BUILDING A DATA WAREHOUSE FOR UNIVERSITY FROM PETROŞANI

Preda Mircea, *Department Mathematics - Informatics, University from Petroşani, mpreda2002@yahoo.com*

ABSTRACT: *This paper presents a concrete way of building a data warehouse suitable for University of Petroșani. For its realization, we started from existing data sources, which consist of a lot of DBF tables based on: specializations, study programs, departments, faculties, disciplines, budget allocations and taxes, etc., which is an important background of information collected over the last 8 years.*

For their operation, they must be normalized, standardized and structured in a database, which will be operated through a complex query mechanism that is capable of extracting information stored according to user requirements and transform them into dynamic, structure or mixed series. These forms of representation offers a multiple range of analysis and graphical representation of data. For graphical representation in this stage was provided an MS Excel export that allow a fast and easy solution of the problem.

KEYWORDS: *data warehouse, architecture of a data warehouse, source of data warehouse, functions of the Data warehouse*

1 THE DISADVANTAGES OF THE ACTUAL SYSTEM

In this moment, The University from Petroşani has two applications: one, that facilitates the income per an academic year, on faculties, on levels of study, on programs of study and years of study and their distribution on departments to determine the coverage of salary costs by departments, colleges and university, and the other application that allows substantiation monthly allowable income on departments, faculties and university and seeks framing the expenditure in income.

These two applications provides information monthly and cumulative from the beginning of academic year for departments, faculty and university management. They collect monthly amounts of information on tax receipts and wage costs made, which then aren't longer used.

In order to operate this information by providing accurate image, clear and simple on the dynamics and structure of each indicator collection designed a database by supplying it with information in data sources specific to the two applications over the years.

1.1 The general architecture of a data warehouse

The data warehouse is designed for managers, analysts and specialists involved in making strategic decisions on the development and future of the organization.

In developing the data warehouse of the university was envisaged the general architecture of a data warehouse that consists four main components:

- the source of data warehouse;
- the data warehouse;
- the update component of the data warehouse;
- the query component of the data warehouse,

which are connected as in Fig.1.

The role of components is as follows:

- Data sources are current, archived and also external databases.
- Updating the data warehouse involves the following steps:
  - data extraction from data sources and transforming them into internal format and structure of the deposit;
  - cleaning of data in order to ensure that data are accurate and can be used for decision making;
  - loading correctly the data into a data warehouse;
  - aggregation of data by calculating a precalculated totals, subtotals, averages, relative size, etc.. expected to be called and used by users.
- Metadata are tables describing the data in the data warehouse and how they are obtained and stored. Through metadata is specified the data structure, their origin, transformation rules, aggregation and calculation rules. They are used in all stages of data loading are accessed and updated throughout the life cycle of the storage. The inclusion of aggregated data in the storage, although it increases the redundancy, it contributes to an average response time much shorter.

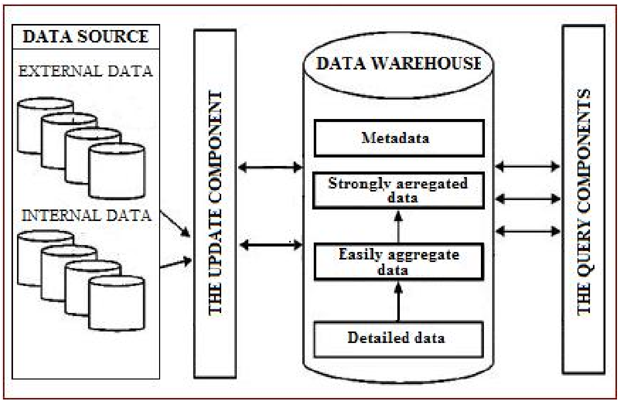


Fig. 1. General architecture of the data warehouse

1.2 Data sources

The data that will be storage in the data warehouse can be found in the following tables that are operational collections used during an academic year, namely:

- ANISTU – contains information's specific years of study;
- CHEL_CAT – contains information's specific departmental costs;
- DISANI – contains subjects on year of study, curricula;
- DISP_V01 – contains taxes distribution values on departments;
- DISP_V0P – contains taxes distribution percentage on departments;
- INCASARI – contains monthly receipts, tuition fees on faculties;
- INCASARE- contains collections from re-examinations, contracts on departments;
- INCAD_DC – contains Ph receipts on holders and departments.

1.3 Functions of the university data warehouse

For the deposit to be functional and to fulfill the role of providing rapid information about certain indicators on different periods, it must meet a number of basic functions that were structured as follows:

1. Update TAB_DIC table with structure of source tables from BAZE;
2. Update STR_XDD table with information from TAB_DIC that don't have empty class;
3. Update / calculation the structure indicators of the field in STR_XDD;
4. Update DD_UPET table with the information offered by sources from TAB_SUR table;
5. Data warehouse DD_UPET query, which involves:
   - 5.1. Marking the desired attribute from STR_XDD table;
   - 5.2. Marking the S type – structure that establish order;
   - 5.3. Extract Simbol, Denumire, Tip in a memory table for the selected fields, and:
     - a. building a variable with the list of selected fields;
     - b. building a list of specific values from the list fields;
     - c. building a filtering expression for the selection from DD_UPET :

- For each field marked set the desired filter in Wfilt0_i, all values, list of values, interval, equal with a value;
- Generate general filter by connecting logical AND filter specific columns.
  - o Drawing up the index key expression;
  - o Generating time series through transposition on the required structure of:
    1. Absolute values;
    2. Development index;
    3. Absolute change;
    4. Development rate

1.4 The executive components of the storage and their role

1. **Pr_gdic –** a procedure that generates / populates TAB_DIC table with the structure of SURSĂ tables defined in BAZE table according to the following chart.

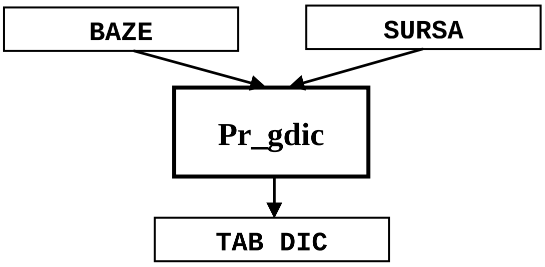


Fig. 2 Diagram of the pr_gdic procedure

- BAZE table - contains the names of all tables and those that INS equal to one contains information that will reach in the data warehouse.
- SURSA table is the generic name of the data source from which will take the fields that will be transferred to the dictionary of the data warehouse DD_UPET.

2. **Pr_addup** – the procedure update the data of DD_UPET with the information from sources specified in TAB_SUR table. For records with data attribute which is - a date - is generated the fields - anu - academic year, an - calendar year in four digits and lu - month calendar, that will be used in queries. Similar is proceeding with records that have luna > 1 but that have the format aaaall where aaaa is the year four-digit year, ll is the month in two digits. Processing is done according to the Fig. 3, where :

- TAB_SUR table contains the name and symbol of the source from where it will take data in the DD_UPET data warehouse.

- SURSA table is the generic name of the ready from TAB_SUR that will be processed by the Pr_addup0 procedure.
- Pr_add0 – is a subordinate procedure of Pr_addup procedure that receives two wsursa parameters - the name of the processed source –and wsim - the symbol of the source.

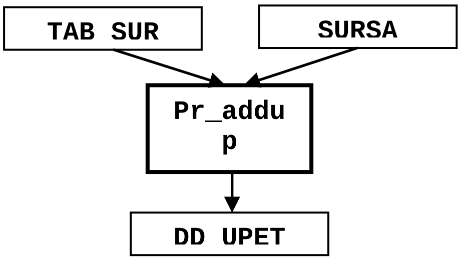


Fig. 3 Diagram of the Pr_addup procedure

Based on the transmitted parameters from STR_XDD table is filtered only the records that describe the fields which are taken from the source specified by wsim parameter who have "X" value that you copied to the memory wstr table. In the next step is opening the source table transmitted through the wsursa parameter from which selects only records specific for the academic year prepared in wanu variable. In the three stage, from each record from the source generates normalized records. A normalized record is that record which have just one “V” type field and others are „S” type. So, from a record source will generate both normal records by “V” type fields are in wstr. The fields transfer from a source in a record are normally as follows:

- If the field is „S” type , it will have the same name and the transfer is made from sursă.câmp field into dd_uper.câmp field.
- If the field is “V” type, the transfer is made as follows:
    - The field name moves in dd_upet.atribut field;
    - The field value moves in dd_upet.valoare field;

When all fields of a normalized records were transferred in variables of memory m.field type they are put into an empty record from the dd_upet storage.

3. **Fg_anu** – is a function that generate the academic year anu starting from wdata0 function parameter, that can be of N type – numeric or D type - calendar date. Generate process is based on drawing up a value „2010-2011” considering that an academic year starts in one year and end in the next one. The rules for an academic year is given in the following table.

If month ∈ {10,11,12}	If month not ∈ {10,11,12}
Wanu=”year-(year+1)”	Wanu=”(year-1)-year”

Tab. 1 Rules for an academic year generation

where luna and anul are values taken from the wdata0 parameter and wanu is the variable whose result is returned.

4. **Pr_vdd** – complex procedure of database query from dd_upet storage, centralization, transposition, generating statistical indicators specific time series and view the results. It involves the following steps:

a. *determination the value indicators that interest us and the related information.* From STR_XDD table is chosen the value indices that interested us then keep to it: the symbol, name, source and source icon and then select all the S type fields and consults the user on the fields that interest and in what order. For results, are declared the memory vectors wl_val, wl_den, wl_typ in which will keep the symbol, the name and the type fields.

b. *supplying information to vectors, drawing up the list of auxiliary variables.* From STR_XDD for S-type marked fields, their order is established and the following operations are performed:

- the declared vectors are completed with the field values;
- it builds wlfields variable with fields list;
- it builds w_NumeCâmp variables that will keep its values;
- it builds wExpK variable with indexing key expression according to the established order and the type of field;
- attaching atribut and valoare fields to the list of fields.

c. *drawing up for the S-type fields: the list of values, the maximum length filter.* The process consists of:

- drawing up the wlx memory table with the distinct values of each field;
- determine the maximum length in characters of the field values in wlmax ;
- drawing up the filter expression specific to the field in wfilt0;
- drawing up the general filter through concatenation the previous general AND operator and wfilt0 field filter.

d. *the processing, in fact,* which using information previously prepared and includes:

- open the dd_upet storage;
- indexing it after the built key in wExpK;
- adding “and atribut=wfield” the condition to the filter;
- activate the defined filter;
- centralization valoare field after wExpK expression and keeping the result in TOT_0 table;
- viewing the results;
- calling the transposition and generation procedures for time series.

5. **Fc_Maxt** – is the function that calculates and returns the maximum size in characters of elements of an external table wlx , elements that can be either N or C.
6. **Fc_Lval** – function sets and returns a list of distinct values of a field from dd_upet data warehouse, of a certain type and length, which are specified as parameters. To do so, the following steps are performed:
   a. open the dd_upet storage;
   b. indexing it by the field specified in UNIQUE condition;
   c. through sequential scroll, it builds the values list in wlval variable;
   d. the storage is closing;
   e. returns the built value list.
7. **Fc_Sims** – this function sets and returns the source symbol that it receives as a parameter.
8. **Pr_Asdd** – the procedure updates the source symbol from where comes every field of storage that find in TAB_DIC. For this, sequential scroll through the dictionary, searching directly the field in STR_XDD and if it found directly is marked with X, source whence that field. In this way, it ensures the recovery of source information, but it is also an important base for questioning.
9. **Pr_Gsdd** – procedure allows updating the STR_XDD data warehouse structures starting from TAB_DIC. In this way the structure of the deposit is synchronized with the data dictionary.
10. **Pr_L2dic** – the procedure ensures the listing or structure view of STR_XDD storage.
11. **Pr_trs** – is a very complex function that is based on two parameters: wdbf - centralized table name – and wlisc – list of fields on which generates a pivot or transpose table in Excel. The fields order is important and they have the following meanings:
    - C_1 – is the field that translates and its based of generation the time series;
    - C_2 – is the field corresponding to attribute value that it will centralize;
    - C_3-C_n – are structure and territorial fields.

    The process is accomplished in eight stages:

a. In the first stage :
   - fields are copied on the extended version from wlisc of the wdbf table in the table of TMP_1, locate C1 in TMP_1 table, save the attributes type, length, decimals, and then delete the record;
   - locate C1 in TMP_1, save its attributes : type, length, decimals and then delete it.

b. determinate for the other fields in the list:
   - wcondiţie – the expression of centralization or equality between the current record and the previous one;
   - wliscx – list of the new or untransposed fields.

c. for different values of the first field it is generated a recording named x_valoare and the other attributes are inherited from C1, and finally add a new field named Total;

d. create TRANS table (Transposed) from table TMP_1;

e. from the table provided by wdbf parameter, that is centralized to the wcondiţie level, the new table is updated - basically translates under the wcondiţie condition determined in stage b;

f. for fields generated by form x_valoare generated in stage c . including the Total field, it is calculated by adding by the column, a total and the result is recorded in a new record, which will have at the atribut field a value "Total-G";

g. the TRANS table is indexed and centralized by the wexpK0 key of fields generated in step c. in TRANSC table and take from TRANS table the row of "Total-G";

h. generate all the others dbf table type specific to the time series, and finally generates under the form nume.xls:
   - TRANS_P – evolutionary transposed with percent %;
   - TRANS_M – transposed with absolute changes from one period to another;
   - TRANS_R – transposed with the pace of development in %.

12. **Pr_gdbfp, Pr_gdbf_m, Pr_gdbf_r** – are procedures similar in form but different in content and role.
    They are based on the TRANSC table, which has the following form:

C2	C3	...	Cp	Attribute	Value	C1 transposed by value					Total
						X_1	...	X_j	...	X_m	
			1			X_{11}		X_{1j}		X_{1m}	TL_1
			...								...
			i			X_{i1}		X_{ij}		x_{im}	TL_i
			...								...
			n			X_{n1}		x_{nj}		X_{nm}	TL_n
				Total-G		TC_1		TC_j		TC_m	TG

Tab. 2 Structure of TRANSC table

where the significance of the new elements are:

- o X_{ij}– the value corresponding to the i record after transposition, corresponding to j value of C1;
- o TL_i –is the x_{ij} total from the i line;
- o TC_j –is the x_{ij} total from the j column;
- o TG – is the overall total on the line or column.

Particularities of the procedures consist in the calculation way of the x_{ij} elements according to the type of table as follows:

- Pr_gdbfp – generates TRANS_P table in which x_{ij} elements are determinate using the formula:

$$p_{ij} = \frac{x_{ij}}{TC_{-j}}.100, \quad (1)$$

which represents percent (%) from the total on the column.

- Pr_gdbfm – generates TRANS_M table in which x_{ij} elements are the absolute changes in the previous period and it is calculated using the formula:

$$m_{ij} = x_{ij} - x_{ij-1}, \quad (2)$$

which represents the change from the previous period.

- Pr_gdbfr – which generates TRANS_R table in which x_{ij} elements are the rates of evolution from one period to another determined with the formula:

$$r_{ij} = \frac{x_{ij}-x_{ij-1}}{x_{ij-1}}.100, \quad (3)$$

which represents the rate of evolution.

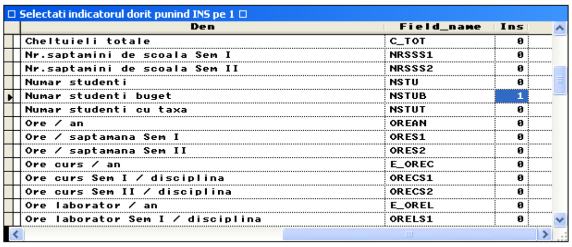
14. **Fc_val** – for a given database, a field of type and length is built a list with the values of the given field in ascending order.

1.5 Example of consultation

The application must be submitted for operating parks steps listed below and it offers a very wide range of extraction, grouping, centralization and presenting the results in multiple forms useful in business analysis and management.

Step 1 Select the desired indicator

Application table has primary statistical indicators, subject to consultation by navigation using next window, concrete case "budget student number" after his passing ins on one press CTRL + W and proceed to Step 2.



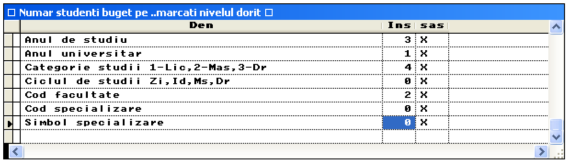
□ Selectati indicatorul dorit punind INS pe 1 □

Den	Field_name	Ins
Cheltuieli totale	C_TOT	0
Nr.saptamini de scoala Sem I	NRSSS1	0
Nr.saptamini de scoala Sem II	NRSSS2	0
Numar studenti	NSTU	0
Numar studenti buget	NSTUB	1
Numar studenti cu taxa	NSTUT	0
Ore / an	OREAN	0
Ore / saptamana Sem I	ORES1	0
Ore / saptamana Sem II	ORES2	0
Ore curs / an	E_OREC	0
Ore curs Sem I / disciplina	ORECS1	0
Ore curs Sem II / disciplina	ORECS2	0
Ore laborator / an	E_OREL	0
Ore laborator Sem I / disciplina	ORELS1	0

Fig. 4 Select indicator NSTUB

Step 2. Establish order of centralization fields

At this stage the structure is established fields of interest and their order, which will be the basis pivot and centralization, and the number one is the one that will pivot. In the present case were chosen fields numbered 1-4 as shown below. For fields marking the next step is to settle the selection rules.



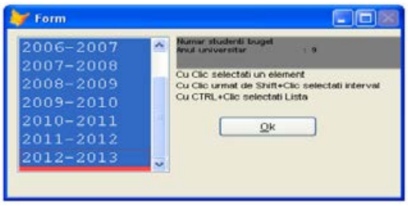
□ Numar studenti buget pe ..marcati nivelul dorit □

Den	Ins	sas
Anul de studiu	3	X
Anul universitar	1	X
Categorie studii 1-Lic,2-Mas,3-Dr	4	X
Ciclul de studii Zi,Id,Ms,Dr	0	X
Cod facultate	2	X
Cod specializare	0	X
Simbol specializare	0	X

Fig. 5 Establishing the order of the fields used

Step 3. Set individual and general filter

Filtering mechanism is similar for all fields set and allows filtering list or range, and after completion click OK. The operation is repeated for each field separately, but changes the list of values and significance of the field. After defining the last filter is construieşte and execute effective filtering for any conditions and displays the result as shown in Fig. 3.



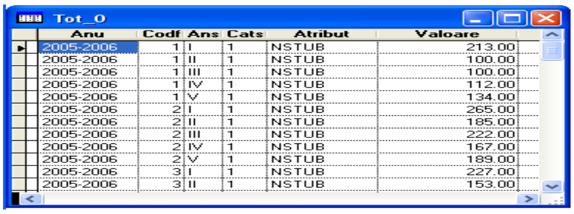
2006-2007
2007-2008
2008-2009
2009-2010
2010-2011
2011-2012
2012-2013

Numar studenti buget
Anul universitar : 9

Cu Clic selectati un element
Cu Clic urmat de Shift+Clic selectati interval
Cu CTRL+Clic selectati Lista

Fig. 3 Window of the filter fixing field

In the figure below you can see filtered results and centralized fixed fields only after defining the order, namely: academic year, college code, year study, study group, attribute and value.



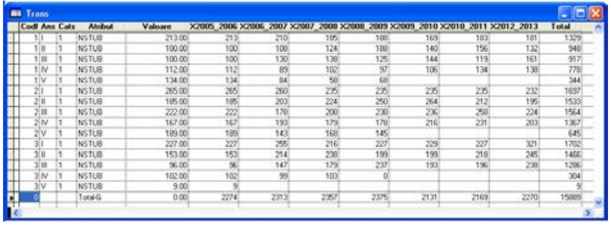
Tot_0

Anu	Codf	Ans	Cats	Atribut	Valoare
2005-2006	1	I	1	NSTUB	213.00
2005-2006	1	II	1	NSTUB	100.00
2005-2006	1	III	1	NSTUB	100.00
2005-2006	1	IV	1	NSTUB	112.00
2005-2006	1	V	1	NSTUB	134.00
2005-2006	2	I	1	NSTUB	265.00
2005-2006	2	II	1	NSTUB	185.00
2005-2006	2	III	1	NSTUB	222.00
2005-2006	2	IV	1	NSTUB	167.00
2005-2006	2	V	1	NSTUB	189.00
2005-2006	3	I	1	NSTUB	227.00
2005-2006	3	II	1	NSTUB	153.00

Fig. 4 Filtration the conditions set

Step 4 Generate tables transposed centralized

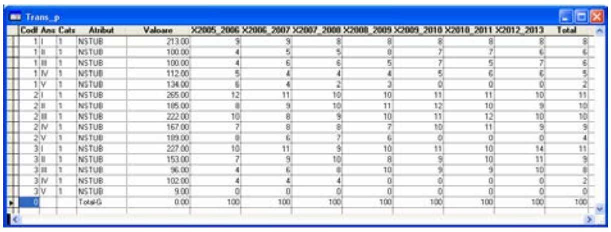
At this stage the previous table is transposed after the first field and rearranged the tables are exported to MS Excel and looks like the following figure. They have the advantage that highlights the absolute and relative progress that facilitates comparison level indicator components present with the other components.



Trans

Codf	Ans	Cats	Atribut	Valoare	X2005_2006	X2006_2007	X2007_2008	X2008_2009	X2009_2010	X2010_2011	X2012_2013	Total
1	I	1	NSTUB	213.00	213	210	185	180	169	183	181	1329
1	II	1	NSTUB	100.00	100	100	124	180	140	156	132	948
1	III	1	NSTUB	100.00	100	130	138	125	144	119	161	917
1	IV	1	NSTUB	112.00	112	89	102	97	106	134	138	778
1	V	1	NSTUB	134.00	134	84	58	68				344
2	I	1	NSTUB	265.00	265	260	235	235	235	235	232	1697
2	II	1	NSTUB	185.00	185	203	224	250	264	212	195	1533
2	III	1	NSTUB	222.00	222	170	208	238	236	258	224	1564
2	IV	1	NSTUB	167.00	167	190	179	170	216	231	203	1367
2	V	1	NSTUB	189.00	189	143	168	145				645
3	I	1	NSTUB	227.00	227	255	216	227	229	227	321	1702
3	II	1	NSTUB	153.00	153	214	230	199	199	218	245	1466
3	III	1	NSTUB	96.00	96	147	179	237	193	196	238	1286
3	IV	1	NSTUB	102.00	102	99	103	0				304
3	V	1	NSTUB	9.00	9							9
0			Total-G	0.00	2274	2313	2357	2375	2131	2169	2270	15889

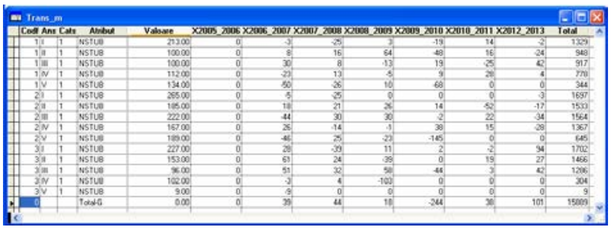
Fig. 6 Absolute transposed academic years



Trans_p

Codf	Ans	Cats	Atribut	Valoare	X2005_2006	X2006_2007	X2007_2008	X2008_2009	X2009_2010	X2010_2011	X2012_2013	Total
1	I	1	NSTUB	213.00	9	9	8	8	8	8	8	8
1	II	1	NSTUB	100.00	4	5	5	8	7	7	6	6
1	III	1	NSTUB	100.00	4	6	6	5	7	5	7	6
1	IV	1	NSTUB	112.00	5	4	4	4	5	6	6	5
1	V	1	NSTUB	134.00	6	4	2	3	0	0	0	2
2	I	1	NSTUB	265.00	12	11	10	10	11	11	10	11
2	II	1	NSTUB	185.00	8	9	10	11	12	10	9	10
2	III	1	NSTUB	222.00	10	8	9	10	11	12	10	10
2	IV	1	NSTUB	167.00	7	8	8	7	10	11	9	9
2	V	1	NSTUB	189.00	8	6	7	6	0	0	0	4
3	I	1	NSTUB	227.00	10	11	9	10	11	10	14	11
3	II	1	NSTUB	153.00	7	9	10	8	9	10	11	9
3	III	1	NSTUB	96.00	4	6	8	10	9	9	10	8
3	IV	1	NSTUB	102.00	4	4	4	0	0	0	0	2
3	V	1	NSTUB	9.00	0	0	0	0	0	0	0	0
0			Total-G	0.00	100	100	100	100	100	100	100	100

Fig. 7 Transposed weighted academic years



Codf	Anx	Cals	Atribut	Valoare	X2005_2006	X2006_2007	X2007_2008	X2008_2009	X2009_2010	X2010_2011	X2012_2013	Total
1	I	1	NSTUB	213.00	0	-3	-25	3	-19	14	-2	1329
1	II	1	NSTUB	100.00	0	8	16	64	-48	16	-24	948
1	III	1	NSTUB	100.00	0	30	8	-13	19	-25	42	917
1	IV	1	NSTUB	112.00	0	-23	13	-5	9	28	4	778
1	V	1	NSTUB	134.00	0	-50	-26	10	-68	0	0	344
2	I	1	NSTUB	265.00	0	-5	-25	0	0	0	-3	1697
2	II	1	NSTUB	185.00	0	18	21	26	14	-52	-17	1533
2	III	1	NSTUB	222.00	0	-44	30	30	-2	22	-34	1564
2	IV	1	NSTUB	167.00	0	26	-14	-1	38	15	-28	1367
2	V	1	NSTUB	189.00	0	-46	25	-23	-145	0	0	645
3	I	1	NSTUB	227.00	0	28	-39	11	2	-2	94	1702
3	II	1	NSTUB	153.00	0	61	24	-39	0	19	27	1466
3	III	1	NSTUB	96.00	0	51	32	58	-44	3	42	1206
3	IV	1	NSTUB	102.00	0	-3	4	-103	0	0	0	304
3	V	1	NSTUB	9.00	0	-9	0	0	0	0	0	9
0			Total-G	0.00	0	39	44	18	-244	38	101	15809

Fig. 7 Transposed with the rhythm of evolution

Conclusion

Through achieve this data warehouse we obtain some facilities:

- achievement a systematic records of the main indicators specific to the entity for a big period of time;
- putting all activities on consultation, processing and analysis of indicators;
- provide a wide range of developments and structures indicators towards the adoption of more accurate decisions;
- provides the ability to connect the results with MS Excel for graphical representation of the structure, evolution and trend analysis indicator;
- provides great flexibility in structuring and consultation the information;
- ease of use by offering convenient tools to define the filtering, ordering and centralization rules.

The application is useful for students to understand the concept of data warehouse, but it is useful to soft developers in decision support in many societies that don't have enough capital to purchase a professional package and can even become a product that can be promoted on the market.

References:

[1] Preda M., *Information systems for decision support*, Universitas publishing house, Petroşani, 2012

[2] Preda M., *Design of Information Systems*, Universitas publishing house, Petroşani, 2002;

[3] Preda M., *Management Information Applications*, Universitas publishing house, Petroşani, 2011;

[4] *** *University management application* ;

[5] *** *MS Office, ON-LINE documentation*;

[6] *** *MS Visual Foxpro, documentation ON-LINE*.