# Entity-Relationship (ER) Diagrams
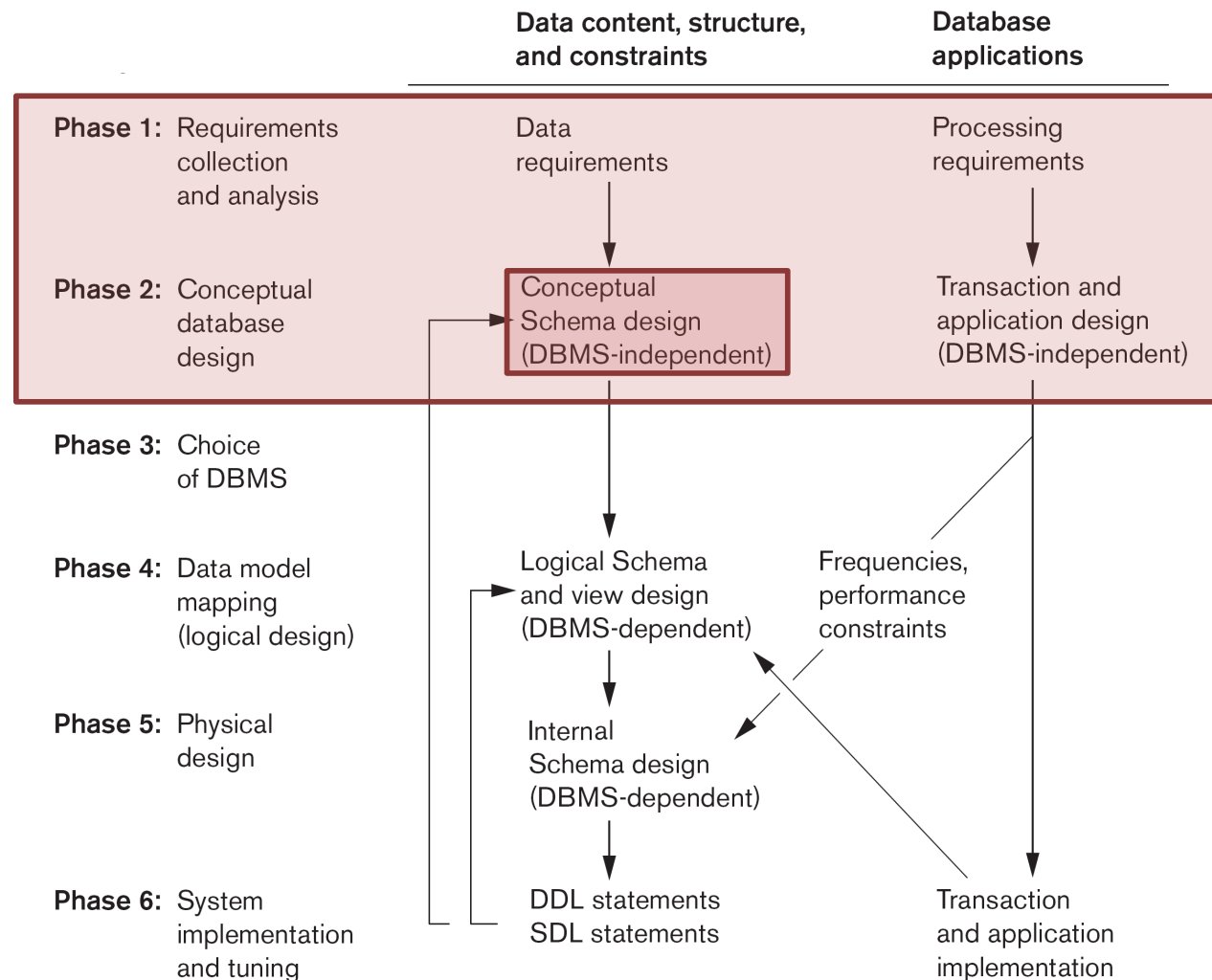
## Lecture 7

# 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

| Phase | | Data content, structure, and constraints | Database applications |
|---|---|---|---|
| **Phase 1:** | Requirements collection and analysis | Data requirements | Processing requirements |
| **Phase 2:** | Conceptual database design | Conceptual Schema design (DBMS-independent) | Transaction and application design (DBMS-independent) |
| **Phase 3:** | Choice of DBMS | | |
| **Phase 4:** | Data model mapping (logical design) | Logical Schema and view design (DBMS-dependent) | Frequencies, performance constraints |
| **Phase 5:** | Physical design | Internal Schema design (DBMS-dependent) | |
| **Phase 6:** | System implementation and tuning | DDL statements SDL statements | Transaction and application implementation |

# Goal of Conceptual Design

Description of data requirements that is…

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

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

<u>Algorithmically Transformable</u>
- Improves application development efficiency, reduces errors

**Entity-Relationship (ER) Diagrams**

# 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

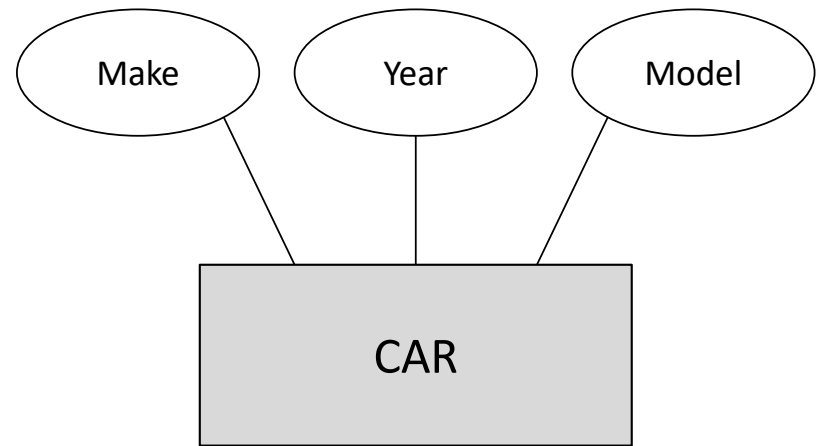- Graphical depiction of an ER model

- Many notations, this class…

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

| Symbol | Meaning |
|---|---|
| | Entity |
| | Weak Entity |
| | Relationship |
| | Indentifying Relationship |
| | Attribute |
| | Key Attribute |
| | Multivalued Attribute |
| | Composite Attribute |
| | Derived Attribute |
| $E_1$ — R — $E_2$ | Total Participation of $E_2$ in R |
| $E_1$ — 1 — R — N — $E_2$ | Cardinality Ratio 1: N for $E_1$:$E_2$ in R |
| R — (min, max) — E | Structural Constraint (min, max) on Participation of E in R |
| C / d / o / S1 S2 S3 | Specialization/Generalization |

**Entity-Relationship (ER) Diagrams**
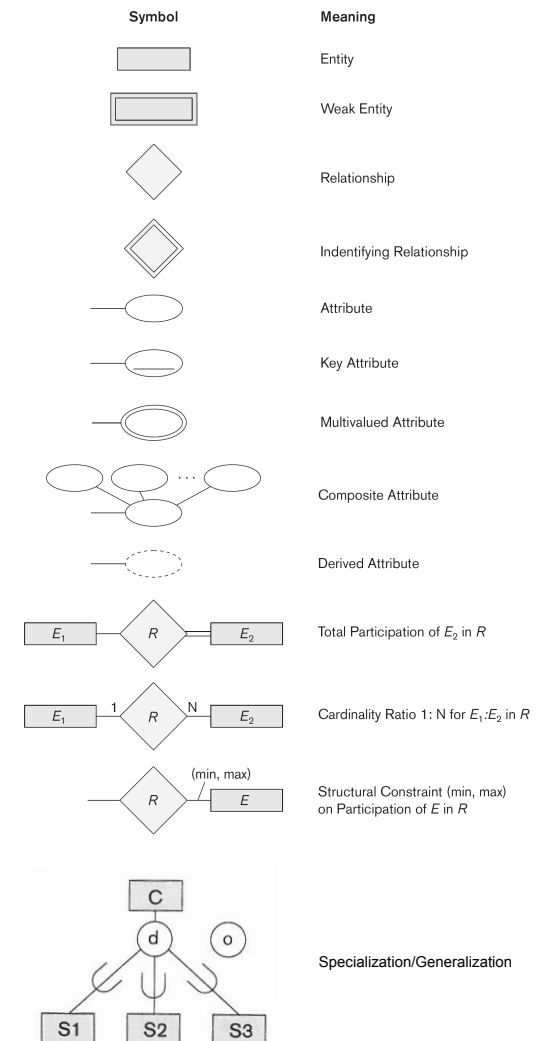
# Entity Sets

Set of entities that have the same attributes

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

```
   ( Make )     ( Year )     ( Model )
       \           |            /
        \          |           /
     ┌─────────────────────────────┐
     │            CAR              │
     │                             │
     └─────────────────────────────┘
```

# Composite Attributes

Can be subdivided into smaller subparts

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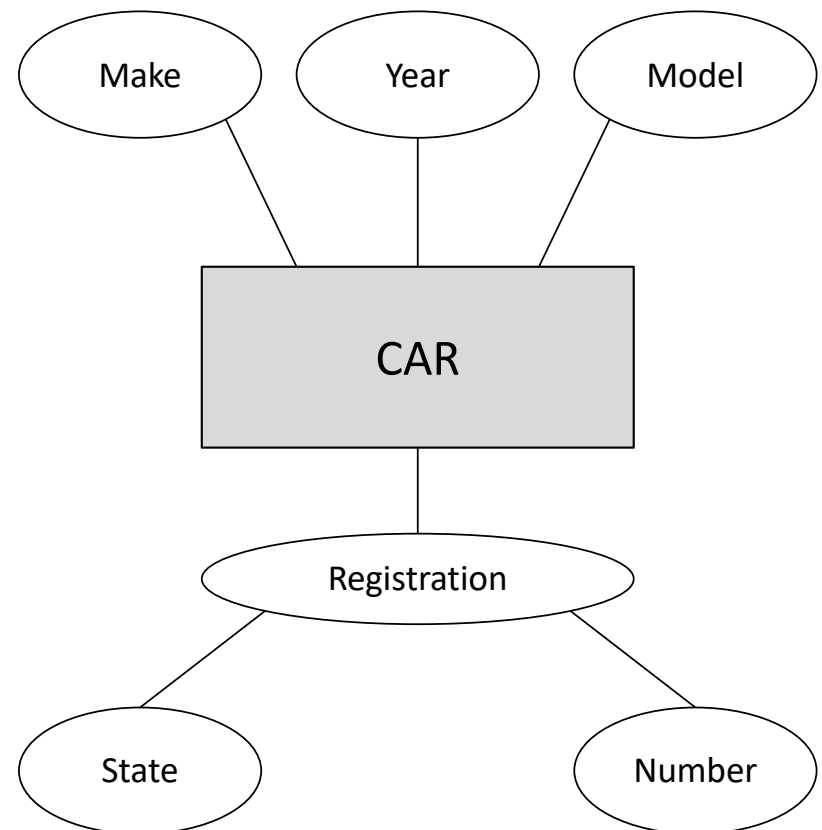
# Multivalued Attributes

Can take a [possibly specified] number of values.

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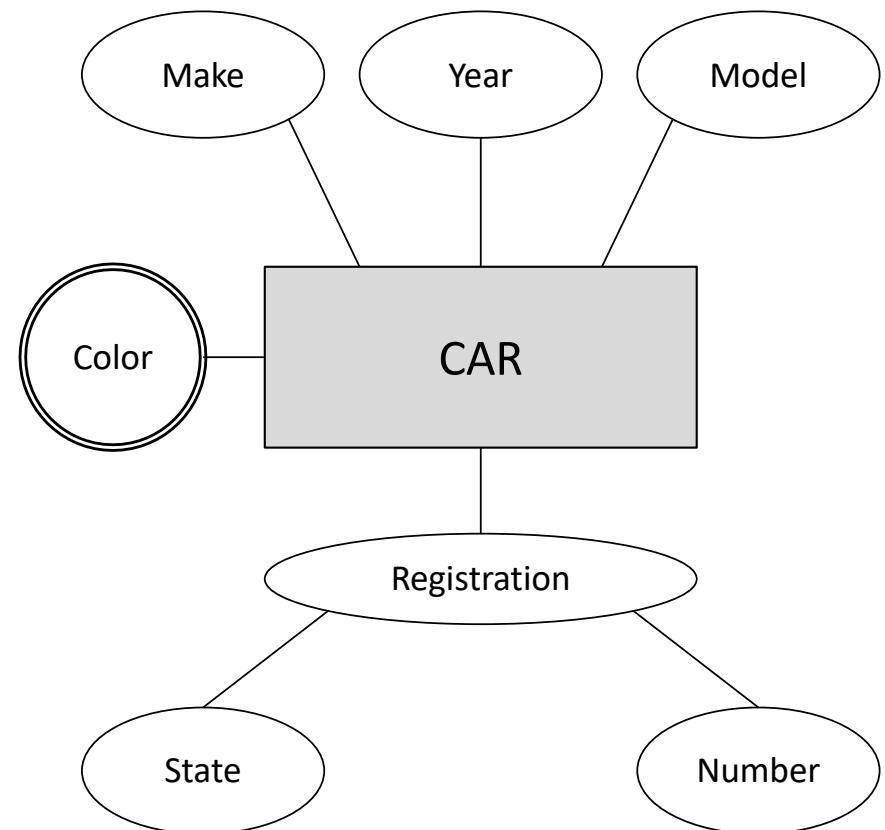
# Key Attributes

The value uniquely identifies each entity

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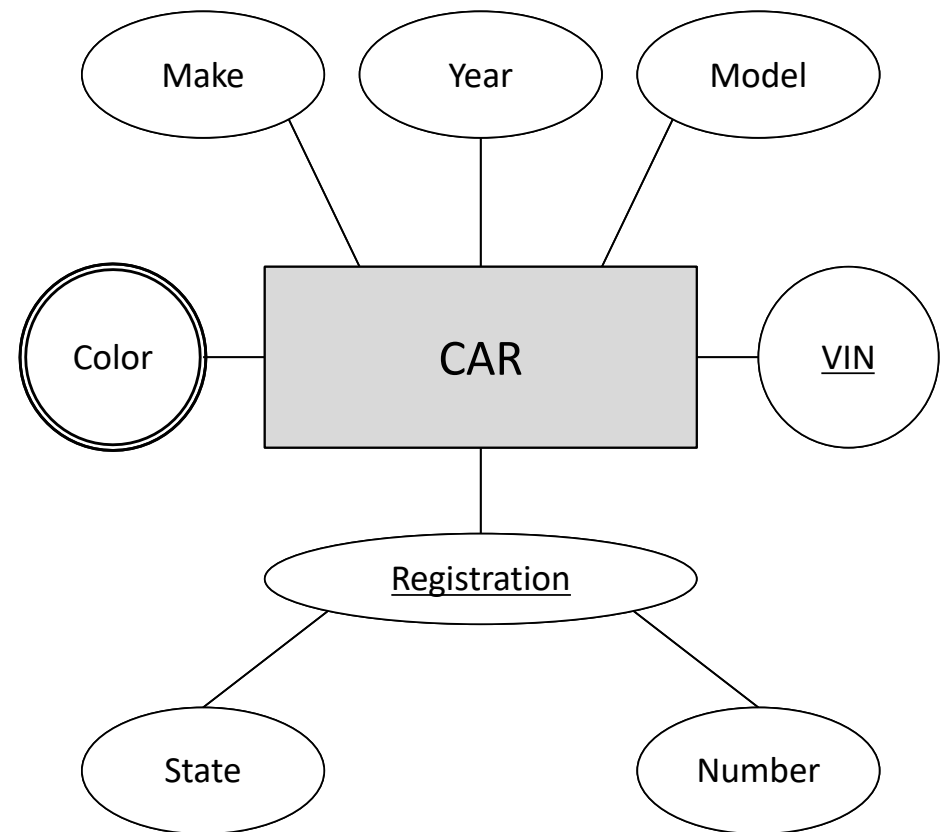
# Potential Pitfall

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

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

**Entity-Relationship (ER) Diagrams**

# 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.*
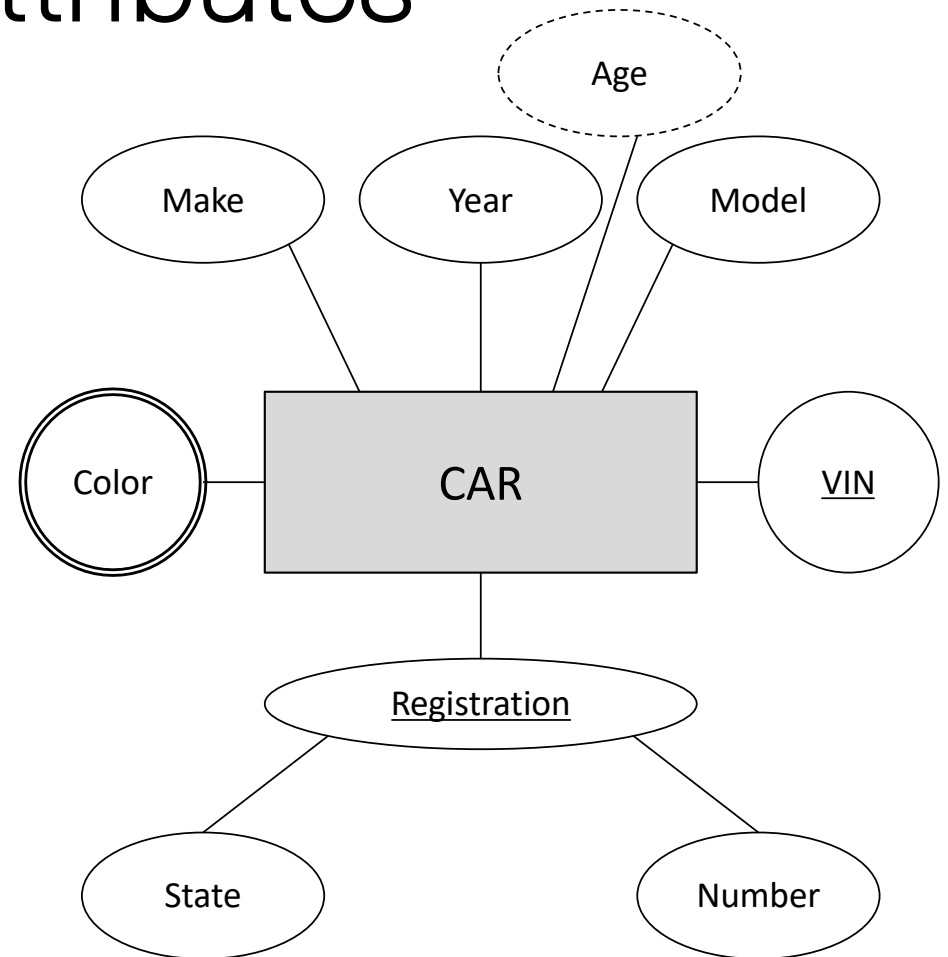
# Exercise

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
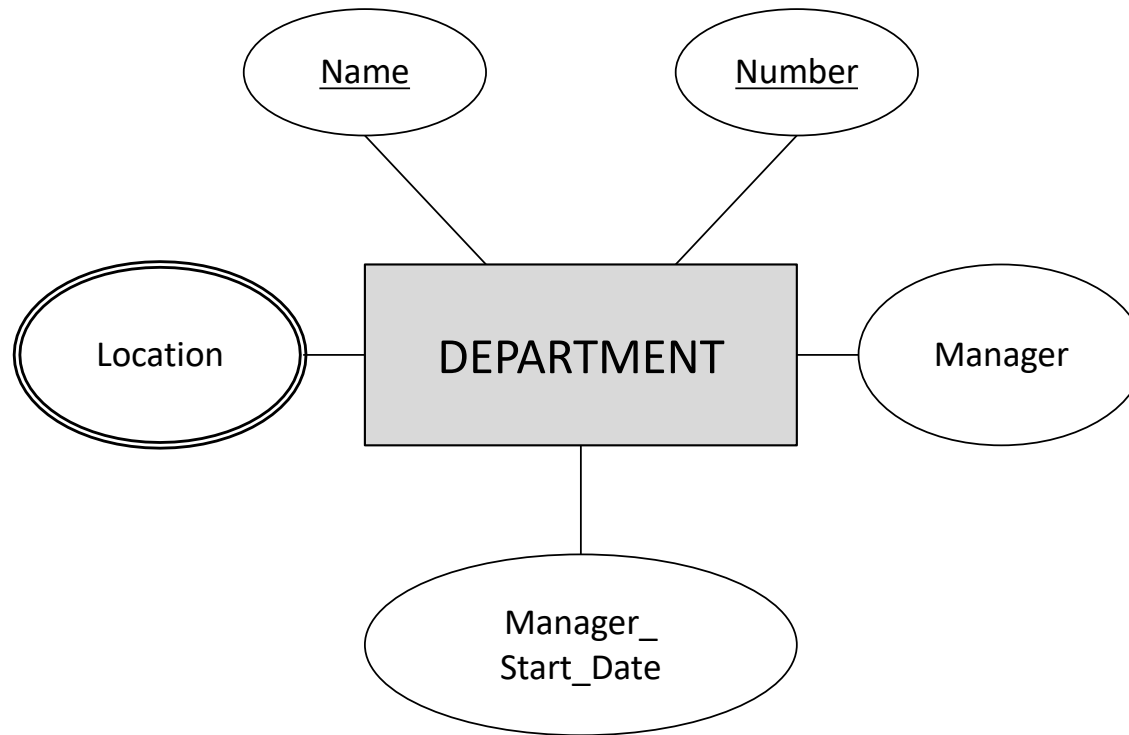
# Exercise

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
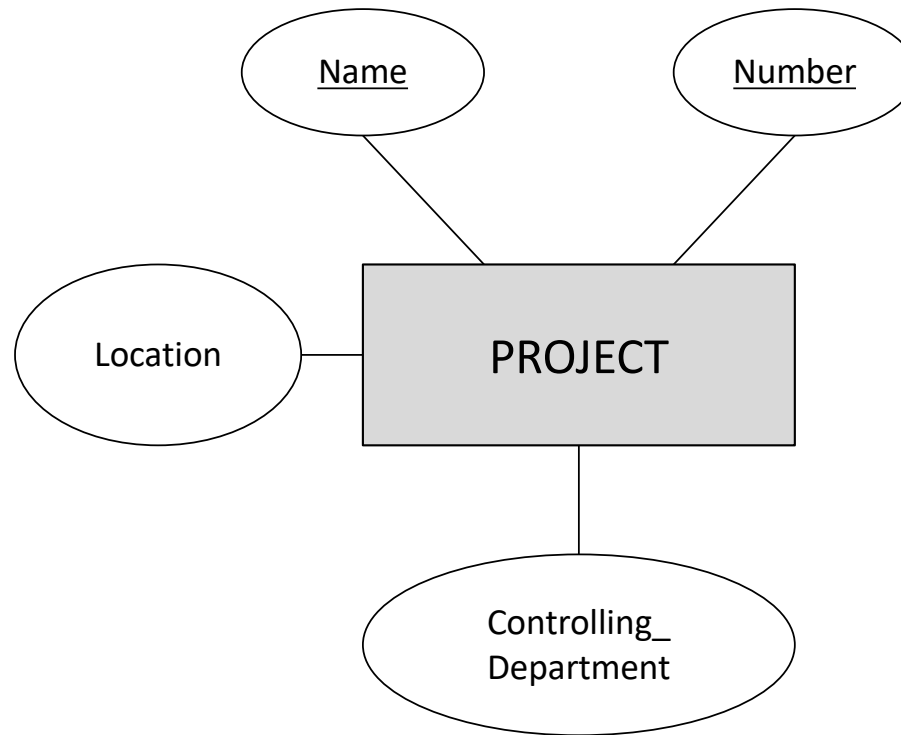
# Exercise

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).*

**Entity-Relationship (ER) Diagrams**
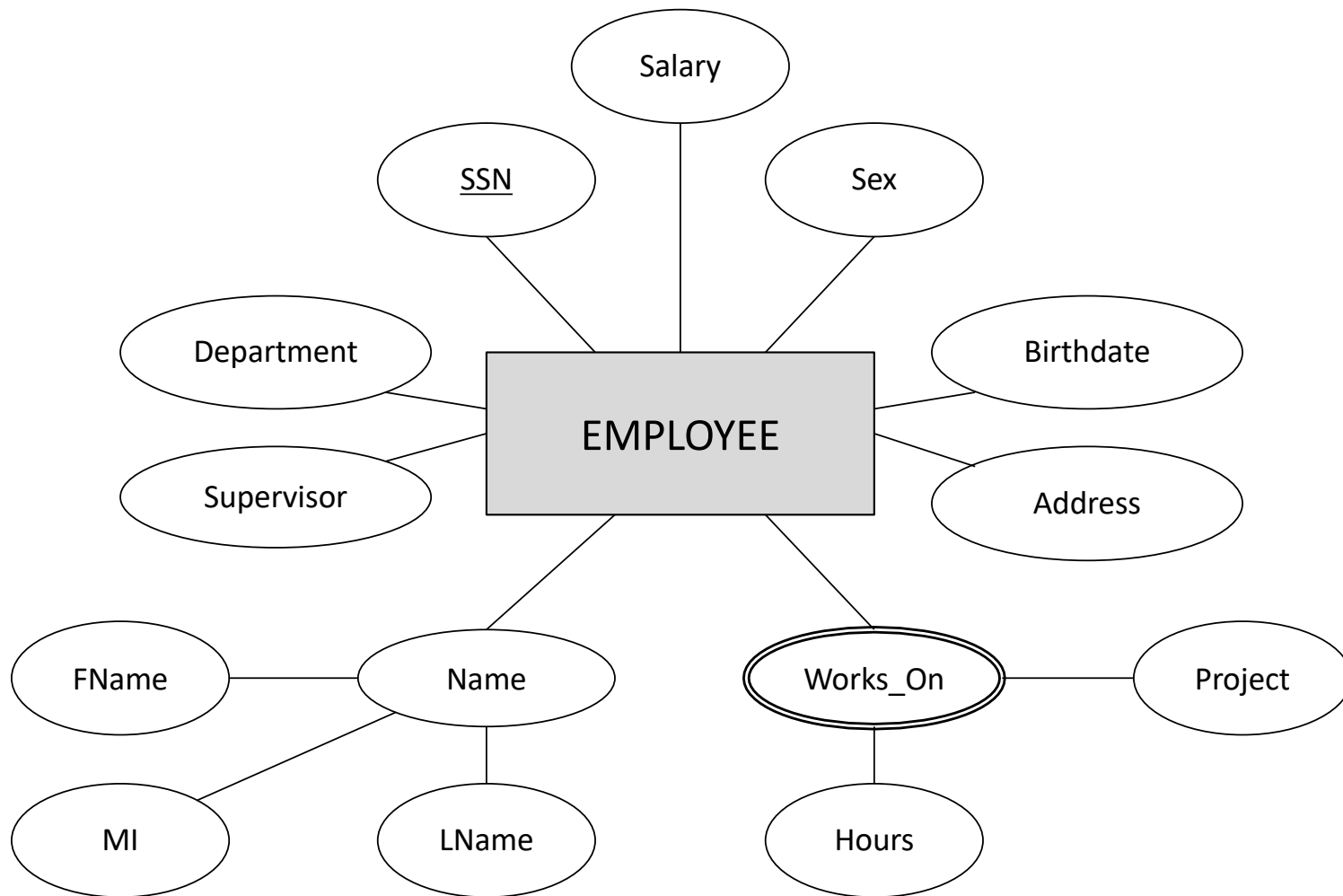
# Answer

# 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

```
                          ┌──────┐
                          │ Sex  │
                          └──────┘
                             │
   ┌──────────┐       ┌─────────────┐       ┌────────────┐
   │ Employee │       │             │       │ DBirthdate │
   └──────────┘───────│  DEPENDENT  │───────└────────────┘
   ┌──────────────┐   │             │
   │ Relationship │───└─────────────┘
   └──────────────┘         │
                       ┌────────┐
                       │ DName  │
                       └────────┘
```

# Relationships

## Associates one or more sets of entities
### – One = recursive (**role** is important)

STUDENT

DEPT
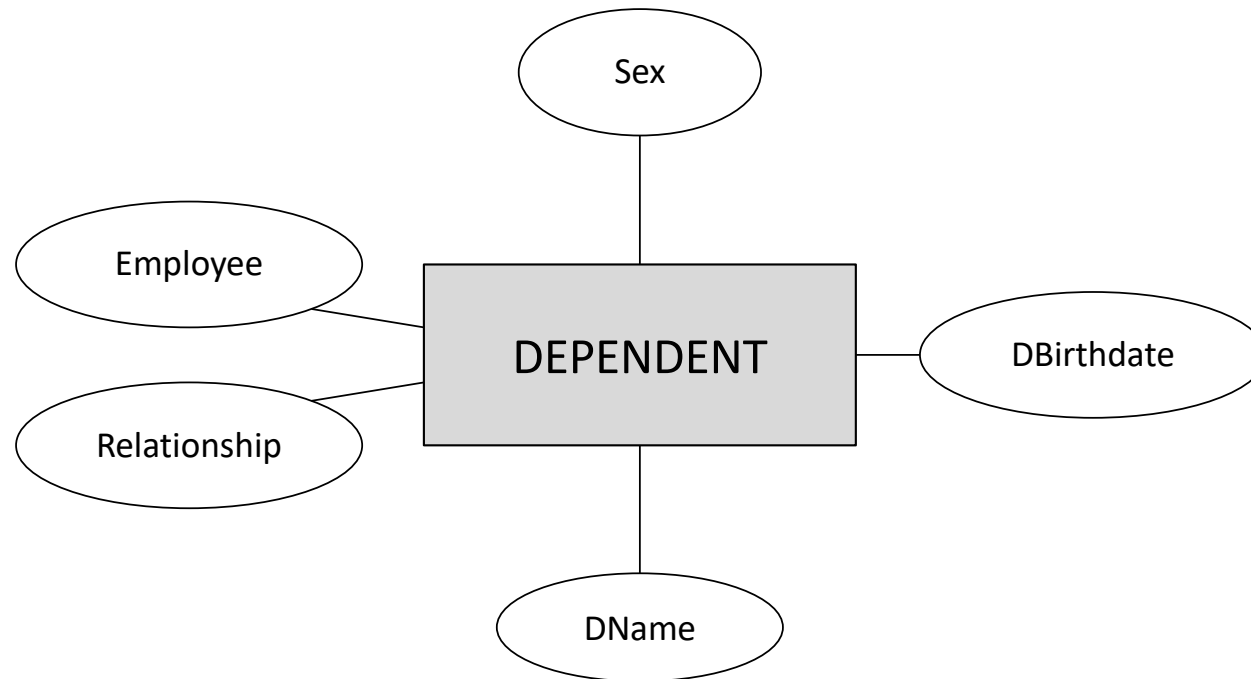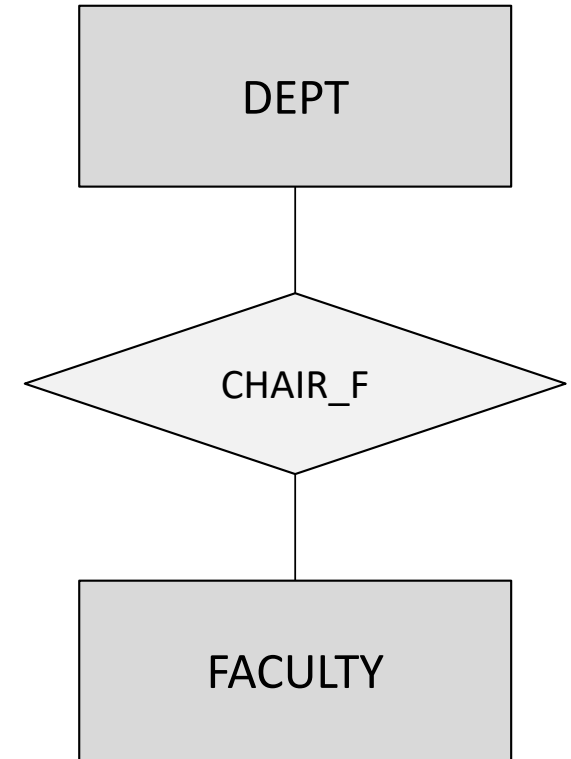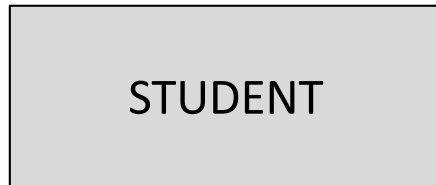
CHAIR_F

FACULTY

*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)
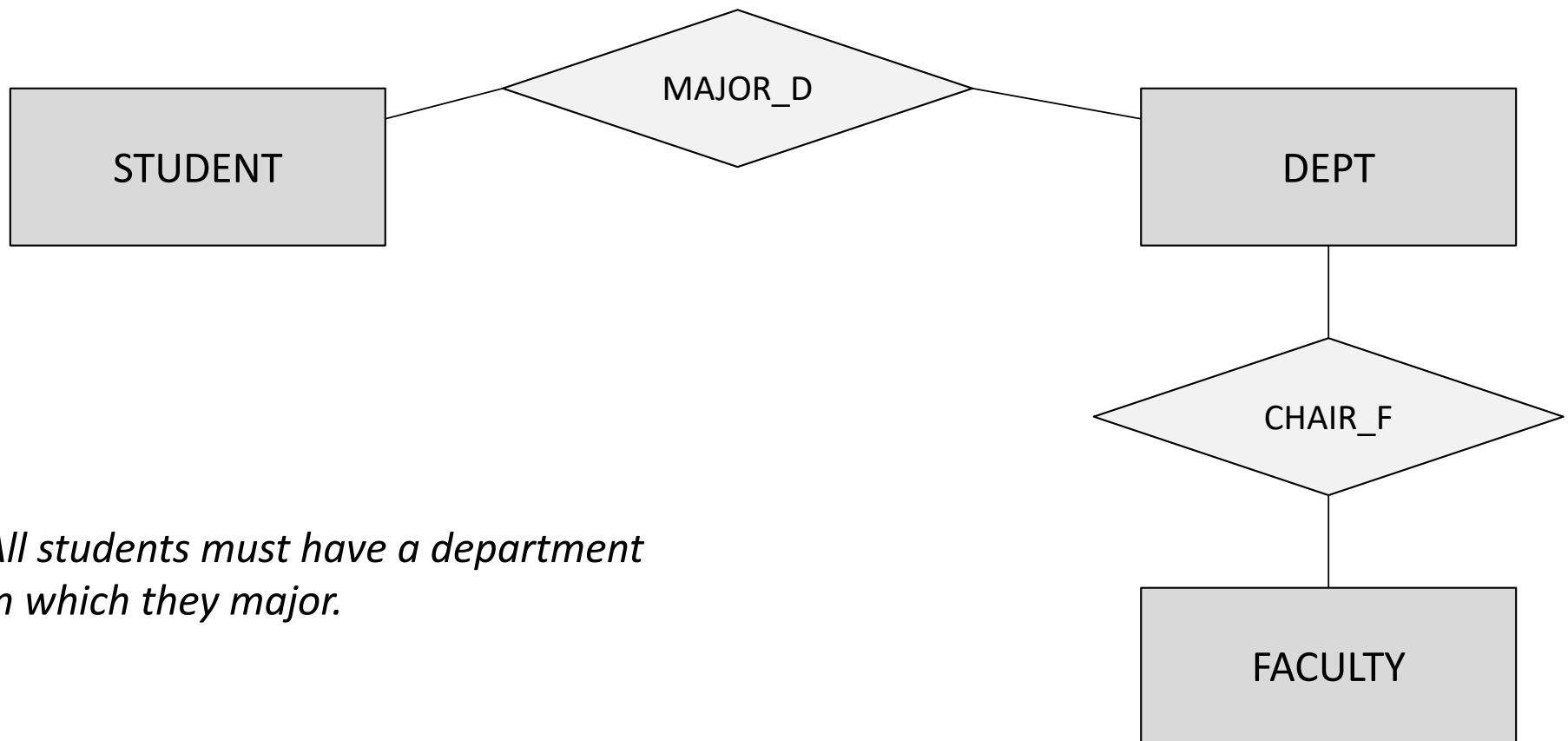


*All students must have a department in which they major.*

# Relationships

## Associates one or more sets of entities
### – One = recursive (**role** is important)
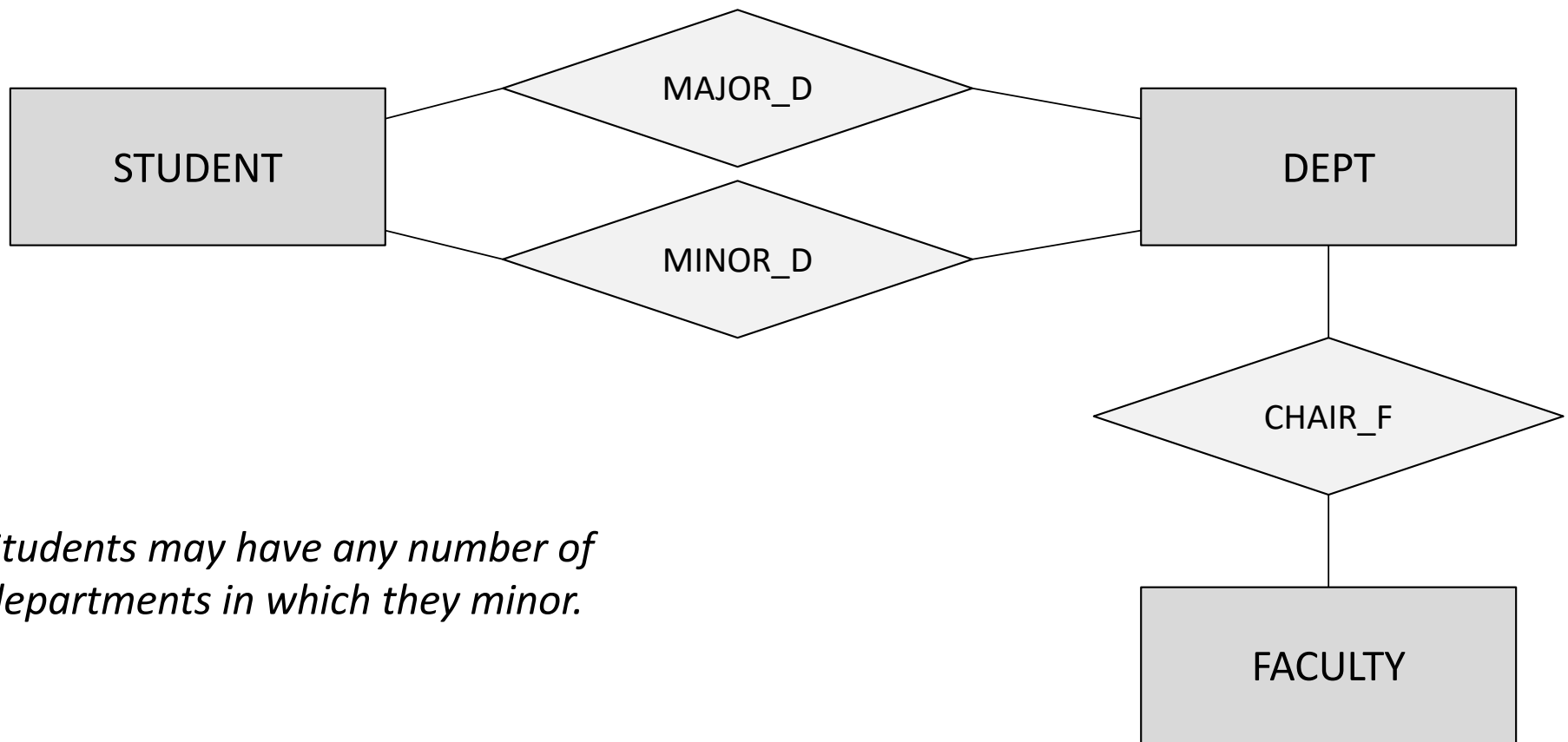


*Students may have any number of departments in which they minor.*

# Relationships

## Associates one or more sets of entities
### – One = recursive (**role** is important)
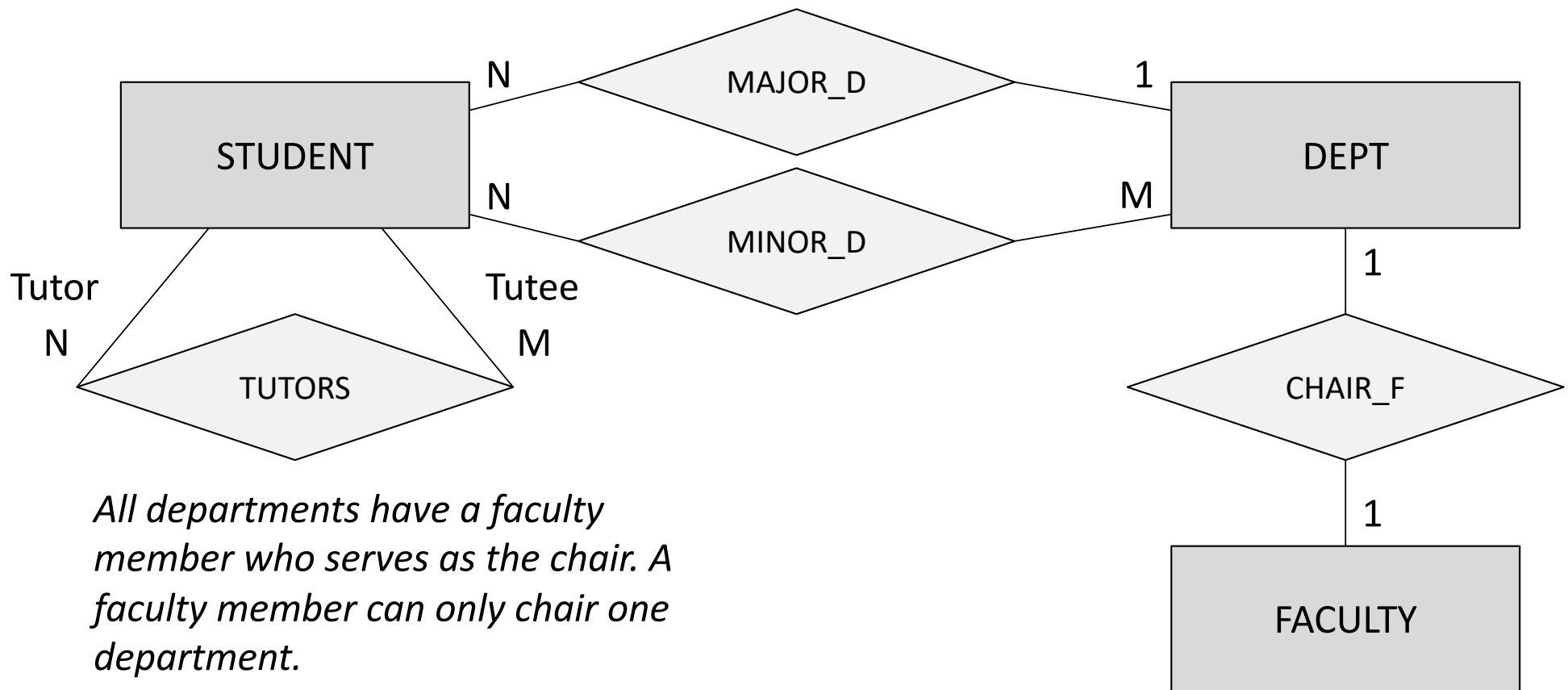


Tutor          Tutee

*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

STUDENT

N — MAJOR_D — 1 — DEPT

N — MINOR_D — M

Tutor          Tutee
N                M

TUTORS

Office — CHAIR_F

DEPT — 1 — CHAIR_F — 1 — FACULTY

*Each department chair has an office.*

# Attributes of Relationships

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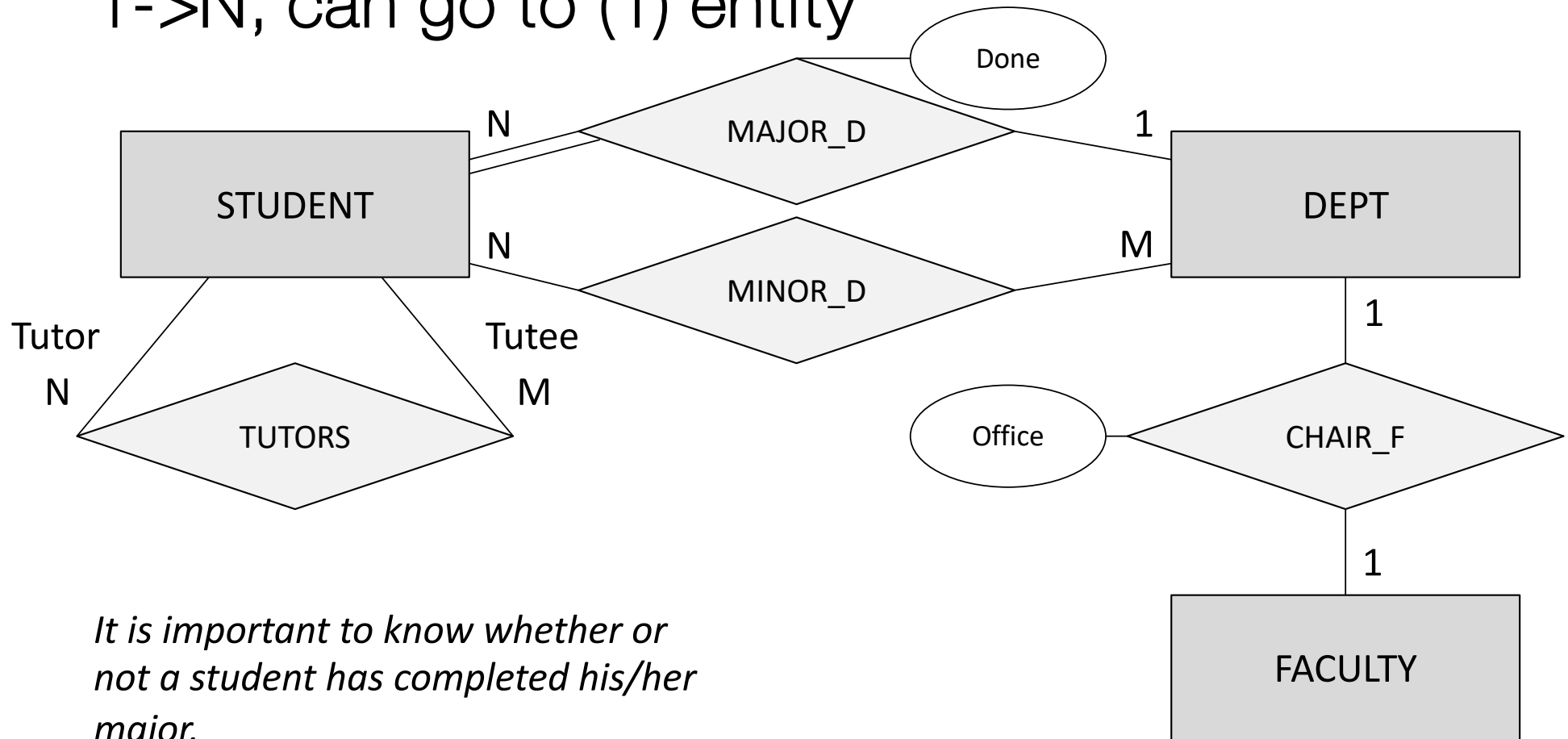
1->N, can go to (1) entity



*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
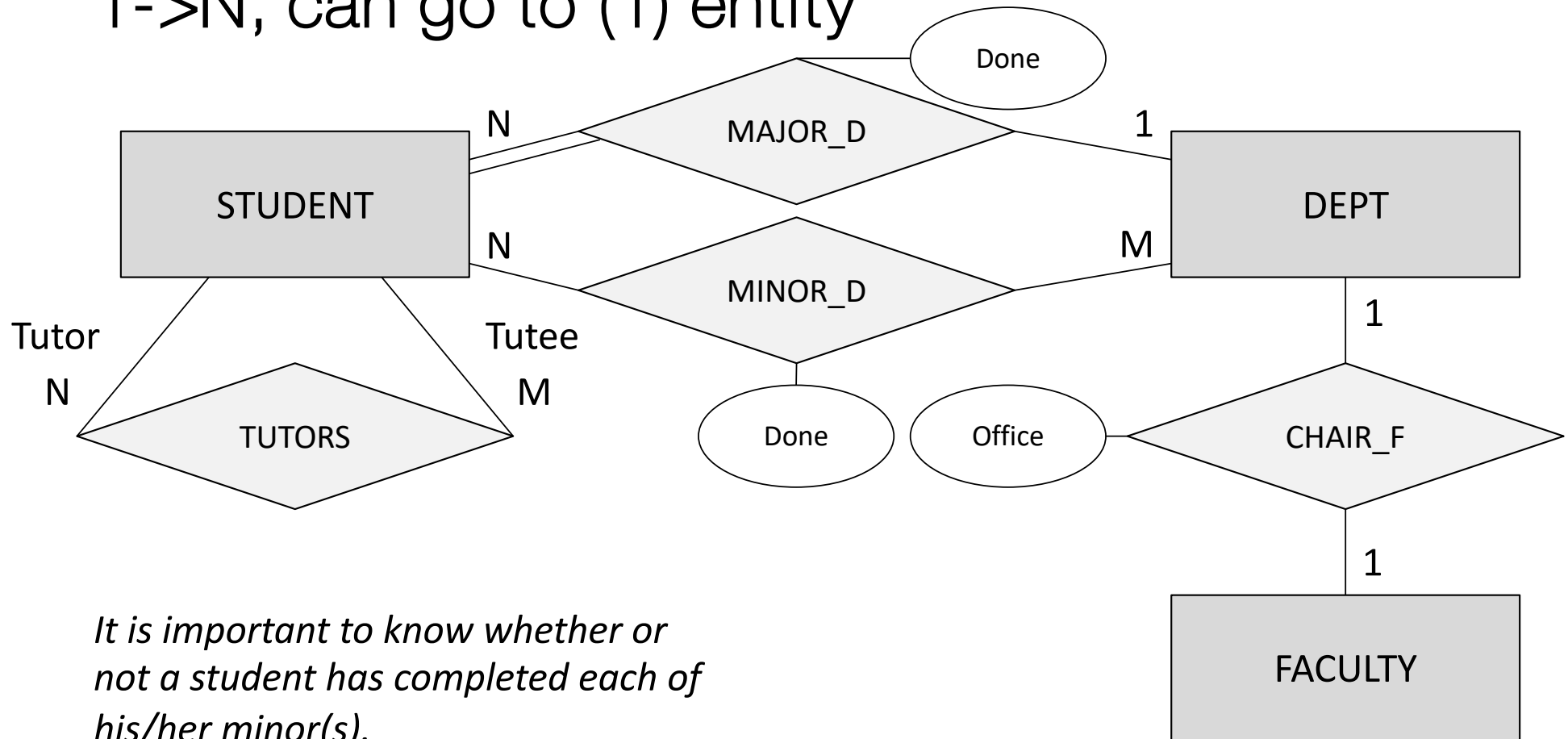


*It is important to know whether or not a student has completed each of his/her minor(s).*

# Attributes of Relationships

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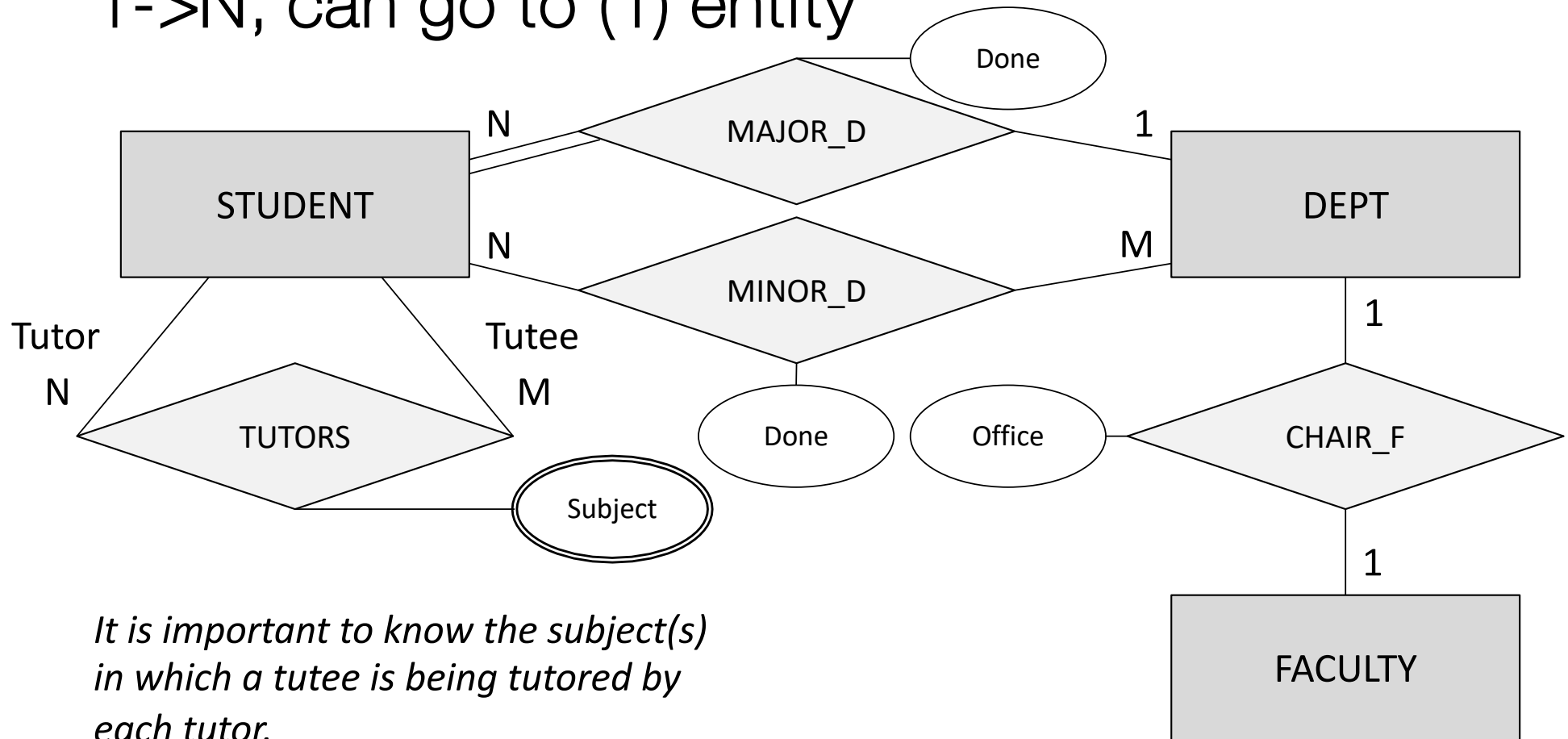


*It is important to know the subject(s) in which a tutee is being tutored by each tutor.*

# 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)
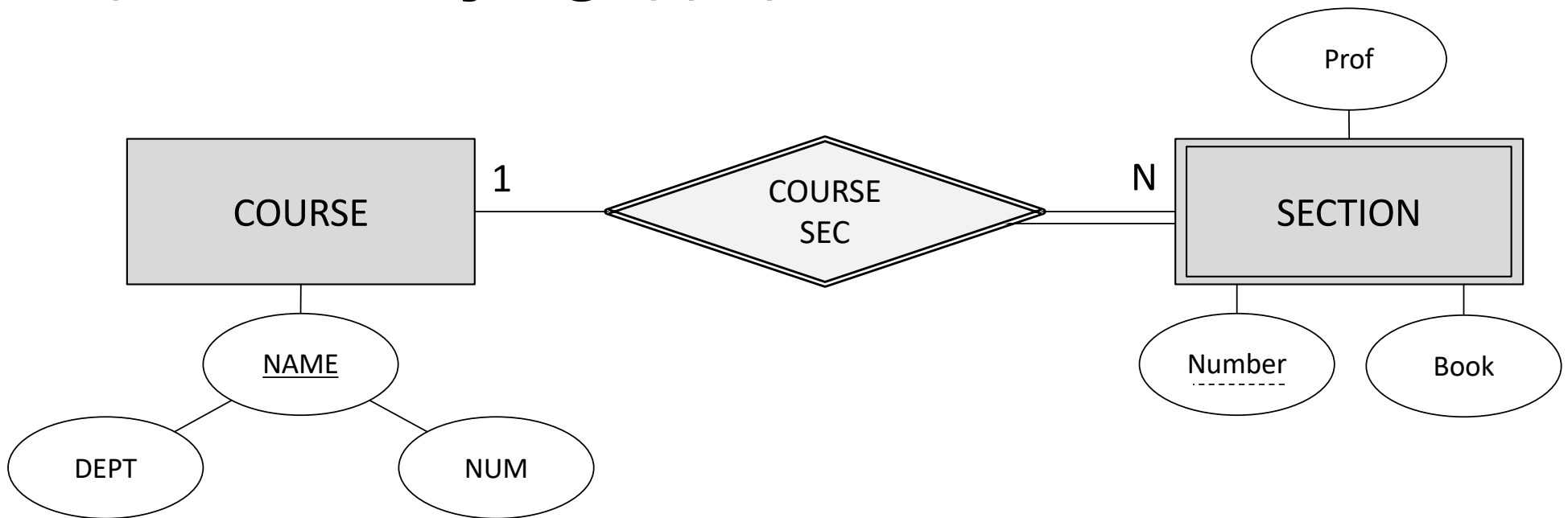
# Revise!

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).**
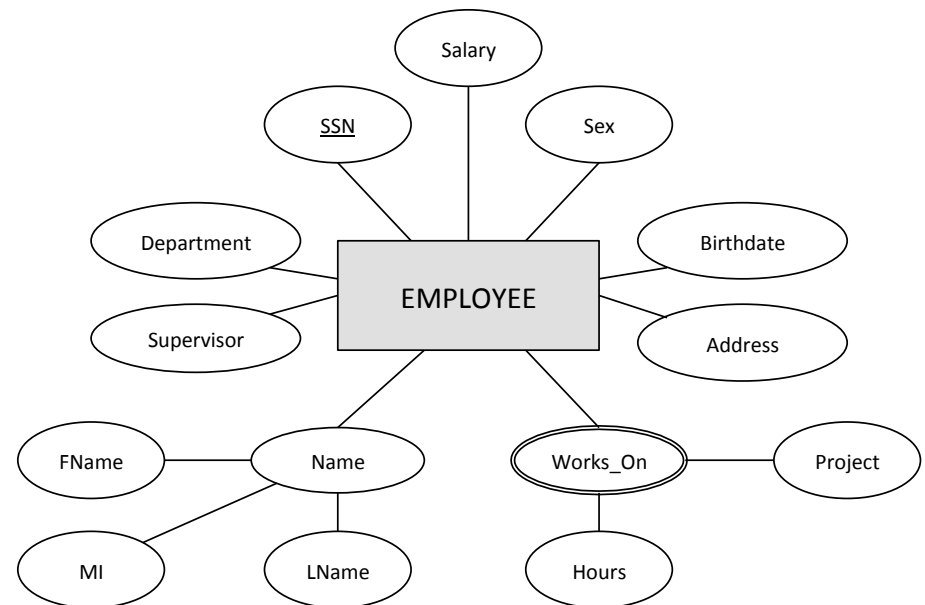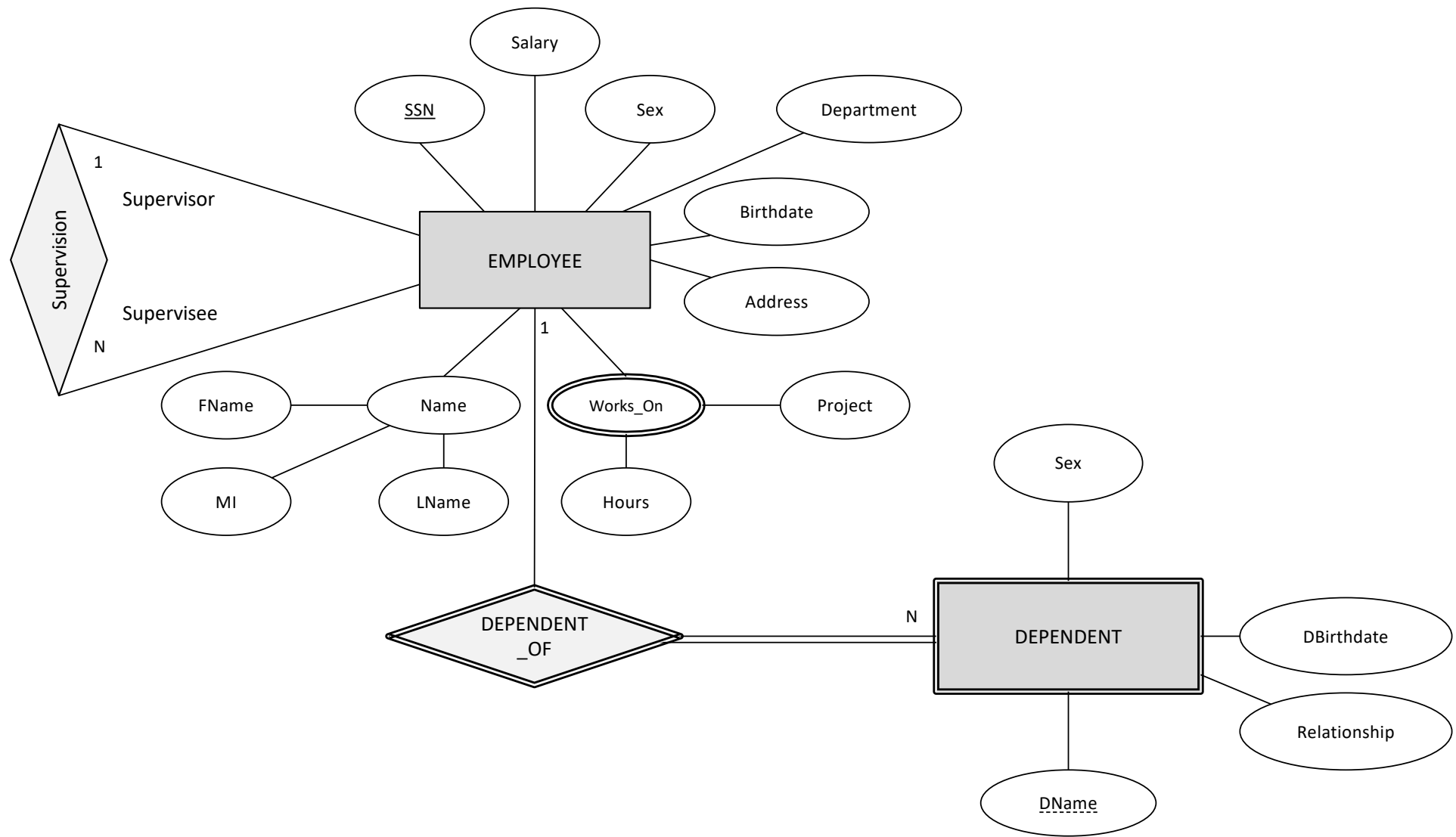
# Answer
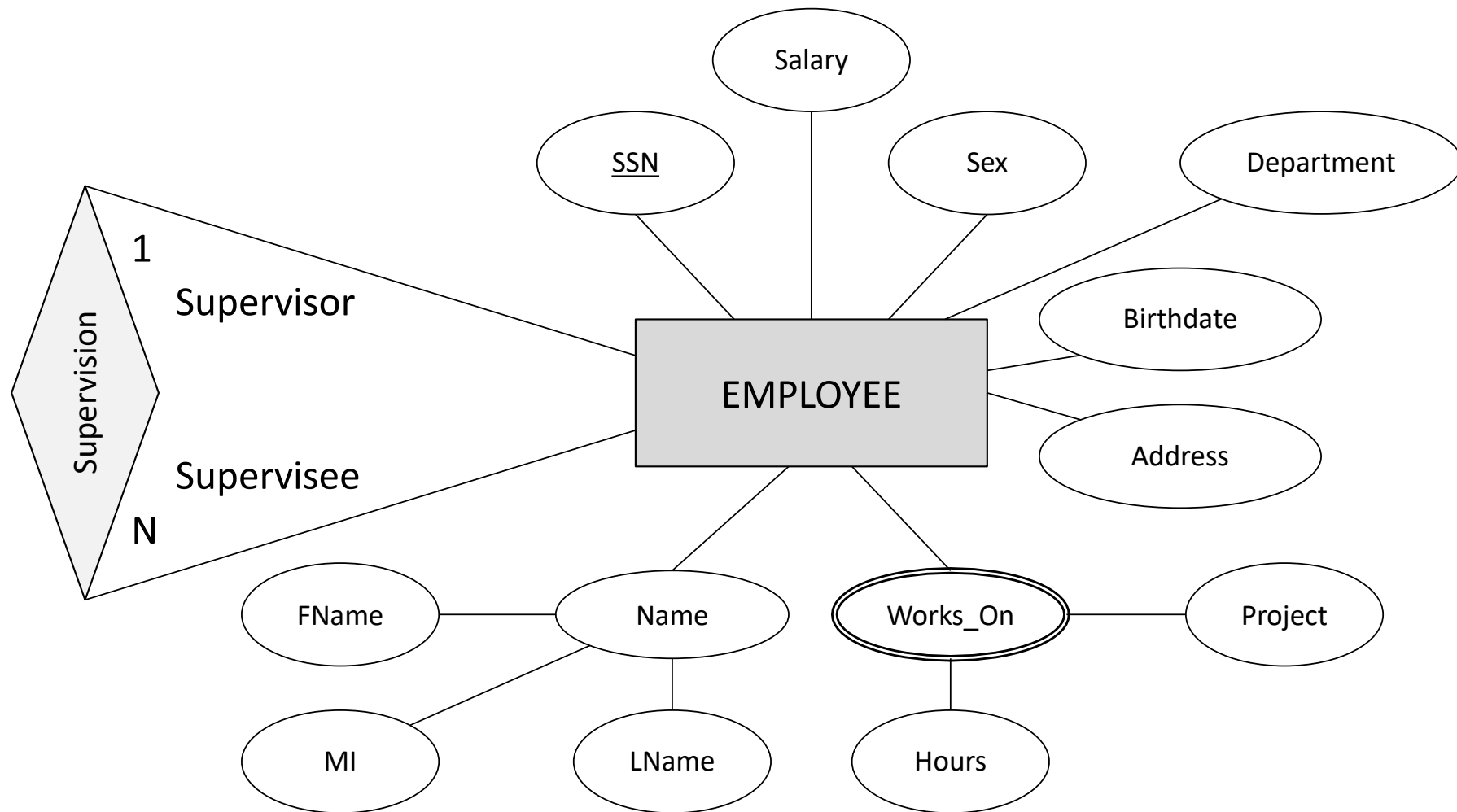
# Revise!

*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.*
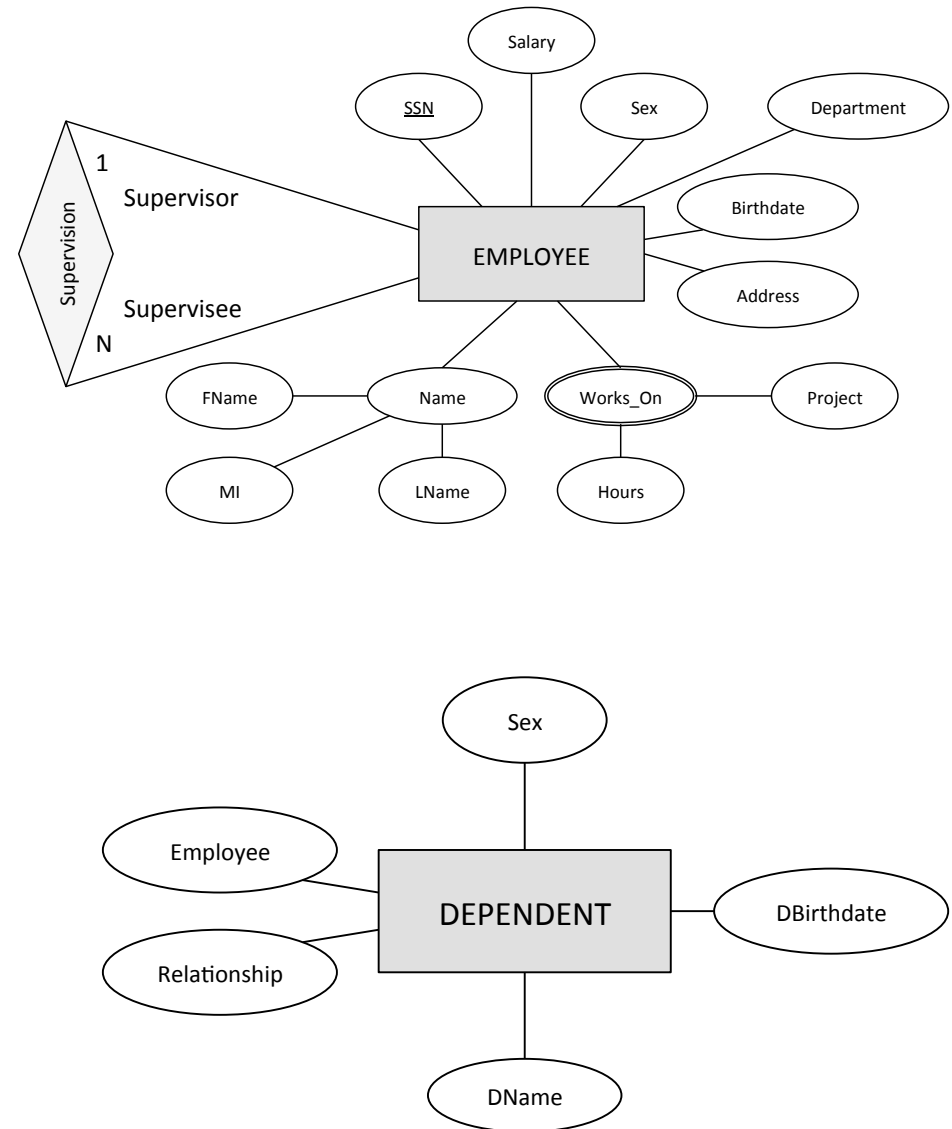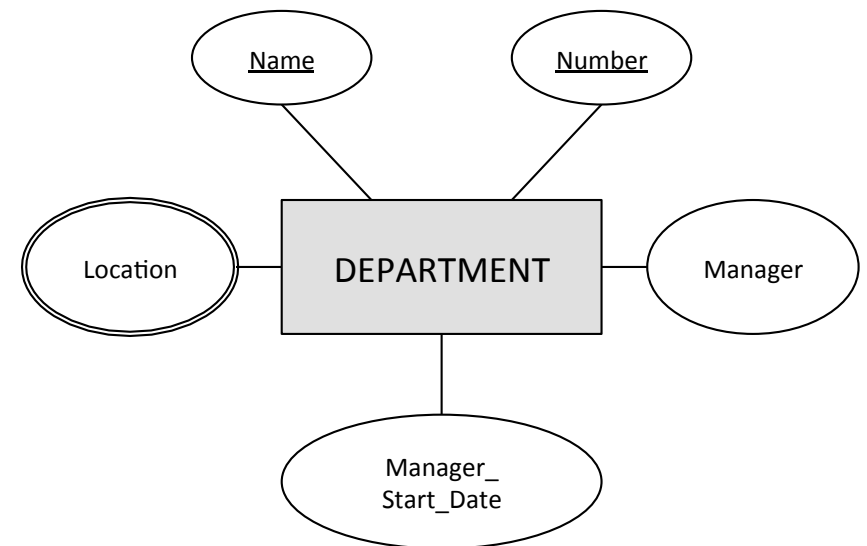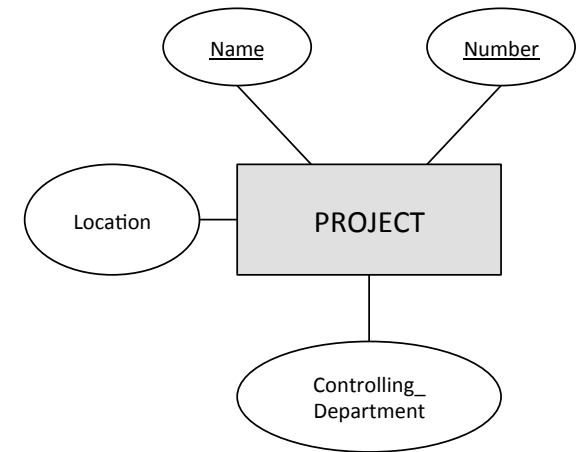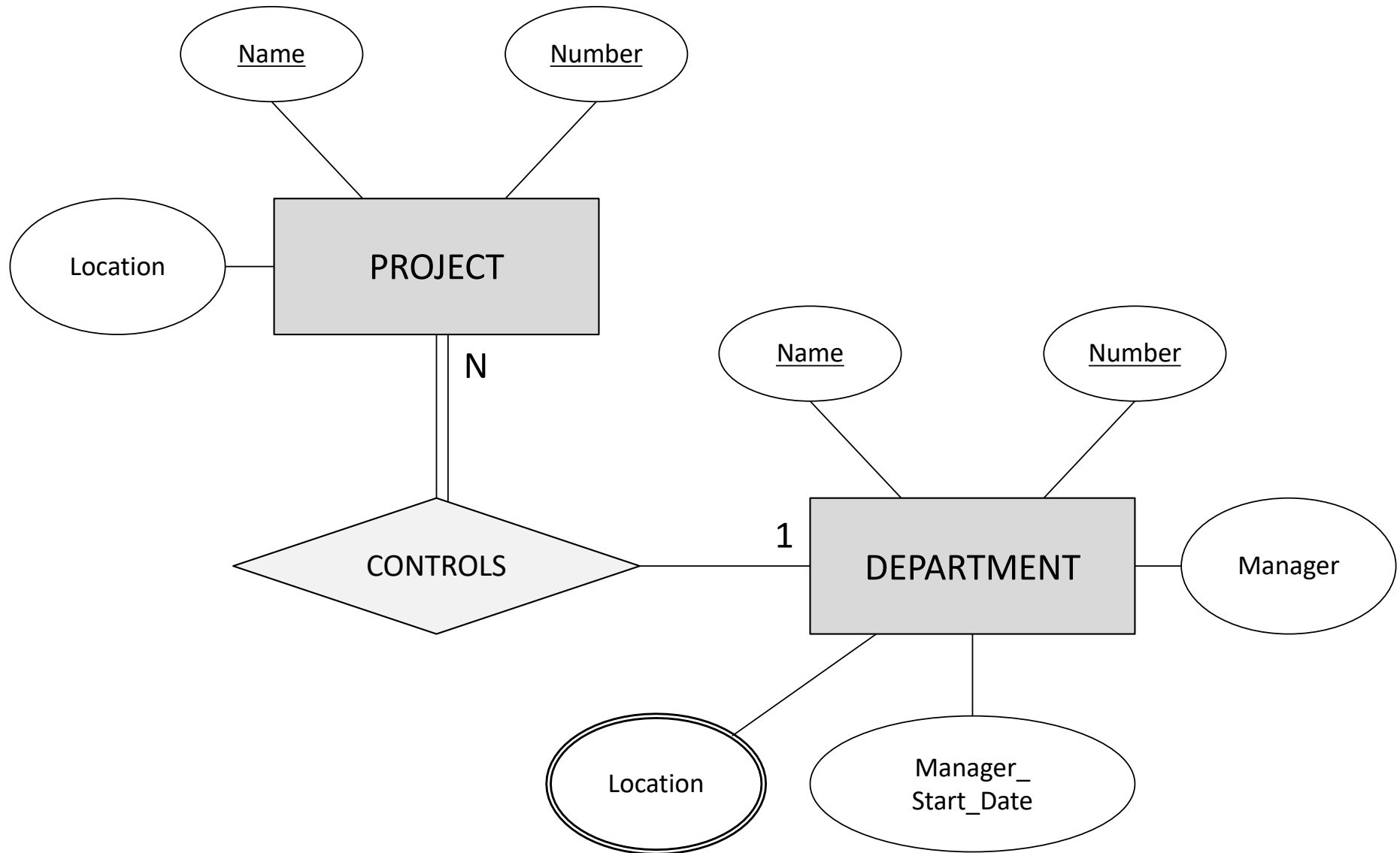
# Answer

# Revise!

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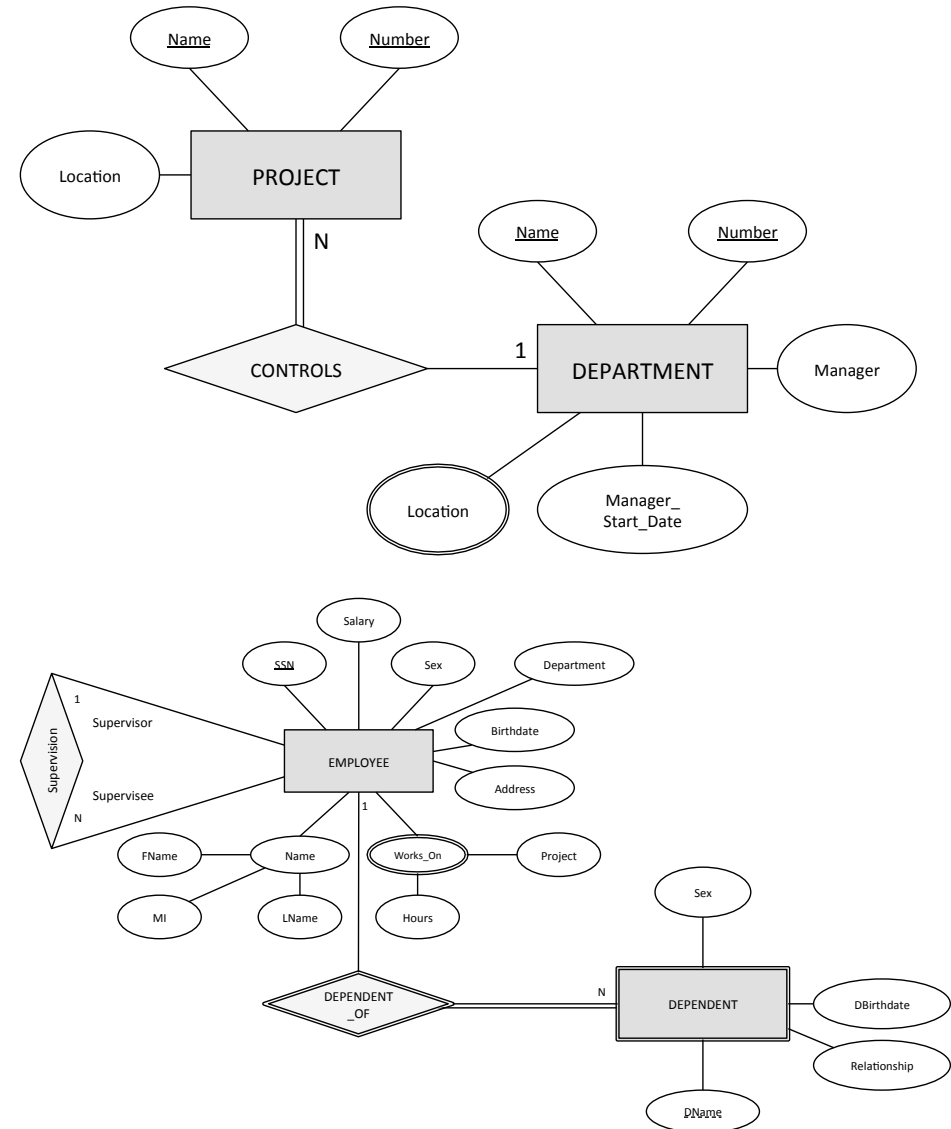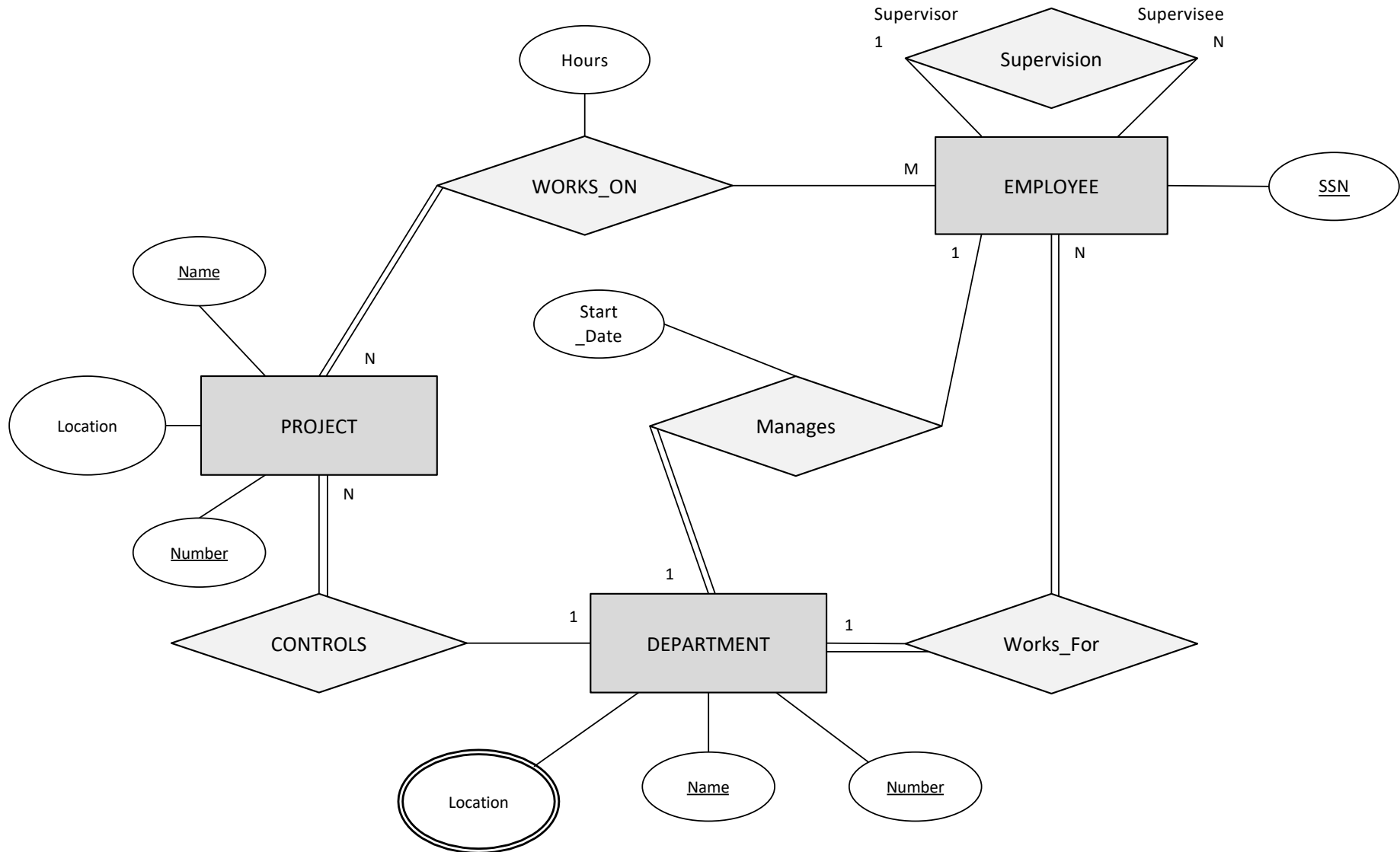
# Answer

# Revise!

*Each department has a … particular employee who manages the department.*
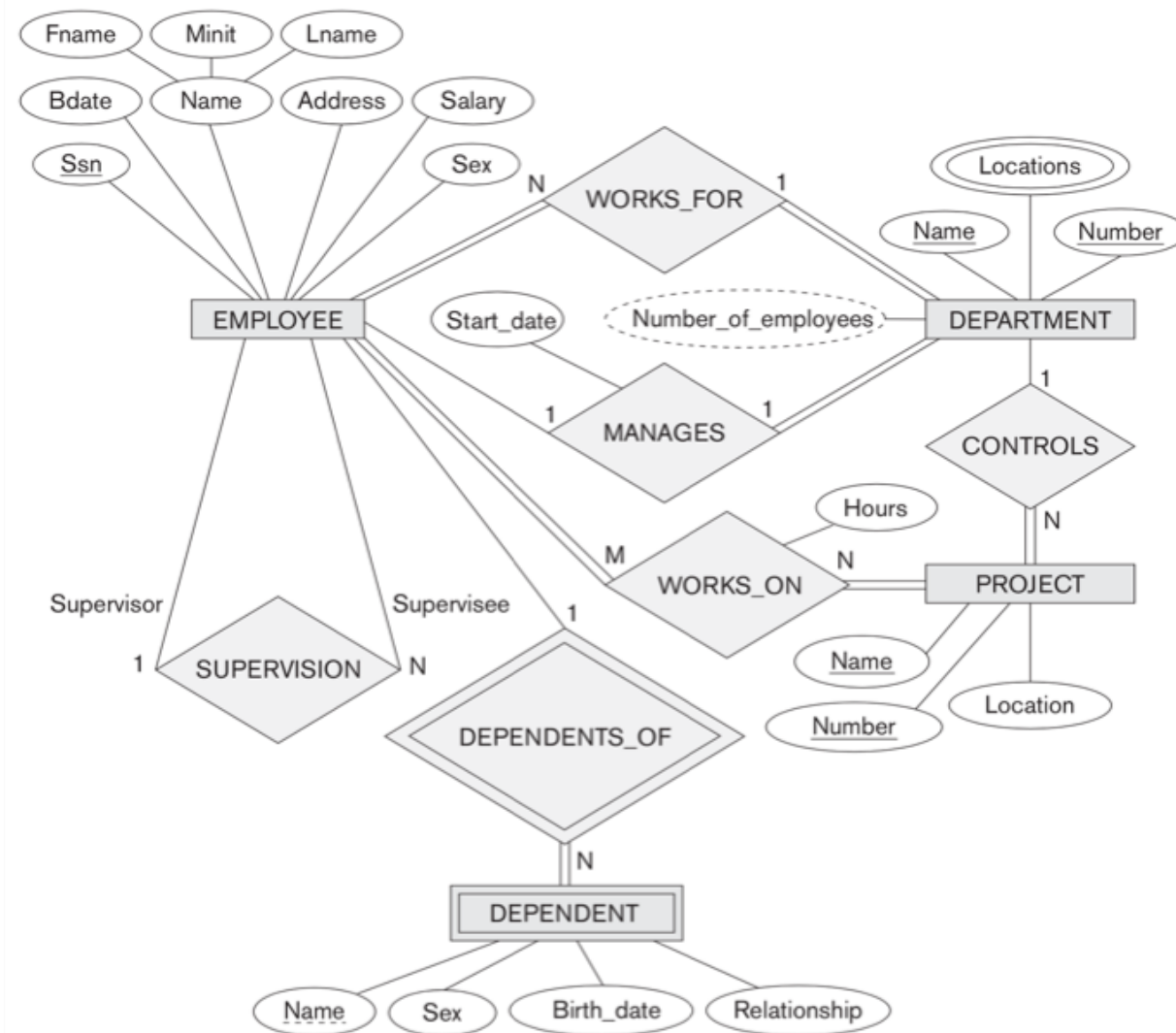
*An employee is assigned to one department, but may work on several projects, which are not necessarily controlled by the same department. We keep track of the current number of hours per week that an employee works on each project.*
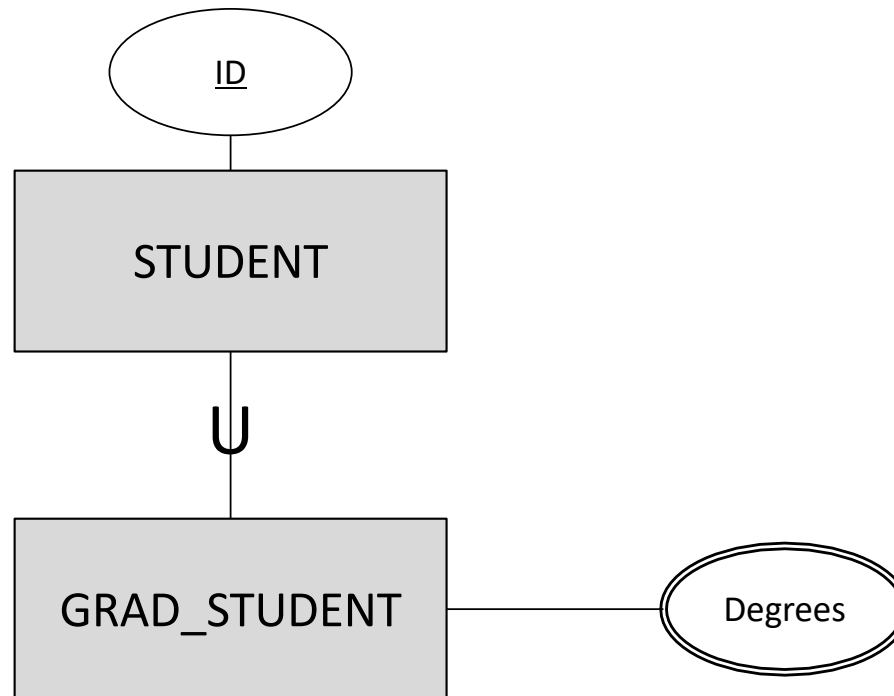
# Answer

# 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

A person can be an employee, an alumnus, and/or a student



SSN

PERSON

o

∪

EMPLOYEE

ALUMNUS

STUDENT

# Multiple Subtypes: Disjointedness

(o)verlap: may be more than one
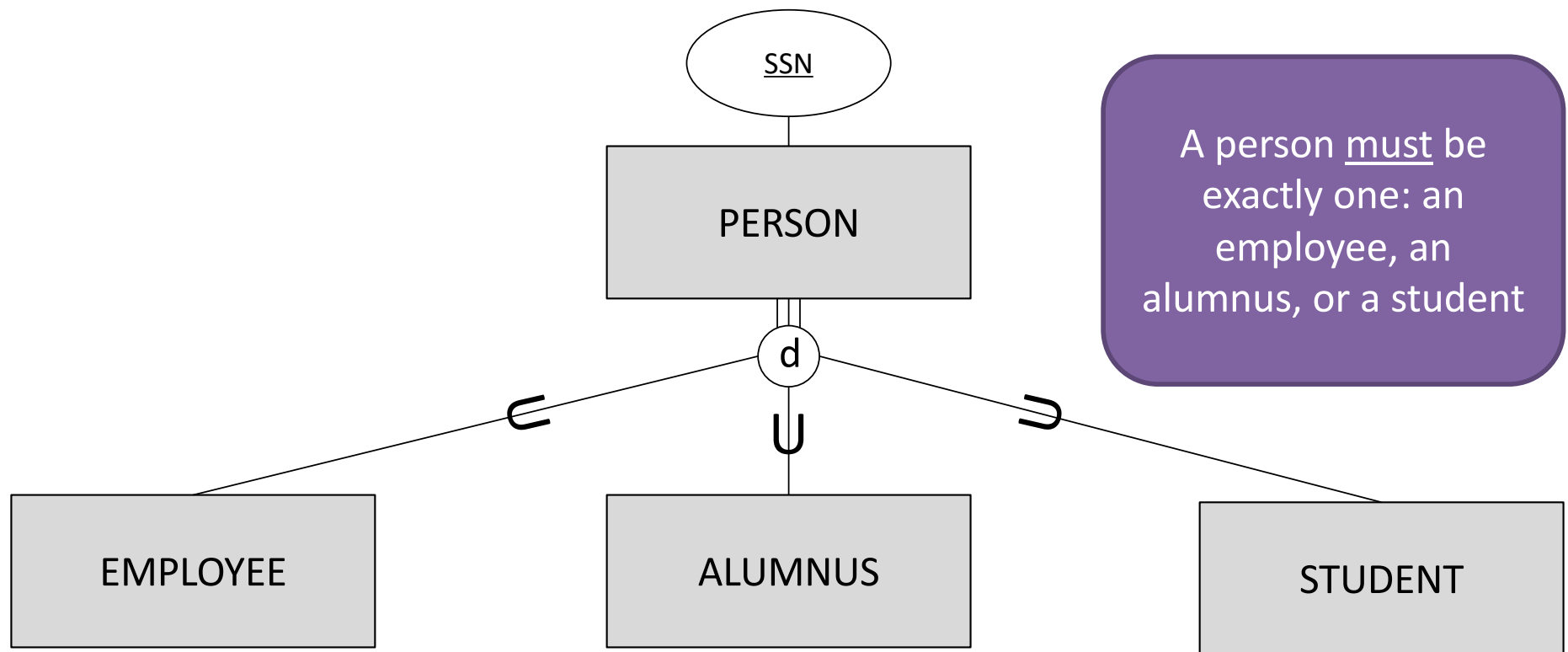
(d)isjoint: entities may *only be one* subtype



A person can be either an employee, an alumnus, or a student
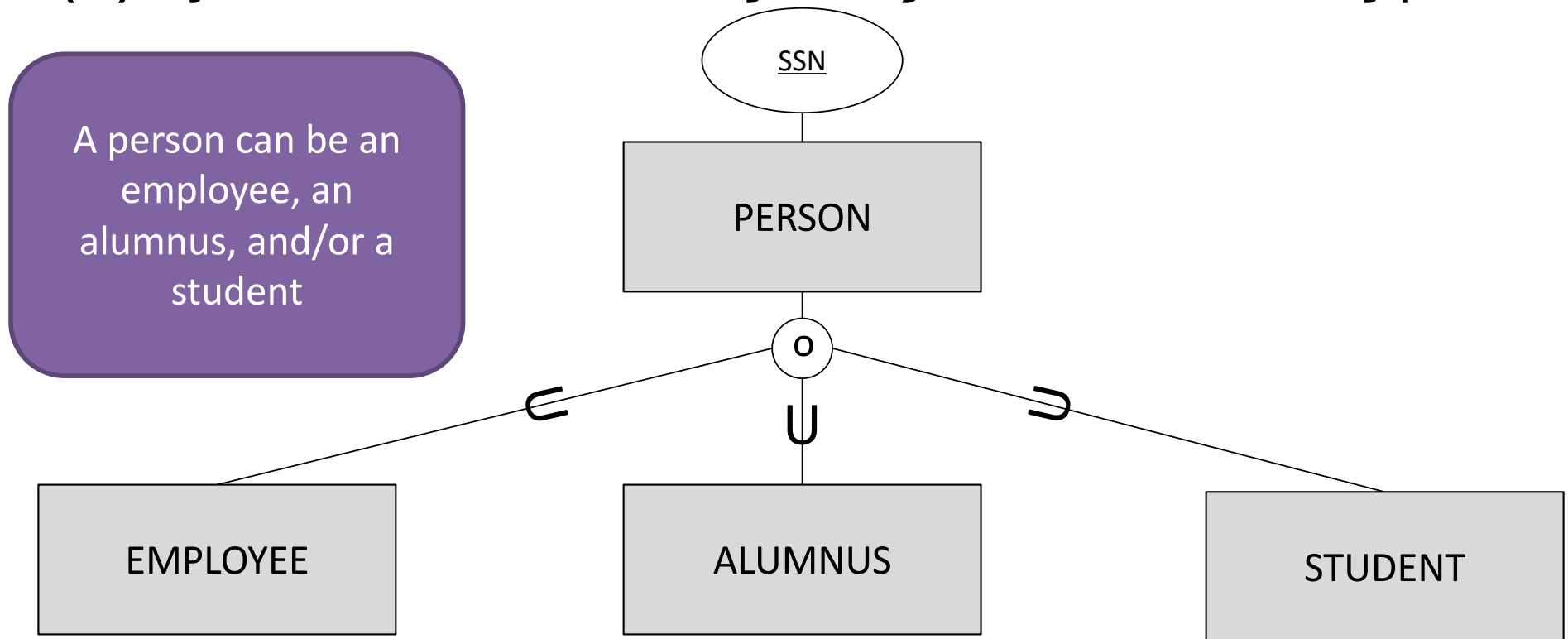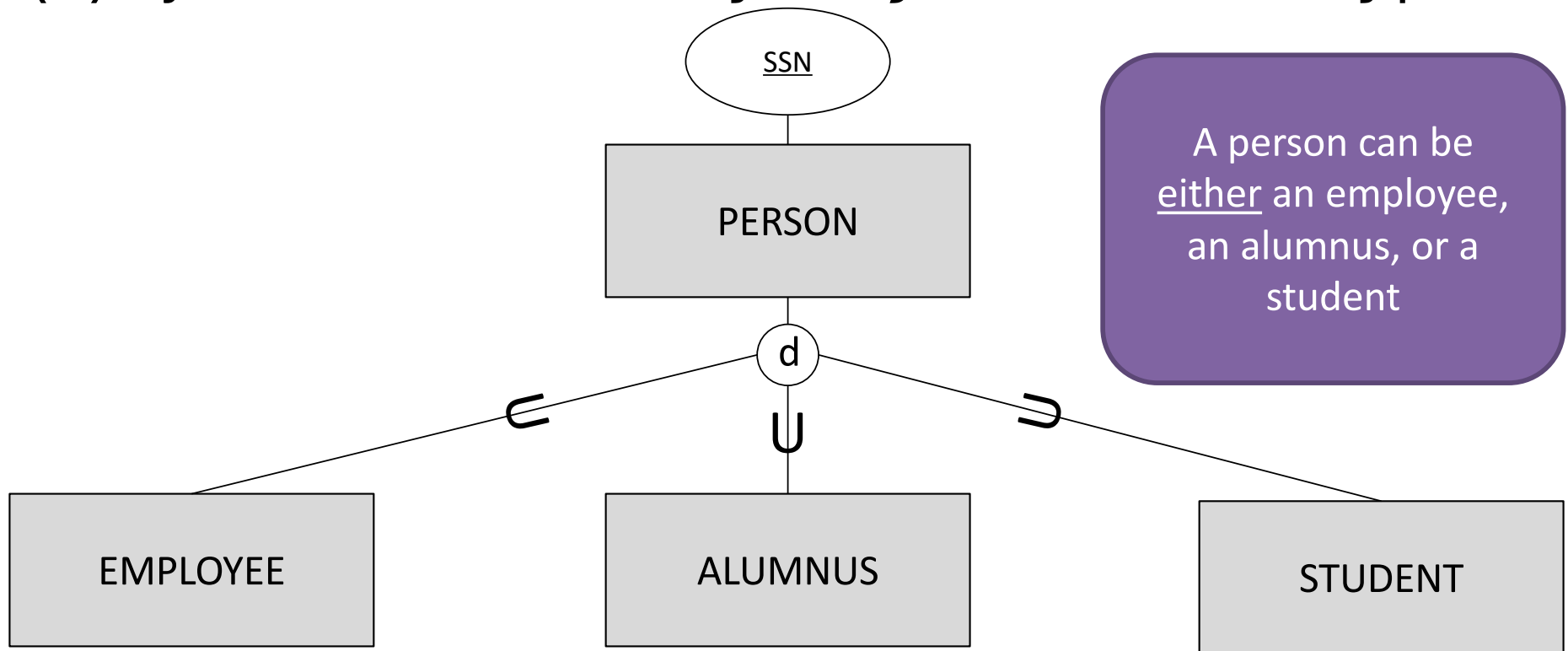
# Multiple Subtypes: Completeness

Similar to relationships; can be total (<u>must</u> belong to subtypes) or partial (<u>can</u> belong)



SSN

PERSON

d

EMPLOYEE     ALUMNUS     STUDENT

A person <u>must</u> be exactly one: an employee, an alumnus, or a student

# 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.
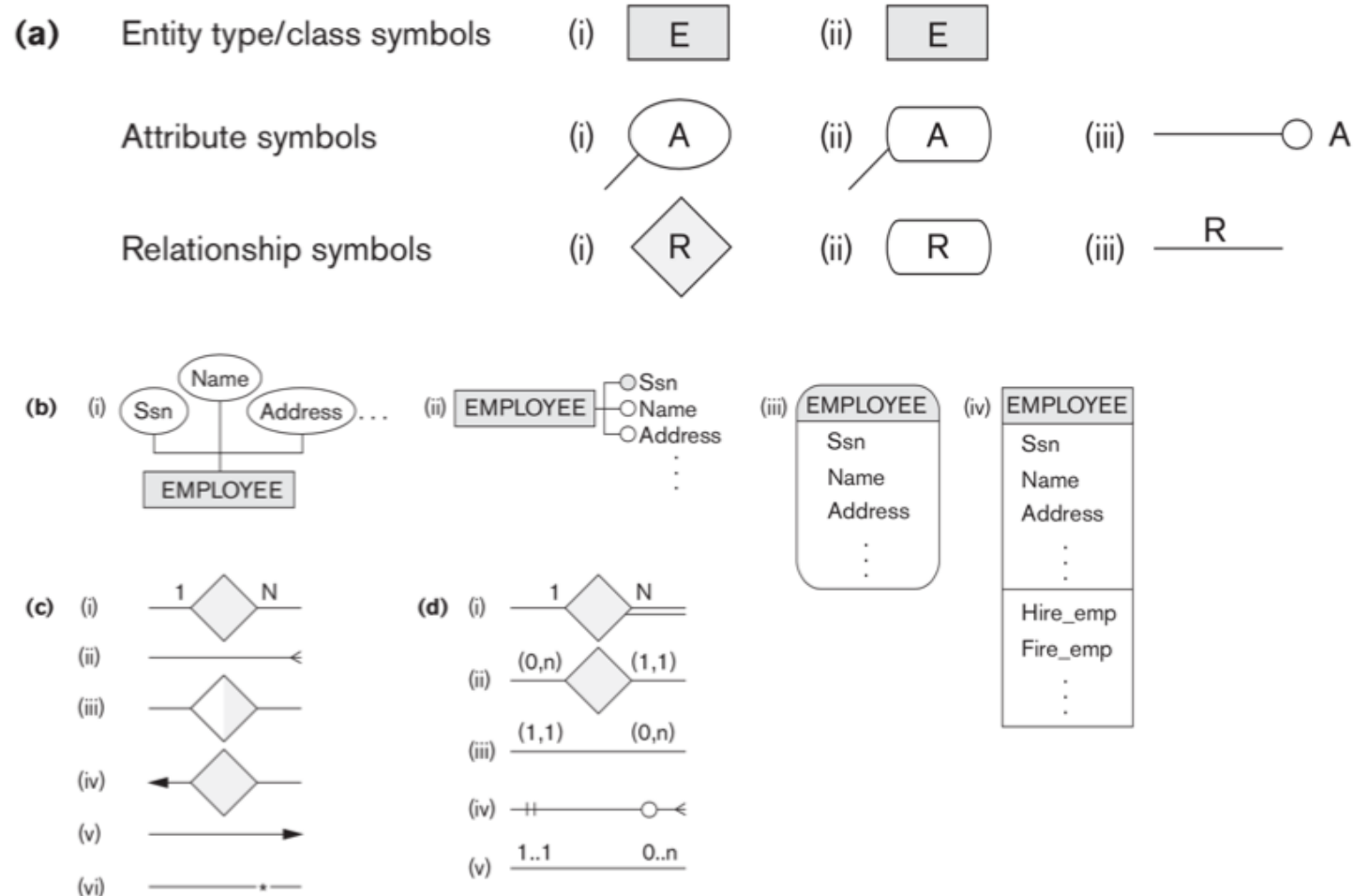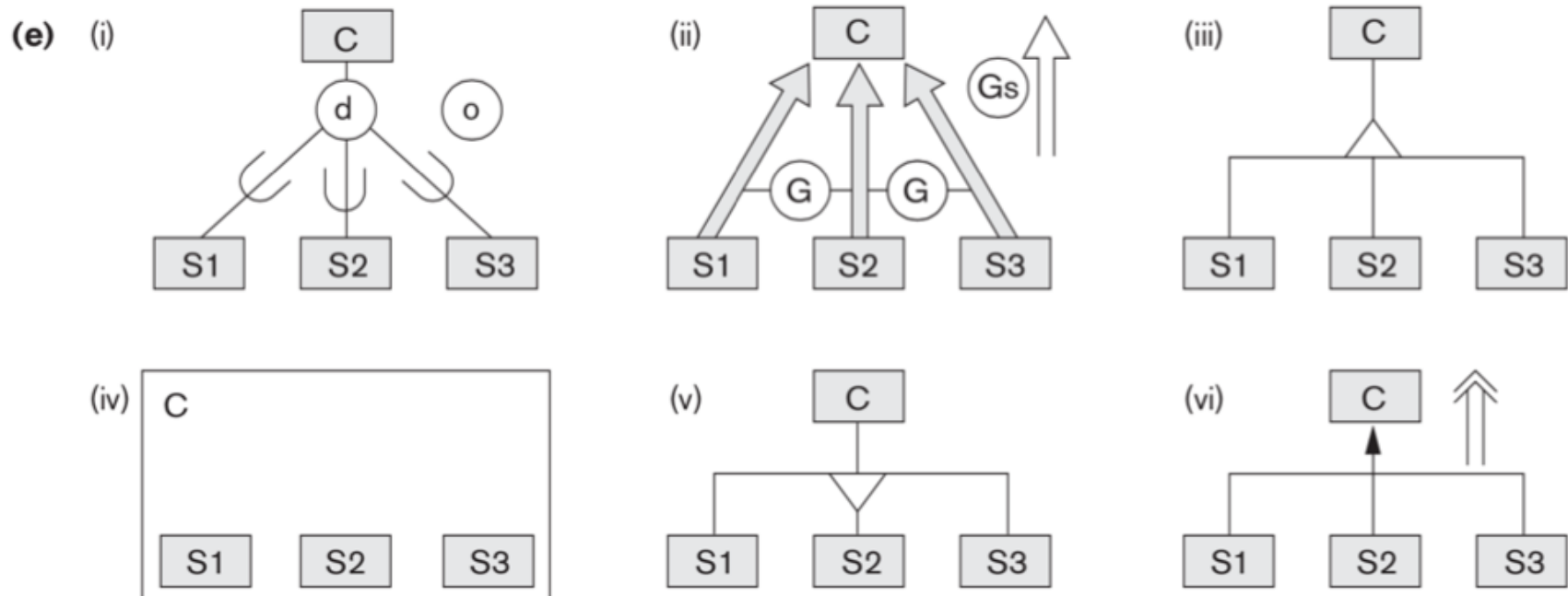
**Entity-Relationship (ER) Diagrams**

# Answer
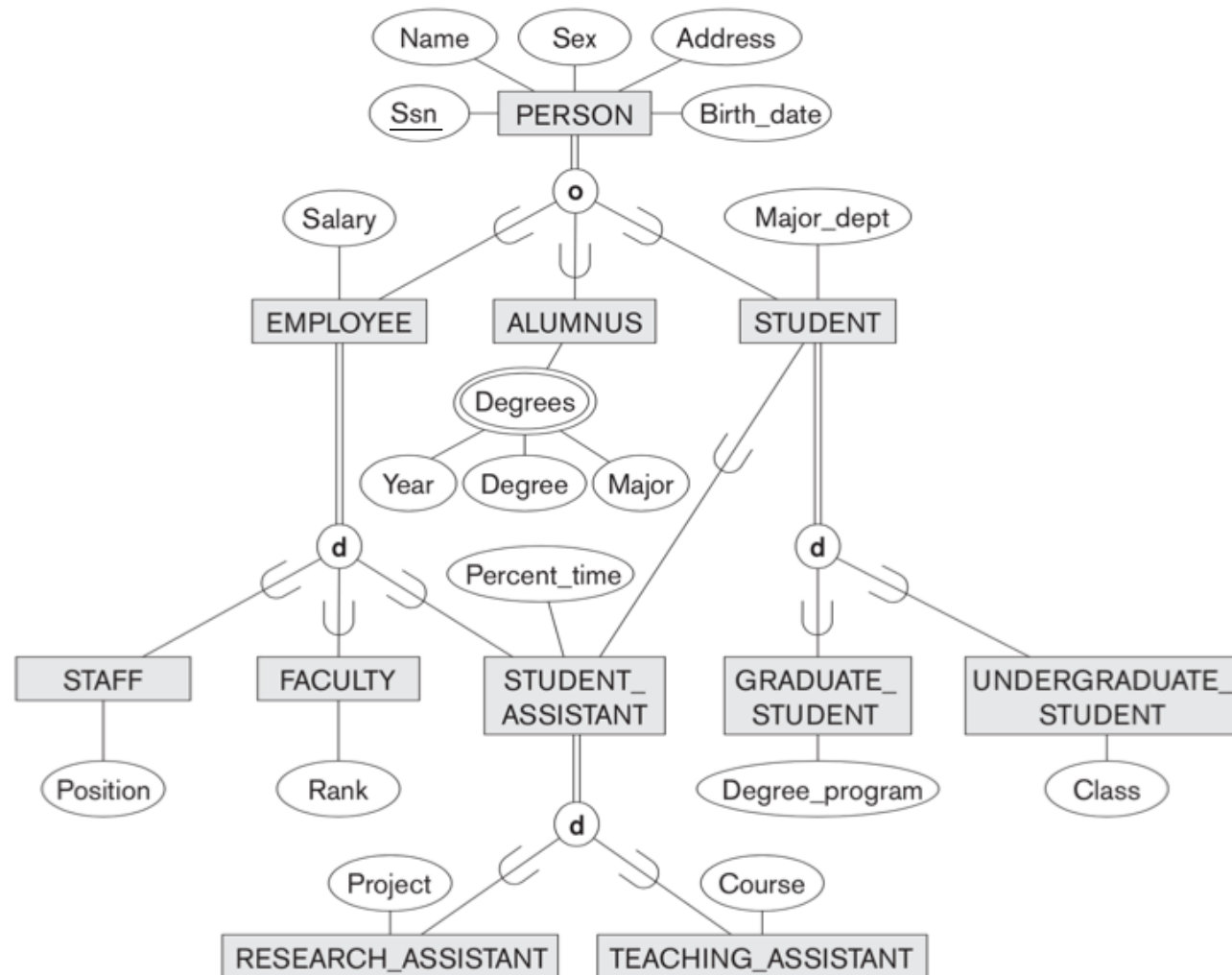
# Alternative Notation (1)
## *Figure A. 1*

# 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*)?

**Entity-Relationship (ER) Diagrams**

# 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
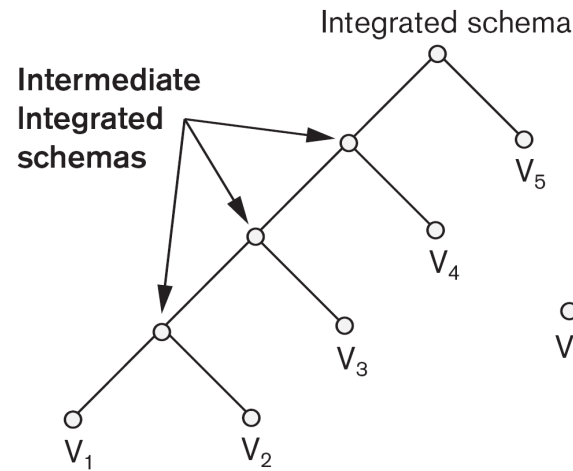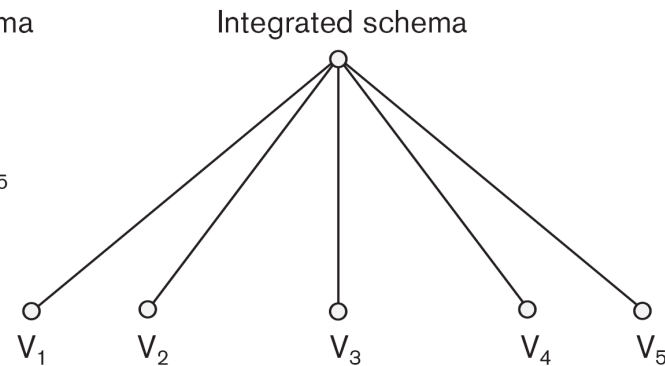
# View Integration (1)

1. Identify correspondences and conflicts

   – Conflicts: names, types, domain, constraints
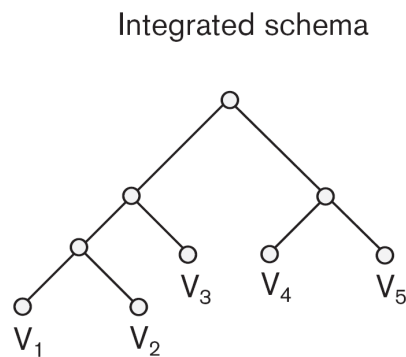
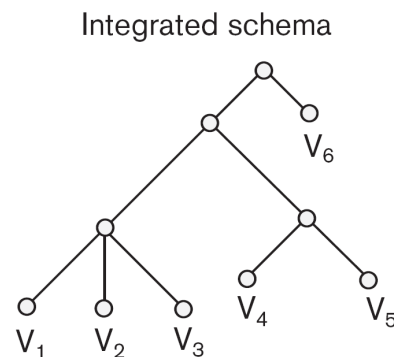2. Modify views to conform

3. Merge

4. Restructure

**Entity-Relationship (ER) Diagrams**

# View Integration (2)



Intermediate Integrated schemas

$V_5$

$V_4$

$V_3$

$V_1$    $V_2$

**Binary ladder integration**

Integrated schema

$V_1$    $V_2$    $V_3$    $V_4$    $V_5$

**N-ary integration**

Integrated schema

$V_3$    $V_4$    $V_5$

$V_1$    $V_2$

**Binary balanced integration**

Integrated schema

$V_6$

$V_4$    $V_5$

$V_1$  $V_2$  $V_3$

**Mixed integration**

**Entity-Relationship (ER) Diagrams**

# Summary

- The goal of conceptual design is to develop a set of *data requirements* that are comprehensive, clear & easy to understand, and algorithmically transformable

- ER Diagrams (ERDs) are one such design model that visually represent the **entities**, **attributes**, and **relationships** of a system

- Requirements elicitation and conceptual design is an iterative process that is a necessary prerequisite to implementing a database

**Entity-Relationship (ER) Diagrams**