Northeastern University
London

Conceptual Database Design

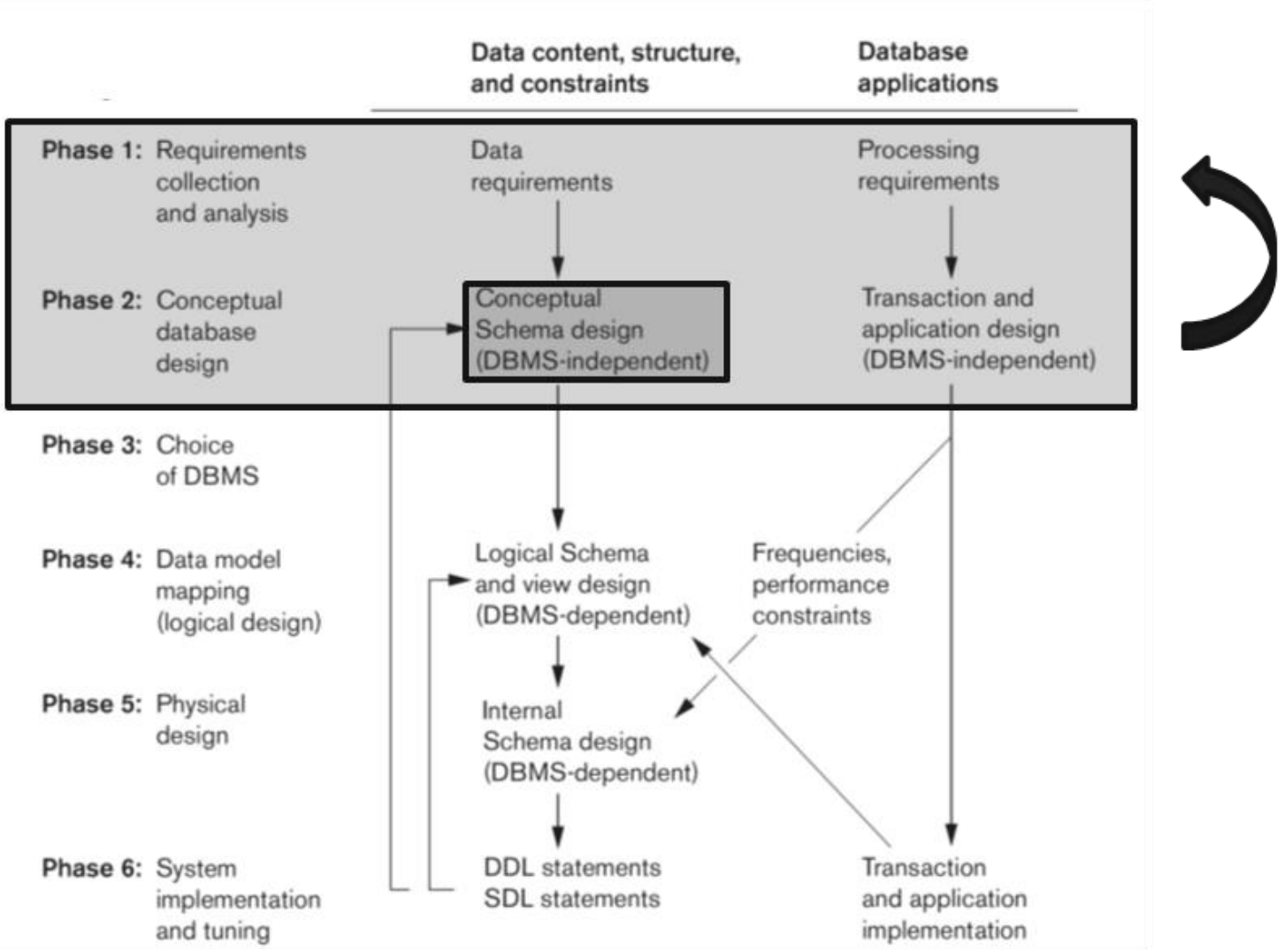
Course Overview

Week No.	Database Design
1	Introduction to Database Design
2	Conceptual Database Design
3	Logical Design
4	Normalization
5	SQL& Relational algebra part 1
6	SQL& Relational algebra part 2
7	*MID TERM*
8	SQL and Web Programming
9	SQL Query Optimizations ***Set Exercises (70%)*** **AE1 Submission 14/11/2025**
10	Database Security and Administration
11	Database Transactions and Concurrency Part I
12	Autumn Break
13	Database Transactions and Concurrency Part I ***Written Assignment(Group & Individual)*** **AE2 Submission 8/12/2025**

Outline

1. Goals of Conceptual Design
2. Entity-Relationship (ER) Model
3. One ER Diagrammatic Notation
4. Requirements Elicitation (lecture & ***lab***)
5. Approaches to Conceptual Design
6. From Requirements to ER ***(lab)***

Database Design and Implementation Process



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

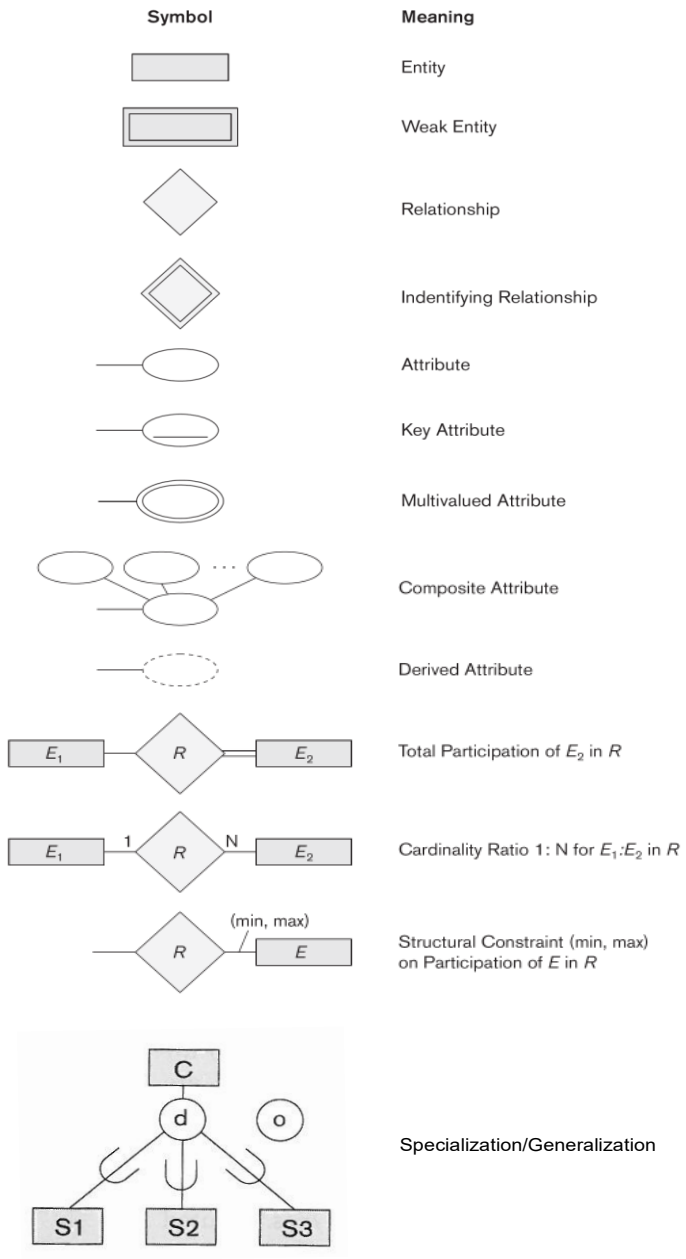
- Property of an entity
- Most of what we store in the database

Relationship

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

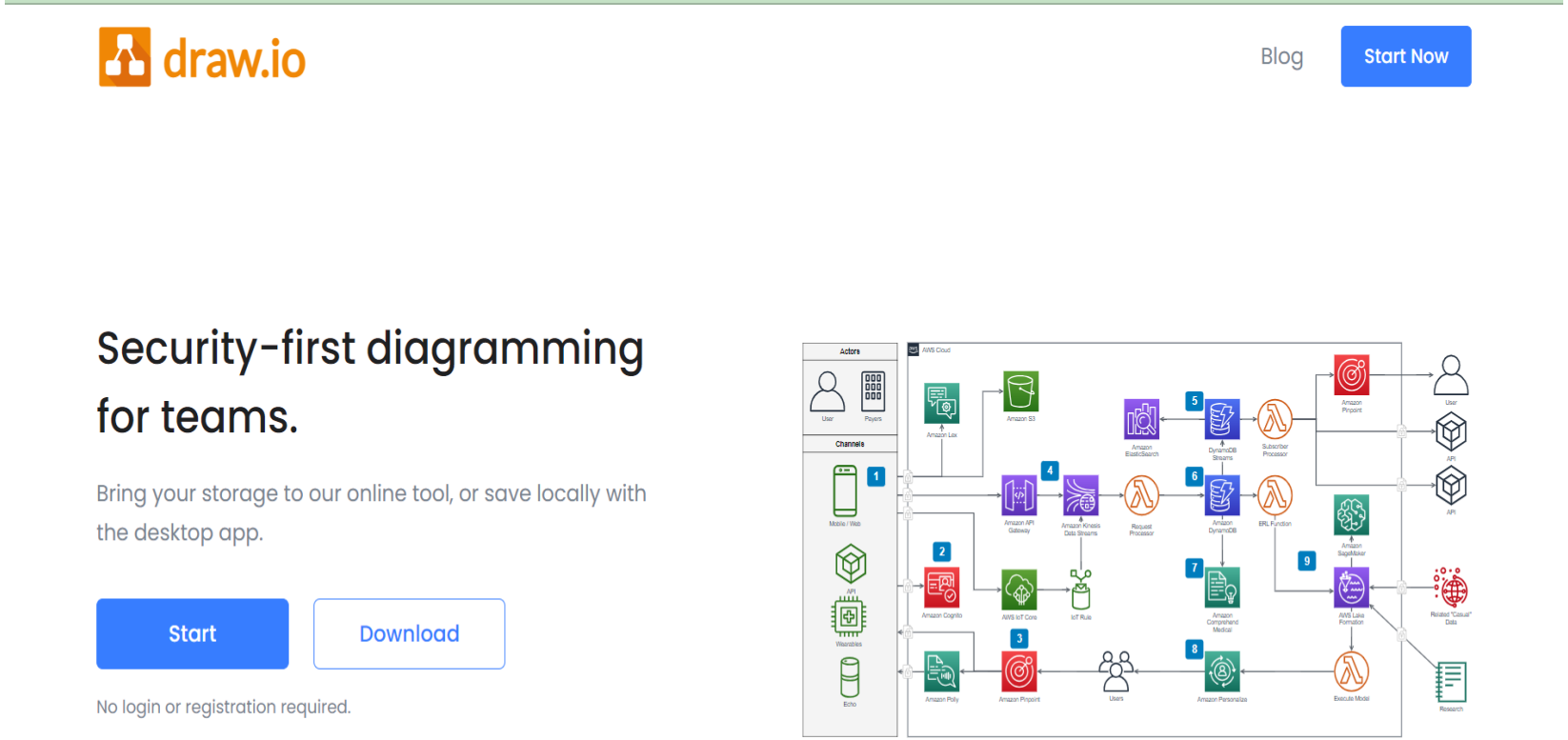
ER Diagrams

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



Tools

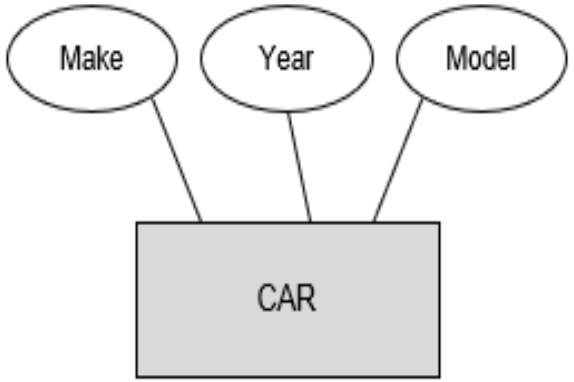
https://app.diagrams.net/
https://www.drawio.com/



Entity Sets

Set of entities that have the same attributes

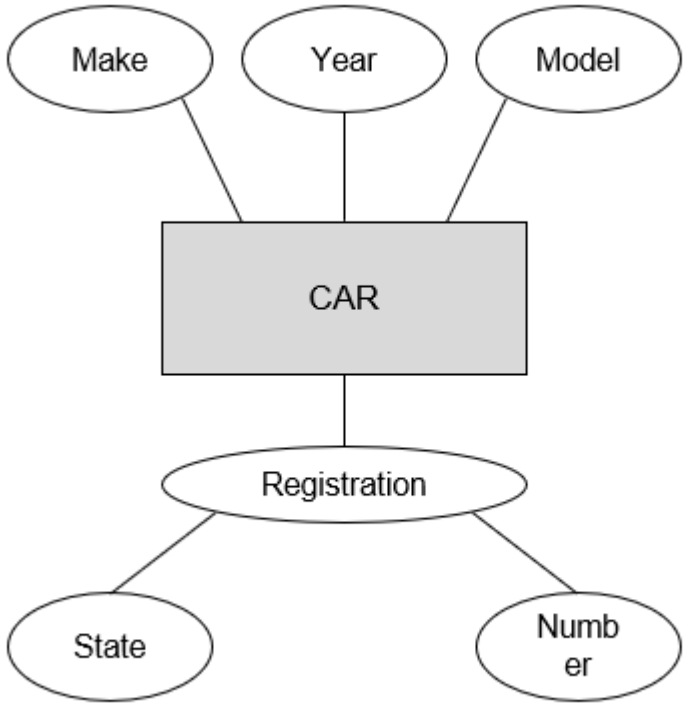
All cars have a year, make, and model.



Composite Attributes

Can be subdivided into smaller subparts

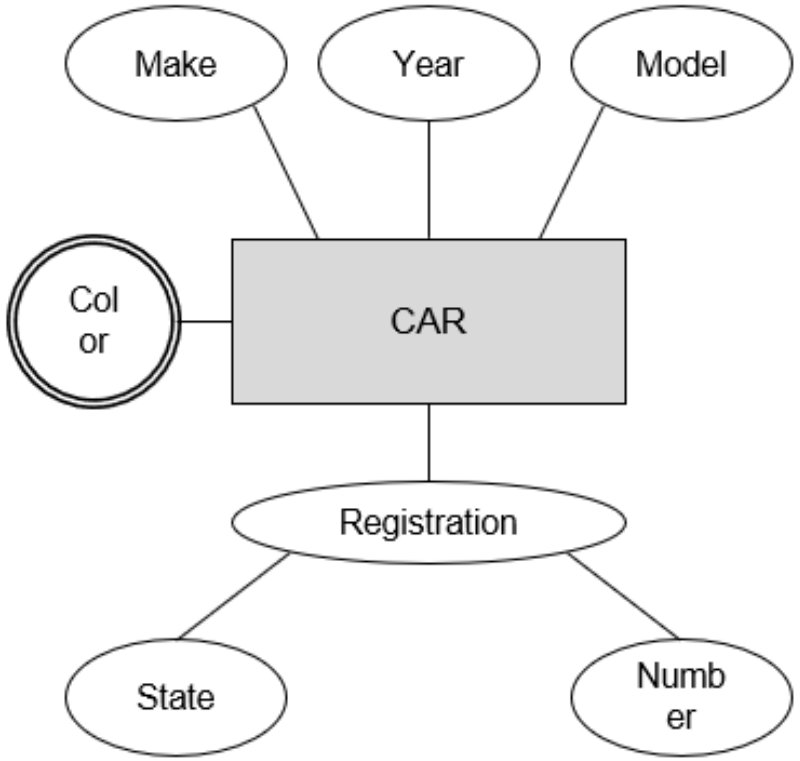
*All cars have a year, make, model, **and registration**.*



Multivalued Attributes

Can take a [possibly specified] number of values.

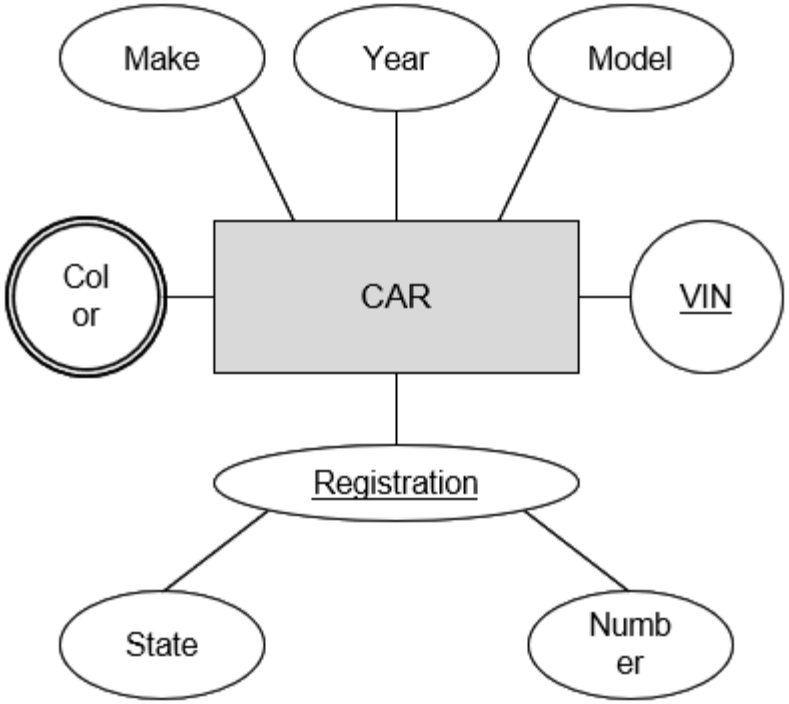
All cars have a year, make, model, registration, and ***some number of colors****.*



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.*



Note

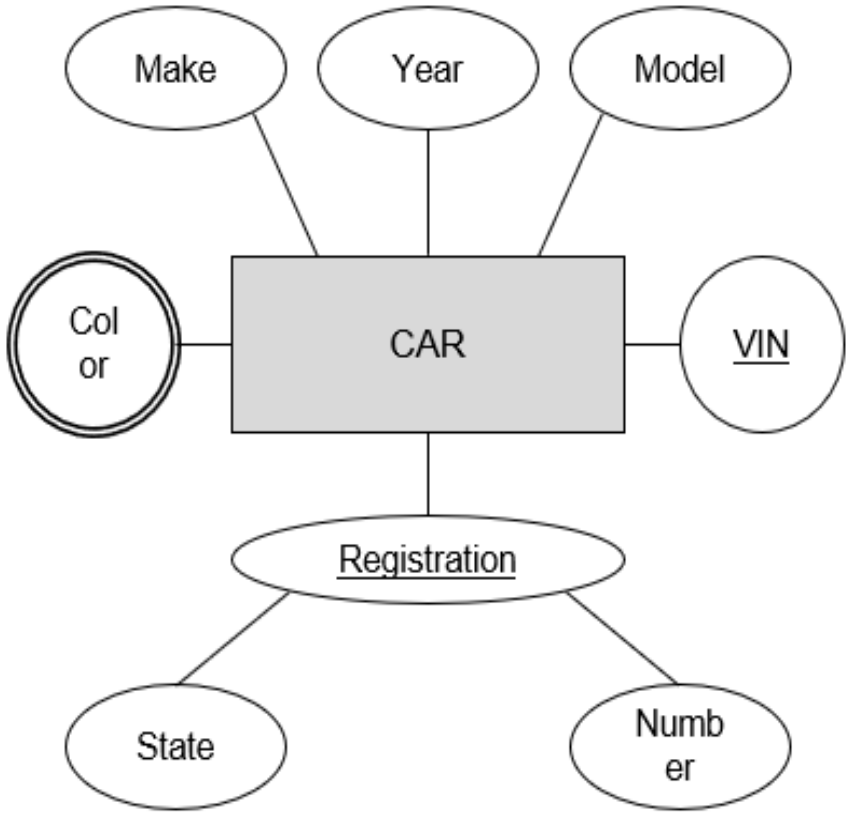
- In relational schema, underlining multiple attributes indicates that for all rows, the *combination* is unique

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

Derived Attributes

The value can be computed

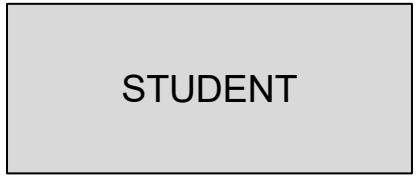
*All cars have a year, **age**, make, model, registration (unique), vehicle number (vin; unique), some number of colors.*



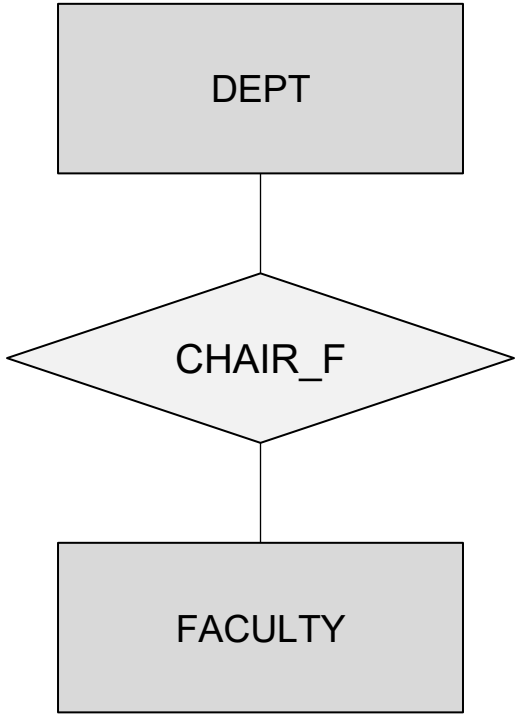
Relationships

Associates one or more sets of entities

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



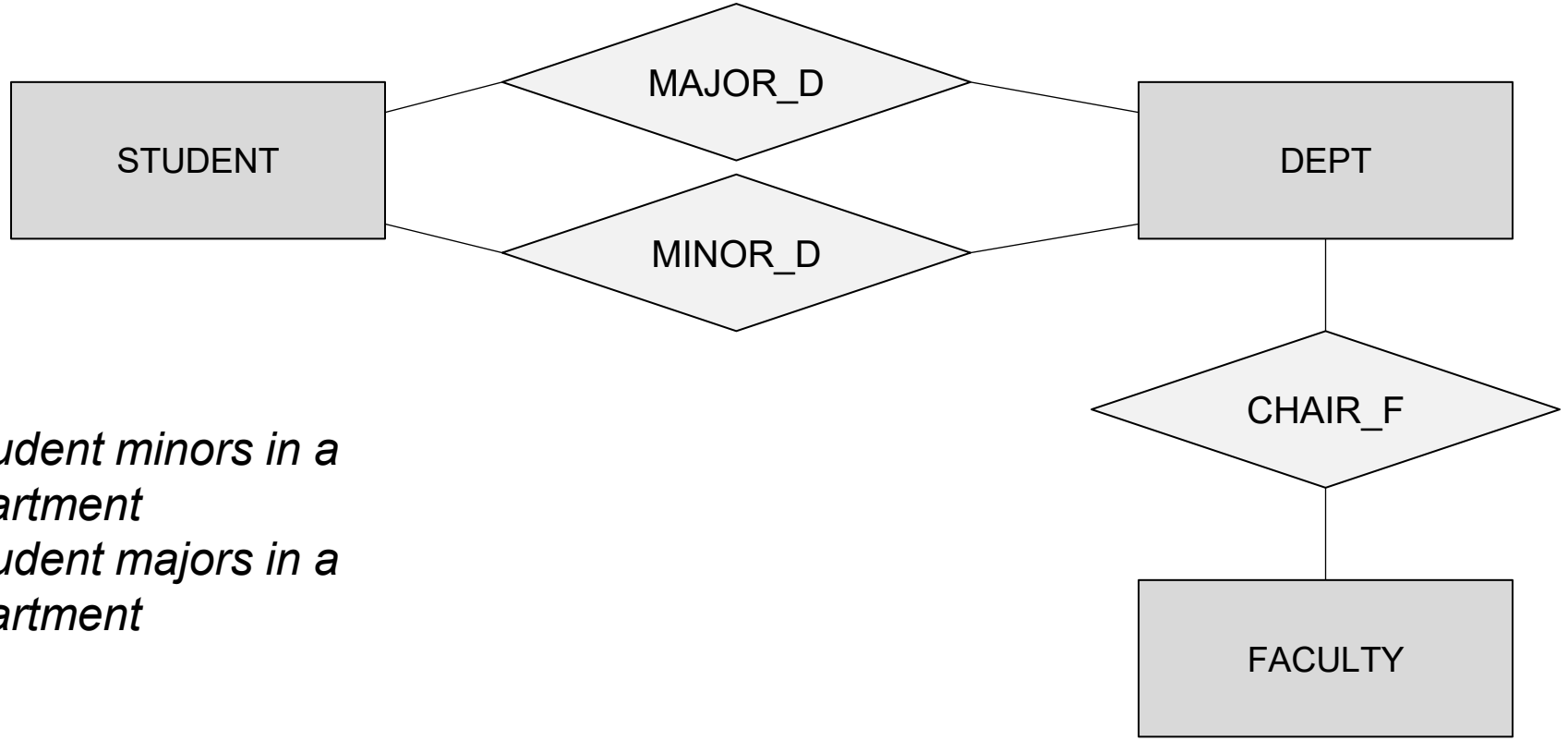
A department is chaired by a faculty member . A faculty member chairs a department.



Relationships

Associates one or more sets of entities

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

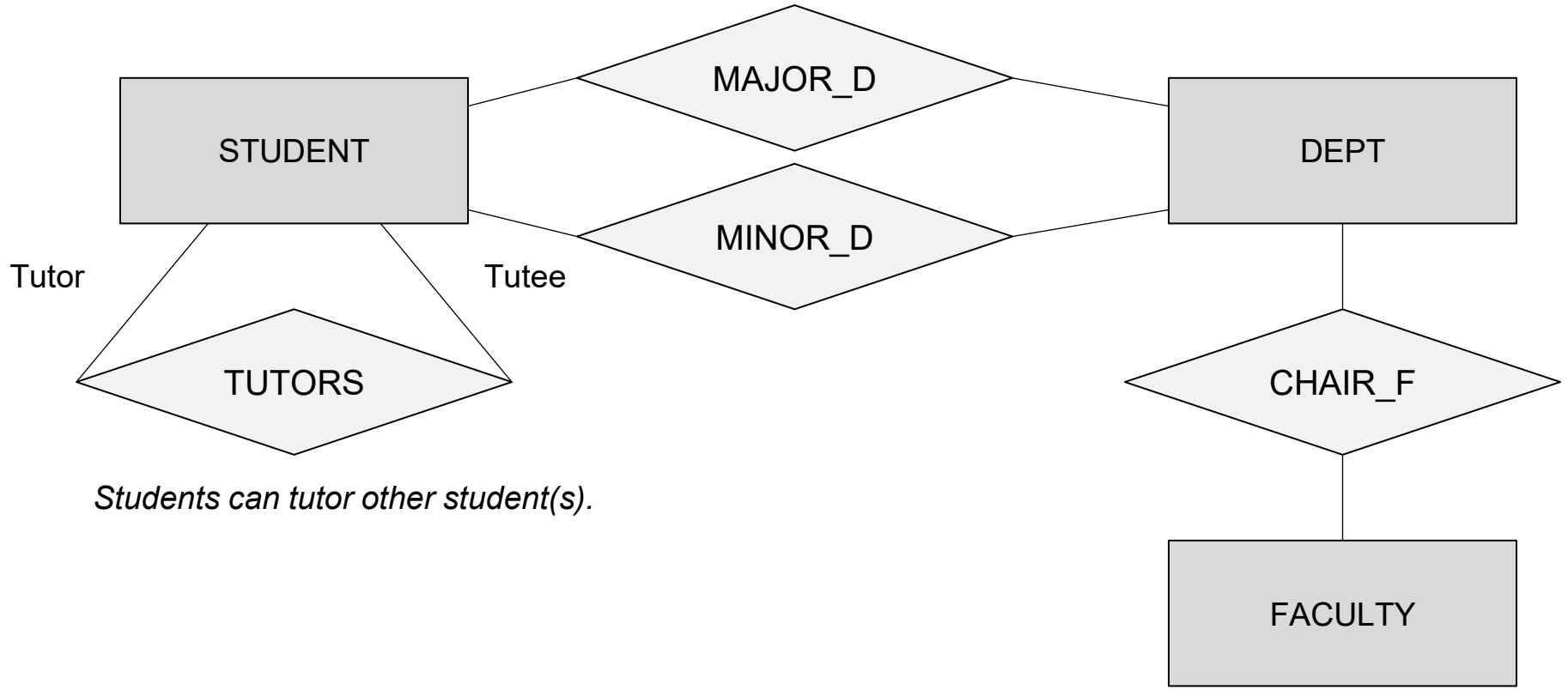


A student minors in a department
A student majors in a department

Relationships

Associates one or more sets of entities

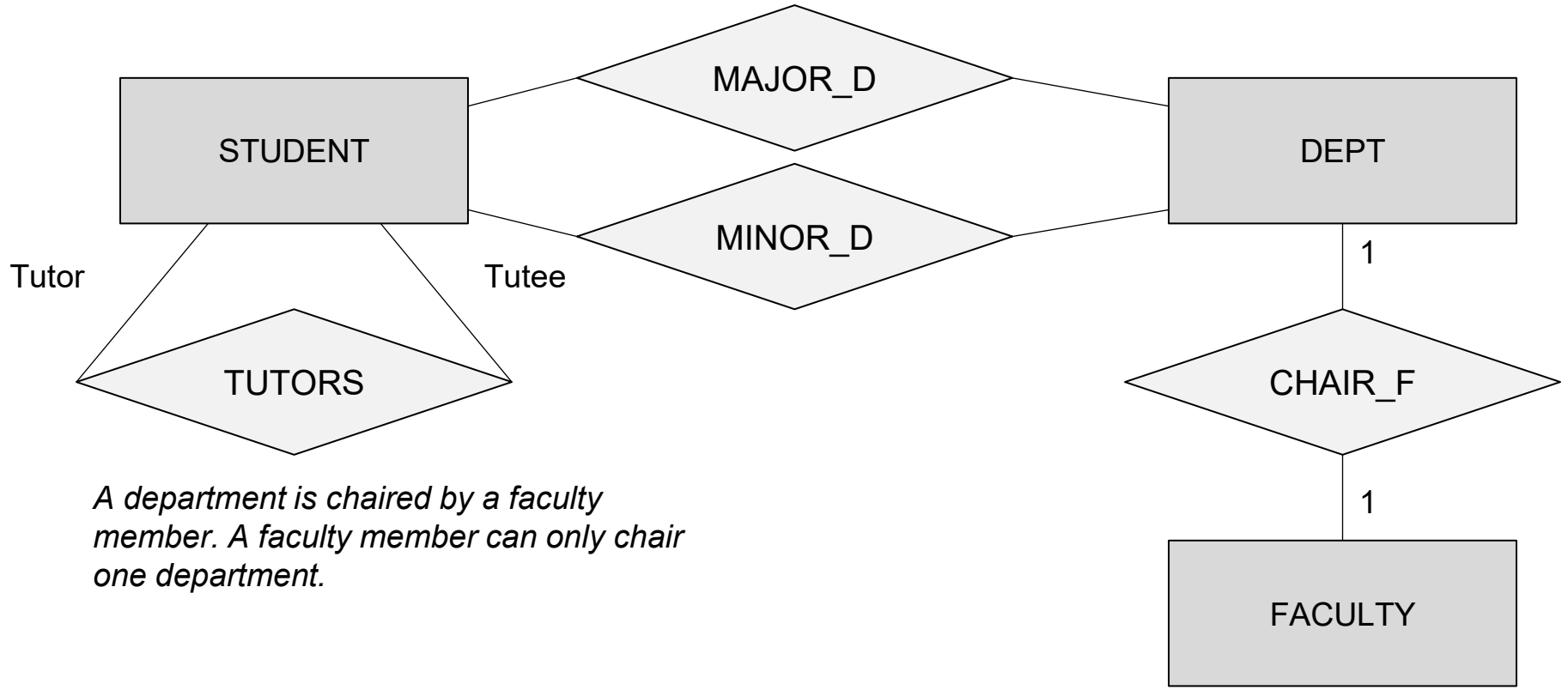
- One = recursive (**role** is important)



Students can tutor other student(s).

Cardinality Ratios

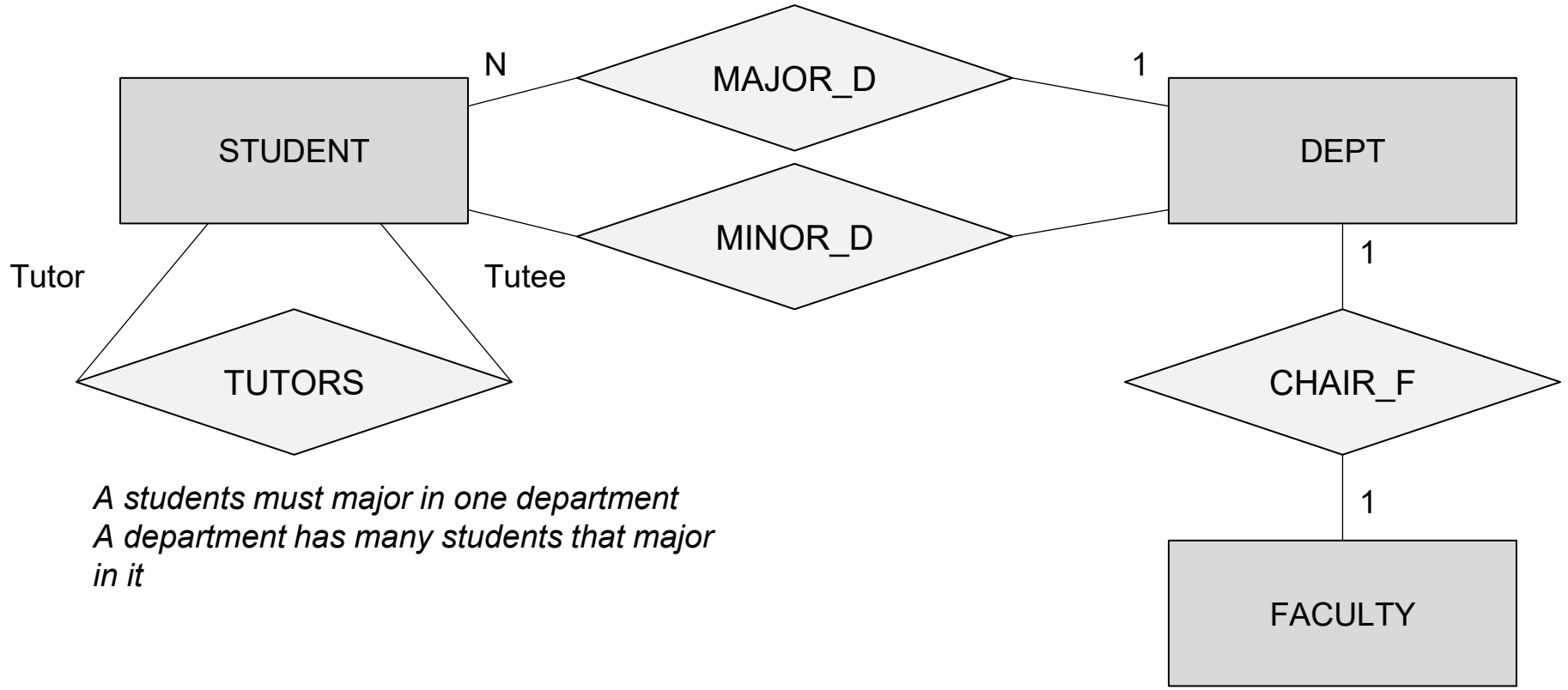
Constrains the number of entities that can participate in each role of the relationship



A department is chaired by a faculty member. A faculty member can only chair one department.

Cardinality Ratios

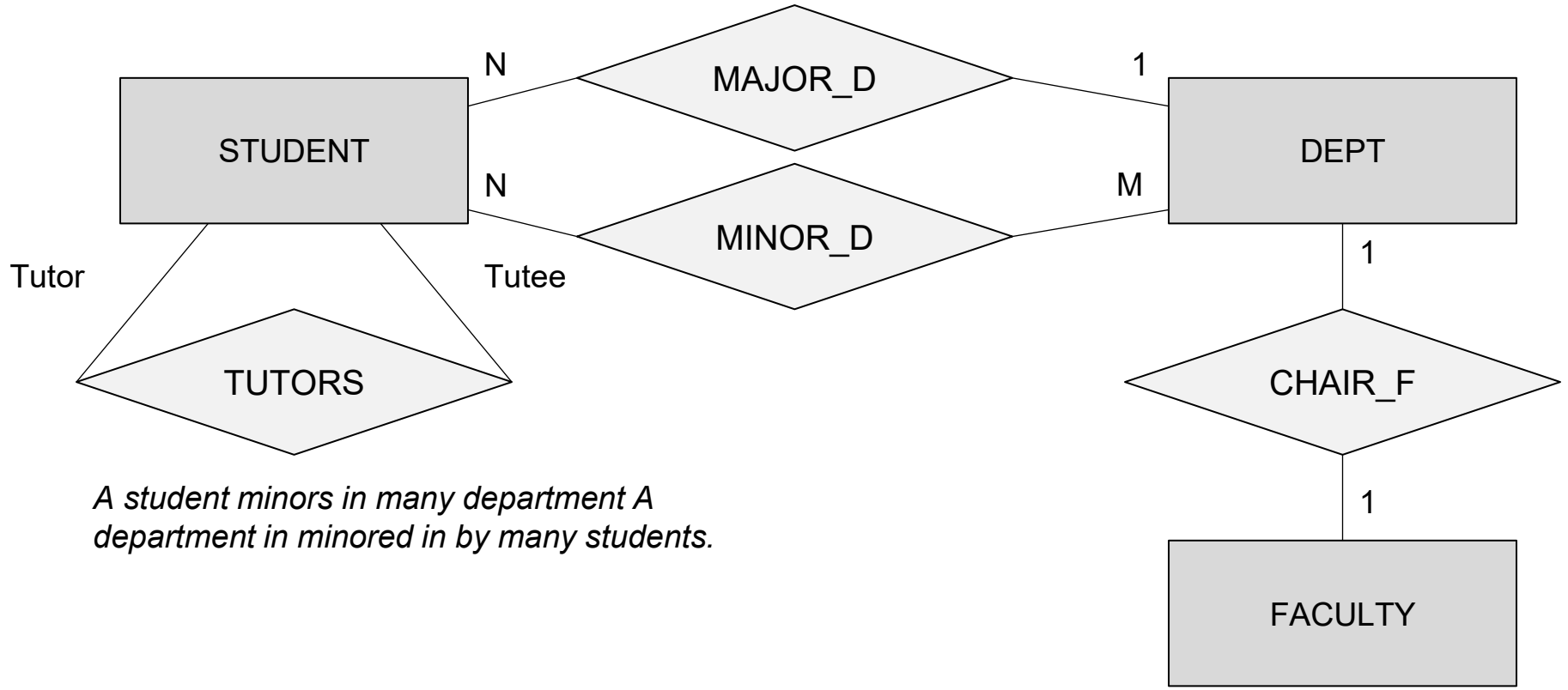
Constrains the number of entities that can participate in each role of the relationship



A students must major in one department
A department has many students that major in it

Cardinality Ratios

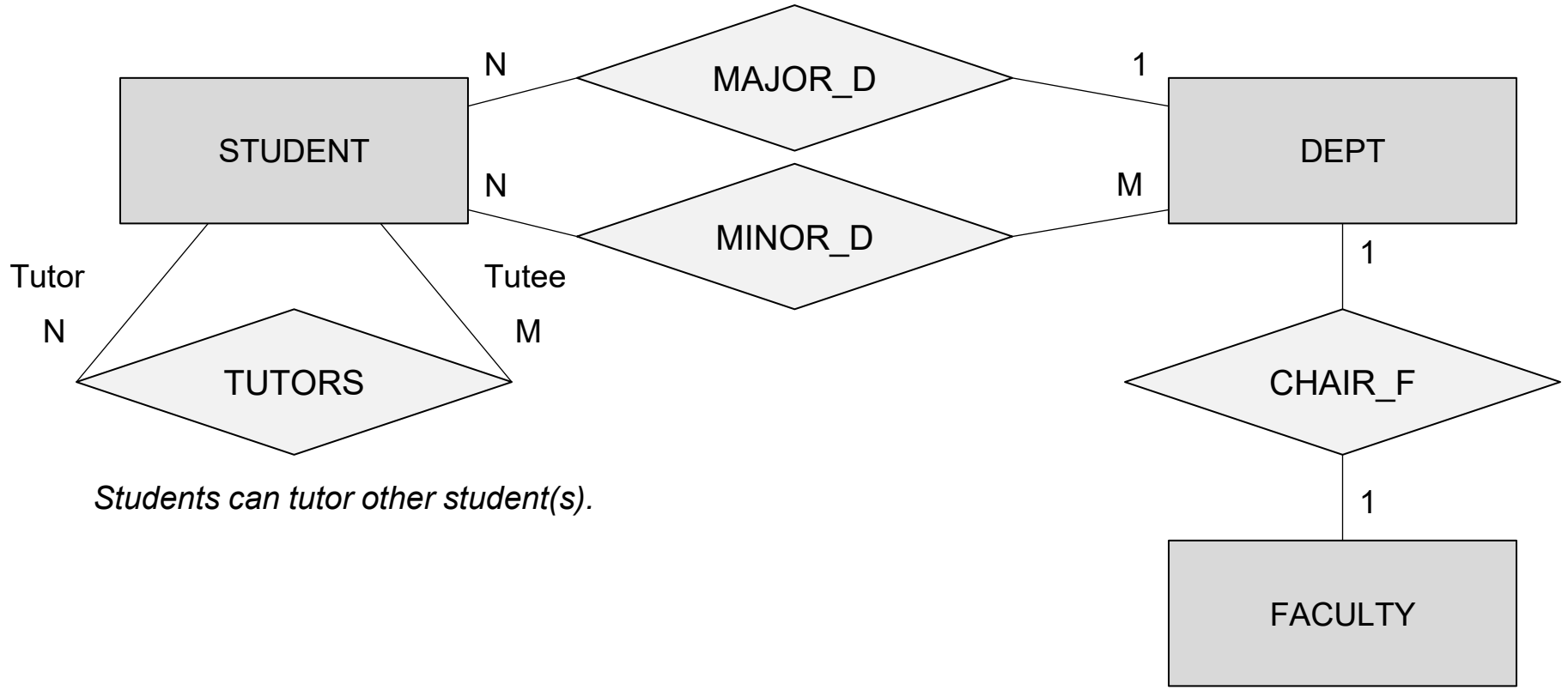
Constrains the number of entities that can participate in each role of the relationship



A student minors in many department A department in minored in by many students.

Cardinality Ratios

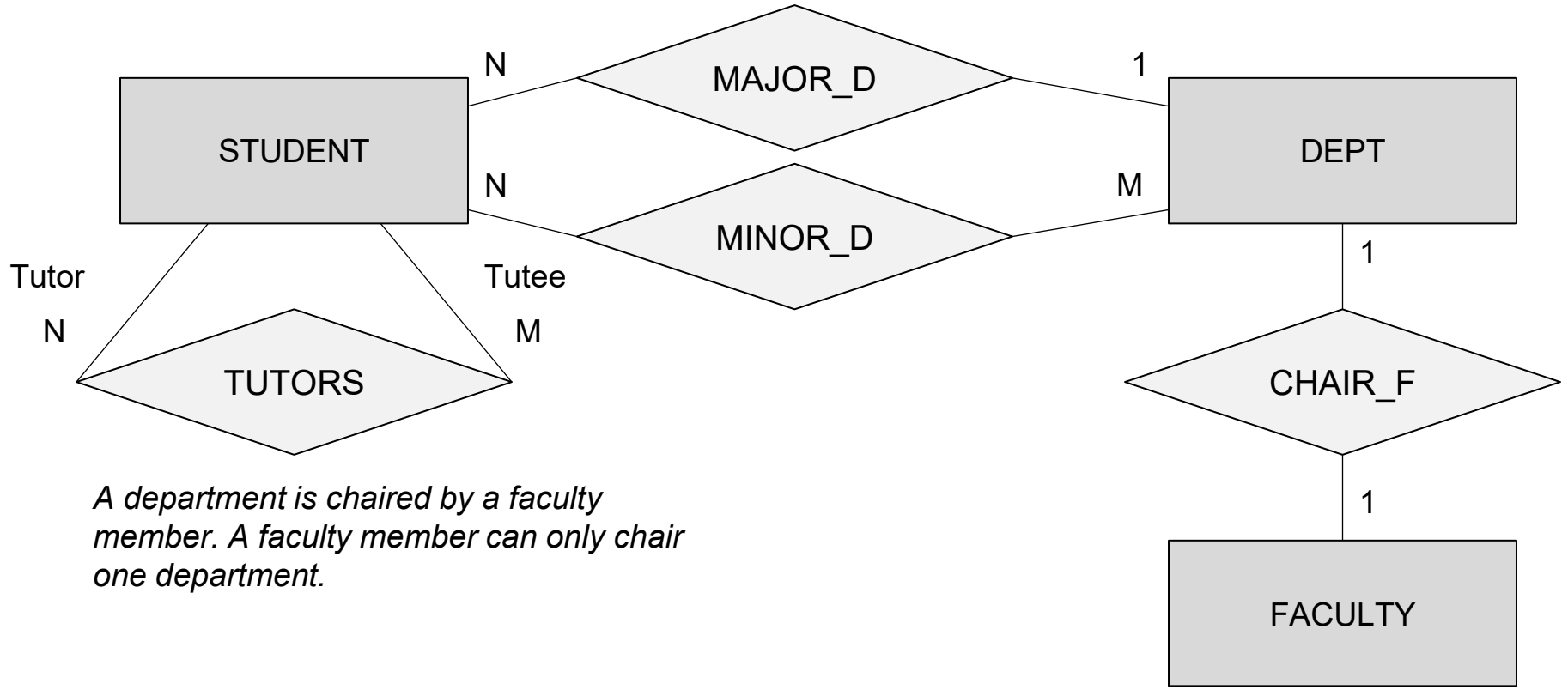
Constrains the number of entities that can participate in each role of the relationship



Students can tutor other student(s).

Structural Constraints

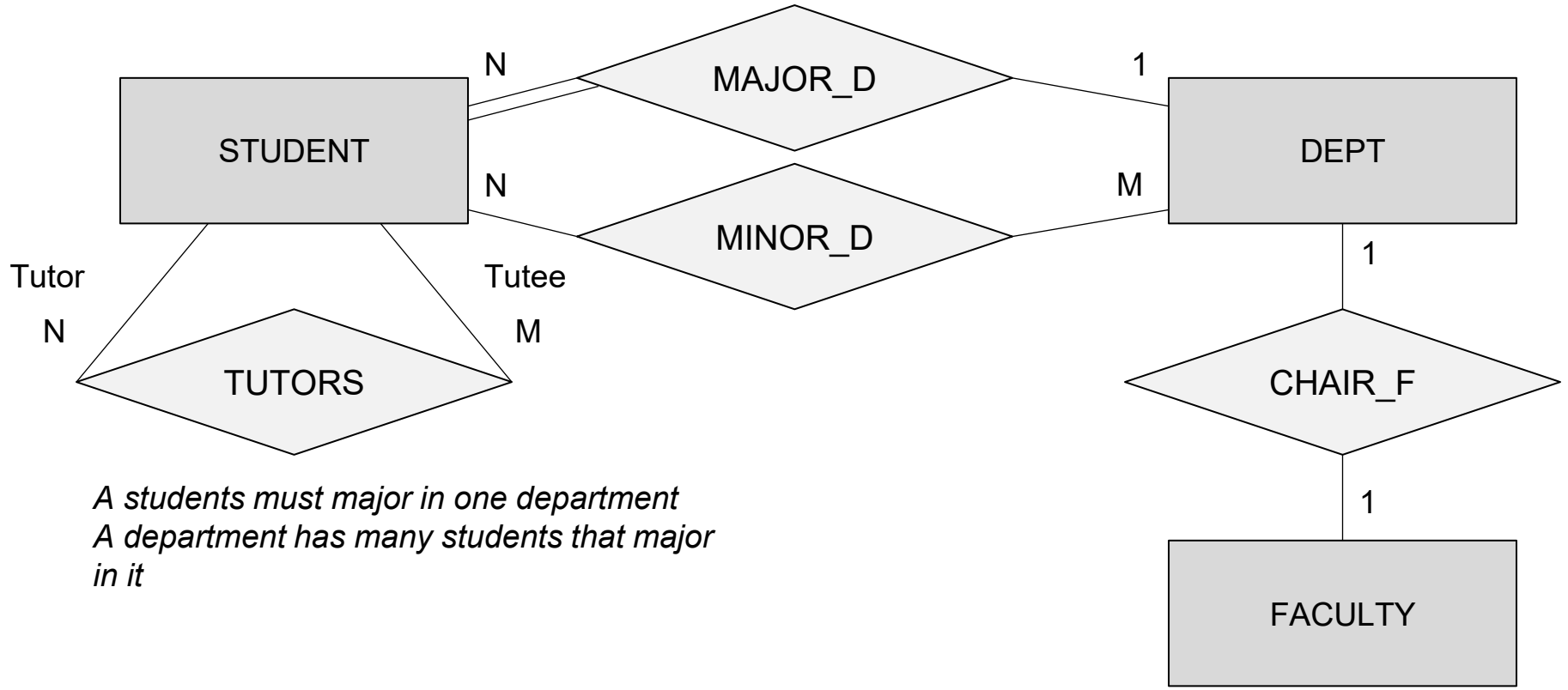
If an entity does not exist unless it appears with an entity in a relationship, the participation is **total** (existence dependency). Else, **partial**.



A department is chaired by a faculty member. A faculty member can only chair one department.

Structural Constraints

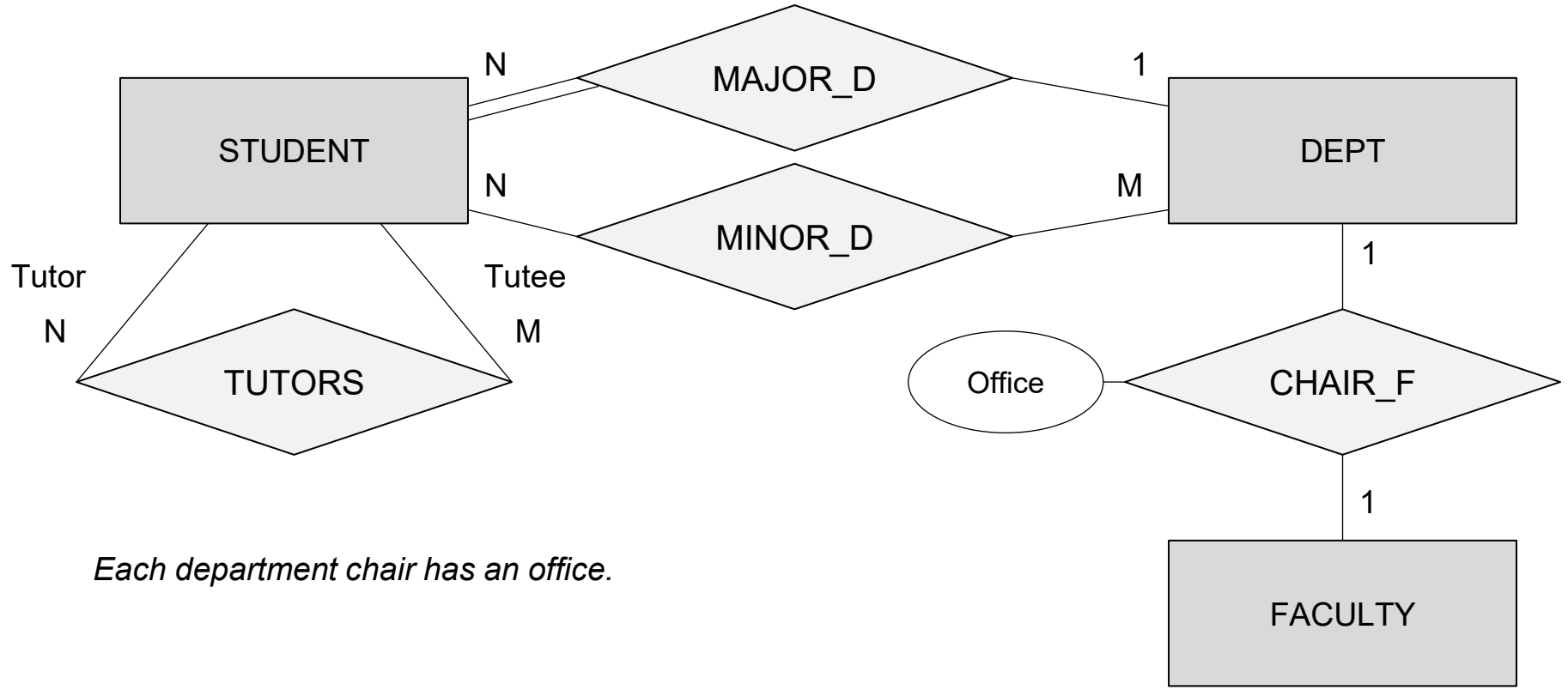
If an entity does not exist unless it appears with an entity in a relationship, the participation is **total** (existence dependency). Else, **partial**.



A students must major in one department
A department has many students that major in it

Attributes of Relationships

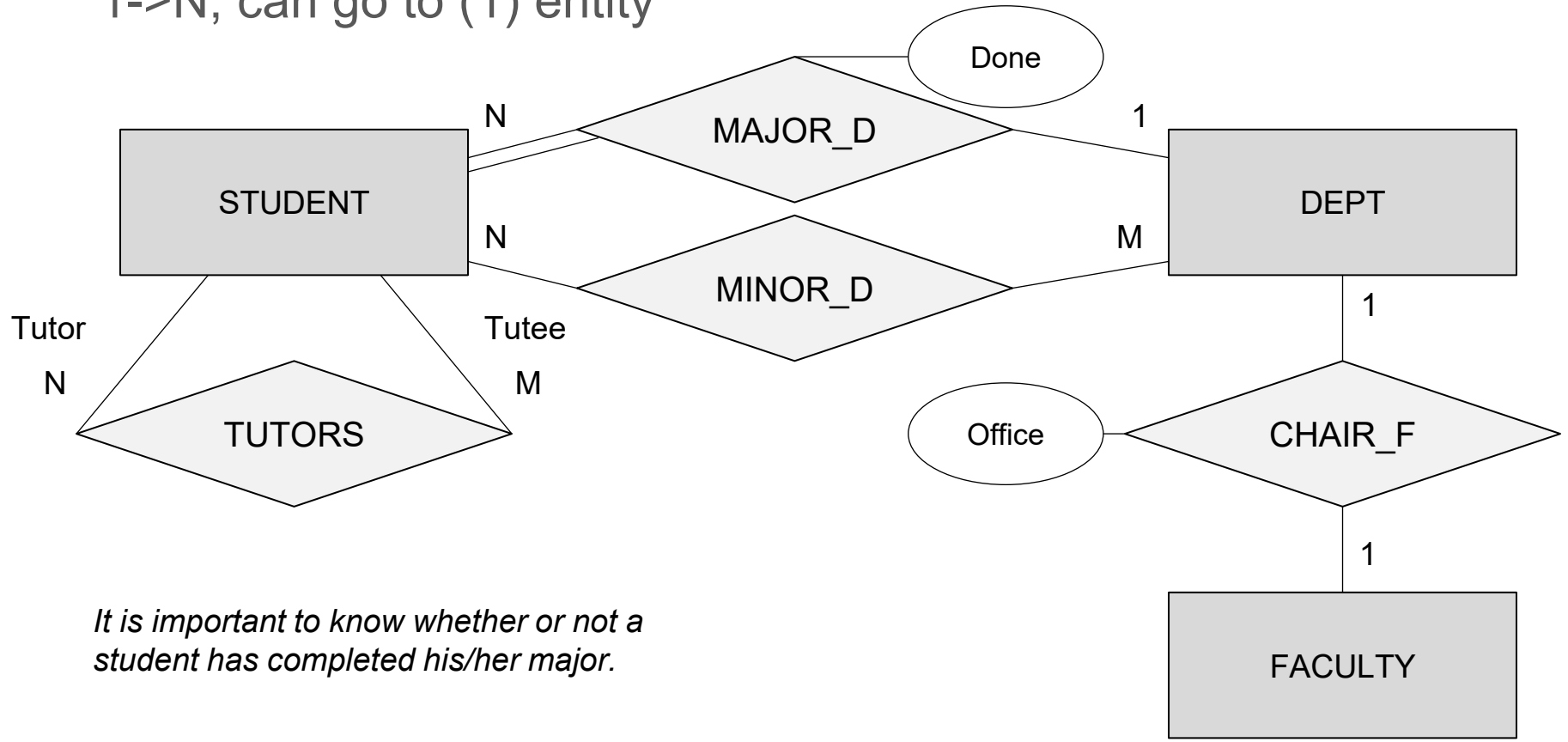
1->1, can go to either entity
1->N, can go to (1) entity



Each department chair has an office.

Attributes of Relationships

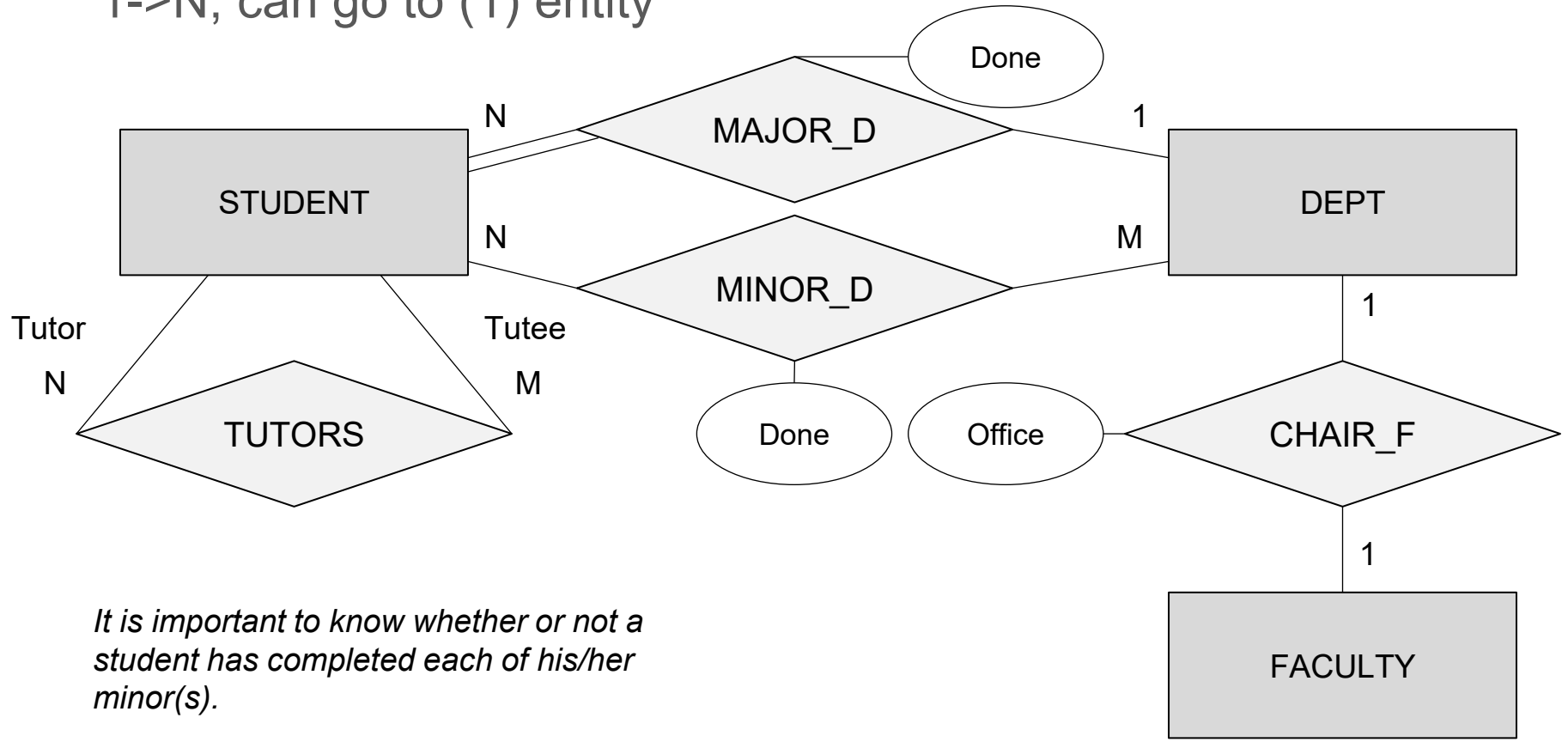
1->1, can go to either entity
1->N, can go to (1) entity



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

Attributes of Relationships

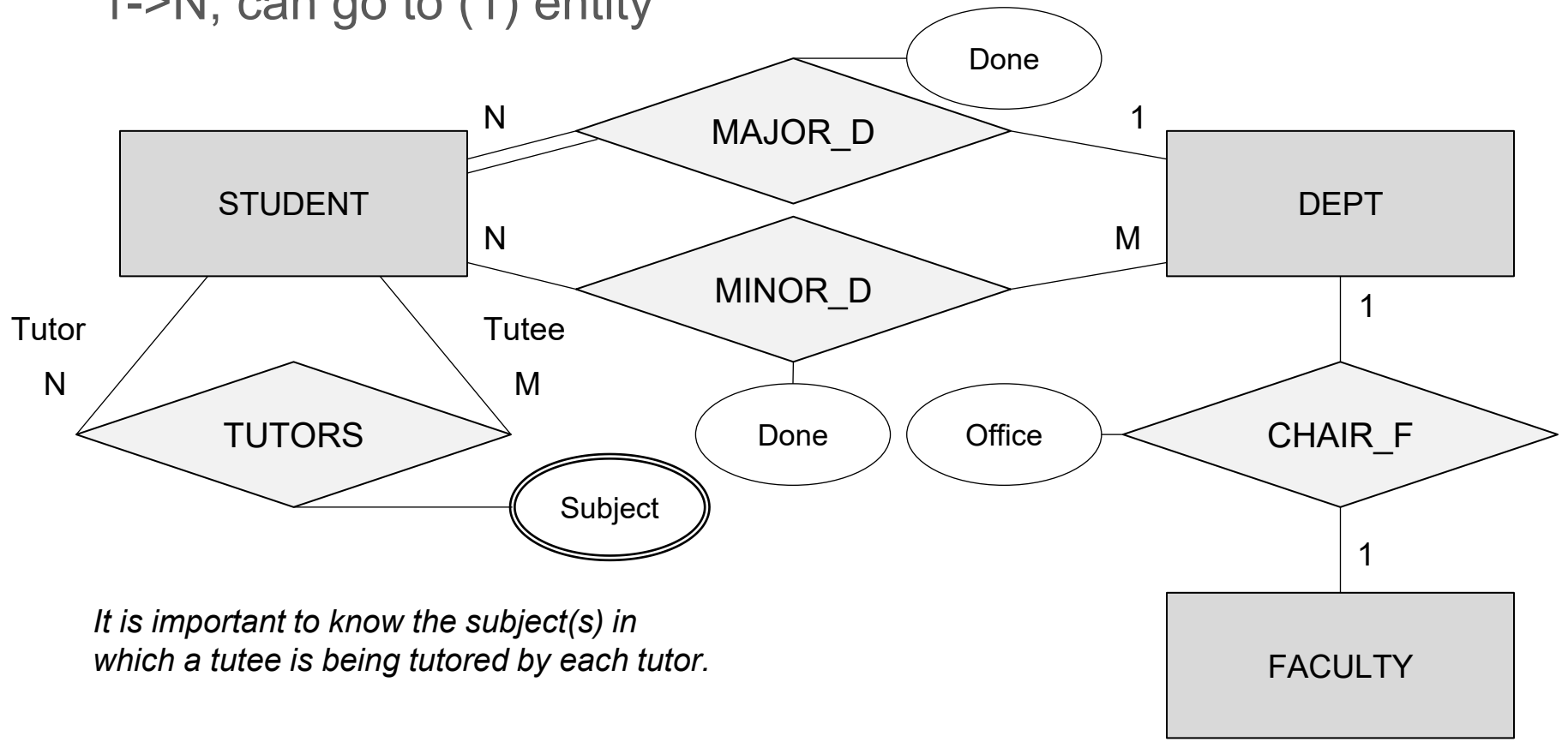
1->1, can go to either entity
1->N, can go to (1) entity



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

Attributes of Relationships

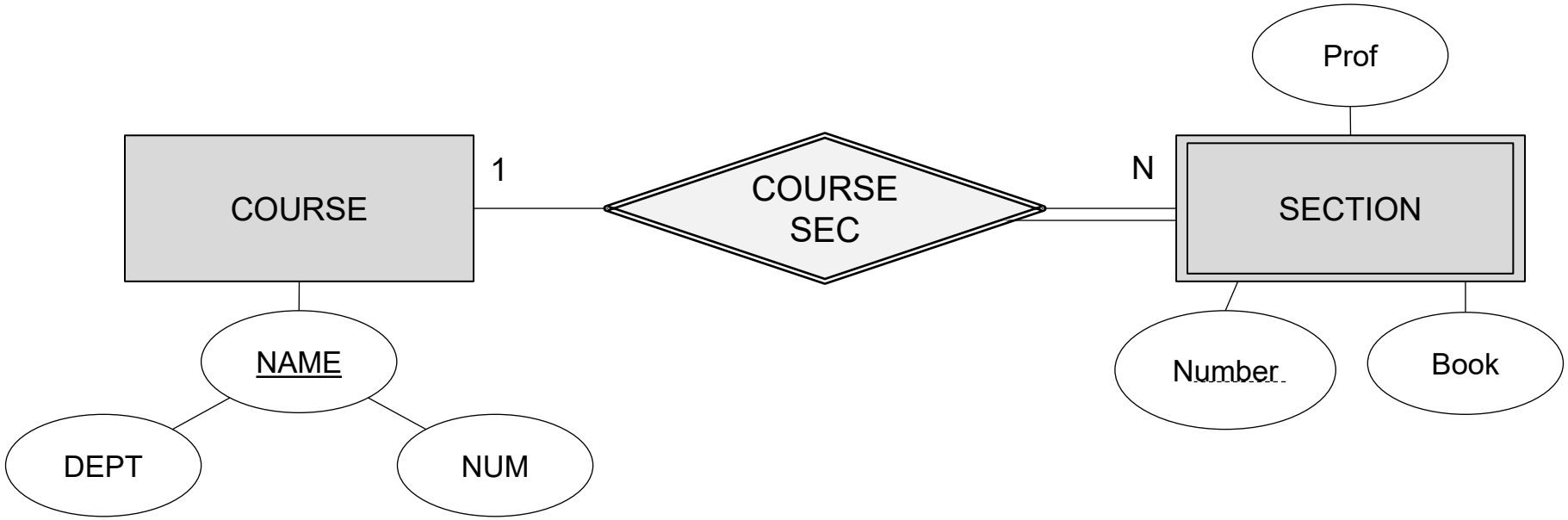
1->1, can go to either entity
1->N, can go to (1) entity



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)



Exercise 1

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.

Exercise 2

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.

Exercise 3

Draw an ERD for the following description:

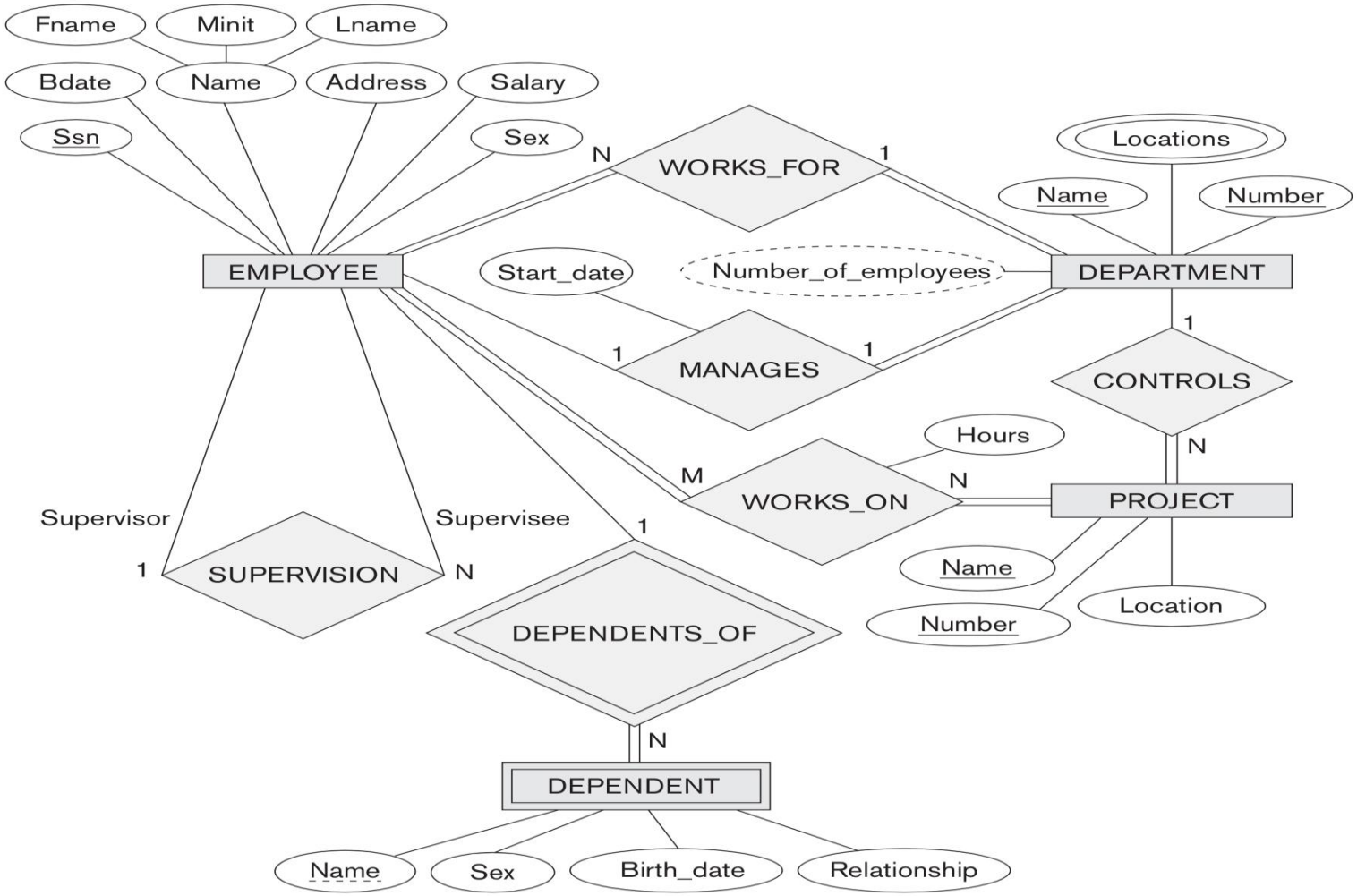
We store each employee's name (first, last, MI), Social Security number (SSN), street address, salary, sex (gender), and birth date. An employee is assigned to one department, but may work on several projects, which are not necessarily controlled by the same department. We keep track of the current number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

Exercise 4

Draw an ERD for the following description:

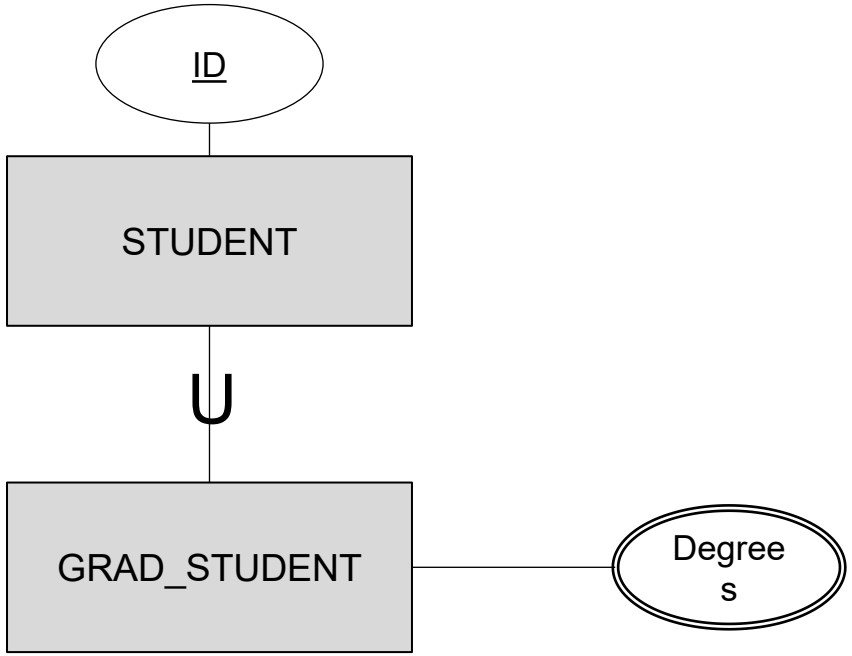
We want to keep track of the dependents of each employee for insurance purposes. We keep each dependent's first name, sex, birth date, and relationship to the employee.

All Together Now!



Specialization/Generalization

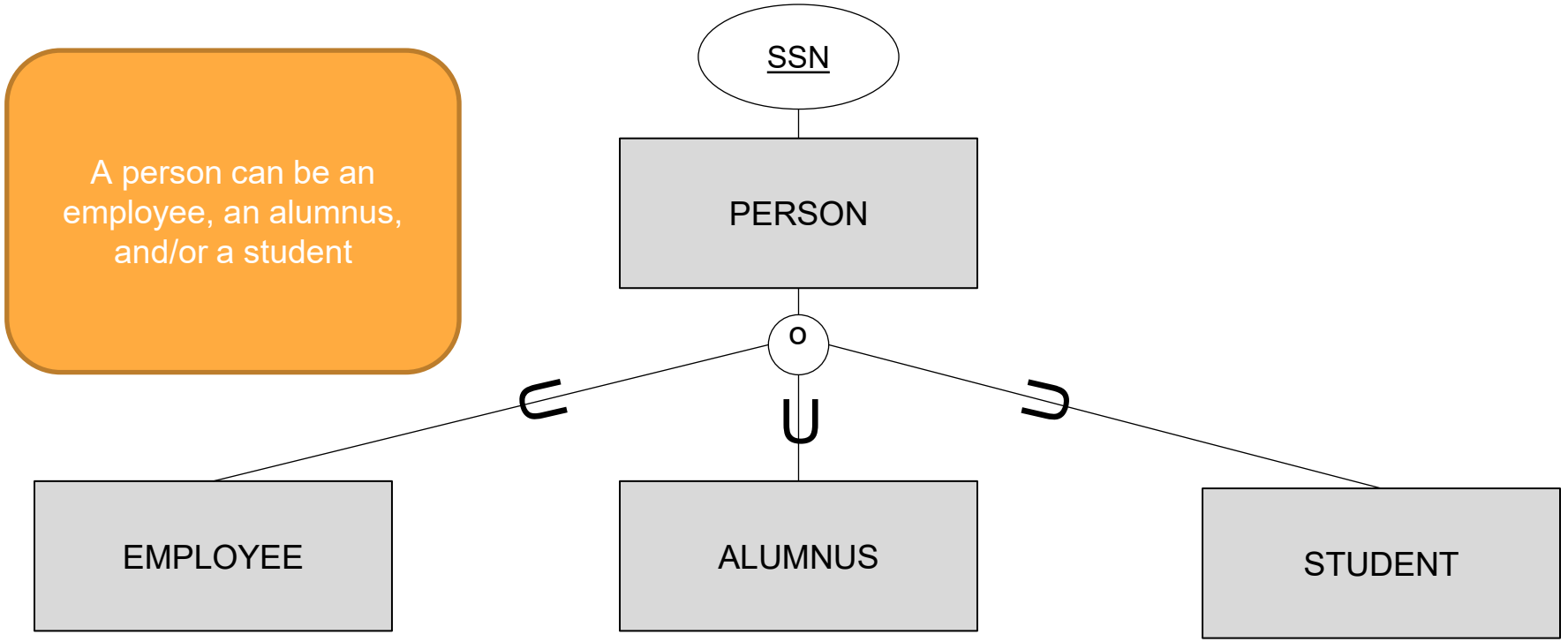
Only a subset of entities within a type have certain attributes or participate in certain relationships



Multiple Subtypes: Disjointedness

(o)verlap: may be more than one

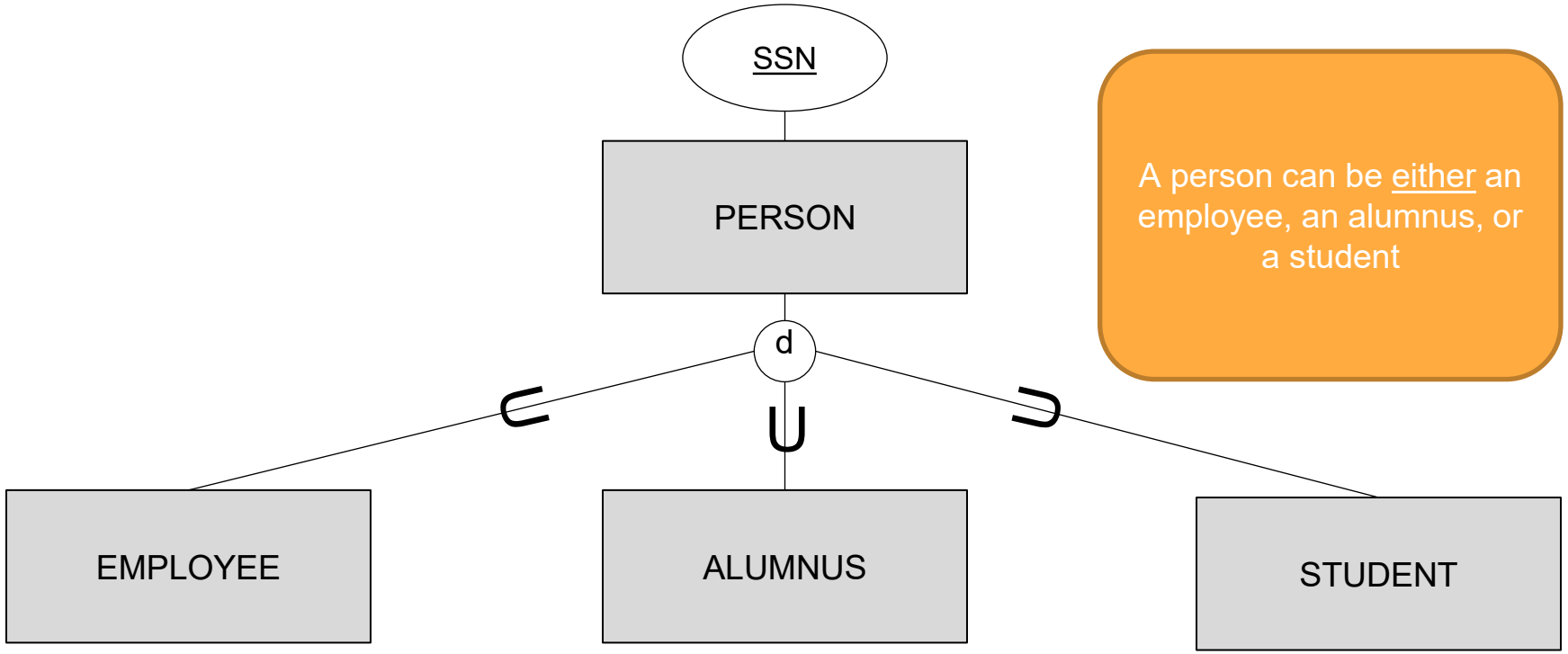
(d)isjoint: entities may *only be one* subtype



Multiple Subtypes: Disjointedness

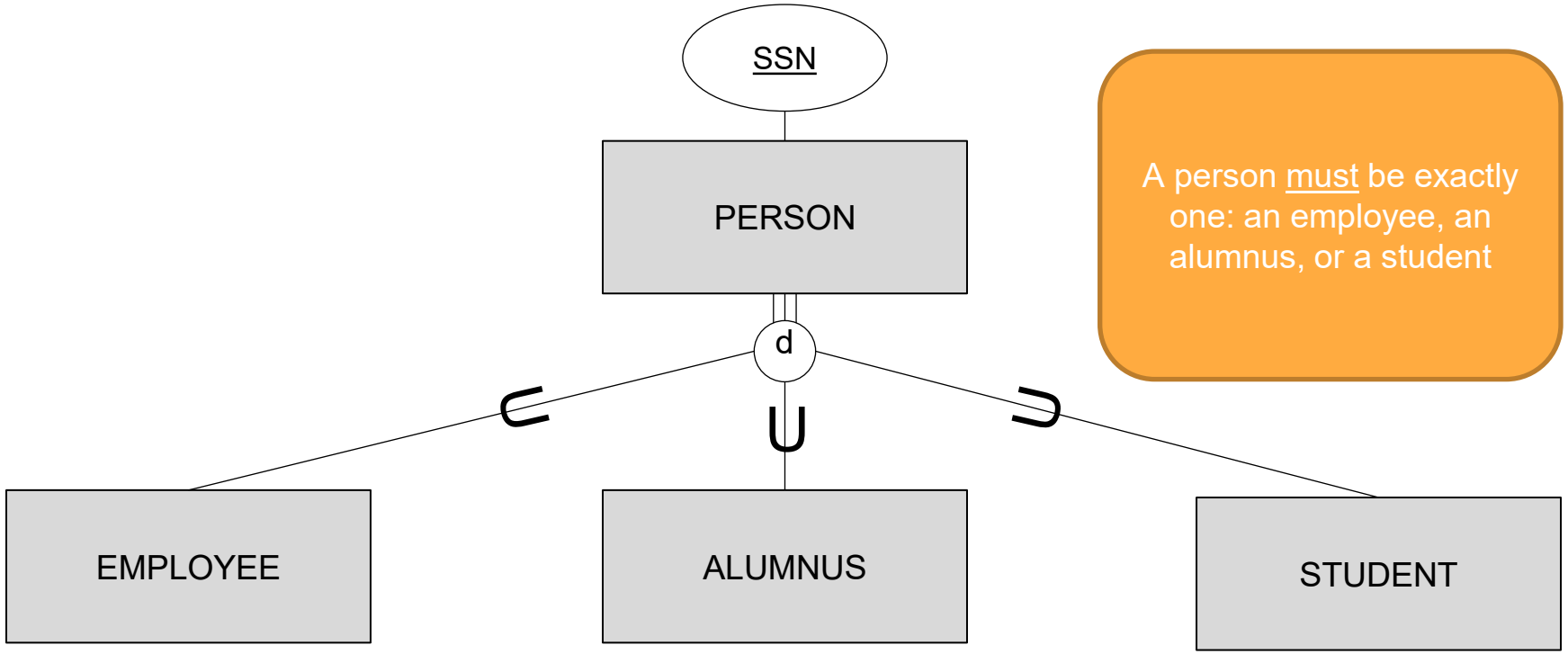
(o)verlap: may be more than one

(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)



Requirements Elicitation (1)

The conceptual model should *inform* requirements elicitation questions:

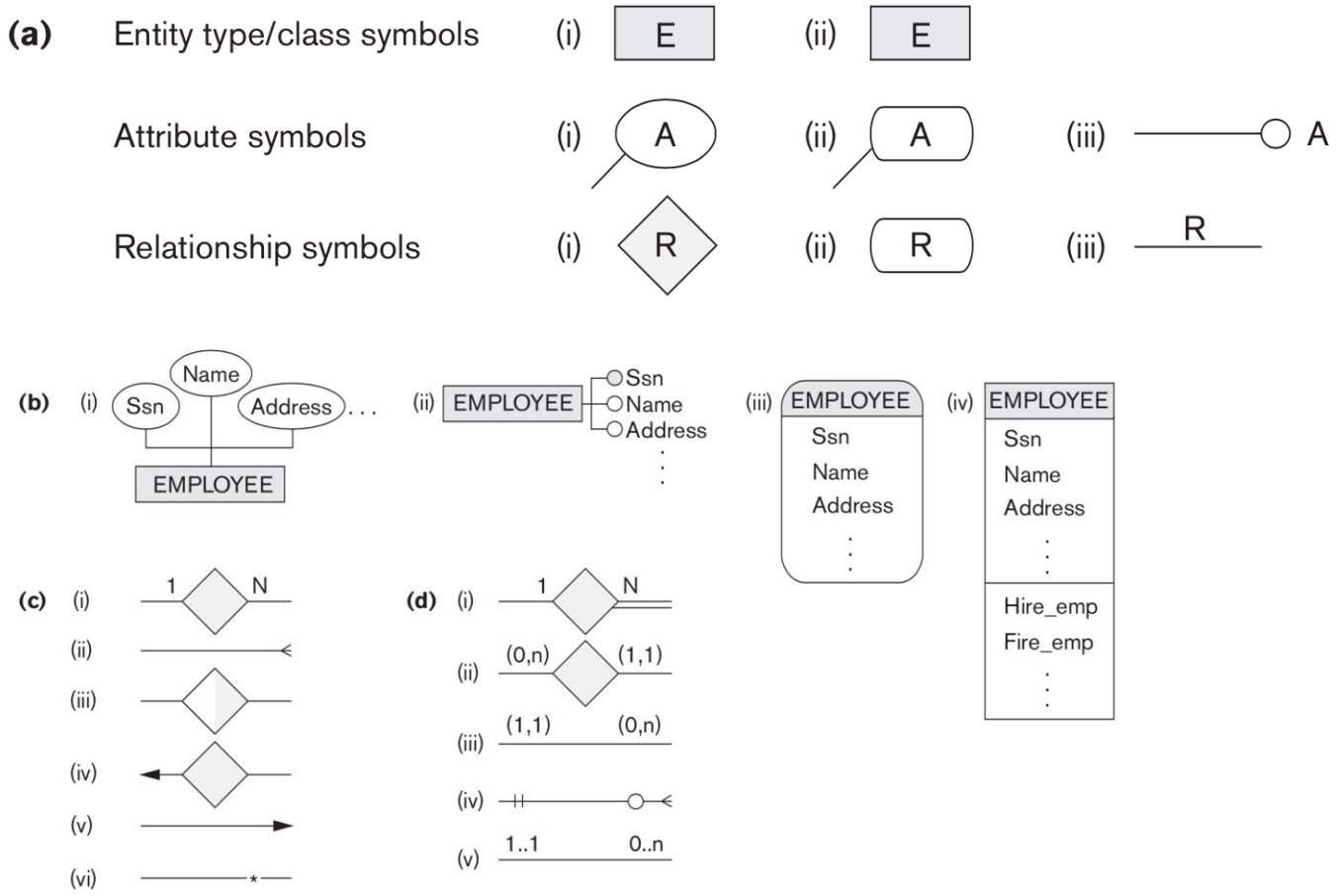
- What are the main kinds of objects to be stored in the database (*entity types*)?
- For each object, what information should be stored (*attributes*, *relationships*)? What information distinguishes one object of a type from another (*keys*, *weak entities*)? Are there different kinds/categories of objects (*specialization*/*generalization*)?
- For each piece of information, what characterizes a valid value (*composite*/*multi-valued*, 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*)?

Requirements Elicitation (2)

More on requirements elicitation in the Lab

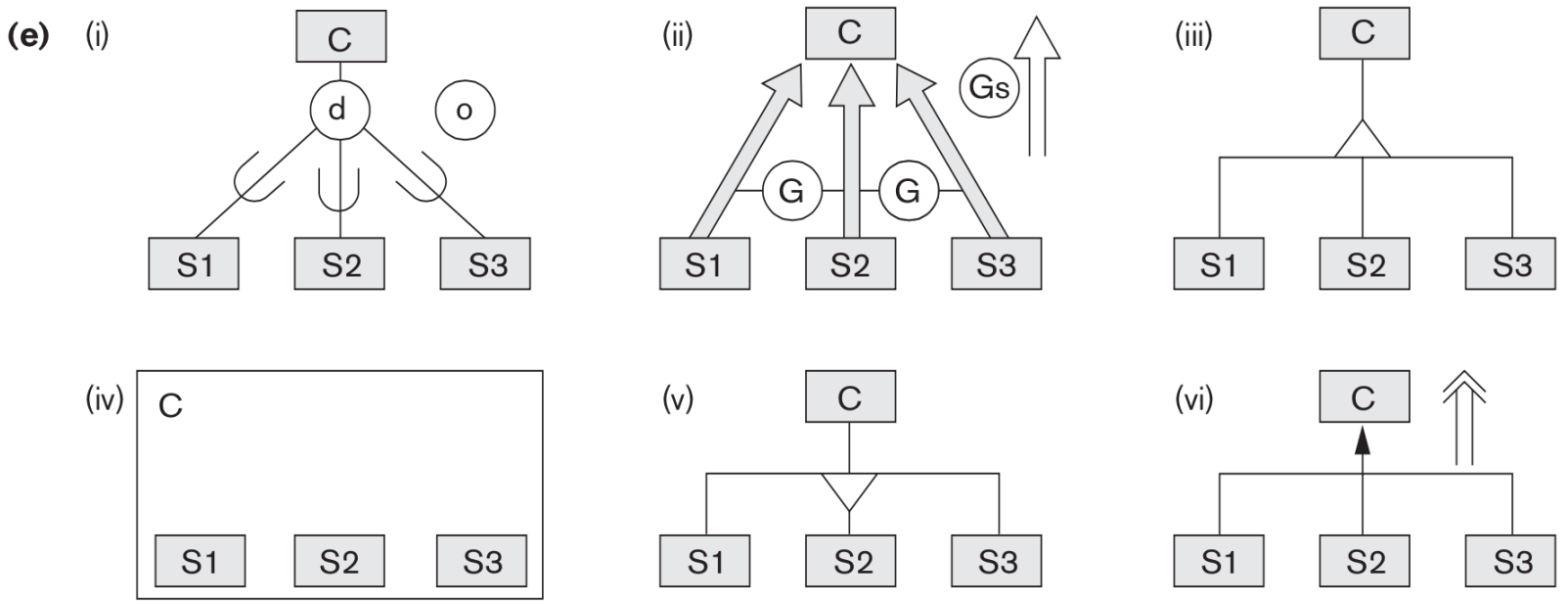
Alternative Notation (1)

Figure A. 1



Alternative Notation (2)

Figure A. 1



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.

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

Acknowledgement

- The major part of the teaching agenda for this course is based on material developed by Nate Derbinsky (https://derbinsky.info/)