The Entity-Relationship Model

Chapter 2

Overview of Database Design

- Conceptual design: (ER Model is used at this stage.)
  - What are the entities and relationships in the enterprise?
  - What information about these entities and relationships should we store in the database?
  - What are the integrity constraints or business rules that hold?
  - A database ‘schema’ in the ER Model can be represented pictorially (ER diagrams).
  - Can map an ER diagram into a relational schema.

ER Model Basics

- **Entity**: Real-world object distinguishable from other objects. An entity is described (in DB) using a set of attributes.
- **Entity Set**: A collection of similar entities. E.g., all employees.
  - All entities in an entity set have the same set of attributes. (Until we consider ISA hierarchies, anyway!)
  - Each entity set has a key.
  - Each attribute has a domain.

ER Model Basics (Contd.)

- **Relationship**: Association among two or more entities. E.g., Atishoo works in Pharmacy department.
- **Relationship Set**: Collection of similar relationships.
  - An n:m relationship set R relates n entity sets E1 ... En each relationship in R involves entities e1 ∈ E1, ..., en ∈ En
  - Same entity set could participate in different relationship sets, or in different “roles” in same set.

Key Constraints

- Consider Worked_In:
  - An employee can work in many departments, a dept can have many employees.
  - In contrast, each dept has at most one manager, according to the key constraint on Manages.

Participation Constraints

- Does every department have a manager?
  - If so, this is a participation constraint: the participation of Departments in Manages is said to be total (e.g., partial).
  - Every did value in Departments table must appear in a row of the Manages table (with a non-null sor value)

Database Management Systems, R. Ramakrishnan and J. Gehrke
**Entity vs. Relationship**
- First ER diagram OK if a manager gets a separate discrete binary budget for each dept.
- What if a manager gets a discretionary budget that covers all managed depts?
  - Redundant budget, which is tied to each dept managed by the manager.
  - Misleading: suggest budget tied to managed dept.

**Binary vs. Ternary Relationships**
- If each policy is owned by just 1 employee:
  - Key constraint on Policies would mean policy can only cover 1 dependent!
- What are the additional constraints in the 2nd diagram?

**Binary vs. Ternary Relationships (Contd.)**
- Previous example illustrated a case when two binary relationships were better than one ternary relationship.
- An example in the other direction: a ternary relation Contracts relates entity sets Parts, Departments and Suppliers, and has descriptive attribute qty. No combination of binary relationships is an adequate substitute:
  - S "can-supply" P, D "needs" P, and D "deals-with" S does not imply that D has agreed to buy P from S.
  - How do we record qty?

**Summary of Conceptual Design**
- Conceptual design follows requirements analysis, yields a high-level description of data to be stored.
- ER model popular for conceptual design
  - Constructs are expressive, close to the way people think about their applications.
  - Basic constructs: entities, relationships, and attributes (of entities and relationships).
  - Some additional constructs: weak entities, ISA hierarchies, and aggregation.
- Note: There are many variations on ER model.

**Summary of ER (Contd.)**
- Several kinds of integrity constraints can be expressed in the ER model: key constraints, participation constraints, and overlap/covering constraints for ISA hierarchies. Some foreign key constraints are also implicit in the definition of a relationship set.
  - Some constraints (notably functional dependencies) cannot be expressed in the ER model.
  - Constraints play an important role in determining the best database design for an enterprise.

**Summary of ER (Contd.)**
- ER design is subjective. There are often many ways to model a given scenario! Analyzing alternatives can be tricky, especially for large enterprises.
- Common choices include:
  - Entity vs. attribute, entity vs. relationship, binary or n-ary relationship, whether or not to use ISA hierarchies, and whether or not to use aggregation.
- Ensuring good database design: resulting relational schema should be analyzed and refined further. FD information and normalization techniques are especially useful.