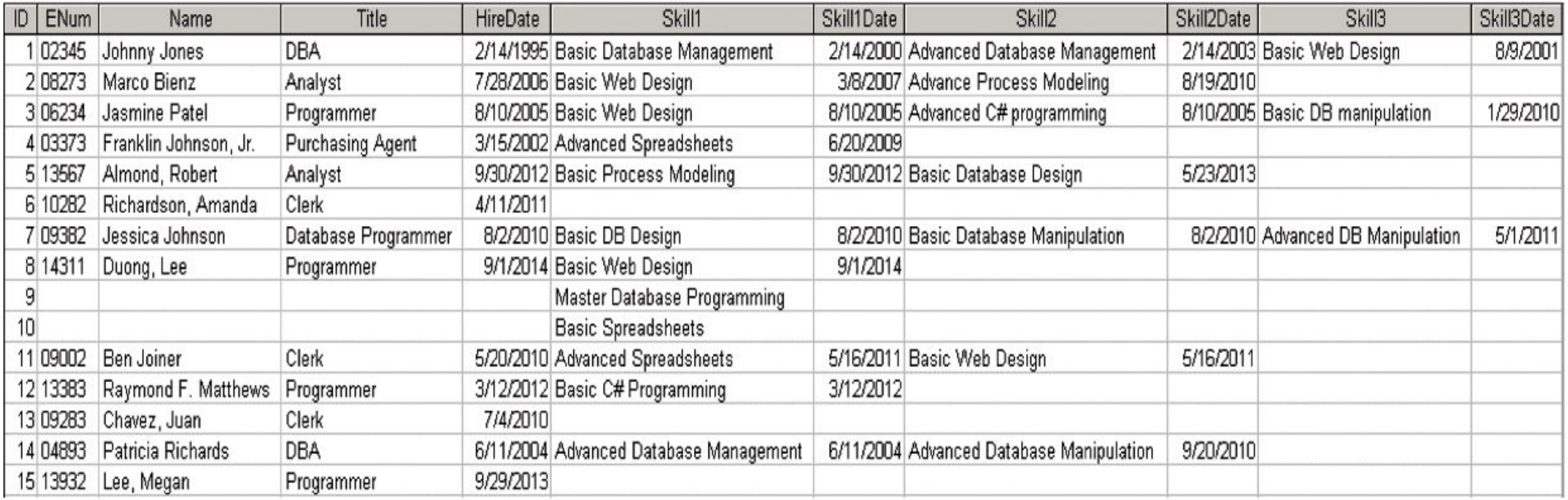
Chapter 1

CSCI 475

Bad Example - Employee Skills Table



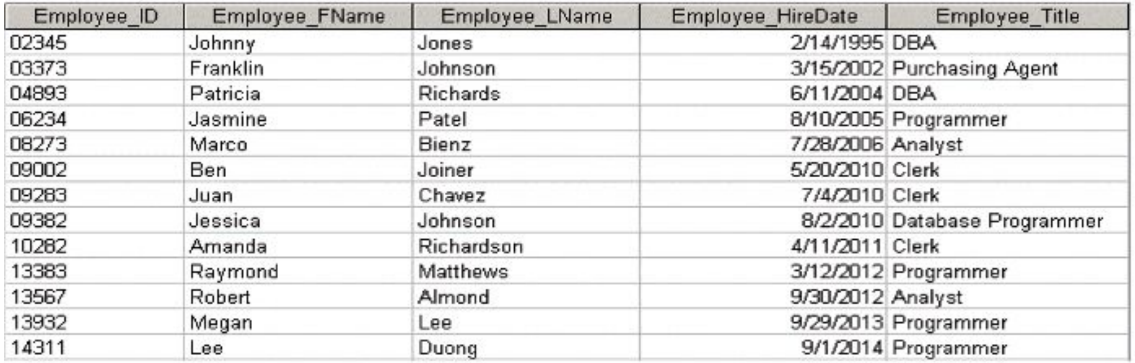
ID	ENum	Name	Title	HireDate	Skill1	Skill1Date	Skill2	Skill2Date	Skill3	Skill3Date
1	02345	Johnny Jones	DBA	2/14/1995	Basic Database Management	2/14/2000	Advanced Database Management	2/14/2003	Basic Web Design	8/9/2001
2	08273	Marco Bienz	Analyst	7/28/2006	Basic Web Design	3/8/2007	Advance Process Modeling	8/19/2010		
3	06234	Jasmine Patel	Programmer	8/10/2005	Basic Web Design	8/10/2005	Advanced C# programming	8/10/2005	Basic DB manipulation	1/29/2010
4	03373	Franklin Johnson, Jr.	Purchasing Agent	3/15/2002	Advanced Spreadsheets	6/20/2009				
5	13567	Almond, Robert	Analyst	9/30/2012	Basic Process Modeling	9/30/2012	Basic Database Design	5/23/2013		
6	10282	Richardson, Amanda	Clerk	4/11/2011						
7	09382	Jessica Johnson	Database Programmer	8/2/2010	Basic DB Design	8/2/2010	Basic Database Manipulation	8/2/2010	Advanced DB Manipulation	5/1/2011
8	14311	Duong, Lee	Programmer	9/1/2014	Basic Web Design	9/1/2014				
9					Master Database Programming					
10					Basic Spreadsheets					
11	09002	Ben Joiner	Clerk	5/20/2010	Advanced Spreadsheets	5/16/2011	Basic Web Design	5/16/2011		
12	13383	Raymond F. Matthews	Programmer	3/12/2012	Basic C# Programming	3/12/2012				
13	09283	Chavez, Juan	Clerk	7/4/2010						
14	04893	Patricia Richards	DBA	6/11/2004	Advanced Database Management	6/11/2004	Advanced Database Manipulation	9/20/2010		
15	13932	Lee, Megan	Programmer	9/29/2013						

This table is designed poorly

Let’s make the data more organized by splitting it up into multiple **related** tables

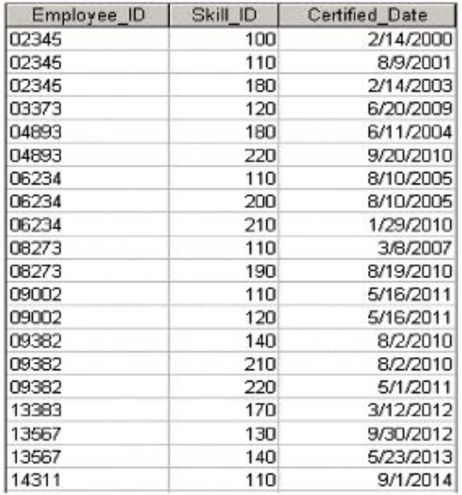
Good Example - Three Related Tables

Table name: EMPLOYEE



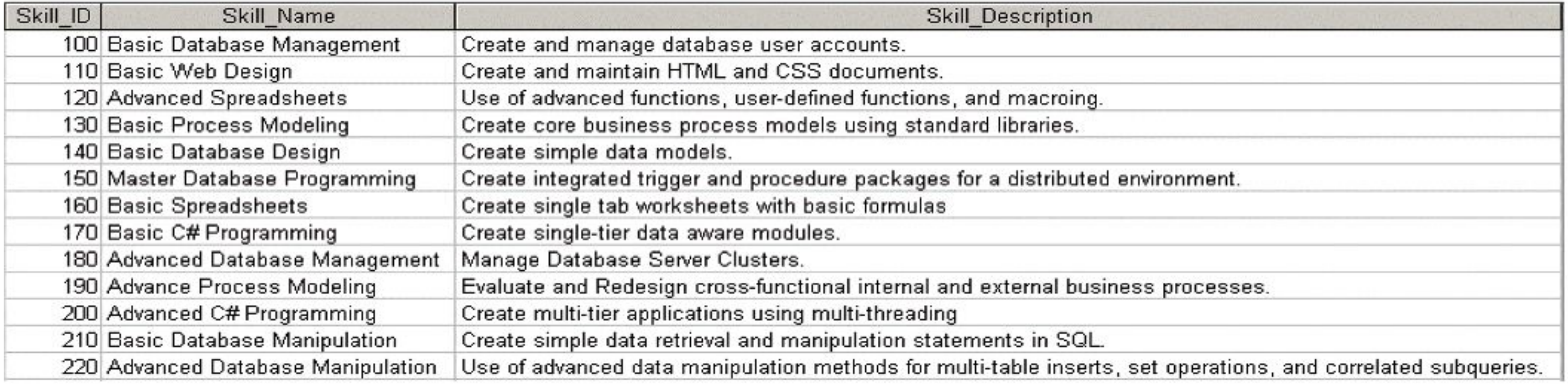
Employee_ID	Employee_FName	Employee_LName	Employee_HireDate	Employee_Title
02345	Johnny	Jones	2/14/1995	DBA
03373	Franklin	Johnson	3/15/2002	Purchasing Agent
04893	Patricia	Richards	6/11/2004	DBA
06234	Jasmine	Patel	8/10/2005	Programmer
08273	Marco	Bienz	7/28/2006	Analyst
09002	Ben	Joiner	5/20/2010	Clerk
09283	Juan	Chavez	7/4/2010	Clerk
09382	Jessica	Johnson	8/2/2010	Database Programmer
10282	Amanda	Richardson	4/11/2011	Clerk
13383	Raymond	Matthews	3/12/2012	Programmer
13567	Robert	Almond	9/30/2012	Analyst
13932	Megan	Lee	9/29/2013	Programmer
14311	Lee	Duong	9/1/2014	Programmer

Table name: CERTIFIED



Employee_ID	Skill_ID	Certified_Date
02345	100	2/14/2000
02345	110	8/9/2001
02345	180	2/14/2003
03373	120	6/20/2009
04893	180	6/11/2004
04893	220	9/20/2010
06234	110	8/10/2005
06234	200	8/10/2005
06234	210	1/29/2010
08273	110	3/8/2007
08273	190	8/19/2010
09002	110	5/16/2011
09002	120	5/16/2011
09382	140	8/2/2010
09382	210	8/2/2010
09382	220	5/1/2011
13383	170	3/12/2012
13567	130	9/30/2012
13567	140	5/23/2013
14311	110	9/1/2014

Table name: SKILL



Skill_ID	Skill_Name	Skill_Description
100	Basic Database Management	Create and manage database user accounts.
110	Basic Web Design	Create and maintain HTML and CSS documents.
120	Advanced Spreadsheets	Use of advanced functions, user-defined functions, and macroing.
130	Basic Process Modeling	Create core business process models using standard libraries.
140	Basic Database Design	Create simple data models.
150	Master Database Programming	Create integrated trigger and procedure packages for a distributed environment.
160	Basic Spreadsheets	Create single tab worksheets with basic formulas
170	Basic C# Programming	Create single-tier data aware modules.
180	Advanced Database Management	Manage Database Server Clusters.
190	Advance Process Modeling	Evaluate and Redesign cross-functional internal and external business processes.
200	Advanced C# Programming	Create multi-tier applications using multi-threading
210	Basic Database Manipulation	Create simple data retrieval and manipulation statements in SQL.
220	Advanced Database Manipulation	Use of advanced data manipulation methods for multi-table inserts, set operations, and correlated subqueries.

1. Display the employees in alphabetical order:

```
SELECT Employee_LName FROM EMPLOYEE ORDER BY Employee_LName ASC;
```

2. Write the query that lists all the names of those employees with the Basic Web Design skill:

```
SELECT Employee_LName FROM EMPLOYEE NATURAL JOIN CERTIFIED

NATURAL JOIN SKILL WHERE SKILL_Name = 'Basic Web Design';
```

```
SELECT Count(*) FROM EMPLOYEE NATURAL JOIN CERTIFIED

NATURAL JOIN SKILL WHERE SKILL_Name = 'Basic Web Design';
```

3. Insert a new Skill for Jasmine Patel – Advanced Spreadsheets.

```
INSERT INTO CERTIFIED VALUES('06234', 120, '2016-01-27');
```

```
INSERT INTO CERTIFIED VALUES((SELECT Employee_ID FROM EMPLOYEE
WHERE Employee_LName = 'Patel'), 120, '2016-01-27');
```

Chapter 2

CSCI 475

Relational Data Model Building Blocks

Entity – person, place, thing or event (think NOUN)

- is all uppercase
- abbreviation of name included in its attributes

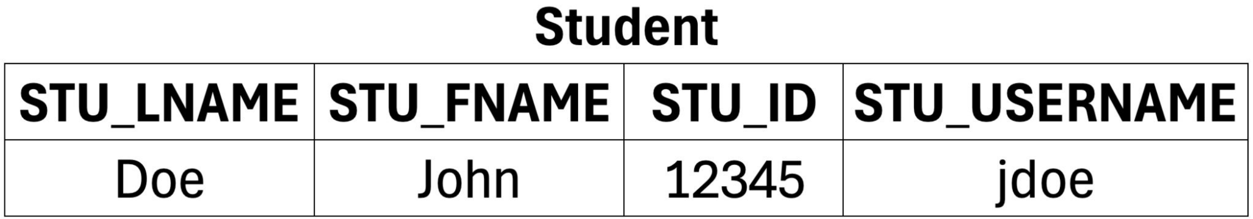
Tuple – set of data

Attribute – characteristic of an entity like first name, last name, etc.

Relationships – (think VERB) – association among entities

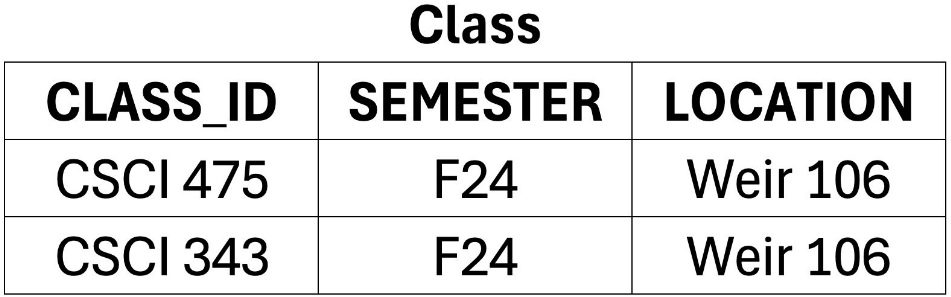
- STUDENT and CLASS entities
- STU_ID takes CLASS_ID (like CSCI 475)

Relational Data Model Building Blocks



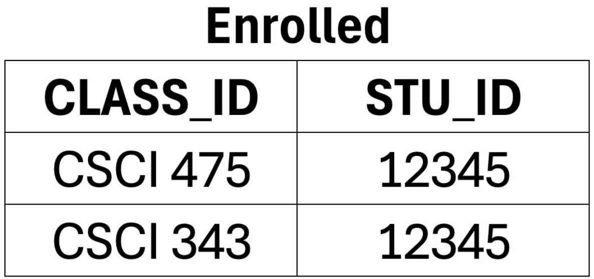
Student

STU_LNAME	STU_FNAME	STU_ID	STU_USERNAME
Doe	John	12345	jdoe



Class

CLASS_ID	SEMESTER	LOCATION
CSCI 475	F24	Weir 106
CSCI 343	F24	Weir 106



Enrolled

CLASS_ID	STU_ID
CSCI 475	12345
CSCI 343	12345

Relationships

Types - BIDIRECTIONAL

- 1:M One-to-Many

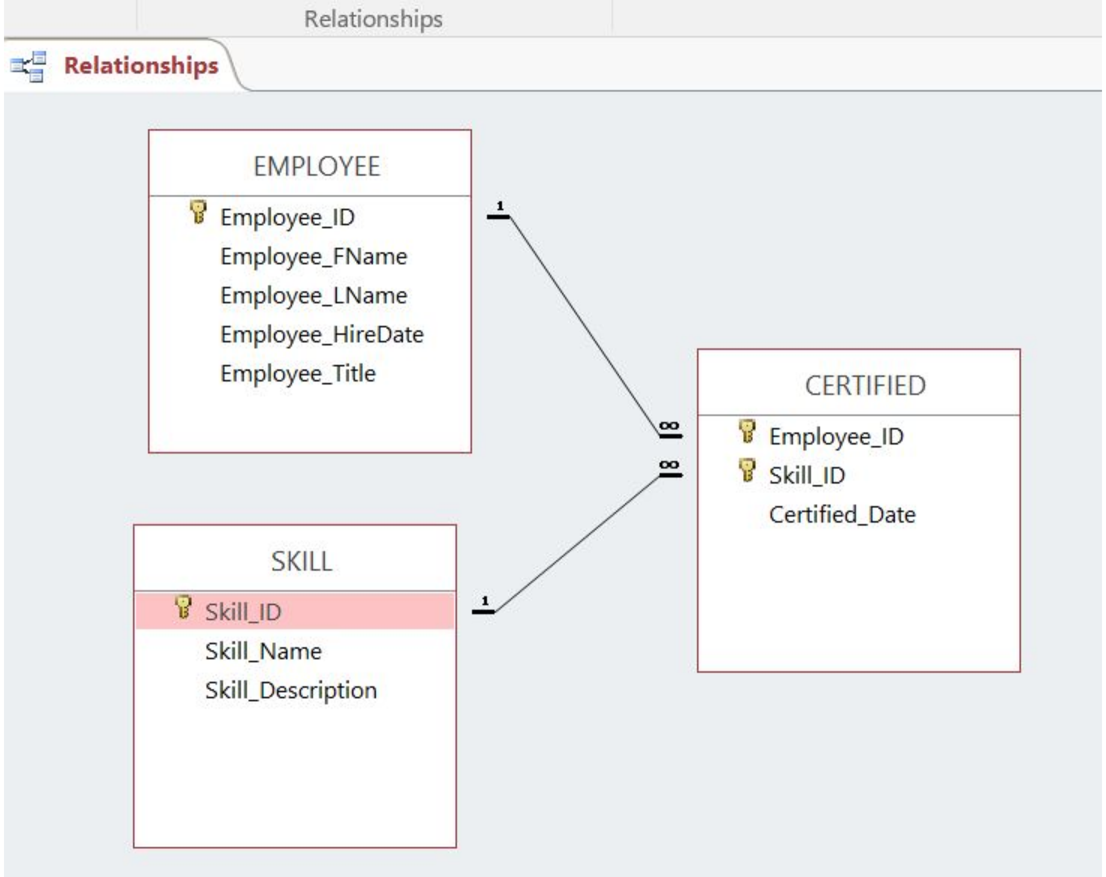
  1 painter produces multiple paintings

- M:N Many-to-Many

  Many employees learn many skills

- 1:1 One-to-One

  1 employee manages 1 store



EMPLOYEE

- Employee_ID
- Employee_FName
- Employee_LName
- Employee_HireDate
- Employee_Title

SKILL

- Skill_ID
- Skill_Name
- Skill_Description

CERTIFIED

- Employee_ID
- Skill_ID
- Certified_Date

Relationships

Schema - conceptual organization of database (database administrator’s view)

STUDENT(STU_LNAME, STU_FNAME, STU_ID, STU_USERNAME)
CLASS(CLASS_ID, SEMESTER, LOCATION)
ENROLLED(CLASS_ID, STU_ID)

Entity Relationship (ER) Model – entities and their relationships represented graphically with ER Diagram

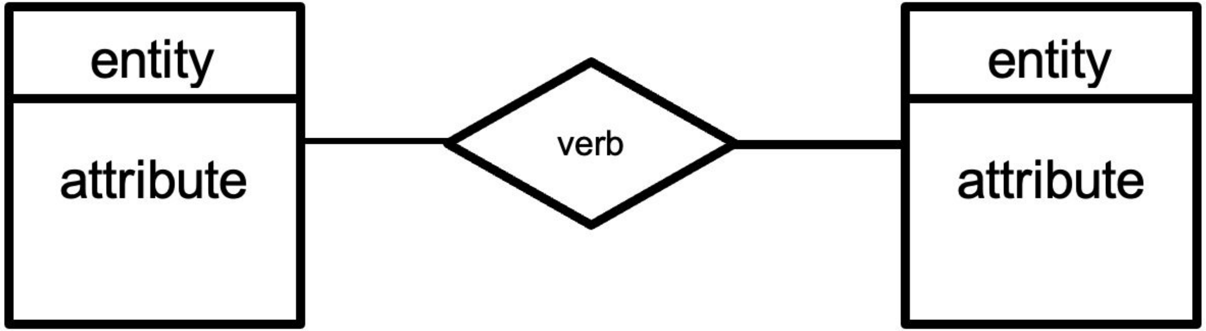
ER Diagram (ERD) – visual representation of entities

Business Rules – description of operation (narrative)

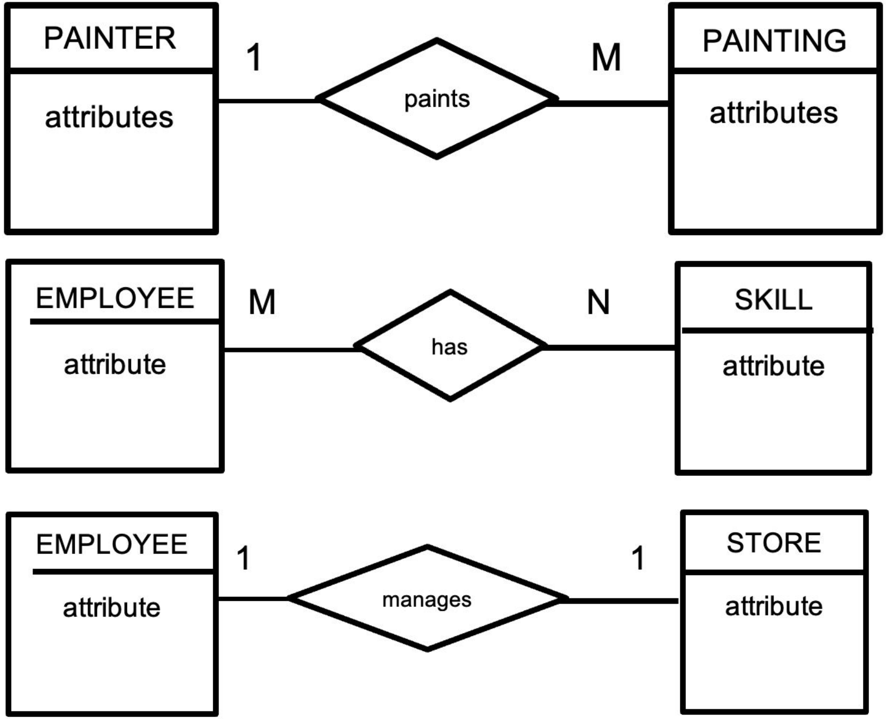
Chen Notation

Conceptualizing Data Model (like UML)

Introduced by Peter Chen in 1976 to conceptualize a data model



Chen Notation



Crow’s Foot Notation

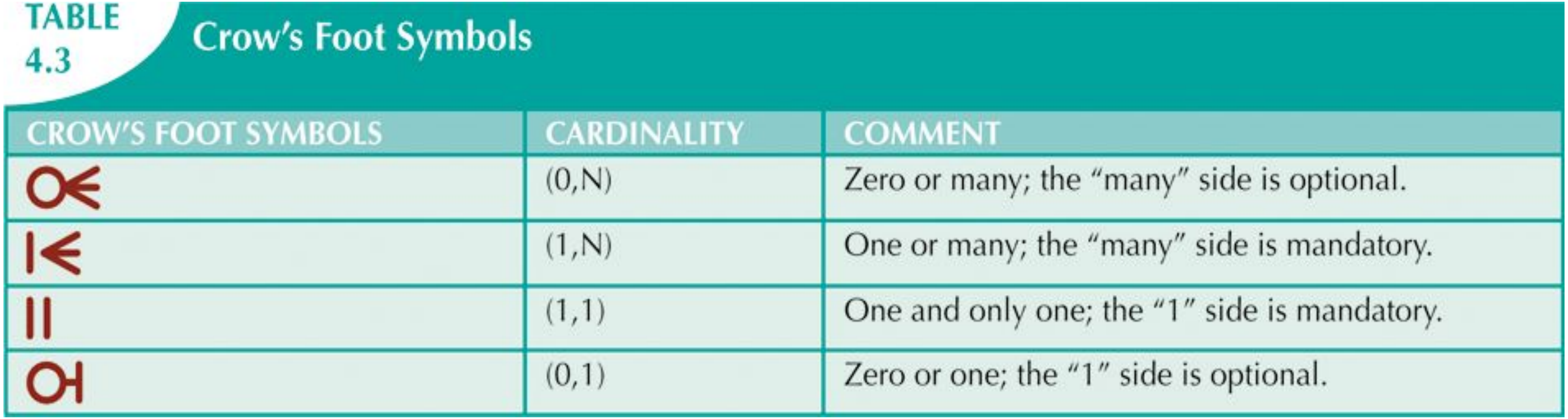
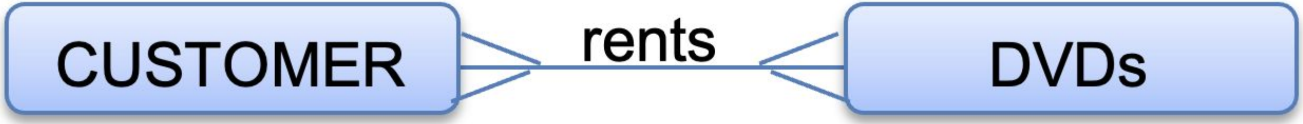


TABLE 4.3 Crow’s Foot Symbols

CROW’S FOOT SYMBOLS	CARDINALITY	COMMENT
○⋲	(0,N)	Zero or many; the “many” side is optional.
|⋲	(1,N)	One or many; the “many” side is mandatory.
||	(1,1)	One and only one; the “1” side is mandatory.
○|	(0,1)	Zero or one; the “1” side is optional.

Crow’s Foot Notation

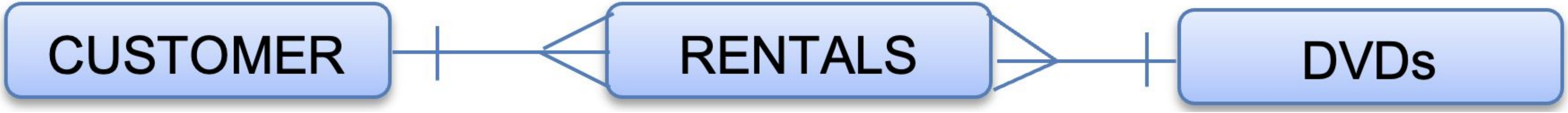
Many-to-Many



This is not implementable in a relational model

Crow’s Foot Notation

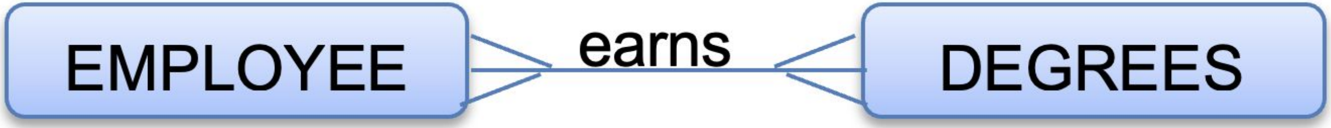
One-to-Many



This can be implemented

Crow’s Foot Notation

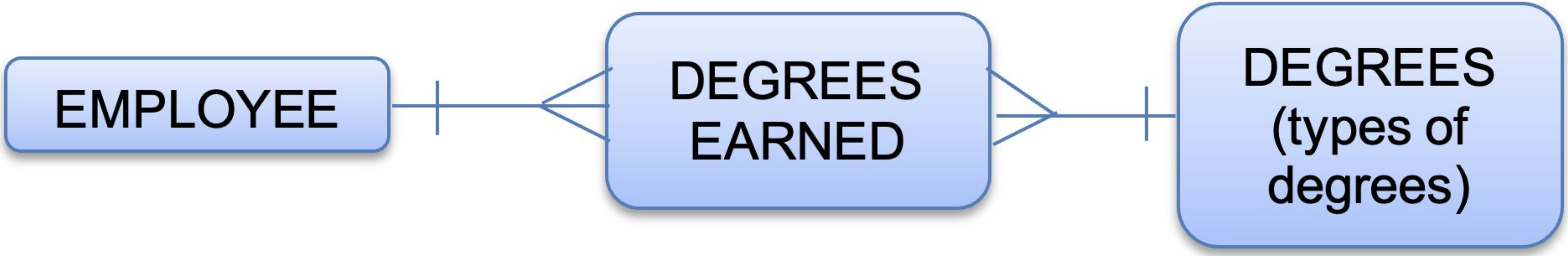
Many-to-Many



Here we have the same issue as before

Crow’s Foot Notation

One-to-Many

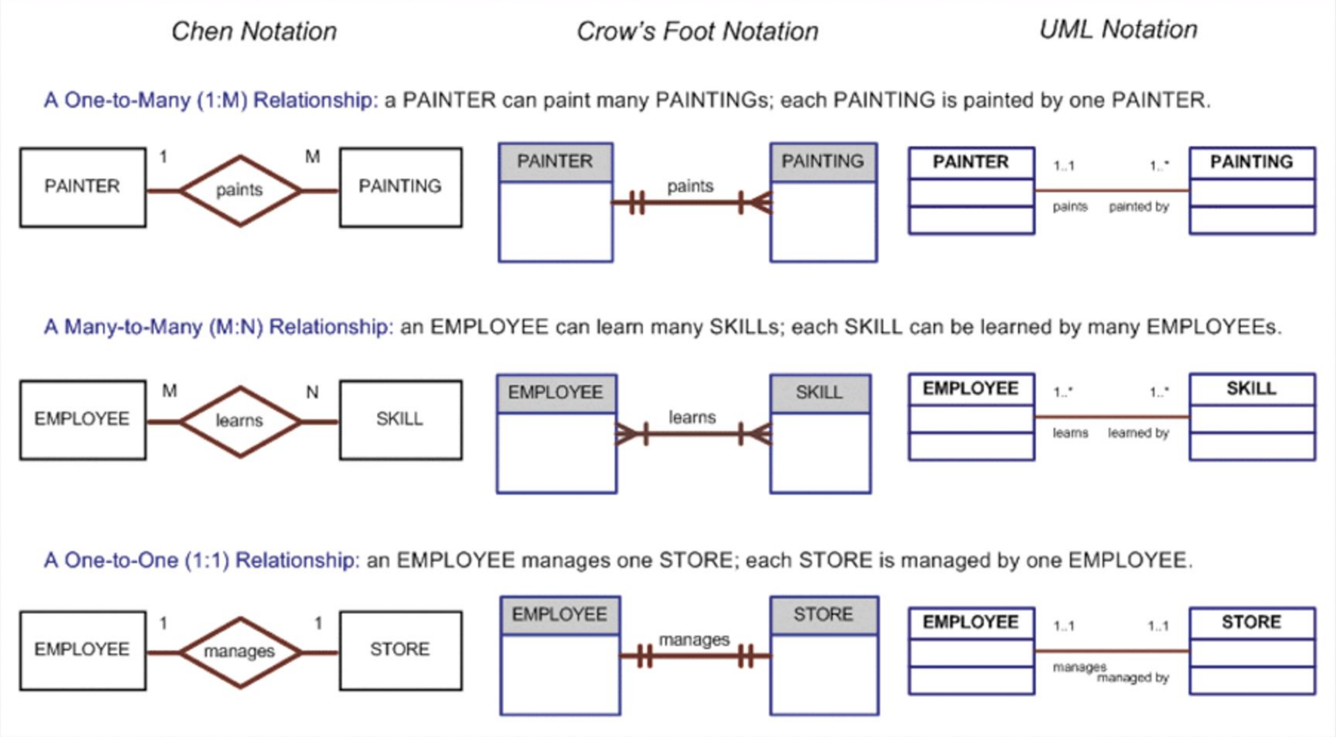


EMPLOYEE

DEGREES EARNED

DEGREES (types of degrees)

FIGURE 2.3 The ER model notations



Chen Notation *Crow's Foot Notation* *UML Notation*

A One-to-Many (1:M) Relationship: a PAINTER can paint many PAINTINGs; each PAINTING is painted by one PAINTER.

A Many-to-Many (M:N) Relationship: an EMPLOYEE can learn many SKILLs; each SKILL can be learned by many EMPLOYEEs.

A One-to-One (1:1) Relationship: an EMPLOYEE manages one STORE; each STORE is managed by one EMPLOYEE.

SOURCE: Course Technology/Cengage Learning