Overview of Database Design

• Conceptual Design
  • Use entity-relationship (aka ER) model represented pictorially as ER diagrams
  • Map ER model to relational schema
• Questions to ask yourself
  • 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?
ER Model Basics: Entities

- **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: Relationships

- **Relationship**: Association among two or more entities.

- **Relationship Set**: Collection of similar relationships.
  - An n-ary relationship set $R$ relates $n$ entity sets $E_1, ..., E_n$; each relationship in $R$ involves entities $e_1 \in E_1, ..., e_n \in E_n$
  - Same entity set could participate in different relationship sets, or in different “roles” in same set.
Cardinality Ratios of Relationships

- Consider binary relationships, i.e., between two entity sets
- Alternate notation: 1:1, 1:M, M:1, M:N
Internet Book Store Example

- Catalog of books
  - ISBN, title, author, qty_in_stock, price, year_published

- Customers
  - CID, Name, address

- Orders
  - ISBN, CID, cardnum, qty, order_date, ship_date
Key Constraints

- Consider Works_In: An employee can work in many depts; a dept can have many employees.
- In contrast, each dept has at most one manager, according to the **key constraint** on Manages.
- Denoted by an arrow: given a dept entity we can uniquely identify the manages relationship in which it appears
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 (vs. partial). Denoted by thick/double line

- Meaning that every Departments entity must appear in an instance of the Manages relationship
Weak Entities

- A weak entity can be identified uniquely only by considering the primary key of another (owner) entity.
- Owner entity set and weak entity set must participate in a one-to-many relationship set (one owner, many weak entities).
- Weak entity set must have total participation in this identifying relationship set.
- Denoted by a box with double or thick lines
IS-A Hierarchies

- If we declare A **ISA** B, every A entity is also considered to be a B entity.

- **Overlap constraints**: Can Joe be an Hourly_Emps as well as a Contract_Emps entity? *(Allowed/disallowed)*

- **Covering constraints**: Does every Employees entity also have to be an Hourly_Emps or a Contract_Emps entity? *(Yes/no)*

- Reasons for using ISA:
  - To add descriptive attributes specific to a subclass.
  - To identify entities that participate in a relationship.
Aggregation

- Used when we have to model a relationship involving (entity sets and) a relationship set.
  - **Aggregation** allows us to treat a relationship set as an entity set for purposes of participation in (other) relationships.

**Aggregation vs. ternary relationship:**
- Monitors is a distinct relationship, with a descriptive attribute.
- Also, can say that each sponsorship is monitored by at most one employee.
Design Choices

- Should a concept be modeled as an entity or an attribute?
- Should a concept be modeled as an entity or a relationship?
- Identifying relationships: Binary or ternary? Aggregation?
- How much semantics to capture in the form of constraints?
Entity vs. Attribute

- Depends upon how we want to use the address information, and the semantics of the data:
  - If we have several addresses per employee, *address* must be an entity (since attributes cannot be set-valued).
  - If the structure (city, street, etc.) is important, e.g., we want to retrieve employees in a given city, *address* must be modeled as an entity (since attribute values are atomic).
Entity vs. Attribute (Contd.)

- Works_In4 does not allow an employee to work in a department for two or more periods.

- Similar to the problem of wanting to record several addresses for an employee: We want to record several values of the descriptive attributes for each instance of this relationship. Accomplished by introducing new entity set, Duration.
Entity vs. Relationship

- ER diagram OK if a manager gets a separate discretionary budget for each dept.
- What if a manager gets a discretionary budget that covers all managed depts?
  - Redundancy: $dbudget$ stored for each dept managed by manager.
  - Misleading: Suggests $dbudget$ associated with department-mgr combination.
Binary vs. Ternary Relationships

- If each policy is owned by just 1 employee, and each dependent is tied to the covering policy, first diagram is inaccurate.

Bad design

Better design
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 a large enterprise. 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.