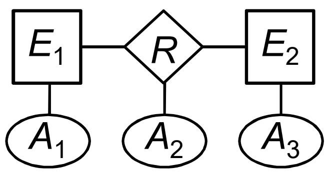
Entity-Relationship Model

The Big Picture (I)

The Big Picture (II)

Entity-Relationship Model

Entities, attributes, relationships



Introduction

- Entity-Relationship model (for short: E-R model) for conceptual database design
- Based on the seminal article:

  Peter P. Chen. The Entity-Relationship Model -Toward a Unified View of Data. *Transactions on Database Systems* 1(1):9-36 (1976)
- E-R model (besides UML) has great importance in practice
- Two-phase procedure for DB design
  - Phase 1: Requirements analysis and design of an E-R model
  - Phase 2: Transformation of the E-R model into a concrete logical model
- Goal: Modeling of an interesting part of the “real world” by abstraction so that questions about its *data* and the *relationships between the data* can be answered with the aid of the model
- E-R model describes the “real world” by entities (objects), attributes (properties), and relationships between entities

Entity Sets

- Entities are distinguishable, independent, self-contained, physically or intellectually existing concepts of the mini-world to be modeled

- Similar entities are collected in an entity set, e.g., the set of all books, the set of all cars, the set of all students, the set of all professors

- An entity is described by a set of pertaining properties (attributes), e.g., each book has an ISBN number, an author, a publisher, ...

- The values of an attribute are from domains or data types such as *integer*, *real*, *string*, ..., e.g., the name of an author is of type *string*

- A *minimal set of attributes* whose values *uniquely* characterize the associated entity among all entities of an entity set is called a key, e.g., an ISBN number uniquely identifies a book, an article number uniquely determines an article

Relationship Sets (I)

- A few set-theoretic concepts and notations are needed
  - Set: A homogeneous collection of distinct elements, e.g., set of integers, set of persons, set of cars
  - Type: Similar term to "set", more used in Computer Science and programming languages
  - Cartesian product or cross product operator "$\times$" as an example of a *type constructor*: Let A, B be sets. Then $A \times B = \{(a, b) \mid a \in A, b \in B\}$
  - Subset predicate "$\subseteq$" : $A \subseteq B \Leftrightarrow \forall\, x \in A : x \in B$
  - Proper subset predicate "$\subset$" :
    $A \subset B \Leftrightarrow ((\forall\, x \in A : x \in B) \wedge (\exists\, x \in B : x \notin A))$
  - Relation as any subset of a Cartesian product: Let $A_1, A_2, \ldots, A_n$ be sets. Let $n \in \mathbb{N}$. Then a relation R is given as $R \subseteq A_1 \times A_2 \times \ldots \times A_n$.

- In the E-R model a relationship describes a connection between several entities, e.g., student Smith *attends* lecture CIS 4301, teaching assistant Benson *works for* professor Meyer

Relationship Sets (II)

- Formally: A relationship set R between the entity sets E_1, E_2, ..., E_n ($n \in \mathbb{N}$) is a relation on the Cartesian product of the entity sets, i.e.,

  $R \subseteq E_1 \times E_2 \times \ldots \times E_n$

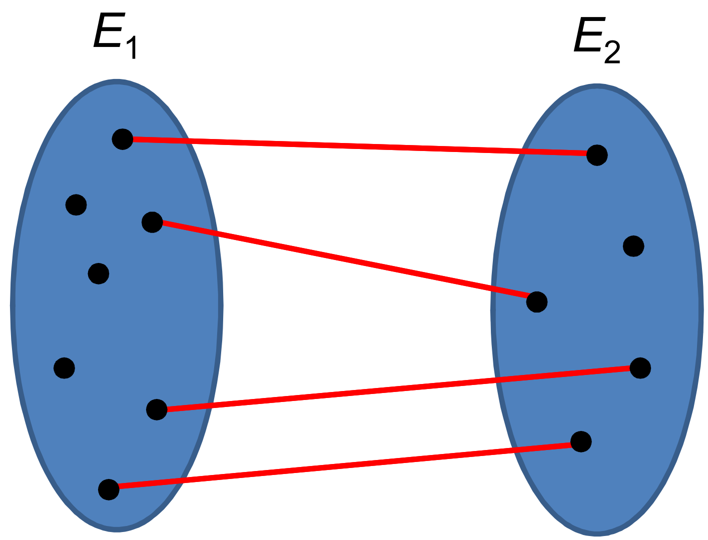
- Practical examples
  - attends_lecture $\subseteq$ students $\times$ lectures
  - works_for $\subseteq$ TAs $\times$ professors
- attributes may characterize relationships, e.g., *frequency* as an attribute for attends_lecture
- An entity set can occur more than once in a relationship set
- If there is only one entity set E participating in a *binary* relationship $R(E, E)$, each of these entity sets can be assigned roles, e.g.,

  is_precondition_of $\subseteq$ lectures $\times$ lectures

  first lecture / second lecture has the role of a predecessor / successor

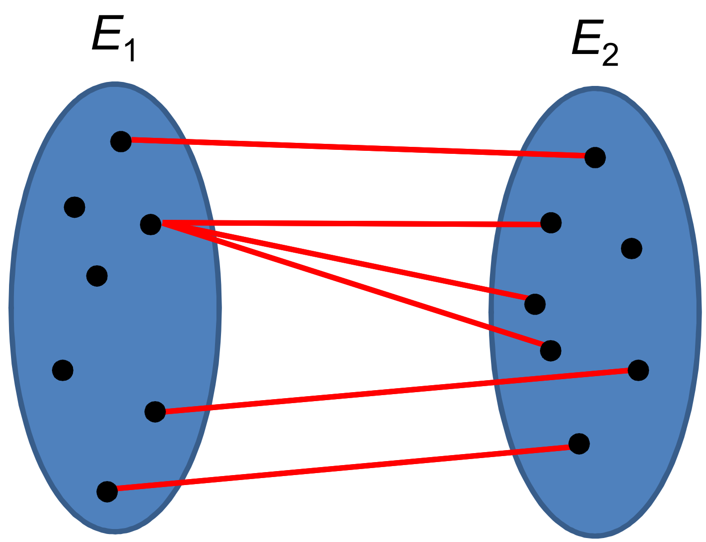
Cardinality of Binary Relationship Sets (I)

- Needed later for creating the actual database
- The cardinality decides about the number of tables that are needed to represent the relationship (performance argument)
- Cardinalities distinguished: 1:1, 1:m, m:1, m:n
- 1:1-relationship (one-to-one relationship)
  If for a binary relationship set $R(E_1, E_2)$ each entity in E_1 is associated with *at most* one entity in E_2, and vice versa



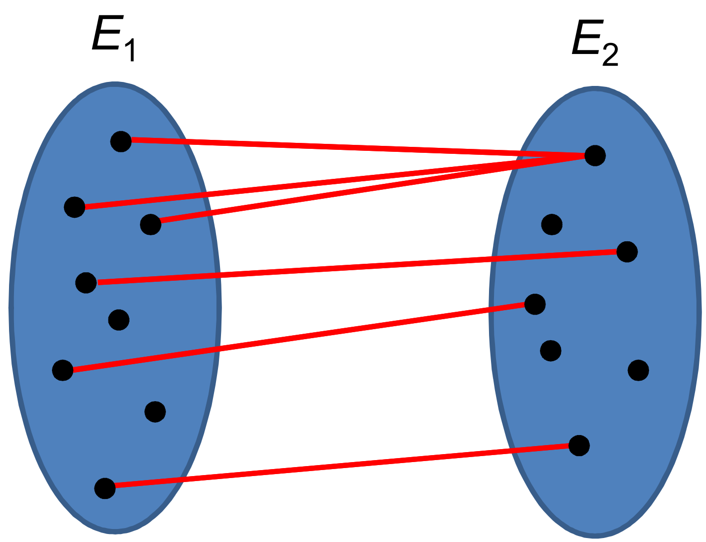
Cardinality of Binary Relationship Sets (II)

- 1:*m*-relationship (one-to-many relationship)
  If for a binary relationship set $R(E_1, E_2)$ each entity in E_1 is associated with any number (*zero or more*) of entities in E_2, and each entity in E_2 is associated with *at most* one entity in E_1



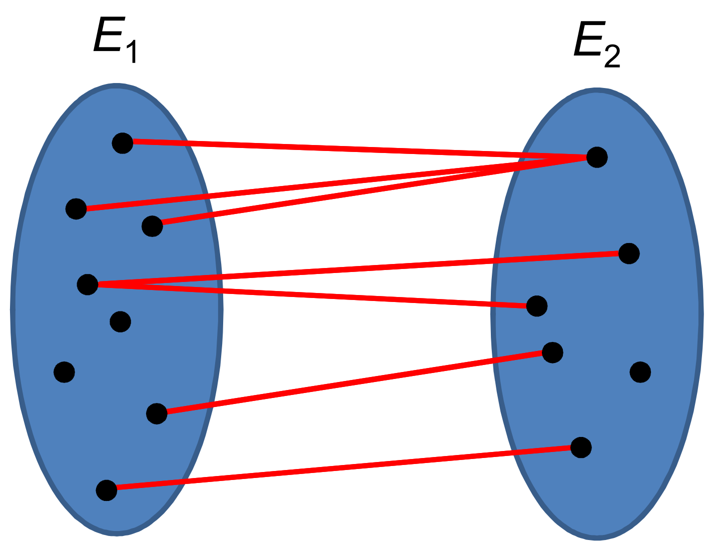
Cardinality of Binary Relationship Sets (III)

- *m*:1-relationship (many-to-one relationship)
  If for a binary relationship set $R(E_1, E_2)$ each entity in E_1 is associated with *at most* one entity in E_2, and each entity in E_2 is associated with any number (*zero or more*) of entities in E_1



Cardinality of Binary Relationship Sets (IV)

- *m*:*n*-relationship (many-to-many relationship)
  If for a binary relationship set $R(E_1, E_2)$ each entity in E_1 is associated with any number (*zero or more*) of entities in E_2, and vice versa

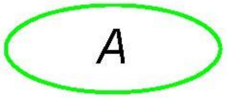


Cardinality of Binary Relationship Sets (V)

- Explanation of relationships at the single entity level: With how many entities of E_2 can an entity of E_1 be in a relationship, and vice versa?
- Examples of 1:1-relationships
  - Relationship set *husband_of* between men and women
  - Relationship set *owns* between persons and cars
- Examples of 1:*m*-relationships
  - Relationship set *teaches* between instructors and classes
  - Relationship set *supervises* between managers and employees
- Examples of *m*:1-relationships
  - Relationship set *works for* between employees and companies
  - Relationship set *belongs to* between cities and states
- Examples of *m*:*n*-relationships
  - Relationship set *attends* between students and classes
  - Relationship set *writes* between persons and books

E-R Diagrams (I)

- Graphical representation of entity sets, relationship sets, and their attributes by means of a graph enhanced by textual labels (colors only to improve clarity of representation, do not belong to the E-R model)
- Rectangles represent entity sets: 

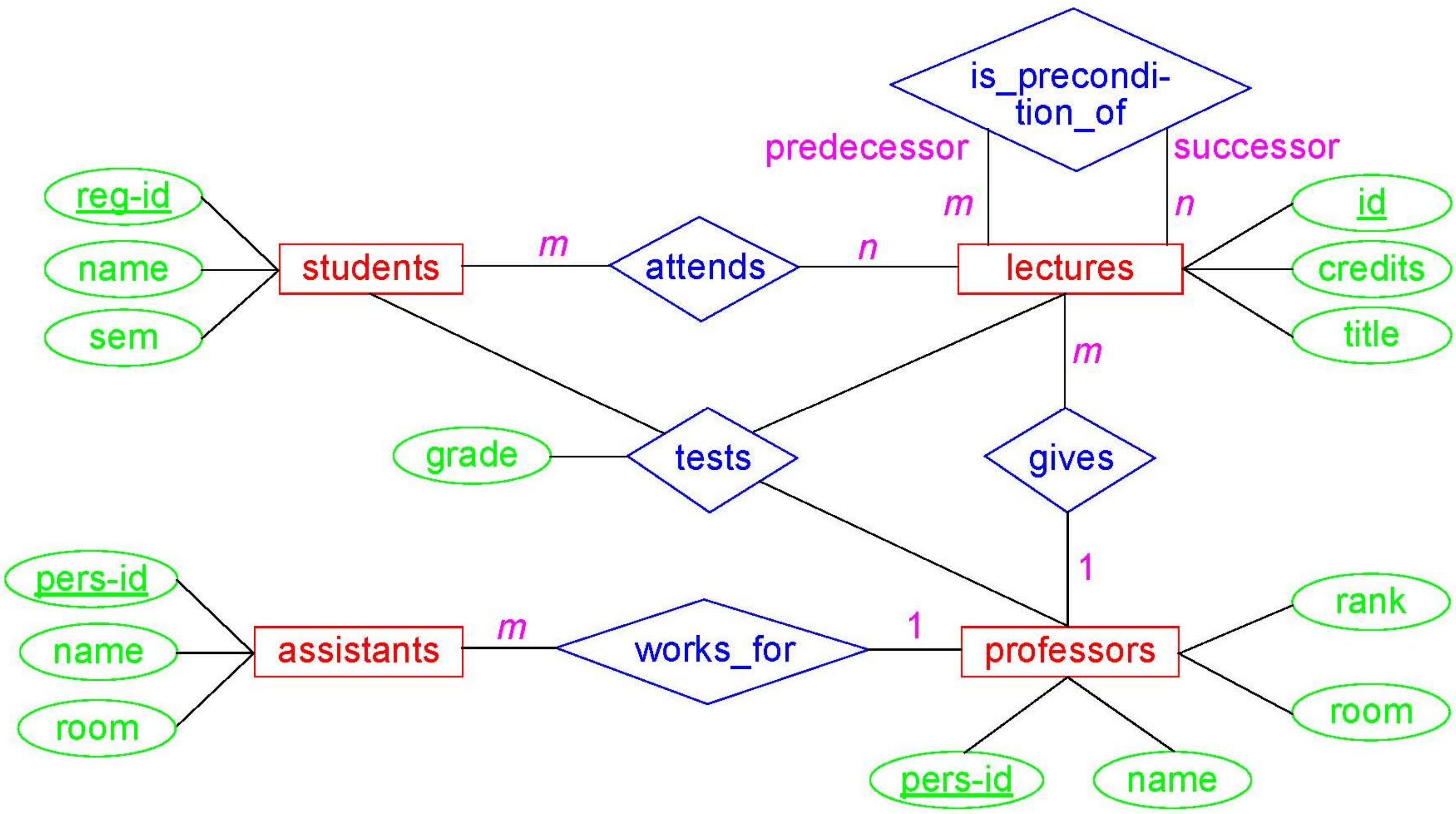
- Ellipses represent attributes: 

  - They are connected with their entity set by undirected edges
  - Key attributes are underlined
- Relationship sets are represented by diamonds: 

  - Relationship sets are connected with their pertaining entity sets by edges
  - Edges carry information about the cardinality of relationships according
- A role of a relationship set is attached to the corresponding edge

E-R Diagrams (II)

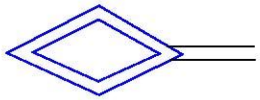
Example: Conceptual university schema



Weak Entity Sets (I)

- Existence dependent (that is, weak) entity set
  - Assumption so far: entities exist autonomously and can be uniquely identified within an entity set by their key attributes (strong entity set)
  - In reality there are also weak entities that do not have sufficient attributes to form a key. These entities are
    - dependent in their existence from another, superior entity and
    - can be uniquely identified only in combination with the key of a superior strong entity
  - But: Examples of weak entity sets are rare and thus difficult to find
  - Superior strong entity set is called identifying or owner entity set
  - Graphical notation: 

Weak Entity Sets (II)

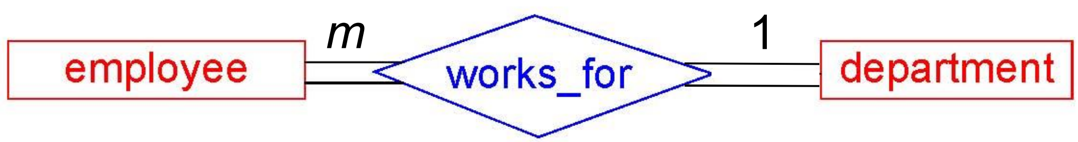
- Identifying relationship set
  - An *identifying relationship set R* associates a *weak entity* set E_1 with an *identifying entity set* E_2 such that the key of E_1 comprises the key of E_2 and *all* entities of E_1 are involved in a relationship to an entity in E_2 (total participation, see below).
  - Relationship from the weak entity set to the superior entity set has usually an *m*:1-cardinality and more seldom a 1:1-cardinality
  - Graphical notation: 

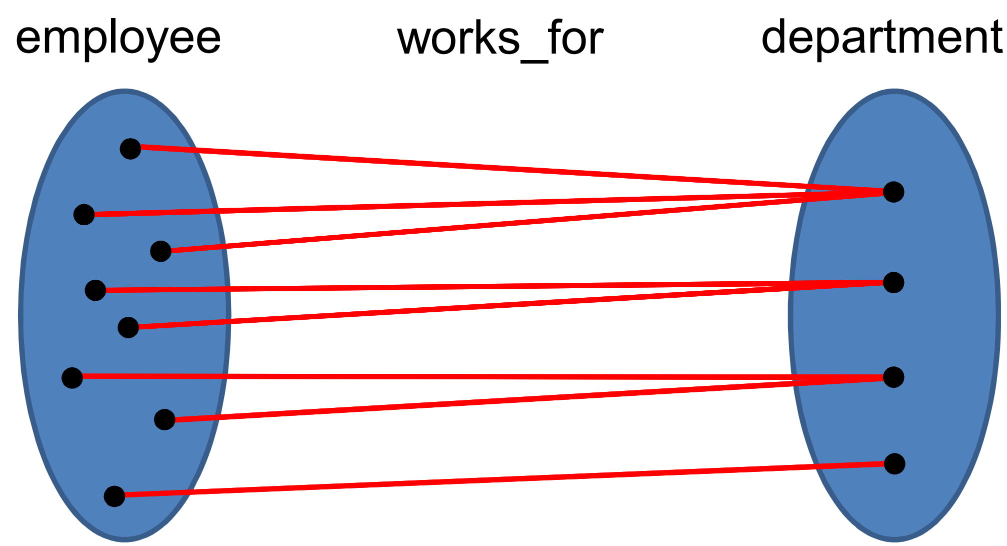
- Example

The weak entity set *room* has a total participation (indicated by the double edge) in the identifying relationship set *lies_in*.

Total Participation

- Total (or mandatory) participation of an entity set in a relationship
  - *All* entities of an entity set E_1 are associated with *at least* one other entity set E_2 by a relationship set R
  - This holds, in particular, for weak entity sets
- Example:





Min-Max Notation (I)

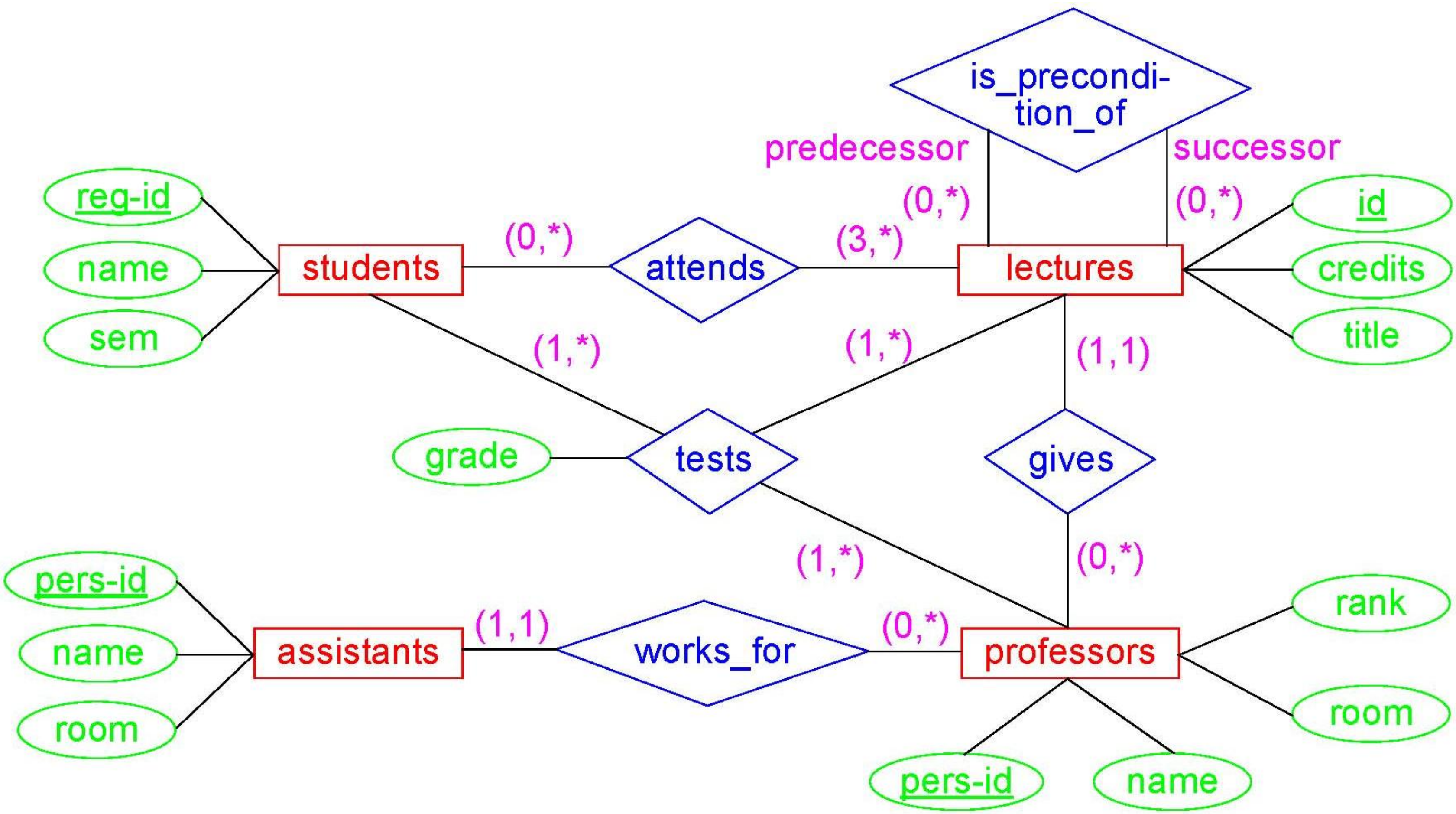
- More precise characterization and constraint of the cardinality of a relationship set by the (*min*, *max*)-notation

- For each entity set *E* participating in a relationship set *R*
  - the value *min* expresses that each entity of *E* participates in *at least min* relationships in *R*
  - the value *max* expresses that each entity of *E* participates in *at most max* relationships in *R*

- Special cases
  - *min* = 0: An entity does not have to be in a relationship (optional)
  - *max* = *: An entity may be in a relationship arbitrarily many times

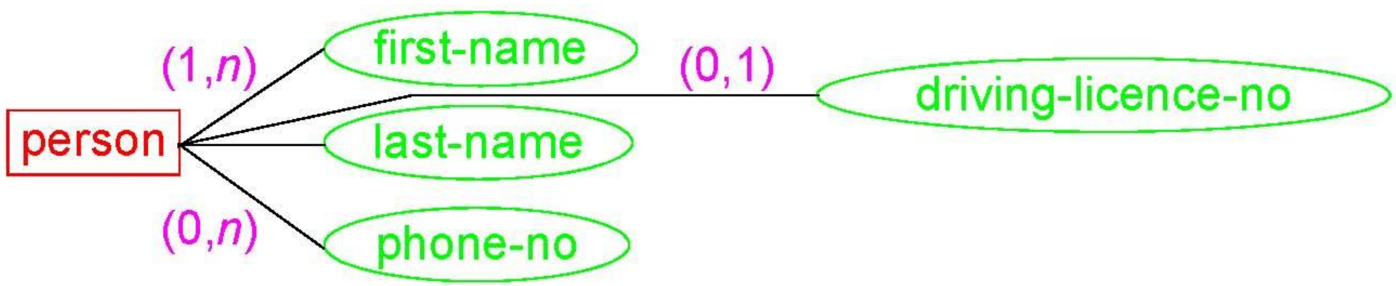
Min-Max Notation (II)

- Example: Conceptual university schema with (*min*, *max*)-notation



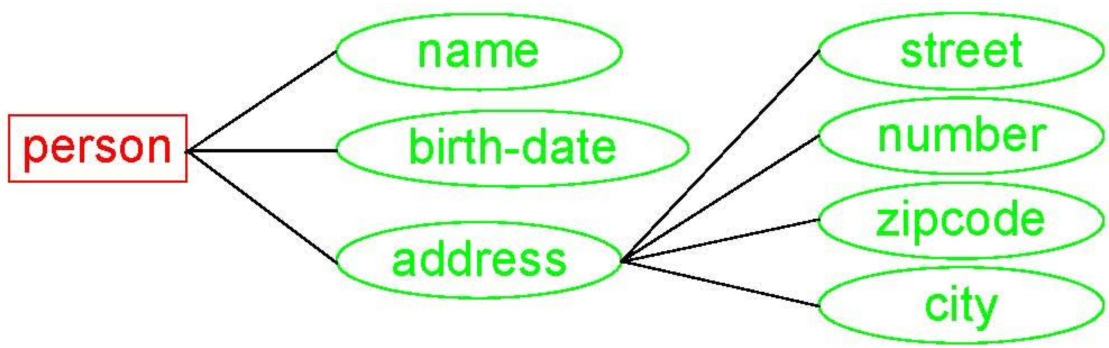
Multivalued Attributes

- Optional attribute: Minimal cardinality is equal to 0
- Simple attribute: Cardinality is equal to 1
- Prescribed attribute: Minimal cardinality is equal to 1
- Multivalued attribute: Maximal cardinality is equal to a natural number *n*
- Example:

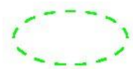


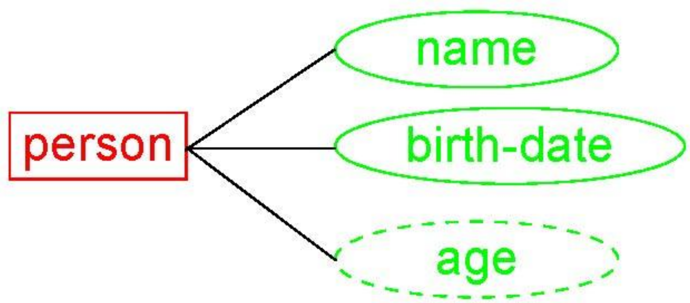
Composite Attributes

- Grouping of attributes of the same entity set or relationship set which are closely related
- Hierarchy of closely related attributes
- Antonym: simple attribute
- Example:



Derived Attributes

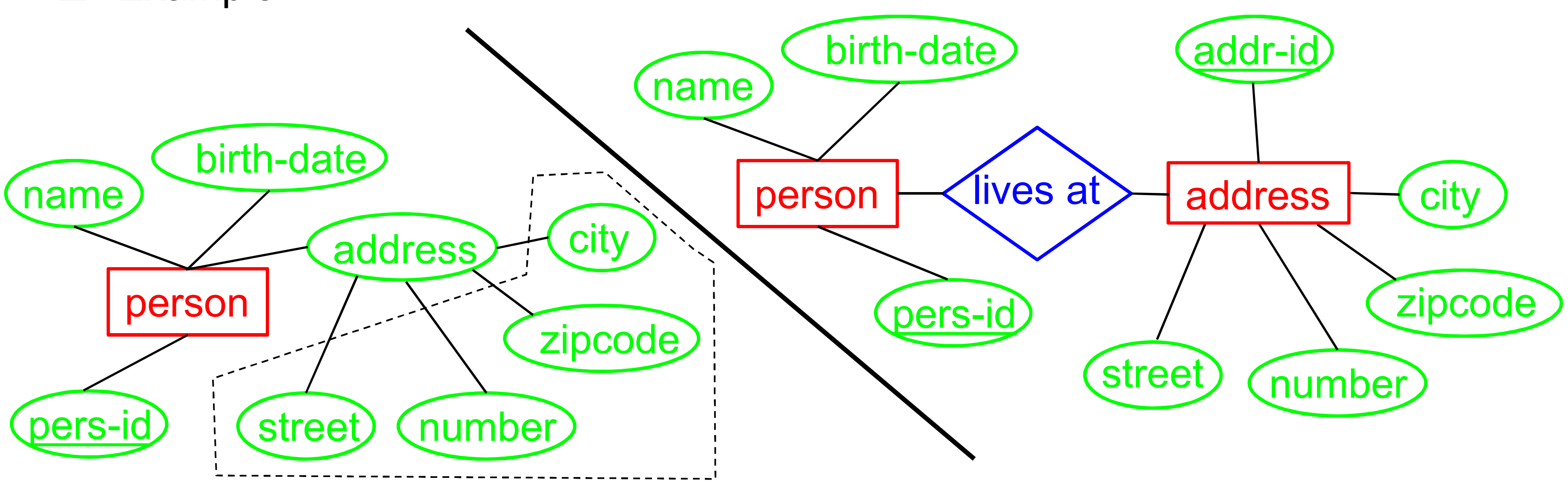
- Attributes that can be derived from one or more attributes
- Antonym: base/stored attribute
- Graphical representation: 
- Example:



- If used in databases
  - They have to be updated regularly, which can be an expensive operation
  - It is better to avoid them and compute them if needed
  - In the example above, the *age* attribute can be derived from the difference of the *system date* and the *birthdate*

Entity Set versus Attribute (I)

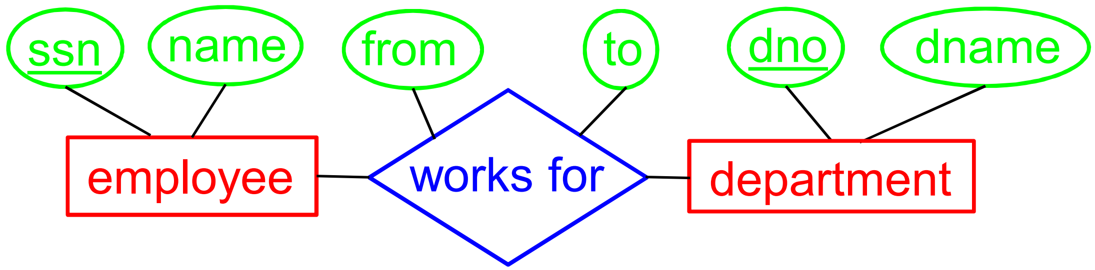
- Should *address* be an attribute of *person* or an own entity set connected to *person* by a relationship set?
- Example



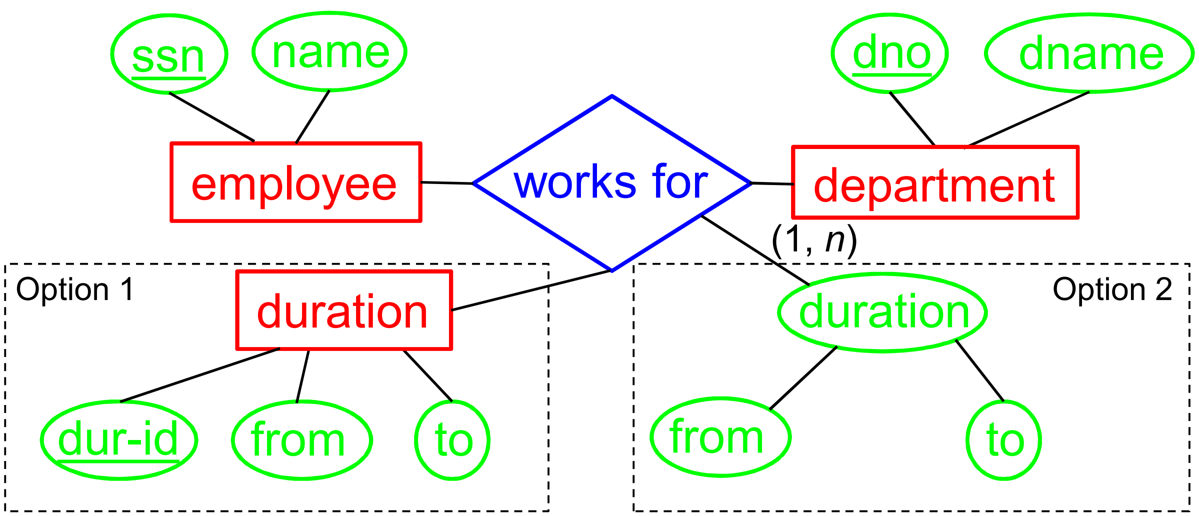
- Both ways are feasible but the attribute variant can introduce redundancy if several persons live at the same address
- For the entity variant it makes sense to introduce an identifier for entity set *address*

Entity Set versus Attribute (II)

- Similar situation for relationship sets: Employee works in a department for a *single* period



- Several values of the descriptive attributes for each instance of this relationship: Employee works in a department for *several* periods

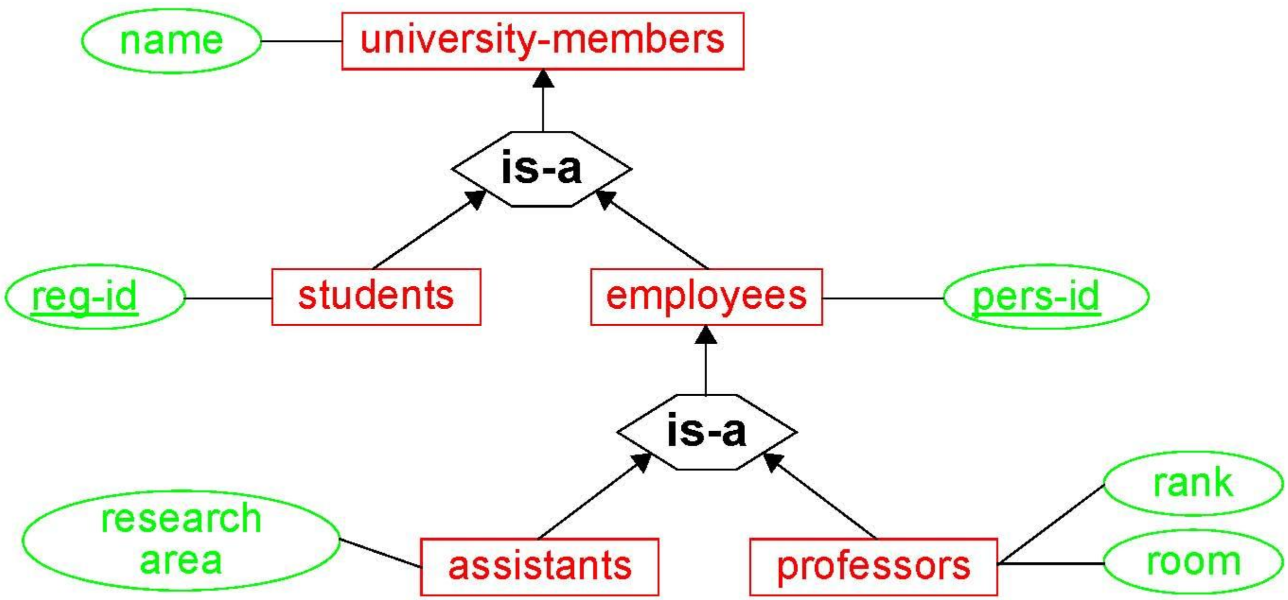


Generalization (I)

- Objectives
  - *Abstraction at the set level* in order not to overwhelm with details but to achieve a more understandable and more concise structuring of entity sets
  - *Abstraction at the instance level* in order to model similar entities by a common entity set
- "Factoring" (extracting) properties (attributes, relationships) of similar entity sets (subclasses, subtypes, categories) to a common superclass (supertype)
- Properties that cannot be extracted remain with the respective subclass, i.e., the subclass is a specialization of the superclass
- Inheritance as the key concept of generalization: A subclass inherits all properties of its superclass
- Entities of a subclass are implicitly considered as entities of the superclass, therefore the relationship is-a is used in the E-R diagrams

Generalization (II)

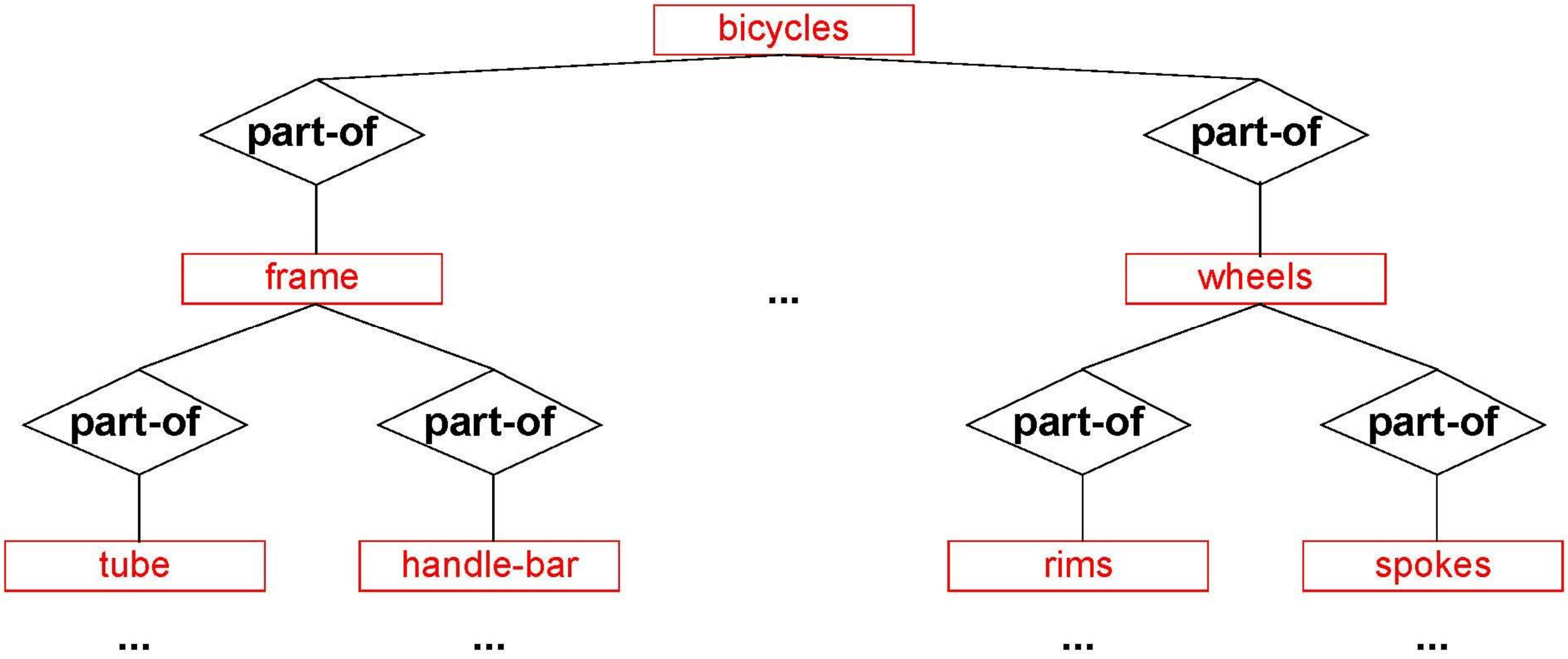
- Example: Conceptual university schema with generalizations



- Two special cases of specialization
  - Disjoint/overlapping specialization: All subclasses of a superclass are pairwise disjoint/overlapping
  - Total specialization: The superclass does not contain explicit elements, but is only given by the union of its subclasses (antonym: partial specialization)

Aggregation

- Objective: Distinct entity sets that together form a “structured” superclass are associated with each other
- An aggregation is a special relationship set named part-of which associates a superior entity set with several subordinate entity sets based on structural composition
- Example: Construction of a bicycle



Main Steps for Designing E-R Diagrams in the Small

An Approach how to design an E-R diagram in the small

- Step 1: Identify (weak) entity sets
- Step 2: Draw (weak) entity sets
- Step 3: Identify (key) attributes of (weak) entity sets
- Step 4: Draw (key) attributes of (weak) entity sets
- Step 5: Identify (identifying) relationship sets
- Step 6: Draw (identifying) relationship sets
- Step 7: Identify attributes of (identifying) relationship sets
- Step 8: Draw attributes of (identifying) relationship sets
- Step 9: Insert cardinalities

Case Study: Gym Management (I)

Suppose that Alachua County wants to centrally control the gyms of the area, and towards this goal you are hired to design a database to store information about the gyms and their employees. The following information is available:

- Each gym has a name, street number, street name, ZIP code, and one or more phone numbers. The gym names are unique.
- An employee is uniquely defined by her SSN and has a name.
- An employee may work at several gyms of Alachua County.
- For every employee the percentage of time is recorded he or she works at each gym.
- Some employees may work in zero or one of the following specializations: manager, receptionist, or personal trainer.
- Every manager manages one or more gyms.
- Each gym has exactly one manager.
- For a personal trainer we also store the zero or one type of certification he/she has. Some examples of certification are yoga, aerobics, and sports nutrition.
- The information stored for a customer is: SSN (unique), name, age.
- Each customer may be going to more than one gym.
- The gyms also allow each customer to have guests/friends associated with him/her. These guests can use the facilities of the gyms their host goes to. The guests are not considered customers of the gym. Only the name and age of the guests are stored in the database, and we assume that for each customer the pair (guest_name, guest_age) is unique.

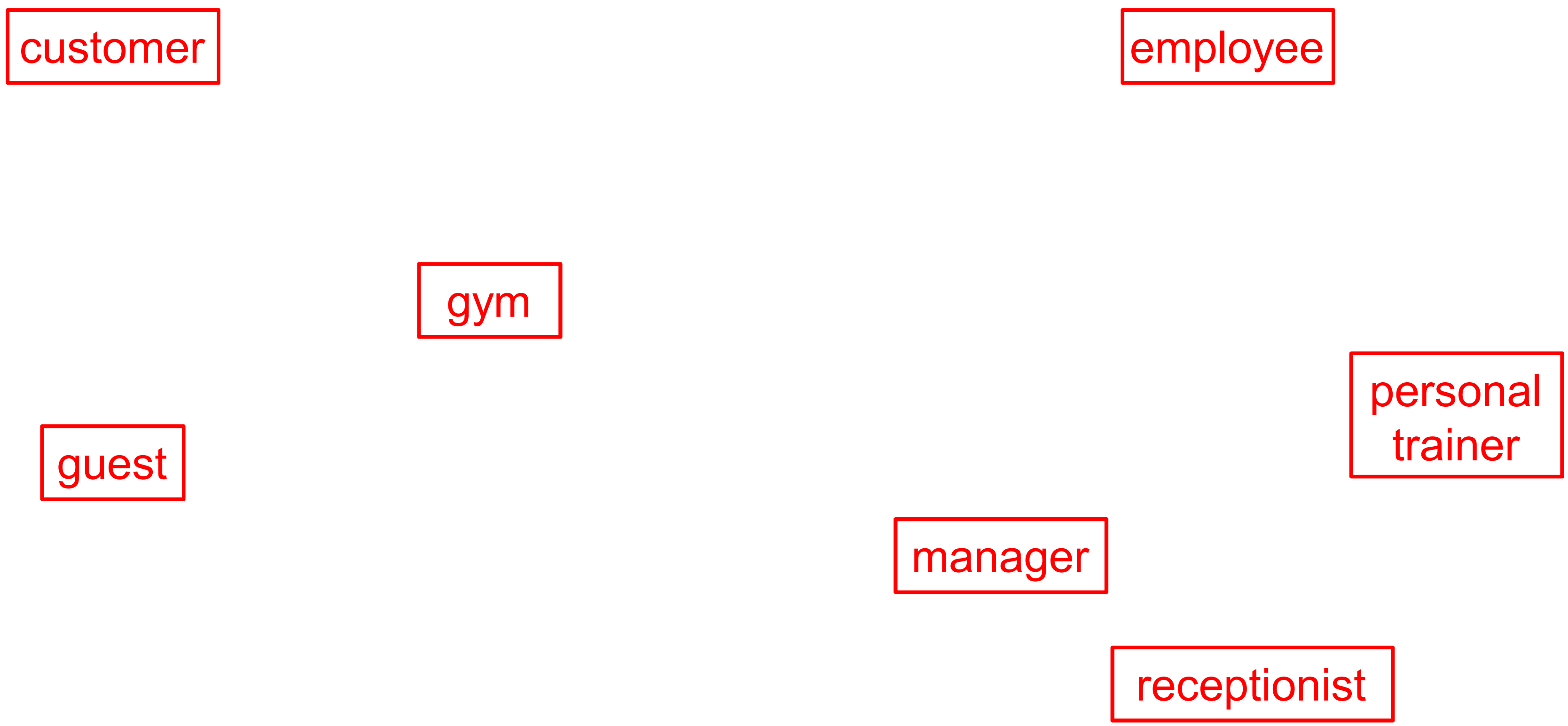
Case Study: Gym Management (II)

Step 1: Identify (weak) entity sets

- Each gym has a name, street number, street name, ZIP code, and one or more phone numbers. The gym names are unique.
- An employee is uniquely defined by her SSN and has a name.
- An employee may work at several gyms of Alachua County.
- For every employee the percentage of time is recorded he or she works at each gym.
- Some employees may work in zero or one of the following specializations: manager, receptionist, or personal trainer.
- Every manager manages one or more gyms.
- Each gym has exactly one manager.
- For a personal trainer we also store the zero or one type of certification he/she has. Some examples of certification are yoga, aerobics, and sports nutrition.
- The information stored for a customer is: SSN (unique), name, age.
- Each customer may be going to more than one gym.
- The gyms also allow each customer to have guests/friends associated with him/her. These guests can use the facilities of the gyms their host goes to. The guests are not considered customers of the gym. Only the name and age of the guests are stored in the database, and we assume that for each customer the pair (guest_name, guest_age) is unique.

Case Study: Gym Management (III)

Step 2: Draw (weak) entity sets



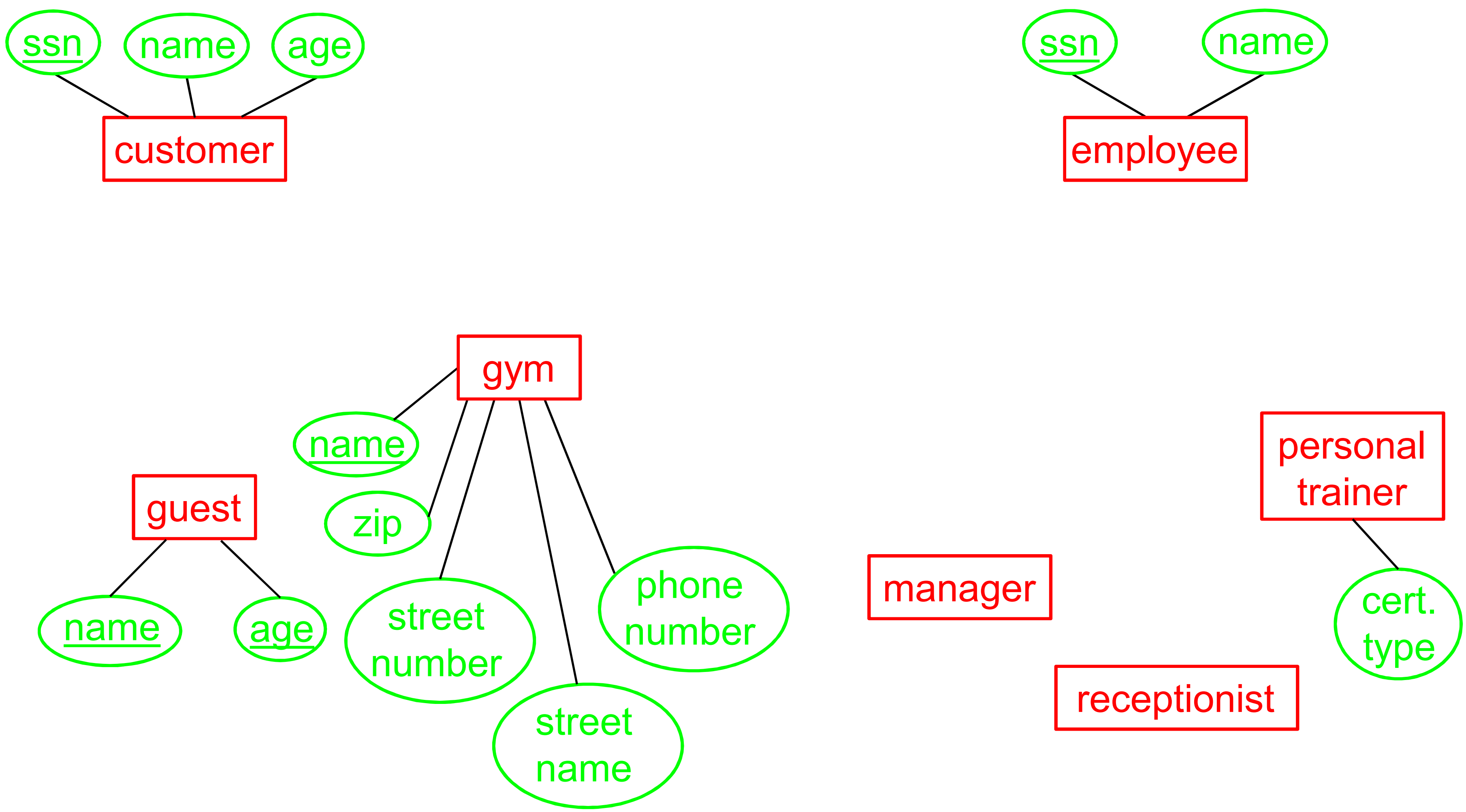
Case Study: Gym Management (IV)

Step 3: Identify (key) attributes of (weak) entity sets

- Each gym has a name, street number, street name, ZIP code, and one or more phone numbers. The gym names are unique.
- An employee is uniquely defined by her SSN and has a name.
- An employee may work at several gyms of Alachua County.
- For every employee the percentage of time is recorded he or she works at each gym.
- Some employees may work in zero or one of the following specializations: manager, receptionist, or personal trainer.
- Every manager manages one or more gyms.
- Each gym has exactly one manager.
- For a personal trainer we also store the zero or one type of certification he/she has. Some examples of certification are yoga, aerobics, and sports nutrition.
- The information stored for a customer is: SSN (unique), name, age.
- Each customer may be going to more than one gym.
- The gyms also allow each customer to have guests/friends associated with him/her. These guests can use the facilities of the gyms their host goes to. The guests are not considered customers of the gym. Only the name and age of the guests are stored in the database, and we assume that for each customer the pair (guest_name, guest_age) is unique.

Case Study: Gym Management (V)

Step 4: Draw (key) attributes of (weak) entity sets



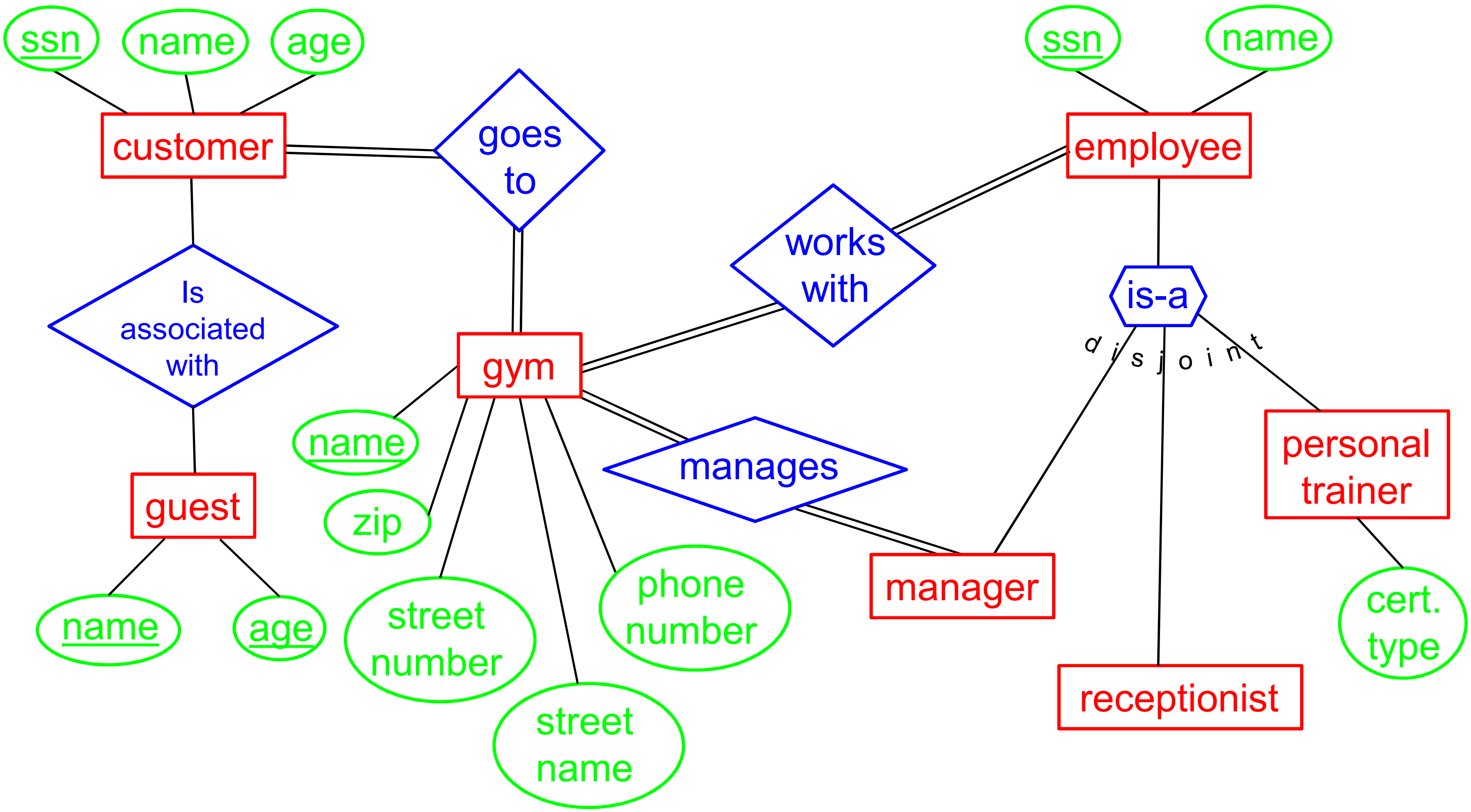
Case Study: Gym Management (VI)

Step 5: Identify (identifying) relationship sets

- Each gym has a name, street number, street name, ZIP code, and one or more phone numbers. The gym names are unique.
- An employee is uniquely defined by her SSN and has a name.
- An employee may work at several gyms of Alachua County.
- For every employee the percentage of time is recorded he or she works at each gym.
- Some employees may work in zero or one of the following specializations: manager, receptionist, or personal trainer.
- Every manager manages one or more gyms.
- Each gym has exactly one manager.
- For a personal trainer we also store the zero or one type of certification he/she has. Some examples of certification are yoga, aerobics, and sports nutrition.
- The information stored for a customer is: SSN (unique), name, age.
- Each customer may be going to more than one gym.
- The gyms also allow each customer to have guests/friends associated with him/her. These guests can use the facilities of the gyms their host goes to. The guests are not considered customers of the gym. Only the name and age of the guests are stored in the database, and we assume that for each customer the pair (guest_name, guest_age) is unique.

Case Study: Gym Management (VII)

Step 6: Draw (identifying) relationship sets



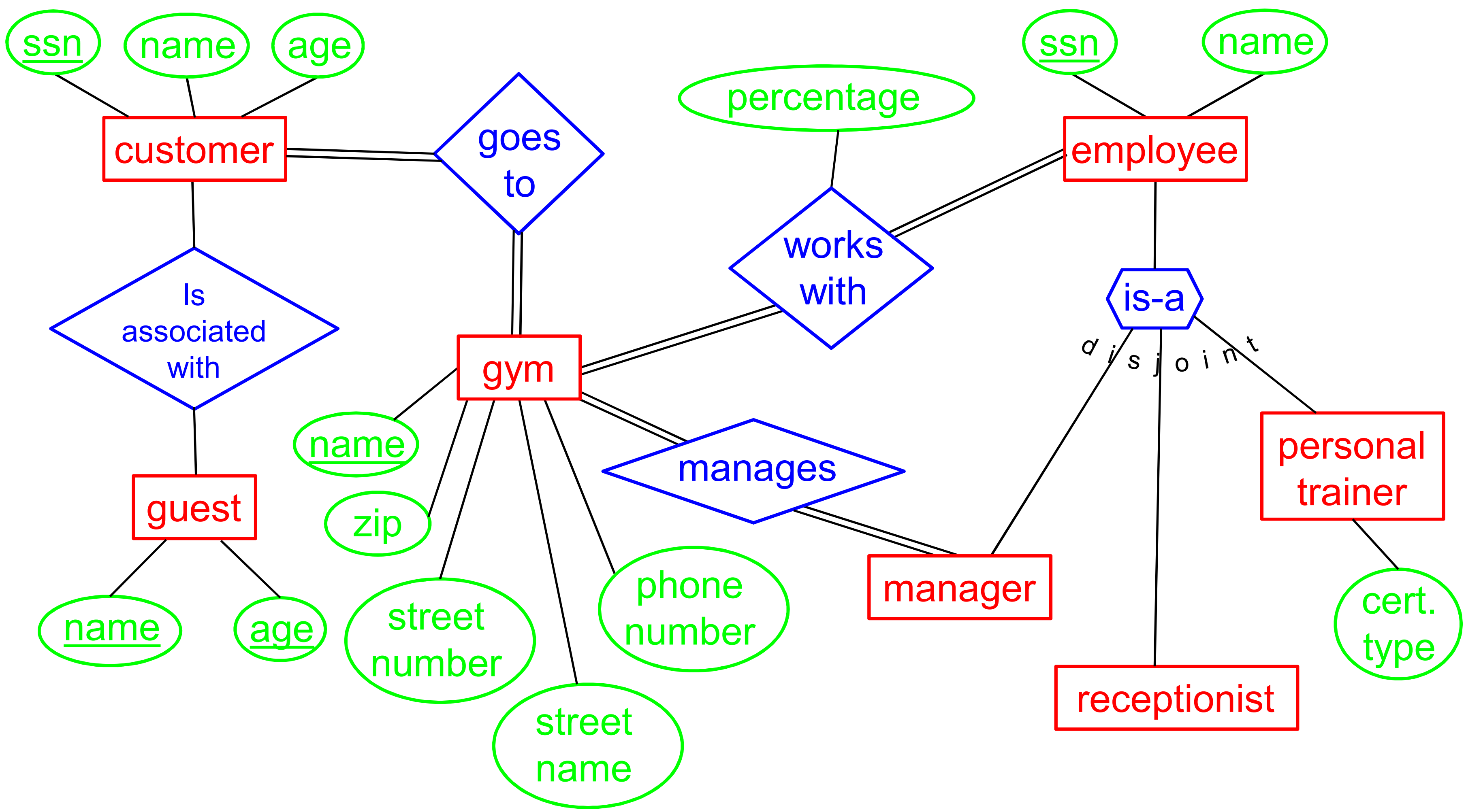
Case Study: Gym Management (VIII)

Step 7: Identify attributes of (identifying) relationship sets

- Each gym has a name, street number, street name, ZIP code, and one or more phone numbers. The gym names are unique.
- An employee is uniquely defined by her SSN and has a name.
- An employee may work at several gyms of Alachua County.
- For every employee the percentage of time is recorded he or she works at each gym.
- Some employees may work in zero or one of the following specializations: manager, receptionist, or personal trainer.
- Every manager manages one or more gyms.
- Each gym has exactly one manager.
- For a personal trainer we also store the zero or one type of certification he/she has. Some examples of certification are yoga, aerobics, and sports nutrition.
- The information stored for a customer is: SSN (unique), name, age.
- Each customer may be going to more than one gym.
- The gyms also allow each customer to have guests/friends associated with him/her. These guests can use the facilities of the gyms their host goes to. The guests are not considered customers of the gym. Only the name and age of the guests are stored in the database, and we assume that for each customer the pair (guest_name, guest_age) is unique.

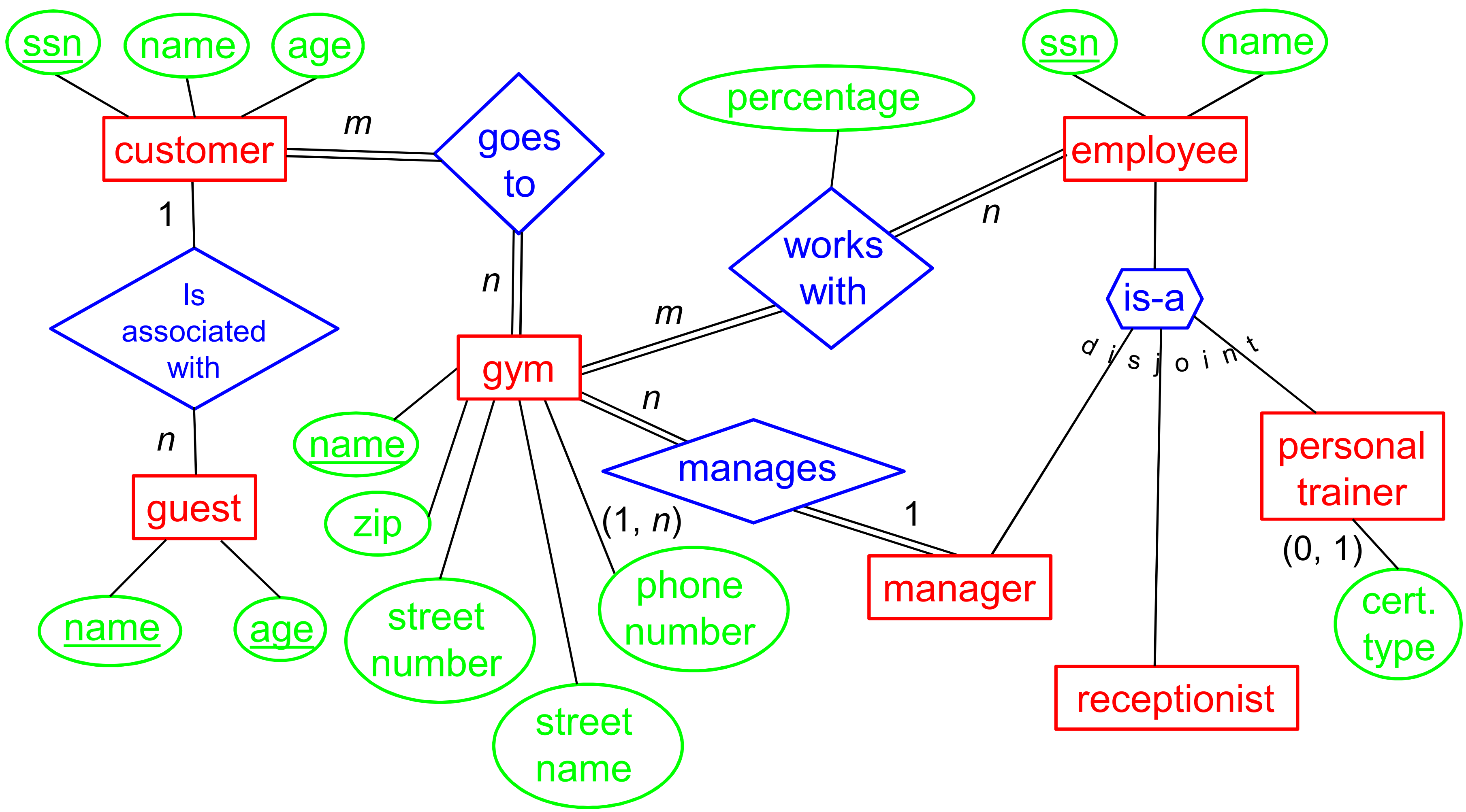
Case Study : Gym Management (IX)

Step 8: Draw attributes of (identifying) relationship sets



Case Study: Gym Management (X)

Step 9: Insert cardinalities

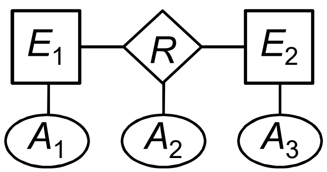


Relational Data Model

The Big Picture (I)

Entity-Relationship Model

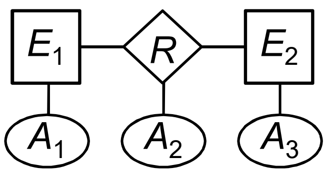
Entities, attributes, relationships



The Big Picture (II)

Entity-Relationship Model

Entities, attributes, relationships



Relational Data Model

Relations

$$
\begin{aligned}
R(&A_1 : D_1,\\
&A_2 : D_2,\\
&\ldots,\\
&A_n : D_n)
\end{aligned}
$$