Entity-Relationship (ER) Diagrams

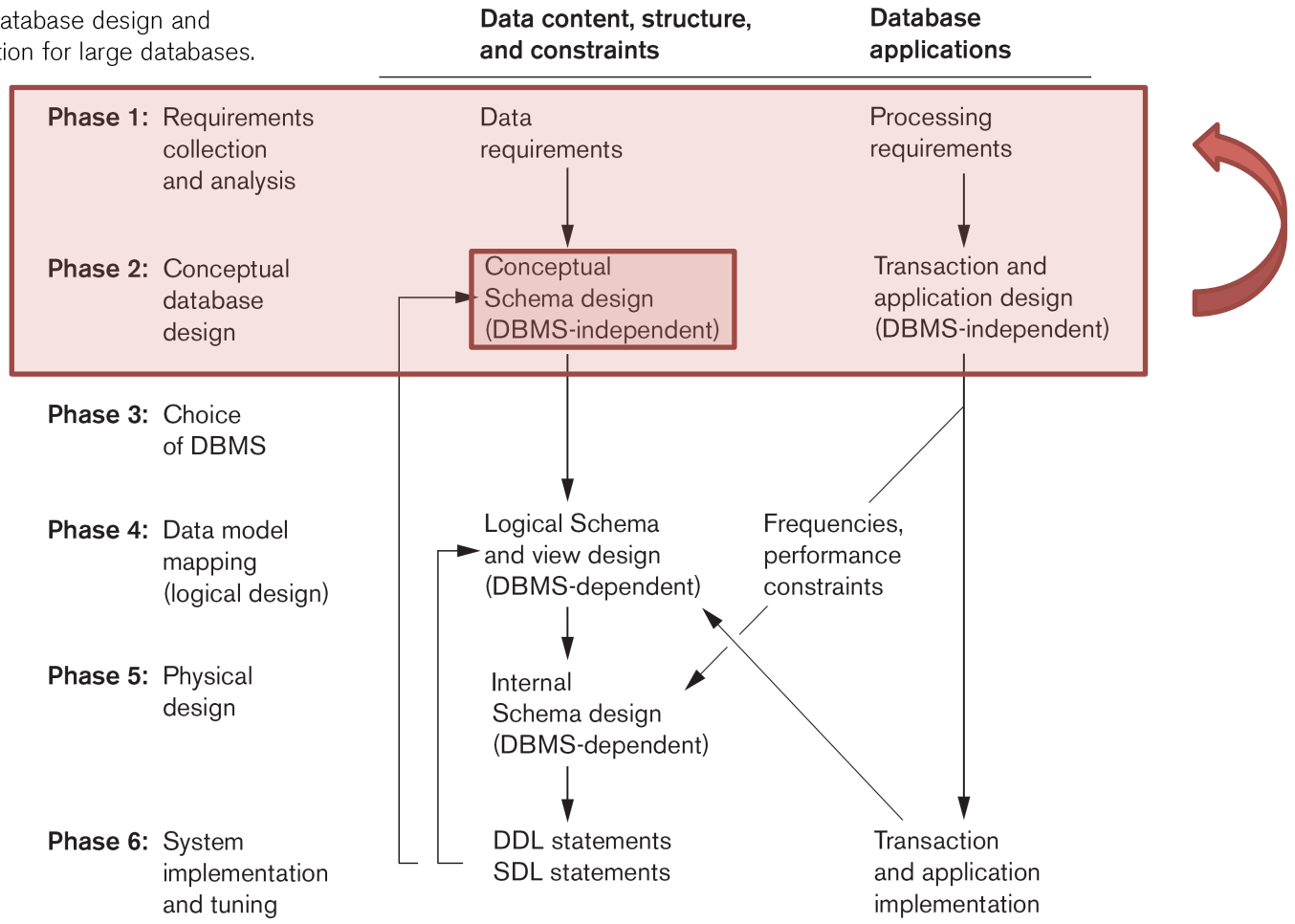
Lecture 7

Outline

1. Context
   - Design & Implementation Process
2. Goals of Conceptual Design
3. Entity-Relationship (ER) Model
4. One ER Diagrammatic Notation
5. Requirements Elicitation
6. Approaches to Conceptual Design

Database Design and Implementation Process

Figure 10.1
Phases of database design and implementation for large databases.



Goal of Conceptual Design

Description of data requirements that is…

Comprehensive
- Entity types, relationships, and constraints
- Sanity check of data & functional requirements
- Reference for [unit/integration] testing/analysis

Concise/High-level
- Easy to understand technically
- Easy to communicate with non-technical users
- Facilitates focus on data (vs. storage/implementation details)

Algorithmically Transformable
- Improves application development efficiency, reduces errors

Entity-Relationship (ER) Model

Entity

- Thing in the real world

Attribute

- Property of an entity
- Most of what we store in the database

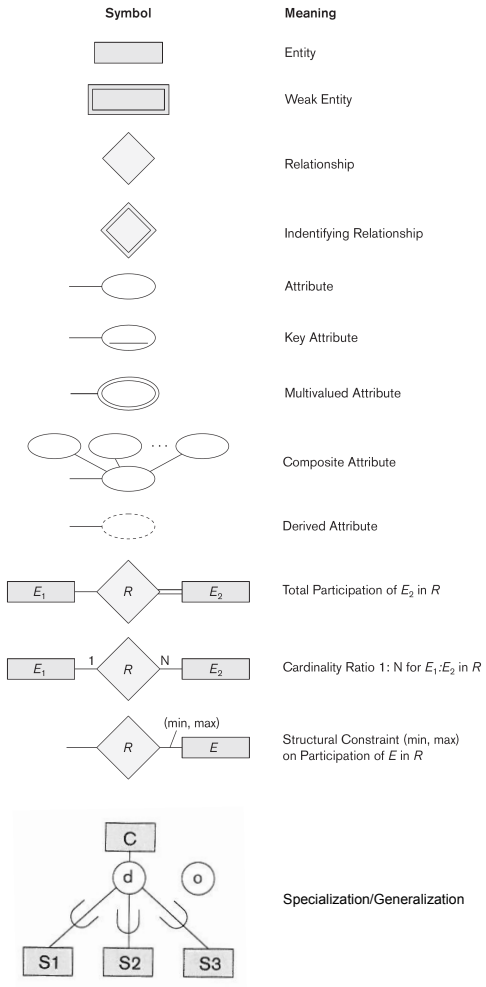
Relationship

- Association between sets of entities
- Possibly with attribute(s)

ER Diagrams

- Graphical depiction of an ER model
- Many notations, this class…

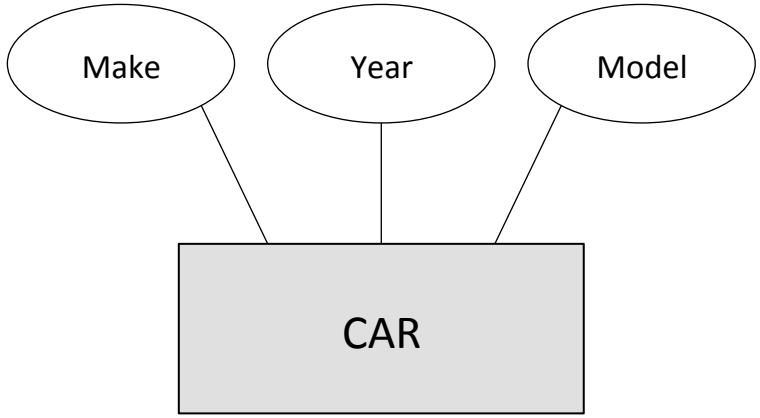
All cars have a year, age, make, model, registration (unique), vehicle number (vin; unique), some number of colors…



Entity Sets

Set of entities that have the same attributes

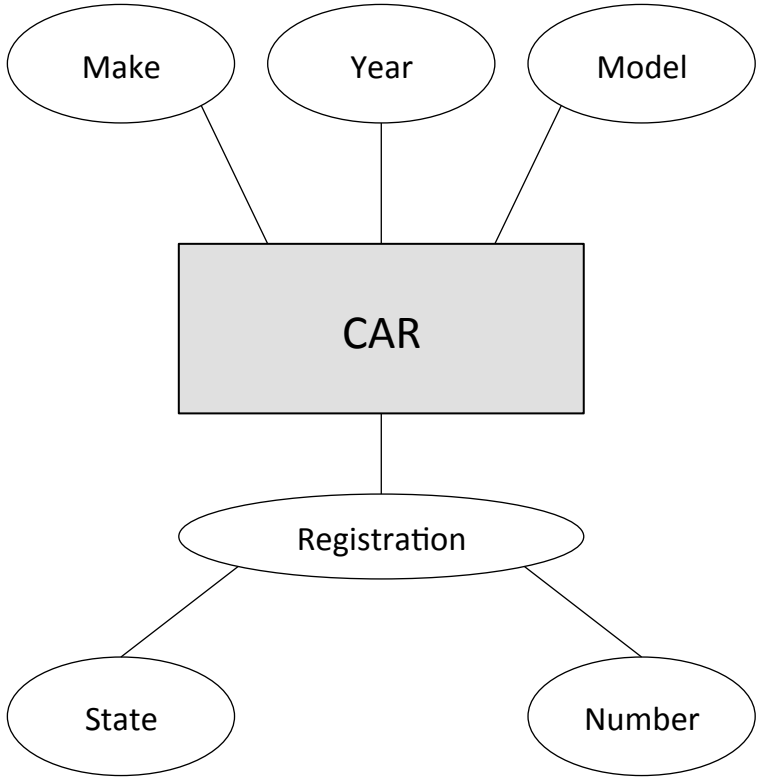
All cars have a year, make, and model.



Composite Attributes

Can be subdivided into smaller subparts

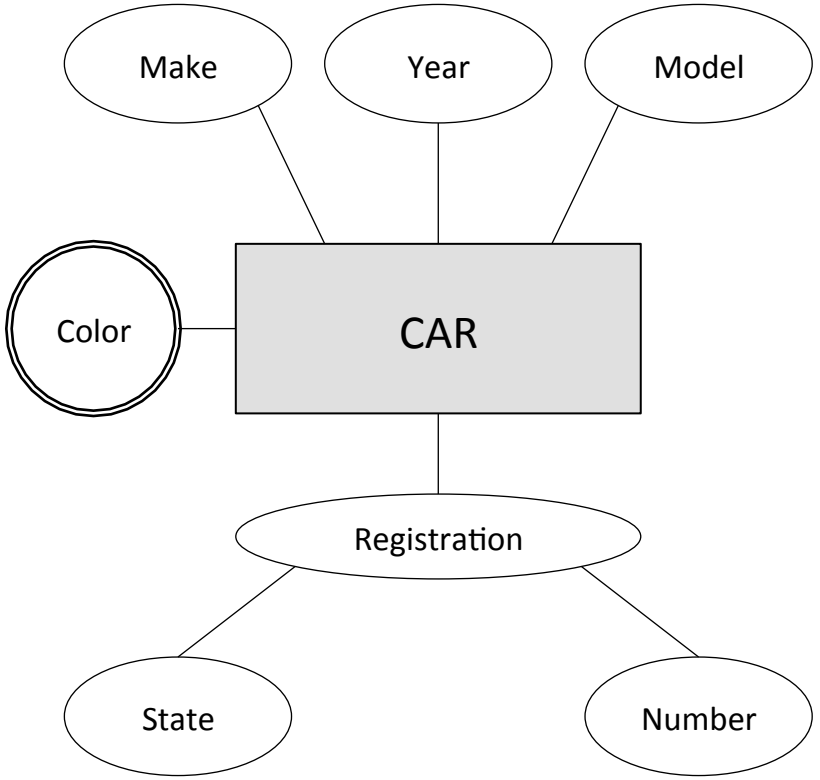
All cars have a year, make, model, ***and registration****.*



Multivalued Attributes

Can take a [possibly specified] number of values.

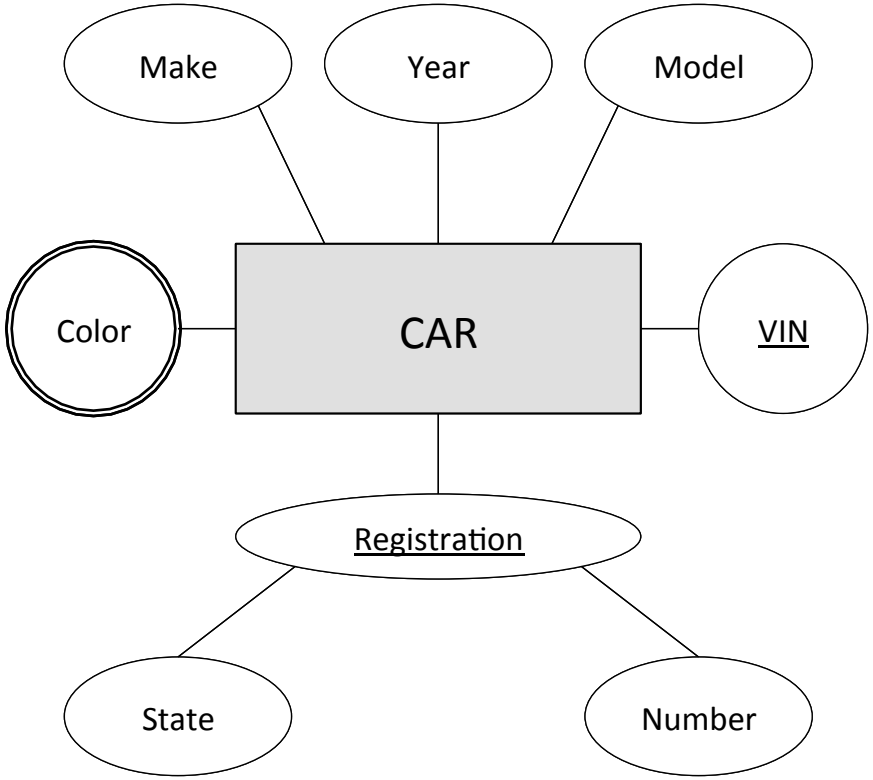
All cars have a year, make, model, registration, and ***some number of colors****.*



Key Attributes

The value uniquely identifies each entity

All cars have a year, make, model, ***registration (unique), vehicle number (vin; unique)****, some number of colors.*



Potential Pitfall

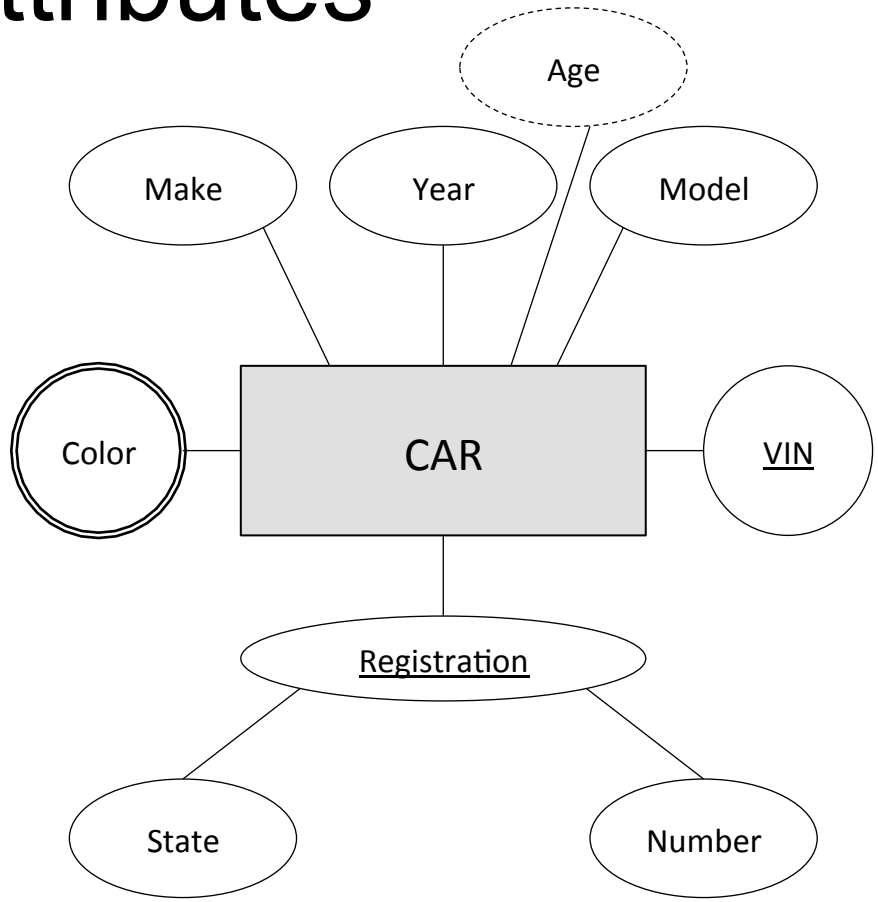
- In relational schema, underlining multiple attributes indicates that for all rows, the *combination* is unique

- In ERDs, underlining multiple attributes indicates that *each individually* can uniquely identify an entity

Derived Attributes

The value can be computed

*All cars have a year, **age**, make, model, registration (unique), vehicle number (vin; unique), some number of colors.*

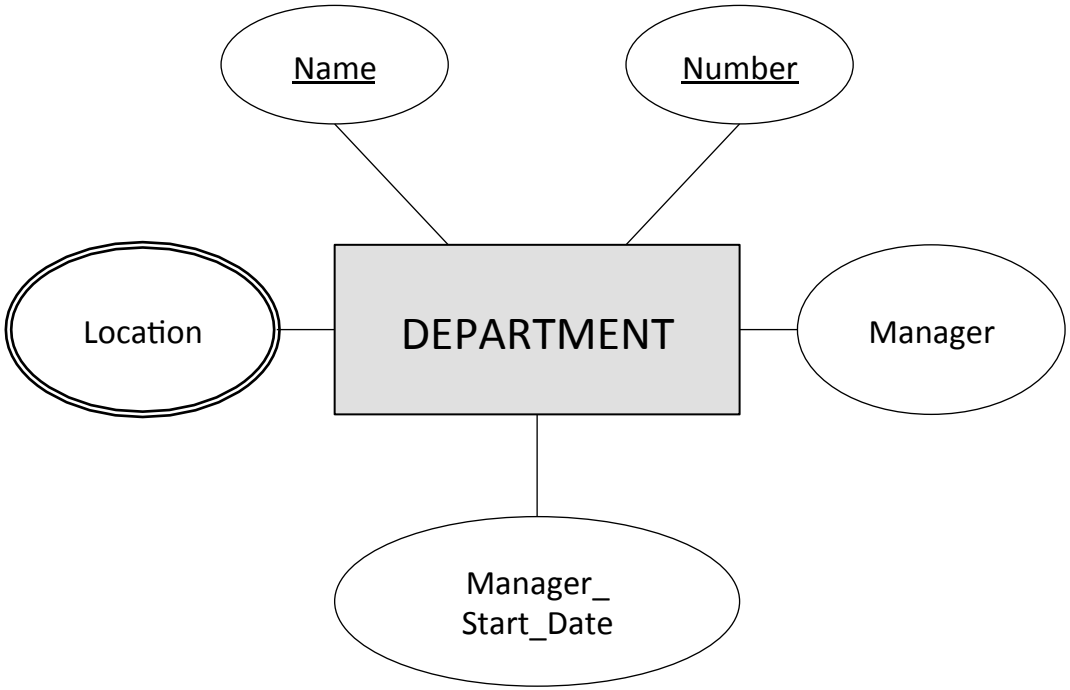


Exercise

Draw an ERD for the following description:

Each department has a unique name, a unique number, and a particular employee who manages the department. We keep track of the start date when that employee began managing the department. A department may have several locations.

Answer

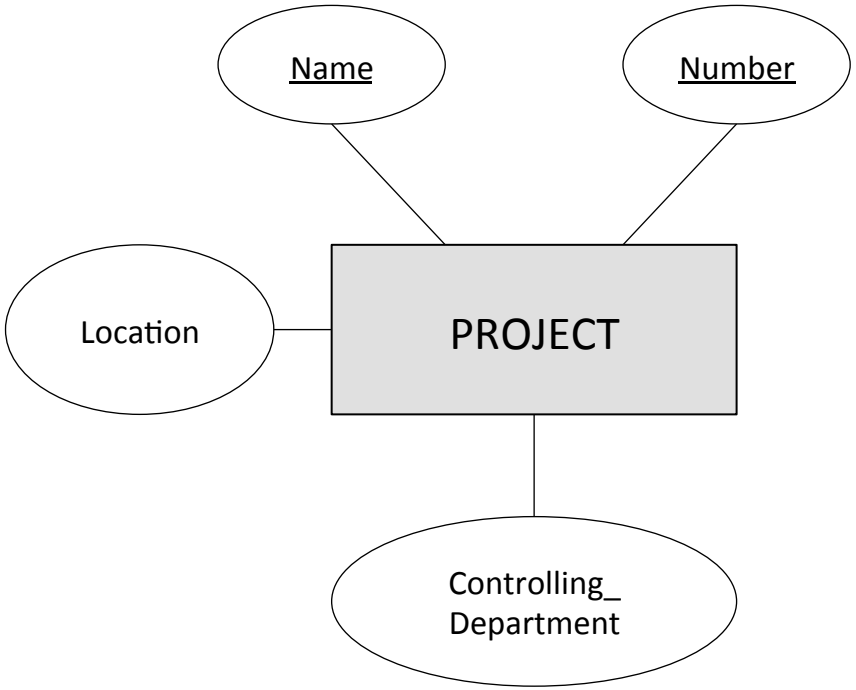

Name
Number
Location
DEPARTMENT
Manager
Manager_
Start_Date

Exercise

Draw an ERD for the following description:

A department controls a number of projects, each of which has a unique name, a unique number, and a single location.

Answer

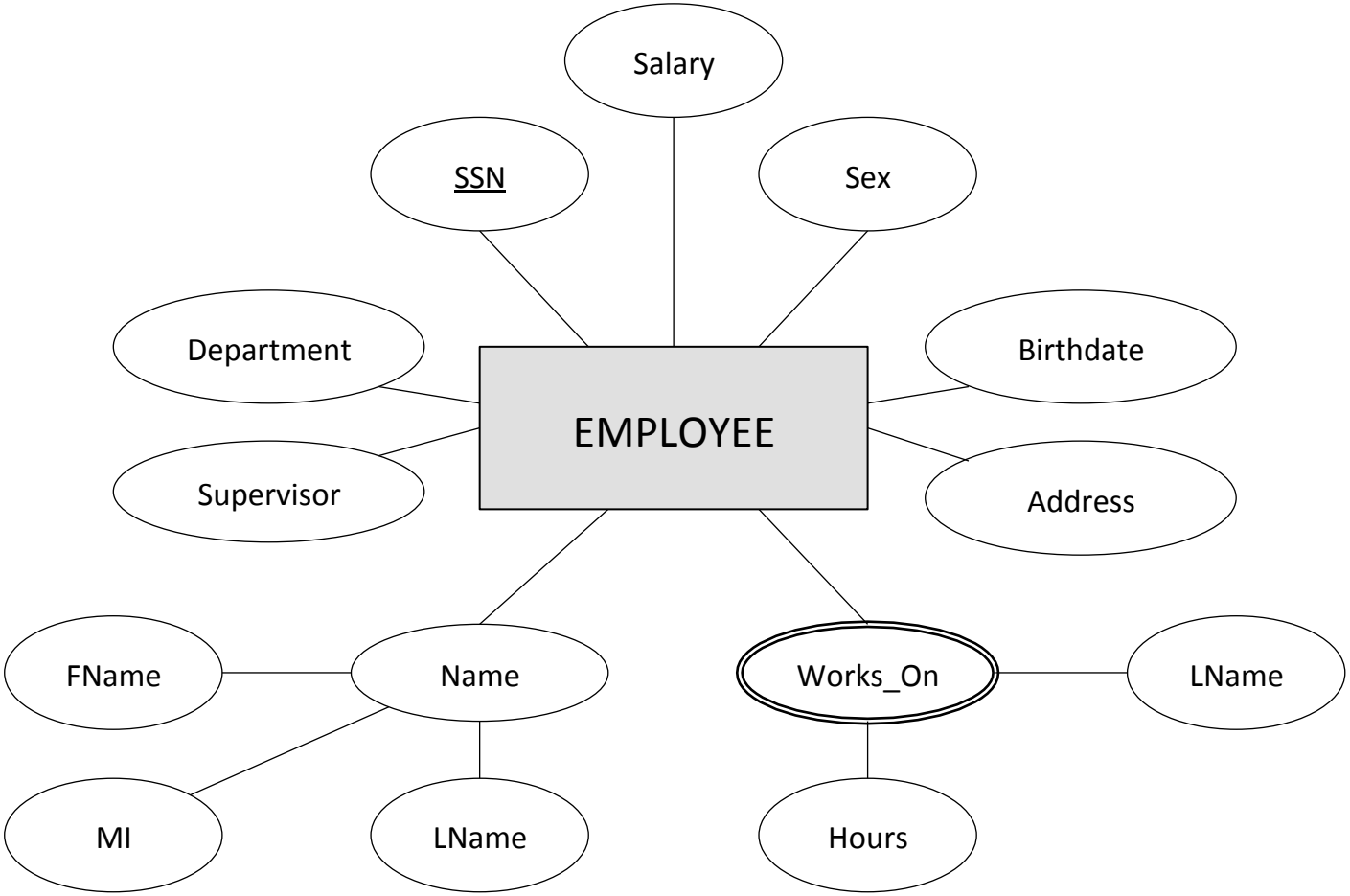


Exercise

Draw an ERD for the following description:

We store each employee's name (first, last, MI), Social Security number (SSN), street address, salary, sex (gender), and birth date. An employee is assigned to one department, but may work on several projects, which are not necessarily controlled by the same department. We keep track of the current number of hours per week that an employee works on each project. We also keep track of the direct supervisor of each employee (who is another employee).

Answer

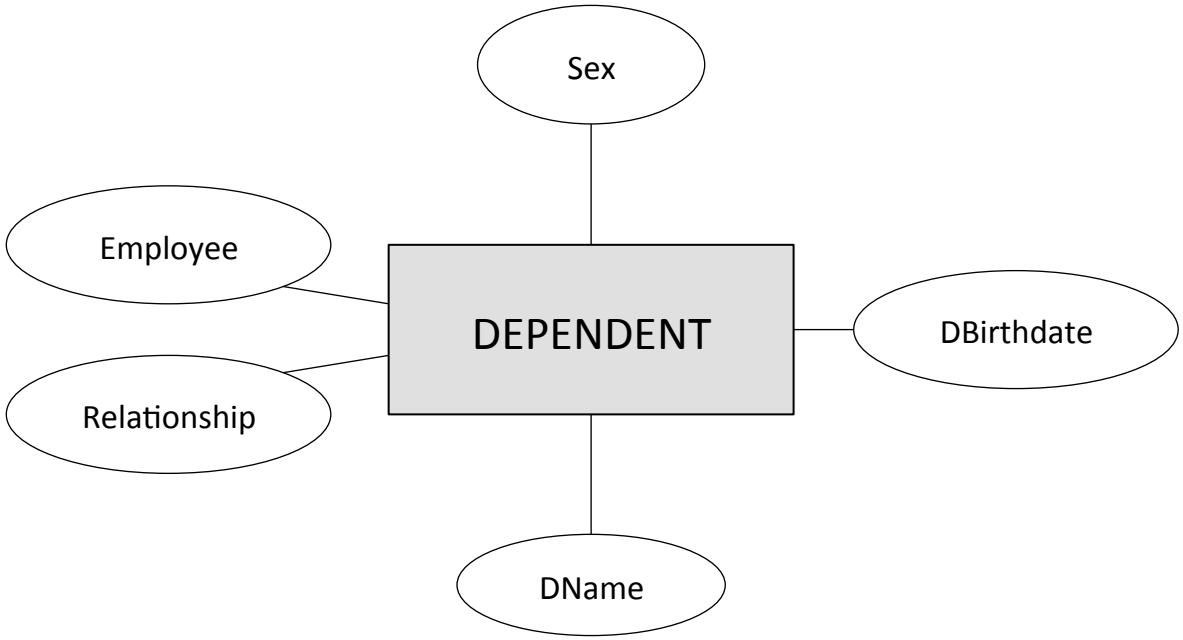

Salary
SSN
Sex
Department
Birthdate
EMPLOYEE
Supervisor
Address
FName
Name
Works_On
LName
MI
LName
Hours

Exercise

Draw an ERD for the following description:

We want to keep track of the dependents of each employee for insurance purposes. We keep each dependent's first name, sex, birth date, and relationship to the employee.

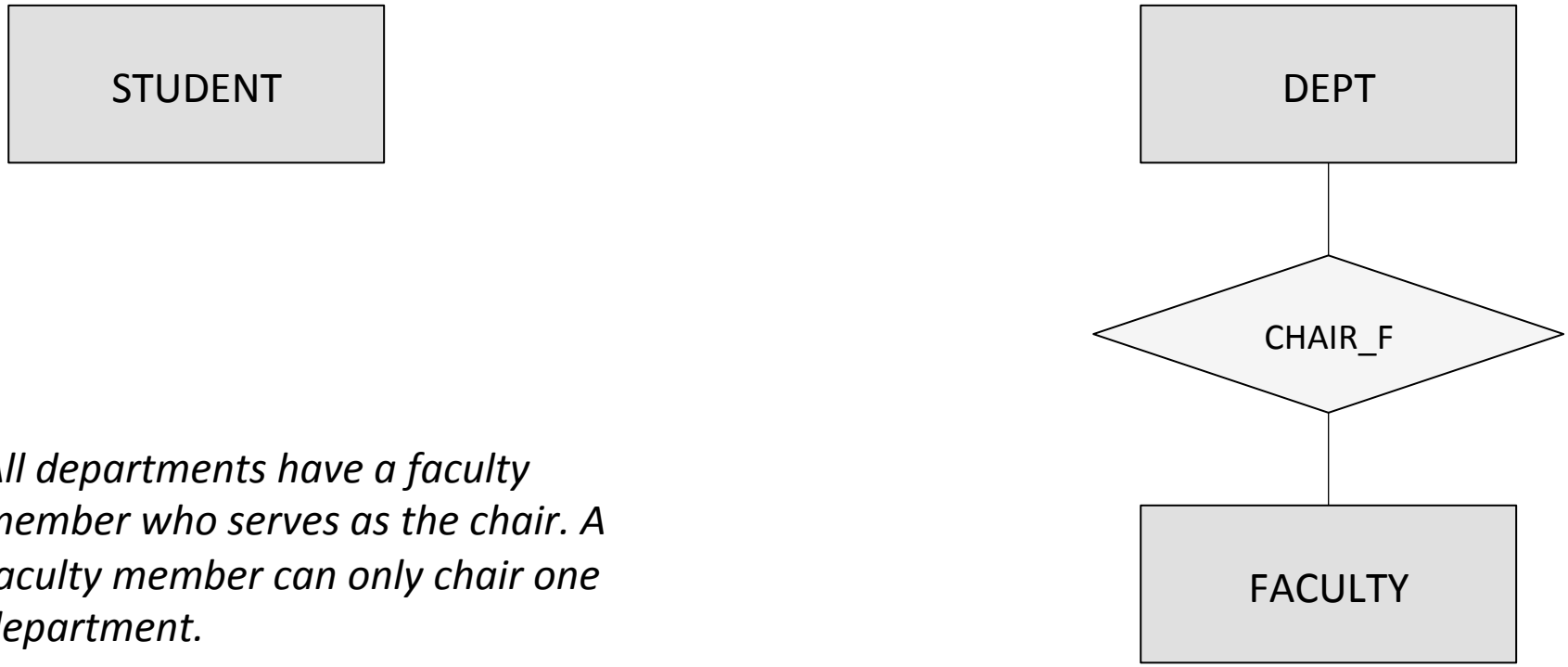
Answer



Relationships

Associates one or more sets of entities

- One = recursive (**role** is important)

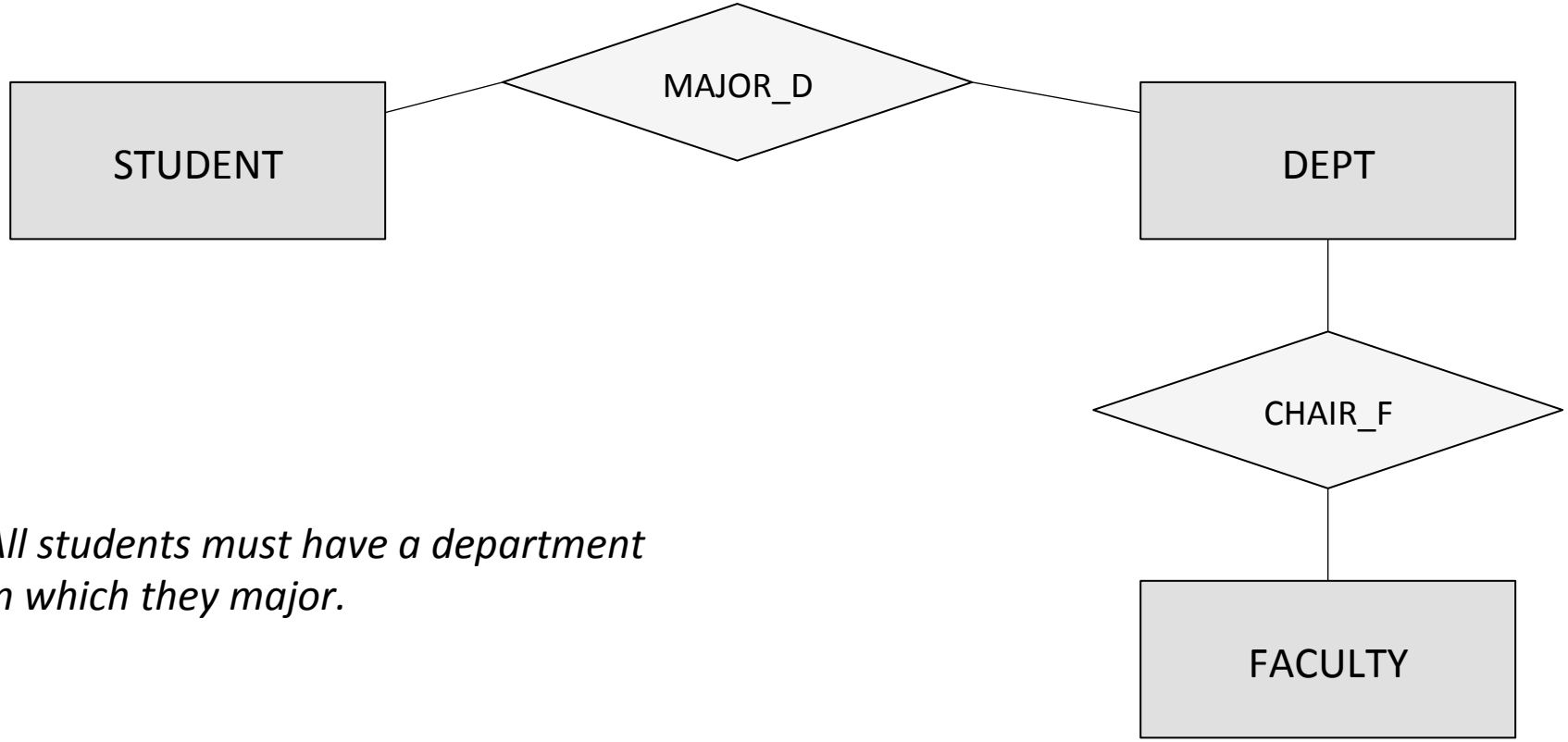


All departments have a faculty member who serves as the chair. A faculty member can only chair one department.

Relationships

Associates one or more sets of entities

– One = recursive (**role** is important)

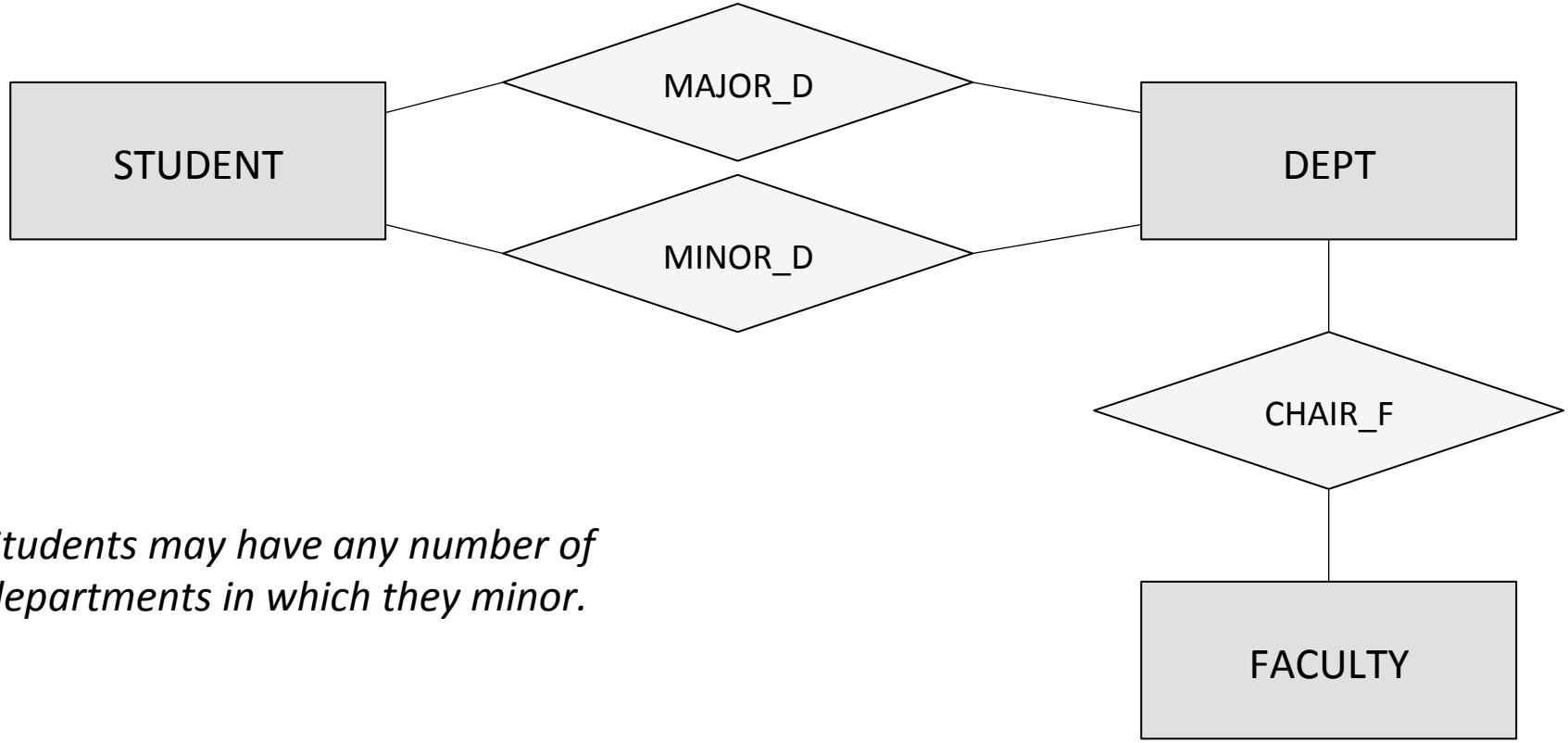


All students must have a department in which they major.

Relationships

Associates one or more sets of entities

– One = recursive (**role** is important)

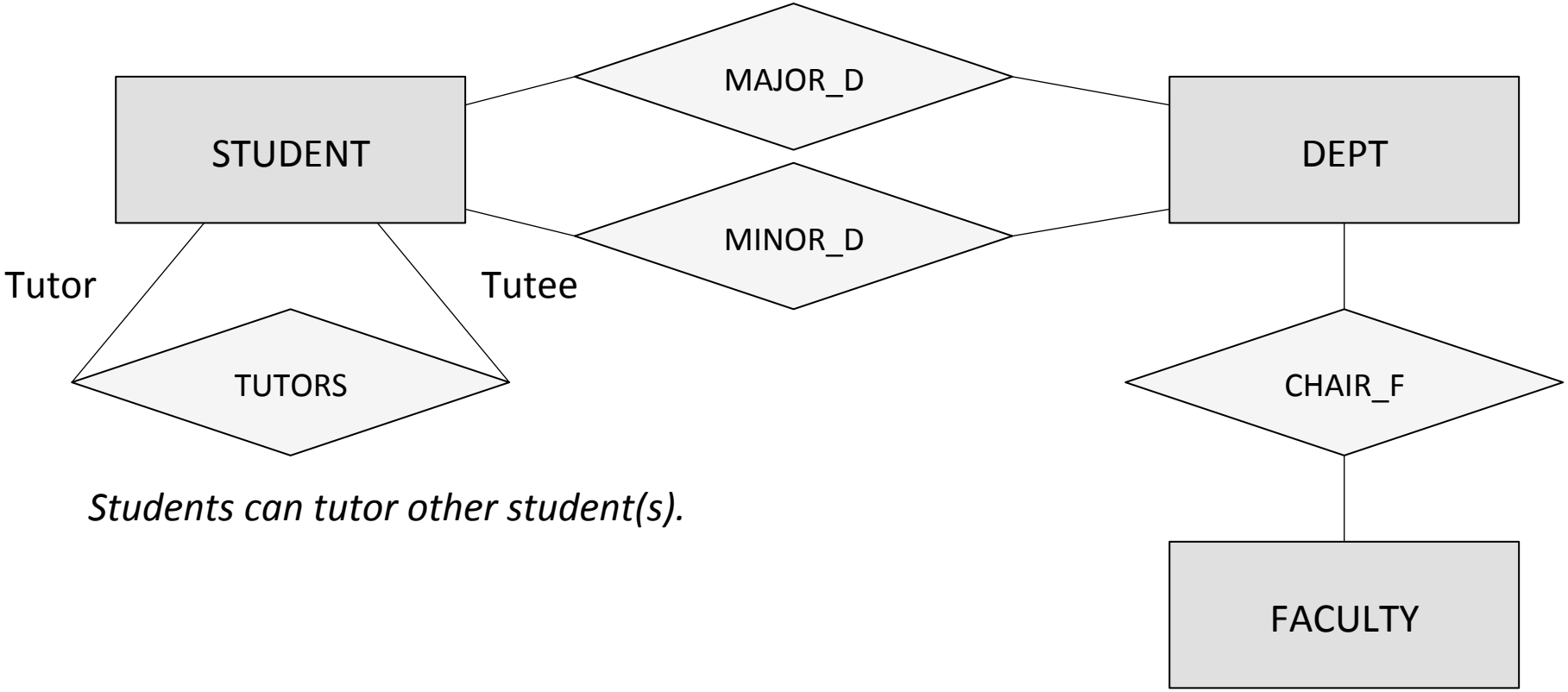


Students may have any number of departments in which they minor.

Relationships

Associates one or more sets of entities

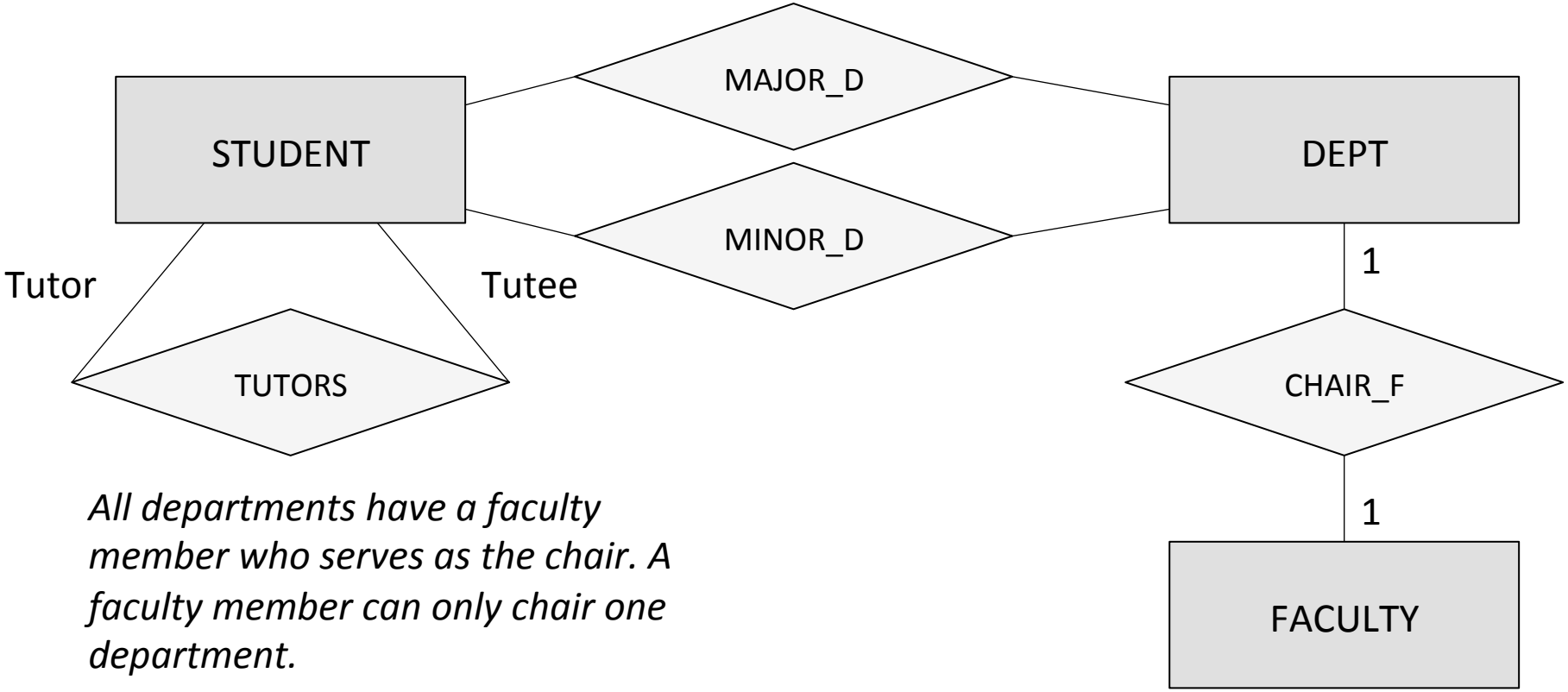
– One = recursive (**role** is important)



Students can tutor other student(s).

Cardinality Ratios

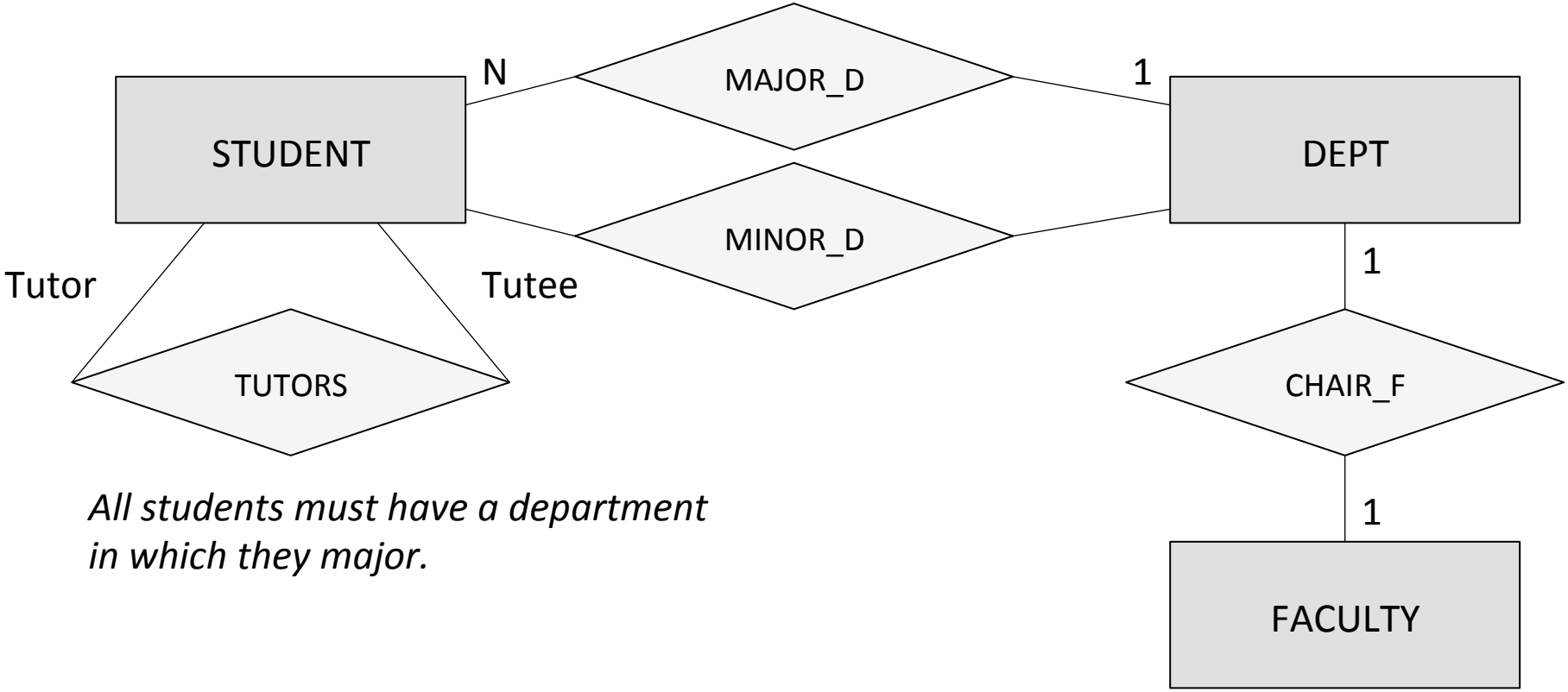
Constrains the number of entities that can participate in each role of the relationship



All departments have a faculty member who serves as the chair. A faculty member can only chair one department.

Cardinality Ratios

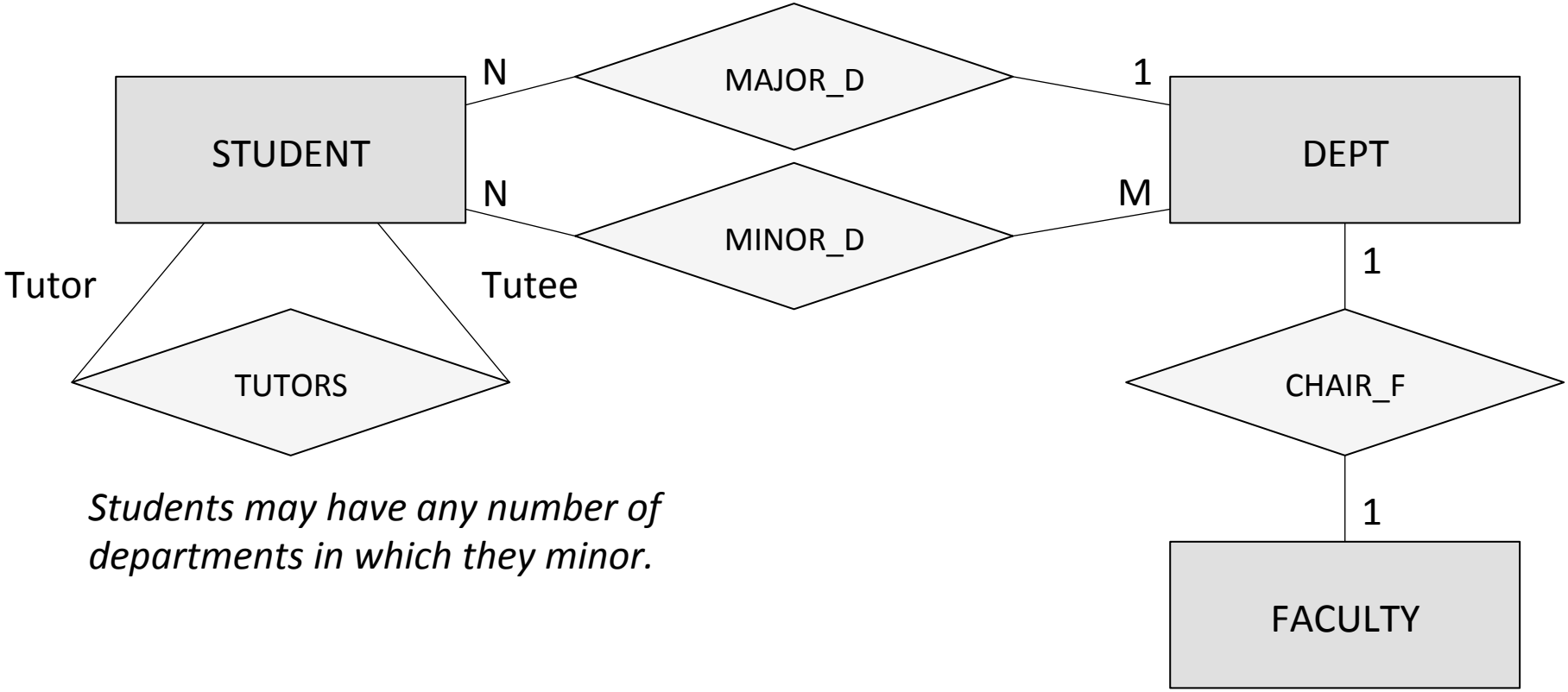
Constrains the number of entities that can participate in each role of the relationship



All students must have a department in which they major.

Cardinality Ratios

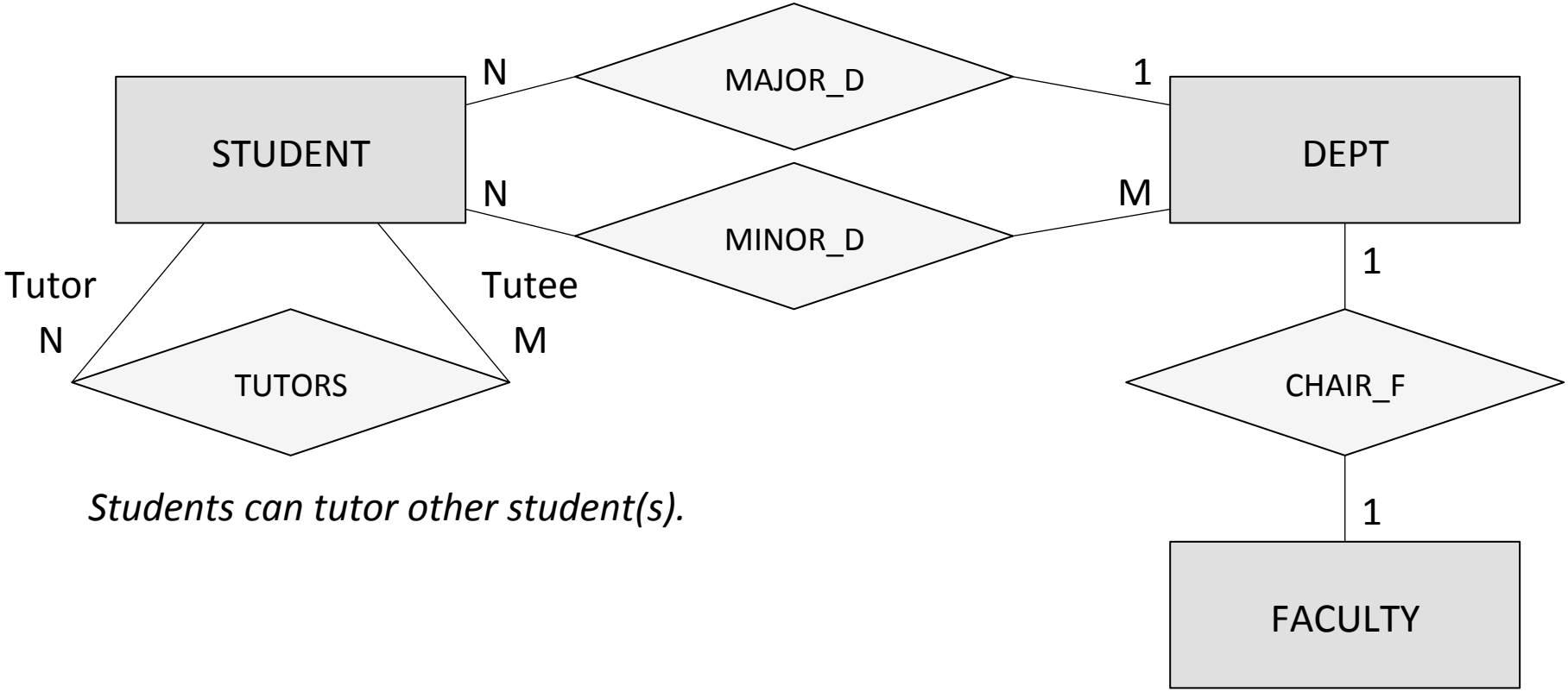
Constrains the number of entities that can participate in each role of the relationship



Students may have any number of departments in which they minor.

Cardinality Ratios

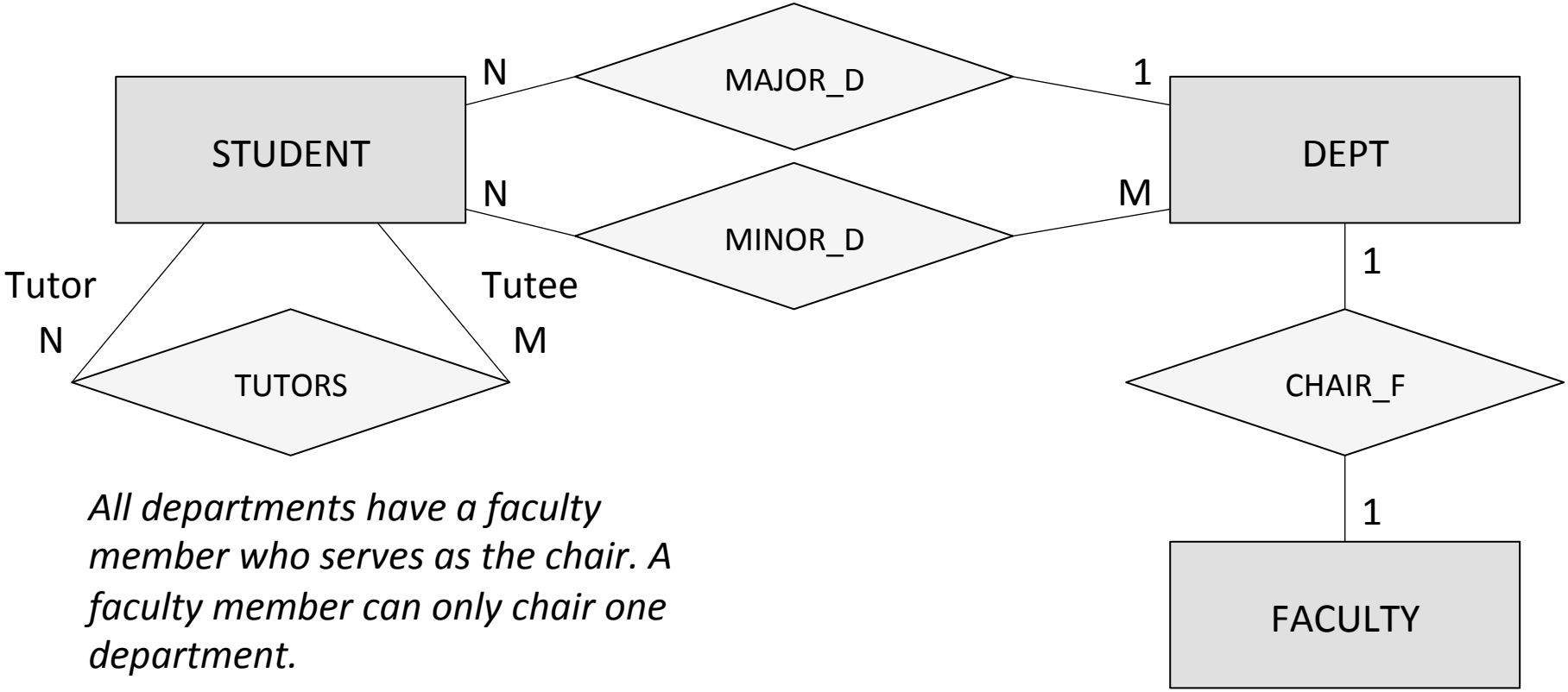
Constrains the number of entities that can participate in each role of the relationship



Students can tutor other student(s).

Structural Constraints

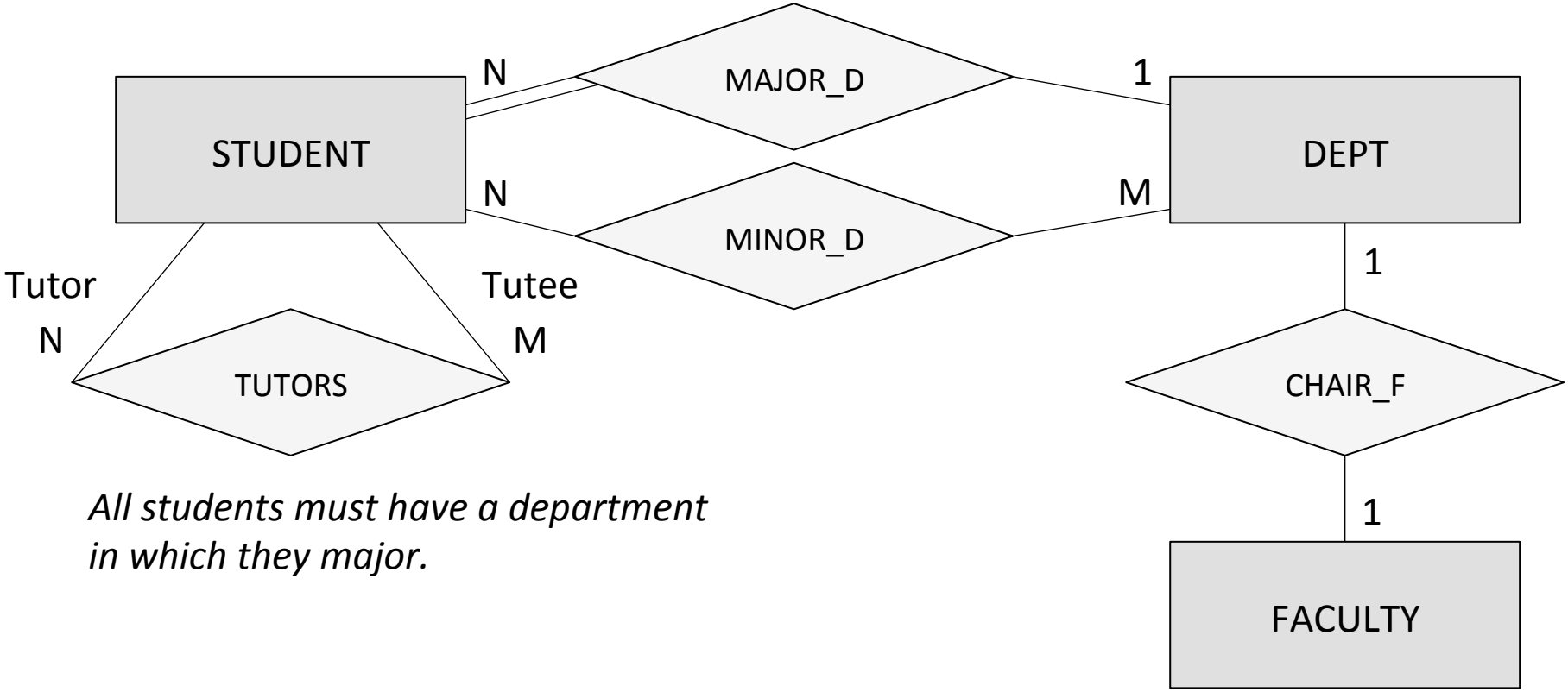
If an entity does not exist unless it appears with an entity in a relationship, the participation is **total** (existence dependency). Else, **partial**.



All departments have a faculty member who serves as the chair. A faculty member can only chair one department.

Structural Constraints

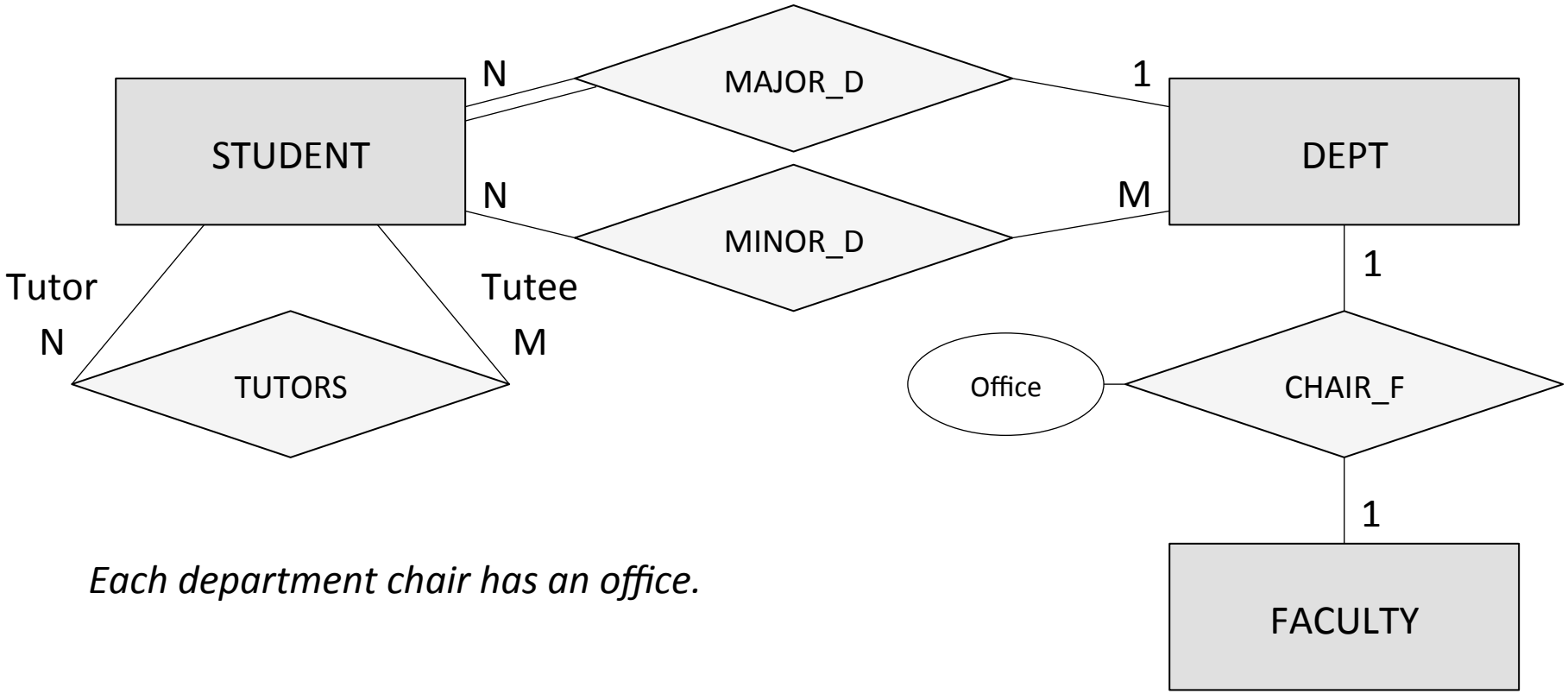
If an entity does not exist unless it appears with an entity in a relationship, the participation is **total** (existence dependency). Else, **partial**.



All students must have a department in which they major.

Attributes of Relationships

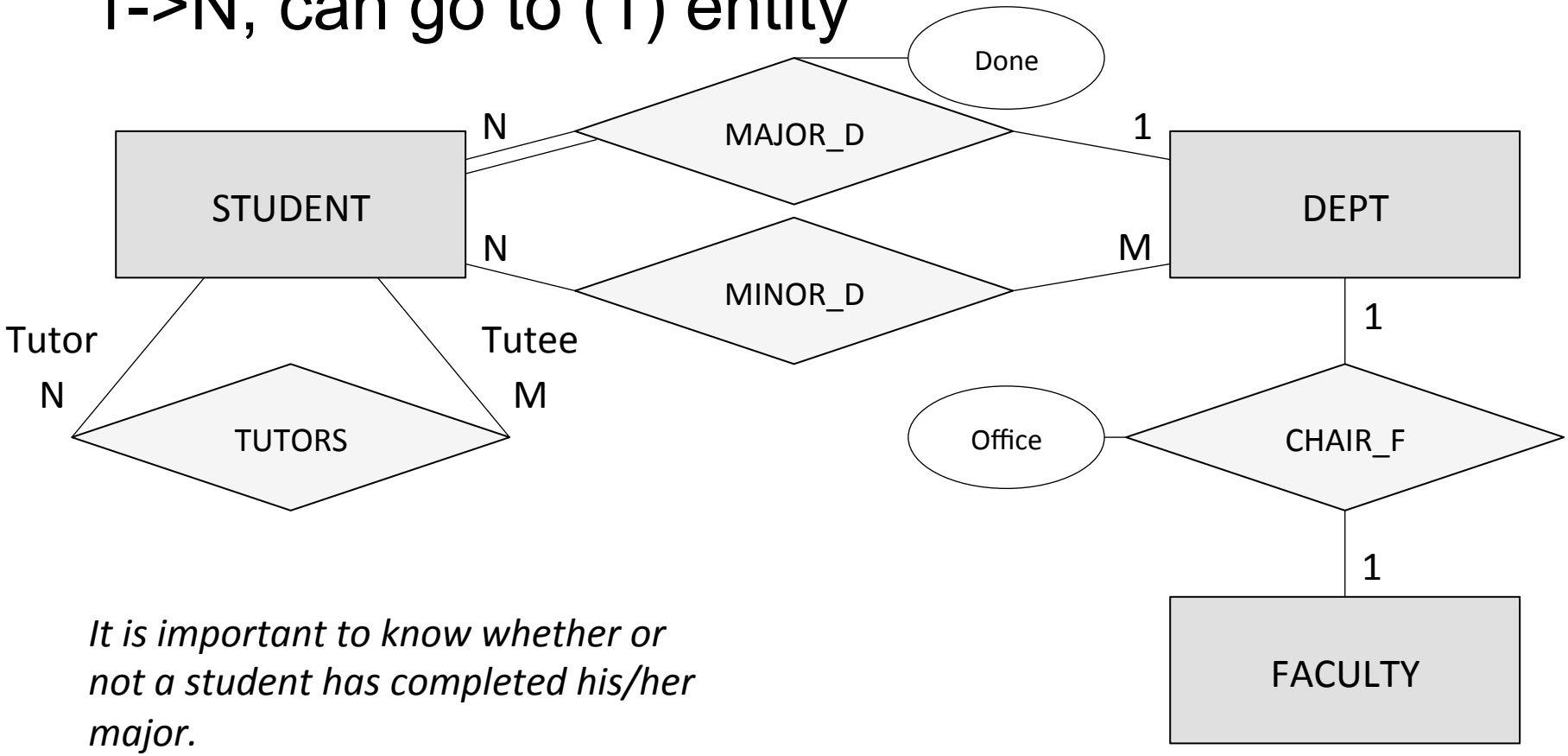
1->1, can go to either entity
1->N, can go to (1) entity



Each department chair has an office.

Attributes of Relationships

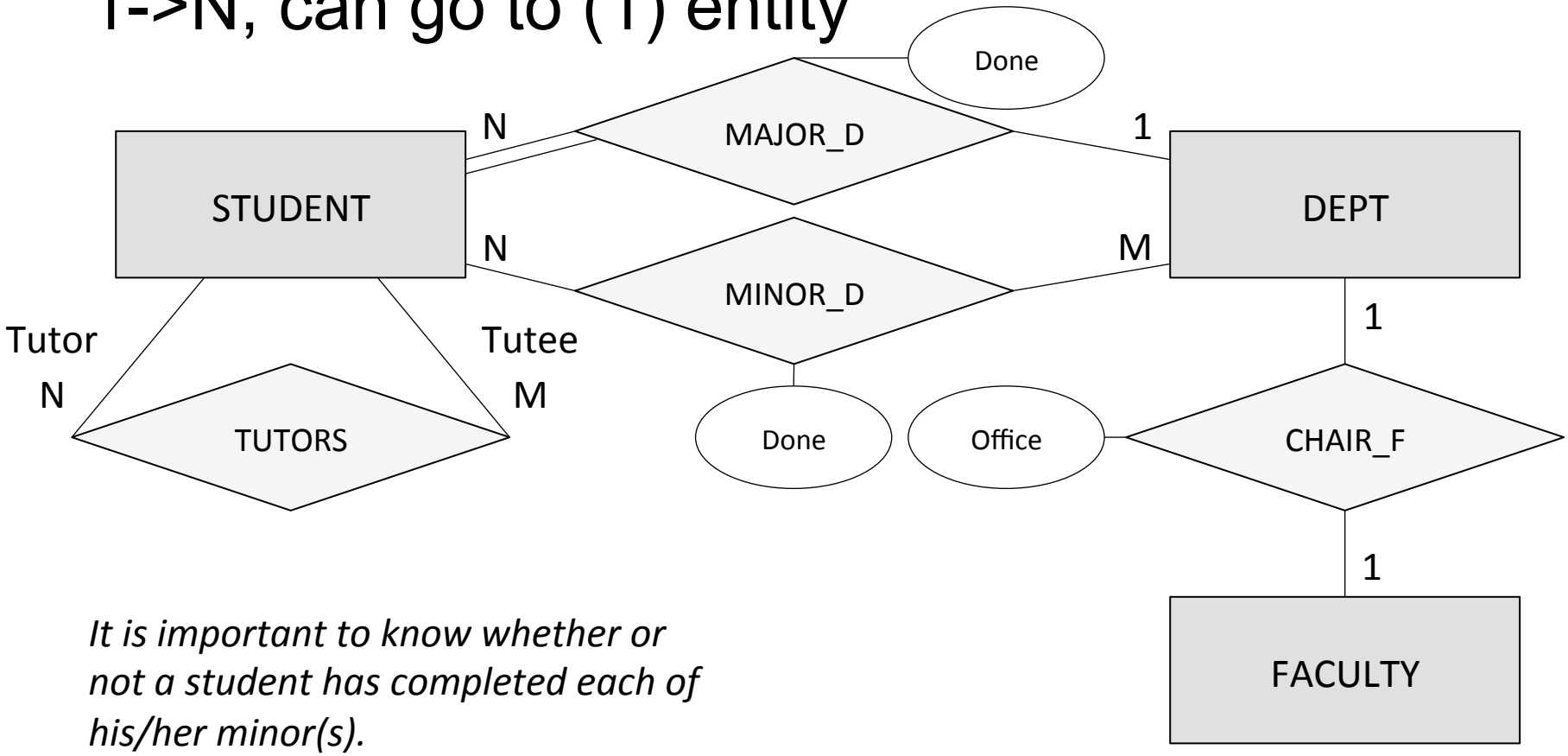
1->1, can go to either entity
1->N, can go to (1) entity



It is important to know whether or not a student has completed his/her major.

Attributes of Relationships

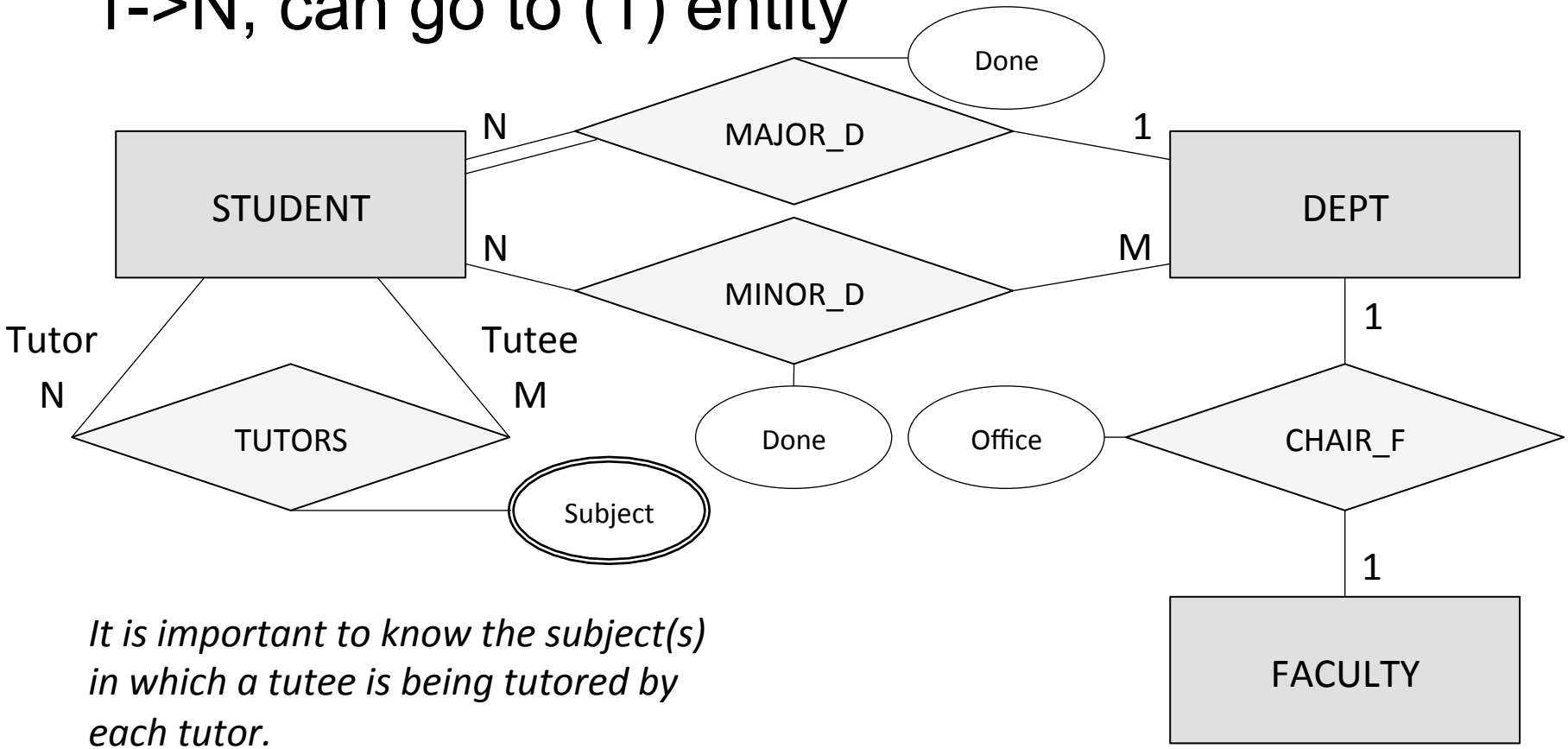
1->1, can go to either entity
1->N, can go to (1) entity



It is important to know whether or not a student has completed each of his/her minor(s).

Attributes of Relationships

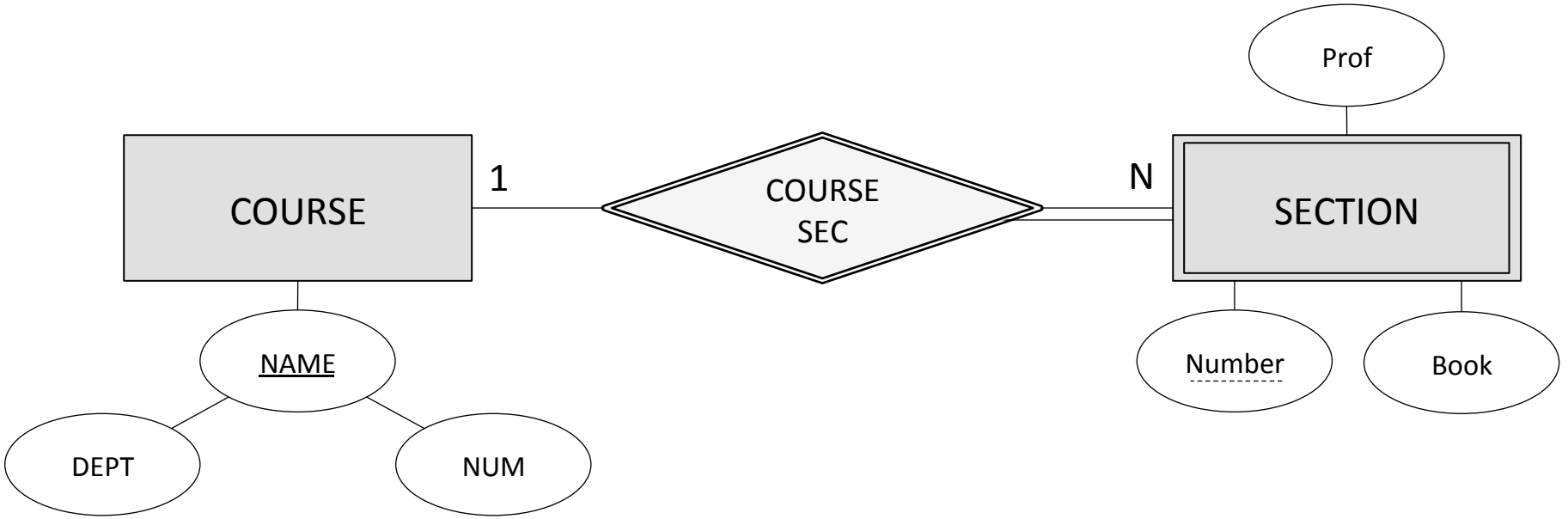
1->1, can go to either entity
1->N, can go to (1) entity



It is important to know the subject(s) in which a tutee is being tutored by each tutor.

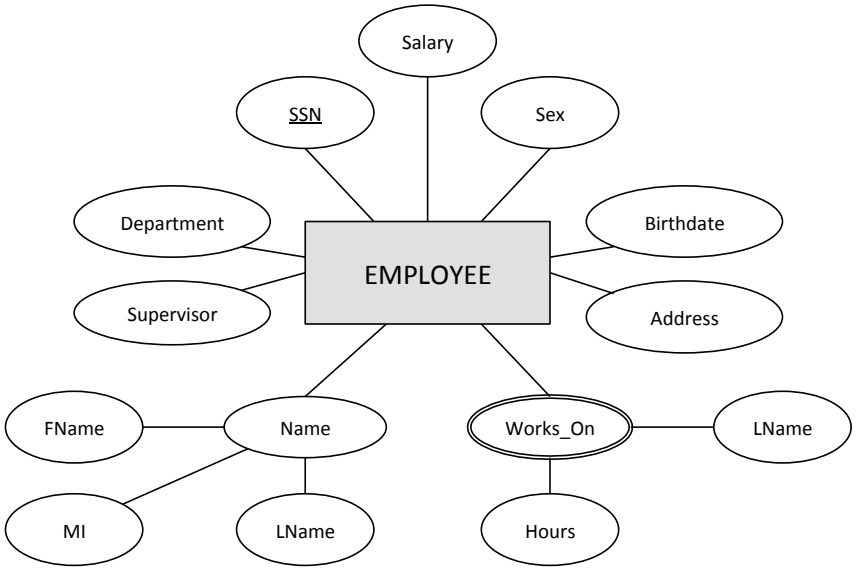
Weak Entities

Entity types that do not have key attributes of their own are **weak**; instead identified by relation to specific entity of another type (the **identifying** type)

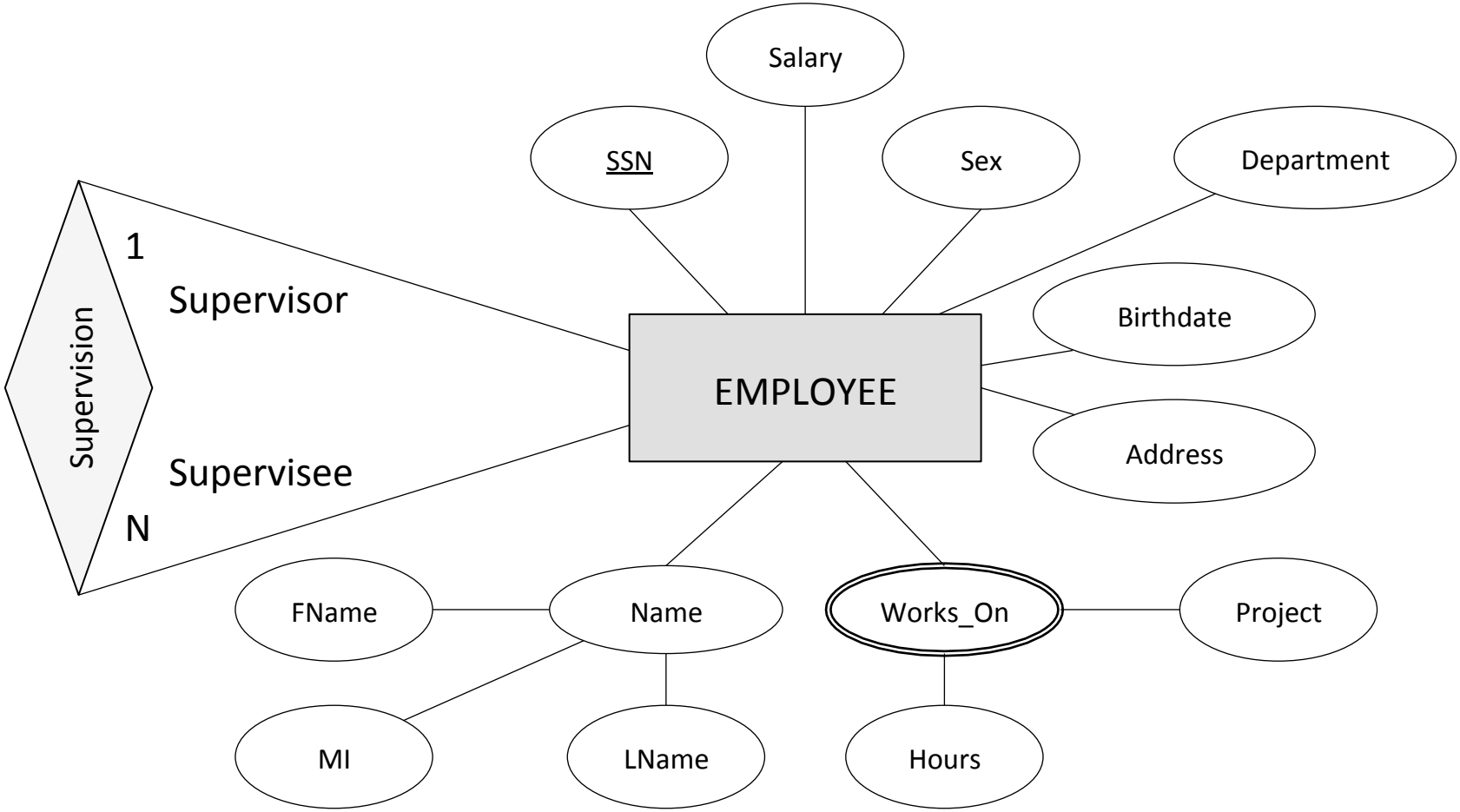


Revise!

We store each employee's name (first, last, MI), Social Security number (SSN), street address, salary, sex (gender), and birth date. An employee is assigned to one department, but may work on several projects, which are not necessarily controlled by the same department. We keep track of the current number of hours per week that an employee works on each project. ***We also keep track of the direct supervisor of each employee (who is another employee).***

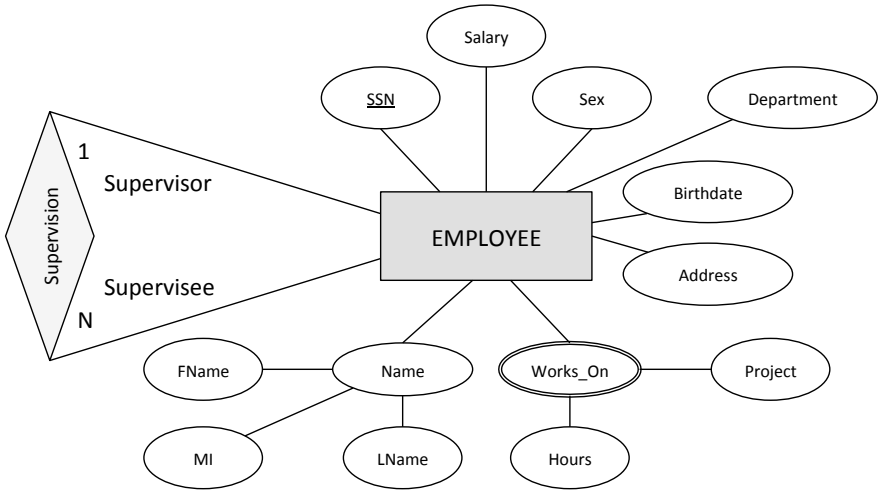


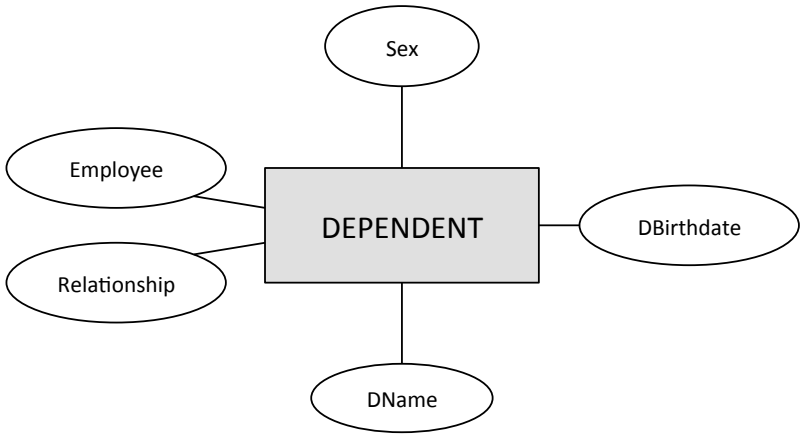
Answer


Salary
SSN
Sex
Department
1
Supervisor
Supervision
Supervisee
N
EMPLOYEE
Birthdate
Address
FName
Name
Works_On
Project
MI
LName
Hours

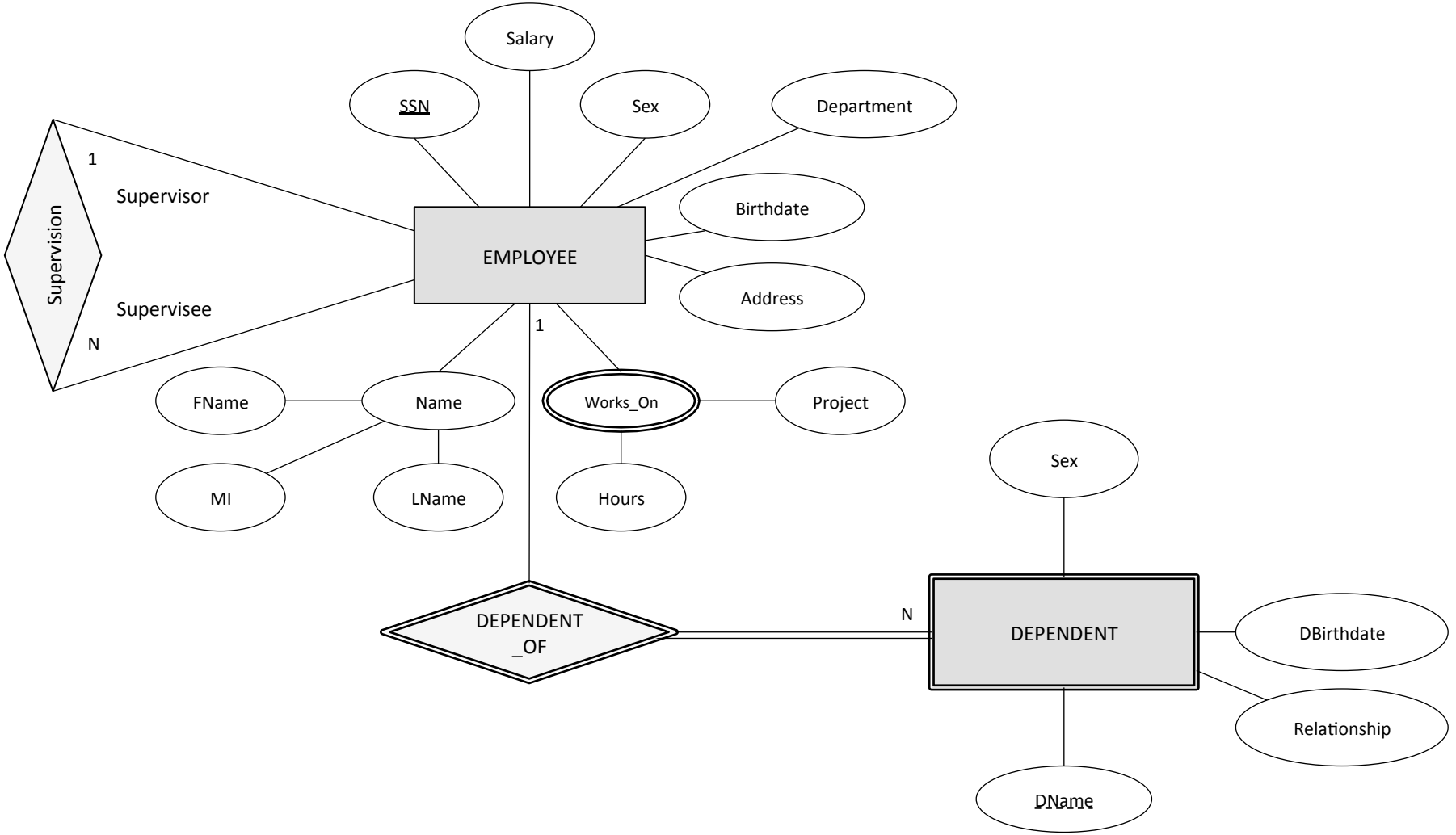
Revise!

*We want to keep track of the dependents **of each employee** for insurance purposes. We keep each dependent's first name, sex, birth date, and relationship to the employee.*



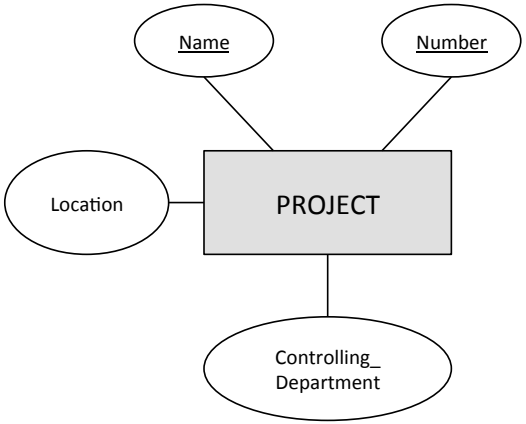


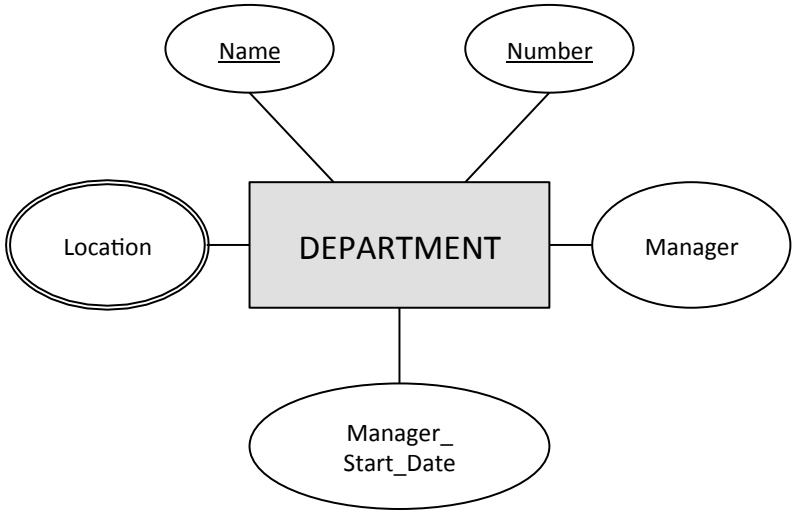
Answer


Salary
SSN
Sex
Department
1
Supervisor
Supervision
Birthdate
EMPLOYEE
Address
Supervisee
N
1
FName
Name
Works_On
Project
MI
LName
Hours
Sex
DEPENDENT
_OF
N
DEPENDENT
DBirthdate
Relationship
DName

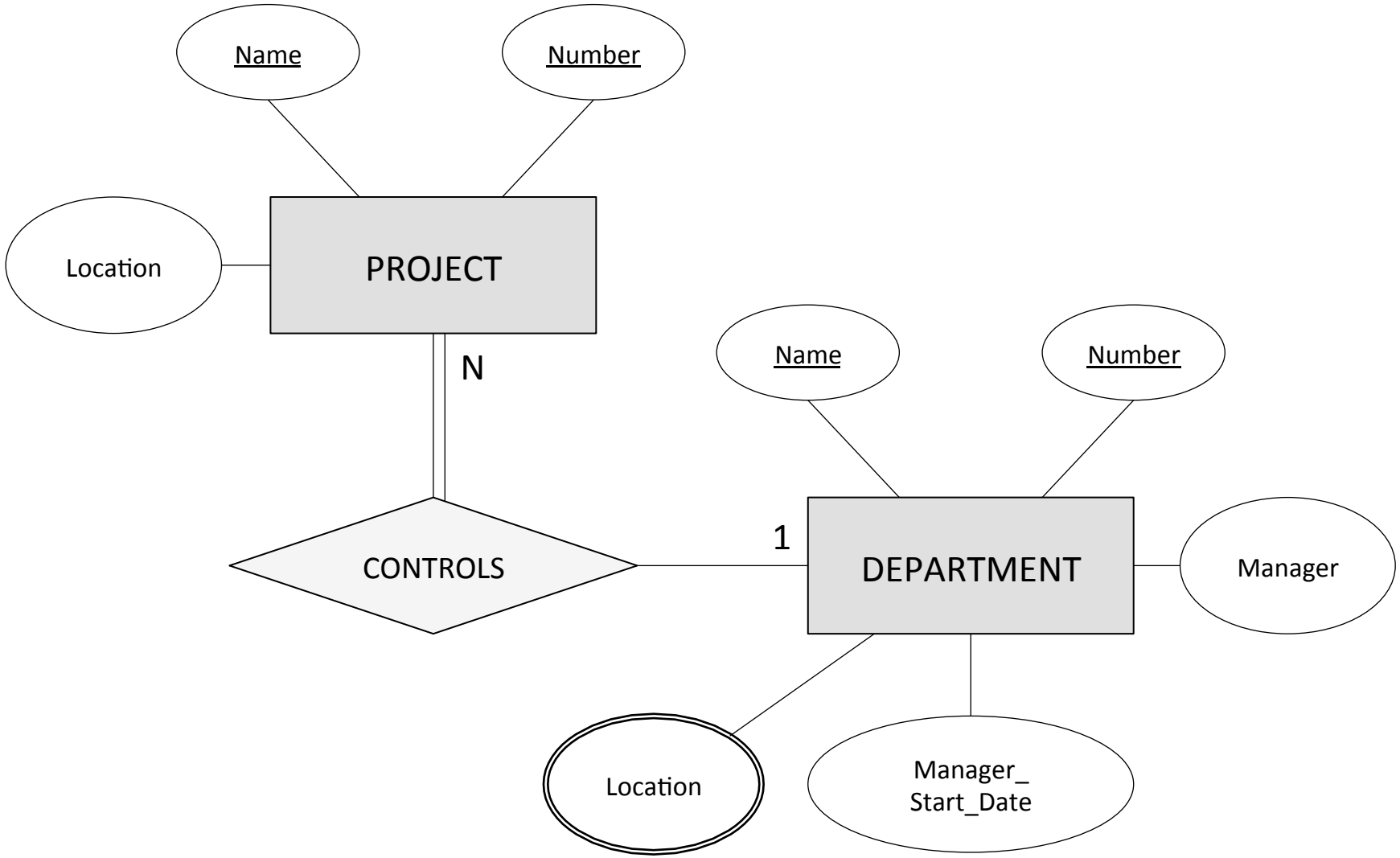
Revise!

A department controls *a number of projects, each of which has a unique name, a unique number, and a single location.*





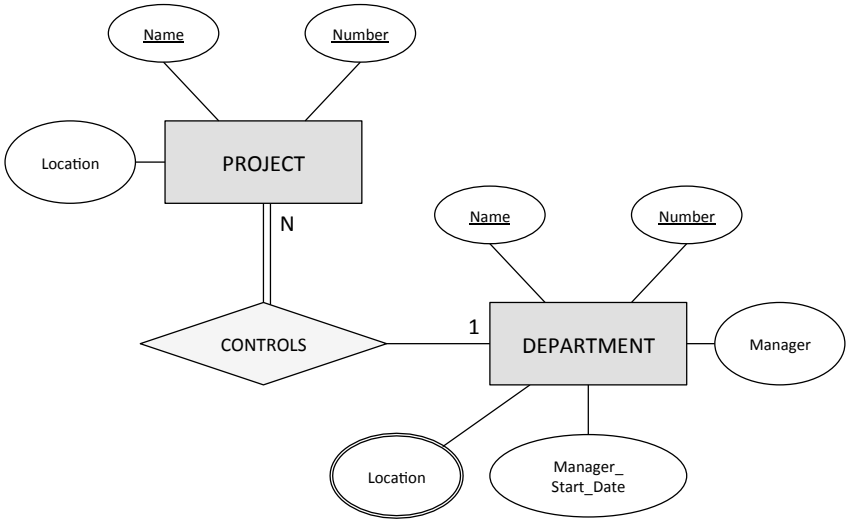
Answer

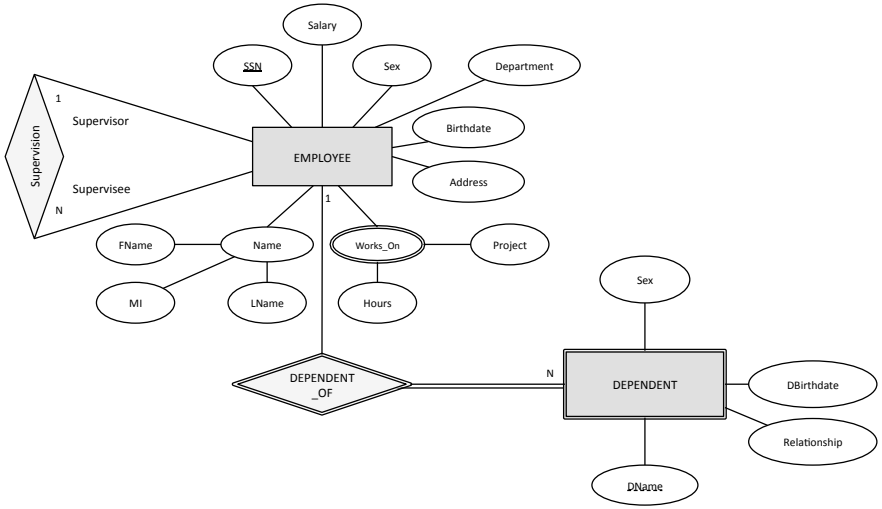

Name
Number
Location
PROJECT
N
Name
Number
CONTROLS
1
DEPARTMENT
Manager
Location
Manager_
Start_Date

Revise!

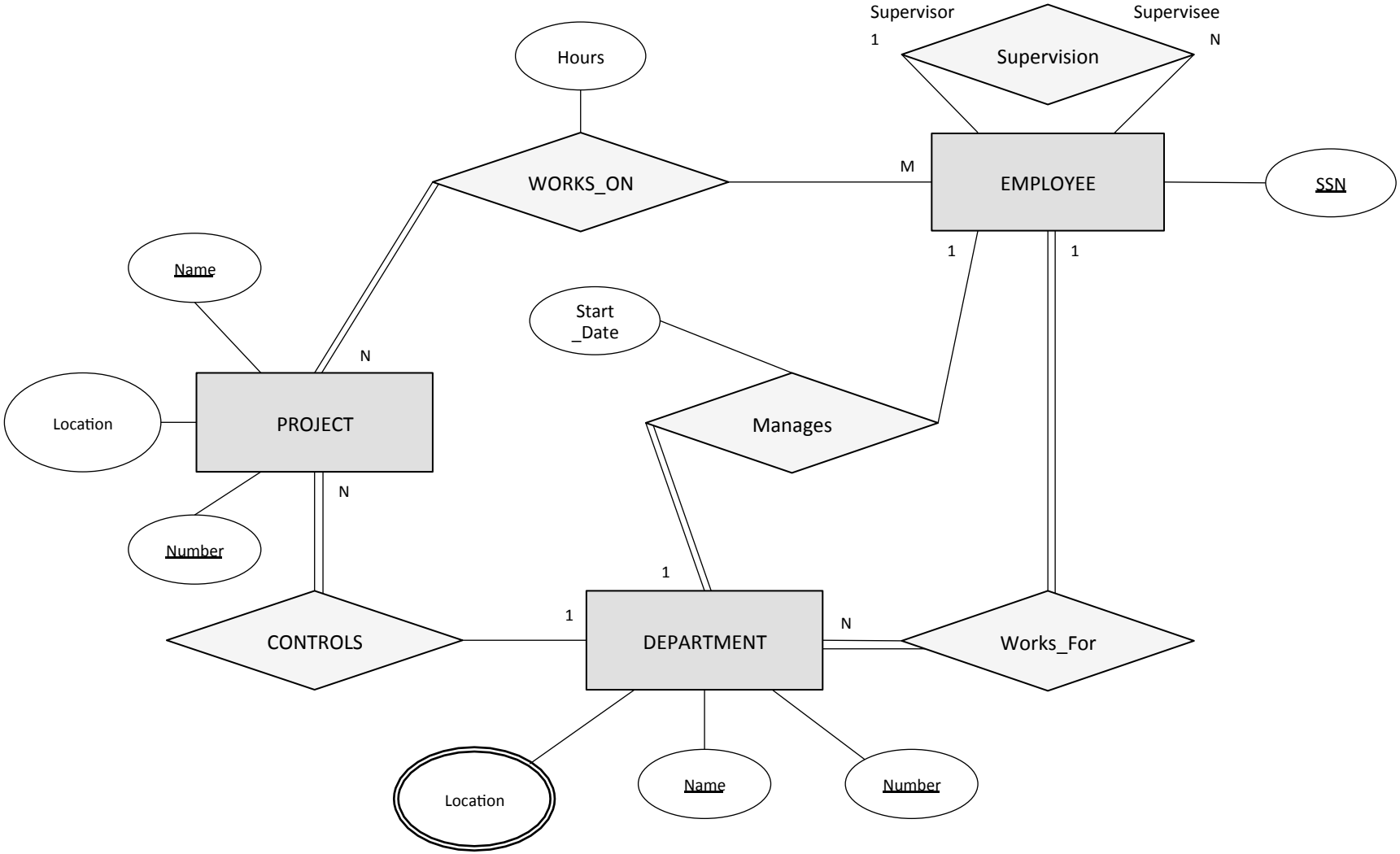
Each department has a … particular employee who manages the department.

An employee is assigned to one department, but may work on several projects, which are not necessarily controlled by the same department. We keep track of the current number of hours per week that an employee works on each project.

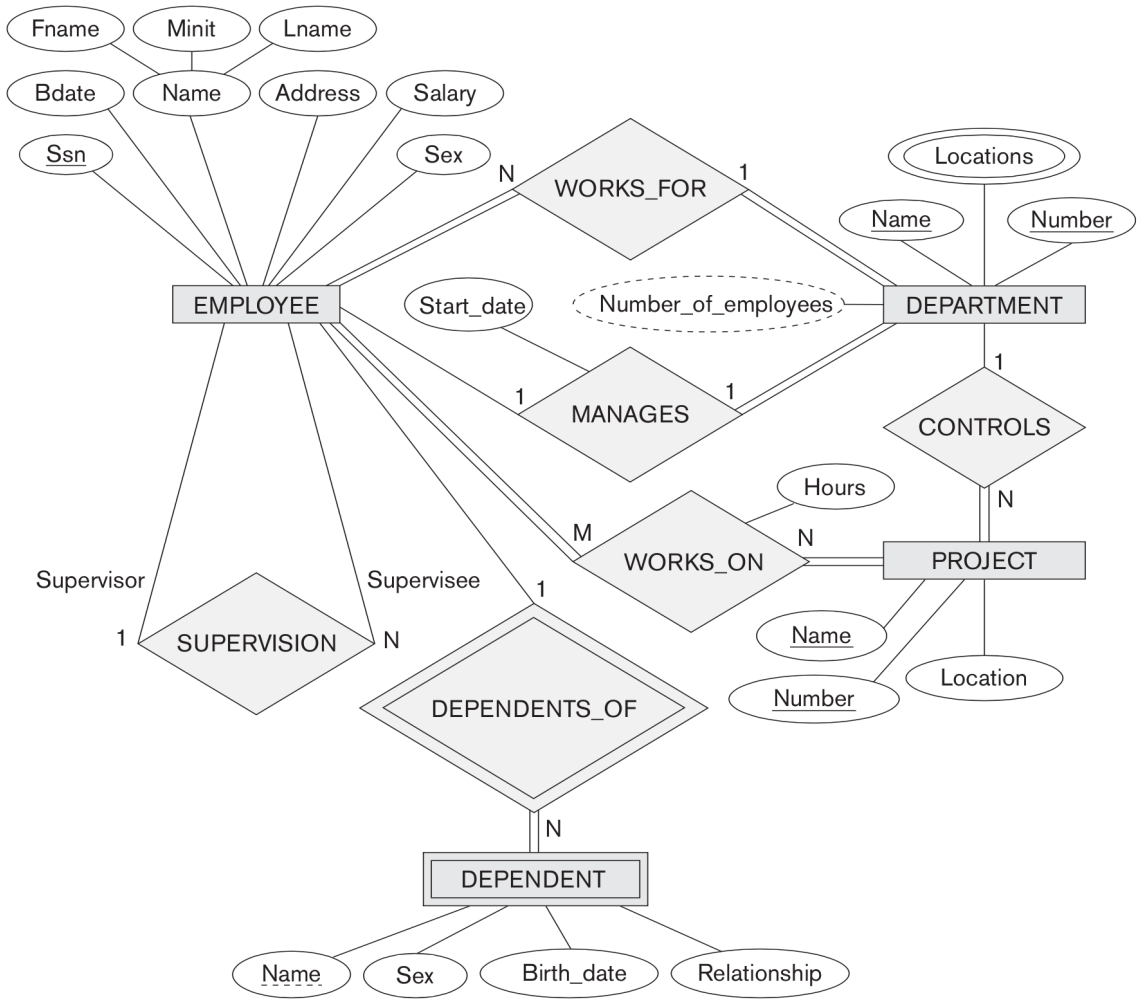




Answer

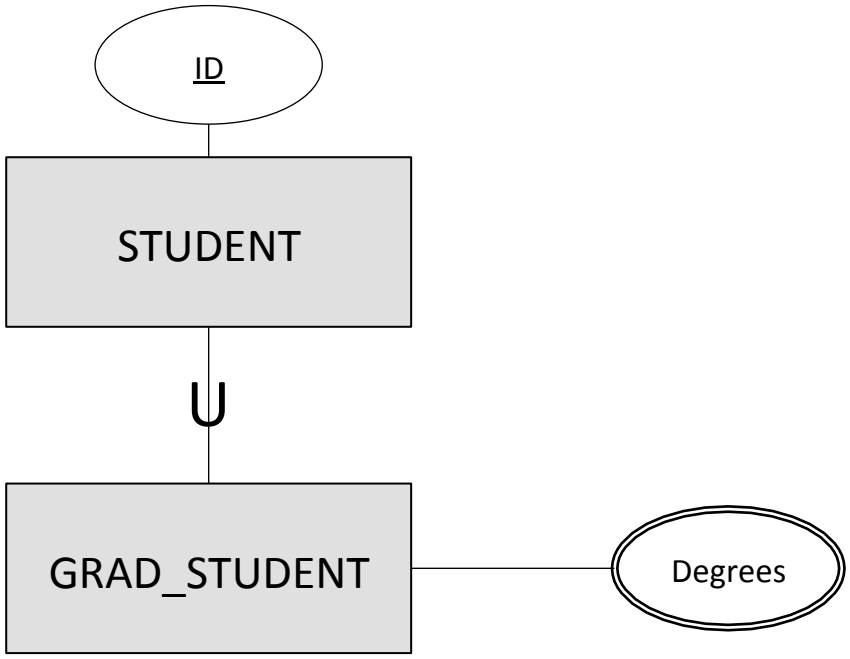

Supervisor
Supervisee
1
N
Supervision
Hours
WORKS_ON
M
EMPLOYEE
SSN
1
1
Name
Start
_Date
N
Location
PROJECT
Manages
N
Number
1
1
CONTROLS
DEPARTMENT
N
Works_For
Location
Name
Number

All Together Now!



Specialization/Generalization

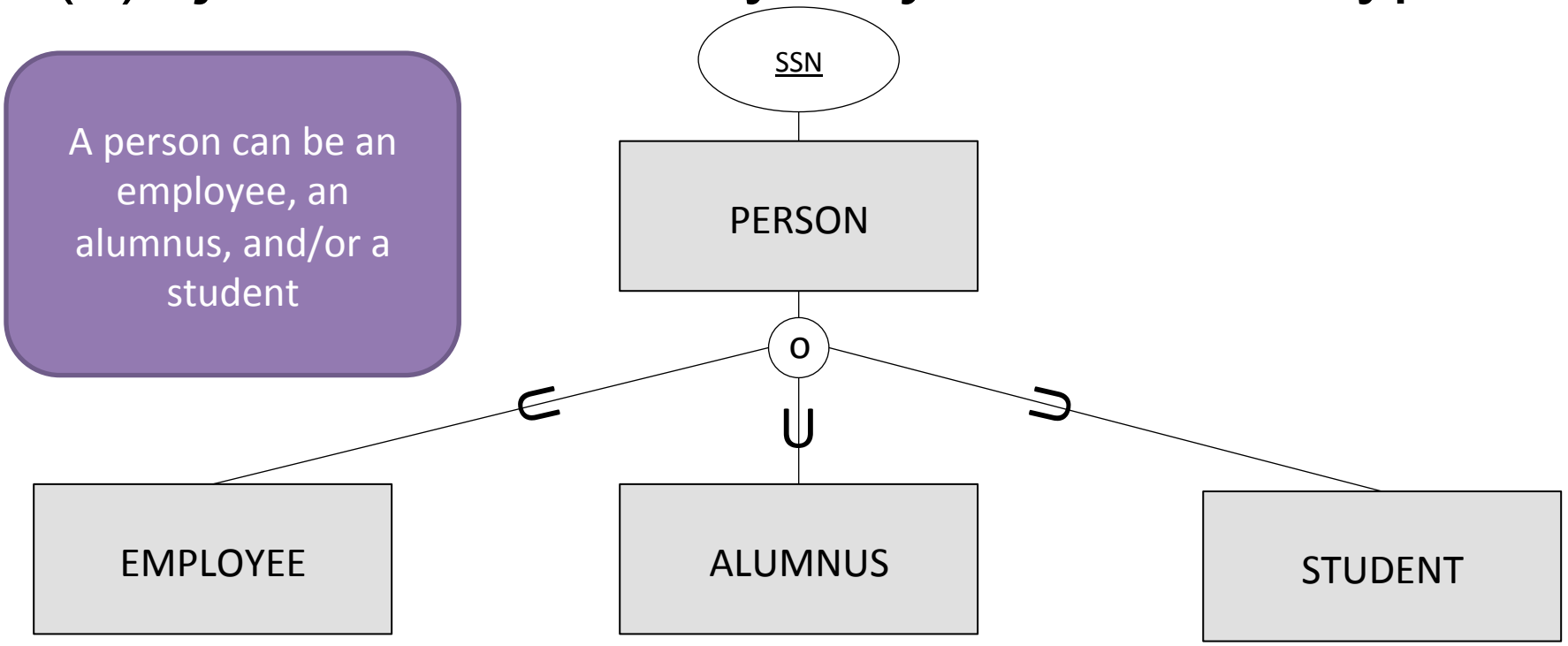
Only a subset of entities within a type have certain attributes or participate in certain relationships



Multiple Subtypes: Disjointedness

(o)verlap: may be more than one

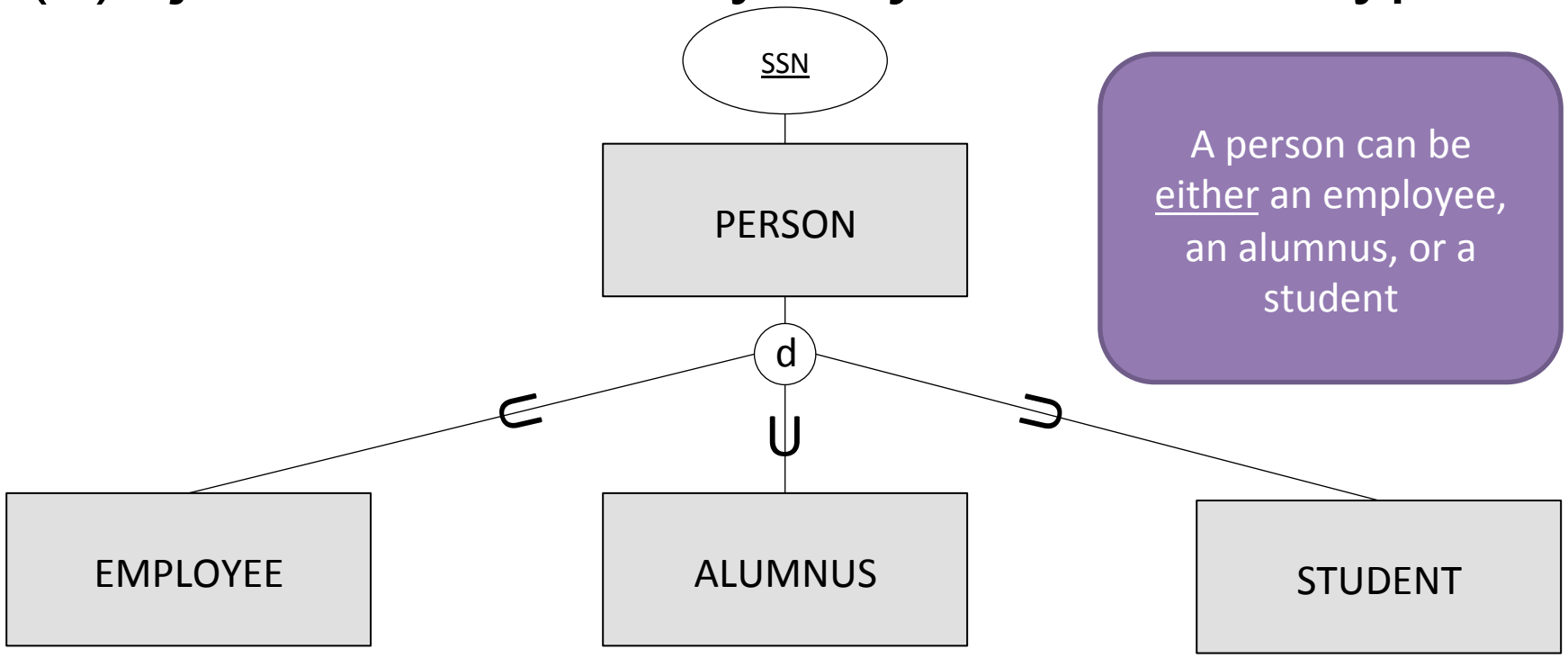
(d)isjoint: entities may *only be one* subtype



Multiple Subtypes: Disjointedness

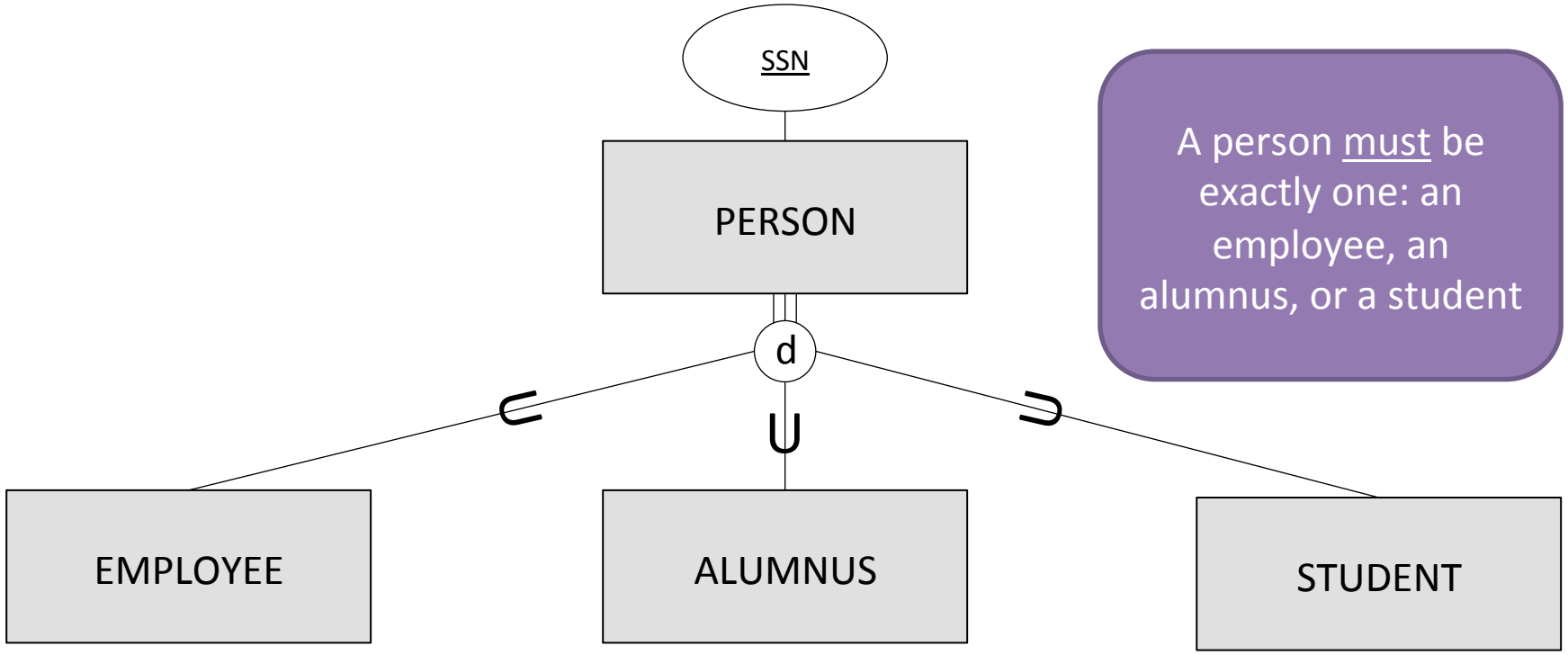
(o)verlap: may be more than one

(d)isjoint: entities may *only be one* subtype



Multiple Subtypes: Completeness

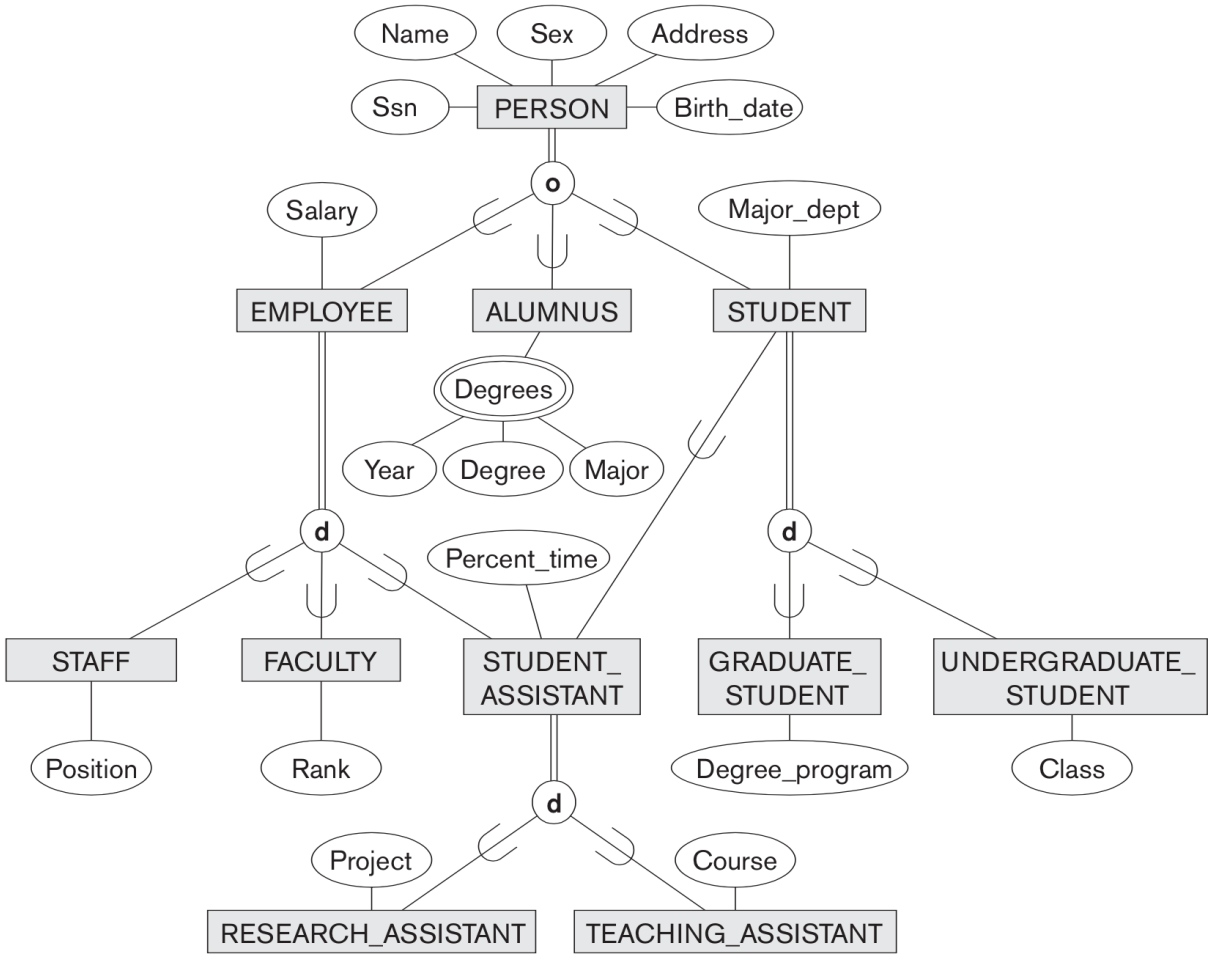
Similar to relationships; can be total (must belong to subtypes) or partial (can belong)



Exercise

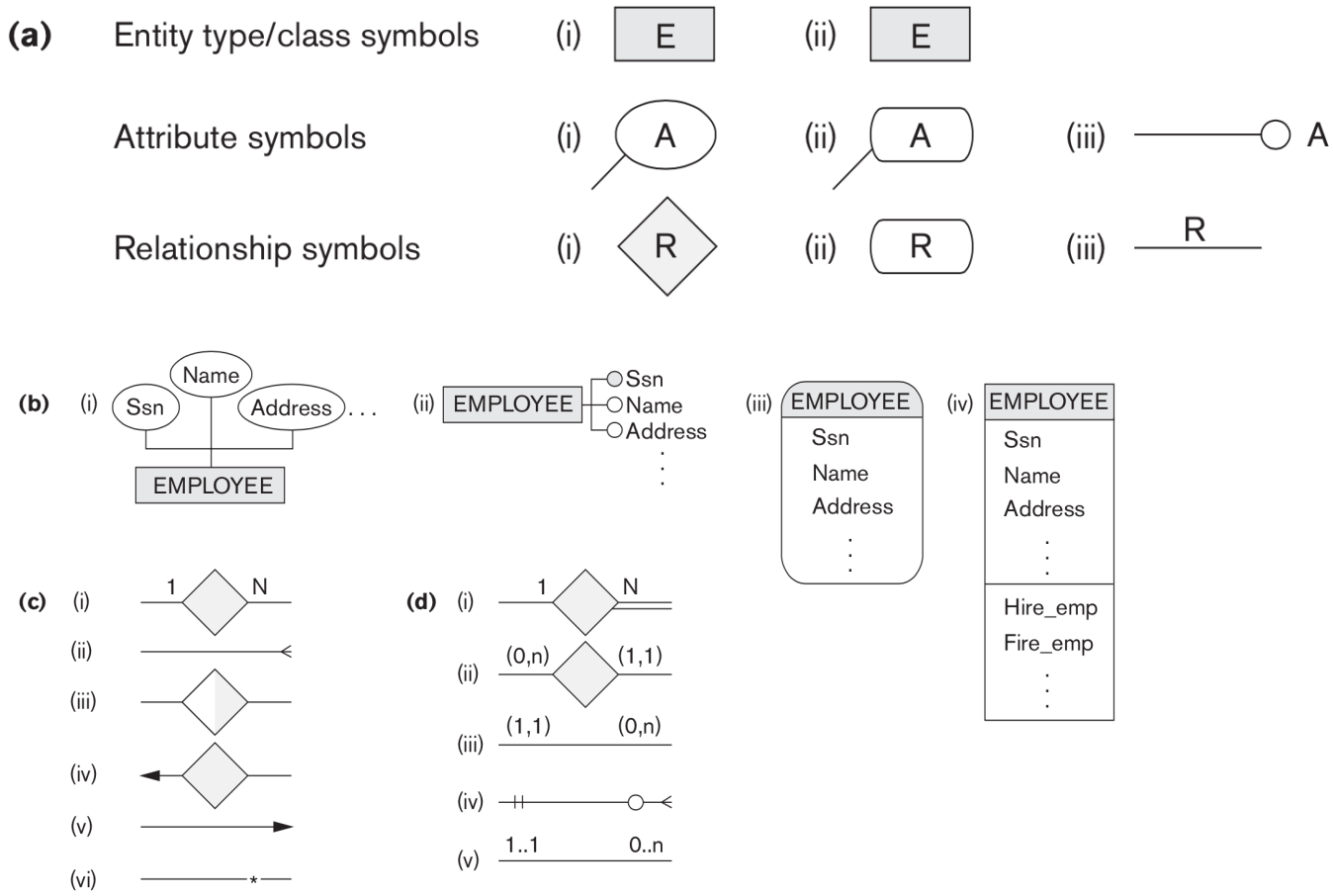
- The database keeps track of three types of persons: employees, alumni, and students. A person can belong to one, two, or all three of these types. Each person has a name, SSN, sex, address, and birth date.
- Every employee has a salary, and there are three types of employees: faculty, staff, and student assistants. Each employee belongs to exactly one of these types. For each alumnus, a record of the degree or degrees that he or she earned at the university is kept, including the name of the degree, the year granted, and the major department. Each student has a major department.
- Each faculty has a rank, whereas each staff member has a staff position. Student assistants are classified further as either research assistants or teaching assistants, and the percent of time that they work is recorded in the database. Research assistants have their research project stored, whereas teaching assistants have the current course they work on.
- Students are further classified as either graduate or undergraduate, with the specific attributes degree program (M.S., Ph.D., M.B.A., and so on) for graduate students and class (freshman, sophomore, and so on) for under- graduates.

Answer


Name
Sex
Address
Ssn
PERSON
Birth_date
o
Salary
Major_dept
EMPLOYEE
ALUMNUS
STUDENT
Degrees
Year
Degree
Major
d
Percent_time
d
STAFF
FACULTY
STUDENT_ ASSISTANT
GRADUATE_ STUDENT
UNDERGRADUATE_ STUDENT
Position
Rank
Degree_program
Class
d
Project
Course
RESEARCH_ASSISTANT
TEACHING_ASSISTANT

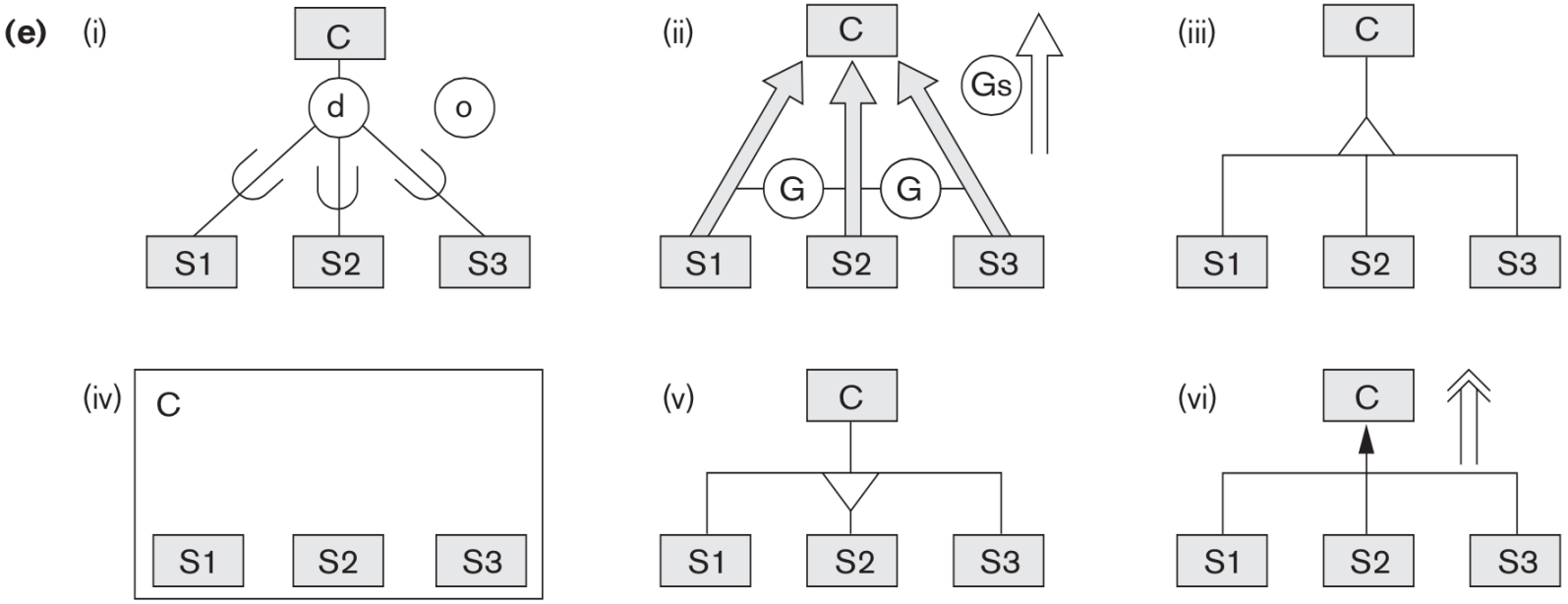
Alternative Notation (1)

Figure A. 1



Alternative Notation (2)

Figure A. 1



Requirements Elicitation

The conceptual model should *inform* requirements elicitation questions:

- What are the main kinds of objects to be stored in the database (entity types)?
- For each object, what information should be stored (attributes, relationships)? What information distinguishes one object of a type from another (keys, weak entities)? Are there different kinds/ categories of objects (specialization/generalization)?
- For each piece of information, what characterizes a valid value (composite/multi-valued, structural, etc.)?
- For related objects x and y, can x exist without y (participation)? How many x's can a y have, and vice-versa (cardinality)?

Approaches to Conceptual Design

Centralized

- Single authority responsible for merging requirements into schema
- Reasonable for smaller applications

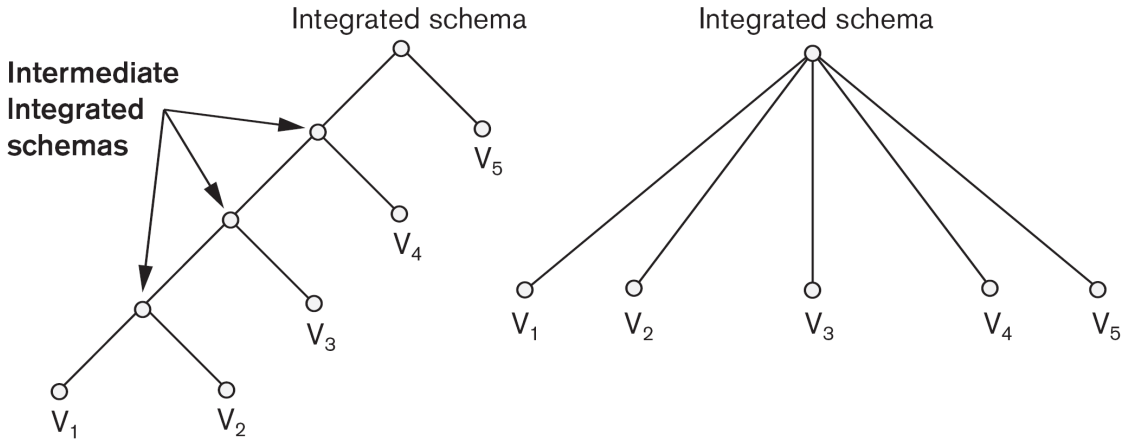
View Integration

- Each stakeholder implements local view
- Individual views integrated into global schema
- Individual views can be reconstructed as external schemas after integration

View Integration (1)

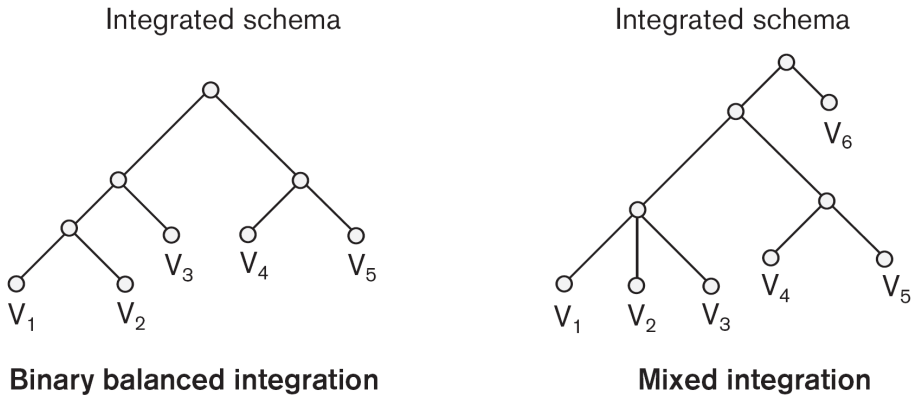
1. Identify correspondences and conflicts
   - Conflicts: names, types, domain, constraints
2. Modify views to conform
3. Merge
4. Restructure

View Integration (2)



Binary ladder integration

N-ary integration



Binary balanced integration

Mixed integration