

# Databases for beginners

02-DBBA

Marek Kręglewski

# About the course

- Steven Roman, Access Database Design & Programming (3rd Edition), O'Reilly 2002
- Curtis D. Frye, Microsoft Office Access 2007 Plain & Simple, Microsoft Press 2007
- <http://office.microsoft.com/en-us/access-help/#>
- <http://office.microsoft.com/en-us/access-help/database-basics-HA010064450.aspx#BMpartsofadatabase>

# Basics about a database?

- A **database** is a structured collection of records.
- Database Management System (**DBMS**)
  - add, remove, update records
  - retrieve data that match certain criteria
  - cross-reference data in different tables
  - perform complex aggregate calculation
- Database consists of columns (attributes) and rows (records).
- Databases versus spreadsheets
  - easy manipulation of data

# Single table database

ISBN	Title	AuID	AuName	AuTel	PubID	PubName	PubTel	Price
0-99-999999-9	Emma	1	Austen	111-111-1111	1	Big House	123-456-7890	20.00 zł
0-91-335678-7	Faerie Queen	7	Spenser	777-777-7777	1	Big House	123-456-7890	17.00 zł
0-91-045678-5	Hamlet	5	Shakespeare	555-555-5555	2	Alpha Press	999-999-9999	20.00 zł
0-103-45678-9	Iliad	3	Homer	333-333-3333	1	Big House	123-456-7890	25.00 zł
0-555-55555-9	Macbeth	5	Shakespeare	555-555-5555	2	Alpha Press	999-999-9999	12.00 zł
0-55-123456-9	Main Street	10	Jones	123-333-3333	3	Small House	714-000-0000	23.00 zł
0-55-123456-9	Main Street	9	Smith	123-222-2222	3	Small House	714-000-0000	23.00 zł
0-12-333433-3	On Liberty	8	Mill	888-888-8888	1	Big House	123-456-7890	25.00 zł
0-321-32132-1	Balloon	2	Sleepy	222-222-2222	3	Small House	714-000-0000	34.00 zł
0-321-32132-1	Balloon	4	Snoopy	444-444-4444	3	Small House	714-000-0000	34.00 zł
0-321-32132-1	Balloon	11	Grumpy	321-321-0000	3	Small House	714-000-0000	34.00 zł

# Disadvantages of a single table database

- Redundancy of data
- Problem with complex data
- Problems in updating in bulk (new phone number)
- Problems in adding incomplete data (new publisher)
- Problems in removing group of data (all books from the publisher)

## Solution:

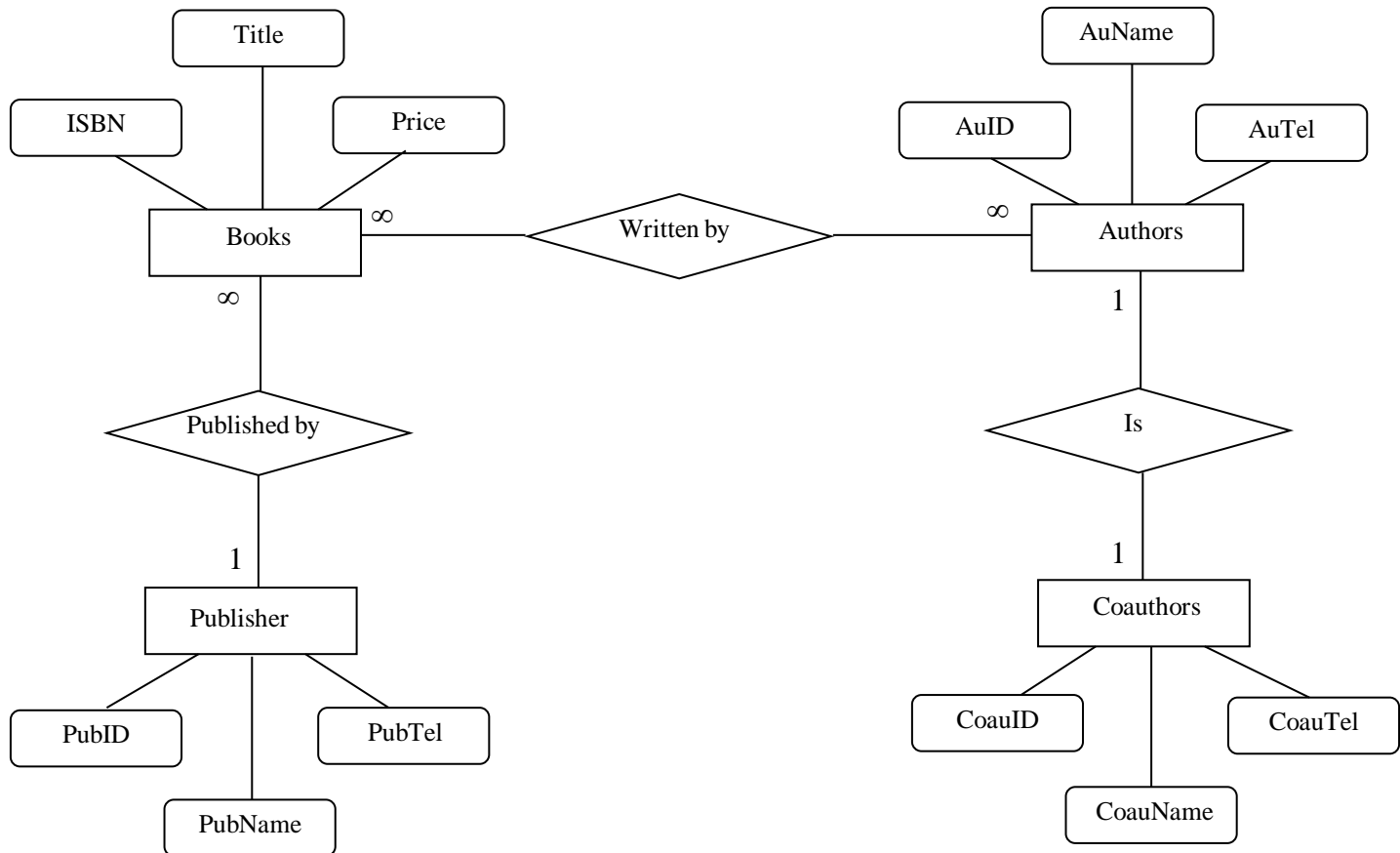
Relational Database Management System (RDBMS)

- E.g. Microsoft Access

# Relational Database

- System of related tables
- Minimum redundancy
- Referential integrity
- Database keys
- The ACID model (guarantee of successful transactions):
  - Atomicity („all or nothing” rule)
  - Consistency (only valid data in)
  - Isolation (order of executed transactions)
  - Durability (committed transaction will not be lost)

# Relations in a database example



# Home library – table Books

ISBN	Title	PubID	Price
0-103-45678-9	Iliad	1	25.00 zł
0-11-345678-9	Moby Dick	3	49.00 zł
0-12-333433-3	On Liberty	1	25.00 zł
0-123-45678-0	Ulysses	2	34.00 zł
0-12-345678-9	Jane Eyre	3	49.00 zł
0-321-32132-1	Balloon	3	34.00 zł
0-55-123456-9	Main Street	3	23.00 zł
0-555-55555-9	Macbeth	2	12.00 zł
0-91-045678-5	Hamlet	2	20.00 zł
0-91-335678-7	Faerie Queen	1	15.00 zł
0-99-777777-7	King Lear	2	49.00 zł
0-99-999999-9	Emma	1	20.00 zł
1-1111-1111-1	C++	1	30.00 zł
1-22-233700-0	Visual Basic	1	25.00 zł



# Home library – table Authors

AuID	AuName	AuTel
1	Austen	111-111-1111
2	Melville	222-222-2222
3	Homer	333-333-3333
4	Roman	444-444-4444
5	Shakespeare	555-555-5555
6	Joyce	666-666-6666
7	Spenser	777-777-7777
8	Mill	888-888-8888
9	Smith	123-222-2222
10	Jones	123-333-3333
11	Snoopy	321-321-2222
12	Grumpy	321-321-0000
13	Sleepy	321-321-1111

# Home library – table Publishers

PubID	PubName	PubTel
1	Big House	123-456-7890
2	Alpha Press	999-999-9999
3	Small House	714-000-0000

# Home library – table Books/Authors

ISBN	AuID
0-103-45678-9	3
0-11-345678-9	2
0-12-333433-3	8
0-123-45678-0	6
0-12-345678-9	1
0-321-32132-1	11
0-321-32132-1	12
0-321-32132-1	13
0-55-123456-9	9
0-55-123456-9	10
0-555-55555-9	5
0-91-045678-5	5
0-91-335678-7	7
0-99-777777-7	5
0-99-999999-9	1
1-1111-1111-1	4
1-22-233700-0	4

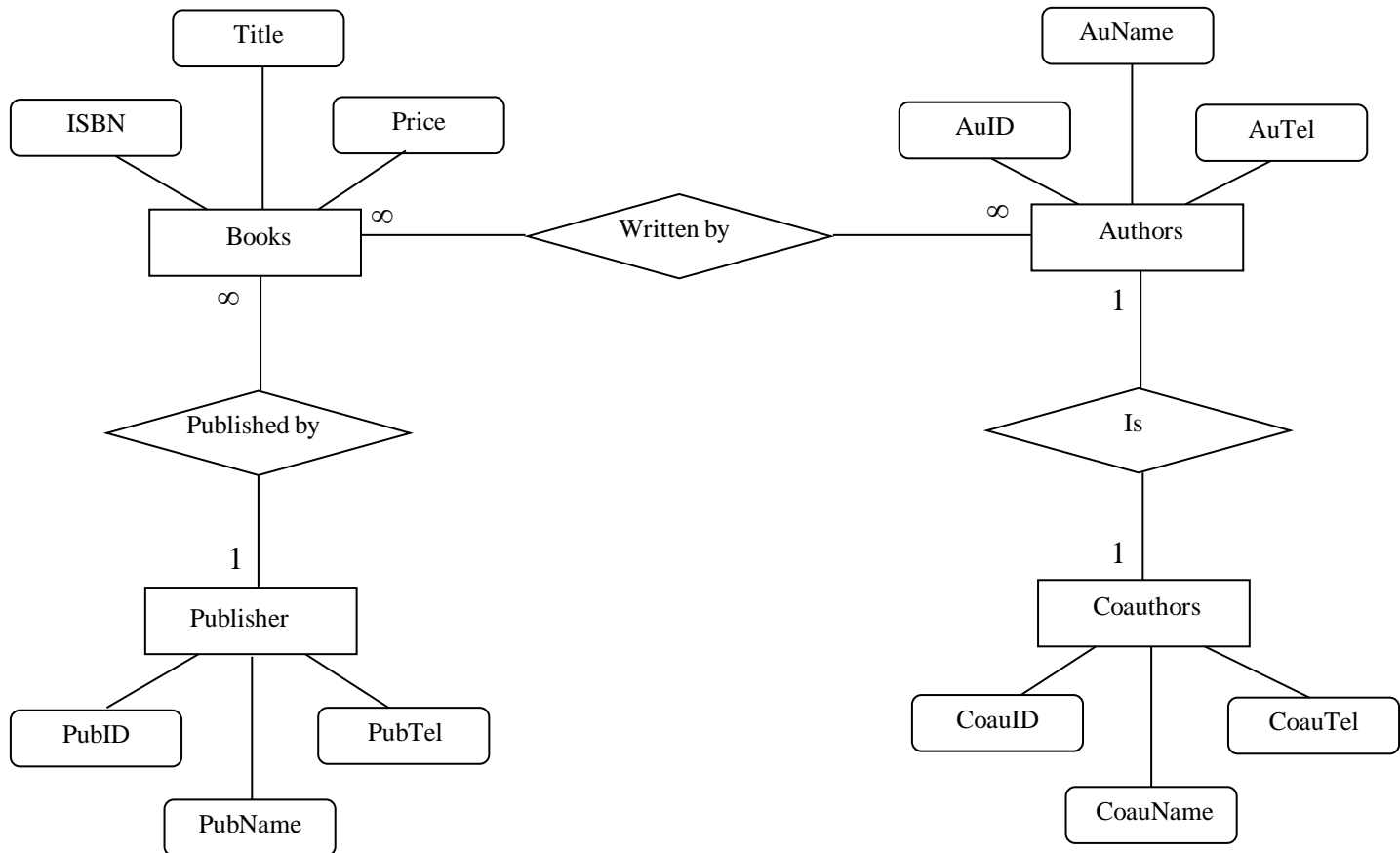
# Table

- Unique name
- Size = # of rows, order =# of columns
- Structure of a table  $\rightarrow T\{ A_1, A_2, \dots, A_n \}$
- All rows different
- Order of rows not important
- Unique headers identify columns
- NULL value in tables

# Database keys

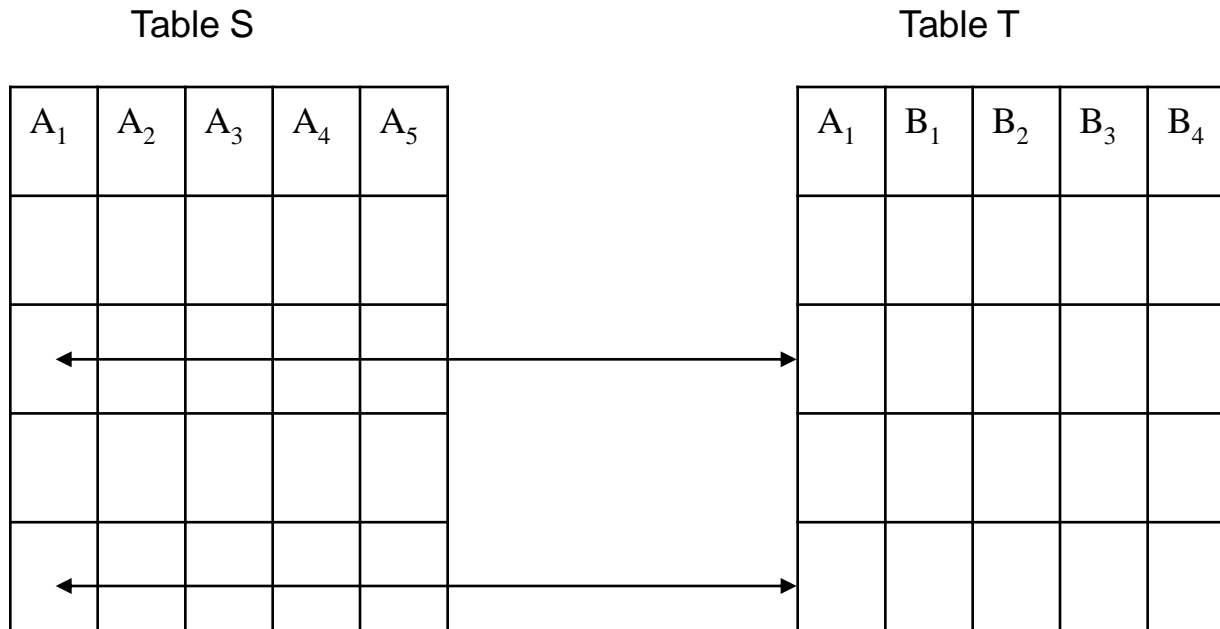
- **Primary key**
  - Value unique for each record in a table
  - This value can not be used twice
  - AutoNumber guarantees uniqueness but does not carry any useful information
- **Foreign keys**
  - Used to create relationships between tables
  - No uniqueness constraint for foreign keys
- **Relation between primary and foreign keys**
  - Same format
  - Same values

# Relations in a database example



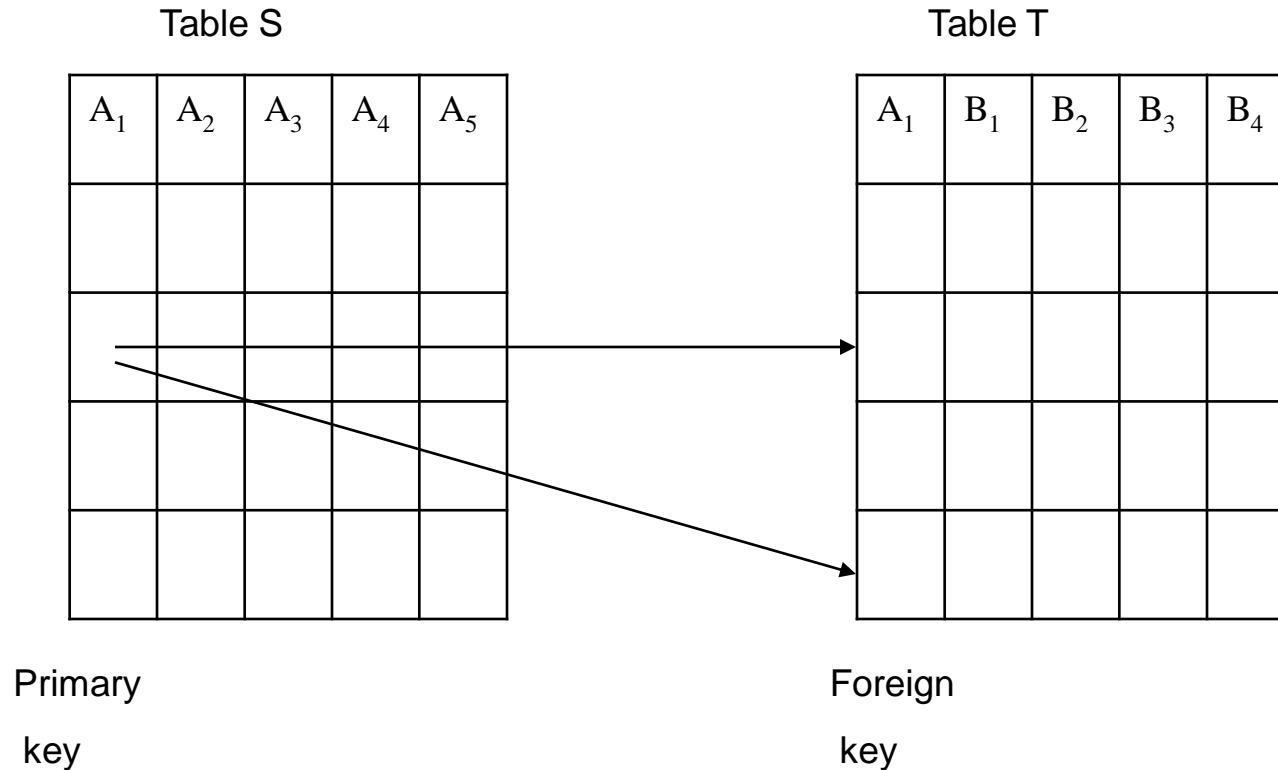
# Building relations

## Relation one-to-one



# Building relations

## Relation one-to-many

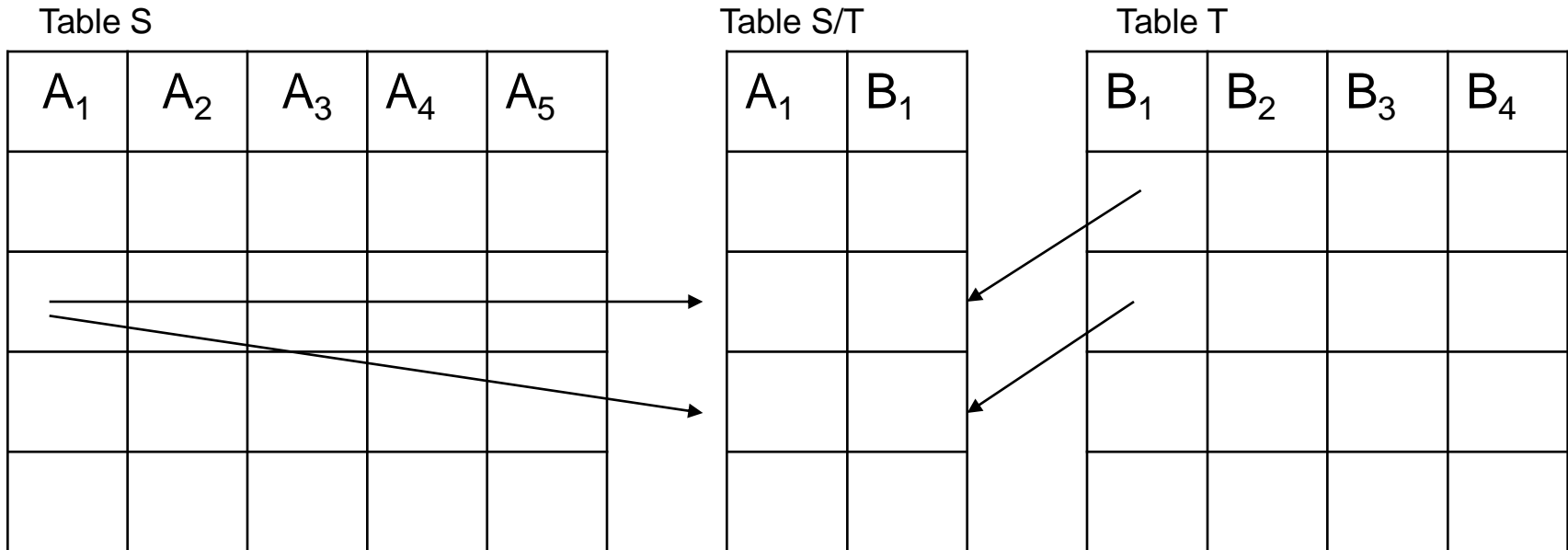


Values of the foreign key can not be different from the values of the primary key.



# Building relations

Relation many-to-many

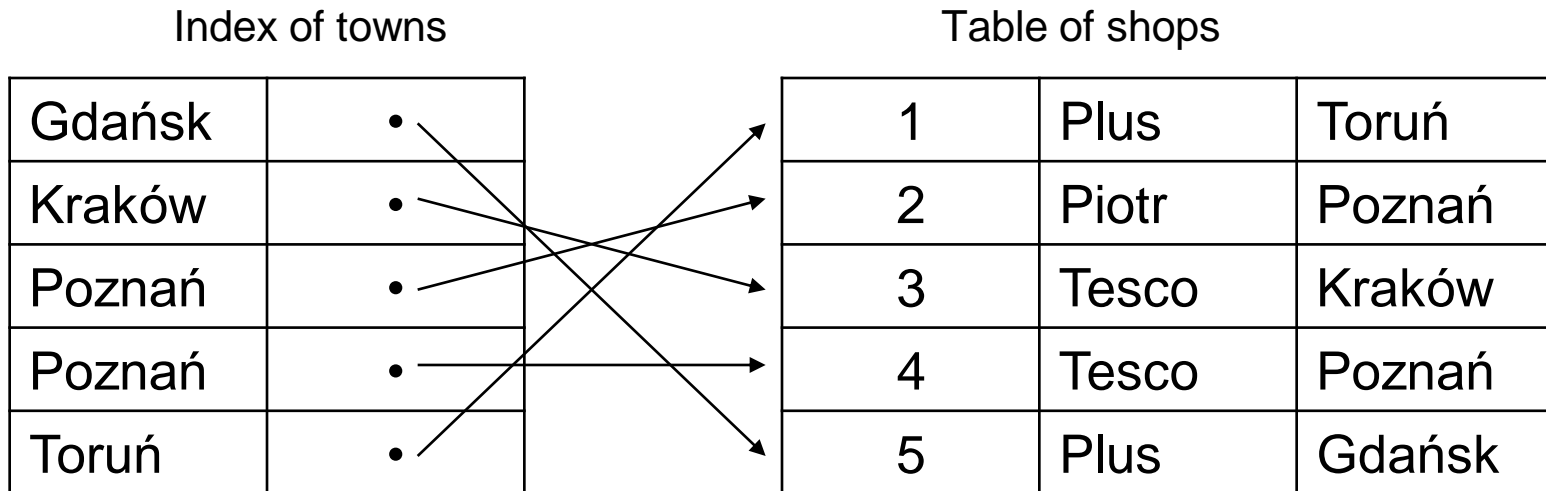


# Enforcing referential integrity

- Cascade Update Related Fields – the values of foreign keys change following changes of the values of the primary key
- Cascade Delete Related Records – deleting a record from the primary field in a relationship causes a deletion of all related records in the second table

# Indexing field values

- Purpose: speed up access to specific data
- Used in large tables
- Updating of all indexes every time a table record is updated or added
- Example



# Principles for building a database

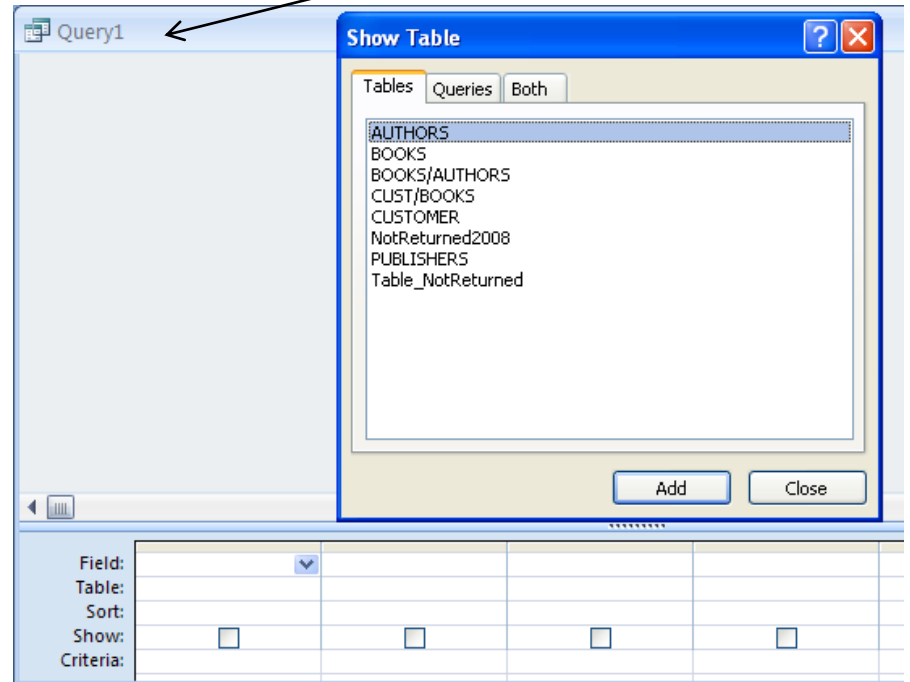
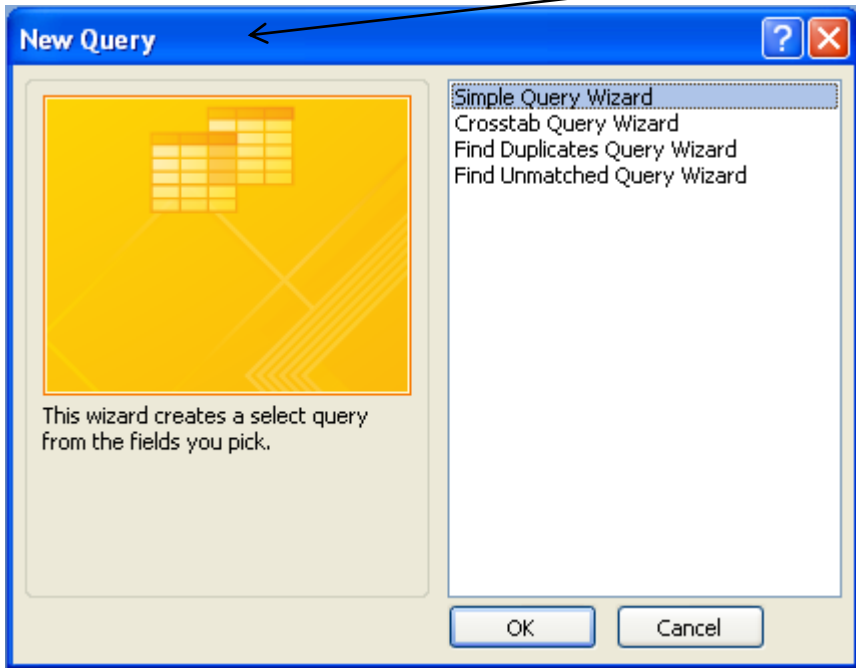
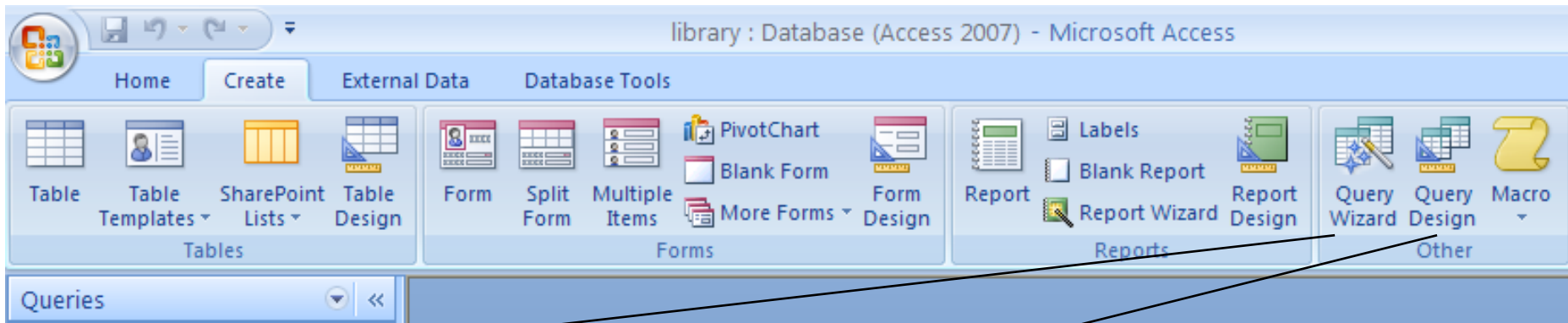
- Types of attributes:
    - Identification
    - Information
    - Identification+information
  - Example 1: {PubID, PubName, PubTel, FoundYear}
- Ident                      Ident+inform                      Inform
- Example 2:

ISBN	Title	PubID	PubName
1-1111-1111-1	Macbeth	1	Big House
2-2222-2222-2	Hamlet	1	
5-5555-5555-5		2	ABC Press

# Queries

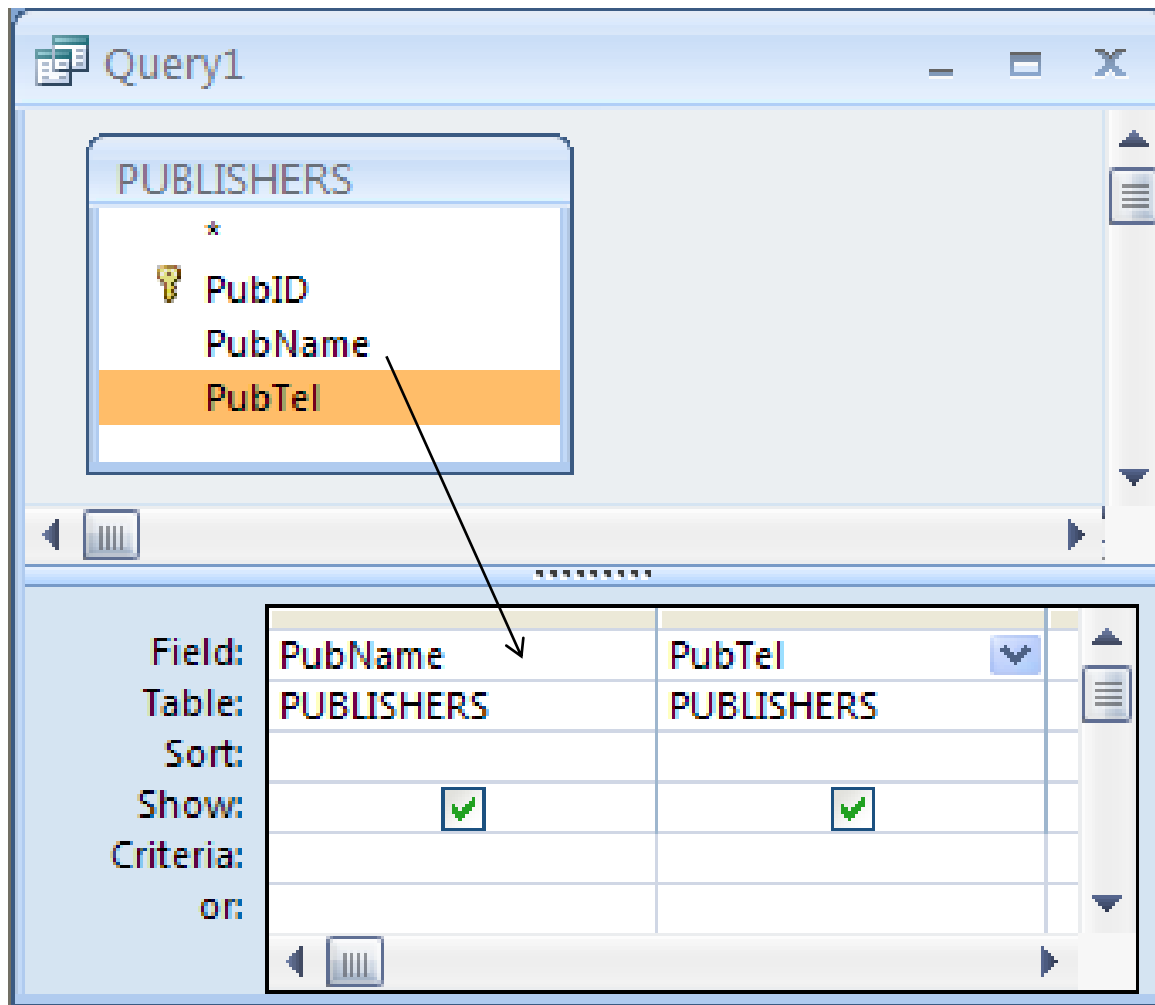
- Database – data located in tables + relations
- Query – primary mechanism for retrieving information from a database, consists of questions presented to the database in a predefined format – an expression stored in a database having a unique name
- Answer to the query – a computed table = Dynaset
- SQL – Structured Query Language
- Types of queries:
  - Select query
  - Action queries (Make-Table, Append, Update, Delete)
  - Crosstab query

# Creating a query in MS Access



# Design View of a query

Drug and drop principle



# Selection criteria

- Specifying criteria:
  - A value of an expression
  - Use of criteria operators: <, >, <=, >=, <>
  - BETWEEN, e.g. BETWEEN 2 AND 5,
  - LIKE, e.g. LIKE „\*[b-d]k[0-5]?#”
- Logical operators: OR, AND
  - e.g. „Smith” OR „Jones”
- Mathematical operators: +, -, \*, /, \, MOD, ^
- Text operator: &
- Date/Time fields
  - Format #2009-06-19# #16:00# #4:00PM#
  - Date/time functions: Date()  
Day(date), Month(date), Year(date), Weekday(date)



# Calculation on groups of records

Function	Computes	AutoNumber	Currency	Date/Time	Number	Yes/No	Text	ALL
Sum	The total of the values in the field	X	X	X	X	X		
Avg	The average of the values in the field	X	X	X	X	X		
Min	The lowest value in the field	X	X	X	X	X	X	
Max	The highest value in the field	X	X	X	X	X	X	
Count	The number of non-blank values in the field							X
StDev	The standard deviation of the values in the field	X	X	X	X	X		
Var	The variance of the values in the field	X	X	X	X	X		
First	The value from the first record							X
Last	The value from the last record							X

# Parameter query

The screenshot shows the Microsoft Access interface for a parameter query named "Titles". The query is based on two tables: BOOKS and PUBLISHERS. The BOOKS table has fields: ISBN (primary key), Title, PubID, Price, and Year. The PUBLISHERS table has fields: PubID (primary key), PubName, and PubTel. The relationship between the tables is 1:8. The query criteria are as follows:

Field:	Title	PubName	Price	Year
Table:	BOOKS	PUBLISHERS	BOOKS	BOOKS
Sort:	Ascending			
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:			<=[The maximum price]	Between 1980 And 1990
or:				

When you run the query, you will be prompted to supply the maximum price

# Crosstab query

**Crosstab Query Wizard**

Which table or query contains the fields you want for the crosstab query results?

To include fields from more than one table, create a query containing all the fields you need and then use this query to make the crosstab query.

Query: Author  
 Query: Averages  
 Query: BooksList  
 Query: Borrowed  
 Query: Customers1  
 Query: Customers2  
 Query: NotBorrowed  
 Query: NumberBooks

View  
 Tables  Queries  Both

Sample:

	Header1	Header2	Header3
TOTAL			

Cancel < Back **Next >** Finish

**Crosstab Query Wizard**

Which fields' values do you want as row headings?

You can select up to three fields.

Select fields in the order you want information sorted. For example, you could sort and group values by Country and then Region.

Available Fields:  
 Osoba  
 DateOut

Selected Fields:  
 Title

Sample:

Title	Header1	Header2	Header3
Title1	TOTAL		
Title2			
Title3			
Title4			

Cancel < Back **Next >** Finish

**Crosstab Query Wizard**

What number do you want calculated for each column and row intersection?

For example, you could calculate the sum of the field Order Amount for each employee (column) by country and region (row).

Do you want to summarize each row?  
 Yes, include row sums.

Fields:  
 DateOut

Functions:  
 Count  
 First  
 Last  
 Max  
 Min

Sample:

Title	Osoba1	Osoba2	Osoba3
Title1	Count(DateOut)		
Title2			
Title3			
Title4			

Cancel < Back **Next >** Finish

**Borrowed\_Crosstab**

Title	Total Of DateC	Anne Brown	John Smith
Ulysses	1	1	
Visual Basic	1		1

Record: 1 of 2 No Filter Search

Current Record

# Database Normalization Basics

- Purpose:
    - eliminating redundant data
    - ensuring logical relations of dependent data
  - The normal forms
    - 1NF, 2NF, 3NF, BCNF
    - guidelines only
    - hierarchical structure of NF
  - First Normal Form (1NF)
    - eliminate duplicative columns
    - create separate tables for each group of related data and define primary keys
- e.g. Authors = Jones, H.; Smith K. (incorrect)

# Database Normalization Basics

- Second Normal Form (2NF)
  - meet all requirements of the 1NF
  - remove subsets of data that apply to multiple rows and place them in separate tables
  - create relationships between new tables using foreign keys
- Example - table of addresses:  
{Town, Street, HouseNumber, HouseColor, SizeOfTown}  
attribute of Town

# Database Normalization Basics

- Third Normal Form (3NF)
  - meet all requirements of the 2NF
  - remove columns that are not dependent upon primary key

- Example

{ISBN, Title, NumberOfBooks, UnitPrice, TotalValue}

where:  $TotalValue = NumberOfBooks * UnitPrice$

Correct form:

{ISBN, Title, NumberOfBooks, UnitPrice}

# Database Normalization Basics

- Fourth Normal Form (Boyce-Codd NF = BCNF)
  - meet all requirements of the 3NF
  - remove all multi-valued dependencies

- Example

{Town, Street, HouseNumber, ZIPcode}

where: combination of {Town, Street} determines  
{Zipcode}

Correct form:

{Street, HouseNumber, ZIPcode}

and {ZIPcode, Town}

# Decomposition of tables

- Relations between data must be conserved

AuID	AuName	PubID
A1	Smith, John	P1
A2	Smith, John	P2

- Decomposition

AuID	AuName
A1	Smith, John
A2	Smith, John

AuName	PubID
Smith, John	P1
Smith, John	P2

- Display all John Smiths

AuID	AuName	PubID
A1	Smith, John	P1
A1	Smith, John	P2
A2	Smith, John	P1
A2	Smith, John	P2



## Example: Relation ORDERS not normalized

No_order	IDsupplier	Name_Supplier	Address_Supplier	IDpart	Name_part	Qty	Warehouse	Address_Warehouse
001	300	VW	Wolfsburg, Rotestrasse 10	53	Carburetor	100	5	Warszawa, Chopina 3
				57	Crankshaft	50	5	Warszawa, Chopina 3
				59	Mudguard	500	6	Warszawa, Mozarta 25
002	400	WSK	Świdnik, Kraszewskiego 5	54	Carburetor	500	5	Warszawa, Chopina 3
				32	Wheel	100	6	Warszawa, Mozarta 25
003	500	VW Polska	Antoninek, Słowackiego 2	88	Engine	15	7	Warszawa, Bacha 3
004	600	FIAT	Bielsko-Biała, Mickiewicza 25	58	Mudguard	400	6	Warszawa, Mozarta 25
				21	Alternator	50	7	Warszawa, Bacha 3
005	300	VW	Wolfsburg, Rotestrasse 10	53	Carburetor	200	5	Warszawa, Chopina 3
				57	Crankshaft	30	5	Warszawa, Chopina 3
006	300	VW	Wolfsburg, Rotestrasse 10	59	Mudguard	20	6	Warszawa, Mozarta 25

## Relation ORDERS in the first normal form (1NF) eliminates duplicative columns

No_order	IDSupplier	Name_Supplier	Address_Supplier	IDpart	Name_part	Qty	Warehouse	Address_Warehouse
001	300	VW	Wolfsburg, Rotestrasse 10	53	Carburetor	100	5	Warszawa, Chopina 3
001	300	VW	Wolfsburg, Rotestrasse 10	57	Crankshaft	50	5	Warszawa, Chopina 3
001	300	VW	Wolfsburg, Rotestrasse 10	59	Mudguard	500	6	Warszawa, Mozarta 25
002	400	WSK	Świdnik, Kraszewskiego 5	54	Carburetor	500	5	Warszawa, Chopina 3
002	400	WSK	Świdnik, Kraszewskiego 5	32	Wheel	100	6	Warszawa, Mozarta 25
003	500	VW	Antoninek, Słowackiego 2	88	Engine	15	7	Warszawa, Bacha 3
004	600	FIAT	Bielsko-Biała, Mickiewicza 25	58	Mudguard	400	6	Warszawa, Mozarta 25
004	600	FIAT	Bielsko-Biała, Mickiewicza 25	21	Alternator	50	7	Warszawa, Bacha 3
005	300	VW	Wolfsburg, Rotestrasse 10	53	Carburetor	200	5	Warszawa, Chopina 3
005	300	VW	Wolfsburg, Rotestrasse 10	57	Crankshaft	30	5	Warszawa, Chopina 3
006	300	VW	Wolfsburg, Rotestrasse 10	59	Mudguard	20	6	Warszawa, Mozarta 25

# Relation ORDERS in the second normal form (2NF)

all attributes fully dependent on primary keys

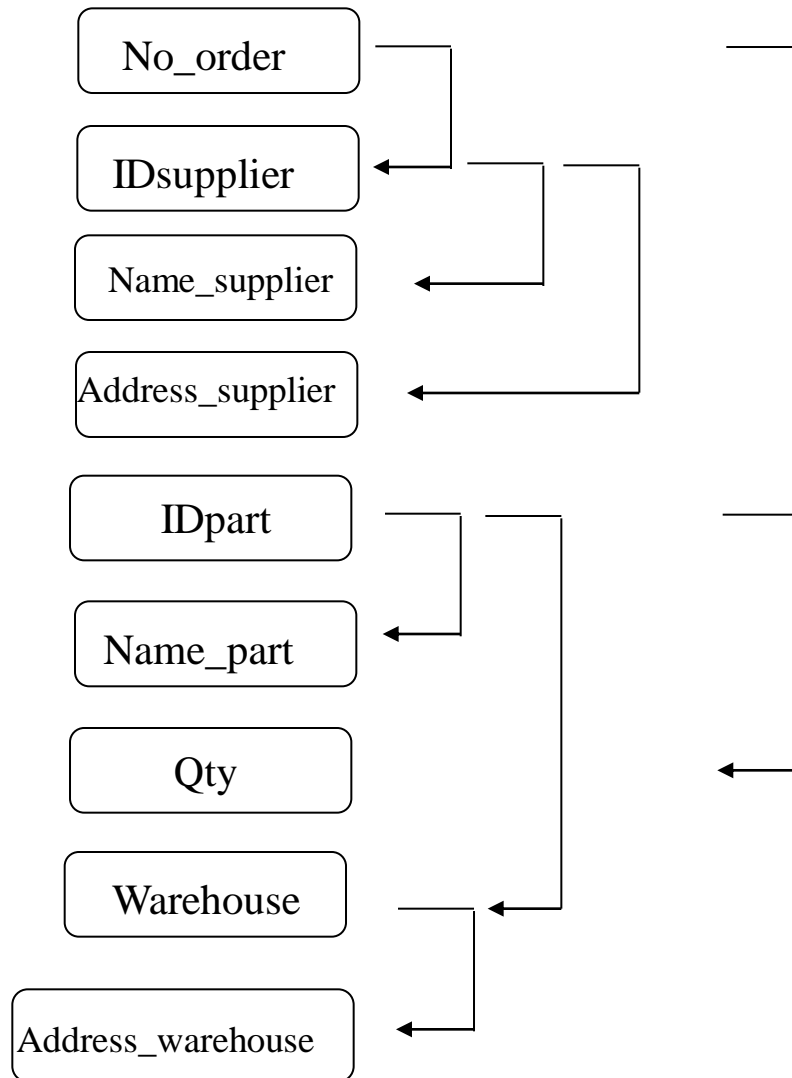


Diagram of functional dependencies in the relation ORDERS

No_order	IDsupplier	Name_supplier	Address_Supplier
001	300	VW	Wolfsburg, Rotestrasse 10
002	400	WSK	Świdnik, Kraszewskiego 5
003	500	VW	Antoninek, Słowackiego 2
004	600	FIAT	Bielsko-Biała, Mickiewicza 25
005	300	VW	Wolfsburg, Rotestrasse 10
006	300	VW	Wolfsburg, Rotestrasse 10

SUPPLIER\_ON\_ORDER

**2NF**

PARTS\_ON\_ORDER

No_order	IDpart	Qty
001	53	100
001	57	50
001	59	500
002	54	500
002	32	100
003	88	15
004	58	400
004	21	50
005	53	200
005	57	30
006	59	20

IDpart	Name_part	Warehouse	Address_Warehouse
53	Carburetor	5	Warszawa, Chopina 3
57	Crankshaft	5	Warszawa, Chopina 3
58	Mudguard	6	Warszawa, Mozarta 25
59	Mudguard	6	Warszawa, Mozarta 25
54	Carburetor	5	Warszawa, Chopina 3
32	Wheel	6	Warszawa, Mozarta 25
88	Engine	7	Warszawa, Bacha 3
21	Alternator	7	Warszawa, Bacha 3

PARTS\_IN\_WAREHOUSE

# Relation ORDERS in the third normal form (3NF)

remove column not dependent upon primary key

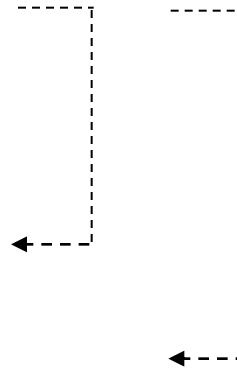
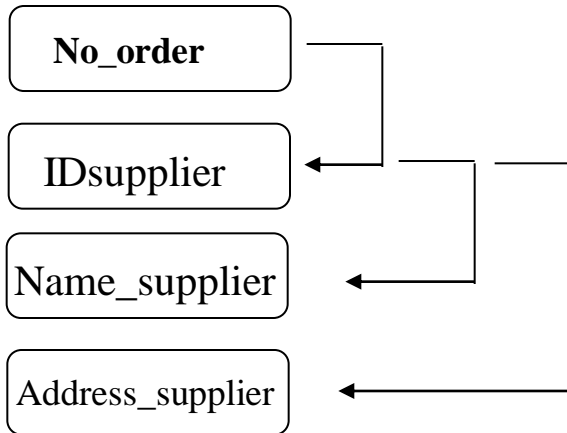


Diagram of functional dependencies in the relation SUPPLIER\_ON\_ORDER

Diagram of functional dependencies in the relation PARTS\_IN\_WAREHOUSE

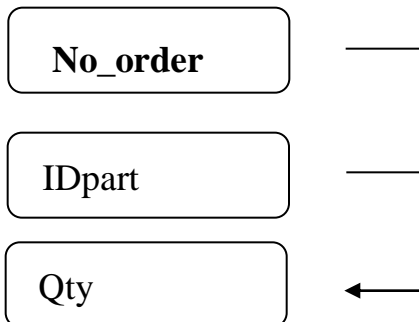
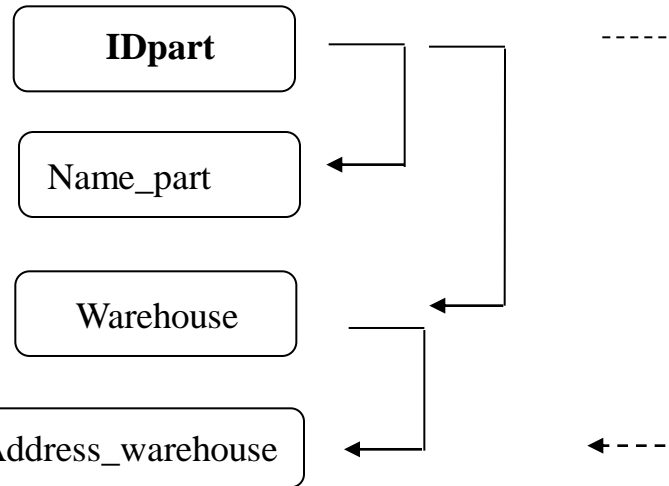
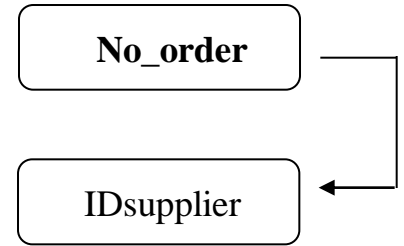


Diagram of functional dependencies in the relation PARTS\_ON\_ORDER

# 3NF

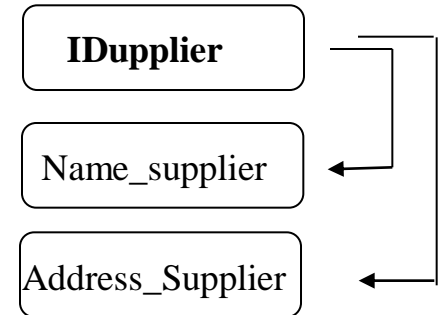
## ORDER\_TO\_SUPPLIER

No_order	IDsupplier
001	300
002	400
003	500
004	600
005	300
006	300



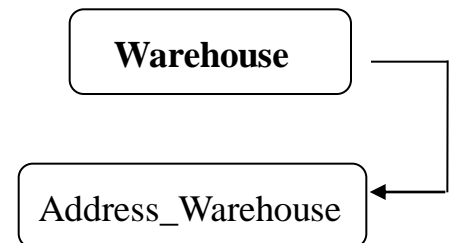
## SUPPLIERS

IDsupplier	Name_Supplier	Address_Supplier
300	VW	Wolfsburg, Rotestrasse 10
400	WSK	Świdnik, Kraszewskiego 5
500	VW	Antoninek, Słowackiego 2
600	FIAT	Bielsko-Biała, Mickiewicza 25



## WAREHOUSES

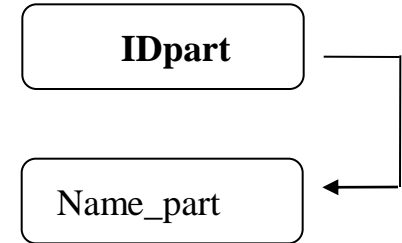
Warehouse	Address_Warehouse
5	Warszawa, Chopina 3
6	Warszawa, Mozarta 25
7	Warszawa, Bacha 3



# 3NF

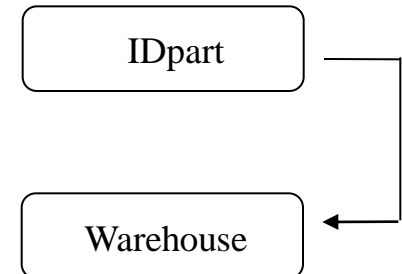
PARTS

IDpart	Name_part
21	Alternator
32	Wheel
53	Carburetor
54	Carburetor
57	Crankshaft
58	Mudguard
59	Mudguard
88	Engine



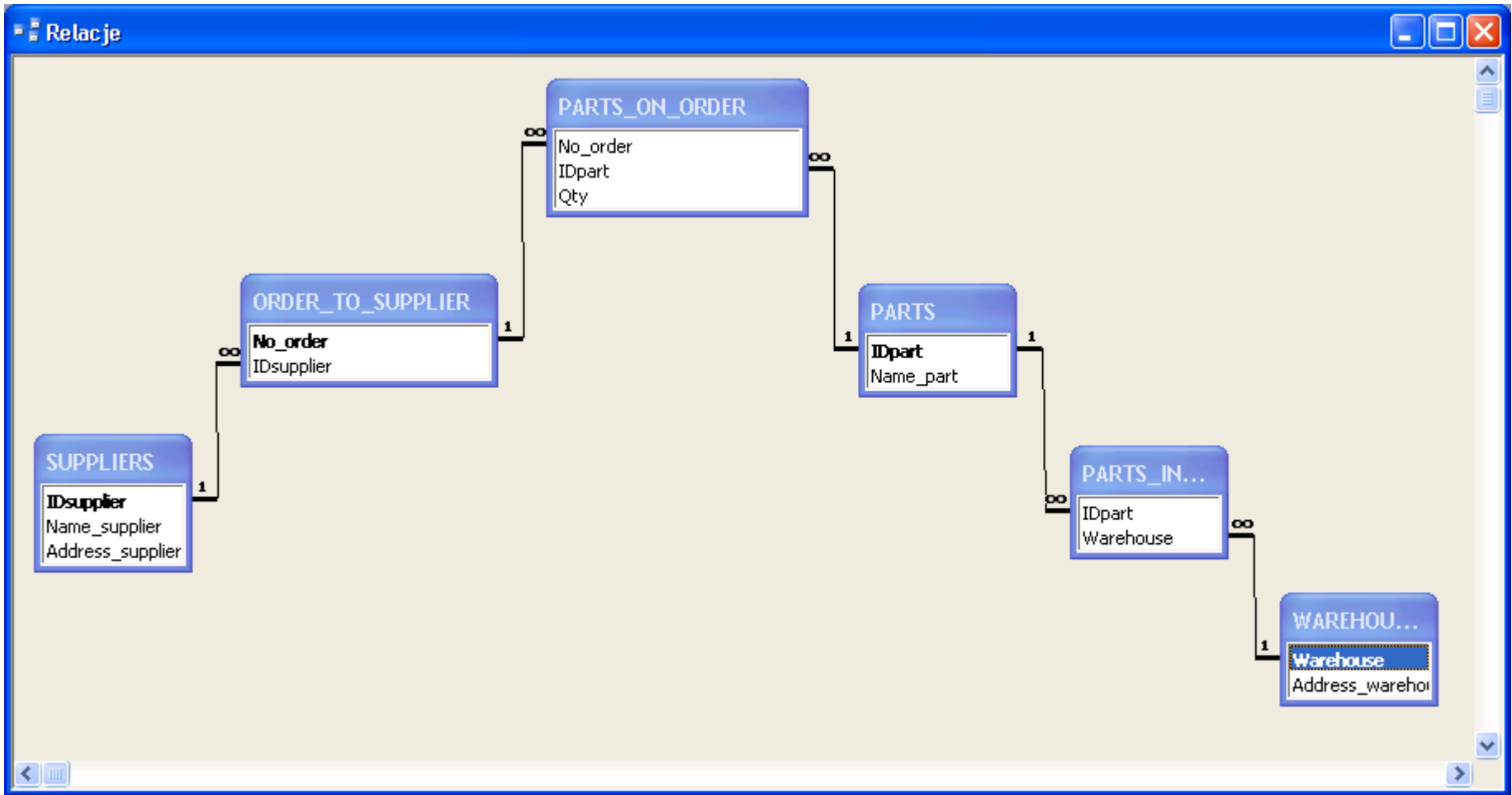
PARTS\_IN\_WAREHOUSE

IDpart	Warehouse
53	5
57	5
58	6
59	6
54	5
32	6
88	7
21	7



PARTS\_ON\_ORDER as above

# Example: ORDERS





# Structured Query Language - SQL

Why to use SQL in addition to the Design View?

- not all SQL functions can be used from the Design View level
- SQL can be used in other applications (Excel, Word, Visual Basic)
- SQL is a standard query language which can be used outside the Access program

One can easily switch between Design View and SQL View

SQL is a procedure language which tells what to do, and not how to do.

SQL components:

- Data Definition Language – DDL
- Data Manipulation Language – DML
- Data Control Language - DCL

# DML component of SQL

Basic instructions:

- SELECT
- UPDATE
- DELETE

Basic structure of the SQL command:

SELECT column1, column2,...

FROM table1, table2,...

WHERE criteria;

List of columns can be replaced with \*.

# Sorting

```
SELECT PUBLISHERS.PubName, PUBLISHERS.PubTel  
FROM PUBLISHERS  
ORDER BY PUBLISHERS.PubName;
```

PubName	PubTel
Alpha Press	999-999-9999
Big House	123-456-7890
Small House	714-000-0000

or

```
ORDER BY attribute DESC
```

# Changing field names

```
SELECT PUBLISHERS.PubName AS [Publisher's Name],  
PUBLISHERS.PubTel AS [Publisher's Phone]  
FROM PUBLISHERS  
ORDER BY PUBLISHERS.PubName;
```

Publisher's Name	Publisher's Phone
Alpha Press	999-999-9999
Big House	123-456-7890
Small House	714-000-0000

# Filtering

```
SELECT COUNT (ISBN) AS [How many books from Alpha Press?]  
FROM BOOKS  
WHERE (PubID=1);
```

How many books from Alpha Press?
6

```
SELECT COUNT(ISBN) AS [Number of books],  
MIN(Price) AS Min_Price,  
MAX(Price) AS Max_Price,  
AVG(Price) AS Avg_Price  
FROM BOOKS;
```

Number of books	Min_Price	Max_Price	Avg_Price
15	12.00 zł	49.00 zł	29.27 zł

# Grouping

```
SELECT PubID, COUNT(*) AS [Number of books]
FROM BOOKS
GROUP BY PubID;
```

PubID	Number of books
1	6
2	4
3	4
4	1

# Group filtering

```
SELECT PubID, COUNT(*) AS [Number of books]
FROM BOOKS
GROUP BY PubID
HAVING COUNT(*)>=2 ;
```

PubID	Number of books
1	6
2	4
3	4

# Aggregation, filtering, grouping, group filtering

```
SELECT PubID, COUNT(*) AS [Number of books]
FROM BOOKS
WHERE Price >= 25
GROUP BY PubID
HAVING COUNT(*)>=2 ;
```

PubID	Number of books
1	4
2	2
3	3



# Elements of the SELECT command

Element	Description	Compulsory?
SELECT	Returns columns or expressions	Yes
FROM	Taking data from tables	Yes if data from tables
WHERE	Filtering rows	No
GROUP BY	Creating groups	Yes for aggregation functions
HAVING	Filtering groups	No
ORDER BY	Sorting of the output	No

# DELETE and UPDATE

```
DELETE BOOKS1.Price  
FROM BOOKS1  
WHERE ((BOOKS1.Price)>40);
```

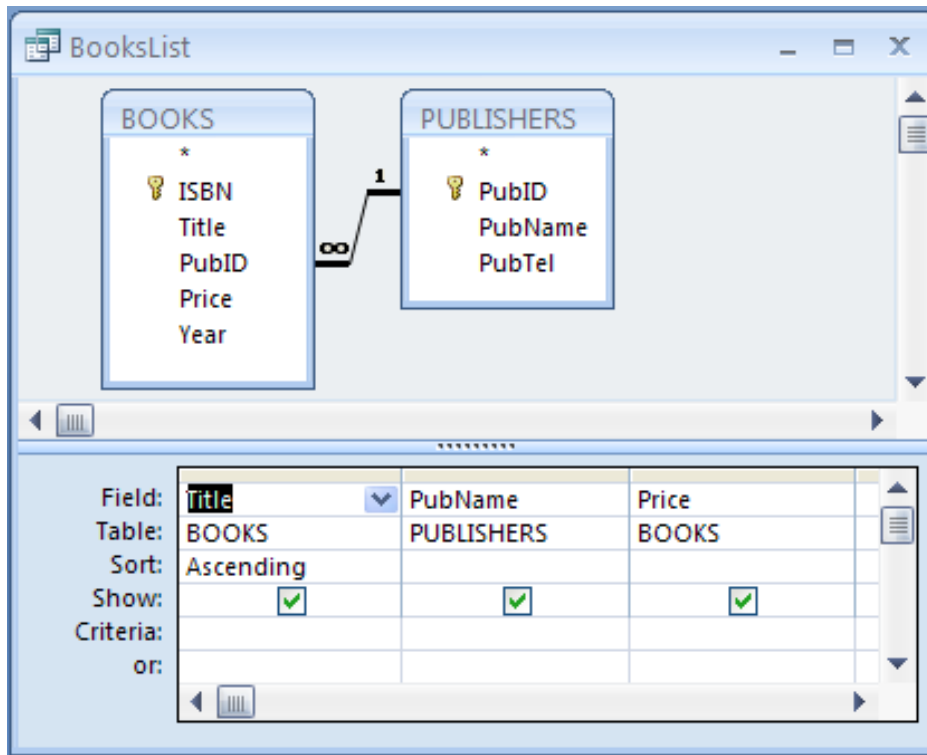
```
UPDATE BOOKS1  
SET BOOKS1.Price= [BOOKS1].[Price]+1;
```

```
UPDATE BOOKS1  
SET BOOKS1.Price= [BOOKS1].[Price]+1  
WHERE ((BOOKS1.Price)<20);
```

# Queries based on multiple tables

```
SELECT Title, PubName, Price
FROM PUBLISHERS, BOOKS
WHERE PUBLISHERS.PubID = BOOKS.PubID
ORDER BY BOOKS.Title;
```

Design View



Dynaset – a computed table

Title	PubName	Price
Balloon	Small House	34.00 zł
C++	Big House	30.00 zł
Emma	Big House	20.00 zł
Faerie Queene	Big House	15.00 zł
Hamlet	Alpha Press	20.00 zł
Iliad	Big House	25.00 zł
Jane Eyre	Small House	49.00 zł
King Lear	Alpha Press	49.00 zł
Macbeth	Alpha Press	12.00 zł
Main Street	Small House	23.00 zł
Moby Dick	Small House	49.00 zł
On Liberty	Big House	25.00 zł
Ulysses	Alpha Press	34.00 zł

# Inner join 1

```
SELECT BOOKS.Title, PUBLISHERS.PubName, BOOKS.Price  
FROM PUBLISHERS, BOOKS  
WHERE PUBLISHERS.PubID = BOOKS.PubID  
ORDER BY BOOKS.Title;
```

```
SELECT BOOKS.Title, PUBLISHERS.PubName, BOOKS.Price  
FROM PUBLISHERS INNER JOIN BOOKS  
ON PUBLISHERS.PubID = BOOKS.PubID  
ORDER BY BOOKS.Title;
```

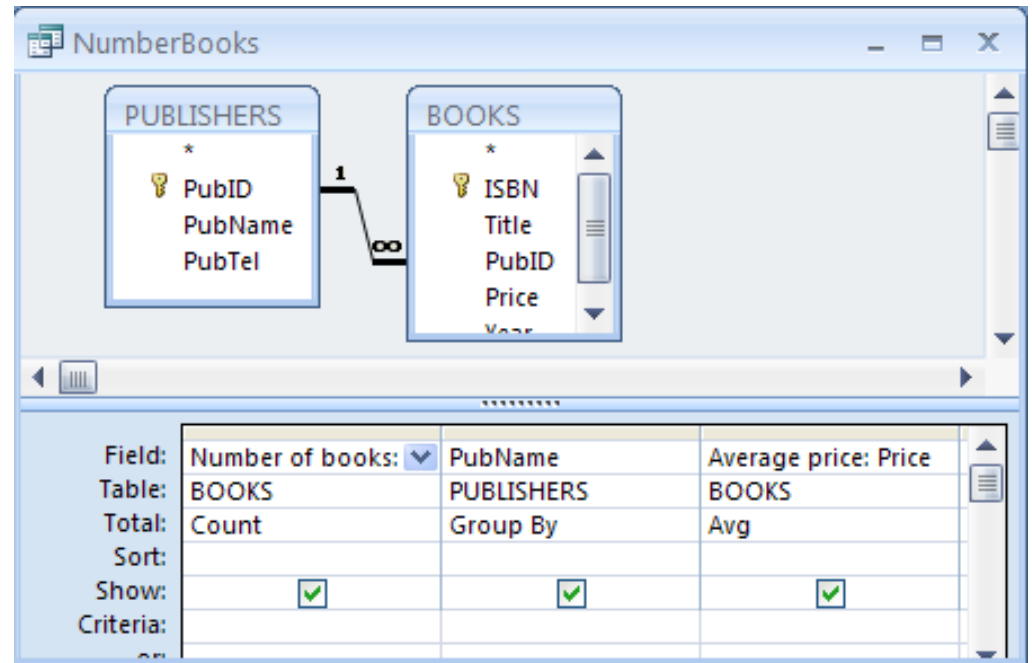
Structure of the inner join one-to-many:

```
TABLE1 INNER JOIN TABLE2
```

```
ON TABLE1.primarykey = TABLE2.foreignkey
```

# Inner join 2 *one-to-many*

```
SELECT Count(BOOKS.Title) AS [Number of books],  
PUBLISHERS.PubName,  
AVG(BOOKS.Price) AS [Average price]  
FROM PUBLISHERS INNER JOIN BOOKS  
ON PUBLISHERS.PubID = BOOKS.PubID  
GROUP BY PUBLISHERS.Name  
ORDER BY PUBLISHERS.Name;
```



Number of books	PubName	Average price
4	Alpha Press	28.75
6	Big House	23.33
4	Small House	38.75

# What happens if WHERE is missing?

```
SELECT BOOKS.Title, PUBLISHERS.PubName, BOOKS.Price  
FROM PUBLISHERS, BOOKS;
```

Title	PubName	Price
Iliad	Big House	25.00 zł
Iliad	Alpha Press	25.00 zł
Iliad	Small House	25.00 zł
Iliad	Edition 2000	25.00 zł
Moby Dick	Big House	49.00 zł
Moby Dick	Alpha Press	49.00 zł
Moby Dick	Small House	49.00 zł
Moby Dick	Edition 2000	49.00 zł
On Liberty	Big House	25.00 zł
On Liberty	Alpha Press	25.00 zł
On Liberty	Small House	25.00 zł
On Liberty	Edition 2000	25.00 zł
Ulysses	Big House	34.00 zł
Ulysses	Alpha Press	34.00 zł
Ulysses	Small House	34.00 zł
Ulysses	Edition 2000	34.00 zł
....		

**Result:**  
A direct product  
of two tables

# Inner join 3 *many-to-many*

```
SELECT BOOKS.Title, BOOKS.ISBN, BOOKS.Price, BOOKS.Year,  
PUBLISHERS.PubName, AUTHORS.AuName  
FROM PUBLISHERS  
INNER JOIN (BOOKS  
INNER JOIN (AUTHORS  
INNER JOIN [BOOKS/AUTHORS]  
ON AUTHORS.AuID = [BOOKS/AUTHORS].AuID)  
ON BOOKS.ISBN = [BOOKS/AUTHORS].ISBN)  
ON PUBLISHERS.PubID = BOOKS.PubID;
```

Title	ISBN	Price	Year	PubName	AuName
Iliad	0-103-45678-9	25.00 zł	1989	Big House	Homer
Moby Dick	0-11-345678-9	49.00 zł	1998	Small House	Melville
On Liberty	0-12-333433-3	25.00 zł	1987	Big House	Mill
Ulysses	0-123-45678-0	34.00 zł	1999	Alpha Press	Joyce
Jane Eyre	0-12-345678-9	49.00 zł	1990	Small House	Austen
Balloon	0-321-32132-1	34.00 zł	1995	Small House	Snoopy
Balloon	0-321-32132-1	34.00 zł	1995	Small House	Grumpy
Balloon	0-321-32132-1	34.00 zł	1995	Small House	Sleepy
Main Street	0-55-123456-9	23.00 zł	1996	Small House	Smith
Main Street	0-55-123456-9	23.00 zł	1996	Small House	Jones
Macbeth	0-555-55555-9	12.00 zł	1991	Alpha Press	Shakespeare
Hamlet	0-91-045678-5	20.00 zł	2000	Alpha Press	Shakespeare

# Query Customers1

```
SELECT [First_name] & " " & [Surname] AS Person,  
BOOKS.Title, [CUST/BOOKS].DateOut, [CUST/BOOKS].DateBack  
FROM CUSTOMER  
INNER JOIN (BOOKS  
INNER JOIN [CUST/BOOKS]  
ON BOOKS.ISBN = [CUST/BOOKS].ISBN)  
ON CUSTOMER.CustID = [CUST/BOOKS].CustID;
```

The screenshot displays the Microsoft Access interface for a query named 'Customers1'. The query design view shows three tables: BOOKS, CUST/BOOKS, and CUSTOMER. The query grid shows the fields selected for the query. The query results view shows a table with 4 records.

Field:	Person: [First_name] & " " & [Surname]	Title	DateOut	DateBack
Table:		BOOKS	CUST/BOOKS	CUST/BOOKS
Sort:				
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:				
or:				

Person	Title	DateOut	DateBack
John Smith	C++	7/5/2005	1/25/2007
John Smith	Visual Basic	11/12/2008	
Anne Brown	Ulysses	1/5/2009	
*			



# Query Customers2 – OUTER JOIN

```
SELECT CUSTOMER!First_name & " " & CUSTOMER!Surname AS Person,  
WhoBorrowsID.Title, WhoBorrowsID.DateOut, WhoBorrowsID.DateBack  
FROM CUSTOMER LEFT {outer} JOIN WhoBorrowsID  
ON CUSTOMER.CustID = WhoBorrowsID.CustID;
```

The screenshot shows the Microsoft Access interface. At the top, a window titled 'Customers2' displays a table with the following data:

Person	Title	DateOut	DateBack
John Smith	Visual Basic	11/12/2008	
John Smith	C++	7/5/2005	1/25/2007
Anne Brown	Ulysses	1/5/2009	
Henry Eigth			
Michael Cooper			
Lara Croft			
Mary Jones			
*			

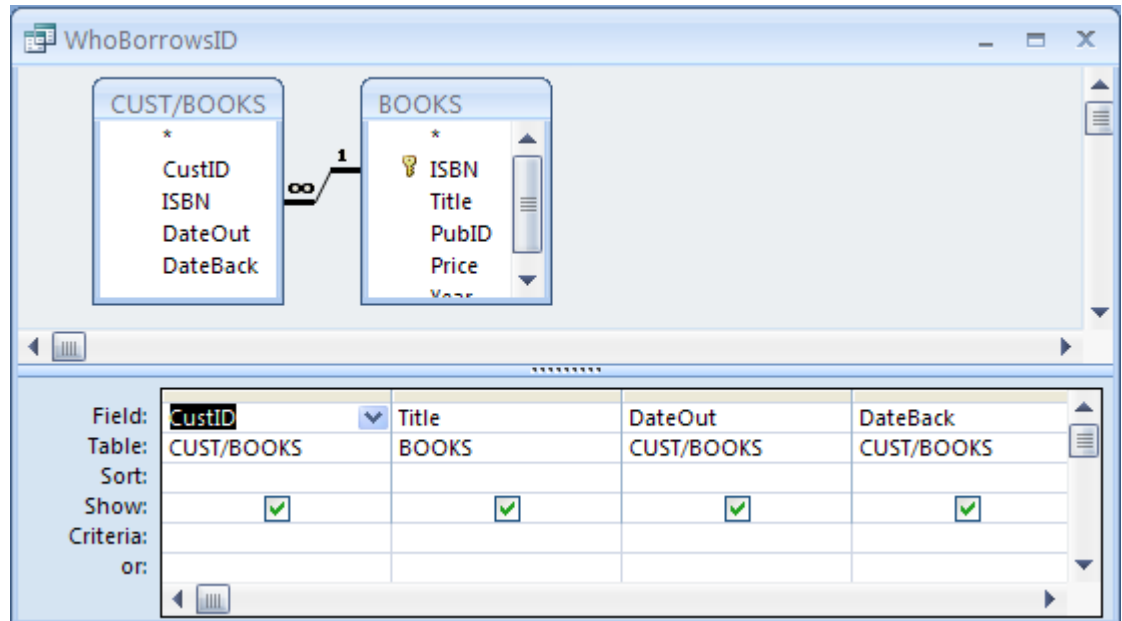
Below the table, the 'Field' list shows the following configuration:

Field:	Title	DateOut	DateBack
Person: CUSTOMER	WhoBorrowsID	WhoBorrowsID	WhoBorrowsID
Table:			
Sort:			
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:			
or:			

**WhoBorrows is a query**

# Query WhoBorrows

```
SELECT [CUST/BOOKS].CustID, BOOKS.Title, [CUST/BOOKS].DateOut,  
[CUST/BOOKS].DateBack  
FROM BOOKS INNER JOIN [CUST/BOOKS]  
ON BOOKS.ISBN = [CUST/BOOKS].ISBN;
```

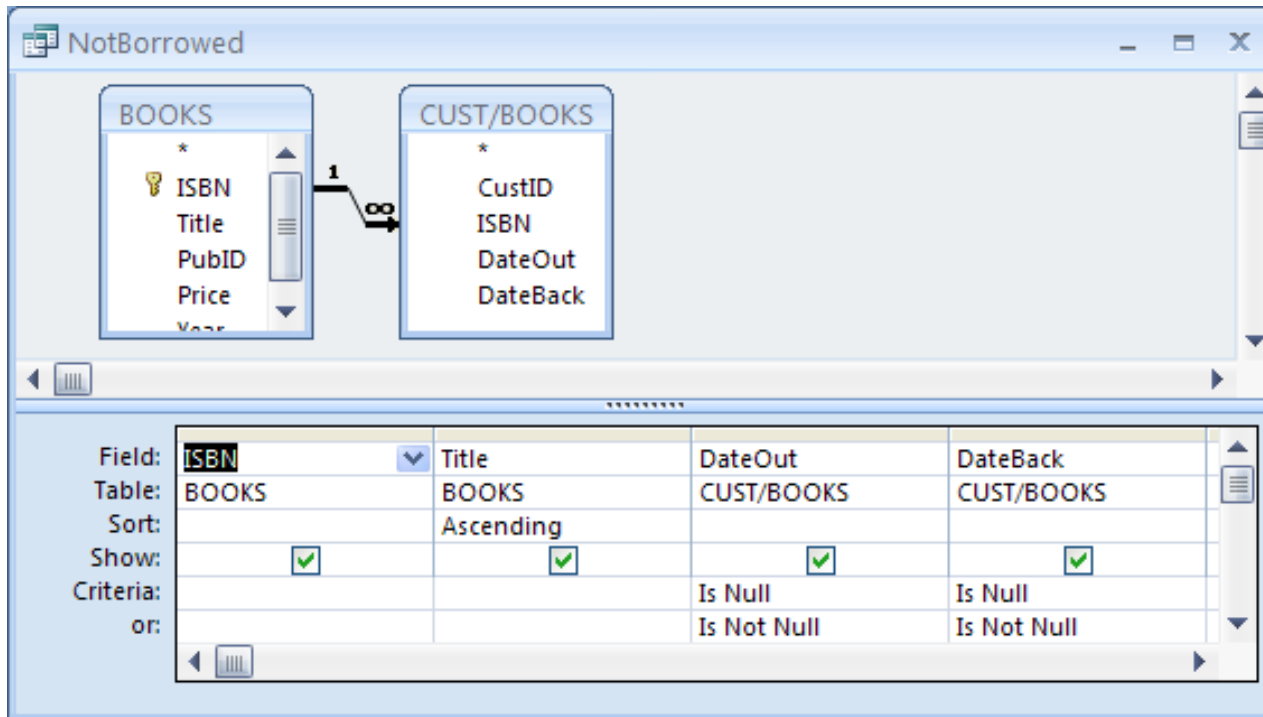


CustID	Title	DateOut	DateBack
1	C++	7/5/2005	1/25/2007
1	Visual Basic	11/12/2008	
2	Ulysses	1/5/2009	

Record: 1 of 3 No Filter Search

# NULL in a query

```
SELECT BOOKS.ISBN, BOOKS.Title, [CUST/BOOKS].DateOut,  
[CUST/BOOKS].DateBack  
FROM BOOKS  
LEFT JOIN [CUST/BOOKS] ON BOOKS.ISBN = [CUST/BOOKS].ISBN  
WHERE  
((( [CUST/BOOKS].DateOut) Is Null) AND (( [CUST/BOOKS].DateBack) Is Null)) OR  
((( [CUST/BOOKS].DateOut) Is Not Null) AND (( [CUST/BOOKS].DateBack) Is Not Null))  
ORDER BY BOOKS.Title;
```

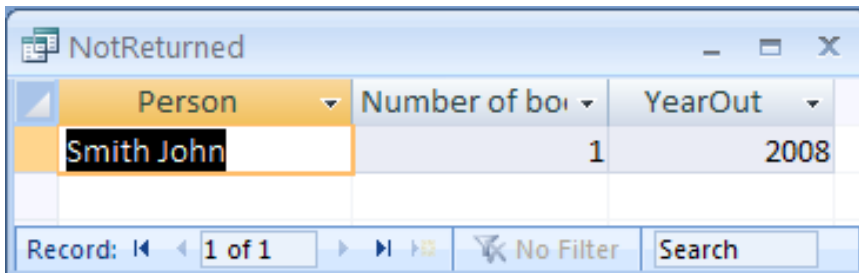


Available books

# Functional queries

```
SELECT CUSTOMER!Surname & " " & CUSTOMER!First_name AS Person,  
       Count([CUST/BOOKS].ISBN) AS [Number of books],  
       Year([CUST/BOOKS]!DateOut) AS YearOut  
INTO Table_NotReturned  
FROM CUSTOMER INNER JOIN [CUST/BOOKS]  
     ON CUSTOMER.CustID=[CUST/BOOKS].CustID  
GROUP BY CUSTOMER!Surname & " " & CUSTOMER!First_name,  
       Year([CUST/BOOKS]!DateOut), [CUST/BOOKS].DateBack  
HAVING ((([CUST/BOOKS].DateBack) Is Null) AND  
        ((Year([CUST/BOOKS]!DateOut))=[Give the year]))  
ORDER BY CUSTOMER!Surname & " " & CUSTOMER!First_name;
```

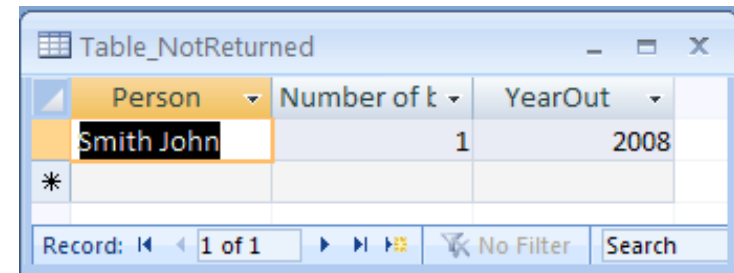
Only SELECT



Person	Number of books	YearOut
Smith John	1	2008

Record: 1 of 1 No Filter Search

Creating a new table



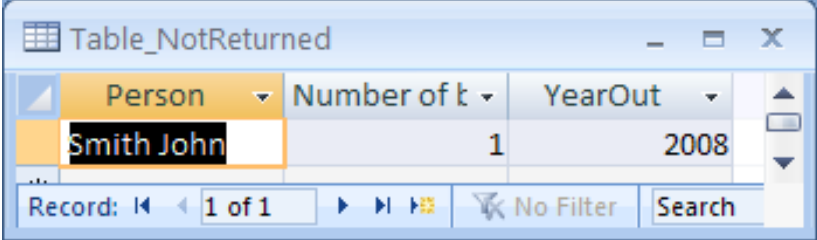
Person	Number of books	YearOut
Smith John	1	2008
*		

Record: 1 of 1 No Filter Search

# Functional queries

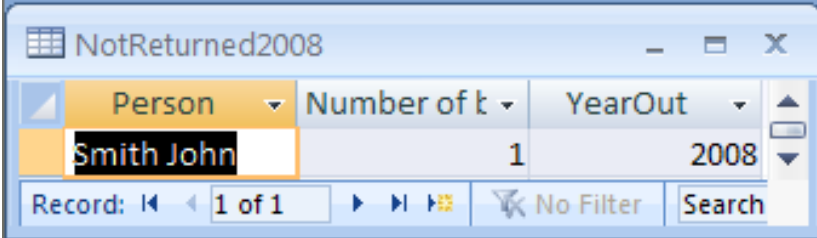
## Creating a copy of a table

```
SELECT *  
INTO NotReturned2008  
FROM Table_NotReturned;
```



Person	Number of k	YearOut
Smith John	1	2008

Record: 1 of 1 | No Filter | Search

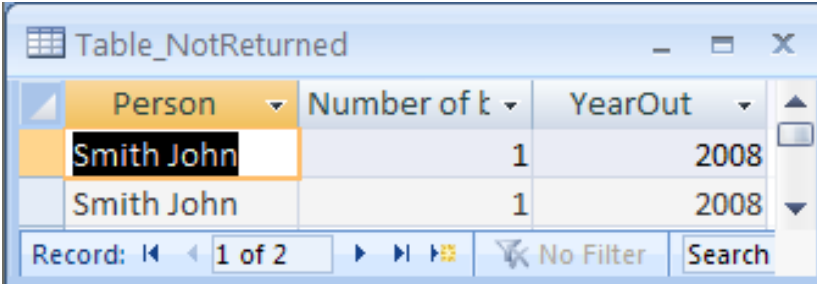


Person	Number of k	YearOut
Smith John	1	2008

Record: 1 of 1 | No Filter | Search

## Merging two tables

```
INSERT INTO Table_NotReturned  
SELECT *  
FROM NotReturned2008;
```



Person	Number of k	YearOut
Smith John	1	2008
Smith John	1	2008

Record: 1 of 2 | No Filter | Search

# Creating reports

## **Characteristic features of a report:**

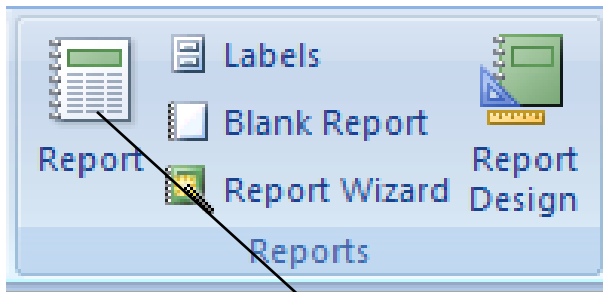
- attractive form
- basic information in headers and footers
- information grouped and sorted
- graphical elements improving the form

## **Methods of creating reports:**

- design view
- Wizard
- AutoReport: Columnar or Tabelar

## **Source of data: tables or queries**

# AutoReport

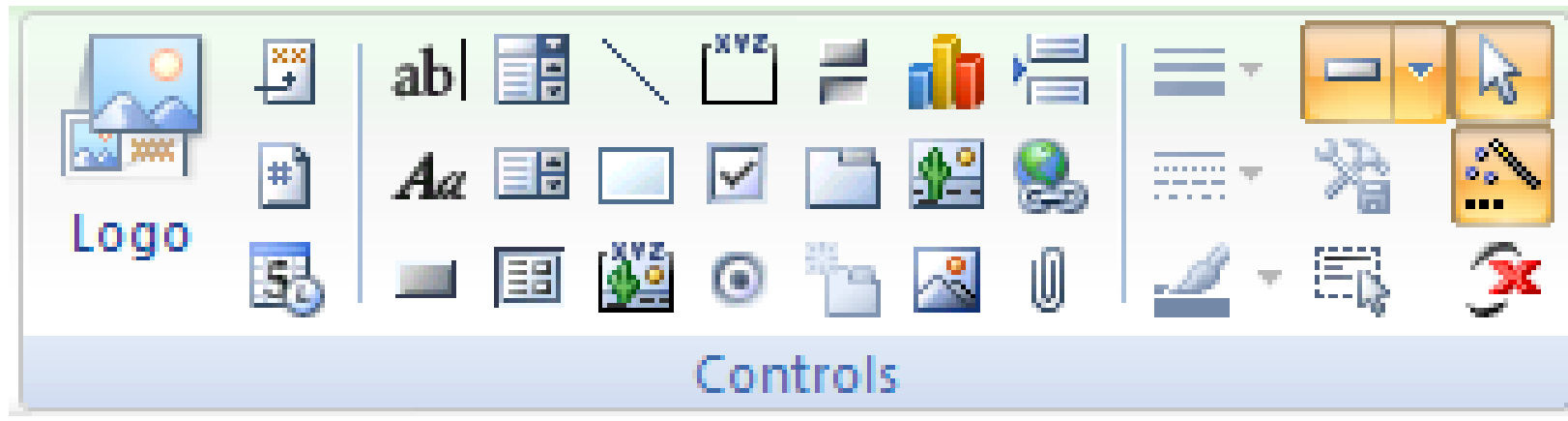
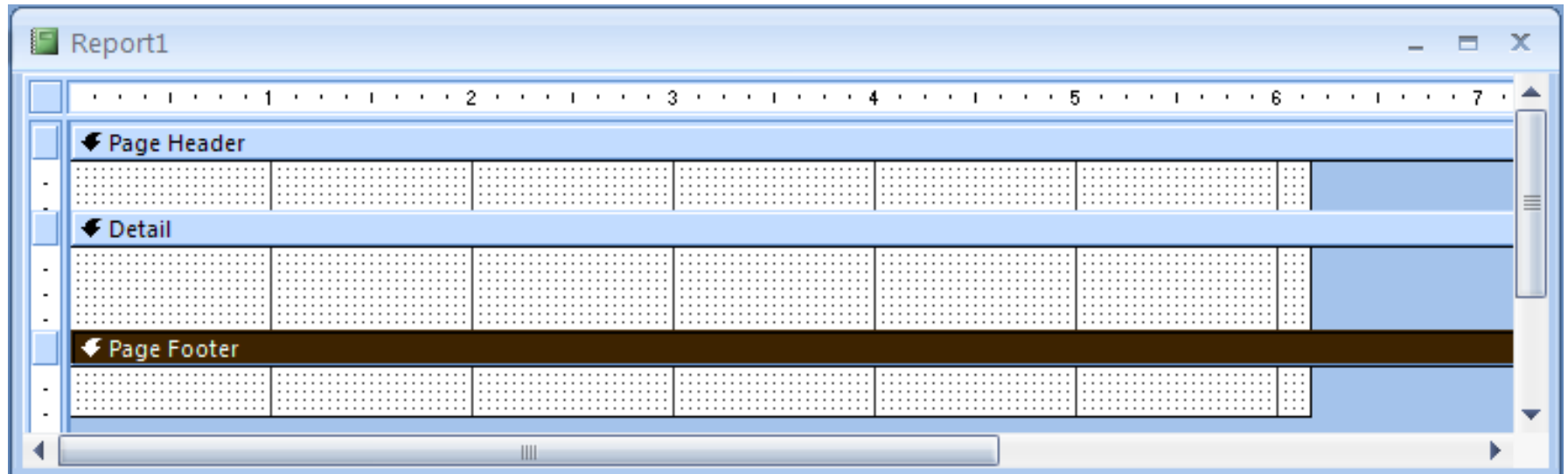


AUTHORS1

AUTHORS

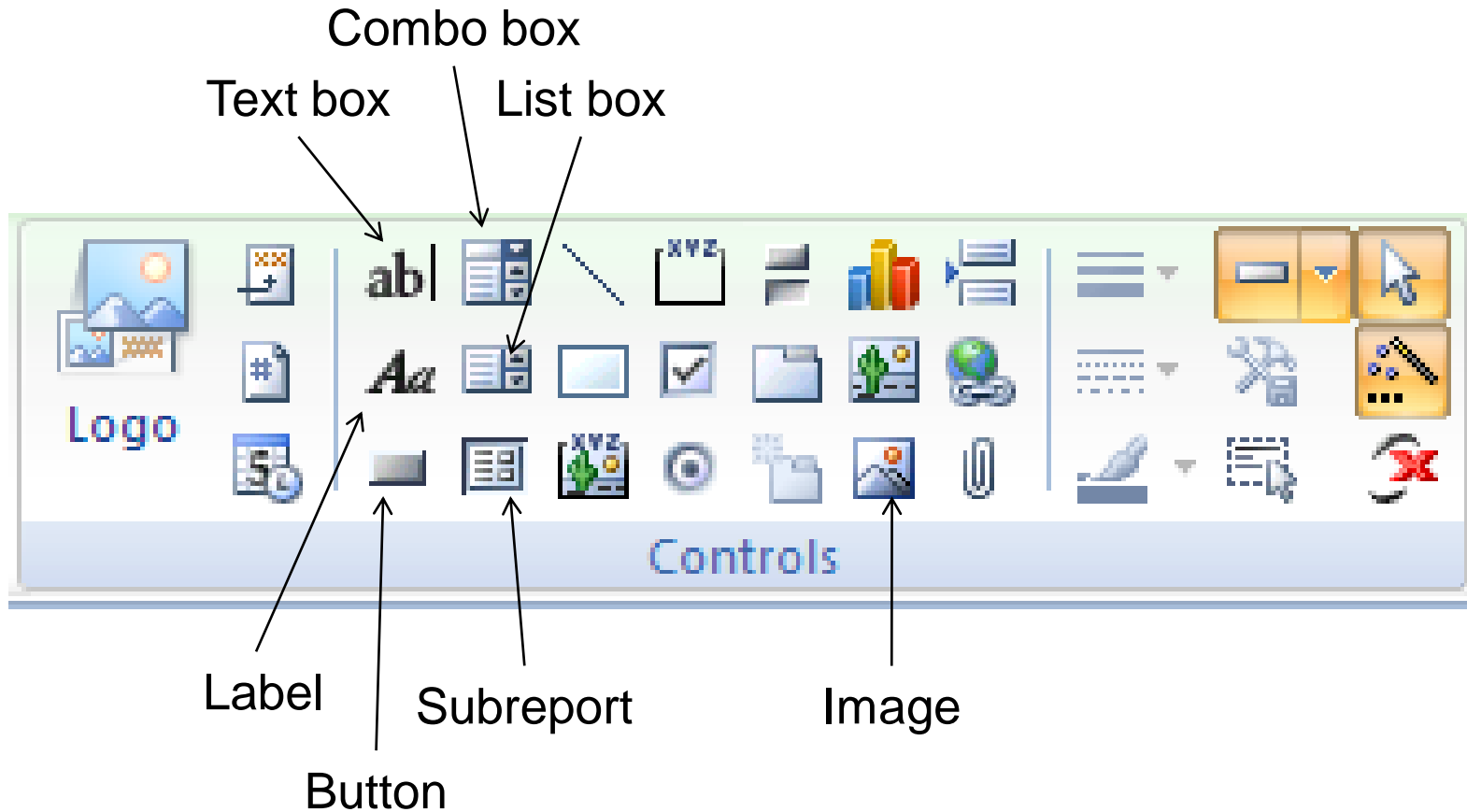
AuID	AuName	AuTel
1	Austen	111-111-1111
2	Melville	222-222-2222
3	Homer	333-333-3333
4	Roman	444-444-4444
5	Shakespeare	555-555-5555
6	Joyce	666-666-6666
7	Spenser	777-777-7777
8	Mill	888-888-8888

# Report – design view





# Toolbox



# Database form

A form is an Access object. It generally serves three purposes:

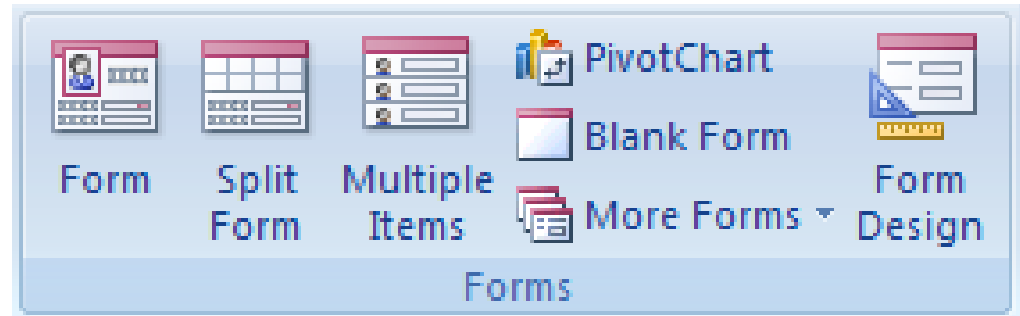
- 1) To allow users to perform data entry. Data can be inserted, updated, or deleted from a table using a Form object.
- 2) To allow users to enter custom information, and based on that information perform a task. For example, you may want to ask a user for parameters before running a report.
- 3) To allow users a method of navigating through the system. For example, you may create a form where a user can select a form to load, a report to run, etc.

## **Forms:** bound or unbound

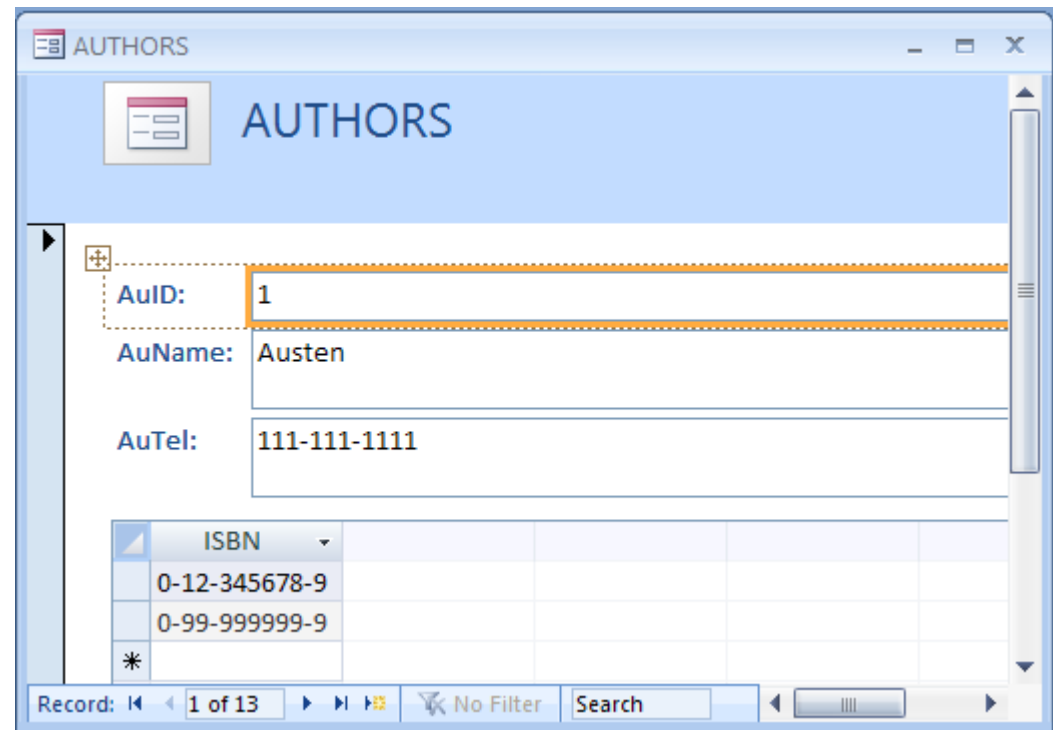
A *bound form* has a RecordSource, a table or query to which the form is "tied" or "based". An *unbound form* does not have a RecordSource, that doesn't mean it can't contain data, but the programmer will have to bring that data in manually.

# Creating Forms in Microsoft Access

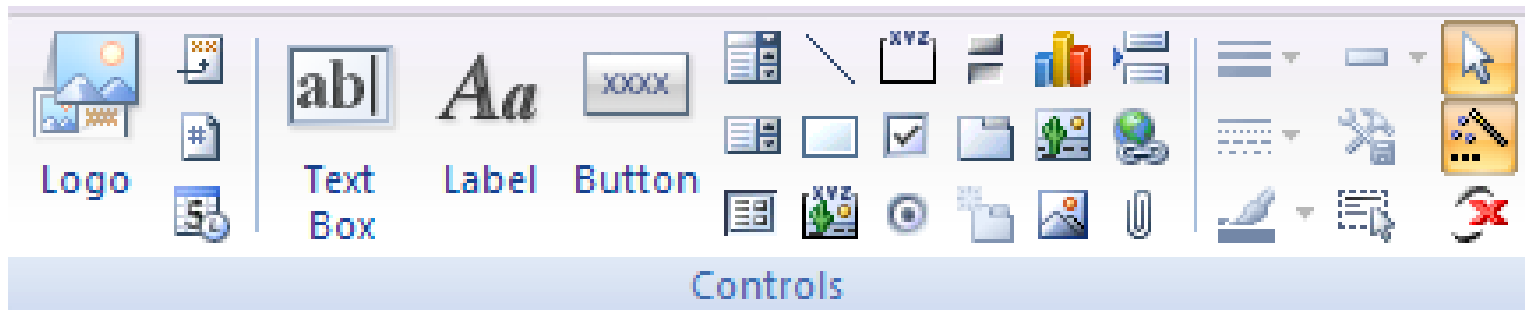
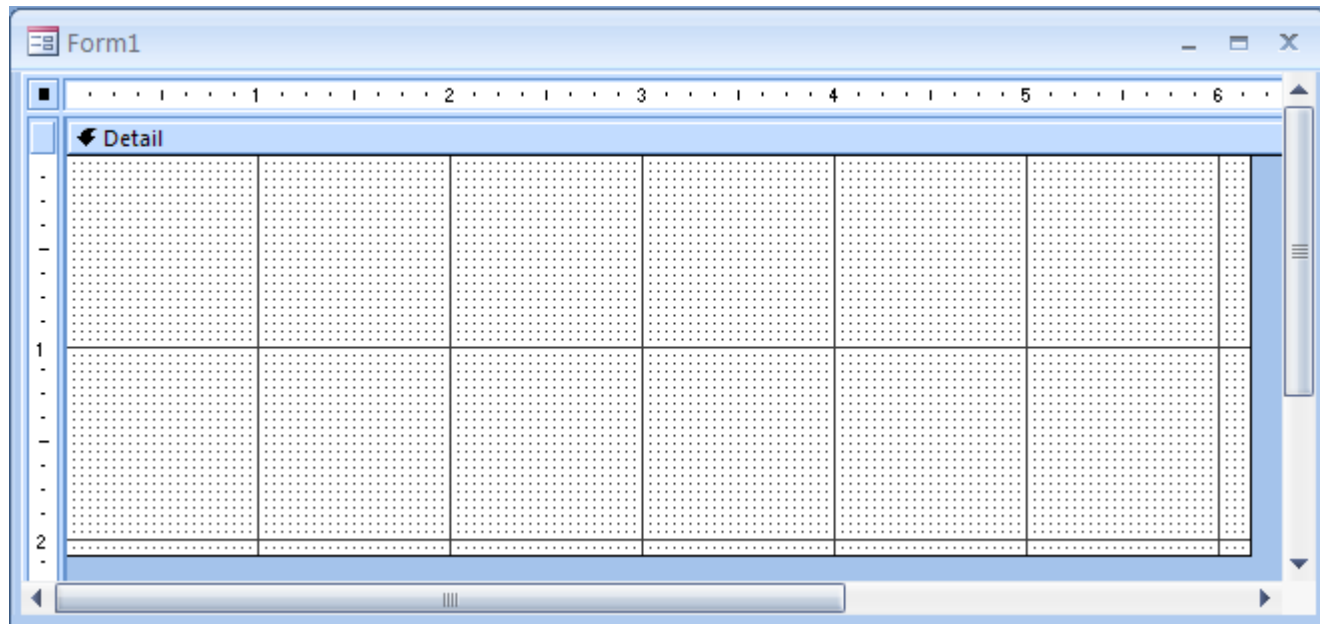
Remember: there is a large number of pre-defined forms.



The Form options quickly create a form based upon a table or query.



# Form Design



# Creating forms – Form Wizard

Form facilitating introduction of a new customer to the database:

Form Wizard

Which fields do you want on your form?  
You can choose from more than one table or query.

Tables/Queries  
Table: CUSTOMER

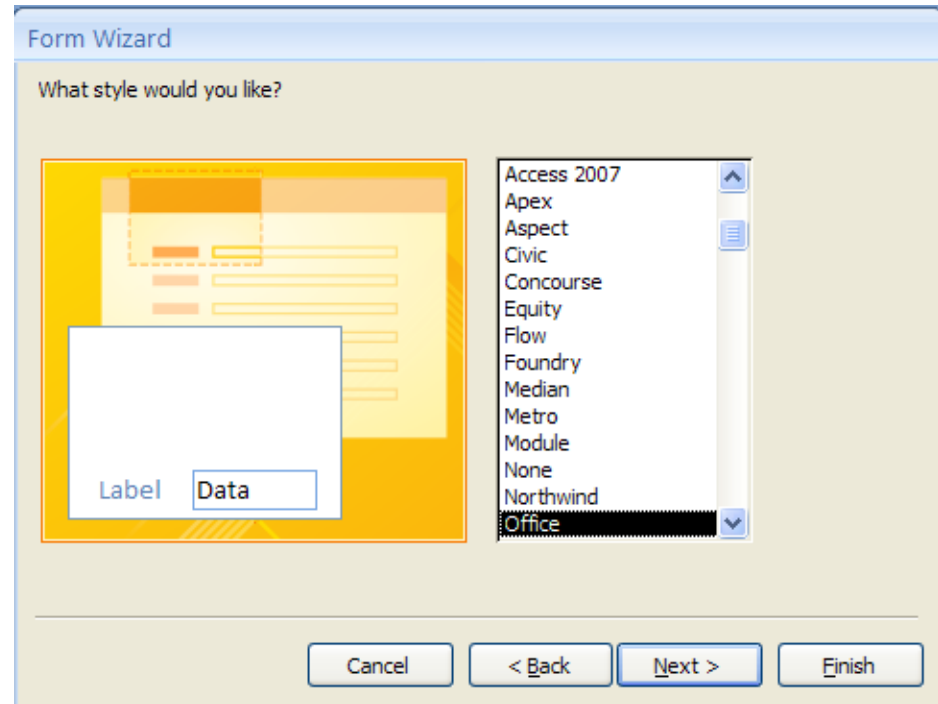
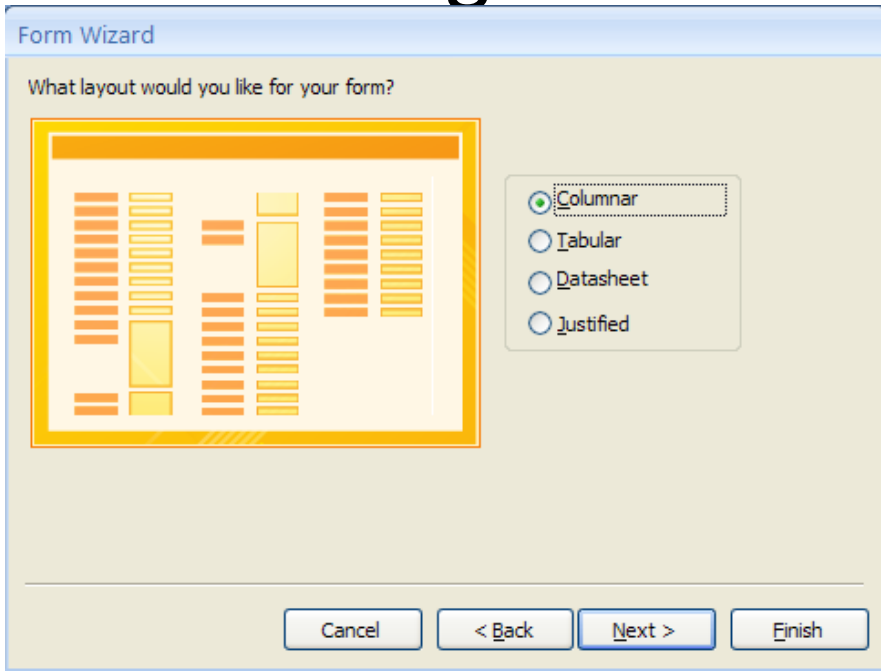
Available Fields:  
CustID

Selected Fields:  
First\_name  
Surname

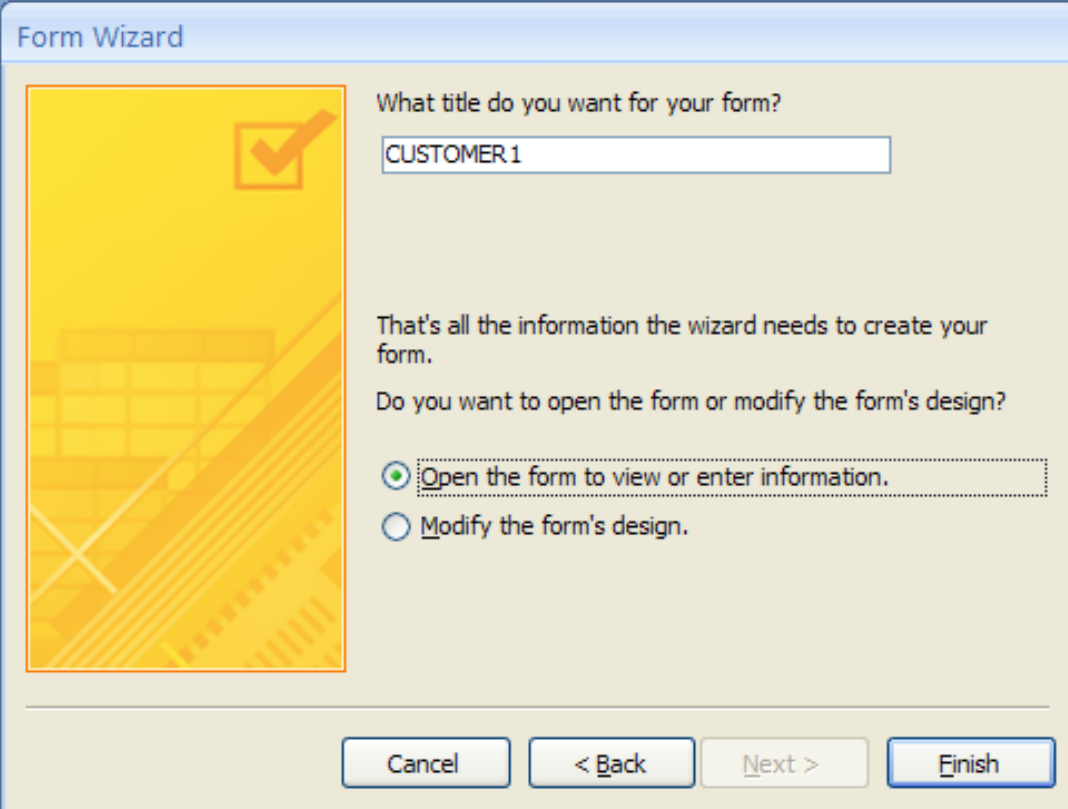
Cancel < Back Next > Finish

- 1) Select the data source
- 2) Select the form fields

# Creating forms – the layout and style



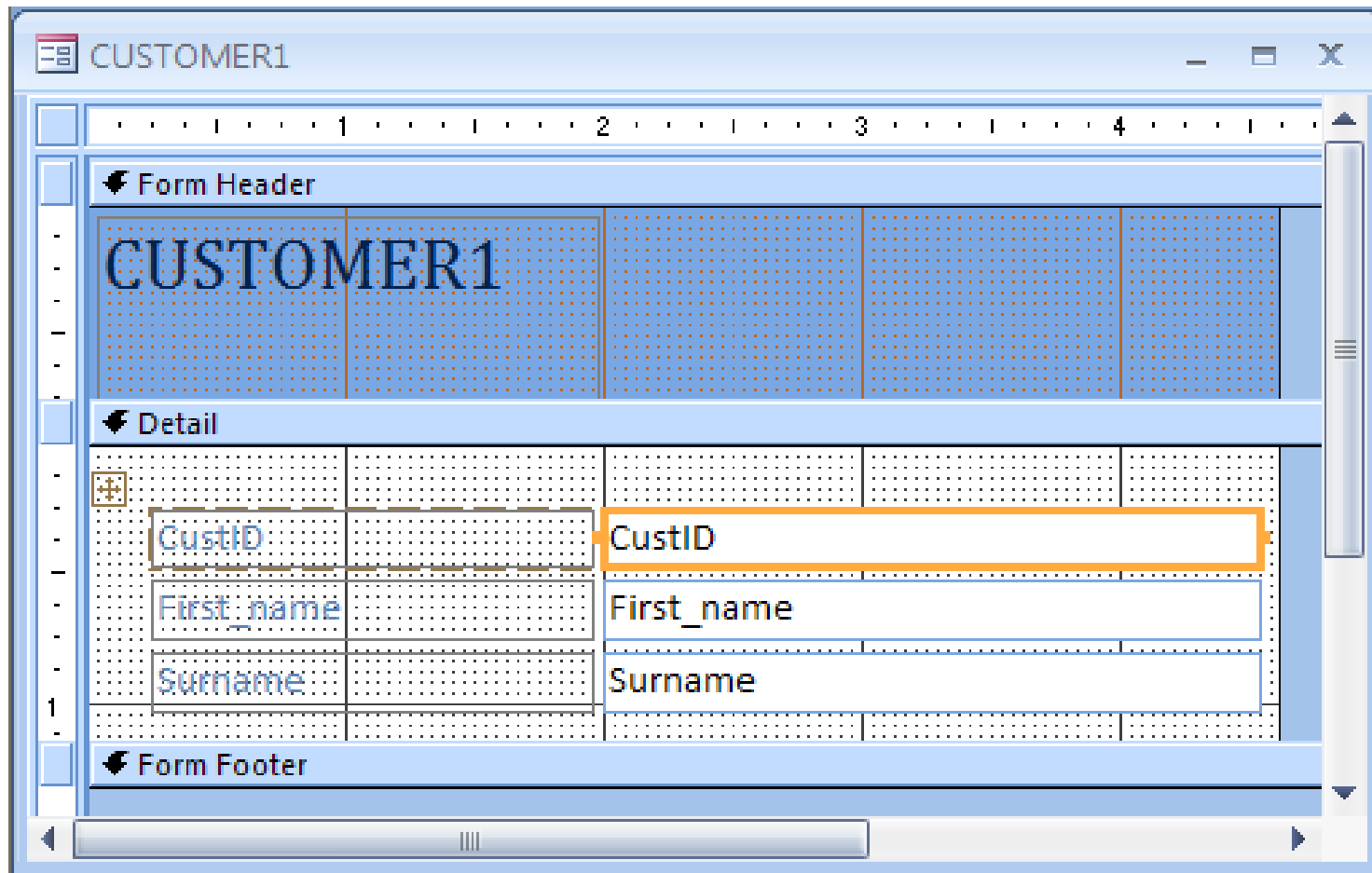
# Creating forms – the form title



The screenshot shows a 'Form Wizard' dialog box. On the left is a yellow graphic with a checkmark and a grid. The main text asks 'What title do you want for your form?' with a text box containing 'CUSTOMER.1'. Below this, it says 'That's all the information the wizard needs to create your form.' and 'Do you want to open the form or modify the form's design?'. There are two radio buttons: 'Open the form to view or enter information.' (selected) and 'Modify the form's design.'. At the bottom are buttons for 'Cancel', '< Back', 'Next >', and 'Finish'.

The form can be modified in Design View mode.

# Creating forms - adjustments





# Using a form

CUSTOMER1

CustID: 1

First\_name: John

Surname: Smith

Record: 1 of 6

Access to the existing records in the table CUSTOMER

CUSTOMER1

CustID: 6

First\_name: Mary

Surname: Jones

Record: 6 of 6

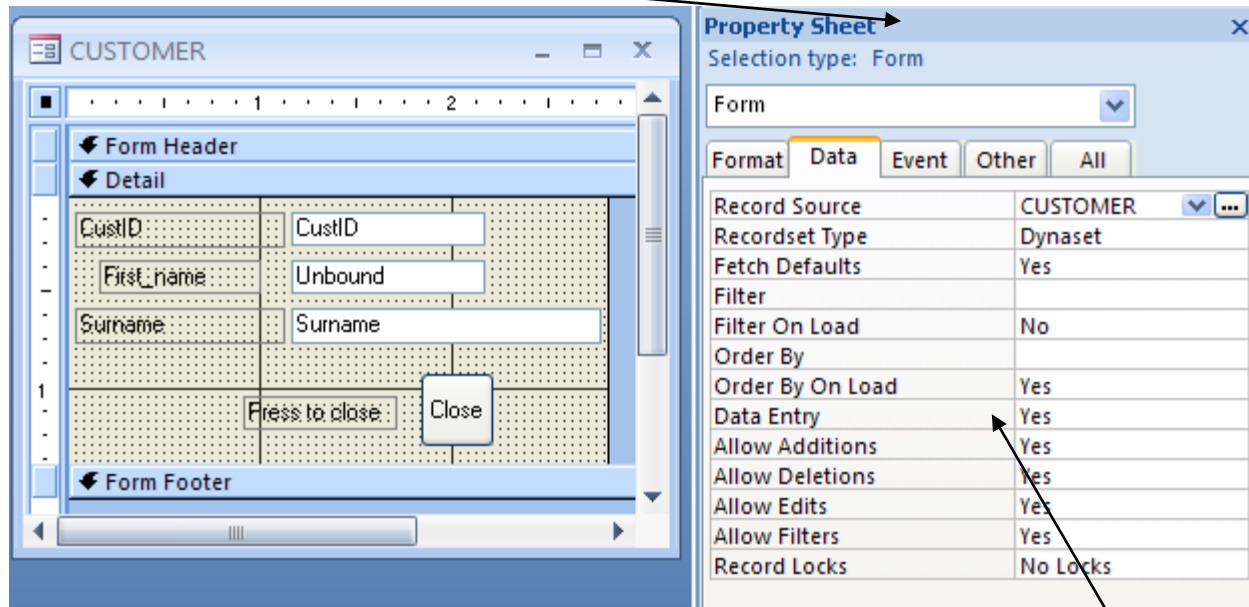
Input of a new record

	CustID	First_name	Surname
+	1	John	Smith
+	2	Anne	Brown
+	3	Henry	Eigth
+	4	Michael	Cooper
+	7	Lara	Croft
+	8	Mary	Jones
*	(New)		

Record: 1 of 6

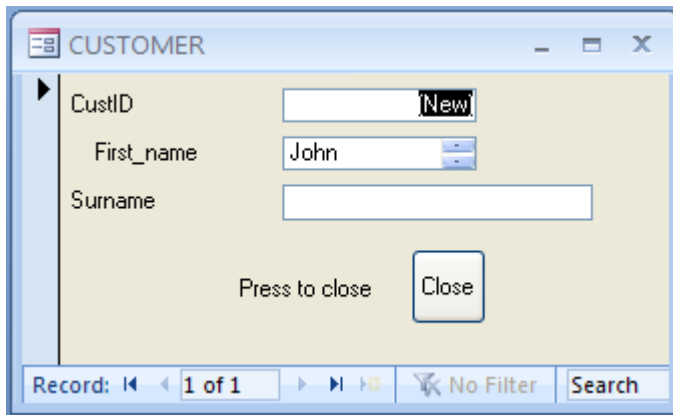
# Editing properties of a form

Properties icon:



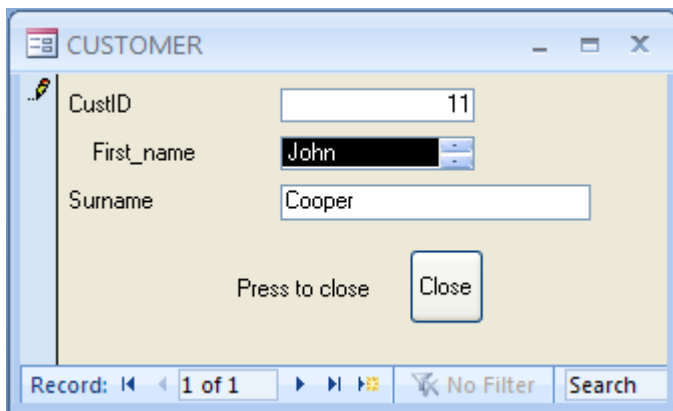
Our original goal is to create a form for data purposes. We don't want to grant employees full access to view or edit customer records. Setting the „Data Entry” property to „Yes” will only allow users to insert new records and modify records created during that session.

# The form after the edition of properties



A screenshot of a Microsoft Access form titled "CUSTOMER". The form has three text boxes: "CustID" containing the text "New", "First\_name" containing "John", and "Surname" which is empty. Below the text boxes is a "Close" button with the text "Press to close" next to it. At the bottom of the form, there is a status bar that reads "Record: 1 of 1", "No Filter", and "Search".

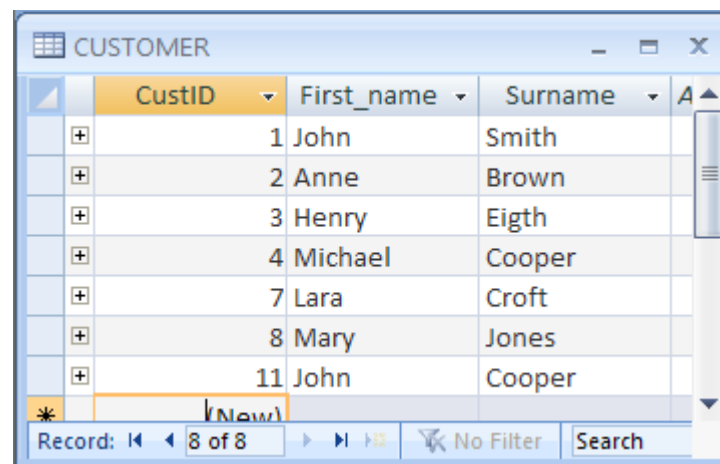
Opening the form: no earlier records can be seen



A screenshot of the same "CUSTOMER" form. The "CustID" text box now contains the number "11". The "First\_name" text box contains "John" and the "Surname" text box contains "Cooper". The "Close" button and status bar remain the same.

The CustID is automatically assigned, First\_name and Surname are typed in.

A new record is introduced into the table CUSTOMER:



A screenshot of a Microsoft Access table view titled "CUSTOMER". The table has four columns: "CustID", "First\_name", "Surname", and a column with a small "A" icon. The table contains eight rows of data. The last row is highlighted in orange and has a "\*" in the first column, indicating it is a new record. The status bar at the bottom reads "Record: 8 of 8", "No Filter", and "Search".

	CustID	First_name	Surname	A
+	1	John	Smith	
+	2	Anne	Brown	
+	3	Henry	Eigth	
+	4	Michael	Cooper	
+	7	Lara	Croft	
+	8	Mary	Jones	
+	11	John	Cooper	
*	(New)			

# Security differences between Access 2007 and earlier versions of Access

Earlier versions:

- security levels (Low, Medium, or High),
- to run potentially unsafe code or not.

Access 2007 disables all potentially unsafe code or other components, regardless of the version of Access that you used to create the database.

Message Bar:



Security Warning Certain content in the database has been disabled

Options...

You can choose to trust or not trust the disabled content in the database.



Security Warning Certain content in the database has been disabled

Options...

## You trust the disabled content:

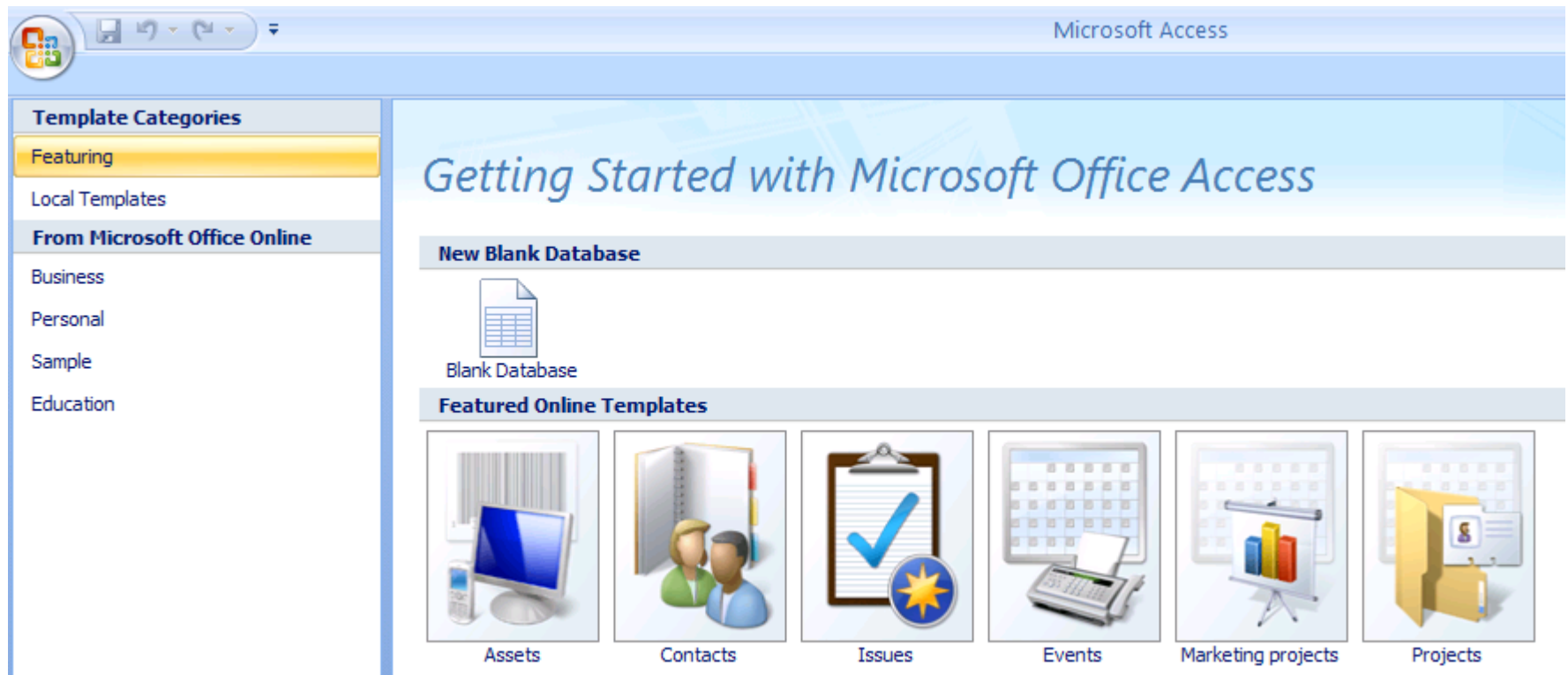
- **Trust the database only for the current session (while the database is open)** Click **Options** on the Message Bar.
- **Trust the database permanently** Place the database in a trusted location — a folder on a drive or network that you mark as trusted.

## You don't trust the database:

Ignore the Message Bar. When you ignore the Message Bar, you can still view the data in the database and use any components in the database that Access has not disabled.

# Trust a database for the current session

1. Start Office Access 2007, and on the **Getting Started with Microsoft Office Access** page, under **Featured Online Templates**, click **Assets** to open the Assets template.



2. In the **File Name** box, type a name for the new database, and then click **Download**. Access downloads the database template and creates a new database, and the Message Bar appears.
3. On the Message Bar, click **Options**. The **Microsoft Office Security Options** dialog box appears.
4. Click **Enable this content**, and then click **OK**.

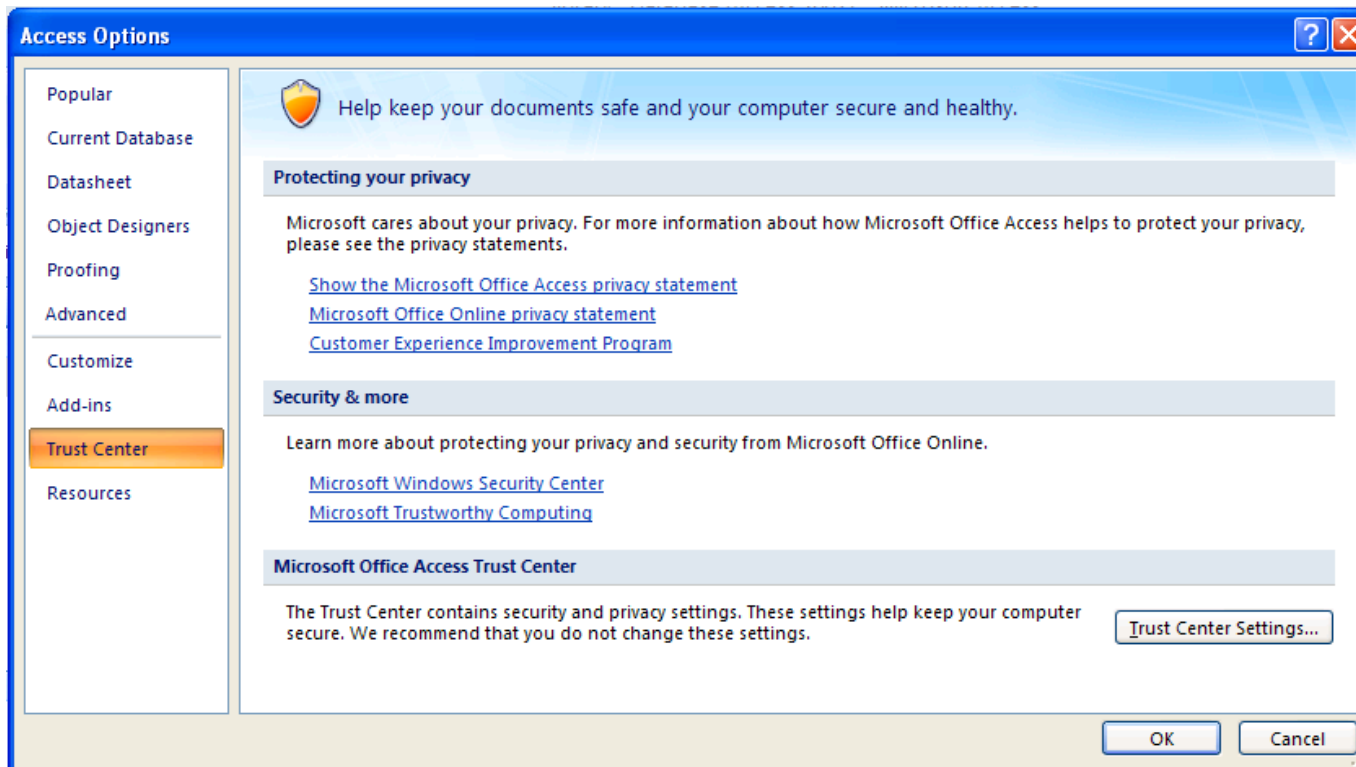


# Create a trusted location

Start Office Access 2007 (you do not need to open a database to complete these steps).

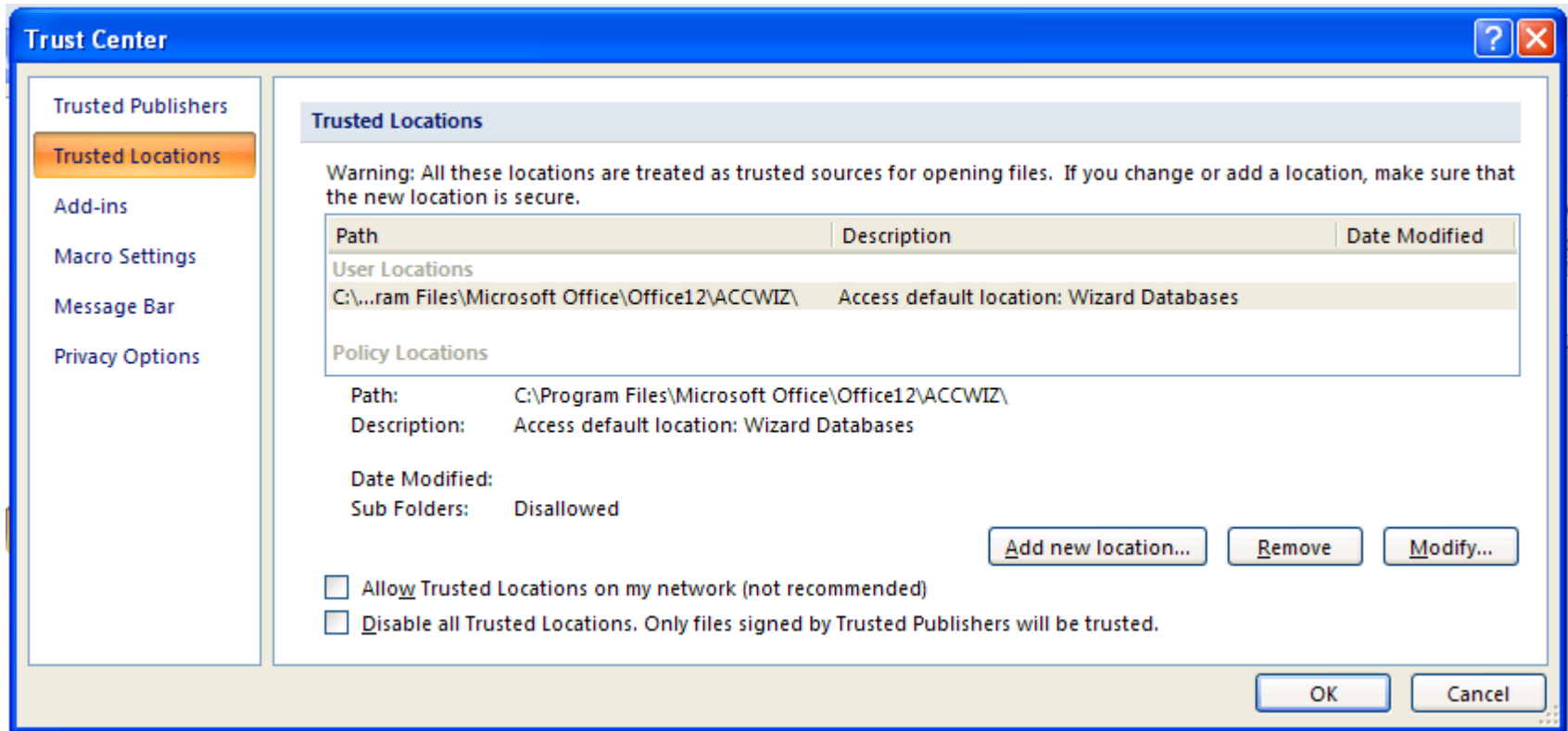
Click the **Microsoft Office Button** , and then click **Access Options**. 

Click **Trust Center**, and then click **Trust Center Settings**.





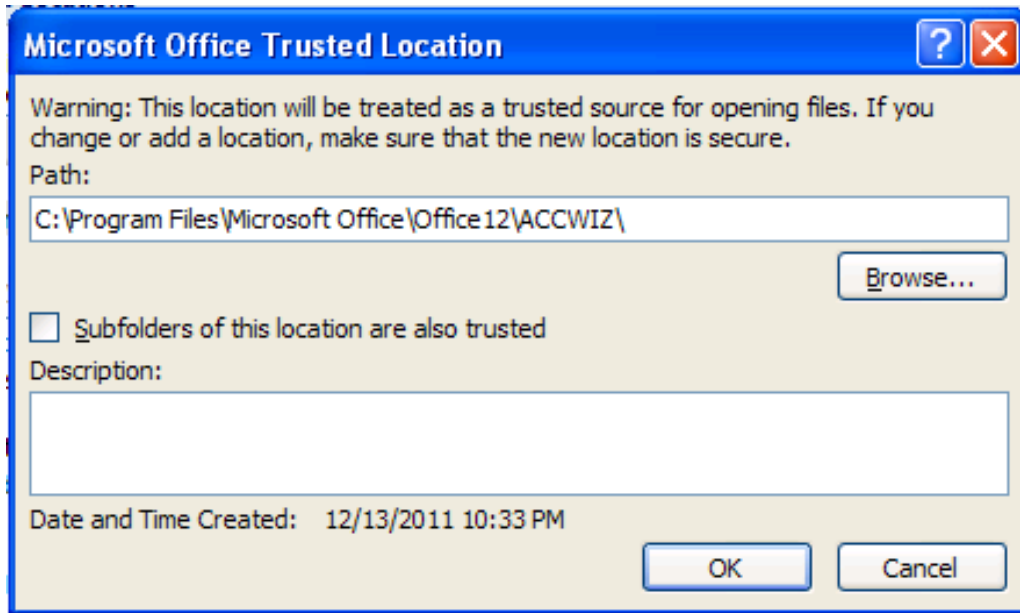
Click **Trusted Locations**.



Click **Add new location**.

The **Microsoft Office Trusted Location** dialog box appears.

The **Microsoft Office Trusted Location** dialog box appears.



In the **Path** box, type the file path and folder name of the location that you want to set as a trusted source, or click **Browse** to locate a folder. By default, the folder must reside on a local drive.

**Note** If you want to allow trusted network locations, in the **Trust Center** dialog box, click **Allow Trusted Locations on my network (not recommended)**.

# Move a database to a trusted location

## If you have a database open

Click the **Microsoft Office Button** .

Point to **Save As**, and under **Save the database in another format**, click one of the available options.

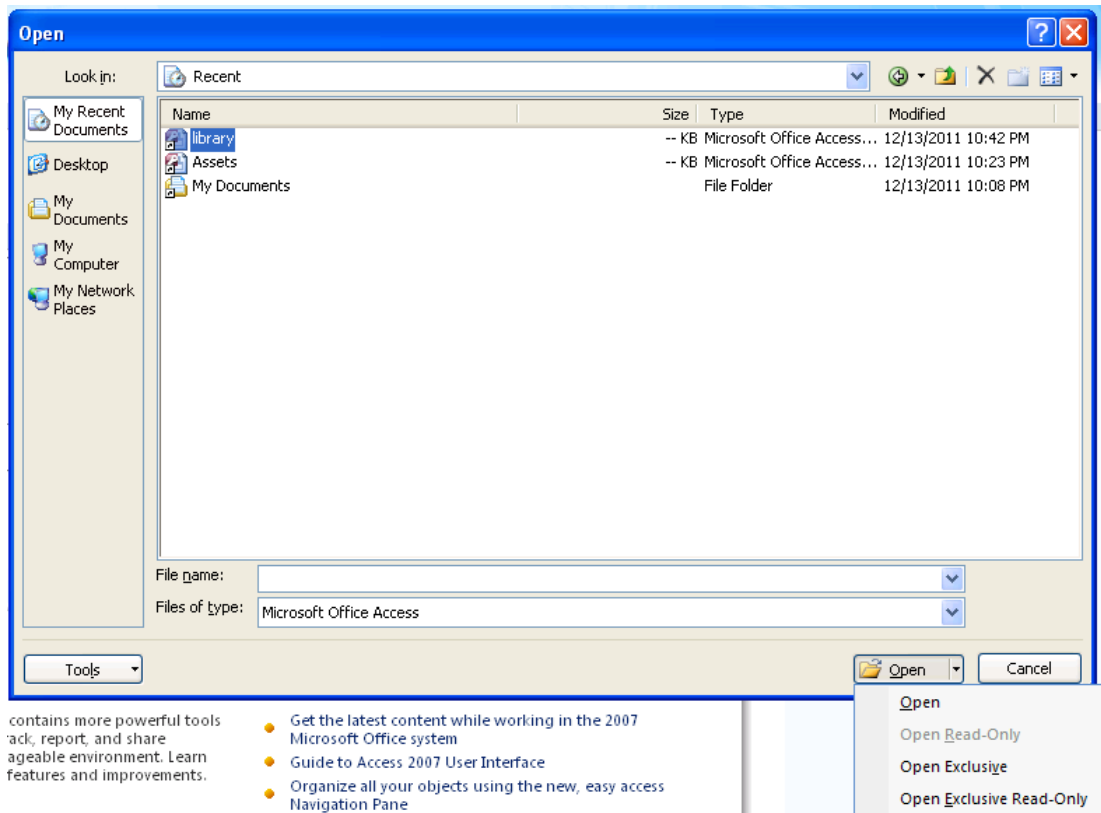
In the **Save As** dialog box, navigate to the trusted location, and then click **Save**.

## If you do not have a database open

Locate and copy your database

# Use a database password to encrypt an Office Access 2007 database

The encryption tool in Office Access 2007 combines and improves on two older tools — database passwords and encoding.

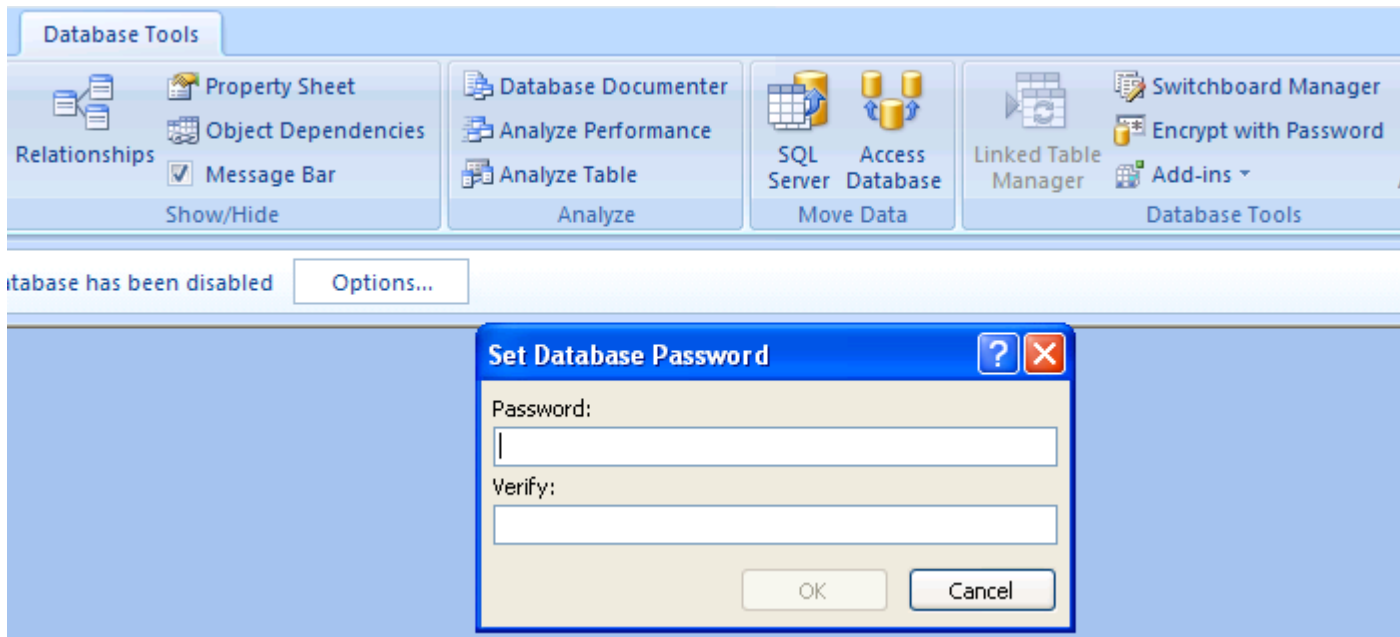


Open in exclusive mode

# Encrypt by using a database password

Open the database (in Exclusive mode) that you want to encrypt.

On the **Database Tools** tab, in the **Database Tools** group, click **Encrypt with Password**. The **Set Database Password** dialog box appears.



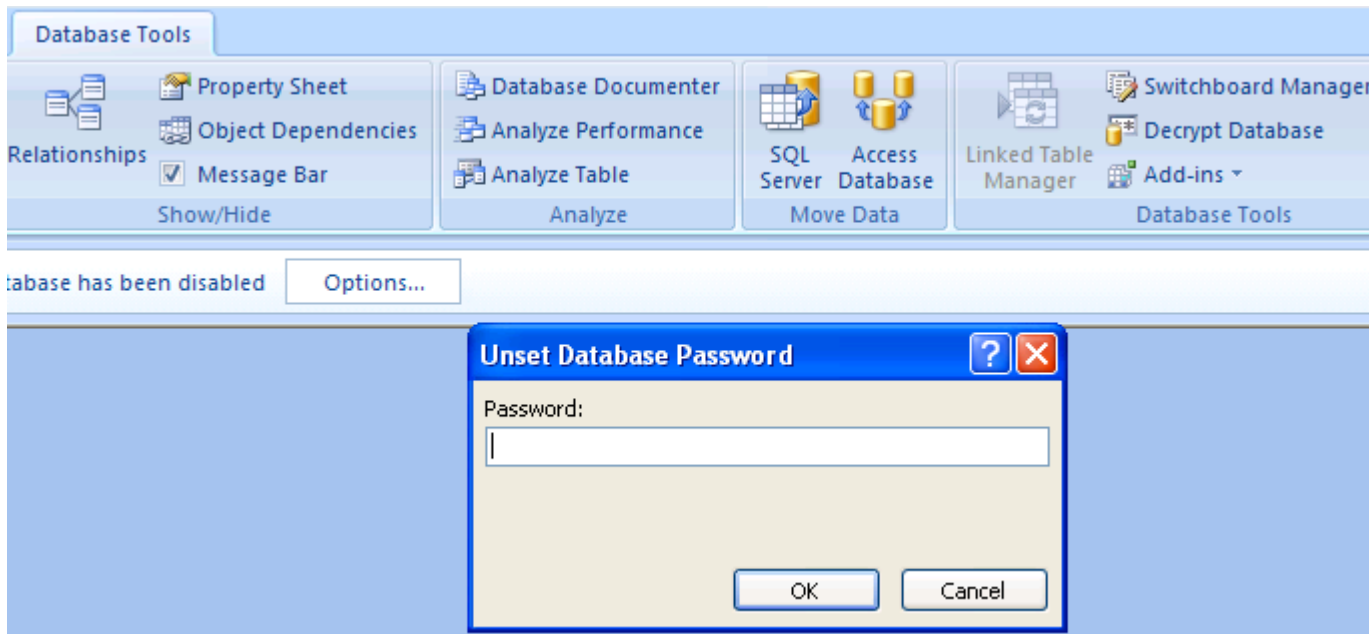
# Decrypt and open a database

1. Open the encrypted database as you open any other database.
2. The **Password Required** dialog box appears.
3. Type your password in the **Enter database password** box, and then click **OK**.

# Remove a password

Open the database in Exclusive mode.

On the **Database Tools** tab, in the **Database Tools** group, click **Decrypt Database**.



# Package, sign, and distribute an Access 2007 database

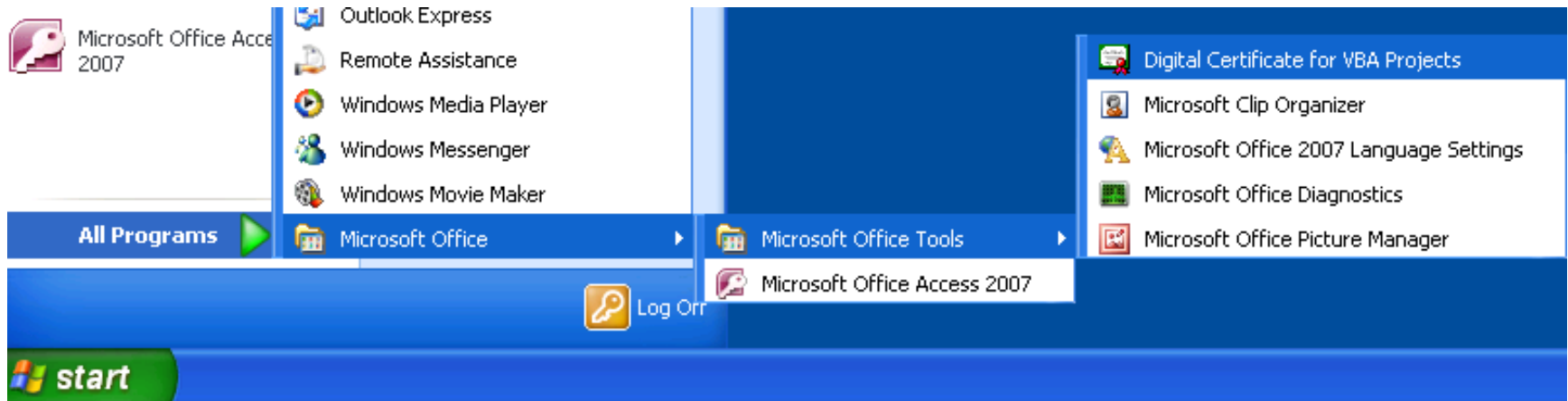
Access 2007 makes it easier and faster to sign and distribute a database. When you create an .accdb file or .accde file, you can package the file, apply a digital signature to the package, and then distribute the signed package to other users.

- ways to convey trust
- add only one database to a package
- signs all of the objects in your databases
- the process also compresses the package file
- a security certificate

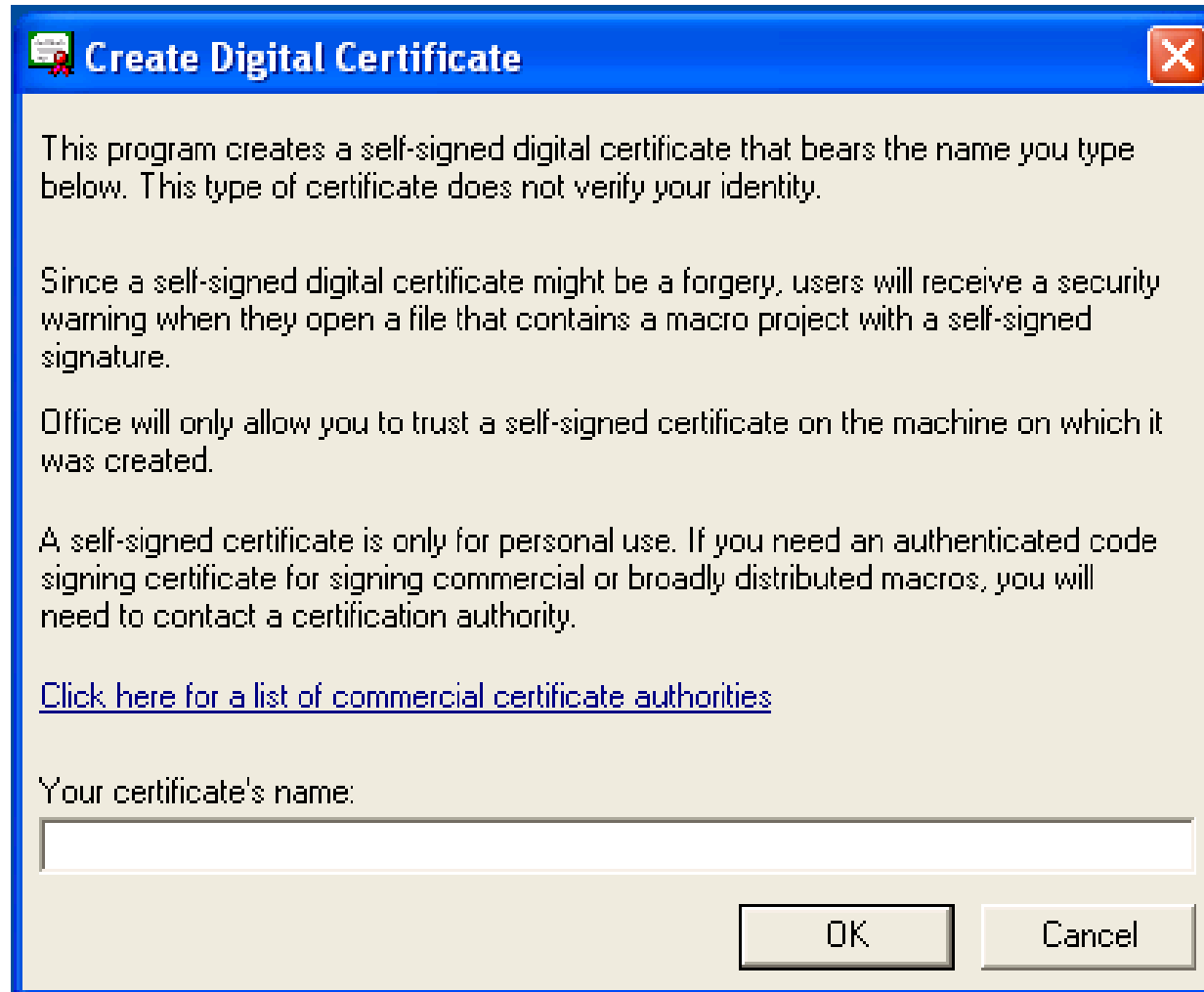


# Create a self-signed certificate

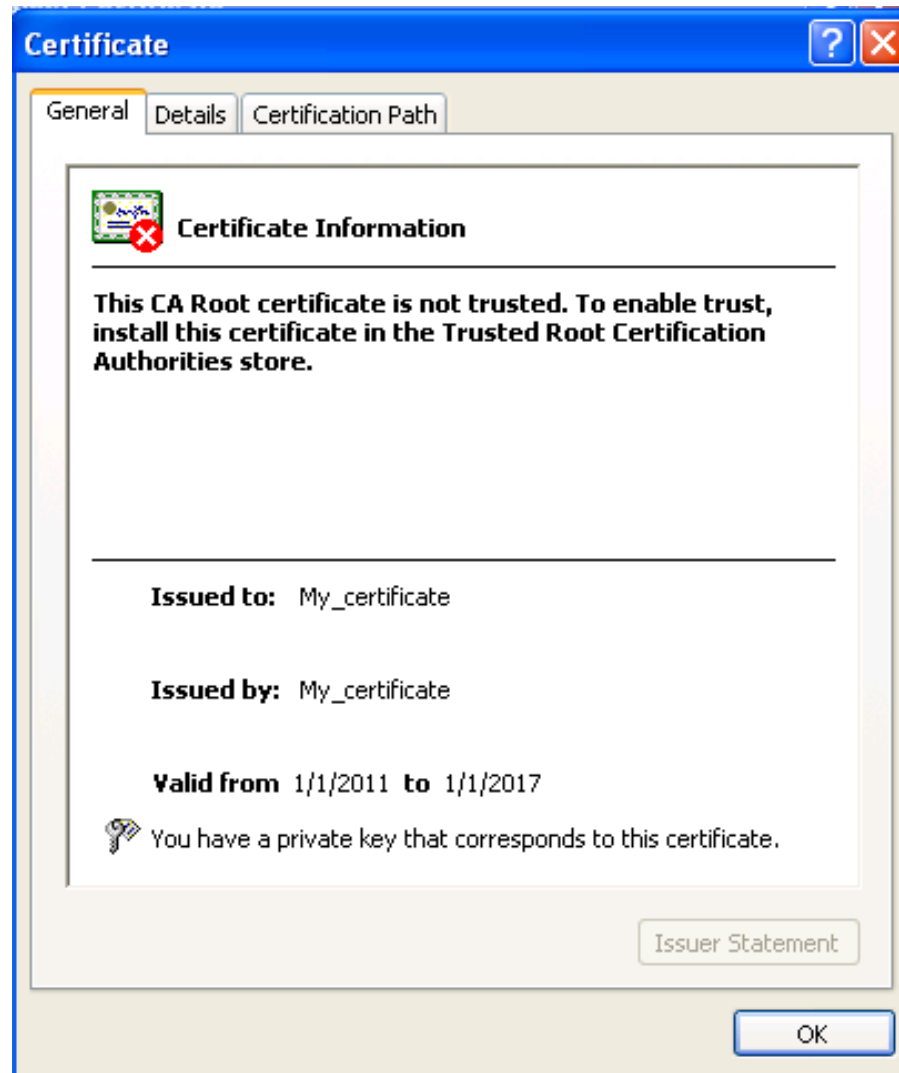
In Microsoft Windows, click the **Start** button, point to **All Programs**, point to **Microsoft Office**, point to **Microsoft Office Tools**, and then click **Digital Certificate for VBA Projects**



# Create a self-signed certificate



# Create a self-signed certificate



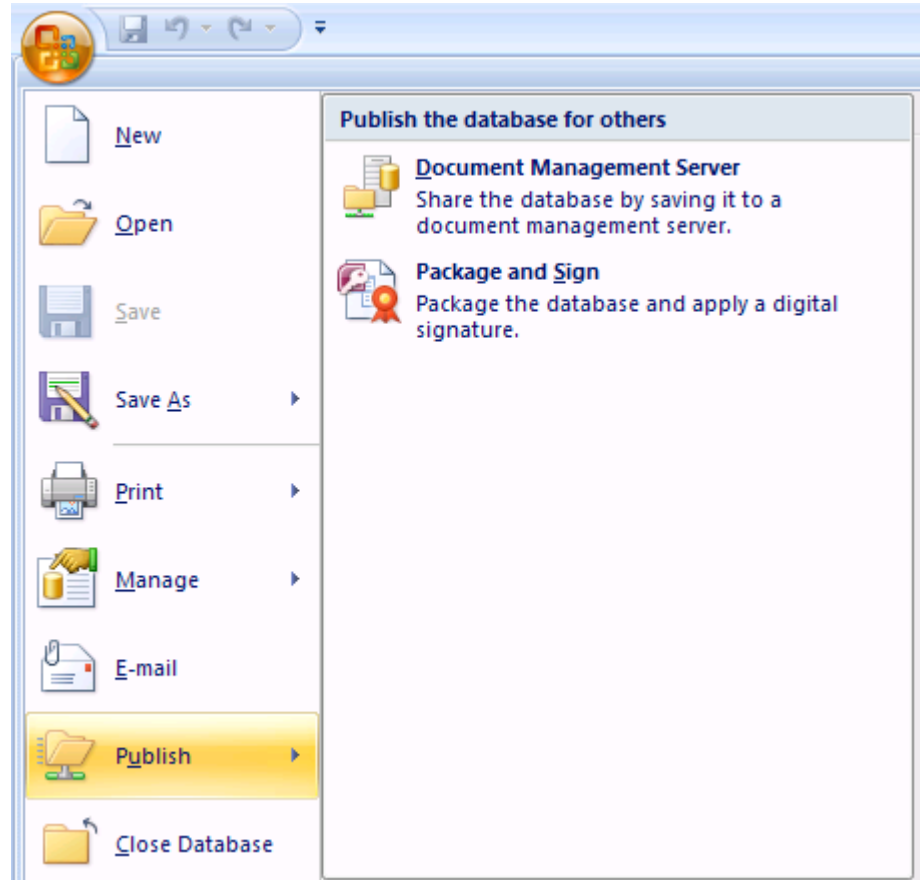
# Create a signed package

Open the database

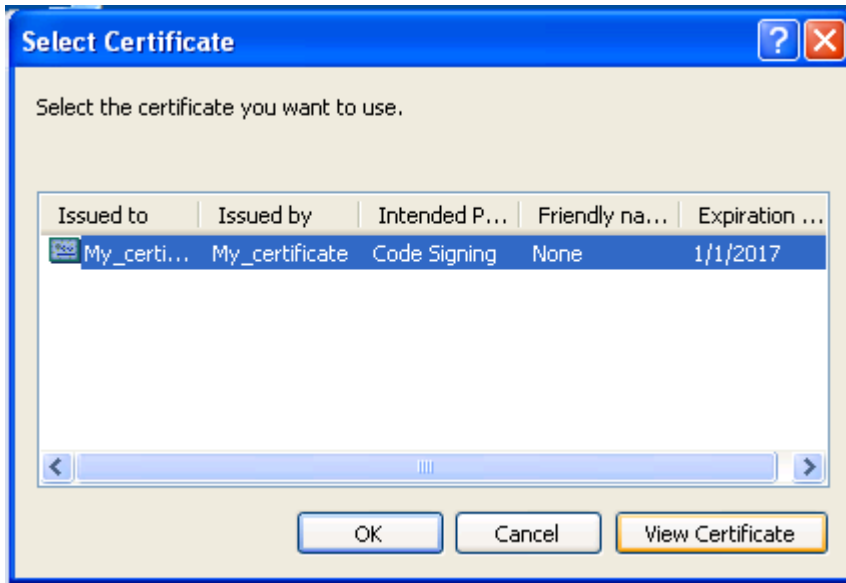
**Microsoft Office Button** ,  
point to **Publish**, and then  
click **Package and Sign**

Select a digital certificate and  
then click **OK**

The **Create Microsoft Office  
Access Signed Package**  
dialog box appears



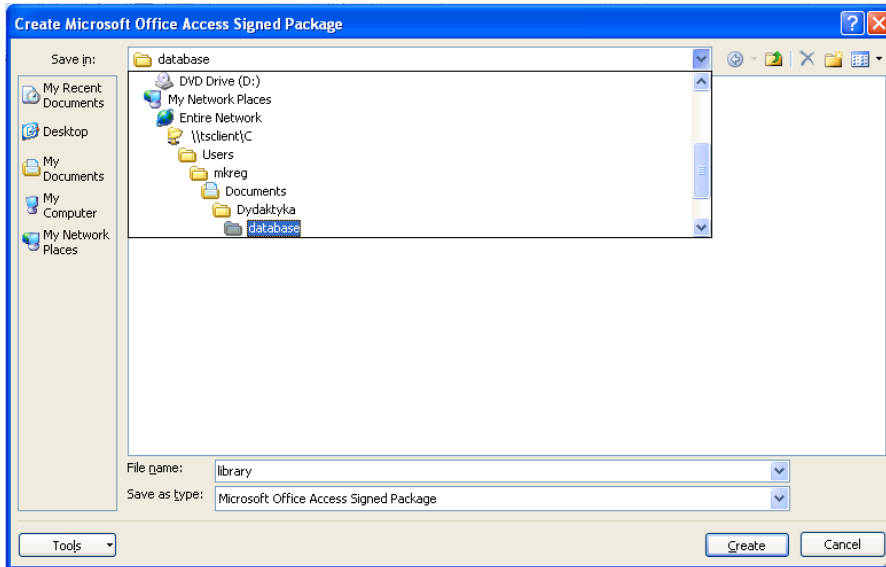
# Create a signed package



Select a location for your signed database package.

Enter a name for the signed package in the **File name** box, and then click **Create**.

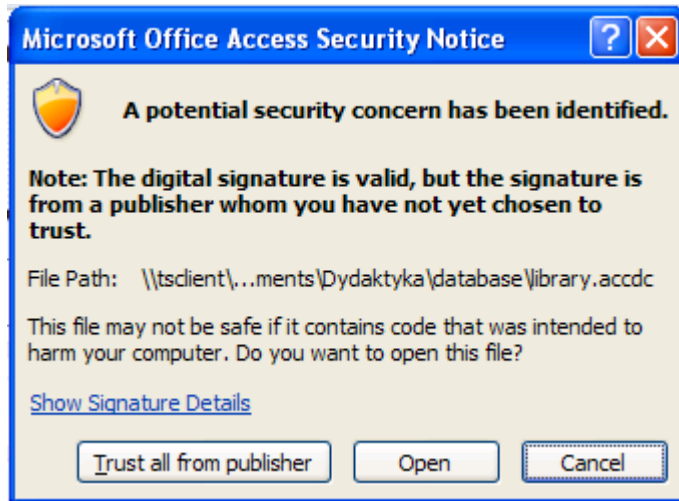
Access creates the .accdc file and places it in the location that you choose



# Extract and use a signed package

1. Click the **Microsoft Office Button** , and then click **Open**.
2. Select **Microsoft Office Access Signed Packages (\*.accdc)** as the file type.
3. Locate the folder that contains your .accdc file, select the file, and then click **Open**.
4. Do one of the following:
  - *If you have earlier chosen to trust the digital certificate that was applied to the deployment package, the **Extract Database To** dialog box appears. Go to the next step.*
  - *If you have not yet chosen to trust the digital certificate, an advisory message appears.*

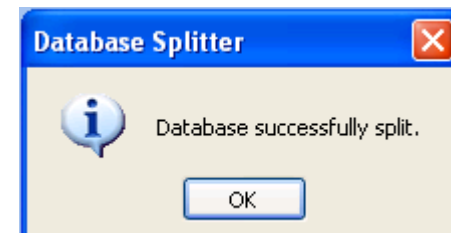
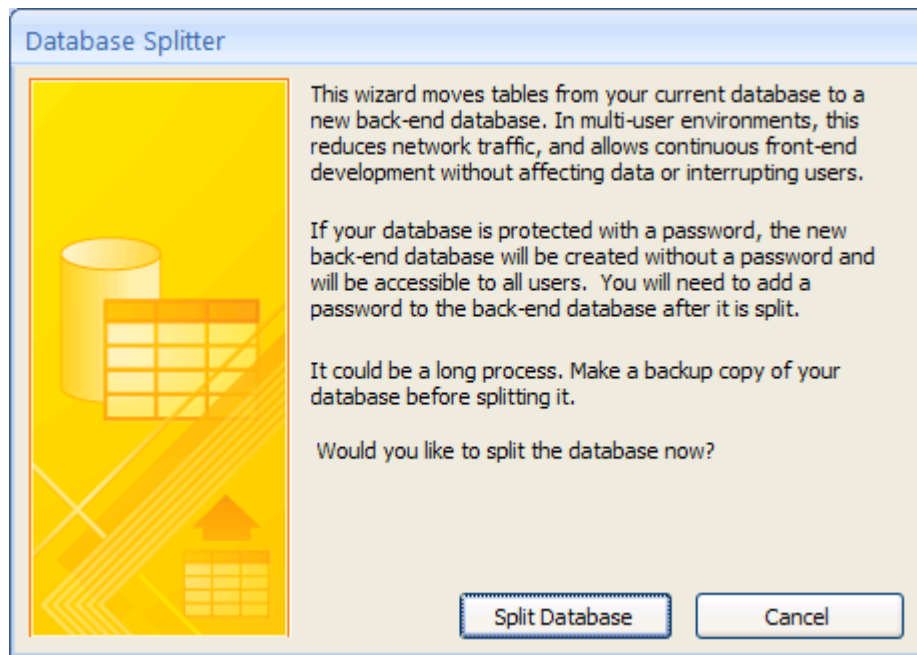
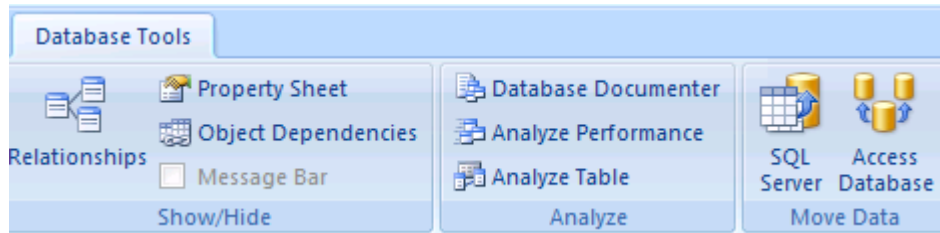
# Extract and use a signed package



If you trust the database, click **Open**. If you trust any certificate from that provider, click **Trust all from publisher** and then click **OK**.

Optionally, select a location for the extracted database, and then in the **File name** box, enter a different name for the extracted database

# Splitting a database

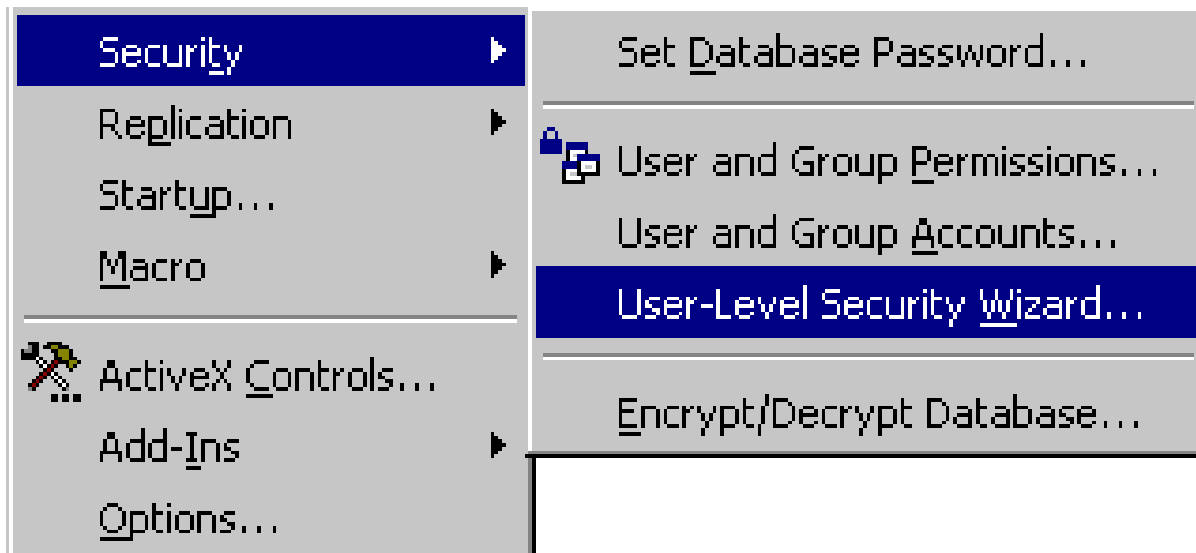




# Hide VBA code from users

If your database contains Visual Basic for Applications (VBA) (Visual Basic for Applications (VBA): A macro-language version of Microsoft Visual Basic that is used to program Windows applications and is included with several Microsoft applications.) code, you can hide that code by saving your Microsoft Office Access database (database: A collection of data related to a particular subject or purpose. Within a database, information about a particular entity, such as an employee or order, is categorized into tables, records, and fields.) in the .accde file format. Saving a database as an .accde file compiles all VBA code modules, removes all editable source code, and compacts the destination database. Your VBA code retains its functionality, but the code cannot be viewed or edited. In general, the database will continue to function as usual — you can still update data and run reports.

# Microsoft Access Security



The Security Wizard in MS Access is a very useful tool. But there is more to security than just running the wizard. The 12 steps will describe how to secure a database from start to finish.

# How to secure a database?

1. You may secure any database that has been created while joined to the System.Mdw.
2. Create a new workgroup file(\*. MDW)
3. Open the unsecured database and create a password for the Admin user.
4. Create a new user account that will be the new workgroup administrator, like "PowerAdmin"
5. Modify the Admins group by adding the new administrator and removing the original Admin account.
6. Re-log into Access as the new workgroup administrator that you created previously.
7. Set up a password for the new workgroup administrator.
8. Run the Security wizard under Tools | Security | User - Level Security.
9. Create any additional group accounts.
10. Create any additional user accounts.
11. Set up the database object's permissions.
12. Any new databases that you create will already be secure.

# System.MDW file

A Workgroup Information File (\*.MDW) stores information to authenticate a user. It stores the user names, group names, and passwords. It does not store any permission or rights to any database. Its main purpose is to verify that a user is really who they say they are. The permissions of the database objects, tables, queries, forms, etc., are stored in each MDB file. The System.mdw is the default workgroup filename created when you install MS Access.

# The Admin user

Every time a user opens the MS Access program, MS Access attempts to login the Admin user with a blank password. If the log in is successful, MS Access continues loading and the user never realizes that they were logged in as Admin. However, if the login is unsuccessful, say for example the Admin user does not have a blank password, then a login dialog box pops up asking the user to specify a username and password

When you create a new User, you will be prompted for a User Name, Password, and a PID or Personal Identification number. A PID can be any text or numbers up to 20 characters long. All three values uniquely identify each user.

# The database.MDB file

MS Access verifies that the user name and password exist in the Workgroup Information File. After the user has been verified, the workgroup information file's job is done. The MDB itself stores security rights and privileges for each user and for each database object.

The MDB will have a list of user id's and the privileges that each user may have. One user may have the rights to open the table, but not delete any records, or change the design of the table. Another user may not have any restrictions at all. The MDB file knows each user's privileges. The distinction between the workgroup information file and the MDB file is one that confuses a lot of developers.

# 1. Workgroup file

All the databases that you have been creating up to this time have been created by the Admin user of the default System.Mdw. The System.Mdw is a generic workgroup file. All System.Mdw on everyone's computers has the exact same Admin account with no password and with a PID of Null, making the Admin user account not very secure to continue to use.

The database does not have to be completely developed to continue setting up security. You may continue to work on it after it is secured.

You can secure a database that has already been create with a different workgroup file, if you can log into the database with the account that created the database and if that account still has full rights to all the database objects.

## 2,3. Create a new workgroup file(\*. MDW)

Use the MS Access Workgroup Administrator program. Do not leave the Workgroup ID blank or anyone will be able to create a workgroup file similar to yours. You can name the workgroup file anything you'd like. It does not have to be named System.MDW. In fact it should have a different name to avoid confusion.

It's a good idea to keep a back up copy of any workgroup file you create file offsite and in a secure location should anything happen to the original.

By establishing a password for the Admin user, Access will ask for a user name and password every time it Access is opened.



# New workgroup administrator

Instead of continuing to use the Admin user, we will disable the Admin user's abilities. The reason for this is that every workgroup file has an Admin user, so using it will make our database less secure. I like to name my new workgroup administrator something like PowerAdmin. Write down the new name and PID. Store this information offsite in a secure location.

# 5. Modifying the Admins group

Any user in the Admins group will have full control of all the database's objects. They can also control other user's access to the database's objects.

By removing the Admin account from the Admins groups, the Admin becomes virtually useless. After our database is secure, if any one successfully logs in as the Admin user, they find that they don't have very many rights. If anyone needs to log in as a user with full rights, they should use the new workgroup administrator that we created in the previous steps.

# 6,7. New workgroup administrator

Re-Log into Access the new workgroup administrator that you created previously.

Up to this point, the new workgroup administrator doesn't have a password. You can set up a password for this account under *Tools | Security | User and Group Accounts | Change Log On Password*.

# 8. Run the Security wizard

Security wizard under *Tools | Security | User - Level Security*.

- 1) This step creates a new database and copies all the current database's objects into it.
- 2) The Wizard sets the owner of the database from Admin to the new workgroup administrator. This is done by creating a new database while logged in as the new workgroup administrator.
- 3) The Security Wizard changes the owner of each object in the database from Admin to the new workgroup administrator.
- 4) The Security Wizard removes all permissions from the Admin user.

# 9. Create any group accounts

For example, you may want to create a group named PowerUsers for users that may need lots of database permissions. You may also create a Personnel group account for those user that only need limited rights.

# 10. Create user accounts

Assign users to the appropriate groups.

If a user belongs to more than one group, the user receives the sum or combination of the permission given to each group.

# 11. The object's permissions

Change the database object's permissions for each group account.

To make permissions easier for you to manage, try and avoid setting permissions for individual user.

Remember that there are several database object types for which you may need to set permissions.

- Database object

  - Table objects

  - Query objects

  - Form objects

  - Report objects

  - Macro objects

  - Module objects

12. Any new databases that you create will already be secure as long as you are still joined to the workgroup file that you created in step #2. Just repeat step #11 to grant permissions to the object in your new database.



# Security gone awry

MS Access help file states: '.if you lose or forget your password, it can't be recovered, and you won't be able to open your database.' If you still need to open your database, try

<http://www.LostPassword.com>.

The site has password recovery tools for the most popular office software, ranging from MS Access to FileMaker and from MS Word to WordPerfect.

Access Key is a Password Recovery Kit module that recovers both database and user-level security accounts passwords. There's also an Access Security Recovery service (available for legal owners/copyright holders only) that comes handy if a Workgroup Information File is lost or corrupted. You can find more info at

<http://ref.lostpassword.com/access.htm?900407>