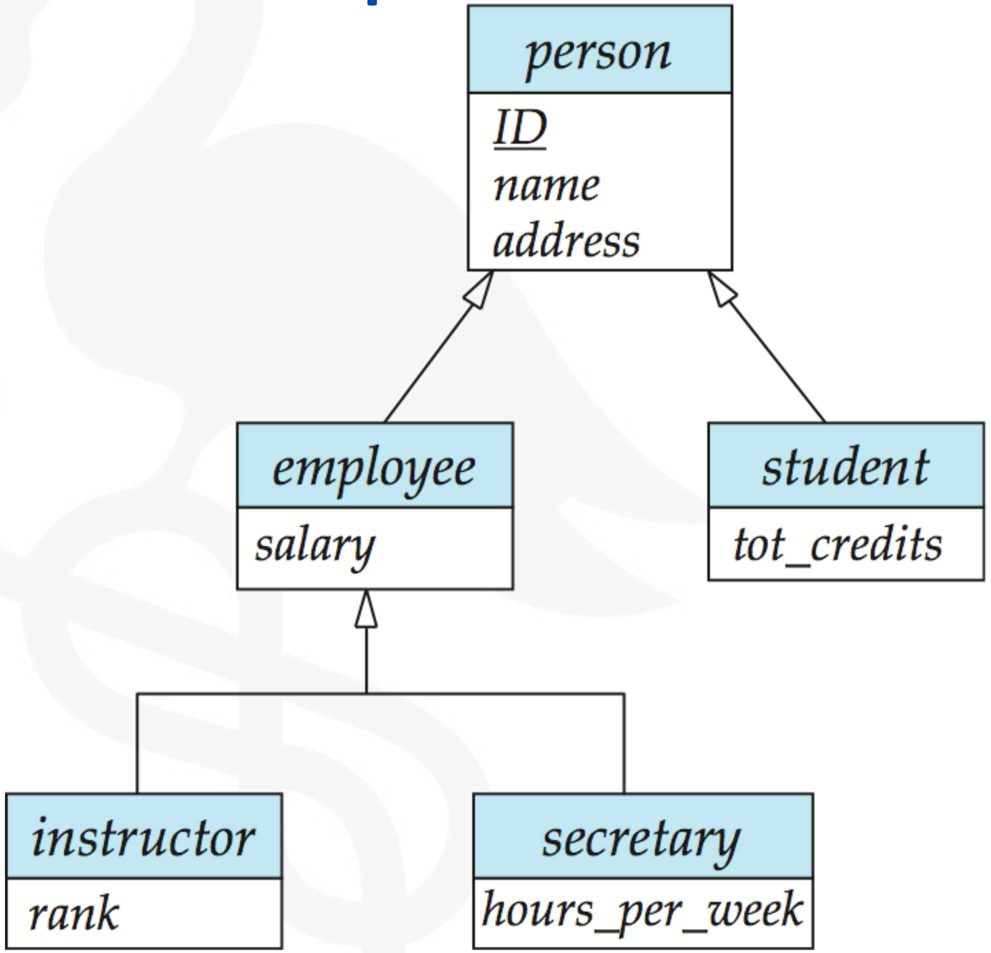
Extended Entity-Relationship Model

Extended E-R Features: Specialization

- Top-down design process; we designate subgroupings within an entity set that are distinctive from other entities in the set.
- These subgroupings become lower-level entity sets that have attributes or participate in relationships that do not apply to the higher-level entity set.
- Depicted by a *triangle* component labeled ISA (E.g., *instructor* “is a” *person*).
- **Attribute inheritance** – a lower-level entity set inherits all the attributes of the higher-level entity set to which it is linked.

Specialization Example

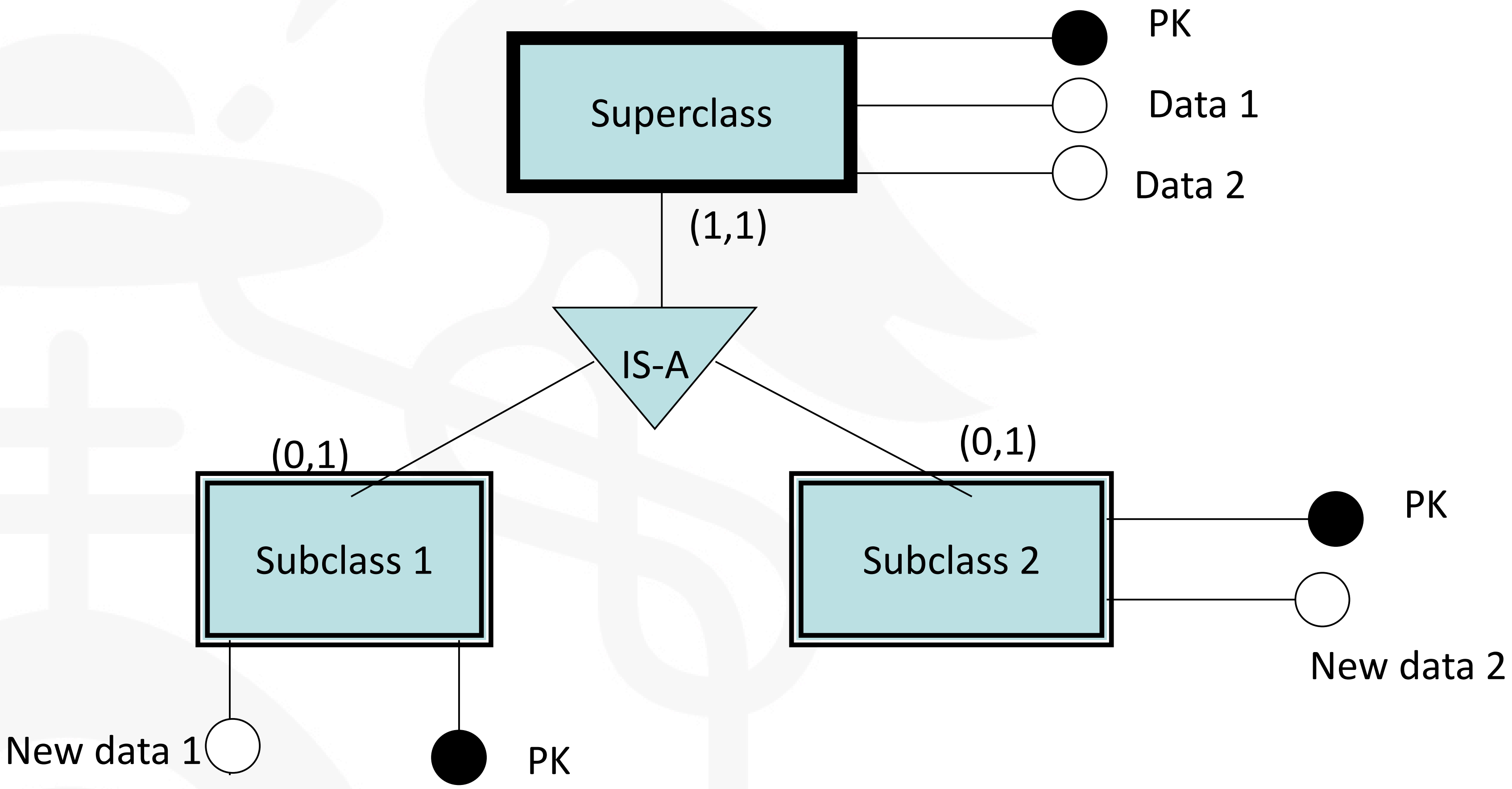


Extended ER Features: Generalization

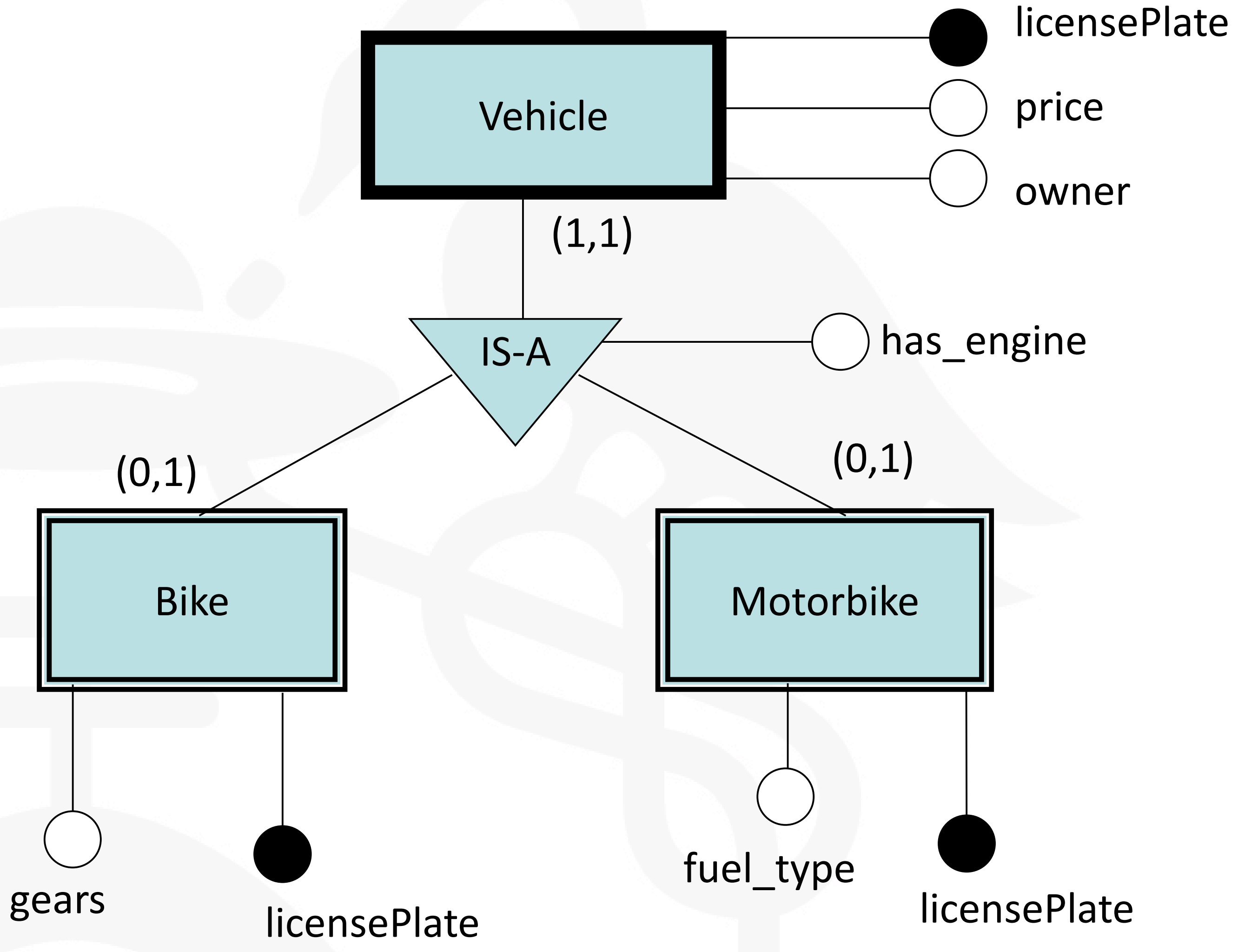
- **A bottom-up design process** – combine a number of entity sets that share the same features into a higher-level entity set.
- Specialization and generalization are simple inversions of each other; they are represented in an E-R diagram in the same way.
- The terms specialization and generalization are used interchangeably.
- The ISA relationship is also referred to as **superclass - subclass** relationship

Specialization and Generalization (Cont.)

- Specialization is represented by triangle labelled IS-A



Specialization and Generalization (Cont.)



The attribute has_engine belongs to the Vehicle entity, but is drawn in the hierarchy symbol. Its value is inherited by the subclasses.

Design Constraints

- Constraint on whether or not entities may belong to more than one lower-level entity set.
  - **Disjoint**
    - an entity can belong to only one lower-level entity set (person is a man or a woman)
  - **Overlapping**
    - an entity can belong to more than one lower-level entity set (a student can be a football player and a baseball player)

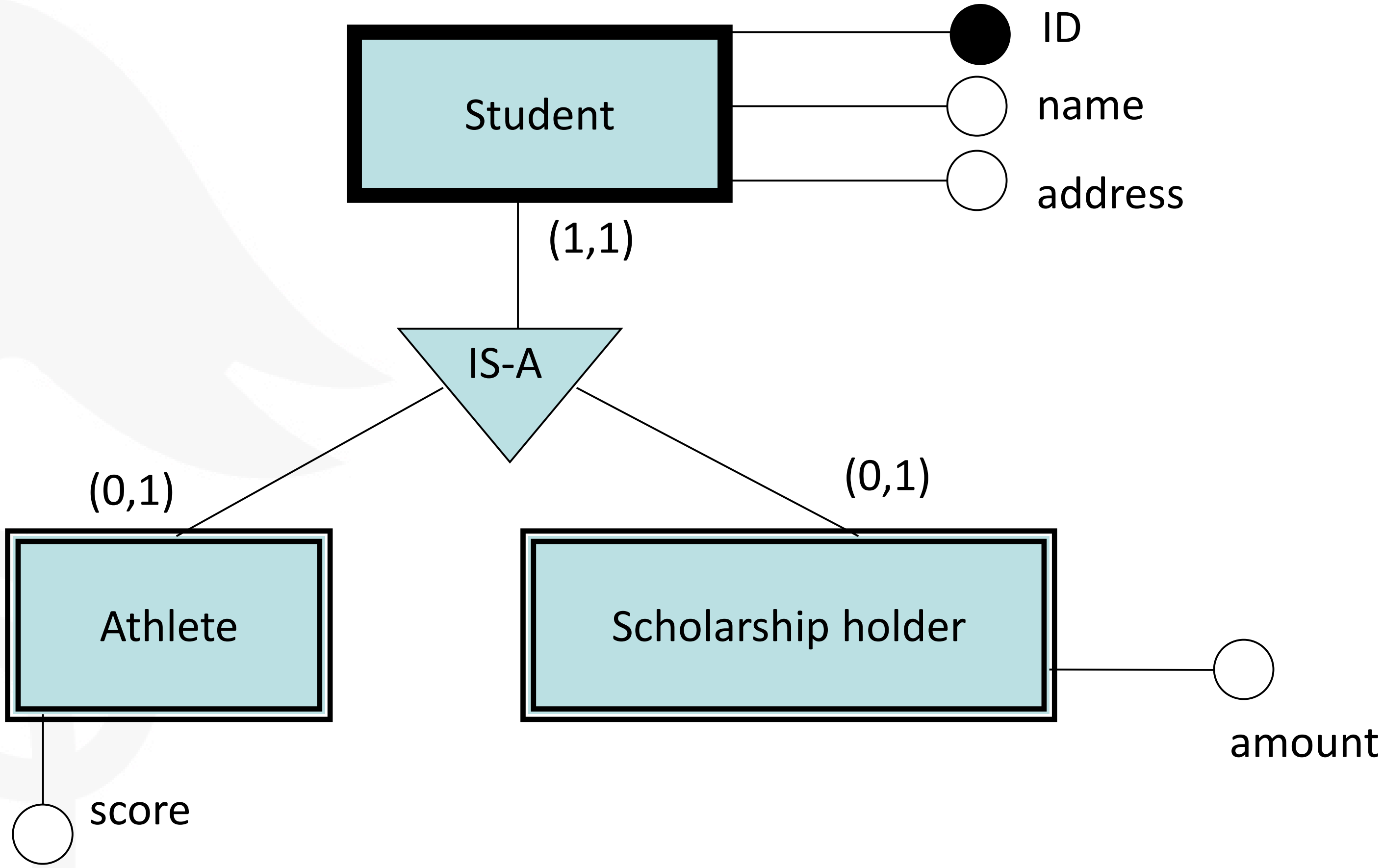
Design Constraints

- Constraint on whether or not an entity in the higher-level entity set must belong to at least one of the lower-level entity sets.
  - **total**: an entity must belong to one of the lower-level entity sets (every person is a man or a woman)
  - **partial**: an entity need not belong to one of the lower-level entity sets (a student is a football player or a baseball player. Not all students play sports)

Specialization/Generalization Examples

Partial and overlapping

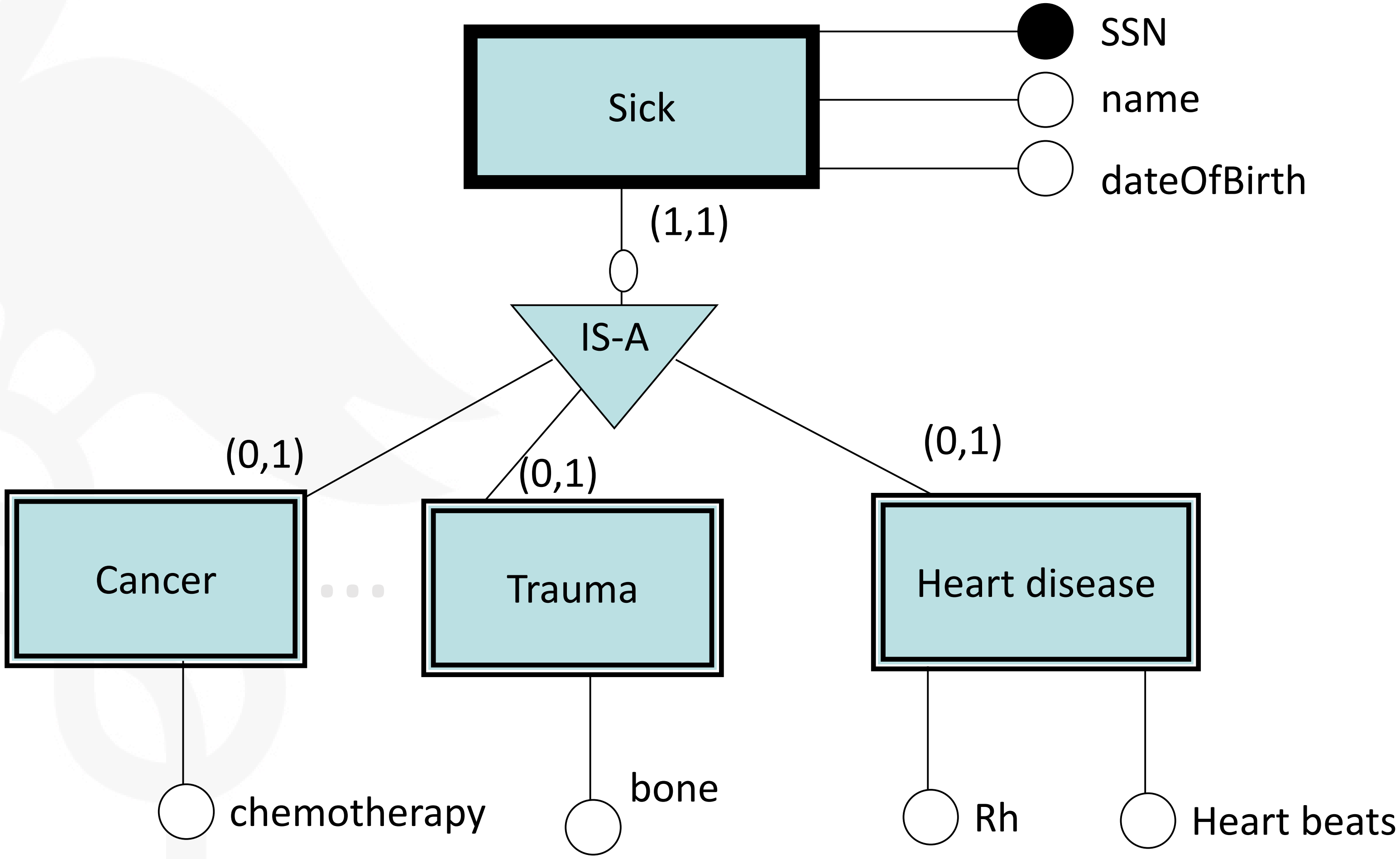
- Partial because there are students that aren’t athletes nor scholarship holders
- Overlapping because a student can be an athlete and a scholarship holder at the same time.



Specialization/Generalization Examples

Total and overlapping

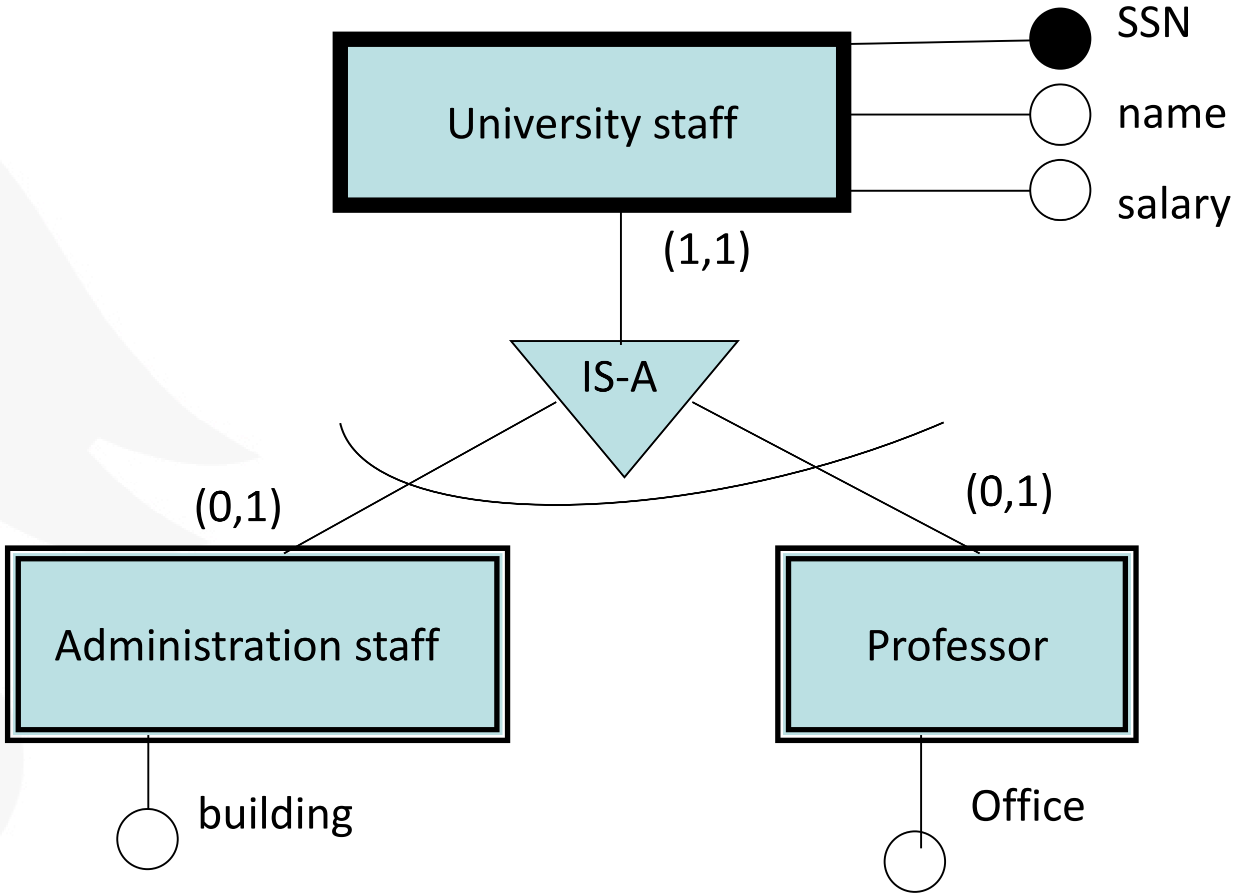
- Total because a sick person has some type of disease
- Overlapping because a person can have cancer and heart disease at the same time



Specialization/Generalization Examples

Disjoint and partial

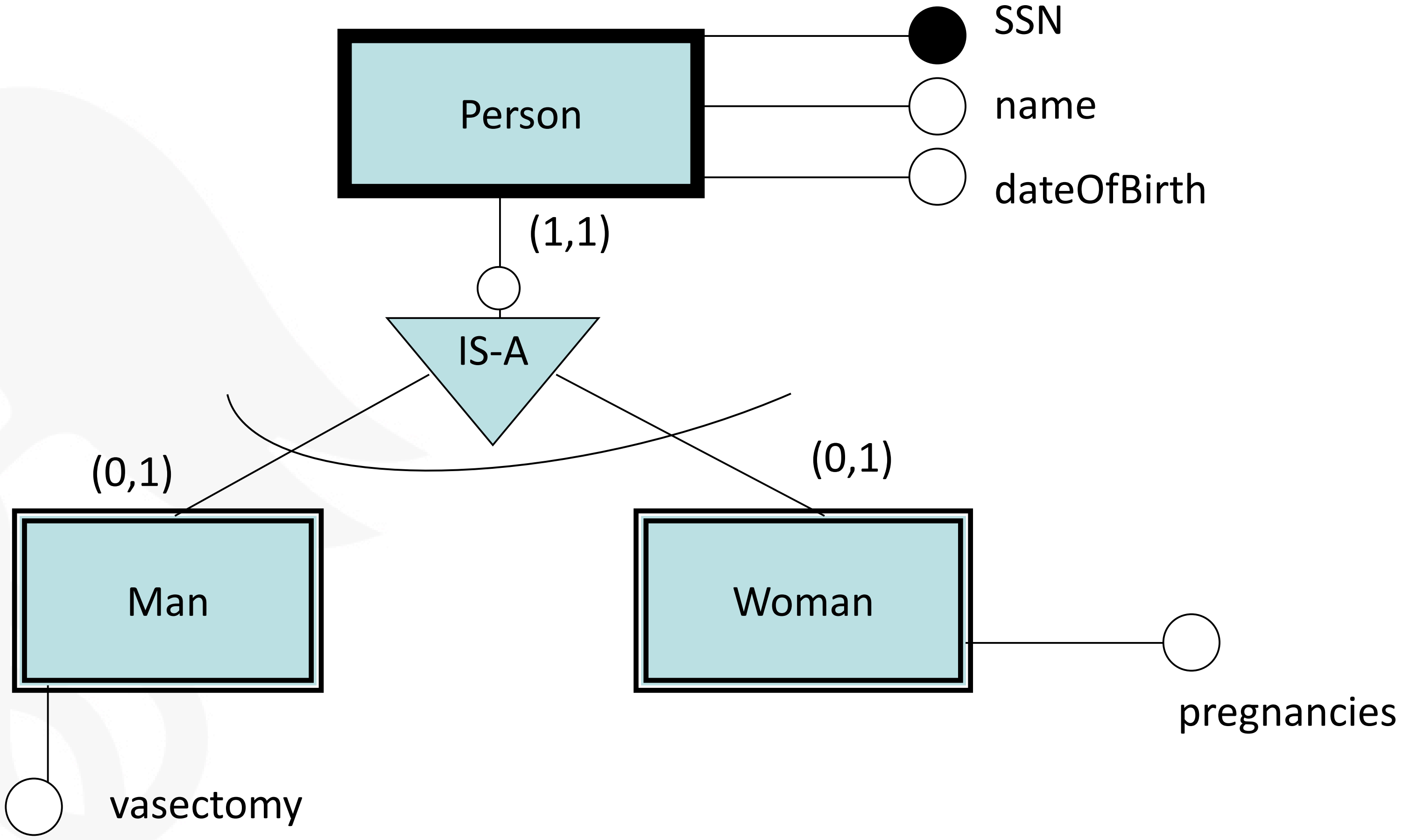
- Disjoint because a person who works in the university can be a professor or can work in the administration of the university. A person can't have both jobs at the same time.
- Partial because there are other employees in the university (cleaners, concierges, etc.)



Specialization/Generalization Examples

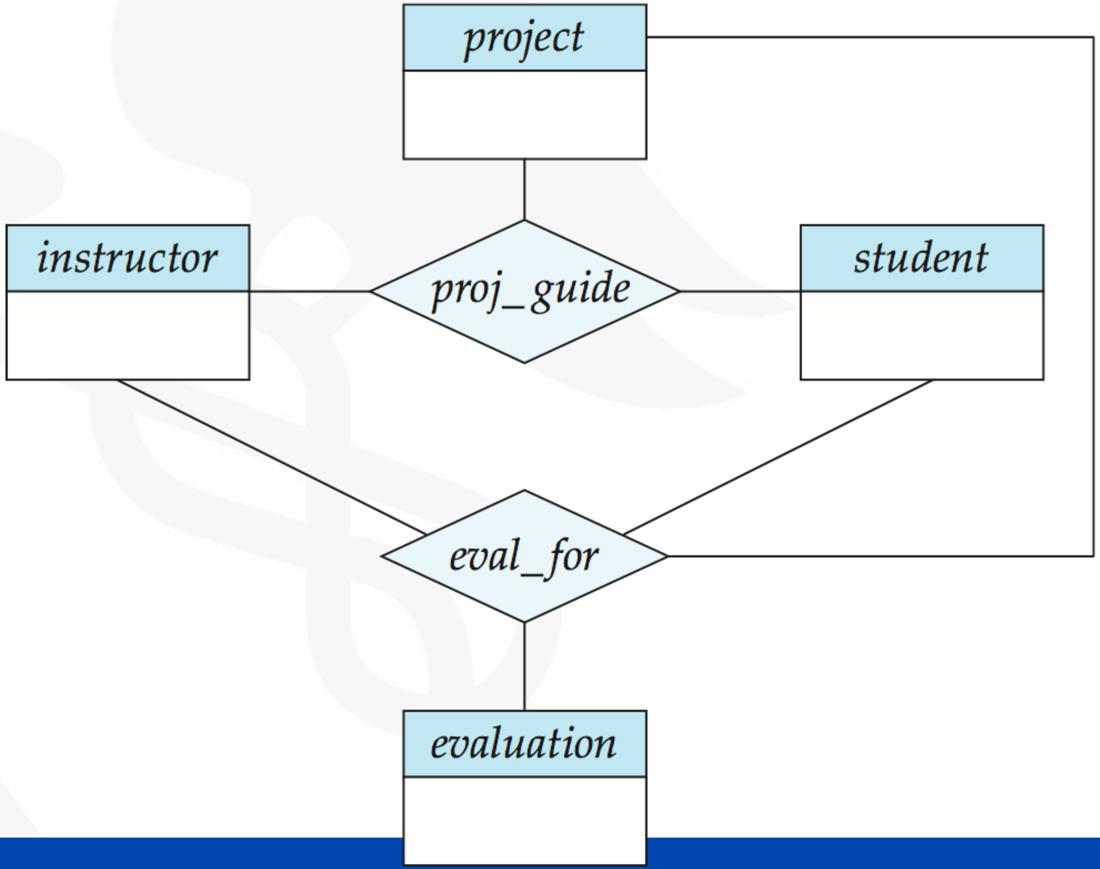
Disjoint and total

- ○ Disjoint because a person is a man or is a woman.
- ○ Total because every person is a man or a woman.



Aggregation

- Consider the ternary relationship *proj_guide*
- Suppose we want to record evaluations of a student by a guide on a project

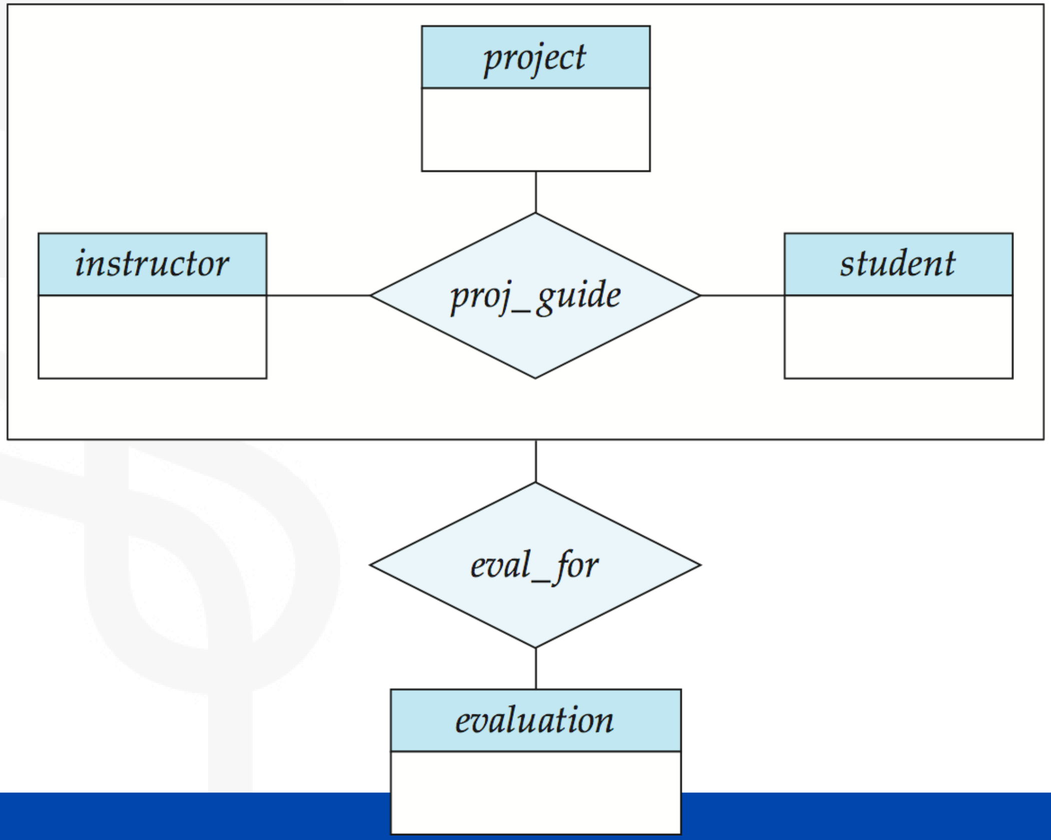


Aggregation (Cont.)

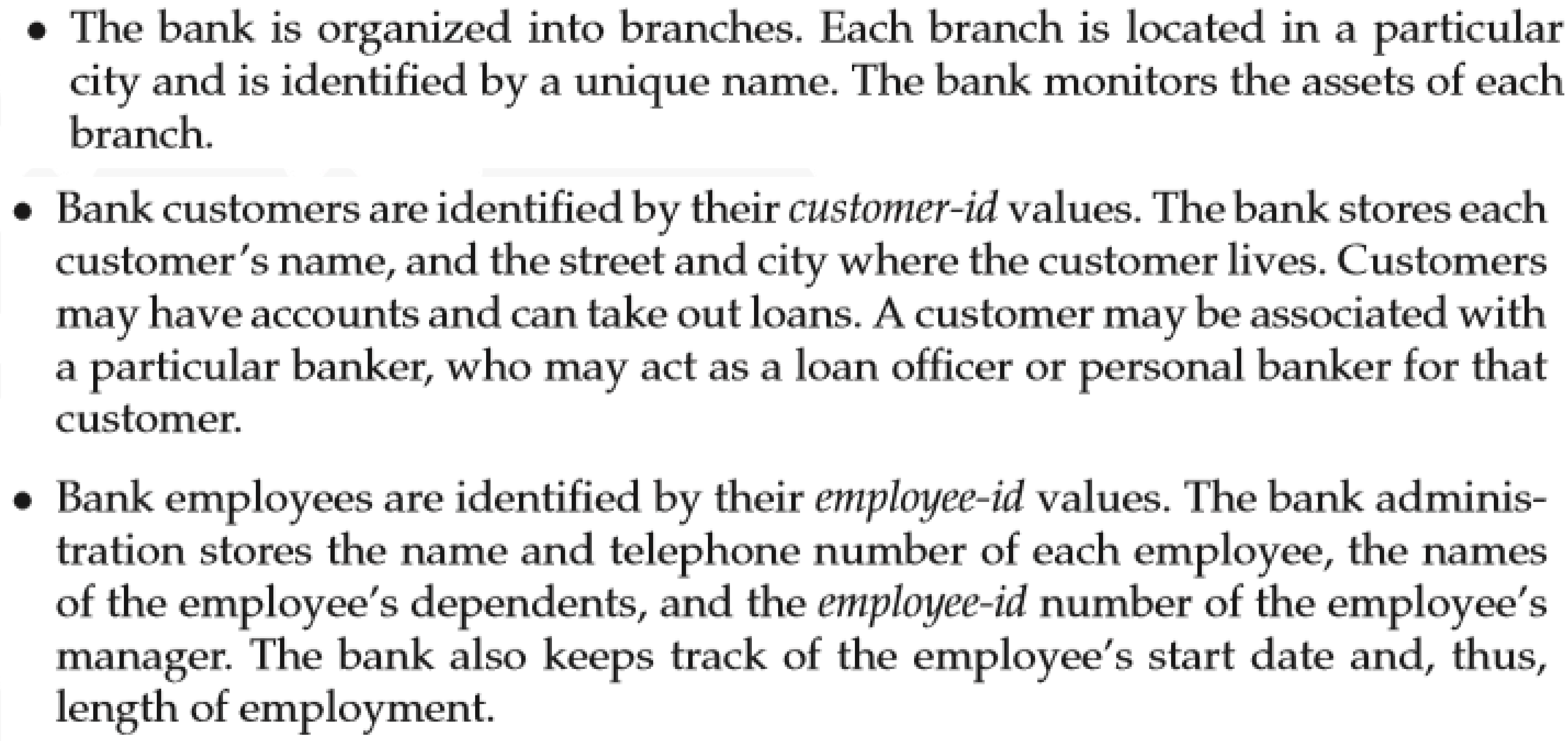
- Relationship sets *eval_for* and *proj_guide* represent overlapping information
  - Every *eval_for* relationship corresponds to a *proj_guide* relationship
  - However, some *proj_guide* relationships may not correspond to any *eval_for* relationships
    - So we can't discard the *proj_guide* relationship
- We eliminate this redundancy via *aggregation*
- Aggregation is an abstraction through which relationships are treated as higher-level entities.

Aggregation (Cont.)

- Without introducing redundancy, the following diagram represents:
  - A student is guided by a particular instructor on a particular project
  - A student, instructor, project combination may have an associated evaluation



Exercises

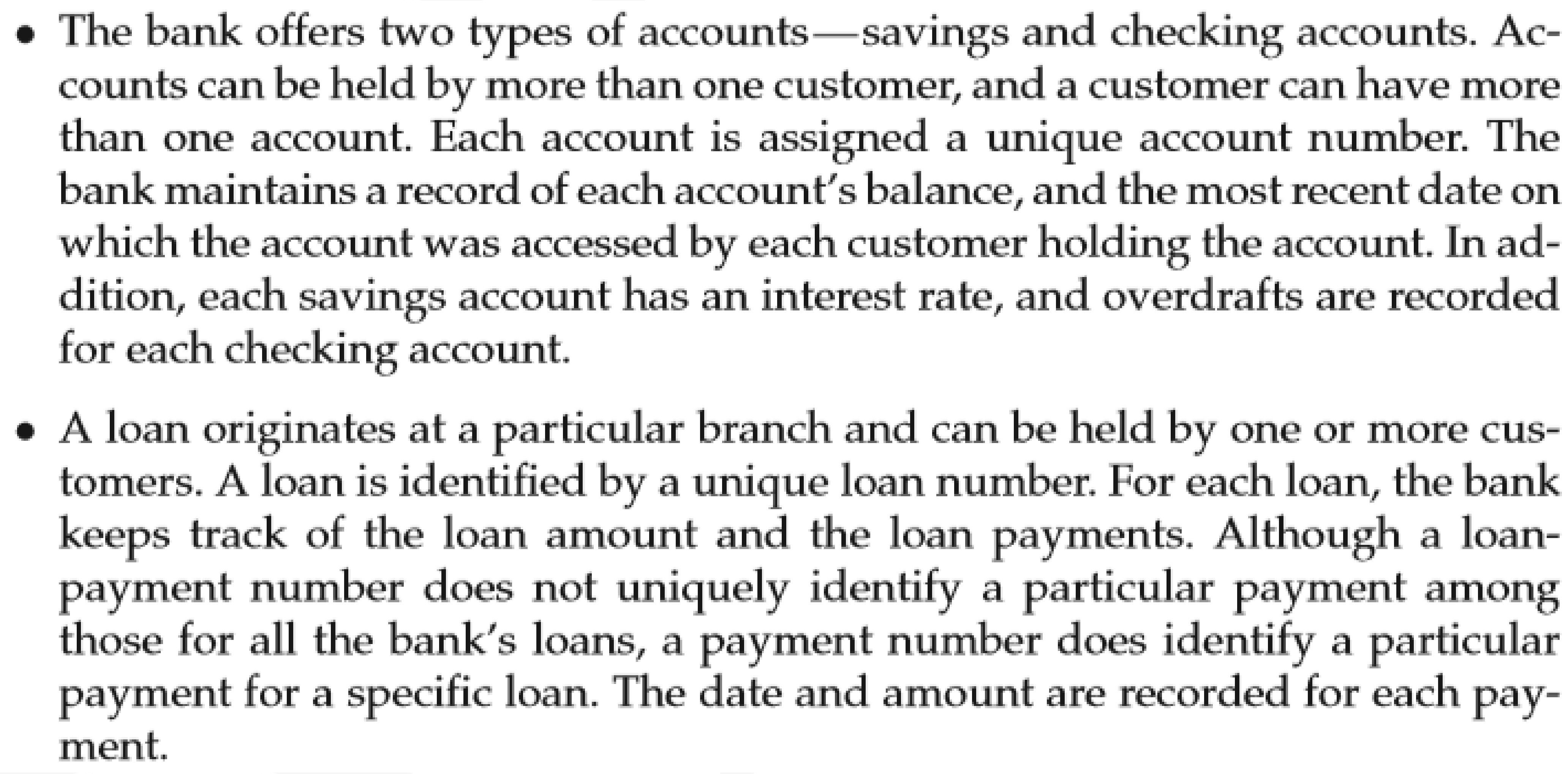


- The bank is organized into branches. Each branch is located in a particular city and is identified by a unique name. The bank monitors the assets of each branch.

- Bank customers are identified by their *customer-id* values. The bank stores each customer's name, and the street and city where the customer lives. Customers may have accounts and can take out loans. A customer may be associated with a particular banker, who may act as a loan officer or personal banker for that customer.

- Bank employees are identified by their *employee-id* values. The bank administration stores the name and telephone number of each employee, the names of the employee's dependents, and the *employee-id* number of the employee's manager. The bank also keeps track of the employee's start date and, thus, length of employment.

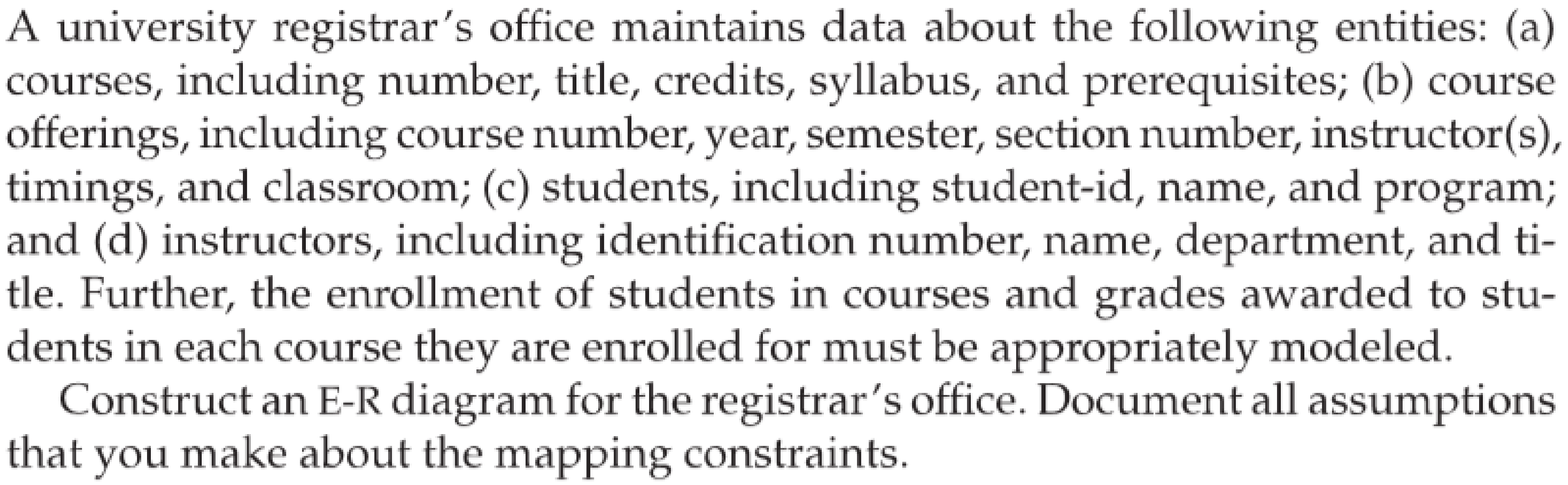
Exercises



- The bank offers two types of accounts—savings and checking accounts. Accounts can be held by more than one customer, and a customer can have more than one account. Each account is assigned a unique account number. The bank maintains a record of each account's balance, and the most recent date on which the account was accessed by each customer holding the account. In addition, each savings account has an interest rate, and overdrafts are recorded for each checking account.

- A loan originates at a particular branch and can be held by one or more customers. A loan is identified by a unique loan number. For each loan, the bank keeps track of the loan amount and the loan payments. Although a loan-payment number does not uniquely identify a particular payment among those for all the bank's loans, a payment number does identify a particular payment for a specific loan. The date and amount are recorded for each payment.

Exercises



A university registrar's office maintains data about the following entities: (a) courses, including number, title, credits, syllabus, and prerequisites; (b) course offerings, including course number, year, semester, section number, instructor(s), timings, and classroom; (c) students, including student-id, name, and program; and (d) instructors, including identification number, name, department, and title. Further, the enrollment of students in courses and grades awarded to students in each course they are enrolled for must be appropriately modeled.

Construct an E-R diagram for the registrar's office. Document all assumptions that you make about the mapping constraints.

Exercises

- A company needs to store in a database the national identity document, the full name, address, email and telephone of every employee. When an employee logs into the company network, a username and password are required.
- The company has a set of computers. Computers are classified into servers and laptops. Only one user can use a laptop at a given time. Several services run in each server.
- Computers have an identifier, a name, a hard disk, a processor, and RAM memory.
- You want to keep track of what services are running in each server. Each service has a name. There are several servers running services with the same name, but a server never runs more than one service with a given name. Services have a name (mail, web, etc) and the date it began to run on the server. Services on different servers must be considered different.

Exercises

- Each laptop has a set of applications installed. Each application has a name, a version and an expiration date of the license. It is clear that two applications should not be considered equal if they have the same name but different version (Word 98 is not the same as Word 2000 ...). In addition, the installation dates must be registered on each laptop.
- For each application it is also interesting to register what services it requires for its execution and what other applications it depends on.
- From each session of a user in a given laptop, it is interesting to record the date and time and what applications he/she used from those installed in that laptop