Min,Max) notation encapsulates both types of structural constraints.
This is more general than previous notation.

Min >=1 implies TOTAL participation
Min = 0 implies PARTIAL participation

Relationship types with degree ¿ 2

Figure 3.16 (a), (b) represent slightly different information.

Figure 3.16 (c) in situations where only binary relationships are allowed.

Figure 3.17

Figure 3.18