METHOD AND DEVICE FOR SEARCHING AND FINDING DATA FROM A FILE

A database or data bank is a collection of related data for determined applications and is generally available to a large number of users. In practice use is mainly made of so-called relational databases in which the data is stored in the form of tables or relations. In designing the structure of a relational database care is taken that in different tables links (relations) are made between the tables by means of the data itself, in the form of characteristics thereof. In practice use is often further made of the fourth generation language SQL (Structured Query Language) which in addition to retrieving data from the database, via so-called queries, also provides the option of making use of a (different) programming language, for instance in order to define, change and consult the data files.

In the currently usual practice for setting up and maintaining a data file, and for the command to be given by the user to obtain particular data from the file, an extensive analysis of the requirements of a large number of users is usually made by a specialist. Not all users will be satisfied with the chosen and implemented structure and the possible commands related thereto. Changed insights may after a certain period of time also entail new requirements for the users. Modification of a structure once it has been chosen and the commands it is possible to give is laborious and time-consuming, i.e. expensive, if use is made of one or more external specialists.

Known from the article by L.M. Burns et al: "A Graphical Entity-Relationship Database Browser", 1988 IEEE, is an entity-relationship interface wherein the model of the database is directly derived from the

entity-relationship diagram in the database, whereby it is not possible to display data in a form desired by the user in an automatic manner.

In the article by Peter Kraft: "Correspondence between user interfaces and data models - an entity relation approach", Amsterdam, 1990, is described a design of a user interface wherein the correspondence between the displayed data and stored data is given. Although this involves a certain projection between database and a user model, the implemented structure according to the present invention is not described here.

The present invention has for its object to obviate one or more of the above stated drawbacks.

The present invention provides a method for searching and finding data from one or more files (databases), wherein a model of the database(s) desired by a user is made via an interface with the user, wherein a link is made between that model and the structure (preferably an entity relation diagram) of this (these) database(s), wherein on the basis of the lay-out of a report desired by the user, in which data from the files is presented in the desired form, commands are generated to the data files, and wherein this report lay-out with data is recorded/made visible on a data carrier, a monitor and/or paper.

In the method according to the present invention, on the basis of a model made by the user of his problem area (reality), an entity relation diagram is made of that model in automated manner, wherein information concerning the structure of the database is defined. On the basis thereof as well as on the basis of the queries the user wishes to make of the database, commands (in particular SQL commands) are generated. The queries to be made by the user are set in the form of a report lay-out. This report lay-out also determines the manner in which the obtained data with the generated commands is recorded/made visible on a data carrier, on a monitor and/or on paper, while the user can amend this

lay-out as required at any desired moment, which prevents the commands for selecting being limited to commands determined by others, for instance external specialists.

The invention further provides a system for searching and finding data, comprising:

- one or more data files, databases;
- a schematic description of this (these) database(s);
- a model of this (these) database(s) as desired by the user;
- an engine which is connected to the data files and the user model thereof; and
- a user interface which is connected to the model and the engine for presenting a report to the user in the lay-out desired by him/her.

Further advantages, features and details of the present invention will be elucidated on the basis of the following description of a preferred embodiment thereof with reference to the annexed drawing, in which:

fig. 1 shows a simplified example of a model of limited size configured by one or more users;

fig. 2 shows a diagram elucidating the preferred embodiment of the method according to the present invention;

fig. 3 shows a diagram to elucidate the storage of a chosen model in a preferred embodiment of the device according to the present invention;

fig. 4 shows an example of a report lay-out obtained using the method with the preferred embodiment of the method according to the present invention;

fig. 5 shows a relation diagram of a simplified example to explain the method and system according to the present invention;

fig. 6 shows a diagram for further explanation of a preferred embodiment of the method and system according to the present invention;

fig. 7 shows a diagram of the (mapping) of the user model on the database model for indicators; and

fig. 8 shows a diagram explaining a further feature of the method and system according to the present invention.

In the diagram of the simplified model according to fig. 1 are shown the data files of a financial services company. Shown vertically herein are different groups, 1, 2, 3, 4 and 5 of data (indicators) which a user wishes to know from the database. Block 1 relates for instance to the number of productive hours, i.e. the data (indicators) of the number of hours worked, overtime hours and the like. In block 2 are the hours, overtime hours and the like worked by management. In the horizontal blocks (6-18) are shown the categories with which a user wishes to study the indicators (data). Block 6 relates for instance to all orders which have been carried out and block 11 relates for instance to all the offices of the company.

In the matrix-like diagram of fig. 1 circles designate which indicators are available for which categories in the database and can be studied by the user. A cross at the intersection of lines of indicators and categories indicates that while the user would like to set the relation in the model, no data is available for this purpose in the database. It is possible to define functions for this purpose, so that the relation between an indicator and such a category can nevertheless be studied. If such a function is selected, the data obtained should only be considered as approximations, since the data is not obtained directly from the file.

According to the present invention a user can change the model as desired. In a system 50 (fig. 2) a model 54 is included as imaginary route map of the data bank. In the diagram of fig. 2 is shown the model 54 derived from a database 51 in addition to the three sub-engines, i.e. model engine 53, query engine 52 and projection engine 55 which together can be viewed as one common engine. Via an interface 57 a user can define a desired report lay-out which is implemented in block 52

which is connected to projection engine 55 and model engine 53. Block 58 designates for instance a monitor or other device on which the report is made visible or printed in the desired lay-out. The query engine 52 serves to retrieve indicators (profit) from the database on the basis of the model and a determined request for information, for instance how much is the profit for the Amersfoort office in the year 1992. For this purpose the query engine generates SQL commands for the database 51. In order to make these commands use is made of the model 54 which is configured by the model engine. In contrast to conventional systems it is not necessary according to the present invention to preset commands since these commands are generated in a query engine on the basis of the model and on the basis of queries to be made by the user.

The projection engine 55 incorporates the definitions for a report lay-out configured by the user and on the basis thereof generates reports on monitors, printers and other data carriers. The projection engine communicates with the query engine to cause the correct commands to be generated by this query engine, which commands are therefore, in contrast to conventional systems, dependent on the wishes of the user.

Fig. 3 shows the manner in which the model is stored in models. Block 20 generally designates a category (blocks 6-18 of fig. 1). The blocks 6-18 are included in a table. The blocks 1-5 of fig. 1 are included in the same manner in a table (block 21).

Block 22 (named allocation) shows which indicators are not present for which categories in the database. This in fact means that a connection as designated with a cross in fig. 1 is included in block 22. It is however possible by means of determined functions to derive data from other data included in the database.

The user model of fig. 1 can be stored and defined in the blocks 20, 21 and 22. Blocks 23, 24, 25

and 26 of fig. 3 represent tables (or entities) called respectively property type, aggregation, aggregation differential level and differential level.

The block property type 23 which is connected via line 27 representing "appears as" to a block 21 (indicator) describes for an indicator in which column this latter can be found in a table in the database, wherein a mapping takes place of the user model to the database model and subsequently to the database.

The block 26 (differential level) is connected via line 28 representing "is available for" to block 20 (category) and via line 29 representing "appears in aggregation" to block 25 (aggregation level). Each category also occurs in the block 26 (differential level), in which is included in which tables in the database the persons, commands and the like are found, wherein a mapping takes place from the user to the database model and to the database.

Block 24 which is connected via line 30 representing "has property type" to block 23 and which represents "aggregation" records the names of the blocks 1-5 of fig. 1 and in which tables in the database these groups of indicators can be found. Aggregation differential level (block 25) makes a connection between block 24 (aggregation) and block 26 (differential level) wherein the circles in fig. 1 are in fact set. Such a circle implies that all combinations of categories situated vertically beneath it and horizontally the name of the block (aggregation) are included in the block aggregation differential level in addition to the associated column name in the database. This is performed in this way for each circle in fig. 1.

The lines 31 and 32 both represent a "resorts above" relation respectively at block 20 (category) and block 26 (differential level) with which the connections between categories (for instance customer order) are stored.

The stated modulation also offers the possibility of grouping this available data into one concept for the user, such as costs, in a so-called production database. The concept of costs in the user model is shown on the blocks of the database structure in fig. 3. Different data in a database can be made into a single item, at least from the point of view of the user. The reverse situation also offers the possibility of obtaining data such as costs in different tables in the database. Depending on the queries made by the user, the data is then retrieved from this table which produces the fastest result.

In addition to indicators in the model which may or may not be available in the database, it is also possible according to the present invention to define derived data, for instance the average number of productive hours, i.e. the number of productive hours divided by the production. In contrast to conventional systems, such a derived quantity is not then stored in the database but is derived when it is requested, which, in contrast to conventional systems, makes the management and consistency of such data superfluous.

The definitions for the report lay-out are abstract definitions in the form of queries which a user wants to make to the database. In the example of fig. 4 this is elucidated with reference to two relatively simple queries: what is the production of clients 1 to 5 and what is the number of productive hours for clients 1 to 5. If the user wishes to alter this report because he/she wishes to place different years instead of production and the number of productive hours and to place the number of productive hours at top left, the correct commands for the desired lay-out are generated via the projection engine together with the model engine and the query engine and the requested data appears for instance on the screen of the user.

In starting up the system according to the present invention, for instance to obtain the data

required in fig. 4 on a screen, the model engine is activated first of all. The model engine internally makes an entity relation diagram of the data file on the basis of the model of the information and the information concerning the structure of the database. The blocks of the diagram of fig. 1 form this entity relation diagram. These blocks represent the model for the user and the relations thereof in the database.

Once a desired report lay-out has been read by the projection engine, as designated in fig. 4, wherein the user wishes to obtain the number of productive hours and the production for client 1 to 5, the desired lay-out for the report, and therefore for the monitor, is configured by the projection engine and the projection engine is connected to the model engine in order to obtain the desired texts in the report (and on the screen), while the requested data is retrieved from the file by the query engine.

For this purpose an SQL command is generated by the query engine on the basis of the information in the model. The commands are thus generated as required by the user, whereby the drawbacks of the conventional systems do not occur. The users can define reports in accordance with their own ideas, wherein the commands to the database are automatically generated by the query engine within the possibilities of the model. It is also possible to change the model (i.e. the route map) of the data file without any great effort, in contrast to the effort required in conventional systems for data files.

A simplified example of a model of the present invention is shown in fig. 5. Further added in the annex are tables 1-12 elucidating the data from the database and the relations. In line 1 of table 1 a turnover of 100 is realized against costs of 50 in month 1 of 1995 at the Amsterdam office (AMS) for a client of the Alkmaar office (ALK) as an order for the work unit Noord-Holland (NH) for the client in work unit Noord-Holland (NH). The lines of table 1 can be read in like manner, wherein GRO

represents the city of Groningen, GR the province of Groningen, APE the town of Apeldoorn, GE the work unit Gelderland and ARN the Arnhem office. On the basis of the example of table 1 are defined the tables 6-12 which are necessary for the explanation of the example according to the present invention.

Data from more than one table can be unified in one or more databases to a concept which can be utilized by the user. The concept turnover for instance occurs twice in the database in the tables "mr client month costs & turnover" and in "mr client costs and turnover". The present invention enables the user to define the concept of turnover while the concept can be linked to both data relating to turnover in the database (see fig. 6). If a user requests the turnover, the correct data is retrieved from the database, since the user model is projected on the database model.

This takes place using the meta-information of the blocks of fig. 3, such as:

- the concepts employed by the user (user model)
- the database.

In the simplified example of fig. 5, fig. 3 is implemented in practical manner and, using this implementation, a program code is generated which retrieves the correct data from the database at the request of the user. If a user requests the turnover for a particular month, the system according to the invention retrieves the data from "mr month costs & turnover". If the user desires the costs per client the data is retrieved from "mr client costs & turnover". The query for the second table will be faster than for the first table since the second table is smaller.

In contrast to conventional systems, the location from which the data originates is determined by the system itself by means of the information about the database, the user model and the query made by the user.

The information the invention requires to generate the code is stored in the model tables as follows. In block 21 (ir) of the database the concepts of cost and turnover are recorded as a user sees them in the user model. In the related table 23 (pt) a registration is made for each indicator as to where in the database(s) data for this concept can be found. Two tables with data are available in the database for the indicators cost and turnover. Where in table 21 there are two entries for turnover and costs, there are as a result four, i.e. two times two, entries in table 24. The indicators are hereby described at two levels: users and database. The connection between block 21 (the concept for the user) and block 23 (the concept for the database) is set by means of connection 27. The method and the system according to the present invention is therefore capable of allowing a user to define a limited set of concepts (model) and of allowing these to be used, wherein the optimal translation to the same or different concepts is always made to the database. This is elucidated in fig. 7.

In addition to unifying multiple database data into the user concepts, the reverse is also possible. For the categories (concepts with which the user wishes to make a selection, for instance country is the Netherlands) there is the possibility of employing more user concepts for a database concept. In the example the concept of client, office and ordering office is used in the user model. A table with all offices is available In the database. The user wishes however to have two concepts of office in his model, since the one concept comprises the offices for which the orders are carried out (ordering offices) and the other concept comprises the offices to which the clients belong (client offices). While the method and system according to the invention offers the option of having client office refer to the same table, it also offers the user the option of performing different selections for both concepts. This in contrast to conventional systems. The query can for

instance be made as to costs for client office Amsterdam and ordering office Apeldoorn.

This possibility is set in fig. 3 by means of the blocks 26 and 20 and the connection 28. Block 26 (table 11) sets the categories as the user wishes to see them. The user knows the concepts of ordering office and client office which are recorded in block 26. Both are offices of his company. In the database only one table office is available. In block 20 (table 10) only one concept of office is set as available in the database. The connection 28 determines that ordering office and client office are equal to each other at the level of the database. If the user makes a query, such as turnover for client office, the method and system is capable of translating this query by means of connection 28 to the correct concepts in the database and of retrieving the desired information.

Queries formulated by a user give more extensive results than is usually the case in conventional systems. According to the method and the system according to the present invention a so-called non-available is generated for instance for months for which no turnover (or profit) is available in table 1. The final results are produced on the basis of the months in the table month and not on the basis of the months for which the turnover (or profit) is available in "mr month costs & turnover".

This is realized in that data which is present in the database (January 530; February 620) is retrieved and in that for the missing months which are present in the model of the user (month comprises all months) non-available symbols are added. The user does not have to adapt his query herein.

According to a further feature of the method and system according to the present invention a concept such as profit can also be described in terms of turnover and costs. If the user defines that profit is turnover minus costs, the user can make any query relating to the

profit which he can also make for turnover. The method and the system according to the present invention then determines on the basis of turnover minus costs that the turnover as well as the costs are required in this situation and calculates the results by means of this formula.

The present invention is not limited to the above described embodiment; the requested rights are determined by the following claims.

*****

THIS SHEET HAS BEEN REPLACED

This Sheet Has Been Replaced

* C L M S *

CLAIMS

1. Method for searching and finding data from one or more files (databases), wherein a model of the database(s) desired by a user is made via an interface with the user, wherein a link is made between that model and the structure (preferably an entity relation diagram) of this (these) database(s), wherein on the basis of the lay-out of a report desired by the user, in which data from the files is presented in the desired form, commands are generated to the data files, and wherein this report lay-out with data is recorded/made visible on a data carrier, a monitor and/or paper.

2. System for searching and finding data, comprising:

- one or more data files, databases;
- a schematic description of this (these) database(s);
- a model of this (these) database(s) as desired by the user;
- an engine which is connected to the data files and the user model thereof; and
- a user interface which is connected to the model and the engine for presenting a report to the user in the lay-out desired by him/her.

3. System as claimed in claim 2, wherein the engine comprises a model engine for assembling an entity relation diagram of the database(s) on the basis of the model of this (these) database(s) desired by the user and the schematic description of this (these) database(s).

4. System as claimed in claim 2 or 3, wherein the engine comprises an enquiry engine for generating the desired commands to the database(s).

5. System as claimed in claim 2, 3 or 4, wherein the engine comprises a projection engine for making the data visible in the desired report lay-out.