The Entity-Relationship (ER) Model
(Skim Chapter 2)
Overview of Database Design

❖ **Conceptual design**: What are the *entities* and *relationships*?
  - What are the attributes of these?
  - What are the *integrity constraints* or *business rules*?
**Entity**: An object distinguishable from other objects with 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.
- Each entity set has a **key**.
- Each attribute has a **domain**.
**ER Model Basics**

- **Relationship:** Association among two or more entities. e.g., David works in the Math department.

- **Relationship Set:** Collection of similar relationships.
  - An *n*-ary relationship set, $R$, relates $n$ entity sets $E_1 \ldots E_n$; each relationship in $R$ involves entities.
  - Same entity set could participate in different relationship sets, or in different “roles” in same set.
ISA (‘is a’) Hierarchies

- It is often useful to subdivide entities into classes, like in an OOL
- 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.
Conceptual Design

❖ **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?

❖ **Constraints in the ER Model:**
  - A lot of data semantics can (and should) be captured.
  - But some constraints not captured in ER diagrams.
**Entity vs. Attribute**

❖ Should *address* be an attribute of Employees or an entity (connected to Employees by a relationship)?

❖ Depends upon the use we want to make of address information, and the semantics of the data:

➢ If we have several addresses per employee, *address* must be an entity since attributes cannot themselves be multivalued.

➢ 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).
First is 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.