

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Запорізький національний технічний університет

МЕТОДИЧНІ ВКАЗІВКИ

до вивчення курсу “Програмні засоби проектування ЕМС» (частина 1)
та виконання лабораторних робіт для студентів спеціальності 141
"Електроенергетика, електротехніка та електромеханіка" (освітні
програми «Електричні та електронні апарати»)
з англійською мовою навчання

Методичні вказівки до вивчення курсу «Програмні засоби проектування ЕМС» (частина 1) та виконання лабораторних робіт для студентів спеціальності 141 "Електроенергетика, електротехніка та електромеханіка" (освітні програми «Електричні та електронні апарати») з англійською мовою навчання / Укл.: Л.О.Бондаренко,
– Запоріжжя: ЗНТУ, 2018.– с.58.

Укладачі:

Л.О. Бондаренко

Рецензент:

М.О.Поляков

Відповідальний за випуск:

П.Д.Андрієнко

Затверджено
на засіданні кафедри
“Електричні та електронні апарати”
Протокол № 11 від 1.06.2018 р.

Рекомендовано до видання НМК
Електротехнічного факультету
протокол № 11 від 21.06.2018 р.

CONTENTS

1. Laboratory work №1. Creating a database in Relational	
DataBase Management System (RDBMS) Microsoft Access.....	5
1.1 Purpose of the work.....	5
1.2 Brief theoretical information	5
1.3 The order of performance of the laboratory work	8
1.4 Review and discussion questions	15
1.5 The contents of the report.....	11
2. Laboratory work №2. MS Access Queries	16
2.1 Purpose of the work.....	16
2.2 Brief theoretical information	16
2.2.1 Simple query.....	17
2.2.2 Selection query	20
2.2.3 Multi-table queries.....	27
2.2.4 Create a calculated field in a query	28
2.3 The order of performance of the laboratory work	29
2.4 Review and discussion questions	29
2.5 The contents of the report.....	30
3. Laboratory work №3. Working with Forms and Reports	31
3.1 Purpose of the work.....	31
3.2 Brief theoretical information	31
3.2.1 Creating forms by using the Form Tools.....	32
3.2.2 Formatting forms	36
3.2.3 Create reports by using a wizard	40
3.3 Tasks for the laboratory work	43
3.4 Review and discussion questions	45
3.5 The contents of the report.....	46
4. Laboratory work №4. Creating and modifying a form	
in design view.....	47
4.1 Purpose of the work.....	47
4.2 Brief theoretical information	47
4.3 Examples of development of forms for a DB "Library"	50
4.3.1 Forms of data input.....	50
4.3.2 Forms of viewing of the data by various choice criteria ...	51
4.3.3 Forms for editing the data.....	53
4.3.4 Modifying a form frmMainReaders created	

in Laboratory work #3	55
4.4 Tasks for the laboratory work	56
4.5 Review and discussion questions	57
4.6 The contents of the report.....	57
Literature	57

1 LABORATORY WORK №1

CREATING A DATABASE IN RELATIONAL DATABASE MANAGEMENT SYSTEM (RDBMS) MICROSOFT ACCESS

1.1 Purpose of the work

Learning of the interface and main objects of a DBMS Microsoft Access.

1.2 Brief theoretical information

What is a Database?

At the most basic level, a database is simply an organized collection of data. A database management system (DBMS) such as Microsoft Access, Oracle or SQL Server provides you with the software tools you need to organize that data in a flexible manner. It includes facilities to add, modify or delete data from the database, ask questions (or queries) about the data stored in the database and produce reports summarizing selected contents.

Concrete steps of the process:

- determine the purpose of your database;
- determine the tables that you need in your database;
- determine the fields that you need in the tables;
- identify fields with unique values;
- define the relationships between tables;
- refine the design;
- create tables and insert data;
- create other system objects.

Determining the purpose of the database

Make sure that you do answer the following questions:

1. What type of data should the system keep track of?
2. What would the user want to know about the data?
3. What would the user want to do to the data?

Briefly: make a design plan.

Designing the tables you need

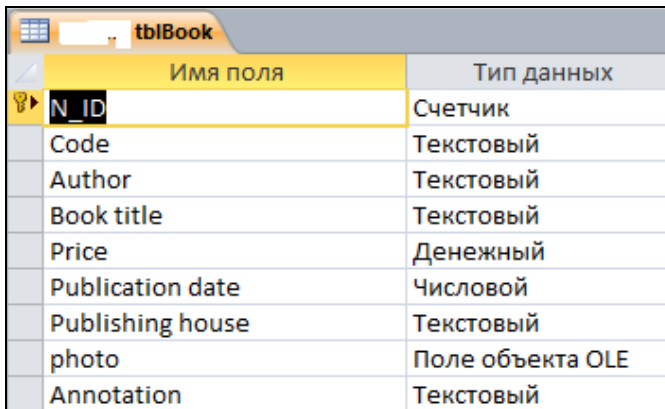
When you design your tables, divide up pieces of information by

keeping following fundamental design principles in mind:

1. Each piece of information is stored in only one table.
2. A table should not contain duplicate information.
3. Each table should contain information about only one subject.

Define the fields and identify a field per a table with unique values. Order of the fields is normally very natural - from main properties to minor (but also necessary) properties. One field contains a piece of information, such as Author, Name of the book, Publication date and so on

There must be at least one field (or a set of fields) having unique values for each record (fig. 1.1). Normally it's a code/ID.



Имя поля	Тип данных
N_ID	Счетчик
Code	Текстовый
Author	Текстовый
Book title	Текстовый
Price	Денежный
Publication date	Числовой
Publishing house	Текстовый
photo	Поле объекта OLE
Annotation	Текстовый

Figure 1.1 - The field list of tblBook table in Library. N_ID field - unique book number

Defining the relationships between the tables

Each table should be joined at least with one other table. If defining primary keys is made correctly the relationships will have the right type (1-to-1 OR 1-to-many). If a table can't be joined with any other table, it means that this table is unnecessary or a table is missing (between this and some other table).

Refining the design

The process of specifying and defining tables, keys, columns and relationships in order to create an efficient database is called normalization.

Normalization is a part of successful database design. Without normalization, database systems can be inaccurate, slow and inefficient and they might not produce the data you expect.

By normalising, you store data where it logically and uniquely belongs. When normalizing a database you should achieve four goals:

1. Arranging data into logical groups such that each group describes a part of the whole.
2. Minimizing the amount of duplicated data stored in a database.
3. Building a database in which you can access and manipulate the data quickly and efficiently without compromising the integrity of the data storage.
4. Organizing the data such that, when you modify it, you make the changes in only one place.

Creating tables and inserting records

First of all, you have to create and name a new database file. Then it is possible to create the tables and define their fields. Each table should have a primary key. Finally, define the relationships. Now you are ready to insert records (data entities) to the tables. You can use also forms (see the next lab) to enter records .

Creating other database objects

After creating the tables you can create other database objects. The most important objects are the following ones:

Query - An object to select, update, insert and delete records and to perform calculations with data. It bases on tables or (other) queries.

Form - An object to enter, update and view data, and to build a user interface. It bases on tables or queries.

Report - An object to summarize data. It normally bases on queries but also on tables.

Macro - An object to automate common tasks.

Module - A collection of Visual Basic for Applications declarations and procedures that are stored together as a unit.

Create DataBase in MS Access

When you open Microsoft Access, you will start in the Open window (fig. 1.2). From here, you can open Recent files, Open Other Files, or create a new database, either a blank database or one from a variety of templates. If you are already in Access, you can open a file by choosing the windows File tab, then select Open.

If you select the Blank desktop database template, you will be asked to give the database a file name and location. Notice that the extension on

database files is .accdb Click Create.

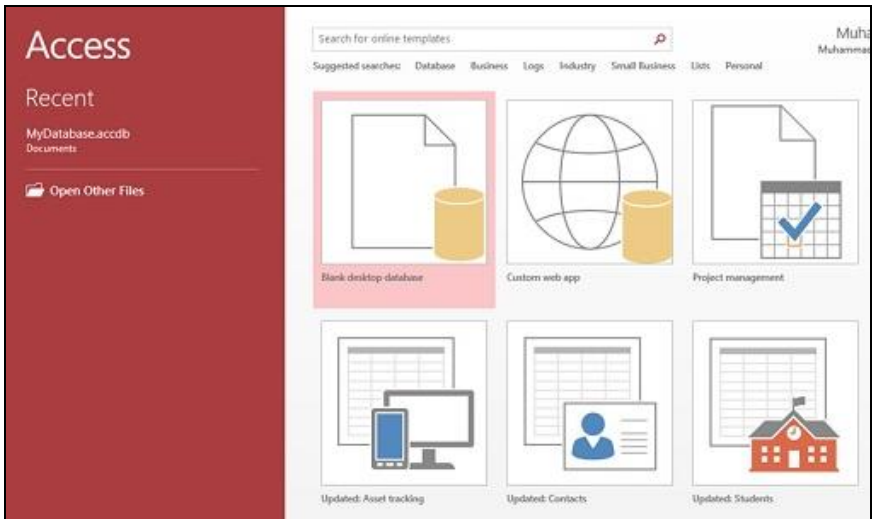


Figure 1.2 – Start window MS Access

Access will create a new blank database and will open up the table which is also completely blank (fig.1.3).

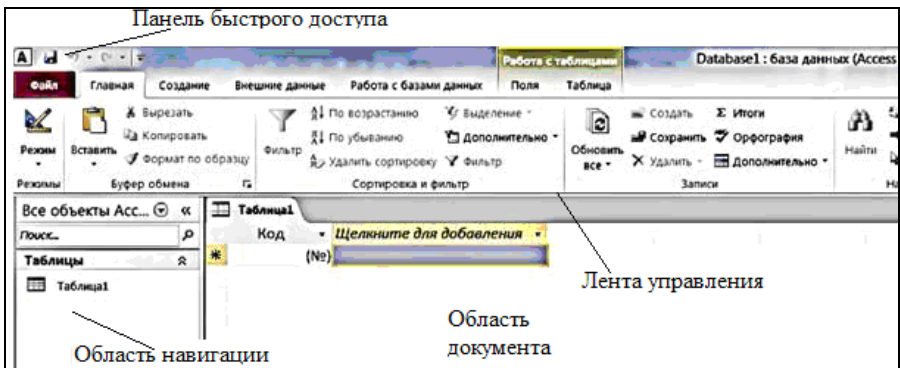


Figure 1.3 - Elements of MS Access interface

Once you have done this, the database will automatically save when you enter new information and when you exit the database.

1.3 The order of performance of the laboratory work

In the given work we start to study a *DBMS MSAccess* on an example of creation of the database "Library". A final cause of the first

three laboratory works is the creation of the application, which allows to fulfill the following operations:

- Support of the catalogs, i.e. input and update the data about the books, readers and books which are read;
- Search of the necessary books by the author, title and other criteria;
- Output of different lists of the readers, for example those, who work on EA chair etc.;
- Obtaining the information about the books, which are given to the concrete reader;
- Output of the list of the books with the overdue term of return to the library for each reader, with calculated fine on each book, and also with total sum of a fine;
- And so on.

Step 1. Creation of the DB Library

Open Microsoft Access. Click on the *Blank Database* icon. Access asks you to enter the new database. Type in the database name **Library** and press the *Create* button.

Step 2. Creation of the tables

Let's create three tables, which contain the information about operation of the library of a university:

- **tblBook** - contains the information about the books, which are kept in the library;
- **tblReadBook** - contains the information about the books, which are given to the readers;
- **tblReader** - contains the information about the readers of the library.

For creation of the table click on the *Table Design* button in the *Tables* group on the *CREATE* tab (fig.1.4)

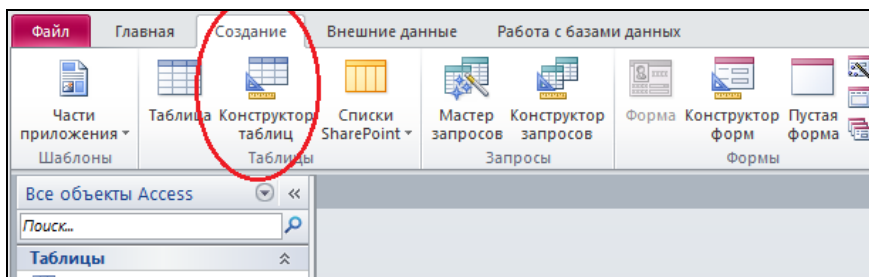


Figure 1.4 - For creation of the table using the *Table Design* button in the *Tables* group

on the **CREATE** tab

In the *Design View* window (fig. 1.5) it is necessary to set for each field its name, type and properties.

The data type is selected from the list, which is opened; the description allows to define more particularly the assignment of a field and the information feature, which is stored in it. To fill the column *Description* is not necessary.

In the fields properties :

Format - the size and type of data to enter in the field. By default system installs a size of a field equal to 50 positions for string data. For number data the most common settings are **Double** and **Long Integer**. If joining this field to **AutoNumber** field in a many-to-one relationship, this field must be **Long Integer**.

Caption – the label for the field when used on a form.

Имя поля	Тип данных
N_ID	Счетчик
Code	Текстовый
Author	Текстовый
Book title	Текстовый
Price	Денежный
Publication date	Числовой
Publishing house	Текстовый
photo	Поле объекта OLE
Annotation	Текстовый

Свойства поля

Общие	
Размер поля	Длинное целое
Новые значения	Последовательные
Формат поля	
Подпись	
Индексированное поле	Да (Совпадения не допускаются)
Смарт-теги	
Выравнивание текста	Общее

Figure 1.5 – The window “Design View”

Validation rule (*Условие на значение*) - an logical expression that

limits the values that can be entered in the field.

Validation text (Сообщение об ошибке) - contains the message to the user at input of error values.

Indexed – an index speeds up searches and sorting on the field. Selecting “Yes-No duplicates” prohibits duplicates values in the field.

One of fields of the table is normally assigned as *key*. Values in this field are uniquely determinate a record. This field should be assigned as *required* and it is necessary to add, that this field is *indexed*. To assign this field as key, it is necessary to select this field and to press the button "Key" on the toolbar.

Create the table **tblBook**

The titles of fields and data about 3 books are shown in the table 1.1. Enter these data and information about other 7 books into the table. A key field in the table **tblBook** are field **N_ID**

Table 1.1 - The names of fields and data about 3 books

№ record	Name of the field	Data
1	N_ID	1
	Code	681.3.06/Г32
	Author	Гарнаев А.Ю.
	Book title	Самоучитель VBA
	Town	Санкт-Петербург
	Publishing house	БХВ
	Publication date	1999
	Price	19
	Annotation	
2	N_ID	2
	Code	618.3/К54
	Author	Камарада, Билл
	Book title	Использование MS Word 97: Пер. с англ.
	Town	Москва
	Publishing house	Издательский дом "Вильямс"
	Publication date	1998
	Price	36
	Annotation	

№ record	Name of the field	Data
3	N_ID	3
	Code	618.3/X21
	Author	Дж. Хоникатт и др.
	Book title	Использование Internet: Пер. с англ.
	Town	Москва
	Publishing house	Издательский дом "Вильямс"
	Publication date	1998
	Price	24
	Annotation	

Create the table "tblReader" with such fields:

Card - number of the reader's ticket;

Reader- surname of the reader;

Chair - chair, on which the reader works, or the group, in which studies the reader;

Telephone - office number of the reader.

Key field in this table is the field **Card**, because it uniquely determinates each record. This field should be indexed and required. Fill in this table by data.

Create the table "tblReadBook" with such fields:

Inventory number - accession number of the book given to the reader;

Card - number of the reader's ticket;

Date1 - date, when the reader took the book from the library;

Date2 - date, when the reader should return the book to the library.

Select types and sizes of fields, which correspond to a contents. The fields **N_ID** and **Card** should be indexed and required. In this table it is possible to not assign a key field.

You must allow not producing the book to the nonexistent readers, since in this case you can not set necessary links between the tables.

You must allow that the data in all three tables were compounded, i.e. to not give out which books is not present in library and to not give out books to those readers who are not registered in table **tblReader**.

Step 3. Creation of link between the tables

Click the *Relationships* button on the *Database* toolbar . The *Show Table* property sheet comes up. From here, you will specify what tables (or queries) will be used when building your relationship (s). Arrange the layout of your relationship window so that the **tblReadBook** table be in the middle of the **tblBook** table on its left and the **tblReader** table on its right. The originating table uses its *Primary Key* and associates it to the field you choose in the target table. The target field is referred to as the *Foreign Key*.

The link between them is set by the help of the mouse by a method «drag-and-drop ». Select a field **N_ID** in the table **tblBook** and drag it up to the same field in the table **tblReadBook**. The *Edit RelationSips* dialog box is display. Check the field names displaed in the two columns to ensure they are correct. You can change them is necessary. Set the relationship options: *Enforce Referential Integrity* (Обеспечение целостности данных), *Cascade Update Related Fields*, *Cascade Delete Related Records* if necessary (fig.1.6).

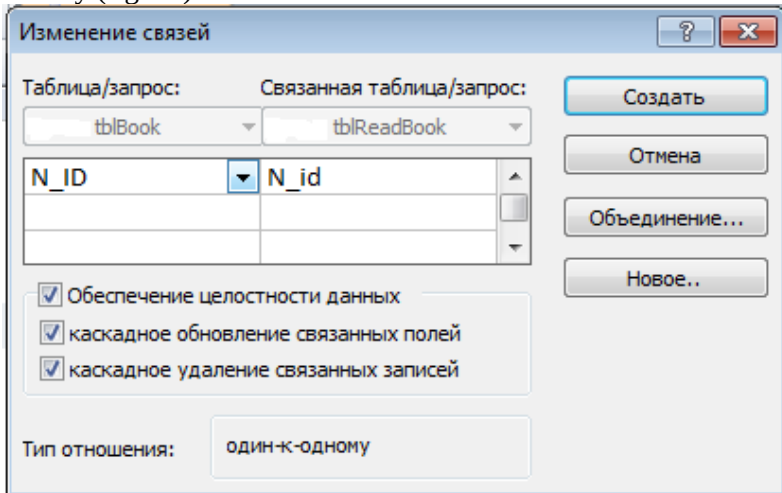


Figure 1.6 - The *Edit RelationSips* dialog box

On the scheme the line which joins these fields will appear. Similarly install link between two other tables by a field **Card**. The example of a *Relationships* window is represented on fig. 1.7 .

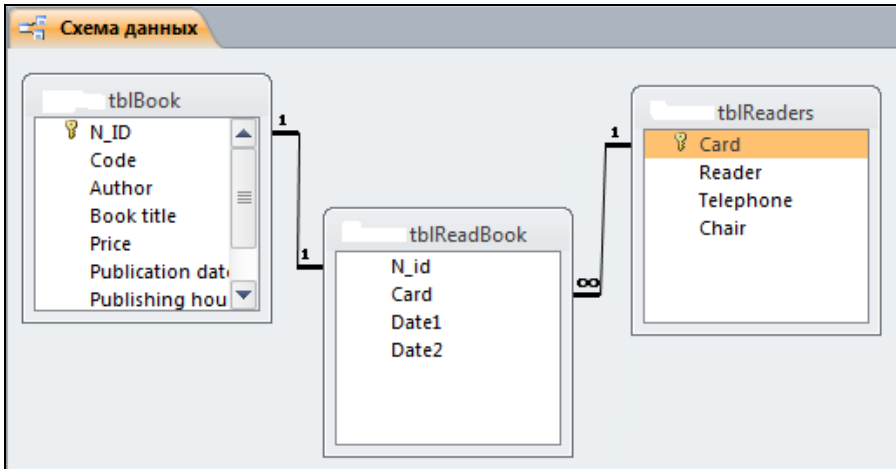


Figure 1.7 – The Relationships window

Step 4. Creation of the forms for review and data input

Go to a tab *Forms* and create the simple forms in the mode *AutoForm* for each of the tables. Try some different types of the forms (*Columnar*, *Tabular*, *Datasheet*) and select for yourself most convenient one. Enter new data into the tables, using the forms.

Step 5. Operation with the tables in the browse mode

Open one of three tables of a DB in the browse mode and in practice take possession of the following tools:

- *Navigating under the table* – using arrow keys, keyboard shortcut (Ctrl-Pgup, Ctrl-Pgdown, Ctrl-Home, Ctrl-End), by the help of a mouse, using if necessary horizontal and vertical scrollbars. For moving on records it is used *the navigation panel* which is located at the left bottom of the window of the table;
- *Input and editing of data*;
- *deleting of rows and columns*;
- *search and replacement of data*;
- *sorting*;
- *concealment and mapping of columns*;
- *fixing of columns*;
- *substitution*;
- *usage of filters*.

1.4 Review and discussion questions

1. Define main concepts: *Database, Table, Field, Record*.
2. Creating a *Table* in *Design View*.
3. Understanding *Field Properties*.
4. Indexing a field.
5. Purpose a *Primary Key* and a *Foreign Key*.
6. Concept of a filter. Creation and usage of filters.
7. Understanding *Table Relationships*.
8. Understanding *Relationship Types*.
9. Purpose the relationship option: Enforce *Referential Integrity*.
10. What different types of the *Forms* do you know?

1.5 The contents of the report

1. Subject and purpose of work.
2. Protocol of work.
3. Conclusions.

2 LABORATORY WORK №2 MS ACCESS QUERIES

2.1 Purpose of the work

Create variety of queries that analyze and manipulate database information.

2.2 Brief theoretical information

MSAccess contains different types of queries: selection queries, parameter queries, crosstab queries, action queries (make-table, delete, update, append queries), SQL queries.

A query is a request you present to the database, and the database displays its response to you.. You use queries to view, change, and analyze data in different ways. You can also use them as the source of records for forms, reports..

We shall use the Library database. With MS Access you can create a query in multiple ways; we will examine the use of the *Query Designer* (fig.2.1).

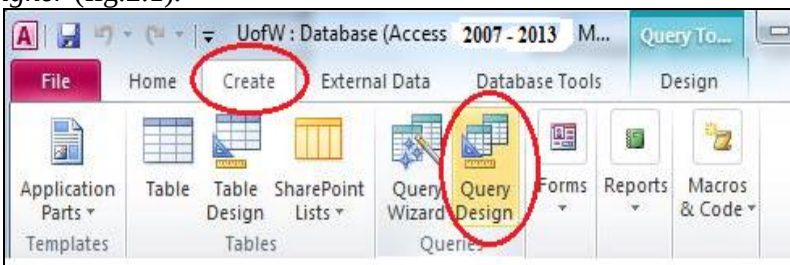


Figure 2.1 - To create a query, click the *Create* tab and then click the *Query Design* icon

As a result MS Access opens a Query By Example (QBE) window that you use to specify components of a query (fig 2.2).

This window comprises two areas: *Relationships* and *Grid*. The Relationships area will show each table that needs to be accessed and the relationships to be used with those tables.

The Grid area is used to specify:

- The *Field* row of the design grid contains the names of the fields actually included in the query.

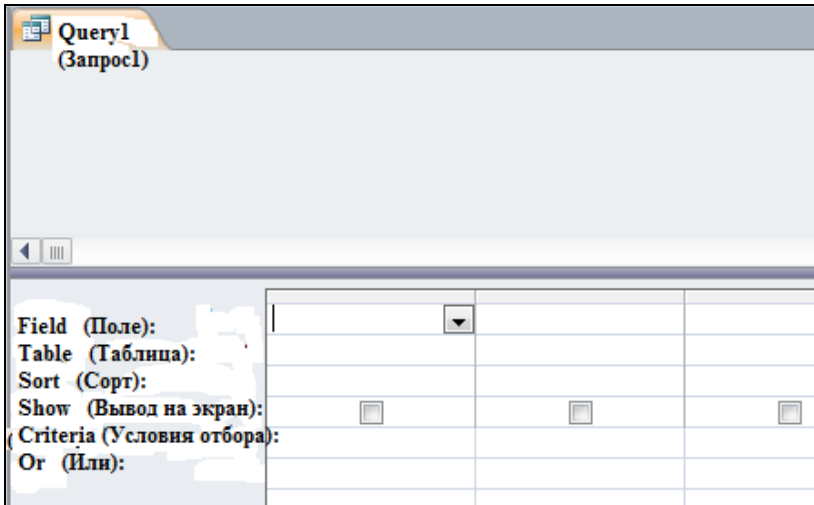


Figure 2.2 - QBE window

- *The Table row* shows which table each field belongs to.
- *The Sort row* indicates which field(s) the query results will be sorted on, if any.
- *A selected check box* in the Show row means that the field will be displayed in the results datasheet. (If the check box isn't selected, the field can be used in determining the query results, but it won't be displayed.)
- *The Criteria row* can contain criteria that determine which records will be displayed.
- *The Or row* sets up alternate criteria.

2.2.1 Simple Query

The simplest query is one that displays a complete table – all rows and columns.

Example 1

Suppose we want to list all books in the library. The process of creating the query is as follows:

- Click on the *Create* tab if necessary and then click on the *Query Design* icon. Now you can right-click in the *Relationships* area and choose the *Show Table* option
- A window pops up, and from the list of tables you must double-

click **tblBook**(fig. 2.3).

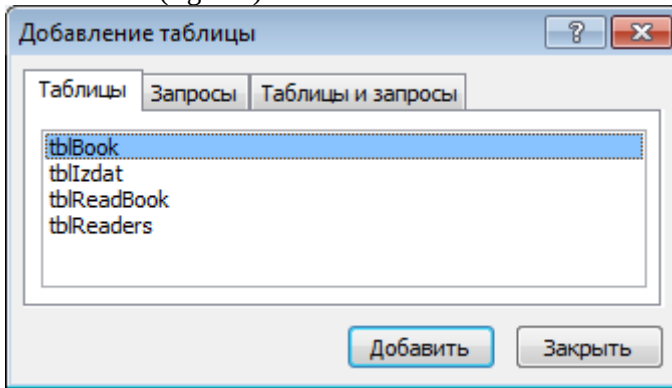


Figure 2.3 - The QBE List of table

– Choose *Close* from the *Show Table* pop up window.

MS Access displays the **tblBook** table and its fields in the *Relationships* area. The first in this list is an * which stands for *all attributes* – double-click the *. This results in the following (fig. 2.4):

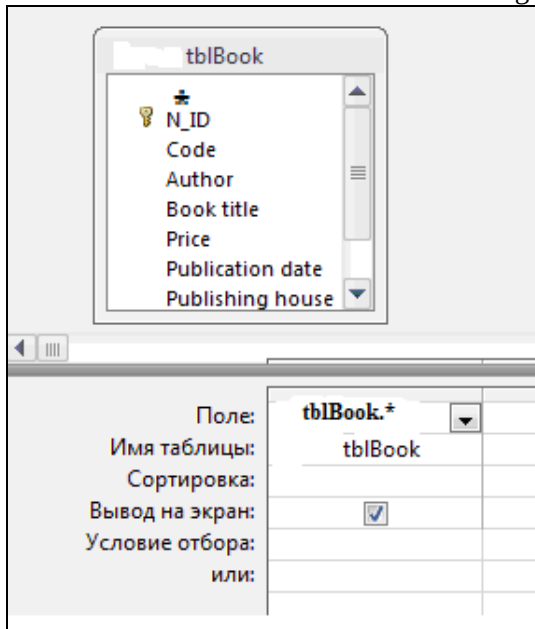


Figure 2.4 - Choosing all fields of the table

We can run the query to test it and confirm it does what we expect:

list all rows in **tblBook**. To run a query, click the **Run** icon(fig 2.5).



Figure 2.5 - Run a query

There are other views of a query. If you click the drop down just below the View icon (fig.2.6):

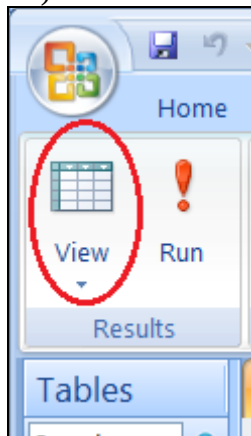


Figure 2.6 - Several views for a query

You can see all the ways of viewing a query, including:

- *Datasheet View*,
- *Design View*,
- *SQL View*.

You can also run a query by choosing *Datasheet View*. When developing a query one often alternates between *Datasheet View* and *Design View* in order to get the query working as required. When you run a query, MS Access will retrieve the information requested. In this case the

results of running the query are:

You can also run a query by choosing Datasheet View. When developing a query one often alternates between Datasheet View and Design View in order to get the query working as required. When you run a query, MS Access will retrieve the information requested. In this case the results of running the query are (fig. 2.7):

tblBook			
N_ID	Code	Author	Book title
17	QA 76.9	David Hay	Data model patterns : conventions of thought
14	QA 76.73	Joe Celko	Joe Celko's SQL puzzles & answers
16	QA 76.9	T A Halpin	Information modeling and relational databases
15	QA 76.76	Zigurd R Me	Programming Android
11	11.A	Акунин Б.	Турецкий гамбит
12	10.A	Акунин Б.	Левиафан1
1	10.A	Арбузов А.	Бедный Марат
4	12	Битов А.	Пушкинский дом
2	681.Г	Гарнаев А.	VBA
3	681.Г	Гарнаев А.	Visual Basic 6.0

Figure 2.7 - Query results

You can save the query and now you can see the query listed as a database object. The query can be run any time by an end user. The results of the query are not stored or saved – only the definition of the query. Whenever a user runs the query the current contents of the Book table are accessed.

2.2.2 Selection Query

Use expressions as query criteria

You use criteria in a query to narrow down query results. You enter the criteria as an expression, and Access returns only those rows that match the expression. To create a conditional expression in the *Criteria row*:

1. Click in the *Criteria cell* in the column for which you want to enter your criteria.
2. To manually create your expression, type your criteria expression. Do not precede the criteria expression with the = operator.
3. To create your expression by using the *Expression Builder*, on the ribbon, click *Design*, and then in the *Query Setup group*, click *Builder* .

4. If you want a larger area in which to edit the expression, place the cursor in the *Criteria cell* and then press *SHIFT+F2* to display the Zoom box.

Conditional expressions consist of operators. The following details the five types of operators and how you use them.

Arithmetic operators

They calculate the value of at least two numbers or change a number to either positive or negative. The following details all of the arithmetic operators:

- + – Addition,
- – Subtraction,
- * – Multiplication,
- / – Division,
- \ – Round to the nearest integer, divide, then truncate to an integer,
- ^ – Exponent,
- Mod – Divide, and then show only the remainder.

Comparison operators are perhaps the most common for databases as the primary purpose of a database is to review and analyze data. The following are the comparison operators, and the result indicates the relation of the first value to the other data. For example, < indicates that the first value is less than the second value in the comparison.

- < – Less than
- <= – Less than or equal to
- > – Greater than
- >= – Greater than or equal to
- = – Equal to
- <> – Not equal to

Null – Either the first or second value is null because comparisons cannot include unknown values.

Concatenation operators combine text values into a single value.

- & – Creates one string from two strings
- + – Creates one string from two strings, including a null value when one of the strings is null

Special operators result in a True or False response.

Is Null/Is Not Null – Analyzes if a value is Null

Like – Finds string values matching the entry after Like; wildcards help widen the search. The wildcard characters are in the table 2.1.:

Table 2.1 – Wildcard characters

Wildcard Character	Matching criteria	Example
*	Matches any number of characters	Like “l*” matches all text strings that start with “l”
?	Matches any single character	Like “a?c” matches “aac”, “abc”, “acc”, etc. but does not match longer strings such as “aacc” or “xabc”
#	Matches any single numeric character	Like “b#b” would match “b2b” and “b7b” but not “bam”
[]	Matches any single character within the brackets	Like “j[ai]m” matches “jim” and “jam” but not “jaim”
!	Used with [] when you do not want to match any of the enclosed characters	Like “b[!ao]b” matches “bim” and “bub” but not “bam” or “bob”
-	Used with [] to specify a range of matching characters (given in ascending sequence)	Like “b[0-9]b” would match to “b2b” but not to “bam” Like “b[a-c]b” would match “bab”, “bbb”, and “bcb”

Between – Compares values to the specified range after Between:

Between 1900 AND 20017

Between #1.01.2018# AND #20.01.2018#

In(...) – The IN operator can be used if you need to determine if a field value is in specific list of values. The list of values is a comma-separated list enclosed in parentheses.

Example 2. To list books published by the "Найка", "Эксмо" or "БХВ - Петербург" you need a list of operate In() (fig. 2.8):

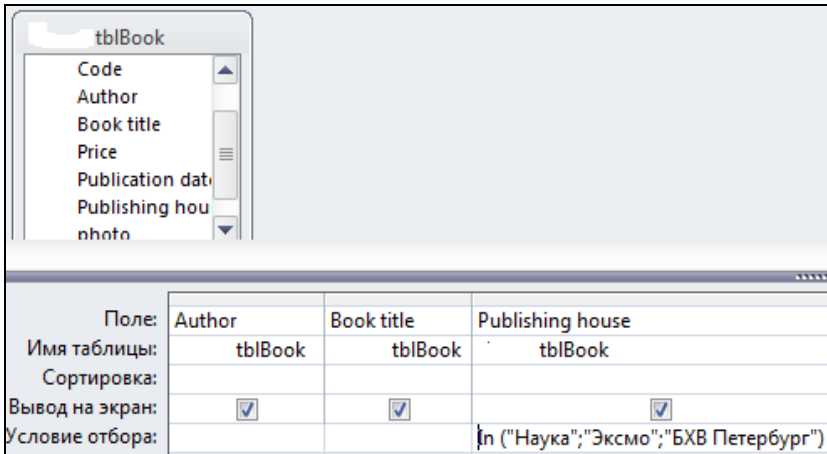


Figure 2.8 – Using IN()

Logical expression

Sometimes we need to retrieve data based on multiple criteria which are expressed as logical expressions involving the logical operators and, or, and not. MS Access provides a way for us to specify multiple criteria using the *Criteria* and *Or lines* in the *Grid*. We will consider each of the operators *And*, *Or*, and *Not*. We will consider each of the operators *And*, *Or*, and *Not*.

AND (fig. 2.9)

Suppose we want to list Celko's books on SQL. In this case there are two criteria a book must meet:

- criteria 1: the author's name must end with "Celko",
- criteria 2: "SQL" must appear in the title.

Criteria 1 must be true *and* criteria 2 must be true – we say the two criteria are *anded*. When using QBE we must place these two criteria on the same criteria line in order that MS Access finds rows that match both criteria.

Example 2. In this example we are looking for titles that have the text SQL anywhere within the title. MS Access provides a way for us to define such a pattern. The character * when used in a text string is a wildcard character that matches any number (zero or more) of characters. For criteria 1 we need two wildcards and so we specify the pattern that title must match: *Like* **"*SQL*"**.

For criteria 2 we specify the pattern that author must match: *Like*

“*Celko”.

The screenshot shows a query builder window for the **tblBook** table. The field list on the left includes **N_ID** (primary key), **Code**, **Author**, **Book title**, **Price**, and **Publication date**. The criteria grid below is as follows:

Поле:	Author	Book title
Имя таблицы:	Копия tblBook	Копия tblBook
Сортировка:	по возрастанию	
Вывод на экран:	☒	☒
Условие отбора:	Like "*Celko"	Like "*SQL*"
или:		

Figure 2.9 - Criteria on one criteria line are ANDed

OR (fig. 2.10)

Instead of books with titles containing “SQL” and authored by Celko, suppose the end user wants a list of books with “SQL” in the title or where Celko is the author. In this situation we place the criteria on separate lines.

The screenshot shows the same query builder window for the **tblBook** table. The criteria grid below is as follows:

Поле:	Author	Book title
Имя таблицы:	Копия tblBook	Копия tblBook
Сортировка:	по возрастанию	
Вывод на экран:	☒	☒
Условие отбора:	Like "*Celko"	
или:		Like "*SQL*"

Figure 2.10 - Criteria on different criteria lines are ORed

MS Access *ORs* the criteria; a row is selected for the result set if either or both of the criteria are true for a row.

If one specifies multiple criteria on both the Criteria and Or lines the criteria on each line is *ANDed*, and those evaluations are then *ORed*. If for some row either one or both of the sub-expressions evaluate to true, then the row will be selected for display.

Suppose we need a list of Celko's books on SQL and a list of Garnaev's books on VBA. Logically we can express this as:

(Author="Celko" AND Book title="SQL")

OR

(Author="Гарнаев" AND Book title="VBA")

We code this in QBE as on figure 2.11.

Поле:	Author	Book title
Имя таблицы:	Копия tblBook	Копия tblBook
Сортировка:	по возрастанию	
Вывод на экран:	☒	☒
Условие отбора:	Like "*Celko"	Like "*SQL*"
или:	Like "*Гарнаев"	Like "*VBA*"

Figure 2.11 - Two expressions that are *ORed*

NOT

To get a list of all Celko's books, besides books on SQL one could use the criteria: NOT "VBA" which, written in long form is :

(Author="Celko" AND (NOT (Book title="SQL")))

Coding this in QBE we have on figure 2.12.

tblBook		
* N_ID Code Author Book title Price Publication date		
Поле:	Author	Book title
Имя таблицы:	Копия tblBook	Копия tblBook
Сортировка:	по возрастанию	
Вывод на экран:	☒	☒
Условие отбора:	Like "*Celko"	Not Like "*SQL*"

Figure 2.12 - Using NOT

Example 3.

Suppose we want to list all books published in the period from 2010 to 2015 by Наука. Publishers and books published after 2010 by Наукв Publishers, or Эксмо, or ВХВПетербург. Selected information must be sorted on *Authors* and *Title Book* fields.

Coding this in QBE we have on figure 2.13

tblBook				
Code Author Book title Price				
Поле:	Author	Book title	Publication date	Publishing house
Имя таблицы:	tblBook	tblBook	tblBook	tblBook
Сортировка:				
Вывод на экран:	☒	☒	☒	☒
Условие отбора:			Between 2019 And 2015	"Наука"
или:			> 2015	"Наука"
			> 2015	"Эксмо"
			> 2015	"ВХВ Петербург"

Figure 2.13 – Design window of a query for example 3

2.2.3 Multi-table queries

When you add tables to a query, Access creates *inner joins* that are based on relationships that have been defined between the tables. You can manually create joins in queries, even if they do not represent relationships that have already been defined. If you use other queries (instead of or in addition to tables) as sources of data for a query, you can create joins between the source queries, and also between those queries and any tables that you use as sources of data.

How you know, there are three basic types of joins: inner joins, left joins, right joins. You must know each type of join you can use, why you use each type, and how to create the joins.

Joins are to queries what relationships are to tables: an indication of how data in two sources can be combined based on data values they have in common. Here's an illustration of a join in query Design View, with the join's properties open in a dialog box:

The screenshot shows a query Design View with two tables: **tblReaders** and **tblReadBook**. **tblReaders** has fields: Card (primary key), Reader, Telephone, and Chair. **tblReadBook** has fields: N_id, Card, Date1, and Date2. A 1:∞ join line connects the **Card** field of **tblReaders** to the **Card** field of **tblReadBook**.

Поле:	tblReaders.*	Card	Date1
Имя таблицы:	tblReaders	tblReadBook	tblReadBook
Сортировка:			
Вывод на экран:	☒	☒	☒
Условие отбора:			

Параметры объединения

Левая таблица	tblReaders	Правая таблица	tblReadBook
Левый столбец	Card	Правый столбец	Card

☒ 1: Объединение только тех записей, в которых связанные поля обеих таблиц совпадают.
☐ 2: Объединение ВСЕХ записей из "tblReaders" и только тех записей из "tblReadBook", в которых связанные поля совпадают.
☐ 3: Объединение ВСЕХ записей из "tblReadBook" и только тех записей из

Figure 2.14 - Illustration of a join in query Design View, with the join's properties open in a dialog box

Inner joins: only related data from both tables combined

An inner join is one in which Access only includes data from a table if there is corresponding data in the related table, and vice versa. Most of the time, you will use inner joins. When you create a join and don't specify what kind of join it is, Access assumes you want an inner join. Inner joins are useful because they let you combine data from two sources based on shared values – so you only see data when there's a complete picture.

Outer joins: all the related data combined correctly, plus all the remaining records from one table

An outer join is like an inner join, but adds the remaining rows from one of the tables. Outer joins are directional: a left outer join includes all the records from the left table – the first table in the join – and a right outer join includes all the records from the right table – the second table in the join.

2.2.4 Create a calculated field in a query

Suppose you're designing a query, and you want to display the results of a calculation using other fields in the query. To create the calculated field, you enter an expression in a blank cell in the Field row in your query. For example, if you have a query that contains a **Quantity** field and a **Unit Price** field, you can multiply the two to create a calculated field for **Extended Price** by entering the following expression in the Field row of the query:

Extended Price: [Quantity] * [Unit Price]

Prefacing the expression with the text **Extended Price:** names the new column **Extended Price**. This name is often called an alias. If you do not supply an alias, Access will create one, such as **Expr1**.

Procedure:

1. In the *Navigation Pane*, right-click the query that you want to change, and then click *Design View* on the shortcut menu.
2. Click the *Field cell* in the column where you want to create the calculated field.
3. To manually create your expression, type your expression.

Do not precede the criteria expression with the = operator; instead, begin the expression with a descriptive label followed by a colon. For example, type **Extended Price:** to provide the label for an expression that creates a calculated field called **Extended Price**. Then, enter the criteria

for your expression after the colon.

4. To create your expression by using the *Expression Builder*, on the ribbon, click *Design*, and then in the *Query Setup* group, click *Builder*.

2.3 The order of realization

In query Design View create the following queries.

1. Show the list of readers that work, for example, at department of electric devices.
2. Show the list of readers which work, on faculty electric devices and which read books today.
3. Show the list of readers which don't read books today.
4. Select readers and books that have been received in library by those readers (query to three tables);
5. Create the query **quList1** for selection of list of books that have been received in library.

The list must include information about a fine that reader has to pay in case of delay. This query will be used in the next laboratory work as a source of data for subforms creation.

In order to create query **quList1** pass on to the supplementary sheet *Queries* and press *Create*. Add tables **tblBook** and **tblReadBook** to the query. The relationship between tables on the field **Inventory number** must be shown in the window of data scheme. Move all fields you want to be included into query (for example **Author, Name of the book, Price, Inventory number, Date1, Date2, Card**) to the lower part of the window. The last field will be necessary to connect the query with the table **tblReaders**. Create a calculated field with the name **Fine** in a first free column of the lower part of the window. For that type in an upper row (where the name of field is situated) the following text:

Fine:`if([Date2]<Date();DateDiff("d";[Date2];Date())*0,01*[Price];0)`

Fields names are written in square brackets.

The query creation is finished. Save it as **quList1**. Make sure that everything is done correctly, look through your query using button *Open*.

2.4 Review and discussion questions

1. Purpose of the object Query.
2. Creating a Query in Design View.

3. The types of Queries.
4. Creating a calculated fields.
5. Working with Expressions and the Expression Builder.
6. Using IIF Function.

2.5 The contents of the report

1. Subject and purpose of work.
2. Protocol of work.
3. Conclusions.

3 LABORATORY WORK №3

WORKING WITH FORMS AND REPORTS

3.1 Purpose of the work

Creating Forms and Reports with *Wizard*

3.2 Brief theoretical information

A form is an organized and formatted view of some or all of the fields from one or more tables. Forms work interactively with the tables in a database. You use controls in the form to enter new information, to edit or remove existing information, or to locate information. The controls you will use most frequently in an Access form are as follows:

- *Text box controls*. You can view or enter information in these controls. Think of a text box control as a little window through which you can insert data into the corresponding field of the related table or view information that is already in that field.

- *Label controls*. You can tell what type of information you are looking at in the corresponding text box control, or what you are expected to enter in the text box control by using these controls.

An Access form can also include a variety of other controls that transform the form into something very much like a Windows dialog box or wizard page.

Like forms, reports give people easy access to the information stored in a database. However, there are several differences between forms and reports, including the following:

- Forms are used to view, enter, and edit information. Reports are used only to view information.

- Forms are usually displayed on the screen. Reports can be previewed on the screen, but they are usually printed.

- Forms generally provide a detailed look at records, and they are usually for people who actually work with the database. Reports are often used to group and summarize data, and they are often for people who don't work with the database but who use the information stored in the database for other tasks.

There are several ways of new form or new report creation (as well as of any database object). In this laboratory work we will study means of forms and reports creation by using *Wizard*.

3.2.1 Creating forms by using the Forms Tools

In a Microsoft Access database, you can choose to build a combination of *Basic Forms*, *Multiple Item Forms*, *Datasheet Forms*, *Split Forms*, or *Subforms*.

- *Basic Forms* display one record of data on a form at a time.
- *Multiple Item Forms* display multiple records of data at the same time. These are also referred to as Continuous Forms.
- *Datasheet Forms* display the data records in a datasheet view making the form look and function exactly like a table.
- *Split Forms* are like a combination of a Basic Form and a Multiple Item Form all wrapped into one big form. On a Split Form, you can view a single record in its entirety and also view multiple detailed records of data at the same time. Split Forms are so named because you can view and interact with the data in two ways simultaneously.

Access gives you three main ways to create a form: with a single mouse click, with the *Form Wizard*, or in *Design View*.

Creating a form with a single mouse click

There are three types of forms that can be created with a single mouse click:

- Basic (Simple) Form,
- Split Form,
- Multiple Items Form.

You can begin using the new form immediately, or you can modify it in Layout view or Design view to better suit your needs.

Example 1. To create a form with a single click.

1. Open the table **tblBook** upon which you want to base the form.
2. To create a form **frmBook1** on which all fields from the underlying table or query are placed, displaying one record at a time, on the *Create* tab, click *Form* (fig. 3.1).

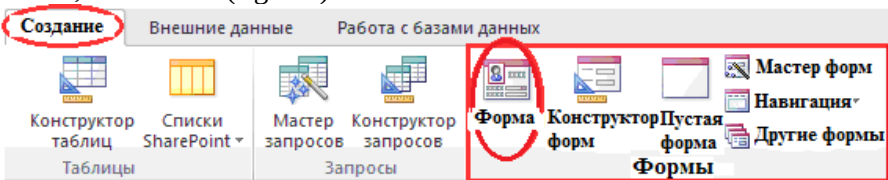


Figure 3.1 – Using Form Tool

Your form will be created and opened in *Layout view* (fig. 3.2).

N_ID	14
Code	681.C 76.73
Author	Joe Celko
Book title	Joe Celko's SQL puzzles & answers
Price	295,00
Publication date	2014
Publishing house	Эксмо
photo	
Annotation	

Figure 3.2 - Your form will be created and opened in *Layout view*

3. To create a split (разделенная) form **frmBook2** on the Create tab, Click *More Forms*, and then *Split Form* (fig.3.3). The two views and connected to the same data source and synchronized with other at all times. Selecting a field in one part of the form select the same field in the other part of the form. You can add, edit and delete data from either part (fig. 3.4).

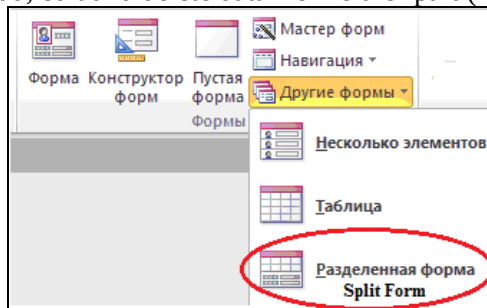


Figure 3.3 – Using Split Form

N_ID	Code	Author	Book title
1	10.A	Арбузов А.	Бедный Марат
2	681.Г	Гарнаев А.	VBA
3	681.Г	Гарнаев А.	Visual Basic 6.0
4	10.Б	Битов А.	Пушкинский дом
5	681.Т	Турчак	Численные методы
6	10.Ч	Чейз	Мышеловка
7	11.Ч	Чехов А.	Пьесы

Figure 3.4 – Split form in Layout

4. To create a form on which all fields from the underlying table or query are placed, displaying multiple records at a time, on the *Create tab*, click *More Forms*, then *Multiple Items* (fig.3.5). The form that Access creates resembles a datasheet. The data is arranged in rows and columns, and you see more than one record at a time (fig.3.6). However, a Multiple Items form give you more customization options than a datasheet, such as the ability to add graphical elements, buttons and other controls.

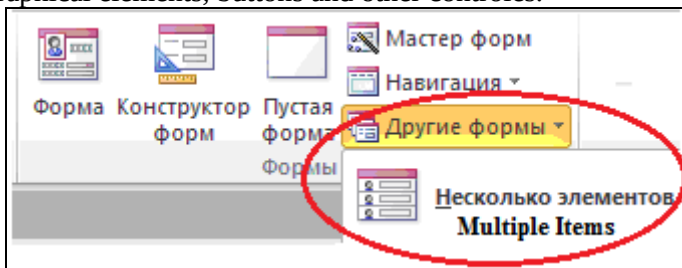


Figure 3.5 – Using Multiple Items

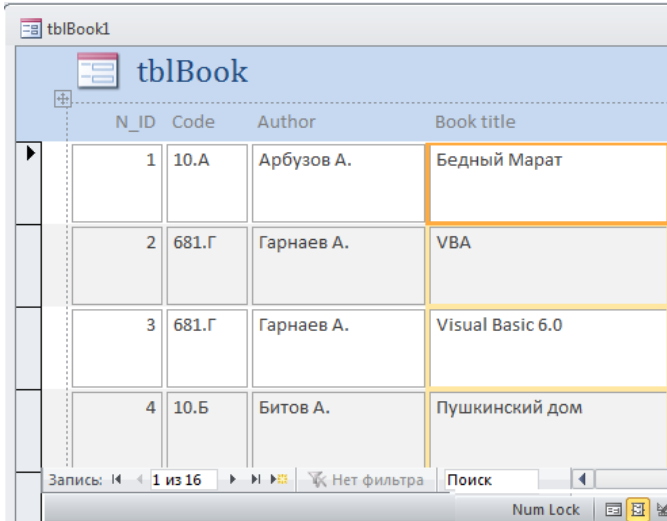


Figure 3.6 Form created with Multiple Items

Creating a form with the Form Wizard

To be more selective about which fields appear on your form, you can use the Form Wizard. You can also use fields from more than one table or query, provided specified the relationships between the tables and queries beforehand. You will need to tell the Wizard:

- The table or query on which to base the form.
- Which fields to use on the form.
- Which form layout to apply.
- Which visual style to apply.

To use the Form Wizard:

1. On the Create tab click Form Wizard (fig.3.7).

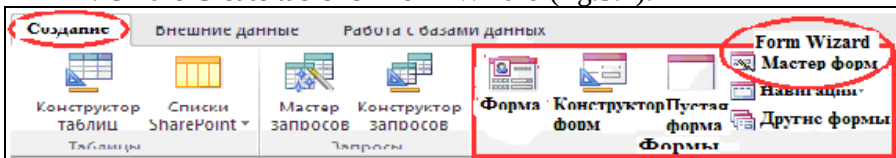


Figure 3.7 – Using Form Wizard

2. Follow the onscreen steps.
3. Additional customization can be done in *Design View*.

3.2.2 Formatting forms

When you create a form by using the Form tool the form includes every or some fields in the table on which it is based. Each field is represented on the form by a *text box control* and its associated *label control*. The form is linked, or bound, to the table, and each text box is bound to its corresponding field. The table is called the record source, and the field is called the control source.

Forms and their controls have properties that determine how they behave and look. A form inherits some of its properties from the table on which it is based. For example, each text box name on the form reflects the corresponding field name in the source table. The text box label also reflects the field name, unless the field has been assigned a Caption property, in which case it reflects the caption. The width of each text box is determined by the Field Size property in the table.

Even though a form is bound to its table, the properties of the form are not bound to the table's properties. After you have created the form, you can change the properties of the form's fields independently of those in the table. You might want to change these properties to improve the form's appearance—for example, you can change the font, font size, alignment, fill color, and border.

Example 2. In this exercise, you'll first apply a theme to a form **frmBook1** and add a logo to the form's title. Then you'll change the form properties that control its colors and text attributes.

1. In the *Navigation pane*, in the *Forms group*, right-click **frmBook1**, and then click *Layout View*.
2. On the *Design tool tab*, in the *Themes group*, click the *Themes button* to display (fig. 3.8) a menu containing the *Themes gallery*(fig 3.9).

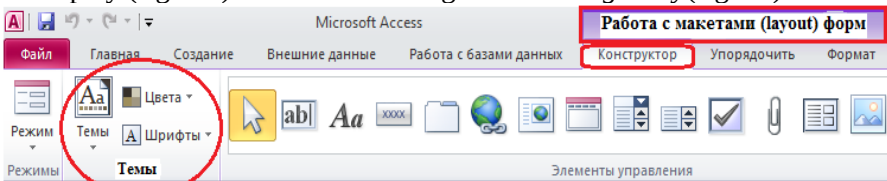


Figure 3.8 - Themes group

3. Point to each thumbnail in turn, pausing until its name appears in a *ScreenTip*.
4. Click on the selected thumbnail to apply that theme.
5. Now let's replace the form icon to the left of the title in the form header

with a logo. On the *Design* tab, in the *Header/Footer* group, click the *Logo* button.

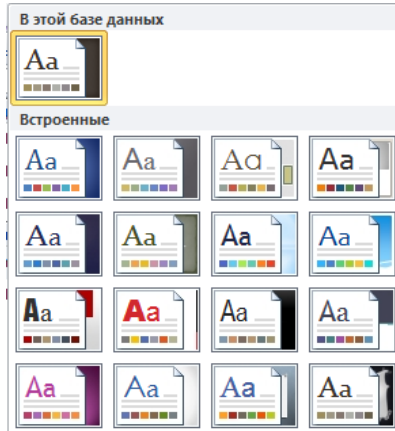


Figure 3.9 - Themes gallery

6. In the Insert Picture dialog box, navigate to the your file picture folder, and double-click the selected Logo picture. You have inserted a custom logo in the form header.

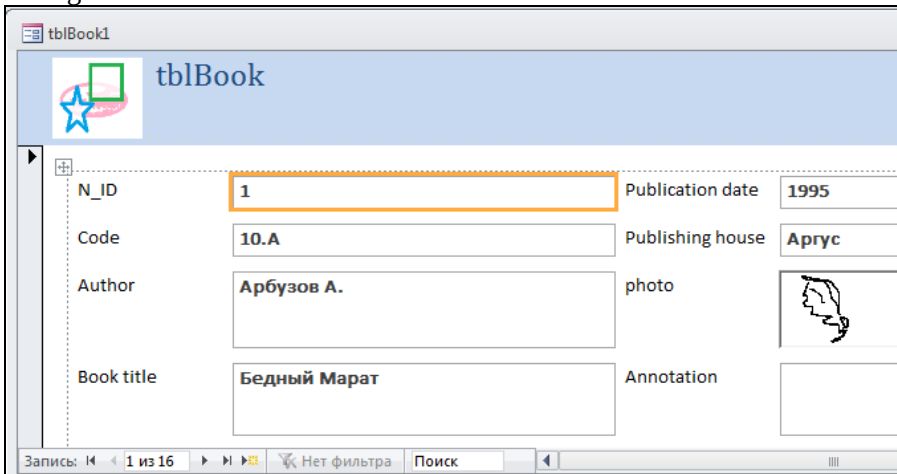


Figure 3.10 - On the **frmBook1** form, click the *Code* label control (not its text box) to select it for manipulation

Next let's experiment with properties.

7. On the **frmBook1** form, click the *Code* label control (not its text box) to select it for manipulation (fig. 3.10).

8. On the *Format tool tab*, in the *Font group*, click the *Font Size arrow*, and then in the list, click 16 to make the label text significantly larger.
9. Click the **Code text box** control (not its label), and then on the *Design tool tab*, in the *Tools group*, click the *Property Sheet button* to open the *Property Sheet pane* (fig.3.11).

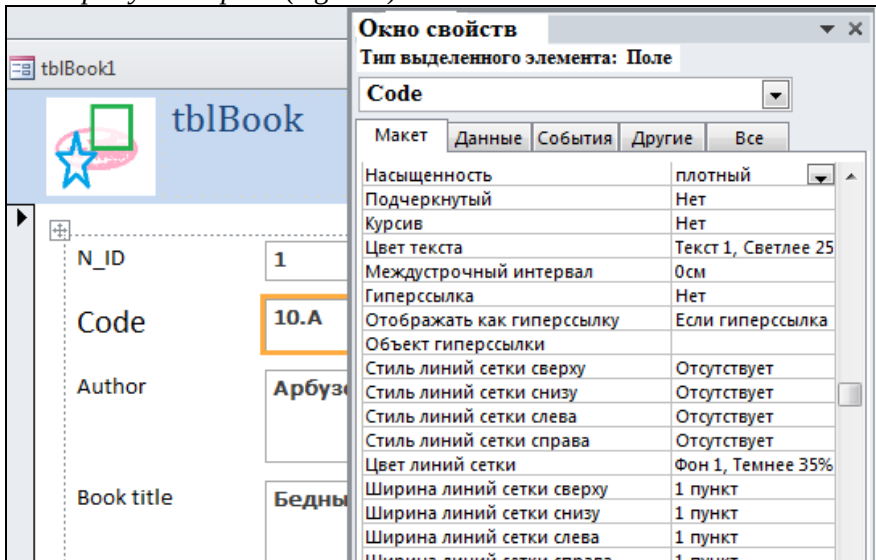


Figure 3.11 – Property Sheet pane of the form frmBook1

The properties of the **Code** text box control are organized on four pages: *Format*, *Data*, *Event*, and *Other*. As its name suggests, the *All* page displays all the properties on one page.

10. Notice that the *Property Sheet pane* displays the properties for the object whose name appears in the text box at the top of the pane, and that above the box, the type of object is identified.
11. In the *Property Sheet pane*, click the *Format tab*. The *Format page* of the *Property Sheet pane* displays all the commands available in the *Font group* on the *Format tab* of the ribbon (plus a few more).
12. On the *Format page*, click *Font Size*, click the arrow to the right of the adjacent property, and in the list, click 16.
13. Set the *Font Weight* property to *Bold*.
14. Let's use a faster method. Select all items on the form. In the *Property Sheet pane*, notice that the selection type is *Multiple selection*, and the box below is blank. Only the settings that are the same for all the

selected controls are displayed. Because the changes you made in the previous steps are not shared by all the selected controls, the *Font Size* and *Font Weight* settings are now blank.

15. Repeat steps 12 and 13 to set the Font Size and Font Weight properties of the selected controls to 16 and Bold.

Arranging the *layout* (maket) of forms

If the default layout doesn't suit your needs or preferences, you can customize it. Most of the rearranging you are likely to want to do can be accomplished in *Layout view*, where you can view the impact on the underlying data. If you want to make more extensive changes to the layout of a database, you can switch to *Design view*.

In Layout view, you can do the following to improve the form's layout and make it attractive and easy to use:

- Add and delete a variety of controls.
- Change the size of controls.
- Move controls.
- Change text alignment.
- Change the margins of controls.

Example 3. In this exercise, you'll align, size the label and text box controls in a form.

1. In the *Navigation pane*, in the *Forms group*, right-click **frmBook1**, click *Layout View*.
2. Click the **Code label** control (not its text box), and on the Arrange (Упорядочить) tool tab, in the Rows & Columns group, click the Select Column button.
3. With all the label controls selected, on the *Format page* of the *Property Sheet pane*, set the *Text Align property* to right-align all the labels in their controls. You can efficiently adjust the alignment of multiple selected controls by changing the *Text Align property* in the *Property Sheet pane* (fig. 3.12).
4. The Property Sheet pane indicates that the Width property of the label controls is 3,092см. Let's make them narrower. Point to the right border of the **Code label**. In the *Property Sheet pane*, adjust the *Width property* to 2 см, and press Enter.
5. Select the **Code text box** (not its label), change its Width property to 3см, and press Enter. Notice that when you size one control in a Stacked layout, all the controls in the same column are adjusted, not just the

selected control.

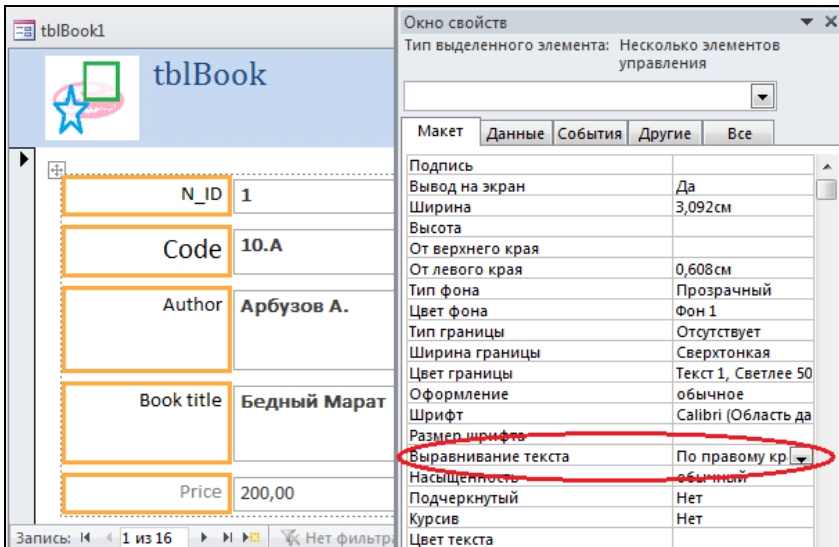


Figure 3.12 - You can efficiently adjust the alignment of multiple selected controls by changing the *Text Align* property in the *Property Sheet* pane

3.2.3 Create reports by using a wizard

You can divide the content of an Access report into two general categories: information derived from records in one or more tables, and everything else. The everything else category includes the title, page headers and footers, introductory and explanatory text, and any logos and other graphics.

Just as you can create a form that includes all the fields in a table by using the *Form tool*, you can create a report that includes all the fields by using the *Report tool*, which is located in the *Reports group* on the *Create* tab. But such a report is merely a prettier version of the table, and it does not summarize the data in any meaningful way. You are more likely to want to create a report based on only some of the fields in the table, and that is a job for the *Report wizard*.

The *Report wizard* leads you through a series of questions and then creates a report based on your answers. So the first step in creating a report is to consider the end result you want and what information you need to include in the report to achieve that result. After you provide that information, the wizard creates a simple report layout and adds a text box

control and its associated label control for each field you specify.

Example 4. For example, you might want to use **tblBook** table as the basis for a report that groups books by Author. When you give the grouping instruction to the wizard, it first sorts the table based on the author, and then sorts the books of each author by books title. In the space at the top of each group (called the group header), the wizard inserts the name of the author.

1. On the *Create tab*, in the *Reports group*, click the *Report Wizard button* to start the wizard. If the **tblBook** table was still selected in the *Navigation pane*, that table is specified in the wizard's *Tables/Queries* box and its fields are listed in the *Available Fields* box. Let's tell the wizard that we want to base this report on the **tblBook** table instead.
2. Display the *Tables/Queries list*, and then click Table: **tblBook** to list that table's fields in the *Available Fields* box.
3. In the *Available Fields* list, double-click **Author**, **Book title**, **Publication date** and **Price** to move them to the *Selected Fields* box (fig. 3.13).

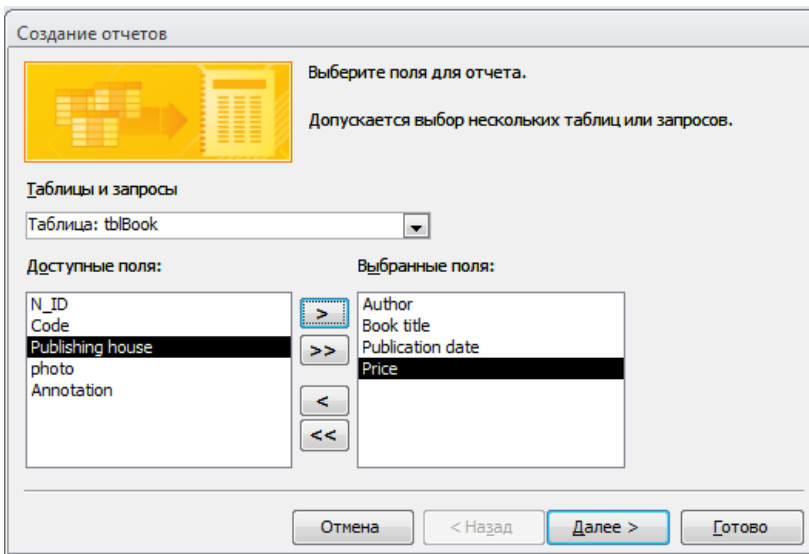


Figure 3.13 – Report Wizard

4. At the bottom of the page, click *Next*.
5. The wizard asks whether you want to group the records. When you group by a field, the report inserts a group header at the top of each group of

records that have the same value in that field. Let's specify that we want the books grouped by the first letter of their author. In the field list on the left, double-click **Author** to move that field into the group header area of the box on the right (fig.3.14)..

- 6 In the lower-left corner of the page, click *Grouping Options* to open the *Grouping Intervals* dialog box. In the *Grouping intervals* list, click **1st Letter**, and then click OK (fig.).

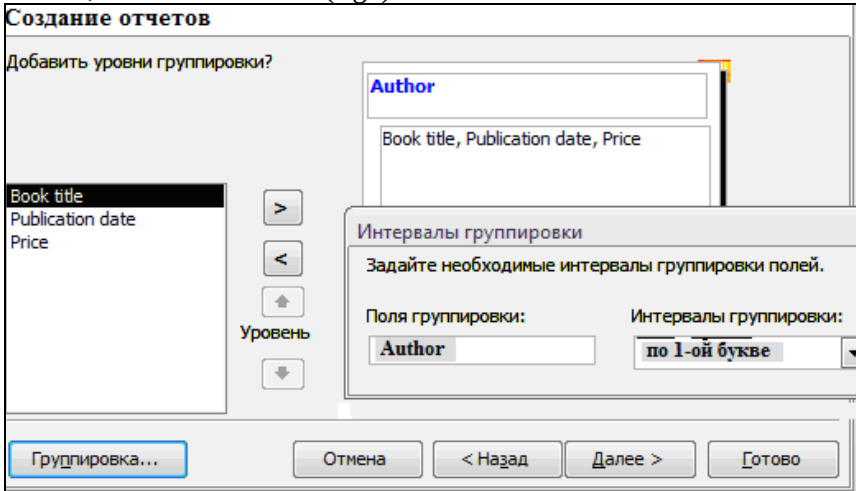


Figure 3.14 - The wizard asks whether you want to group the records

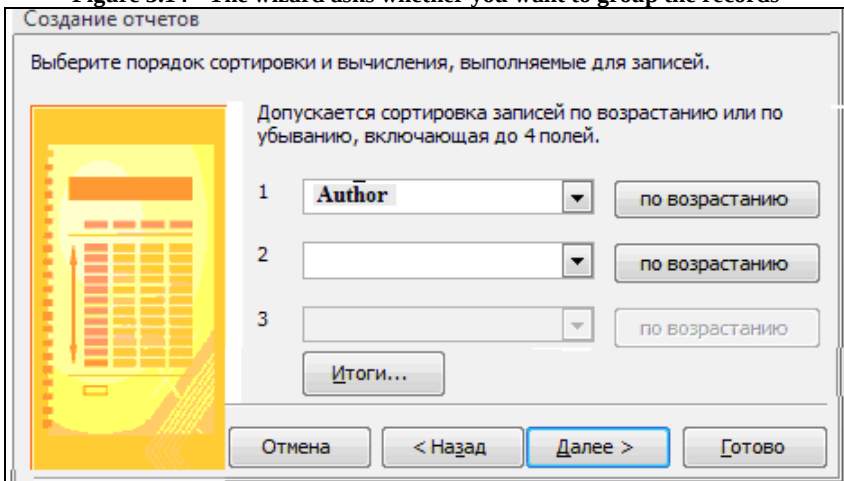


Figure 3.15 - Page on which you can sort and summarize the grouped records

7. Click *Next* to move to a page on which you can sort and summarize the grouped records (fig. 3.15).
8. With *Portrait* selected in the *Orientation* area and the *Adjust the field width so all fields fit on a page* check box selected, click *Next*.
9. Let's give this report a title that easily conveys its content. In the title box, enter **Alphabetical List of Books**, and then with *Preview* the report selected, click *Finish* to create the report (fig. 3.16) and display it in *Print Preview*.

rptBook2

Alphabetical List of Books

Author	A A Halpin		
Book title	Publication date	Price	
Information modeling and relational da	2017	280,00	

Author	A David Hay		
Book title	Publication date	Price	
Data model patterns : conventions of th	2014	190,00	

Figure 3.16 - Report, created by Report Wizard

3.3 Tasks for the laboratory work

Task 1. Run examples of Section 3.2

Task 2. Create forms for viewing data in the **tblReaders** table using the following tools:

- *Simple Form*,
- *Split Form*,
- *Multiple Items Form*.

Task3. Create a form to view the data in the **tblBook** table using the *Form Wizard*.

Task 4. Format the forms created in Tasks 2, 3 in the *Layout View*.

Task5. Create a form with subform using the *Form Wizard*.

A subform is a form within a simple (primary) form. A primary form

is called the main form and the form within the form is called the subform. Subforms are especially useful when you want to show data from tables or queries with one-to-many relationships. For example create a form that for every reader gives a list of readable books. All information for the subform will be taken from the query **quList1**(author, title, date of issue and date of return), that was created during previous laboratory work.

To create a form with subform:

1. On the Create tab click *Form Wizard*.
2. The window *Form Wizard* appears. Select the table **tblReaders** from list. Transport the fields you want to be included from *Available Fields* to *Selected Fields*, namely **Card, Name, Chair** and **Telephone**. Then in the same dialog box select the query **quList1** and transport all *Available fields* to *Selected fields*.
3. When you click *Next*, if you set up the relationships correctly before starting the wizard, the wizard asks which table or query you want to view by. Click *by Readers (no Chumamenu)* because main form is the form that shows information about readers. Select the *Form With Subform* option and click *Next*.
4. In the next window choose the layout of subform. It would be more convenient to see the information about books being read now in a table view so select *Columnar* or *Tabular* layout and click *Next*.
5. Select a style for the main form. You can see its view at just as you selected one of them. Click *Next*.
6. Next window asks you to give titles for you forms (both for the main and for the subform). Access created two forms related to each other. Nevertheless you can edit them in design view separate from each other and you may also use subform independently of the main one. Name the main form **frmMainList1**, and subform – **frmSubList1**. Click *Finish* and you will see the created form.

Task 6. Create a report that outtypes a list of library readers ordered by departments and a list of books for each reader, as shown in Figure 3.17. It must also calculate a fine for each reader and for all readers. For creation of such a report we are to use the table **tblReader** and the query **quList1** that contains numerical field **Fine**.

List of readers and readable books by each reader					
Кафедра		ВМ			
Прізвище	Белая Н.И.				
Телефон	3-32				
	Дата видачі	Автор	Назва книги	Дата повернення	Пеня
	12.10.98	Гарнаев А.	Эффективная работа с Экс	01.11.98	110.17
	15.02.00	Ахаян	Access	16.03.00	0
	15.02.00	Гарнаев А.	Самоучитель VBA	16.03.00	0
	19.02.00	Пинчук	Объектно-ориентированное	20.03.00	0
Итого для 'ВМ' = 830 (4 записей)					
Sum				110.17	
Итого для 'Кафедра' = ВМ (4 записей)					
Sum				110.17	
Кафедра		ЭА			
Прізвище	Бондаренко Л.А.				
Телефон	4-30				
	Дата видачі	Автор	Назва книги	Дата повернення	Пеня
	01.01.98	Макашарипов	Базы данных	01.03.98	72.4
	10.10.98	Керри Н.Праг	Библия пользователя	10.11.98	202.1

Figure 3.17 - Report that outtypes a list of library readers ordered by departments and a list of books for each reader

3.4 Review and discussion questions

1. Ways to create forms and reports in Access.
2. What forms can create with a single mouse click?
3. Definition of the Basic Form.
4. Definition of Split Form.
5. Definition of Multiple Items (Continues) Form.
6. Definition Datasheet Form.
7. What forms can create with Form Wizard?
9. What is the records source of the form?
10. Structure and purpose of the form properties window
11. The distinction between the assignment of a form and a report.

12. How can change the properties of the form and form's fields?

3.5 The contents of the report

1. Subject and purpose of work.
2. Protocol of work.
3. Conclusions.

4 LABORATORY WORK № 4 CREATING AND MODIFYING A FORM IN DESIGN VIEW

4.1 Purpose of the work

Studying Building and Working with Forms

4.2 Brief theoretical information

How you knowing, Microsoft Access has numerous wizards that help you in the creation of database objects. I recommend that you use the *Form Wizard* to help you when creating a new form. The *Form Wizard* provides a great framework that you can customize and build upon, which saves a lot of time. Often, the *Form Wizard* can build approximately 70% of the form for you instead of you having to build every single piece of the form “from scratch” in *Design View*.

Once the form is created, it can be viewed in the *Form View*, *Layout View*, or the *Design View*. Each view has its distinct purpose, advantages, and disadvantages, which are summarized here. While building database forms, you will often switch between these views.

- *Form View* is the end-user view of the form used for entering and displaying data. You can fully update and work with the form data in this view, but you cannot make any changes to the form design or layout.

- *Design View* provides the most powerful way to work in the design and layout of the form. In this view, you cannot see or work with any of the form data.

- *Layout View* is a combination of *Form View* and *Design View*. It enables you to make most design changes to the form while viewing live form data, but you cannot update the data you are viewing. It is very powerful because it instantly allows you to see how design changes will look on the form with data.

Before diving into the details of building a form, it is important to be familiar with a little more terminology related to forms and to understand the principle of inheritance when building a form.

First, anything you see on a form is called a control. This includes items such as labels, textboxes, combo boxes, option groups, checkboxes, images, command buttons, and even graphical lines.

Controls can be further classified as *Bound*, *Unbound*, or *Calculated*:

- A *bound* control is bound or “connected” to a specific field in an underlying table or query. When the data in a bound control is updated, it is ultimately updating the field in the table from which the form is built. Textboxes, combo boxes, option groups, and checkboxes are some common controls that are usually bound to an underlying data source.

- An *unbound* control is not connected to any underlying data source in a table or query. Images, labels, and other graphics are common examples.

- A *calculated* control is based on a formula, function, or expression that displays a calculated result. Although calculated controls often use underlying data fields as part of the calculation, they are unable to update any data in the table like a bound control can. As the underlying fields used in the calculation are updated, the result displayed in the calculated control is also updated. The results of calculated controls are usually displayed in textboxes on a form.

Create Form in Design View:

1. In the *Database* window, click the *Create Tab*.
2. In the *Forms Group*, click the *Form Design* button (see fig. 4.1).

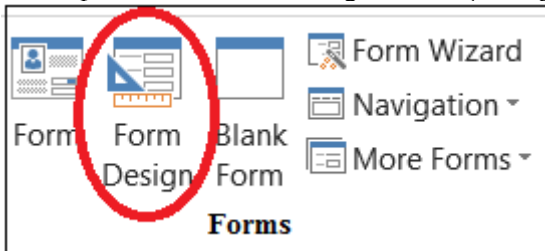


Figure 4.1 - To open the Form Designer, click on the Form Design button

3. The *Form Design* window will display (fig.4.2).
4. The *Form Design Tools* tab will appear at the top of the window.

All forms have a *Detail* section, but a form can also include *Form Header* and *Form Footer* sections. To add or remove these sections click *Form Header/Footer* on the *View* menu.

Add Existing Fields

All the information on a form is contained in controls. So far, the only kind of “control” you have used on your forms has been the text box:

1. Click the *Form Design Tools*. *Design Tab* if necessary.
2. In the *Tools Group*, click the *Add Existing Fields* button (see fig.4.3)

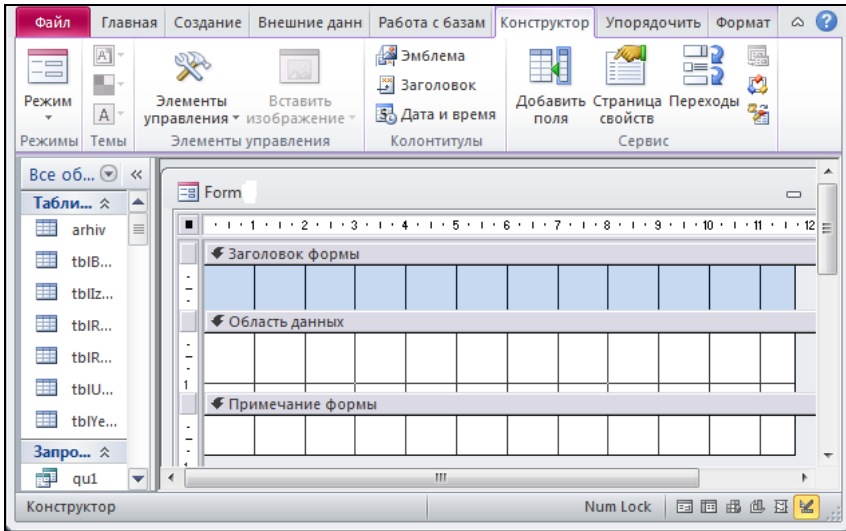


Figure 4.2 - The Forms designer window

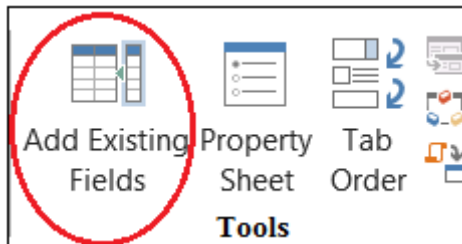


Figure 4.3 - Add Existing Fields

However, Access provides other controls. Microsoft Access includes the different types of controls, which are all accessible through the Toolbox (fig.4.4).

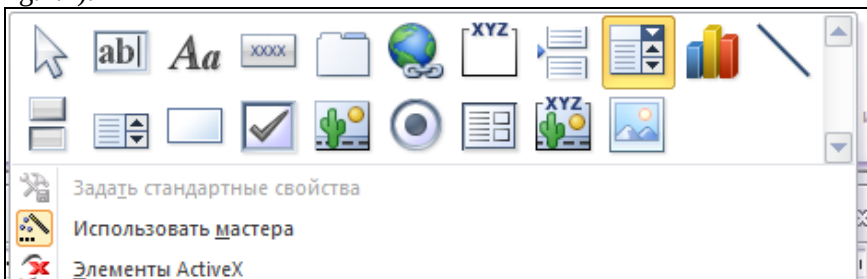


Figure 4.4 - Microsoft Access includes the different types of controls

All properties of the form, its sections, and the elements of management located on the form, are displayed and edited in *the Window of Properties* (fig.4.5).

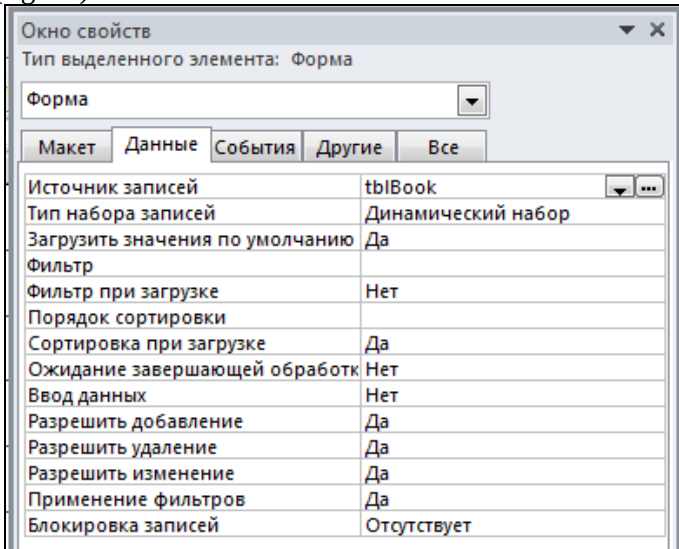


Figure 4.5 - All properties of the form, its sections, and the controls are displayed and edited in the Window of Properties

4.3 Examples of development of forms for a DB "Library"

4.3.1 Forms of data input

Use *the Form Wizard* to create forms (**frmBook** and **frmReaders**) of data input into tables **tblBook** and **tblReader**. Change the *Data input* property (*tab Data*) into **ON**.

After end of work of *the Form Wizard*, you may decide to modify it to add additional features, and also to add if necessary heading and figures. Improvement of forms can be carried out also due to addition of a choice of the requisite information for a field **Publishing house** out of *Combo Box* control.

Creating the form **frmReadBook**, for data input into the table **tblReadBook**, it is necessary to trace, that book given out to the reader really is in library.

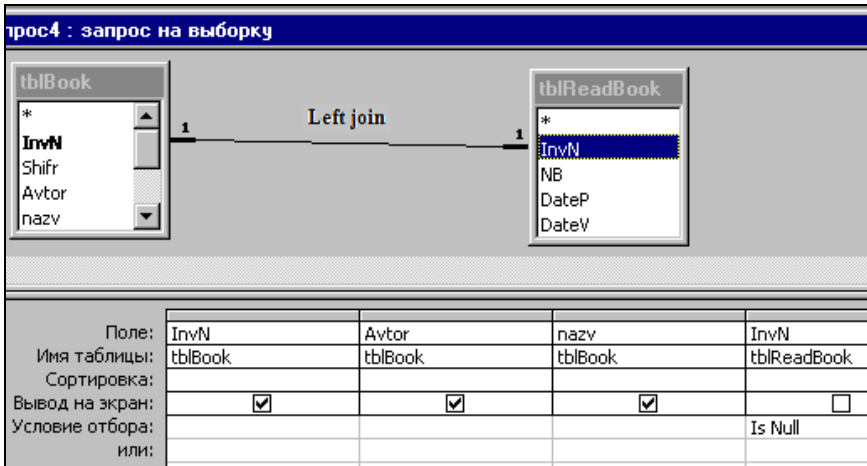


Figure 4.6 - Inquiry quListReadBook

The decision of a problem: create inquiry as a source of the data for the form (for example, under a name **quListReadBook**) which creates a set of records with the information only about books available in library at the moment. The inquiry **quListReadBook** is displayed on the fig. 4.6.

4.3.2 Forms of viewing of the data by various choice criteria

1) We shall create the form **frmLookBook** for search of the necessary book «By a surname of the author», «By the name of the book ». We shall arrange on the form the following elements of management:

- Two elements of management *ComboBox* for input of search criterion. We shall appropriate it names – **cmbAvtor** and **cmbNazv**. We shall arrange them on different sheets an element of management *the Set of supplementary sheets (Tab)*;
- The attached framework of object and text fields for a conclusion of the information about found, according to the chosen criterion of search, the book.

The form **frmLookBook** at different stages of work of system is submitted on fig.4.7, 4.8.

Creation of fields with the list (**cmbAvtor** and **cmbNazv**)

If you, having clicked preliminary in the panel of elements on the button of *the Wizard*, will choose an element *ComboBox* and place it on the form, on the screen of the monitor the first window of the master will

appear. On a first step it is offered to choose a way of reception of lines of values. It is necessary to choose one of three variants:

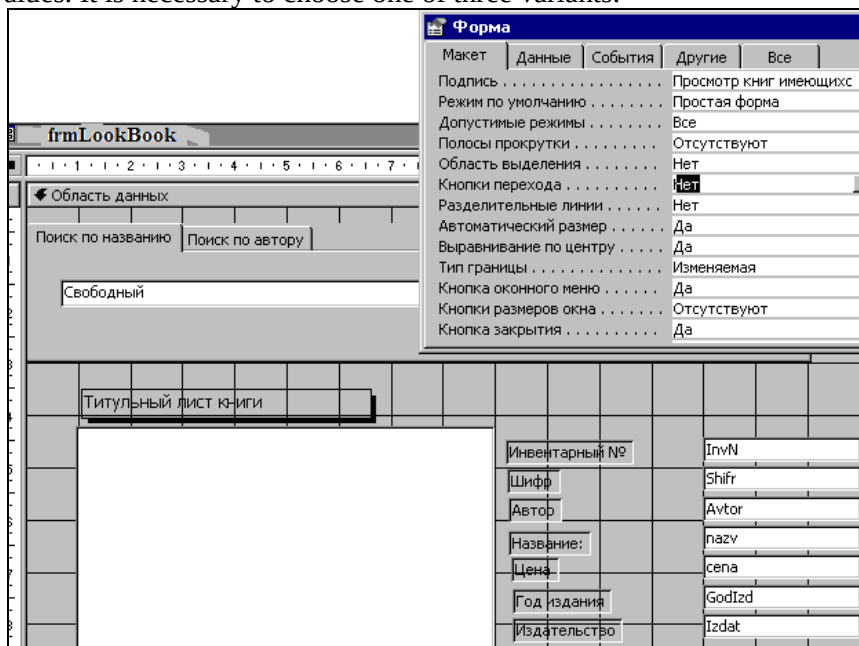


Figure 4.7 - The Form frmLookBook in a designer mode with the deduced window of form properties

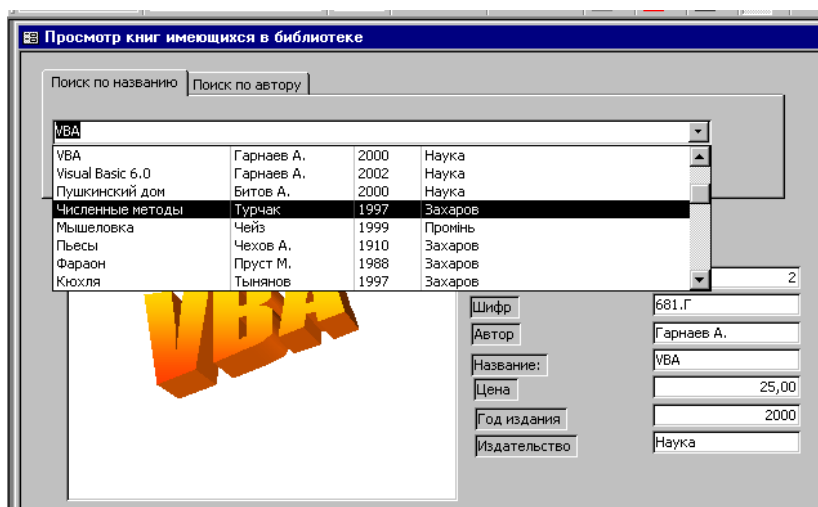


Figure4.8 - The Form frmLookBook

- *Object the Field with the list* will use values from the table or inquiry. In this case the list of values will be based on the inquiry choosing necessary lines from the base table or inquiry.
- *The fixed set of values will be entered.* A source of lines for a field with the list will be the fixed set of values entered by the user.
- *Search of record in the form on the basis of value which contains a field with the list.* The field with the list can be used for transition to record in a source of the data of the form. In this case a source of lines for a field with the list automatically becomes a source of the data of the form.

It is necessary to choose *Search of record in the form on the basis of value which contains a field with the list*. Click on the button *Further* to proceed to the following window of the master. At this stage it is necessary to choose fields which values will make columns the dropping out list. Make a necessary choice and click on the button *Further*. The further sequence of windows of the Wizard will allow you:

- To establish desirable width columns (it can be made and later when the field with the list will be already created);
- To specify a field which value will represent value of a field with the list. In other words, you should choose a column which value is kept in variable forms;
- To specify a name of the created field with the list.

The kind of the created elements of management **cmbAvtor** and **cmbNazv** is determined through properties which need to be established in a window of *Property of a field with the list*.

4.3.3 Forms for editing the data

In such forms the opportunity of editing of the information in the table (or tables), being by a source of the data for the required form should be given to the user, namely:

- To move on the necessary record;
- To change the information directly in a field;
- To add new records;
- To delete records.

Поиск по названию | Поиск по автору

Гарнаев А.

Кнопка для удаления книги из каталога библиотеки: [X]

Регистрация новых поступлений книг в библиотеку: [+]

Обновление формы после регистрации новых книг: [-]

Титульный лист книги

Инвентарный №	2
Шифр	681.Г
Автор	Гарнаев А.
Название:	VBA
Цена	25,00
Год издания	2000
Издательство	Наука

Перемещение по записям: [←] [→]

Figure 4.9 - The Example of the form for editing the data in the table **tblBook**

As an example we shall create the form **frmEditBook** (see fig. 4.9) for editing the information in the table **tblBook**. For this purpose we shall finish the form **frmLookBook**. We shall arrange on it the following buttons:

- cmdAdd** - for addition of new record in the table **tblBook**;
- cmdRefresh** - for updating the information displayed in the form;
- cmdDelete** - removals of record of the table **tblBook**;
- cmdGo** - for moving on records.

Buttons we shall add with the help of the Wizard of buttons (*Command Button Wizard*). For creation of buttons **cmdAdd**, **cmdDelete**, **cmdGo**:

1. Of a window of the designer of the form be convinced, that the button of the Master (*Control Wizards*) on the panel of elements is made active, then press the button the Button (*Command Button*) and click a mousy in that place of the form in which be going to create the button. On the screen the dialogue window *Creation of buttons (Command Button Wizard)*, containing the list of categories and actions for the chosen category will appear. The window *Creation of buttons* is resulted in the

figure 4.10.

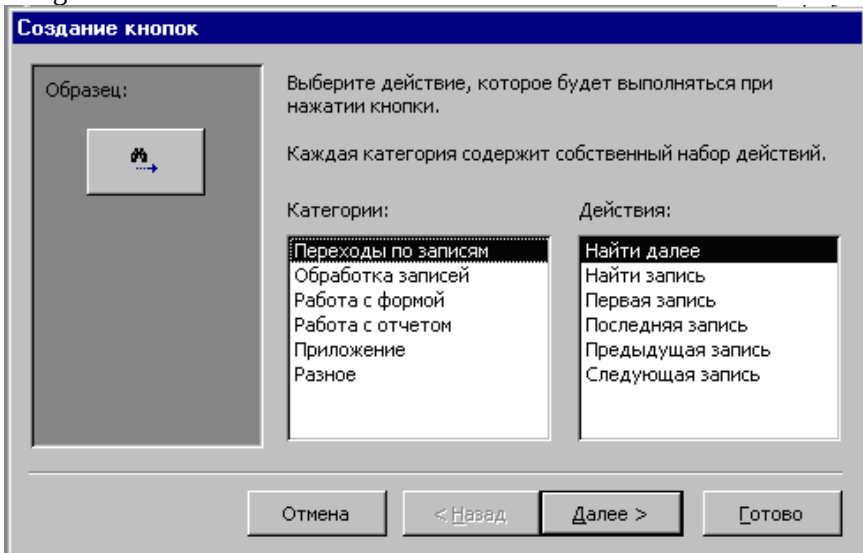


Figure 4.10 - The Window Creation of buttons

2. Choose item *Transitions on records* or *Processing of records* in the list of the *Category (Categories)*, then item{point} corresponding to purpose{assignment} of the button in the list *Action (the Previous record, the Following record To remove record To add record)*.
3. For creation of the button **cmdRefresh** choose item *Work with the form (Form Operation)* in the list of the *Category*, then item *To update the given forms* in the list of *Action*.

4.3.4 Modifying a form frmMainReaders created in Labrotory work #3

After you create a form, you may decide to modify it to add additional features. For example, you must to add control that display the total value of fine for each reader. With the help of *Form Designer* execute the following steps:

1. Create calculated control with name **TotalFine** in the subform **frmSubList1**;

Open subform **frmSubList1** in *Design View*. In a *Form Footer* of the form create a calculated control (*Text Box*). Name it **Total Fine** and write the following formula to the *Data properties*: **=Sum([Fine])**. The total value of fine for each reader will be calculated by this formula. Save all

changes and open the form for review. The created field is not visible because the subform **frmSubList1** is represented in datasheet view. Nevertheless the field exists and all calculations are performed.

2. Create calculated control in the main form **frmMainReaders** (fig. 4.11), that takes information from the control **TotalFine**.

Open the form **frmMainReaders** in *Design view* and add a control item *TextBox*. Right-click the control and select *Properties* from the shortcut menu. Click the *Data tab* and enter the following expression to the *Control Source* box:

=[Forms]![frmMainReaders]![frmSubList1].[Form]![TotalFine]

Figure 4.11 - The main form frmMainReaders with created calculated control

4.4 Tasks for the laboratory work

Create for your database the forms realizing basic of functions of any appendix Access, namely:

- Forms for data input in tables;
- Forms for viewing the data;
- Forms for editing the data;
- Forms for display of the data from the several connected tables. For example, the form with the subform which contains calculated fields, final values, et;

Use opportunities of a mode *the Designer* for creation of forms and

toolkit of Wizards, for construction of elements of management.

4.5 Review and discussion questions

1. Creation of forms in a mode of the designer.
2. Structure and properties of forms.
3. Elements of management, purpose of each element, their property.
4. Two ways of use of an element *the Field with the list*.
5. Use of *the Master* for creation of buttons. Kinds of *Categories* and *Actions* of a window *Creation of buttons*.
6. Concept of the subordinated form, the connected form. Creation of calculated fields with use of *"the Builder of expressions"*.

4.6 The contents of the report

1. Subject and purpose of work.
2. Protocol of work.
3. Conclusions.

LITERATURE

Basic

1. David Murray. Microsoft Access 2013. Tutorial and Lab Manual. University at Buffalo. 2014 by E-Assign LLC, -122p.
1. Гурвиц Г. А. Microsoft Access 2010. Разработка приложений на реальном примере. – Издательство: BHV, 2010. – 496с.
3. Мирошниченко Г. А. Реляционные базы данных: практические приемы оптимальных решений. –СПб: БХВ-Петербург, 2005. –400с.
4. Єрьоміна Н. В. Проектування баз даних: Навчальний посібник. –К. : КНЕУ, 1998. – 208 с.
5. Джеффри Д. Ульман, Дженнифер Уидом. Введение в системы баз данных. – Издательство «Лори», 2000. – 376 с.
6. Методичні вказівки до самостійної роботи з курсу “Програмні засоби проектування електромеханічних систем” для студентів денної та заочної форм навчання напряму підготовки спеціальності 141 "Електроенергетика, електротехніка та електромеханіка – / Укл.: Л.О.Бондаренко, О.О.Каплієнко. – Запоріжжя: ЗНТУ, 2015.– с.70. №5706е

Internet resources

1. Сайт додатків Office корпорації Microsoft [Електронний ресурс].
<http://office.microsoft.com/ru-ru/access-help/>
2. Сайт комп'ютерної літератури «Таурион» [Електронний ресурс].–
Режим доступу: <http://www.taurion.ru/access>
3. Сайт «AccessSoft» по розробці баз даних за допомогою Access
[Електронний ресурс]. – Режим доступу:
<http://www.accessoft.ru/index.html>
4. Сайт «Библиотека on-line» по розробці баз даних [Електронний
ресурс]. – Режим доступу: <http://citforum.ru/database>

Additional

1. Ivankov V. F. Calculation of CFD-thermal models of oil-cooled transformer equipment / V. F. Ivankov, A. V. Basova // Електротехніка та електроенергетика. - 2016. - № 2. - С. 19-32.
2. Ніценко В. В. Дослідження похибок трансформаторів струму у системах релейного захисту в усталених та перехідних режимах енергосистеми / В. В. Ніценко, Д. О. Кулагін, П. В. Махлін // Електротехніка та електроенергетика. - 2016. - № 2. - С. 59-71.
3. Ніценко В. В. Дослідження похибок трансформаторів струму у системах релейного захисту в усталених та перехідних режимах енергосистеми / В. В. Ніценко, Д. О. Кулагін, П. В. Махлін // Електротехніка та електроенергетика. - 2016. - № 2. - С. 59-71.
4. Дивчук Т. Е. Особенности определения параметров силовых трансформаторов методами схемно-полевого моделирования / Т. Е. Дивчук, Д. К. Мимоход, С. А. Кутилин, А. Э. Кузнецов, Ю. В. Гуразда, И. С. Сырых // Електротехніка та електроенергетика. - 2017. - № 1. - С. 61-70.
5. Дьяченко В. В. Формирование программы энергосбережения для систем электроснабжения / В. В. Дьяченко // Електротехніка та електроенергетика. - 2015. - № 1. - С. 70–76.