

Internal distribution code:

- (A) [-] Publication in OJ
- (B) [-] To Chairmen and Members
- (C) [-] To Chairmen
- (D) [X] No distribution

**Datasheet for the decision
of 10 August 2022**

Case Number: T 1576/20 - 3.5.07

Application Number: 13851963.2

Publication Number: 2901344

IPC: G06F17/30

Language of the proceedings: EN

Title of invention:

System and method for flexible distributed massively parallel processing (MPP) database

Applicant:

Huawei Technologies Co., Ltd.

Headword:

Database partitioning/HUAWEI

Relevant legal provisions:

EPC Art. 54, 56

RPBA 2020 Art. 12(3), 12(5), 12(6)

Keyword:

Novelty - main request and auxiliary request 1 (no)
Inventive step - auxiliary requests 2, 4 and 5 (no)
Request not admitted by the examining division - auxiliary
request 3 - admitted (no)



Beschwerdekammern
Boards of Appeal
Chambres de recours

Boards of Appeal of the
European Patent Office
Richard-Reitzner-Allee 8
85540 Haar
GERMANY
Tel. +49 (0)89 2399-0
Fax +49 (0)89 2399-4465

Case Number: T 1576/20 - 3.5.07

D E C I S I O N
of Technical Board of Appeal 3.5.07
of 10 August 2022

Appellant: Huawei Technologies Co., Ltd.
(Applicant) Huawei Administration Building
Bantian
Longgang District
Shenzhen, Guangdong 518129 (CN)

Representative: Körber, Martin Hans
Mitscherlich PartmbB
Patent- und Rechtsanwälte
Sonnenstrasse 33
80331 München (DE)

Decision under appeal: **Decision of the Examining Division of the
European Patent Office posted on 27 January 2020
refusing European patent application
No. 13851963.2 pursuant to Article 97(2) EPC**

Composition of the Board:

Chair J. Geschwind
Members: R. de Man
P. San-Bento Furtado

Summary of Facts and Submissions

I. The appellant (applicant) appealed against the decision of the examining division refusing European patent application No. 13851963.2, which was published as international application WO 2014/067449.

II. The contested decision cited the following documents:

D1: "Partition (database)", Wikipedia, 10 August 2012, retrieved from [https://en.wikipedia.org/w/index.php?title=Partition_\(database\)&oldid=506792667](https://en.wikipedia.org/w/index.php?title=Partition_(database)&oldid=506792667);

D2: A. Zitelli: "Oracle 11g Reference Partitioning - Benefits, Hazards & Other Considerations", NoCOUG Spring Conference, 20 May 2010, retrieved from http://www.nocoug.org/download/2010-05/Zitelli-Reference_Partitioning_NoCOUG.pdf.

The examining division decided that the subject-matter of claim 1 of the main request and of auxiliary requests 1 and 2 lacked inventive step over document D2. Auxiliary request 3 was not admitted into the proceedings under Rule 137(3) EPC.

III. In its statement of grounds of appeal, the appellant requested that the decision under appeal be set aside and that a patent be granted on the basis of the claims of the main request or, in the alternative, of one of auxiliary requests 1, 2 and 3.

IV. In a communication accompanying the summons to oral proceedings, the board referred to the following document cited in the international search report:

D3: US 2005/0187977 A1, 25 August 2005.

It expressed the preliminary view that the subject-matter of claim 1 of the main request and auxiliary request 1 lacked novelty over document D3, that the subject-matter of claim 1 of auxiliary request 2 lacked inventive step over document D3, and that auxiliary request 3 should not be admitted into the appeal proceedings.

- V. With a letter submitted in preparation for the oral proceedings, the appellant filed new auxiliary requests 4 and 5. It did not provide further comments on the main request and auxiliary requests 1 to 3.
- VI. In a further written submission, the appellant withdrew its request for oral proceedings. In response, the board cancelled the oral proceedings.
- VII. The appellant requests that the decision under appeal be set aside and that a patent be granted on the basis of the claims of the main request or, in the alternative, of one of auxiliary requests 1 to 5.
- VIII. Claim 1 of the main request reads as follows:

"A method for logically dividing a database (24) into multiple independently operated smaller databases, characterized in that the method comprises:

assigning a primary key to a first table (26) in the database (24) and a foreign key to a second table (28) in the database (24), the foreign key of the second table (28) is identical to the primary key of the first table (26), wherein the primary key and foreign key are assigned to the first table (26) and

second table (28) based on the type of data or values stored in each column of the first table (26) and the second table (28);

determining a number of partition groups (32) desired for the database (24);

partitioning the first table (26) into a plurality of first partitions (34) based on the primary key assigned and the number of partition groups (32) desired;

partitioning the second table (28) into a plurality of second partitions (36) based on the foreign key assigned and the number of partition groups (32) desired, wherein the number of first partitions (34) and second partitions (36) is the same; and

distributing the first partitions (34) and the second partitions (36) to the partition groups (32) as desired;

wherein each one of the partition groups (32) comprises one of the first partitions (34) and one of the second partitions (36); and wherein each one of the partition groups (32) is an independently operated database, and contains partitions holding data that is linked through some attribute."

IX. Claim 1 of auxiliary request 1 differs from claim 1 of the main request in that the following text has been added at the end of the claim:

"; and wherein each of the partition groups (32) is assigned to an independent processor and/or and independent memory".

X. Claim 1 of auxiliary request 2 differs from claim 1 of the main request in that the following text has been added at the end of the claim:

"; the partition groups (32) comprises [sic] indexes, catalogs, and permissions relating to the data in corresponding partition group [sic]".

XI. Claim 1 of auxiliary request 3 differs from claim 1 of the main request in that the text after "partitioning the second table ... is the same;" has been replaced with:

"establishing the partition groups (32) as desired; wherein each one of the partition groups (32) comprises one of the first partitions (34) and one of the second partitions (36); and wherein each one of the partition groups (32) is an independently operated database, and contains partitions holding data that is linked through some attribute; and

after the partition groups have been established, each of the partition groups is assigned to an independent processor and/or an independent memory".

XII. Claim 1 of auxiliary request 4 differs from claim 1 of auxiliary request 1 in that the following text has been added at the end of the claim:

", and the partition groups (32) comprise indexes, catalogs, and permissions relating to the data in corresponding partition group [sic]; and

wherein the method further comprises:

assigning a third foreign key to a third table in the database, the third foreign key of the third table is identical to the primary key of the first table, wherein the third foreign key is assigned to the third table based on the type of data or values stored in each column of the third table;

partitioning the third table into a plurality of third partitions based on the third foreign key

assigned and the number of partition groups desired, wherein the number of first partitions and second partitions and the number of third partitions are the same; and

distributing the third partitions to the partition groups as partitioned".

XIII. Claim 1 of auxiliary request 5 differs from claim 1 of auxiliary request 4 in that "the third foreign key of the third table is identical to the primary key of the first table" has been replaced with "the third foreign key of the third table is identical to a primary key of the second table".

XIV. The appellant's argument, where relevant to this decision, are discussed in detail below.

Reasons for the Decision

1. The application relates to partitioning a relational database into "partition groups", in particular in the context of a massively parallel processing database management system based on a shared-nothing architecture (see paragraphs [0002] to [0006] of the published application).

Main request

2. *The invention as defined by claim 1*

2.1 Claim 1 is directed to a method for "logically dividing a database into multiple independently operated smaller databases".

2.2 First, a primary key is assigned to a first table and a corresponding foreign key to a second table. These assignments are "based on the type of data or values stored in each column of the first table and the second table".

The board notes that primary and foreign keys are well-known concepts in the field of relational databases. A relational database table normally has a primary key that uniquely identifies each row of the table, and it may use foreign keys which are the primary keys of other tables to refer to rows of those other tables. Such primary and foreign keys are naturally "based on the type of data or values stored in each column" of the database tables.

For example, a (first) department table which contains a row for each department in a company may use a department ID as the primary key to uniquely identify each department. A (second) employee table which contains a row for each employee of a company may use an employee ID as the primary key to uniquely identify each employee, and it may use the department ID as a foreign key to link each employee row to the row in the department table corresponding to the department in which the employee works.

Hence, this first step merely sets the context for the invention: first and second relational database tables with primary and foreign keys.

2.3 Next, a desired number of partition groups is determined. This "desired" number may be determined, for example, by a human operator.

2.4 The first table is then partitioned on the basis of the primary key into the desired number of partitions. The second table is likewise partitioned on the basis of the same key (which is a foreign key for the second table).

To continue the example, if the desired number of partition groups is two and the primary key is the department ID which runs from 1 to 10, then the rows of the department table with department ID 1 to 5 may form a first partition, and the rows with department ID 6 to 10 may form a second partition. The rows in the employee table, which include the department ID as foreign key, are then partitioned accordingly.

2.5 Finally, the partitions are "distributed" to "partition groups", each partition group being an "independently operated database" and "holding data that is linked through some attribute".

In terms of the board's example, a first database is formed with a department table and an employee table for departments and employees with department ID 1 to 5, and a second database is formed with a department table and an employee table for departments and employees with department ID 6 to 10. Each of these databases holds data that is linked through the department ID.

2.6 The appellant did not contest the board's interpretation of claim 1 as set out above.

3. *Novelty*

- 3.1 Document D3 relates to a "shared-nothing" parallel database system comprising a master node and multiple slave nodes (paragraphs [0007] and [0034]).

It discloses, with reference to the sample database schema shown in Figure 4, a database partitioning method which, for a given database, horizontally partitions the fact table (the "LINEITEM" table in Figure 4) and one of the dimension tables (the "ORDERS" table in Figure 4) by means of hash partitioning using a common key and distributes the partitions across slave nodes (paragraphs [0007], [0008], [0043] and [0044]).

Paragraph [0043] discloses that the common key is the primary key of the dimension table and a foreign key of the fact table. The dimension table thus corresponds to the "first table" of claim 1, and the fact table to the "second table".

- 3.2 The board notes that horizontally partitioning a table means partitioning the rows of the table (see document D1, "Partition methods"). Since the partitioning method of document D3 applies the same partitioning criterion to both the fact table and the dimension table, an equal number of partitions is obtained for both tables.
- 3.3 Hence, the subject-matter of claim 1 lacks novelty over document D3 (Article 54(1) and 54(2) EPC).

Auxiliary request 1

4. Claim 1 of auxiliary request 1 adds to claim 1 of the main request that each of the partition groups is

assigned to an independent processor and/or an independent memory.

5. Since the slave nodes of document D3 include a processor and a memory (see Figure 1 and paragraph [0036]), the subject-matter of claim 1 of auxiliary request 1 also lacks novelty over document D3 (Article 54(1) and (2) EPC).

Auxiliary request 2

6. Claim 1 of auxiliary request 2 adds to claim 1 of the main request that the partition groups comprise "indexes, catalogs, and permissions relating to the data in corresponding partition group [sic]".
7. The added features render the subject-matter of claim 1 new over document D3. However, since indexes, catalogs and permissions are commonplace features of relational database systems, it is an obvious choice to include them in the corresponding partitions of the independently operated databases.

Hence, the subject-matter of claim 1 of auxiliary request 2 lacks inventive step over document D3 (Article 56 EPC).

Auxiliary request 3

8. *Admission into the appeal proceedings*
 - 8.1 The examining division decided not to admit auxiliary request 3 into the proceedings under Rule 137(3) EPC for being late filed and not *prima facie* allowable. In its statement of grounds of appeal, the appellant did

not give any reason why auxiliary request 3 should be admitted into the appeal proceedings.

Hence, in respect of auxiliary request 3, the statement of grounds of appeal does not comply with Article 12(3) RPBA 2020, which requires that the statement of grounds of appeal contains the appellant's complete appeal case and sets out clearly and concisely why it is requested that the decision under appeal be set aside.

The board therefore has the discretion not to admit this request into the appeal proceedings (Article 12(5) RPBA 2020).

- 8.2 Moreover, according to Article 12(6), first sentence, RPBA 2020, the board may admit a request which was not admitted in the first-instance proceedings only if the decision not to admit the request suffered from an error in the use of discretion or if the circumstances of the appeal case justify its admission.

The presence of such an error or such circumstances was not argued by the appellant and is not apparent to the board.

- 8.3 The board therefore does not admit auxiliary request 3 into the appeal proceedings.

Auxiliary requests 4 and 5

9. *Admission into the appeal proceedings*

Auxiliary requests 4 and 5 were filed after the notification of the board's summons to oral proceedings and represent reasonable reactions to the novelty and inventive-step objections based on document D3, which

were raised for the first time in the board's communication accompanying the summons to oral proceedings. The admission of these requests into the appeal proceedings is therefore justified by an exceptional circumstance, as required by Article 13(2) RPBA 2020.

10. *Inventive step*

10.1 Claim 1 of auxiliary request 4 is based on claim 1 of auxiliary request 1.

Like claim 1 of auxiliary request 2, it adds that the partition groups comprise "indexes, catalogs, and permissions relating to the data in corresponding partition group".

It further adds features relating to a third database table with a third foreign key identical to the primary key of the first table. This third table is partitioned into a plurality of third partitions based on the third foreign key. The number of third partitions is equal to the number of first and second partitions, and the third partitions are also distributed to the partition groups.

10.2 Claim 1 of auxiliary request 5 corresponds to claim 1 of auxiliary request 4, except that the "third foreign key" is identical to the primary key of the second table.

10.3 Since indexes, catalogs and permissions are commonplace features of relational database systems, it is an obvious choice to include them in the corresponding partitions of the independently operated databases.

10.4 As for the features relating to the third database table, the board notes that, in auxiliary request 4, this third table plays exactly the same role as the second table.

To continue the example given in point 2. above, the third table may be a printer table which contains a row for each printer in the company and which uses a printer ID as the primary key to uniquely identify each printer and the department ID as a foreign key to link each printer row to the row in the department table corresponding to the department to which the printer belongs. The rows in the printer table are then partitioned according to the value of their department ID foreign key.

The board notes that both the second table and the third table of auxiliary request 4 are "child tables" of the first table in the sense that they have a foreign key corresponding to the primary key of the first table.

10.5 The third table of auxiliary request 5 is not a child table of the first table but a child table of the second table (and therefore a "grandchild table" of the first table).

The claim does not specify what it means to partition the third table "based on the third foreign key", which is now not identical to the primary key of the first table. However, the skilled person understands that what is intended is that a row of the third table is assigned to the same partition as the row of the second table to which it is linked via the third foreign key.

To continue the example, the third table of auxiliary request 5 may be a laptop table which contains a row for each laptop in the company and which uses a laptop ID as the primary key to uniquely identify each laptop and the employee ID as a foreign key to link each laptop row to the row in the employee table corresponding to the employee to which the laptop belongs. A row in the laptop table is assigned to the same partition as the row of the employee table to which it is linked via the employee ID.

- 10.6 Referring to paragraphs [0008], [0010], [0011], [0043], [0044] and [0046], the appellant argued that in document D3 only the fact table and the first dimension table were partitioned on a common key. The remaining dimension tables were not partitioned on a common key or on a third foreign key identical to the primary key of the fact table (i.e. the second table) but instead by row or by column.
- 10.7 It is true that in document D3 the remaining dimension tables are not necessarily partitioned on the common key. One reason for this is that the remaining dimension tables do not necessarily have a foreign key which is linked, directly or indirectly, to the primary key of the first table, in which case there is no such common key.
- 10.8 In document D3, the remaining tables may be horizontally partitioned, i.e. by row, or vertically partitioned, i.e. by column (see paragraphs [0010] and [0048]).

Horizontal partitioning is described in paragraph [0051], which explains that an algorithm is used to assign rows to partitions/slave nodes. One such

algorithm is date partitioning (paragraphs [0052] and [0053]), which assign rows to partitions/slaves nodes on the basis of relevant dates within the data. Another algorithm is hash partitioning, whereby a row is assigned to a partition/slave node on the basis of an appropriate hash key (paragraphs [0054] and [0068]).

- 10.9 Hence, document D3 discloses that the first and second tables are horizontally partitioned, i.e. by row, by using the primary key of the first table (which is the foreign key of the second table) as hash key, and that the remaining tables may also be horizontally partitioned on the basis of an appropriate hash key.
- 10.10 In the board's view, the skilled person reading document D3 would understand that partitioning the second table in accordance with the partitioning of the first table (by using the primary key of the first table as hash key) is advantageous in that it ensures that related records in the first and second tables are assigned to the same partition, which may speed up the processing of queries (see D3, paragraph [0043]). The skilled person would also understand that the same advantages can be achieved by partitioning a remaining table in accordance with the partitioning of the first table, where that is possible.
- 10.11 Hence, if a remaining table is a child of the first table, i.e. has a foreign key identical to the primary key of the first table, it would be obvious to partition that remaining table in the same way as the second table by using the primary key as hash key, as expressed in claim 1 of auxiliary request 4.
- 10.12 Likewise, if a remaining table is indirectly linked to the first table via a foreign key, for example if the

remaining table is a child of the second table, which itself is a child of the first table, it would be obvious to partition that remaining table by using the primary key as hash key, the value of which can be accessed via the foreign key of the remaining table (which links to a row in the second table, which itself links to a row in the first table via its foreign key). This is expressed in claim 1 of auxiliary request 5.

- 10.13 In the embodiment described in document D3, which is based on the database schema shown in Figure 4, the "ORDERS" table is the first dimension table, corresponding to the first table in the independent claims, and the "LINEITEM" table is the fact table, corresponding to the second table in the independent claims (see paragraphs [0042] and [0043]). In this embodiment, none of the other tables shown in Figure 4 corresponds to the third table of either auxiliary request 4 (i.e. another child table of the "ORDERS" table) or auxiliary request 5 (i.e. a child table of the "LINEITEM" table).

However, paragraph [0041] confirms that the schema shown in Figure 4 is only meant as an example and explains that "possible host schemas include, but are not limited to, star schemas, snowflake schemas and normalised schemas". The board indeed cannot see any reason why the partitioning techniques disclosed in document D3 could not be applied to other relational database schemas, including the schema with department, employee, printer and laptop tables described in points 2. and 10.4 and 10.5 above. This schema, with a parent table having two child tables and a grandchild table, is a common basic example of a relational database schema, and in the board's judgment there is no inventive merit in applying the partitioning

techniques of document D3 to this example. When doing so, the skilled person would carry out a partitioning method falling within the scope of claim 1 of both auxiliary request 4 and auxiliary request 5.

10.14 Hence, the subject-matter of claim 1 of auxiliary requests 4 and 5 lacks inventive step (Article 56 EPC).

Order

For these reasons it is decided that:

The appeal is dismissed.

The Registrar:

The Chair:



S. Lichtenvort

J. Geschwind

Decision electronically authenticated