Paper No. 19

UNITED STATES PATENT AND TRADEMARK OFFICE

_____

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

_____

Ex parte KARELLE L. CORNWELL, BARBARA T. COUCH, ROBERT W. LYLE, JAMES ZU-CHIA TENG and JULIE A. WATTS

_____

Appeal No. 2001-1605
Application 08/735,168

_____

ON BRIEF

_____

Before FLEMING, DIXON, and BARRY, **Administrative Patent Judges**.

FLEMING, **Administrative Patent Judge**.

DECISION ON APPEAL

This is a decision on appeal from the final rejection of claims 1 through 30, all the claims pending in the instant application.

Invention

The invention relates to a system and method within the field of database management for providing concurrent access to database contents by individually locking partitions of a table

without locking the entire table. See page 1 of Appellants' specification. The need to provide concurrent access to database content is a recurring requirement in current database technology. Concurrency applies to multiple applications requiring access to the same data at the same time through one database management system. The virtually universal technique of concurrency control is locking. In this regard, an application will acquire a lock on an object in the database in which it has an interest for reading, inserting, deleting, or changing. In order to ensure that the object will not change while the application is accessing it, the database management system provides a lock giving the application access to the object, while preventing other applications from modifying the object so long as the application holds the lock. See pages 1 and 2 of Appellants' specification.

In relational database systems, contents of a database are represented as tables of database values. Each table corresponds to a relation. In a relational database, a table can be divided into partitions. Each partition contains a portion of the data in the table. By partitioning a table, partitions containing more frequently-used data can be placed on faster devices, and parallel processing of data can be improved by spreading

partitions over different DASD volumes, with each I/O stream on a separate channel path. However, when access is granted to an application, the entire table is locked, even if a subset of the data (e.g. data for only a few partitions) will be accessed. When a table is partitioned, locking the entire table may degrade concurrency and database system performance. It is an object of Appellants' invention to provide a means for serializing access to a partitioned table in a relational database without requiring locking of the entire table when a serialized application requires access to less than all of the partitions of the table. See page 2 of Appellants' specification. Appellants provide this object by the use of selective partition locking that allows the database system to lock only those partitions of a partitioned table space to which access is sought by the first application. See page 3 of Appellants' specification.

Independent claim 1 present in the application represent Appellants' claimed invention and is reproduced as follows:

1. A method for controlling concurrency of access to data in a database system, comprising:

partitioning a table in the database system into a plurality of partitions;

receiving a request for access to data in the table;

determining a partition of the plurality of partitions that contains the data;

locking the partition in response to the request; and

granting access to the partition.

References

The references relied on by the Examiner are as follows:

Crus et al. (Crus)	4,716,528	Dec. 29, 1987
Bhide et al. (Bhide)	5,625,811	Apr. 29, 1997 (Filing date Oct. 31, 1994)
Bireley et al. (Bireley)	5,692,174	Nov. 25, 1997 (Filing date Oct. 5, 1995)

Rejections at Issue

Claims 1 through 9, 11 through 15 and 17 through 29 stand rejected under 35 U.S.C. § 102 as being anticipated by Bireley. Claims 10, 16 and 30 stand rejected under 35 U.S.C. § 103 as being unpatentable over Bireley in view of Crus. Claims 1, 5 through 7, 9, 21, 26, 27 and 29 stand rejected under 35 U.S.C. § 102 as being anticipated by Bhide.

Throughout the opinion, we will make reference to the briefs[1] and the answer for the respective details thereof.

[1] Appellants filed an appeal brief on February 19, 1999. Appellants filed a reply brief on June 1, 1999. The Examiner mailed out an office communication on August 16, 1999, stating that the reply brief has been entered.

OPINION

With full consideration being given to the subject matter on appeal, the Examiner's rejections and the arguments of Appellants and the Examiner, for the reasons stated **infra**, we reverse the Examiner's rejection of claims 1 through 9, 11 through 15 and 17 through 29 under 35 U.S.C. § 102, and we reverse the Examiner's rejection of claims 10, 16 and 30 under 35 U.S.C. § 103.

Rejection of Claims 1 through 9, 11 through 15 and 17 through 29 under 35 U.S.C. § 102 as being anticipated by Bireley

It is axiomatic that anticipation of a claim under § 102 can be found only if the prior art reference discloses every element of the claim. **See In re King**, 801 F.2d 1324, 1326, 231 USPQ 136, 138 (Fed. Cir. 1986) and **Lindemann Maschinenfabrik GMBH v. America Hoist & Derrick Co.**, 730 F.2d 1452, 1458, 221 USPQ 481, 485 (Fed. Cir. 1984).

Appellants argue that Bireley fails to anticipate the steps of partitioning a table in a database system into a plurality of partitions; receiving a request for access to data in a table; determining a partition of the plurality of partitions that contains the data; and locking the partition in response to the request as recited in Appellants' claims. See pages 4 through 7 of Appellants' brief. Appellants further argue that Bireley

fails to teach the sequences of the steps as recited in Appellants' claims. Appellants argue that the claims establish a relationship between individual elements of the claim (the steps) that must be taken into account when analyzing the claim for anticipation. Appellants note that the Examiner randomly hopscotches through Bireley and has not established clear sequence other than arbitrarily signified by the Examiner. See page 7 of the brief.

In response, the Examiner maintains that Bireley discloses partitioning a database into several partitions and reading an access request to such data wherein access to a partition table is deemed to contain a plurality of partitions. The Examiner points us to column 12, lines 20 through 63, and column 13, lines 13 through 27, of Bireley. The Examiner argues that the Bireley partition tables are located in separate physical storage devices which are accessed by query of a clarity. The Examiner argues that the fact that the partition tables are located in separate physical storage devices inherently requires a plurality of partitions. The Examiner points us to column 6, lines 39 through 48, of Bireley. The Examiner also maintains that Bireley discloses a lock manager which is deemed to lock a table partition. The Examiner points us to column 20, lines 40 through

67. The Examiner points out that Bireley states that his invention provides a global locking of resources and Bireley describes for illustrative purposes that the page resource is a lockable resource. The Examiner argues that a table partition is an accessible database resource and thereby is deem lockable.

In response, Appellants argue that the Examiner has only offered conclusionary statements about inherency and has not brought forth extrinsic evidence to establish that those steps, elements and limitations which he holds to be inherent are necessarily present in Bireley. Appellants argue that the Examiner as a result has not set forth a ***prima facie*** case of anticipation.

Our reviewing court states that "[t]o establish inherency, the extrinsic evidence 'must make clear that the missing descriptive matter is necessarily present in the thing described in the reference, and that it would be so recognized by person of ordinary skill.'" **In re Robertson** 169 F.3d 743, 745, 49 USPQ 1949, 1950-51 (Fed. Cir. 1999) **citing Continental Can Co. v. Mosanto Co.**, 948 F.3d 1264, 1268, 20 USPQ2d 1746, 1749 (Fed. Cir. 1991). "Inherency, however, may not be established by probabilities or possibilities. The mere fact that a certain

thing may result for a given set of circumstances is not sufficient." **Id. citing Continental Can Co. v. Monsanto Co.,** 948 F.3d 1264, 1269, 20 USPQ2d 1746, 1749 (Fed. Cir. 1991).

We note that Bireley is dealing with the problem of shared data database management systems comprising multiple database management systems and a single copy of data. Bireley decomposes long-running complex queries into a plurality of parallel tasks allocated across the database management systems of a shared database management system. See column 1, lines 5 through 11. Bireley solves the problem of a shared database management system having multiple database management systems in which each database management system decomposes a large data intensive query into multiple parallel tasks and allocates the parallel tasks across the database management systems. Bireley's shared database management system comprises multiple database management systems each of which has direct access to data stored on multiple storage devices. Each database management system does not have to communicate through another database management system to access data. Each database management system is further capable of decomposing a long-running complex query into multiple parallel tasks. See column 5, lines 30 through 42, of Bireley.

Bireley also solves the problem of dynamically disabling one or more modes of parallelism at the system level without the need to reconfigure the originating database management system. Bireley includes a dynamic mechanism providing selective disablement of the different modes of parallelism within a shared database management system. The mechanism allows a system administrator to dynamically respond to system utilization levels by selectively disabling parallel modes. See column 6, line 66, through column 7, line 6, of Bireley.

Finally, Bireley solves the problem of reducing workfile overhead in a shared database management system by providing a consuming database management system read-only access to a producing database management's working files. Bireley also includes a system and method for a consuming database management system in a shared database management system to only have read access to a producing database management working file. See column 7, lines 35 through 45, of Bireley. We will refer to these four problems as query, parallelism, selective disabling of parallel modes, and bufferpool coherency for working file data.

Figures 3 and 4 describe how Bireley solves the problem of query parallelism. See column 10, lines 53 through 55. The high level control flow of query parallelism is described in figures 5A and 5B. See column 11, lines 36 through 40. In step 504, shown in figure 5A, the originating database management system binds the query thereby creating a plan for executing the query. See column 11, lines 47 through 52, of Bireley. The operation of the originating database management system during the buying time shown in step 504 is represented by a control flow diagram shown in figure 6. Thus, the portion of Bireley which the Examiner relies on for disclosing partitioning a database into several partitions and reading an access request to such data wherein access to a partition database is deemed to contain a plurality of partitions is directed to the method of determining query parallelism. This portion of Bireley, column 12, lines 20 through 63, is describing the operation of the original database management during the buying time in which the database management system decomposes the query into multiple parallel tasks and allocates the parallel tasks across to the other database management systems. We agree with the Examiner that this portion of Bireley does teach that the data is in partitioned tablespace located in separate physical storage

devices. However, we fail to find that this requires a plurality of partitions because physical storage space does not neccessarily dictate the makeup of a database.

The selectively disabling of parallel modes is described in section 5, Originating the Database Management Systems at Run Time in column 13, line 28, of Bireley. The operation of the original originating database management system during run time is represented by a control flow diagram in figure 7. See column 13, lines 29 through 31, of Bireley. In section 6, Originating Database Management processing results at Run Time is described. See column 17, line 33, of Bireley. The operation of the original database management receiving processing results from parallel tasks during run time is represented by a control flow diagram in figure 8. See column 17, lines 33 through 36. Finally, Bireley describes in section 7, Parallel Tasks at Run Time. See column 19, line 51. The operation of the assisting database management system during run time is represented by a control flow diagram in figure 9. See column 19, lines 51 through 54.

Bireley finally addresses how to solve the buffer coherency for working file data problems in section 8, Setting Bufferpool Allocation Thresholds. See column 20, line 55. Section 9,

Global Compatible Class Locking, Bireley discusses the problem of parallel tasks causing contention on locking database resources during the processing of the query. See column 22, lines 31 through 33. Bireley teaches that it is imperative that shared database management systems treat all locking requests resulting from the parallel task of a query as belonging to the same family. Under such a scheme, one parallel task executing on one database management system can share database resources with other parallel tasks distributed across the database management systems. Bireley discloses that figure 11 describes global compatible class locking of the invention. Bireley uses a globally compatibility token as a key for locking and unlocking database resource along parallel tasks of a query.

From our review of Bireley, we fail to find that the Examiner has established that Bireley teaches locking a partition of a table in a database having a plurality of partitions. The only portion of Bireley that discusses locking does not teach or suggest locking only a single partition of a table having multiple partitions. Bireley only teaches about locking in columns 22 and 23. The portions that the Examiner relies on for partitioning is addressing a completely different problem. Therefore, we will not sustain the Examiner's rejection.

Rejection of Claims 10, 16 and 30 under 35 U.S.C. § 103.

We note that claims 10, 16 and 30 recite the above limitations discussed above due to their dependencies. We further note that the Examiner relies on Bireley for the above limitations. Furthermore, we find that Crus does not provide the missing pieces. Therefore, we will not sustain the rejection of claims 10, 16 and 30 under 35 U.S.C. § 103 as being unpatentable over Bireley in view of Crus.

Rejection of Claims 1, 5 through 7, 9, 21, 26, 27 and 29 under 35 U.S.C. § 102 as being Anticipated by Bhide.

Appellants argue that Bhide does not teach or suggest partitioning a table of a database into a plurality of partitions receiving a request for the access of the table, determining the partition that contains data, locking the partition in response to the request and granting a request to the partition as recited in Appellants' claim. In response, the Examiner maintains that Bhide's teaching of the locking of blocks reads on locking partitions as claimed. See page 12 of Examiner's answer.

Appellants argue that Bhide's teaching of locking blocks can-not read on Appellants' claims because a block denotes a unit of physical storage and not a partition of a database table that

is indicated by a range of a partitioning key. See page 10 of Appellants' brief and Appellants' reply brief.

We find that Bhide is directed to the storage of data involving physically partitioning the data and distributing the resulting partitions to responsible or owner nodes in system which become responsible for transitions involving their own corresponding partitions. See columns 1 and 2 of Bhide. Bhide does teach that during the transition period of transferring logical ownership from one logical owner to another, node will maintain block locks of physical blocks accessed by transitions at node and new transitions at node. See column 10, lines 41 through 50, and column 11, lines 50 through 57. However, we fail to find a teaching of partitioning a table of a database into a plurality of partitions, receiving a request for access to the table, determining a partition for containing the data, and locking the partition in response to the request as recited in Appellants' claims. Therefore, we will not sustain the Examiner's rejection of claims 1, 5 through 7, 9, 21, 26, 27 and 29 under 35 U.S.C. § 102.

In view of the foregoing, we have not sustained the Examiner's rejection of claims 1 through 9, 11 through 15 and 17 through 29 under 35 U.S.C. § 102. Furthermore, we have not sustained the rejection of claims 10, 16 and 30 under 35 U.S.C. § 103.

REVERSED

MICHAEL R. FLEMING Administrative Patent Judge	)	
JOSEPH L. DIXON Administrative Patent Judge	)	BOARD OF PATENT APPEALS AND INTERFERENCES
LANCE LEONARD BARRY Administrative Patent Judge	)	

MRF:pgg

Joseph A. Sawyer Jr.
Sawyer Law Group
P.O. Box 51418
Palo Alto, CA 94303