The Entity Relationship Model

The Entity Relationship Model (ERM)

- Forms the basis of an entity relationship diagram (ERD)
  - Conceptual database as viewed by end user
  - Relational model is a logical design of databases
- The main components of a database:
  - Entities
  - Attributes
  - Relationships

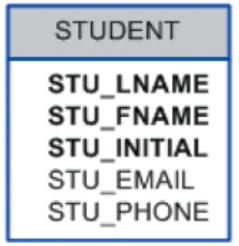
Entities

- Object of interest to the end user
  - Refers to the entity set and not to a single entity occurrence
- ERM corresponds to a table—not to a row—in the relational environment
  - ERM refers to a table row as an entity instance or entity occurrence
- In Chen, Crow's Foot, and UML notations, an entity is represented by a rectangle that contains the entity's name
  - The entity name (a noun) is usually written in all capital letters

Attributes

- Characteristics of entities:
  - **Required attribute**: must have a value and cannot be left empty
  - **Optional attribute**: does not require a value and can be left empty
  - **Attribute domain**: set of possible values for a given attribute

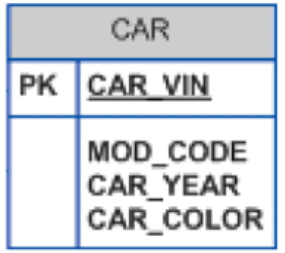
Crow's Foot Model



  - **Entity identifier**: one or more attributes that uniquely identify each entity instance (PK)
  - **Composite identifier**: primary key composed of more than one attribute

Crow's Foot Model

  - **Simple attribute**: attribute that cannot be subdivided
  - **Composite attribute**: attribute that can be subdivided to yield additional attributes



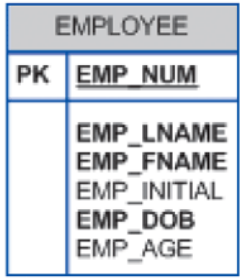
  - **Single-valued attribute**: attribute that has only a single value
  - **Multi-valued attributes**: attributes that have many values

Attributes

- Requirements of multi-valued attributes
  - Create several new attributes, one for each component of the original multivalued attribute
  - Develop a new entity composed of the original multivalued attribute's components

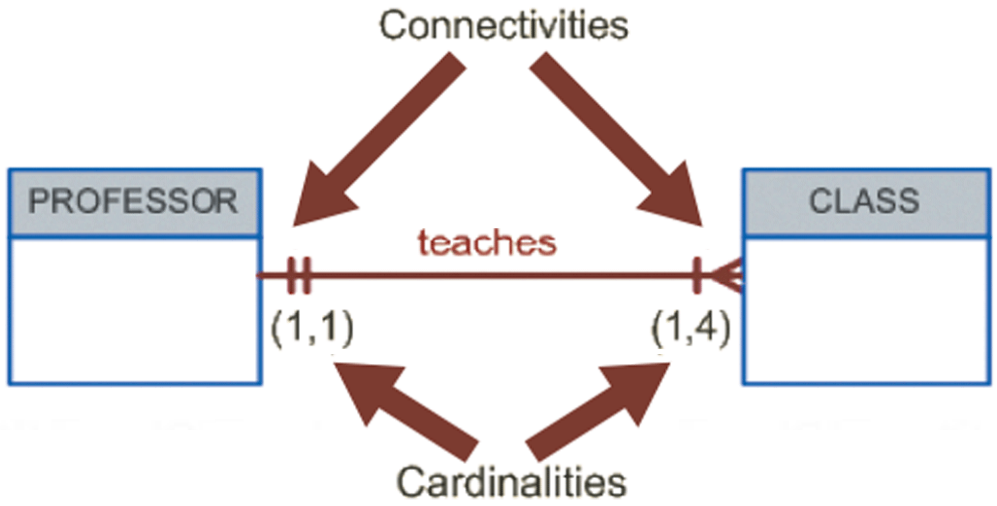
- **Derived attribute**: attribute whose value is calculated from other attributes
  - Derived using an algorithm

Crow's Foot Model



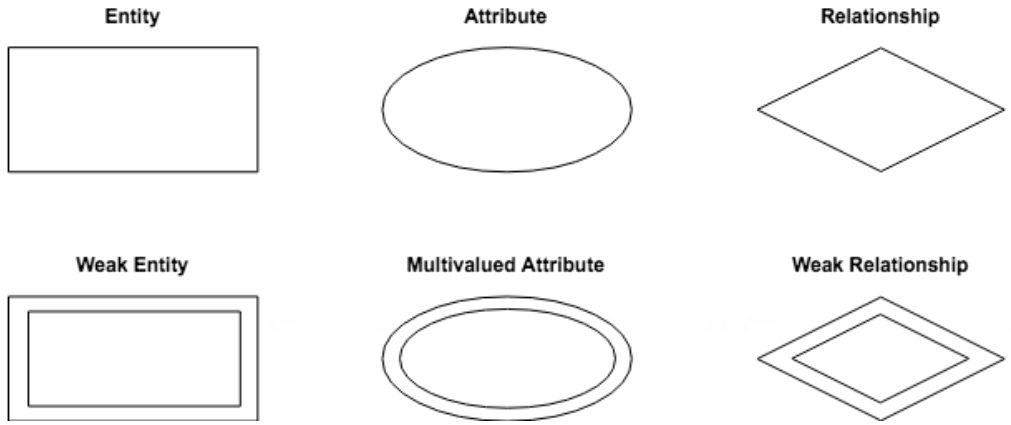
Relationships, Connectivity, and Cardinality

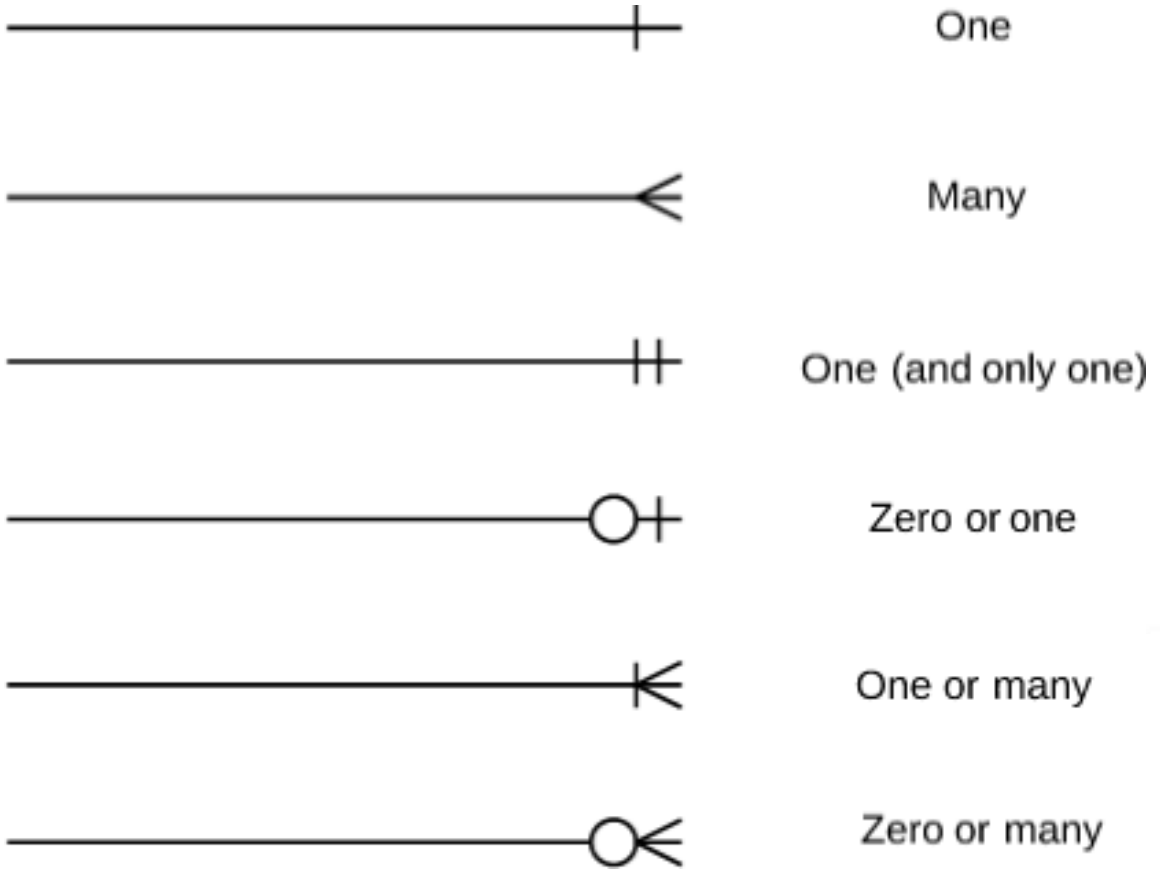
- Association between entities that always operate in both directions
  - Participants: entities that participate in a relationship
- Connectivity: describes the relationship classification
  - Include 1:1, 1:M, and M:N
- Cardinality: expresses the minimum and maximum number of entity occurrences associated with one occurrence of related entity
  - In the ERD, cardinality is indicated by placing the appropriate numbers beside the entities, using the format (x, y)

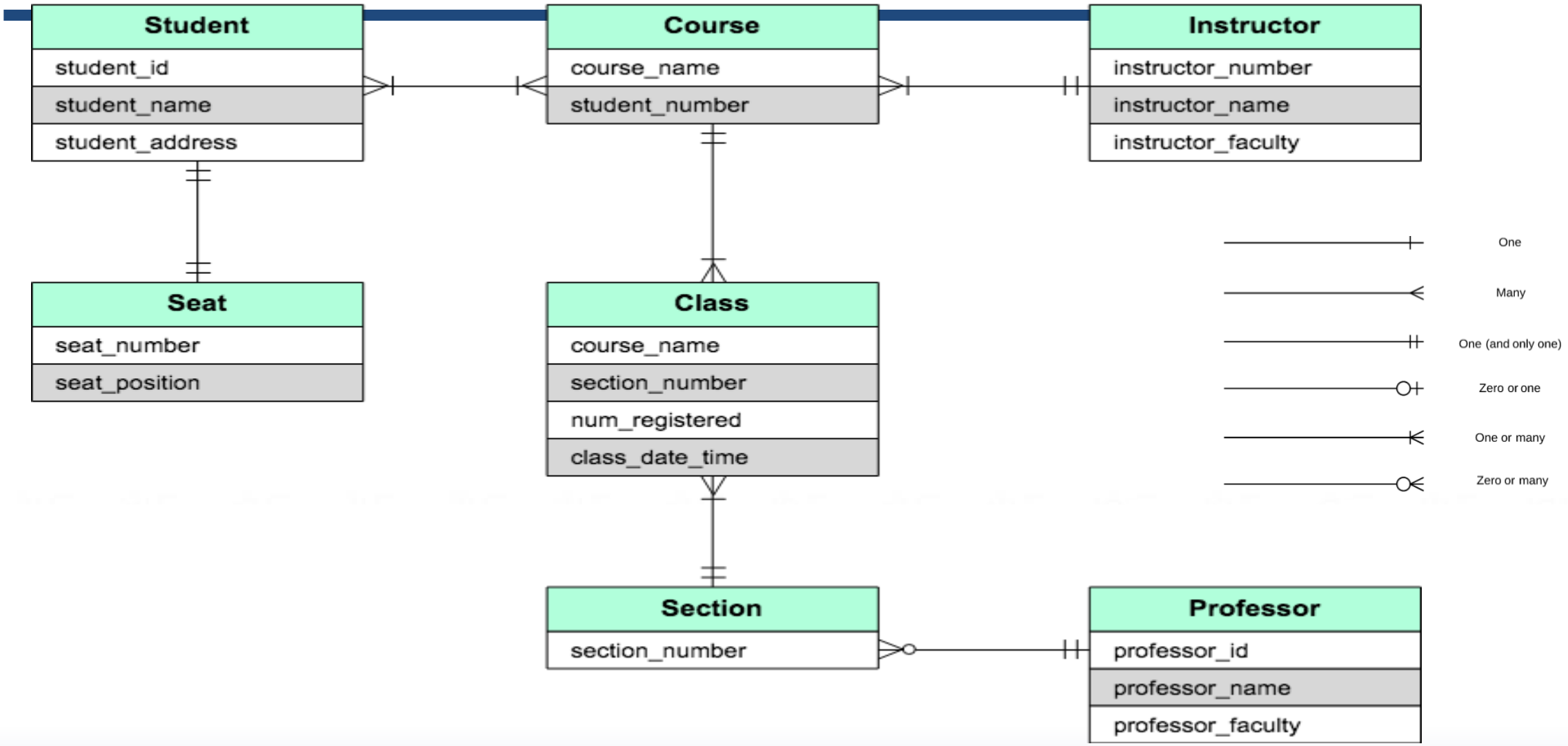


ERD Cardinality

MySQL Workbench, Visio, Lucidchart, draw.io etc

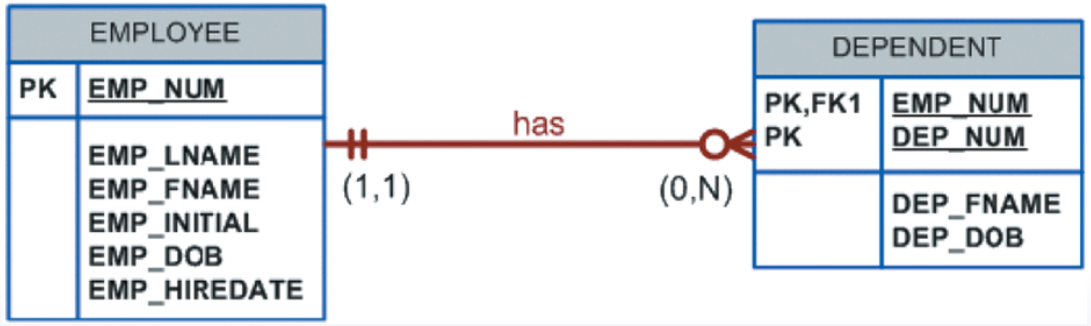





Student
student_id
student_name
student_address
Course
course_name
student_number
Instructor
instructor_number
instructor_name
instructor_faculty
Seat
seat_number
seat_position
Class
course_name
section_number
num_registered
class_date_time
Section
section_number
Professor
professor_id
professor_name
professor_faculty
One
Many
One (and only one)
Zero or one
One or many
Zero or many

Existence Dependence

- Existence independence
  - Entity exists apart from all its related entities
  - Referred to as a **strong entity** or **regular entity**
- Existence dependence
  - Entity exists in the database only when it is associated with another related entity occurrence
  - Referred to as **weak entity**
  - Has a primary key that is partially or totally derived from parent entity in the relationship
  - Database designer determines whether an entity is weak ➔ based on business rules

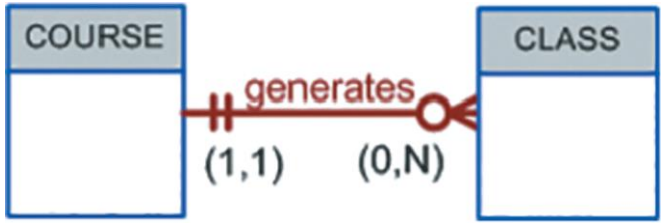


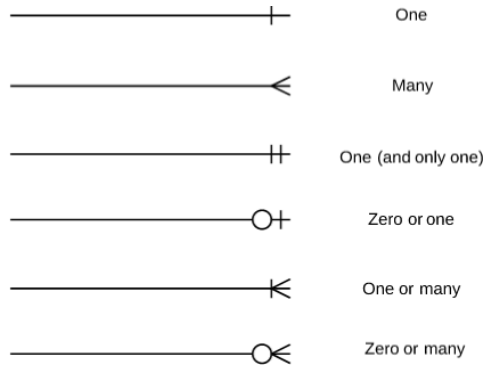
Relationship Strength

- Weak or Non-identifying relationships
  - Primary key of the related entity does not contain a primary key component of the parent entity
- Strong or Identifying relationships
  - Primary key of the related entity contains a primary key component of the parent entity

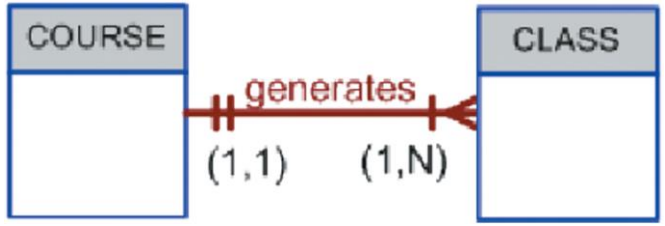
Relationship Participation

- Optional participation
  - One entity occurrence does not require a corresponding entity occurrence in a particular relationship



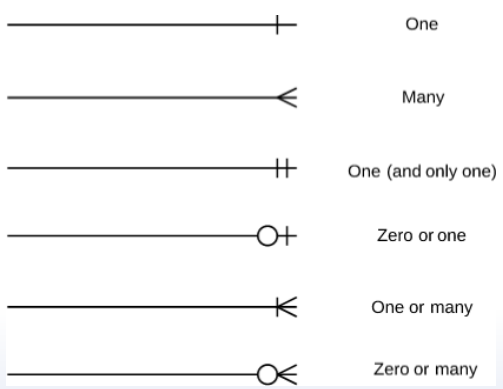


- Mandatory participation
  - One entity occurrence requires a corresponding entity occurrence in a particular relationship



Relationship Degree

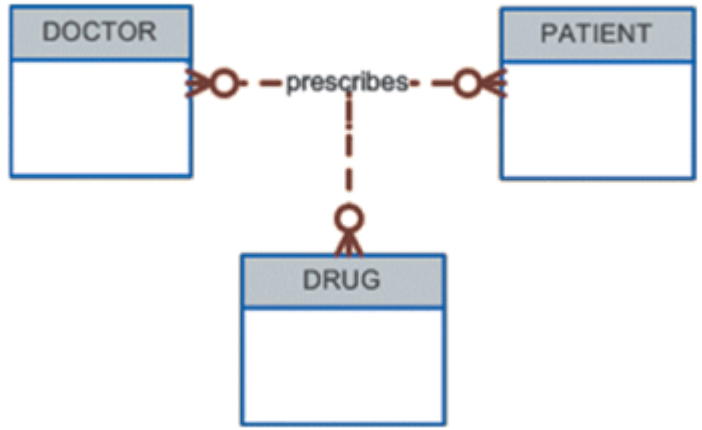
- Indicates the number of entities or participants associated with a relationship
  - Unary relationship: association is maintained within a single entity
  - Binary relationship: two entities are associated
  - Ternary relationship: three entities are associated
  - Recursive relationship: relationship exists within a single entity type



Unary relationship

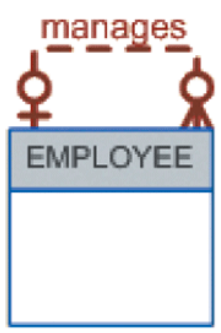
Binary relationship

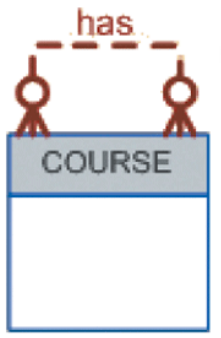
Ternary relationship (Conceptual)



Recursive Relationships

- Relationship can exist between occurrences of the same entity set
  - Naturally, such a condition is found within a unary relationship
    - Common in manufacturing industries
- One common pitfall when working with unary relationships is to confuse participation with referential integrity
  - Similar because they are both implemented through constraints on the same set of attributes





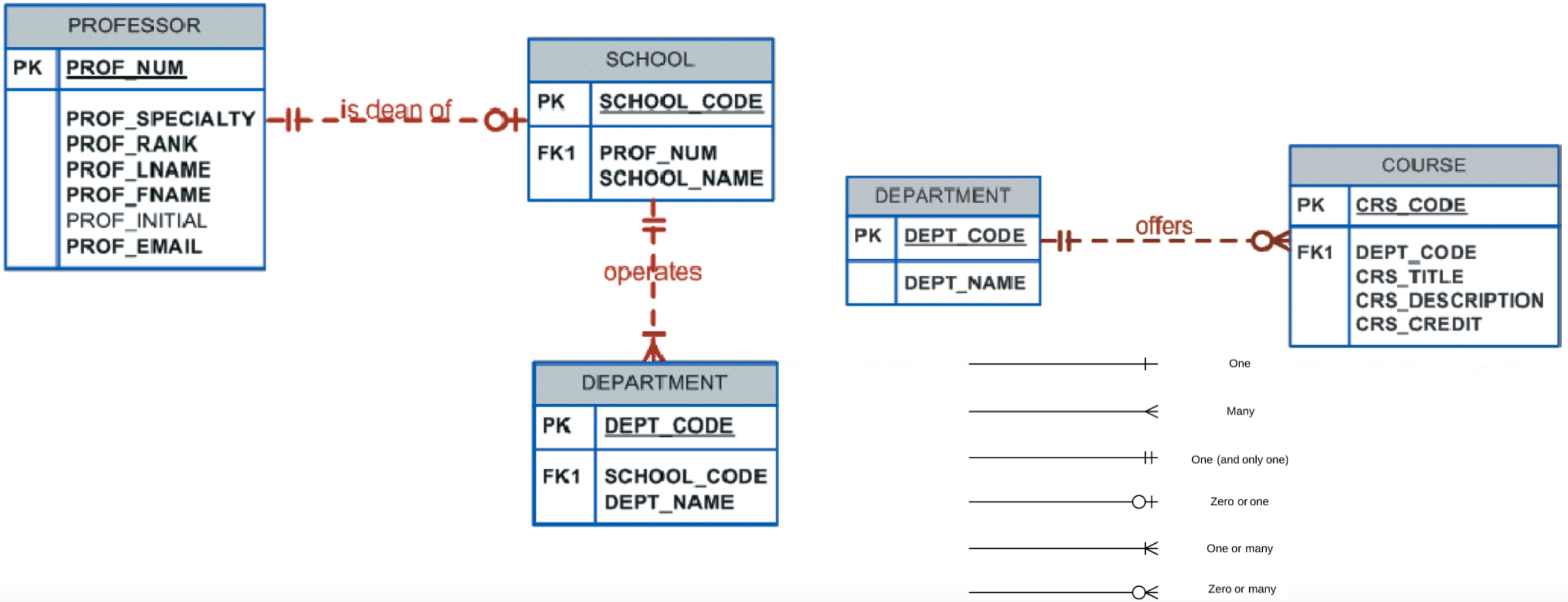
Developing an ER Diagram

- Activities involved in building and ERD
  - Create a detailed narrative of the organization's description of operations
  - Identify business rules based on the descriptions
  - Identify main entities and relationships from the business rules
  - Develop the initial ERD
  - Identify the attributes and primary keys that adequately describe entities
  - Revise and review ERD

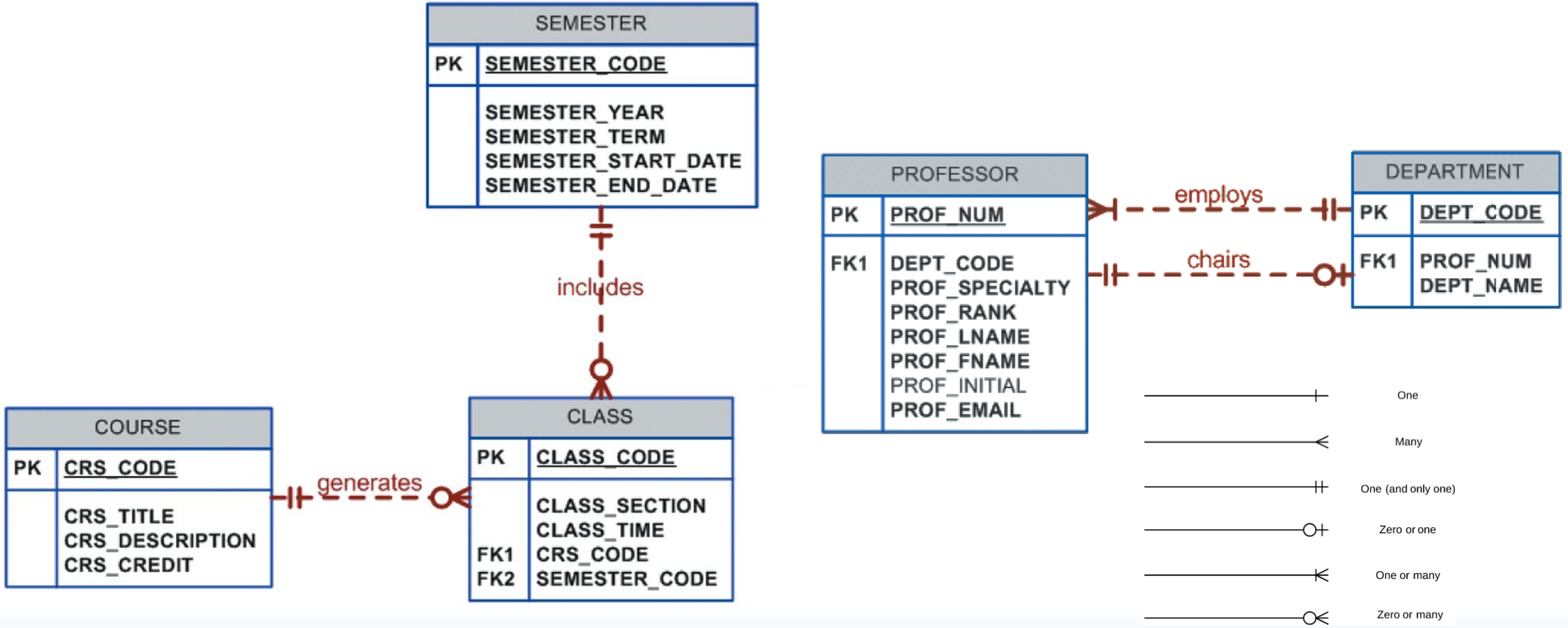
PROFESSOR	SCHOOL	DEPARTMENT
COURSE	CLASS	SEMESTER
STUDENT	BUILDING	ROOM

ENROLL (the associative entity between STUDENT and CLASS)

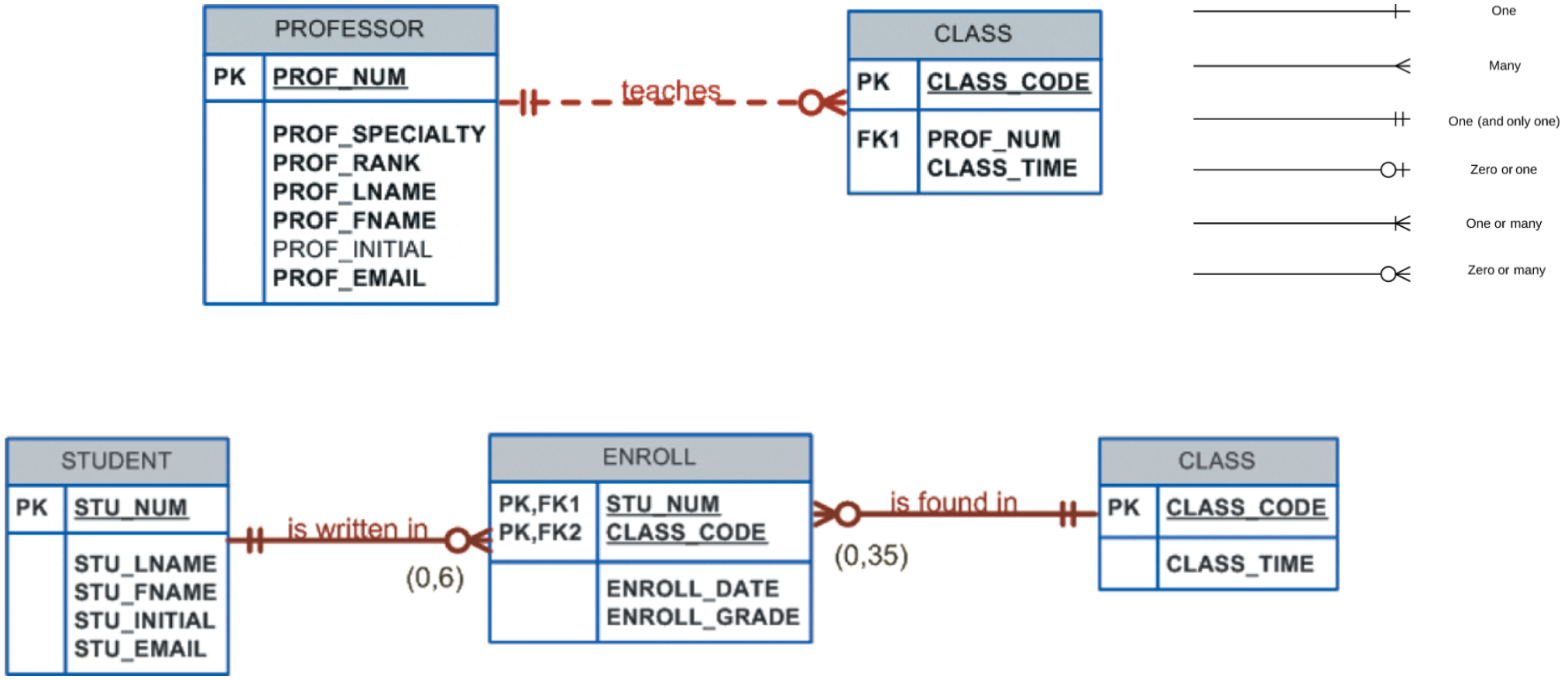
Developing an ER Diagram



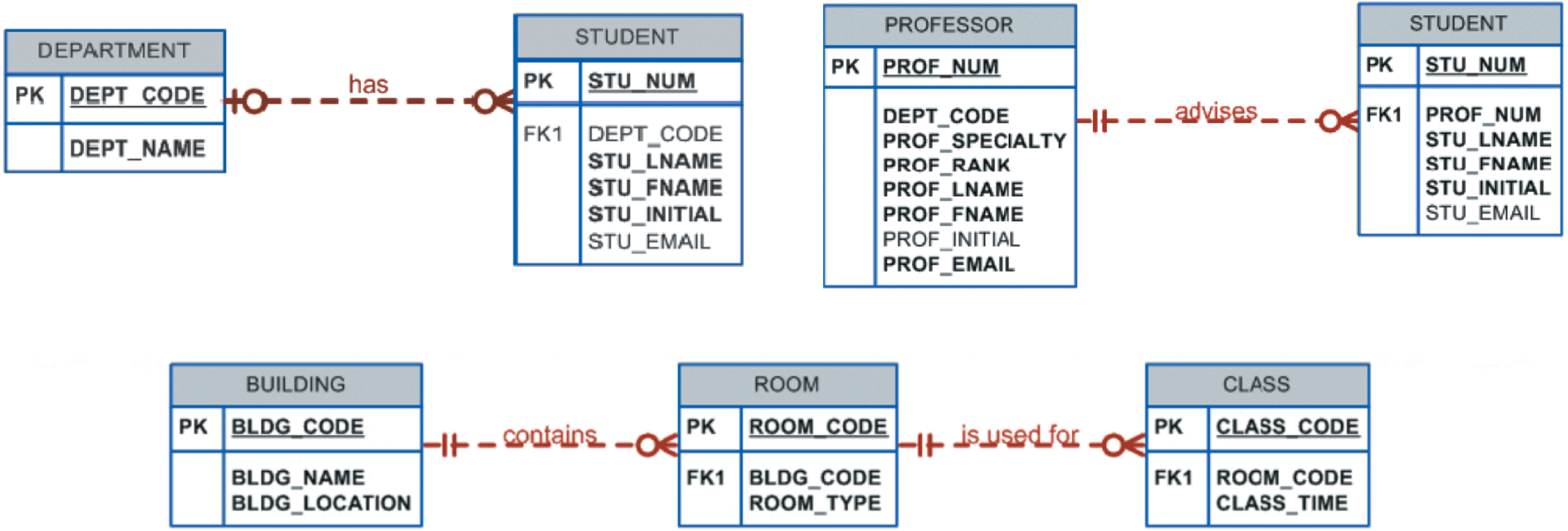
Developing an ER Diagram

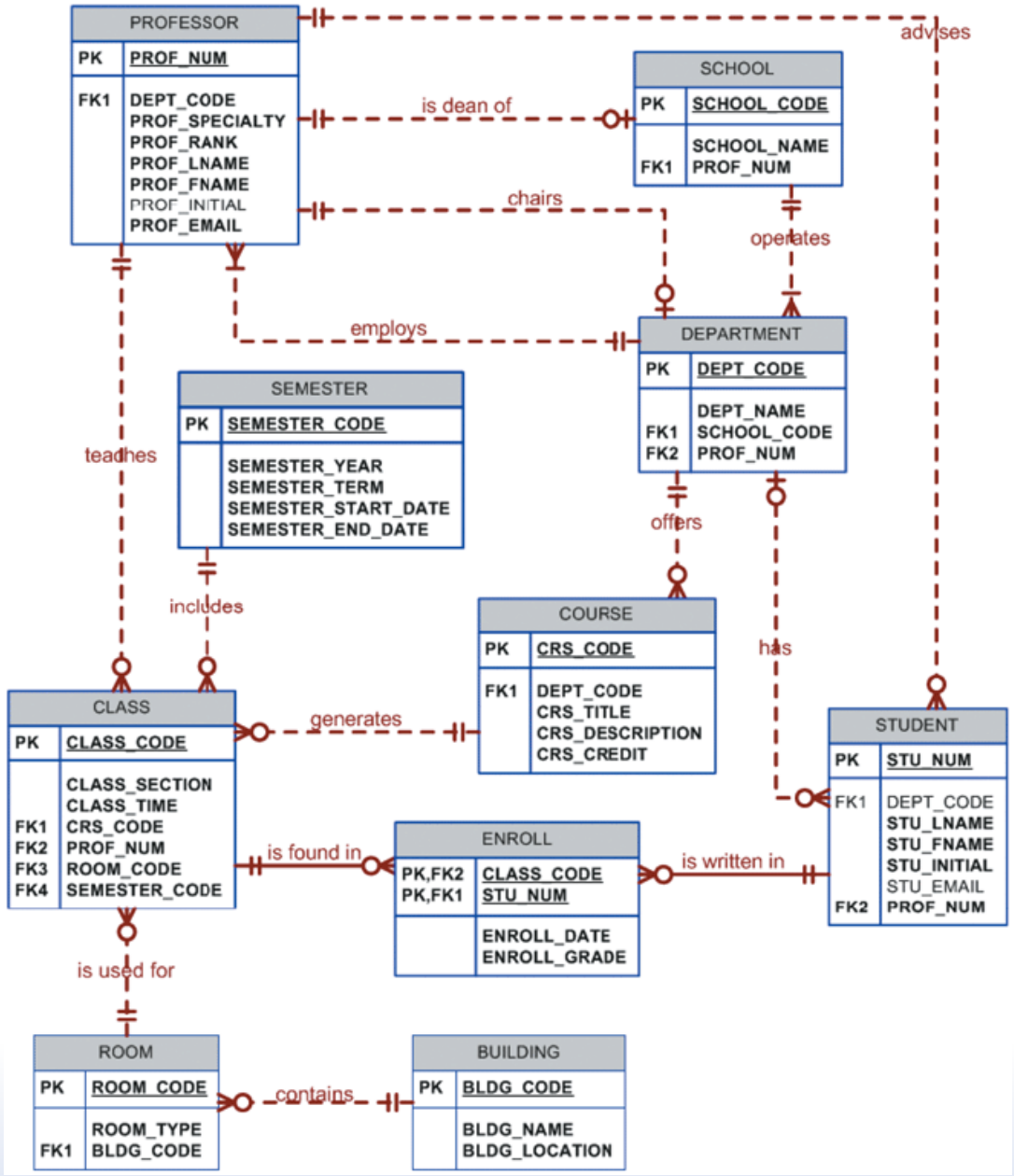


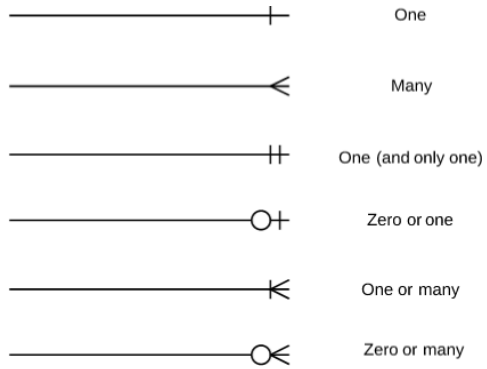
Developing an ER Diagram



Developing an ER Diagram




PROFESSOR
PK PROF_NUM
FK1 DEPT_CODE
PROF_SPECIALTY
PROF_RANK
PROF_LNAME
PROF_FNAME
PROF_INITIAL
PROF_EMAIL
SCHOOL
PK SCHOOL_CODE
FK1 SCHOOL_NAME
PROF_NUM
DEPARTMENT
PK DEPT_CODE
FK1 DEPT_NAME
FK2 SCHOOL_CODE
PROF_NUM
SEMESTER
PK SEMESTER_CODE
SEMESTER_YEAR
SEMESTER_TERM
SEMESTER_START_DATE
SEMESTER_END_DATE
COURSE
PK CRS_CODE
FK1 DEPT_CODE
CRS_TITLE
CRS_DESCRIPTION
CRS_CREDIT
CLASS
PK CLASS_CODE
CLASS_SECTION
CLASS_TIME
FK1 CRS_CODE
FK2 PROF_NUM
FK3 ROOM_CODE
FK4 SEMESTER_CODE
ENROLL
PK,FK2 CLASS_CODE
PK,FK1 STU_NUM
ENROLL_DATE
ENROLL_GRADE
STUDENT
PK STU_NUM
FK1 DEPT_CODE
STU_LNAME
STU_FNAME
STU_INITIAL
STU_EMAIL
FK2 PROF_NUM
ROOM
PK ROOM_CODE
ROOM_TYPE
FK1 BLDG_CODE
BUILDING
PK BLDG_CODE
BLDG_NAME
BLDG_LOCATION
advises
is dean of
chairs
operates
employs
teaches
includes
offers
has
generates
is found in
is written in
is used for
contains


One
Many
One (and only one)
Zero or one
One or many
Zero or many

Database Design Challenges: Conflicting Goals

- Database designers must often make design compromises that are triggered by conflicting goals
  - Database design must conform to **design standards**
  - **High processing speed** may limit the number and complexity of logically desirable relationships
  - **Complex information requirements** (the extraction, transformation, and loading (ETL) process) may dictate data transformations, and they may expand the number of entities and attributes within the design.

Developing an ER Diagram (Review again)

- Activities involved in building and ERD
  - Create a detailed narrative of the organization's description of operations
  - Identify business rules based on the descriptions
  - Identify main entities and relationships from the business rules
  - Develop the initial ERD
  - Identify the attributes and primary keys that adequately describe entities
  - Revise and review ERD

PROFESSOR	SCHOOL	DEPARTMENT
COURSE	CLASS	SEMESTER
STUDENT	BUILDING	ROOM
ENROLL (the associative entity between STUDENT and CLASS)		

Modeling Relationships (ER to Tables)

- An entity type is usually modeled as a relation.
- Relationships are usually modeled by having the tuples of one relation point to the tuples of another relation.
- One-to-one, one-to-many, and many-to-one relationships between two entities (relations) can be modeled directly with foreign keys.
- Many-to-many relationships are more of a problem.
  - Must introduce a new relation to model the many-to-many relationship.
  - This relation has foreign keys pointing to tuples in the associated relations (entities).

ER Diagram Common Errors

- An ER diagram is not a flow chart. An ER diagram trys to capture structure of data but not the decision logic behind a set of tasks.
- Every ER diagram contains at least one entity and some attributes. If relationships exist, relationship cardinalities should be modeled.
- Weak entity type must be related to at least one non-weak entity type. Be careful about the relationship cardinalities for such relationship.
- All entities must have keys, full or partial.

Practice Question 1

- Professors have an SSN, a name, an age, a rank, and a research specialty.
- Projects have a project number, a sponsor name (e.g., NSF), a starting date, an ending date, and a budget.
- Graduate students have an SSN, a name, an age, and a degree program (e.g., M.S.or Ph.D.).
- Each project is managed by one professor (known as the project's principal investigator).
- Each project is worked on by one or more professors (known as the project's co-investigators).
- Professors can manage and/or work on multiple projects.
- Each project is worked on by one or more graduate students (known as the project's research assistants).
- When graduate students work on a project, a professor must supervise their work on the project. Graduate students can work on multiple projects, in which case they will have a (potentially different) supervisor for each one.
- Departments have a department number, a department name, and a main office.
- Departments have a professor (known as the chairman) who runs the department.
- Professors work in one or more departments, and for each department that they work in, a time percentage is associated with their job.
- Graduate students have one major department in which they are working on their degree.
- Each graduate student has another, more senior graduate student (known as a student advisor) who advises him or her on what courses to take

Practice Question 2

- Each bank can have multiple branches, and each branch can have multiple accounts and loans.
- Each bank has a unique number, name and address. Each branch as address and branch number.
- Each customer has account(s) in a bank branch and may get loan(s).
- Each account has a unique account number, type, and balance.
- Each loan has a unique loan number, type and amount.
- Each customer has SSN, Name, phone, address.

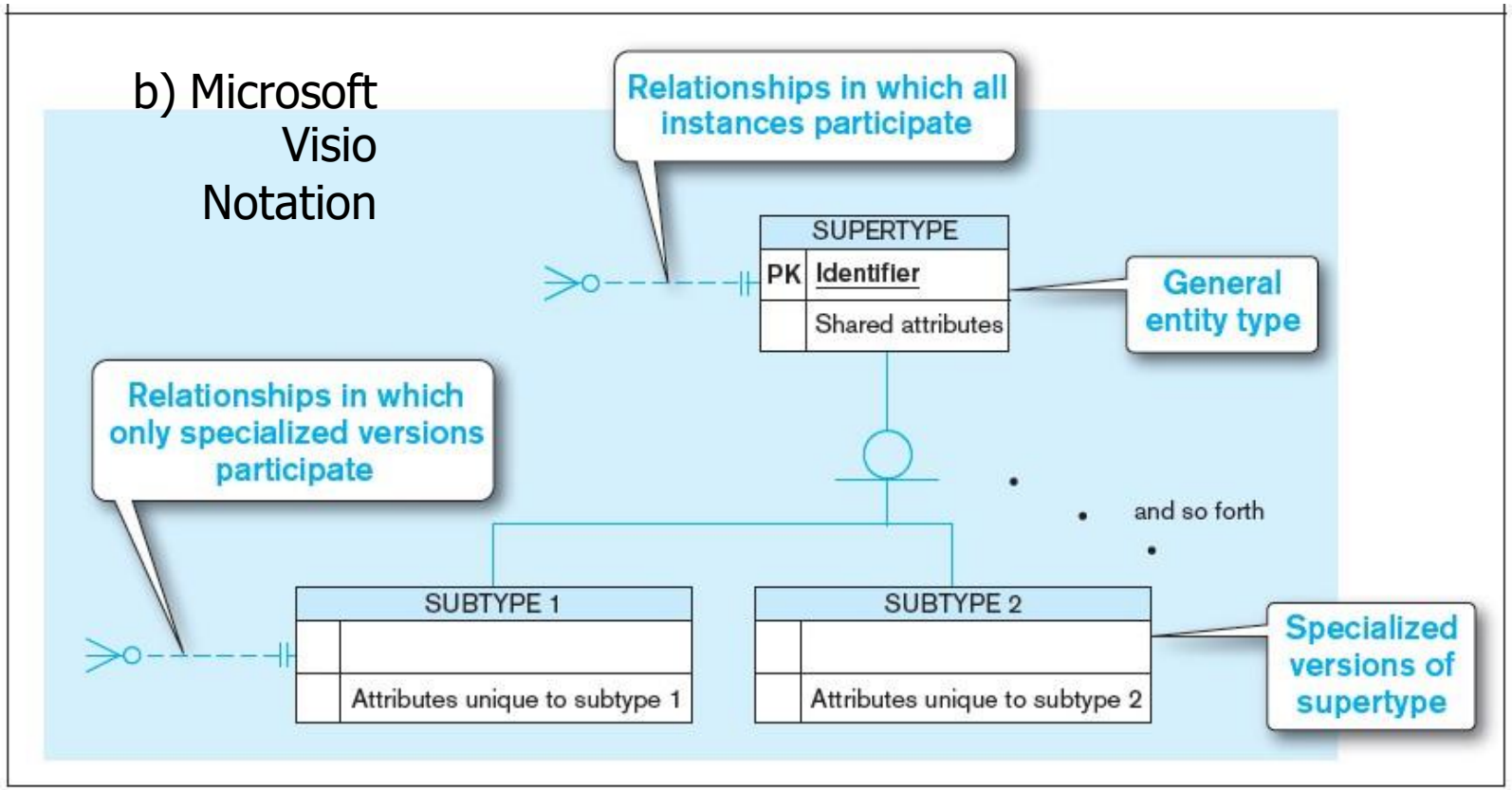
Enhanced ER Model

- In addition to ER model, EER includes the concepts of subclass and superclass
- Subclass: Specialization (top down)
  Generalization: (bottom up)
- Ex) Employee are grouped into secretary, engineer, manger, and technician
  These subgrouping is called subclass of the employee entity type
- Attribute inheritance: a member of a subclass inherits all the attributes of the entity
- Specialization: process of defining a set of subclasses of an entity type.

Specialization

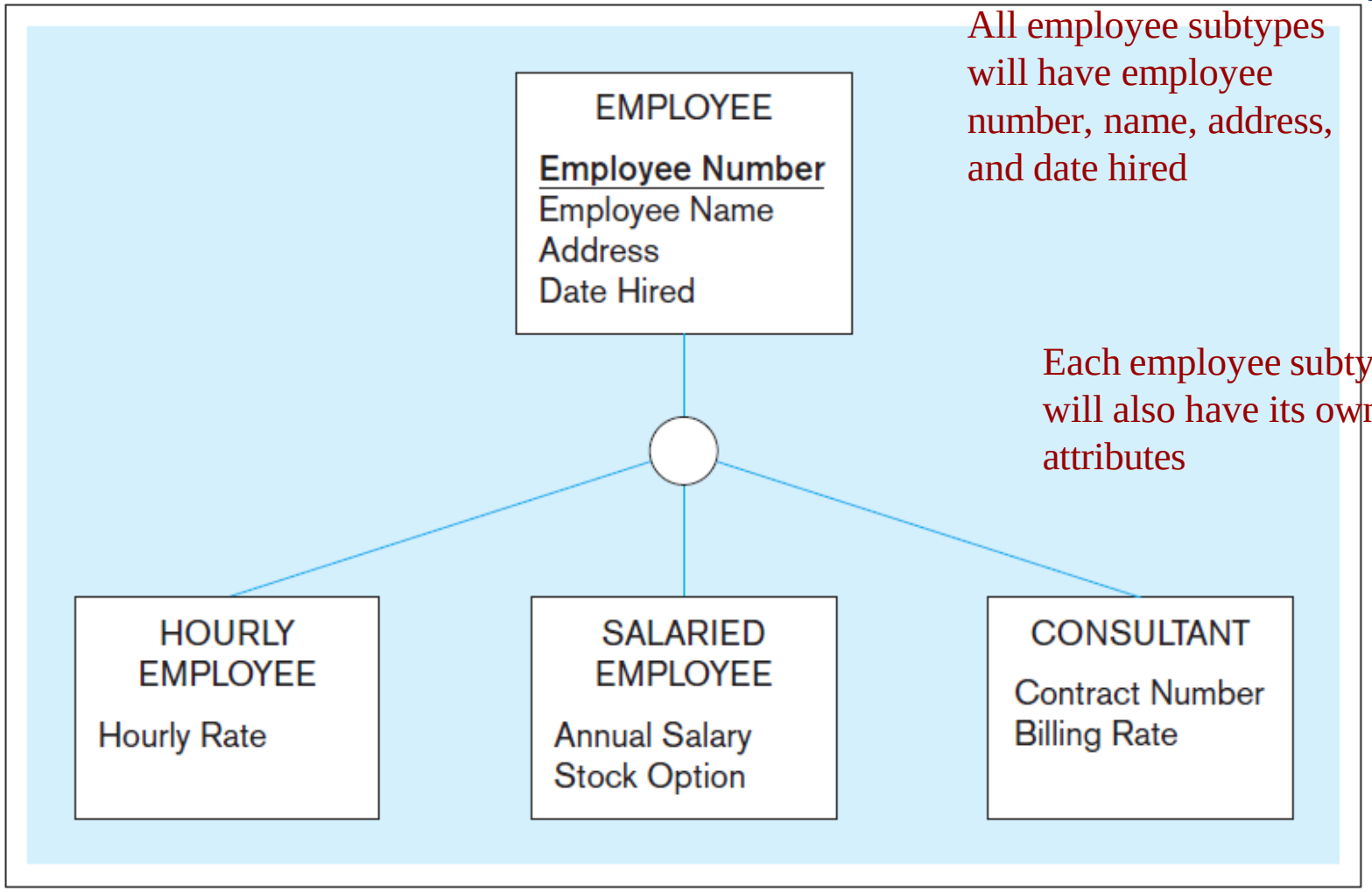
- Ex) {Secretary, engineer} : specialization on the job type.
- Two constraints may apply to specialization
  - disjointness constraints (d) : subclasses of the specialization must be disjoint
  - Otherwise (o) overlapped.
  - Completeness constraints: either total or partial
- Generalization is a reverse process of abstraction in which identify common features and generalize into a superclasses.

Basic notation for supertype/subtype notation



Different modeling tools may have different notation for the same modeling constructs.

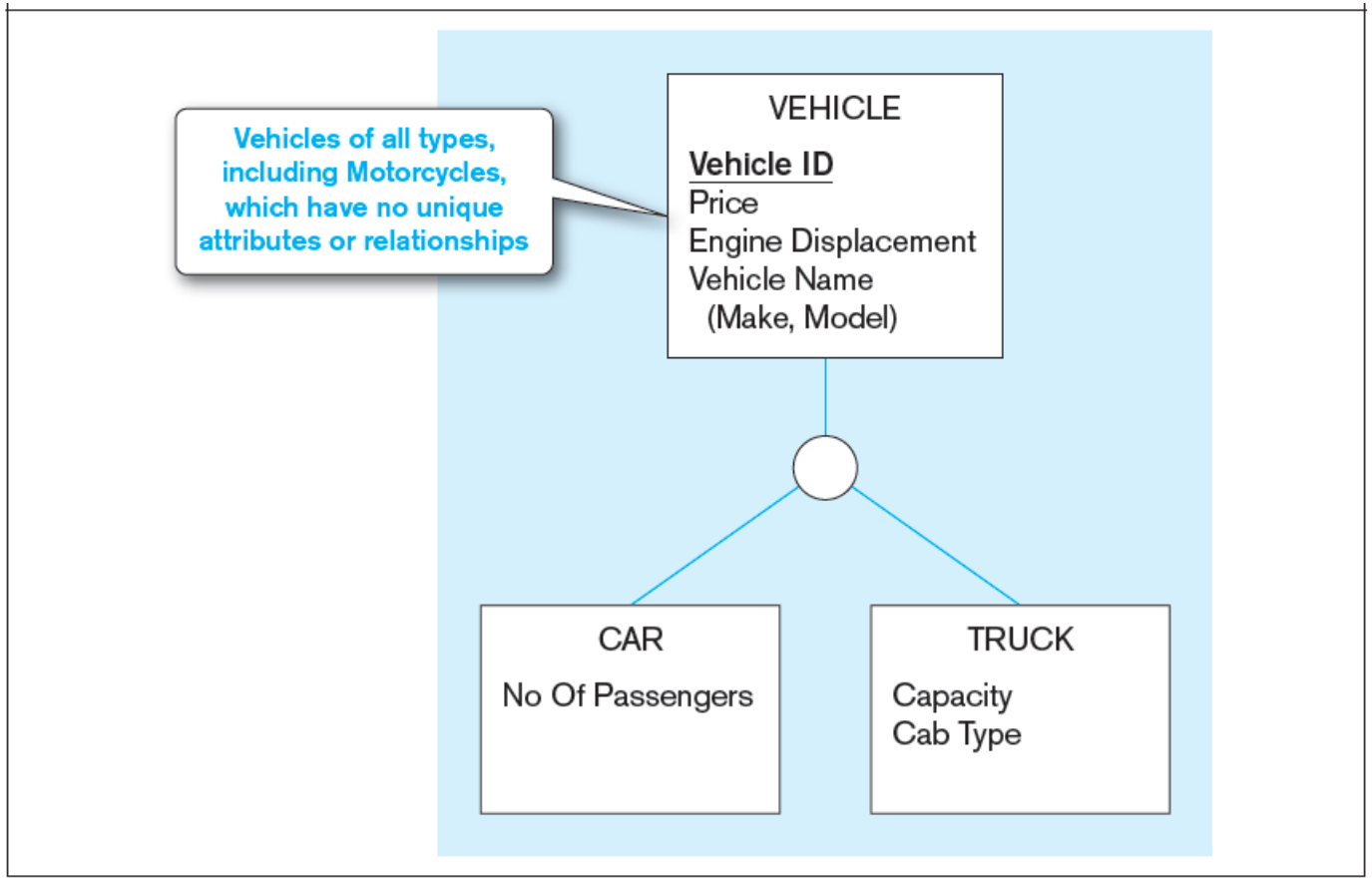
Employee supertype with subtypes



All employee subtypes will have employee number, name, address, and date hired

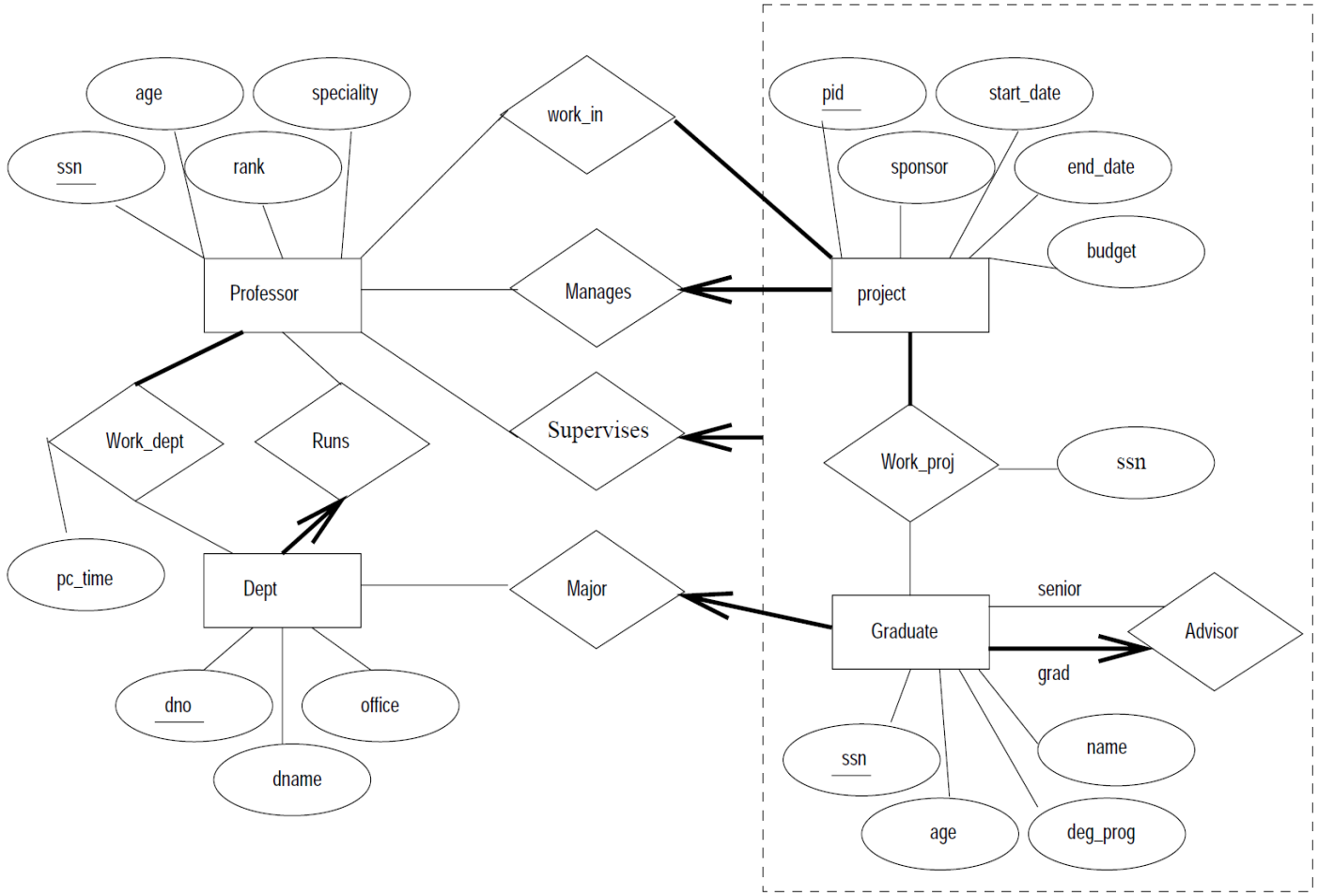
Each employee subtype will also have its own attributes

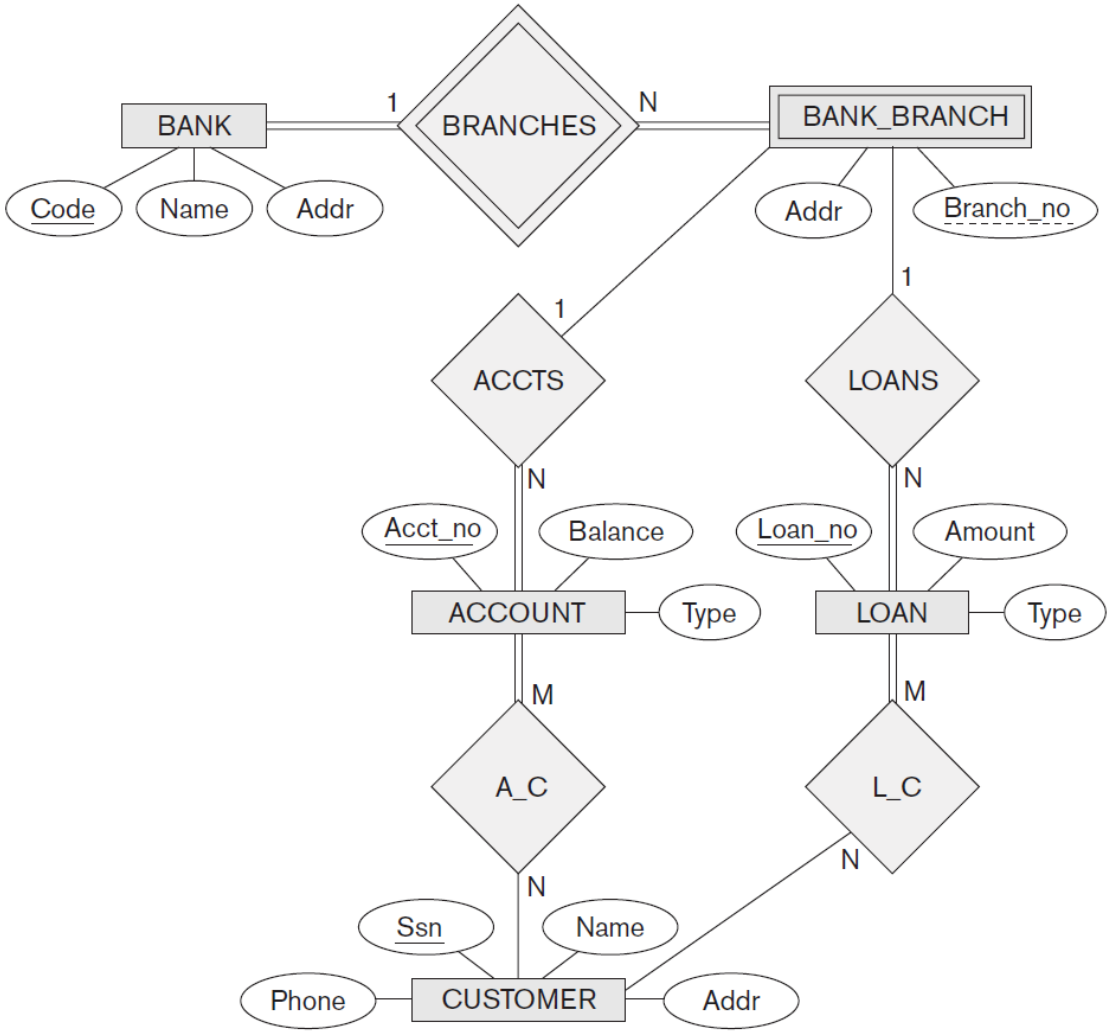
Generalization to VEHICLE supertype



So we put the shared attributes in a supertype

Note: no subtype for motorcycle, since it has no unique attributes


age
speciality
ssn
rank
work_in
pid
start_date
sponsor
end_date
budget
Professor
Manages
project
Work_dept
Runs
Supervises
Work_proj
ssn
pc_time
Dept
Major
senior
Graduate
Advisor
grad
dno
office
dname
ssn
name
age
deg_prog


BANK
1
BRANCHES
N
BANK_BRANCH
Code
Name
Addr
Addr
Branch_no
1
1
ACCTS
LOANS
N
N
Acct_no
Balance
Loan_no
Amount
ACCOUNT
Type
LOAN
Type
M
M
A_C
L_C
N
N
Ssn
Name
Phone
CUSTOMER
Addr