



Database Intro

INFO/CSE 100, Spring 2006
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.

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}

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					

Schema for Example table:

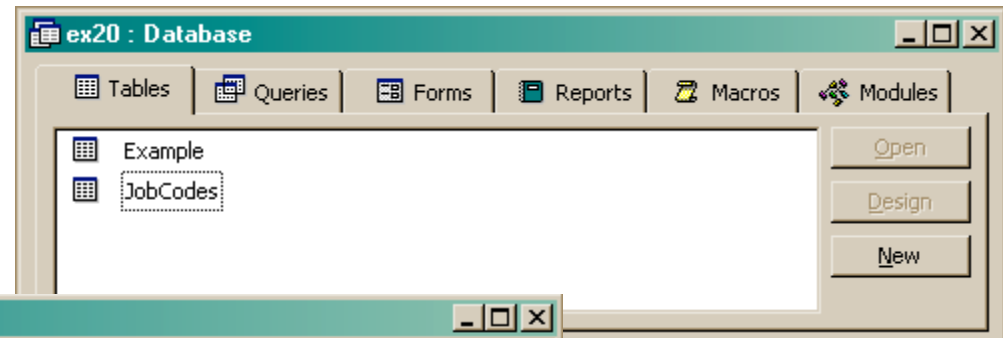
ID	number	unique number(Key)
Last	text	person's last name
First	text	person's first name
JobCode	number	current position
Hire	date	first day on job
...		

instance

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					

Record: 7 of 7

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

Record: 1 of 4

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 location
 - Inconsistent data is worse than no data
 - Cut down on the typos and mis-keyed entries
 - » Keep a *single copy* of any data
 - Reduces memory and data processing costs
 - if it is needed in multiple places, associate it with a key and store key rather than the data
 - » Effort to update is high

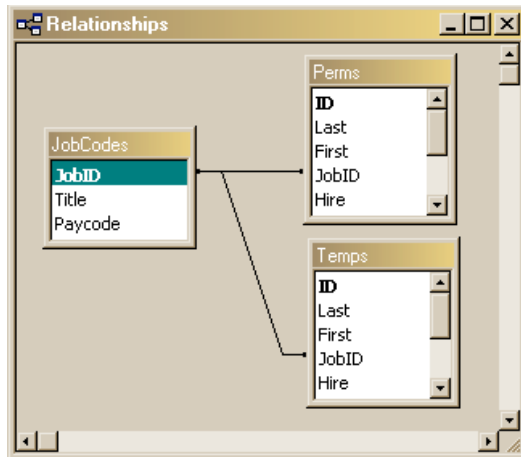




Relationships between tables

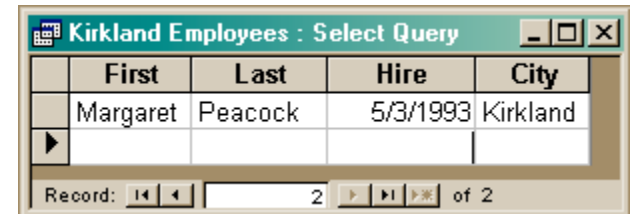
ID	Last	First	JobID	Hire	Street	City	State	Country
1	Davalino	Nancy	0	01-May-92	507 20th Ave E	Seattle	WA	USA
2	Fuller	Andrew	3	14-Aug-92	908 W. Capital Way	Seattle	WA	USA
3	Wooster	Berton	1	01-Apr-93	722 Moss Bay Blvd	Seattle	WA	USA
4	Peacock	Margaret	2	03-May-93	4110 Old Redmond Rd	Kirkland	WA	USA
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
	CEO	8
1	VP	7
2	Engineer	4
3	Administrative	6
*	0	0



“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 $\leq 4:00$?
 - Table of runners



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

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/Join** -- 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					

Record: 1 of 6

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					

Record: 1 of 2



Project

- Project creates a table from the columns of another table

Project Last, First From Example

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					

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

The screenshot displays three database tables in a grid view:

- Perms : Table** (6 records):

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	Seattle	WA	USA
3	Wooster	Berton	1	01-Apr-93	722 Moss Bay E	Seattle	WA	USA
4	Peacock	Margaret	2	03-May-93	4110 Old Redm	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					
- Temps : Table** (6 records):

ID	Last	First	JobID	Hire	Street	City	State	Country
101	Soggy	Peter	0	01-Jun-04	1300 20th Ave W	Seattle	WA	USA
102	Morken	Xavier	3	14-Sep-03	100 Eastlake Dr	Seattle	WA	USA
103	Wilshire	Bruce	1	01-Mar-98	34 15th Ave NE	Seattle	WA	USA
104	Brazely	Tanya	2	03-Mar-02	103 25th Ave NW	Seattle	WA	USA
105	Compton	Sarah	3	17-Nov-99	4034 NW 50th St	Seattle	WA	USA
106	Zanzy	Ovid	2	12-Jan-99	4502 NW 52nd	Seattle	WA	USA
0			0					
- All employees : Union Query** (12 records):

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 E	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 W	Seattle	WA	USA
102	Morken	Xavier	3	9/14/2003	100 Eastlake Dr	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 NW	Seattle	WA	USA
105	Compton	Sarah	3	11/17/1999	4034 NW 50th St	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

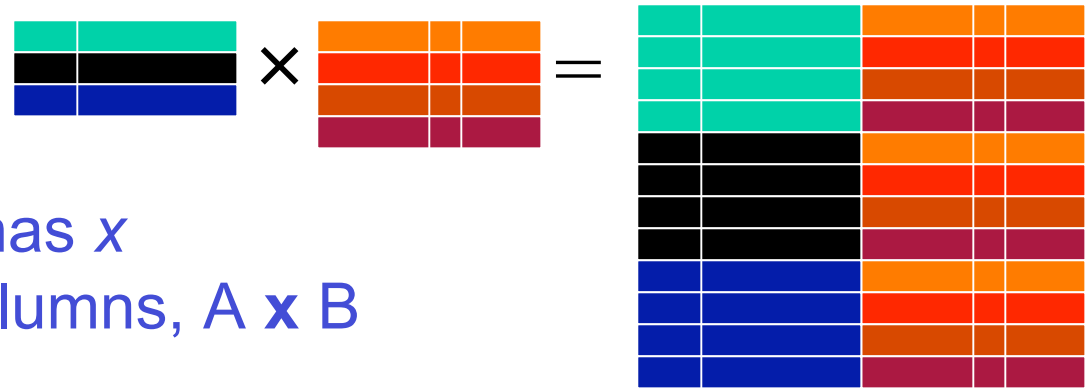
WestCoast : Table		
Name	Capitol	Sight
Washington	Olympia	Mt. Rainier
Oregon	Salem	Crater Lake
California	Sacramento	Golden Gate

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 \times Reds



Column Rule: If A has x columns, B has y columns, $A \times B$ has $x+y$ columns

Row Rule: If A has m rows, B has n rows $A \times B$ has mn rows

Join

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

Employee List = Perms ⋈ 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					

Record: 1 of 6

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

Record: 5 of 5

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
*					

Record: 1 of 6

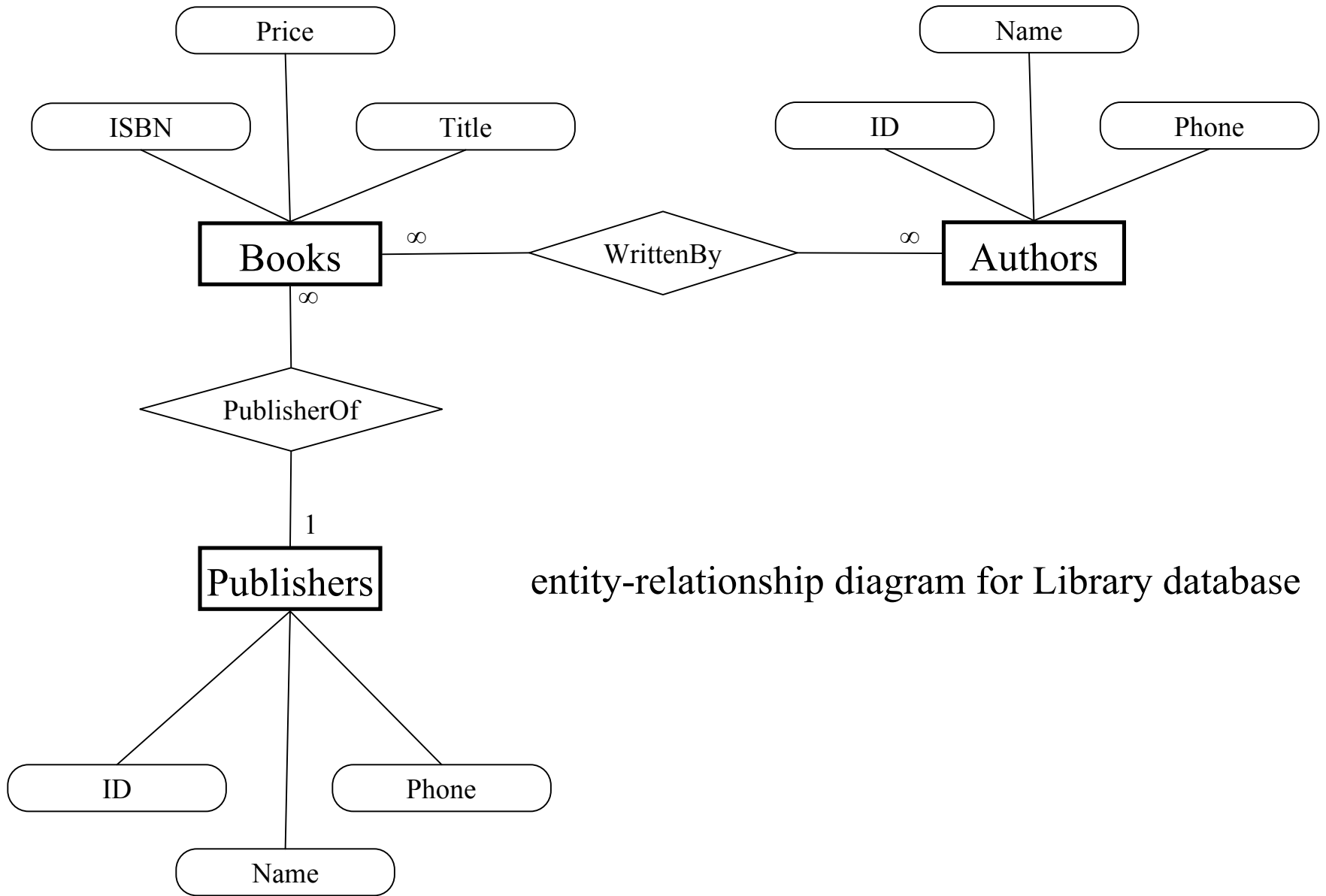
DB Operations

- 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

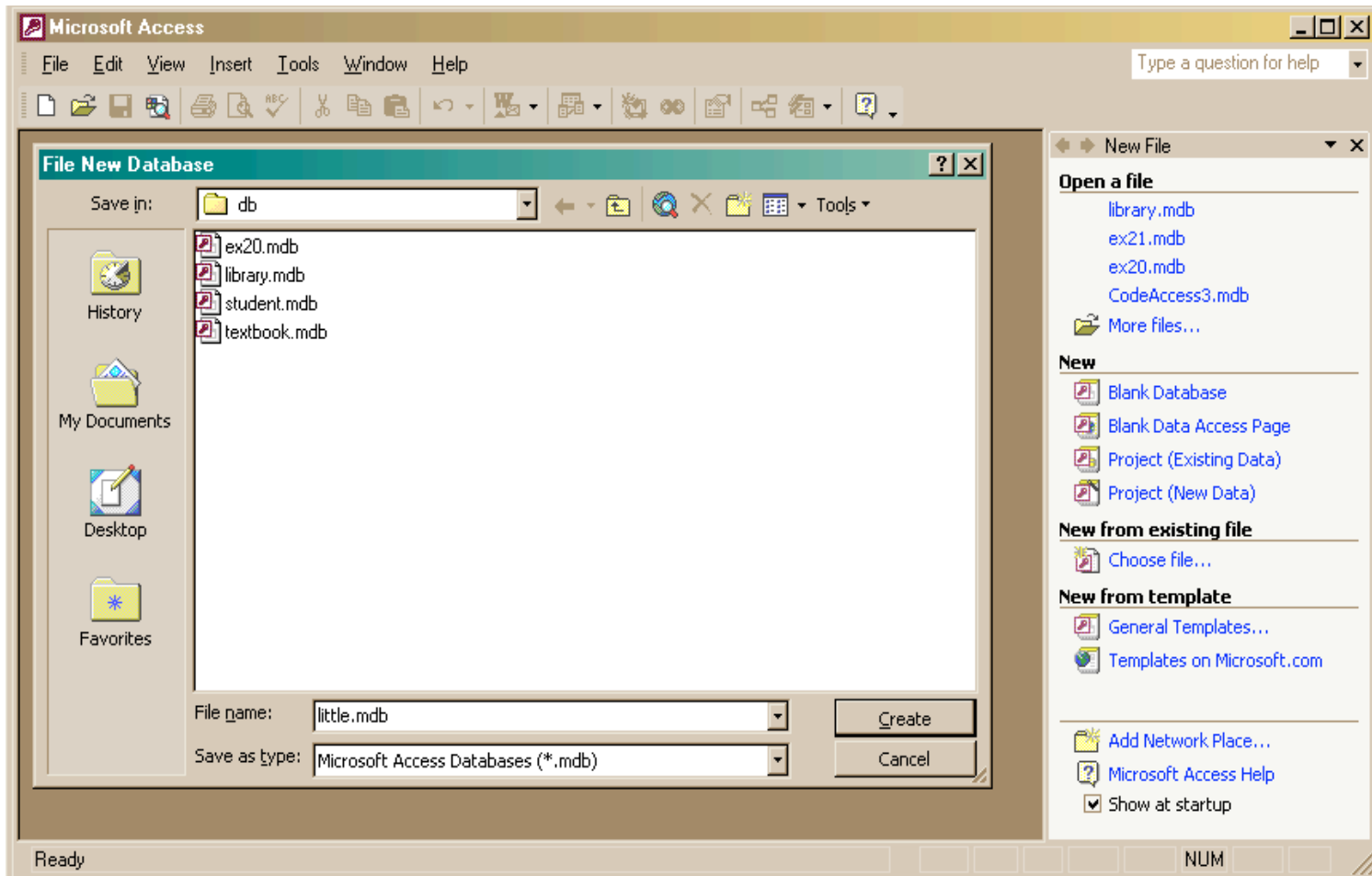
- A database contains one or more *tables*
 - » Tables include *entities* with *attributes*
 - » There are *relationships* defined between the entities in the various tables
 - » Retrieve information from the tables using *queries*
 - » Create GUI front ends (*forms* and *reports*) for users
- First, design the database or create the schema
 - » What are the entities?
 - » What are the attributes of each entity?
 - » What are the relationships between tables?



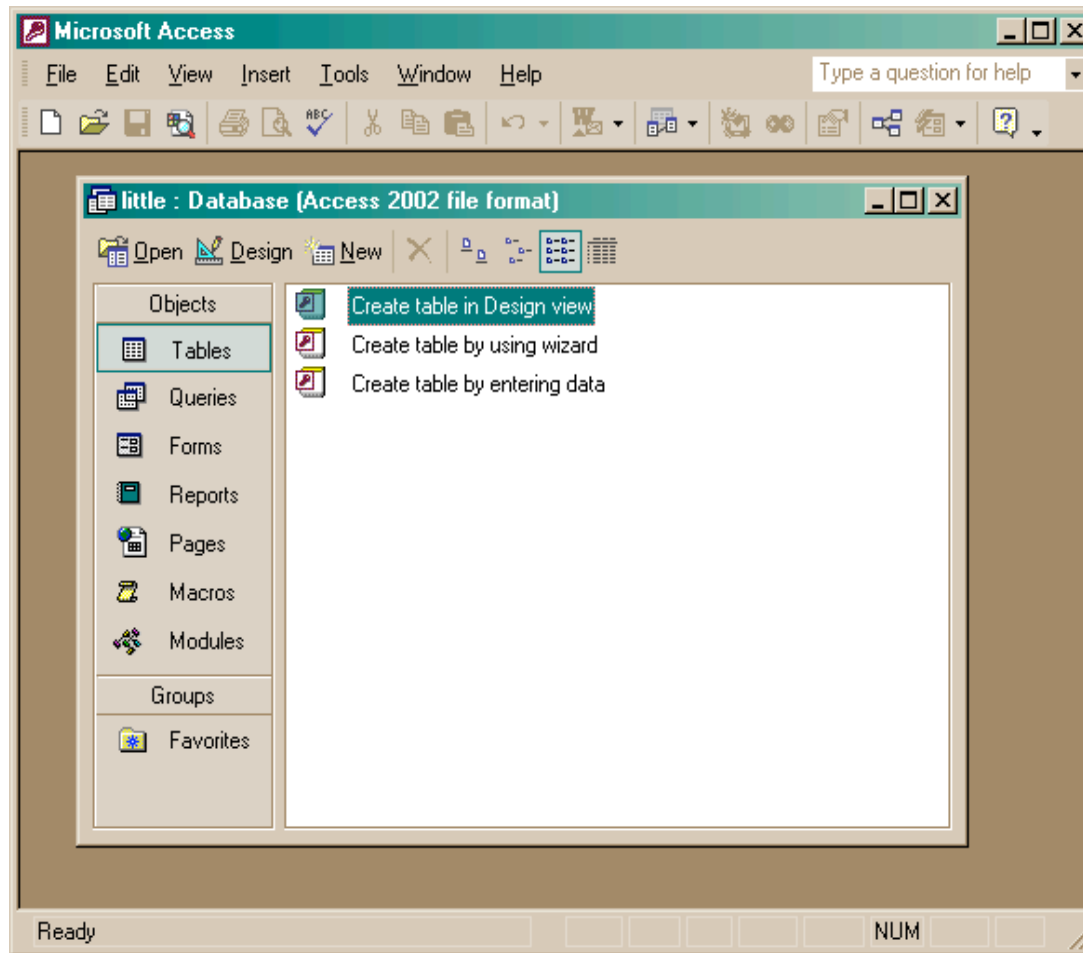
entity-relationship diagram for Library database



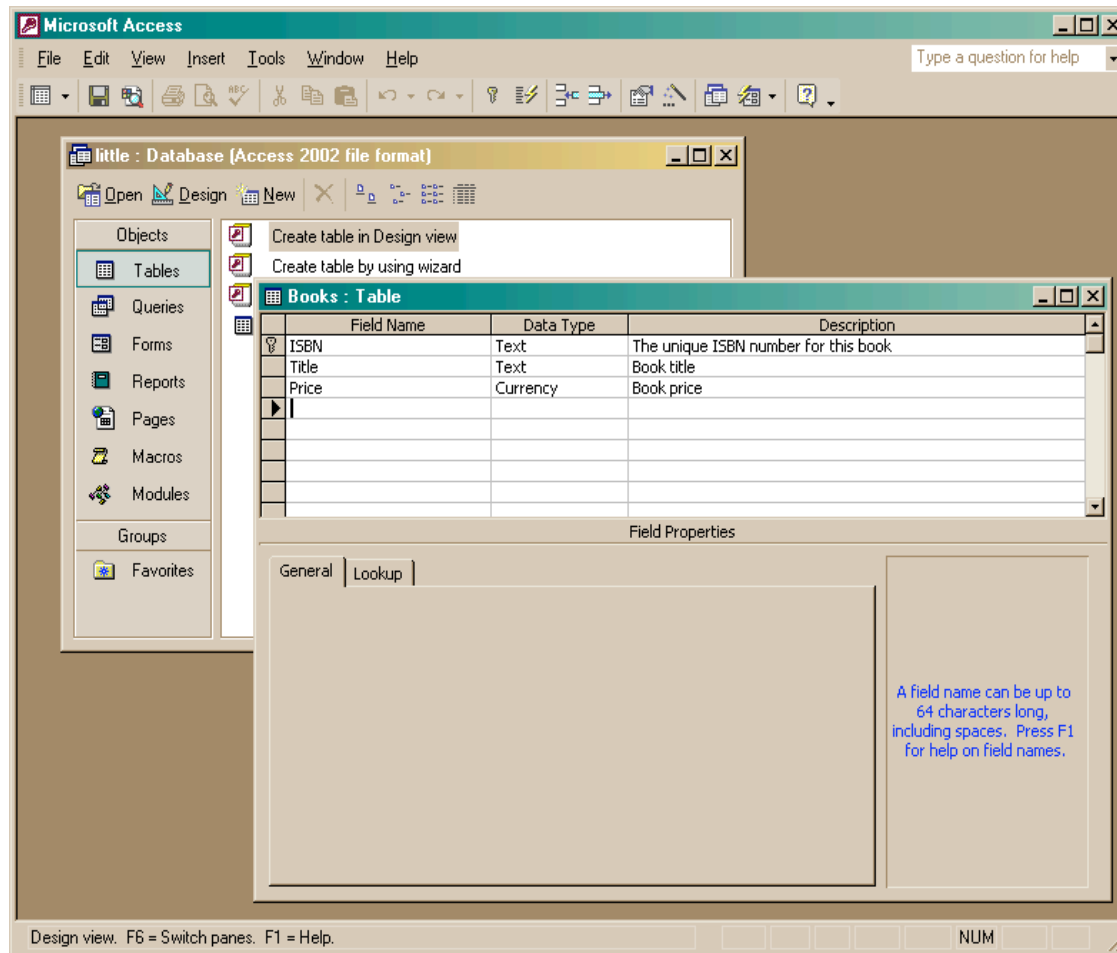
Create a new database



Create a new table in the database



Creating a table in Design view





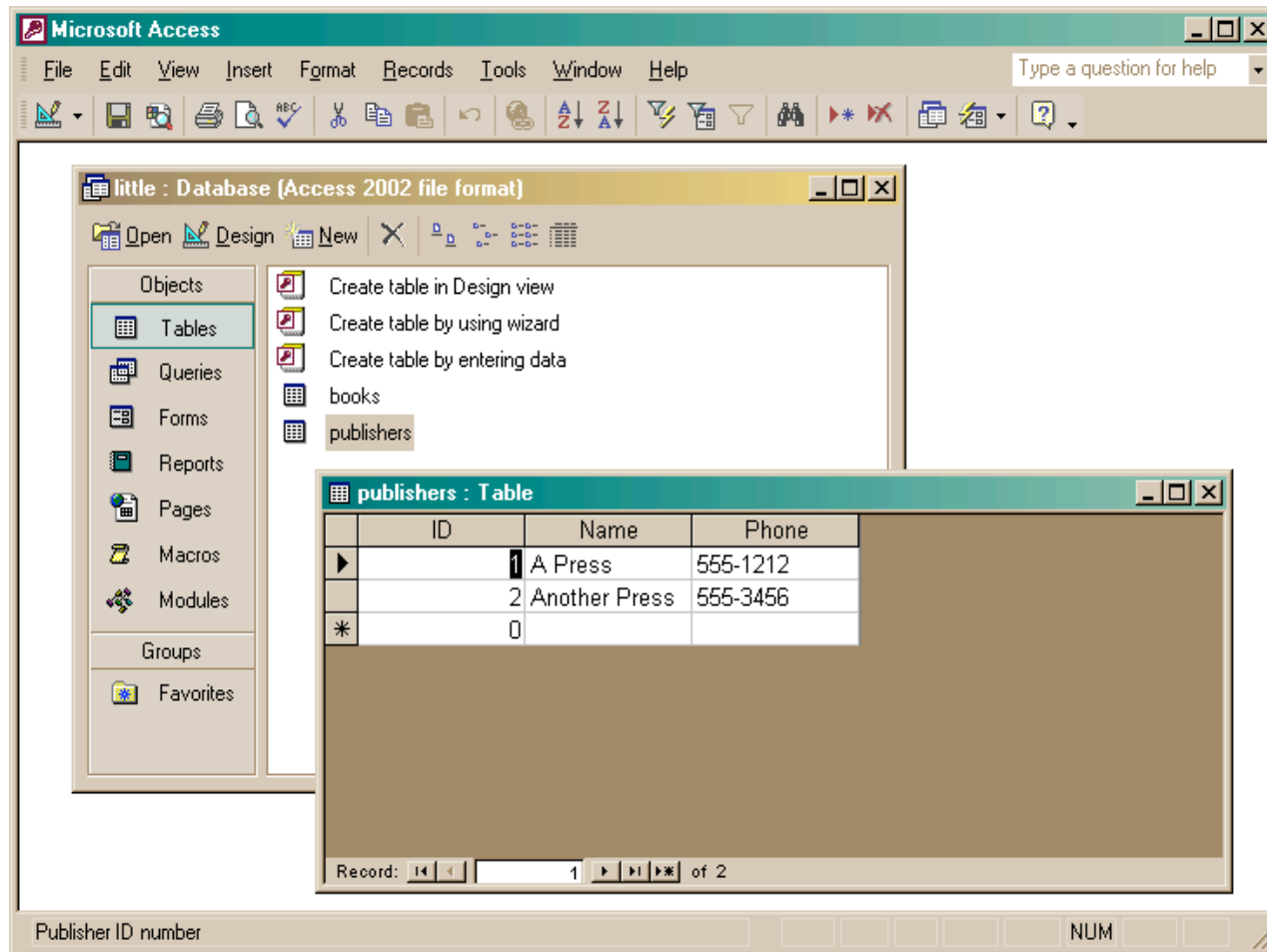
Entering Table Data

The screenshot shows the Microsoft Access interface. A window titled 'Database [Access 2002 file format]' is open, displaying the 'Objects' pane on the left with 'Tables' selected. A 'Books : Table' window is open, showing a table with the following data:

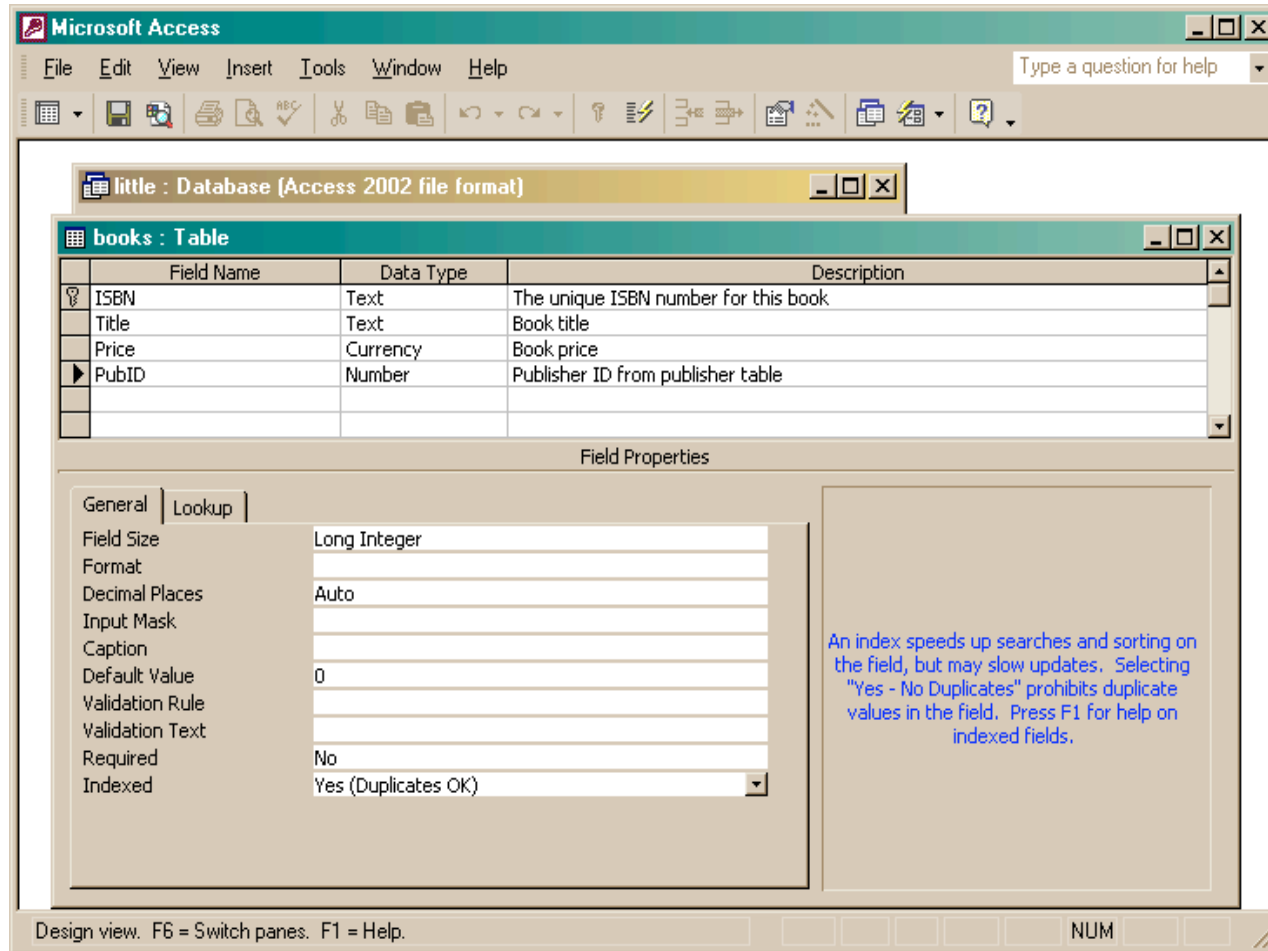
ISBN	Title	Price
1-1	My Reader	\$10.00
1-2	Your Reader	\$12.00
		\$0.00

The status bar at the bottom indicates 'Record: 3 of 3' and 'NUM'.

Build another table



Add publisher ID to books



The screenshot shows the Microsoft Access interface. The main window displays the 'books' table in Design view. The table has the following fields:

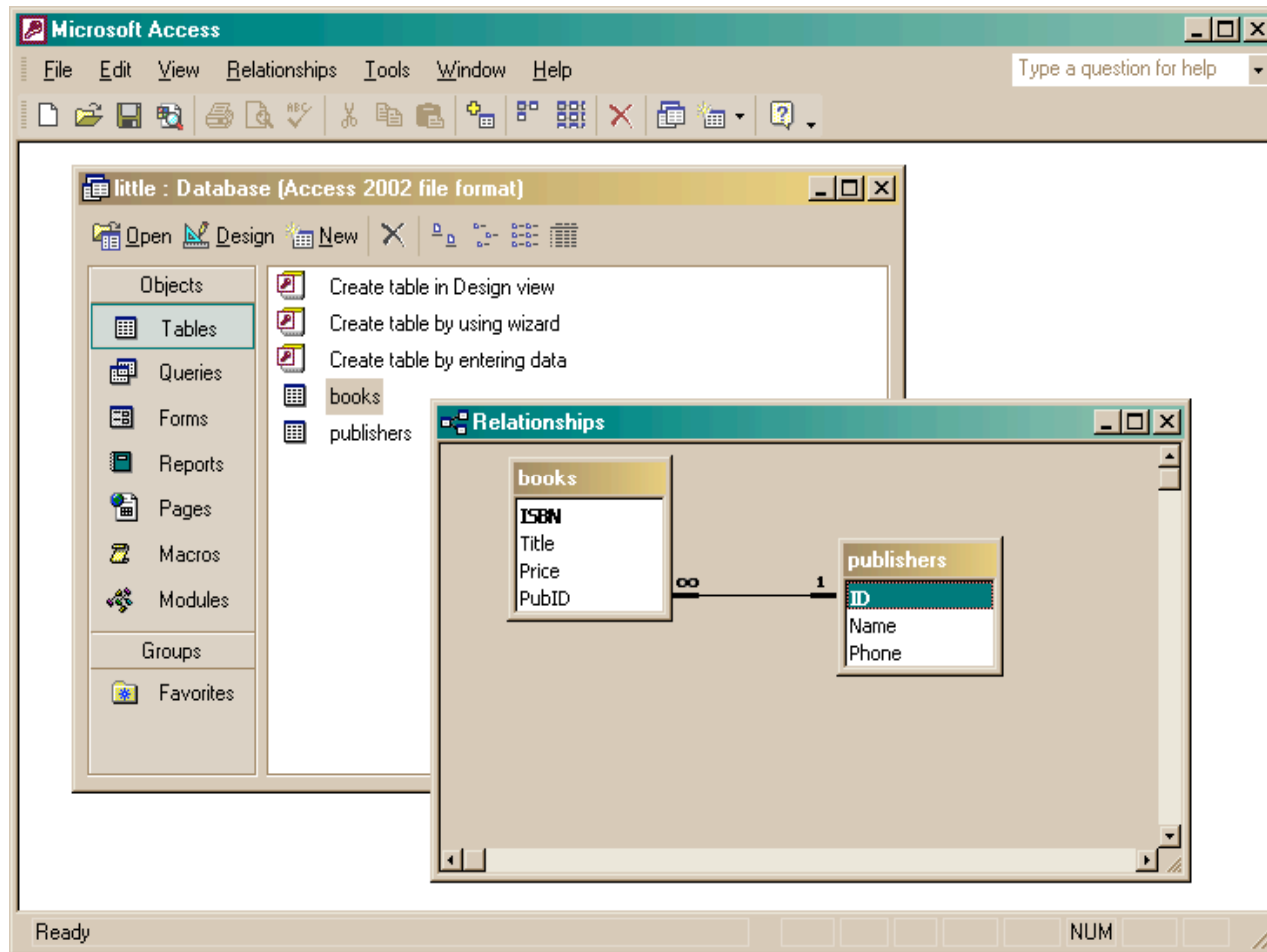
Field Name	Data Type	Description
ISBN	Text	The unique ISBN number for this book
Title	Text	Book title
Price	Currency	Book price
PubID	Number	Publisher ID from publisher table

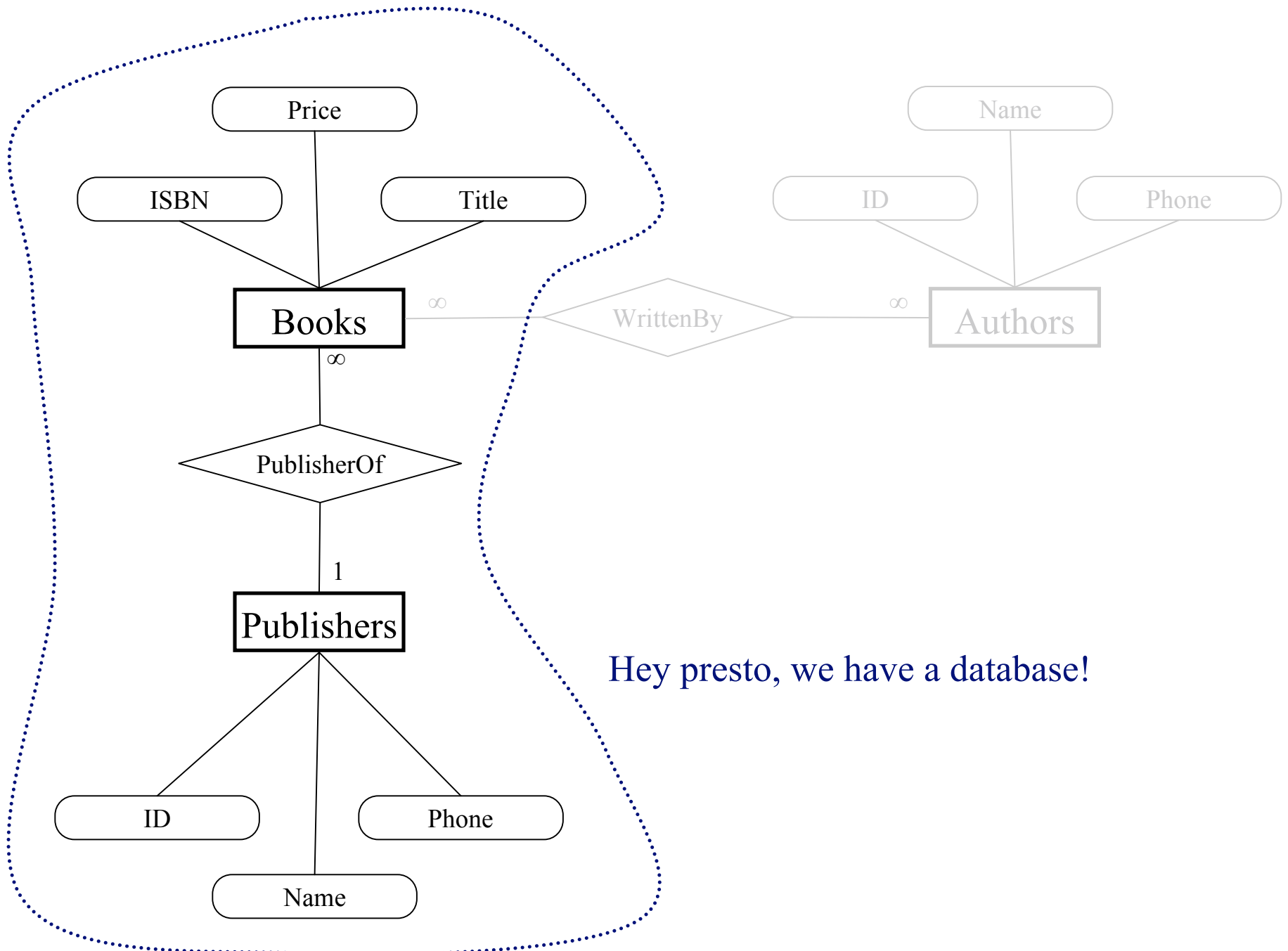
The 'Field Properties' pane for the 'PubID' field is open, showing the following properties:

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

A note on the right side of the Field Properties pane reads: "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."

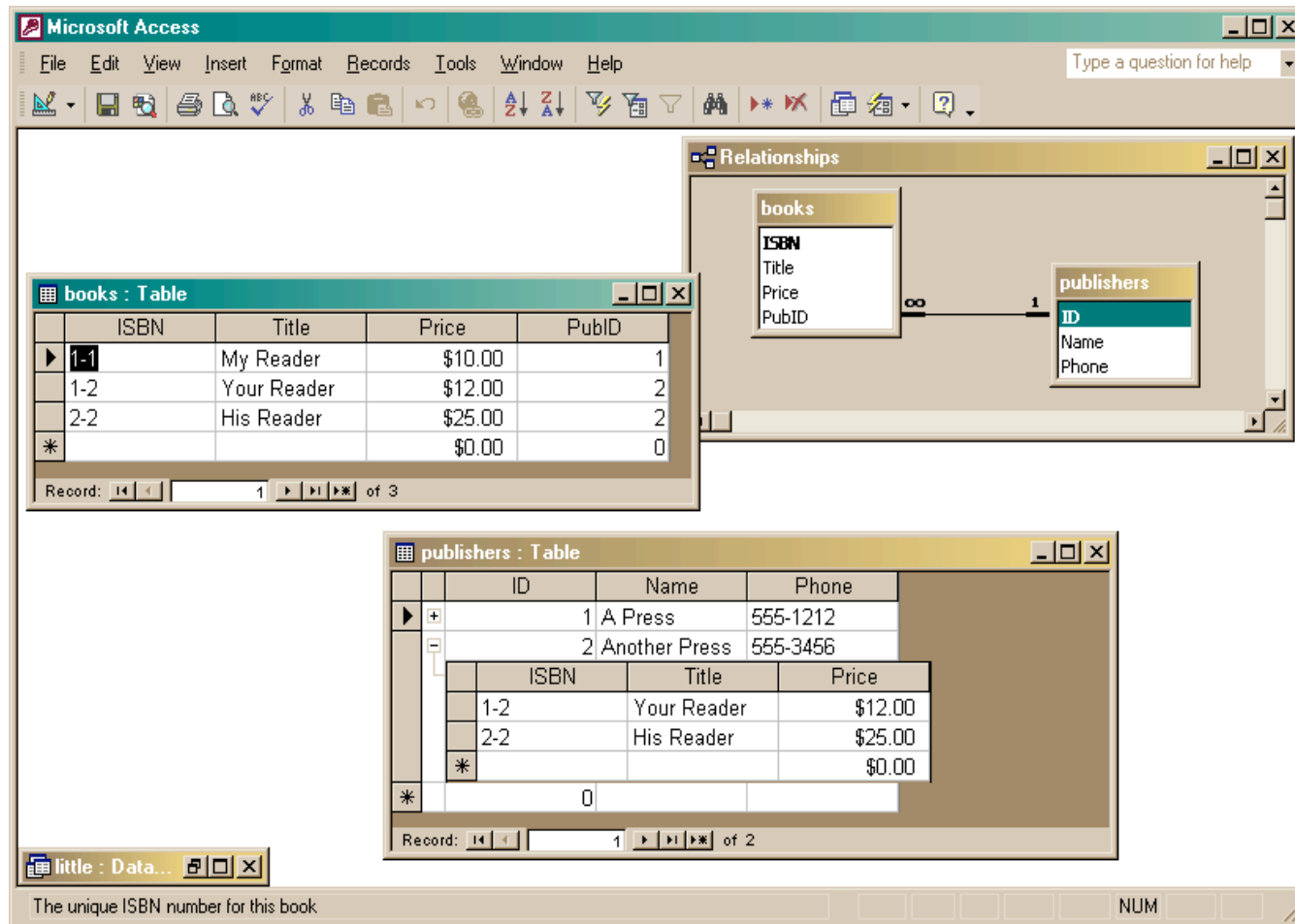
Create the link between the tables





Hey presto, we have a database!

Two tables with a relationship



The screenshot shows Microsoft Access with three windows open:

- 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
- publishers : Table**

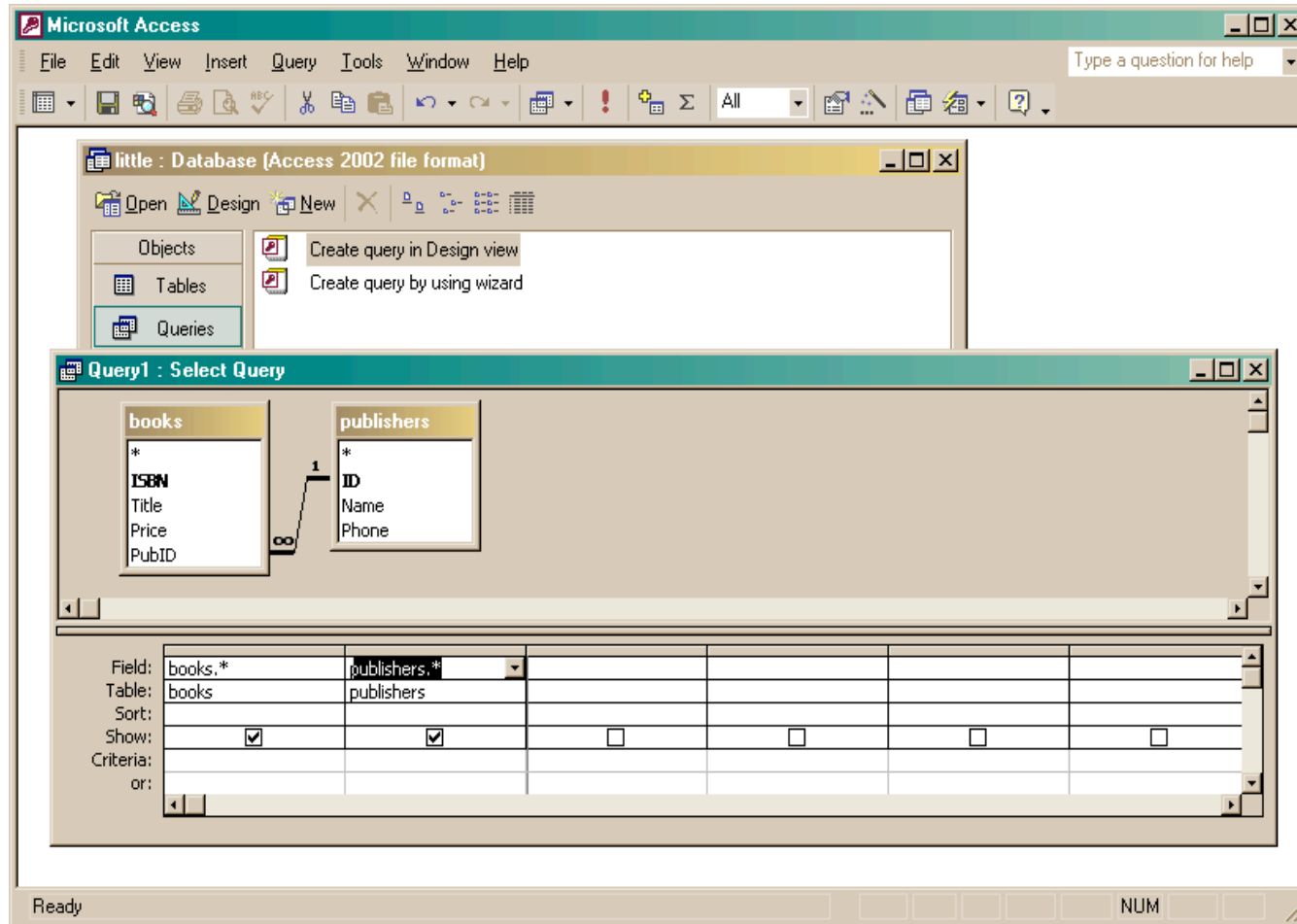
	ID	Name	Phone
▶ +	1	A Press	555-1212
-	2	Another Press	555-3456
		ISBN	Title
		1-2	Your Reader
		2-2	His Reader
*			\$0.00
*			0
- Relationships**

Diagram showing a one-to-many relationship between 'publishers' (ID) and 'books' (PubID). The 'publishers' side has a '1' and the 'books' side has an '∞'.

At the bottom, a status bar indicates: "The unique ISBN number for this book NUM"



Create a query





The query produces a new (virtual) table

The screenshot shows the Microsoft Access interface. A window titled "All Books w/ all fields : Select Query" is open, displaying a table with the following data:

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
*						

At the bottom of the window, it says "Record: 1 of 3". Below the window, there is a status bar with the text "The unique ISBN number for this book" and a field labeled "NUM".



Project (select particular columns)

The screenshot shows Microsoft Access with a database named 'little : Database (Access 2002 file format)'. The 'Queries' pane on the left shows a query named 'Title & Publisher'. The main window displays the 'Title & Publisher : Select Query' in Design view, showing a relationship between the 'books' table (with fields ISBN, Title, Price, PubID) and the 'publishers' table (with fields ID, Name, Phone). A one-to-many relationship is indicated between the 'books' table and the 'publishers' table.

Below the design view, a table shows the field specifications for the query:

Field:	ISBN	Title	Name
Table:	books	books	publishers
Sort:			
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:			
or:			

Overlaid on the right is a data view window titled 'Title & Publisher : Select Query' showing the following data:

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

Record: 1 of 3



Select particular rows

The screenshot shows Microsoft Access with a query design view for 'Costly books'. The design view shows two tables, 'books' and 'publishers', connected by a one-to-many relationship. The 'books' table has fields: ISBN, Title, Price, and PubID. The 'publishers' table has fields: ID, Name, and Phone. The criteria for the query are: Price > 15. Below the design view is a data table view showing the results of the query.

Field:	ISBN	Title	Price	Name
Table:	books	books	books	publishers
Sort:				
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Criteria:			>15	
or:				

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

Record: 2 of 2

SQL behind the scenes

```
All Books w/ all fields : Select Query
SELECT books.*, publishers.*
FROM publishers INNER JOIN books ON publishers.ID=books.PubID;
```

```
Title & Publisher : Select Query
SELECT books.ISBN, books.Title, publishers.Name
FROM publishers INNER JOIN books ON publishers.ID=books.PubID;
```

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