DESIGNING AND AUTOMATION OF A DATABASE WITH TECHNICAL SPECIFICATIONS FOR ACQUISITION

Mariana BARBARESSO*

*Military Equipment and Technologies Research Agency, Bucharest, Romania

Abstract: *This paper presents the designing and automation of a database with technical specifications for acquisition. Design database stages and relationships between tables are presented. It also shows how automation of database is made by expanding Excel capabilities using Visual Basic for Application.*

Keywords: *database design, VBA, Microsoft Excel*

1. INTRODUCTION

For an efficient management of technical specifications for acquisition has been created a Microsoft Excel database.

Microsoft Excel organizes data into tables: lists of rows and columns. Each row is called a record and each column is called a field. A record is a significant and consistent way to combine certain information. A field is a single item of information - an item type that appears in any record.

A properly designed database provides access to accurate and actual information.

In database designing took into account several general principles of database theory. The first principle is that the redundant information have a negative influence as they consume space and increase the likelihood of errors and inconsistencies. The second principle is the importance of the accuracy and completeness of information. If a database contains incorrect information, any reports that extract information from the database will also contain incorrect information. Therefore, any decision taken on the basis of these reports will be mislead.

2. DESIGNING OF DATABASE

Design is the process of transforming data requirement from reality, defined by the need of a user in a database structure.

The design process of a database with technical specifications for acquisition has gone through following steps:

- ***Finding and organizing the required information***. This stage involved gathering all kinds of information that would be recorded in the database, such as name of technical specifications for acquisition, the dates when were issued the requirements, structure which issued requirements etc.
- ***Dividing information in tables (worksheet)***. Elements of information divided in entities and major issues such as procurement technical specification, development stage, beneficiaries or contracting authority. Then each major topic becomes a table (worksheet).
- ***Transformation of information into columns***. We decided what information will be stored in each table. Each item becomes a field that is displayed as a column in the table. For example, *Autoritate_Contractanta table* includes

fields such as: name, abbreviation, address, manager etc.

- ***Identify primary keys***. For each table we set the primary key. A primary key is a column that is used to uniquely identify each row. For example, for the *Centralizator* table, the primary key can be quite technical specification code, which is a string of alphanumeric characters, defined uniquely.
- ***Configuring relationships between tables (worksheets)***. Once you have divided the information into tables, it takes a combination of useful information. For example, figure 1 shows relationships between *Centralizator*, *Beneficiar*, *Autoritate_Contractanta* and *Revizie* tables. Defining relationships between tables allows associating information in useful ways.

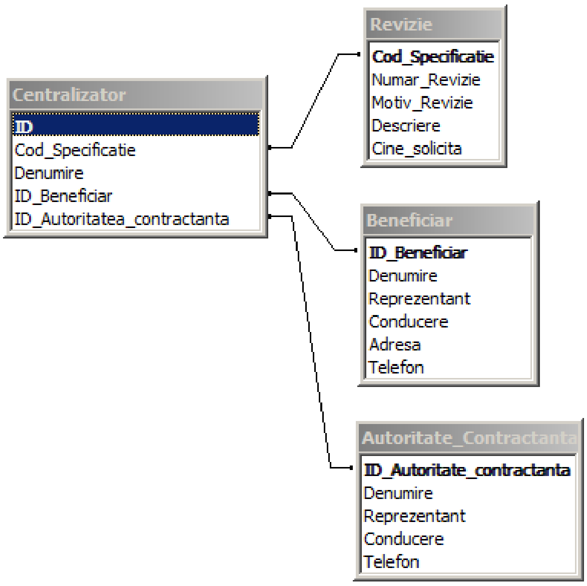


Figure 1 Relationships between tables.

One-to-many relationship. A beneficiary may have more than one technical specification for acquisition. Thus, for any beneficiary from *Beneficiar* table can have multiple specifications in *Centralizator* table. Relationship between *Beneficiar* table and *Centralizator* table is a relation of type one-to-many. In this case, the *Centralizator* table will have a *ID_Beneficiar* column that connects with data beneficiary from *Beneficiar* table. *ID_Beneficiar* column is the primary key of the *Beneficiar* table and ID_Beneficiar column from *Centralizator* table is foreign key, because is the primary key for *Beneficiar* table. This type of relationship allows storing data beneficiary once. The reports, it will extract the full data about the beneficiary and will be displayed for each technical specification.

Many-to-many relationship. This type of relationship has not been found for database with technical specifications of acquisition. Many-to-many relationship requires that a beneficiary will have many technical specifications for acquisition and one technical specifications of acquisition will have more than one beneficiary. This is impossible because one single beneficiary issued the requirements for technical specifications of acquisition. In practice, other structures can use the same technical specifications for acquisition. If the same product is intended to be purchased by other structures, coordinating structure sends to the new beneficiaries technical specification to verify the desired requirements. Subsequent changes to requirements leads to a new technical specification with a new code.

One-to-one relationships. This type of relationship was used for recording additional information about technical specifications of acquisition, information applies only occasionally or for a few products. For example, information about revision relates only to certain technical specifications of acquisition. Storage information in *Centralizator* table generate blank records for each technical specification to which they do not apply. Thus, information about revision (*Număr_Revizie*, *Motiv_Revizie*, *Descriere* and *Cine_solicită*) were stored separately in *Revizie* table. Cod_Specificație field, as in *Centralizator* table, is used with primary key role. Relationship between additional *Revizie* table and *Centralizator* table is a relationship type one-to-one. For each record in additional *Revizie* table is one recording in *Revizie* table that fits.

Refining the design. Once you have set all tables, fields and relationships between tables, the database was populated with sample data to try to work with information stored, creating queries, adding new records, generating reports, etc. This helps to highlight potential problems (eg, you may need to add a new column or to split a table into two tables to eliminate duplication). To check whether you can use the database to get the answers you want.

Database normalization. Normalization rules were applied as a step within the design process. These rules were used to check if the tables are structured correctly. The application of the design rules database is called database normalization. Database normalization is the process of organizing the fields and tables of a database to minimize redundancy and dependency. Normalization usually involves dividing large tables into smaller (and less redundant) tables and defining relationships between them. The objective is to isolate data so that additions, deletions, and modifications of a field can be made in just one table and then propagated through the rest of the database via the defined relationships.

First normal form (1NF) assumes that there is only one value at each intersection of a row and a column in the table. A relation is in first normal form if the domain of each attribute contains only atomic values, and the value of each attribute contains only a single value from that domain. For example, there may not exist several beneficiary structures in *Beneficiar* field.

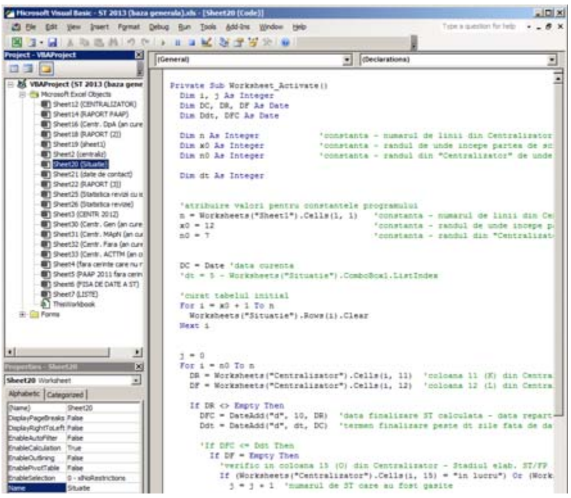
Second normal form (2NF) requires that each column is fully dependent on the primary key, not just part of the key. For database with technical specification acquisition, second normal form is respected because any column is completely dependent on the primary key. Each table in the database have the primary key consists only of a single column.

Third normal form (3NF) requires that each key column is dependent on the entire primary key, and all columns that are not keys are mutually independent. Another way of saying this is that each non-key column must be independent of the other columns. Independent column means you can change any column that is not a key, without affecting the other columns.

3. DATABASE AUTOMATION

Automation for a database with technical specifications for acquisition was done through the development and expansion of capabilities of the host application (Excel) using Visual Basic for Applications (VBA). VBA provides support for Visual Basic language, form interface, controls, objects and data access technologies. When hosted by another application, such as Excel, VBA provides the means to interact with the host application objects.

Using VBA editor, was created, edited, debugged and executed program code associated with database with technical specifications of acquisition (Figure 2).


Microsoft Visual Basic - ST 2013 (baza generala).xls - [Sheet20 (Code)]
File
Edit
View
Insert
Format
Debug
Run
Tools
Add-Ins
Window
Help
Type a question for help
Project - VBAProject
VBAProject (ST 2013 (baza gene
Microsoft Excel Objects
Sheet12 (CENTRALIZATOR)
Sheet14 (RAPORT PAAP)
Sheet16 (Centr. DpA (an cure
Sheet18 (RAPORT (2))
Sheet19 (sheet1)
Sheet2 (centrala)
Sheet20 (Situatie)
Sheet21 (date de contact)
Sheet22 (RAPORT (3))
Sheet25 (Statistica revizi cu s
Sheet26 (Statistica revizie)
Sheet3 (CENTR 2012)
Sheet30 (Centr. Gen (an cure
Sheet31 (Centr. MApN (an cu
Sheet32 (Centr. Fara (an cure
Sheet33 (Centr. ACTTM (an c
Sheet4 (fara cerinte care nu r
Sheet5 (PAAP 2011 fara cerin
Sheet6 (FISA DE DATE A ST)
Sheet7 (LISTE)
ThisWorkbook
Forms
Properties - Sheet20
Sheet20 Worksheet
Alphabetic
Categorized
(General)
(Declarations)

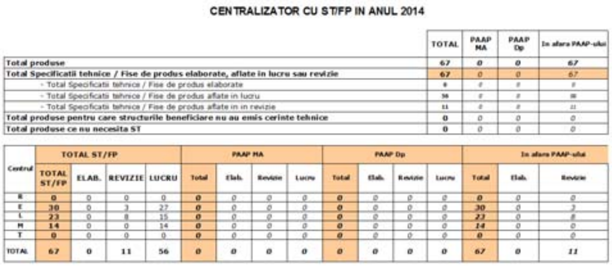
(Name)	Sheet20
DisplayPageBreaks	False
DisplayRightToLeft	False
EnableAutoFilter	False
EnableCalculation	True
EnableOutlining	False
EnablePivotTable	False
EnableSelection	0 - xlNoRestrictions
Name	Situatie

```
Private Sub Worksheet_Activate()
  Dim i, j As Integer
  Dim DC, DR, DF As Date
  Dim Ddt, DFC As Date

  Dim n As Integer           'constanta - numarul de linii din Centralizator
  Dim x0 As Integer          'constanta - randul de unde incepe partea de sc
  Dim n0 As Integer          'constanta - randul din "Centralizator" de unde

  Dim dt As Integer


  'atribuire valori pentru constantele programului
  n = Worksheets("Sheet1").Cells(1, 1)   'constanta - numarul de linii din Ce
  x0 = 12                                'constanta - randul de unde incepe p
  n0 = 7                                 'constanta - randul din "Centralizat


  DC = Date 'data curenta
  'dt = 5 - Worksheets("Situatie").ComboBox1.ListIndex

  'curat tabelul initial
  For i = x0 + 1 To n
    Worksheets("Situatie").Rows(i).Clear
  Next i


  j = 0
  For i = n0 To n
    DR = Worksheets("Centralizator").Cells(i, 11)  'coloana 11 (K) din Centra
    DF = Worksheets("Centralizator").Cells(i, 12)  'coloana 12 (L) din Centra

    If DR <> Empty Then
      DFC = DateAdd("d", 10, DR)  'data finalizare ST calculata - data repart
      Ddt = DateAdd("d", dt, DC)  'termen finalizare peste dt zile fata de da

      'If DFC <= Ddt Then
        If DF = Empty Then
          'verific in coloana 15 (O) din Centralizator - Stadiul elab. ST/FP
          If (Worksheets("Centralizator").Cells(i, 15) = "in lucru") Or (Work
            j = j + 1  'numarul de ST care au fost gasite
```

Figure 2 VBA Editor.

There have been 9 reports that automates the work with the database. The source code for each report individually run automatically when activating the appropriate worksheet. This was possible by writing a procedure associated with activation event of the worksheet that will display the results of the report. A procedure, similar to other programming languages, means a series of instructions that are executed by one call unit.

For example, to compile the report "Centralizator cu ST/FP" has made the *Numarare* procedure associated with *Activate* event for the worksheet "*Centralizator*" (Figure 3). The *Numarare* procedure goes through the entire database and do a count of technical specifications based on their stage, they carry center and approved procurement plan.



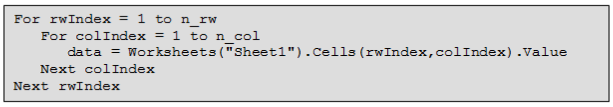
CENTRALIZATOR CU ST/FP IN ANUL 2014

	TOTAL	PAAP MA	PAAP Dp	In afara PAAP-ului
Total produse	67	0	0	67
Total Specificatii tehnice / Fise de produs elaborate, aflate in lucru sau revizie	67	0	0	67
- Total Specificatii tehnice / Fise de produs elaborate	0	0	0	0
- Total Specificatii tehnice / Fise de produs aflate in lucru	56	0	0	56
- Total Specificatii tehnice / Fise de produs aflate in in revizie	11	0	0	11
Total produse pentru care structurile beneficiare nu au emis cerinte tehnice	0	0	0	0
Total produse ce nu necesita ST	0	0	0	0

Centrul	TOTAL ST/FP				PAAP MA				PAAP Dp				In afara PAAP-ului		
	TOTAL ST/FP	ELAB.	REVIZIE	LUCRU	Total	Elab.	Revizie	Lucru	Total	Elab.	Revizie	Lucru	Total	Elab.	Revizie
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E	30	0	3	27	0	0	0	0	0	0	0	0	30	0	3
L	23	0	8	15	0	0	0	0	0	0	0	0	23	0	8
M	14	0	0	14	0	0	0	0	0	0	0	0	14	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TOTAL	67	0	11	56	0	0	0	0	0	0	0	0	67	0	11

Figure 3 *Centralizator* Worksheet.

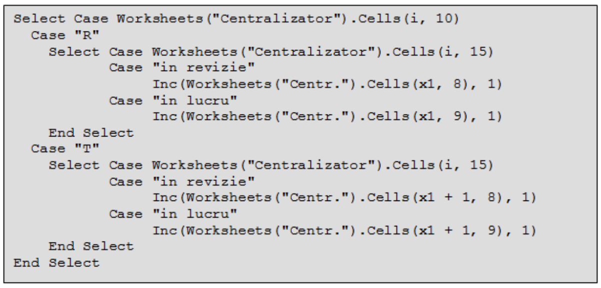
The browsing the database is made through a looping instruction and by accessing each cell, using references by index (Figure 4).



```
For rwIndex = 1 to n_rw
   For colIndex = 1 to n_col
      data = Worksheets("Sheet1").Cells(rwIndex,colIndex).Value
   Next colIndex
Next rwIndex
```

Figure 4 Accessing each cell in BD.

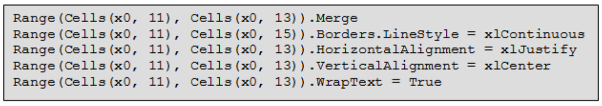
In order to count technical specifications for acquisition according to criteria established it is necessary the ramification of thread of execution after the verification of conditions. This was done using *Case* decision statement. Such a statement, found in any other programming languages, execute a group of instructions in response to comparing the same expression with multiple values different from each other. If the evaluated expression matches an item in a *Case* list, will execute the following instructions until the next *Case* statement or until *End Select* statement. Control program then passes execution to the statement following *End Select* final line (Figure 5).



```
Select Case Worksheets("Centralizator").Cells(i, 10)
  Case "R"
    Select Case Worksheets("Centralizator").Cells(i, 15)
           Case "in revizie"
                Inc(Worksheets("Centr.").Cells(x1, 8), 1)
           Case "in lucru"
                Inc(Worksheets("Centr.").Cells(x1, 9), 1)
    End Select
  Case "T"
    Select Case Worksheets("Centralizator").Cells(i, 15)
           Case "in revizie"
                Inc(Worksheets("Centr.").Cells(x1 + 1, 8), 1)
           Case "in lucru"
                Inc(Worksheets("Centr.").Cells(x1 + 1, 9), 1)
    End Select
End Select
```

Figure 5 *Case* statement.

Each of the nine reports appear automatically on the activation of corresponding worksheet.



```
Range(Cells(x0, 11), Cells(x0, 13)).Merge
Range(Cells(x0, 11), Cells(x0, 15)).Borders.LineStyle = xlContinuous
Range(Cells(x0, 11), Cells(x0, 13)).HorizontalAlignment = xlJustify
Range(Cells(x0, 11), Cells(x0, 13)).VerticalAlignment = xlCenter
Range(Cells(x0, 11), Cells(x0, 13)).WrapText = True
```

Figure 6 Text formatting.

It automatically build tables displaying the results and format text by using special functions (Figure 6).

4. CONCLUSIONS

Visual Basic for Application extended host application capabilities of Microsoft Excel, allowing automation of database with technical specifications of acquisition. VBA, through the functions it provides, reduced the time of preparation of situations and statistics, they practically accomplished automatically, accessing proper worksheet. However, through automation, were eliminated errors that may be introduced by human factor.

REFERENCES

1. Katz, A., *Beginning Microsoft Excel 2010*, Apress, 2010.
2. Walkenbach, J., *Excel VBA Programming*, Wiley, 2010.