Developing ERD

Lecture 5

Learning Outcomes

- In this chapter, you will learn:
  - The characteristics of entity relationship components
  - How relationships between entities are defined, refined, and incorporated into the database design process
  - How ERD components affect database design and implementation

Developing an ER Diagram

- The process of database design is an ***iterative*** rather than a linear or sequential process. The process is repeated until the end users and designers agree that the ER diagram.
- It usually begins with a general narrative of the organization's operations and procedures.

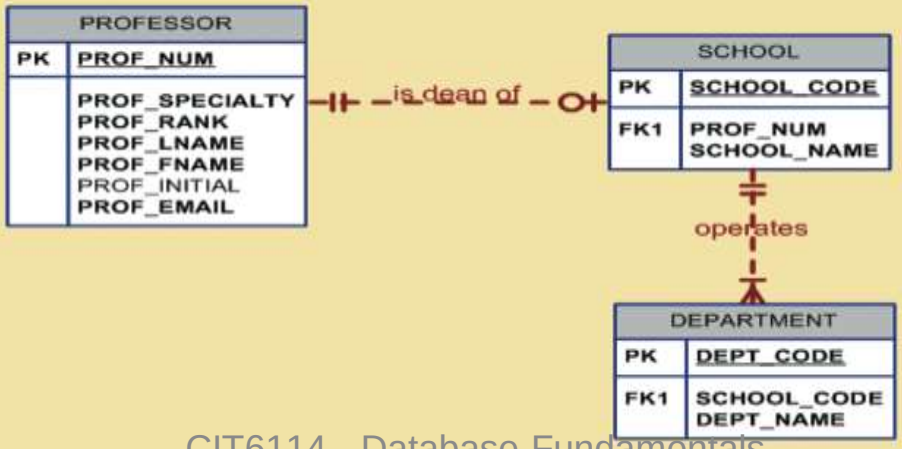
Developing an ER Diagram

- Building an ERD usually involves the following activities:
  - Create detailed narrative of organization's description of **operations**
  - Identify **business rules** based on description of operations
  - Identify main **entities** and **relationships** from business rules
  - **Develop** initial ERD
  - Identify **attributes** and **primary keys** that adequately describe entities
  - **Revise** and **review** ERD

Developing an ER Diagram

- Tiny College Database
  - Each school operates several departments.
  - The smallest number of departments operated by a school is one, and the largest number of departments is indeterminate (N).
  - Each department belongs to *only* a single school.

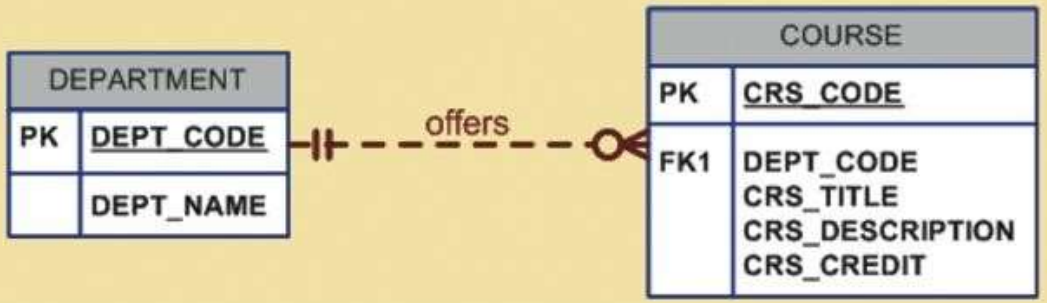
FIGURE 4.26 The first Tiny College ERD segment



Developing an ER Diagram

- Tiny College Database
  - Each department may offer several courses.

FIGURE 4.27 The second Tiny College ERD segment



Developing an ER Diagram

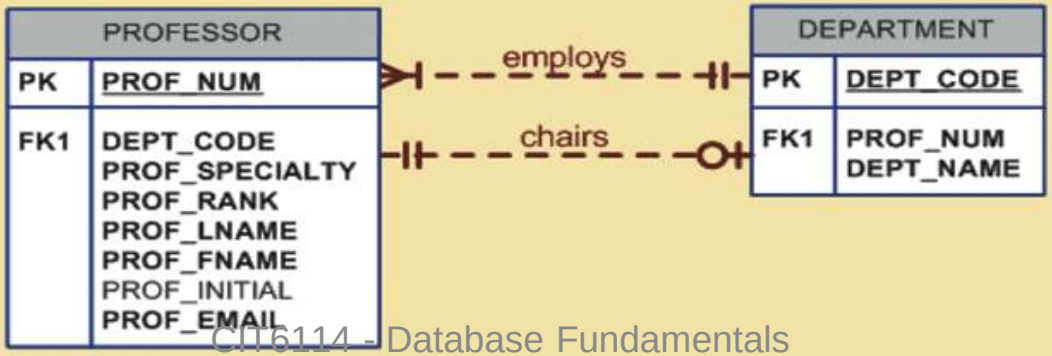
- Tiny College Database
  - A department may offer several classes of the same course.
  - A 1:M relationship exists between COURSE and CLASS.
  - CLASS is *optional* to COURSE

FIGURE 4.28 The third Tiny College ERD segment

Developing an ER Diagram

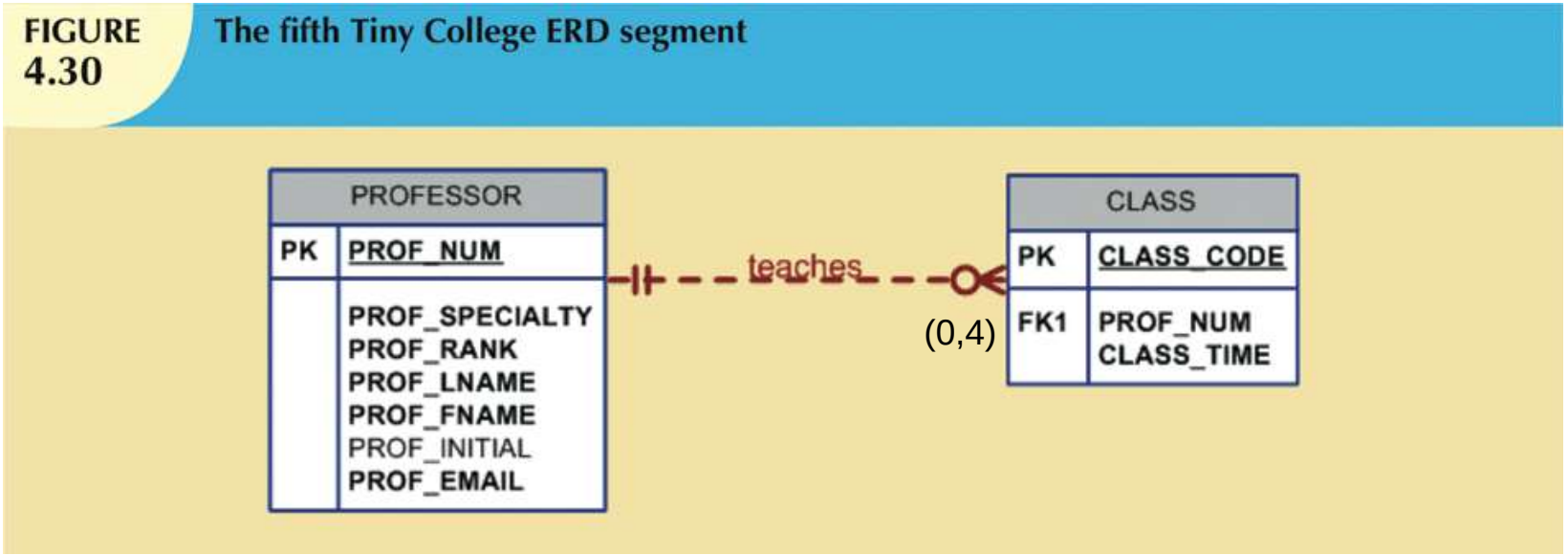
- Tiny College Database
  - Each department employs many professors.
  - One of those professors may chair the department. Only one of the professors can chair the department.
    - DEPARTMENT is optional to PROFESSOR in the "chairs" relationship.

FIGURE 4.29 The fourth Tiny College ERD segment



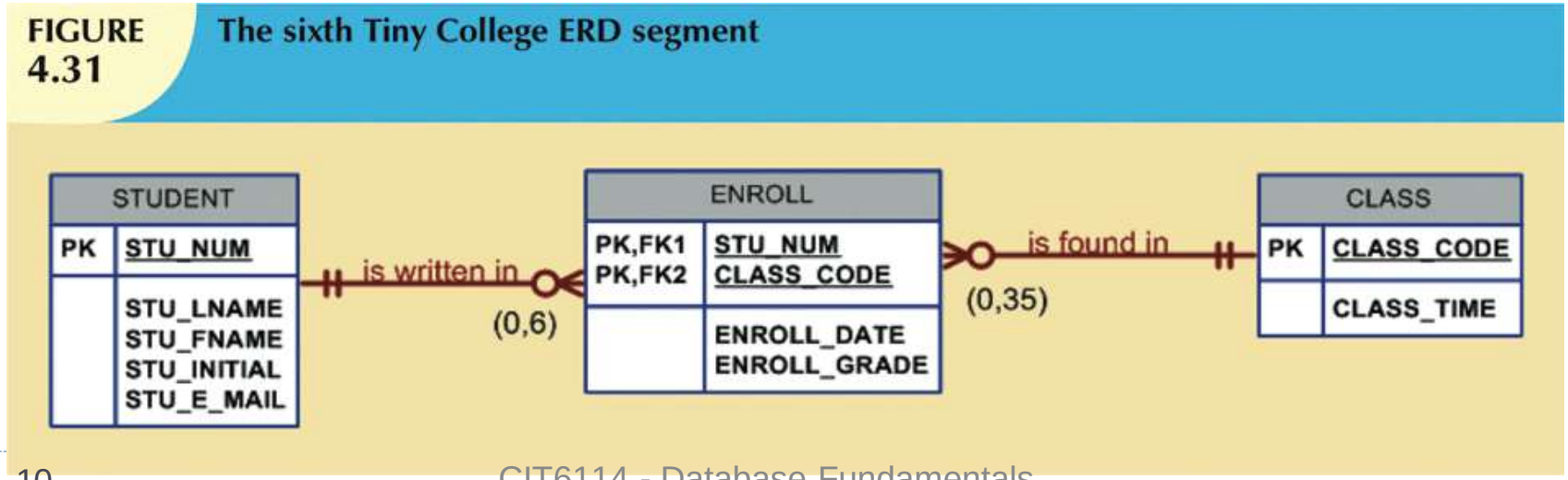
Developing an ER Diagram

- Tiny College Database
  - Each professor may teach up to four classes.
  - A professor may also be on a research contract and teach no classes.



Developing an ER Diagram

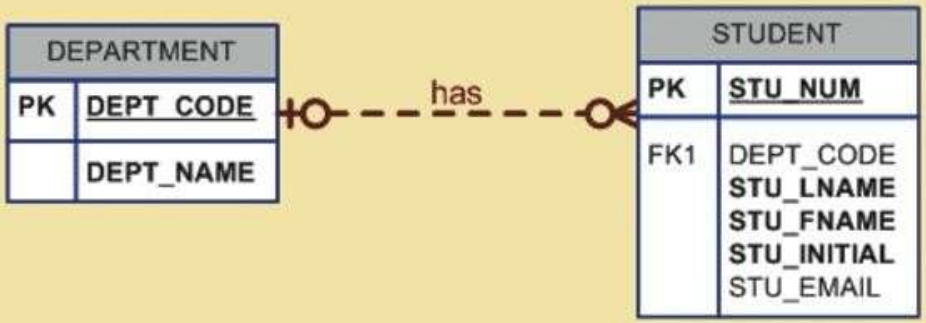
- Tiny College Database
  - A student may enroll in several classes. Each student may enroll in up to six classes.
  - Each class may have up to 35 students in it.
  - STUDENT is optional to CLASS (class may exist without student enrolling it).



Developing an ER Diagram

- Tiny College Database
  - Each department may have several students whose major is offered by that department.
  - Each student may have a single major and associated with a single department.

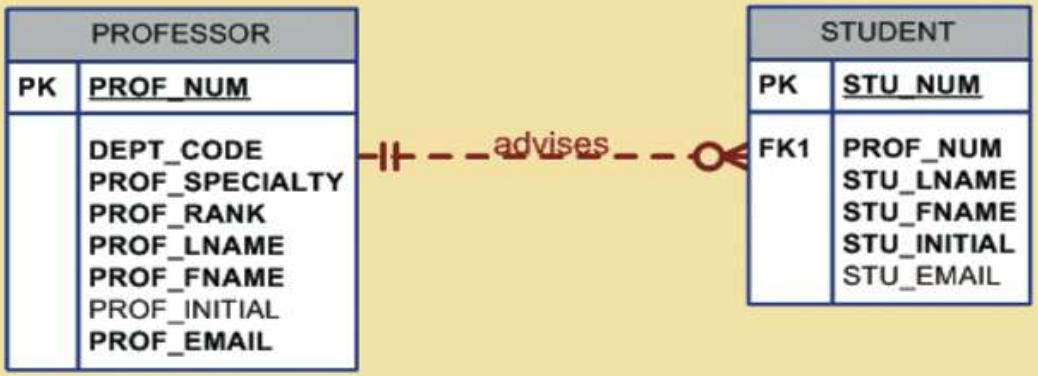
FIGURE 4.32 The seventh Tiny College ERD segment



Developing an ER Diagram

- Tiny College Database
  - Each student has an advisor in his or her department; each advisor counsels several students.
  - An advisor is also a professor, but not all professors advise students.

FIGURE 4.33 The eighth Tiny College ERD segment



Mapping ER Model to Relational (Internal) Model

- General Rules Governing Relationships among Tables
  - All primary keys must be defined as **NOT NULL**.
  - Define all foreign keys to conform to the following requirements for binary relationships.
    - 1:M Relationship
    - M:N Relationship
    - 1:1 Relationship

Mapping ER Model to Relational (Internal) Model

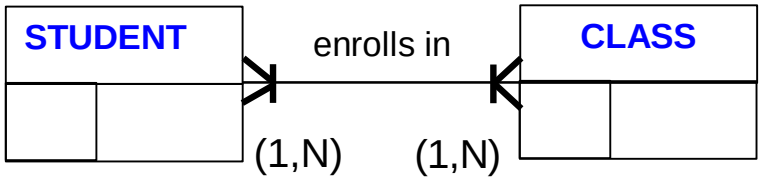
- **1:M Relationships**
  - Create the foreign key by putting the primary key on the "many" side.

- **M:N Relationship**
  - Convert the M:N relationship to a composite (bridge) entity consisting of (at least) the parent tables' primary keys.

Mapping ER Model to Relational (Logical) Model

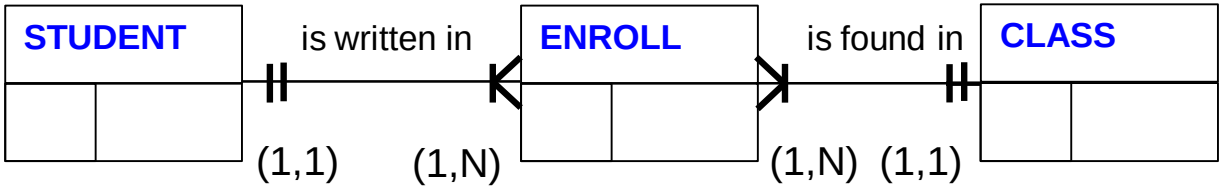
- **1:1 Relationships**
  - If both entities are in mandatory participation in the relationship and they do not participate in other relationships, it is most likely that the two entities should be part of the same entity.

CASE 1: M:N, Both Sides MANDATORY



The revised (implementable) two sets of relationships:

- 1:M relationship between STUDENT and ENROLL
- 1:M relationship between CLASS and ENROLL

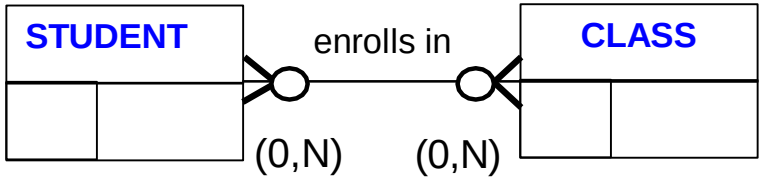


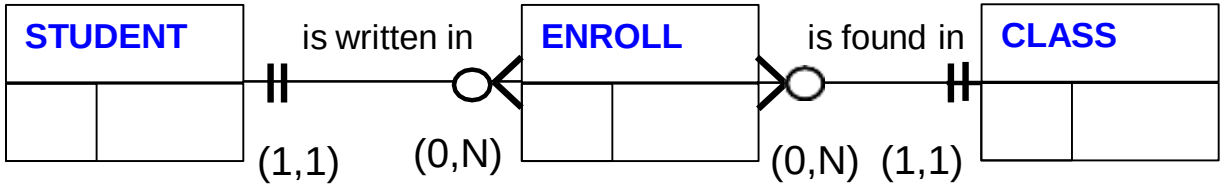
STUDENT (STU_NUM, STU_LNAME)

ENROLL (CLASS_CODE, STU_NUM, GRADE)

CLASS (CLASS_CODE, CLASS_SEC, CLASS_TIME, ROOM_CODE, PROF_NUM)

CASE 2: M:N, Both Sides OPTIONAL





Note: Neither STUDENT nor CLASS needs to appear in ENROLL

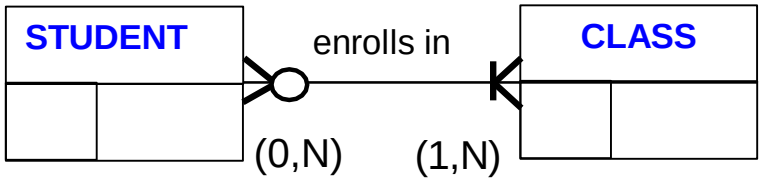
STUDENT (STU_NUM, STU_LNAME)

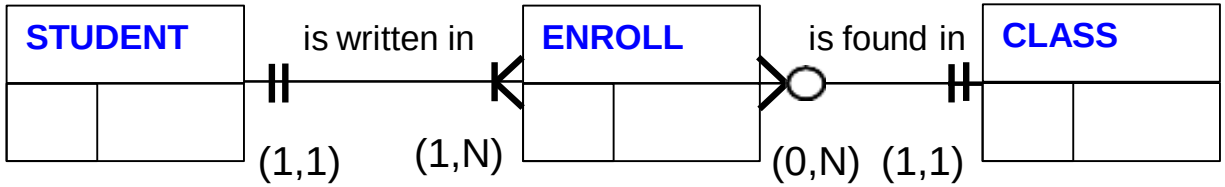
ENROLL (CLASS_CODE, STU_NUM, GRADE)

CLASS (CLASS_CODE, CLASS_SEC, CLASS_TIME, ROOM_CODE, PROF_NUM)

CASE 3: M:N, One Side OPTIONAL

Note: STUDENT is optional to CLASS, i.e., a class may not (yet) have students in it





Because it is possible that no student has signed up for a class, it is possible for a class not to show up in the ENROLL table

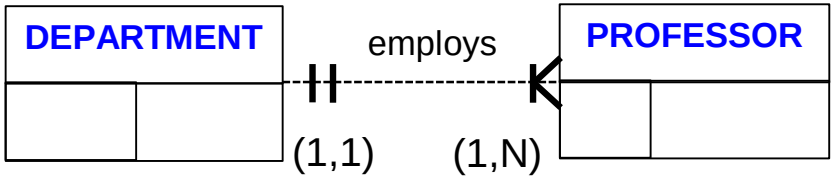
STUDENT (STU_NUM, STU_LNAME)

ENROLL (CLASS_CODE, STU_NUM, GRADE)

CLASS (CLASS_CODE, CLASS_SEC, CLASS_TIME, ROOM_CODE, PROF_NUM)

CASE 4: 1:M, Both Sides MANDATORY

Foreign key always on the “many” side

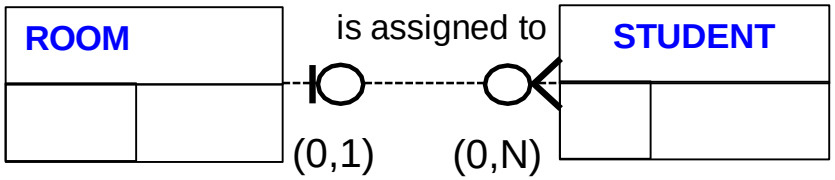


DEPARTMENT (DEPT_CODE, DEPT_NAME, DEPT_ADDRESS)

PROFESSOR (PROF_NUM, PROF_LNAME, PROF_FNAME, PROF_INITIAL, DEPT_CODE)

CASE 5: 1:M, Both Sides OPTIONAL

Foreign key always on the “many” side

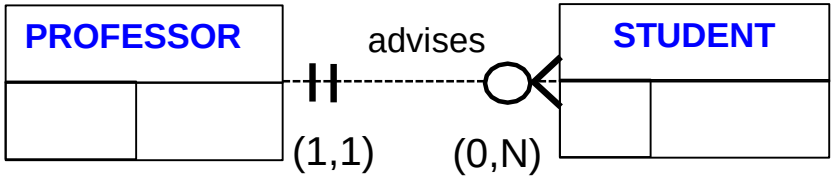


ROOM (ROOM_CODE, BLDG_CODE)

STUDENT (STU_NUM, STU_LNAME, STU_FNAME, STU_INITIAL, ROOM_CODE)

CASE 6: 1:M, Many Side OPTIONAL, One Side MANDATORY

Foreign key always on the “many” side

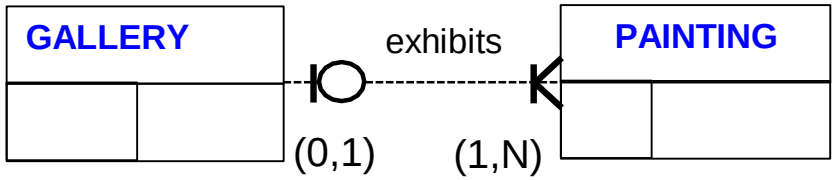


PROFESSOR (PROF_NUM, PROF_LNAME, PROF_FNAME, PROF_INITIAL)

STUDENT (STU_NUM, STU_LNAME, STU_FNAME, STU_INITIAL, PROF_NUM)

CASE 7: 1:M, One Side OPTIONAL, One Side MANDATORY

Foreign key always on the “many” side

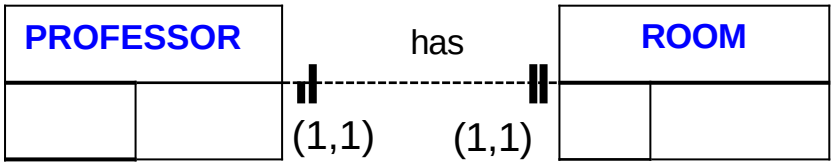


GALLERY (GAL_NUM, GAL_OWNER,GAL_PHONE, GAL_RATE)

PAINTING (PTNG_NUM, PTNG_TITLE, PTNG_PRICE, GAL_NUM)

CASE 8: 1:1, Both Sides MANDATORY

Foreign key is the strong entity or in the most frequently accessed entity

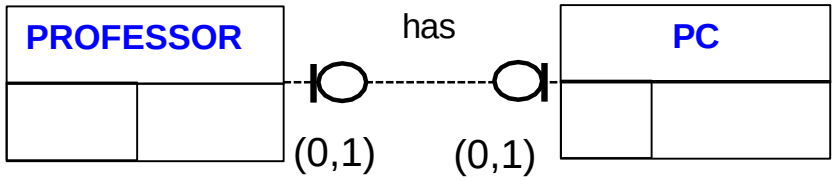


PROFESSOR (PROF_NUM, PROF_LNAME, PROF_FNAME, PROF_INITIAL, ROOM_CODE)

ROOM (ROOM_CODE, BLDG_CODE)

CASE 9: 1:1, Both Sides OPTIONAL

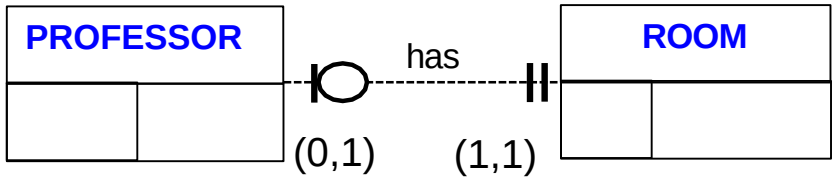
Foreign key is the strong entity or in the most frequently accessed entity



PC (PC_NUM, PC_BRAND, PROF_NUM)

PROFESSOR (PROF_NUM, PROF_LNAME, PROF_FNAME, PROF_INITIAL)

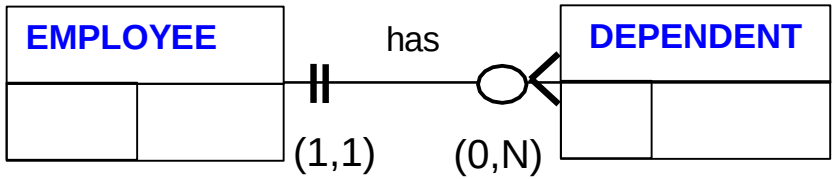
CASE 10: 1:1, One Side OPTIONAL, One Side MANDATORY



PROFESSOR (PROF_NUM, PROF_LNAME, PROF_FNAME, PROF_INITIAL, ROOM_CODE)

ROOM (ROOM_CODE, BLDG_CODE)

CASE 11: Weak Entity



EMPLOYEE (EMP_NUM, EMP_LNAME, EMP_FNAME, EMP_INITIAL, EMP_DOB)

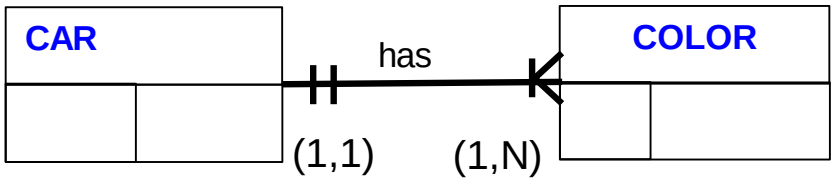
DEPENDENT (EMP_NUM, DEP_NUM, DEP_FNAME, DEP_DOB)

[Recap]: Weak Entity meet two conditions:

1. *Existence-dependent (an entity cannot exist without the entity that it has a relationship)*
2. *Primary key is partially/totally derived from parent entity*

CASE 12: Multivalued Attributes

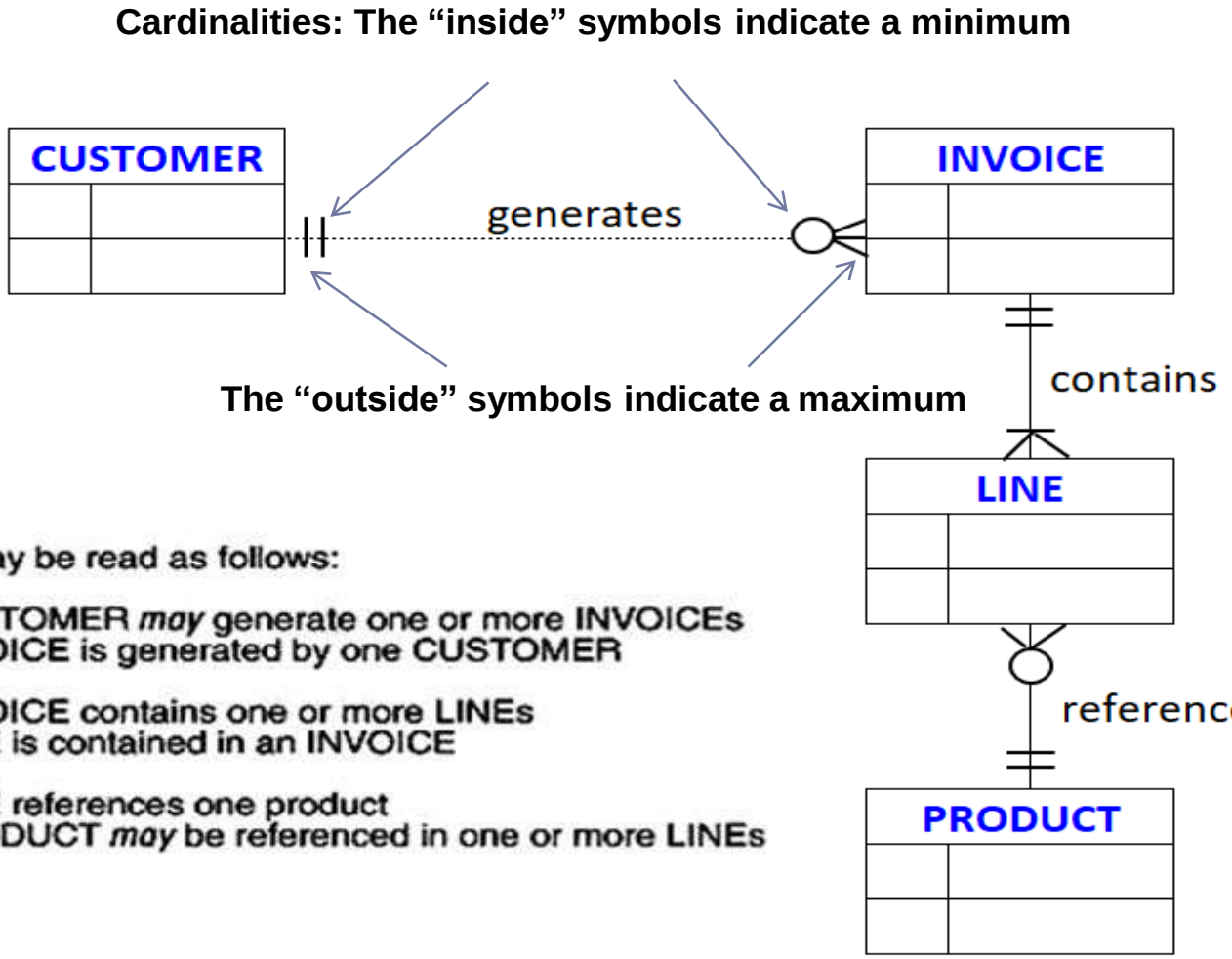
New table in 1:M relationship, foreign key (CAR_NUM) in the new table.
Conform to weak entity rules.



CAR (CAR_NUM, YEAR, MODEL)

COLOR (CAR_NUM, SECTION, COLOR)

Crow's Foot Model for INVOICING



This model may be read as follows:

each CUSTOMER *may* generate one or more INVOICEs
each INVOICE is generated by one CUSTOMER

each INVOICE contains one or more LINEs
each LINE is contained in an INVOICE

each LINE references one product
each PRODUCT *may* be referenced in one or more LINEs