Conceptual Design, ER Diagrams

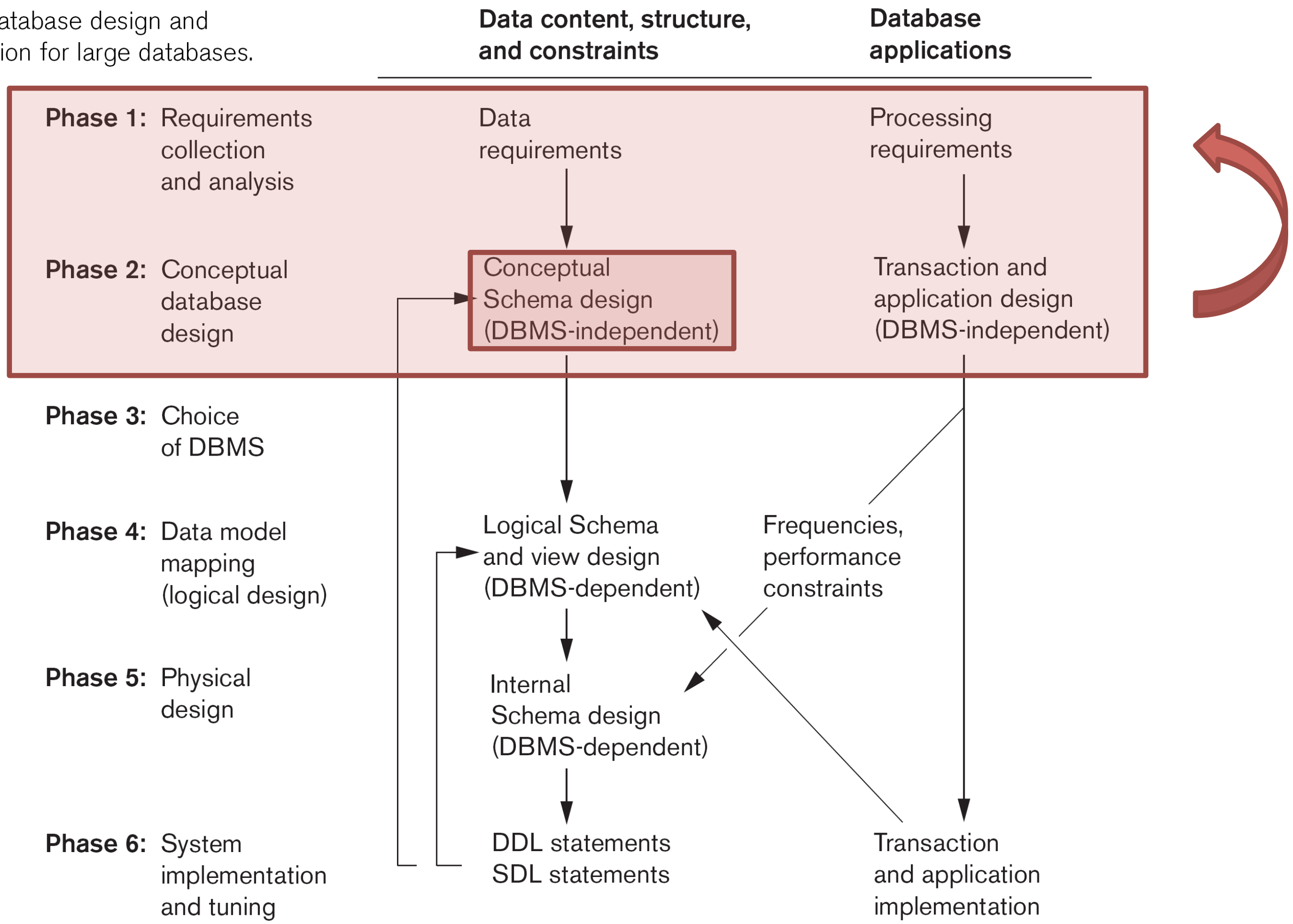
Lecture 2

Outline

1. Context: Design & Implementation Process
2. Goals of Conceptual Design
3. The 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.



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…

- 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 into implementation data model
  - 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

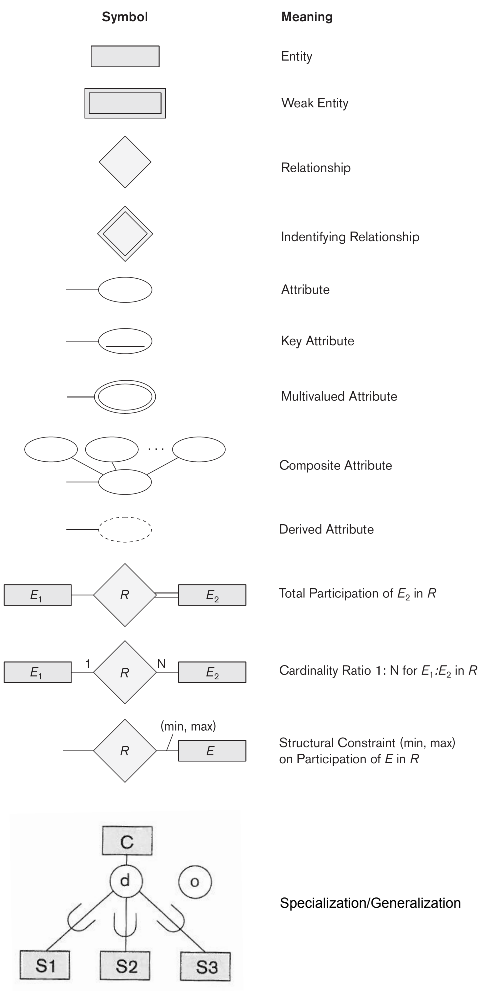
Relationship

- 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.



Symbol

Meaning

Entity

Weak Entity

Relationship

Indentifying Relationship

Attribute

Key Attribute

Multivalued Attribute

Composite Attribute

Derived Attribute

Total Participation of E_2 in R

Cardinality Ratio 1: N for E_1:E_2 in R

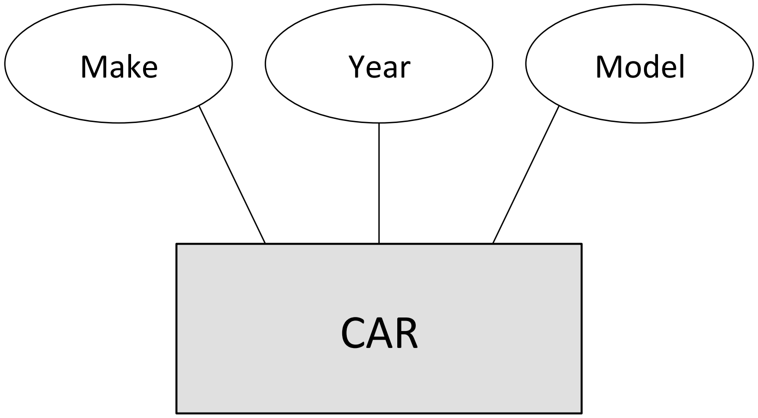
Structural Constraint (min, max) on Participation of E in R

Specialization/Generalization

Entity Sets

Set of entities that have the same attributes

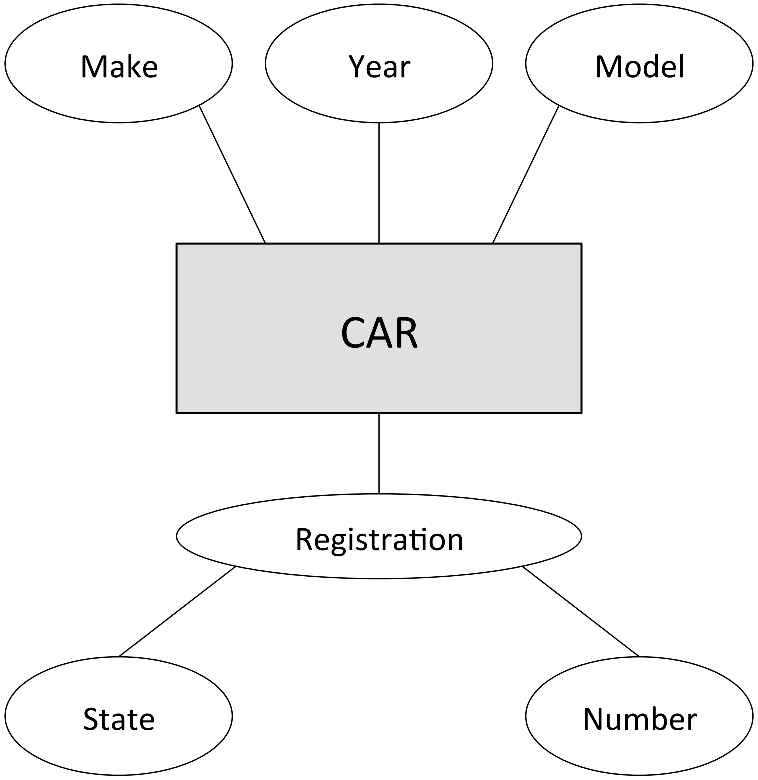
All cars have a year, make, and model.



Composite Attributes

Can be subdivided into smaller subparts

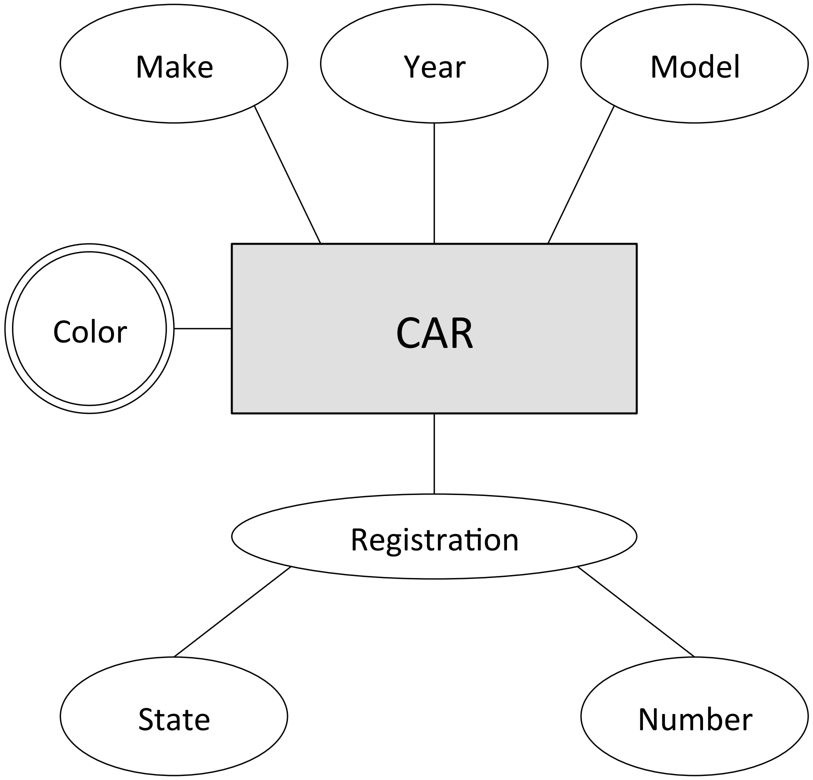
All cars have a year, make, model, ***and registration****.*



Multivalued Attributes

Can take a [possibly specified] number of values.

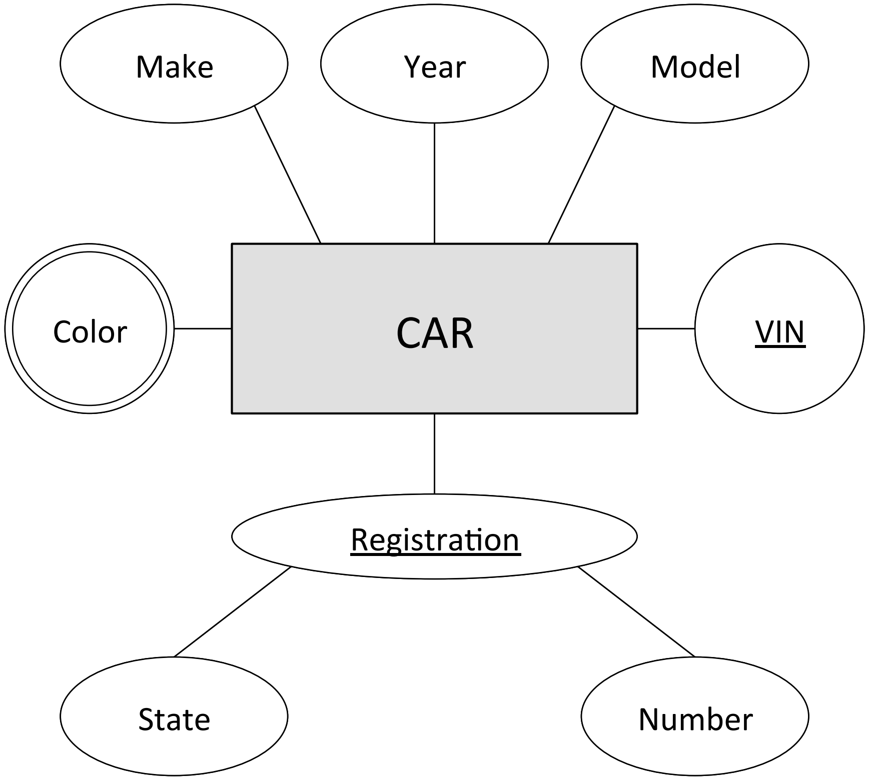
All cars have a year, make, model, registration, and ***some number of colors****.*



Key Attributes

The value uniquely identifies each entity

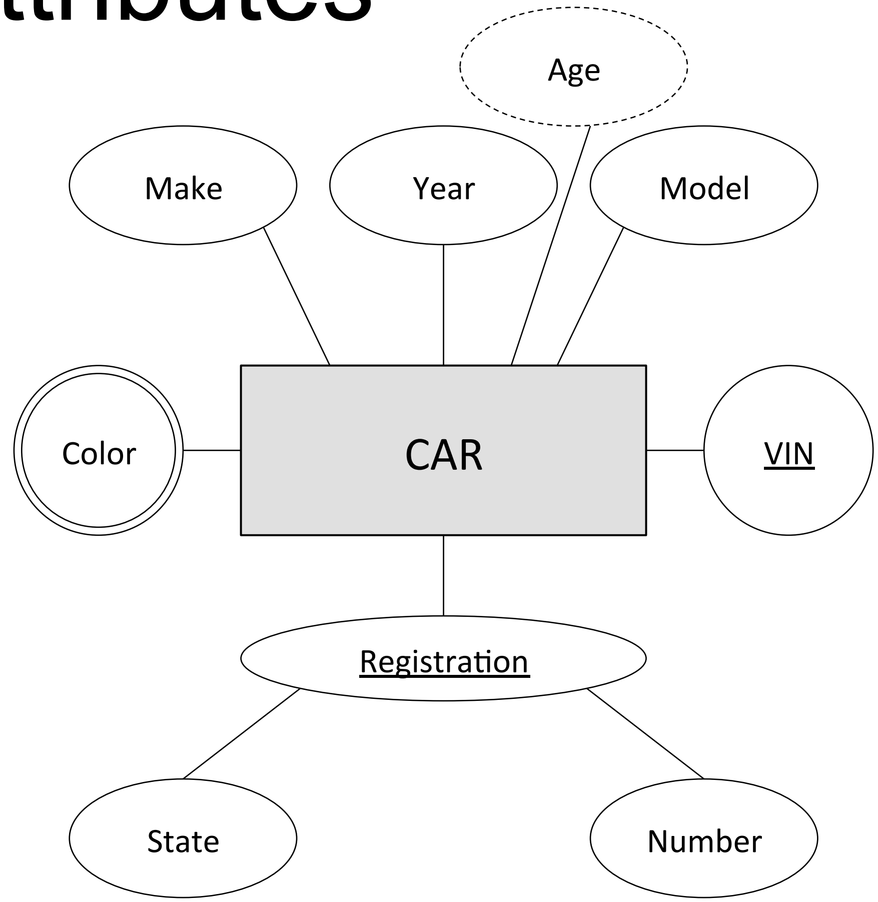
*All cars have a year, make, model, **registration (unique), vehicle number (vin; unique)**, some number of colors.*



Derived Attributes

The value can be computed

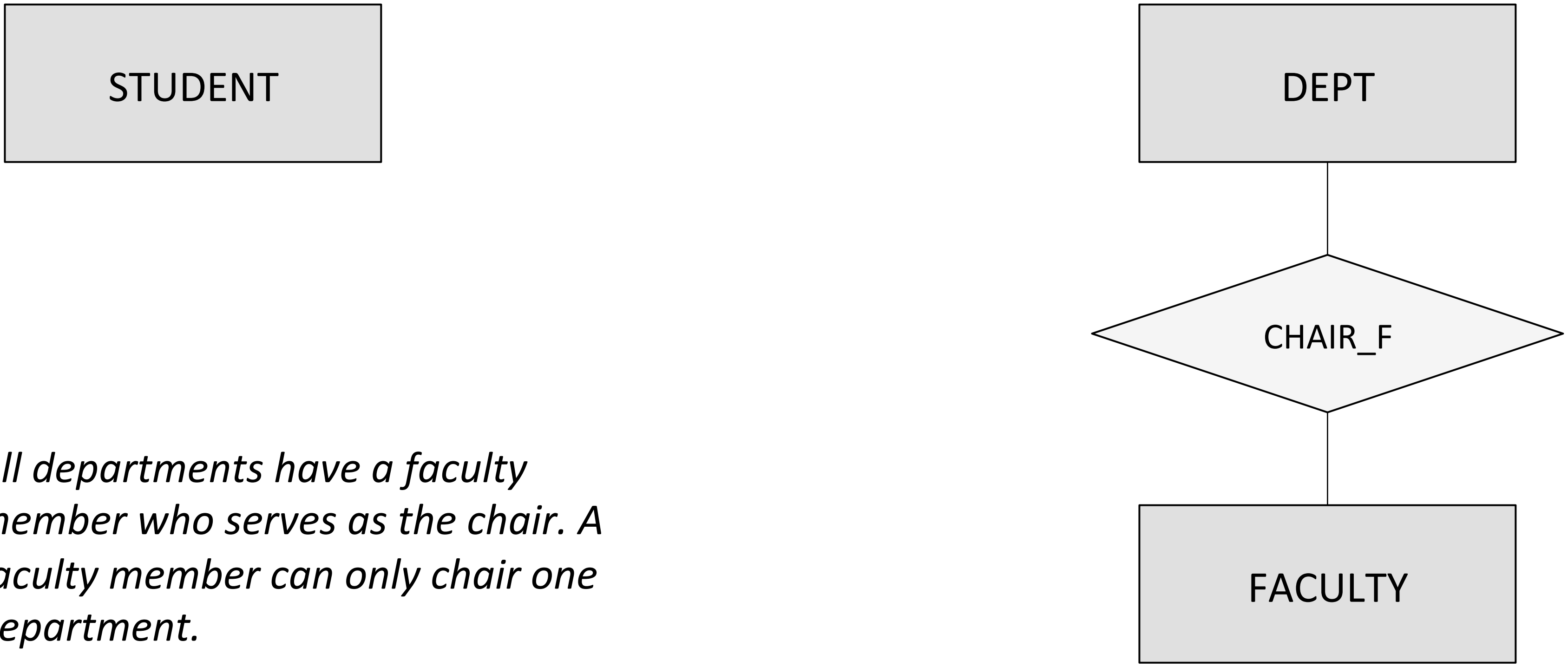
All cars have a year, ***age****, make, model, registration (unique), vehicle number (vin; unique), some number of colors.*



Relationships

Associates one or more sets of entities

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

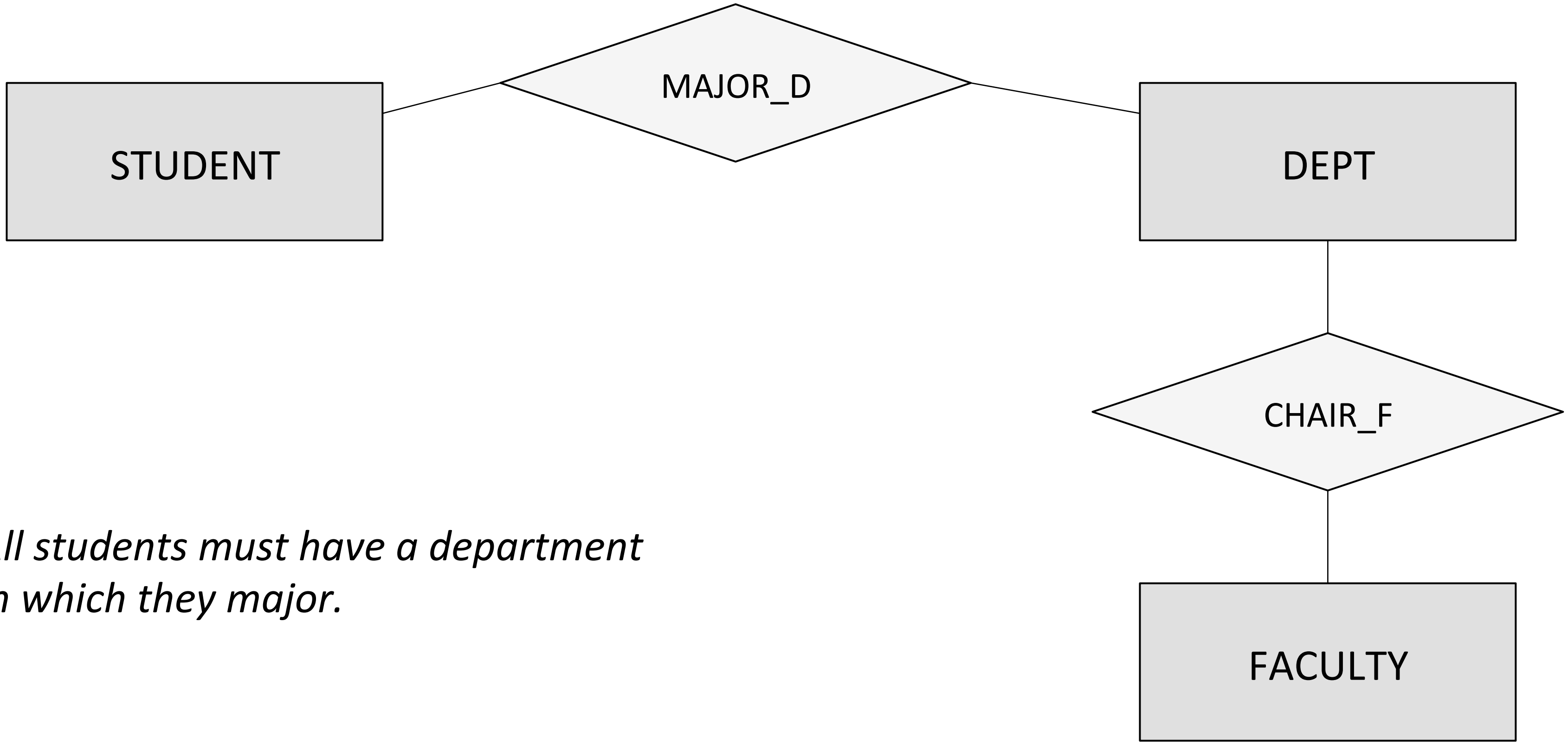


All departments have a faculty member who serves as the chair. A faculty member can only chair one department.

Relationships

Associates one or more sets of entities

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

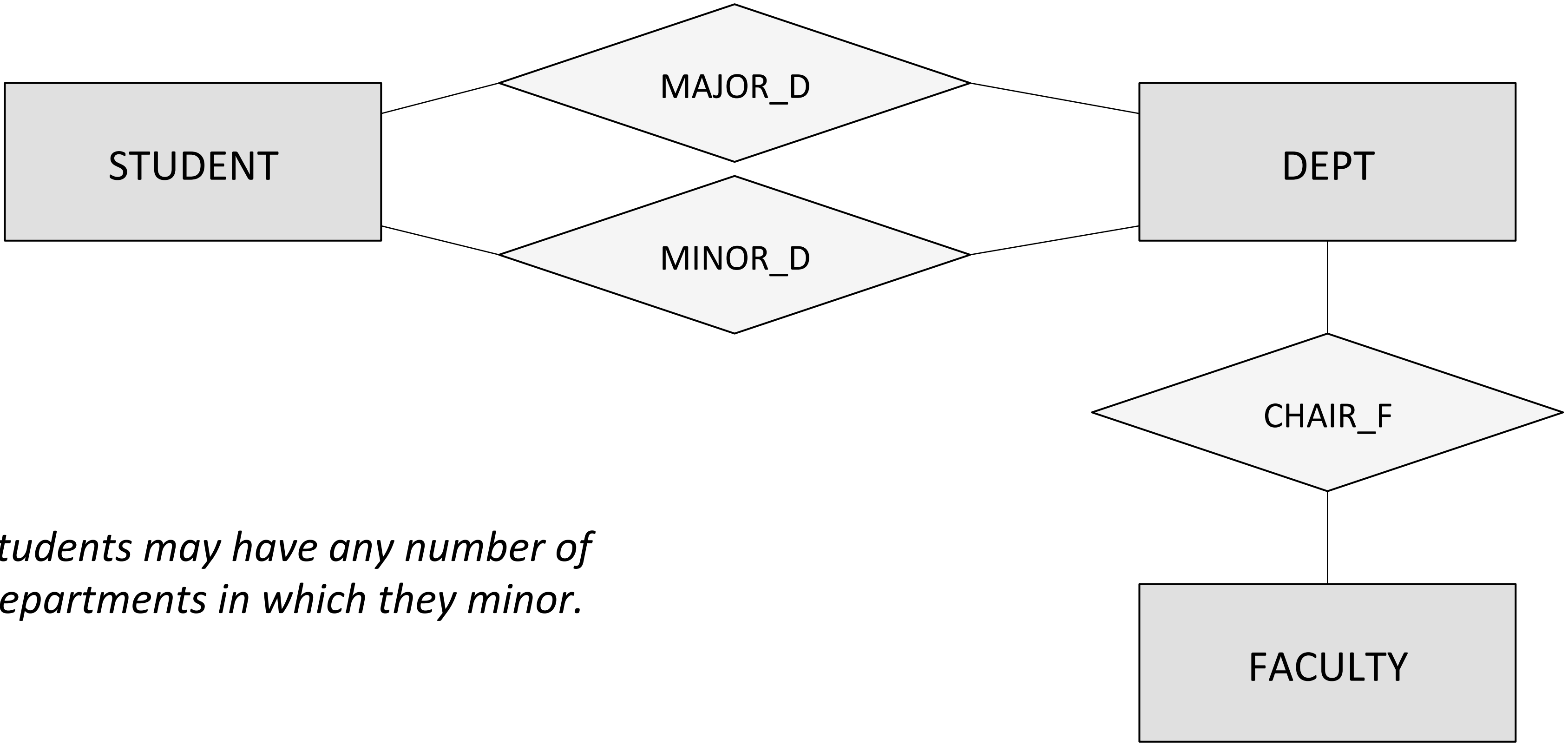


All students must have a department in which they major.

Relationships

Associates one or more sets of entities

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

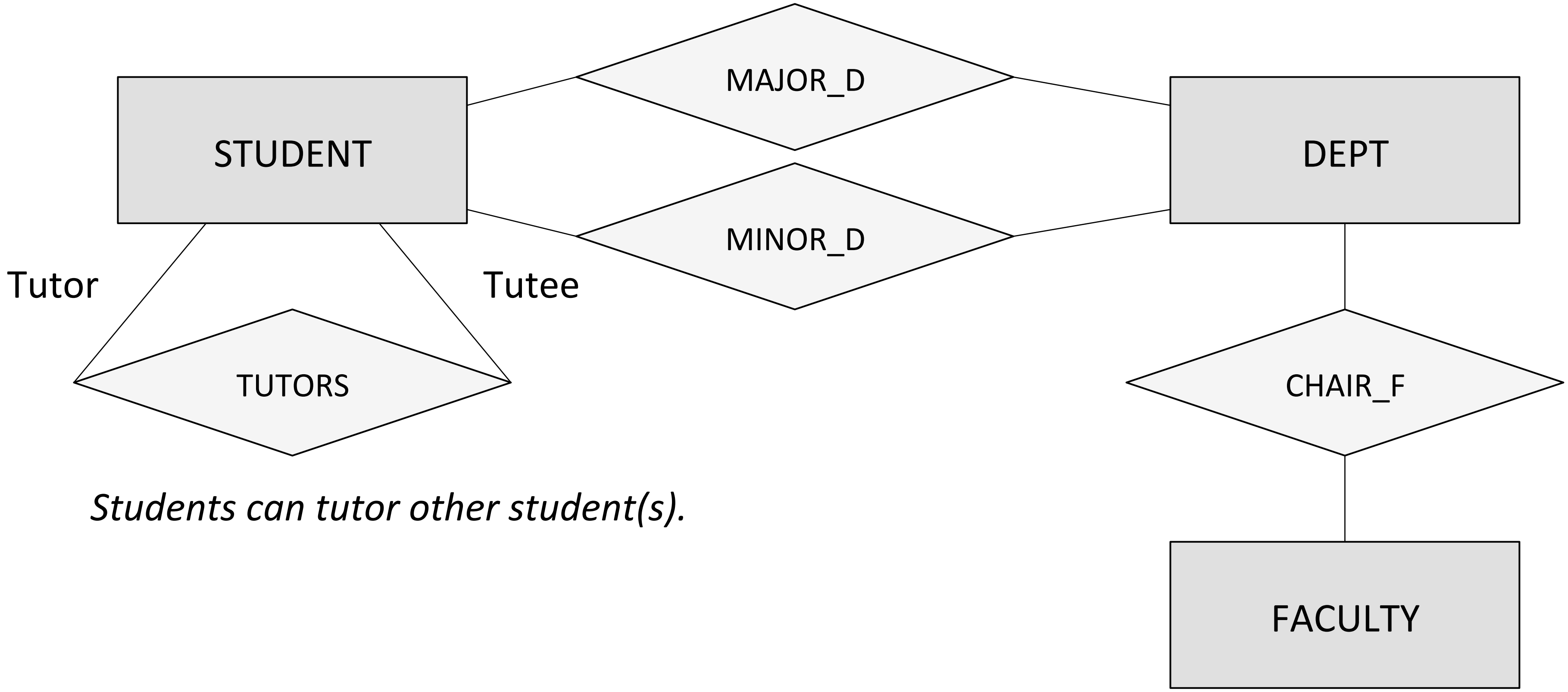


Students may have any number of departments in which they minor.

Relationships

Associates one or more sets of entities

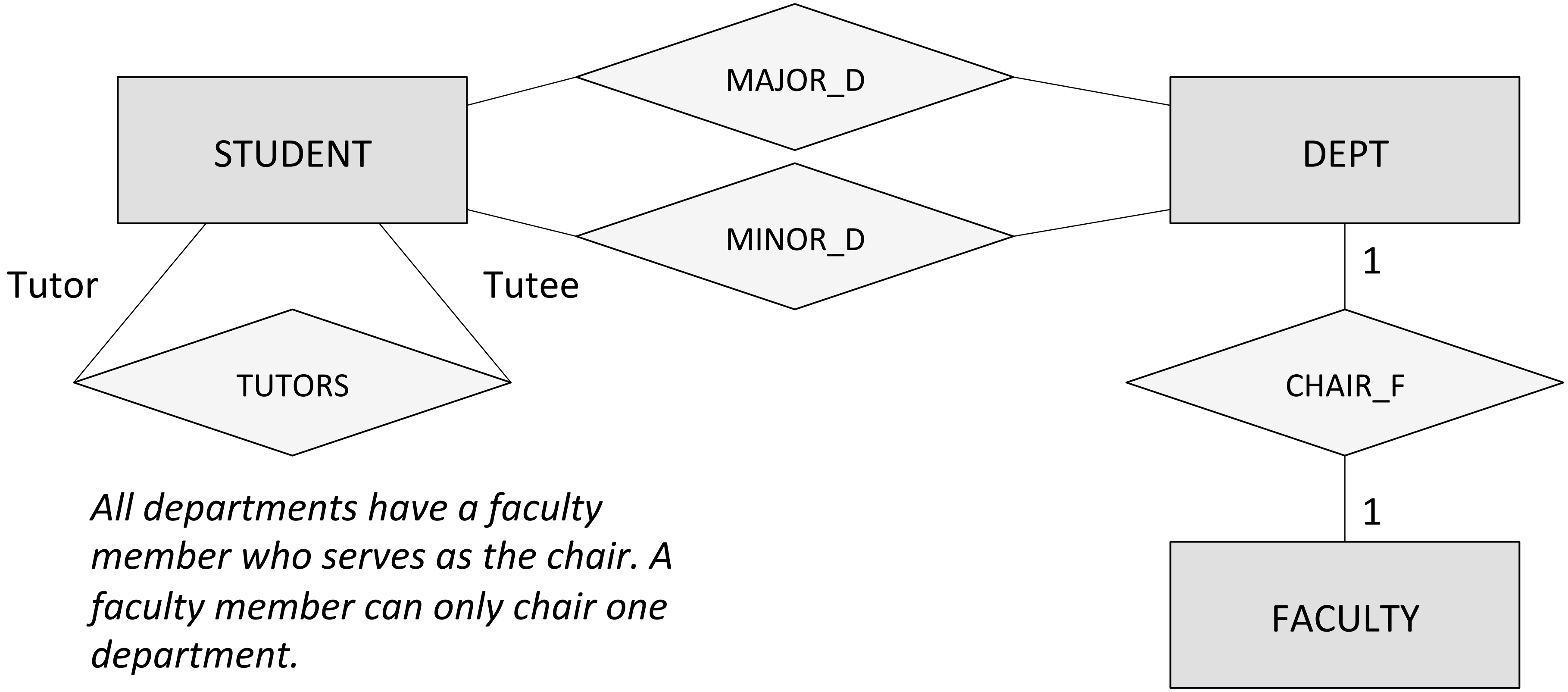
- One = recursive (**role** is important)



Students can tutor other student(s).

Cardinality Ratios

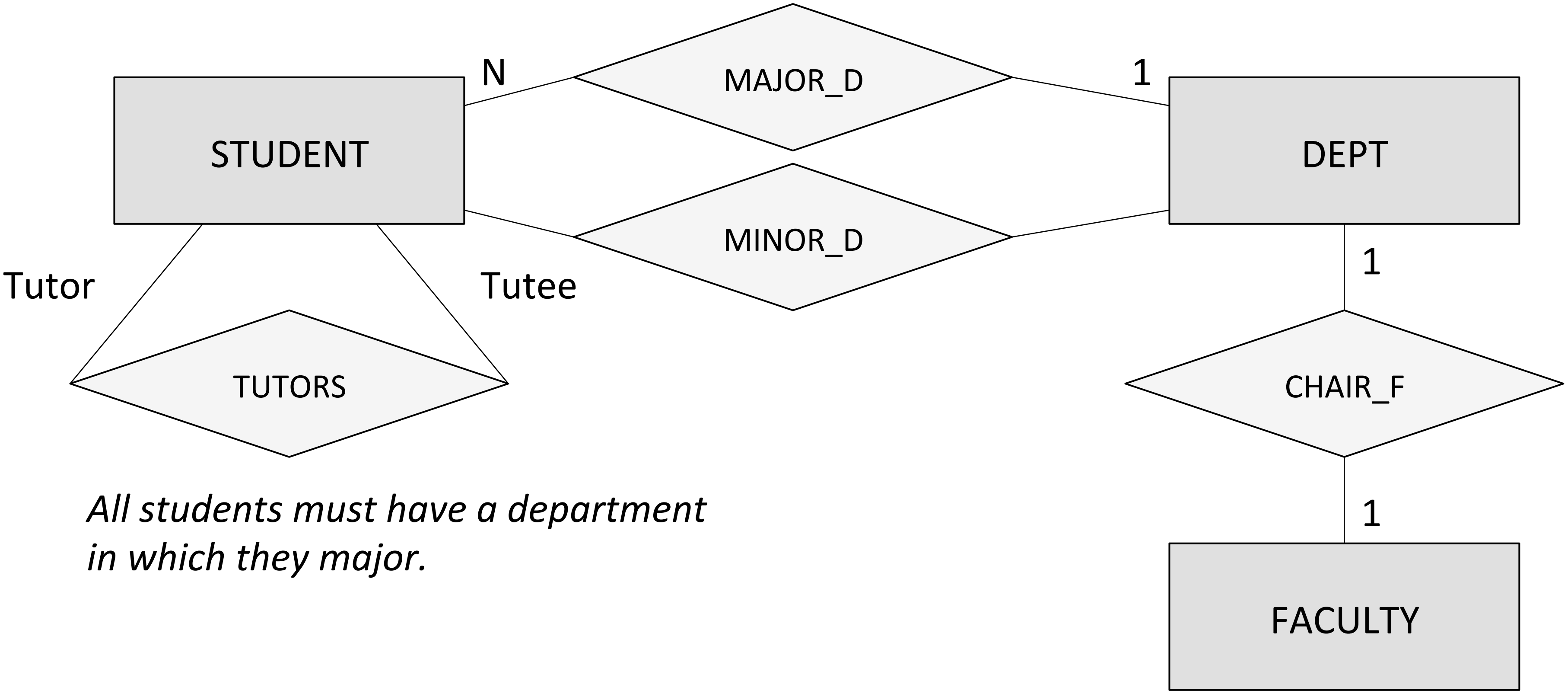
Constrains the number of entities that can participate in each role of the relationship



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

Cardinality Ratios

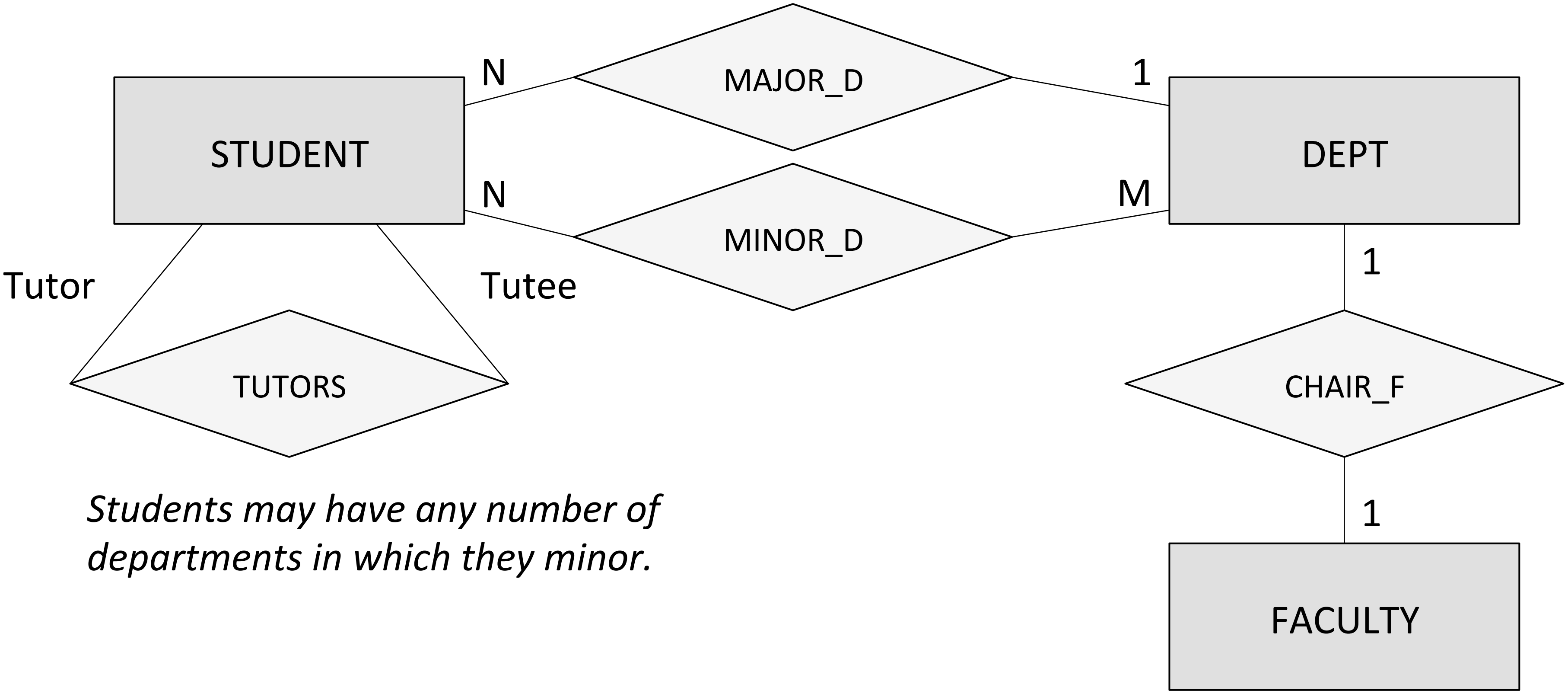
Constrains the number of entities that can participate in each role of the relationship



All students must have a department in which they major.

Cardinality Ratios

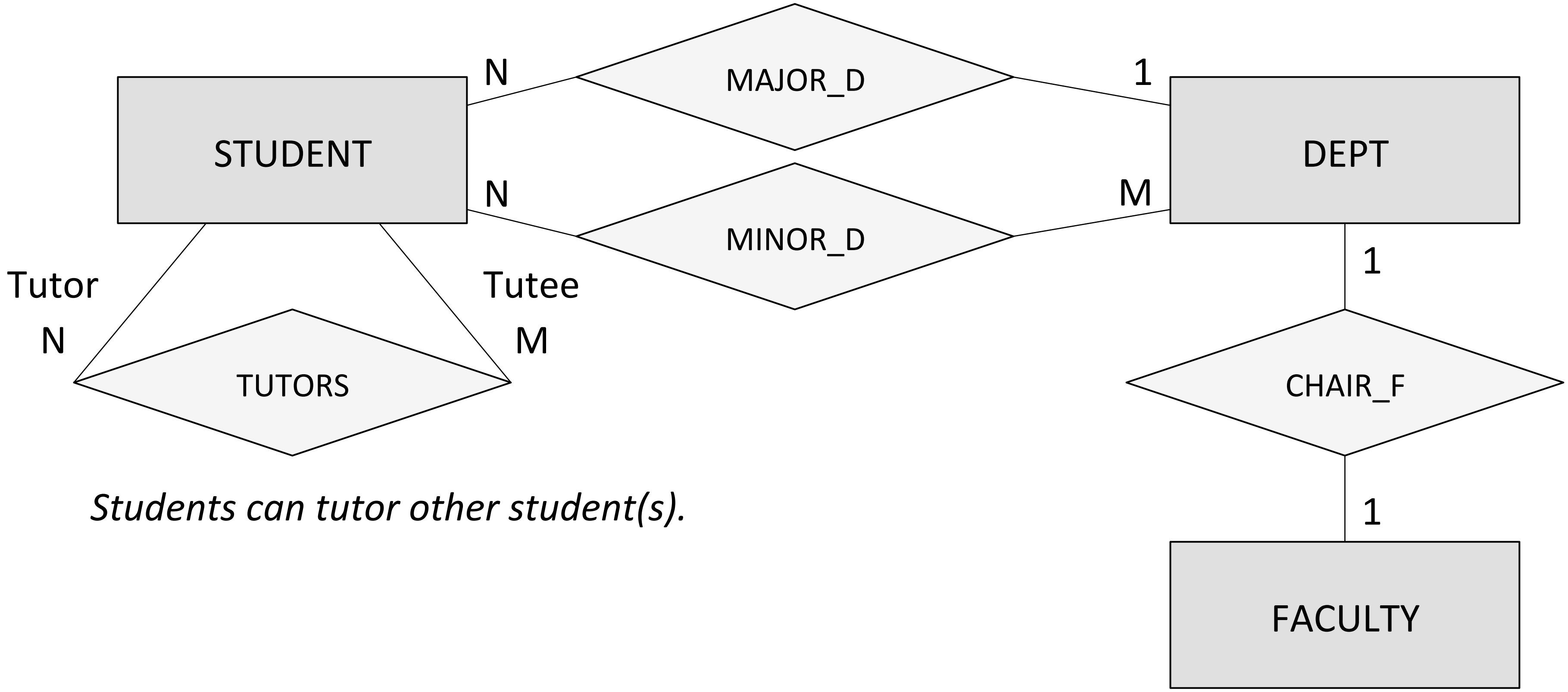
Constrains the number of entities that can participate in each role of the relationship



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

Cardinality Ratios

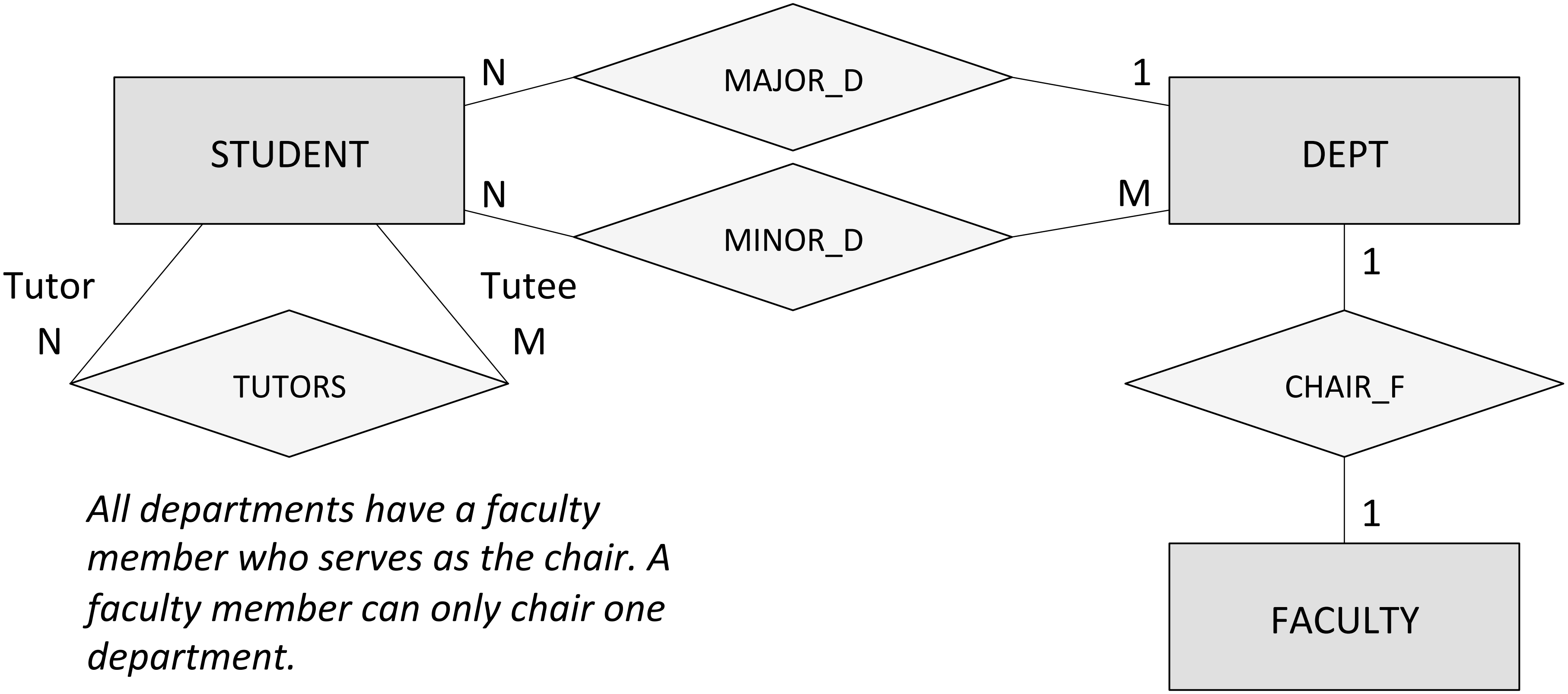
Constrains the number of entities that can participate in each role of the relationship



Students can tutor other student(s).

Structural Constraints

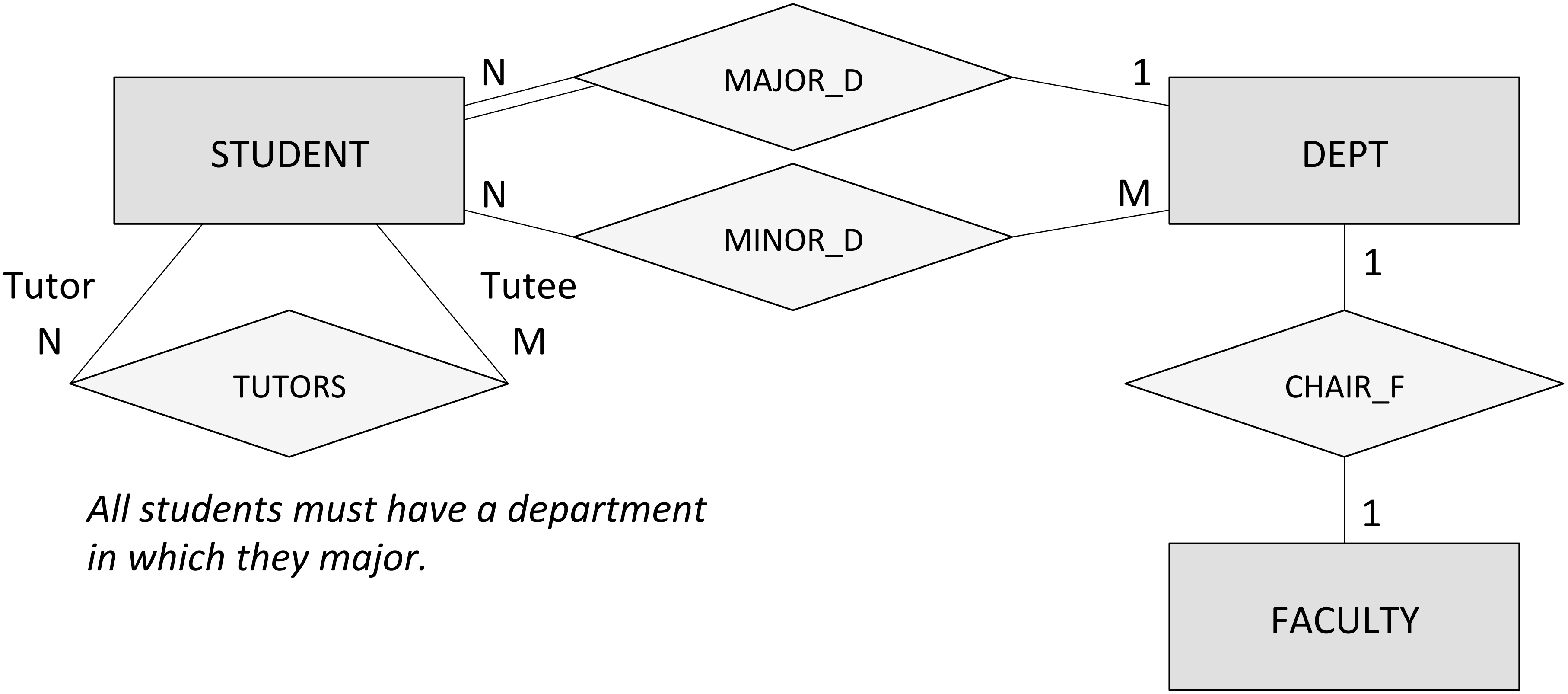
If an entity does not exist unless it appears with an entity in a relationship, the participation is **total** (existence dependency). Else, **partial**.



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**.

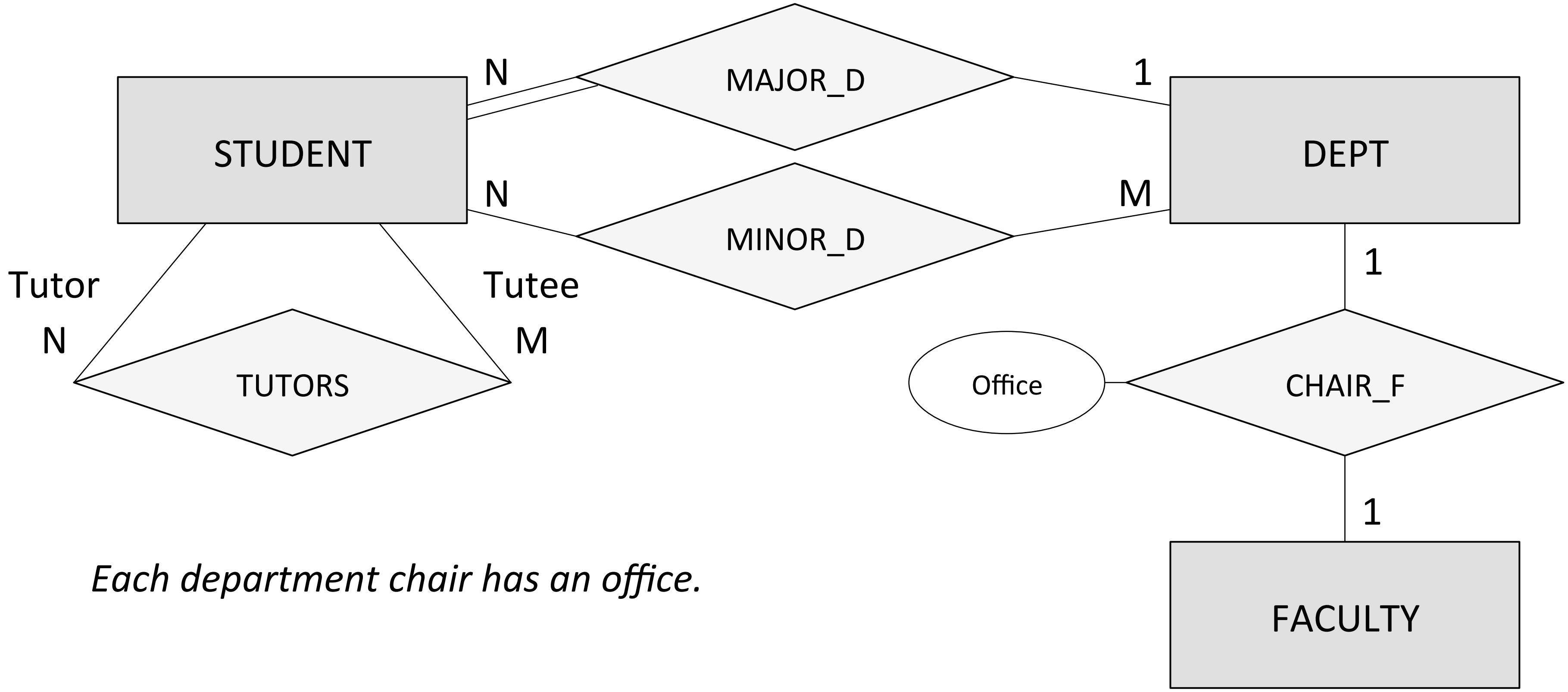


All students must have a department in which they major.

Attributes of Relationships

1->1, can go to either entity

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

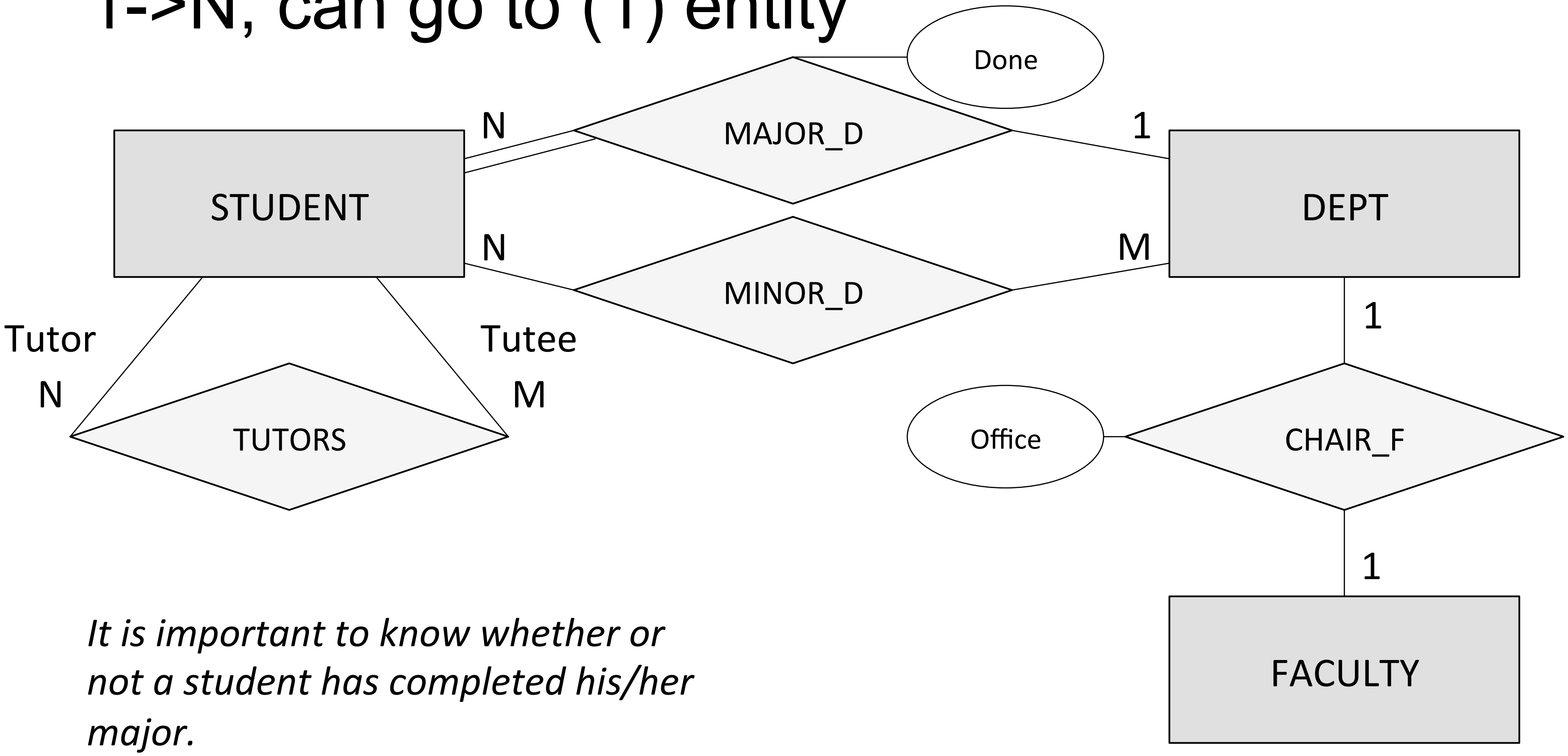


Each department chair has an office.

Attributes of Relationships

1->1, can go to either entity

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

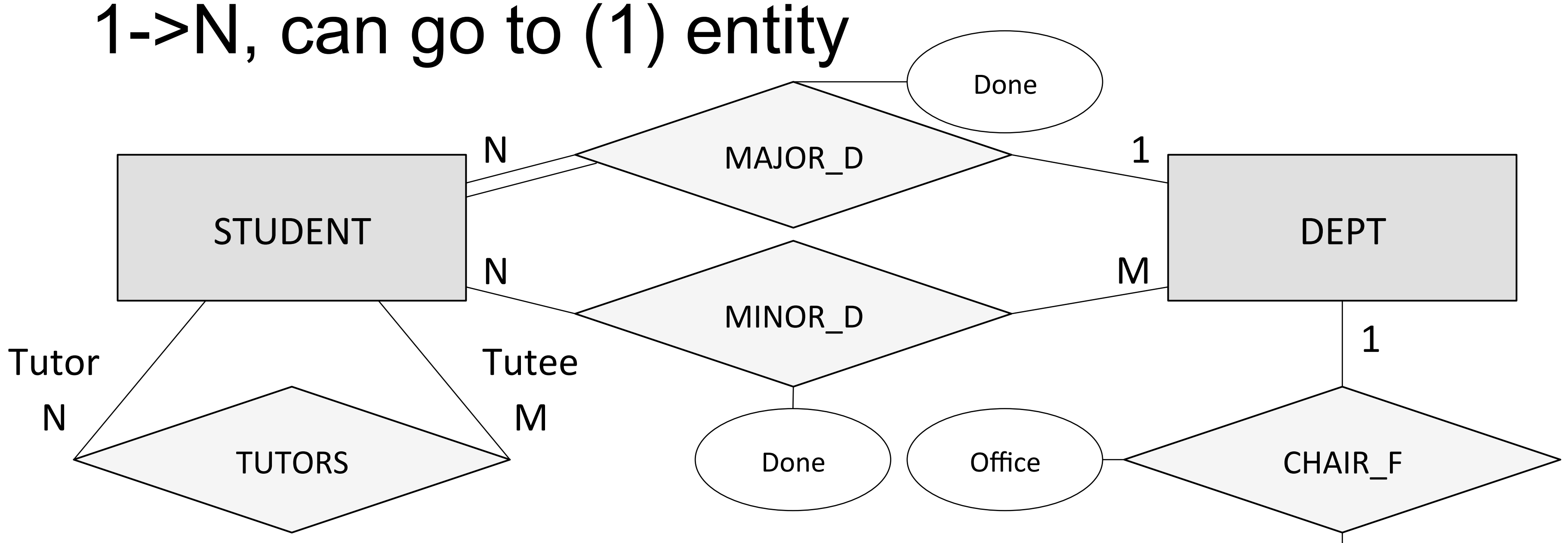


It is important to know whether or not a student has completed his/her major.

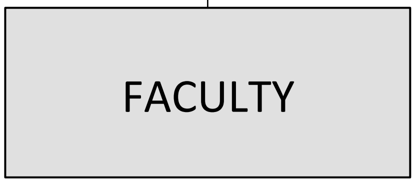
Attributes of Relationships

1->1, can go to either entity

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



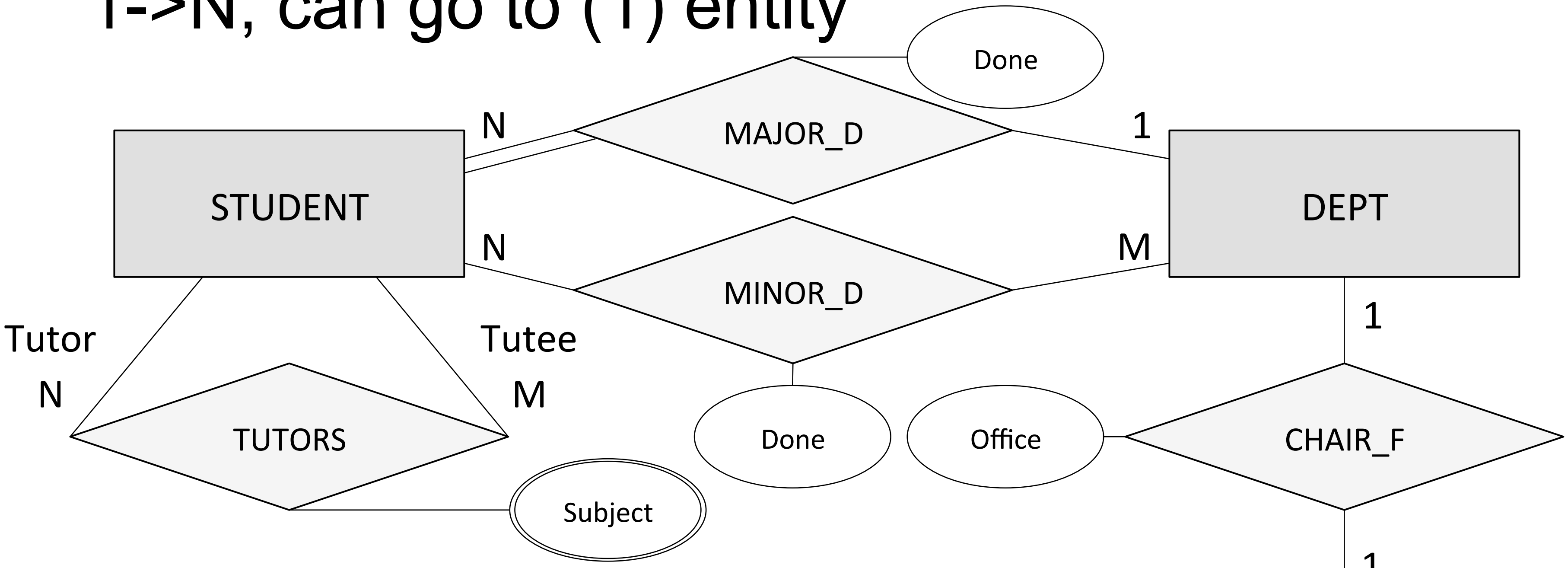
It is important to know whether or not a student has completed each of his/her minor(s).



Attributes of Relationships

1->1, can go to either entity

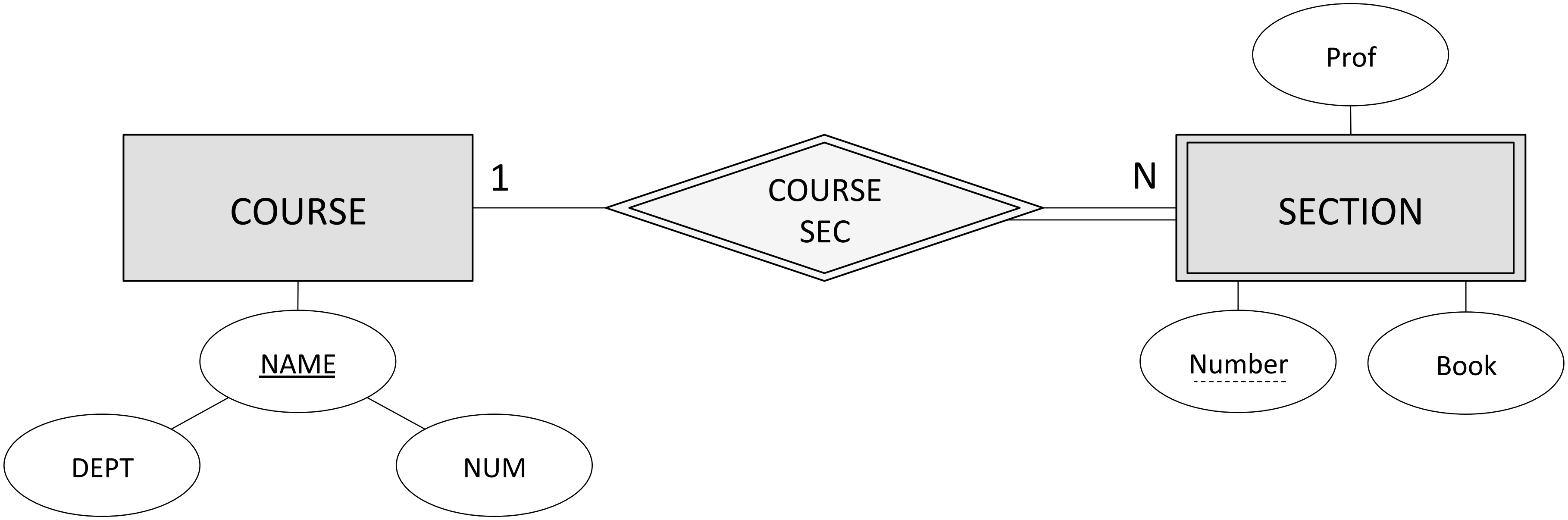
1->N, can go to (1) entity



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

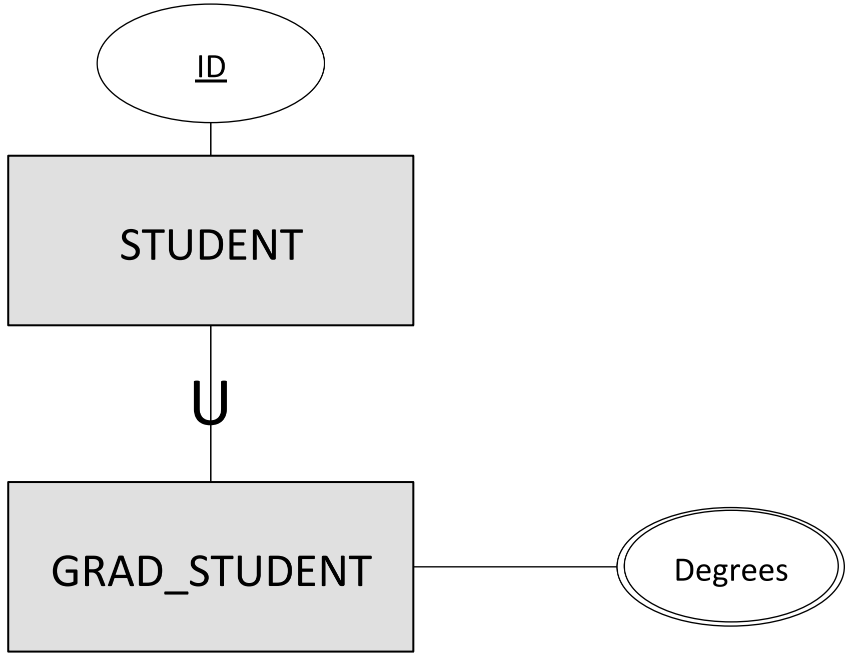
Weak Entities

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)



Specialization/Generalization

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

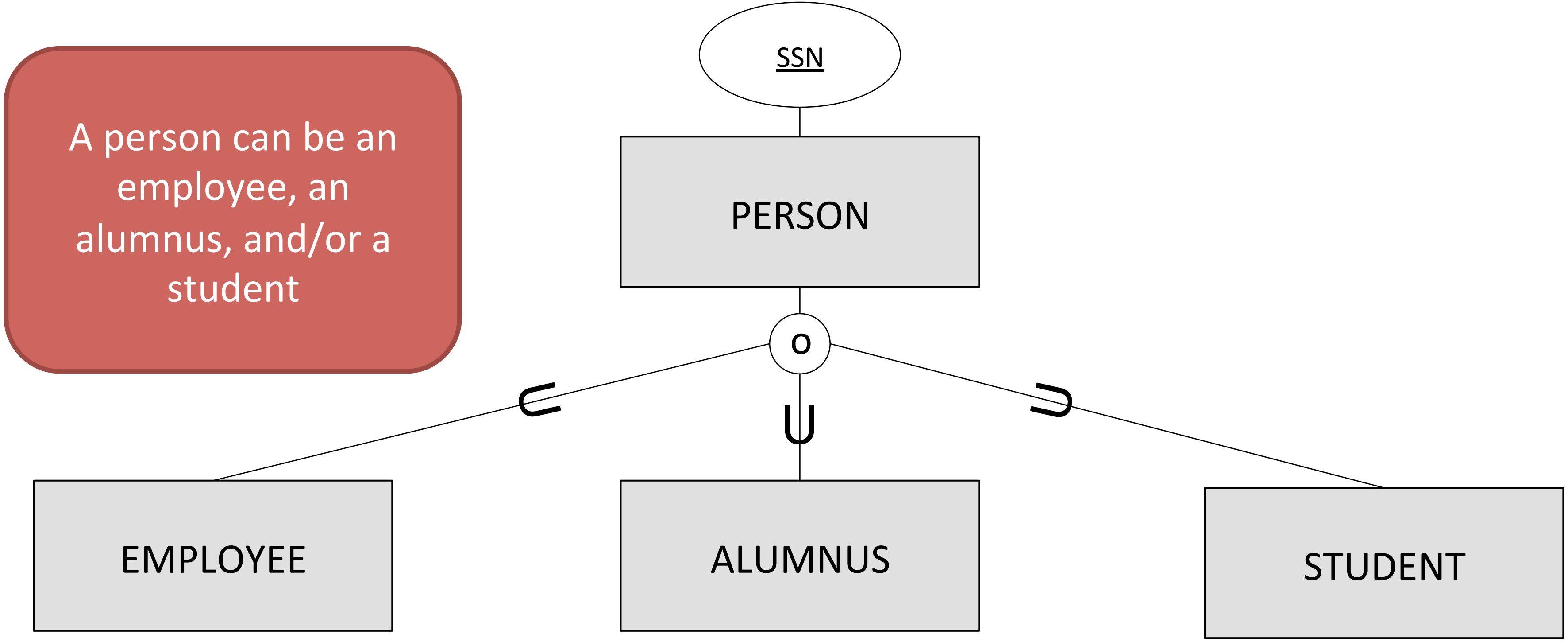


Multiple Subtypes: Disjointedness

(o)verlap: may be more than one

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

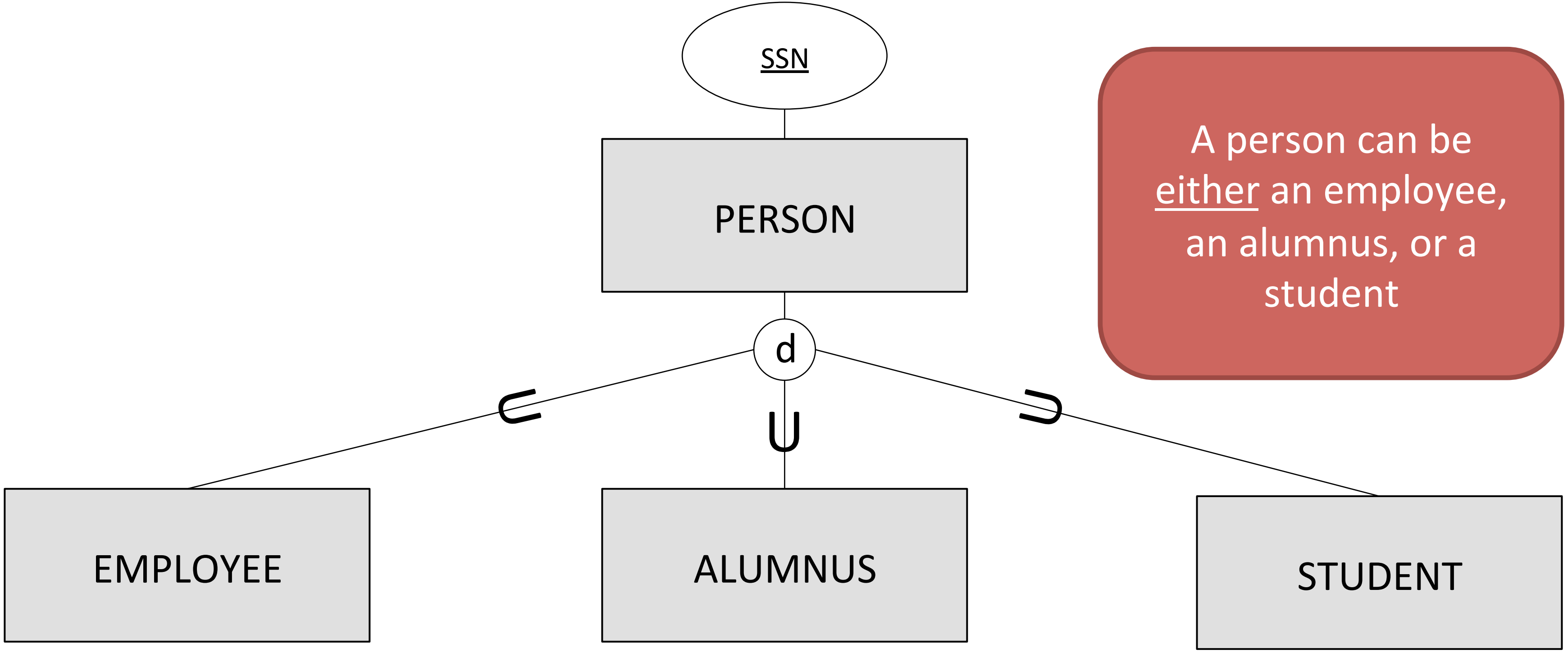
A person can be an employee, an alumnus, and/or a student



Multiple Subtypes: Disjointedness

(o)verlap: may be more than one

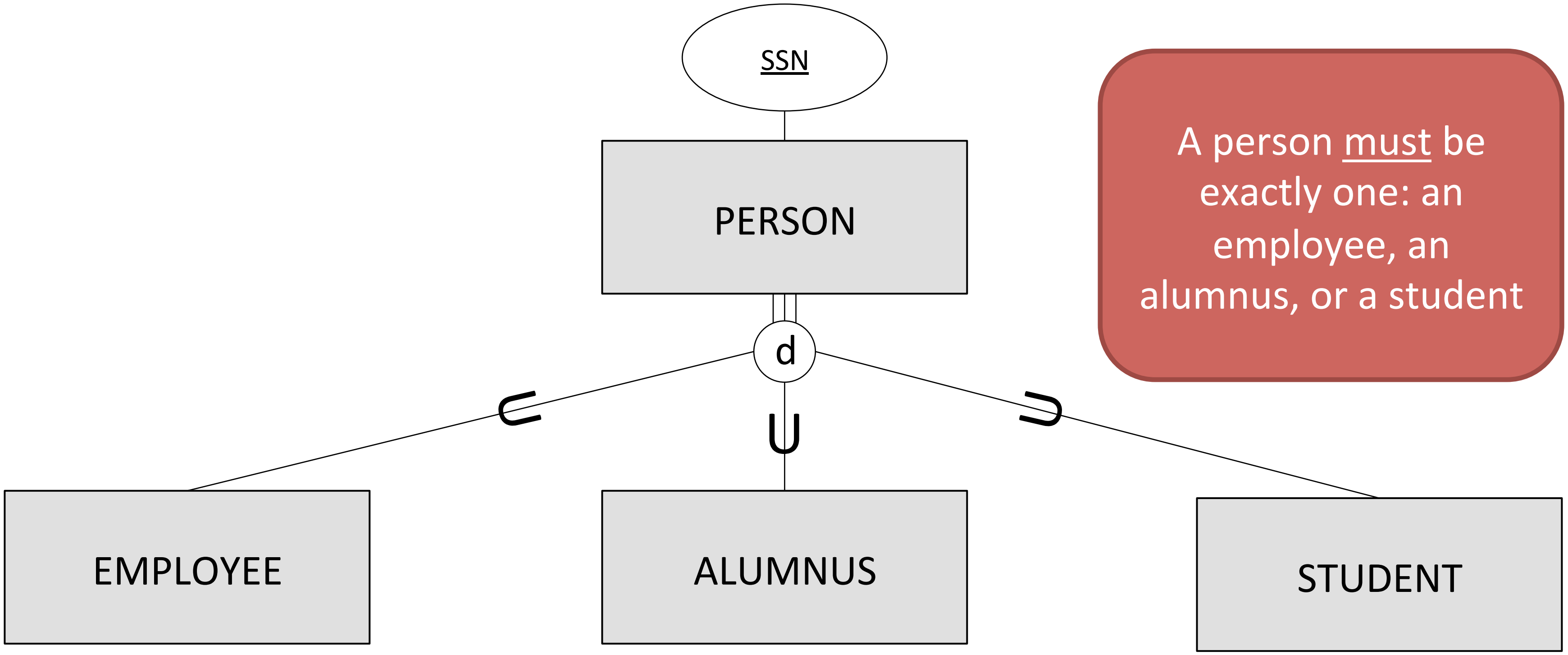
(d)isjoint: entities may *only be one* subtype



A person can be <u>either</u> an employee, an alumnus, or a student

Multiple Subtypes: Completeness

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



A person must be exactly one: an employee, an alumnus, or a student

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

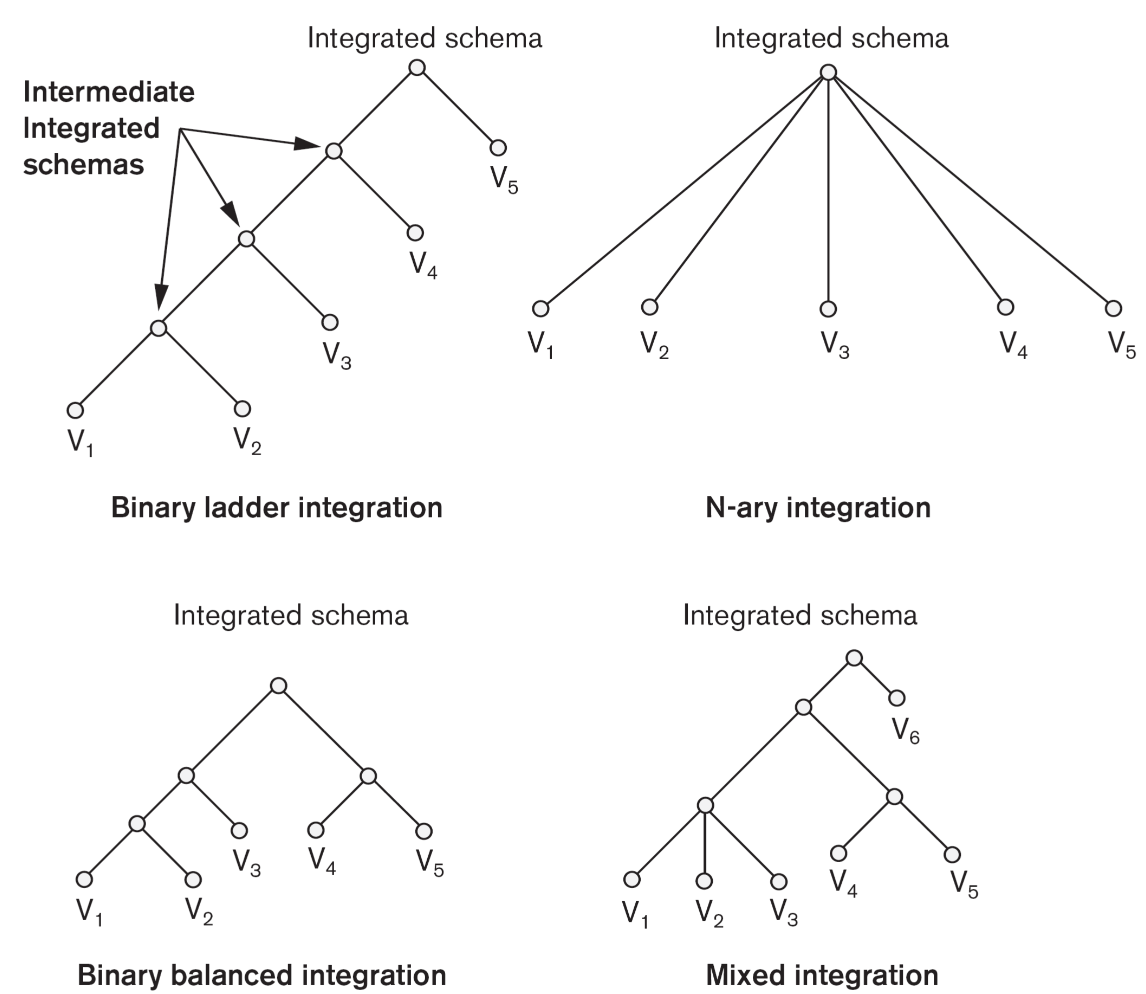
View Integration

- 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

Figure 10.6
Different strategies for the view integration process.