

## PIVOT REPORTING TOOL

Nikolay Pavlov, Maria Dobрева

**Abstract:** In this paper we describe a tool for data analysis via pivot tables, implemented as part of the Framework for Distributed Business Applications – FDBA. We present the primary functional requirements and describe how they were implemented. Our tool can retrieve and analyze data from multiple related database tables. A key feature is proper summarization of data values from parent database entities.

**Keywords:** data analysis, pivot table, FDBA

### 1. Introduction

Businesses create huge amounts of potentially valuable data during the course of their daily operations. Obtaining information from that data is critical for the successful planning and making both strategic and operational decisions. Statistics indicate that workers waste many hours searching for, sorting, and assessing information, incurring a significant organizational productivity cost [2]. For example, International Data Corp. estimates that an enterprise with 1,000 knowledge workers loses a minimum of \$6 million a year in the time workers spend searching for – and not finding – needed information [3].

The Framework for Distributed Business Applications – FDBA is a powerful platform for developing state-of-art enterprise resource planning (ERP) and customer relation management (CRM) software products [7]. No such system can be complete without an adequate tool for business intelligence. Business intelligence (BI) is the set of techniques and tools for the transformation of raw data into meaningful and useful information for business analysis purposes. BI technologies are capable of handling large amounts of unstructured data to help identify, develop and otherwise create new strategic business opportunities. The goal of BI is to allow for the easy interpretation

of these large volumes of data. Identifying new opportunities and implementing an effective strategy based on insights can provide businesses with a competitive market advantage and long-term stability [1].

The goal of this project is to implement a new standard functionality for FDBA that will empower users create aggregated reports and analyze data from various perspectives – Pivot Reporting Tool (PRT).

## 2. Requirements

### **Pivot table**

Data reporting and visualization has come a long way. At first there were simple reports and queries; then came multidimensional analysis. While cubes and star schemas are still very popular, there is a movement now toward more complex, deeper analytics using data mining, statistical methods and technologies that support pure ad hoc, unplanned forms of analytics. Perhaps it is just the natural evolution of business intelligence toward a more mature environment. Certainly, the level of business expertise is much more sophisticated and advanced than it was 10 years ago, and the ease with which even non-skilled users can perform these complicated analyses increases their adoption and utilization [4].

Then we want to give users a dynamic, interactive tool with which they can obtain an overview of data from different perspectives, data summarization on multiple levels, and ability to drill-down into data to inspect details. To satisfy this requirement, we decided to use a pivot table.

A pivot table allows users to create an interactive view of their datasets. Pivot tables can be used to create unique perspectives of user's data, with custom calculations and summarizations which do not change the underlying data [5]. A pivot report consists of *row*, *column* and *data* fields. Row and column fields give view on the summarized data from different perspectives. Multiple row and column fields are organized in a hierarchy. The first row or column field is the top level of the hierarchy and give summary of most data. Subsequent row or column fields give a finer level of detail and give a summary of less data. The farther one field from the start, the less data it summarizes. Data fields are fields on which summarization or some other calculation is performed.

### **Accessible**

There are many powerful and feature-rich Business Intelligence and Reporting tools available. The biggest disadvantage of existing solutions is that all of them

require that users who create reports have good understanding in relational databases, and the internal database structure of the application. As a result, report creators are usually software developers or very technically proficient users. This increases the costs of obtaining customized reports. We aim to hide the complexities of the relational database structure as much as possible from users. It should feature a simple, user-friendly interface. Users must operate with well-known entities, such as relations, policies, claims, etc., and the relations between these entities must be clearly and unambiguously presented.

### **Extensibility**

While keeping the tool accessible, we want to retain the power of developers to create complex calculations. The PRT must be extensible, so that developers can create reports with more sophisticated and domain-specific calculations. Such extensions will appear as application-specific functionality in applications, based on FDBA.

## **3. Implementation**

The PRT consists of three major components:

- a. Pivot Configuration Tool;
- b. Pivot Table viewer;
- c. Extensions support.

### **3.1. Pivot Configuration Tool**

1. The Pivot Configuration Tool (PCT) lets users configure their pivot report by selecting row fields, column fields and data fields with summarization function. PRT makes a heavy use of the Database Dictionary (DbD) of FDBA. DbD [6] is an extension to the metadata of the application's relational database. Among other things, DbD:
  2. Provides user-friendly names of database entities and attributes with multi-lingual support.
  3. Provides a topology map with the database entities, their attributes (fields) and the links between them.
  4. Hides "internal" data attributes. Every database has certain data attributes, which are for internal use by application developers and database administrators, and are not for application users to access.

Through the use of DbD it achieves two goals:

1. Users can create complex queries against multiple database tables. The PCT hides the notion of database tables from users. Instead, it creates a visual tree with fields, where child nodes represent fields from linked database entities.
2. Field names are displayed with their names as they appear in the user interface of application. In this way users work with well-known objects. Since DbD has multi-lingual support, the names may appear in different languages as well according to the language preferences of the user. Users see a clear description of their data, and allow them to properly choose the data they want to see in their reports.

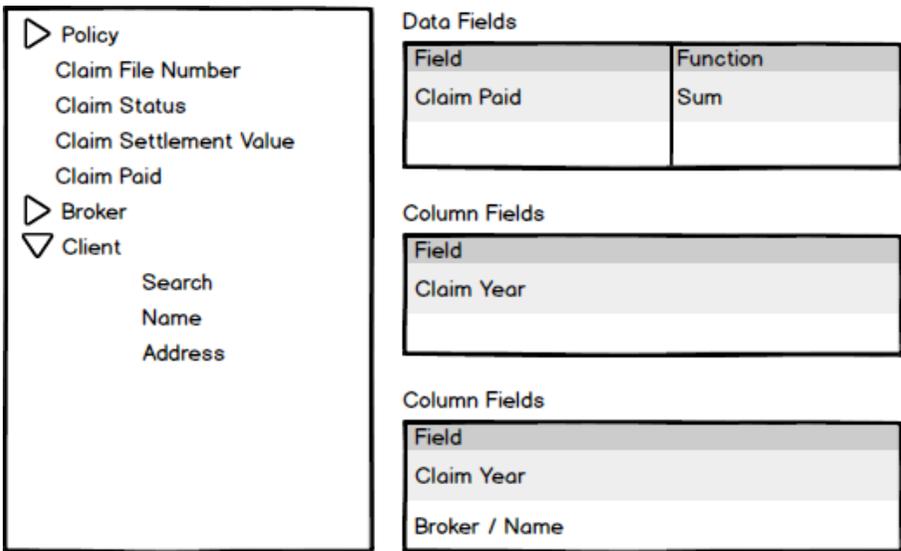


Figure 1. Pivot Configuration Tool

Only fields from the current database entity and its parent entities (and the parents of the parents and so on) are allowed. Fields from children entities cannot be used. Users must be trained how to properly identify the starting point for their pivot reports.

**Data fields**

Only numeric fields can be used as data fields. Standard summarization functions include:

- Count – The number of values (excluding Null and DBNull).

- Count Distinct – The number of distinct values.
- Max – The largest value.
- Min – The smallest value.
- Average – the average of the values.

$$\bar{v} = \frac{1}{n} \sum_i v_i$$

- StdDev – An estimate of the standard deviation of a population where the sample is a subset of the entire population.

$$StdDev = \sqrt{\frac{1}{n-1} \sum_i (v_i - \bar{v})^2}$$

- StdDevp – The standard deviation of a population where the population is all of the data to be summarized.

$$StdDevp = \sqrt{\frac{1}{n} \sum_i (v_i - \bar{v})^2}$$

- Sum – The sum of the values.

$$Sum = \sum_i v_i$$

- Var – An estimate of the variance of a population, where the sample is a subset of the entire population.

$$Var = \frac{1}{n-1} \sum_i (v_i - \bar{v})^2$$

- Varp – The variance of a population, where the population is all of the data to be summarized.

$$Var = \frac{1}{n} \sum_i (v_i - \bar{v})^2$$

One field can appear more than once in the list of data fields within a single report, each time with different function.

### Column and Row Fields

Column and row fields are used to group data and provide data analysis from different perspectives. Only string, alphanumeric, integer and date fields can be used as data fields. Other field types are disallowed because they are not suitable for grouping. For example, floating-point numeric types do not have the precision to

support the correct uniqueness, needed for successful grouping operation. One field can be either a column or row field, but not both at the same time.

### 3.2. Pivot Table

Pivot tables are well-known instrument. We follow the generally accepted behavior of pivot tables. Figure 1 demonstrates how our implementation of the pivot table looks like.

			2009				
Insurance Type ▲	Policy ▲	Claims ▲	Premium	Claim Paid In Cl...	Costs paid	Total Claim	Loss Ratio
[-] 8610 Total			20,266.79	214,788.02	40,956.73	255,746.14	1,261.90 %
[-] 2009.00251 Total			626.81	214,788.02	18,081.18	232,870.59	37,151.70 %
A2009.0005			626.81	89,882.94	18,081.18	107,964.12	17,224.38 %
A2009.0087			626.81	19,901.31	0.00	19,901.31	3,175.01 %
A2009.0088			626.81	52,884.60	0.00	52,884.60	8,437.10 %
A2009.0095			626.81	0.00	0.00	0.00	0.00 %
A2009.0165			626.81	0.00	0.00	0.00	0.00 %
A2009.0166			626.81	3,896.76	0.00	3,896.76	621.68 %
A2009.0332			626.81	0.00	0.00	0.00	0.00 %
A2009.0458			626.81	0.00	0.00	0.00	0.00 %
A2009.0582			626.81	21,084.26	0.00	21,084.26	3,363.74 %
A2009.0743			626.81	27,138.15	0.00	27,138.15	4,329.57 %
A2009.0744			626.81	0.00	0.00	1.39	0.22 %
[+] 2009.00254			19,639.98	0.00	22,875.55	22,875.55	116.47 %
[+] 8650			454.43	0.00	0.00	0.00	0.00 %
Grand Total			20,721.22	214,788.02	40,956.73	255,746.14	1,234.22 %

Figure 2. Pivot table

### 3.3. Extensions support

Pivot Reports will be extensible. Developers will be able to define custom calculations to be performed on data and produce specific reports.

There are two scopes of custom calculations: *server-side* and *client-side*.

Server-side calculations will be performed by application server after data is retrieved from the database, but before being transferred to the client. For example, Dupi Statistics will calculate total claim based on selected options and convert amounts to other currencies. Normally, all custom computations must be performed server-side.

Client-side calculations will be performed by the Pivot control of the Client. They are provided for specific calculation of aggregated values, when built-in standard functions (see below) are not applicable. For example, the total may need to be calculated by a formula from the other sub-total values, and not from all the underlying data.

Calculations are performed on the data after being retrieved from the database. We define an interface with three methods.

```
interface IDataProcessor
{
    void InitProcessing(ShellApplication application,
        SelectedColumn[] selectedColumns, List<ParamValue> paramValues);
    void ProcessRecord(DataRow row);
    void ProcessResult(List<DataRow> rows);
}
```

Method `InitProcessing` is used to initialize the state of the class that will implement the interface. It provides the class with the metadata about the data it will need to process and with additional options about the calculations to be performed. These options can be defined by the user in a custom dialog in the user interface of the application.

Method `ProcessRecord` is invoked once for every record from the retrieved data. The implementation of this method is expected to modify one or multiple fields from the row according to the specific business case.

Method `ProcessResult` is invoked only once after the last call to `ProcessRecord`. This method is intended for cases ranking calculations, i.e. when the calculations require overview of all data.

When developers implement this interface in a class, they need to link it to the entity on the database, where the calculations will be required or applicable. An entity defined in the DBD can have an attribute `DataProcessorClass` which is used to make all custom calculations for particular entity. The `DataProcessorClass` must implement the `IDataProcessor` interface.

```
<Entity PhysicalName="VIEW$DPI_STATISTICS" Name="DpiStatistics" IsReport="true"
    DataProcessorClass="DpiStatisticsDataProcessor" MainTable="CLAIMS" />
```

It is the responsibility of the developers to guarantee that the fields, required for the custom calculations will always be present in the pivot report.

### Summarizing Parent Data

When a data field is not from the entity of the report but from one of its parents, its value must be considered only once per unique occurrence.

For example, consider the following structure. A policy has a field value with its premium. Claims are related to policies, i.e. for a policy we can have zero, one or many claims. In a report on claims, where policy premium is selected as data field, a policy can have multiple claims and thus appear many times in the result. The sum of the total premium must consider only the unique policies, not the policy for every claim record.

The PRT automatically detects fields from parent tables. When calculating the values, it eliminates all unwanted repetitions and produces correct results.

Let’s consider the example of two policies, where one has two claims, and the second has no claims, as shown in Table 1.

Policy Number	Policy Premium	Claim Number	Claim Premium
2015.00001	1000	A2015.0001	500
2015.00001	1000	A2015.0002	400
2015.00002	800	-	-
<b>Total</b>	<b>1800</b>		

*Table 1. Policies with claims*

Standard pivot tables will list 2800 because they will aggregate policy 2015.00001 twice, which is incorrect. The RPT properly calculates that total premium is 1800.

#### 4. Conclusion

This paper has described the functional requirements, architecture and implementation of a pivot reporting tool as a standard functionality to the Framework for Distributed Business Applications. The approach we have chosen provides users with an easily accessible tool to perform data analysis.

In future, we will investigate methods to allow users to add summarized values to their current reporting entity from children entities.

#### Acknowledgements

This work is supported by the IT15-FMIIT-004 project of the Scientific Fund of the University of Plovdiv “Paisii Hilendarski”, Bulgaria.

## References

- [1] Rud, O., *Business Intelligence Success Factors: Tools for Aligning Your Business in the Global Economy*, Hoboken, N.J: Wiley & Sons, 2009, ISBN 978-0-470-39240-9.
- [2] Rao, R., From unstructured data to actionable intelligence, *IT Professional* 5 (6): 29, 2003, ISSN 1520-9202.
- [3] Feldman, S., Sherman C., The High Cost of Not Finding Information, IDC, Apr.2003, IDC #29127.
- [4] Imhoff, C., Three Trends in Business Intelligence Technology, 2006, BeyeNETWORK, <http://www.b-eye-network.com/view/2608>, retrieved on November 3, 2015.
- [5] Jelen, B., M. Alexander, Pivot table data crunching, Indianapolis, p. 274, 2006, ISBN 0-7897-3435-4.
- [6] Pavlov, N., Rich Metadata Model for Business Applications with Database Dictionary, *International Journal of Applied Science and Technology* (IJAST), Vol. 4, No. 2, 2014, ISSN 2221-0997, pp 58-66.
- [7] Pavlov, N., *Object-Oriented Framework for Development of Distributed Business Applications*, Ph.D. Thesis, Plovdiv University, Plovdiv, Bulgaria, 2011, (in Bulgarian).

Faculty of Mathematics and Informatics  
Plovdiv University "Paisii Hilendarski"  
236 Bulgaria blvd.  
4003 Plovdiv, Bulgaria  
E-mails: nikolayp@uni-plovdiv.bg, maria.d.dobрева@gmail.com

## ИНСТРУМЕНТ ЗА АНАЛИЗ НА ДАННИ

**Николай Павлов, Мария Добрева**

**Резюме:** В тази статия е описан инструмент за анализ на данни чрез Pivot таблици, разработен като част от рамката за разпределени бизнес приложения – FDBA. Представени са основните функционални изисквания и

тяхната реализация. Нашият инструмент може да извлича и анализира данни от няколко таблици на базата данни. Основна характеристика е правилното обобщаване на стойности от свързани таблици на базата данни.

**Ключови думи:** анализ на данни, пивот таблица, FDBA