CLAIMS

What is claimed is:

1. A method of querying, with a relational database management query, a distributed ledger platform that implements a distributed ledger including transaction data, comprising:

creating, using at least one processor of the distributed ledger platform, a database on the distributed ledger and recording information about the database on the distributed ledger;

converting, using the at least one processor, a received relational database management query into a query operation that may be processed by the database on the distributed ledger;

executing, using the at least one processor, the query operation on the database on the distributed ledger to generate a query result; and

logging, using the at least one processor, the query result to the database on the distributed ledger in a form that may be processed by the database on the distributed ledger for inclusion on the distributed ledger.

2. A method as in claim 1, further comprising generating, using the at least one processor, the relational database management query as a structured query language (SQL) query from one of (1) a user via a user interface; (2) a user application via an application program interface; and (3) a smart contract.

3. A method as in claim 2, further comprising parsing, using the at least one processor, the SQL query into an SQL operation comprising one of a data manipulation language (DML) write operation, a DML read operation, and a data definition language (DDL) operation, and creating a JavaScript Object Notation (JSON) representation of the SQL operation and any relational data included in the SQL operation.

4. A method as in claim 3, further comprising determining, using the at least one processor, whether a transaction resulting from execution of the SQL operation has progressed toward acceptance into the distributed ledger to a platform-relevant extent.

5. A method as in claim 4, further comprising adding, using the at least one processor, an operation broadcast status (OPSTAT) instruction at the end of a DML Write operation to fetch a more definitive status of a corresponding transaction.

6. A method as in claim 3, wherein executing the query operation on the distributed ledger comprises confirming, using the at least one processor, that an entity invoking the SQL query has authority to perform the SQL operation.

7. A method as in claim 3, wherein executing the query operation on the distributed ledger comprises creating, using the at least one processor, a JSON representation of the query operation and a result of the query operation that may be processed and stored on the distributed ledger.

8. A method as in claim 7, wherein the SQL operation is an SQL DML write operation, further comprising retrieving, using the at least one processor, data from the distributed ledger as required to execute the SQL DML write operation, executing, using the at least one processor, the SQL DML write operation including the retrieved data, preparing, using the at least one processor, JSON representations of any new or updated records in the query result, and committing, using the at least one processor, the SQL DML write operation and any updated records to the distributed ledger in the form that may be processed by the database on the distributed ledger for inclusion on the distributed ledger.

9. A method as in claim 8, further comprising monitoring, using the at least one processor, the distributed ledger for progress of the SQL DML write operation toward acceptance into the distributed ledger and informing, using the at least one processor, an entity invoking the SQL query whether the SQL DML write operation made such progress to a platform-relevant extent.

10. A method as in claim 7, wherein the SQL operation is an SQL DML read operation, further comprising retrieving, using the at least one processor, data from the distributed ledger as required to execute the SQL DML read operation, executing, using the at least one processor, the SQL DML read operation including the retrieved data, preparing, using the at least one

processor, a JSON representation of the query result, and logging, using the at least one processor, the JSON representation of the query result to the distributed ledger in the form that may be processed by the database on the distributed ledger for inclusion on the distributed ledger.

11. A method as in claim 10, further comprising returning, using the at least one processor, the JSON representation of the query result to an entity invoking the SQL query.

12. A method as in claim 1, further comprising creating, using the at least one processor, the database within the distributed ledger by executing a configuration transaction that creates a channel and defines which peer computers will maintain a copy of the database and a regular transaction that writes metadata about the database, and creating, using the at least one processor, code defining transaction data in the distributed ledger and transaction instructions for modifying the transaction data.

13. A method as in claim 12, further comprising creating, using the at least one processor, code defining attributes of the database in a JavaScript Object Notation (JSON) as the transaction data in the distributed ledger and causing, using the at least one processor, the database to be updated with the database attributes.

14. A method as in claim 1, further comprising creating, using the at least one processor, a parsed and translated structured query language (SQL) query that accounts for the form that may be processed by the database on the distributed ledger for inclusion on the distributed ledger.

15. A system comprising:

a distributed ledger platform that implements a distributed ledger including transaction data;

a database on the distributed ledger; and

at least one processor that executes instructions to implement a relational data management and organization system that performs operations comprising:

recording information about the database on the distributed ledger;

converting a received relational database management query into a query operation that may be processed by the database on the distributed ledger;

executing the query operation on the database on the distributed ledger to generate a query result; and

logging the query result to the database on the distributed ledger in a form that may be processed by the database on the distributed ledger for inclusion on the distributed ledger.

16. A system as in claim 15, further comprising means for generating the relational database management query as a structured query language (SQL) query comprising one of: (1) a user interface to the relational data management and organization system through which a user provides the SQL query; (2) a user application that passes the SQL query to the relational data management and organization system via an application program interface to the relational data management and organization system; and (3) a smart contract.

17. A system as in claim 16, wherein the at least one processor further executes instructions to parse the SQL query into an SQL operation comprising one of a data manipulation language (DML) write operation, a DML read operation, and a data definition language (DDL) operation, and to create a JavaScript Object Notation (JSON) representation of the SQL operation and any relational data included in the SQL operation.

18. A system as in claim 17, wherein the at least one processor further executes instructions to determine whether a transaction resulting from execution of the SQL operation has progressed toward acceptance into the distributed ledger to a platform-relevant extent.

19. A system as in claim 18, wherein the at least one processor further executes instructions to add an operation broadcast status (OPSTAT) instruction at the end of a DML Write operation to fetch a more definitive status of a corresponding transaction.

20. A system as in claim 17, wherein the at least one processor executes the query operation on the distributed ledger by executing instructions to confirm that an entity invoking the SQL query has authority to perform the SQL operation.

21. A system as in claim 17, wherein the at least one processor further executes instructions to create a JSON representation of the query operation and a result of the query operation that may be processed and stored on the distributed ledger.

22. A system as in claim 21, wherein the SQL operation is an SQL DML write operation, and wherein the at least one processor further executes instructions to retrieve data from the distributed ledger as required to execute the SQL DML write operation, to execute the SQL DML write operation including the retrieved data, to prepare JSON representations of any new or updated records in the query result, and to commit the SQL DML write operation and any updated records to the distributed ledger in the form that may be processed by the database on the distributed ledger for inclusion on the distributed ledger.

23. A system as in claim 22, wherein the at least one processor further executes instructions to monitor the distributed ledger for progress of the SQL DML write operation toward acceptance into the distributed ledger and to inform an entity invoking the SQL query whether the SQL DML write operation made such progress to a platform-relevant extent.

24. A system as in claim 21, wherein the SQL operation is an SQL DML read operation, and wherein the at least one processor further executes instructions to retrieve data from the distributed ledger as required to execute the SQL DML read operation, to execute the SQL DML read operation including the retrieved data, to prepare a JSON representation of the query result, and to log the JSON representation of the query result to the distributed ledger in the form that may be processed by the database on the distributed ledger for inclusion on the distributed ledger.

25. A system as in claim 24, wherein the at least one processor further executes instructions to return the JSON representation of the query result to an entity invoking the SQL query.

26. A system as in claim 15, wherein the at least one processor executes instructions to create the database within the distributed ledger by executing a configuration transaction that

creates a channel and defines which peer computers will maintain a copy of the database and a regular transaction that writes metadata about the database, and to create code defining transaction data in the distributed ledger and transaction instructions for modifying the transaction data.

27. A system as in claim 26, wherein the at least one processor further executes instructions corresponding to code that defines attributes of the database in a JavaScript Object Notation (JSON) as the transaction data in the distributed ledger and to cause the database to be updated with the database attributes.

28. A system as in claim 15, wherein the at least one processor further executes instructions to create a parsed and translated structured query language (SQL) query that accounts for the form that may be processed by the database on the distributed ledger for inclusion on the distributed ledger.