

CLAIMS

What is claimed is:

1. A method for managing primary and secondary databases, comprising the following steps:

5 determining whether a lock held by a primary database expires, wherein the primary database and secondary databases share the lock;

 in response to determining that the lock held by the primary database has expired, determining whether a lock renewal request of the primary database is received; and

 in response to determining that the lock renewal request of the primary database
10 is not received, selecting one of the secondary databases as a new primary database, and controlling the primary database to be switched to a secondary database.

2. The method for managing primary and secondary databases according to claim 1, wherein after the determining that the lock held by the primary database has expired, the method further comprises:

15 sending a lock expiration notification to the secondary database; and

 receiving a lock request sent by the secondary database based on the lock expiration notification, and recording a receiving time of the lock request.

3. The method for managing primary and secondary databases according to claim 2, wherein the selecting one of the secondary databases as a primary database
20 specifically comprises:

 selecting, as the primary database, a secondary database with the earliest receiving time.

4. The method for managing primary and secondary databases according to claim 1, wherein a priority of the primary database is higher than a priority of the secondary
25 database.

5. The method for managing primary and secondary databases according to claim 1, wherein an update cycle of the lock is T1, the primary database sends, at a cycle T2, a lock request for querying a status of the lock, and the secondary database sends, at a cycle T3, a lock request for querying the status of the lock, wherein T2 is less than T1,
30 and T3 is greater than or equal to T1.

6. The method for managing primary and secondary databases according to claim 1, wherein the secondary database comprises a hot standby database and a disaster

recovery database, the primary database and the hot standby database are located in a same data center, and the primary database and the disaster recovery database are located in different data centers.

7. The method for managing primary and secondary databases according to claim 4 or 5, wherein after the original primary database is restored, the method further comprises:

receiving a lock request sent by the original primary database, and controlling the lock to be held by the original primary database after the lock expires, so that the original primary database is restored to a primary database, and the current primary database is restored to a secondary database.

8. The method for managing primary and secondary databases according to claim 1, wherein after the selecting one of the secondary databases as a new primary database, the method further comprises:

continuing to determine whether a lock held by the current primary database expires;

in response to determining that the lock held by the current primary database has expired, determining whether lock requests of the original primary database, the current primary database, and other secondary databases are received; and

in response to determining that the lock request of the original primary database is received, restoring the original primary database to a primary database, and restoring the current primary database to a secondary database; or

in response to determining that the lock request of the original primary database is not received and the lock requests of the current primary database and the other secondary databases are received, maintaining the current primary database as a primary database; or

in response to determining that the lock requests of the original primary database and the current primary database are not received and the lock requests of the other secondary databases are received, selecting one of the other secondary databases as a primary database, and restoring the current primary database to a secondary database.

9. A lock arbitration server, comprising:

a first determining module, configured to determine whether a lock held by a primary database expires, wherein the primary database and secondary databases share the lock;

a second determining module, configured to: in response to determining that the

lock held by the primary database has expired, determine whether a lock renewal request of the primary database is received; and

a first processing module, configured to: in response to determining that the lock renewal request of the primary database is not received, select one of the secondary
5 databases as a new primary database, and control the primary database to be switched to a secondary database.

10. The lock arbitration server according to claim 9, further comprising:

a first sending module, configured to send a lock expiration notification to the secondary database after the first determining module determines that the lock held by
10 the primary database has expired; and

a record module, configured to: receive a lock request sent by the secondary database based on the lock expiration notification, and record a receiving time of the lock request.

11. The lock arbitration server according to claim 10, wherein the first processing
15 module is further configured to:

select, as the primary database, a secondary database with the earliest receiving time.

12. The lock arbitration server according to claim 9, wherein a priority of the primary database is higher than a priority of the secondary database.

20 13. The lock arbitration server according to claim 9, wherein

an update cycle of the lock is T1, the primary database sends, at a cycle T2, a lock request for querying a status of the lock, and the secondary database sends, at a cycle T3, a lock request for querying the status of the lock, wherein T2 is less than T1, and T3 is greater than or equal to T1.

25 14. The lock arbitration server according to claim 9, wherein

the secondary database comprises a hot standby database and a disaster recovery database, the primary database and the hot standby database are located in a same data center, and the primary database and the disaster recovery database are located in different data centers.

30 15. The lock arbitration server according to claim 12 or 13, further comprising:

a first receiving module, configured to: after the original primary database is restored, receive a lock request sent by the original primary database, wherein

the first processing module is further configured to control the lock to be held by the original primary database after the lock expires, so that the original primary

database is restored to a primary database, and the current primary database is restored to a secondary database.

16. The lock arbitration server according to claim 9, wherein

the first determining module is further configured to continue to determine
5 whether a lock held by the current primary database expires;

the second determining module is further configured to: in response to determining that the lock held by the current primary database has expired, determine whether lock requests of the original primary database, the current primary database, and other secondary databases are received;

10 a second processing module is configured to: in response to determining that the lock request of the original primary database is received, restore the original primary database to a primary database, and restore the current primary database to a secondary database;

a third processing module is configured to: in response to determining that the
15 lock request of the original primary database is not received and the lock requests of the current primary database and the other secondary databases are received, maintain the current primary database as a primary database; and

a fourth processing module is configured to: in response to determining that the lock requests of the original primary database and the current primary database are
20 not received and the lock requests of the other secondary databases are received, select one of the other secondary databases as a primary database, and restore the current primary database to a secondary database.

17. A primary database server, comprising:

a second sending module, configured to send a lock renewal request to a lock
25 arbitration server when a lock held by a primary database expires, so that the lock arbitration server selects one of secondary databases as a new primary database when the lock arbitration server does not receives the lock renewal request; and

a switching module, configured to control the primary database to be switched to a secondary database.

30 18. The primary database server according to claim 17, wherein a priority of the primary database is higher than a priority of the secondary database.

19. The primary database server according to claim 17, wherein an update cycle of the lock is T1, the primary database server sends, at a cycle T2, a lock request for querying a status of the lock, and the secondary database sends, at a cycle T3, a lock

request for querying the status of the lock, wherein T2 is less than T1, and T3 is greater than or equal to T1.

20. The primary database server according to claim 17, wherein

the secondary database comprises a hot standby database and a disaster recovery
5 database, the primary database and the hot standby database are located in a same data center, and the primary database and the disaster recovery database are located in different data centers.

21. The primary database server according to claim 18 or 19, wherein the primary database server further comprises:

10 a third sending module, configured to: after the original primary database is restored, send a lock request to the lock arbitration server, so that the lock arbitration server receives the lock request sent by the original primary database, and controls the lock to be held by the original primary database after the lock expires, so that the original primary database is restored to a primary database, and the current primary
15 database is restored to a secondary database.