

## SEMINAIRE

## СЕМИНАР

## SEMINAR

STATISTICAL COMMISSION AND  
ECONOMIC COMMISSION FOR EUROPECONFERENCE OF EUROPEAN  
STATISTICIANSDistr.  
GENERALCES/SEM.43/5  
13 March 2000

ENGLISH ONLY

Seminar on integrated statistical information  
systems and related matters (ISIS 2000)  
(Riga, Latvia, 29-31 May 2000)

Topic I: Data warehousing and the development and use  
of statistical databases in a network environment

**THE APPLICATION OF DATA WAREHOUSE TECHNIQUES IN A STATISTICAL ENVIRONMENT****Invited paper**Submitted by Statistics Netherlands<sup>1</sup>**I. PRINCIPLES OF THE DIMENSIONAL MODEL**

1. The concept of the data warehouse is, of course, nothing new to Statistics Netherlands (CBS). Those who can remember boxes of punch cards know that they were in fact a data warehouse, with the punch cards representing the fact table and the code lists the dimensions. Maybe it's time to resume what we were doing...

**I.1 Query versus transaction****I.1.1 What is OLAP?**

2. The abbreviation OLAP stands for *On Line Analytical Processing*, which means that the database can be expected to give a very fast response to "big" questions. To this end, the database not only possesses special middle-tier applications, but is also specially modelled, with redundancy preferred to standardisation.

**I.1.2 Is the production and analysis of statistics OLAP?**

3. Yes, basically, although a distinction can be made between two types of statistical production. First, there is standard, regular production, i.e. pre-determined overviews, which will eventually form part of a CBS publication. This type of output is certainly OLAP, but it does not provide the necessary and sufficient conditions for a data warehouse. Later on, we shall see why we still think a data warehouse is the right solution.

4. The second type of statistical production analysis, is an activity for which a data warehouse can be very valuable. After all, next to machine-based methods of **data mining**, human analysis is still the best method of obtaining the most

---

1 Prepared by Marton Vucsan.

interesting facts from raw data. You can think of this as a refining process: the raw data in the *data warehouse* have to be correlated in various ways by humans, so that information can be extracted from them.

5. Lastly, the future of the statistical production process is one in which "ready-made" statistics presented in paper form will increasingly be displaced by automated means which, within certain limits, will allow *ad hoc* statistics for remote and other users to be compiled electronically. One of the CBS' traditional activities, the compilation of aggregates, will become a threatened species.

## **I.2 Consistency**

### ***I.2.1 OLAP is consistent at global level***

6. A data warehouse works completely differently to an OLTP database. Not only do a number of data have a "calculated accuracy", but not all data might be present. "Calculated accuracy" denotes the fact that, when an event or "fact" such as a store sale has a "margin" column, that margin is of course a number which originated in the accounts and which has worked its way down from the global level. At this level, the figure may be incorrect. Is the margin on a bar of chocolate really 0.001 cents? If, however, we compile aggregates from the data warehouse and end up with the totals for this column, then we are back where the concept of "margin" was defined.

7. The fact that not all the data may be present is not, of course, due to a structural lack of input, but to the arbitrary removal of obvious inaccuracies or minor errors in data handling. If they are arbitrary, no harm is done, as long as correct grids can be compiled.

### ***I.2.2 OLAP is consistent over time***

8. Once data have been entered in a data warehouse, they remain there for good. The result for a given period stays the same. A data warehouse is essentially a time series. But we will discuss more about that later.

## **I.3 The dimensional model**

### ***I.3.1 The data cube***

9. This concept is of particular importance in statistics. The idea is that you can always distribute data over several axes. The cube metaphor is appropriate because two variables are viewed over time, thus creating a kind of three-dimensional space, i.e. a cube, possibly with marginal values. This term is also used for thematic areas in a data mart. StatLine makes use of this technique.

### ***I.3.1 The star join***

10. Storing cubes in a relational database is not too difficult as long as one realises that the cubes tend to be 'sparse', i.e. the data are distributed sparsely among the available cells. Storing these cells, of course, entails no more than creating a table with fields for all the dimension keys of the cube and the cell values and to start the loading process.

11. The dimensions are therefore the column headers, no more and no less. Of course, we put these dimensions into tables (the dimension tables). Think, for example, of municipal codes.

12. Querying the cube therefore always entails joining the table containing the data and the dimension tables. The 'WHERE' clause is the limiting condition applied to the data in the dimension tables, for example 'where municipality name = AMSTERDAM'

13. A star join therefore always involves a large table (the fact table) and a number of small tables (the dimensions). The fact table is measured in GIGabytes and

the dimension tables are of the order of 100 MEGabytes. The database optimizer has special solutions.

14. So, what's new? We have always been doing that at the CBS. What is new is the idea that we are dealing with a new discipline, rather than a one-off solution for a software problem.

### **I.3.3 The data mart**

15. A star-join configuration with a fact table and several dimension tables is known as a data mart. A data mart contains information about a specific subject. That subject is almost always a process. The following are some examples:

- Store sale of product to customer at a particular time
- Customer's journey in vehicle to and from destination
- Delivery of goods from loading platform to customer on a particular date
- Treatment of patient in hospital at a particular time
- Balance of customer account at a particular time
- Presence of bird in breeding grounds at a particular time.

The nice thing about a data mart is that the data are not free standing, but are (or should be) part of a data warehouse.

### **I.3.4 The data warehouse**

16. The strength of a data warehouse is that it consists of a number of data marts which are dovetailed to each other. This is done by harmonising the dimensions. The municipality code table or the dimension 'LOCATION' for all the data marts must of course be either identical or a subset or superset.

17. A data warehouse, which comprises data marts in this manner, has a very high added value. Of course, one can relate any number of processes to each other, for example, the store sales data mart and the purchases data mart, to see how much has been stolen or broken.

18. It is important to realise that dovetailing the various data marts and their dimensions is a logical concept. It is not at all necessary for two data marts to be located in the same physical database or in the same computer. We will never approach two data marts with a single SQL statement because that takes much too long. There are much better solutions, such as running two separate queries and letting the client application merge the results.

## **I.4 Fact and dimension tables**

### **I.4.1 What is a dimension table?**

19. We have already looked at an example of a municipality code table as a (fairly simple) dimension. Basically, a dimension table is just the name of the column. Things become more complicated if we look more closely. Let us take another look at the municipal codes. The dimension "municipality" is unlikely to be of much use in many data marts: something along the lines of a generalised location dimension is probably more suitable.

20. What does a location dimension look like? Firstly, of course, we will want to enter the municipality codes. But we will also want to enter the hamlets and the provinces. We now see that the municipal code may not be sufficient for our needs, and we move up to a 4- or 5-digit integer. This is just a number like any other and has no intrinsic significance. We give each record a new number, which then becomes the **primary key**.

21. This becomes clear if we look at a fragment of this dimension:

	<i>key</i>	<i>province</i>	<i>municipality</i>	<i>hamlet</i>
•	00234	South Holland	Alkmaar	Oude-pekela
•	00233	South Holland	Alkmaar	Nieuwe-pekela
•	00232	South Holland	Beverwijk	Beverwijk

This dimension allows us to join not only at municipal level, but also at hamlet and provincial levels.

#### **I.4.2 What is a fact table?**

22. A fact table contains the process variables which we are interested in. If we look at the process 'store sale', we want to know not only when what was sold to whom, but also how much was sold and what it cost. It does not make much sense to create a dimension containing a vast array of amounts and quantities. We may also want to add them up some time. What we are doing is creating a record with a primary key which consists of "foreign" keys to the dimensions and one or more attributes which denote quantities or other variables which can be expressed numerically.

23. This becomes apparent if we look at a fragment of the fact table:

	<i>loc.</i>	<i>time</i>	<i>prod</i>	<i>customer</i>	<i>no.</i>	<i>amount</i>
•	00234	00011	88234	211154	2	400
•	00233	00003	78986	329809	1	3400

24. Using the combination of the dimension keys *location*, *time*, *product*, *customer*, it is now possible to record when "something" is sold "somewhere" by "someone". The only information which was not yet known is the quantity and cost, but that information can be found elsewhere in the record. The four foreign keys together form a combination of textual attributes which describe the event perfectly. The fact table contains nothing more than numerical data. The two main reasons for this are compactness and the desire for the data in the fact table to be countable. The desire for the attributes in the fact table to be countable means that the amount is a total, otherwise **number\*price** would constantly recur in the SQL, which would seriously impact on performance. If we want to know the price per item, we can do so either in the product dimension (if the price does not change too often) or by calculating **price/number**.

25. Another important feature is that the keys have no intrinsic meaning and consist only of numbers which have been assigned to them. This is to avoid conflicts with changes made to the data formats by the suppliers of the data.

#### **I.5 Time series**

26. The statistical process could be described in abstract (and highly simplified) terms as the *observation of variables at a particular moment in time*. With a little effort, it is possible to see any set of statistics as a time series.

##### **I.5.1 A data warehouse is a time series**

27. Data stored in a data warehouse have, to put it mildly, a static character. The aim is not for data already stored in the data warehouse to be amended, although that is an option. A data warehouse is only ever topped up, it is never refilled. The data in a data warehouse describe processes and the course which they follow over time. This makes data warehouses extremely well suited for compiling statistics. Today's data warehouses are almost all used for compiling statistics, although aggregation is, of course, done by the client himself.

### ***1.5.2 Changing dimensions***

28. One of the knottiest problems associated with statistics is that of changing codes. This problem also occurs in data warehouses and has given rise to a number of solutions which are very similar to ones already adopted by the CBS. The main solution for changing codes is to use a denatured key, i.e. a arbitrary integer. If one of the attributes has changed, a new record with a new key is made. As soon as new events occur, the new combination is used. The nice thing about this solution is that code changes do not present the slightest problem.

#### **Example:**

29. In the dimension **location**, the hamlet of Baarsland is transferred from the Municipality of Rijnsburg to the Municipality of Rijnsoude (both of which are in the Province of South Holland) as of 1 November 1996. Queries concerning South Holland are not affected by the change. Nor, of course, are queries using the underlying grid. Queries which aggregate up to municipal level should reveal a shift affecting both municipalities. The question now is whether this is desirable.

30. A table in the statistical yearbook cannot simply record a shift between two municipalities without further explanation, since the user would have no way of telling what had caused the shift. With a data warehouse, the user is expected to interpret the results himself. We can therefore expect a user who discovers anomalies in the data for the two municipalities to run a few simple queries and compare the data by year, municipality, etc. in order to check that the shift in the first query was not caused by relatively minor things like changes to municipal borders. One may conclude that entering this sort of change in the dimension table does not give rise to problems. Time will tell.

### ***1.5.3 Time is a separate dimension***

31. Time is, of course, a separate dimension. No one would dispute that. Yet, the manner in which we treat it in a dimensional model is not what you might expect. At first glance it would seem reasonable to use the date in the internal database format in the records of the fact table, or a number which represents **yyymmdd**, which is familiar from a wide range of statistics. This approach is not very practical, however.

32. Let us take a look at some of the problems associated with entering dates in fact tables:

- Tiresome conversions to week numbers, etc.
- Calculating dates means that queries take a long time to answer.
- Not until AFTER the fact table has been accessed do we know if Q1 76 is in the data base.

33. All these problems can be avoided by using a separate dimension table for time. The table stores all possible time representations using denatured numerical keys. The key (an arbitrary number) is then used as a link to the fact table.

34. The following is an example of what a time dimension table can look like:

key	day	month	quarter	year	leave status
00093	Wednesday	12	4	1977	0
00094	Thursday	12	4	1977	1
00095	Friday	01	1	1978	1

It is clear what is going on here: all possible times and dates are given in completely denormalized form. If I want to know what happened on Thursdays in the course of the years, the join is:

blabla WHERE day of week = 'Thursday' etc.

This is not very difficult and quite fast. Other tasks, like compiling monthly aggregates, can also be accomplished more easily.

## **I.6 Aggregates**

### **I.6.1 The need for aggregates**

35. With fact tables of more than several gigabytes, it makes sense to compile a few aggregates for some of the more frequently requested data. Generally speaking, the use of precompiled aggregates is the most important means of enhancing data warehouse performance. The reasons are obvious.

### **I.6.2 Automatic navigation**

36. Where aggregates are present in a data warehouse, using them to answer queries was previously a major problem, because it required fast access to a sort of data dictionary and software which allowed automatic navigation. The compilation of separate fact tables containing the aggregate and smaller tables for the aggregated dimensions ultimately proved the best solution for storing aggregates in the database. Incidentally, modern OLAP engines which provide the end user with data cubes have made aggregate management and navigation completely automatic.

### **I.6.3 Management**

37. Thanks to automatic navigation, the management of the aggregates has become a task for the database administrator alone. He is the one who uses the performance data obtained in the course of the day to decide if aggregates are required and, if so, which ones. Compiling aggregates is therefore a dynamic process which belongs in the management sector, this makes designing a data warehouse much easier.

## **II. COMPLICATIONS IN THE DESIGN OF STATISTICAL DATA MARTS**

38. The supermarket model is not appropriate for the design of a data mart which is to be used in a statistical environment. Things start to go wrong trying to define the process under observation. What process should be observed and recorded for examining population data?

39. Nor is the situation much better as regards the design of the dimensions. Not only is it unclear what dimensions there are: their attributes are usually difficult to identify.

40. In statistical environments, allowance also has to be made for the fact that a number of codes, which have traditionally had an important role in statistics, have to be included in the dimensions. The reason is that, in an analysis which requires aids other than standard query tools, these codes are often needed to avoid complex operations when extracting a data subset with a view to translating the texts back into code. Also, the design is usually influenced not only by data needs, but also by data availability.

## **III. APPLICATION OPTIONS IN THE STATISTICAL PRODUCTION PROCESS**

41. If we take a critical look at the conceptual framework of data warehousing, we see that we are dealing with a simple method of compiling statistics. If we can adapt the technology, it should be possible for us to put it to good use.

### **III.1 Checks and corrections**

42. If we bear in mind that statisticians have a considerable need for insights into the masses of data with which they are confronted, it becomes clear that a data warehouse can be a powerful aid. After all, as long as data are stored in flat

files, we need the help of a programme and ask the right question in order to get an answer, and that is all we get. The use of a data warehouse allows us to check the data visually, as it were. Not only can practically any aggregate be produced within seconds, one cannot avoid noticing anomalies in the subtotals, since they appear on screen.

43. We have found that statisticians want to load data at ever earlier stages of correction so as to gain a firmer grip on the correction process. This would not be the first time that checking and correction software "corrected" genuine phenomena because they were formerly implausible. A likely use is the repeated loading of a data mart in the checking and correction cycle, so as to steer the cycle. A first step in this direction was taken with the current population statistics project, by working with a provisional load before all the processing operations were completed and all the secondary data were known. The risk of publishing incorrect figures is very small, because the data warehouse manager ensures that users know what they are doing by giving the cubes names and making the material available to selected users.

### III.2 Analysis

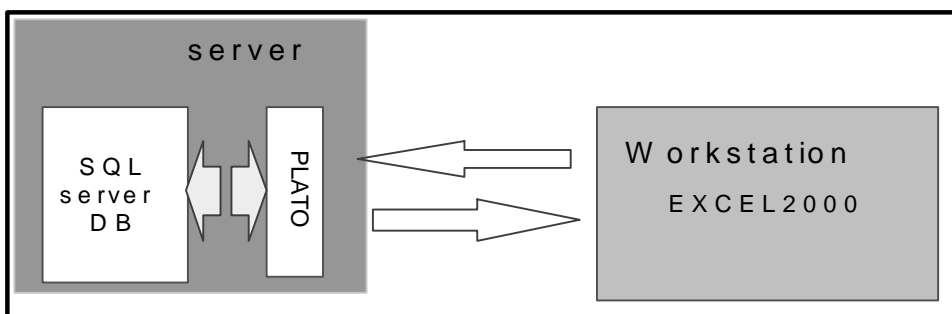
44. In the statistical analysis phase, a data warehouse is important not only as a replacement for *ad hoc* query systems, but also as a means of obtaining insights into the cleaned population. For further analysis, it is important to be able to distinguish data subsets. Data warehouses are not the most suitable tools for model-based estimates, special tables with SPSS, etc. We do expect data warehouses to make checking, correction and analysis activities increasingly interwoven.

### III.3 Output

45. Data warehouses make very suitable output media, but it is too early to use them or access them outside the CBS: security procedures are not yet in place, particularly procedures protecting against repeated consultation and recombination of data. The CBS is using the StatLine program for this activity. StatLine is constructed on data warehousing principles and will serve for a few more years to come.

## IV. TECHNICAL IMPLEMENTATION AT THE CBS

46. We decided to use Microsoft software for the first data mart. The CBS had recently decided that Microsoft software would be the standard. Microsoft has taken a conscious decision to make data warehousing available for mass use, in the form of SQLserver 7. Not only does SQLserver contain the Plato cube engine as standard, but tools such as EXCEL2000 dovetail neatly with the back-end software.



The diagram shows how the server components interact and how the link to the client application (EXCEL2000) is created. The link between the work station and the server goes via OLE-DB.

#### **IV.1 The data warehouse consists of data marts**

47. Given the CBS' decision to decentralise and downsize much of the processing work, it was decided not to put the CBS data warehouse on a large computer. On the contrary, we started from the premise that a data warehouse is a logical unit consisting of numerous data marts and lends itself to distribution among a large number of machines. It was decided, for practical reasons, that a data mart would be indivisible between computers. This has a number of advantages.

#### **IV.2 Star configurations in a relational SQLserver database**

48. Although the Microsoft DSS engine is suited for making cubes from an OLTP database with a normalised model, it was decided at the outset not to take that avenue. There were two reasons for this decision. Firstly, articles were beginning to appear in the specialist press, saying that perhaps this wasn't such a good idea after all, because it would be extremely difficult to store historical events etc. in this way. Secondly, because our aim (in addition to consulting cubes using interactive tools) was to extract data subsets from the data warehouse. We therefore opted to create genuine star schemas. These star schemas were then stored in the relational database with an SQLserver 7 kernel.

#### **IV.3 Aggregates and cubes in PLATO**

49. The Plato engine is complementary to the SQLserver software. It can run either on the database server itself, or on its own server. The software also operates with databases produced by manufacturers other than Microsoft. Its official name is Microsoft DSS services.

50. If we really want to do things by the book, the aggregates should be made in the database and maintained during loading. For our purposes, however, it was sufficient that we could use the DSS engine to compile aggregates as a means of automatically creating more space or improving efficiency.

51. Constructing the various cubes is the task of the data warehouse manager (decentral) and is fairly straightforward. With a graphic tool, it is easy to indicate which fields from which dimensions and which numerical data are to be made into a cube. The relatively simple tasks of definition and processing can then begin.

52. By opting for a DSS server (Plato, pivot table services, etc.: the beast has many different names) as a means of providing the interactive end user with data, we also opted to allow access via cubes. In this context, cubes are thematic areas within a data mart (star configuration). What happens is that the data warehouse manager chooses a number of dimensions from among all the dimensions in the mart. He then selects a numerical item from the fact table and has it pre-processed to form a cube. This ensures a better performance than is achievable via direct queries of the star in the relational database.

#### **IV.4 End user tools**

##### **IV.4.1 EXCEL2000**

53. For statistical applications, spreadsheets are an excellent aid for browsing through a data warehouse via a DSS engine. EXCEL2000 can contact the Plato OLAP provider (Microsoft DSS services) via OLE-DB, and so make cubes available to the user relatively easily. Spreadsheets dovetail neatly with statisticians' skills and have the immediate effect of raising productivity.

##### **IV.4.2 Our extraction tool**

54. Yes, we had to get our hands dirty and build our own tool. It should be possible to extract not only *ad hoc* aggregates but also data subsets from a data



warehouse in a statistical environment. The Microsoft pivot table services are not the most appropriate. That is why we developed a simple programme for this project which can write selections to a file and which makes it easy for a query to be formulated without the user having to understand SQL. The user can take a look at the SQL statement, however, and use it as a kind of semi-manufactured product.

## **V. THE ANNUAL STRUCTURAL SURVEY OF NETHERLANDS MUNICIPALITIES** (Annual enumeration of the whole population)

55. The Department of Population Statistics carries out an annual census based on the population data available to the municipalities. Hitherto, the census consisted of a large number of large sequential files for which special software was used for calculations. Not only is the management of these files a difficult task, because of their size and number, but consulting them also had to be done on a planned basis. Through-put times (including de-archiving) of up to 60 hours were not exceptional. Two years ago, they decided, in consultation with the Automation Division, to undertake a pilot study of whether a data warehouse could solve the problems. We began a joint project during which it gradually became clear how a data warehouse would have to function in a statistical environment, and that the model could not be constructed entirely along the lines of the familiar supermarket model.

### **V.1 Structure of the model**

56. Initially, it looked as if just one data mart would be sufficient to meet our information needs, but it soon became clear that the ADDRESSPERSON (ADRESPERSON) data mart could not provide information on the relationship between persons living at the same address. It was therefore decided to create a second data mart, called ADDRESSFAMILYRELATIONSHIP (ADRESGEZINRELATIE), which we now refer to as simply FAMILY (GEZIN). The overall structure of the two data marts is as follows.

#### **V.1.1 PERSON data mart**

roles of the particular dimension (every role is a key in the facttable)	
Dimensions	
Date	10 immigration, visum, change mar.stat., marriage, divorce, firstmarriage, etc.
Duration (years)	8 age, age parents, duration mar. status, etc.
Gender and marital status	1
Address	1
Postal area	1
City	1
Address type	1
Visum	1
Country	8 birth, mother, father, etc.
Person	1
Family profile	1

57. The PERSON data mart is dedicated to the analysis of a person's residence at a particular address. All the data about this person, insofar as they are available from the Population Register, are stored in the data mart.

**V.1.2 The ADDRESSRELATIONSHIP data mart**

roles of the particular dimension (every role is a key in the facttable)

**Dimensions**

Date	8 birth, immigration, visum, change mar.stat., marriage, divorce, etc.
Duration (years)	7 age, age parents, duration mar. status, etc.
Gender and marital status	2 reference person and partner
Address	1
Postal area	1
City	1
Address type	1
Country	6 birth, mother, father, etc.
Person	4
Family profile	4

58. In this data mart, the relationship between two persons is the most important fact. The keys of both persons and those of their youngest and oldest children are also included in the fact table. Numerous data were also taken from PERSON so as to avoid having to constantly combine two marts.

**V.2 Physical implementation**

59. As already mentioned, we decided on Microsoft products for the database, DSSservice and query tool. The main reason was that the software was already available to the CBS. We do not believe in departing from existing standards without a very good reason.

**V.2.1 Custom software for loading data**

60. Loading the data warehouse was more problematic. Firstly, no suitable Microsoft tool was available. Secondly, the search for commercially available loading software was frustrated because the many tools that claimed to support data marts did nothing of the sort. As our timetable was under threat, we decided to write some custom software using Microsoft's Visual Basic. After the pilot study, the choice was easy. The loading software was ready and it seemed reasonable to develop this software for production purposes.

**V.3 Management**

61. The entire project has been transferred to its owner, the Population Division. This means that, although we (IT and Applications Development) continue to do research and provide support, it is their data warehouse and they are responsible for it.

**V.3.1 Contents management**

62. A locally based statistician is responsible for managing the contents. Although he has acquired the knowledge necessary for making cubes and defining and adding users and roles, he remains first and foremost a statistician. This ensures that the data warehouse does not degenerate into a "mere" technical *tour de force*, but remains a working statistical tool.

**V.3.2 Technical management**

63. Management of the software and data model has been transferred to the local computer experts, who are answerable to the contents manager.

## **VI. CONCLUSION**

64. Data marts can offer statistical offices a solution for the production of statistics and may replace existing methods where more and more data come to comprise an integrated data warehouse. It has been firmly established that significant gains can be made in terms of checking, correction and analysis.

## **References**

The Data Warehouse Toolkit, Ralph Kimball, Wiley; ISBN 0-471-15337-0  
SQLserver7 Data Warehousing, Michael Corey et al, Osborne; ISBN 0-07-211921-7  
The Data Warehouse Lifecycle Toolkit, Ralph Kimball, Wiley; ISBN 0-471-25547-5  
Building the Data Warehouse, W.H. Inmon, Wiley; ISBN 0-471-14161-5