ALLEGHENY COLLEGE

Principles of Database Systems
CS 312
Tables and Schemas

Spring 2019

Oliver Bonham-Carter

Database Development

Design

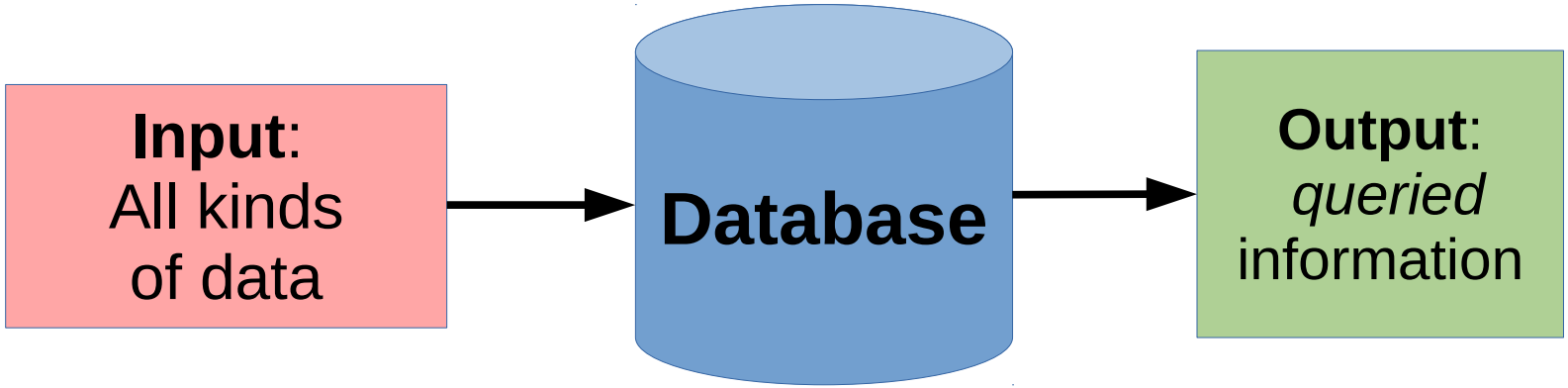
Development

Database Development

How do we *think* about designing a database?

- Steps for development:
  - Analysis of information
    - Purpose of the db
  - Design of system
  - Construction
    - (programming)
  - Data Entry
  - Information Retrieval

With some help from Uncle Jeff Bezos (CEO of Amazon), book seller and computer guru extraordinaire...



ALLEGHENY COLLEGE

Let's Imagine: Our Client

Fantastic Books

The Client Says…

I WANT EVERYTHING!!!
I need data about… Customers!
Wholesalers! Sales! Purchases!
Employees! Receipts!
And reports! Lots of reports!
And profits!!!

FYI: Clients **never** know exactly what they need!

The Challenge:

We are to change the current "system"…

The Grand Ambition!

Fantastic Books

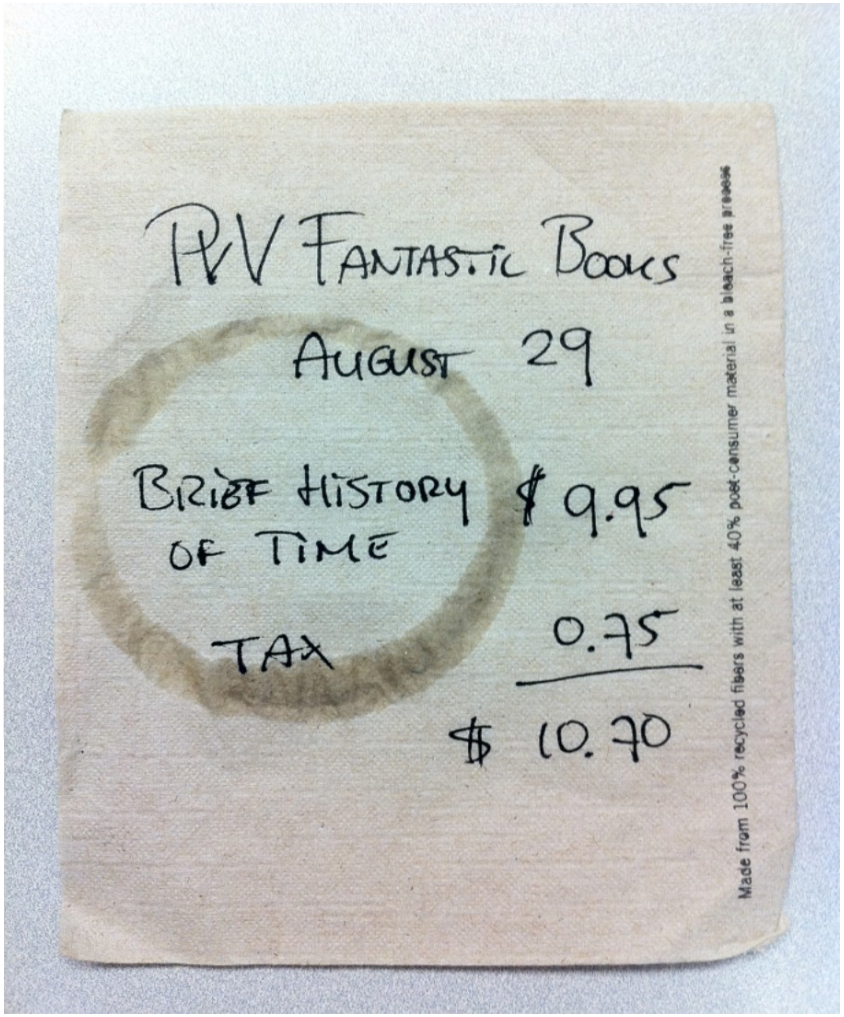
J. Smith
1234 PKI Road
Omaha, NE 68123

September 15, 2013

Zero History	1	$ 15.95
Anathem	1	8.90
The Hobbit	1	14.25
The Subtle Knife	1	7.30
		========
Total		$ 46.40
Tax		3.48
		========
Total Due		$ 49.88
Cash Payment		$ 49.88

Exchange only w/ receipt in 7 days.
Sign up for our newsletter
at pvv-fantasticbooks.com

The Sad Reality



What information do I have?

What data is available?
What is this database supposed to do?
What reports are important to make?
What kinds of queries?

Users?
Software?
Hardware?
How populated?
How queried?

Maintenance?
Centralized?
Networked?
Points of use?

Entities & Attributes *Version 1.0*

Customer: first name, last name, address, city, state, zip, phone number, e-mail address

Employee: first name, last name, address, city, state, zip, phone number, e-mail address

Book: ISBN, title, author(s), publisher, publication date, genre, number in stock, number on order

Sale: date, customer first name, last name, ISBN, title, quantity sold, total due, price paid, payment method, employee first name, last name

Purchase: date, wholesaler name, ISBN title, quantity bought, total due, amount paid,

Wholesaler: wholesaler name, address, city, state, zip, phone number, account number, website, e-mail address

How to differentiate rows, tables?

Entities & Attributes *Version 2.0*

Customer: customer-ID, first name, last name, address, city, state, zip, phone number, e-mail address

Employee: employee-ID, first name, last name, address, city, state, zip, phone number, e-mail address

Book: ISBN, title, author(s), publisher, publication date, genre, number in stock, number on order

Sale: sale-ID, date, customer first name, last name, ISBN, title, quantity sold, total due, price paid, payment method , employee first name, last name

Purchase: purchase-ID, date, wholesaler name, ISBN, title, quantity bought, total due, amount paid,

Wholesaler: wholesaler-ID, wholesaler name, address, city, state, zip, phone number, account number, website, e-mail address

What about linking your tables?

Entities & Attributes *Version 3.0*

Customer: **customer-ID**, first name, last name, address, city, state, zip, phone number, e-mail address

Employee: **employee-ID**, first name, last name, address, city, state, zip, phone number, e-mail address

Book: **ISBN**, title, author(s), publisher, publication date, genre, number in stock, number on order

Sale: sale-ID, date, **customer-ID**, **ISBN**, quantity sold, total due, price paid, payment method, **employee-ID**

Purchase: purchase-ID, date, **wholesaler-ID**, ISBN, quantity bought, total due, amount paid,

Wholesaler: **wholesaler-ID**, wholesaler name, address, city, state, zip, phone number, account number, website, e-mail address

Are you sure that there is no redundancy in your tables?

Entities & Attributes *Version 4.0*

Customer: customer-ID, first name, last name, address, city, state, zip, phone number, e-mail address

Employee: employee-ID, first name, last name, address, city, state, zip, phone number, e-mail address

Book: ISBN, title, publisher-ID, publication date, genre, number in stock, number on order

Sale: sale-ID, date, customer-ID, total due, price paid, payment method, employee-ID

Purchase: purchase-ID, date, wholesaler-ID, total due, amount paid,

Wholesaler: wholesaler-ID, wholesaler name, address, city, state, zip, phone number, account number, website, e-mail address

Publisher: Publisher-ID, publisher name, address, city, state, zip, phone number

Special case: connect Books to Authors

Author: Author-ID, first name, last name

Sale-Line: sale-ID, ISBN, quantity sold

Purchase-Line: Purchase-ID, ISBN, quantity bought

I like the connectivity.
Now, what about the ***normalization***?

ALLEGHENY COLLEGE

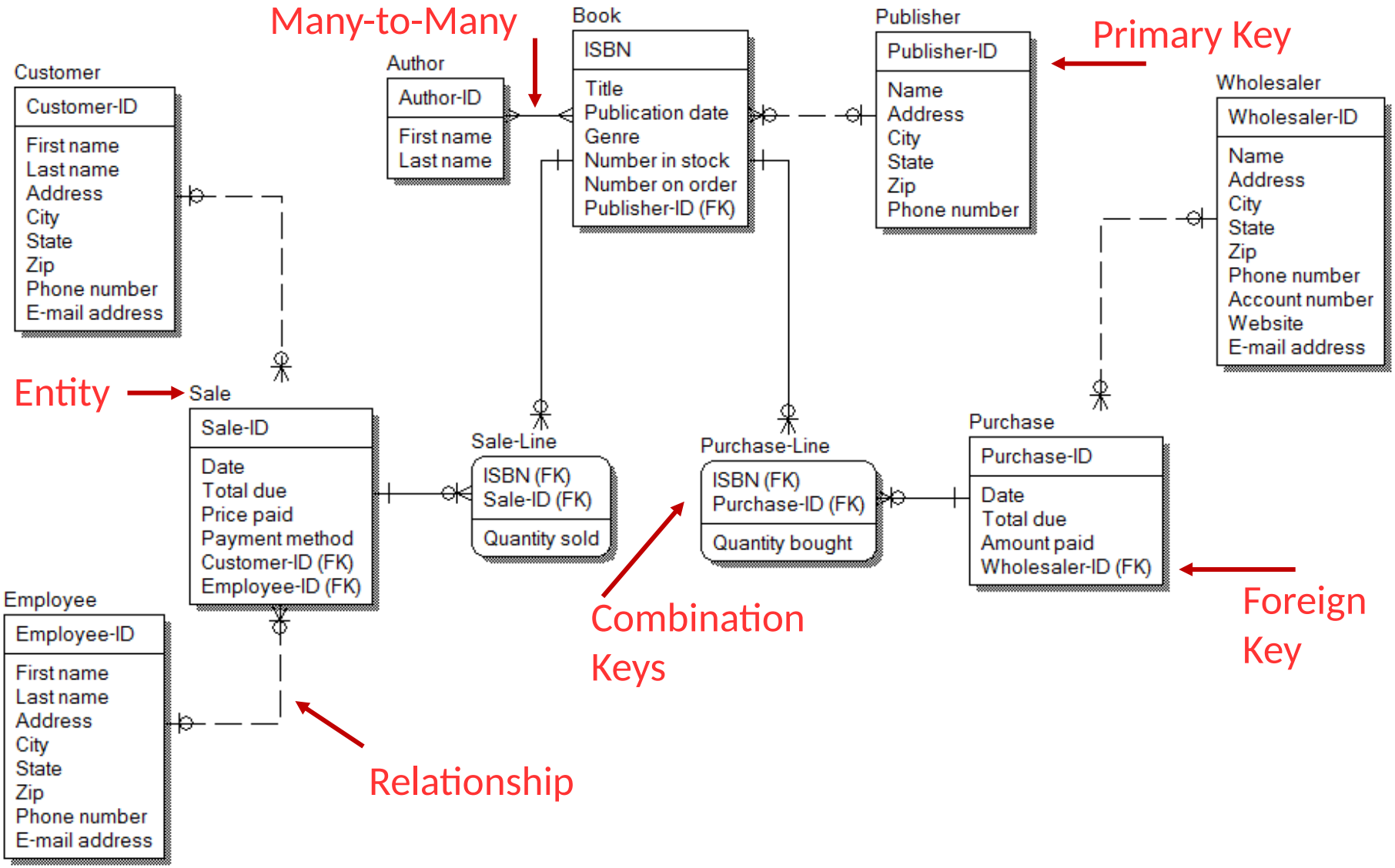
“Normalization” of a Base

***Normalization**:*
The process of organizing data in a database. This includes creating tables and establishing relationships between them according to rules designed both to protect the data and to make the database more flexible by eliminating redundancy and inconsistent dependency.

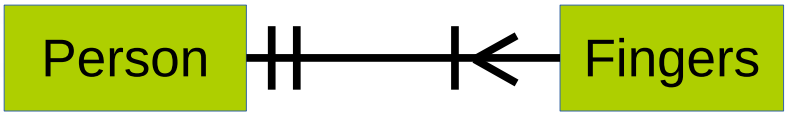
Inconsistant Dependancies

- Each of the fields in a table should have a *dependency* on the table (because it logically belongs there and nowhere else.)
- A sales representative (the person who visits the customer and generates sales) does interact with the customer and so this person's name belongs in the **Customers Table** but we only really need a link to this person in the **Representatives Table**.
- If the representative's email (or other information concerning representatives) appeared in the "**Customers Table**" table, then we would say that there was an "**inconsistent dependency**."

Entity Relationship Diagram

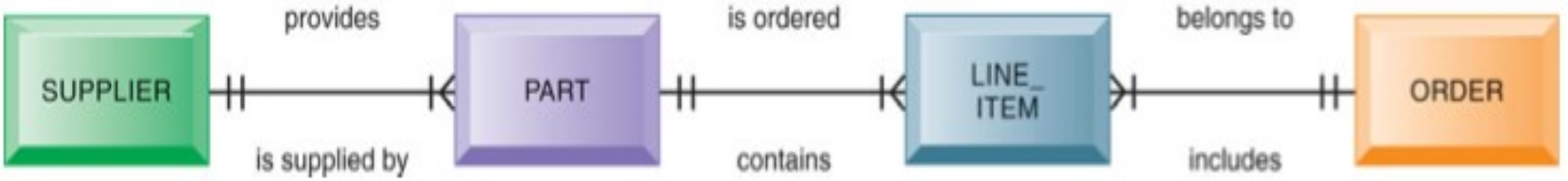


The *One to Many* Relationship



Each person has
many fingers,
But each finger belongs
to only one person.

FIGURE 6-11 AN ENTITY-RELATIONSHIP DIAGRAM



Note: Other books use double-strikes, *greater-than* and *less-than* notation to denote these relationships

The *One to Many* Relationship

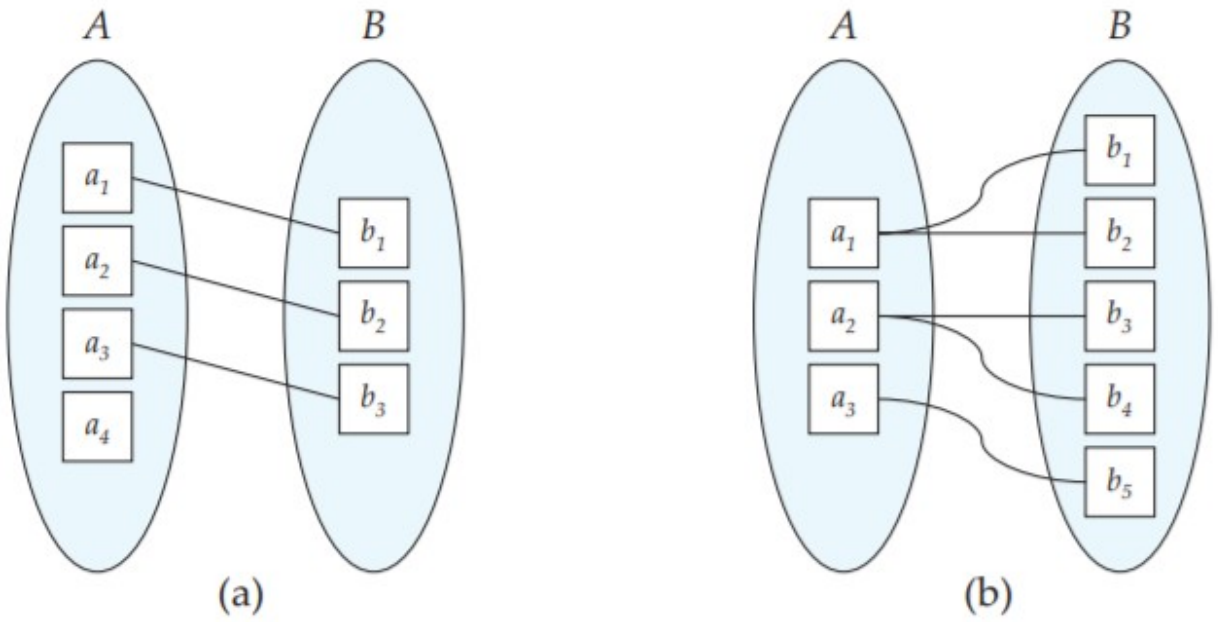


Figure 7.5 Mapping cardinalities. (a) One-to-one. (b) One-to-many.

- **One-to-one**. An entity in A is associated with at most one entity in B, and an entity in B is associated with at most one entity in A. (See Figure 7.5a.)
- **One-to-many**. An entity in A is associated with any number (zero or more) of entities in B. An entity in B, however, can be associated with at most one entity in A. (See Figure 7.5b.)

The Many to Many Relationship

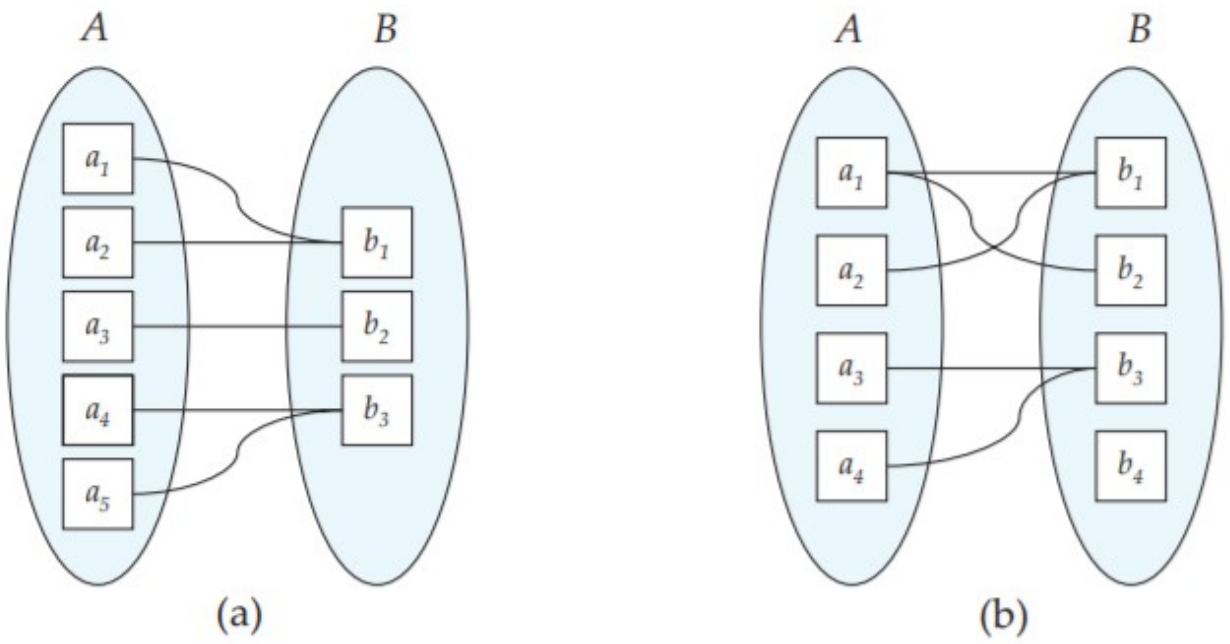


Figure 7.6 Mapping cardinalities. (a) Many-to-one. (b) Many-to-many.

- **Many-to-one**. An entity in A is associated with at most one entity in B. An entity in B, however, can be associated with any number (zero or more) of entities in A. (See Figure 7.6a.)
- **Many-to-many**. An entity in A is associated with any number (zero or more) of entities in B, and an entity in B is associated with any number (zero or more) of entities in A. (See Figure 7.6b.)

The Many to Many Relationship

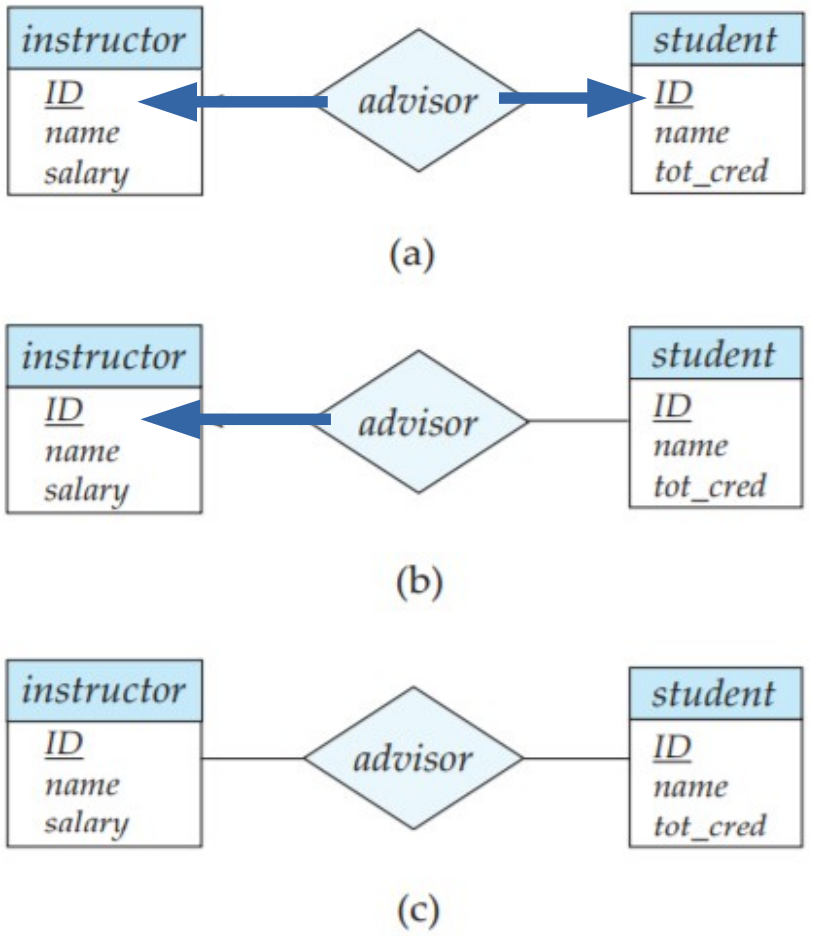


Figure 7.9 Relationships. (a) One-to-one. (b) One-to-many. (c) Many-to-many.

Note: The book uses arrows to denote these relationships

The Weak Entity Relationships

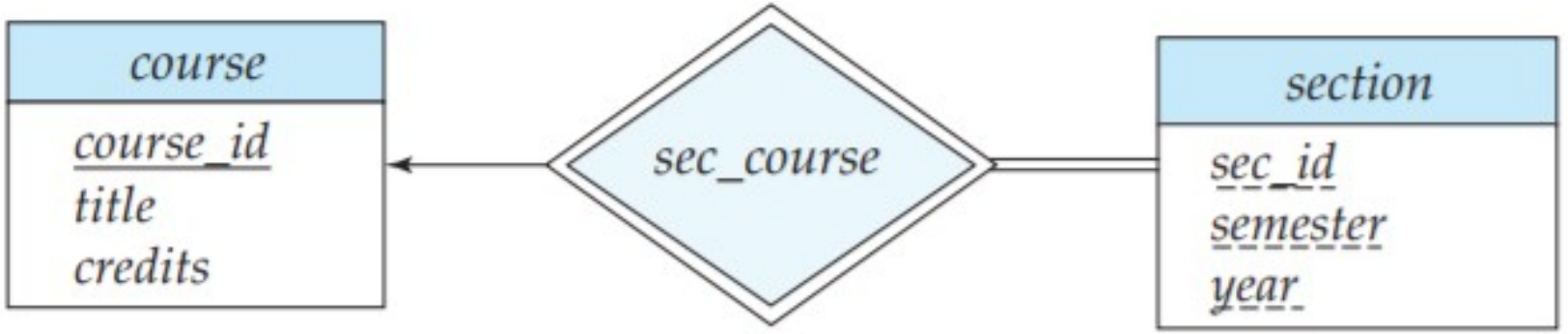
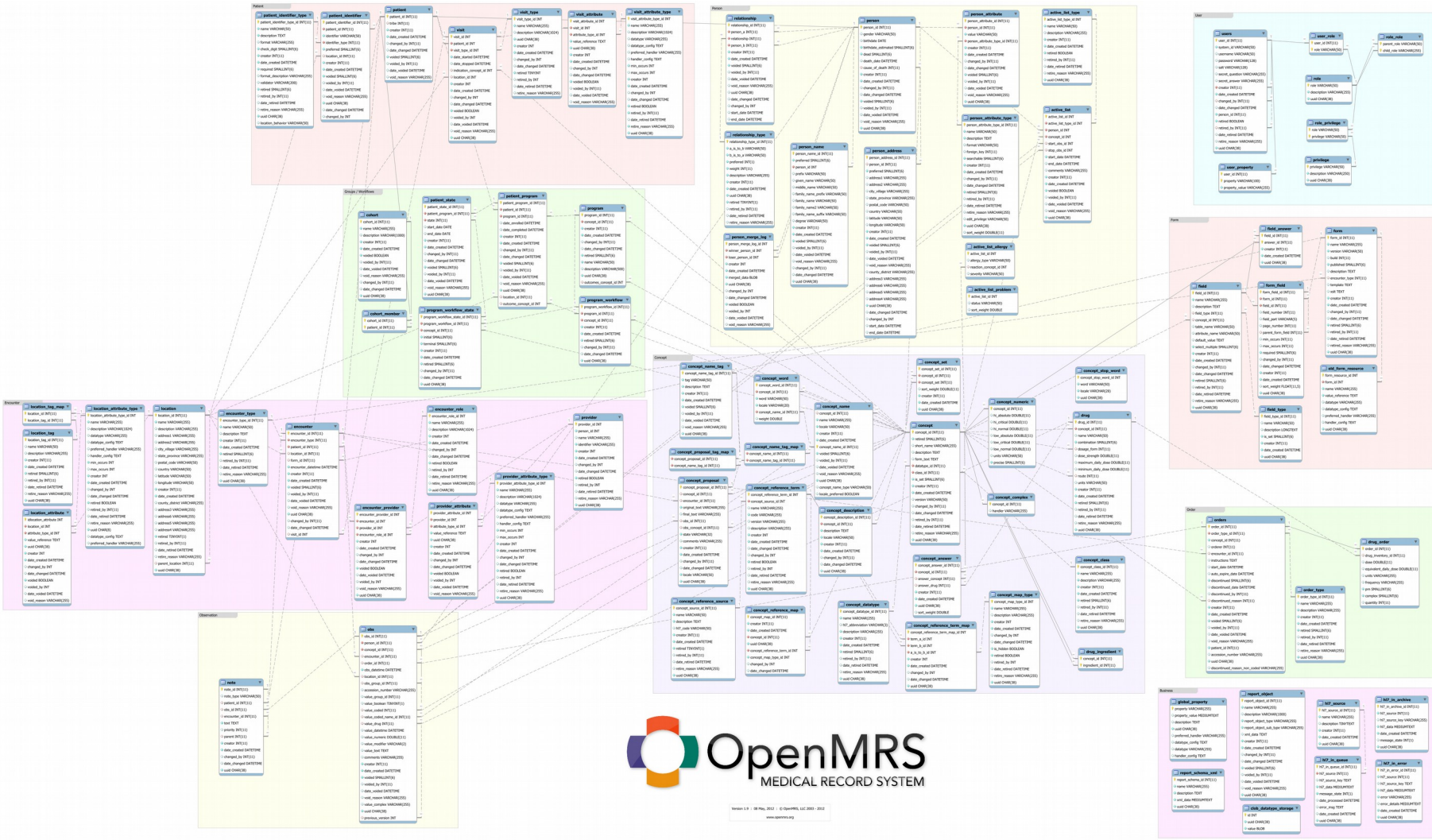


Figure 7.14 E-R diagram with a weak entity set.

- Weak entity sets do not have primary keys
- We use a **discriminator** (see underlined) to distinguish (allowing distinction between entities)
- Here, the discriminator of the weak entity set section consists of the attributes *sec id*, *year*, and *semester*, since, for each course, this set of attributes uniquely identifies one single section for that course
- The discriminator of a weak entity set is also called the *partial key* of the entity set

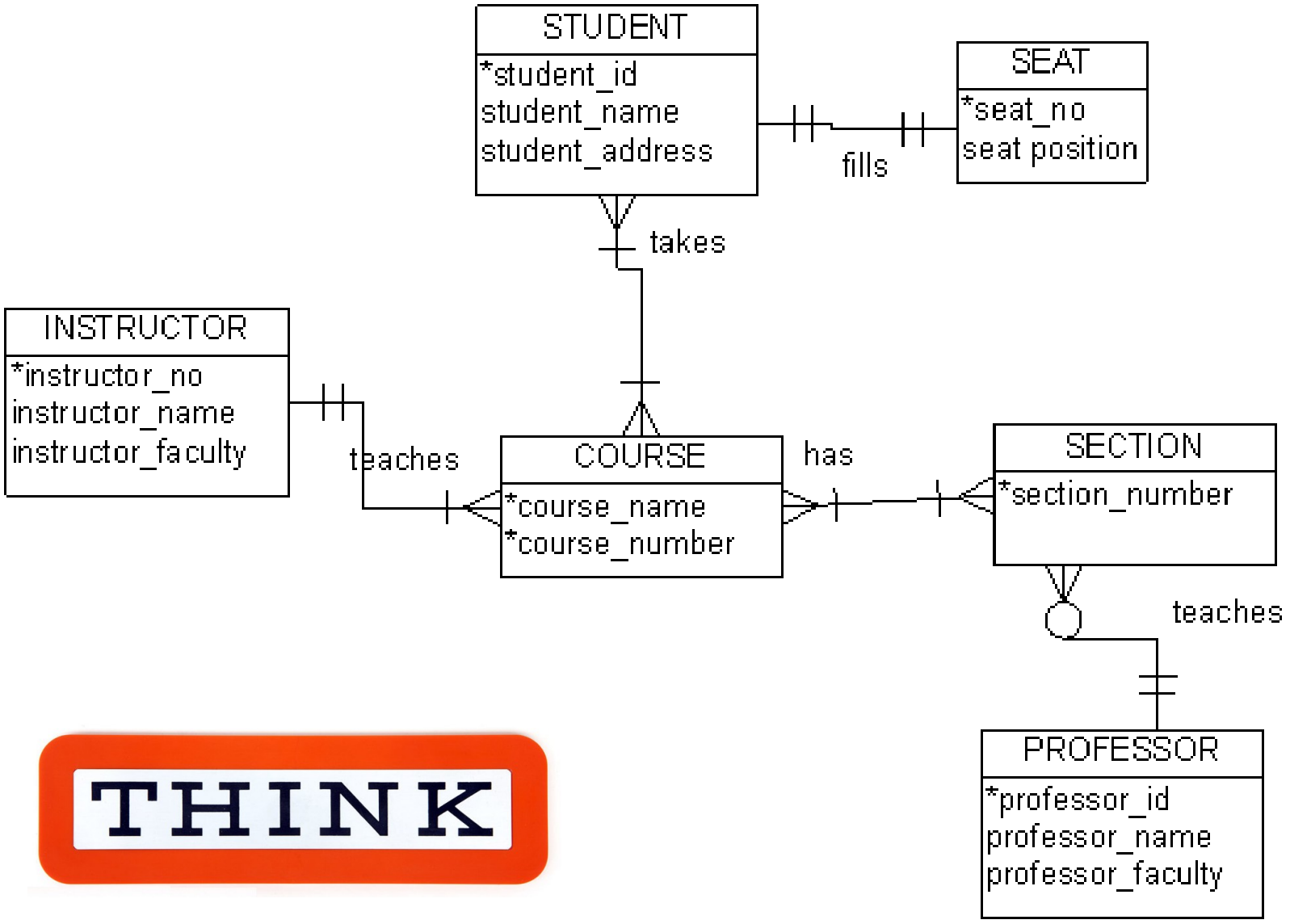
Note: The book uses double lines to denote weak entity relationships

Schema: A Medical Database

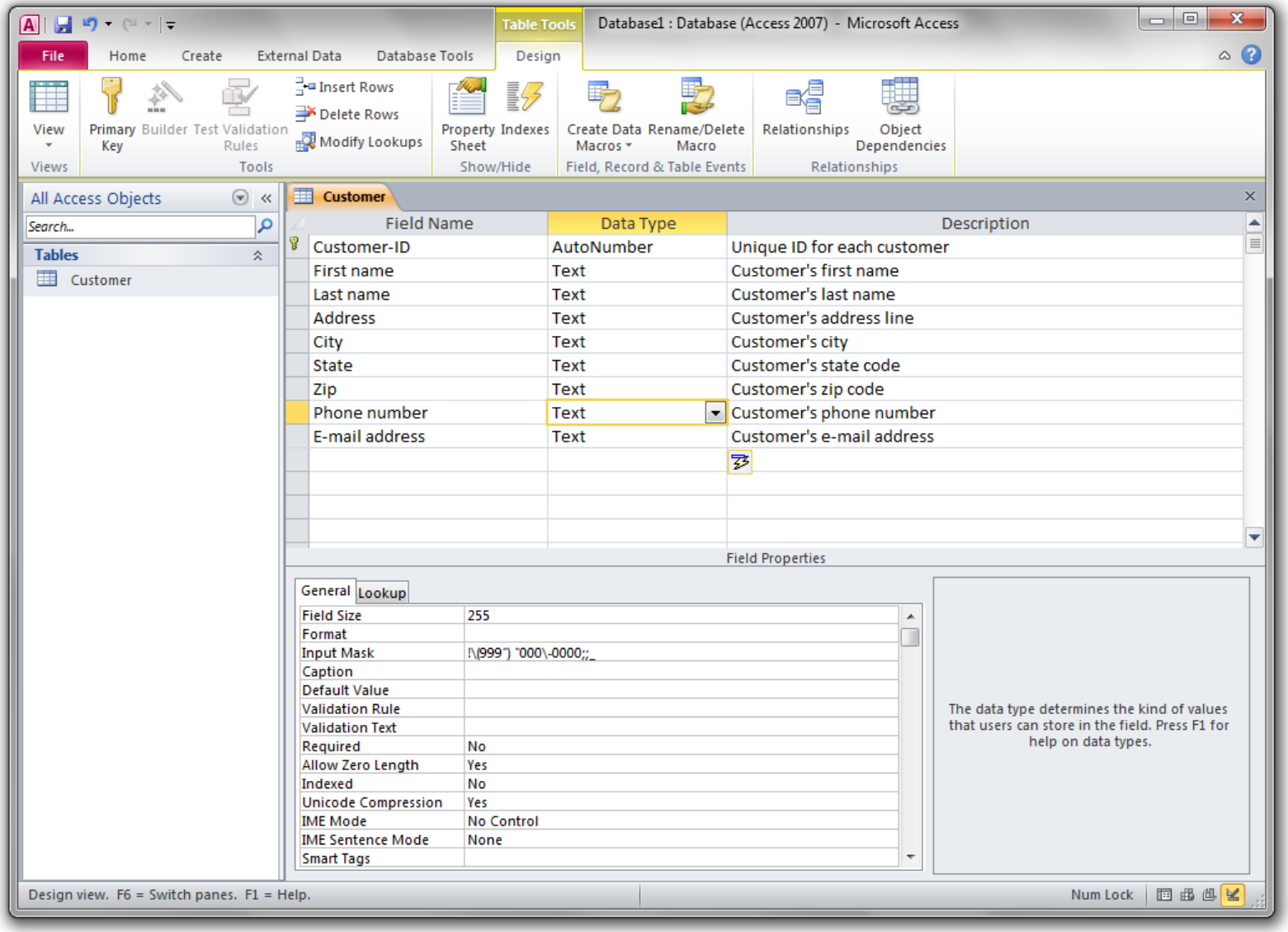


Consider This Schema!

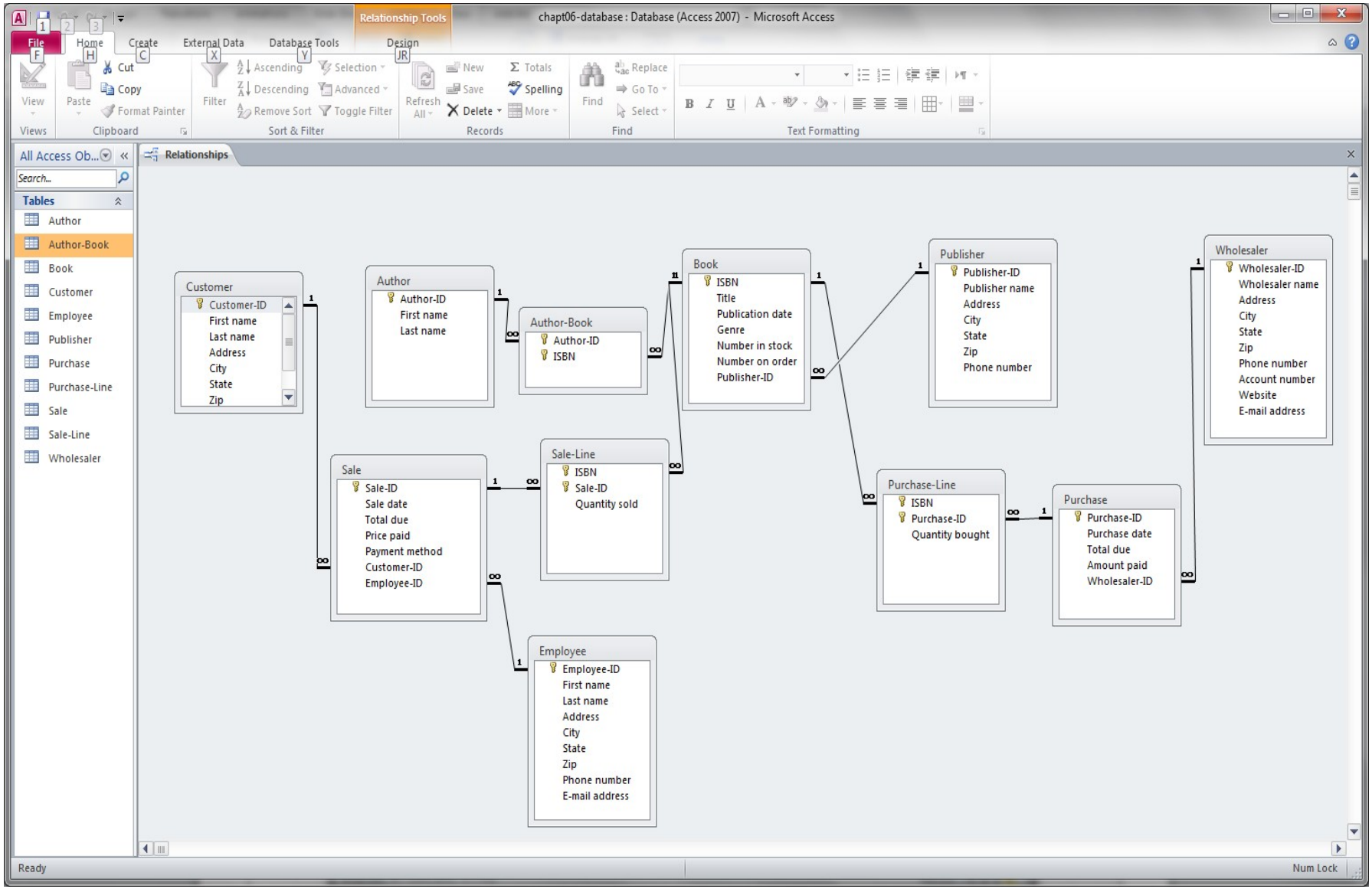
Can you explain the *one-to-many* relationships?



Software to Help Create Tables



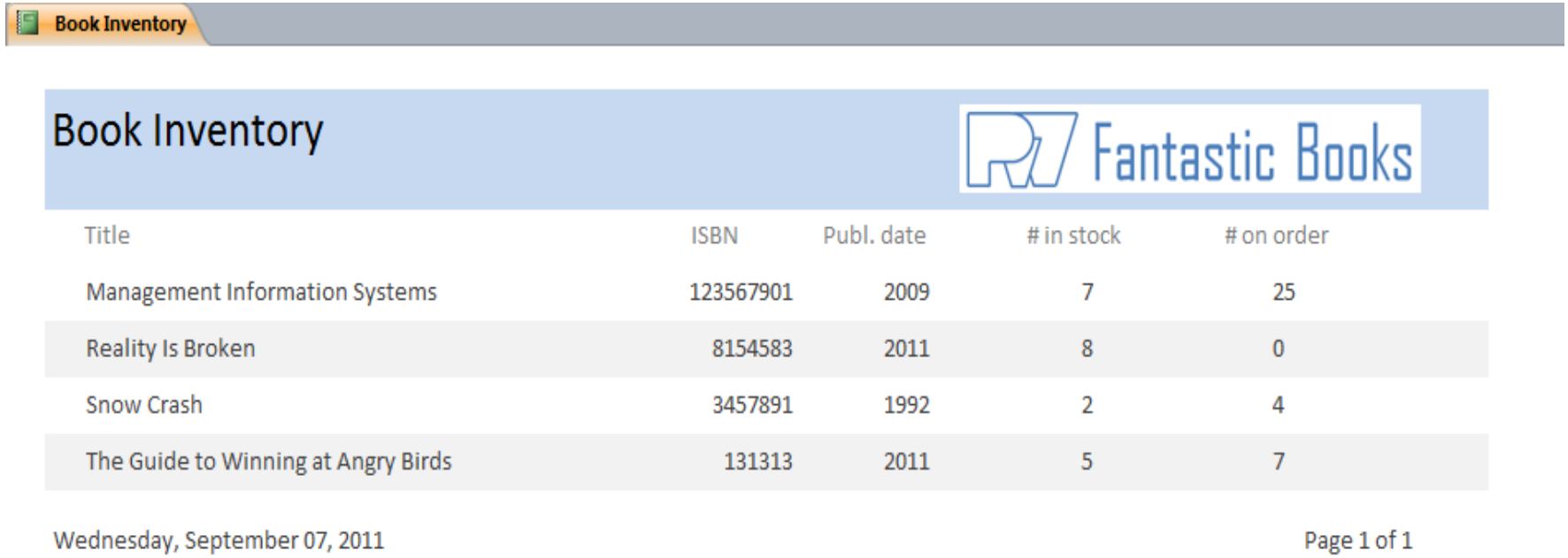
Creating all Tables and Relationships



ALLEGHENY COLLEGE

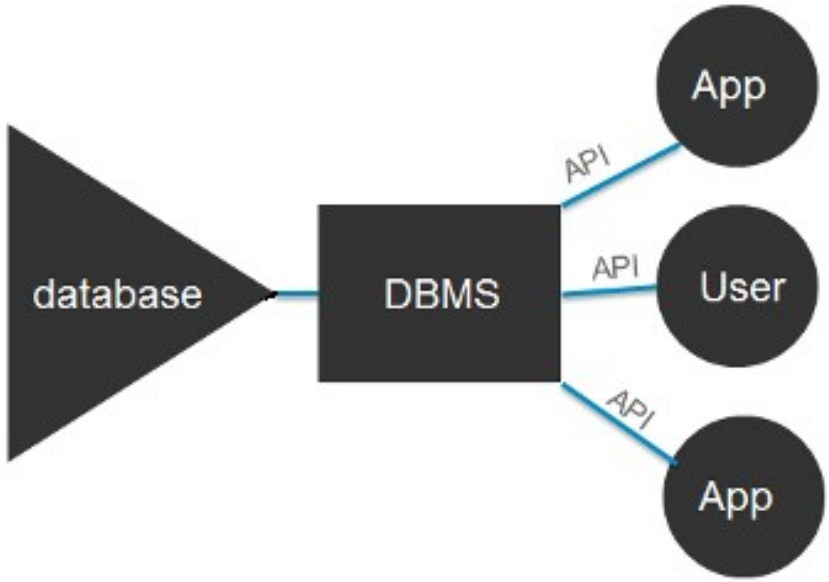
Forms Capture Data

Reports Provide Output



Database Administration

- Responsibilities
  - Defines & organizes database structure and content
  - Develops security procedures
  - Develops database documentation
  - Maintains database management system (DBMS)
    - software that handles the storage, retrieval, and updating of data in a computer system.



ALLEGHENY COLLEGE

So, Why a Database, Again?

- To keep track of the warehouse (Amazon, for example…)

ALLEGHENY COLLEGE

Data Warehouses vs Databases

- Data Warehouses
  - Non-volatile
  - Non-editable: no changes to data
  - Archived data from fiscal periods

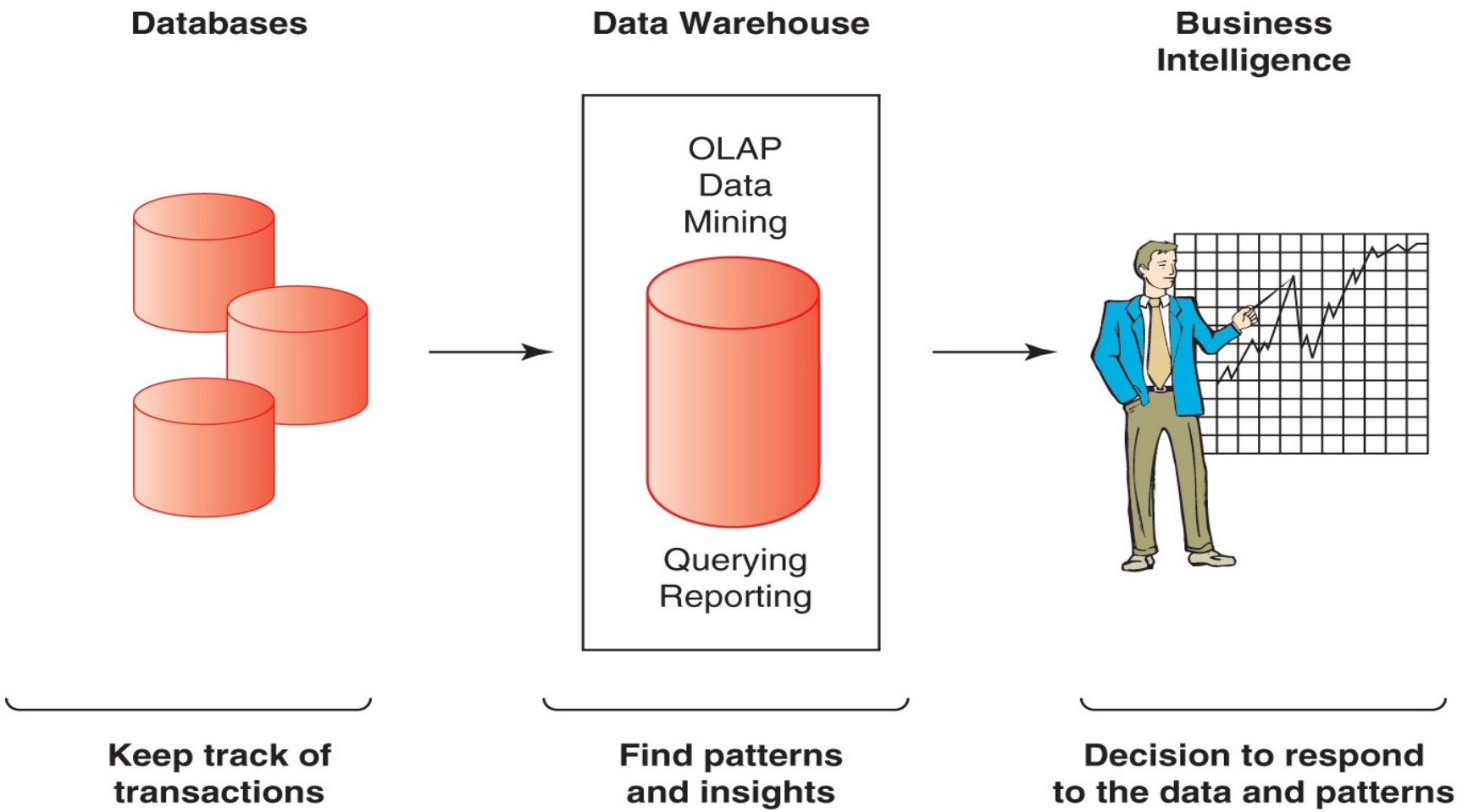
Storage of Data

- Databases
  - Dynamic
  - Able to change data
  - Able to be updated

Access and Ability to Update the Data

ALLEGHENY COLLEGE

Database Trends – Data Warehouse & Data Mining



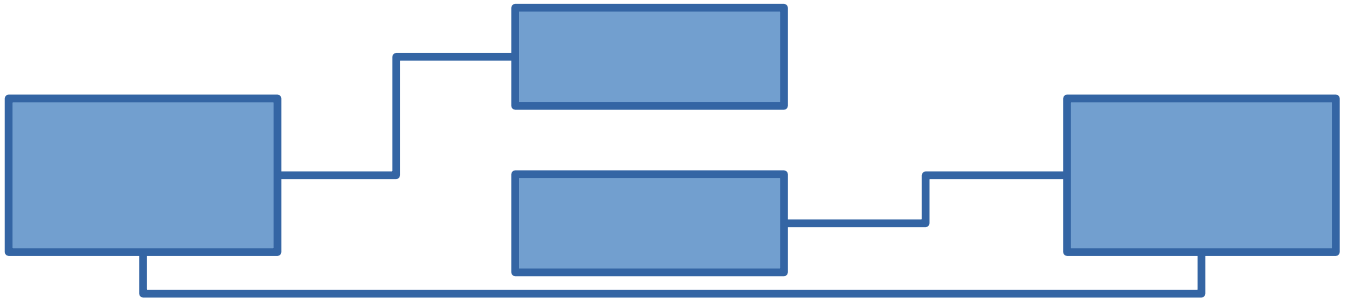
OLAP: Online Analytical Processing

Consider this...

1) Can you organize these terms into tables

student_id	section_number
course_name	seat_position
seat_number	student_address
professor_id	instructor_name
student_name	instructor_number
course_number	professor_department
professor_name	instructor_department

2) Connect these tables based on some logic.



THINK