A METHOD AND APPARATUS FOR CONTROLLING ACCESS TO A DATABASE

This invention relates to a method and apparatus for controlling access to a database.

In database systems it is usual for a number of users to be able to interact with the system and to utilise the database. Such systems are thus called multi-user systems. A problem occurs in such systems, where the database contains information or data which is in some way sensitive, that is to say, it should only be available for a certain user or a member of a certain class of user. Where this is so, a security policy is implemented to restrict the data available to the class of user.

An example of a database management system which supports multiple users is ORACLE (registered trademark of Oracle Corporation). ORACLE is a relational database management system. In a relational database, only one type of data structure exists and this is the table which is a two dimensional structure of rows and columns of data. A query language called Structured Query Language (SQL) may be used to access data in a database in a non-procedural way.

There are a number of ways in which a security policy has been implemented on database management systems. For example, in one method each class of user is provided with its own copy of that part of the data held in the central database for which it is appropriate for that group to have access to. This method has been called the replication method because it results in the data being replicated since at least some of the data will exist in more than one copies. Clearly, such a method is very inefficient in terms of memory usage. Further, if one copy of the data is changed in some way by, for example, a user of a particular group updating a value, then a number of other copies of that data held by

This Sheet Has Been Replaced



other groups will have to be updated. This will be time consuming and the way in which the system is administered will have to be very precise to ensure that data is maintained in a consistent state if, for example, the system crashes.

According to the invention there is provided a method for controlling access by a user to data in a database comprising configuring at least part of the database such that at least some of the data of the configured database is associated with a security tag, configuring a storage structure of user identifiers and associated user tags, configuring a storage structure of user tags and associated security tags and mapping a user identifier to at least a subset of the data by determining from the storage structure of user tags and associated security tags a security tag or tags appropriate to the user tag of the user identifier and allowing access to the data from the configured database associated with the security tag or tags.

By providing a storage structure of user tags and security tags it is possible to change the security policy by modification of the values in the storage structure alone without any need to reconfigure the database or to change the user tags associated with the user name. By security tag it is meant a data entry which indicates that the associated data has particular attributes such as security classification. It may be a number or a character or other data entry. By storage structure it is meant some grouping of data, for example, in the form of trees, records, sets, tables or other storage structures.

According to a second aspect of the invention there is provided a method for controlling access to a database by a user comprising creating a database of data and associated security tags, creating a first storage structure of identifiers and identifier tags, creating a second storage

This Sheet Has Been Replaced



structure of identifier tags and associated security tags and creating a view onto the database appropriate to at least one of the identifiers by determining from the first storage structure an associated identifier tag and determining from the second storage structure at least one security tag appropriate to that determined identifier tag and selecting from the database, data associated with that security tag or tags.

By view it is meant a subset of the data which may be manipulated by the user as required. In SQL a view may be created for a user which allows the user to perform operations in the data. By providing a SQL CHECK option when defining the view, the view may be made secure that is to say alterations made by a user to the data may not be written back to the database without the CHECK option being satisfied. This is particularly useful where a user is not to have write access to the data in the database but is permitted to read the data and to manipulate it for his own use only.

A system operating in accordance with the invention has to have one copy of the data only which is shared by the users and thus avoids at least some of the problems associated with the known replication methods such as storage memory inefficiency or the need for additional data management and processing time.

It should be noted that not all of the database need be configured for the sake of security. Some parts may be public and thus open to all users.

The storage structure of user identifiers and user tags and the storage structure of user tags and security tags may be configured such that at least some of the users cannot gain access to all parts of the storage structures. In this way, the precluded user is prevented from determining what

This Sheet Has Been Replaced



security rating he has and the system can be thought of as "transparent" that is, the user is not aware of the security policy and has no evidence for believing that access is being denied to some of the central database.

According to a further aspect of the invention there is provided apparatus for controlling access by a user to a database comprising means to configure: at least part of the database such that at least some of the data of the configured database is associated with a security tag; a storage structure of user identifiers and associated user tags; a storage structure of user tags and associated security tags; and means to map a user identifier to at least a subset of the data by determining from the storage structure of user tags and associated security tags a security tag or tags appropriate to the user tag of the user identifier and allowing the user access to the data from the configured database associated with the security tag or tags.

The apparatus or method could be used for the access of employee records or other information such as information on a telecommunications network. Such information could be used to configure the telecommunications network by a network manager. The apparatus could also be configured to combine both a network managing function and a database security function.

A specific embodiment of the invention will now be described, by way of example only, with reference to the drawing in which:

Figure 1 shows, in schematic block diagram form, hardware of a database system operating in accordance with an embodiment of the invention;

Figure 2 shows a set of database accounts supported by the database system;

This Sheet Has Been Replaced



Figure 3 shows a software view of the database system;

Figures 4 to 6 show, in schematic form, tables used in the database system;

Figure 7 is an explanatory diagram of operation of the database system;

Figure 8 is an explanatory diagram of a mapping operation carried out by the database system;

Figure 9 is a further explanatory diagram showing steps in the mapping operation; and

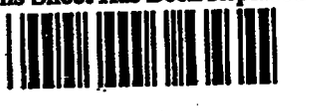
Figure 10 shows in schematic block diagram form the database system being used in a network management application.

With reference to figure 1, a database system 1 comprises a number of elements including a mainframe computer 2 of well known type such as a DEC Micro Vax connected to a number of user terminals 3, 4 and 5 each of which comprises a microcomputer of well known type such as an IBM PC. The connection is made by means of coaxial cable 6 of well known type and the communication between the elements of the system 1 is achieved by a well known communications protocol such as Transmission Control Protocol/Internet Protocol (TCP/IP).

The user terminals 3,4 and 5 are nominally identical. Each has a microprocessor 3a, 4a and 5a; memory 3b, 4b and 5b; an input/output device 3c, 4c and 5c; a buffer 3d, 4d and 5d; a visual display unit (VDU) 3e, 4e and 5e, and a keyboard 3f, 4f and 5f.

As will be readily appreciated, the memory 3b, 4b and 5b can be in the form of random access memory, read only memory or combinations of the both. The memory may be of solid state

This Sheet Has Been Replaced



form as semiconductor "chips" or disc (optical or magnetic) or a combination of these forms.

Whatever the form, the memory comprises a number of memory locations. These locations will contain instructions for governing the operation of the microprocessor 3a, 4a and 5a with which the particular memory is associated. The microprocessor 3a, 4a and 5a accesses the memory to obtain the instructions. A program for governing the operation of the terminal is held in the memory as a set of instructions located at a number of the memory locations. The instructions will be in the form of a hexadecimal number.

The memory is linked to the microprocessor by a databus in a manner well know. The databus also links the microprocessor to the other elements of the terminal. The input/output device 3c, 4c and 5c acts as an interface between the terminal and the other computers in the system.

The keyboard and VDU of each terminal interacts with the terminal's microprocessor via the buffer in a well known manner. Collectively, they provide an interface between the system and a user wishing to interact with the system.

The main frame computer 2 has a processor 2a, memory 2b, an input/output device 2c, a peripheral buffer 2d and associated VDU 2e and keyboard 2f. Thus, it will be seen that the mainframe computer 2 is of the same form as the user terminals 3, 4 and 5. The major difference is that the storage capacity of the memory 2b is far greater than that of the memory of the user terminals. An administrator of the database system can access the system by utilising the keyboard 2f and VDU 2e.

The terminals 3, 4 and 5 and the mainframe computer 2 are interconnected by the coaxial cable 6 which extends between the input/output devices 3c, 4c and 5c of the terminals and

This Sheet Has Been Replaced

the input/output device 2c of the mainframe. As earlier mentioned a protocol called TCP/IP is used for communication between elements of the system 1.

The memory 2b contains a database of information. This information can be accessed by the users from their terminals. However the extent to which each user is allowed to access the information may vary between users. The system administrator will have access to all the database.

The system can be considered as providing a set of database accounts, as depicted in figure 2. The administrator will have an administrator account 21 and the users will have user accounts 22, 23 and 24. In this case the users are named Brown, Smith and Jones and the accounts are labelled accordingly.

The memory 2b holds, as well as the database, a program for controlling the processor 2a, in particular, the way in which the database is accessed by each of the users. Thus, under software control the processor 2a acts as a database engine.

As is shown in figure 3, Smith, Brown and Jones can input requests into the database engine 31 and the engine will process the request accessing the database 32 as required. The database engine 31 then outputs a response to the querying user. These requests will be carried by transmission over the coaxial cable 6.

The database 32 is subdivided into three parts, each part being an SQL table. The first subdivision is a table 33 called "EMPLOY". The second subdivision is a table 34 called "SECURITY" and the third subdivision is a table 35 called "USER". The database system 1 utilises a programming language called ORACLE SQL (registered trade mark of the Oracle Corporation) to set up and utilise the tables. The

This Sheet Has Been Replaced

way in which the tables are initially created will be described later.

The "EMPLOY" table 33 comprises information about employees in a company. It comprises a number of datafields as shown in figure 4.

The datafields include a datafield 33a called "EMP NO" which includes the employee reference numbers for the employees of a particular company.

There is a datafield 33b called "NAME" which includes the names of employees of the company held as a string of thirty characters (CHAR) or less.

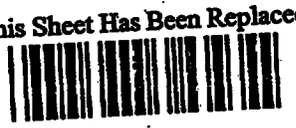
The next datafield is a datafield 33c which is called "POSITION". The "POSITION" datafield 33c contains information about the position of a particular employee in the company, for example, the employee may be a manager, clerk or secretary. This information is also stored as a string of ten characters or less.

The next datafield is datafield 33d and this is called "SAL". This contains information about each employees salary expressed numerically in seven digits.

Datafield 33e is called "DEPT" and this includes the name of the department within which the employees work. This information is held as a string of ten characters.

Datafield 33f is called "ROW_TAG". This datafield contains a one character string indicative of a security status of the row of information to which it belongs. This field is of particular significance to the way in which access is allowed to particular rows of the "EMPLOY" table 33.

This Sheet Has Been Replaced



The "SECURITY" table 34 comprises two datafields, a first datafield 34a called "ROW_TAG" and a second datafield 34b called "USER_TAG", as shown in figure 5.

The "ROW_TAG" datafield 34a will include the same characters as held in the "ROW_TAG" datafield 33f of the "EMPLOY" table. This will permit a mapping operation to be explained later in which rows of the "EMPLOY" table are selected by selecting these rows having a "ROW_TAG" the same as the "ROW_TAG" of the "SECURITY" table 34.

The "USER_TAG" datafield 34b holds one character data. The function of this field is to enable the mapping operation mentioned above and this will be more fully explained later.

The "SECURITY" table 34 is thus named because the system security policy is embodied in the table. The "ROW_TAG" and "USER_TAG" of this table are termed security tags since the security policy is governed by these tags. The security policy may be conveniently modified by modifying this table. This aspect of the system will be more fully explained later.

The "USER" table 35 is shown in figure 6 and comprises a "USER_TAG" field 35a and a "USER_NAME" field 35b.

The "USER_TAG" field 35a holds one character data which will include the same characters as those held in the "USER_TAG" field 34b of the "SECURITY" table 34. This will permit the aforementioned mapping operation to be performed as will be described later.

The "USER_NAME" field 35b holds the names of users of the system in the form of character strings.

A flow chart of the system operation is shown in figure 7. A first step in the operation is initialisation, as represented by box 70. In this step the terminals 3, 4 and

This Sheet Has Been Replaced



5 are switched on, as is the main computer 2, and readied for use.

A second step, as represented by box 71, is to create the tables in memory 2b. This is done by the database engine 31 using a SQL command CREATE TABLE in the following way.

For the "EMPLOY" table 33 the following command statement is implemented by the database engine 31:-

```
CREATE TABLE EMPLOY
(
EMP NUMBER (4)
NAME   CHAR (30),
POSITION CHAR (10),
SAL    NUMBER (7),
DEPT   CHAR (10),
ROW_TAG CHAR (1)
);
```

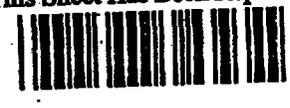
The "SECURITY" table 34 is configured by use of the SQL CREATE TABLE command in the following way.

```
CREATE TABLE SECURITY
(ROW_TAG CHAR (1),
USER_TAG CHAR (1)
);
```

The "USER" table 35 is configured, by the SQL CREATE COMMAND in the following way.

```
CREATE TABLE USER
(
USER_TAG CHAR (1),
USER_NAME CHAR (10)
);
```

This Sheet Has Been Replaced



In a next step 72 the tables are populated with data. This is done by the network administrator utilising the database engine 31 and the SQL INSERT command in the following way.

For example to insert a row of data about an employee called Stuart Fitchett into the "EMPLOY" table the following command is issued by the administrator.

```
INSERT INTO EMPLOY
VALUES (10, 'Stuart Fitchett', 'CLERK', 1000, 'CS', 'U');
```

Thus, it is recorded that employee number 10 is called Fitchett, he is a clerk earning £1000 per month in the customer services department (abbreviated CS) and the security tag required to read this information is 'U'.

Data is entered into the "SECURITY" table 34 in a similar way. For example the first seven rows of data may be entered in the following manner.

```
INSERT INTO SECURITY
VALUES ('U', 'A');
INSERT INTO SECURITY
VALUES ('V,' 'A');
INSERT INTO SECURITY
VALUES ('U,' 'B');
INSERT INTO SECURITY
VALUES ('W' 'B');
INSERT INTO SECURITY
VALUES ('W' 'C');
```

The USER table is completed in a similar manner.

```
INSERT INTO USER
VALUES ('A', 'SMITH');
INSERT INTO USER
VALUES ('B', 'JONES');
```

This Sheet Has Been Replaced



```
INSERT INTO USER
VALUES ('C', 'BROWN');
```

The database engine 31 then awaits a request from one of the users for information from the database 32 as represented by box 73 of figure 7.

When a request is received it is processed, as represented by box 74. Upon completion of the processing the system returns to the await request step 73.

The completed tables are schematically shown in figure 8.

The process request step 74 will now be described in more detail with reference to figure 9. It is this processing step that utilises the above mentioned mapping operation that implements the security policy governing the system 1.

A first step is for the database engine 31 to identify the user making the request for access to information stored in the database 32, as represented by box 90 of figure 9.

A next step is for the database engine 31 to utilise the "USER_TABLE" 35 to obtain a "USER_TAG" appropriate for the identified user, as represented by box 91.

A further step, as represented by box 92, is for the database engine 31 to utilise the "SECURITY" table 34 to obtain a "ROW_TAG" appropriate for the "USER_TAG" identified in step 91.

A final step, as represented by box 93, is for the database engine 31 to return from the "EMPLOY" table 33 a row or rows of data where a "ROW_TAG" associated with the row matches the "ROW_TAG" identified in the previous step, step 92,

This Sheet Has Been Replaced



An example will now be used to illustrate the way in which the database engine 31 processes a request.

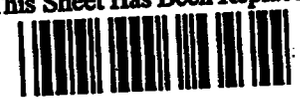
A user having a USER_NAME SMITH has been assigned a USER_TAG A and this may be mapped via the "SECURITY" table 34 to ROW_TAGS U and V. This enables SMITH to gain access to rows of the EMPLOY table 33 which have been assigned ROW_TAGS U or V. Thus SMITH is mapped onto rows ROW1, ROW2 and ROW3 of table EMPLOY and can view the data of those rows.

In more detail, suppose SMITH wishes to access all the information that he can from the "EMPLOY" table 33. To do this SMITH sends a SQL SELECT command SELECT * FROM EMPLOY; to the database engine 31.

The database engine 31 attributes an argument to this command of SMITH to identify the user and then performs the following mapping operation where USER_NAME is "SMITH". The mapping operation being provided in the programme governing the operation of the database engine 31.

```
CREATE VIEW SECURE_EMPLOY AS
SELECT EMP,
       NAME,
       POSITION,
       SAL,
       DEPT,
FROM EMPLOY
WHERE EMPLOY.ROW_TAG IN
(
       SELECT SECURITY.ROW_TAG
       FROM SECURITY
       WHERE SECURITY.USER_TAG IN
       (
       SELECT USER.USER_TAG
       FROM USER
       WHERE USER.USER_NAME = username
```

This Sheet Has Been Replaced



```
    )
)
```

This returns to the user SMITH the rows of table "EMPLOY" which have a ROW_TAG 'U' or 'V' that is to say rows ROW 1, ROW 2 and ROW 3 of the table 33. It should be noted that since the SELECT command did not include the "ROW_TAG" column of the "EMPLOY" table 33 this information is not returned to the user and the security policy governing the data cannot be determined by the user. Thus, the security policy is transparent to the user.

The system caters for access by BROWN and JONES in a similar way with the user name argument attributed by the database engine being BROWN or JONES as appropriate.

If a change in the security policy is to be implemented, for example, the class of user having a USER_TAG of 'A', formerly having access to rows in the EMPLOY table 33 having TAG 'U' or 'V', is to be restricted to only rows having TAG 'U', then it is only necessary for the network administrator to amend the SECURITY table 34 by deleting from the security tags assigned to user tag A. This could be achieved by using the SQL DELETE FROM table command thus:

```
DELETE FROM SECURITY
WHERE SECURITY.USER_TAG = 'A'
AND SECURITY.ROW_TAG = 'V';
```

Thus the second row of the "SECURITY" table 34 as shown in figure 8 is deleted, eliminating that mapping path from user SMITH to the second row of the "EMPLOY" table 6 having "TAG" 'V'.

To extend the amount of the EMPLOY table 33 accessible to a class of user having a USER_TAG 'B', for example, to enable rows having a ROW_TAG 'V' or ROW_TAG 'X' to be accessed, the SQL command INSERT INTO table is used thus:

This Sheet Has Been Replaced



```
INSERT INTO SECURITY
VALUES ('V', 'B');
INSERT INTO SECURITY
VALUES ('X', 'B');
```

This will provide the two further mapping paths from the user table to the EMPLOY table as shown in broken outline in figure 8.

In some database security systems it will be desirable to prevent data being updated in the database by some classes of users. One way in which this may be achieved is creating a view onto the EMPLOY table 33 which is secure in the sense of a check option being provided to prevent a user inserting an entry into a part of the database which he cannot subsequently delete information from. Insertion of data or updates of data are only allowed if a SQL "WHERE" statement is satisfied. For example:

```
CREATE VIEW SECURE EMPLOY_2 AS
SELECT*
FROM EMPLOY
WHERE EMPLOY. ROW_TAG IN
(
        SELECT SECURITY. ROW_TAG
        FROM SECURITY
        WHERE SECURITY. USER_TAG IN
        (
        SELECT USER. USER_TAG
        FROM USER
        WHERE USER. USER_NAME=USERNAME
        )
)
WITH CHECK OPTION
```

In the first described embodiment the database contained information about employees that is to say the database was

This Sheet Has Been Replaced



a personnel database. Other types of information could be stored.

Figure 10 shows a system 1 in accordance with the invention, being used in a network manager. In this system the database includes information such as configuration management information on a telecommunications network 100 comprising a number of network elements 101 to 103 and their element managers 104 to 106. Users of the system 1, such as network managers concerned with the operation and control of the network 100, can then be provided with access to different parts of the database in the same way as earlier described.

This Sheet Has Been Replaced



This Sheet Has Been Replaced

CLAIMS

1. A method for controlling access by a user to a database comprising configuring at least part of the database such that at least some of the data of the configured database is associated with a security tag, configuring a storage structure of user identifiers and associated user tags, configuring a storage structure of user tags and associated security tags and mapping a user identifier to at least a subset of the data by determining from the storage structure of user tags and associated security tags a security tag or tags appropriate to the user tag of the user identifier and allowing the user access to the data from the configured database associated with the security tag or tags.

2. A method as claimed in claim 1 wherein data entered into the database is associated with a security tag appropriate to the user tag of the user identifier of the user entering the data.

3. A method as claimed in claim 1 or claim 2 wherein the database and or storage structures are tables.

4. A method for controlling access to a database by a user comprising creating a database of data and associated security tags, creating a first storage structure of identifers and associated identifier tags, creating a second storage structure of identifier tags and associated security tags and creating a view onto the database appropriate to at least one of the identifiers by determining from the first storage structure an associated identifier tag and determining from the second storage structure at least one security tag appropriate to that determined identifier tag and selecting from the database, data associated with that security tag or tags.

* C L M S *

5. A method as claimed in any preceding claim wherein the database is a telecommunications network database.

6. Apparatus for controlling access by a user to a database comprising means to configure: at least part of the database such that at least some of the data of the configured database is associated with a security tag; a storage structure of user identifiers and associated user tags; a storage structure of user tags and associated security tags; and means to map a user identifier to at least a subset of the data by determining from the storage structure of user tags and associated security tags a security tag or tags appropriate to the user tag of the user identifier and allowing the user access to the data from the configured database associated with the security tag or tags.

7. Apparatus as claimed in claim 6 comprising means to associate data entered into the database with a security tag appropriate to the user tag of the user identifier of the user entering the data.

8. Apparatus for managing a telecommunications network including apparatus as claimed in claim 6 or claim 7.

9. A set of data, accessed by a method as claimed in any one of claims 1 to 5

This Sheet Has Been Replaced



