Database Management System -I

INTRODUCTION

0301203

DATABASE MANAGEMENT SYSTEM - I

0301206

PRACTICAL ON DBMS - I

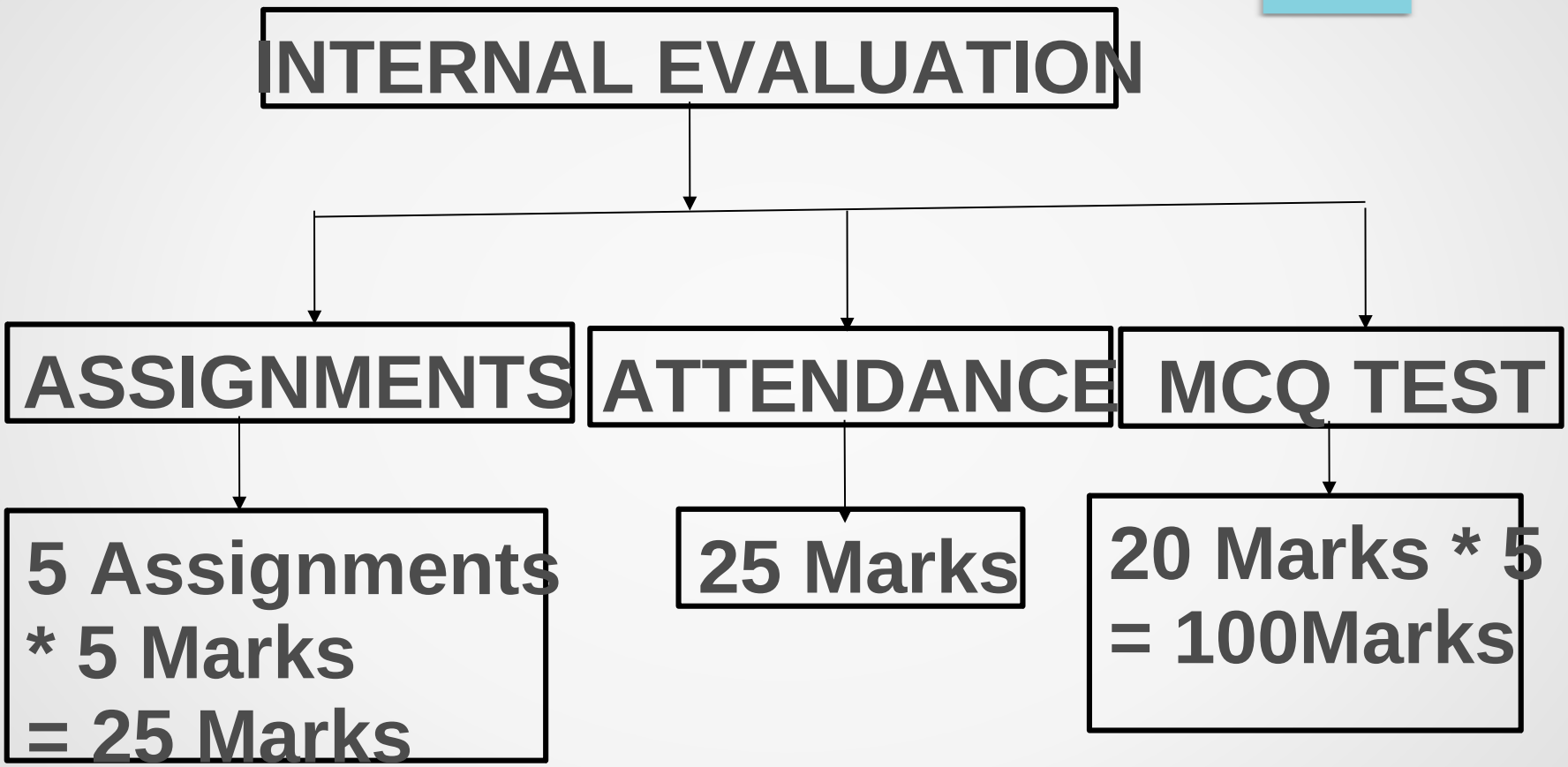
BY:

Prof. (Dr.) Ankit Bhavsar

0301203 Database Management System - I

UNIT	MODULES	WEIGHTAGE
1	Introduction to DBMS	20 %
2	Introduction to RDBMS	20 %
3	Inroduction to Normalization	20 %
4	Open Source Database Management Software	20 %

INTERNAL EVALUATION



UNIT - 2 Introduction to RDBMS

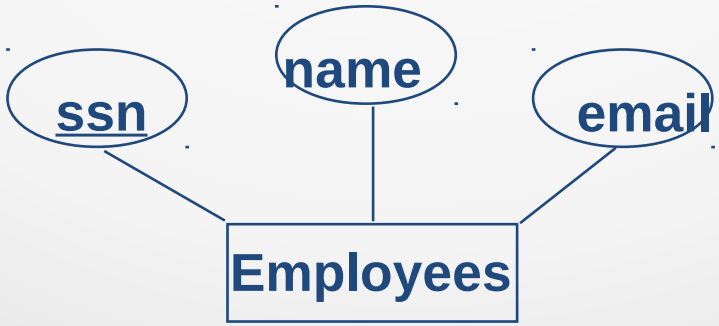
- Introduction to Enitity Relationship Model
  - Entity & Attribute
  - Connectivity and Cardinality
  - Types of Notation
  - Types of Entity
  - Degree of Relationship
  - Creating and E-R Diagram

Why we need ER diagram

“ER diagrams are easy for non-technical people to understand, and thus are typically used by database designers before the schema ever exists”

Entity

- An **entity** is something that exists by itself.
- *Entity:* Real-world object distinguishable from other objects. An entity is described using a set of *attributes*.
- The entity name, a noun, is written in capital letters.



Examples of entities

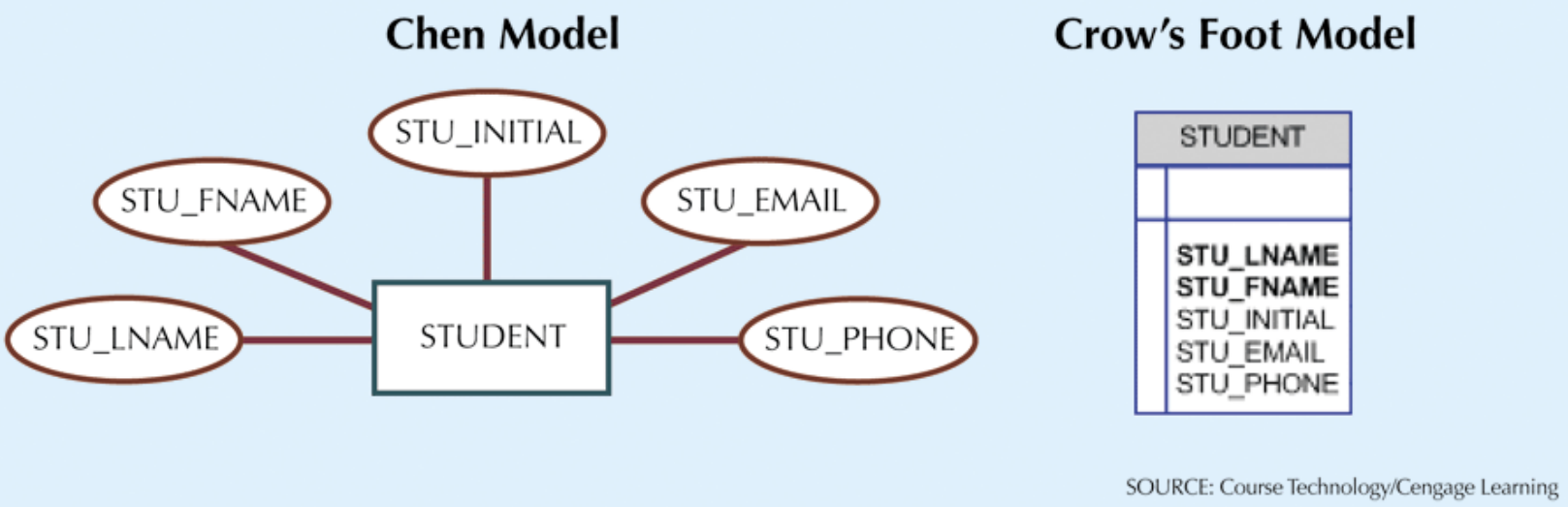
- **Person: EMPLOYEE, STUDENT, PATIENT**
- **Place: STORE, WAREHOUSE**
- **Object: MACHINE, PRODUCT, CAR**
- **Event: SALE,REGISTRATION, RENEWAL**
- **Concept: ACCOUNT, COURSE**

Attributes

- Example of entity types and associated attributes:

  STUDENT: Student_ID, Student_Name, Home_Address, Phone_Number, Major

FIGURE 4.1 The attributes of the STUDENT entity: Chen and Crow's Foot



SOURCE: Course Technology/Cengage Learning

Attribute types

- ***Simple* and *composite* attributes.**
  - A **simple attribute** is an attribute that cannot be subdivided. For example, age, sex, and marital status would be classified as simple attributes
  - A **composite attribute**, not to be confused with a composite key is an attribute that can be further subdivided to yield additional attributes. For example,
  - the attribute ADDRESS can be subdivided into street, city, state, and zip code. Similarly,
  - the attribute PHONE_NUMBER can be subdivided into area code and exchange number.

Attribute types

- ***Single-valued* and *multi-valued* attributes**
  - A **single-valued** attribute is an attribute that can have only a single value. For example, a person can have only one Social Security number, and a manufactured part can have only one serial number
  - **Multivalued attributes** are attributes that can have many values. For instance, a person may have several college degrees, and a household may have several different phones, each with its own number
- ***Derived* attributes**
  - **Can be computed from other attributes**
  - Example: age, given date_of_birth

Attributes

- Are characteristics of entities
- Chen notation: attributes represented by ovals connected to entity rectangle with a line
  - Each oval contains the name of attribute it represents
- Crow's Foot notation: attributes written in attribute box below entity rectangle

Types of Entity

Entities based on their characteristics are classified as follows.

- Strong Entities
- Weak Entities
- Composite Entities

Existence Dependence

Existence dependence

Entity exists in database only when it is associated with another related entity occurrence.

An entity is existence-dependent if it has a mandatory foreign key. (foreign key attribute cannot be null)

Example:

Spouse entity is existence dependent on the employee

Payment entity is existence dependent on the loan.

Existence independence

Entity can exist apart from one or more related entities

Sometimes such an entity is referred to as a strong or regular entity

Example:

Parts and Vendor (Part is supplied by Vendor but some parts are made inhouse then Part is existence independence from Vendor)

Types of Entities

Weak Entities

- These tables are existence dependent.
- They cannot exist without entity with which it has a relationship.
- Primary key is derived from the primary key of the parent entity
- The spouse table is a weak entity because its PK is dependent on the employee table. Without a corresponding employee record, the spouse record could not exist

Strong Entities

- If an entity can exist apart from all of its related entities it's considered a strong entity.
- A table without a foreign key is or a table that contains a foreign key which can contain NULLS is a strong entity.

FIGURE 4.8 A weak (non-identifying) relationship between COURSE and CLASS

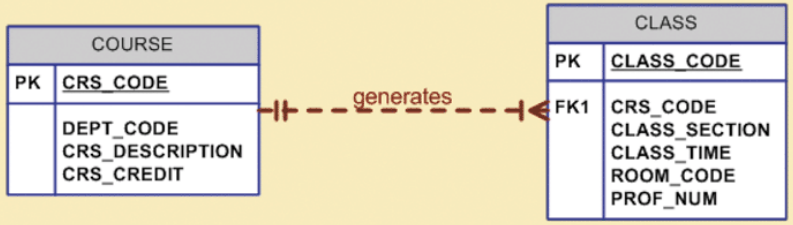


Table name: COURSE **Database name: Ch04_TinyCollege**

CRS_CODE	DEPT_CODE	CRS_DESCRIPTION	CRS_CREDIT
ACCT-211	ACCT	Accounting I	3
ACCT-212	ACCT	Accounting II	3
CIS-220	CIS	Intro. to Microcomputing	3
CIS-420	CIS	Database Design and Implementation	4
MATH-243	MATH	Mathematics for Managers	3
QM-261	CIS	Intro. to Statistics	3
QM-362	CIS	Statistical Applications	4

Table name: CLASS

CLASS_CODE	CRS_CODE	CLASS_SECTION	CLASS_TIME	ROOM_CODE	PROF_NUM
10012	ACCT-211	1	MWF 8:00-8:50 a.m.	BUS311	105
10013	ACCT-211	2	MWF 9:00-9:50 a.m.	BUS200	105
10014	ACCT-211	3	TTh 2:30-3:45 p.m.	BUS252	342
10015	ACCT-212	1	MWF 10:00-10:50 a.m.	BUS311	301
10016	ACCT-212	2	Th 6:00-8:40 p.m.	BUS252	301
10017	CIS-220	1	MWF 9:00-9:50 a.m.	KLR209	228
10018	CIS-220	2	MWF 9:00-9:50 a.m.	KLR211	114
10019	CIS-220	3	MWF 10:00-10:50 a.m.	KLR209	228
10020	CIS-420	1	W 6:00-8:40 p.m.	KLR209	162
10021	QM-261	1	MWF 8:00-8:50 a.m.	KLR200	114
10022	QM-261	2	TTh 1:00-2:15 p.m.	KLR200	114
10023	QM-362	1	MWF 11:00-11:50 a.m.	KLR200	162
10024	QM-362	2	TTh 2:30-3:45 p.m.	KLR200	162
10025	MATH-243	1	Th 6:00-8:40 p.m.	DRE155	325

SOURCE: Course Technology/Cengage Learning

FIGURE 4.9 A strong (identifying) relationship between COURSE and CLASS

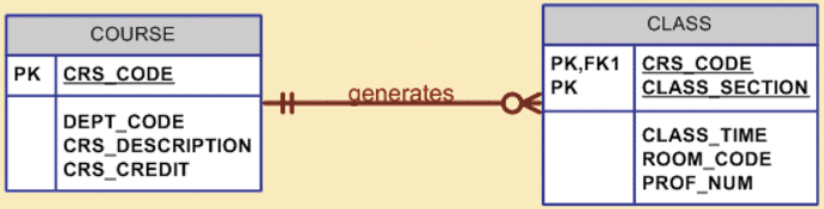


Table name: COURSE **Database name: Ch04_TinyCollege_Alt**

CRS_CODE	DEPT_CODE	CRS_DESCRIPTION	CRS_CREDIT
ACCT-211	ACCT	Accounting I	3
ACCT-212	ACCT	Accounting II	3
CIS-220	CIS	Intro. to Microcomputing	3
CIS-420	CIS	Database Design and Implementation	4
MATH-243	MATH	Mathematics for Managers	3
QM-261	CIS	Intro. to Statistics	3
QM-362	CIS	Statistical Applications	4

Table name: CLASS

CRS_CODE	CLASS_SECTION	CLASS_TIME	ROOM_CODE	PROF_NUM
ACCT-211	1	MWF 8:00-8:50 a.m.	BUS311	105
ACCT-211	2	MWF 9:00-9:50 a.m.	BUS200	105
ACCT-211	3	TTh 2:30-3:45 p.m.	BUS252	342
ACCT-212	1	MWF 10:00-10:50 a.m.	BUS311	301
ACCT-212	2	Th 6:00-8:40 p.m.	BUS252	301
CIS-220	1	MWF 9:00-9:50 a.m.	KLR209	228
CIS-220	2	MWF 9:00-9:50 a.m.	KLR211	114
CIS-220	3	MWF 10:00-10:50 a.m.	KLR209	228
CIS-420	1	W 6:00-8:40 p.m.	KLR209	162
MATH-243	1	Th 6:00-8:40 p.m.	DRE155	325
QM-261	1	MWF 8:00-8:50 a.m.	KLR200	114
QM-261	2	TTh 1:00-2:15 p.m.	KLR200	114
QM-362	1	MWF 11:00-11:50 a.m.	KLR200	162
QM-362	2	TTh 2:30-3:45 p.m.	KLR200	162

SOURCE: Course Technology/Cengage Learning

Types of Entities

- **Associative (Composite) Entities**
  - Also known as bridge entities
  - Used to implement M:N relationships
  - Composed of primary keys of each of the entities to be connected
  - May also contain additional attributes that play no role in connective process

Types of Entities

FIGURE 4.23 Converting the M:N relationship into two 1:M relationships

Table name: STUDENT — **Database name: Ch04_CollegeTry**

STU_NUM	STU_LNAME
321452	Bowser
324257	Smithson

Table name: ENROLL

CLASS_CODE	STU_NUM	ENROLL_GRADE
10014	321452	C
10014	324257	B
10018	321452	A
10018	324257	B
10021	321452	C
10021	324257	C

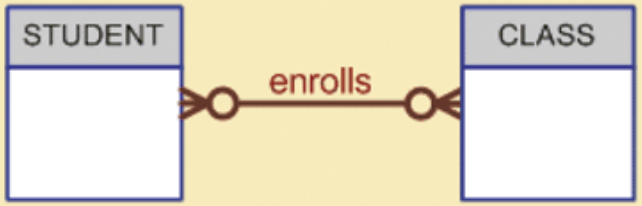
Table name: CLASS

CLASS_CODE	CRS_CODE	CLASS_SECTION	CLASS_TIME	ROOM_CODE	PROF_NUM
10014	ACCT-211	3	TTh 2:30-3:45 p.m.	BUS252	342
10018	CIS-220	2	MWF 9:00-9:50 a.m.	KLR211	114
10021	QM-261	1	MWF 8:00-8:50 a.m.	KLR200	114

SOURCE: Course Technology/Cengage Learning

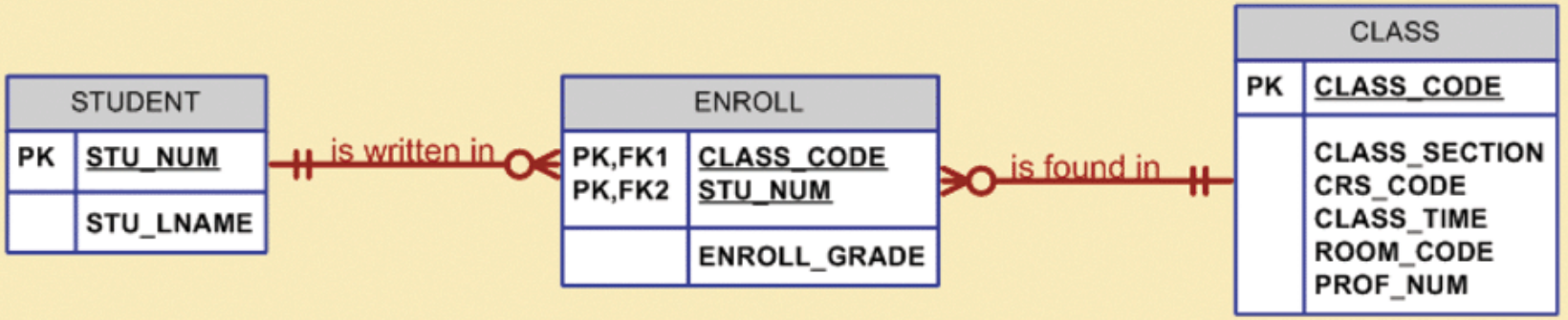
Types of Entities

FIGURE 4.24 The M:N relationship between STUDENT and CLASS



SOURCE: Course Technology/Cengage Learning

FIGURE 4.25 A composite entity in an ERD



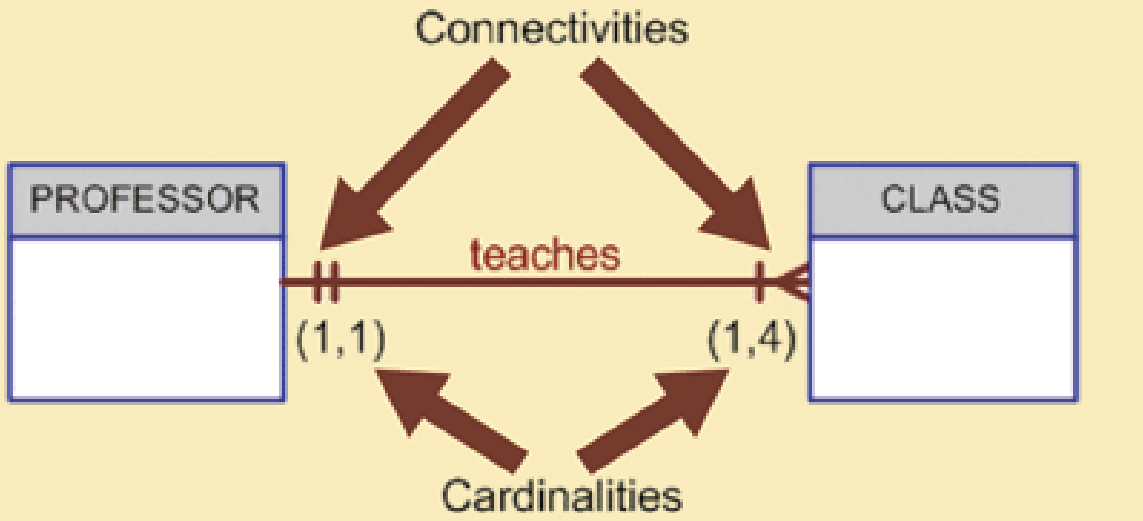
SOURCE: Course Technology/Cengage Learning

Connectivity and Cardinality

- Connectivity is used to **describe the relationship** classification.
- Cardinality **expresses the minimum and maximum number of entity occurrences** associated with one occurrence of the related entity.

Connectivity and Cardinality

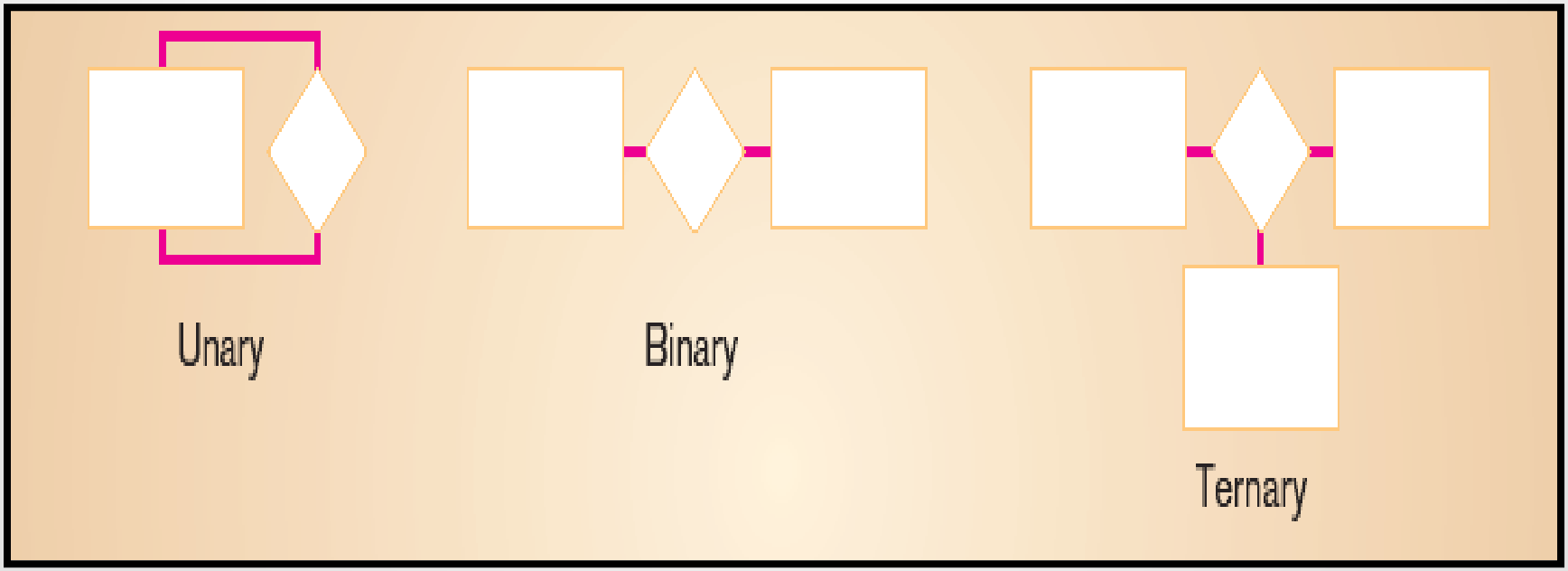
FIGURE 4.7 Connectivity and cardinality in an ERD



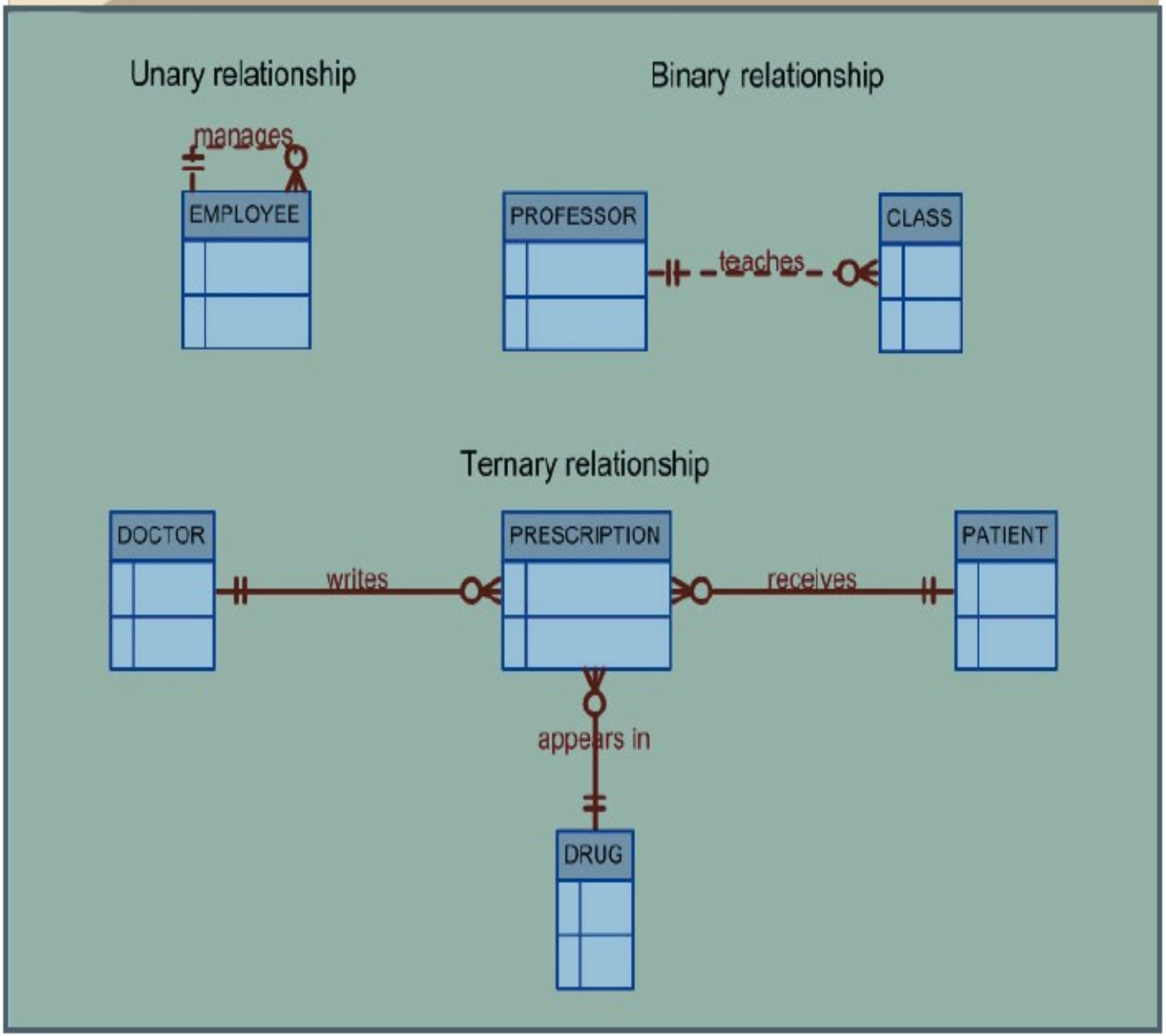
SOURCE: Course Technology/Cengage Learning

Degree of Relationships

- A relationship degree indicates the number of entities or participants associated with a relationship.Three cases
  - **Unary:**A unary relationship exists when an association is maintained within a single entity
  - **Binary:** A binary relationship exists when two entities are associated in a relationship
  - **Ternary:** A ternary relationship implies an association among three different entities.



4.15


Unary relationship
manages
EMPLOYEE
Binary relationship
PROFESSOR
teaches
CLASS
Ternary relationship
DOCTOR
writes
PRESCRIPTION
receives
PATIENT
appears in
DRUG

Recursive Relationships

Relationship can exist between occurrences of the same entity set

Naturally found within unary relationship

FIGURE 4.19 Another unary relationship: "PART contains PART"

Table name: PART_V1 Database name: CH04_PartCo

PART_CODE	PART_DESCRIPTION	PART_IN_STOCK	PART_UNITS_NEEDED	PART_OF_PART
AA21-6	2.5 cm. washer, 1.0 mm. rim	432	4	C-130
AB-121	Cotter pin, copper	1034	2	C-130
C-130	Rotor assembly	36		
E129	2.5 cm. steel shank	128	1	C-130
X10	10.25 cm. rotor blade	345	4	C-130
X34AW	2.5 cm. hex nut	879	2	C-130

SOURCE: Course Technology/Cengage Learning

FIGURE 4.22 Implementation of the 1:M recursive relationship "EMPLOYEE manages EMPLOYEE"

Database name: Ch04_PartCo

Table name: EMPLOYEE_V2

EMP_CODE	EMP_LNAME	EMP_MANAGER
101	Waddell	102
102	Orincona	
103	Jones	102
104	Reballoh	102
105	Robertson	102
106	Deltona	102

SOURCE: Course Technology/Cengage Learning

Functional Dependency

- Functional dependency is a relationship that exists when one attribute uniquely determines another attribute. If R is a relation with attributes X and Y, a functional dependency between the attributes is represented as X->Y, which specifies Y is functionally dependent on X.
- The attributes of a table is said to be dependent on each other when an attribute of a table uniquely identifies another attribute of the same table.
- If column A of a table uniquely identifies the column B of same table then it can represented as A->B (Attribute B is functionally dependent on attribute A)

- For example:
  Suppose we have a student table with attributes:

  Stu_Id, Stu_Name, Stu_Age.

  Here **Stu_Id** attribute uniquely identifies the **Stu_Name** attribute of student table because if we know the student id we can tell the student name associated with it.

- This is known as functional dependency and can be written as:

  Stu_Id->Stu_Name

  or we can say **Stu_Name** is functionally dependent on **Stu_Id.**