8/2 (2010), 197–210
tmcs@math.klte.hu
http://tmcs.math.klte.hu

Teaching
Mathematics and
Computer Science

Information System’s experiences of EGERFOOD project making use of it in the education of the database management

Tibor Radványi, Emőd Kovács and János Kormos

Abstract. We present in this article the background of a developed food safety tracking system searched and formed in the Regional Knowledge Centre of Eszterházy Károly College, the requirements following from this, and by way of the requirements towards the information system appearing expectations. The development of the consumer centre system is a complex task which provides fast and cost-effective information for consumers, food producers and concerned authorities. It accomplishes severe expectations of the tracking system in connection with data security and encryption beside all this. We demonstrate in this article that forming of database model why we chose the general model. We also demonstrate what kind of SQL server we chose for buffer servers and central data warehouse. We wish to support our choosing with the result of done efficiency examinations. It is important viewpoint what kind of database planning principles we base these examinations on and how we match them to the requirements of the system. As software engineers took part in the development effectively from the first minute of the planning of the system, we can examine with what this project work was able to raise students’ qualification and knowledge in addition to the general curricular substance.

Key words and phrases: database management, information system, performance of queries.

ZDM Subject Classification: P20, R50.

Antecedents

In the focus of the research there are service activities currently environment protection and food analytics works, from which the activities that are connected to food analytics and food safety are the most important. It follows from this

that established food safety and analytics examination centre can be regarded as the logical continuation of the activities until now, the forming of certain new priorities and extension of economical the most relevant research topics.
The Regional Knowledge Centre created at Eszterházy Károly College wants to contribute to increasing competitiveness of home foods with improvement of innovation abilities and spreading the results of the home food safety research in keeping with North Hungarian Innovation Strategy.

Informatics tasks

1. Creation and constant operation of the web system which operates the inner communication of the project.
2. Defining the structure of the tracing database, development of the software and hardware form of the data transmission. The marrow of the information system is the database of the tracing system. We analysed data and gathered demands so we created the data model of the information system on the basis of it.
3. We worked out an algorithm and a code system , regarding the long-term development strategy of the tracing system, that are suitable for product identification, the certificate of guarantee of the product justifies appearing in tracking system and carries a code with code system. It is important that data is stored and moved with right cryptographic method.
4. Our tasks are to plan and produce user interfaces for different data collecting and querying activities.
5. We developed the aspects of the communication on WAP and Internet with consumers. Appearing common food safety information on different platforms has been carried out.
6. We planned and realized the safety requirements of the full information system and the safety procedures.

1. Claims affecting the implementation plan

The central data warehouse undertakes to store data finally and serve querying and display modules. The first question is how to send data extracted over to the data warehouse. There are more alternative possibilities, the first one is data sources in online contact by supplying data continuously towards centre. The

second option is, that the data source is offline and we update periodically. The second opportunity was selected, because the depots (data sources) have totally different data traffic opportunities.
Look at the data sources: [5]

Data come from two large areas: received results of college research laboratories and received data on remote industrial sites during production. These data can come through 24 hours while results of laboratories come periodically according to measurement experiments. The outer sites are places of production of six different food industry companies, their geographical positions; their equipment and facilities are totally different. Food industry products made by them are also different so composition of examined data and time intensity of their origin are different on product curves because of difference of manufacturing technology. Following from these differences the network and the information system, which realizes data storage, has to be prepared for it. We can see that building a large-sized central data warehouse is important because expected amount of data. Time intensity of incoming data demands a large network cross section.

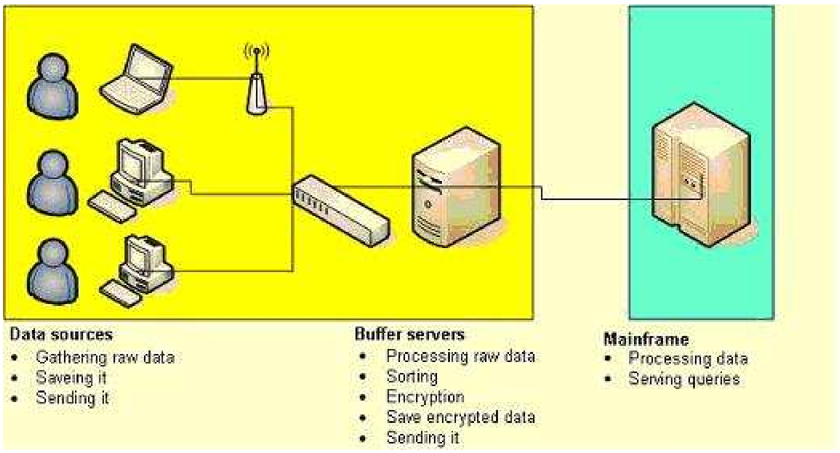

Figure 1. From data source to Mainframe

The task of the local servers placed out to industrial firms, call them buffer servers, is to pre-send the data of computers and measuring instruments belonged to their territory and to send data to the central server grouped in a right system. This can be seen on Figure 1. Software on these computers encrypt data, too. [3]
Important question is how we can prevent data loss. We have to examine its two

sides. One of them is the long-term defence of incoming data, which is ensured by well designed archiving order. The other one is the short- and long-term defence of the data that were made at the data sources but are not on the central server. The short-term defence is the kind of possibility which means that the first autosave has to be done as soon as possible to the moment of the data origin. Then it is necessary to save the pre-processed, encrypted, prepared local data for sending. So data has already been stored in two independent places before getting into the data warehouse of the central server. Such measure data redundancy seems to be quite source of energy demanding, but these measurements seem to be necessary if we want to meet the requirements.

There is a very important question that what kind of database management system should be used. We involved the students into the decision mechanism and utilized the advantages of cooperative project teamwork. We investigated the teams' suggestions together and we realised plausible problems. These problems can be caused by different beliefs or economical calculations and project meeting results in belief debate or economical counting. The consequence is made by the students themselves; without more investigation and mensuration the decision can't be made. A well-planned investigation of efficiency is required.

2. Investigating database efficiency

Data model

After reviewing the data we have decided to make a general data model. We can see in Figure 2. – which can be used to store data without reference to any companies. [5] Its usage is more complicated but it can speed up the later development significantly.

Two tables are containing the data of the companies and their products participating in the consortium. (*Company, Product*)

This allows to add companies to the investigation later and to add new products for the companies that are already added.

There are two tables containing the attributes of the data: its type and size (*Attribute, Attribute_Type*).

It's recommended for making the right way of conversions.

There are three tables containing the logs, its data lines and the elementary data stored in the lines. (*Log, Log_Row, Row_Element*)

The *Log* table contains data about a companies' products shift.

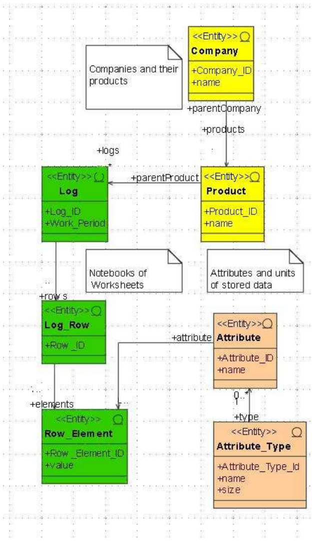

Figure 2. The Database plan

There are company identifier, product identifier and shift identifier. The shift identifier can be divided into minor units depending on the apportionment of production. A *Log* entry can include an optional line which means adding data. Only the logically cohesive data are going to be recorded. e.g. Mary Smith on the 15th of December, 2006, during second shift at the 10th tent culled 3kg mushroom. The *Row_Element* table contains the elementary data, the line identifier and the attribute identifier. According to this the data can be securely retrieved for the data adding forms or for the investigations.

A dictum

Using the open source database management system can be 60 percent cheaper than using a commercial database management system.

Leastwise, according to the freshly released study of Forrest Research marketing research institution, which contrasts the benefits and disadvantages of these two different types.

Forest’s study admits that the commercial database management systems usually offers higher level functions, more services for the developers according to the open source ones, but this advantage is not used by most of the time.

According to the study, 30% of the higher level functions were used and even the open source ones could offer these functions most of the time.

According to the authors, at the same time we must not forget that there are some parts where the open source software can't come short of the commercial ones because of lacking their performance, security and availability.

Most open source database management software are less crucial, usually working with low – less than 200GB – amount of data; similar solutions like MySQL and Ingres can even produce better results than the expensive Microsoft, IBM or Oracle commercial software.

According to the summary of the study the open source system's most indisputable benefit is its price, in most cases that is why they are used not because of their technological details.

Primary users are developers, students and test project managers or rather some market where the expensive solutions can't come into question because of low profitability.

Examined databases:

1. Oracle 10g; 2. MSSQL 2005; 3. PostgreSQL 8.2 the operating system: Windows 2003.

Uploading Data

A software has been made to upload data to the SQL servers.

The data used in uploading can be random but picked from the given list by the sample and using the ratio between the companies.

Companies: Fish(3), Steak Hammerer(6), Rolling-pinner(6), Wine stiffer(1), Cookie Baker(5), Mushroom Picker(7), Researcher(12)

The numbers in brackets are the ratio of the companies. These can be the probabilities, divide them by 40. We used these probabilities in the example

Products:

Company	**Product**	**Probabilities**
Fish	Tinned fish	0,075
Steak hammerer	Devil Sausage	0,15
Rolling-pinner	Noodle	0,15
Wine stiffer	White Sweety	0,025
Cookie baker	Coco speciality	0,125
Mushroom Picker	Agaricus_Bisporus	0,175
Researcher	Own cookie	0,3

By these data the Company and Product tables can be loaded. The Attribute and Attribute_Type tables can be loaded necessarily by the recordable data.

Let’s see an example

On 15th of December in 2006 Mary Smith culled at the second shift at the 10th tent 3kg mushroom. To debug the Mushroom Picker company’s Agaricus_Bisporus product on the 15th of December in 2006 at the second shift storing the data like this:

Date	worker	shift	Place	Product	kg
15.12.2006	Mary Smith	2	10	Champignon	3

<table>
<tr><td><i>Company</i> table:</td><td>Mushroom Picker (id: 3) it’s given, we just need make a reference</td></tr>
<tr><td><i>Product</i> table:</td><td>Agaricus_Bisporus (id 17), and the Mushroom Picker company’s ID (3, foreign key: FK)</td></tr>
<tr><td><i>Log</i> table:</td><td>Log_id, Product_id, Work_Period
Log_id to generate, (200100)
Product_id FK → search (17)
Work_period: given data (2)</td></tr>
<tr><td><i>Log_Row</i> table:</td><td>Log_Row_id, Log_Id, Date
Log_Row_Id to generate, (100103)
Log_Id FK → search, (200100)
date: given data (2006.12.15)</td></tr>
<tr><td><i>Row_Element</i> table:</td><td>Row_Element_ID, Log_Row_Id, Attribute_Id, Value
Row_element_Id: to generate
Log_Row_Id: FK, search
Attribute_ID: FK, search
Value: the given cell’s type is string. Pl.: Mary Smith
<table>
<tr><th>Row_Element_ID</th><th>Log_Row_Id</th><th>Attribute_Id</th><th>Value</th></tr>
<tr><td>12345</td><td>100103</td><td>15</td><td>Mary Smith</td></tr>
<tr><td>12346</td><td>100103</td><td>19</td><td>10</td></tr>
<tr><td>12347</td><td>100103</td><td>3</td><td>3</td></tr>
</table></td></tr>
<tr><td><i>Attribute</i> table:</td><td>Attribute_ID to generate
Attribute_Type_ID: FK search
Name: string , the attributes name, description For example: name or tent
<table>
<tr><th>Attribute_ID</th><th>Attribute_Type_ID</th><th>Name</th></tr>
<tr><td>15</td><td>51</td><td>Name</td></tr>
<tr><td>19</td><td>64</td><td>Tent</td></tr>
<tr><td>3</td><td>34</td><td>Agaricus_Bisporus</td></tr>
</table></td></tr>
</table>

<table>
<tr><td><i>Attribute_Type</i> table:</td><td>Attribute_Type_ID: to generate
Name: the attributes date type is string
Size: Size of the data type
<table>
<tr><th>Attribute_Type_ID</th><th>Name</th><th>Size</th></tr>
<tr><td>51</td><td>String_80</td><td>80</td></tr>
<tr><td>64</td><td>Integer</td><td>4</td></tr>
<tr><td>34</td><td>kg</td><td>4</td></tr>
</table>
</td></tr>
</table>

The *Log, Log_Row, Row_Element* tables are loaded, first of all is the *Row_Element* table.
The *Attribue* and the *Attribute_Type* tables are impregnating after several dozens or maybe a hundred records of data. Thus like, during the query we must choose of the order of data.
To upload the records, choose from the given companies with a given probability then use the one appertaining product.
Decide the shift number randomly from the following probabilities:

1st shift: $0,4\,(40\%)$
2nd shift: $0,4\,(40\%)$
3rd shift: $0,2\,(20\%)$

Choose a date from the $[01.01.2006, 31.12.2006]$ interval with the following method:
It should be able to change the number of lines at the log for the Wine stiffer company. For the other companies the given probabilities will determine the multiplier.
For example if we pick 10 lines from the Wine Stiffler company it results 50 lines from the Cookie Baker. In this case there are 400 lines generated a day. This means 1500 records a day which is 550000 records in the period under survey.
The attributes are going to be generated.

The examination of SQL servers [2][4][1]

The project was finished, which uploads the basic data, and later it fills *Log, Log_rows, Row_elements* tables of the MSSQL 2005 and the Oracle with data. It handles the database under PostgreSQL server. This software was made with Visual Studio 2005, in C# language, as far as more software of project. [6] The students could be taken up with the usage of latest development tool of OOP. They could use their knowledge and they had to find connection between the software and three kinds of SQL server. About 3 million records were filled into the databases with the assistance of the software. Their repartition is aforementioned according to method. The time-requirement was examined already under the filling. Behaviour of database-servers was measured from the point of view of data-insert. The experience was that there wasn't divergence between MSSQL2005 and Oracle, but the PostgreSQL 8.2 demanded 2,5 times longer time. So the PostgreSQL server accomplished very weakly. The windows version has serious performance problems, but the Linux version hasn't it.

3. Examination of queries

The second part of examination studied the effectiveness of the queries with the help of uploaded data. [7]
a) Restoring the given row of the given log file.

```
SELECT    Company.Name, Product.Name AS productName, [Log].Work_Period,
[Log].Log_ID, Log_Row.Datum, Log_Row.Log_Row_ID, Row_Element.Value,
                  Attribute.Name AS AttName, Attribute_Type.Name AS
                  AttTypeName
FROM          Product INNER JOIN
                  Company ON Product.Company_ID = Company.Company_ID
                  INNER JOIN
                  [Log] ON Product.Product_ID = [Log].Log_ID INNER JOIN
                  Log_Row ON [Log].Log_ID = Log_Row.Log_ID INNER JOIN
                  Row_Element ON Log_Row.Log_Row_ID =
                  Row_Element.Log_Row_ID
INNER JOIN
                  Attribute INNER JOIN
                  Attribute_Type ON Attribute.Attribute_Type_ID =
Attribute_Type.Attribute_Type_ID ON Row_Element.Attribute_ID =
Attribute.Attribute_ID
WHERE (Log_row.Log_Row_ID= 2)
```

b) In given time-interval made log files of given company and their rows.

```
SELECT    Company.Name, Product.Name AS Prod, [Log].Log_ID,
[Log].Work_Period, Log_Row.Datum
FROM          Company INNER JOIN
                  Product ON Company.Company_ID = Product.Company_ID
                  INNER JOIN
                  [Log] ON Product.Product_ID = [Log].Product_ID INNER
                  JOIN
                  Log_Row ON [Log].Log_ID = Log_Row.Log_ID
WHERE (Company.Company_id = 3) and (Log_Row.Datum between '2008-01-01' AND
'2008-01-31')
```

c) All log file ID of a company.

```
SELECT    Company.Name, Product.Name AS Prod, [Log].Log_ID,
Company.Company_ID
FROM          Company INNER JOIN
                  Product ON Company.Company_ID=Product.Company_ID INNER
JOIN
                  [Log] ON Product.Product_ID=[Log].Product_ID
WHERE    (Company.Company_ID=1)
```

These basic queries were run. It was an important question what kind of indexes should be used. The students drew up formulas to queries in a database and used indexes according to their own experiences. So they attained the knowledge of the database-system in this practical way.

What kind of indexes we should use

Microsoft Server SQL Manager recommends the following indexes to extend effectiveness:
a) query:

```
use [TestDatabase]
go
CREATE NONCLUSTERED INDEX [_dta_index_Row_Element_10_21575115__K2_K3_4] ON
[dbo].[Row_Element]
([Log_Row_ID] ASC, [Attribute_ID] ASC)
INCLUDE ([Value]) WITH (SORT_IN_TEMPDB=OFF, DROP_EXISTING = OFF,
IGNORE_DUP_KEY=OFF, ONLINE=OFF) ON [PRIMARY]
go
CREATE NONCLUSTERED INDEX [_dta_index_Log_Row_10_2137058649__K1_K2_3] ON
[dbo].[Log_Row]
([Log_Row_ID] ASC, [Log_ID] ASC)
INCLUDE ([Datum]) WITH (SORT_IN_TEMPDB=OFF, DROP_EXISTING=OFF,
IGNORE_DUP_KEY=OFF, ONLINE=OFF) ON [PRIMARY]
go
CREATE STATISTICS [_dta_stat_2137058649_2_1] ON [dbo].[Log_Row]([Log_ID],
[Log_Row_ID])
go
CREATE NONCLUSTERED INDEX [_dta_index_Log_10_5575058__K1_3] ON [dbo].[Log]
([Log_ID] ASC)
INCLUDE ([Work_Period]) WITH (SORT_IN_TEMPDB=OFF, DROP_EXISTING=OFF,
IGNORE_DUP_KEY=OFF, ONLINE=OFF) ON [PRIMARY]
go
```

b) query:

```
use [TestDatabase]
go
CREATE CLUSTERED INDEX [_dta_index_Log_c_10_5575058__K1_K2] ON [dbo].[Log]
([Log_ID] ASC, [Product_ID] ASC) WITH (SORT_IN_TEMPDB = OFF, DROP_EXISTING
= OFF, IGNORE_DUP_KEY = OFF, ONLINE = OFF) ON [PRIMARY]
go
CREATE STATISTICS [_dta_stat_5575058_1_2] ON [dbo].[Log]([Log_ID],
[Product_ID])
go
CREATE NONCLUSTERED INDEX [_dta_index_Log_Row_10_2137058649__K3_K2] ON
[dbo].[Log_Row]
([Datum] ASC, [Log_ID] ASC )ITH (SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF,
IGNORE_DUP_KEY = OFF, ONLINE = OFF) ON [PRIMARY]
go
```

c) query:

```
use [TestDatabase]
go
CREATE NONCLUSTERED INDEX [_dta_index_Log_10_5575058__K2_1] ON [dbo].[Log]
([Product_ID] ASC)
INCLUDE ([Log_ID]) WITH (SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF,
IGNORE_DUP_KEY = OFF, ONLINE = OFF) ON [PRIMARY]
go
```

The measuring gave the following results:

There isn't any difference between the response time to queries of MSSQL 2005 Server and Oracle Server. We decided to use the MSSQL 2005 Server as the information system is developed in VS 2005 C# programming language.

Utilization of the project's experiences: subjects, connection between subjects

Database Systems

The students have to learn the basic concepts of database management:

- data independence,
- data integrity,
- data protection.

In connection with the planning, they become acquainted with the entity-attribution-relationship model. They familiarize oneself with the features of relational data model and base on this with the parts and structure of the SQL language. On the practises SQL language is in the focus. First of all, creating queries and applying them in one and multi-table databases.

Advanced DBMS

The lecture's subject is the PL/SQL language. After learning the basic parts, data types the expressions, executable commands, control structures are studied. The known SQL commands are examined how to use in PL/SQL. Study the program units: block, subroutine, respectively exception handling. Creating stored subroutines then examine the cursors, cursor variables. Study triggers, respectively packages. On the practises these are implemented in Oracle DBMS with SQL Developer.

High-Level Programming

Primarily the students study C# and Java languages. During the two semesters they are getting to know the language's syntax. They become acquainted with the basic framework classes. They get acquainted with the principles of object oriented programming and the utilization of them in case of the above-mentioned languages. On the practises they get acquainted with the Visual Studio's services through simpler programs coding. Within the confines of the course there isn't enough time to code bigger volume projects. Windows Application respectively ASP.NET systems developing.[6] The aim is to practise the basic algorithms.

Consequences

Let's sum it up, what kind of knowledge should have the students used and improved in this project. During the planning of the database they have participated in a demand survey. After, the forthcoming discussion they have experienced the problems arisen and they've participated finding the answers, too. They've efficiently participated in the preparation phase of the test database, the test program's development, measurement, and evaluation of the obtained result.

Therefore they could have tested and expanded their programming and database skills they've learned at the lessons during an intense project. They could have developed using the latest technology improving their skills.

Results:

- The universal database structure makes boundless expandability possible, that is to say however many company can adapt the model system.
- The recall of sub-standard product can realize within a short time, with necessary accuracy, with a minimal expense, with the possible smallest prestige losing.
- The tracing information is stored in central data-warehouse, therefore the chance of product's falsification and manipulation become minimal.
- The technological parameters ahead from a granted province reaching truth on his case generate automatic alarm.
- The origin identity of the product on a reliable manner is verifiable.
- The consumers themselves may query the food quality data with the help of a WAP telephone or internet.

On the whole it could be found that participating in the project has helped improving their knowledge on the way to put it in a good use after they finish their studies, taking up a position in the economy.

The students' carried out reviews were that the participation in such projects is advantage of the work in new jobs.

It is mostly recommended to manage projects like this to participate students.

References

[1] A. Ailamaki and M. Shao, DBMbench: Microbenchmarking Database Systems in a Small, Yet Real World, *Confidential*, submitted to ICDE 2004.

[2] J. Gray, *The Benchmark Handbook for Database and Transaction Processing Systems*, 2nd edition, Morgan Kaufman Publishers Inc., 1993.

[3] K. Liptai, G. Kusper and T. Radványi, Cryptographycal Protocols in the Egerfood Information System, *Annales Mathematicae et Informaticae*, Eger, Hungary (2007), 61–70.

[4] T. Radványi, Examination of the MSSQL Server from the user'point of view considering data insertion, *Acta Academiae Paedagogicae Agriensis, Sec. Mathematicae* (2004), 69–77.

[5] T. Radványi and G. Kusper, *Requirement Analyzes and a Database Model for the Project EGERFOOD Food Safety Knowledge Center*, 7th International Conference on Applied Informatics, Eger, Hungary, January 28–31, 2007, 15–25.

[6] Microsoft: Improving.NET, Application Performance and Scalability, 2004, 639–682.

[7] Mike Ruthruff (Microsoft Co.), *Microsoft SQL server 2000 Index Defragmentation Best Practices*, 2003.

TIBOR RADVÁNYI and EMŐD KOVÁCS
ESZTERHÁZY KÁROLY COLLEGE

E-mail: dream@aries.ektf.hu

E-mail: emod@aries.ektf.hu

JÁNOS KORMOS
UNIVERSITY OF DEBRECEN

E-mail: janos.kormos@econ.unideb.hu

(Received August, 2008)