Institute of Information and Communication and Technology
Bangladesh University of Engineering and Technology

ICT 5103: Database Design and Management

Lecture 2

Instructor: Samin Rahman Khan
Lecturer, IICT, BUET

The slides are taken from the following book.

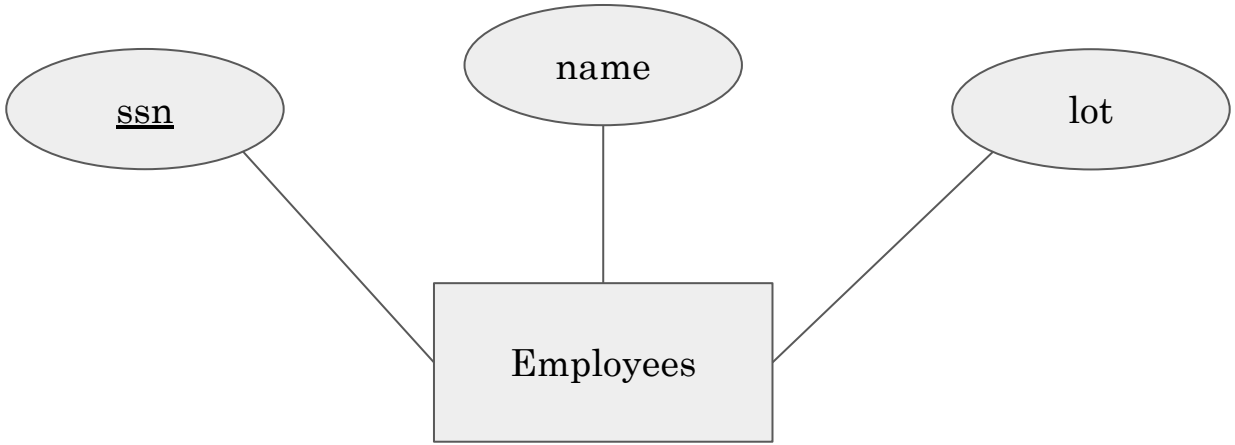
© Database Management Systems 3ed, R. Ramakrishnan and J. Gehrke

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**: A real-world object distinguishable from other objects
  - An object is described (in DB) with a set of **attributes**.



ER Model Basics (Contd.)

- **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** – a minimal set of attributes whose values uniquely identify an entity in the set.
  - There could be more than **candidate key**; one of them is designated and is known as **primary key**.
  - Each attribute has a **domain** – a set of permitted values.

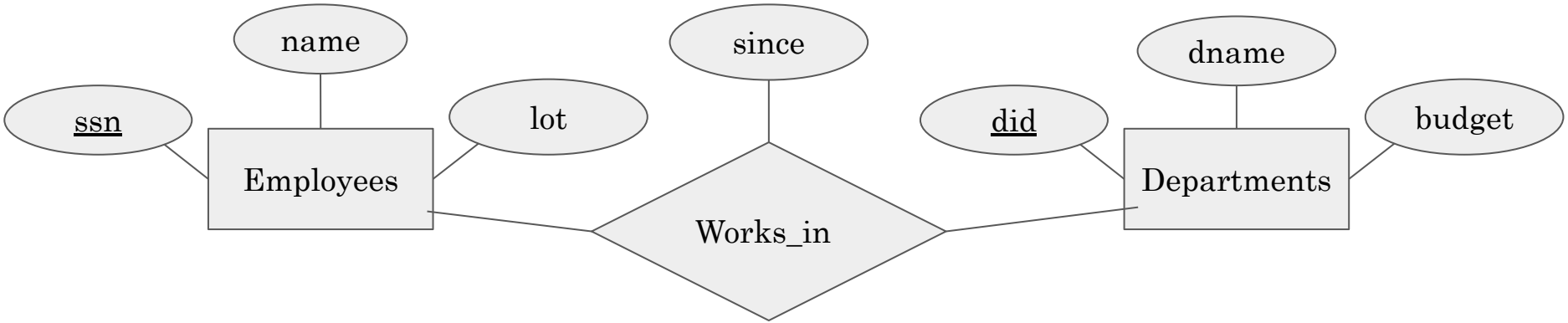
ER Model Basics (Contd.)

- **Relationship:** Association among two or more entities. E.g., Attishoo works in the Physics department.
- **Relationship Set**: Collection of similar relationships.
  - An n-ary relationship set R relates n entity sets
    - Entity Set: $E_1 \ldots E_n$.
    - Relationship Set: $\{(e_1, \ldots, e_n) \mid e_1 \in E_1, \ldots, e_n \in E_n\}$.
  - Same entity set could participate in different relationship sets, or in different “roles” in the same set.

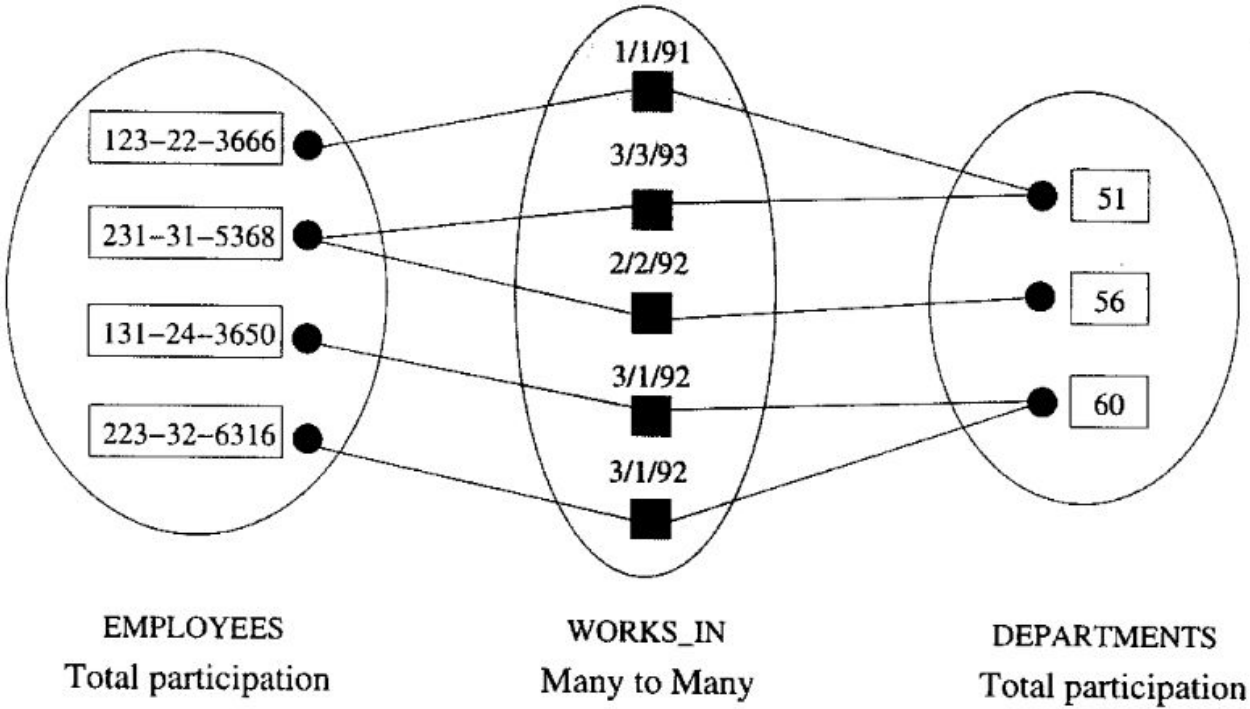
ER Model Basics (Contd.)

- A relationship can also have descriptive **attributes**.
  - Record information about the relationship; rather than any participating entities.
- A relationship can be identified by the participating entities.

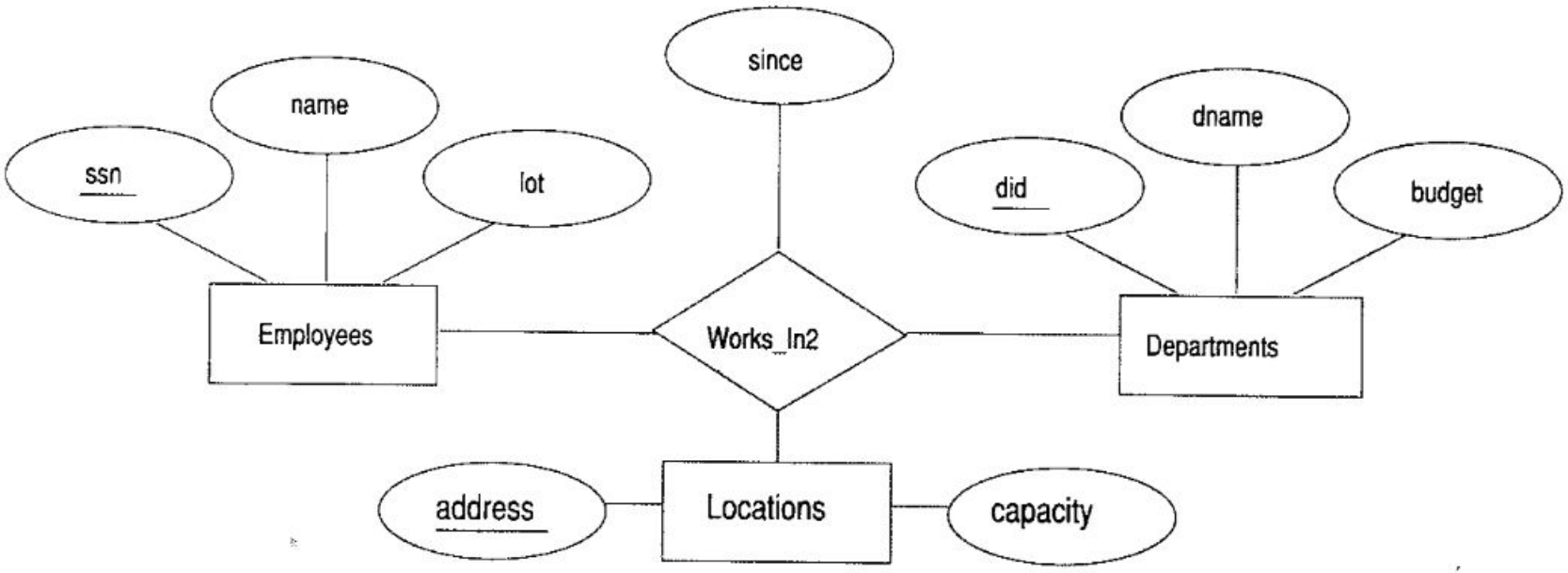
ER Model Example



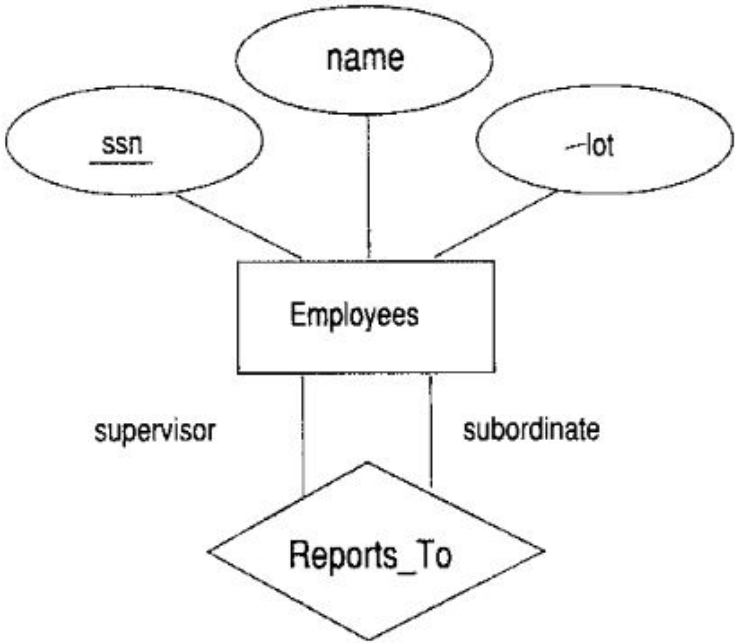
An Instance of Works_in



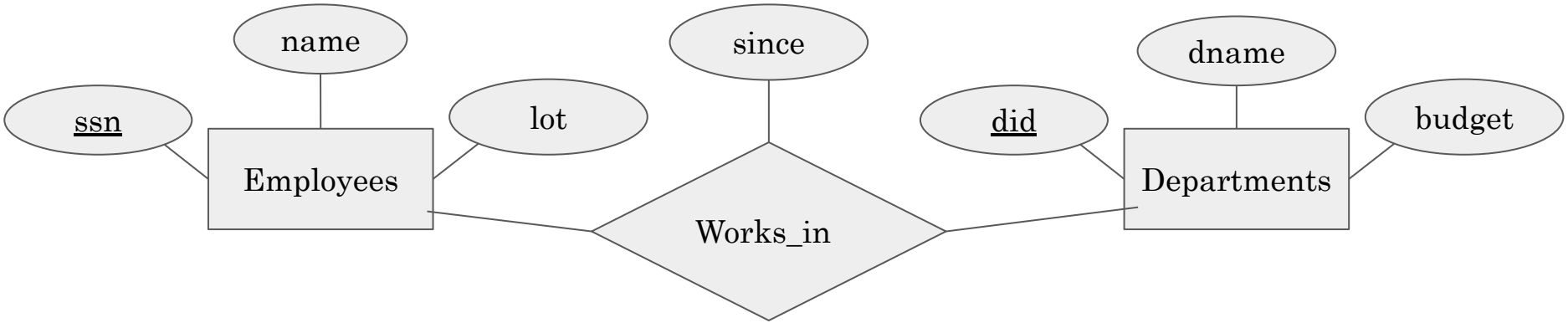
Ternary Relationship



Relationships with roles

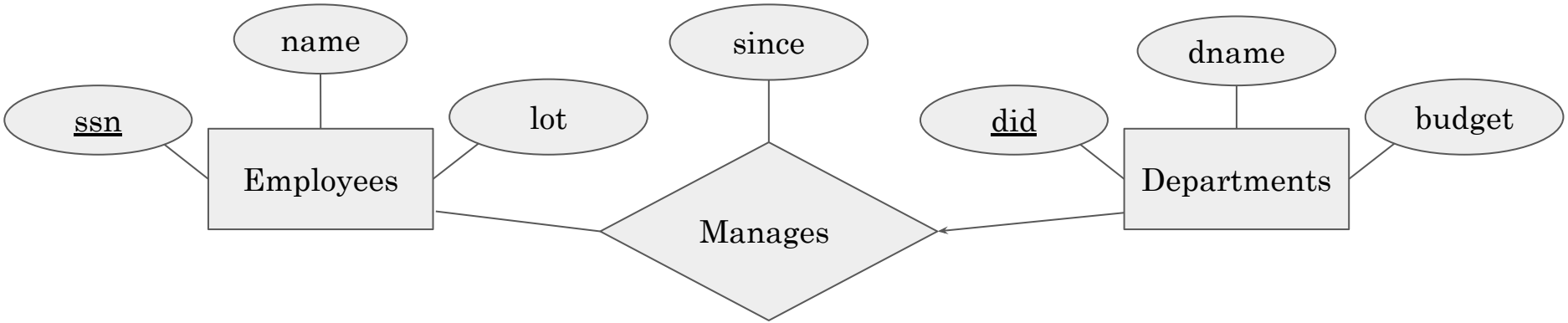


Key Constraints



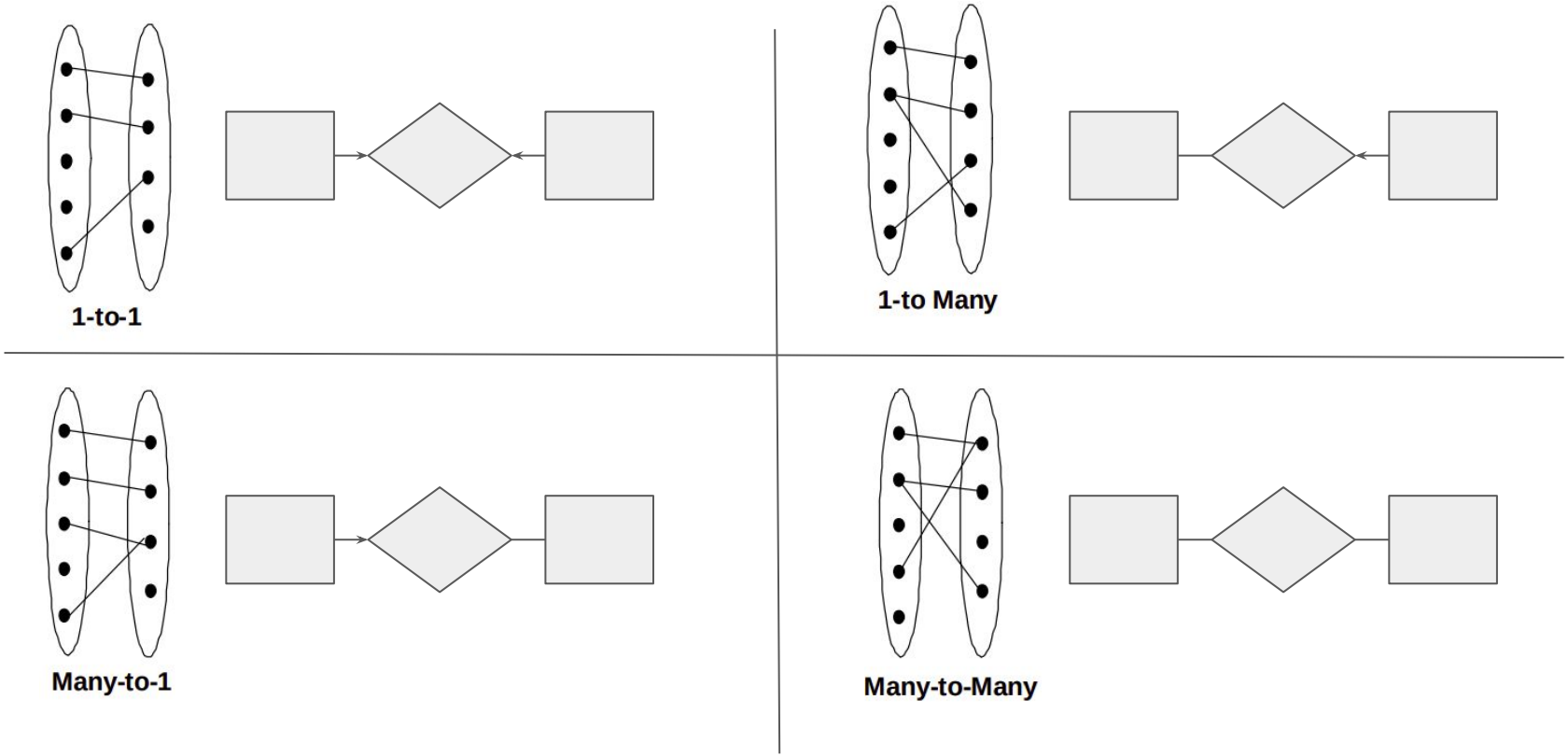
- An employee can work in many departments
- A department can have many employees

Key Constraints (Contd.)

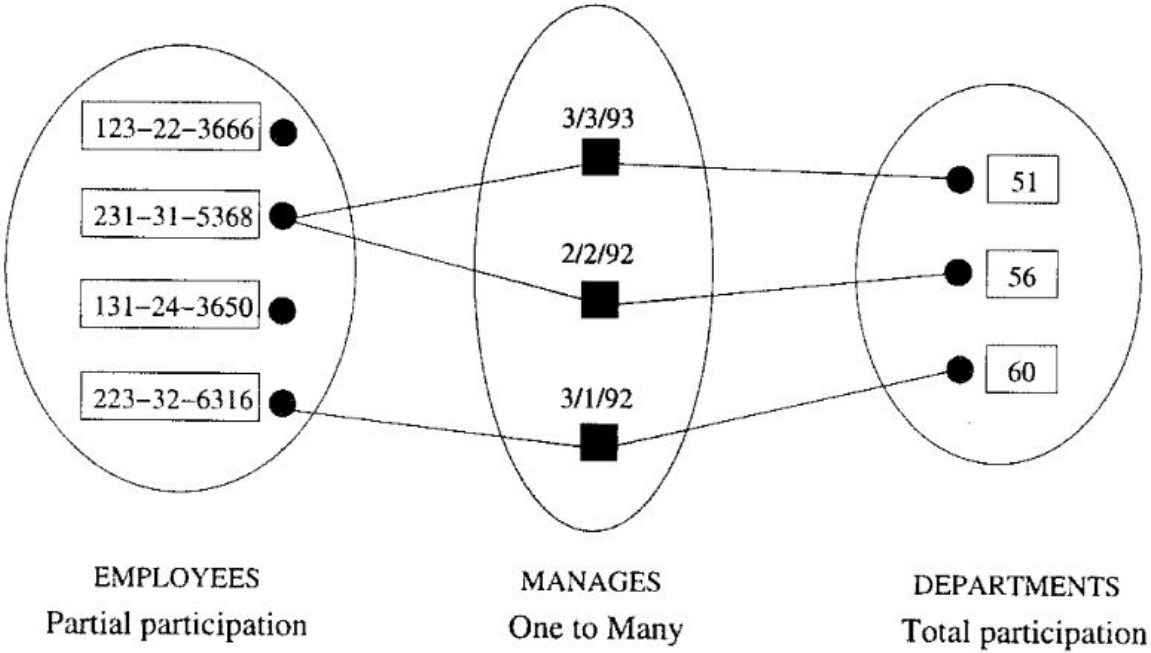


- Each department has at most one manager
  - 1-to-many

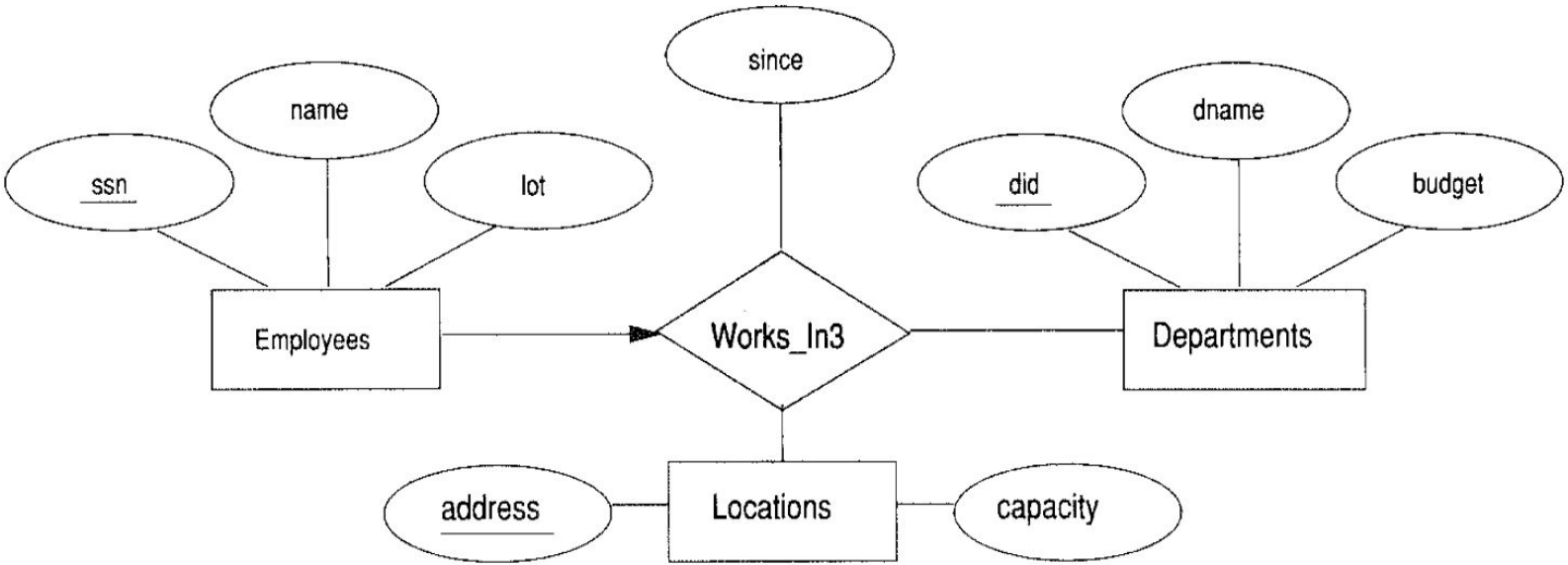
Key Constraints (Contd.)



Instance of Manages

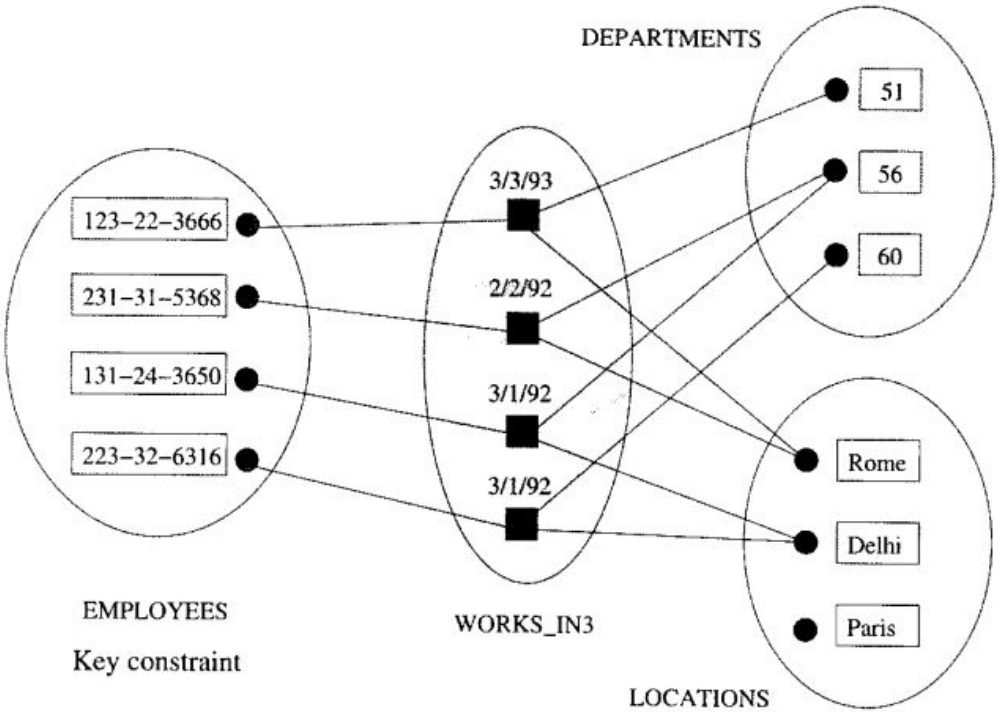


Key Constraints for Ternary Relationships



- Each employee works in at most one department and at a single location.
- Each department can be associated with several employees and each location can be associated with several departments and employees.

Key Constraints for Ternary Relationships (Contd.)



Exercise!!

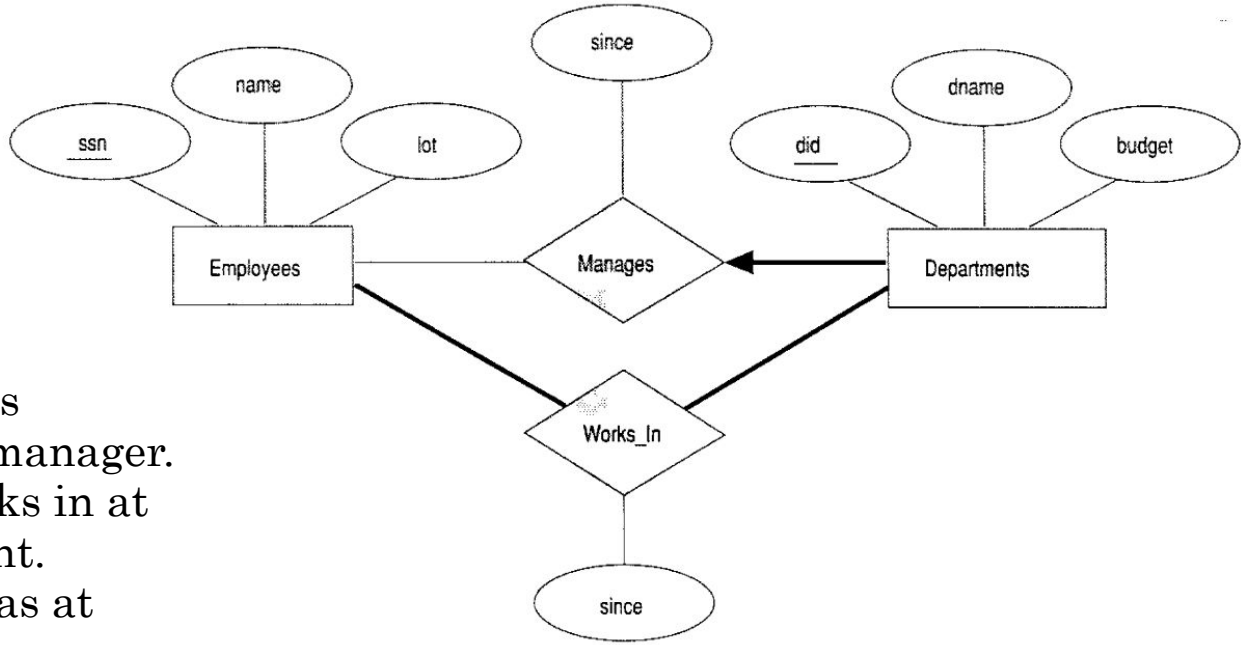
- A university database contains information about professors (identified by instructorid), courses (identified by courseid) and departments (identified by departmentid).
- Professors from can offer courses. We need to record which professor took which course in which semester.
- A department is managed by a single professor and one professor is in charge of a single department.

Answer

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 did value in Departments table must appear in a row of the Manages table (with a non-null ssn va
- If the participation of an entity set in a relationship set is total, the two are connected by a **thick line**.

Participation Constraints

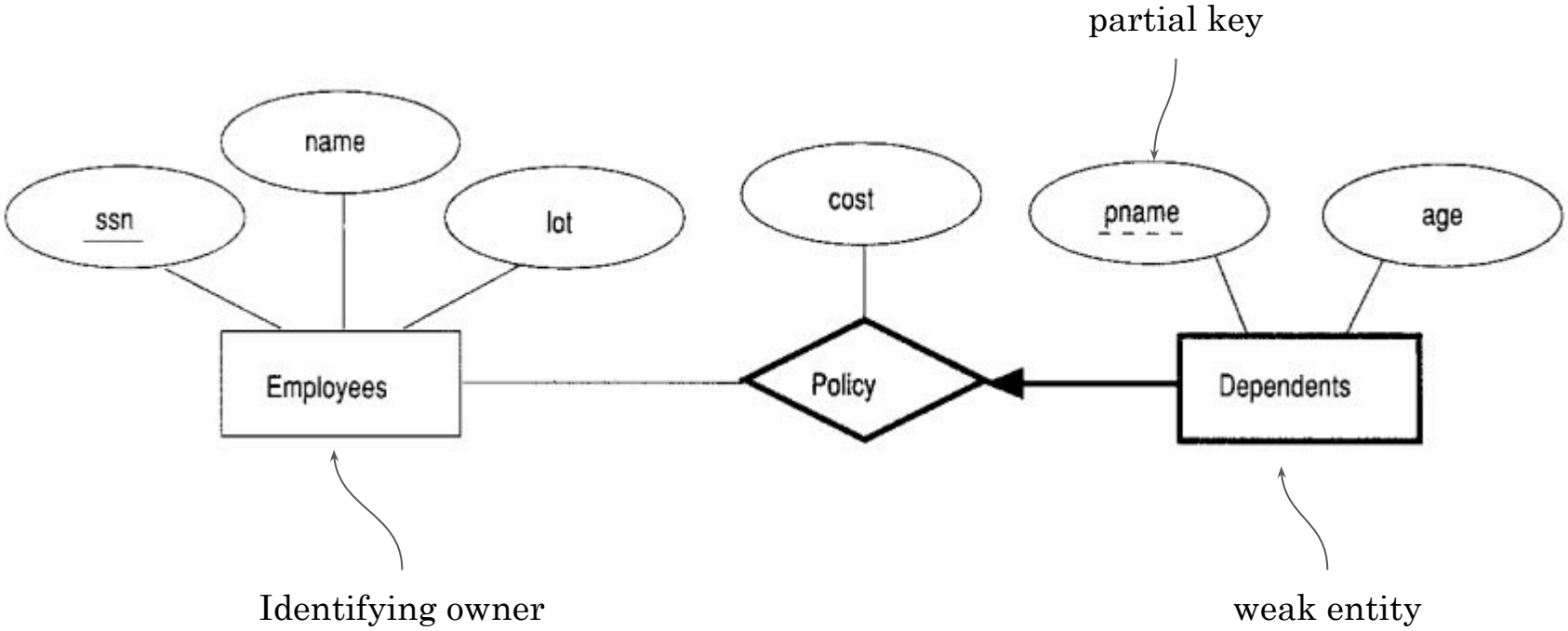


1. Every department is required to have a manager.
2. Each employee works in at least one department.
3. Each department has at least one employee.

Weak Entities

- **Weak Entity**: do not have key attributes. E.g.,
- A **weak entity** can be identified uniquely by considering the some of **its attributes** in conjunction with **primary key of another (owner) entity**, which is called **identifying owner**.
- The set of attributes of a weak entity set that uniquely identify a weak entity for a given owner entity is called a **partial key** of the weak entity set.

Weak Entities (Contd.)



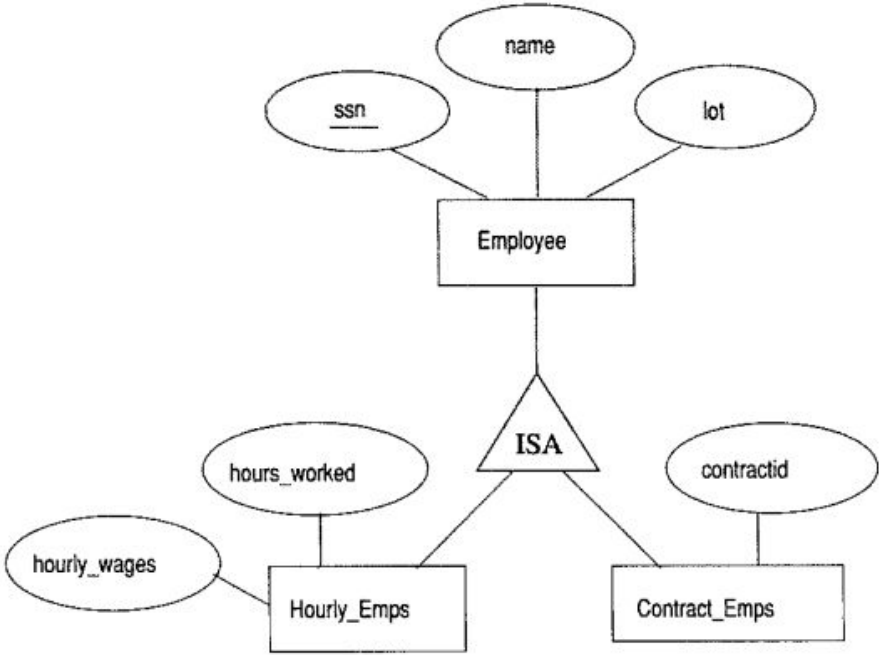
Weak Entities (Contd.)

- 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**.

Class (ISA) Hierarchies

- Like in object-oriented programming, attributes can be **inherited**.
- **A ISA B** = every A entity is also a B entity
- Example:
  - An hourly employee (Hourly_emps) is also an employee (Employees)
  - The attributes for hourly employees (Hourly_emps) include the attributes of employees (Employees) in addition to attributes that the exclusive to itself.

Class (ISA) Hierarchies (Contd.)



Two Ways to View Class Hierarchy

- **Specialization**
  - Process of identifying subsets of an entity set (the **superclass**) that share some distinguishing characteristic.
  - Typically, the **superclass** is defined first, the **subclasses** are defined next, the subclass-specific attributes and relationship sets are then added.
- **Generalization**
  - Process of identifying some common characteristics of a collection of entity sets and creating a new entity that contains entities possessing these common characteristics.
  - Typically, the subclasses are defined first, the superclass is defined next, and any relationship sets that involve the superclass are then defined.

Constraints in ISA hierarchy

- **Overlap constraints**
  - Determine whether two subclasses are allowed to contain the same entity.
  - E.g., Can Attishoo be both a Hourly_Emps and a Contract_Emps (No). Can he be both a Contract_Emps and a Senior_Emps (Yes). Denoted by writing Contract_Emps OVERLAPS Senior_Emps.
- **Covering constraints**
  - Determine whether the entities in the subclasses collectively include all entities in the superclass.
  - E.g., does every Employees entity have to belong to one of its subclasses? (No). If every Motor_Vehicle attribute is in either Motorboats or Cars entity, we write Motorboats AND Cars COVER Motor_Vehicles.

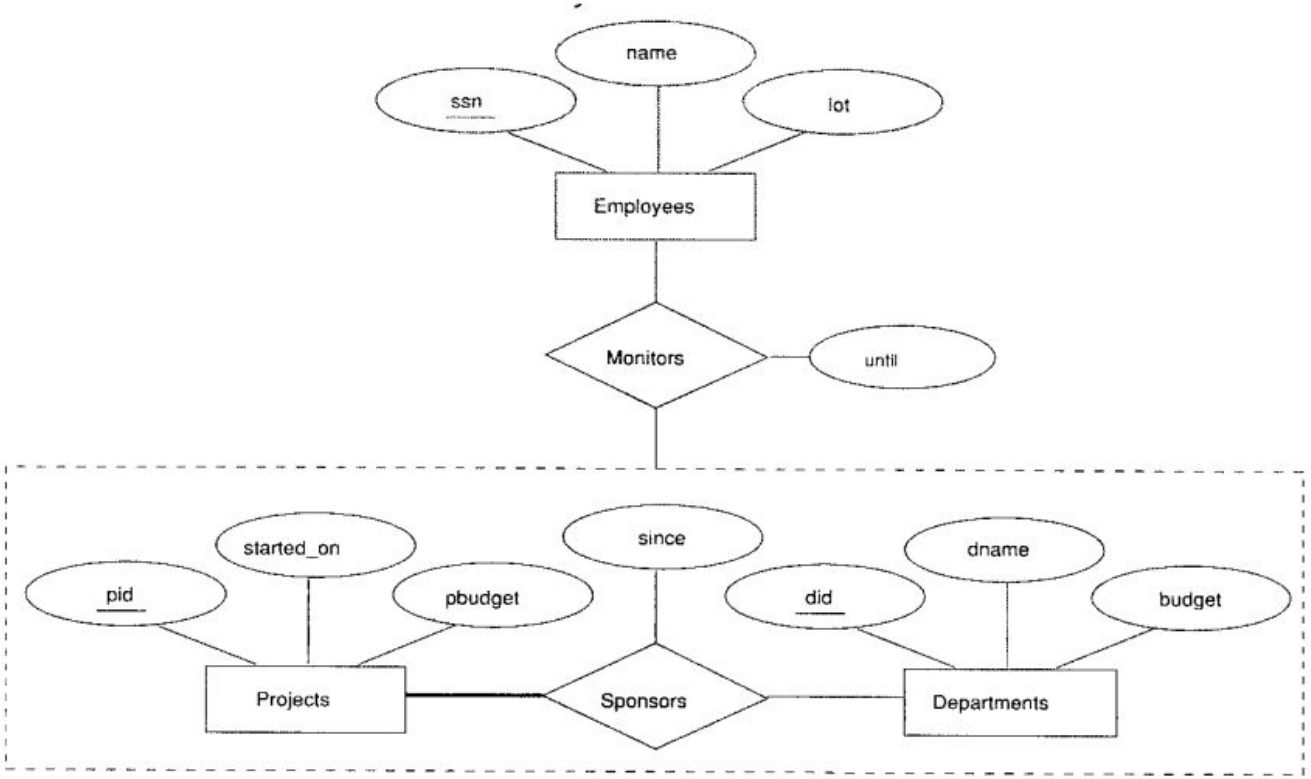
Reasons for using ISA

- To add descriptive attributes specific to a subclass.
  - E.g., hourly_wages is specific to Hourly_Emps.
- To identify entities that participate in a relationship.
  - E.g., Motorboats and Cars may have different descriptive attributes (say tonnage and number of doors), but as Motor_Vehicles entities, they must be licensed. The licensing information can be captured by a Licensed_To relationship between Motor_Vehicles and an entity set called Owners.

Aggregation

- Used when we have to model a relationship involving (entity sets and) a relationship set.
- Aggregation allow us to treat a relationship set as an entity set for purposes of participation in (other) relationships.
- We use dashed box around the relationship set to denote aggregation.

Aggregation (Contd.)



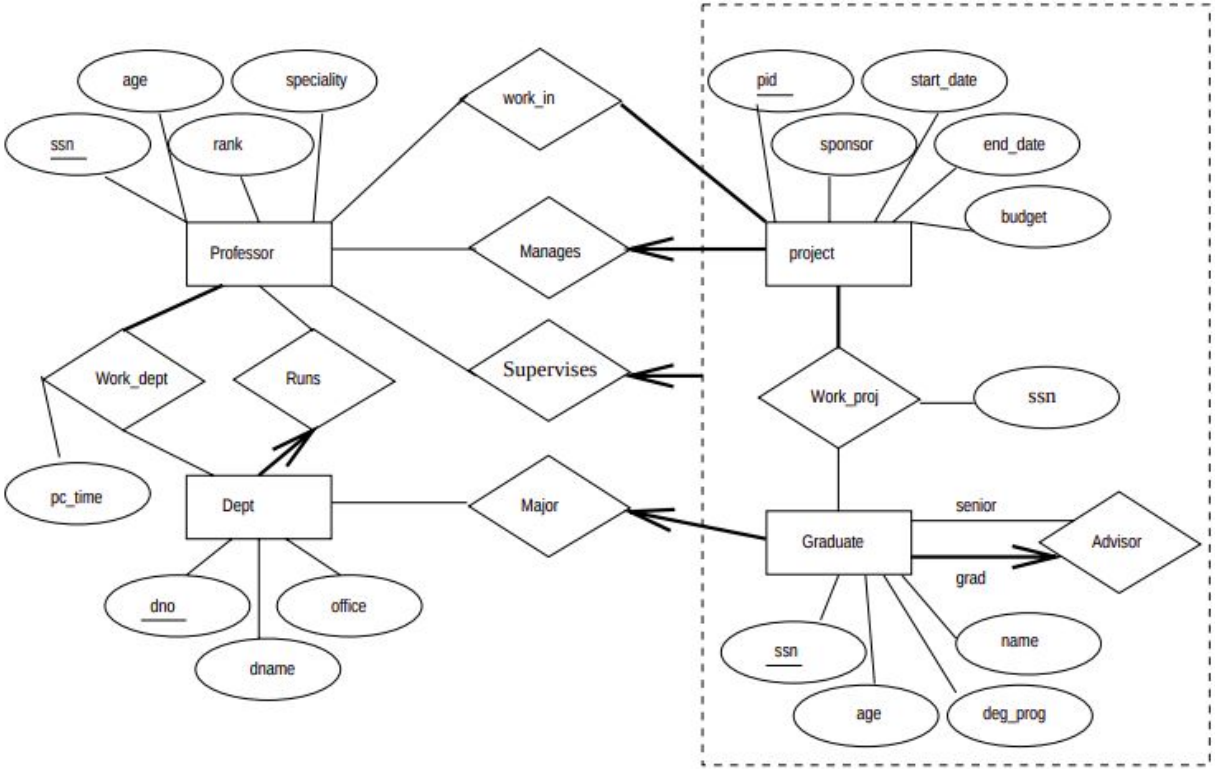
Exercise!!

Consider the following information about a university database:

1. Professors have an SSN, a name, an age, a rank, and a research specialty.
2. Projects have a project number, a sponsor name (e.g., NSF), a starting date, an ending date, and a budget.
3. Graduate students have an SSN, a name, an age, and a degree program (e.g., M.S. or Ph.D.).
4. Each project is managed by one professor (known as the project's principal investigator).
5. Each project is worked on by one or more professors (known as the project's co-investigators).
6. Professors can manage and/or work on multiple projects.
7. Each project is worked on by one or more graduate students (known as the project's research assistants).
8. When graduate students work on a project, a professor must supervise their work on the project. Graduate students can work on multiple projects, in which case they will have a (potentially different) supervisor for each one.
9. Departments have a department number, a department name, and a main office.
10. Departments have a professor (known as the chairman) who runs the department.
11. Professors work in one or more departments, and for each department that they work in, a time percentage is associated with their job.
12. Graduate students have one major department in which they are working on their degree.
13. Each graduate student has another, more senior graduate student (known as a student advisor) who advises him or her on what courses to take.

Design and draw an ER diagram that captures the information about the university.

Answer


age
speciality
ssn
rank
work_in
pid
start_date
sponsor
end_date
budget
Professor
Manages
project
Work_dept
Runs
Supervises
Work_proj
ssn
pc_time
Dept
Major
senior
Graduate
Advisor
grad
dno
office
dname
ssn
name
age
deg_prog

Conceptual Design with ER Model

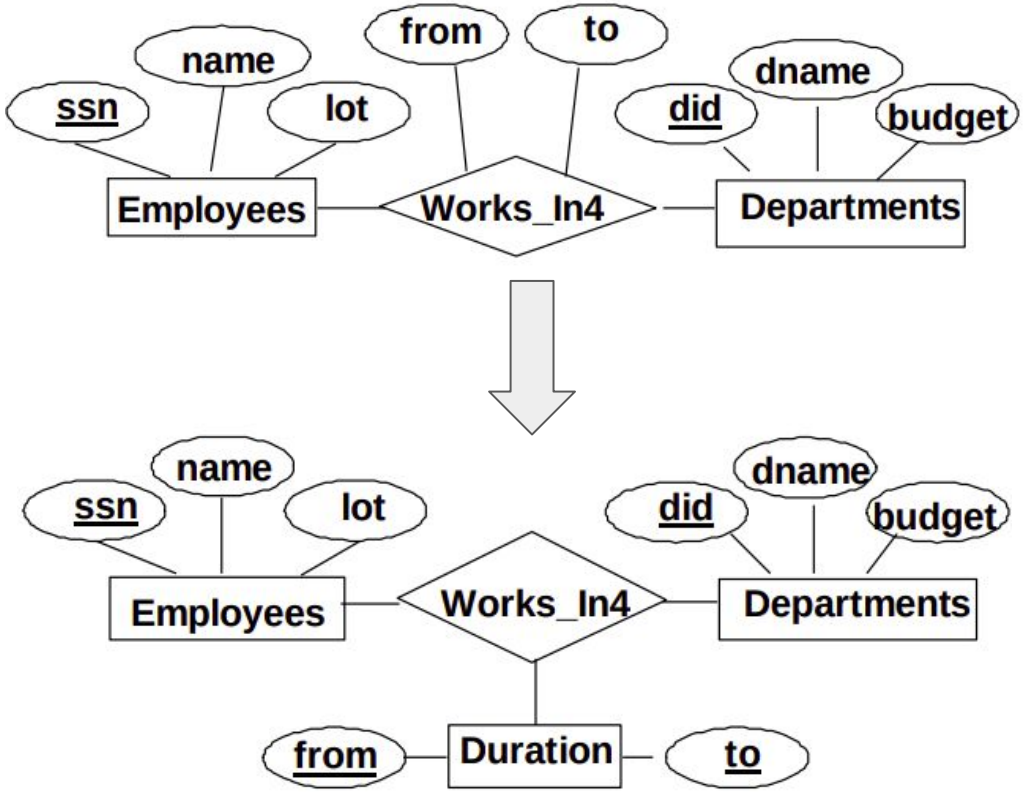
- Design choices:
  - Should a concept be modeled as an entity or an attribute?
  - Should a concept be modeled as an entity or a relationship?
  - Identifying relationships: Binary or ternary?
  - Should we use Aggregation?
- Constraints in the ER Model:
  - A lot of data semantics can (and should) 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:
  - **Multi-valued**: If we have several addresses per employee, address must be an entity (since attributes cannot be set-valued).
  - **Composite**: 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).

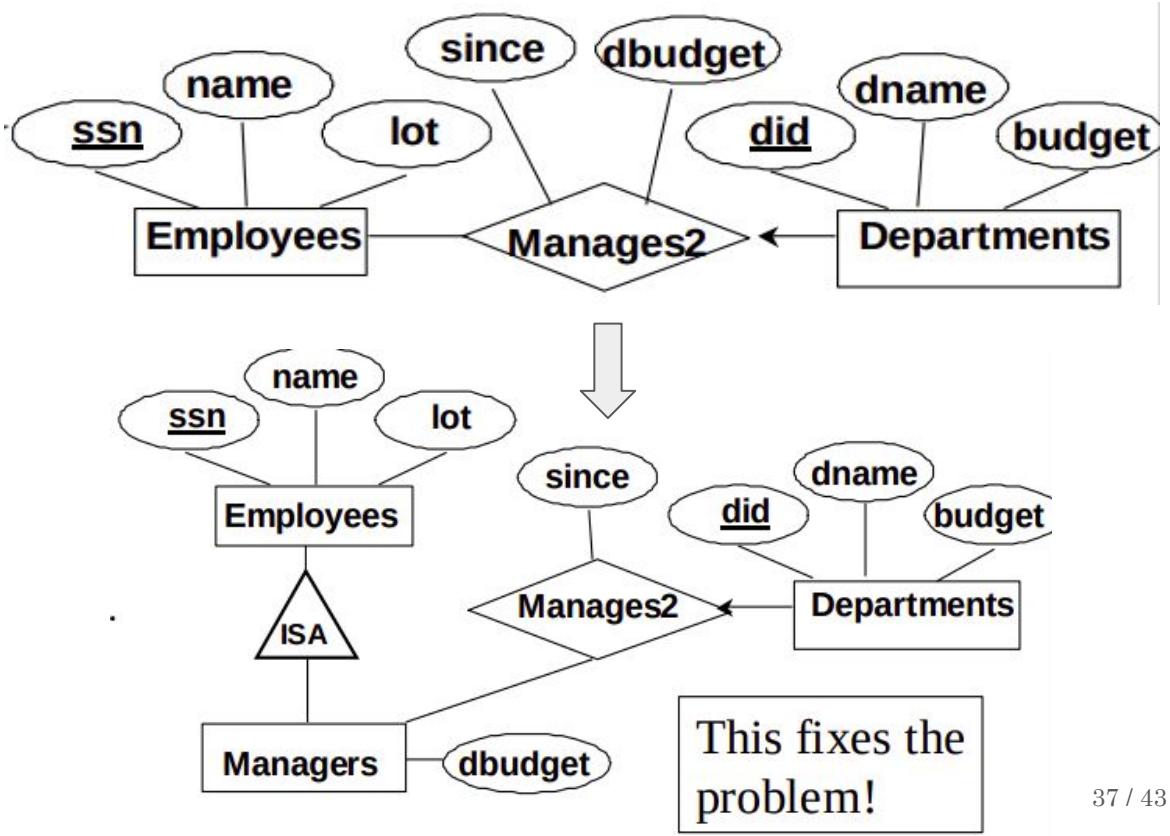
Entity vs. Attribute (Contd.)

- 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.



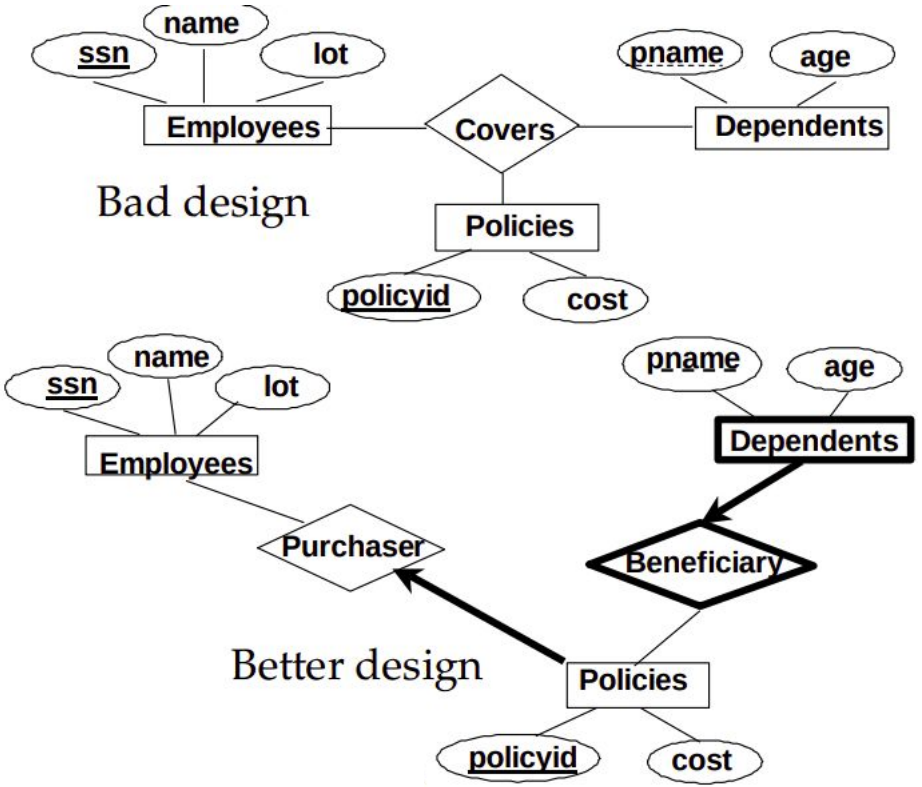
Entity vs. Relationship

- 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.



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.
- What are the additional constraints in the 2nd diagram?

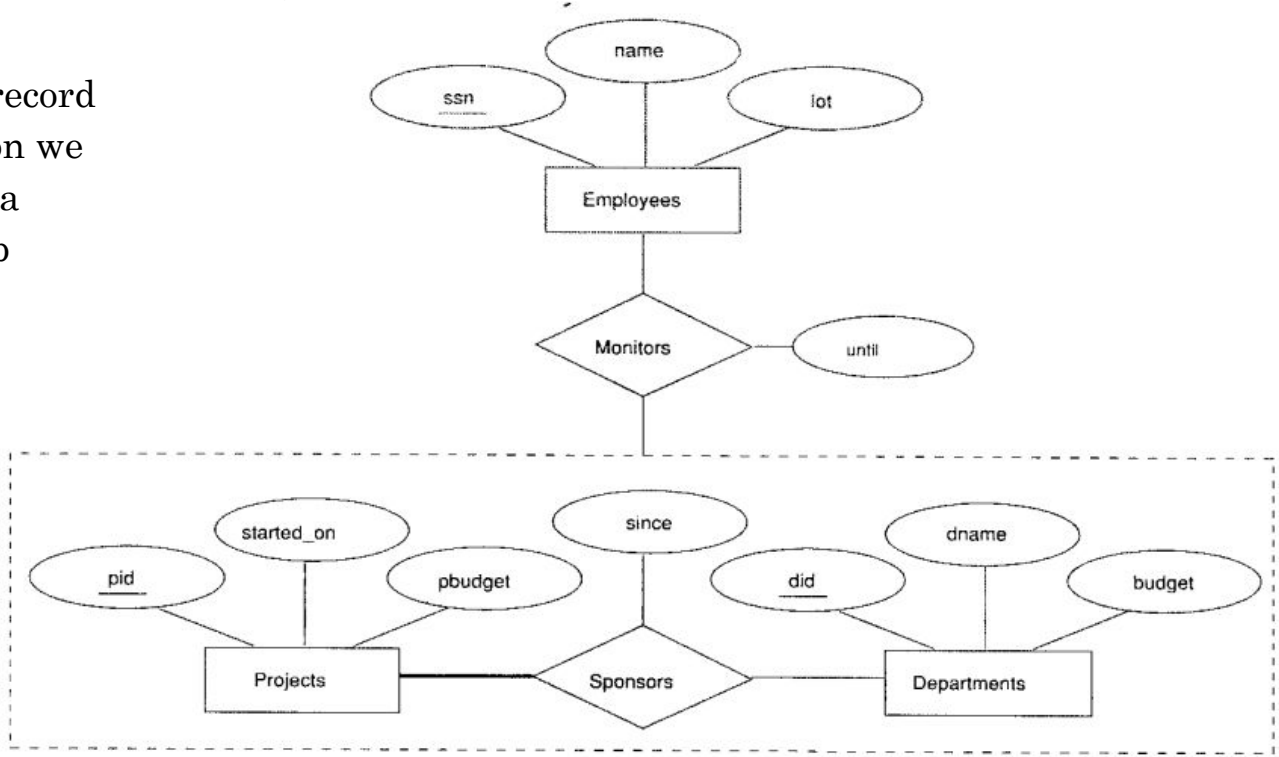


Binary vs. Ternary Relationships (Contd.)

- 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?

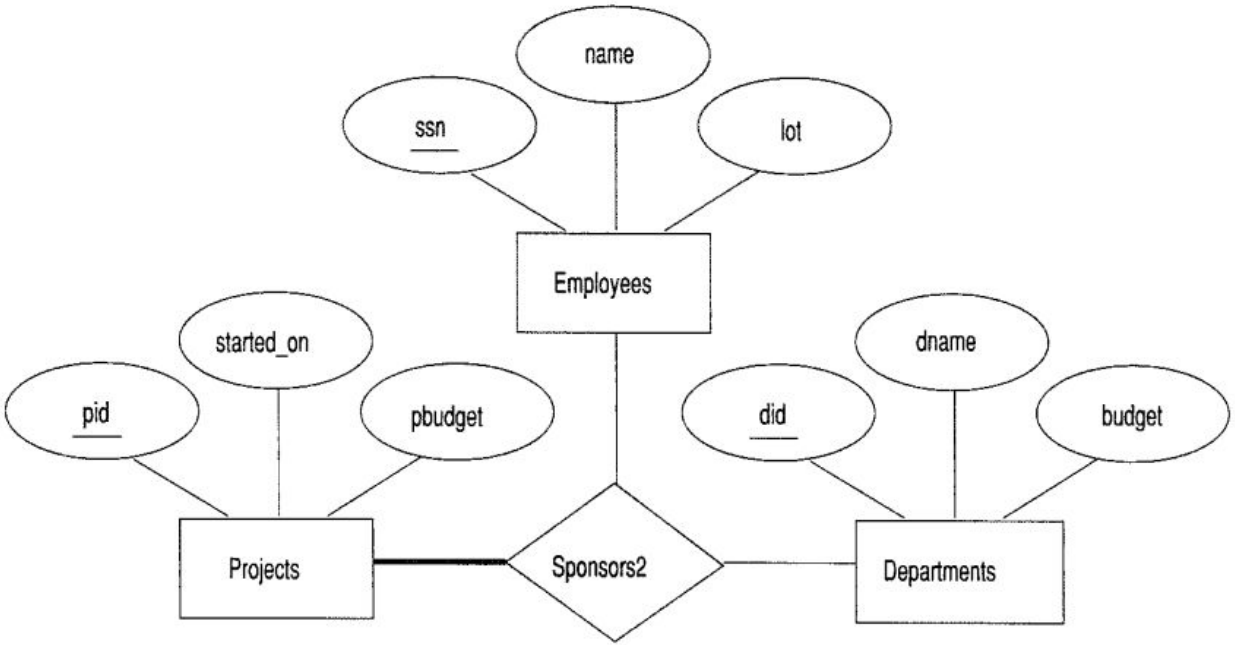
Aggregation vs. Ternary Relationships

- If we don't need to record the until information we can reasonably use a ternary relationship



Aggregation vs. Ternary Relationships (Contd.)

- If we don't need to record the until information we can reasonably use a ternary relationship



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.

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.