Database Intro

INFO/CSE 100, Spring 2005

Fluency in Information Technology

http://www.cs.washington.edu/100

Readings and References

- Reading
  - » *Fluency with Information Technology*
    - Chapter 13, Introduction to Database Concepts
- References
  - » *Access Database: Design and Programming*
    - by Steve Roman, published by O'Reilly

Why Study Databases?

- Some of us want to compute, but all of us want information …
  - Much of the archived information is in tables
  - Databases enhance applications, e.g. Web
  - Once you know how to create databases, you can use them to personal advantage
  - Databases introduce interesting ideas

The Internet Movie Database

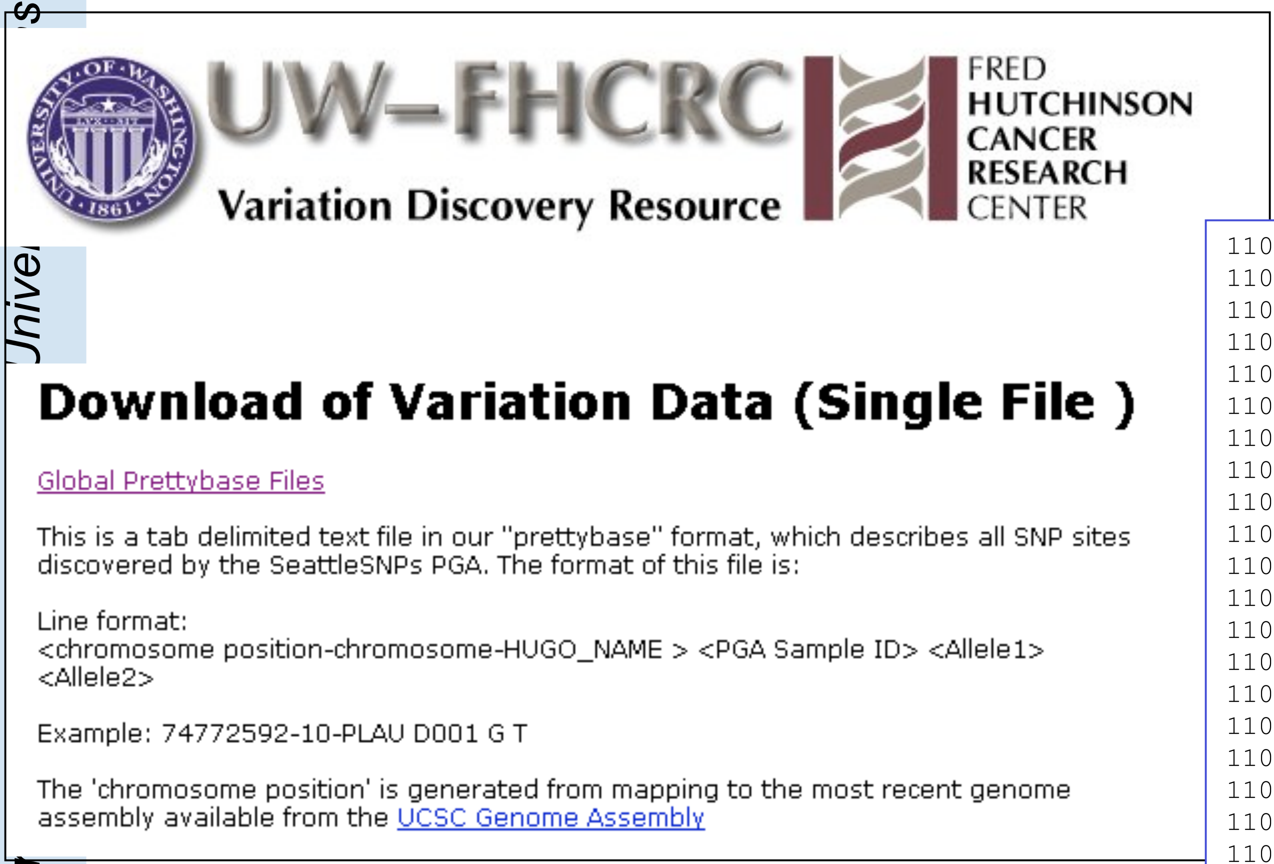
Visited by over 20 million movie lovers each month!

Welcome to the Internet Movie Database, the biggest, best, most award-winning movie site on the planet.

How to organize the data?

- Before relational databases (the kind we study) there were only “flat files”
  - » Structural information is difficult to express
  - » All processing of information is “special cased”
    - custom programs are needed
  - » Information repeated; difficult to combine
  - » Changes in format of one file means all programs that ever process that file must be changed
    - eg, adding ZIP codes

tab-delimited file example


UW–FHCRC
Variation Discovery Resource
FRED HUTCHINSON CANCER RESEARCH CENTER

Download of Variation Data (Single File)

Global Prettybase Files

This is a tab delimited text file in our "prettybase" format, which describes all SNP sites discovered by the SeattleSNPs PGA. The format of this file is:

Line format:
<chromosome position-chromosome-HUGO_NAME > <PGA Sample ID> <Allele1> <Allele2>

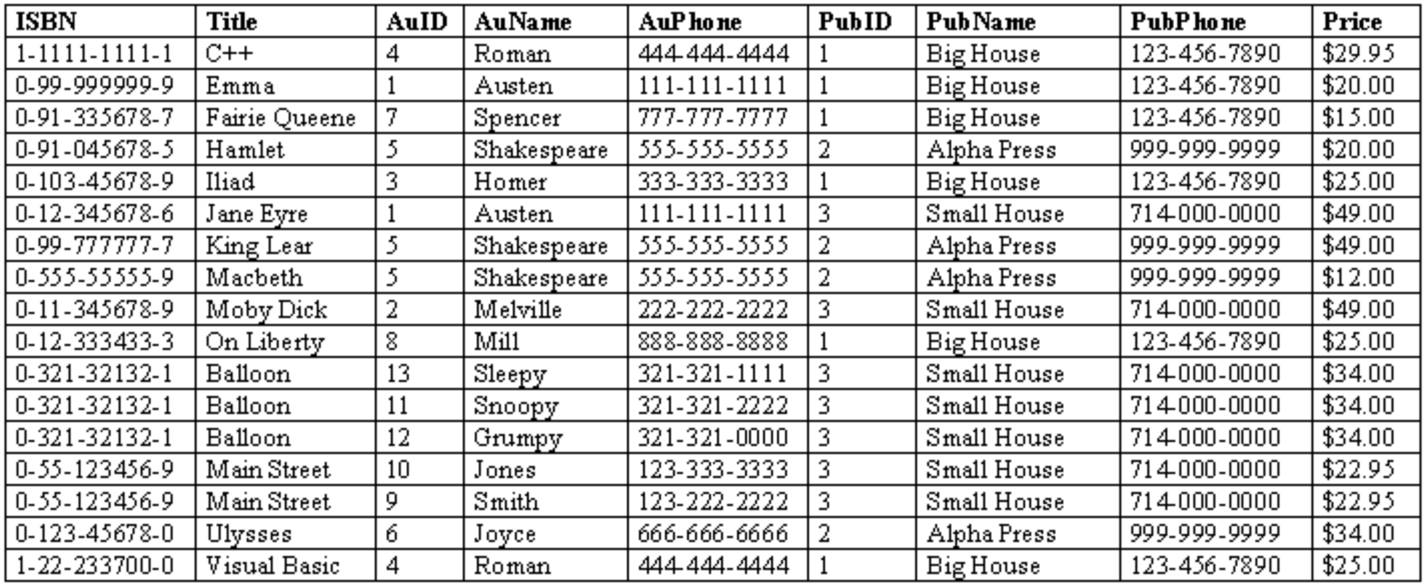
Example: 74772592-10-PLAU D001 G T

The 'chromosome position' is generated from mapping to the most recent genome assembly available from the UCSC Genome Assembly

```
1100322-IL3RA-X      D001    N   N
1100322-IL3RA-X      D002    G   G
1100322-IL3RA-X      D003    G   G
1100322-IL3RA-X      D004    G   G
1100322-IL3RA-X      D005    G   G
1100322-IL3RA-X      D006    G   G
1100322-IL3RA-X      D007    G   G
1100322-IL3RA-X      D008    G   G
1100322-IL3RA-X      D009    A   G
1100322-IL3RA-X      D010    N   N
1100322-IL3RA-X      D011    N   N
1100322-IL3RA-X      D012    N   N
1100322-IL3RA-X      D013    G   G
1100322-IL3RA-X      D014    A   G
1100322-IL3RA-X      D015    N   N
1100322-IL3RA-X      D016    N   N
1100322-IL3RA-X      D033    A   G
1100322-IL3RA-X      D034    A   G
1100322-IL3RA-X      D035    G   G
1100322-IL3RA-X      D036    A   G
1100322-IL3RA-X      D037    A   A
1100322-IL3RA-X      D038    G   G
1100322-IL3RA-X      D039    G   G
1100322-IL3RA-X      D040    G   G
...
```

Library example

notice the redundancy



ISBN	Title	AuID	AuName	AuPhone	PubID	PubName	PubPhone	Price
1-1111-1111-1	C++	4	Roman	444-444-4444	1	Big House	123-456-7890	$29.95
0-99-999999-9	Emma	1	Austen	111-111-1111	1	Big House	123-456-7890	$20.00
0-91-335678-7	Fairie Queene	7	Spencer	777-777-7777	1	Big House	123-456-7890	$15.00
0-91-045678-5	Hamlet	5	Shakespeare	555-555-5555	2	Alpha Press	999-999-9999	$20.00
0-103-45678-9	Iliad	3	Homer	333-333-3333	1	Big House	123-456-7890	$25.00
0-12-345678-6	Jane Eyre	1	Austen	111-111-1111	3	Small House	714-000-0000	$49.00
0-99-777777-7	King Lear	5	Shakespeare	555-555-5555	2	Alpha Press	999-999-9999	$49.00
0-555-55555-9	Macbeth	5	Shakespeare	555-555-5555	2	Alpha Press	999-999-9999	$12.00
0-11-345678-9	Moby Dick	2	Melville	222-222-2222	3	Small House	714-000-0000	$49.00
0-12-333433-3	On Liberty	8	Mill	888-888-8888	1	Big House	123-456-7890	$25.00
0-321-32132-1	Balloon	13	Sleepy	321-321-1111	3	Small House	714-000-0000	$34.00
0-321-32132-1	Balloon	11	Snoopy	321-321-2222	3	Small House	714-000-0000	$34.00
0-321-32132-1	Balloon	12	Grumpy	321-321-0000	3	Small House	714-000-0000	$34.00
0-55-123456-9	Main Street	10	Jones	123-333-3333	3	Small House	714-000-0000	$22.95
0-55-123456-9	Main Street	9	Smith	123-222-2222	3	Small House	714-000-0000	$22.95
0-123-45678-0	Ulysses	6	Joyce	666-666-6666	2	Alpha Press	999-999-9999	$34.00
1-22-233700-0	Visual Basic	4	Roman	444-444-4444	1	Big House	123-456-7890	$25.00

from Access Database book, Steve Roman

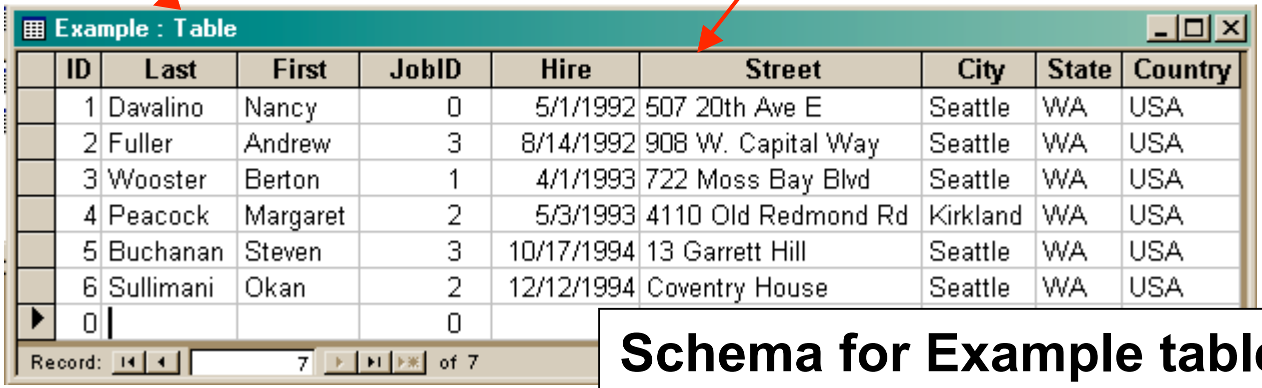
Relational Databases

- Information is stored in tables
  - » Tables store information about *entities*
  - » Entities have characteristics called *attributes*
  - » Each row in a table represents a single entity
    - Each row is a set of attribute values
    - Every row must be unique, identified by a *key*
  - » Relationships -- associations among the data values are stored

Table structure = schema
Table contents = instance

A Table in a Database

Tables have names, attributes {fields}, entities {rows}



Example : Table

ID	Last	First	JobID	Hire	Street	City	State	Country
1	Davalino	Nancy	0	5/1/1992	507 20th Ave E	Seattle	WA	USA
2	Fuller	Andrew	3	8/14/1992	908 W. Capital Way	Seattle	WA	USA
3	Wooster	Berton	1	4/1/1993	722 Moss Bay Blvd	Seattle	WA	USA
4	Peacock	Margaret	2	5/3/1993	4110 Old Redmond Rd	Kirkland	WA	USA
5	Buchanan	Steven	3	10/17/1994	13 Garrett Hill	Seattle	WA	USA
6	Sullimani	Okan	2	12/12/1994	Coventry House	Seattle	WA	USA
0			0					

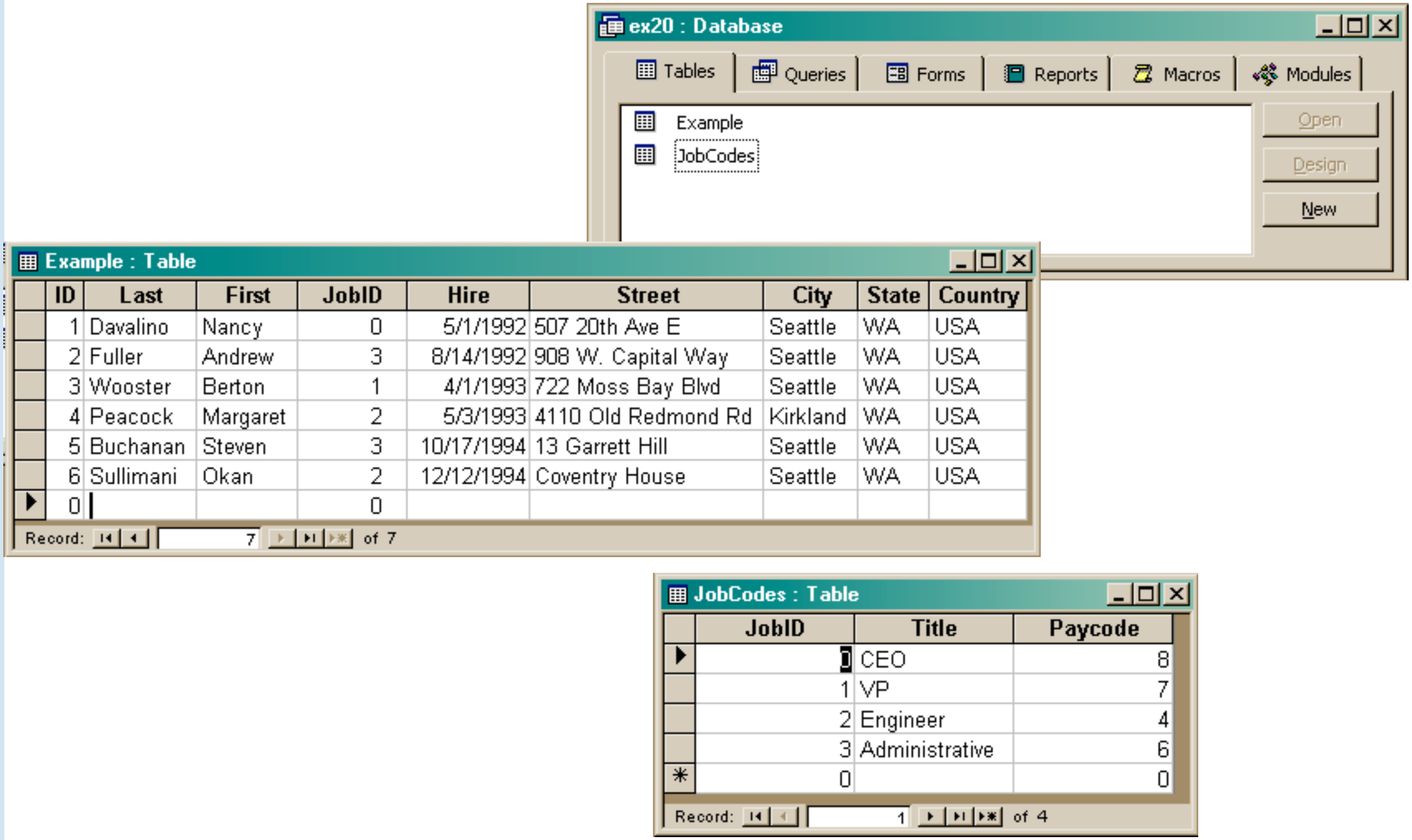
instance

Schema for Example table:

ID	number	unique number(Key)
Last	text	person’s last name
First	text	person’s first name
JobCode	number	current position
Hire	date	first day on job
...		

schema

Two tables in a database



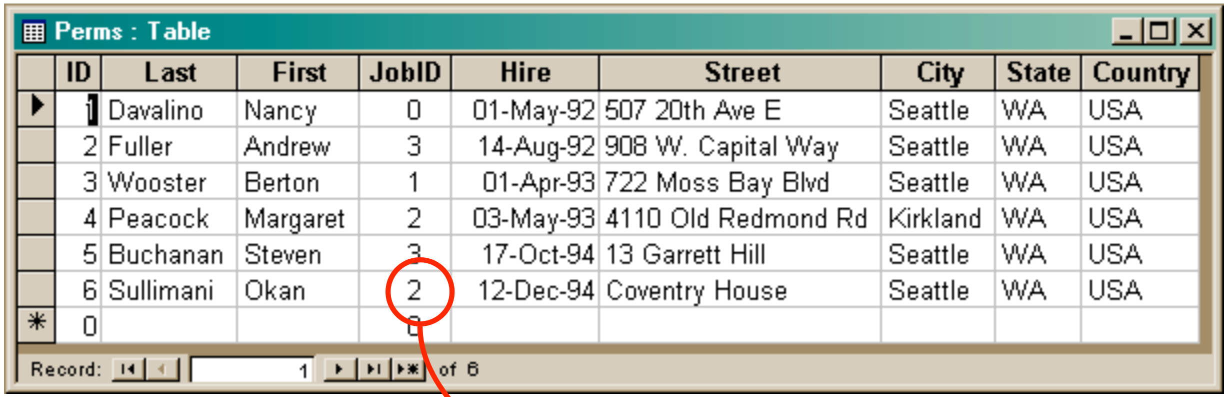
ID	Last	First	JobID	Hire	Street	City	State	Country
1	Davalino	Nancy	0	5/1/1992	507 20th Ave E	Seattle	WA	USA
2	Fuller	Andrew	3	8/14/1992	908 W. Capital Way	Seattle	WA	USA
3	Wooster	Berton	1	4/1/1993	722 Moss Bay Blvd	Seattle	WA	USA
4	Peacock	Margaret	2	5/3/1993	4110 Old Redmond Rd	Kirkland	WA	USA
5	Buchanan	Steven	3	10/17/1994	13 Garrett Hill	Seattle	WA	USA
6	Sullimani	Okan	2	12/12/1994	Coventry House	Seattle	WA	USA
0			0					

JobID	Title	Paycode
0	CEO	8
1	VP	7
2	Engineer	4
3	Administrative	6
0		0

Redundancy in a database is Very Bad

- Not every assembly of tables is a good database
- Repeating data is a bad idea
  - » Replicated data can differ in its different locations, e.g. multiple addresses can differ
    - Inconsistent data is worse than no data
  - » Keep a *single copy* of any data
    - if it is needed in multiple places, associate it with a key and store key rather than the data

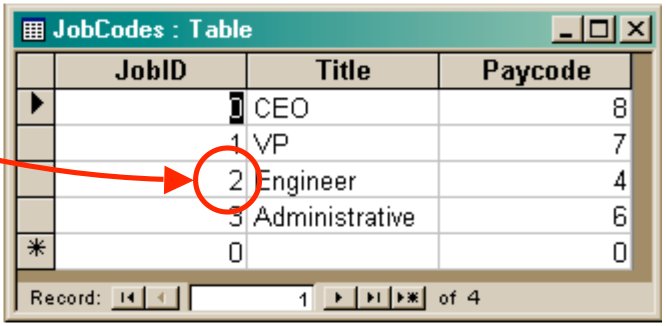
Relationships between tables



Perms : Table

	ID	Last	First	JobID	Hire	Street	City	State	Country
▶	1	Davalino	Nancy	0	01-May-92	507 20th Ave E	Seattle	WA	USA
	2	Fuller	Andrew	3	14-Aug-92	908 W. Capital Way	Seattle	WA	USA
	3	Wooster	Berton	1	01-Apr-93	722 Moss Bay Blvd	Seattle	WA	USA
	4	Peacock	Margaret	2	03-May-93	4110 Old Redmond Rd	Kirkland	WA	USA
	5	Buchanan	Steven	3	17-Oct-94	13 Garrett Hill	Seattle	WA	USA
	6	Sullimani	Okan	2	12-Dec-94	Coventry House	Seattle	WA	USA
✳	0			0					

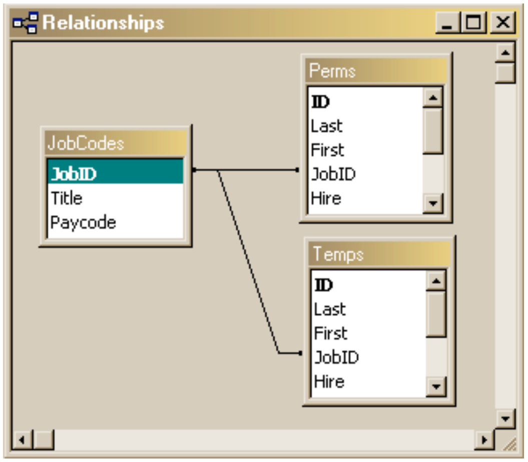
Record: 1 of 6



JobCodes : Table

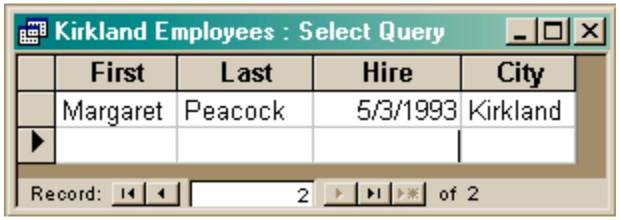
	JobID	Title	Paycode
▶	0	CEO	8
	1	VP	7
	2	Engineer	4
	3	Administrative	6
✳	0		0

Record: 1 of 4



“You can look it up”

- When looking for information, a single item might be the answer, but a table is more likely
  - » Which employees live in Kirkland?
    - Table of employees
  - » Who is taking INFO/CSE 100?
    - Table of students
  - » Whose mile run time ≤ 4:00?
    - Table of runners



Kirkland Employees : Select Query

First	Last	Hire	City
Margaret	Peacock	5/3/1993	Kirkland

Record: 2 of 2

Query to a database (set of tables) produces a new table

Relational Algebra: Tables From Tables

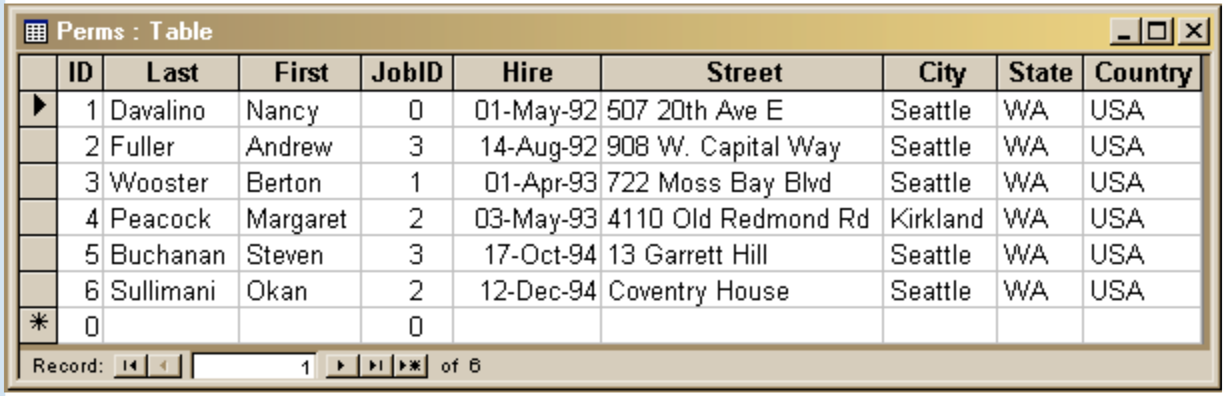
- There are five basic "algebraic" operations on tables:
  - Select -- pick rows from a table
  - Project -- pick columns from a table
  - Union -- combine two tables w/like columns
  - Difference -- remove one table from another
  - Product -- create "all pairs" from two tables

From this basis, many more complicated operations can be built up

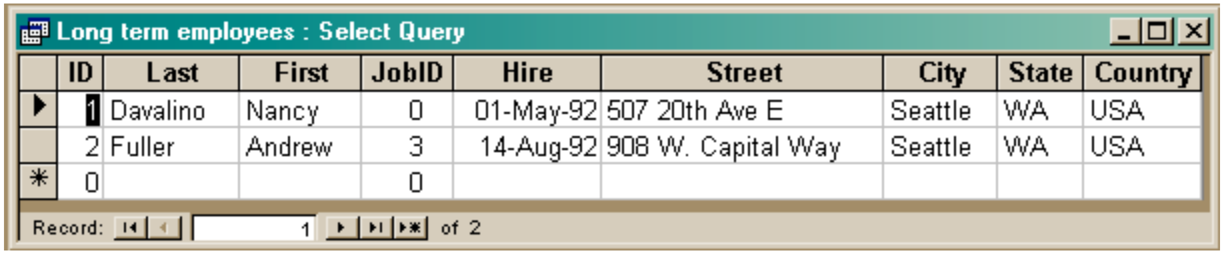
Select Operation

- Select creates a table from the rows of another table meeting a criterion

Select_from Example On Hire < 1993



	ID	Last	First	JobID	Hire	Street	City	State	Country
▶	1	Davalino	Nancy	0	01-May-92	507 20th Ave E	Seattle	WA	USA
	2	Fuller	Andrew	3	14-Aug-92	908 W. Capital Way	Seattle	WA	USA
	3	Wooster	Berton	1	01-Apr-93	722 Moss Bay Blvd	Seattle	WA	USA
	4	Peacock	Margaret	2	03-May-93	4110 Old Redmond Rd	Kirkland	WA	USA
	5	Buchanan	Steven	3	17-Oct-94	13 Garrett Hill	Seattle	WA	USA
	6	Sullimani	Okan	2	12-Dec-94	Coventry House	Seattle	WA	USA
*	0			0					

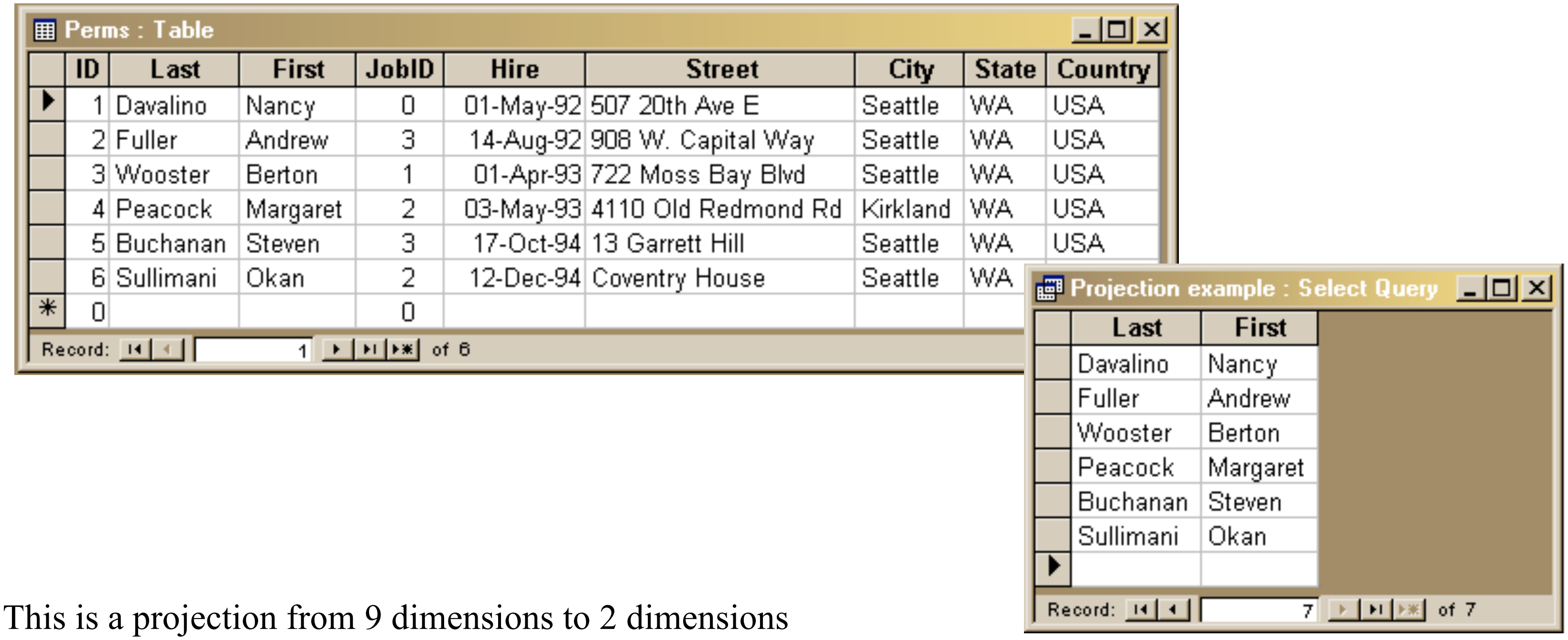


	ID	Last	First	JobID	Hire	Street	City	State	Country
▶	1	Davalino	Nancy	0	01-May-92	507 20th Ave E	Seattle	WA	USA
	2	Fuller	Andrew	3	14-Aug-92	908 W. Capital Way	Seattle	WA	USA
*	0			0					

Project

- Project creates a table from the columns of another table

Project Last, First **From** Example



Perms : Table

	ID	Last	First	JobID	Hire	Street	City	State	Country
▶	1	Davalino	Nancy	0	01-May-92	507 20th Ave E	Seattle	WA	USA
	2	Fuller	Andrew	3	14-Aug-92	908 W. Capital Way	Seattle	WA	USA
	3	Wooster	Berton	1	01-Apr-93	722 Moss Bay Blvd	Seattle	WA	USA
	4	Peacock	Margaret	2	03-May-93	4110 Old Redmond Rd	Kirkland	WA	USA
	5	Buchanan	Steven	3	17-Oct-94	13 Garrett Hill	Seattle	WA	USA
	6	Sullimani	Okan	2	12-Dec-94	Coventry House	Seattle	WA	
*	0			0					

Projection example : Select Query

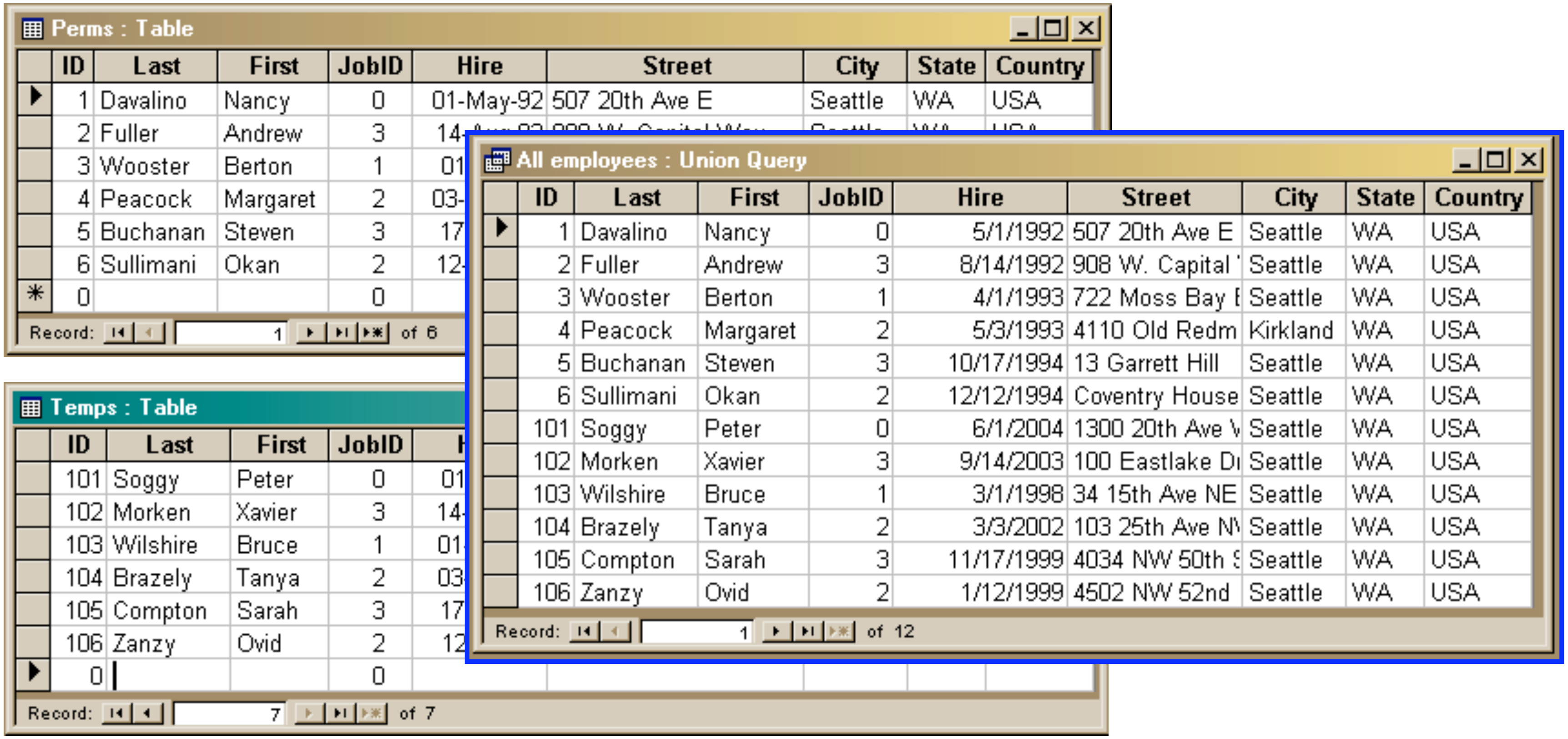
	Last	First
	Davalino	Nancy
	Fuller	Andrew
	Wooster	Berton
	Peacock	Margaret
	Buchanan	Steven
	Sullimani	Okan
▶		

This is a projection from 9 dimensions to 2 dimensions

Union

- Union combines two tables with *same attributes*

All employees = perms UNION temps



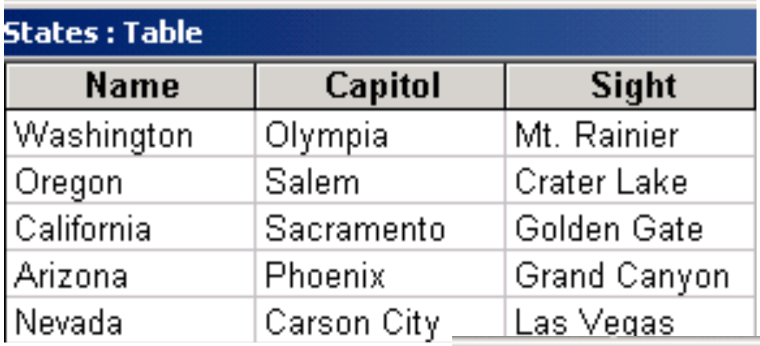
	ID	Last	First	JobID	Hire	Street	City	State	Country
▶	1	Davalino	Nancy	0	01-May-92	507 20th Ave E	Seattle	WA	USA
	2	Fuller	Andrew	3	14-Aug-92	908 W. Capital Way	Seattle	WA	USA
	3	Wooster	Berton	1	01-				
	4	Peacock	Margaret	2	03-				
	5	Buchanan	Steven	3	17				
	6	Sullimani	Okan	2	12-				
✳	0			0					

	ID	Last	First	JobID	H
	101	Soggy	Peter	0	01
	102	Morken	Xavier	3	14
	103	Wilshire	Bruce	1	01
	104	Brazely	Tanya	2	03
	105	Compton	Sarah	3	17
	106	Zanzy	Ovid	2	12
▶	0			0	

	ID	Last	First	JobID	Hire	Street	City	State	Country
▶	1	Davalino	Nancy	0	5/1/1992	507 20th Ave E	Seattle	WA	USA
	2	Fuller	Andrew	3	8/14/1992	908 W. Capital '	Seattle	WA	USA
	3	Wooster	Berton	1	4/1/1993	722 Moss Bay I	Seattle	WA	USA
	4	Peacock	Margaret	2	5/3/1993	4110 Old Redm	Kirkland	WA	USA
	5	Buchanan	Steven	3	10/17/1994	13 Garrett Hill	Seattle	WA	USA
	6	Sullimani	Okan	2	12/12/1994	Coventry House	Seattle	WA	USA
	101	Soggy	Peter	0	6/1/2004	1300 20th Ave V	Seattle	WA	USA
	102	Morken	Xavier	3	9/14/2003	100 Eastlake Dı	Seattle	WA	USA
	103	Wilshire	Bruce	1	3/1/1998	34 15th Ave NE	Seattle	WA	USA
	104	Brazely	Tanya	2	3/3/2002	103 25th Ave N\	Seattle	WA	USA
	105	Compton	Sarah	3	11/17/1999	4034 NW 50th S	Seattle	WA	USA
	106	Zanzy	Ovid	2	1/12/1999	4502 NW 52nd	Seattle	WA	USA

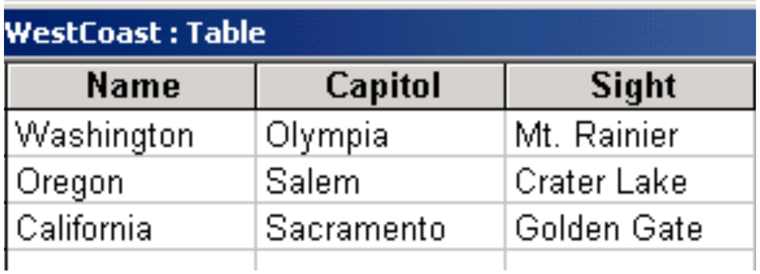
Difference

- Difference (written like subtraction) removes 1 table's rows from another
  - Eastern = States - WestCoast



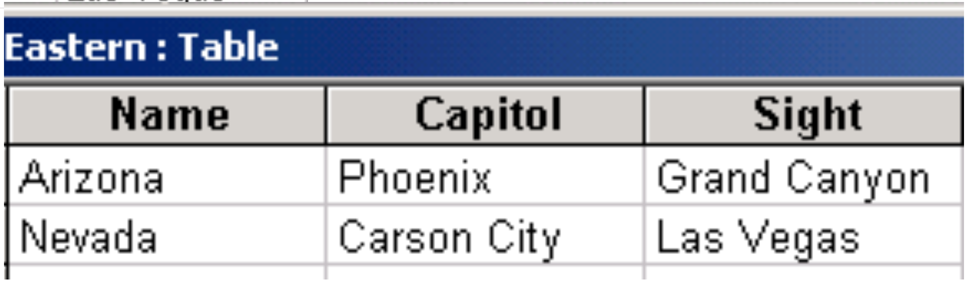
States : Table

Name	Capitol	Sight
Washington	Olympia	Mt. Rainier
Oregon	Salem	Crater Lake
California	Sacramento	Golden Gate
Arizona	Phoenix	Grand Canyon
Nevada	Carson City	Las Vegas



WestCoast : Table

Name	Capitol	Sight
Washington	Olympia	Mt. Rainier
Oregon	Salem	Crater Lake
California	Sacramento	Golden Gate



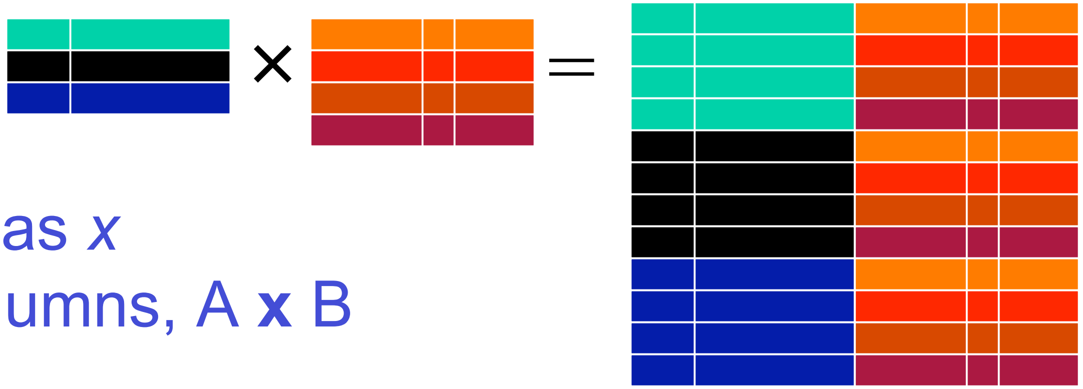
Eastern : Table

Name	Capitol	Sight
Arizona	Phoenix	Grand Canyon
Nevada	Carson City	Las Vegas

Product

- Product (written like multiplication) combines columns and pairs all rows

Colors = Blues **x** Reds



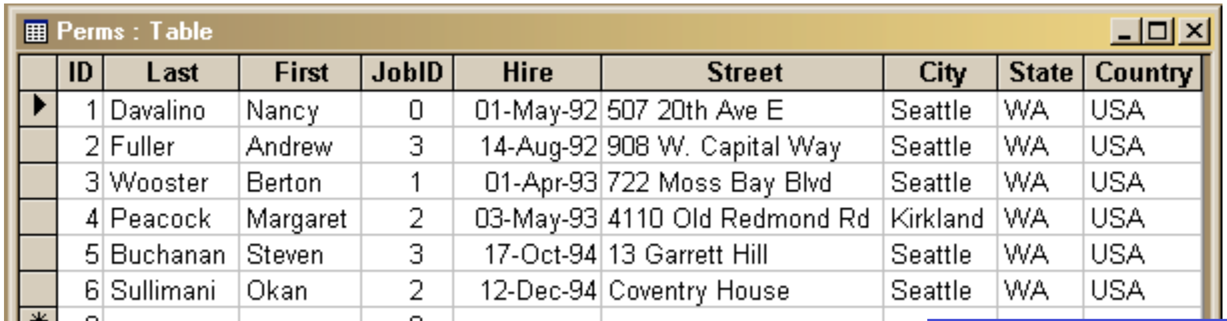
Column Rule: If A has *x* columns, B has *y* columns, A **x** B has *x*+*y* columns

Row Rule: If A has *m* rows, B has *n* rows A **x** B has *mn* rows

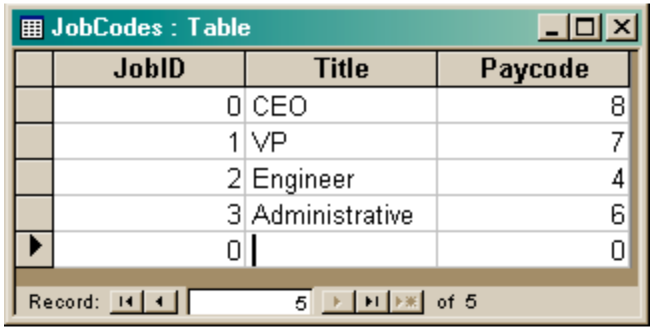
Join

- Join (written like a bow tie) combines rows if common field matches

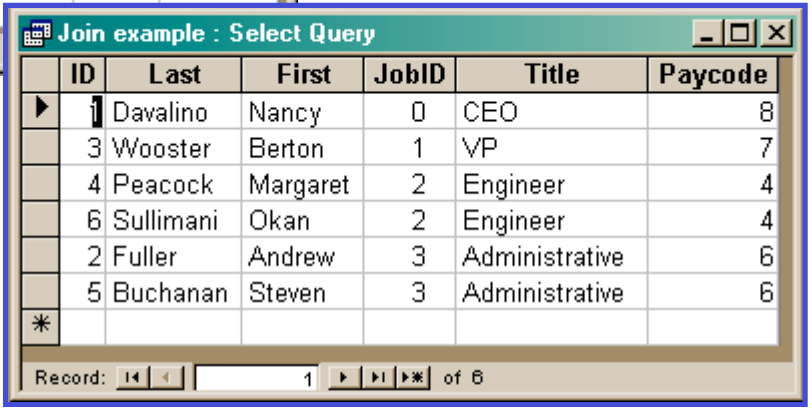
Employee List = Perms $\bowtie$ JobCodes



ID	Last	First	JobID	Hire	Street	City	State	Country
1	Davalino	Nancy	0	01-May-92	507 20th Ave E	Seattle	WA	USA
2	Fuller	Andrew	3	14-Aug-92	908 W. Capital Way	Seattle	WA	USA
3	Wooster	Berton	1	01-Apr-93	722 Moss Bay Blvd	Seattle	WA	USA
4	Peacock	Margaret	2	03-May-93	4110 Old Redmond Rd	Kirkland	WA	USA
5	Buchanan	Steven	3	17-Oct-94	13 Garrett Hill	Seattle	WA	USA
6	Sullimani	Okan	2	12-Dec-94	Coventry House	Seattle	WA	USA
0			0					



JobID	Title	Paycode
0	CEO	8
1	VP	7
2	Engineer	4
3	Administrative	6
0		0



ID	Last	First	JobID	Title	Paycode
1	Davalino	Nancy	0	CEO	8
3	Wooster	Berton	1	VP	7
4	Peacock	Margaret	2	Engineer	4
6	Sullimani	Okan	2	Engineer	4
2	Fuller	Andrew	3	Administrative	6
5	Buchanan	Steven	3	Administrative	6

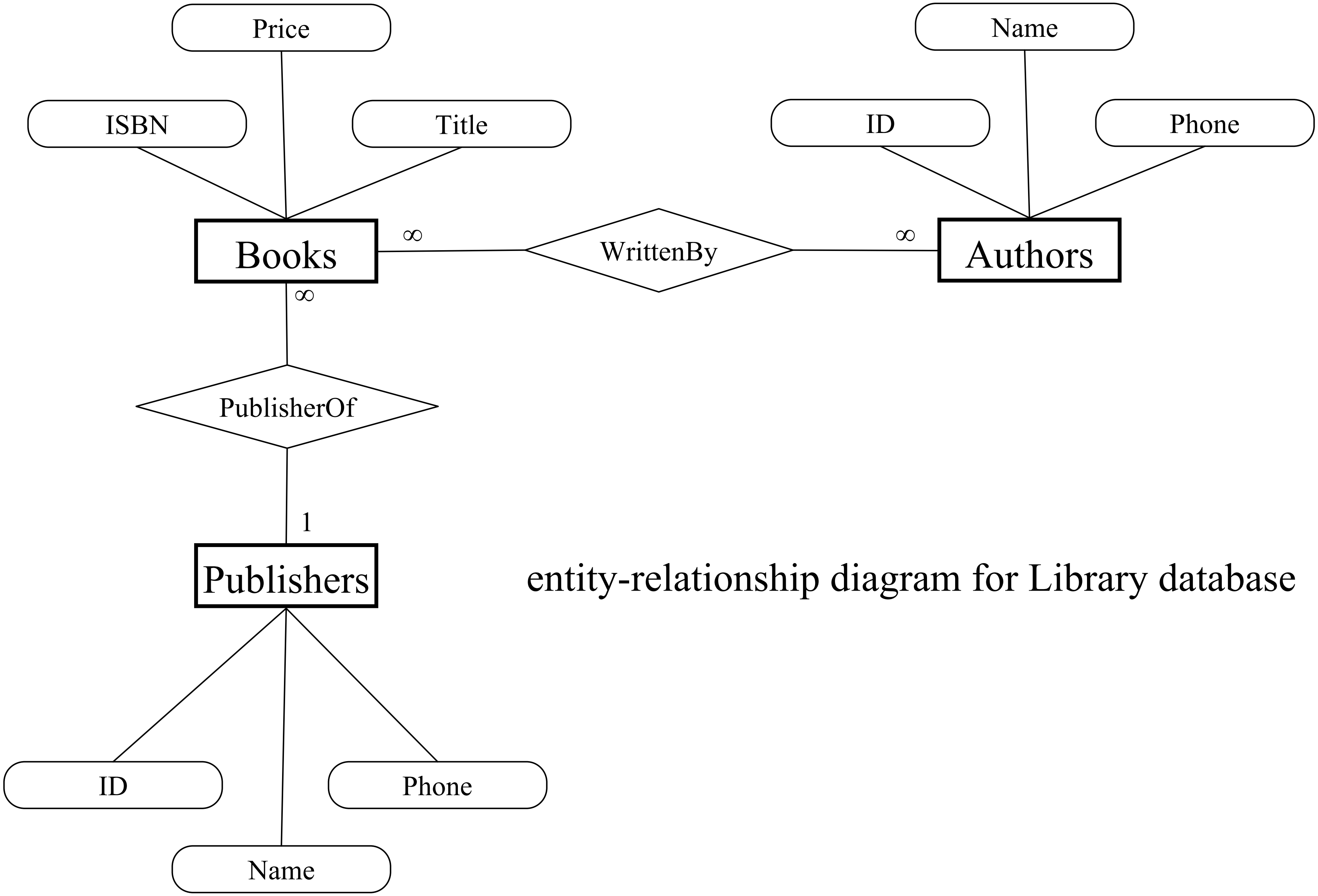
DB Operations

- The five DB Operations can create any table from a given set of tables
  - All modern database systems are built on these relational operations
  - Join is not primitive, but can be built from 5
  - Join, select and project are used most often
  - The operations are not usually used directly, but are used indirectly from other languages
- Structured Query Language (SQL) is the language that we talk to the database in

SQL, the DB language we learn, is built on basic 5

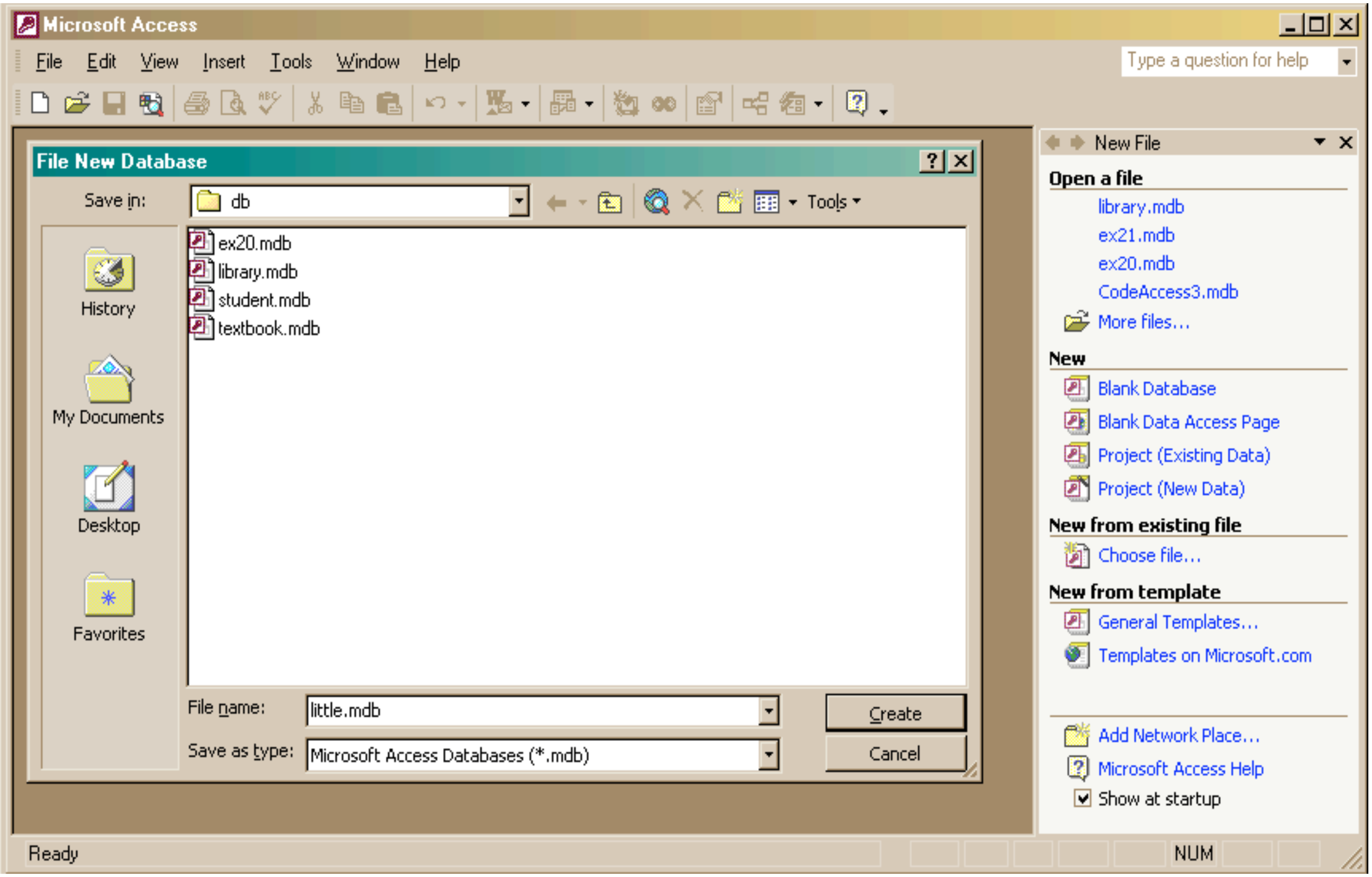
Database Structure

- A database contains one or more *tables*
  - » Tables include *entities* with *attributes*
  - » There are *relationships* defined between the entities in the various tables
  - » Retrieve information from the tables using *queries*
- First, design the database
  - » What are the entities?
  - » What are the attributes of each entity?
  - » What are the relationships between tables?



entity-relationship diagram for Library database

Create a new database



Save in: db

ex20.mdb
library.mdb
student.mdb
textbook.mdb

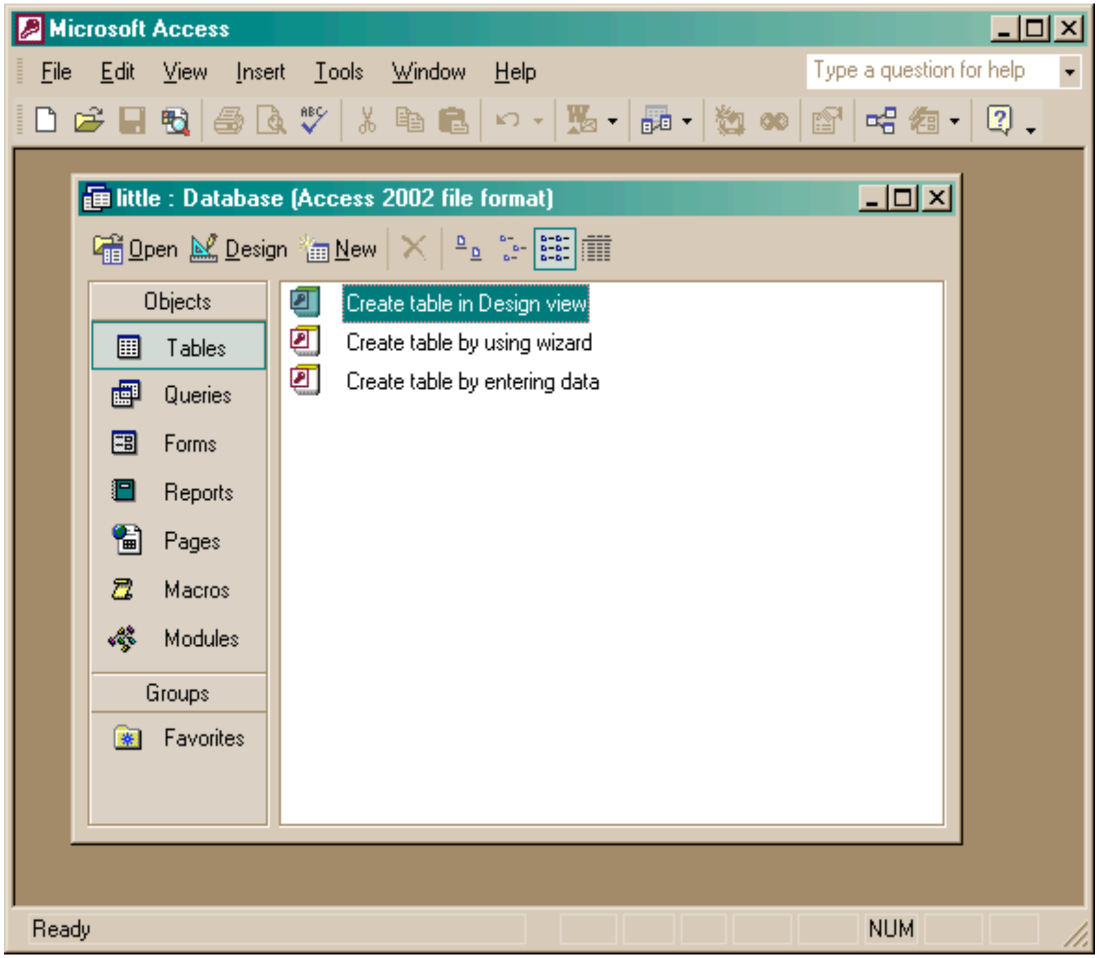
File name: little.mdb

Save as type: Microsoft Access Databases (*.mdb)

☑ Show at startup

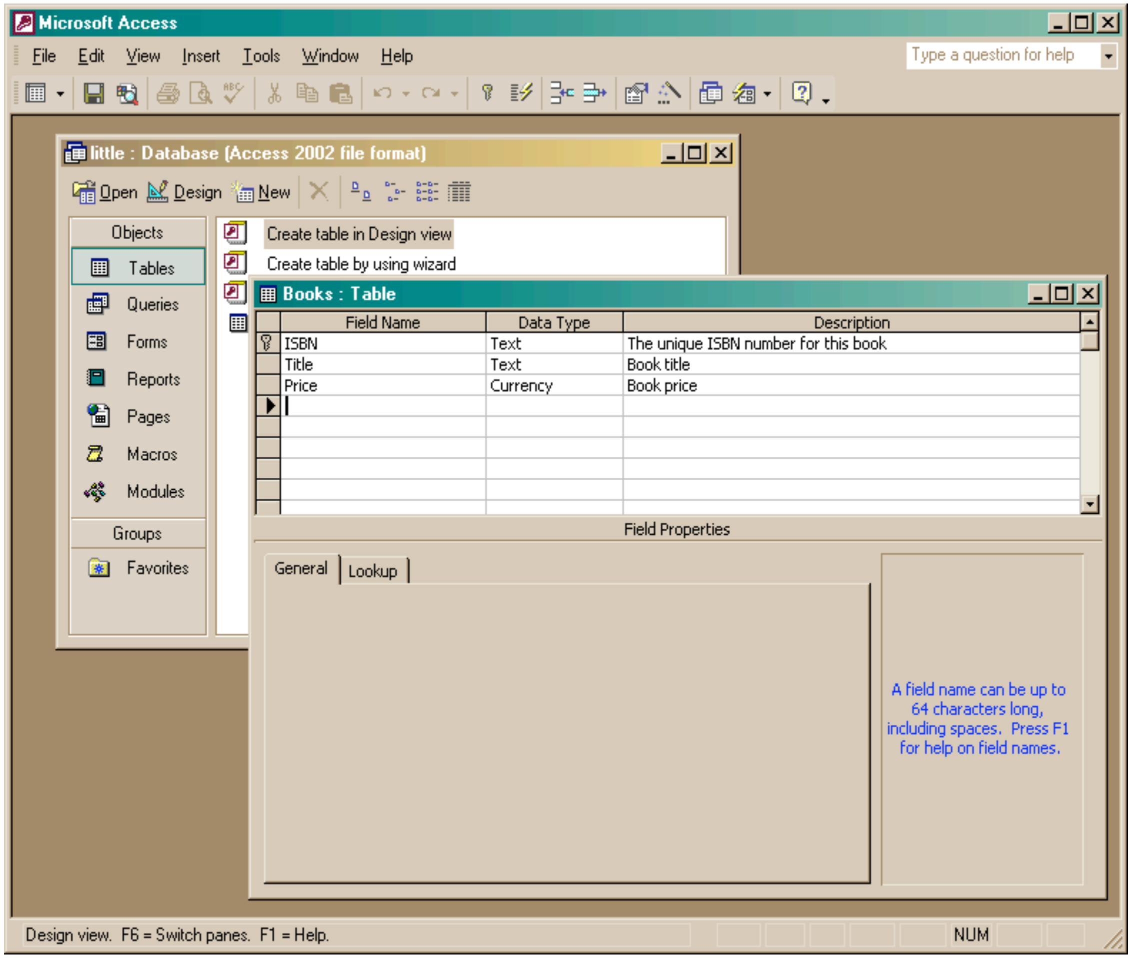
The Information School of the University of Washington

Create a new table in the database



Create table in Design view
Create table by using wizard
Create table by entering data

Creating a table in Design view



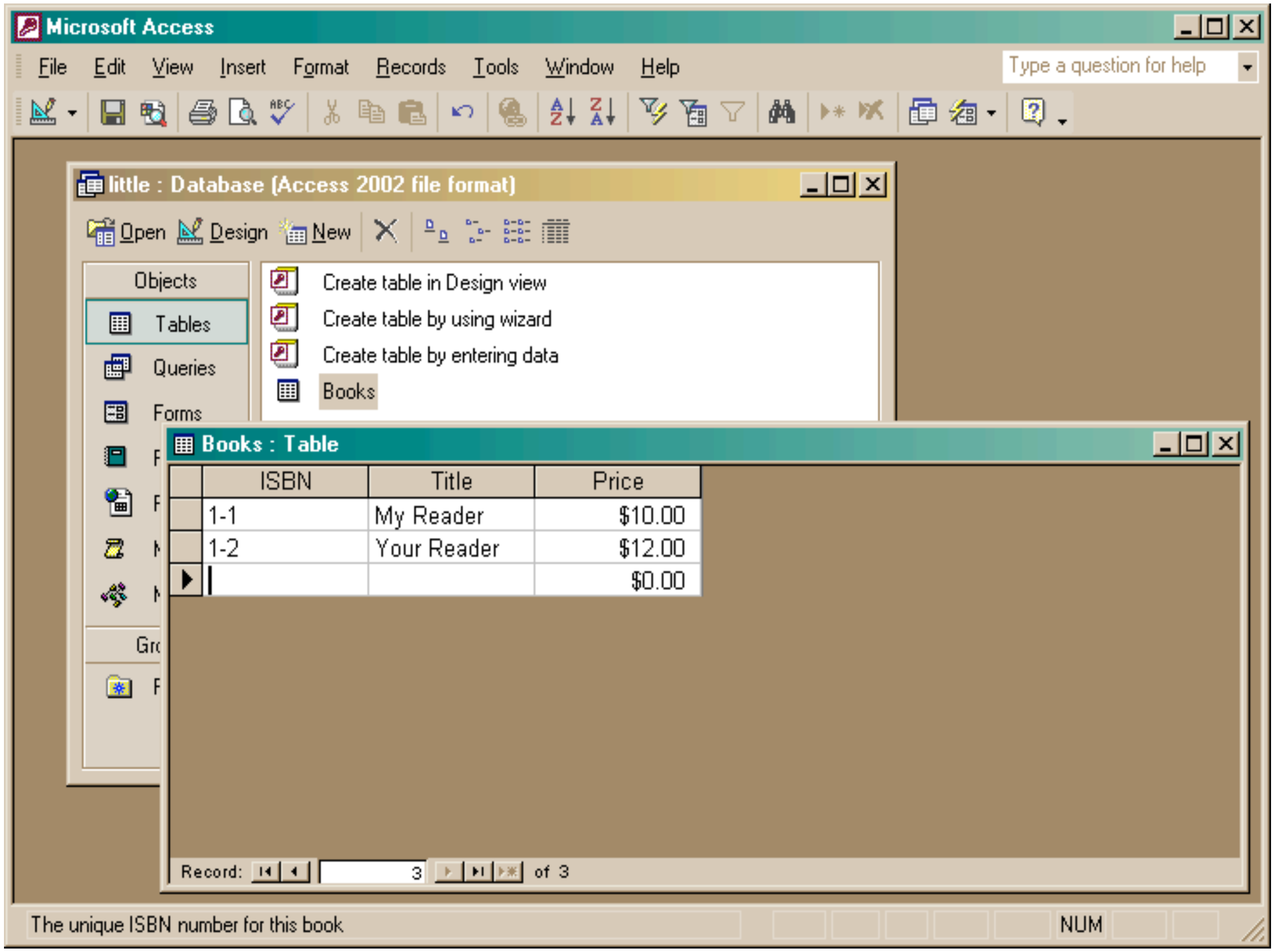
Field Name	Data Type	Description
ISBN	Text	The unique ISBN number for this book
Title	Text	Book title
Price	Currency	Book price

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

Design view. F6 = Switch panes. F1 = Help.

NUM

Entering Table Data

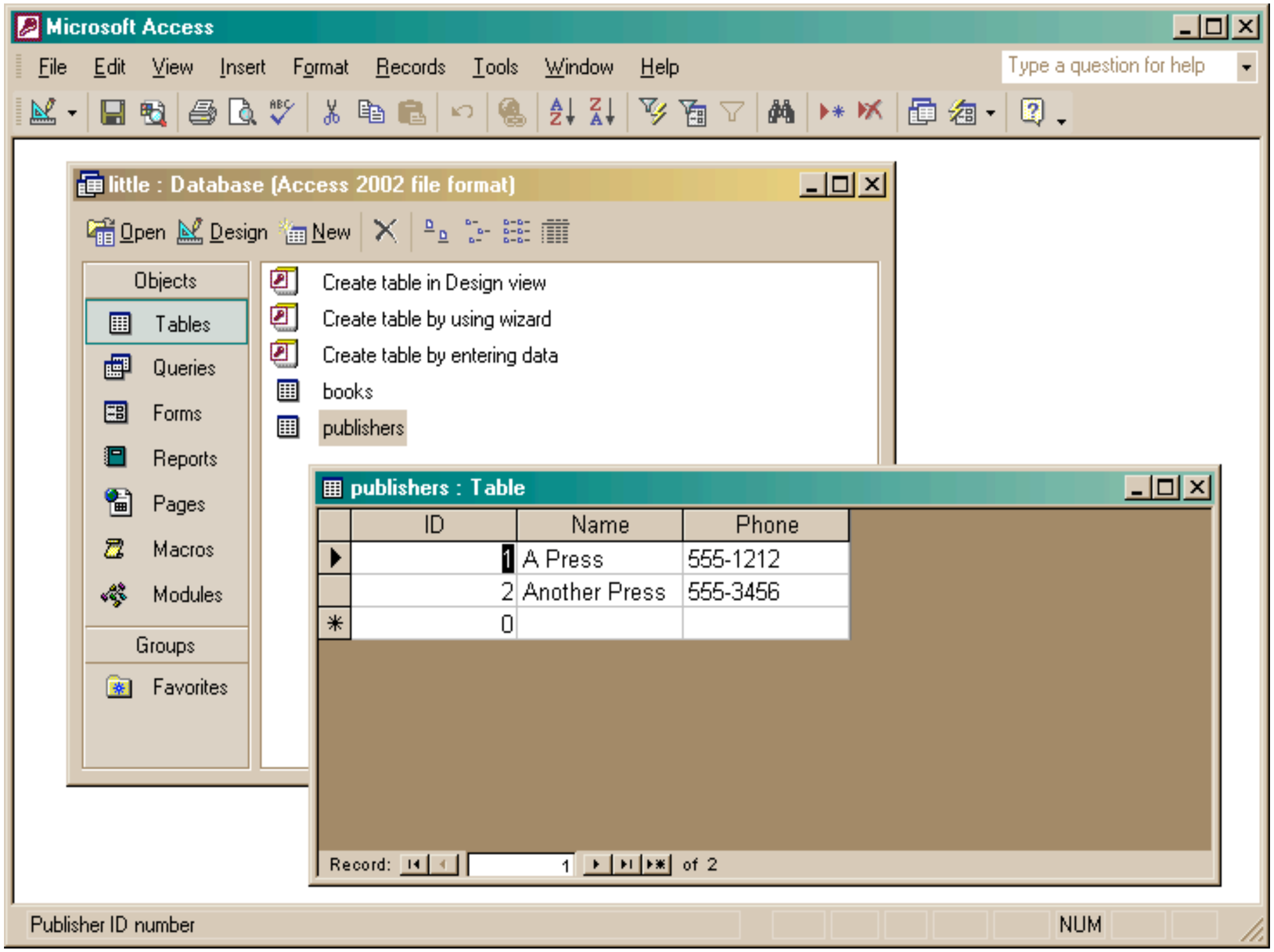


Microsoft Access
File Edit View Insert Format Records Tools Window Help
Type a question for help
little : Database (Access 2002 file format)
Open Design New
Objects
Tables
Queries
Forms
Create table in Design view
Create table by using wizard
Create table by entering data
Books
Books : Table
Record: 3 of 3
NUM

ISBN	Title	Price
1-1	My Reader	$10.00
1-2	Your Reader	$12.00
		$0.00

The unique ISBN number for this book

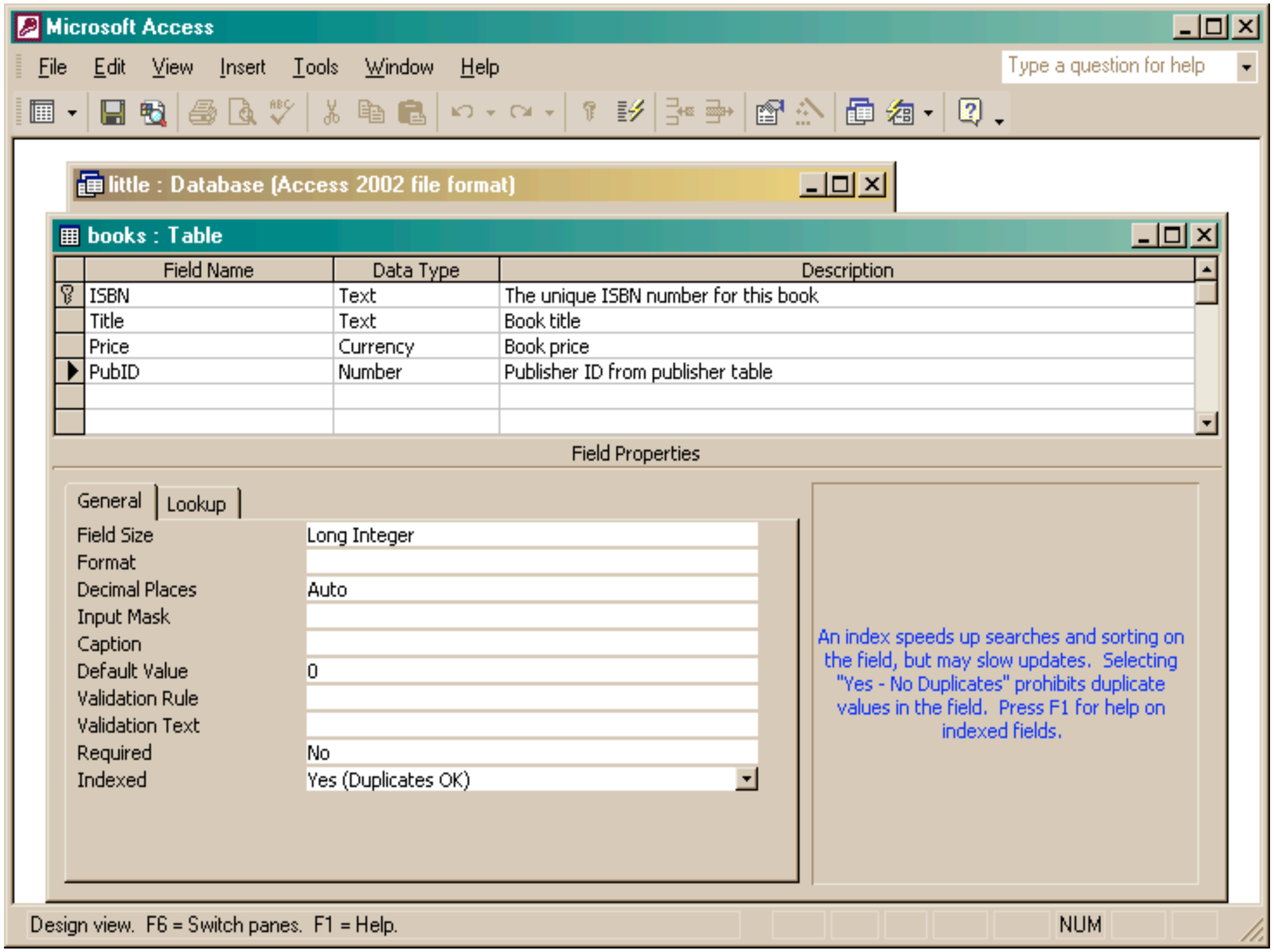
Build another table


Microsoft Access
File Edit View Insert Format Records Tools Window Help
Type a question for help
little : Database (Access 2002 file format)
Open Design New
Objects
Tables
Queries
Forms
Reports
Pages
Macros
Modules
Groups
Favorites
Create table in Design view
Create table by using wizard
Create table by entering data
books
publishers
publishers : Table
Record: 1 of 2
NUM

ID	Name	Phone
1	A Press	555-1212
2	Another Press	555-3456
0		

Publisher ID number

Add publisher ID to books



Field Name	Data Type	Description
ISBN	Text	The unique ISBN number for this book
Title	Text	Book title
Price	Currency	Book price
PubID	Number	Publisher ID from publisher table

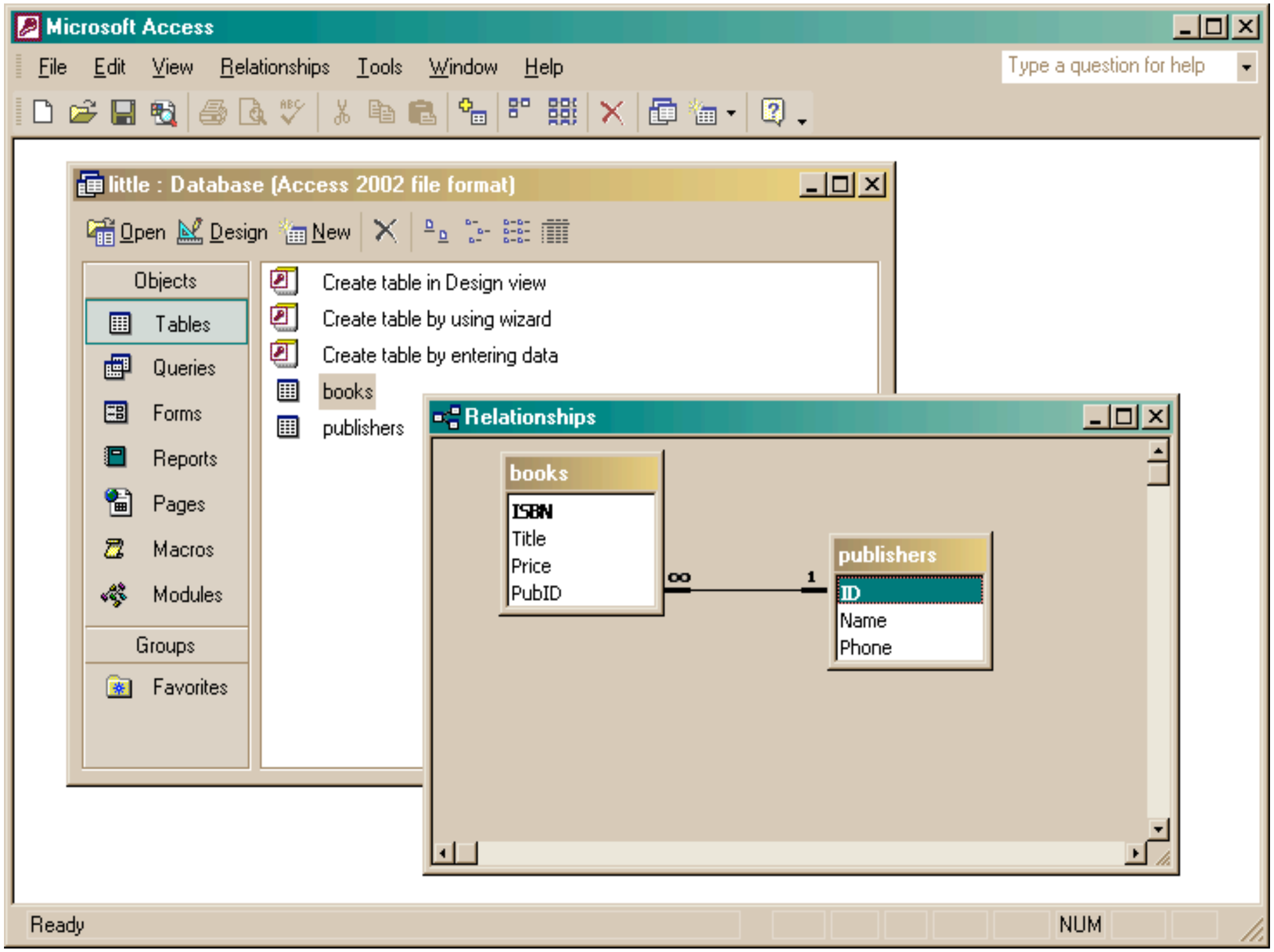
Field Size	Long Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	0
Validation Rule	
Validation Text	
Required	No
Indexed	Yes (Duplicates OK)

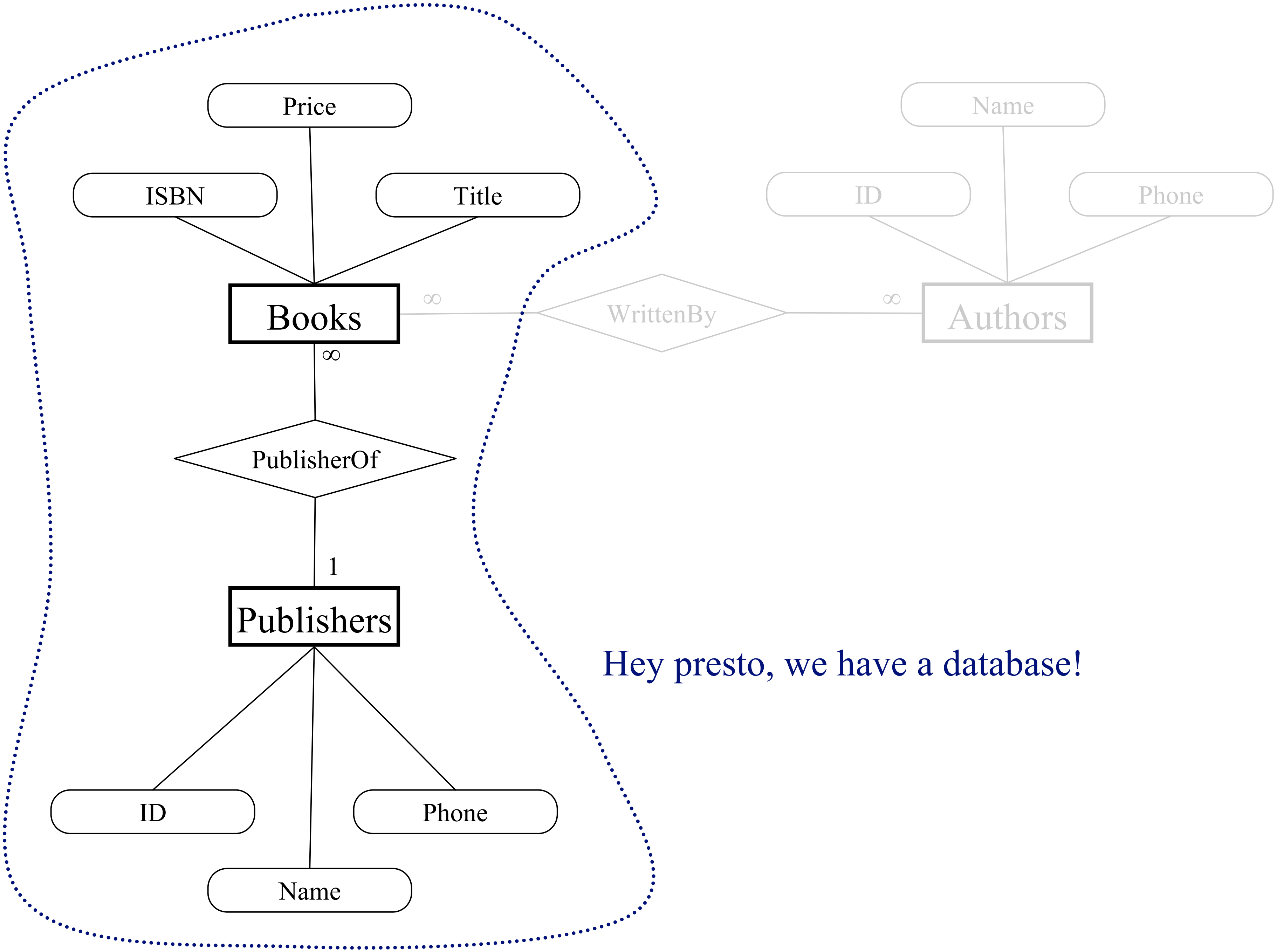
An index speeds up searches and sorting on the field, but may slow updates. Selecting "Yes - No Duplicates" prohibits duplicate values in the field. Press F1 for help on indexed fields.

Design view. F6 = Switch panes. F1 = Help.

NUM

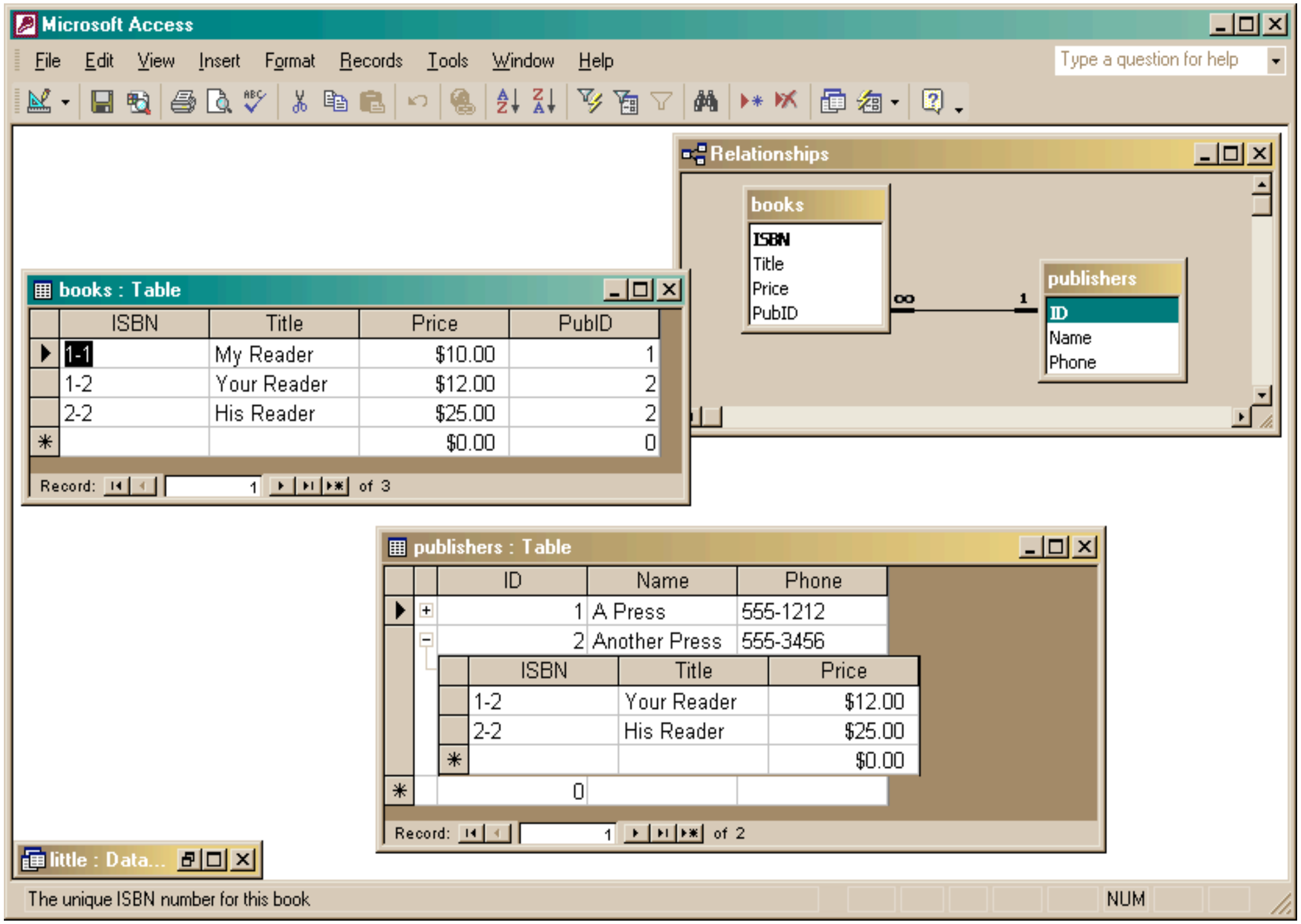
Create the link between the tables





Hey presto, we have a database!

Two tables with a relationship



books : Table

	ISBN	Title	Price	PubID
▶	1-1	My Reader	$10.00	1
	1-2	Your Reader	$12.00	2
	2-2	His Reader	$25.00	2
*			$0.00	0

Record: 1 of 3

publishers : Table

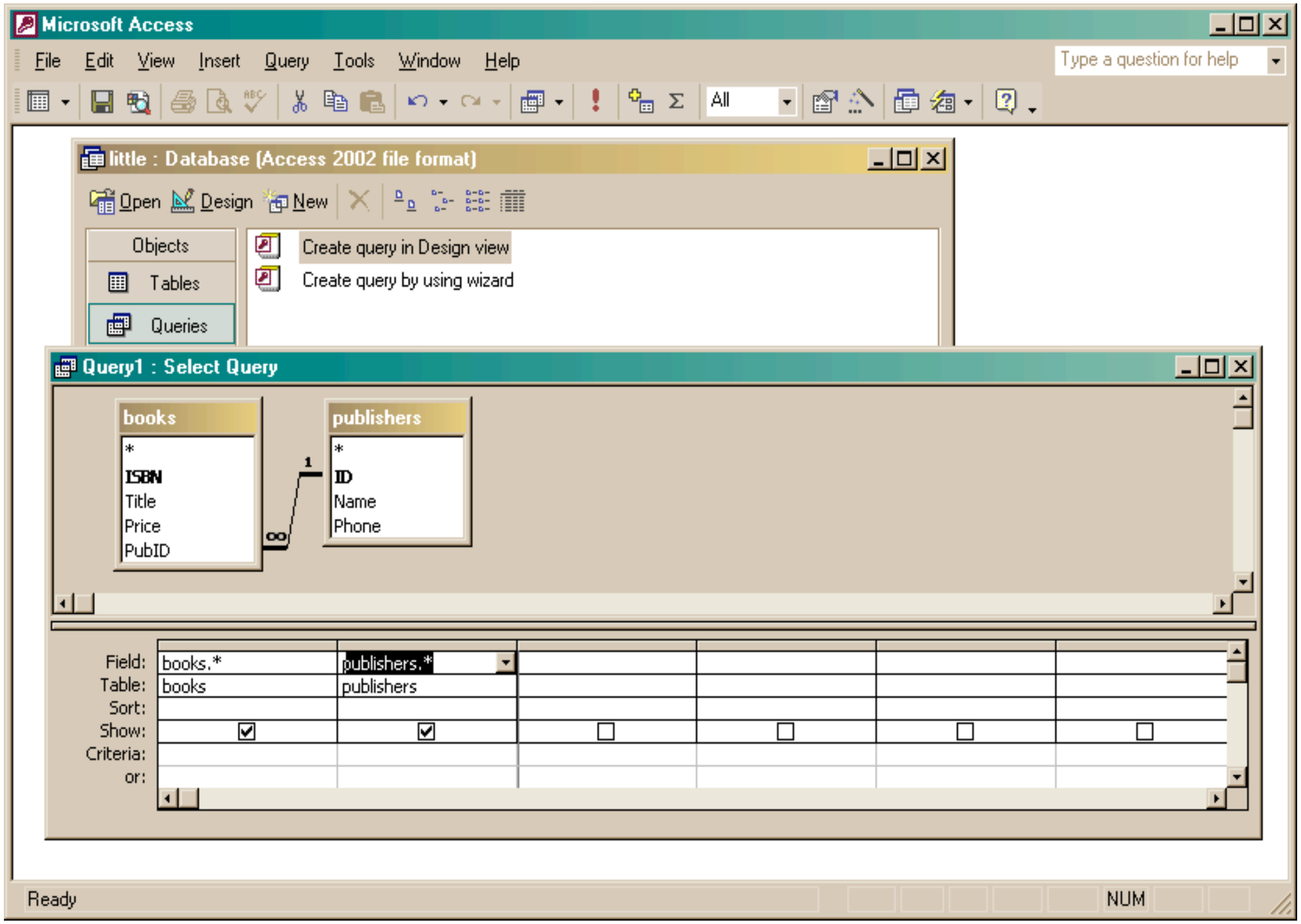
		ID	Name	Phone
▶	+	1	A Press	555-1212
	−	2	Another Press	555-3456
*		0		

	ISBN	Title	Price
	1-2	Your Reader	$12.00
	2-2	His Reader	$25.00
*			$0.00

Record: 1 of 2

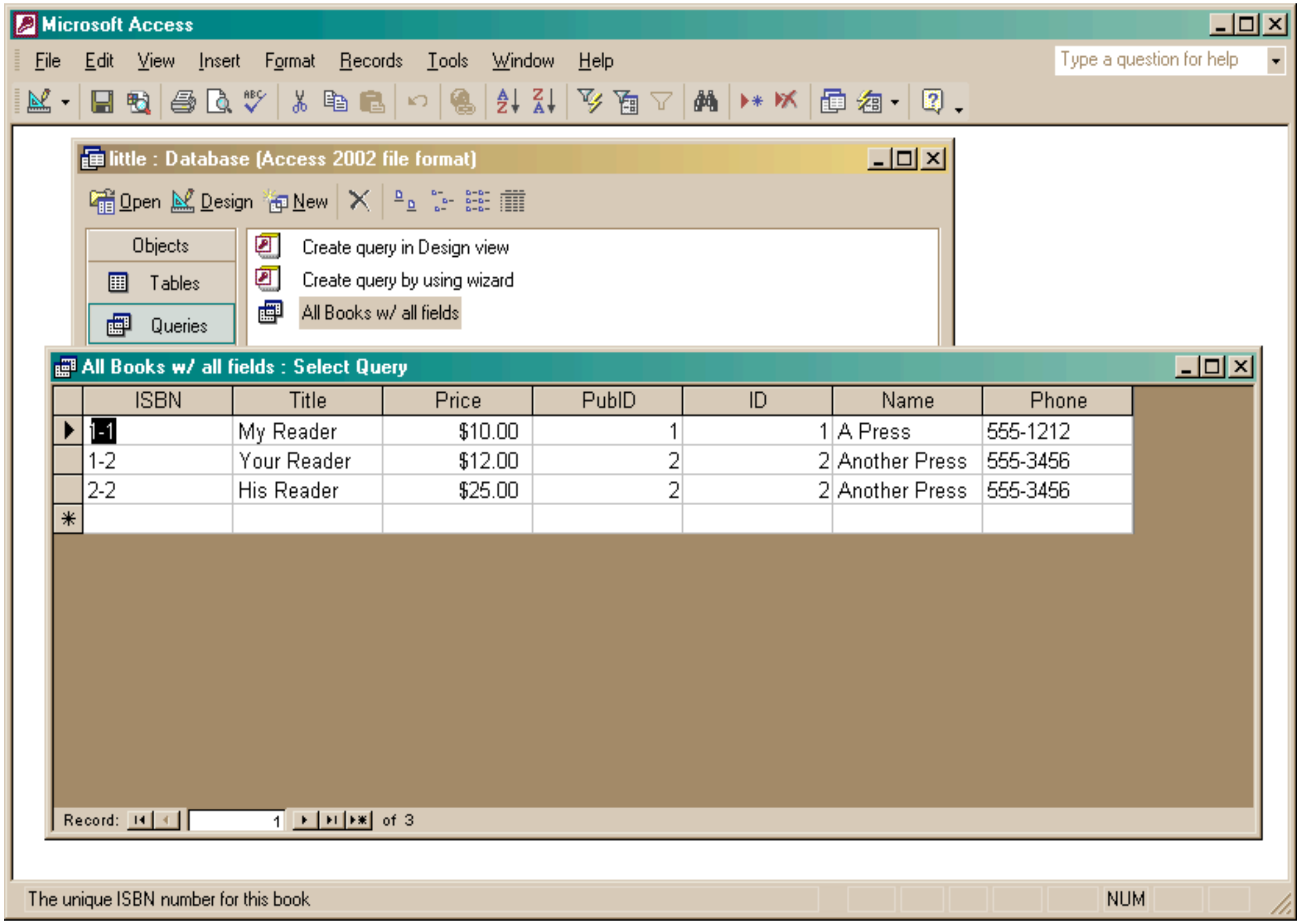
The unique ISBN number for this book

Create a query



Field:	books.*	publishers.*				
Table:	books	publishers				
Sort:						
Show:	☑	☑	□	□	□	□
Criteria:						
or:						

The query produces a new (virtual) table


Microsoft Access
File Edit View Insert Format Records Tools Window Help
Type a question for help
little : Database (Access 2002 file format)
Open Design New
Objects
Tables
Queries

Create query in Design view

Create query by using wizard

All Books w/ all fields

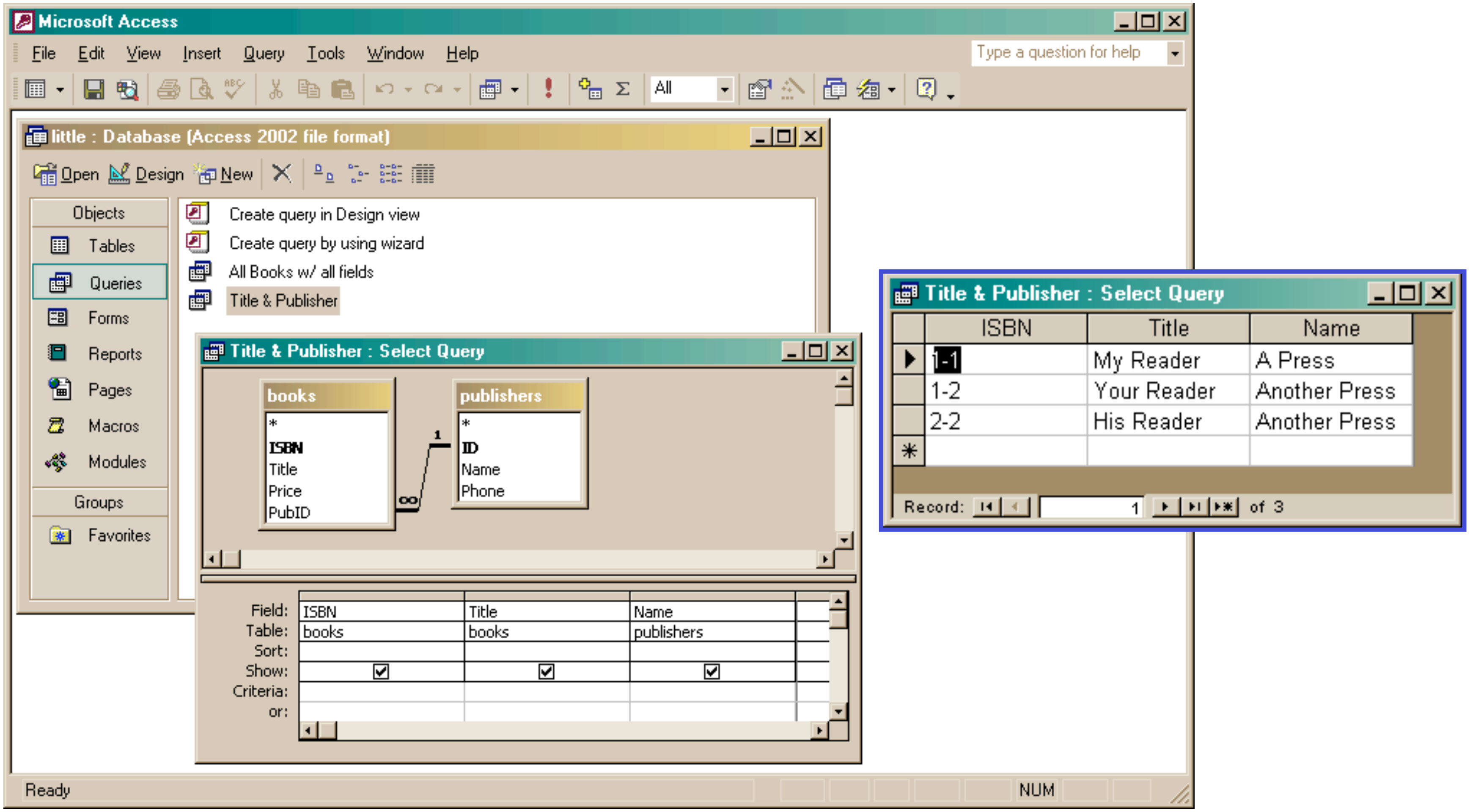
All Books w/ all fields : Select Query

ISBN	Title	Price	PubID	ID	Name	Phone
1-1	My Reader	$10.00	1	1	A Press	555-1212
1-2	Your Reader	$12.00	2	2	Another Press	555-3456
2-2	His Reader	$25.00	2	2	Another Press	555-3456

Record: 1 of 3

The unique ISBN number for this book

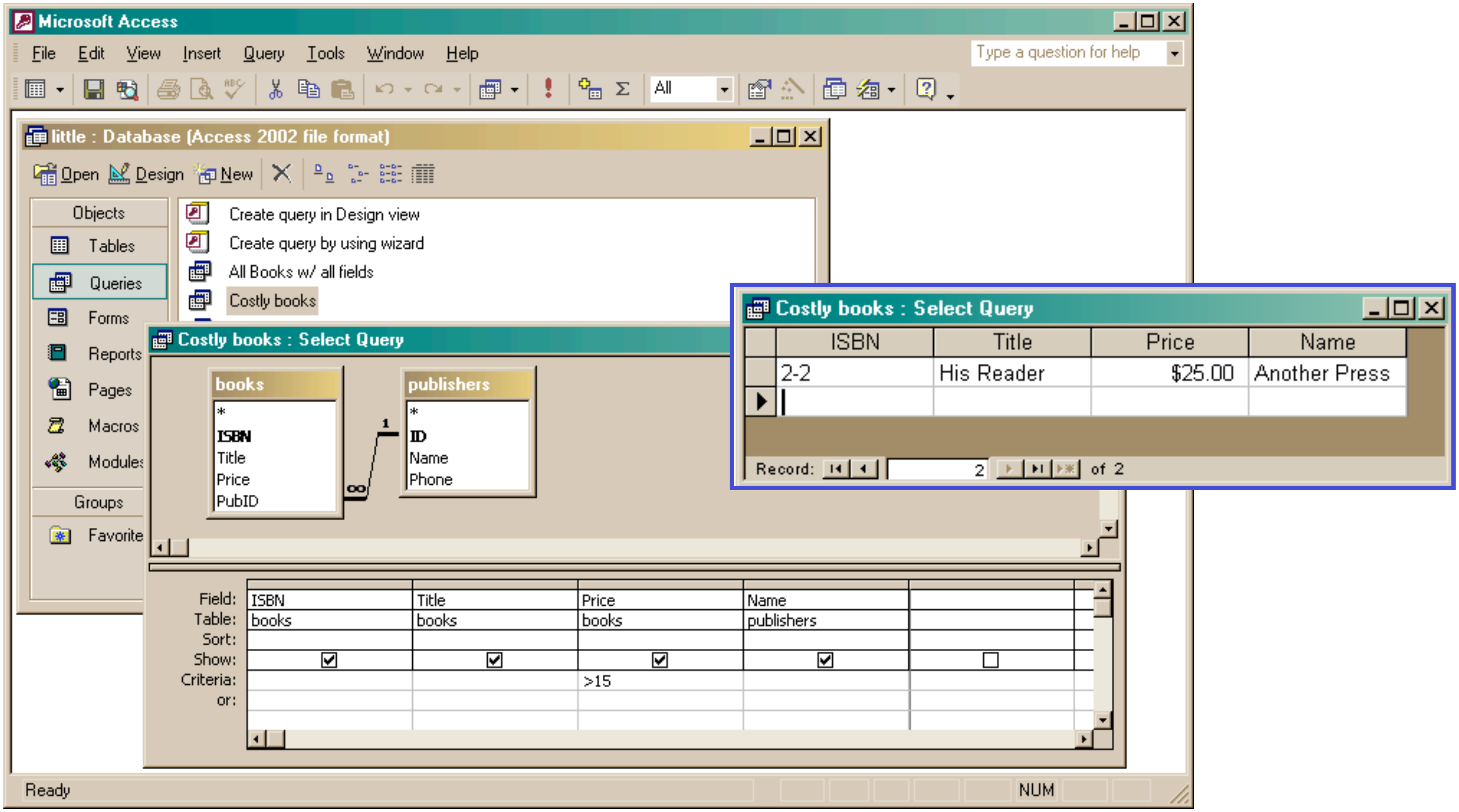
Project (select particular columns)



Field:	ISBN	Title	Name
Table:	books	books	publishers
Sort:			
Show:	☑	☑	☑
Criteria:			
or:			

ISBN	Title	Name
1-1	My Reader	A Press
1-2	Your Reader	Another Press
2-2	His Reader	Another Press

Select particular rows

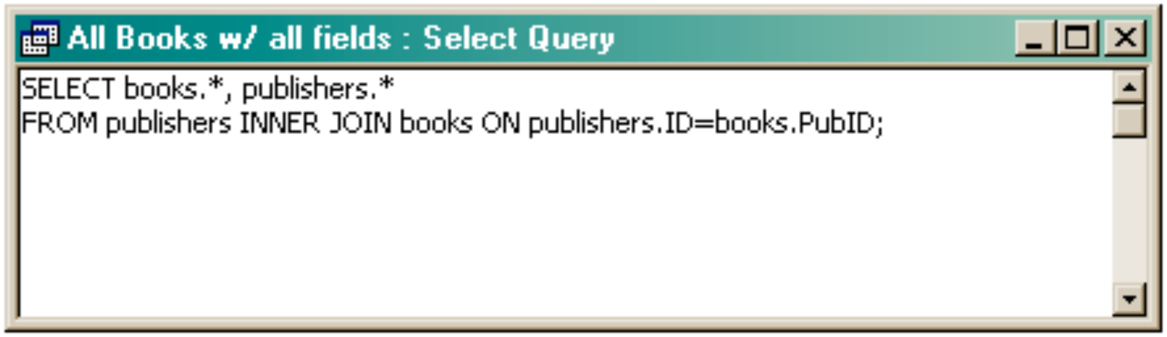


Field:	ISBN	Title	Price	Name
Table:	books	books	books	publishers
Sort:				
Show:	☑	☑	☑	☑
Criteria:			>15	
or:				

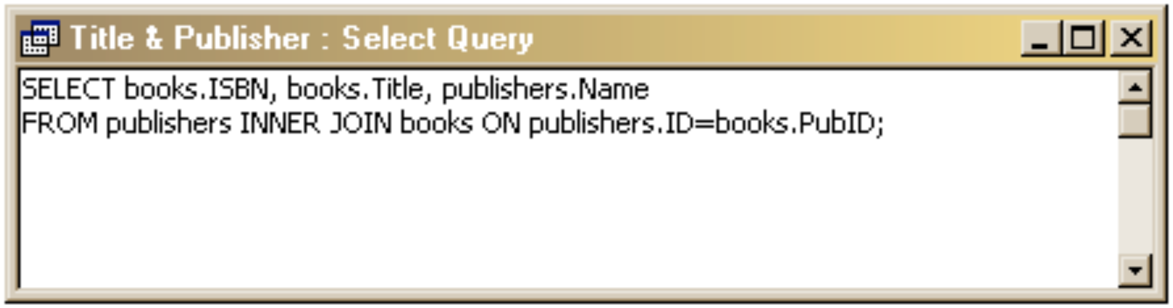
Costly books : Select Query

ISBN	Title	Price	Name
2-2	His Reader	$25.00	Another Press

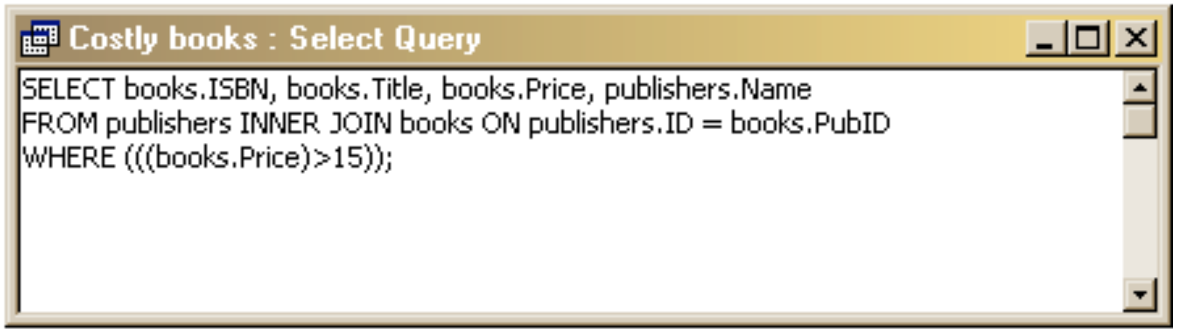
SQL behind the scenes



```
SELECT books.*, publishers.*
FROM publishers INNER JOIN books ON publishers.ID=books.PubID;
```



```
SELECT books.ISBN, books.Title, publishers.Name
FROM publishers INNER JOIN books ON publishers.ID=books.PubID;
```



```
SELECT books.ISBN, books.Title, books.Price, publishers.Name
FROM publishers INNER JOIN books ON publishers.ID = books.PubID
WHERE (((books.Price)>15));
```