ALFRED NOBEL UNIVERSITY,
DNIPROPETROVS'K

N.O. Rizun
L.N. Savchuk

INFORMATICS

Access

Part 3

Workshop

ALFRED NOBEL UNIVERSITY,
DNIPROPETROVS'K

N.O. RIZUN
L.N. SAVCHUK

INFORMATICS

PART 3

WORKSHOP

FIELD: 0305 − ECONOMICS AND ENTREPRENEURSHIP
for:
6.030503 − INTERNATIONAL ECONOMICS

Dnipropetrovs'k
2014

УДК 004
ББК 32.81
Р 49

Reviewer:
B.E. Momot, Doctor of Economics, Full Professor,
Head of Innovation Management and International Logistics Department.

Лабораторний практикум з дисципліни «Інформатика» (частина 3) призначений для студентів економічних спеціальностей, що навчаються за англомовною програмою, і передбачає отримання практичних навичок роботи із системою управління базами даних MS Access 2010 (модуль 3).

The workshop in Informatics (part 3) is intended for students of economic specialties, who study by the international (English) program, and provides for acquiring practical skills to work with relational database management system MS Access 2010 (Module 3).

Responsible for the publication:
S.B. Vakarchuk, Doctor of Physics and Mathematics, Full Professor,
Head of Informatics and Mathematical Methods in Economics Department.

ISBN 978-966-434-292-3

© N.O. Rizun, L.N. Savchuk, 2014
© Alfred Nobel University,
Dnipropetrovs’k, design, 2014

CONTENTS

INTRODUCTION

In the framework of discipline "Informatics" (the "Relational Database Management System"), students study one of the most powerful and popular at the present time Relational Database Management System – MS ASSESS 2010.

Relational Database Management System MS ASSESS 2010 is part of the Microsoft Office package, and in fact has become the industry standard for professional work to create a relational database type. For user interaction with the data stored in the database, the database contains the MS ASSESS 2010:

– support tools tables and relationships between related tables;

– advanced user interface that allows you to enter and modify, find, and present information in text or graphic form

– high-level programming tools with which you can create your own applications.

The purpose of this workshop is to provide a laboratory of theoretical and practical skills of independent work in MS ASSESS in solving real-world economic problems. In the study of the functionality of MS ASSESS identified the following steps:

– creation and organization of single-table and multi-table databases;

– the use of forms for data entry and editing;

– creation aggregate, crosstab, parameter and action queries;

– report generation.

Stages of performing the workshop include:

– studying the tasks of the workshop;

– performing the tasks using lecture notes, a reference system, textbooks and consultations of the lecturer;

– preparing for giving answers to the testing questions on the workshop theme;

– defending the workshop.

Performing the workshops is done in the computer classrooms of the university at the seminars as well as individual work of the students.

Workshop № 1

CREATION OF TABLES IN THE MS ACCESS 2010

1.1. Creating the Table in Datasheet View

Assignments:

1. Start the ACCESS and create new empty database **CONTACTS** (**Blank Database / Create)**.

2. In *Datasheet View* (tab **Create** / group **Tables** / icon **Table)** create the table **Address_book**, which contains the following fields: *LastName*, *FirstName*, *BusinessPhone*, *MobilePhone*, *Address*, *StateProvince*, *City*, *ZIPPostal*, *CountryRegion*, *Country*, *DateOfBirth*, *CompanyName*, *Credit*, *Notes*. In the process of the adding fields appropriate *Data type* (the list **Click to Add).**

3. Complete the table **Address_book** with not less than 15 records. The main data requirements:

– the list of subscribers should contain citizens of different countries (including *Russians*) and of different cities (including *Kiev* and *Moscow*);

– a few persons should work in the *Dnipropetrovsk's Alfred Nobel University;*

– the list should contain employees of the *Airit* company, who work in subsidiaries in *Kiev* and *Moscow*;

– a few subscribers should have their birthdays in *March*.

4. Do the following structure modifications of the table **Address_book**:

– add the field *CompanyExecutive* after the field *CompanyName* (the command of the contextual menu **Insert Field**)*;*

– add the field *FaxNumber* after the field *BusinessPhone* (tab **Fields** / group **Add & Delete** / icon with the appropriate *Data type*);

– place the field *MobilePhone* before the field *Notes*.

5. Modify the *Structure* and set the *Properties* of the table **Address_book** (tab **Fields** in the *Datasheet View*):

a) for all the fields, which names consist of a few words, set the property **Caption**, the text of which should contain the name of the current field with the gap between words (tab **Fields /** group **Properties**). For example, for the field *LastName* the property **Caption** should be the following: *Last Name*;

b) for fields *LastName*, *FirstName* set the property **Required** – **Yes**;

c) for field *Country* set the property **Default Value** – *Ukraine*, and for field *City – Dnepropetrovsk* (tab **Fields /** group **Properties**);

d) for field *Credit* set (tab **Fields /** group **Formatting**):

– the property **Field Size – Single**;

– the property **Format** – **Standard**;

– the property **Validation Rule**, supposing that the data in this field cannot be negative;

e) for the field *DateOfBirth* set the property **Validation Rule,** supposing that the date of birth should be later than 01.01.1990 (tab **Fields /** group **Field Validation /** icon **Validation / Field Validation Rule**);

f) for all the **Validation Rules,** which were set, formulae the **Validation Text** (tab **Fields /** group **Field Validation /** icon **Validation / Field Validation Message);**

g) add the **Calculated** field **–** *ContactName*, which allows forming the name for a contact with the help of the following **Expression** (tab **Fields /** group **Add & Delete /** icon **More Fields / Calculated Field / Text**):

[FirstName] & " " & [LastName]

6. Check operation of the added **Calculated** field and formed **Properties** of the rest of table fields.

1.2. Creating the Table in Design View

Assignments:

1. In the mode *Design View* for fields *BusinessPhone*, *MobilePhone*, *FaxNumber*, *ZIPPostal*, *DateOfBirth* set the property **Input Mask** (tab **Fields /** group **Views /** icon **Design View**).

2. Check operation of the created **Input Masks** in the *Datasheet View* (tab **Fields /** group **Views /** icon **Datasheet View**).

3. Create a new database **CLIENT'S_ORDERS** (tab **File / New/ Blank Database / Create)**.

4. In the mode *Design View* create the table **Orders** with the following structure:

Structure of the table Orders

Field name	**Data type**	**Description**
OrderCode	Text	Unique value for every order
ClientCode	Text	Unique value for every client
GoodCode	Text	Unique value for every product
DateOfOrder	Date/Time	Date of order
OrderQuantity	Number	Number of the units ordered
DateOfDelivery	Date/Time	Planned date of delivery
Notes	Memo	Additional information about the order

5. For all the fields, which names consist of a few words, set the property **Caption**, the text of which should contain the name of the current field with the gap between words.

6. Organize data control at the moment of entering values into the table **Orders**. To do this:

– for the field *OrderCode* create the **Input Mask**, consisting of *two* required and *one* not required letters, placed in the beginning of the code (the first letter should be automatically transferred to the upper register), and *five* obligatory numbers;

a) for the field *ClientCode* create the **Input Mask**, consisting of *one* required letter, placed in the beginning of the code (the letter should be automatically transferred to the upper register), and *ten* required numbers;

b) for the field *GoodCode* create the **Input Mask**, which begins with letter "*G"* and a gap and consists of *two* required numbers and *two* obligatory letters, separated from the numbers by the symbol "–";

c) for field *OrderQuantity* set the property*:*

– **Field Size –** *Single*;

– **Format** – *Standard*;

– **Decimal Places** – *2*;

– **Validation Rule,** supposing that data in this field *cannot* be negative and cannot have a value *of more than 1000000*;

d) for the field *DateOfOrder* set:

– the property **Validation Rule**, supposing that the date *should not* be in the interval from the beginning of the current year till the current date;

– the property **Default Value –** *current date***;**

e) for field *DateOfDelivery* set the property oa **Validation Rule**, supposing that the date *should not* refer to the previous years;

f) for all the set value properties formulate the text of the property **Validation Text**.

7. Add the **Calculated** field **–** *Day'sTillDelivery*, which allows calculating the number of days between the order and the planned date of delivery with the help of the following **Expression**:

[DateOfDelivery]-[DateOfOrder]

8. Look through the **Description** and check operation of the added

Calculated field and the properties, formed in the mode *Datasheet View* (right bottom corner of the window ***Design View*** / icon **Datasheet View**).

9. Complete the **Validation Rule** for the whole table (button **Property Sheet** on the toolbar of the window ***Design View***), where you should set the following conditions for data control:

– value of the field *DateOfDelivery* should be higher, than that of the *DateOfOrder*;

– and the period between values of fields *DateOfOrder* and *DateOfDelivery* should not be more than 30 days;

10. Check operation of the created **Properties** and **Calculated** field in the *Datasheet View* (tab **Fields /** group **Views /** icon **Datasheet View**).

TESTING QUESTIONS

1. Enumerate all the possible ways (minimum 3) of creating table fields in the mode *Datasheet View*.

2. Name the peculiarities of adding the fields between those already existing in the mode *Datasheet View* (minimum 2).

3. Name the ways of changing the field name in the mode *Datasheet View*?

4. Name the ways of changing the type of data in the field of the mode *Datasheet View*?

5. Name the rules of creating the table fields' names.

6. Name the ways of deleting the table fields in the mode *Datasheet View*?

7. Name the ways of moving the table fields in the mode *Datasheet View*?

8. Name the field properties, which can be changed in the mode *Datasheet View*?

9. Name the data types, which are acceptable in Access?

10. What do we use the modifier *Description* for?

11. Give the definition of the property *Input Mask*. What symbols are used in the process of *Input Mask* creation?

12. What do we use the property *Default Value* for? Give examples.

13. How are the following field properties connected with each other: *Validation Rule* and *Validation text*? Give examples of the use of these properties.

14. What is the difference between the property *Validation Rule*, set for concrete table fields and for the whole table?

ASSISTANCE

The symbols, we can use when create the MASK OF INPUT:

SIGN	DESCRIPTION
0	Figure (from 0 through 9, input is required; signs plus [+] and minus [-] are not accepted).
9	Figure or space (input is not required; signs plus [+] and minus [-] are not accepted).
#	Figure or space (input is not required; empty signs are turned into spaces, signs plus [+] and minus [-] are accepted).
L	Letter (from A till Z or from A till Z, input is required).
?	Letter (from A till Z or from A till Z, input is not required).
A	Letter or figure (input is required).
a	Letter or figure (input is not required).
&	Any sign or space (input is required).
C	Any sign or space (input is not required).
. , : ; - /	Decimal spacer and spacers of thousands, values of dates and time.
<	Point to shit of all following signs to low register.
>	Point to shit of all following signs to upper register.
!	Point to completion of mask of input from the right to the left, not vice versa.

In the table given below you can see description of some masks of input and special symbols of the mask which point to what positions, in what quantity and of what type data may be input.

Description of the mask of input	**Examples of values**
(000) 000-0000	*(206) 555-0248*
(999) 999-9999!	*(206) 555-0248* *( ) 555-0248*
(000) AAA-AAAA	*(206) 555-TELE*
>L0L 0L0	*T2Ф 8M4*
00000-9999	*98115-* *98115-3007*
>L<??????????????	*Mary* *Ivan*
ISBN 0-&&&&&&&&&-0	*ISBN 1-55615-507-7* *ISBN 0-13-964262-5*
>LL00000-	*DB51392-0493*

Property DEFAULT VALUE helps to specify the default value that will appear in this field each time that a new record is added.

For example:

Name of the Field	**Examples DEFAULT VALUE**
City	*Dnipropetrovsk*
Country	*Ukraine*
DateofOrder	*Date()*

Property VALIDATION RULE lets restrict input only with specified values.

For example:

Name of the Field	**Examples VALIDATION RULE**
Salary	*>1000 And <=5000* *or*

	Between 1000 and 5000
DateOfBirth	*(Year(Date())-Year([DateOfBirth]))>=18*
DateOfOrder	*>= #01/01/2013# and <=Date()* *or* *Between #01/01/2013# and Date()*
DateOfOrder	*<=Date()*

Property VALIDATION TEXT lets comment a mistake in case of failure.

For example:

Name of the Field	**Examples VALIDATION TEXT**
Salary	*Salary must be within 1 till 5 thousand*
DataofBirth	*Age of colleagues must be greater than 18*
DateofOrder	*Date of Order mustn't be greater than 01.01.2013 and less than current date*
DateofOrder	*Date of Order mustn't be greater than the current date*

Workshop № 2

SEARCH AND FILTER OF THE DATA IN THE MS ACCESS 2010. CREATING THE QUERY TYPE: SELECT

1.3. Search And Filter of the Data in Datasheet View

Assignments:

1. Start the ACCESS.

2. Open the database **CONTACTS**.

3. In the table **Address_book** in the *DataSheet* mode:

– *Freeze* the fields *LastName* and *FirstName* (tab *Home* / group *Records* / list *More* / *Freeze fields*);

– *Hide* the field *Notes* (tab *Home* / group *Records* / list *More* / *Hide fields*);

– change the *Font*, *Font size*, *Gridlines* and *Alternative Row Color* of the table (tab *Home* / group *Text Formatting*);

– add the *Total* by the fields *Credit* and *MobilePhone* (tab *Home* / group *Records* / icon *Totals*). Use the appropriate functions – *Sum*, *Maximum*, *Minimum* and *Count*. Analyze the results.

4. *Find* the data about addressees (tab *Home* / group *Find* / icon *Find*), who:

– work in the corporation *Airit*;

– have a *Birthday today*;

– have an *Anniversary this year*.

5. Formulate and fulfill different variants of search of the *MobilePhone* of the addressees, in which you remember only:

– first *two* digits;

– *fourth* and *last* digits.

6. With help of the icon *Replace* (tab *Home* / group *Find* / icon *Replace*) make the following corrections of the data:

– change the *Surname* of the woman, who is married;

– change the C*ompanyName – Airit* to *Saturn.*

7. With the help of the *Selection filter* display data only about the addressees (tab *Home* / group *Sort* & *Filter* / icon *Selection*)**:**

– who live in *Ukraine.* Cancel of the Filter (tab *Home* / group *Sort* & *Filter* / icon *Toggle Filter*);

– who live out of *Ukraine.* Cancel of the Filter;

– whose *FirstName* begins with letter *"C".* Cancel of the Filter;

– with the *DateofBirth* between *01/01/80* and *01/01/90.* Cancel of the Filter;

– with the *DateofBirth* in *March*. Don't cancel of the Filter;

– with *Credit greater than or equal to* 1000. Analyze the results.

8. With the help of the *Filter* (tab *Home* / group *Sort* & *Filter* / icon *Filter*) display data only about the addressees:

– from *Dnipropetrovsk*, who work in the *Dnipropetrovsk's Alfred Nobel University*. Cancel of the Filter;

– who use the mobile operator – *KievStar.* Cancel of the Filter;

– with the *DateofBirth* next month. Cancel of the Filter;

– with the *DateofBirth* in *first quarter*. Don't cancel of the Filter;

– whose *LastName* does not contain letter *"a"*. Analyze the results.

9. With help of the *Filter by Form* (tab *Home* / group *Sort* & *Filter* / icon *Advanced / Filter by Form*) display data only about the addressees:

– from *Dnipropetrovsk* or *Kiev*. For applying the conditions of *Filter* use the tab *Home* / group *Sort* & *Filter* / icon *Advanced / Apply Filter & Sort*;

– from *Dnipropetrovsk* and of the *MTS* mobile operator*;*

– without *Credit* and without *MobilePhone*;

– which *LastName* contain letter *"o"* or with *Credit greater* than 10000;

– which *LastName* begins with letters *"A"* or *"B"* or "C".

10. Sort the records of the table **Address_book** in the following way: within the records about addressees of the same *Country* values of the *LastName* must be placed in a way of *ascending* (tab *Home* / group *Sort* & *Filter* / icon *Ascending*).

TESTING QUESTIONS

1. Enumerate instruments of the menu *Format*, available in the *Design view* mode.

2. Enumerate substitutional symbols that can be used in the commands *Find и Replace*.

3. Characterize differences in the usage of substitutional symbols **#, ?, ***. Give examples.

4. Characterize purposes of the additional search options – *Search Fields as Formatted* and *Match Case***.**

5. How can you use the command *Find* to search for data in the *whole table*, but not only in the *current field*?

6. Give the example of the usage of *Selection filter* in the **Address_book**.

7. In what case do we need to use the instrument *Filter by Form* to search for data in the table?

8. Characterize differences in the usage of criteria *AND* and *OR* while applying the *Filter by Form*.

9. Characterize peculiarities of the data sort simultaneously for *several* table fields.

ASSISTANCE

Substitutional signs	***Usage***	***Example***	***Found values***
*	Refer to any quantity of any symbols or digits	*55*	553-3486 123-5561
?	Refer to any one textual symbol	P?trov	Petrov Pitrov Putrov Patrov

Substitutional signs	***Usage***	***Example***	***Found values***
[ ]	Refer to any one symbol among ones in brackets	P[eua]trov	Petrov Putrov Patrov *but not* Pitrov
!	Refer to any one symbol except for ones in brackets	b[!a]ll	bill bell *but not* ball
-	Refer to any symbol of the range. It is important to point this range in a way of ascending (from A to Z, but not from Z to A)	b[a-c]d	bad bbd bcd
#	Refer to any digit	1#3	103 113 123

1.4. Creating the Select Queries

Assignments:

1. Start the **ACCESS**.
2. Open the database **CONTACTS**.
3. With help of the *Query Wizard* mode create the following *Simple* queries (tab *Create* / group *Queries* / icon *Query Wizard / Simple Query Wizard*):

- **Phone_Numbers** – which contain Detail information of the fields *LastName*, *FirstName* and *MobilePhone;*
- **Credits** – which contain Summary information about total (sum) value of the *Credits* of the addressees from every *Country*.

4. With the help of the *Query Design* mode create the following *Select Queries* without using calculated expressions (tab *Create* / group *Queries* / icon *Query Design*):

- **Ukraine_contacts** – which contains information only about the addressees from *Ukraine* (the field *Country* must not be shown). Change the part of the results of selection, which must be shown on the screen (tab *Design* / group *Query Setup* / *Return*);
- **Foreign_contacts** – which contains information only about *foreign* addressees, who have no *BusinessPhone;*
- **MTS_subscriber** – which contains information about the addressees, who live in *Dnipropetrovsk*, on the *Lenin Street* or *Kirov Street*, and who are subscribers *MTS* and have no *Credits*;
- **Credit_contact** – which contains information about the addressees, who work in *Dnipropetrovsk's Alfred Nobel University* or *Airit*, live not in *Dnipropetrovsk* and have *Credit* more then *10000*;
- **Range_ contact** – which contains information about the addressees, who work in the *Company*, the *CompanyName* of which begins with letters *"A" or* "*B*" or "*C*" or "*D*"; who live in *Kiev* or *Dnipropetrovsk* and were born in *April* (use *three* ways for creation the *Criteria*).

5. With the help of the *Query Design* (tab *Create* / group *Queries* / icon *Query Design*) create the following *Select Queries* with using calculated expressions (*for performing this item of the laboratory use <u>assistance</u> information)*:

- **Birthday_1** – which contains information about the addressees, whose *DateofBirth* falls on the second part of *October* (with using of 2 ways);
- **Birthday_2** – which contains information about the addressees, whose *DateofBirth* falls on the next 10 days (<u>without</u> accounting of moving to the next month);
- **Age** – which lets to calculate the age of addressees;
- **Birthday_3** – which contains information only about the addressees, whose *DateofBirth* falls on the last day of each winter's month;

- **Birthday_4** (*the task of higher level of complexity*) – which contains information about the addressees, whose *DateofBirth* falls on the next 20 days (with accounting of moving to the next month).

6. Add to the table **Address_book** information about new addressees and analyze the results of running your *Queries*.

TESTING QUESTIONS

1. What is meant under the *query* in Access? Name all possible types of queries.

2. What are the main characteristics of *Select Queries?*

3. Define peculiarities of creating queries with the help of *Query Wizard.*

4. Characterize elements of the window *Query Design*.

5. Give examples of the query with the condition of *exact mismatch* of values in a field.

6. Give examples of the query with the condition of inexact mismatch with the value of a field.

7. Give examples of the query with the condition of choosing a record by *the range of values*.

8. Give examples of the query, in which you used logical operators *OR* and *AND*.

9. Characterize peculiarities of creation of *calculated expressions* in queries. Give examples of such a query in the laboratory work.

ASSISTANCE

Functions of the category DATE/TIME:

⇒ ***DATE ()*** – returns a *Variant (Date)* containing the current system date.

⇒ ***DAY («date»)*** – returns a *Variant (Integer)* specifying a whole number between 1 and 31, inclusive, representing the day of the month.

⇒ ***MONTH («date»)*** – returns a *Variant (Integer)* specifying a whole number between 1 and 12, inclusive, representing the month of the year.

⇒ ***YEAR («date»)*** – returns a *Variant (Integer)* containing a whole number representing the year.

⇒ ***DAY (Date())*** – the current day

⇒ ***MONTH (Date())*** – the current month

⇒ ***YEAR (Date())* –** the current year

⇒ ***DATESERIAL* (*«year», «month», «day»*) –** returns a *Variant (Date)* for a specified year, month, and day.

The ***DATESERIAL*** function syntax has these named arguments:

Part	**Description**
year	Required; Integer. Number between 100 and 9999, inclusive, or a numeric expression.
month	Required; Integer. Any numeric expression.
day	Required; Integer. Any numeric expression.

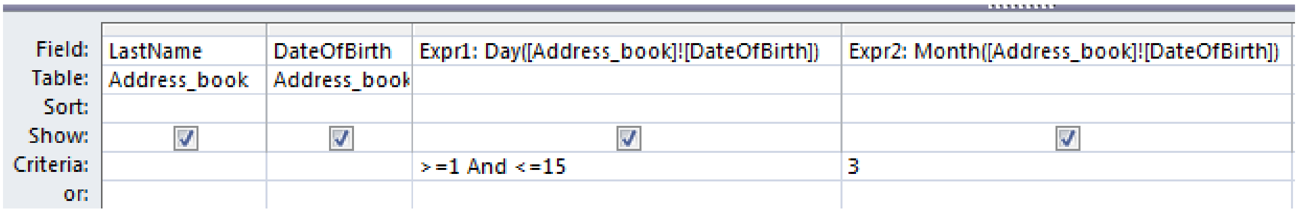
☑ **The examples of using function *DATESERIAL*:**

The addressees, whose DateofBirth falls on14st of March	**DateSerial(Year([Address_book]![DateOfBirth]),3,14)**
The addressees, whose DateofBirth falls on current day	**DateSerial(Year([Address_book]![DateOfBirth]), Month(Date()),Day(Date()))**
The last day of the current month	**DateSerial(Year(Date()), Month(Date()) + 1, 0)**
The last day of the next month	**DateSerial(Year(Date()), Month(Date()) + 2, 0)**

The first day of the previous month	**DateSerial(Year(Date()), Month(Date()) - 1,1)**
The last day of the previous month	**DateSerial(Year(Date()), Month(Date()),0)**

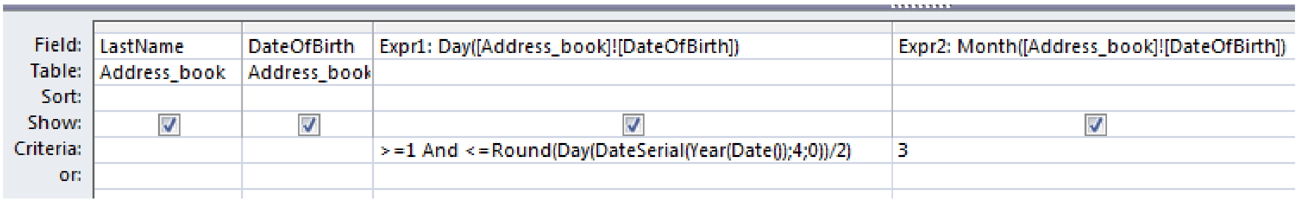
☑ **The example of query,** which shows all addressees whose *DateofBirth falls on the first part of March:*

Variant 1 (*without using function* ***DateSerial***)



Field:	LastName	DateOfBirth	Expr1: Day([Address_book]![DateOfBirth])	Expr2: Month([Address_book]![DateOfBirth])
Table:	Address_book	Address_book		
Sort:				
Show:	☑	☑	☑	☑
Criteria:			>=1 And <=15	3
or:				

Variant 2 (*with using function* ***DateSerial***)



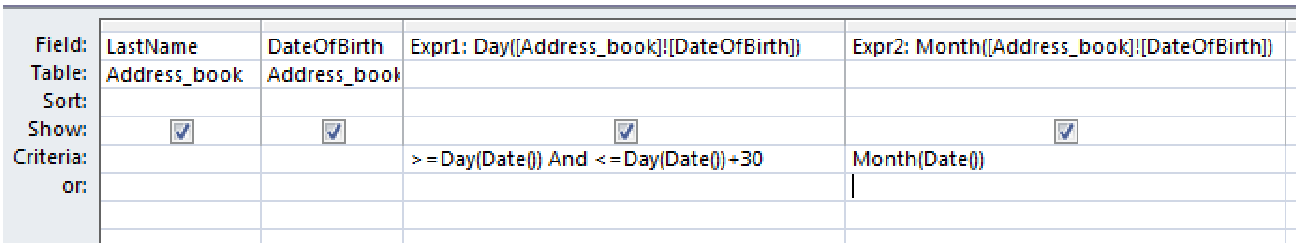
Field:	LastName	DateOfBirth	Expr1: Day([Address_book]![DateOfBirth])	Expr2: Month([Address_book]![DateOfBirth])
Table:	Address_book	Address_book		
Sort:				
Show:	☑	☑	☑	☑
Criteria:			>=1 And <=Round(Day(DateSerial(Year(Date());4;0))/2)	3
or:				

where

Round(Day(DateSerial(Year(Date()),4,0))/2) – *value of the half quantity of the days in March, which are rounded to the integer number*

☑ **The example of query,** *which shows all* addressees *whose DateofBirth falls on the next 30 days*

<u>Variant 1</u> (*<u>without</u> accounting of removing to the next month*):



Field:	LastName	DateOfBirth	Expr1: Day([Address_book]![DateOfBirth])	Expr2: Month([Address_book]![DateOfBirth])
Table:	Address_book	Address_book		
Sort:				
Show:	☑	☑	☑	☑
Criteria:			>=Day(Date()) And <=Day(Date())+30	Month(Date())
or:				

<u>Variant 2</u> (*<u>with</u> accounting of removing the found values to the next month*):

The algorithm of the selection contains the following steps:

✓ **add** the <u>first</u> auxiliary field – for calculating the quantity of days till the last day of the current month:

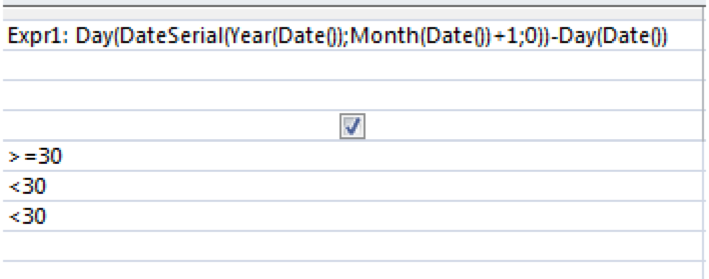
Expr1: Day(DateSerial(Year(Date()),Month(Date())+1,0))-Day(Date())

✓ **add** the <u>second</u> and the <u>third</u> auxiliary fields – for extracting the day and month from the field *DateOfBirth*:

Expr2: Day([Address_book]![DateOfBirth])

Expr3: Month([Address_book]![DateOfBirth])

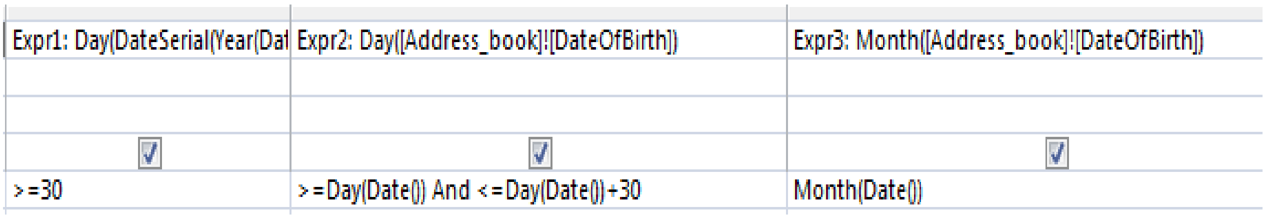
✓ **create** <u>THREE criteria for the first</u> auxiliary field – for analyzing quantity of days till last day of the current month – in the cases, when value of this field can be greater or equal, or less than 30:



Expr1: Day(DateSerial(Year(Date());Month(Date())+1;0))-Day(Date())
☑
>=30
<30
<30

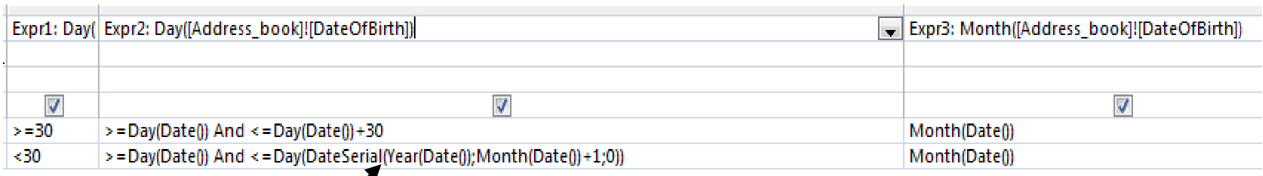
✓ **create** THREE groups of criteria for the second and third auxiliary fields:

- the **first** group – for situations, when quantity of *days till the last day of the current month* is greater or equal 30:



Expr1: Day(DateSerial(Year(Dat	Expr2: Day([Address_book]![DateOfBirth])	Expr3: Month([Address_book]![DateOfBirth])
☑	☑	☑
>=30	>=Day(Date()) And <=Day(Date())+30	Month(Date())

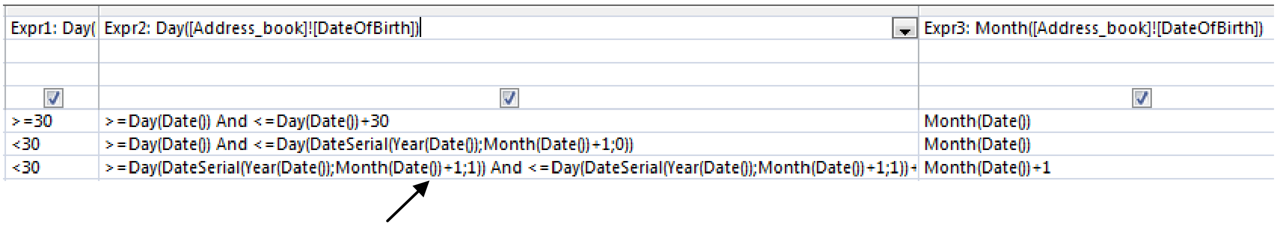
- the **second** group – for situations, when quantity of *days till the last day of the current month* is less than 30 and for selection, in this case, the part of specified in the *query quantity of days (30),* which is within the *current month:*



Expr1: Day(	Expr2: Day([Address_book]![DateOfBirth])	Expr3: Month([Address_book]![DateOfBirth])
☑	☑	☑
>=30	>=Day(Date()) And <=Day(Date())+30	Month(Date())
<30	>=Day(Date()) And <=Day(DateSerial(Year(Date());Month(Date())+1;0))	Month(Date())

from current day till the last day of the current month

- the **third** group – for situations, when quantity of *days till the last day of the current month* is less than 30 and for selection, in this case, the remainder – the difference between specified in the *query quantity of days (30)* and quantity of *days till the last day of the current month* – which *move from the current month to the next one:*



Expr1: Day(	Expr2: Day([Address_book]![DateOfBirth])	Expr3: Month([Address_book]![DateOfBirth])
☑	☑	☑
>=30	>=Day(Date()) And <=Day(Date())+30	Month(Date())
<30	>=Day(Date()) And <=Day(DateSerial(Year(Date());Month(Date())+1;0))	Month(Date())
<30	>=Day(DateSerial(Year(Date());Month(Date())+1;1)) And <=Day(DateSerial(Year(Date());Month(Date())+1;1))+	Month(Date())+1

>=Day(DateSerial(Year(Date()),Month(Date())+1,1)) And <=Day(DateSerial(Year(Date()),Month(Date())+1,1))+(30-(Day(DateSerial(Year(Date()),Month(Date())+1,0))-Day(Date())))

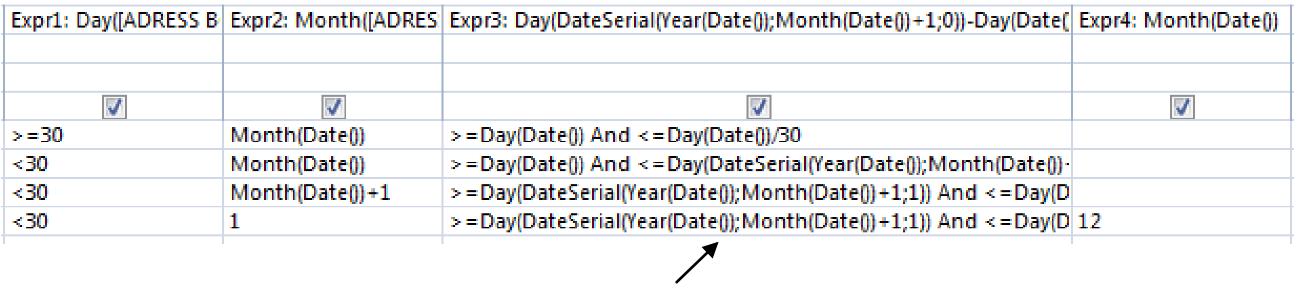
where

(30-(Day(DateSerial(Year(Date()),Month(Date())+1,0))-Day(Date()))) – the difference between specified in the *query quantity of days (30)* and quantity of *days till the last day of the current month*

✓ **add** the <u>fourth</u> auxiliary field – for situation, when the current month is *December*:

Expr1: Month(Date())

✓ **create** <u>for groups of criteria for the all four </u>auxiliary fields for situations, when quantity of *days till the last day of the current month* is less than 30 and for selection, in this case, the remainder – the difference between specified in the *query quantity of days (30)* and quantity of *days till the last day of the current month* – which *move from December to January*:

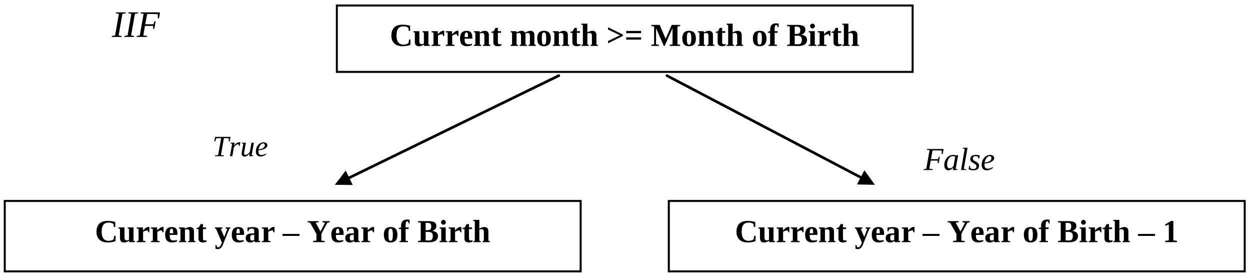


Expr1: Day([ADRESS B	Expr2: Month([ADRES	Expr3: Day(DateSerial(Year(Date());Month(Date())+1;0))-Day(Date(	Expr4: Month(Date())
☑	☑	☑	☑
>=30	Month(Date())	>=Day(Date()) And <=Day(Date())/30	
<30	Month(Date())	>=Day(Date()) And <=Day(DateSerial(Year(Date());Month(Date())-	
<30	Month(Date())+1	>=Day(DateSerial(Year(Date());Month(Date())+1;1)) And <=Day(D	
<30	1	>=Day(DateSerial(Year(Date());Month(Date())+1;1)) And <=Day(D	12

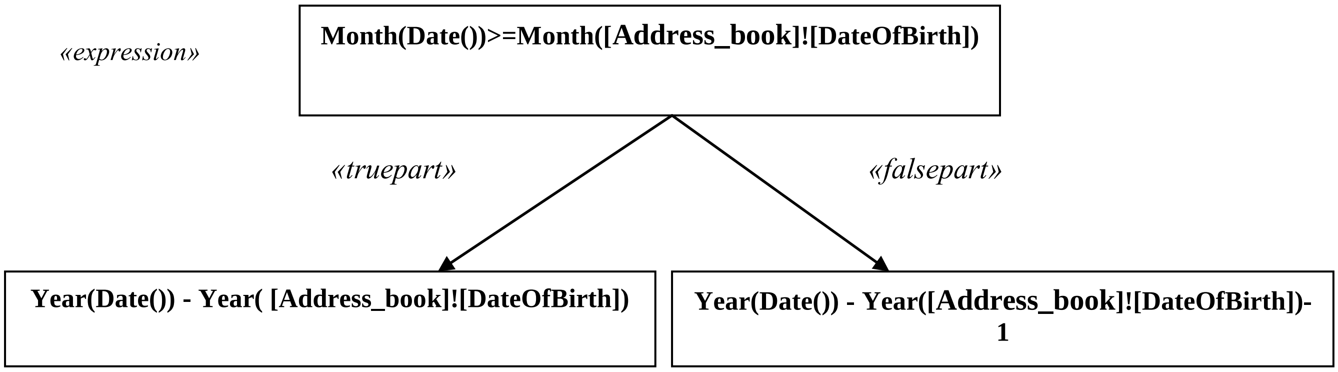
>=Day(DateSerial(Year(Date()),Month(Date())+1,1)) And <=Day(DateSerial(Year(Date()),Month(Date())+1,1))+(30-(Day(DateSerial(Year(Date()),Month(Date())+1,0))-Day(Date())))

The **example** of query**,** *which* show the ***AGE*** of addressees.

The algorithm of the age calculation (depending on the current month) has the following view:


IIF
Current month >= Month of Birth
True
False
Current year – Year of Birth
Current year – Year of Birth – 1

With the usage of functions ***IIF*** and functions of the group *Date/Time*, this algorithm will have the following view:


«expression»
Month(Date())>=Month([Address_book]![DateOfBirth])
«truepart»
«falsepart»
Year(Date()) - Year([Address_book]![DateOfBirth])
Year(Date()) - Year([Address_book]![DateOfBirth])-1

Workshop № 3

CREATING THE SUBDATASHEETS IN THE MS ACCESS 2010. USING THE FORMS FOR INPUTTING AND EDITING DATA. SUBFORMS. NAVIGATION FORMS

2.1. Creating the Subdatasheets

Assignments:

1. Start the ACCESS.
2. Open the database **CONTACTS**.
3. Project the *Subdatasheets*. To do that:

3.1. Create two reference tables: **Country** – including one text field *Country*; **City** – including one text field *City;* **Company_Name** – including one text field *CompanyName.*

3.2. Set *Fields Property* for the data fields of the tables – **Indexed: Yes (No duplicates)**.

3.3. Open the tables **Country**, **City**, **Company_Name** and **Address_book** in the *Datasheet View.*

3.4. Do the sequential copying of the content of the fields *Country, City* and *CompanyName* from the table **Address_book** into the appropriate created assistant tables (*Copy / Paste*). Pay attention to the messages about the impossibility of adding the *duplicate value* to these fields.

3.5. For each created assistant table set the *Subdatasheet* – the table **Address_book.** Choose the fields *Country, City* and *CompanyName* as the *Link Child Fields* and *Link Master Fields* (open the table **Country** (or **City**, or **Company_Name** correspondently) in *Datasheet View / Ribbon / tab Home / group Records / icon More / Subdatasheet / Subdatasheet).*

3.6. In each assistant table switch on and off the option of showing the *subrecords* in the table **Address_book,** using the signs ⊞ and ⊟ or the modes *Ribbon / tab Home / group Records / icon More / Subdatasheet / Expand All* and *Collapse All).*

3.7. In the table **Address_book** sequently create *Combo Boxes* for the fields *Country, City* and *CompanyName*, using the tables **Country, City** and **Company_Name** as the *Row Source* (*Design View* / *Field Properties* / tab *Lookup* / *Display Control – Combo Box / Row Source –* correspondingly tables **Country, City** and **Company_Name).**

3.8. Check the work of the created *Combo Boxes* in the *Datasheet View* by means creating a record about a new addressee.

4. Watch the structure of the projected *Subdatasheets* in the mode *Relationships* (*Ribbon* / tab *Database Tools* / group *Relationships* / icon *Relationships;* add the

Database tables with the help of the mode *Show table* – contextual tab *Relationship Tools* / *Show table*). Close the window *Relationships* .

2.2. Creating the Forms

A. Continuous Forms

1. For the table **Address_book** create the *Continuous Form* **Address_Continuous**, that shows multiple records at once (select the table **Address_book** and choose *Ribbon* / tab *Create* / group *Forms* / icon *More Forms* / *Multiple items*). Analyze the *View* of this form in the *Layout View* (by default). Switch to the *Form View* (tab *Home* / group *View / Form View).*

2. Switch to the *Design View*:

2.1. Change the *height* of the *Text Boxes*, which show the values in the fields of the form.

2.2. Chose the appropriate *Themes*, *Font* and *Colors* of *Form (Ribbon /* contextual tab *Design* / group *Themes).*

2.3. Add the appropriate picture as the *Logo* (*Ribbon* / contextual tab *Design* / group *Header / Footer / Logo*).

2.4. Set the mode of changing the picture size – *Zoom* (*Ribbon* / contextual tab *Design* / group *Tools* / *Property Sheet* / Navigation Pane *Property Sheet* / tab *Format / Size Mode: Zoom*);

3. Switch to the *Layout View:*

3.1. Change the *width* of the tables, which show the values in the fields of the form.

3.2. Set *Condition formatting* (*Ribbon* / contextual tab *Format* / group *Control Formatting / Condition formatting / New rule*):

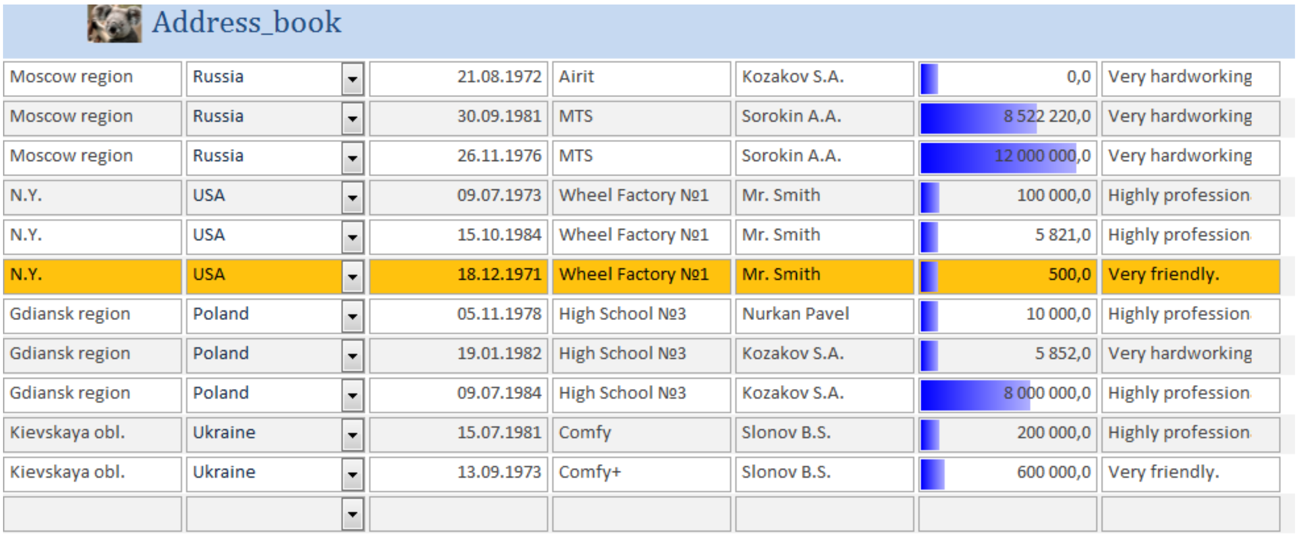
⇒ for the field *LastName* – the condition of highlighting the font (background) with another color when it is focused: **Select a rule type**: *Check values in the* *current record or use an expression*; **Format only cells where the**: *Field has focus*; choose the necessary type of *Font* and *Background Color*;

⇒ for all the form fields – the condition of highlighting the background with another color for the addressees, whose birthday is in the current month (preliminary select all the columns): **Select a rule type**: *Check values in the current record or use an expression*; **Format only cells where the**: *Expression Is: Month(date())=Month([DateOfBirth]);* choose the necessary type of *Font* and *Background Color*;

⇒ for the field *Credit* – the condition of highlighting the field value using the bar of different length (according to the relevant value of the number in this field): **Select a rule type**: *Compare to the other records*; **Shortest Bar**: *Type*: *Lowest value*; **Longest Bar**: *Type*: *Highest value*; choose the necessary *Bar color*.

3.3. Open the *Property* of *Form* and make sure that the mode is set to reflect the form – *Continuous Form* (*Ribbon* / contextual tab *Design* / group *Tools* / *Property Sheet* / Navigation Pane *Property Sheet* / Selection type: *Form* / tab *Format* / Default mode*: Continuous Form*).

4. Switch to the *Form View*. Analyze the advantages and disadvantages of the given method of reflecting the form fields.



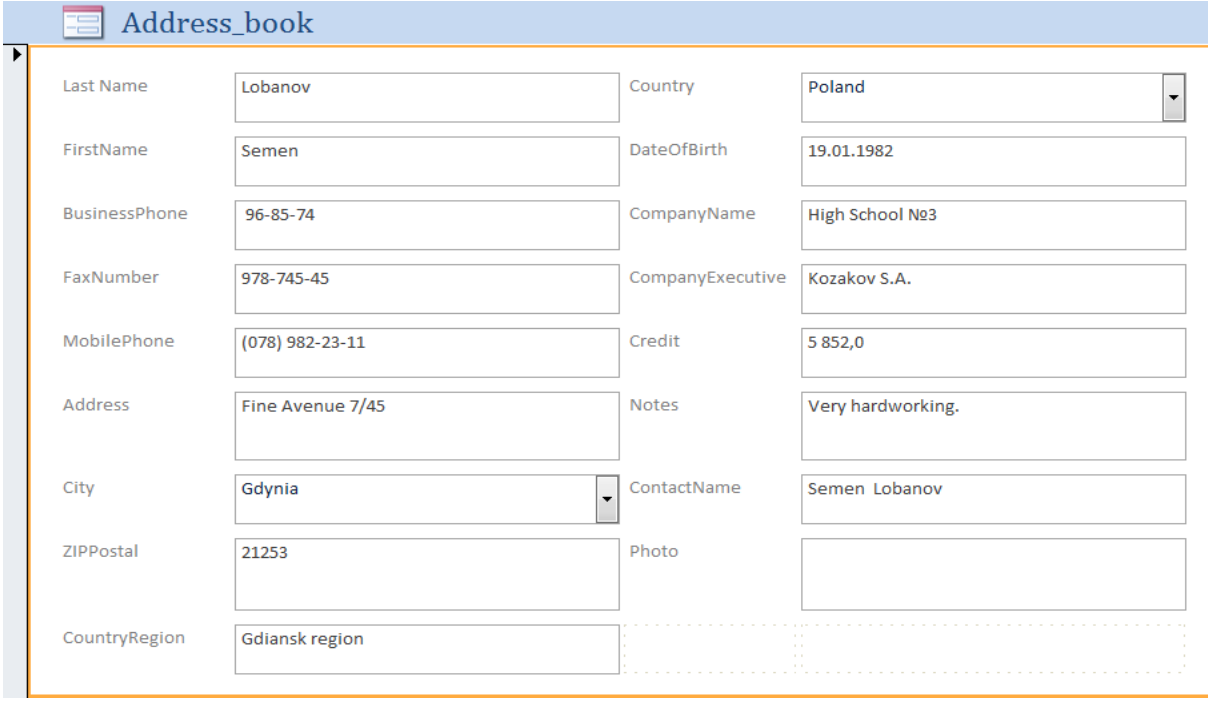
Address_book

Moscow region	Russia	21.08.1972	Airit	Kozakov S.A.	0,0	Very hardworking
Moscow region	Russia	30.09.1981	MTS	Sorokin A.A.	8 522 220,0	Very hardworking
Moscow region	Russia	26.11.1976	MTS	Sorokin A.A.	12 000 000,0	Very hardworking
N.Y.	USA	09.07.1973	Wheel Factory №1	Mr. Smith	100 000,0	Highly profession
N.Y.	USA	15.10.1984	Wheel Factory №1	Mr. Smith	5 821,0	Highly profession
N.Y.	USA	18.12.1971	Wheel Factory №1	Mr. Smith	500,0	Very friendly.
Gdiansk region	Poland	05.11.1978	High School №3	Nurkan Pavel	10 000,0	Highly profession
Gdiansk region	Poland	19.01.1982	High School №3	Kozakov S.A.	5 852,0	Very hardworking
Gdiansk region	Poland	09.07.1984	High School №3	Kozakov S.A.	8 000 000,0	Highly profession
Kievskaya obl.	Ukraine	15.07.1981	Comfy	Slonov B.S.	200 000,0	Highly profession
Kievskaya obl.	Ukraine	13.09.1973	Comfy+	Slonov B.S.	600 000,0	Very friendly.

B. *Single and Split Forms*

1. Add the field *Photo* with the data type *Attachment* to the table **Address_book**.

2. For the table **Address_book** create the *Single Form* – **Address_Single**, that lets you enter information for one record at a time (select the table **Address_book /** *Ribbon* / tab *Create* / group *Forms* / *Form*). Analyze the *View* of this form in the *Layout View* (by default).



Address_book

Last Name	Lobanov	Country	Poland
FirstName	Semen	DateOfBirth	19.01.1982
BusinessPhone	96-85-74	CompanyName	High School №3
FaxNumber	978-745-45	CompanyExecutive	Kozakov S.A.
MobilePhone	(078) 982-23-11	Credit	5 852,0
Address	Fine Avenue 7/45	Notes	Very hardworking.
City	Gdynia	ContactName	Semen Lobanov
ZIPPostal	21253	Photo	
CountryRegion	Gdiansk region		

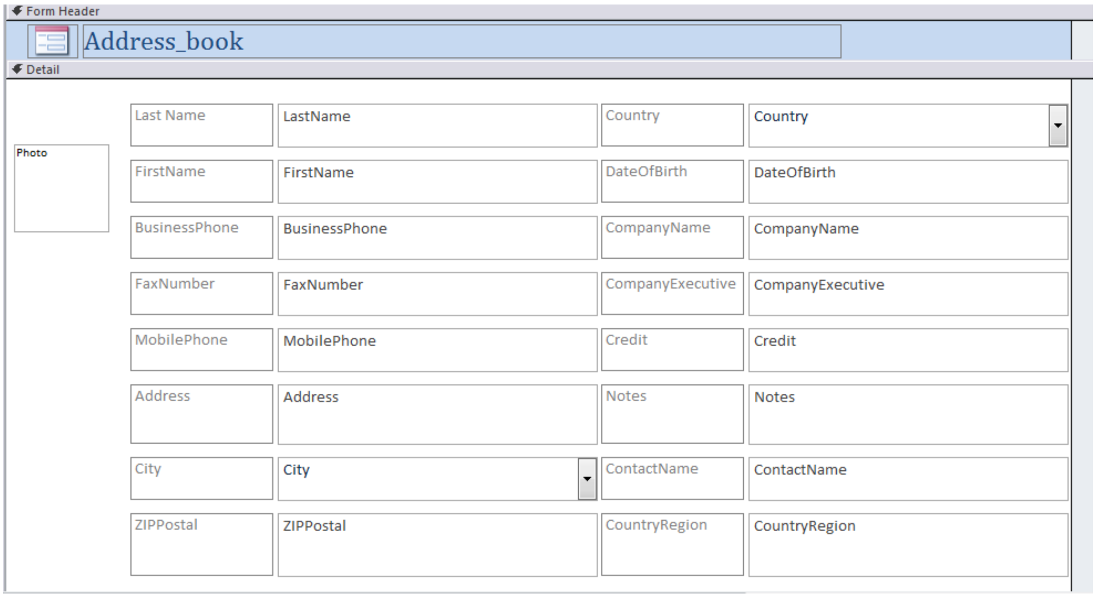
3. Switch to the *Design View*:

3.1. Delete *Label* of the field *Photo.*

3.2. Move block with all fields to the right.

3.3. Move the *Text Box* of the field *Photo* to the left corner of the *Detail* part of the *Form*.

3.4. Move the *Label* and *Text Box* of the field *CountryRegion* to the right column.



Address_book

Last Name	LastName	Country	Country
FirstName	FirstName	DateOfBirth	DateOfBirth
BusinessPhone	BusinessPhone	CompanyName	CompanyName
FaxNumber	FaxNumber	CompanyExecutive	CompanyExecutive
MobilePhone	MobilePhone	Credit	Credit
Address	Address	Notes	Notes
City	City	ContactName	ContactName
ZIPPostal	ZIPPostal	CountryRegion	CountryRegion

4. Switch to the *Form View*. With the help of the sign 📎 sequently add addressees' photos to each record.

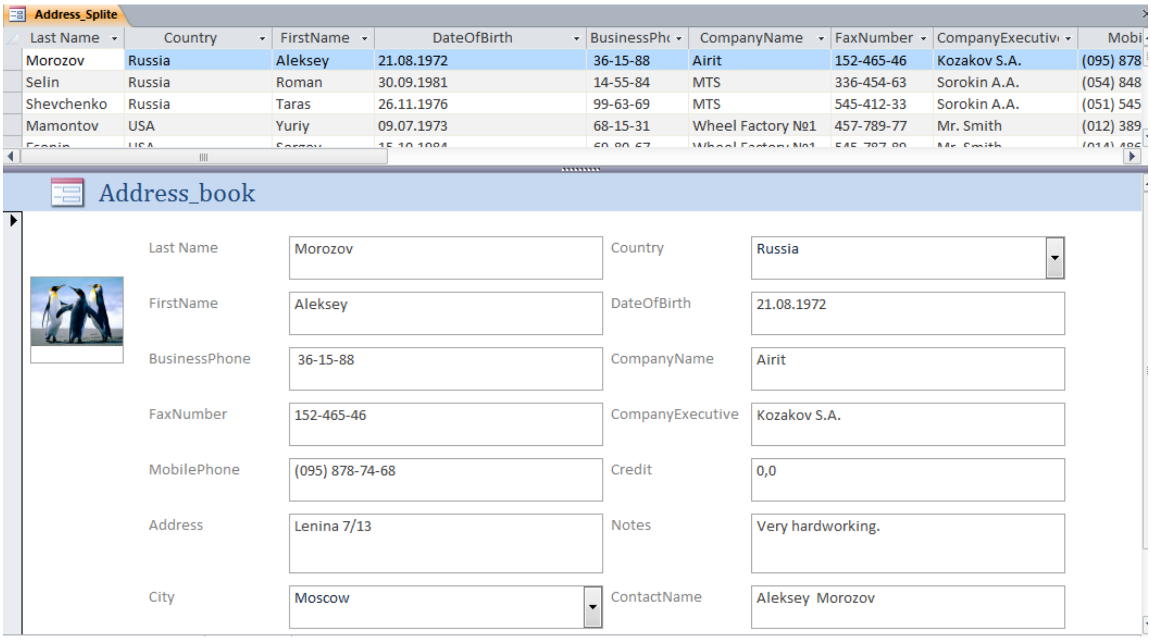
5. If needed change the size of the field *Photo* in the *Design View*.

6. Save and close the created form.

7. Create the copy of the form **Address_Single** and give it the following name – **Address_Split** *(Copy / Paste)***.**

8. Open the form **Address_Split** in the *Design View*.

9. Change the mode of the form reflection to the *Split Form* (*Ribbon* / contextual tab *Design* / group *Tools* / *Property Sheet* / Navigation Pane *Property Sheet* / Selection type: *Form* / tab *Format* / Default mode*: Split Form*).



Last Name	Country	FirstName	DateOfBirth	BusinessPh	CompanyName	FaxNumber	CompanyExecutiv	Mobi
Morozov	Russia	Aleksey	21.08.1972	36-15-88	Airit	152-465-46	Kozakov S.A.	(095) 878
Selin	Russia	Roman	30.09.1981	14-55-84	MTS	336-454-63	Sorokin A.A.	(054) 848
Shevchenko	Russia	Taras	26.11.1976	99-63-69	MTS	545-412-33	Sorokin A.A.	(051) 545
Mamontov	USA	Yuriy	09.07.1973	68-15-31	Wheel Factory №1	457-789-77	Mr. Smith	(012) 389
[illegible]	USA	Sergey	15.10.1984	69-80-67	Wheel Factory №1	545-787-89	Mr. Smith	(014) 486

Address_book

Last Name	Morozov	Country	Russia
FirstName	Aleksey	DateOfBirth	21.08.1972
BusinessPhone	36-15-88	CompanyName	Airit
FaxNumber	152-465-46	CompanyExecutive	Kozakov S.A.
MobilePhone	(095) 878-74-68	Credit	0,0
Address	Lenina 7/13	Notes	Very hardworking.
City	Moscow	ContactName	Aleksey Morozov

10. Switch to the *Form View* and analyze the advantages and disadvantages of the given method of reflecting the form values.

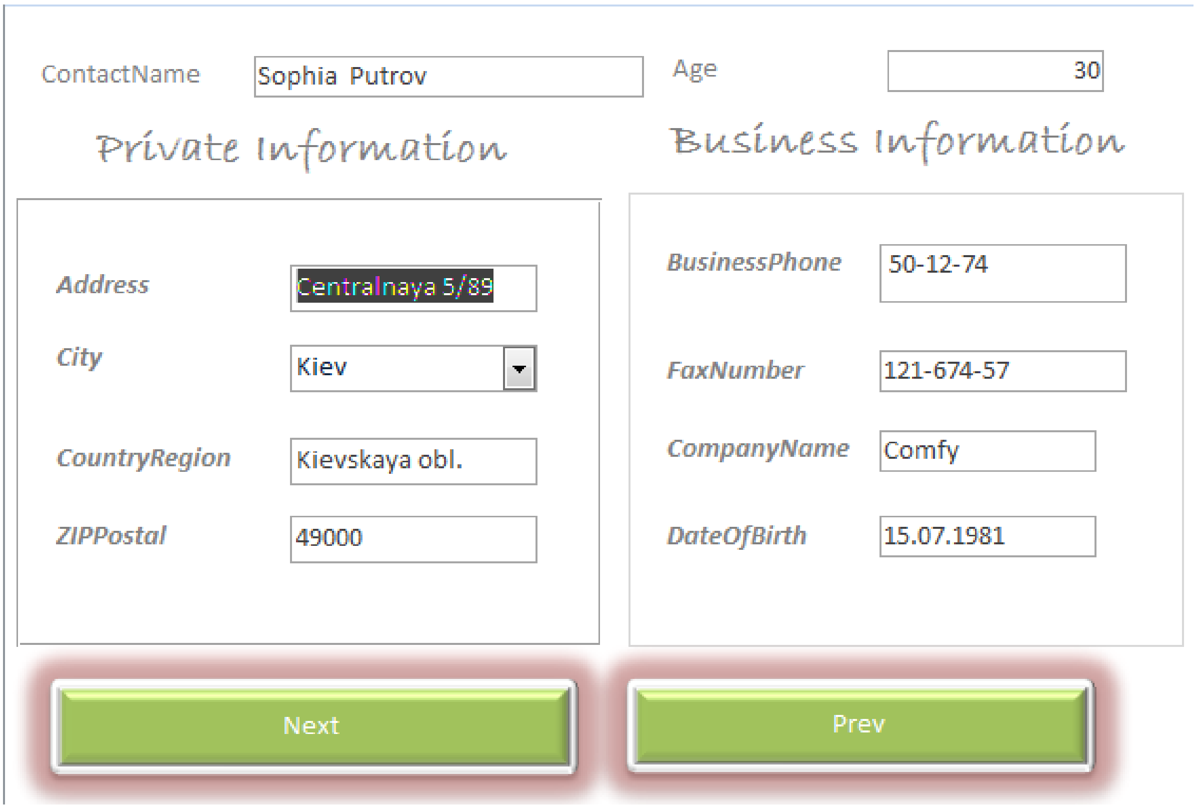
C. Blank Forms in the Base of Query

1. Create the *Blank Form* (*Ribbon* / tab *Create* / group *Forms* / icon *Blank Form*).
2. In the *Design View:*

2.1. Set the *Record Source* for this form – query **Ukraine_contacts** (*Ribbon /* contextual tab *Design* / group *Tools* / *Property Sheet* / Navigation Pane *Property Sheet /* Selection type: *Form* / tab *Data / Record Source:* **Ukraine_contacts**).

2.2. Add the areas *Form Header / Footer (*contextual menu *Form Header / Footer).*

2.3. Project and save this form with the name **Ukraine_Address** in accordance with the given example.



ContactName: Sophia Putrov

Age: 30

Private Information

Business Information

Address: Centralnaya 5/89

City: Kiev

CountryRegion: Kievskaya obl.

ZIPPostal: 49000

BusinessPhone: 50-12-74

FaxNumber: 121-674-57

CompanyName: Comfy

DateOfBirth: 15.07.1981

To do that:

⇒ with the help of *Field List* (*Ribbon /* contextual tab *Design* / group *Tools / Add Existing Field / Show only fields in current record source*) add fields from the query **Ukraine_contacts**;

⇒ create subscripts *Private Information* and *Business Information* with the help of the instrument **Controls** – *Label* (*Ribbon /* contextual tab *Design /* group *Controls*);

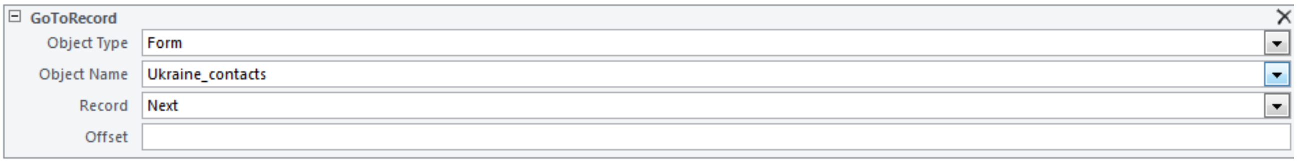
⇒ create rectangular frames with the help of the instrument **Controls** – *Rectangle* (*Ribbon /* contextual tab *Design* / group *Controls*);

⇒ add the appropriate picture as a *Logo* (*Ribbon /* contextual tab *Design /* group *Header / Footer / Logo*)*;*

⇒ display a *Title* in a form (*Ribbon /* contextual tab *Design* / group *Header / Footer / Title*)*;*

⇒ create *control buttons* to switch to the **Next** and **Previous** records. <u>The chain of actions:</u>

- add the element *Button* (*Ribbon /* contextual tab *Design* / group *Controls / Button*);
- open the window *Property Sheet* (*Ribbon /* contextual tab *Design* / group *Tools / Property Sheet /* Navigation Pane *Property Sheet /* Selection type: *Command);*
- rename the *Buttons* correspondently into *Next* and *Previous* (tab *Format / Caption*);
- choose the *Macro Builder* (tab *Event / On Click /*open the *Builder* … / *Macro Builder*);
- create the following *Macros:*



⊟ GoToRecord

Object Type	Form
Object Name	Ukraine_contacts
Record	Next
Offset	

- change the format of the created *Buttons (Ribbon /* contextual tab *Format* / group *Control Formatting / Shape Effects*);

⇒ switch off the form element *Navigation Buttons* (*Ribbon* / contextual tab *Design* / group *Tools / Property Sheet /* Navigation Pane *Property Sheet /* Selection type: *Form /* tab *Format / Navigation Buttons: No);*

⇒ create the calculating field *Age*. <u>The chain of actions</u>:

- add the element *Text Box* (*Ribbon* / contextual tab *Design* / group *Controls / Text Box*);
- rename the *Label Text №* into **Age** *(Ribbon* / contextual tab *Design* / group *Tools* / *Property Sheet* / Navigation Pane *Property Sheet* / Selection type: *Label /* tab *Format / Caption)***;**
- open the window *Property Sheet* for *Text Box* (*Ribbon /* contextual tab *Design* / group *Tools / Property Sheet /* Navigation Pane *Property Sheet* / Selection type: *Text Box /* tab *Data / Control Source* / open the *Builder* ... *);*
- create the *expression* (pay attention to the fact that the *expression* must not contain the name of the table):

IIF (Month(Date())>=Month([DateOfBirth]), Year(Date()) - Year([DateOfBirth]), Year(Date()) - Year([DateOfBirth])-1)

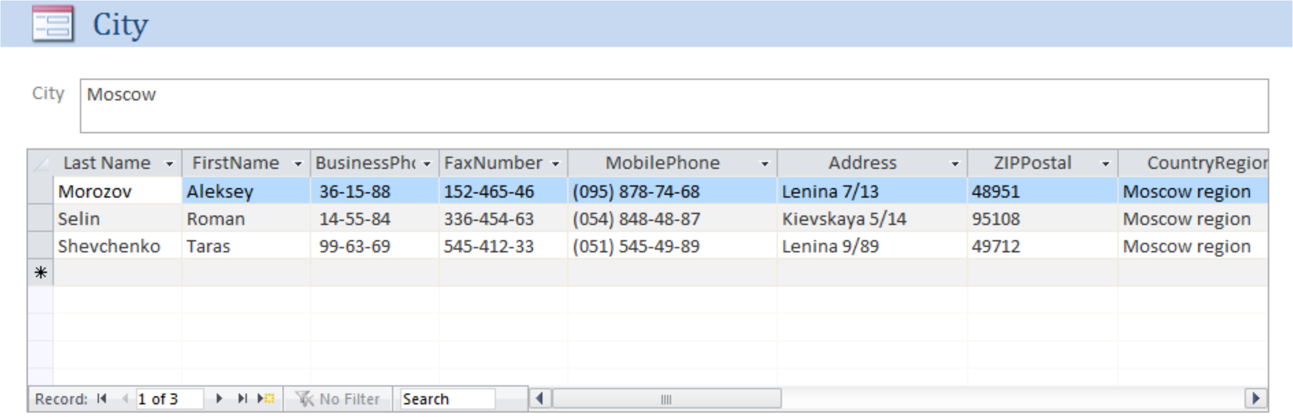
3. Switch to the *Form View* and analyze the results of data reflection in the form.

D. Subforms

1. Create the *Subform* for the table **City** (select the table **/** *Ribbon /* tab *Create* / group *Forms /* icon *Form)*. Rename this form into **City_sub.**

🕮 *Take into account the fact, that the creation of Subform is the result of preliminary set connections (Subdatasheet) between the tables **City** and **Address_book**.*

2. Analyze and modify the *View* of this form in the *Layout View* (by default). Switch to the *Form View (Ribbon /* contextual tab *Design* / group *View / Form View).*



City

City: Moscow

Last Name	FirstName	BusinessPh	FaxNumber	MobilePhone	Address	ZIPPostal	CountryRegior
Morozov	Aleksey	36-15-88	152-465-46	(095) 878-74-68	Lenina 7/13	48951	Moscow region
Selin	Roman	14-55-84	336-454-63	(054) 848-48-87	Kievskaya 5/14	95108	Moscow region
Shevchenko	Taras	99-63-69	545-412-33	(051) 545-49-89	Lenina 9/89	49712	Moscow region

3. Create *Subforms* for the tables **Country** and **Company_Name.** To do that in the *Design View* sequently:

3.1. Create the *Blank Form* (*Ribbon /* tab *Create /* group *Forms /* icon *Blank Form*).

3.2. Set the *Record Source* for this form – table **Country** or **Company_Name** correspondently (*Ribbon /* contextual tab *Design* / group *Tools / Property Sheet /* Navigation Pane *Property Sheet /* Selection type: *Form* / tab *Data /* Record Source **Country** or **Company_Name** correspondently).

3.3.Add the areas *Form Header / Footer* (contextual menu *Form Header / Footer).*

3.4.Add the fields *Country* or *CompanyName* correspondently to the areas *Detail* (*Ribbon /* contextual tab *Design* / group *Tools / Add Existing Field / Show only fields in current record source*).

3.5.Display a *Title* in a form (*Ribbon /* contextual tab *Design* / group *Header / Footer / Title*).

4. Analyze the results of reflecting the following form in the *Form View*.

5. Return to the *Design View* and add *Subforms* – **Addressess_Continious** to the created forms. To do that:

5.1. Add the element *Subform/ Subreport* (*Ribbon /* contextual tab *Design* / group *Controls / Subform/ Subreport*).

5.2. Open the window *Property Sheet* (*Ribbon* / contextual tab *Design* / group *Tools / Property Sheet /* Navigation Pane *Property Sheet /* Selection type: *Child №).*

5.3. Set the form **Addressess_Continious** as the *Source Object* (tab *Data / Source Object* / **Addressess_Continious**).

5.4. Correct the size of *Subform/ Subreport.*

6. Switch off the elements *Navigation Buttons* for the *Subform* **Addressess_Continious** (*Ribbon /* contextual tab *Design* / group *Tools / Property Sheet /* Navigation Pane *Property Sheet /* Selection type: *Form /* tab *Format / Navigation Buttons: No).*

7. Save these forms with names correspondently – **Country_sub** and **Company_Name_sub.**

8. Analyze the results of reflecting the created form in the *Form View.*

E. Navigation Forms

1. Create the *Blank Form* (*Ribbon /* tab *Create /* group *Forms /* icon *Blank Form*).

2. In the *Design View:*

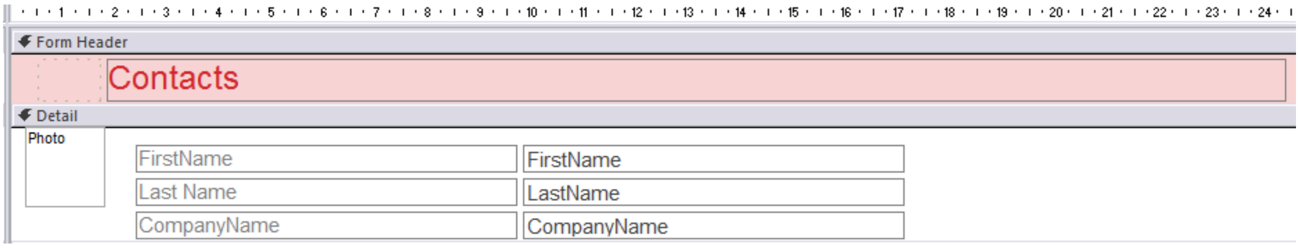
2.1. Set the *Record Source* for this form – table **Address_book** (*Ribbon /* contextual tab *Design* / group *Tools / Property Sheet /* Navigation Pane *Property Sheet /* Selection type: *Form* / tab *Data /* Record Source*:* **Address_book**).

2.2. Add areas *Form Header / Footer (*contextual menu *Form Header / Footer).*

2.3. Project and save this form with the name **Address_short** according to the given example:

🕮 Take into account the fact, that this form contains only a number of fields from the table **Address_book,** and the regime of form reflection is – *Continuous Form.*

⇒ model of the form in the *Design View:*



Form Header

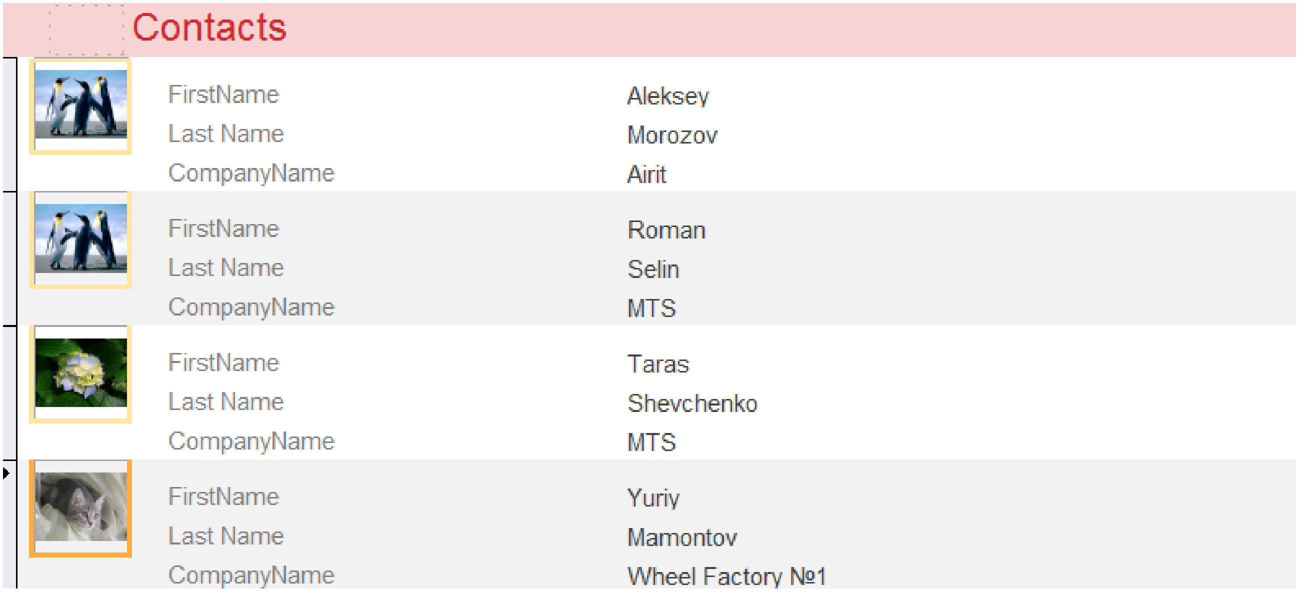
Contacts

Detail

Photo

FirstName	FirstName
Last Name	LastName
CompanyName	CompanyName

⇒ model of the form in the *Layout View:*



Contacts

FirstName	Aleksey
Last Name	Morozov
CompanyName	Airit
FirstName	Roman
Last Name	Selin
CompanyName	MTS
FirstName	Taras
Last Name	Shevchenko
CompanyName	MTS
FirstName	Yuriy
Last Name	Mamontov
CompanyName	Wheel Factory №1

2.4. Create the control button *Detail Information* to reflect the detail information about the addressee from the form **Address_Single**. Chain of actions:

⇒ add the element *Button* (*Ribbon* / contextual tab *Design* / group *Controls / Button*);

⇒ set the *Shape Fill* and *Shape Outline – Transparent* (*Ribbon* / contextual tab *Format* / group *Control Formatting*);

⇒ open the window *Property Sheet* (*Ribbon* / contextual tab *Design* / group *Tools / Property Sheet /* Navigation Pane *Property Sheet /* Selection type: *Command);*

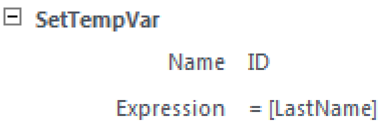
⇒ rename the *Buttons* into *Detail Information* (tab *Format / Caption*);

⇒ change the type of the *Cursor on Hover* (tab *Other / Cursor on Hover / Hyperlink hand);*

⇒ choose the *Macro Builder* for creating the *Macros* (tab *Event / On Click* /open the *Builder* … / *Macro Builder*);

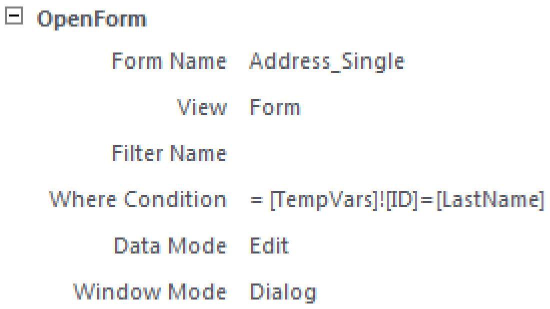
⇒ create the following *Macros*, which assumes a sequential implementation of the following actions*:*

a. Creation of the temporary variable **ID**:



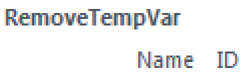
```
⊟ SetTempVar
        Name   ID
  Expression   = [LastName]
```

b. Opening of the form **Address_Single**:



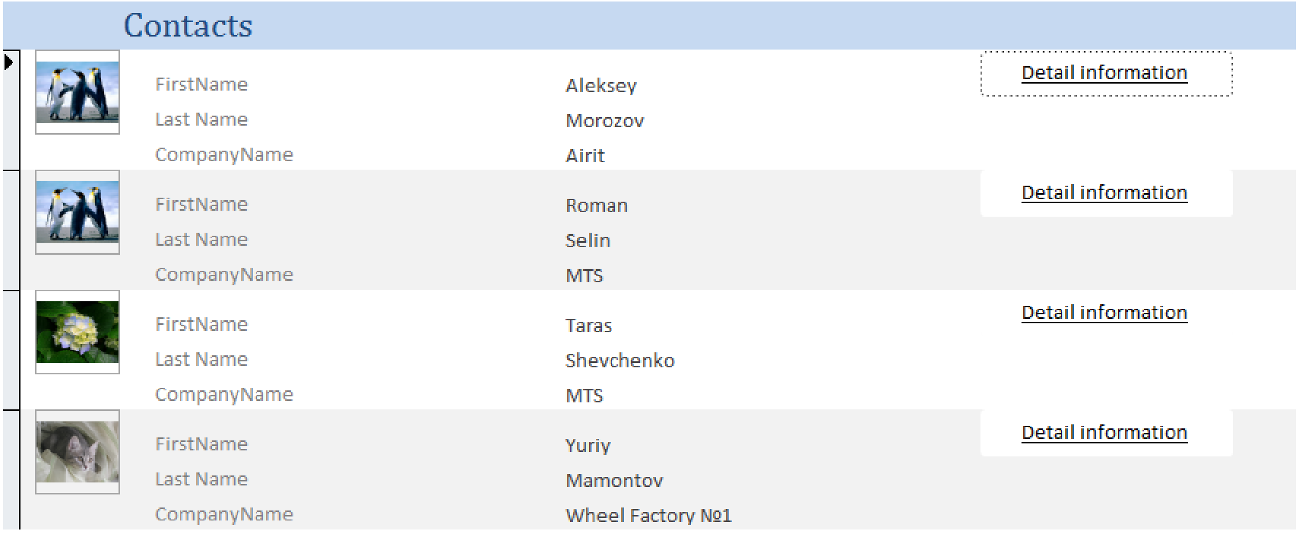
```
⊟ OpenForm
        Form Name   Address_Single
             View   Form
      Filter Name
  Where Condition   = [TempVars]![ID]=[LastName]
        Data Mode   Edit
      Window Mode   Dialog
```

c. Deleting of the temporary **ID**:



```
RemoveTempVar
        Name   ID
```

3. Analyze the results of this form reflection in the *Form View*.



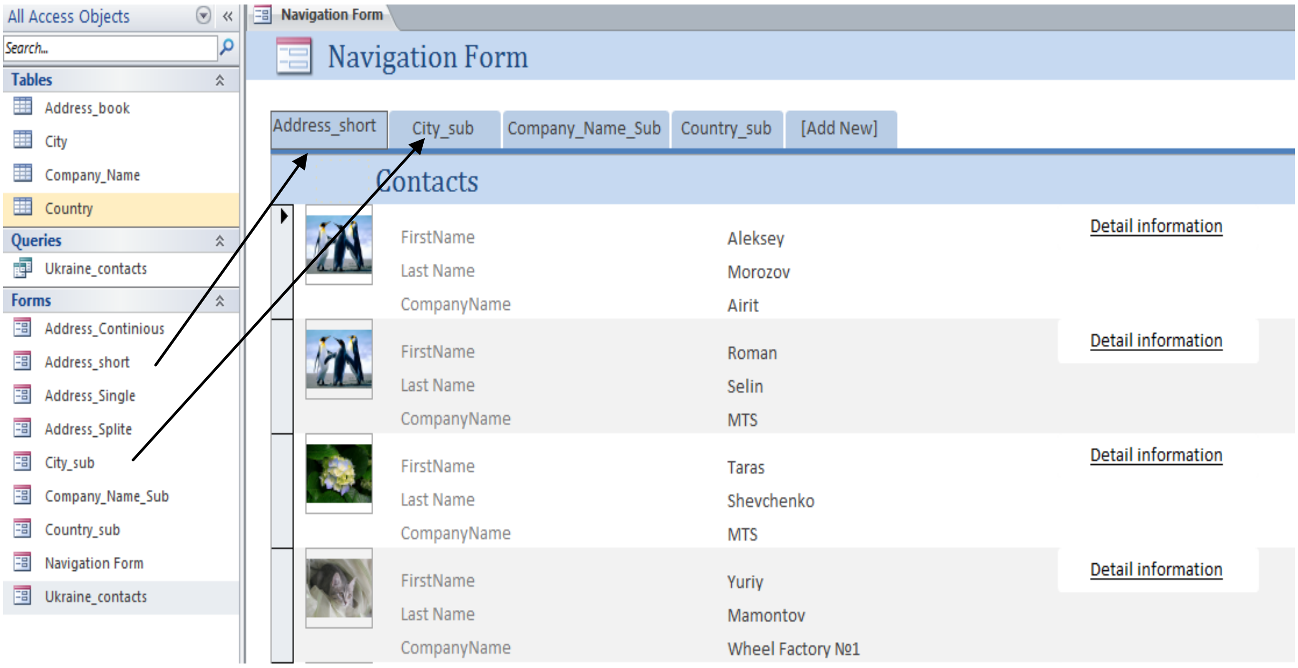
Contacts

FirstName Last Name CompanyName	Aleksey Morozov Airit	Detail information
FirstName Last Name CompanyName	Roman Selin MTS	Detail information
FirstName Last Name CompanyName	Taras Shevchenko MTS	Detail information
FirstName Last Name CompanyName	Yuriy Mamontov Wheel Factory №1	Detail information

4. Create *Navigation Form*, using the forms created before: **Address_short**, **City_sub, Country_sub** and **Company_Name_sub**. To do that:

4.1. Create the form *Navigation Form (Ribbon* / tab *Create* / group *Forms* / icon *Navigation / Horizontal Tab*).

4.2. In the *Layout View* move the names of above-mentioned forms from the list *Objects* of the database to the places of new tabs *Navigation Form* [Add New]:



Navigation Form

Contacts

FirstName Aleksey
Last Name Morozov
CompanyName Airit

FirstName Roman
Last Name Selin
CompanyName MTS

FirstName Taras
Last Name Shevchenko
CompanyName MTS

FirstName Yuriy
Last Name Mamontov
CompanyName Wheel Factory №1

5. Chose *Themes*, *Font* and *Colors* of *Form (Ribbon* / contextual tab *Design* / group *Themes)* for the created *Navigation Form.*

6. Add an appropriate picture as a *Logo* (*Ribbon* / contextual tab *Design* / group *Header / Footer / Logo*).

7. Display a *Title* in a form (*Ribbon* / contextual tab *Design* / group *Header / Footer / Title*).

8. Analyze the results of reflecting the created form in the *Form View*.

TESTING QUESTIONS

1. What is the way of setting *Subdatasheets* for the current table?
2. List the possible ways of displaying *Child Records* of the *Subdatasheets* for the current (*master*) table?
3. How can you create the *Display Control – Combo Box*?
4. How can you set the *Row Source Type* and *Row Source* for the *Combo Box*?
5. What is the way of attaching the file for the field of the type *Attachment*?
6. Enumerate all the possible ways (minimum 3) of creating the *Form*.

7. Enumerate all the possible *type* of the *Form*.

8. Enumerate all the possible *views* of the *Form*.

9. List the main differences between *Layout* and *Design Views*.

10. Give sequence of operations while working with *calculated field* of the *Form*. What is the difference between *calculated fields* if *Query* and *Form*?

11. How can you set the *Record Source* of the *Form*?

12. How can you display the *List of Existing Fields* in the *Layout* or *Design Views* of the *Form*?

13. Give the definition of the *Navigation Form*. Characterize peculiarities of creation this type of the *Form*.

14. List the possible ways of creating the *Subforms*.

15. Enumerate main *areas* and *elements* of the *Form*.

16. Enumerate main *Properties* of the *Form* and certain *Form's elements*.

Workshop № 4

RELATIONAL DATABASES PROJECTING IN THE MS ACCESS 2010. RELATIONSHIPS BETWEEN TABLES

4.1. Creating the Structure of the Database. Primary Key

Assignments:

1. Create a new database **INVENTORY** (tab **File / New/ Blank Database / Create)**.

2. In the *Design View* create 5 tables as follows:

Table 1: **EMPLOYEES**		Table 2: **SUPPLIERS**	
Field name	**Data type**	**Field name**	**Data type**
ID_Employees	*AutoNumber*	ID_Suppliers	*AutoNumber*
FirstName	*Text*	Company	*Text*
LastName	*Text*	FirstName	*Text*
JobTitle	*Text*	LastName	*Text*
E-mailAddress	*Text*	E-mailAddress	*Text*
BusinessPhone	*Text*	JobTitle	*Text*
MobilePhone	*Text*	BusinessPhone	*Text*
HomePhone	*Text*	MobilePhone	*Text*
Address	*Text*	FaxNumber	*Text*
City	*Text*	Address	*Text*
State	*Text*	City	*Text*
ZIP/PostalCode	*Text*	State	*Text*
Country	*Text*	Zip/PostalCode	*Text*
WebPage	*Hyperlink*	Country	*Text*
DateOfBirth	*Date/Time*	WebPage	*Hyperlink*
Salary	*Currency*	Notes	*Memo*
Notes	*Memo*	Attach	*Attachment*
Attach	*Attachment*		

Table 3: **TRANSACTION_TYPE**

Field name	**Data type**
ID_Transaction	*AutoNumber*
Description	*Text*
Add/Remove	*Text*

Table 4: **INVENTORY**

Field name	**Data type**
ID_Inventory	*AutoNumber*
Description	*Memo*
Category	*Text*
Location	*Text*
Supplier	*Number*
Manufacturer	*Text*
Model	*Text*
ReOrderLevel	*Number*
TargetStockLevel	*Number*
Attach	*Attachment*
Discontinued	*Yes/No*
Comments	*Memo*

Table 5: **INVENROTY_TRANSACTION**

Field name	**Data type**
ID_Inv_Tr	*AutoNumber*
InventoryItem	*Number*
Employee	*Number*
TransactionType	*Number*
Quantity	*Number*
CreatedDate	*Date/Time*
PONumber	*Text*
Comments	*Memo*

3. For all the fields, names of which consist of a few words, add the property **Caption**, where the text will include the current field's name with spaces between words.

4. Set the *Properties* of the tables **EMPLOYEES** and **SUPPLIERS**:

In the Design View

a) set fields *ID_Employees* and *ID_ Suppliers* as a **Primary Key** (tab **Design /** group **Tools /** icon **Primary Key** Primary Key);

b) for fields *LastName, FirstName* set the property **Required** – **Yes**;

c) for fields *BusinessPhone*, *MobilePhone*, *HomePhone ZIP/PostalCode*, *DateOfBirth*, set the property **Input Mask**;

In the Datasheet View

a) for the field *Country* set the property **Default Value** – *Ukraine*, and for the field *City* – *Dnepropetrovsk* (tab **Fields /** group **Properties**);

b) for the field *Salary* set (tab **Fields /** group **Formatting**):

– property **Format** – **Currency**;

– property **Decimal Places** – **2;**

– property **Validation Rule**, assuming that data in this field cannot be less than the value, that you specify themselves;

c) for the field *DateOfBirth* with the help of calculated expression set the property **Validation Rule**, assuming that employees' age cannot be less than 18 (tab **Fields /** group **Field Validation /** icon **Validation /** property **Validation Rule**);

d) for **Validation Rule,** which was set, formulae the **Validation Text** (tab **Fields /** group **Field Validation /** icon **Validation /** property **Validation Message**)**;**

e) for fields **Attach** add photos of employees (suppliers) *DatasheetView*.

5. In the table **TRANSACTION_TYPE** (*in the DesignView*)

a) set the field *ID_Transaction* as a **Primary Key** (tab **Design /** group **Tools /** icon **Primary Key**);

b) create *Combo Boxes* for the field *Add/Remove* (*Field Properties* / tab *Lookup* / *Display Control – Combo Box)* and set: *Row Source Type* – **Value List**; as a value of the property *Row Source* consecutively add values of the list **Addition; Removal** (using *List Items Editor* ...).

6. In the table **INVENTORY** (*in the DesignView*)**:**

a) the field *ID_Inventory* as a **Primary Key** (tab **Design /** group **Tools /** icon **Primary Key**);

b) for the field *Category* create **Combo Boxes** (*Design View* / *Field Properties* / tab *Lookup* / *Display Control – Combo Box)* and set: *Row Source Type* – **Value List**; as a value of the property *Row Source* consecutively add values of the list

Category 1; Category 2; Category 3 and etc. (using *List Items Editor* …); *Allow Value List Edits* – **Yes**;

c) for the field *Location* create **Combo Boxes** (*Design View* / *Field Properties* / tab *Lookup* / *Display Control – Combo Box)* and set: *Row Source Type* – **Value List**; as a value of the property *Row Source* consecutively add values of the list **Stock 1; Stock 2; Stock 3** and etc. (using *List Items Editor* …); *Allow Value List Edits* – **Yes**;

d) for the field *Supplier* create **Combo Boxes** (*Design View* / *Field Properties* / tab *Lookup* / *Display Control – Combo Box)* and set: *Row Source Type* – **Table/Query**; as a value of the property *Row Source* – table fields **Supplier** – *ID_Suppliers*, *Company*, *LastName*, *FirstName* (using *Query Builder* …); as a value of the property *Column Count* – 4; *Allow Value List Edits* – **Yes**.

e) for the whole table complete the **Validation Rule** (use **Property Sheet** on the toolbar of the ***Design View***) and set data control with the following conditions: the value of the field *ReorderLevel* should be less than the value of the field *TargetStockLevel*.

7. In the table **INVENTORY_TRANSACTIONS** (*in the DesignView*)**:**

a) Set the field *ID_Inv_Tr* as a **Primary Key** (tab **Design /** group **Tools /** icon **Primary Key**);

b) for fields *CreatedDate* and *PONumber* set the property **Input Mask** using substitution symbols for numbers and letters;

c) for the field *InventoryItem* create **Combo Boxes** (*Design View* / *Field Properties* / tab *Lookup* / *Display Control – Combo Box)* and set: *Row Source Type* – **Table/Query**; as a value of the property *Row Source* – table fields **Inventory**– *ID_Inventory*, *Description* (using *Query Builder* …); as a value of the property *Column Count* – 2; *Allow Value List Edits* – **Yes**;

d) for the field *Employee* create **Combo Boxes** (*Design View* / *Field Properties* / tab *Lookup* / *Display Control – Combo Box)* and set: *Row Source Type* – **Table/Query**; as a value of the property *Row Source* – table fields **Employees** –

ID_Employees, *FirstName*, *LastName* (using *Query Builder* …); as a value of the property *Column Count* – 3; *Allow Value List Edits* – **Yes**;

e) for the field *TransactionType* create **Combo Boxes** (*Design View* / *Field Properties* / tab *Lookup* / *Display Control – Combo Box)* and set: *Row Source Type* – **Table/Query**; as a value of the property *Row Source* – table fields **Transaction_Type** – *ID_ Transaction*, *Add/Remove* (using *Query Builder* …); as a value of the property *Column Count* – 2.

4.2. Creating Relationships Between Tables

A. Working in the Window Relationships

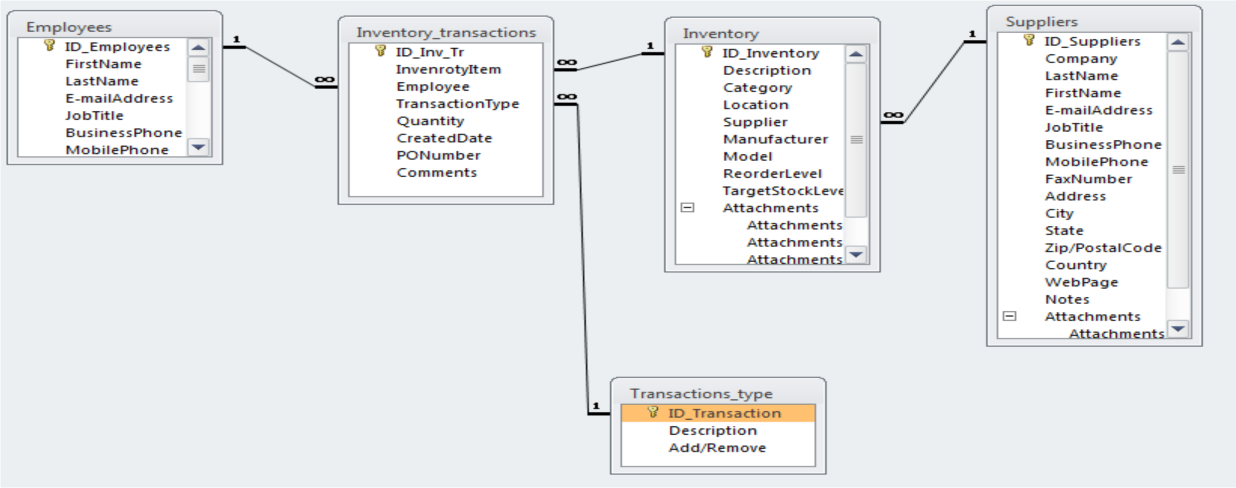
1. Create the structure of the projected *Relational database* in the mode *Relationships* (add the *Database tables* with the help of the mode *Show table* – contextual tab *Relationship Tools* / *Show table*):

a) open the window *Relationships (Ribbon* / tab *Database Tools* / group *Relationships /* icon *Relationships);*

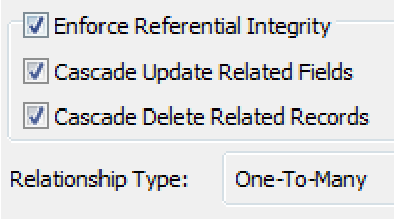
b) add the *Database tables* with the help of the mode *Show table* – contextual tab *Relationship Tools* / *Show table;*

c) create the **relationship** between appropriate table fields – *Primary* and *Foreign keys* (via dragging the name of one table to the name of the other table) in accordance with scheme on the Frame 1.

Frame 1



a) set the parameters of **Data integrity** in the database (pay attention to the fact that relationship type *One-To-Many* is identified automatically in case of following the rules of relationship setting – data type between related fields should be the same).



☑ Enforce Referential Integrity

☑ Cascade Update Related Fields

☑ Cascade Delete Related Records

Relationship Type: One-To-Many

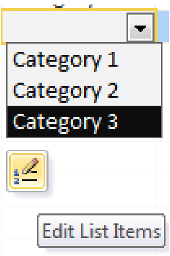
b) Close the window *Relationships* Close .

B. *Creating the Edit Forms*

1. Create *Single* forms for tables **EMPLOYEES, SUPPLIERS** and **TRANSACTION_TYPE.** Organize the entry of a few records into each of these tables.

2. In the *DatasheetView* try to enter a few records into the field **INVENTORY.** At that:

a) to enter values of the fields **Category** and **Location** use both the *Combo Box* values, entered before, and new values, which you add while you are entering records into the table .

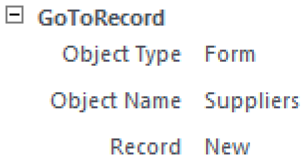


b) in order to leave the possibility of making changes in the *Combo Box* list of the field **Supplier,** previously:

– open the *Single* form **Suppliers;**

– in the window of form properties open the *Macro Builder* (*Ribbon /* contextual tab *Design* / group *Tools* / *Property Sheet* / Navigation Pane *Property Sheet /* Selection type: *Form* / tab *Event / On Load* /open the *Builder* ... / *Macro Builder*);

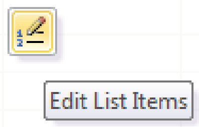
– create the macros, which allows opening the form **Supplier** in the mode of records entering:



⊟ GoToRecord

Object Type	Form
Object Name	Suppliers
Record	New

c) in the *Design View* of the table **INVENTORY** add the property *List Item Edit Form* – **Supplier** (*Field Properties* / tab *Lookup*);

d) enter the following record into the table **INVENTORY,** using information about a new *Supplier*, added into the *Combo Box* with the help of the form **Supplier**, that you created **.**

3. Create *Single* forms for tables **INVENTORY.**

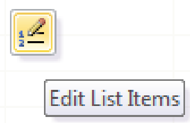
4. In order to leave the possibility of making changes in the *Combo Box* list of the fields **InventoryItem** and **Employee** in the table **INVENTORY_TRANSACTIONS,** previously:

a) open the *Single* form **Inventory (Employees);**

b) in the property window of each form open the *Macro Builder* (*Ribbon /* contextual tab *Design* / group *Tools / Property Sheet /* Navigation Pane *Property Sheet /* Selection type: *Form* / tab *Event / On Load* /open the *Builder* ... / *Macro Builder*);

c) create the macros, which allows opening the forms **Inventory** and **Employees** in the regime of new records entering;

d) in the *Design View* of the table **INVENTORY_TRANSACTIONS** add into the property *List Item Edit Form* for the fields **InventoryItem (Employee)** correspondingly the following name forms – **Inventory** (**Employees)** (*Field Properties* / tab *Lookup*).

5. Enter a few new records into the table **INVENTORY_TRANSACTIONS,** using information about new *Inventories* and *Employees*, added into the *Combo Box* with the help of the form, which you created .

TESTING QUESTIONS

1. For what do we use *Normalization* of relational databases?

2. Enumerate main requirements to the fields, between which relationships are set. Is it possible to set relationships between fields with *different names* or with *different data types*?

3. Enumerate possible *types* of *relationship* between tables in databases. Define types of relationship between the tables in your database.

4. How can we change the existing relationship between tables in a database?

5. How can we add (remove) a table in the *Relationships scheme*?

6. How can we open the *Design View* window for a table not leaving the window *Relationships scheme*?

7. Give definition to the meaning of *Data Integrity*. Enumerate restrictions, which work after the data integrity is defined.

8. For what do we use regimes of *Cascade Deleting* records and *Updating* of *Related Fields* in the relational database? In what task did you use these regimes?

9. What do we need to do, if we want information about the employee's transaction not to be deleted from the employee's list when the employee is removed from this list?

Workshop № 5

JOIN PROPERTIES AND JOIN TYPES. AGGREGATE, PARAMETER AND ACTION QUERIES IN THE MS ACCESS 2010

5.1. Join Properties and Join Types

Assignments:

1. Start the ACCESS.
2. Open the database **INVENTORY**.
3. Create **INNER-JOIN** queries (tab **Create /** group **Queries/ Query Design**) for the following conditions:

– the query **Category –** to receive information about all the *Transaction* of *Inventories* with <u>Category 1</u> or <u>Category 2</u>, carried out by *Employees* that do not have a *WebPage* and whose *LastName* begin with letter "*A*", but do not contain letters *b, c* and *d*;

– the query **Create_Date –** to receive information about all the *Transaction*, which were carried out during the previous week with the *Inventory* with the *Location* in *Stock 1* with *Discounting*, <u>OR</u> with *Location* in *Stock 3* in the *Quantity* from 100 till 10000;

– the query **City –** to receive information about all the *Transactions* in the previous month of the *Inventory, Supplier* of which lives in the same *City* as the *Employees* that is responsible for this transaction;

– the query **Tr_Type –** to receive information about *Employees* who <u>did not carry out</u> *Inventory Transactions* with the type **Addition**;

– the query **Actual_Quantity –** to receive information about the actual *Quantity* of *Transactions* in the current month, depending on *Transaction Type* – using the following algorithm:

If ***TransactionType* = *"Addition"***,

then ***Actual_Quantity*** *of transaction will be equal to* **positive** *value of Quantity,*

else ***Actual_Quantity*** *of transaction will be equal to* **negative** *value of Quantity***;**

– the query **Target_Stock_Level –** to receive information about *Inventories*, the *ReorderLevel* of which is less than a half of the value *TargetStockLevel*.

4. Create **OUTER-JOIN** queries (tab **Create /** group **Queries/ Query Design**) for the following conditions:

– the query **Outer_Inventory –** to receive information about *Inventories*, where *Inventory Transactions* **were not carried out***;*

– the query **Outer_Suppliers –** to receive information about *Suppliers,* whose *Inventories* **there were no** *Inventory Transactions;*

– the query **Outer_Employee –** to receive information about *Employees* that **did not carry out** *Inventory Transactions* (the field *EmployeesName* should be formed, using the values of fields **FirstName** and **LastName**).

5.2. Aggregate Queries

Create AGGREGATE QUERIES (tab **Create /** group **Queries/ Query Design**) for the following conditions:

– the query **Aggregate_Count –** to receive information about the quantity (function *Count*) of *Transactions* of each *Inventory*;

– the query **Aggregate_Month –** to receive information about the total value of *Quantity* (function *Sum*) of *Transactions* of each *Supplier* in the current month;

– the query **Aggregate_Max –** to receive information about the maximum value of *Quantity* (function *Max*) of removal *Transactions* for every *Employee* during two previous weeks;

– the query **Aggregate_ Actual_Quantity –** to receive information about the total value of *Actual_Quantity* in each *Inventory,* taking into account *Transaction Type* (see query **ActualQuantity**);

– the query **Aggregate_Actual_Quantity_1** (in the base of query **Aggregate_Actual_Stock**) – to receive only information about *Inventories,* for which the value of *Actual_Quantity* is less or equal to the value of the *ReorderLevel*.

5.3. Parameter Queries

Create PARAMETER QUERIES for the following conditions:

– to change the query **Aggregate_Actual_Stock1** in a way that it would be possible to use it in order to receive information about *Inventories* of any *Category,* where the value of **Actual Stock** is less or equal to the *ReorderLevel*.

– to change the query **Aggregate_Max** in a way that it would be possible to use it in order to receive information about the maximum value of *Quantity* of removal *Transaction* during two previous weeks – only for the *Employees,* whose *LastName* begin with any two letters that you set;

– to change the query **Aggregate_Month** in a way that it would be possible to use it in order to receive information about the quantity of *Transactions* for each *Inventory* during the current month, where the *Quantity* is in any limits that you set.

5.4. Action Queries

1. Create **Make-Table Queries** (tab **Create /** group **Queries/ Query Design**) for the following conditions:

– the query **Address** – to create a new table **Suppliers_mail**, which will contain information about required for the sending (*Company, FirstName, LastName, E-mail, Address* and *Category)* letter to *Suppliers*, whose *Inventories* had *Transactions* in the 2 recent *Months*;

☑ *Creating this query you should sequentially use different variants of creating the Criteria – functions MONTH, DATESERIAL, DATEADD.*

– the query **Actial_QTY_Month** – to create a new table **Turnover_per_Month,** which will contain information about the *Total* value of *Actual Stock* (taking into account *Transaction Type*) per each *Inventory* per previous *Months;*

☑ *Creating this query you should use function MONTHNAME for formatting the field Month.*

– the query **Previous_Year** – to create a new table **Inv_Trans_Prev_Year,** which will contain information about all *Transactions* during previous *Year;*

2. Create **Append Queries** (tab **Create /** group **Queries/ Query Design**) for the following conditions:

– the query **Actial_QTY_Current_Month** – to add to the table **Turnover_per_Month** information about the *Total* value of *Actual Stock* (taking into account *Transaction Type*) per each *Inventory* during the current *Month;*

– the query **Contacts** – to add from the table **Address_book** (database **INVENTORY**) to the table **EMPLOYEES** (database **CONTACTS**) information about a person, who lives in *Dnipropetrovs'k* and works in a particular *Company* (information about *Transaction Inventory* which you store in your database **INVENTORY**).

☑ *For creating the query* **Contacts***:*

*a) in the database **INVENTORY** temporary remove relationships between tables **EMPLOYEES** and **INVENTORY_TRANSACTION;***

*b) in the table **EMPLOYEES** temporary cancel the key field;*

*c) to create the query go to the database **CONTACTS**, which is the source of the information that you add to the table **EMPLOYEES;***

d) after running and completing the query restore the removed relationships and the key word for the table ***EMPLOYEES*** *in the database* ***INVENTORY.***

3. Create **Delete Queries** (tab **Create /** group **Queries/ Query Design**) for the following conditions:

– the query **Delete_Mail** – to delete from the table **Suppliers_mail** records about *Suppliers,* whose *Inventories* belong to the *Category 1*;

– the query **Delete_Previous_Year** – to delete from the table **INVENTORY_TRANSACTION** information about all *Transactions* during previous *Year. Check and analyze the results of this query in the main table* (**INVENTORY_TRANSACTION)** and the one related to it.

☑ *Before creating the query* **Delete_Previous_Year***:*

▪ *You have to check the relationships between related tables in database* ***INVENTORY****;*

▪ *You have to check if the Enforced the Referential Integrity for the relationships between your chosen tables have been switched on;*

▪ *You have to check if the Cascade Delete Related Records for the relationships have been switched on.*

4. Create **Update Queries** (tab **Create /** group **Queries/ Query Design**) for the following conditions:

– the query **Increase_Reorder** – to update information about **ReOrderLevel** for all **Inventories** of *Category 1* by means of its increasing the by 10%;

– the query **Update _Comments** – to update information about *Comments* about *Inventories*; assign to the given field the value of the field *WebPage* of the correspondent *Supplier.*

– correspondent *Supplier.*

TESTING QUESTIONS

1. Give the definition *of Inner-Join* and *Outer-Join* queries.
2. What is the way of changing the *Join Properties*?
3. What means if the window *Join Properties* terms *Left Table*, *Right Table*?
4. Enumerate the possible ways of creating the *Aggregate* queries.
5. What the peculiarities of adding fields to the *Aggregate* queries?
6. How to specify the condition for the field, which is used for calculation the *group* values?
7. Give the definition of *Parameter Queries*.
8. Enumerate the possible ways of creating the *Parameters*.
9. Enumerate all types of *Action Queries*.
10. What type of query you have used before *Action Queries*?
11. List the main purpose of using and peculiarities of creating *Make-Table Queries.*
12. List the main of using and peculiarities of creating of *Append Queries*.
13. List the main of using and peculiarities of creating of *Delete Queries*.
14. What the specificity of deleting the records in the database with switched on *Data integrity* parameters.
15. List the main of using and peculiarities of creating of *Update Queries*.

Workshop № 6

CROSSTAB QUERIES. REPORTS IN THE MS ACCESS 2010

6.1. Crosstab Queries

Assignments:

1. Start the ACCESS.
2. Open the database **INVENTORY**.
3. Create **CrossTab Queries** (tab **Create /** group **Queries/ Query Design**) for the following conditions:

3.1.The query **CrossTotal** – to receive information about *Total* value (<u>Value</u>) of *Actual Stock* (taking into account *Transaction Type*) per *Inventory* (<u>Row Heading)</u> per *Month* (<u>Column Heading</u>);

> ☑ *Creating this query you should use function FORMAT for formatting the field MONTH*

3.2.The query **CrossMax** – to receive information about maximum value of *Quantity* (<u>Value</u>) from *Transactions* per *Employee* per *Category* (<u>Row Headings</u>) per *Year* (<u>Column Heading</u>);

> ☑ *Creating this query you should use operator "&" for forming the field EMPLOYEES_NAME*

3.3.The query **CrossCount** – to receive information about <u>quantity</u> (<u>Value</u>) of <u>removal</u> *Transactions* (<u>not shown</u>), which value of *Quantity* less than 10000 units (<u>not shown</u>), per *Suppliers* (<u>Row Heading</u>) per *City* (<u>Column Heading</u>);

6.2. Reports

A. *Task 1*

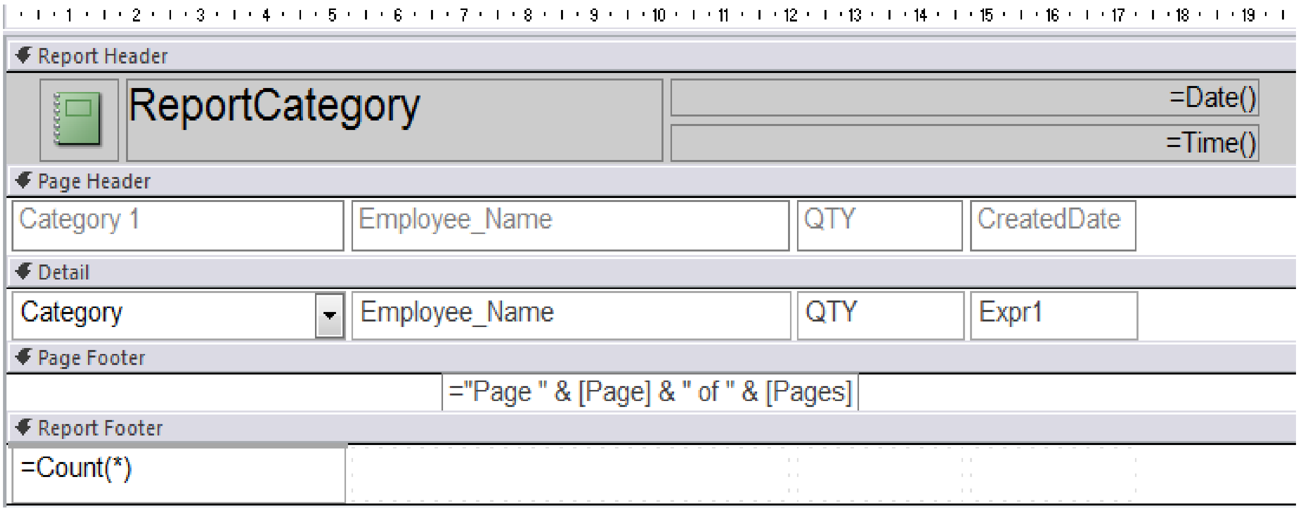
Assignments:

1. Create the query **ReportCaterogy** – to receive information about *Actual Quantity* (field *QTY*) of *Transaction* by *Inventories* (field *Category*), carried out by *Employees* (field *Employees_Name*) per *Date* (field *CreatedDate*).

☑ *Creating this query you should use functions IIF and operator "&"*

2. Create the draft version of report **Report_Caterogy** in the base of the query **ReportCaterogy** (select the query **ReportCaterogy** and choose *Ribbon* / tab *Create* / group *Reports* / icon *Report*).

3. Switch to the *Design View*:



Report Header

ReportCategory

=Date()

=Time()

Page Header

Category 1 | Employee_Name | QTY | CreatedDate

Detail

Category | Employee_Name | QTY | Expr1

Page Footer

="Page " & [Page] & " of " & [Pages]

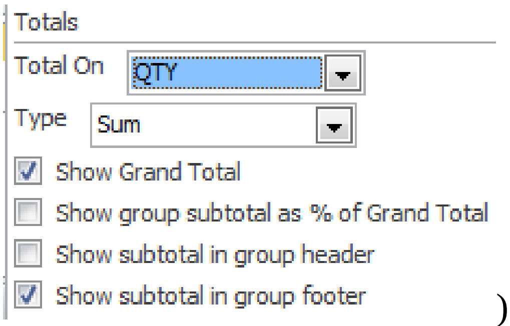
Report Footer

=Count(*)

3.1.Add a *Group* on *Category (Ribbon* / contextual tab *Design* / group *Grouping & Totals / Group & Sort).*

3.2.Remove *Header Section* and add *Footer Section* to this *Group* (area *Group, Sort and Total* / More ▶ / *Without a Header Section / With a Footer Section).*

3.3.Add *Totals* by QTY (area *Group, Sort and Total* / More ▶ / in the line with Category totaled ▾ select the next options).



Totals

Total On QTY ▾

Type Sum ▾

☑ Show Grand Total

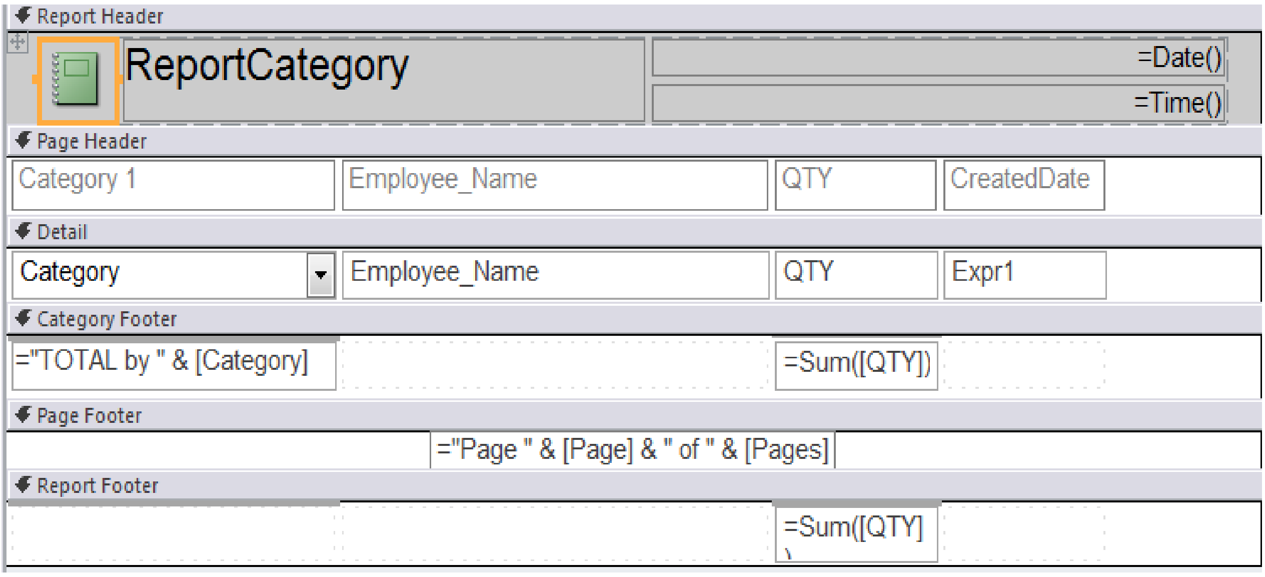
☐ Show group subtotal as % of Grand Total

☐ Show subtotal in group header

☑ Show subtotal in group footer

3.4.To the area *Category Footer* add the *Text box (Ribbon* / contextual tab *Design* / group *Controls / Text box)* and create the expression (*Ribbon* / contextual tab *Design* / group *Tools* / icon *Property Sheet* / Navigation Pane *Property Sheet* / tab *Data* / row *Control Source)*

="TOTAL by " & [Category]



Report Header

ReportCategory =Date() =Time()

Page Header

Category 1 | Employee_Name | QTY | CreatedDate

Detail

Category | Employee_Name | QTY | Expr1

Category Footer

="TOTAL by " & [Category] | =Sum([QTY])

Page Footer

="Page " & [Page] & " of " & [Pages]

Report Footer

=Sum([QTY]

4. Switch to the *Layout View*:

4.1.Switch off the option of repeating the duplicate values in the field *Category* (*Ribbon* / contextual tab *Design* / group *Tools* / icon *Property Sheet* / Navigation Pane *Property Sheet* / Selection type: *Category* / tab *Format* / row *Hide Duplicates* / *Yes*).

4.2.Set the format of reflecting the numbers in the field *QTY* to *Standard* (*Ribbon* / contextual tab *Design* / group *Tools* / icon *Property Sheet* / Navigation Pane *Property Sheet* / Selection type: *QTY* / tab *Format* / row *Format* / *Standard*).

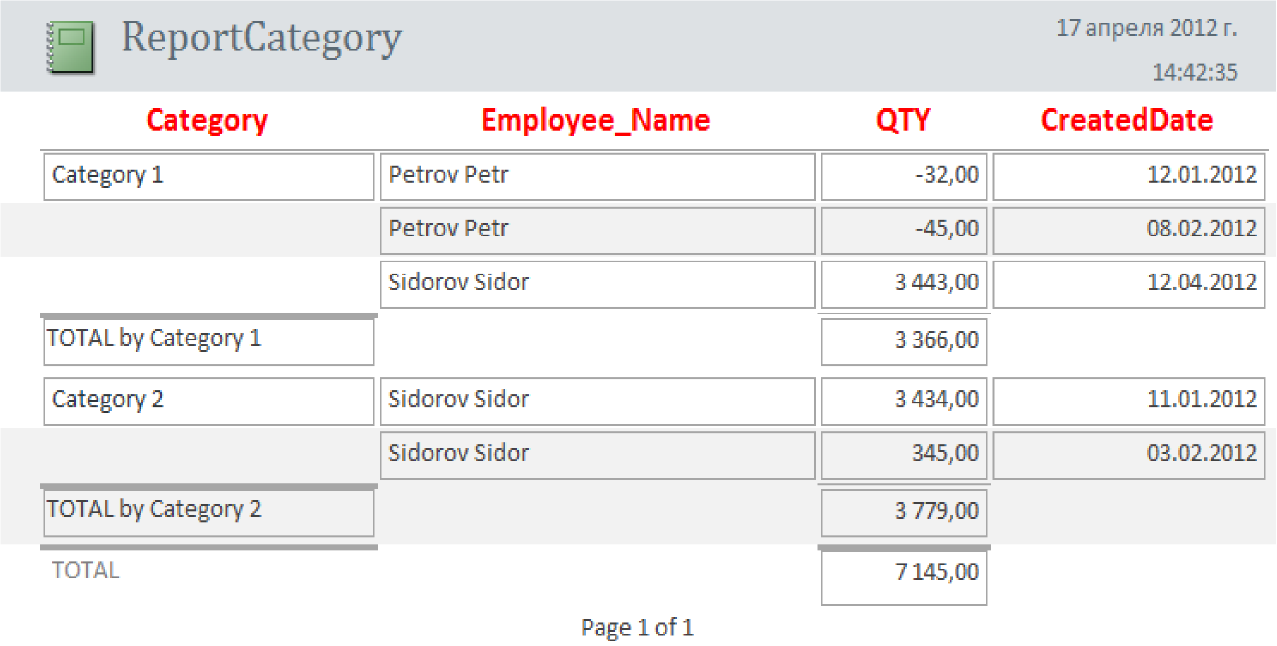
4.3.Set the format of displaying the information about current page Page 1 of 1 – *Print only (Ribbon /* contextual tab *Design /* group *Tools /* icon *Property Sheet /* Navigation Pane *Property Sheet /* Selection type: *Text_N /* tab *Format /* row *Display When / Print only).*

4.4.Chose the appropriate *Themes*, *Font* and *Colors* of *Form (Ribbon /* contextual tab *Design /* group *Themes).*

4.5.Add the appropriate picture as the *Logo* (*Ribbon /* contextual tab *Design /* group *Header / Footer / Logo*).

4.6.To the area *Report Footer* add the *Label (Ribbon /* contextual tab *Design /* group *Controls / Label)* and input the text “TOTAL”.

5. Switch to the *Report View* and *Print Preview*.



ReportCategory

17 апреля 2012 г.
14:42:35

Category	Employee_Name	QTY	CreatedDate
Category 1	Petrov Petr	-32,00	12.01.2012
	Petrov Petr	-45,00	08.02.2012
	Sidorov Sidor	3 443,00	12.04.2012
TOTAL by Category 1		3 366,00	
Category 2	Sidorov Sidor	3 434,00	11.01.2012
	Sidorov Sidor	345,00	03.02.2012
TOTAL by Category 2		3 779,00	
TOTAL		7 145,00	

Page 1 of 1

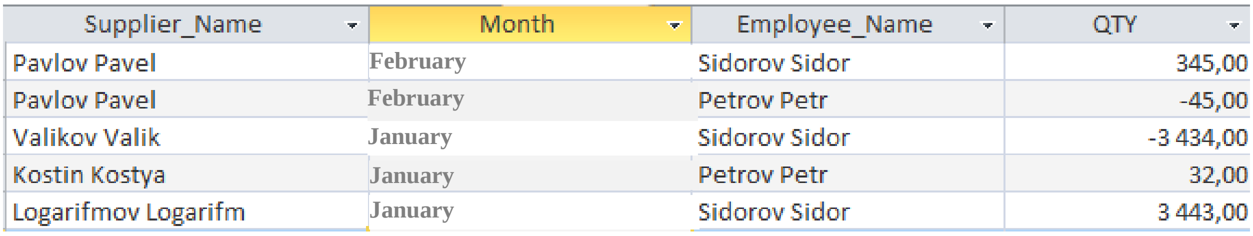
6. Save this Report.

B. *Task 2*

1. Create copy of query **ReportCaterogy** and change it in accordance with following conditions – to receive information about *Actual Quantity* (field QTY) of *Transaction* by *Supplier* (field *Supplier_Name*), carried out by *Employees* (field *Employees_Name*) per *Month* (field *Month_Name*). Save its query by name **ReportMonth.**

☑ *Creating this query you should use functions IIF and MONTHNAME*

For example, the result of running the query **ReportCaterogy:**



Supplier_Name	Month	Employee_Name	QTY
Pavlov Pavel	February	Sidorov Sidor	345,00
Pavlov Pavel	February	Petrov Petr	-45,00
Valikov Valik	January	Sidorov Sidor	-3 434,00
Kostin Kostya	January	Petrov Petr	32,00
Logarifmov Logarifm	January	Sidorov Sidor	3 443,00

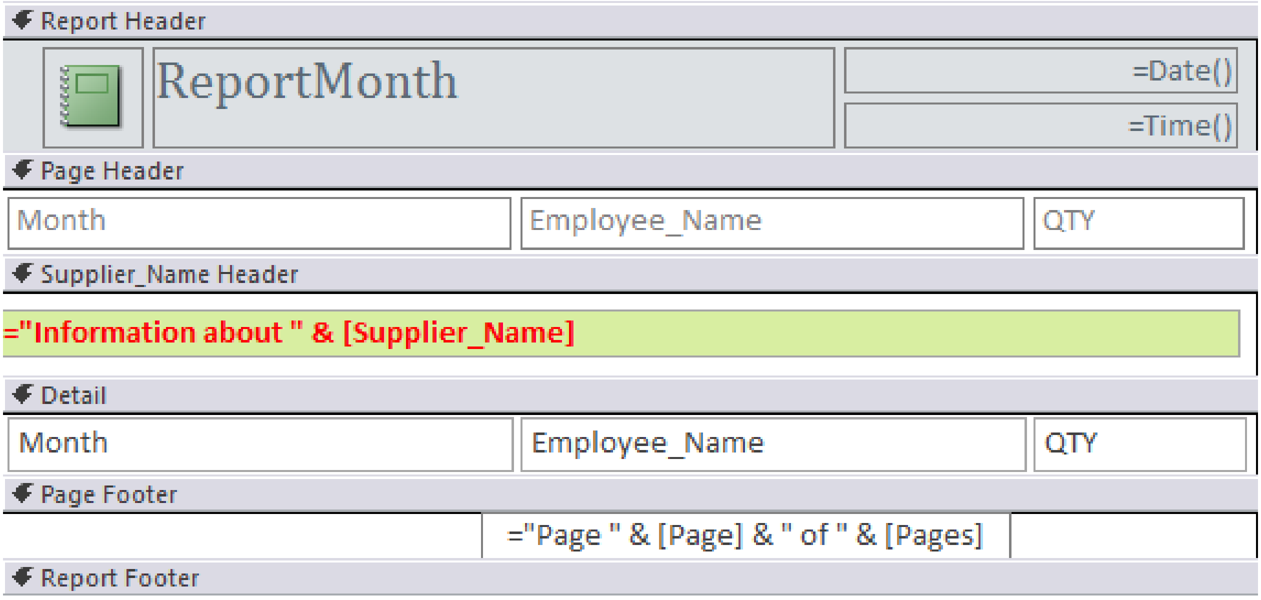
2. Create the draft version of report **Report_Month** in the base of the query **ReportMonth** (select the query **ReportMonth** and choose *Ribbon* / tab *Create* / group *Reports* / icon *Report*).
3. Switch to the *Design View*:

3.1. Add a *Group* on *Supplier_Name.*

- ✓ Add *Footer Section* to this *Group.*
- ✓ To the area *Supplier_Name Header* add the *Text box* and create the expression

="Information about " & [Supplier_Name]

- ✓ Remove the field *Supplier_Name* from the areas *Detail* and *Page Header* and change the places of another fields in this areas.



Report Header

ReportMonth

=Date()

=Time()

Page Header

Month	Employee_Name	QTY

Supplier_Name Header

="Information about " & [Supplier_Name]

Detail

Month	Employee_Name	QTY

Page Footer

="Page " & [Page] & " of " & [Pages]

Report Footer

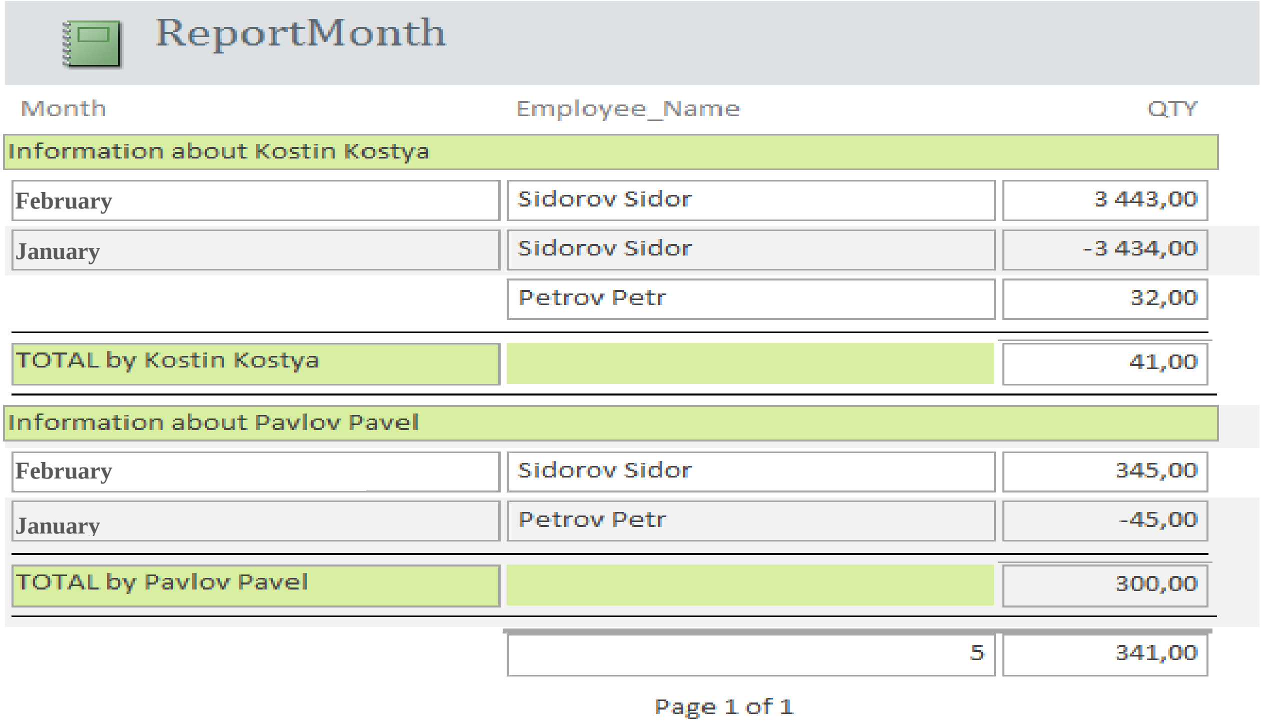
- ✓ Add *Totals* by *QTY* in the group *Supplier_Name (*select the field *QTY* and use *Ribbon* / contextual tab *Design* / group *Grouping & Totals / Group & Sort*).

For example, the result of *Report View***:**

3.2. Add a *Group* on *Month.*

- ✓ Add *Footer Section* to this *Group.*
- ✓ To the area *Month Header* add the *Text box* and create the expression:

="Information about " & [Month]



ReportMonth

Month	Employee_Name	QTY
Information about Kostin Kostya		
February	Sidorov Sidor	3 443,00
January	Sidorov Sidor	-3 434,00
	Petrov Petr	32,00
TOTAL by Kostin Kostya		41,00
Information about Pavlov Pavel		
February	Sidorov Sidor	345,00
January	Petrov Petr	-45,00
TOTAL by Pavlov Pavel		300,00
	5	341,00

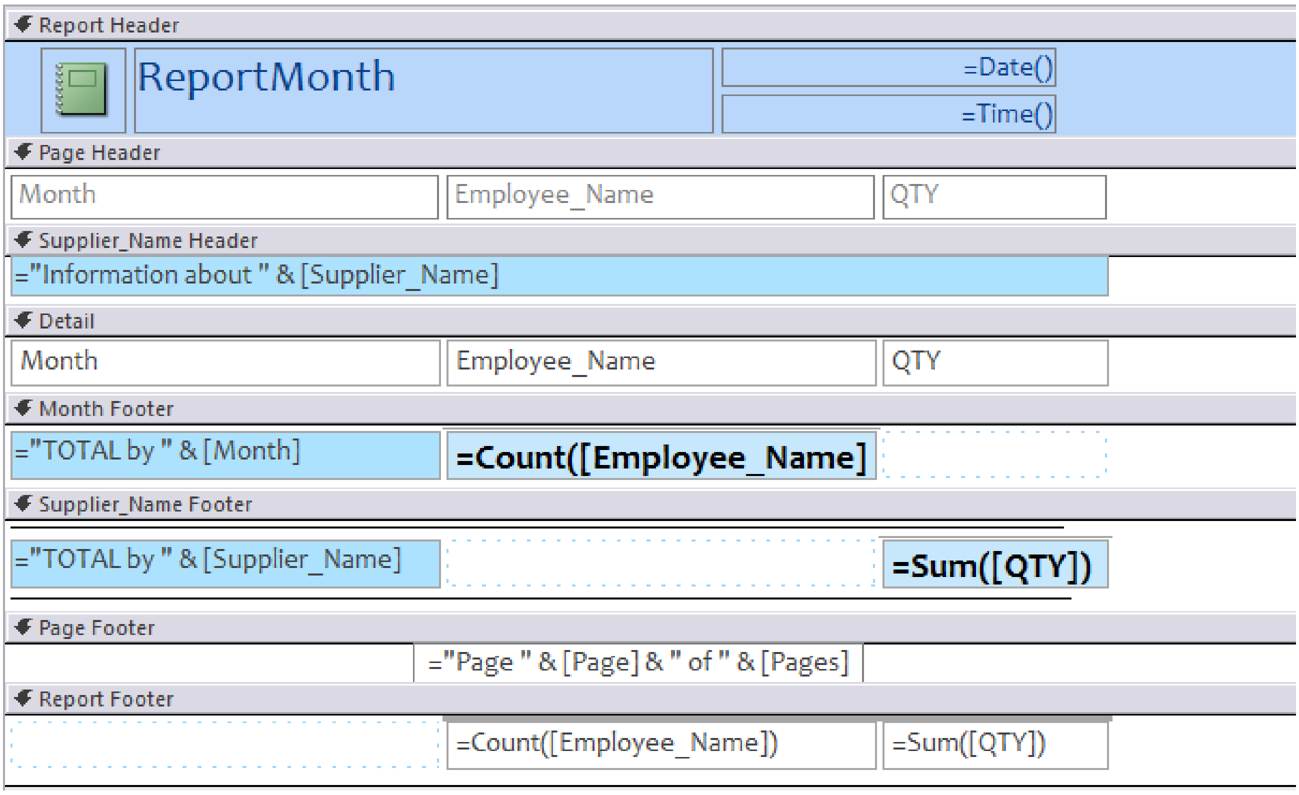
Page 1 of 1

✓ To the area *Month Footer* add the *Text box (Ribbon /* contextual tab *Design* / group *Controls / Text box)* and create the expression (*Ribbon /* contextual tab *Design* / group *Tools /* icon *Property Sheet /* Navigation Pane *Property Sheet /* tab *Data /* row *Control Source):*

="TOTAL by " & [Month]

3.3. Change *Background Color* for *Text Boxes* with information about *Totals*.

For example, the result of *Report View***:**



Report Header

ReportMonth

=Date()

=Time()

Page Header

Month | Employee_Name | QTY

Supplier_Name Header

="Information about " & [Supplier_Name]

Detail

Month | Employee_Name | QTY

Month Footer

="TOTAL by " & [Month] | **=Count([Employee_Name])**

Supplier_Name Footer

="TOTAL by " & [Supplier_Name] | **=Sum([QTY])**

Page Footer

="Page " & [Page] & " of " & [Pages]

Report Footer

=Count([Employee_Name]) | =Sum([QTY])

4. Switch to the *Layout View*:

4.1. Switch off the option of repeating the duplicate values in the field *Month* (*Ribbon /* contextual tab *Design* / group *Tools /* icon *Property Sheet /* Navigation Pane *Property Sheet /* Selection type: *Month / tab Format* / *row Hide Duplicates / Yes).*

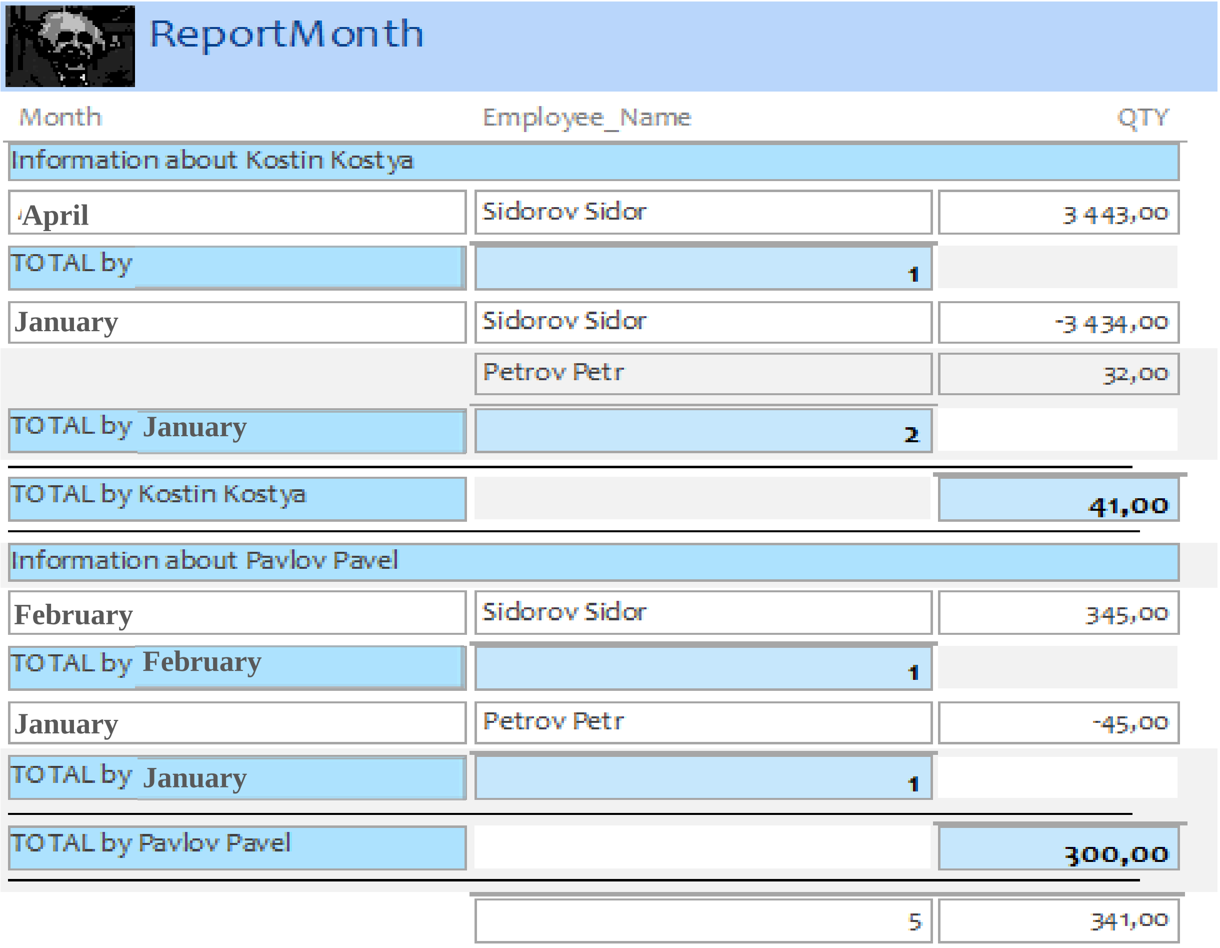
4.2. Set the format of reflecting the numbers in the field *QTY* to *Standard* (*Ribbon* / contextual tab *Design* / group *Tools* / icon *Property Sheet* / Navigation Pane *Property Sheet* / Selection type: *QTY / tab Format* / *row Format / Standard).*

4.3. Set the format of displaying the information about current page Page 1 of 1 – *Screen only (Ribbon* / contextual tab *Design* / group *Tools* / icon *Property Sheet* / Navigation Pane *Property Sheet* / Selection type: *Text_N* / tab *Format* / row *Display When/ Screen only).*

4.4. Chose the appropriate *Themes*, *Font* and *Colors* of *Form (Ribbon /* contextual tab *Design* / group *Themes).*

4.5. Add the appropriate picture as the *Logo* (*Ribbon /* contextual tab *Design* / group *Header / Footer / Logo*).

4.6. Switch to the *Report View* and *Print Preview*.



ReportMonth

Month	Employee_Name	QTY
Information about Kostin Kostya		
April	Sidorov Sidor	3 443,00
TOTAL by	1	
January	Sidorov Sidor	-3 434,00
	Petrov Petr	32,00
TOTAL by January	2	
TOTAL by Kostin Kostya		**41,00**
Information about Pavlov Pavel		
February	Sidorov Sidor	345,00
TOTAL by February	1	
January	Petrov Petr	-45,00
TOTAL by January	1	
TOTAL by Pavlov Pavel		**300,00**
	5	341,00

TESTING QUESTIONS

1. Give the definition of *CrossTab Queries.*

2. Characterize each purpose of *Row Heading, Column Heading* and *Value* elements in *CrossTab Queries*.

3. Define the specificity of using function *FORMAT* for creating the heading of the *CrossTab Queries.*

4. Give the definition of *Reports.* What the difference between *Forms* and *Reports.*

5. Enumerate all section of *Report.*

6. Characterize peculiarities of using *Report Header* section.

7. Characterize peculiarities of using *Page Header* section.

8. Characterize peculiarities of using *Detail* section.

9. Characterize peculiarities of using *Page Footer* section.

10. Characterize peculiarities of using *Report Footer* section.

11. List the main differences between *Layout* and *Design Views*.

12. Characterize peculiarities of using *Group Header* section.

13. Characterize peculiarities of using *Group Footer* section.

14. List the sequence of creating the totals for *Group*.

15. List the sequence of creating the totals for *Report*.

GLOSSARY

Relational Database Management System	Система Управление Реляционными Базами Данных
Field	Поле
Record	Запись
Properties	Свойства
Required	Обязательный
Optional	Не Обязательный (Выборочный)
Validation Rule	Условие на Значение
Design View	Режим Конструктора
Input Mask	Маска Ввода
Datasheet View	Режим Таблицы
Caption	Подпись
Validation Text	Сообщение об Ошибке
Freeze The Fields	Закрепить Поля
Hide Fields	Спрятать Поля (Столбцы)
Substitutional Symbols	Подстановочные Символы
Brackets	Квадратные Скобки
Select Queries	Запросы на Выборку
Query Wizard	Мастер Создания Запросов
Query Design	Конструктор Запросов
Exact Mismatch	Точное Совпадение

Subdatasheets	Подчиненные Таблицы
Reference Tables	Справочные Таблицы
Subrecords	Подчиненные Записи
Layout View	Вид Просмотра Формы в Режиме Разметки
Continuous Form	Ленточная Форма
Multiple Items	
Condition Formatting	Условное Форматирование
Single Form	Форма в Столбец
Form View	Вид Просмотра Формы в Режиме Формы
Split Form	Разделенная Форма
Blank Form	Бланк Формы
Header / Footer	Заголовок / Примечание
Subforms	Подчиненные Формы
Navigation Forms	Навигационные Формы
Macro Builder	Построитель Макросов
Relational Databases	Реляционная База Данных
Primary Key	Первичный Ключ
Foreign Key	Внешний Ключ
Data Integrity	Целостность Данных
Normalization	Нормализация
Referential Integrity	Ссылочная Целостность
Cascading Update Related Fields	Каскадное Обновление Связанных Полей

Cascading Delete Related Fields	Каскадное Удаление Связанных Полей
Join Properties	Свойства Объединения
Inner-Join	Включающая Связь
	Внутреннее Объединение
Outer-Join	Исключающая Связь
	Внешнее Объединение
Aggregate Queries	Итоговые Запросы
Parameter Queries	Параметрические Запросы
Action Queries	Выполняемые Запросы
	Модифицирующие Запросы
Crosstab Queries	Перекрестные Запросы
Report	Отчет

REFERENCES

1. *Microsoft Office 2010 QuickStep*. McGraw-Hill Osborne Media, 2010. – 368 p. ISBN: 0071741607 (base-type).

2. Habraken J. *Microsoft Office 2010*. In Depth. Que, 2011. – 969 p.

3. Michael R. Groh. *Access 2010 Bible*. Wiley. ISBN-13: 978-0470475348 – 1392 p.

4. Roger Jennings. *Microsoft Access 2010*. In Depth. Que. ISBN-13: 978-0789743077. – 1200 p.

5. Hart-Davis G. *Office 2010 Made Simple*. Apress, 2011. – 704 p. - ISBN: 978-1-4302-3575-0 (http://www.twirpx.com/file/628321/).

6. Joyce Cox, Joan Lambert, Curtis Frye. *Microsoft Office Professional 2010. Step by Step* (http://www.twirpx.com/file/155399/).

7. Hoffman S., Smith C., Stillman M., Heald G. *Microsoft Office 2010*. Digital Classroom. Wiley Publishing, Inc., 2011. – 430 p. – ISBN: 0470577770 (http://www.twirpx.com/file/462159/).

8. Koers D. *Picture Yourself Learning Microsoft Office 2010*. Course Technology PTR, 2010. – 416 pages. (http://www.twirpx.com/file/600195/)

FOR NOTES

Rizun N.O.

Р 49 Informatics: Workshop. Part 3 / N.O. Rizun, L.N. Savchuk. – Dnipropetrovs'k: Alfred Nobel University, Dnipropetrovs'k, 2014. – 68 p.

ISBN 978-966-434-292-3

The workshop in Informatics (part 3) is intended for students of economic specialties, who study by the international (English) program, and provides for acquiring practical skills to work with relational database management system MS Access 2010 (Module 3).

УДК 004
ББК 32.81

Навчальне видання

Різун Ніна Олегівна
Савчук Лариса Миколаївна

ІНФОРМАТИКА

Лабораторний практикум

Частина 3

(англійською мовою)

Комп'ютерна верстка О.М. Гришкіної

Підписано до друку 5.02.2014. Формат 70×100/16.
Ум. друк. арк. 5,48. Тираж 100 пр. Зам. № .

ПВНЗ «Дніпропетровський університет імені Альфреда Нобеля».
49000, м. Дніпропетровськ, вул. Набережна В.І. Леніна, 18.
Тел. (056) 778-58-66, e-mail: rio@duep.edu
Свідоцтво ДК № 4611 від 05.09.2013 р.

Віддруковано у ТОВ «Роял Принт».
49052, м. Дніпропетровськ, вул. В. Ларіонова, 145.
Тел. (056) 794-61-05, 04
Свідоцтво ДК № 4121 від 27.07.2011 р.