Data Models & Relational Model Characteristics

Lecture 2

Learning Outcomes

- In this chapter, you will learn:
  - terminologies related to database design
  - data models
  - relational database operators

Data Modeling

- ***Data modeling***
  - It is the *first step* in designing a database
  - It refers to the *process of creating* a specific data model for a problem domain

- ***Data model***
  - It is a *simple representation* (usually *graphical*) of complex real-world data structures
  - A **model** is an abstraction of complex real-world object

The Importance of Data Models

- Facilitate *interaction* among the *designer*, *programmer*, and the *end user*
- Data model is an *abstraction* ("**blueprint**")
  - A high-level description of your database.

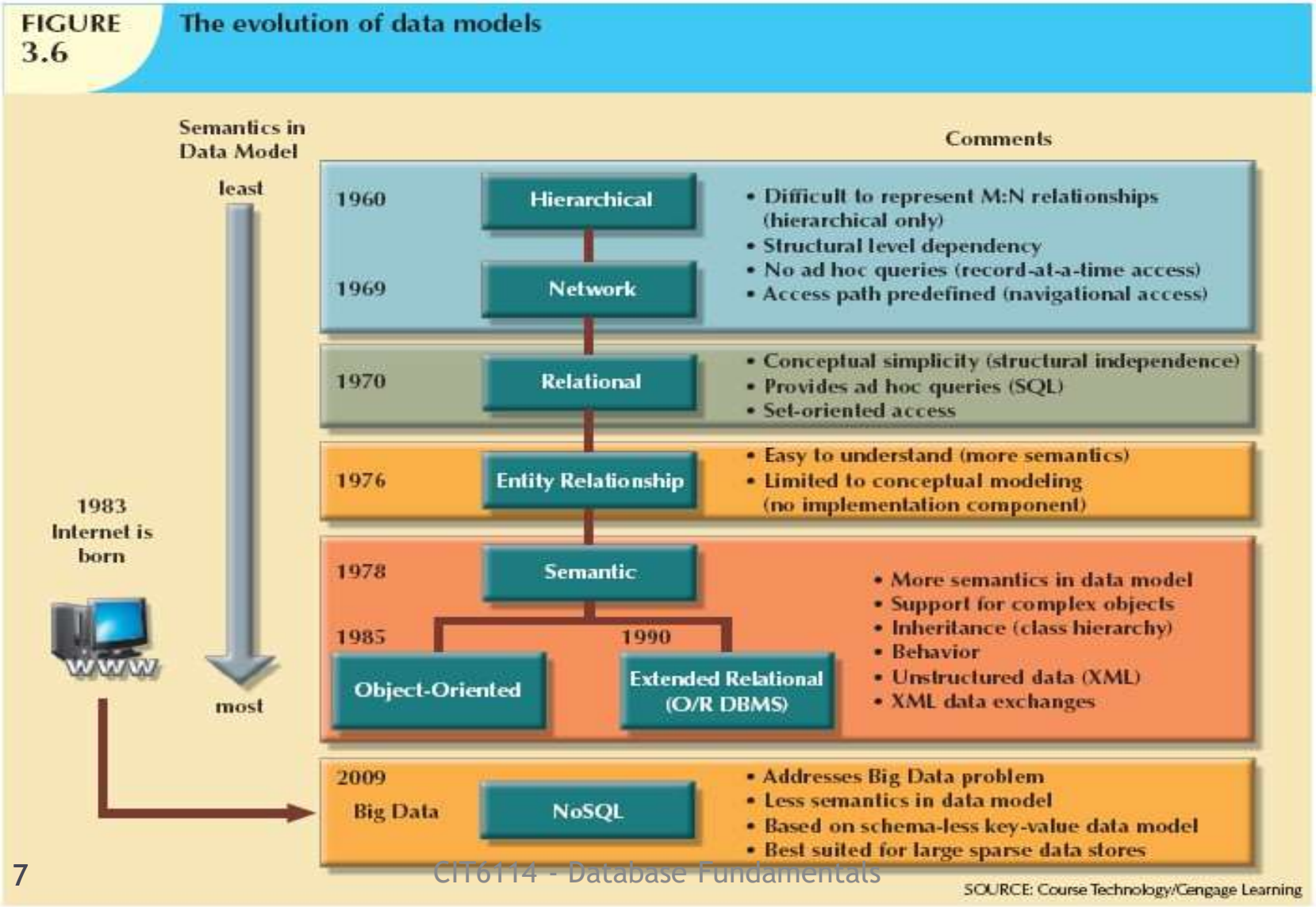
Basic Building Blocks

- ***Entity***: is a *person*, *place*, *thing* or *event* about which data will be collected and stored.
- ***Attribute***: a characteristic of an entity
- ***Relationship***: describes an association among entities
  - *One-to-many (1:M)*
  - *Many-to-many (M:N or M:M)*
  - *One-to-one (1:1)*
- ***Constraint***: a restriction placed on the data
  - E.g., student's GPA must be between 0 and 4

Business Rules

- is a ***brief***, ***precise***, ***unambiguous*** description of a policy, procedure or principle within a specific organization
- Properly written business rules are used to define entities, attributes, relationships and constraints.
- Example:
  - *An agent can serve many customers*
  - *A training session cannot be scheduled for fewer than 10 employees or for more than 30 employees*

Evolution of Data Models



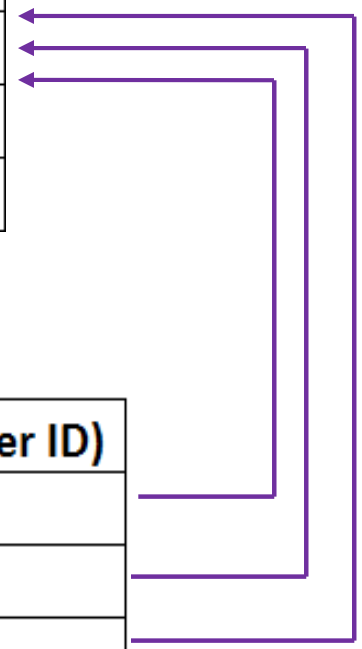
Hierarchical Model

- Data are represented as a **tree-like structure**
- The hierarchical structure contains levels
  - A higher layer is perceived as the **parent** of the level directly beneath it, which is called the **child**
- The hierarchical model depicts a set of **one-to-many** relationships between a parent and its children
  - Each parent can have many children, but each child has only one parent

Hierarchical Model

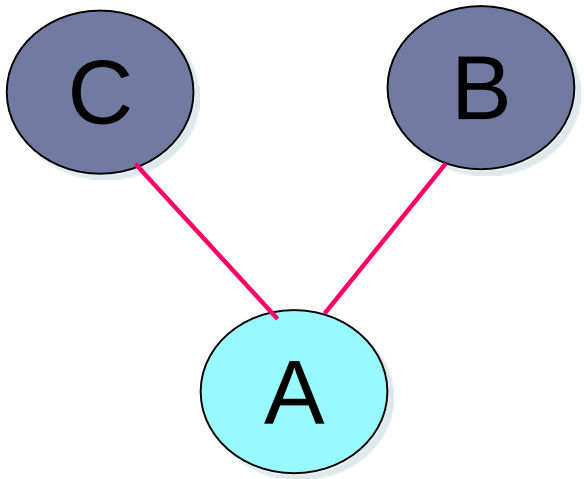
Lecturer ID	First Name	Last Name	Dept
L1001	Ben	Steven	FCI
L1002	John	Lincoln	FCI
L1003	Ali	Abu	FCI

Equipment Type	Serial No	User (Lecturer ID)
Computer	X-123-WW345	L1001
Monitor	M9878-222	L1001
Printer	P0989003	L1001
Computer	X-456-WW345	L1002



Network Model

- Unlike the hierarchical model (each child has only one parent), the **network model** allows a record to have more than one parent
  - Example: an *order* is related to both a *customer* who placed it and a *salesperson* who made the sale

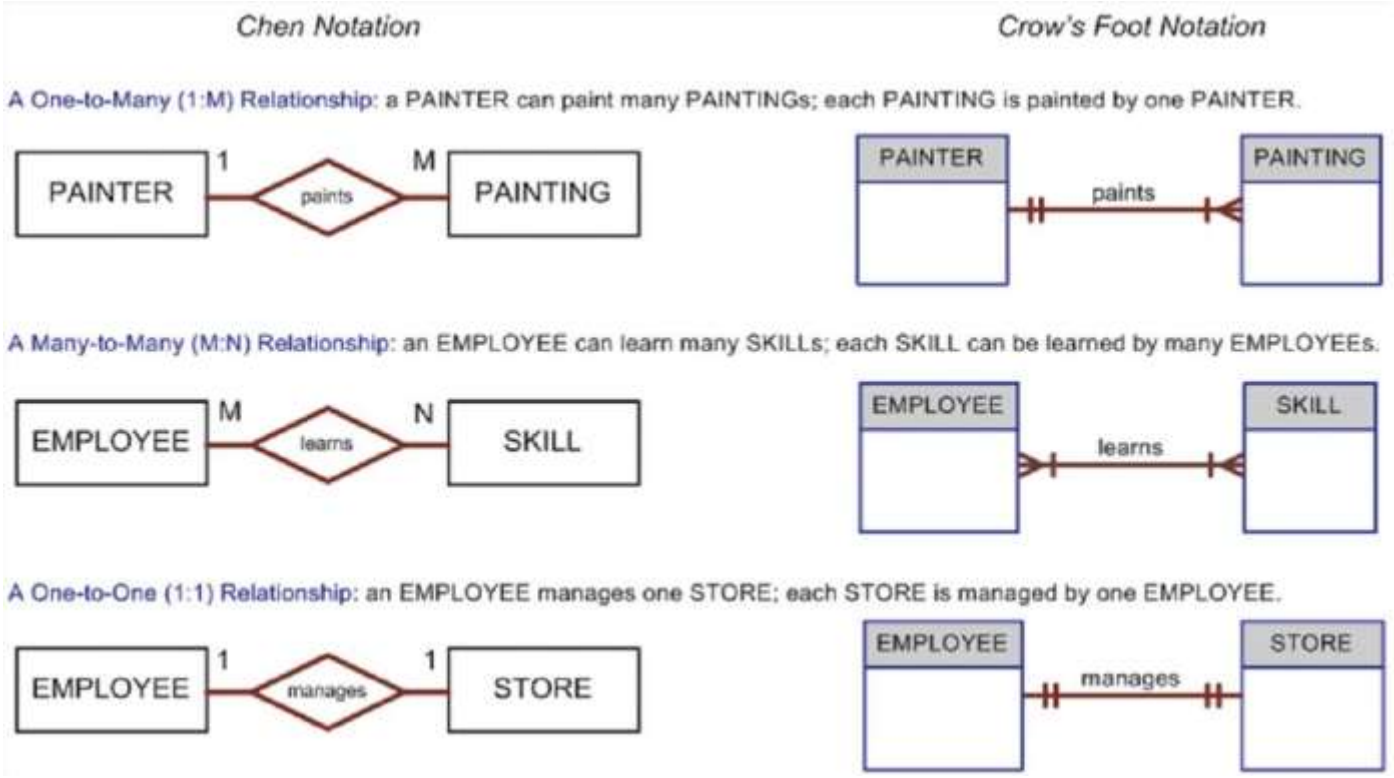


Relational Model

- The basic data structure of the relational model is the **table**
  - **Relation = table**
- Allows designer to focus on the logical representation of the data and its relationships rather than on the physical storage details

Entity-Relationship Diagram

- Based on entities, attributes, and relationships
- Complements the relational data model concepts



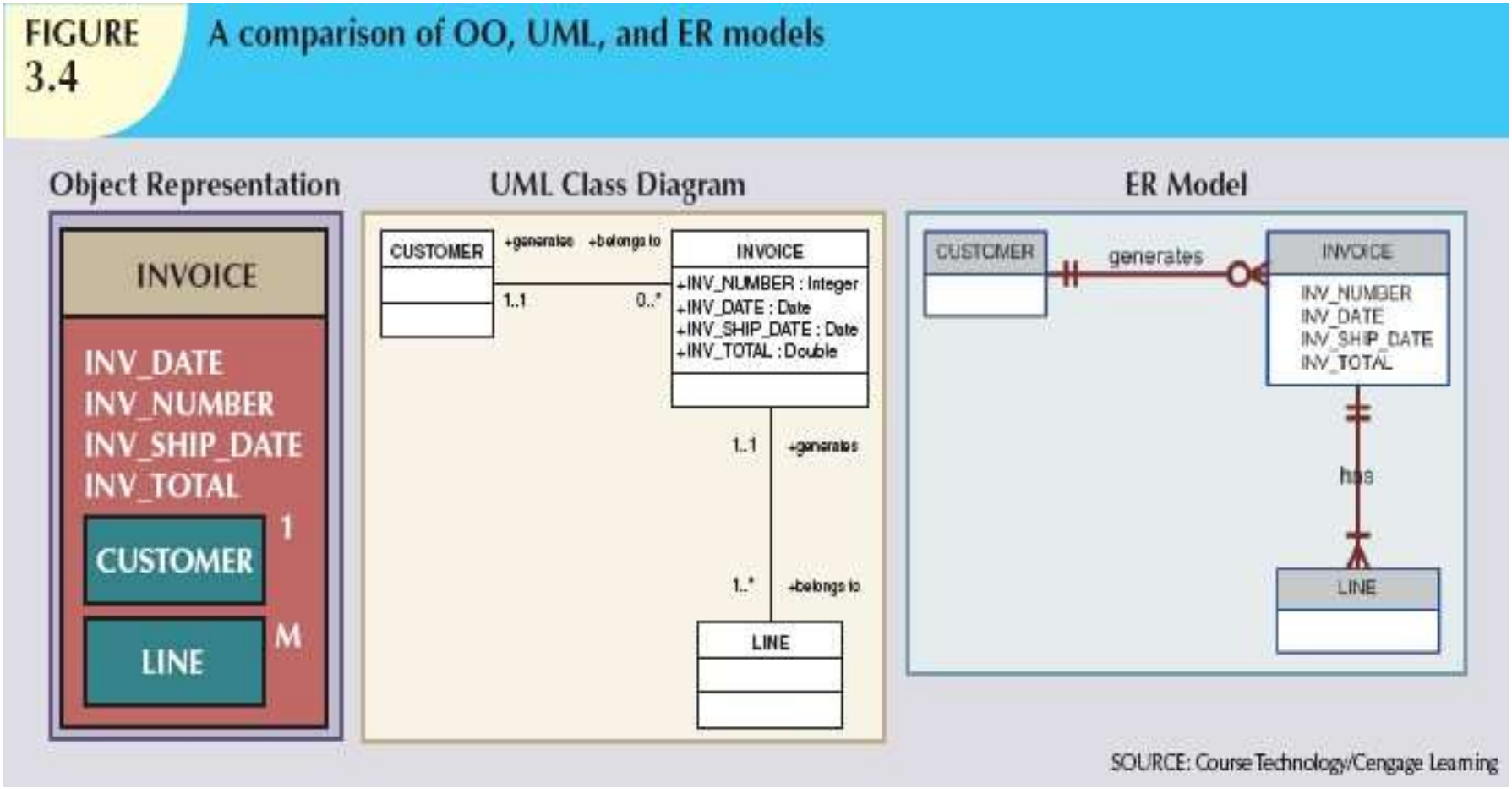
Object-Oriented (OO) Model

- Data and relationships are contained in a single structure known as an ***object***
- An object:
  - is described by its **factual** content (just like entity)
  - Contains all **operations** that can be performed
- Unlike entity, object includes information about
  - relationships between the facts within the object
  - its relationship with other objects
- OODM (object-oriented data model) is the basis for OODBMS

Object-Oriented (OO) Model

- *Attributes* describe the properties of an object
- *Objects* that share similar characteristics are grouped in classes
- *Classes* are organized in a class hierarchy
- *Inheritance*: object inherits methods and attributes of parent class
- UML based on OO concepts that describe diagrams and symbols

Object-Oriented (OO) Model



Emerging Data Models: Big Data and NoSQL

- Big Data
  - Find new and better ways to manage large amounts of Web-generated data and derive business insight from it
  - Relational approach does not always match the needs of organizations with Big Data challenges

- NoSQL databases
  - Not based on the relational model, hence the name NoSQL
  - Supports distributed database architectures
  - Provides high scalability, high availability, and fault tolerance
  - Supports very large amounts of sparse data

Relational Model – Table and Their Characteristics

Student File

Id No.	Name	Faculty	Major
1100	Ali	FIT	SE
1200	Bobby	FIT	MIS
1300	Clement	FCM	MM
1400	David	FCM	MM
1500	Evelyn	FOE	CE

Attribute
Field
Column

Entity Occurrence
Record
Row

Entity
File
Table

Conceptual data modeling notation
File processing Notation
Relational Table Notation

TABLE 4.1

Characteristics of a Relational Table

1	A table is perceived as a two-dimensional structure composed of rows and columns.
2	Each table row (**tuple**) represents a single entity occurrence within the entity set.
3	Each table column represents an attribute, and each column has a distinct name.
4	Each intersection of a row and column represents a single data value.
5	All values in a column must conform to the same data format.
6	Each column has a specific range of values known as the **attribute domain**.
7	The order of the rows and columns is immaterial to the DBMS.
8	Each table must have an attribute or combination of attributes that uniquely identifies each row.

FIGURE 3.1 STUDENT table attribute values

Study this table carefully

Table name: STUDENT **Database name: Ch03_TinyCollege**

STU_NUM	STU_LNAME	STU_FNAME	STU_INIT	STU_DOB	STU_HRS	STU_CLASS	STU_GPA	STU_TRANSFER	DEPT_CODE	STU_PHONE	PROF_NUM
321452	Bowser	William	C	12-Feb-1975	42	So	2.84	No	BIOL	2134	205
324257	Smithson	Anne	K	15-Nov-1981	81	Jr	3.27	Yes	CIS	2256	222
324258	Brewer	Juliette		23-Aug-1969	36	So	2.26	Yes	ACCT	2256	228
324269	Oblonski	Walter	H	16-Sep-1976	66	Jr	3.09	No	CIS	2114	222
324273	Smith	John	D	30-Dec-1958	102	Sr	2.11	Yes	ENGL	2231	199
324274	Katinga	Raphael	P	21-Oct-1979	114	Sr	3.15	No	ACCT	2267	228
324291	Robertson	Gerald	T	08-Apr-1973	120	Sr	3.87	No	EDU	2267	311
324299	Smith	John	B	30-Nov-1986	15	Fr	2.92	No	ACCT	2315	230

STU_NUM = Student number
STU_LNAME = Student last name
STU_FNAME = Student first name
STU_INIT = Student middle initial
STU_DOB = Student date of birth
STU_HRS = Credit hours earned
STU_CLASS = Student classification
STU_GPA = Grade point average
STU_TRANSFER = Student transferred from another institution
DEPT_CODE = Department code
STU_PHONE = 4-digit campus phone extension
PROF_NUM = Number of the professor who is the student's advisor

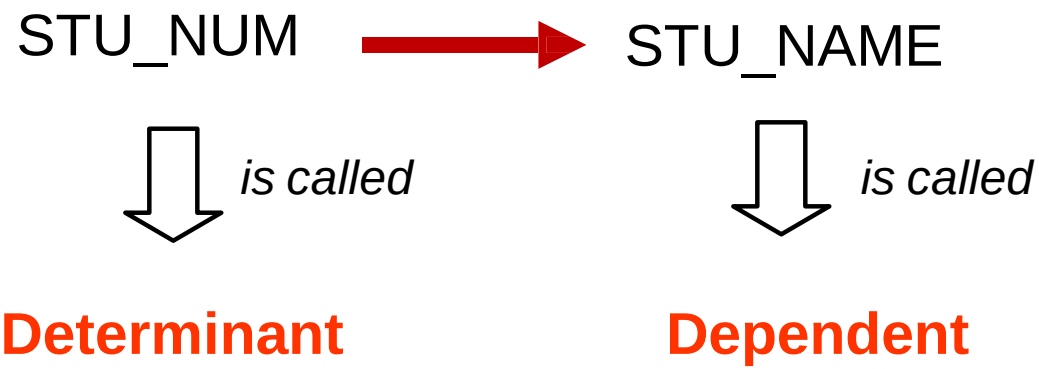
Primary key – is an attribute that uniquely identifies any given row or an entity

Keys

- Key ensures that each row in a table is *uniquely identifiable*
- ***Primary key*** – an attribute that uniquely identifies any given row (e.g., STU_Num)
- Key is based on the concept of determination
  - If you know the value of attribute A, you can determine the value of attribute B
  - E.g., STU_NUM determines STU_NAME, STU_DOB, etc.

Functional Dependence

- The value of *one* or *more* attributes determines the value of ***one*** or ***more*** other attributes
- Notation:



- ***Keys*** *are basically the determinants in functional dependencies*

Keys

- **Composite Key**
  - a key that is *composed of more than one* attribute

(STU_LNAME, STU_FNAME, STU_INIT, STU_PHONE) → STU_HRS

- **Superkey**
  - a key that can uniquely identify any row in the table
    - STU_NUM
    - (STU_LNAME,STUFNAME,STU_INIT,STU_PHONE)

Keys

- **Candidate Key**
  - It is a ***minimal superkey*** that is superkey without any unnecessary attributes
    - STU_NUM
    - (STU_LNAME,STU_FNAME,STU_INIT,STU_PHONE)
    - **(STU_NUM, STU_LNAME)** ✘
- **Secondary Key**
  - A key that is used strictly for data retrieval purposes
    - Customer's phone number

Keys

- **Foreign key**
  - is the primary key of one table that has been placed into another table to create a common attribute

Table name: PRODUCT **Database name: Ch03_SaleCo**
Primary key: PROD_CODE
Foreign key: VEND_CODE

PROD_CODE	PROD_DESCRIPT	PROD_PRICE	PROD_ON_HAND	VEND_CODE
001278-AB	Claw hammer	12.95	23	232
123-21UUY	Houselite chain saw, 16-in. bar	189.99	4	235
QER-34256	Sledge hammer, 16-lb. head	18.63	6	231
SRE-657UG	Rat-tail file	2.99	15	232
ZZX/3245Q	Steel tape, 12-ft. length	6.79	8	235

link

Table name: VENDOR
Primary key: VEND_CODE
Foreign key: none

VEND_CODE	VEND_CONTACT	VEND_AREACODE	VEND_PHONE
230	Shelly K. Smithson	608	555-1234
231	James Johnson	615	123-4536
232	Annelise Crystall	608	224-2134
233	Candice Wallace	904	342-6567
234	Arthur Jones	615	123-3324
235	Henry Ortozo	615	899-3425

Summary

TABLE 3.3 Relational Database Keys

KEY TYPE	DEFINITION
Superkey	An attribute (or combination of attributes) that uniquely identifies each row in a table.
Candidate key	A minimal (irreducible) superkey. A superkey that does not contain a subset of attributes that is itself a superkey.
Primary key	A candidate key selected to uniquely identify all other attribute values in any given row. Cannot contain null entries.
Secondary key	An attribute (or combination of attributes) used strictly for data retrieval purposes.
Foreign key	An attribute (or combination of attributes) in one table whose values must either match the primary key in another table or be null.

Integrity Rules

- Many RDBMs enforce integrity rules automatically
- Application design must conform to the integrity rules:
  - entity integrity
  - referential integrity

Integrity Rules (cont'd.)

- **Entity integrity**
  - is the condition in which each row in the table has its own unique identity
  - To ensure entity integrity, the primary key has two requirements:
    - All of the values in the primary key must be **unique**
    - No **null** value (i.e., absence of any data value) is allowed in any part of the primary key
      - Null values can create logical problems when tables are linked

Integrity Rules (cont'd.)

- **Referential integrity**
  - is the condition in which every foreign key must either be:
    - null (providing that it does not form part of its table's primary key)

    or
    - an entry which matches the primary key value in the table to which it is related

Integrity Rules (cont'd.)

Table name: CUSTOMER
Primary key: CUS_CODE
Foreign key: AGENT_CODE

Database name: Ch03_InsureCo

CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_AREACODE	CUS_PHONE	CUS_INSURE_TYPE	CUS_INSURE_AMT	CUS_RENEW_DATE	AGENT_CODE
10010	Ramas	Alfred	A	615	844-2573	T1	100.00	05-Apr-2010	502
10011	Dunne	Leona	K	713	894-1238	T1	250.00	16-Jun-2010	501
10012	Smith	Kathy	W	615	894-2285	S2	150.00	29-Jan-2011	502
10013	Olowski	Paul	F	615	894-2180	S1	300.00	14-Oct-2010	
10014	Orlando	Myron		615	222-1672	T1	100.00	28-Dec-2010	501
10015	O'Brian	Amy	B	713	442-3381	T2	850.00	22-Sep-2010	503
10016	Brown	James	G	615	297-1228	S1	120.00	25-Mar-2011	502
10017	Williams	George		615	290-2556	S1	250.00	17-Jul-2010	503
10018	Farriss	Anne	G	713	382-7185	T2	100.00	03-Dec-2010	501
10019	Smith	Olette	K	615	297-3809	S2	500.00	14-Mar-2011	503

Table name: AGENT
Primary key: AGENT_CODE
Foreign key: none

AGENT_CODE	AGENT_AREACODE	AGENT_PHONE	AGENT_LNAME	AGENT_YTD_SLS
501	713	228-1249	Alby	132735.75
502	615	882-1244	Hahn	138967.35
503	615	123-5589	Okon	127093.45

Relationships within a Relational Database

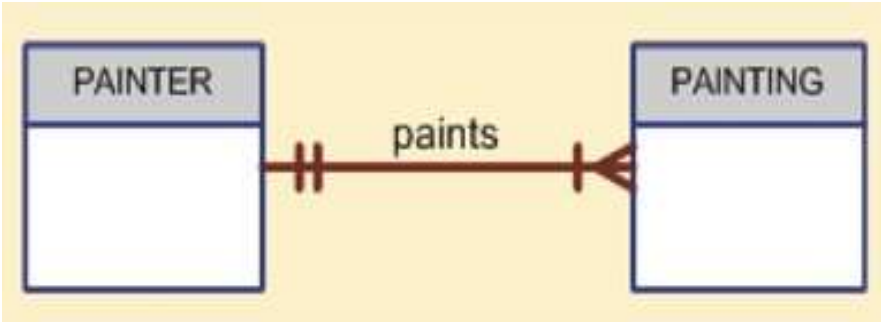
- **1:M relationship**
- **1:1 relationship**
- **M:N relationships**

The 1:M Relationship

- The norm in any relational database design
  - 1:M relationship between painter and painting

Chen's model

Crow's foot model



This course follows Crow's foot model

The 1:M Relationship

FIGURE 3.19 The implemented 1:M relationship between PAINTER and PAINTING

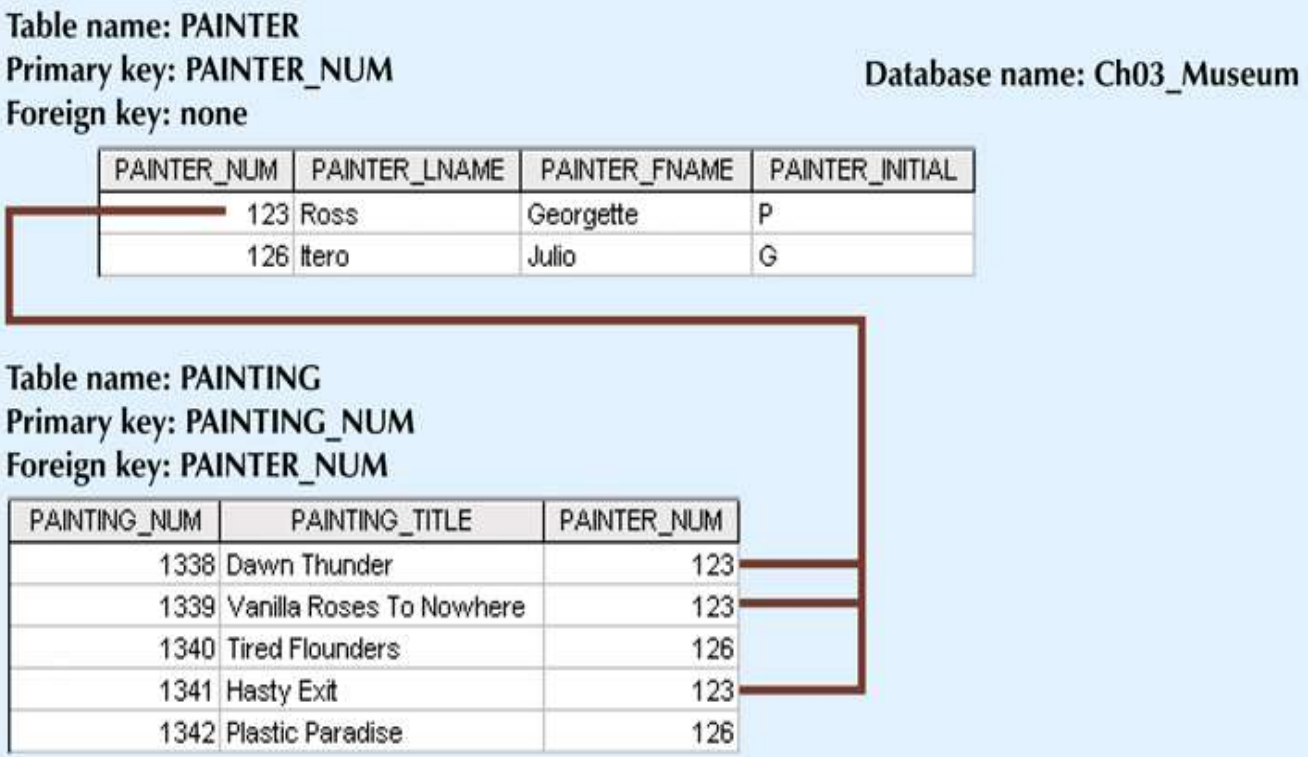


Table name: PAINTER
Primary key: PAINTER_NUM
Foreign key: none

Database name: Ch03_Museum

PAINTER_NUM	PAINTER_LNAME	PAINTER_FNAME	PAINTER_INITIAL
123	Ross	Georgette	P
126	Itero	Julio	G

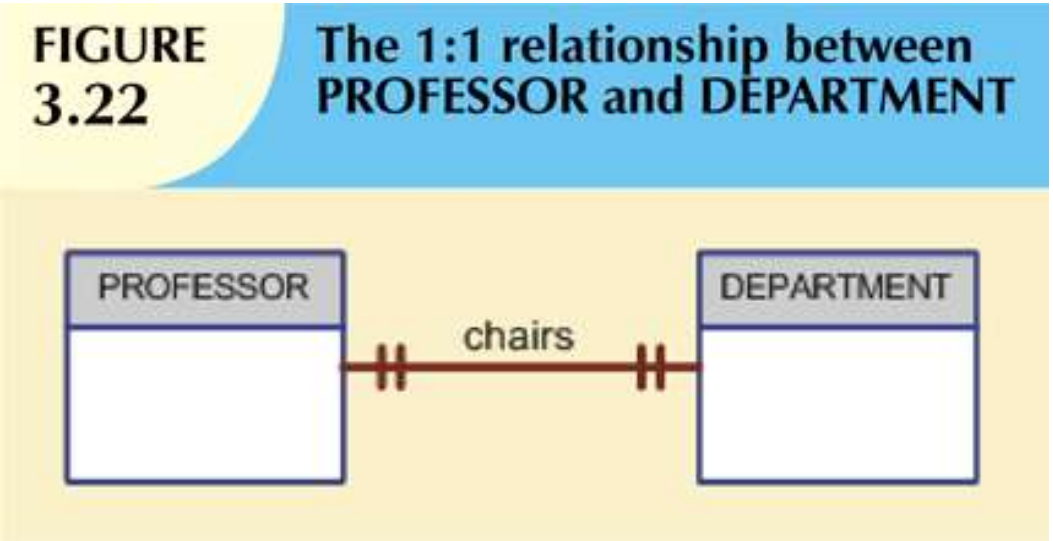
Table name: PAINTING
Primary key: PAINTING_NUM
Foreign key: PAINTER_NUM

PAINTING_NUM	PAINTING_TITLE	PAINTER_NUM
1338	Dawn Thunder	123
1339	Vanilla Roses To Nowhere	123
1340	Tired Flounders	126
1341	Hasty Exit	123
1342	Plastic Paradise	126

The 1:1 Relationship

- One entity related to only one other entity, and vice versa
- Could indicate that:
  - two entities actually belong in the same table
  - entity components were not defined properly

The 1:1 Relationship



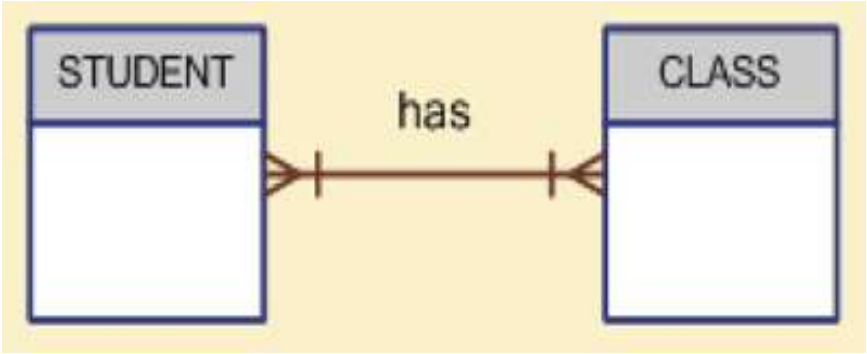
The M:N Relationships

- **Many-to-many (M:N) relationships** - associations among two or more entities in which one occurrence of an entity is associated with many occurrences of a related entity
- To avoid problems inherent to M:N relationships
  - need to break M:N relationships produce two sets of 1:M relationships using a composite entity

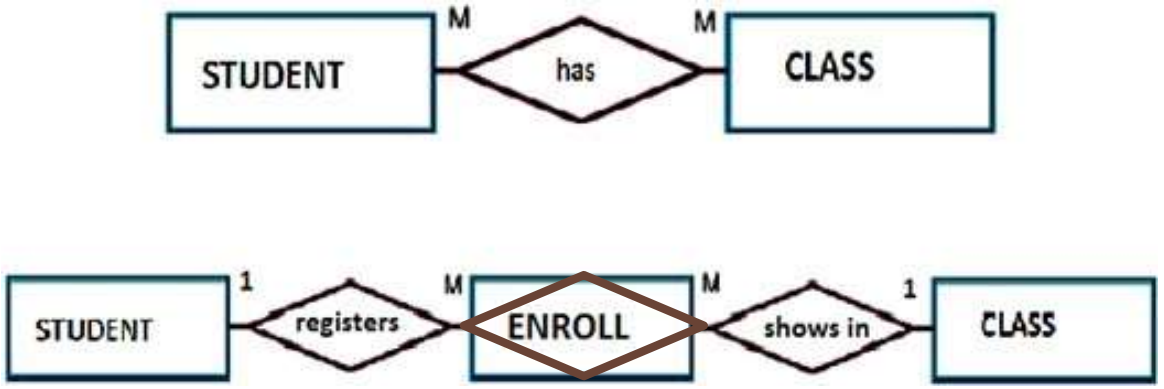
The M:N Relationship

- Chen's model

- Crow's foot model



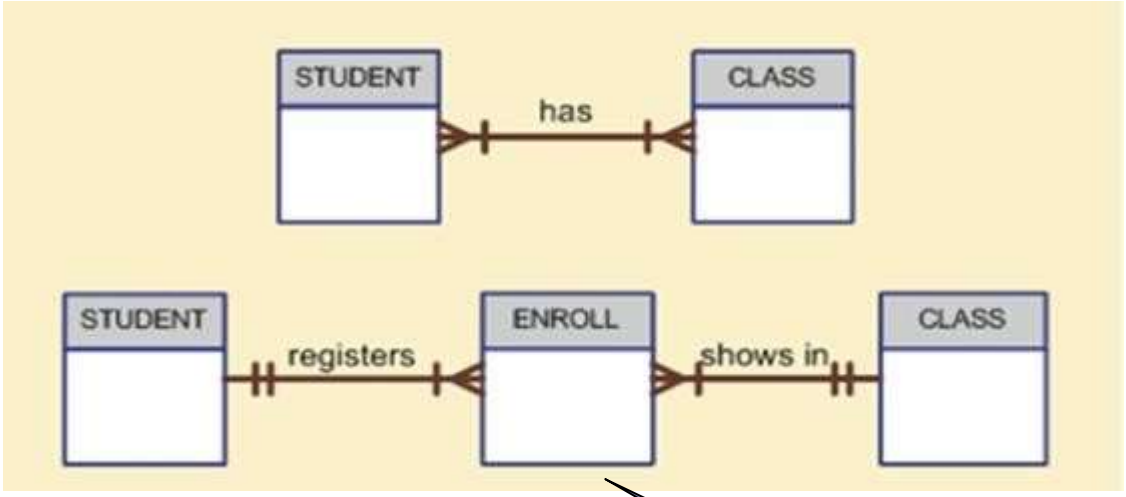
Composite Entity for M:N Relationship



Chen's model

Composite Entity

Composite Entity for M:N Relationship



Crow's foot model

Composite Entity

Converting M:N into 1:M Relationships

Table name: STUDENT
Primary key: STU_NUM
Foreign key: none

Database name: Ch03_CollegeTry2

STU_NUM	STU_LNAME
321452	Bowser
324257	Smithson

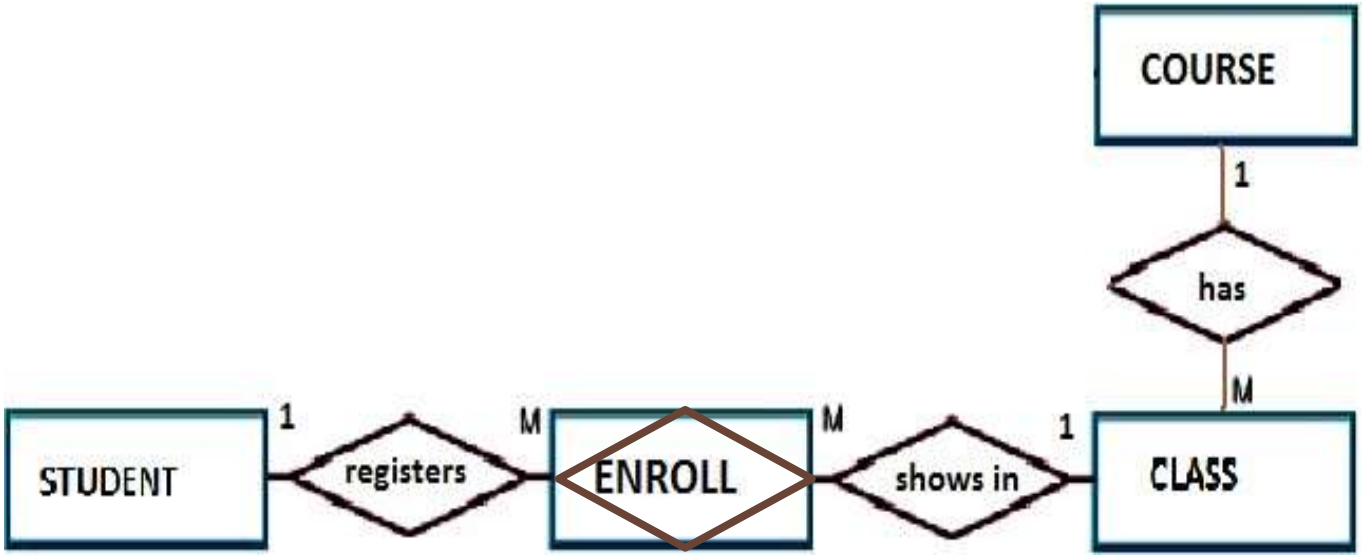
Table name: ENROLL
Primary key: CLASS_CODE + STU_NUM
Foreign key: CLASS_CODE, STU_NUM

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
Primary key: CLASS_CODE
Foreign key: CRS_CODE

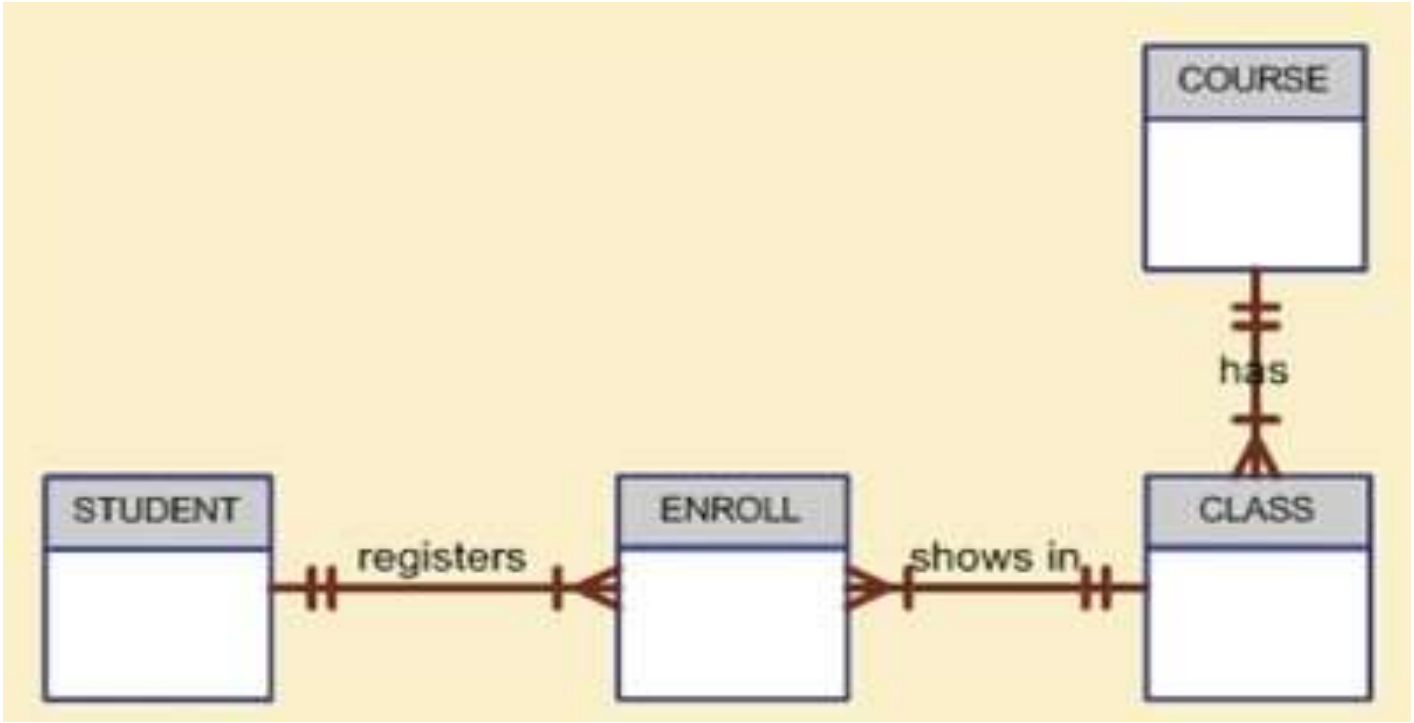
CLASS_CODE	CRS_CODE	CLASS_SECTION	CLASS_TIME	CLASS_ROOM	PROF_NUM
10014	ACCT-211	3	TTh 2:30-3:45 p.m.	BUS252	342
10018	CIS-220	2	MWF 9:00-9:50 a.m.	KLR211	114
10021	QM-261	1	MWF 8:00-8:50 a.m.	KLR200	114

The Expanded ER Model



Chen's model

The Expanded ER Model



Crow's foot model

Relational Set Operators

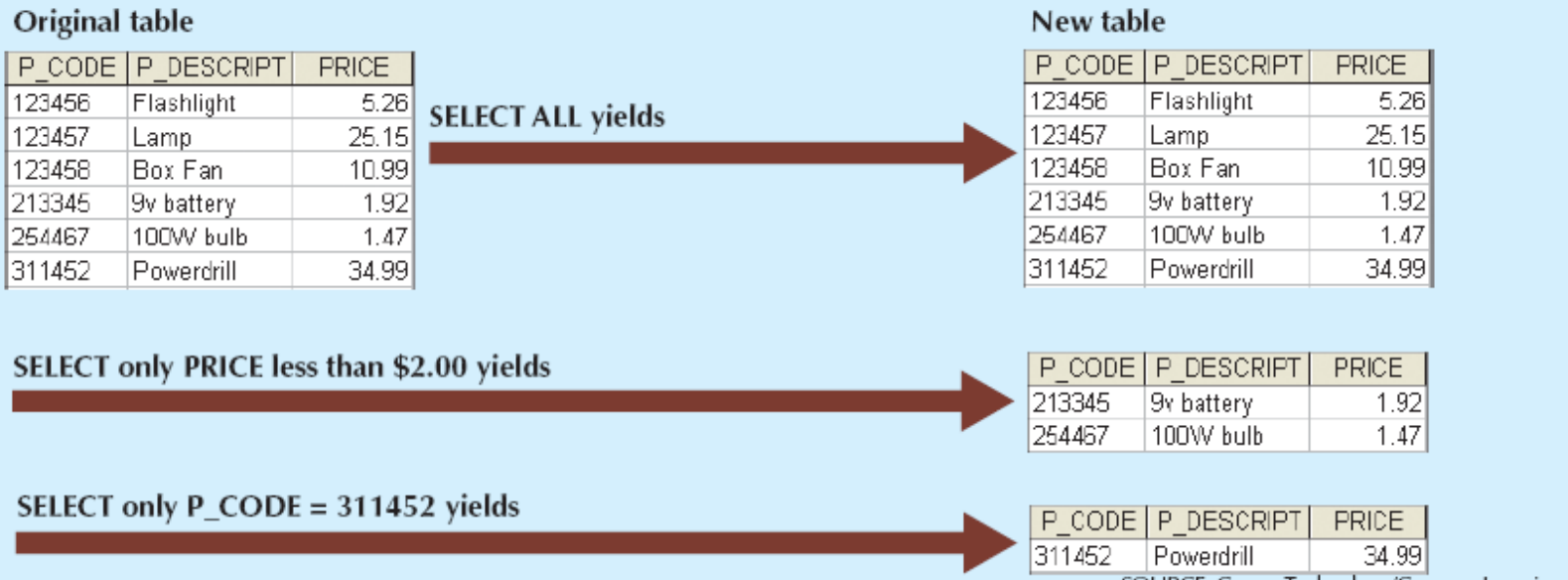
- Relational algebra
  - Defines theoretical way of manipulating table contents using relational operators
  - Use of relational algebra operators on existing relations produces new relations:

- **SELECT**
- **PROJECT**
- **UNION**
- **INTERSECT**
- **DIFFERENCE**
- **JOIN**
- **PRODUCT**
- **DIVIDE**

Select

- Yields values for all rows found in a table.
- It yields a horizontal subset of a table.

FIGURE 4.4 SELECT

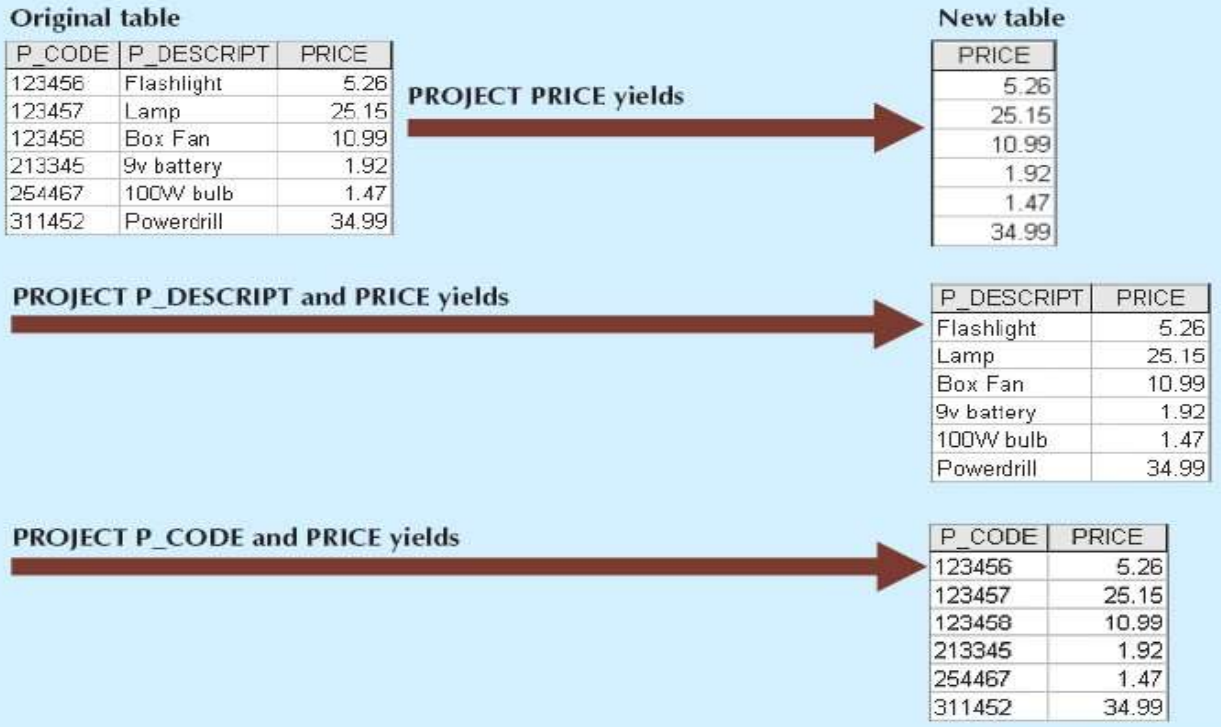


SOURCE: Course Technology/Cengage Learning

Project

- Produces a list of all values for selected attributes
- It yields a vertical subset of a table.

FIGURE 4.5 **PROJECT**



SOURCE: Course Technology/Cengage Learning

Union

- Combines all rows from two tables.
- The tables must be union compatible (same attribute characteristics).

FIGURE 4.6 UNION

P_CODE	P_DESCRIPT	PRICE
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	10.99
213345	9v battery	1.92
254467	100W bulb	1.47
311452	Powerdrill	34.99

UNION

P_CODE	P_DESCRIPT	PRICE
345678	Microwave	160.00
345679	Dishwasher	500.00
123458	Box Fan	10.99

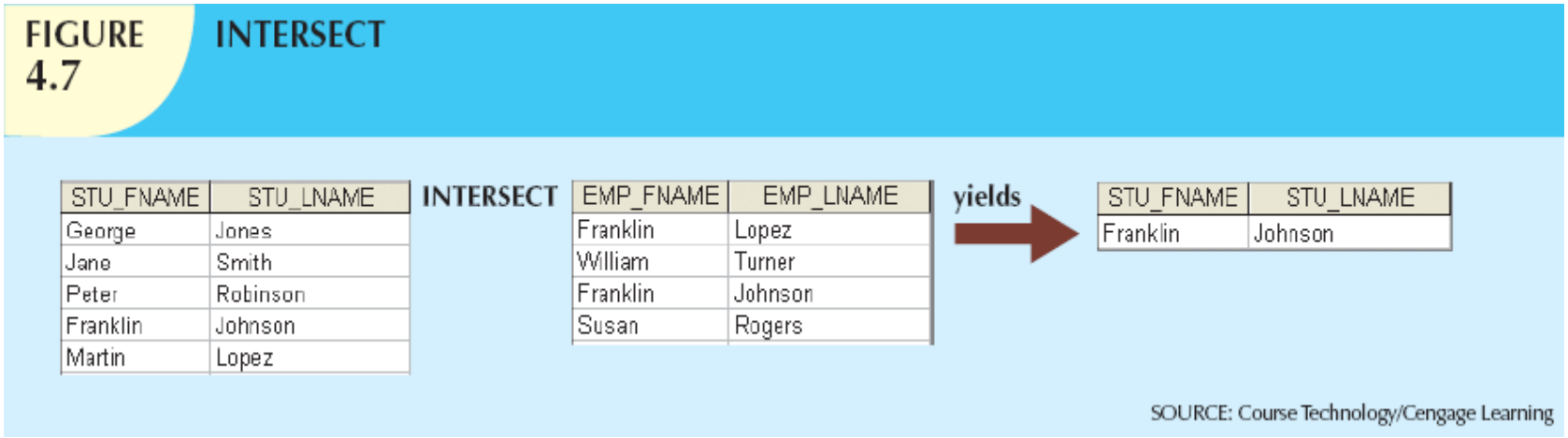
yields

P_CODE	P_DESCRIPT	PRICE
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	10.99
213345	9v battery	1.92
254467	100W bulb	1.47
311452	Powerdrill	34.99
345678	Microwave	160
345679	Dishwasher	500

SOURCE: Course Technology/Cengage Learning

Intersect

- Produces a listing that contains only the rows that appear in both tables.
- The tables must be union compatible (same attribute characteristics).



Difference

- Yields all row in one table that are not found in the other table, i.e. it subtracts one table from the other.
- The table must be union compatible.

FIGURE 4.8 DIFFERENCE

STU_FNAME	STU_LNAME
George	Jones
Jane	Smith
Peter	Robinson
Franklin	Johnson
Martin	Lopez

DIFFERENCE

EMP_FNAME	EMP_LNAME
Franklin	Lopez
William	Turner
Franklin	Johnson
Susan	Rogers

yields

STU_FNAME	STU_LNAME
George	Jones
Jane	Smith
Peter	Robinson
Martin	Lopez

SOURCE: Course Technology/Cengage Learning

Product

- Produces a list of all possible pairs of rows from two tables.

FIGURE 4.9 PRODUCT

P_CODE	P_DESCRIPT	PRICE
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	10.99
213345	9v battery	1.92
254467	100W bulb	1.47
311452	Powerdrill	34.99

PRODUCT

STORE	AISLE	SHELF
23	W	5
24	K	9
25	Z	6

yields

P_CODE	P_DESCRIPT	PRICE	STORE	AISLE	SHELF
123456	Flashlight	5.26	23	W	5
123456	Flashlight	5.26	24	K	9
123456	Flashlight	5.26	25	Z	6
123457	Lamp	25.15	23	W	5
123457	Lamp	25.15	24	K	9
123457	Lamp	25.15	25	Z	6
123458	Box Fan	10.99	23	W	5
123458	Box Fan	10.99	24	K	9
123458	Box Fan	10.99	25	Z	6
213345	9v battery	1.92	23	W	5
213345	9v battery	1.92	24	K	9
213345	9v battery	1.92	25	Z	6
311452	Powerdrill	34.99	23	W	5
311452	Powerdrill	34.99	24	K	9
311452	Powerdrill	34.99	25	Z	6
254467	100W bulb	1.47	23	W	5
254467	100W bulb	1.47	24	K	9
254467	100W bulb	1.47	25	Z	6

SOURCE: Course Technology/Cengage Learning

Join

- **Natural Join**
  - Links tables by selecting rows with common values in common attribute(s)

FIGURE 3.11 Two tables that will be used in join illustrations

Table name: CUSTOMER

CUS_CODE	CUS_LNAME	CUS_ZIP	AGENT_CODE
1132445	Walker	32145	231
1217782	Adares	32145	125
1312243	Rakowski	34129	167
1321242	Rodriguez	37134	125
1542311	Smithson	37134	421
1657399	Vanloo	32145	231

Table name: AGENT

AGENT_CODE	AGENT_PHONE
125	6152439887
167	6153426778
231	6152431124
333	9041234445

Join

- 1st step: **Product**

FIGURE 3.12 Natural join, Step 1: PRODUCT

CUS_CODE	CUS_LNAME	CUS_ZIP	CUSTOMER.AGENT_CODE	AGENT.AGENT_CODE	AGENT_PHONE
1132445	Walker	32145	231	125	6152439887
1132445	Walker	32145	231	167	6153426778
1132445	Walker	32145	231	231	6152431124
1132445	Walker	32145	231	333	9041234445
1217782	Adares	32145	125	125	6152439887
1217782	Adares	32145	125	167	6153426778
1217782	Adares	32145	125	231	6152431124
1217782	Adares	32145	125	333	9041234445
1312243	Rakowski	34129	167	125	6152439887
1312243	Rakowski	34129	167	167	6153426778
1312243	Rakowski	34129	167	231	6152431124
1312243	Rakowski	34129	167	333	9041234445
1321242	Rodriguez	37134	125	125	6152439887
1321242	Rodriguez	37134	125	167	6153426778
1321242	Rodriguez	37134	125	231	6152431124
1321242	Rodriguez	37134	125	333	9041234445
1542311	Smithson	37134	421	125	6152439887
1542311	Smithson	37134	421	167	6153426778
1542311	Smithson	37134	421	231	6152431124
1542311	Smithson	37134	421	333	9041234445
1657399	Vanloo	32145	231	125	6152439887
1657399	Vanloo	32145	231	167	6153426778
1657399	Vanloo	32145	231	231	6152431124
1657399	Vanloo	32145	231	333	9041234445

Join

- 2nd step: **Select**
- Select operation performed on the output of the 1st step, shows only the rows for which the common attribute values match.

FIGURE 3.13 Natural join, Step 2: SELECT

CUS_CODE	CUS_LNAME	CUS_ZIP	CUSTOMER.AGENT_CODE	AGENT.AGENT_CODE	AGENT_PHONE
1217782	Adares	32145	125	125	6152439887
1321242	Rodriguez	37134	125	125	6152439887
1312243	Rakowski	34129	167	167	6153426778
1132445	Walker	32145	231	231	6152431124
1657399	Vanloo	32145	231	231	6152431124

Join

- 3rd step: **Project**
- Performed to yield a single copy of each attribute, thereby **eliminating** the duplicate columns.

FIGURE 3.14 Natural join, Step 3: PROJECT

CUS_CODE	CUS_LNAME	CUS_ZIP	AGENT_CODE	AGENT_PHONE
1217782	Adares	32145	125	6152439887
1321242	Rodriguez	37134	125	6152439887
1312243	Rakowski	34129	167	6153426778
1132445	Walker	32145	231	6152431124
1657399	Vanloo	32145	231	6152431124

Join

- **Outer join**
  - Matched pairs are retained and any unmatched values in other table are left null
  - Two types:
    - Left outer join
    - Right outer join

Example - Left Outer Join

Table name: CUSTOMER

CUS_CODE	CUS_LNAME	CUS_ZIP	AGENT_CODE
1132445	Walker	32145	231
1217782	Adares	32145	125
1312243	Rakowski	34129	167
1321242	Rodriguez	37134	125
1542311	Smithson	37134	421
1657399	Vanloo	32145	231

Table name: AGENT

AGENT_CODE	AGENT_PHONE
125	6152439887
167	6153426778
231	6152431124
333	9041234445

FIGURE 3.15 **Left outer join**

CUS_CODE	CUS_LNAME	CUS_ZIP	AGENT_CODE	AGENT_PHONE
1217782	Adares	32145	125	6152439887
1321242	Rodriguez	37134	125	6152439887
1312243	Rakowski	34129	167	6153426778
1132445	Walker	32145	231	6152431124
1657399	Vanloo	32145	231	6152431124
1542311	Smithson	37134	421	

Example - Right Outer Join

Table name: CUSTOMER

CUS_CODE	CUS_LNAME	CUS_ZIP	AGENT_CODE
1132445	Walker	32145	231
1217782	Adares	32145	125
1312243	Rakowski	34129	167
1321242	Rodriguez	37134	125
1542311	Smithson	37134	421
1657399	Vanloo	32145	231

Table name: AGENT

AGENT_CODE	AGENT_PHONE
125	6152439887
167	6153426778
231	6152431124
333	9041234445

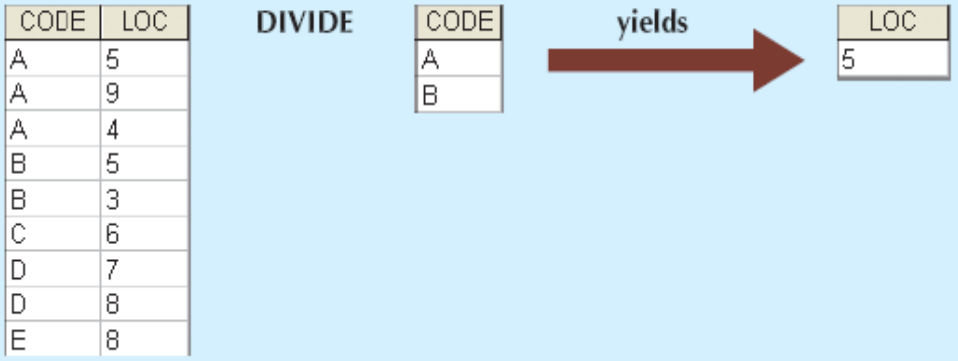
FIGURE 3.16 **Right outer join**

CUS_CODE	CUS_LNAME	CUS_ZIP	AGENT_CODE	AGENT_PHONE
1217782	Adares	32145	125	6152439887
1321242	Rodriguez	37134	125	6152439887
1312243	Rakowski	34129	167	6153426778
1132445	Walker	32145	231	6152431124
1657399	Vanloo	32145	231	6152431124
			333	9041234445

Divide

- Requires the use of one-single column table and one two-column table.

FIGURE 4.16 DIVIDE



SOURCE: Course Technology/Cengage Learning