DATABASE TABLE RECOVERY SYSTEM

FIELD OF THE INVENTION

This present invention relates in general to database recovery systems, and more particularly to a system and method for recovering a table of a database that does not require the entire database to be recovered.

BACKGROUND OF THE INVENTION

A database, such as DB2 from IBM Corp., may utilize tables that depend from a tablespace. The tablespace contains all of the semi-permanent data of the database, and the tables contain user updates and modifications to that data. Users access the database data from the tablespace by creating a subset of that data in a table, and then modify and update the table data. The users periodically update the database to overwrite the data in the tablespace with the table data.

One problem problem that occurs when a user updates the data in a table is that the data updates may be incorrect, such as due to a loss of power during a table update, an error in a data input routine, or other common sources of error. In such situations, it is necessary to rebuild the data in the table without storing the table data to the tablespace, because the corrupted data must not be allowed to be transmitted to the tablespace. For example, it is common to keep a log record file of table updates, such that the tablespace data may be updated using the log record file.

One drawback with known methods of recovering tables is that they require that the tablespace be recovered with the log record files. When the tablespace is recovered, all access to the tablespace must be restricted, including access by unaffected tables. In addition, access to all tables must also be restricted until the tablespace is recovered. As a result, if two

or more users are utilizing the tablespace to create two or more different tables, then all users are unable to access their tables while the data for a single table is recovered.

This drawback may result in complete disruption of work while the tablespace is being recovered.

Based on the foregoing, there is a need for a system and method for recovering a table that substantially eliminate or reduce the problems associated with conventional systems and methods for recovering a table. In particular, a system and method for recovering a table are needed that allow multiple users to access unaffected tables, and that do not require the tablespace to be recovered.

SUMMARY OF THE INVENTION

One embodiment of the present invention is a system and method for recovering a database table that depends on a tablespace. The system receives a backup copy of the tablespace and reads log records associated with the table. The system then applies the log records to the backup copy and builds new table data pages from the backup copy. Finally, the system scans the new table data pages for records of the first table and updates the table from the records.

The present invention provides many advantages. One advantage of the present invention is a system for recovering a table that does not require the tablespace from which the table depends and all other tables that depend from the tablespace to be reconstructed. The present invention returns the reconstructed table to the configuration it was in prior to corruption of the data without also modifying the configuration of the tablespace and other dependent tables.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a diagram of a tablespace with two dependent tables in accordance with one embodiment of the present invention.

Fig. 2 is a block diagram of a table recovery system in accordance with one embodiment of the present invention.

Fig. 3 is a flow chart of steps executed by one embodiment of the present invention for recovering a table.

DETAILED DESCRIPTION

Fig. 1 is a diagram of a tablespace with two dependent tables in accordance with one embodiment of the present invention. A tablespace 10 is broken down into four columns: column A, column B, column C, and column D. In addition, tablespace 10 is broken down into five pages: page one, page two, page three, page four, and page five. The column definitions of each column are uniform across each row and page. The page size of each page is uniform, but may necessarily include varying numbers of rows depending upon the number of characters in variable column fields of each row. For example, if each page contains 4,096 bytes of data, then one page may contain two rows having variable fields that total 4,096 bytes, and another page may contain four rows having variable fields that total 4,096 bytes. Thus, exemplary page one includes four rows, exemplary page two includes two rows, exemplary page three includes three rows, exemplary page four includes four rows, and exemplary page five includes two rows.

Exemplary table 20 and exemplary table 30 are drawn from tablespace 10. Table 20 includes column A data and column C data, while table 30 includes column B data and column D data. Table 20 and table 30 further include rows, but are not broken down into pages of data. In operation, table 20 may become corrupted due to improper data input,

systems operation, or other error sources. Any updates that were made to table 20 since the time that table 20 was last read from tablespace 10 must therefore be incorporated into the data of tablespace 10 before table 20 can be reconstructed.

Tables 20 and 30 include log records 22 and 24, respectively. Each log record 22 and 24 is a record of changes that were made to its corresponding table. In addition, a backup copy of tablespace.10 is maintained by the database. The backup copy may be stored on tape or disk, or any other storage medium.

In accordance with one embodiment of the present invention, the updates made to table 20 are applied to the backup copy of tablespace 10 data, which allows table 30 to be accessed in a read-only mode while table 20 is being recovered. Thus, it is not necessary to apply the log records for table 20 and table 30 to tablespace 10 and thus modify tablespace 10 if the data for table 20 is corrupted but the data for table 30 is not corrupted. Instead, table 20 is rebuilt from the backup copy of tablespace 10 to which the updates from the table 20 log record file have been applied. In this manner, the configuration of table 20, table 30, and tablespace 10 are maintained in the state that they were in prior to the is corruption of the table 20 data.

The present invention eliminates the need to apply the log record files from table 20 and table 30 to tablespace 10 in order to recover table 20. Users may have read-only access to other tables such as table 30, which reduces the disruption of work that may result from recovery of the data in table 20.

In particular, in one embodiment, read-only access is provided for table 30 to prevent any changes from being made to tablespace 10 while table 20 is being recovered. The backup copy of tablespace 10 is then stored into a working data memory. All of the rows of table 20 are then deleted, and any indices of table 20 are locked out from access to prevent changes to

the indices from being made. The log records associated with table 20 are then read from the log.

After the log records are read, they are read to a log record workspace and sorted. The log records are then applied to the backup copy of tablespace 10. Sorting the log records decreases the number of input and output operations that must be made to the backup copy of tablespace 10, which decreases the amount of processing time that is required to recover table 20.

After the log records have been applied to the backup copy of tablespace 10, new table data pages are built with the updated backup copy of tablespace 10. The new table data pages are then scanned for records, or database "rows", that belong to table 20. These records are selected and are used to update table 20. After table 20 is updated, update access to table 20, table 30 and tablespace 10 is allowed.

Fig. 2 is a block diagram of a table recovery system 200 in accordance with one embodiment of the present invention. Table recovery system 200 may be used to recover a table in which corrupted data is stored, such as a table of a DB2 tablespace, without requiring tablespace 10 and all other tables of tablespace 10 to be recovered also. Table recovery system 200 thus returns tablespace 10 and tables to the configuration that they were in prior to the corruption of data in the affected table.

Table recovery system 200 may be implemented in hardware, software, or a suitable combination of hardware and software. Table recovery system 200 is preferably software, such as operating code that operates on a general purpose computing platform. Table recovery system 200 is coupled to tablespace backup storage 202, log record storage system 204, and table storage (spacemap) 206. Tablespace backup storage 202, log is record storage system 204, and table storage 206 are disc storage mechanisms, random access memory, or

other suitable data storage devices that are used to store tablespace backup data, log record data, and table data respectively.

Table recovery system 200 comprises individual subsystems that may each be implemented as software, hardware, or a suitable combination of software and hardware. Furthermore, the subsystems of table recovery system 200 may be individual functional aspects of a single system. For example, each of the subsystems of table recovery system 200 may be functions or commands in a DB2 database system, or may also or alternatively be logic circuits, programmable devices, or other suitable systems or components.

Log records sorter system 210 is used to sort log records obtained from log record storage system 204. Log records sorter system 210 optimizes input/output operations by grouping sets of log records according to data page and log records location.

Data page updater system 212 is coupled to log records sorter system 210 and tablespace backup storage 202. Data page updater system 212 updates the backup copy of tablespace 10 from tablespace backup storage 202 with sorted log records is received from the log records sorter system 210.

Data page scanner system 214 is coupled to data page updater system 212. Data page scanner system 214 is operable to scan the updated tablespace backup copy and identify table rows for extraction.

Page row extractor system 216 is coupled to data page scanner system 214, and extracts the page rows identified by the data page scanner system 214.

Table row inserter system 218 receives the page rows from page row extractor system 216 and inserts them into the table that is being recovered. For example, the table that is being recovered may be stored on table storage system 206, which may be a space map that has been reinitiated with all rows deleted.

In operation, table recovery system 200 is used to recover a table from a tablespace without requiring all tables that depend from tablespace 10 to be rebuilt. For example, table recovery system 200 may operate in a DB2 Database System in which incorrect updates to a table have been made. If the table is not updated, then the incorrect updates will be imposed on tablespace 10, or the updates to the table will be lost. Table recovery system 200 is used to implement the updates to the table without requiring tablespace 10 and all other tables to be reconstructed.

In this manner, the table data may be reconstructed while other tables can be accessed in a "read only" mode. Furthermore, each table and the table space is returned to the configuration that was present prior to the corruption of the data in the affected table.

Fig. 3 is a flow chart of steps executed by one embodiment of the present invention for recovering a table. The steps may be implemented in software, hardware, or a suitable combination of software and hardware, but is preferably software such as operating code that is used with a general purpose computing platform, such as a personal computer or mainframe computer.

At step 302, all updates to tablespace 10 from which the table depends are completed. For example, a DB2 "acquiesce" function may be used to implement all queued updates to a tablespace, so as to prevent changes to tablespace 10 from occurring as table recovery is performed. The method then proceeds to step 304.

At step 304, access to other tables that are dependent from tablespace 10 is set to "read only" mode. The method then proceeds to step 306 where the backup copy of tablespace 10 is received from backup storage to system storage. For example, the backup copy of tablespace 10 may be stored to a random access memory, a magnetic data storage medium, or other suitable data storage devices. The method then proceeds to step 308.

At step 308, all rows of the table that is to be reconstructed are deleted. The method proceeds to step 310 where access to the indices of the table is locked out. The indices of the table may be used to verify that the data in the table has not changed after the recovery of the table has been completed. The method then proceeds to step 312 where log records are read from the log record storage associated with the table. The present invention implements those log updates that have occurred prior to table recovery on the copied tablespace, while leaving the original tablespace and all other dependent tables unaffected. In this manner, the table having corrupted data is returned to its original configuration, and no changes are implemented to tablespace 10 or other dependent tables. The method then proceeds to step 314.

At step 314, the log record file is copied to a log record work space. The log record work space is used at step 316 so that the log records may be sorted. The method then proceeds to step 318, where the log records are applied to tablespace 10 backup copy.

At step 320, new table data pages are built with the updated tablespace backup copy. The method then proceeds to step 322 where the new table data pages are scanned for records that belong to the table 10 that is being recovered. For example, each table comprises table keys that are columns from tablespace 10 that have been marked for use by the table. These table keys are extracted for each row. The method then proceeds to step 324 where the table is updated with the new table data pages from tablespace 10.

In operation, method 300 is used to recover a table after the table data has been corrupted without requiring tablespace from which the table depended on to be recovered. Method 300 allows access to other tables of tablespace 10 without requiring those tables to be rebuilt. Method 300 also optimizes the table recovery process by sorting the log records, such that input/output operations to tablespace 10 are optimized during the table recovery process.

As described, the present invention allows recovery of a table without requiring the tablespace from which the table depends and all other tables that depend from the tablespace to be reconstructed. The present invention returns the reconstructed table to the configuration it was in prior to corruption of the data without also modifying the configuration of the tablespace and other dependent tables.

Several embodiments of the present invention are specifically illustrated and/or described herein. However, it will be appreciated that modifications and variations of the present invention are covered by the above teachings and within the purview of the appended claims without departing from the spirit and intended scope of the invention.

For example, although the embodiments described allows users to access other tables in read-only mode, in other embodiments the users may be able to access the tables in update access mode while the table is recovered.

WHAT IS CLAIMED IS:

1. A system for recovering a table comprising:

a table recovery system; and

a tablespace access system coupled to the table recovery system, wherein the tablespace access system is operable to restrict access to the tablespace to read-only access.

2. The system of claim 1 wherein the table recovery system further comprises a log record sorter system operable to sort log records from a log record file.

3. The system of claim 2 further comprising a data page updater system coupled to the log record sorter system operable to apply log record updates to a data page.

4. The system of claim 3 further comprising a data page scanner system coupled to the data page updater system, the data page scanner system operable to locate records associated with a table in a data page.

5. The system of claim 4 further comprising a page row extractor system coupled to the data page scanner system operable to extract page rows from a data page that has been located by the data page scanner system.

6. The system of claim 5 further comprising a table row inserter system coupled to the page row extractor system operable to write extracted page rows to the table.

7. A method for recovering a table that depends on a tablespace, said method comprising:

allowing at least one other table to have read-only access to the tablespace; and

using a backup copy of the tablespace to restore the table.

8. The method of claim 7, further comprising allowing said at least one other table to have update access to the tablespace when the table is restored.

9. The method of claim 7 wherein using the copy of the tablespace to restore the table comprises:

deleting all rows of the table;

locking out access to indexes of the table;

applying log records to the tablespace backup copy to update the tablespace backup copy;

building new table data pages with the updated tablespace backup copy; and

updating the table with the new table data pages.

10. The method of claim 9 wherein applying log records to the tablespace backup copy to update the tablespace backup copy further comprises:

reading the log records from a log record file to a log record workspace;

sorting the log records; and

applying the log records to the tablespace backup copy.

11. The method of claim 9 wherein building new table data pages with the updated tablespace backup copy comprises scanning the new table data pages for records that belong to the table being recovered.

12. A method of recovering a first table that depends on a tablespace, said method comprising:

receiving a backup copy of the tablespace;

reading log records associated with the first table;

applying the log records to the backup copy;

building new table data pages from the backup copy;

scanning the new table data pages for records of the first table; and

updating the first table from the records.

13. The method of claim 12, further comprising:

limiting access of a second table to the tablespace to read-only before the first table is updated, wherein the second table depends on the tablespace.

14. The method of claim 13, further comprising:

providing update access to the second table after the first table is updated.

15. The method of claim 12, further comprising sorting the log records.

16. The method of claim 12 wherein the first table comprises rows and indices, the method further comprising:

deleting the rows; and

locking out the indices.

