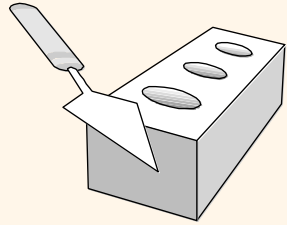
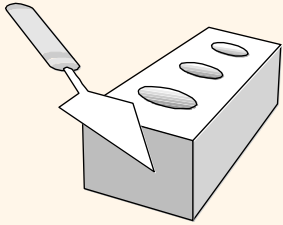


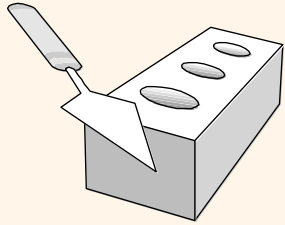
The Entity-Relationship Model

DataBase Modelling



Database Design Process

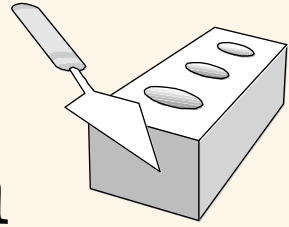
- *Requirement collection and analysis*
  - DB requirements
- *Conceptual DB design using a high-level model*
  - Data to store and relationship between them
- *Logical DB design (data model mapping)*
  - Columns in each table (if relational model)
- *Physical DB design*
  - Internal data structures



Disclamer

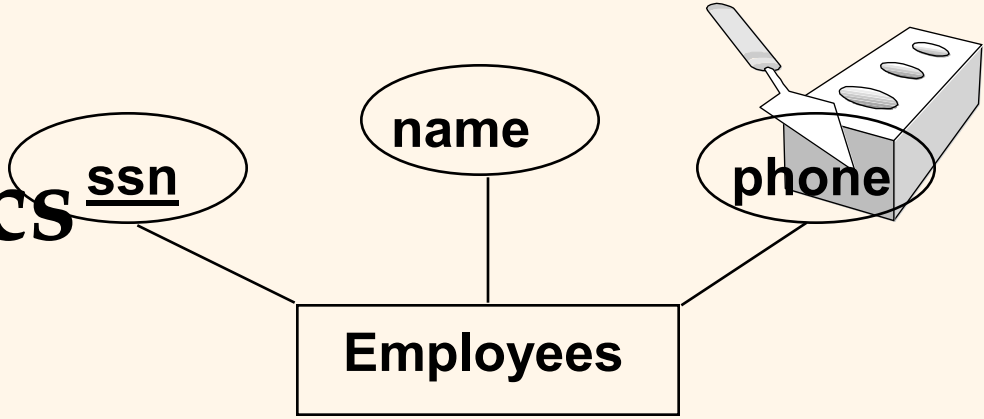
- ER design is *subjective*. There are often many ways to model a given scenario! Analyzing alternatives can be tricky, especially for a large enterprise.

Overview of Database Design



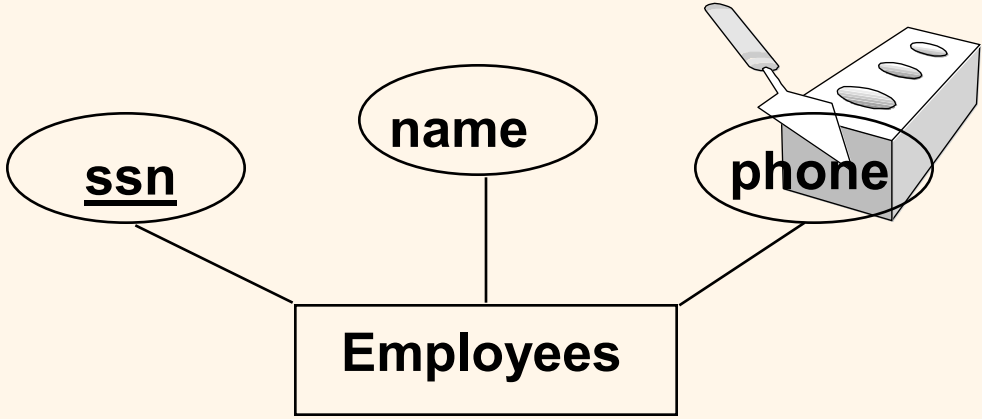
- *Conceptual design*: *(ER Model is used at this stage.)*
  - What are the *entities* and *relationships* in the enterprise?
  - What information about these entities and relationships should we store in the database?
  - What are the *integrity constraints* or *business rules* that hold?
  - A database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

ER Model Basics



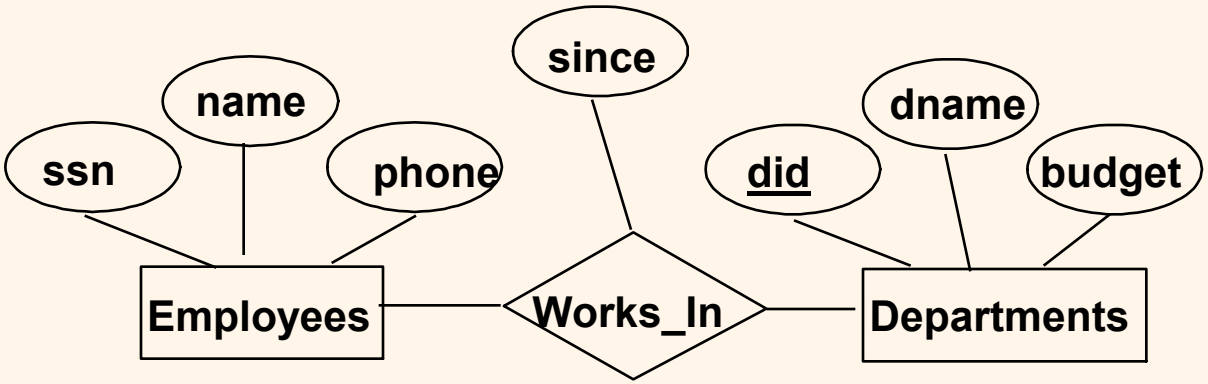
- *Entity:* Real-world object distinguishable from other objects. An entity is described (in DB) using a set of *attributes*.
- *Entity Set*: A collection of similar entities. E.g., all employees.
  - All entities in an entity set have the same set of attributes. (Until we consider ISA hierarchies, anyway!)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

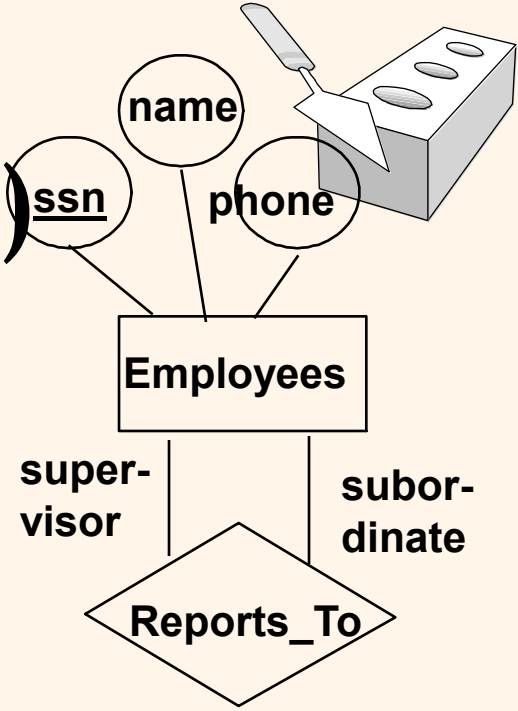
ER Model Basics



- *Key and key attributes:*
  - Key: a unique value for an entity
  - Key attributes: a group of one or more attributes that uniquely identify an entity in the entity set
- *Super key, candidate key, and primary key*
  - Super key: a set of attributes that allows to identify and entity uniquely in the entity set
  - Candidate key: minimal super key
    - There can be many candidate keys
  - Primary key: a candidate key chosen by the designer
    - Denoted by underlining in ER attributes

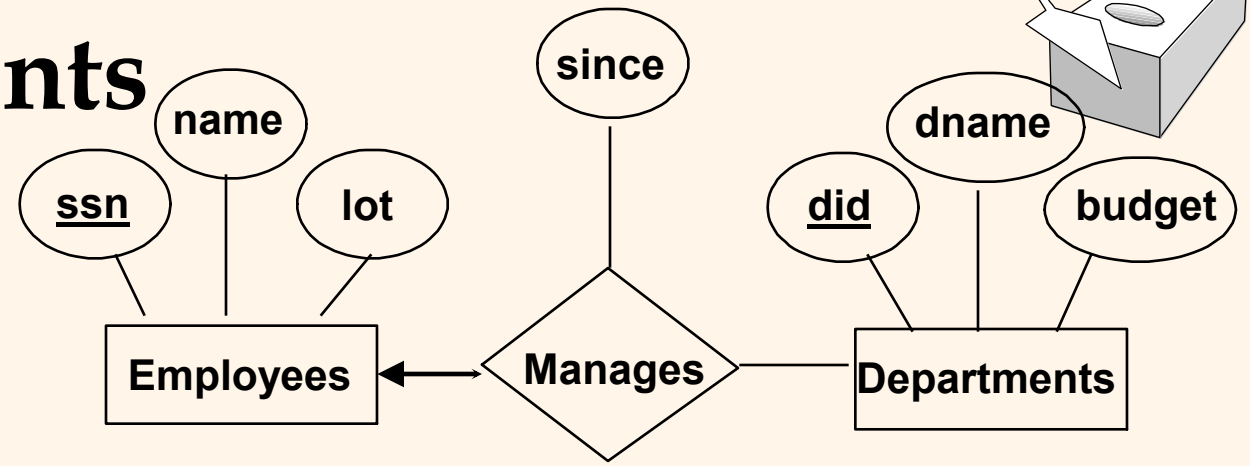
ER Model Basics (Contd.)



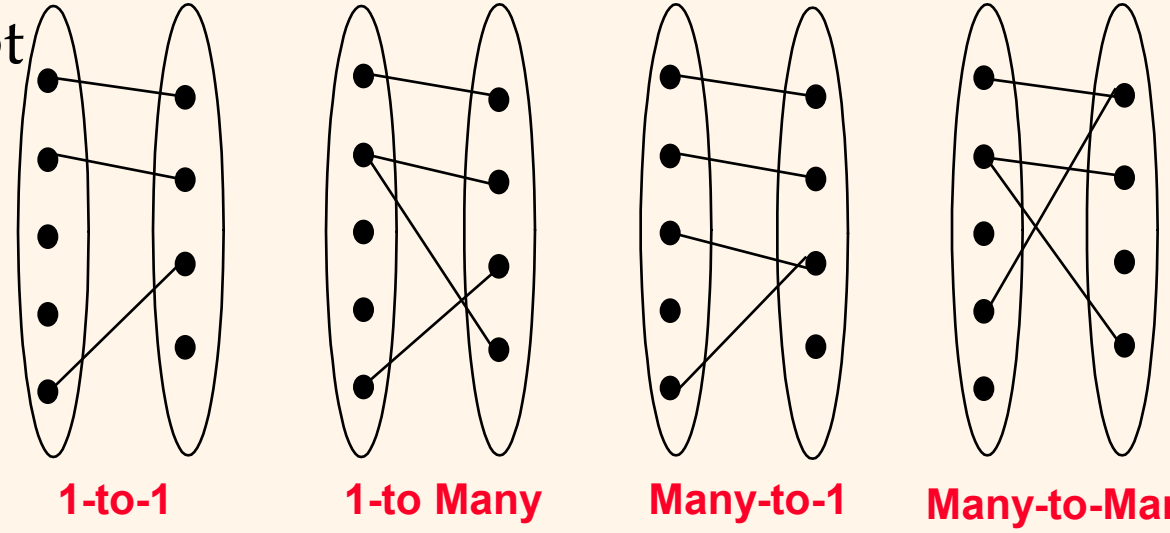


- *Relationship*: Association among two or more entities. e.g., Jack works in Pharmacy department.
- *Relationship Set*: Collection of similar relationships.
  - An n-ary relationship set R relates n entity sets E1 ... En; each relationship in R involves entities e1 in E1, ..., en in En
    - Same entity set could participate in different relationship sets, or in different "roles" in same set.

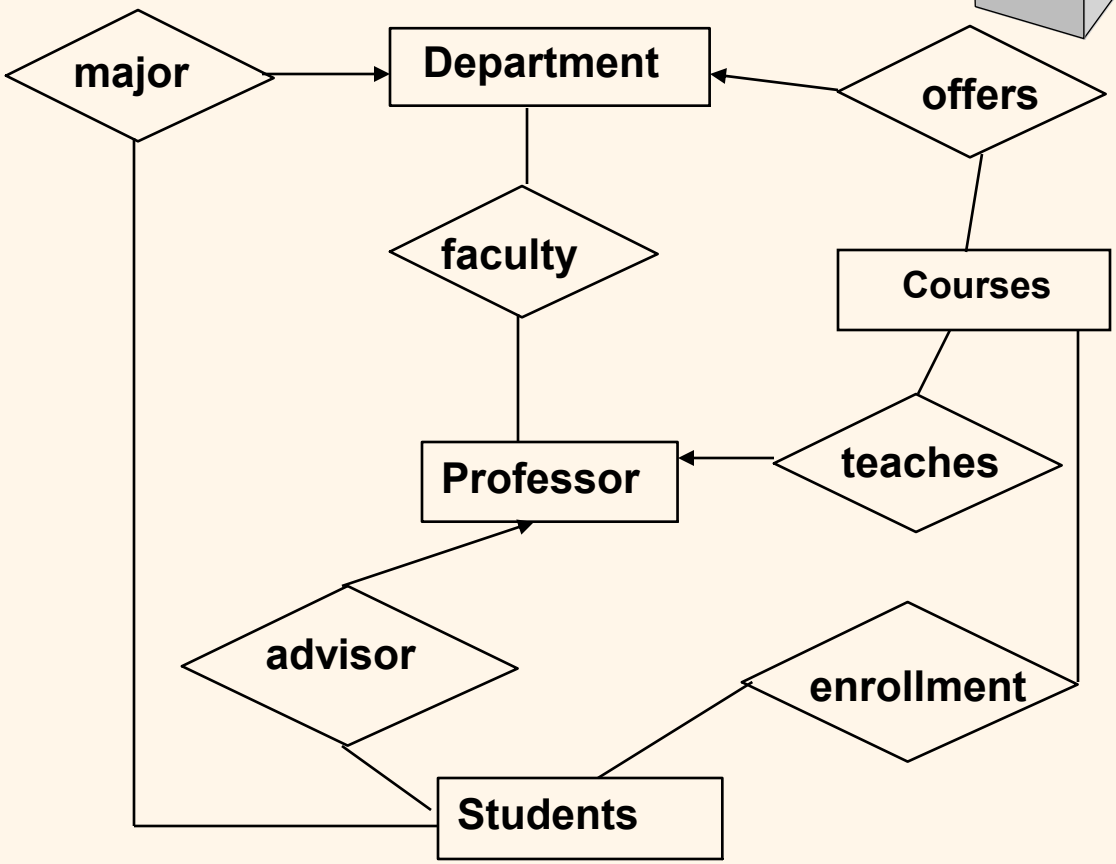
Key Constraints



- Consider Works_In (previous slide): 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.

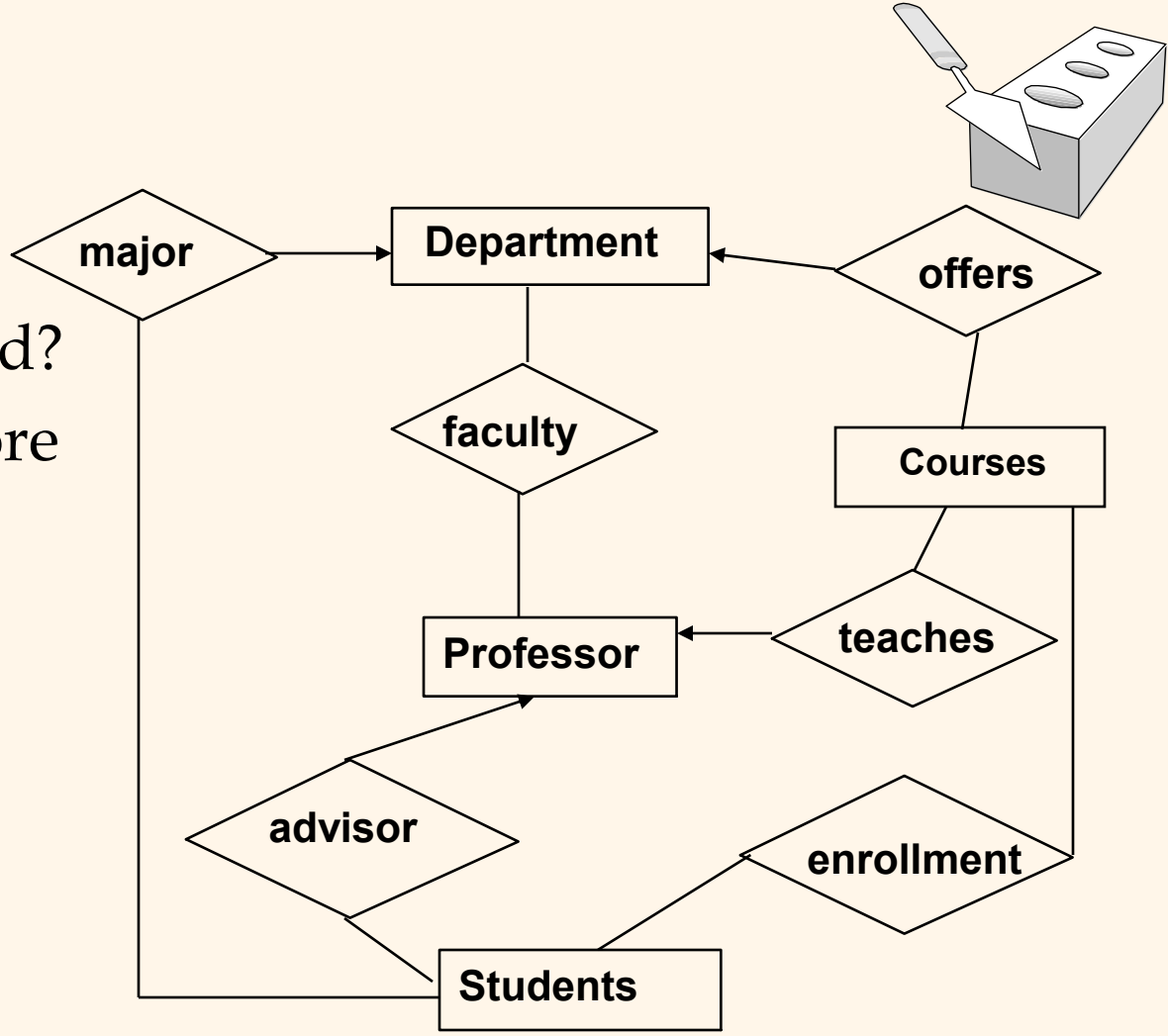


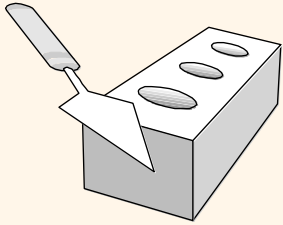
Example ER


major
Department
offers
faculty
Courses
Professor
teaches
advisor
enrollment
Students

Exercises

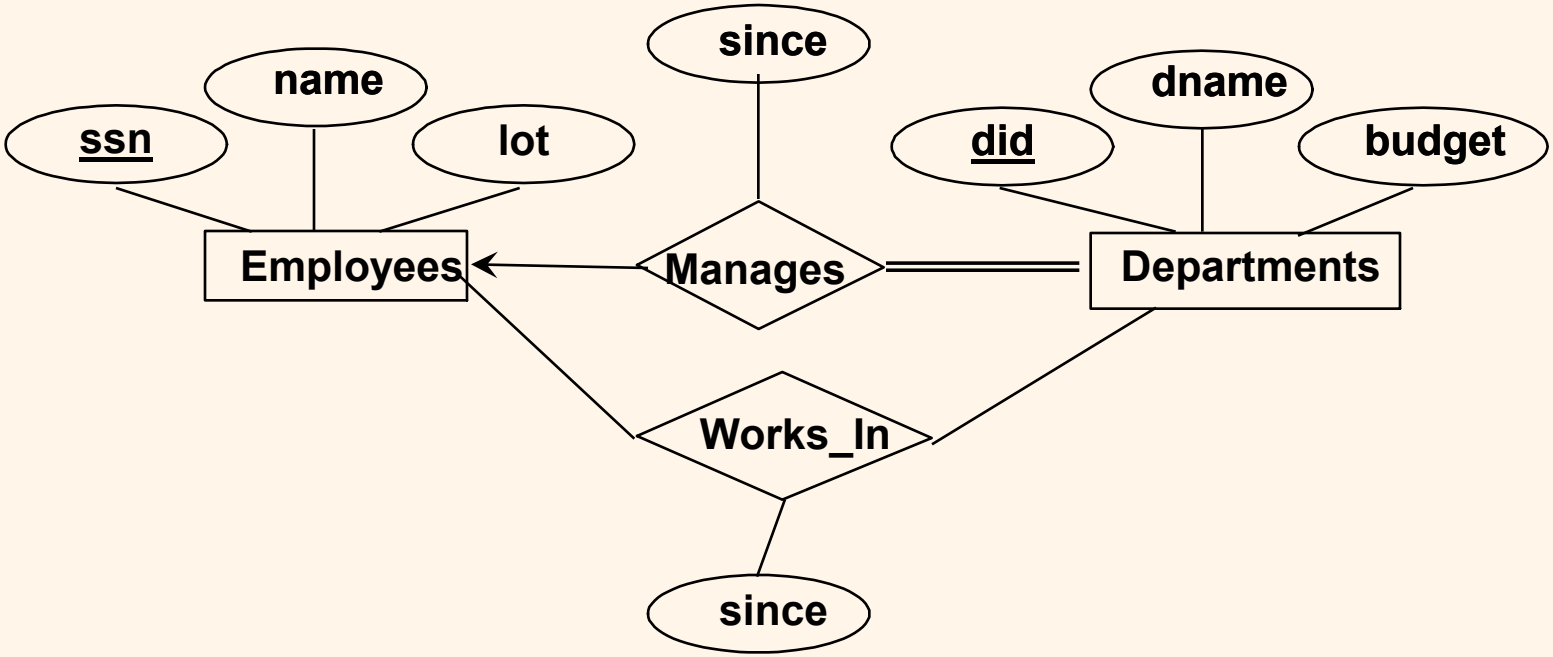
- Is double major allowed?
- Can a student have more than 1 advisor?
- Is joint appointment of faculty possible?
- Can two profs share to teach the same course?
- Can a professor teach more than one course?
- Can a professor stay without affiliated with a department?

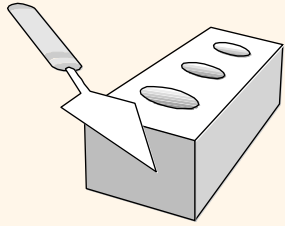




Participation Constraints

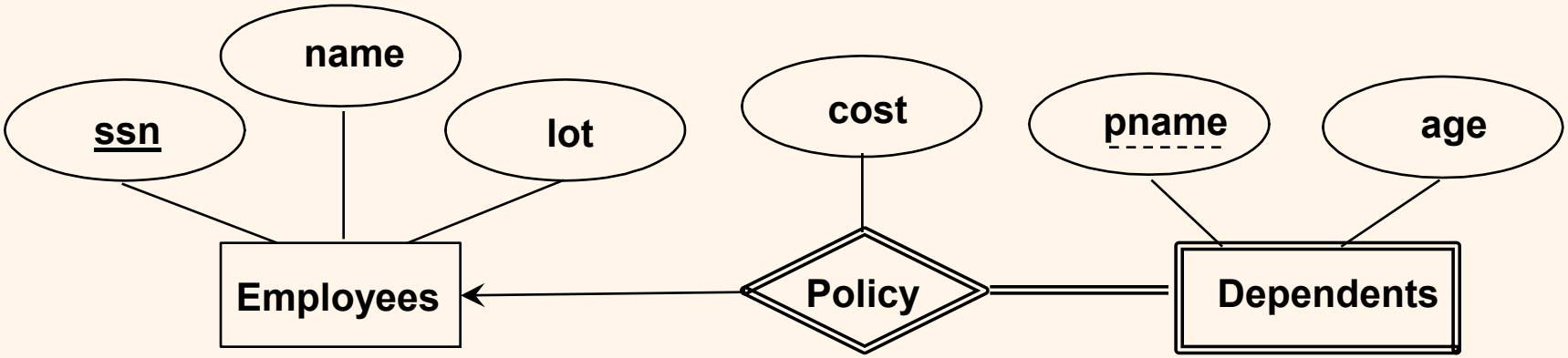
- Does every department have a manager?
  - If so, this is a *participation constraint*: the participation of Departments in Manages is said to be *total* (vs. *partial*).
    - Every Departments entity must appear in an instance of the Manages relationship.



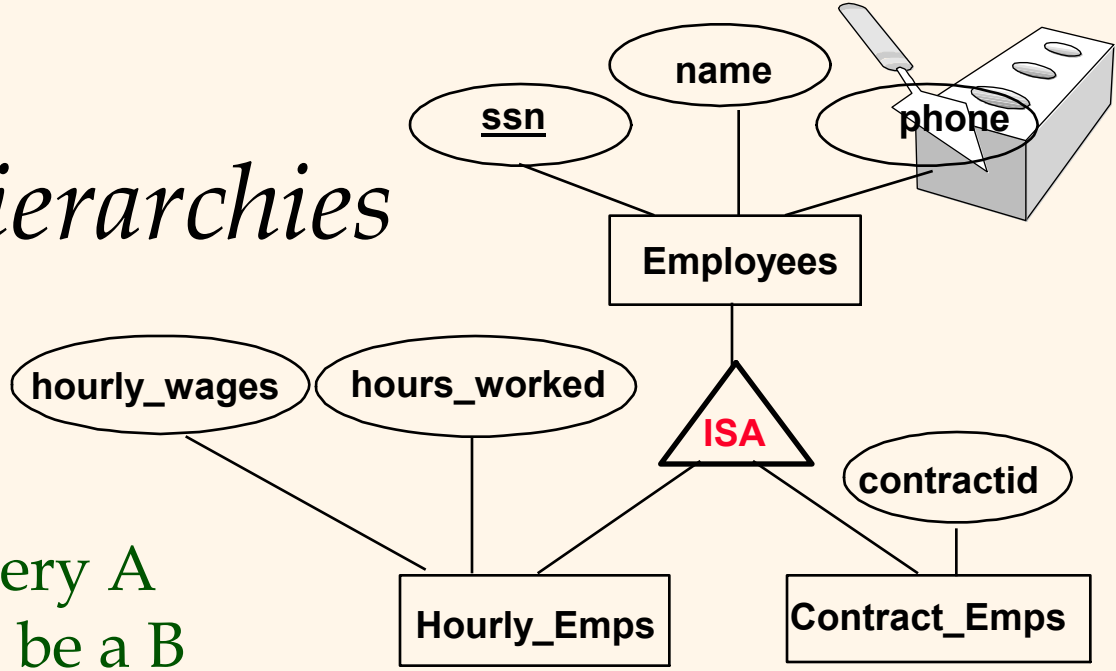


Weak Entities

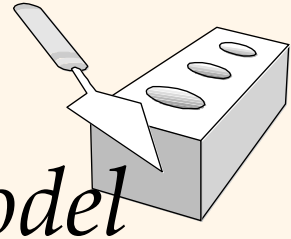
- A *weak entity* can be identified uniquely only by considering the primary key of another (*owner*) entity.
  - Owner entity set and weak entity set must participate in a one-to-many relationship set (one owner, many weak entities).
  - Weak entity set must have total participation in this *identifying* relationship set.



ISA (`is a') Hierarchies

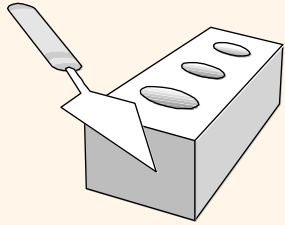


- As in C++, attributes are inherited.
- If we declare A **ISA** B, every A entity is also considered to be a B entity.
  - *Overlap constraints*: Can Joe be an Hourly_Emps as well as a Contract_Emps entity? (*default: disallowed; A overlaps B*)
  - *Covering constraints*: Does every Employees entity also have to be an Hourly_Emps or a Contract_Emps entity? *(default: no; A AND B COVER C)*



Conceptual Design Using the ER Model

- 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?
- Constraints in the ER Model:
  - A lot of data semantics can (and should) be captured.
  - But some constraints cannot be captured in ER diagrams.

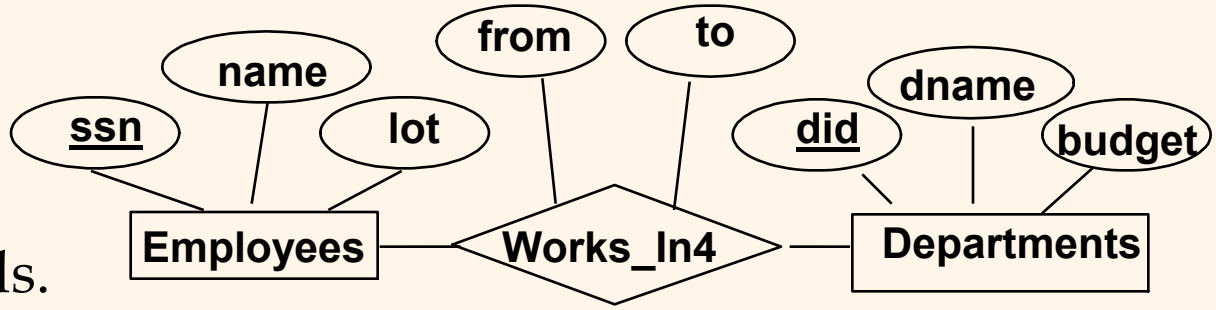


Entity vs. Attribute

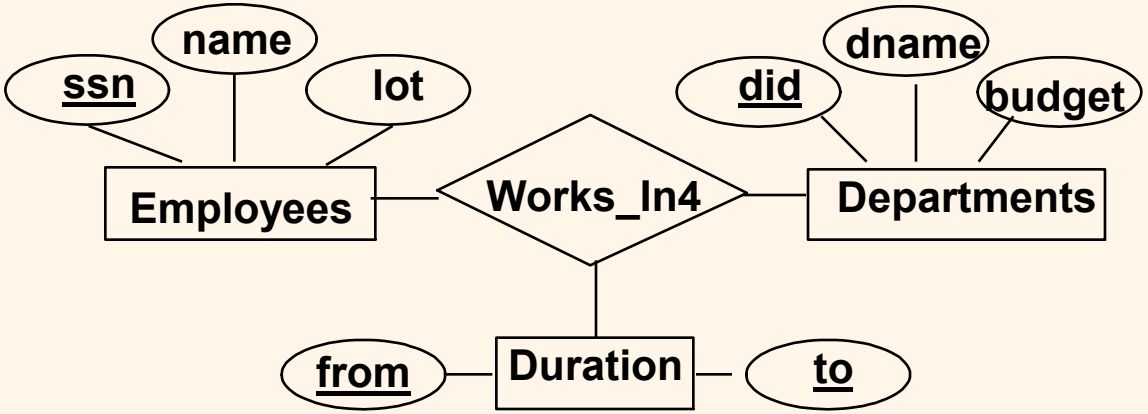
- Should *address* be an attribute of Employees or an entity (connected to Employees by a relationship)?
- Depends upon the use we want to make of address information, and the semantics of the data:
  - If we have several addresses per employee, *address* must be an entity (since attributes cannot be 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.

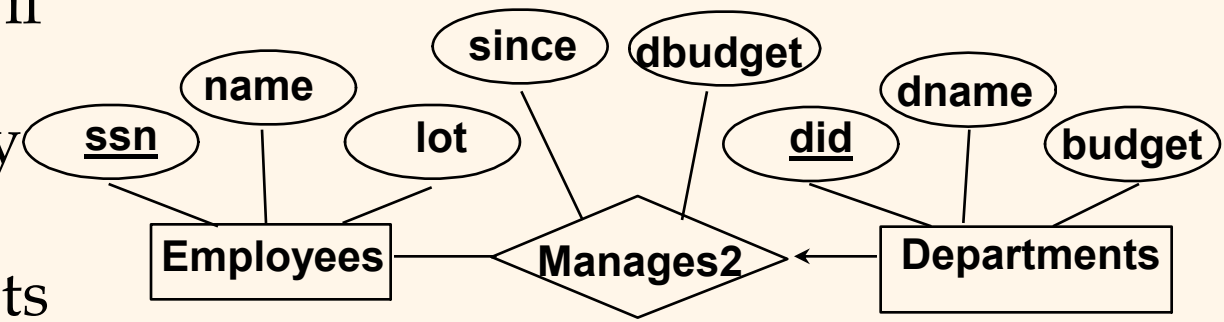


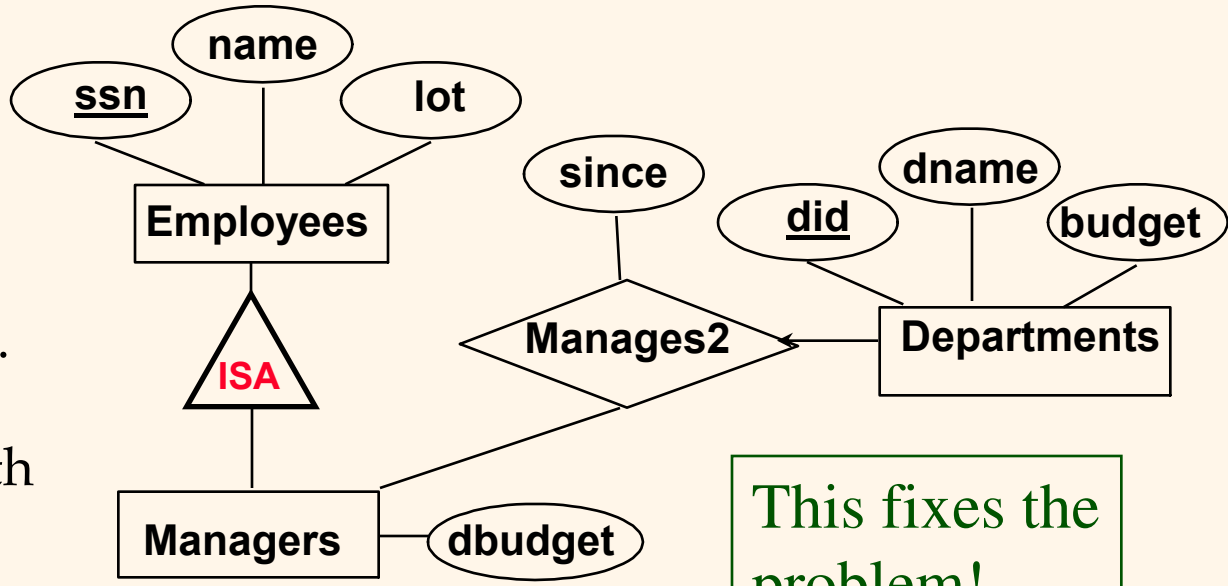
- Similar to the problem of wanting to record several addresses for an employee: We want to record *several values of the descriptive attributes for each instance of this relationship.* Accomplished by introducing new entity set, Duration.



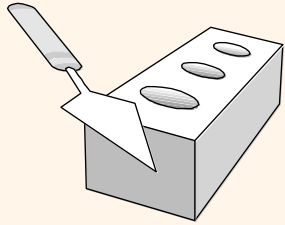
Entity vs. Relationship

- First 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.





This fixes the problem!

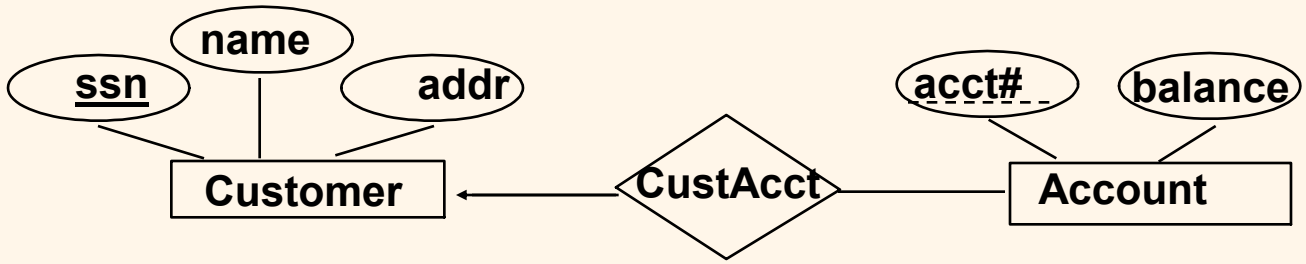


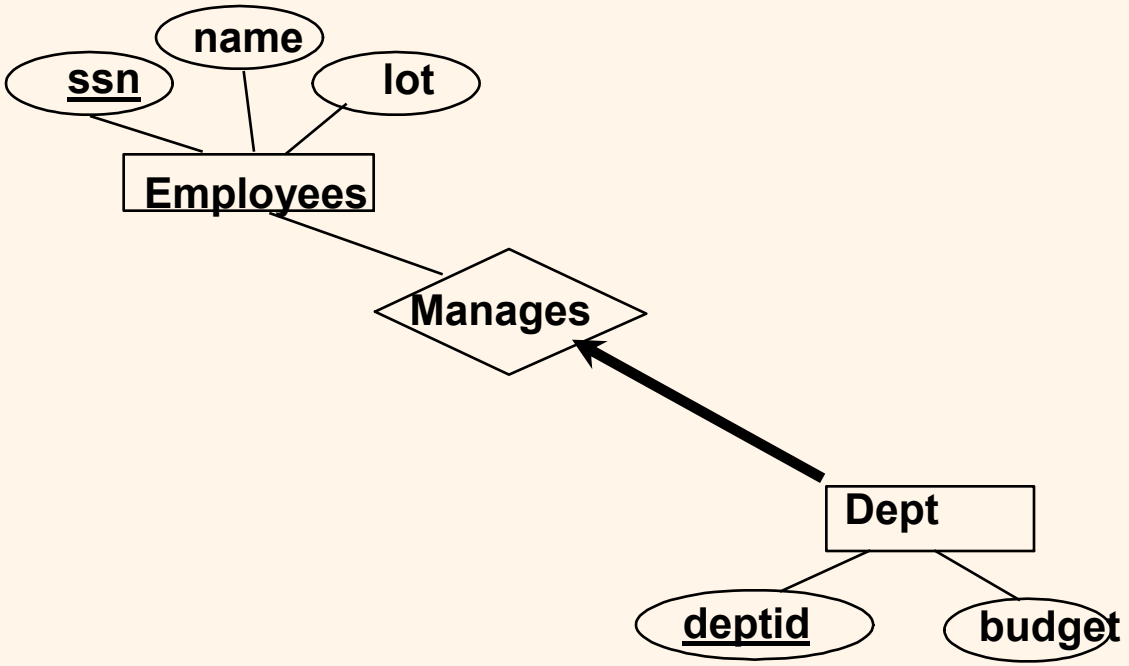
Summary of Conceptual Design

- *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.

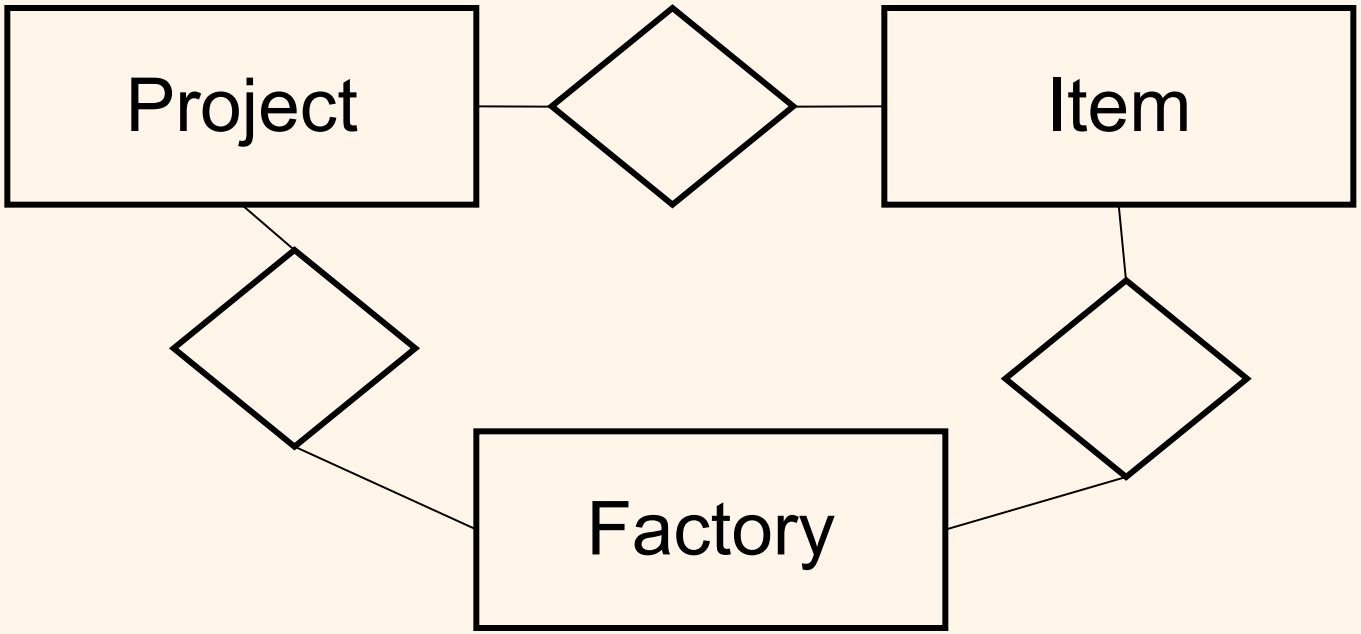
Exercise

- What can you say about policy of the bank from the ER diagram?
- What can you say about the policy of the company?



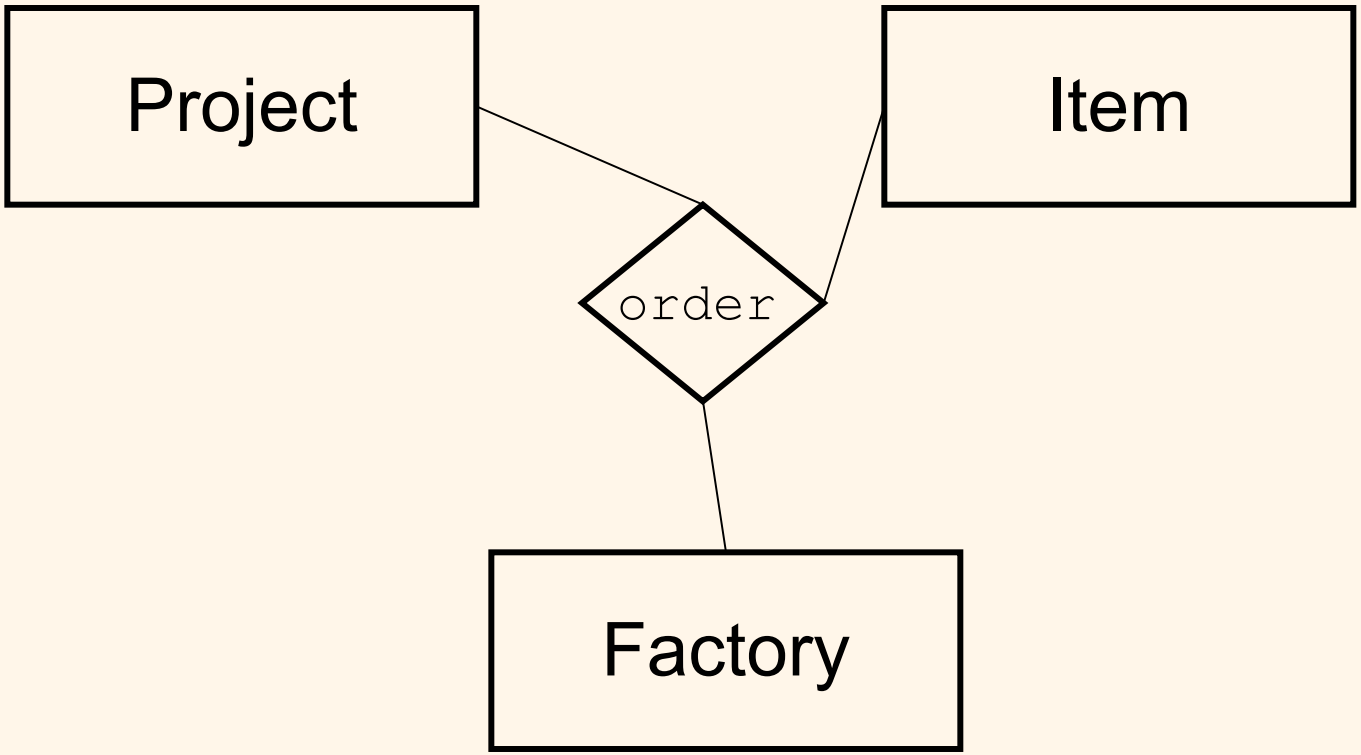


Exercise

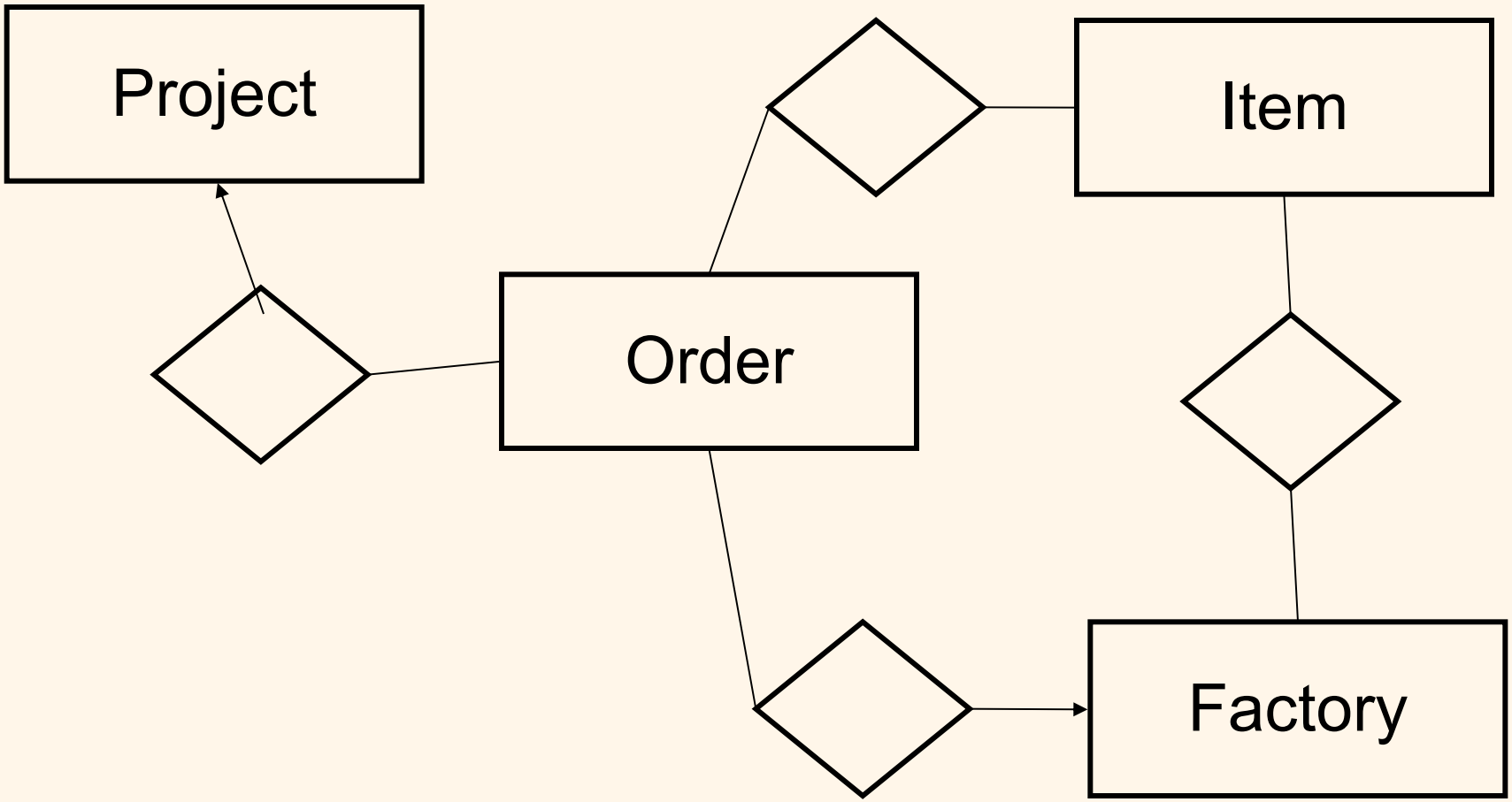


What has been purchase, from whom, which factory produced the item

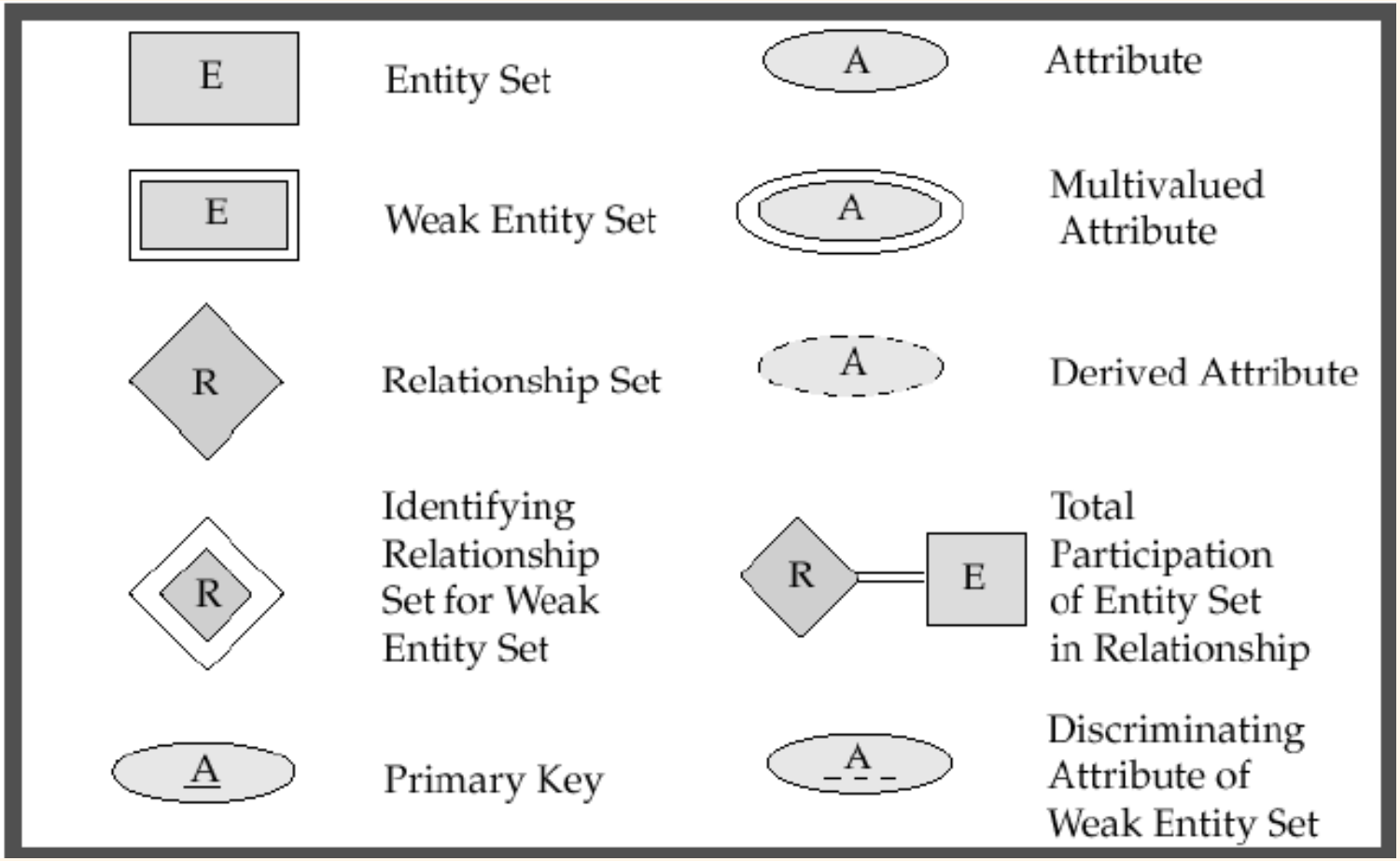
Exercise



Exercise

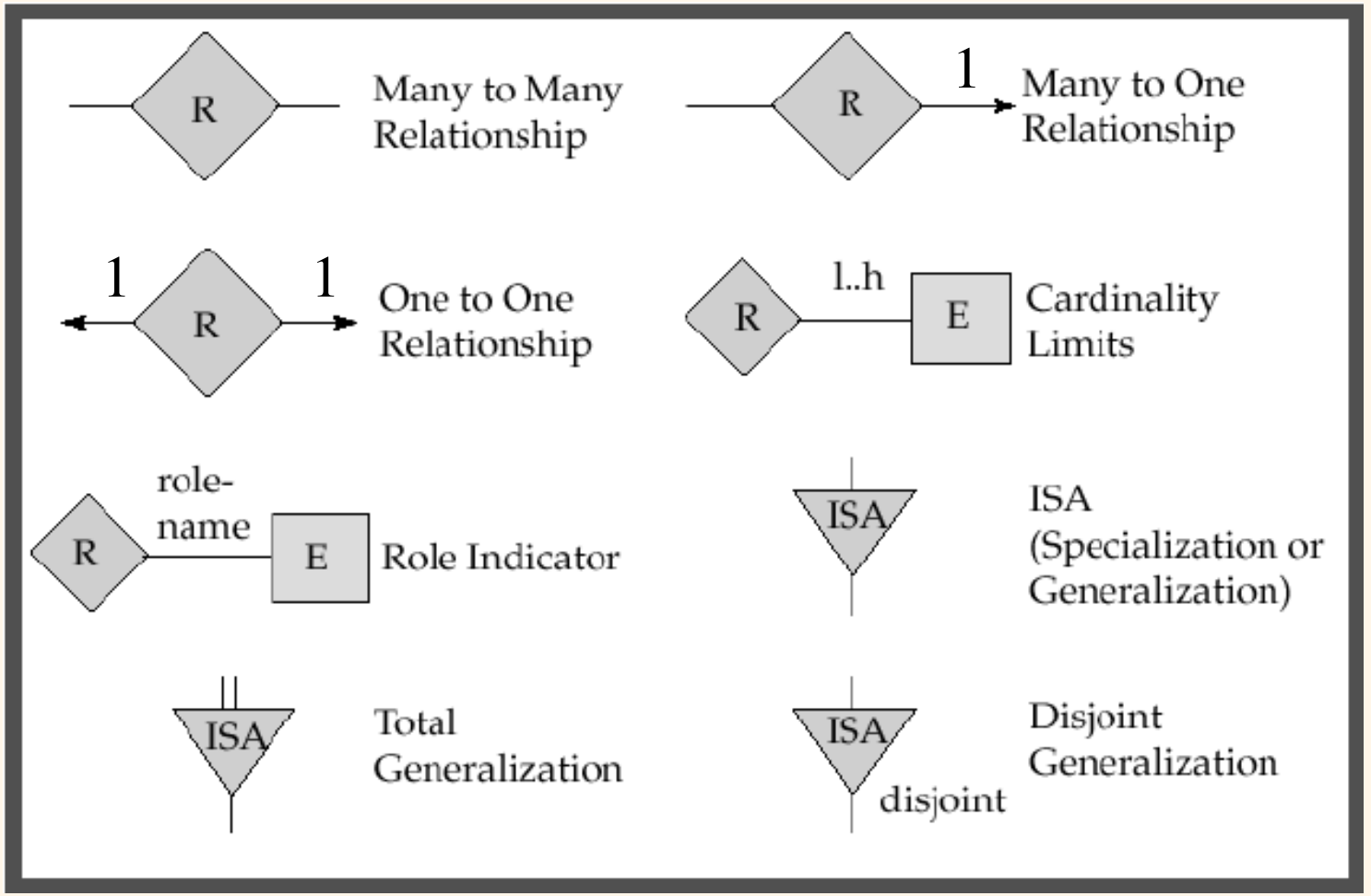


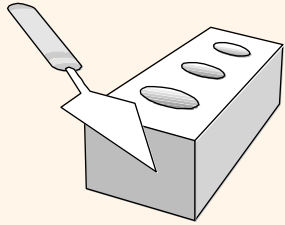
Summary



Página 39, Silberschatz

Summary

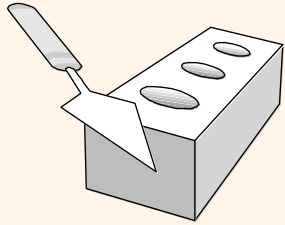

R
Many to Many Relationship
R
1
Many to One Relationship
1
R
1
One to One Relationship
R
l..h
E
Cardinality Limits
R
role-name
E
Role Indicator
ISA
ISA (Specialization or Generalization)
ISA
Total Generalization
ISA
disjoint
Disjoint Generalization



Use Case example

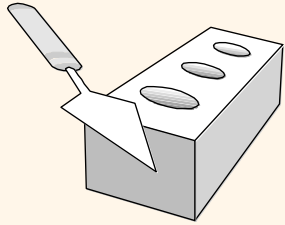
Arrange Meeting

1. The user chooses the option to arrange a meeting.
2. The system prompts user for the names of attendees.
3. The user types in a list of names.
4. The system checks that the list is valid.
5. The system prompts the user for meeting constraints.
6. The user types in meeting constraints.
7. The system searches the calendars for a date that satisfies the constraints.
8. The system displays a list of potential dates.
9. The user chooses one of the dates.
10. The system writes the meeting into the calendar.
11. The system emails all the meeting participants informing them of them appointment



Exercise

A database will be made to store information about patients in a hospital. On arrival, each patient's personal details (name, address, and telephone number) are recorded where possible, and they are given an admission number. They are then assigned to a particular ward (Accident and Emergency, Cardiology, Oncology, etc.). In each ward there are a number of doctors and nurses. A patient will be treated by one doctor and several nurses over the course of their stay, and each doctor and nurse may be involved with several patients at any given time



Constructing a ER Model

1. Identify entities
2. Remove duplicate entities
3. Identify attributes
4. Mark Primary Keys
5. Define Relationships
6. Describe cardinality
7. Remove redundant relationships