Using High-Level Conceptual Data Models for Database Design

- Requirements collection and analysis
  - Database designers interview prospective database users to understand and document data requirements
  - Result: data requirements
  - Functional requirements of the application

Using High-Level Conceptual Data Models (cont’d.)

- Conceptual schema
  - Conceptual design
  - Description of data requirements
  - Includes detailed descriptions of the entity types, relationships, and constraints
  - Transformed from high-level data model into implementation data model
Entity-Relationship Model (ER Model)

- ER Model is a popular high-level conceptual data model
  - It is also frequently used for the conceptual design of database applications
  - Many database design tools employ its concepts
- ER diagrams are diagrammatic notations associated with ER models

Entity-Relationship Model (ER Model)

- An entity is a thing in the real world with an independent existence
- Attributes of an entity describe its properties
  - Composite versus simple (atomic) attributes
  - Single-valued versus multivalued attributes
  - Stored versus derived attributes
  - NULL values
  - Complex attributes
- Relationships
Relationships

- An association b/w at least two entities
  - Descriptive attributes
  - Instance

Ex. COMPANY Data Requirements

- The company is organized into departments. Each dept. has a unique name, unique number and a particular employee who manages the dept. We keep track of the start date when that employee began managing the dept. A dept. may have several locations.
- A dept. controls a number of projects, each of which has a unique name, unique number, and a single location.
- We store each employee’s name, ssn, address, salary, gender, and birth date. An employee is assigned to one dept., but may work on several projects, which are not necessarily controlled by the same dept. We keep track of the current number of hours per week that an employee works on each project. We also keep track of the direct supervisor of each employee (who is another employee).
- We want to keep track of the dependents of each employee for insurance purposes. We keep each dependent’s first name, gender, birth date, and relationship to employee.
ER Diagrams, Conventions, & Design

Key Attributes
- An entity type is represented in ER diagrams as a rectangle box
- Attribute names are enclosed in ovals
- Key attributes are shown with underlines

Initial Conceptual Design of the COMPANY Database
Entity-Relationship Model (ER Model)

- A relationship type $R$ among $n$ entity types $E_1, ..., E_n$ defined a relationship set
  - Each individual instance is a relationship instance
  - Displayed as diamond-shaped boxes in ER diagrams
    - The degree: number of participating entities
    - Cardinality ratios for binary relationship: 1:N, M:N, 1:1

![Employee](n) Works for [1] Department

Entity-Relationship Model (ER Model)

- Ternary Relationship Set
  ![Ternary Relationship Diagram]

Entity-Relationship Model (ER Model)

- Roles Relationship Set
  - Roles indicator
  ![Roles Relationship Diagram]
• Participation constraints
  – Total: every employee must work for a department (employee-department-work_for)
  – Partial: not every employee manages a department (employee-department-manage)

• Weak entity types
  – Cannot be identified without an owner entity
  – Has a partial key to identify a record once the owner’s key is given
  – Weak entity types can sometimes be represented as complex attributes (multi-valued, composite)
Attributes of Relationship Types

- Attributes of 1:1 or 1:N relationship types can be migrated to one entity type
- For a 1:N relationship type
  - Relationship attribute can be migrated only to entity type on N-side of relationship
- For M:N relationship types
  - Some attributes may be determined by combination of participating entities
  - Must be specified as relationship attributes

ER Modeling

- Another way of specifying cardinality and participation
  - (min, max)

ER Diagrams
Class Example: ER diagram for University Database

- (a) The university keeps track of each student’s name, student number, social security number, current address and phone, permanent address and phone, birthdate, sex, class (freshman, sophomore, …, graduate), major department, minor department (if any), and degree program (B.A., B.S., …, Ph.D.). Some user applications need to refer to the city, state, and zip of the student’s permanent address, and to the student’s last name. Both social security number and student number have unique values for each student.
- (b) Each department is described by a name, department code, office number, office phone, and college. Both name and code have unique values for each department.
- (c) Each course has a course name, description, course number, number of semester hours, level, and offering department. The value of course number is unique for each course.
- (d) Each section has an instructor, semester/year, course, and section number. The section number distinguishes different sections of the same course that are taught during the same semester/year. Its values are 1, 2, 3, … up to the number of sections taught during each semester.
- (e) A grade report has a student, section, letter grade, and numeric grade (0, 1, 2, 3, 4 for F, D, C, B, A, respectively).