WHAT IS CLAIMED IS:

1. A method, comprising:

performing by a computer:

receiving a request to create a table in a non-relational data store, wherein the request specifies an identifier of the table and a primary key by which to index items stored in the table; and

in response to said receiving:

creating a scalable table in the non-relational data store, wherein the scalable table is configured to store a plurality of items each of which comprises a value for the primary key, and wherein the scalable table does not have a pre-determined size limit; and

programmatically performing at least one of sizing or partitioning the scalable table in response to receiving one or more requests to access the scalable table.

2. The method of claim 1,

wherein creating the scalable table comprises initiating performance of an asynchronous table creation workflow; and

wherein the method further comprises:

receiving a request to describe one or more tables maintained in the non-relational data store; and

in response to receiving the request to describe one or more tables, returning information about the one or more tables, wherein the information about the one or more tables comprises information indicating whether the scalable table has been successfully created by the asynchronous table creation workflow.

3. The method of claim 1, further comprising:

receiving one or more requests to store a plurality of items in the scalable table; and

in response to receiving the one or more requests:

determining whether the plurality of items can be stored in a single partition of the scalable table; and

in response to determining that the plurality of items cannot be stored in a single partition of the scalable table, performing said programmatically partitioning the scalable table.

4. The method of claim 1,

wherein the primary key comprises a single attribute that is designated as a hash key attribute;

wherein each of the items stored in the scalable table comprises a respective value for the hash key attribute; and

wherein programmatically partitioning the scalable table comprises dividing the items in the scalable table into two or more partitions dependent on a hash of their respective hash key attribute values.

5. The method of claim 1,

wherein the primary key comprises an attribute that is designated as a hash key attribute and another attribute designated as a range key attribute;

wherein each of the items stored in the scalable table comprises a respective value for the hash key attribute and a respective value for the range key attribute;

wherein the range key attribute value uniquely identifies a particular item from among all items in the scalable table having a same hash key attribute value, wherein the items having the same hash key are ordered according to their respective range key attribute values; and

wherein programmatically partitioning the scalable table comprises dividing items in the scalable table having a same hash key attribute value into two or more partitions dependent on their respective range key attribute values.

6. The method of claim 1, wherein said programmatically partitioning the scalable table comprises dividing the scalable table into two or more partitions, and storing each of the two or more partitions on different ones of the plurality of computing nodes.

7. The method of claim 1, wherein said programmatically partitioning the scalable table comprises moving or splitting a partition of the scalable table while one or more replicas of the partition continue to accept and process requests.

8. The method of claim 1, wherein said programmatically partitioning the scalable table is performed in response to receiving an increasing number of requests targeting items in the scalable table, items in a particular partition of the scalable table, or items stored on a same computing node as at least a subset of the items in the scalable table, or in response to receiving a

decreasing number of requests targeting items in the scalable table, items in a particular partition of the scalable table, or items stored on a same computing node as at least a subset of the items in the scalable table.

9. The method of claim 1,

wherein the request to create a table further comprises an indication of one or more user preferences, wherein the one or more user preferences comprise a preferred service request throughput level or a service request throughput level for which a guarantee is requested; and

wherein said programmatically performing at least one of sizing or partitioning the scalable table is performed in response to detecting that one or more of the user preferences is not being met.

10. A system, comprising:

a plurality of computing nodes, each comprising at least one processor and memory, wherein the plurality of computing nodes is configured to implement a data storage service;

wherein in response to receiving a service request to create a table on behalf of a storage service client, and wherein the service request specifies an identifier of the table and a primary key by which to index items stored in the table, the data storage service is configured to create a scalable table in a non-relational data store;

wherein the scalable table is configured to store a plurality of items each of which comprises a value for the primary key;

wherein the scalable table does not have a pre-determined size limit; and

wherein creating the scalable table comprises initiating performance of an asynchronous table creation workflow.

11. The system of claim 10, wherein the data storage service is further configured to programmatically perform at least one of sizing or partitioning the scalable table in response to receiving one or more service requests to store, retrieve, modify, or delete items in the scalable table.

12. The system of claim 10,

wherein in response to receiving one or more service requests to store a plurality of items in the scalable table, the data storage service is configured to:

determine whether the plurality of items can be stored in a single partition of the scalable table; and

in response to determining that the plurality of items cannot be stored in a single partition of the scalable table, performing said programmatically partitioning the scalable table.

13. The system of claim 10, wherein the data storage service is further configured to programmatically partitioning the scalable table is performed in response to receiving an increasing number of requests targeting items in the scalable table, items in a particular partition of the scalable table, or items stored on a same computing node as at least a subset of the items in the scalable table, or in response to receiving a decreasing number of requests targeting items in the scalable table, items in a particular partition of the scalable table, or items stored on a same computing node as at least a subset of the items in the scalable table.

14. The system of claim 10,

wherein the primary key comprises a single attribute that is designated as a hash key attribute and another attribute designated as a range key attribute;

wherein each of the items stored in the scalable table comprises a respective value for the hash key attribute and a respective value for the range key attribute; and

wherein the range key attribute value uniquely identifies a particular item from among all items in the scalable table having a same hash key attribute value, wherein the items having the same hash key are ordered according to their respective range key attribute values; and

wherein the data storage service is further configured to divide items in the scalable table having a same hash key attribute value into two or more partitions dependent on their respective range key attribute values.

15. The system of claim 10, wherein said creating the scalable table further comprises:

gathering information reflecting the status of one or more of the plurality of computing nodes; and

determining one or more computing nodes on which to create the scalable table dependent on the gathered information; and

creating a respective partition for storing at least a subset of the scalable table on each of the one or more computing nodes.

What is claimed is:

1. A system, comprising:

a plurality of computing nodes, each comprising at least one processor and memory, that collectively implement a data storage service;

wherein the plurality of computing nodes comprises:

a front end module that provides a web services interface through which service requests are received and that is configured to parse and dispatch service requests for processing;

an administrative component that is configured to allocate resources in the system, to monitor the status of the system, and to detect anomalies experienced in the system while service requests are being serviced; and

a plurality of storage nodes that collectively implement a non-relational data store;

wherein in response to the front end module receiving a service request to create a table on behalf of a storage service client, and wherein the service request specifies a table name and a primary key by which to partition and index items stored in the table:

the front end module is configured to dispatch the service request to one of the plurality of storage nodes;

in response to receiving the service request, the one of the plurality of storage nodes is configured to create a scalable table in the non-relational data store, wherein the scalable table is configured to store a plurality of items each of which comprises a value for the primary key, and wherein the scalable table does not have a pre-determined size limit; and

subsequent to the scalable table being created, the administrative component is configured to programmatically cause the scalable table to be sized or partitioned in response to detecting an anomaly in the system or in response to receiving one or more service requests to store, retrieve, modify, or delete items in the scalable table.

2. The system of claim 1,

wherein creating the scalable table comprises initiating performance of an asynchronous table creation workflow; and

wherein in response to receiving a service request to describe one or more tables maintained by the data storage service, the non-relational data store is configured to return information about the one or more tables, wherein the information about the one or more tables comprises information indicating whether the scalable table has been successfully created by the asynchronous table creation workflow.

3. The system of claim 1,

wherein in response to receiving one or more service requests to store a plurality of items in the scalable table, the administrative component is configured to:

determine whether the plurality of items can be stored in a single partition of the scalable table; and

in response to determining that the plurality of items cannot be stored in a single partition of the scalable table, performing said programmatically partitioning the scalable table.

4. The system of claim 1,

wherein the primary key comprises a single attribute that is designated as a hash key attribute; and

wherein partitioning the scalable table comprises dividing the items in the scalable table into two or more partitions dependent on a hash of their respective hash key attribute values.

5. The system of claim 1,

wherein the service request to create a table further comprises an indication of one or more user preferences, wherein the one or more user preferences comprise a preferred service request throughput level or a service request throughput level for which a guarantee is requested; and

wherein the administrative component is configured to programmatically cause the scalable table to be sized or partitioned in response to detecting that one or more of the user preferences is not being met.

6. The system of claim 1, wherein programmatically causing the scalable table to be partitioned is performed in response to the front end module receiving an increasing number of requests targeting items in the scalable table, items in a particular partition of the scalable table, or items stored on a same computing node as at least a subset of the items in the scalable table.

7. A method, comprising:

performing, by one or more computer processors:

receiving a request through a web services interface to create a table in a non-relational data store, wherein the request specifies an identifier of the table and a primary key by which to index items stored in the table;

in response to said receiving:

creating a scalable table in the non-relational data store, wherein the scalable table is configured to store a plurality of items each of which comprises a value for the primary key, and wherein the scalable table does not have a pre-determined size limit;

receiving one or more requests to store one or more items in the scalable table;

in response to said receiving one or more requests to store the plurality of items in the scalable table, determining whether the scalable table should be partitioned; and

programmatically performing partitioning the scalable table in response to determining that the scalable table should be partitioned.

8. The method of claim 7, wherein the one or more requests to access the scalable table comprise one or more requests to store, retrieve, modify, or delete items in the scalable table.

9. The method of claim 7, wherein creating the scalable table comprises:

gathering information reflecting the status of one or more of the plurality of computing nodes; and

determining one or more computing nodes on which to create the scalable table dependent on the gathered information.

10. The method of claim 7,

wherein creating the scalable table comprises initiating performance of an asynchronous table creation workflow; and

wherein the method further comprises:

receiving a request to describe one or more tables maintained in the non-relational data store; and

in response to receiving the request to describe one or more tables, returning information about the one or more tables, wherein the information about the one or more tables comprises information indicating whether the scalable table has been successfully created by the asynchronous table creation workflow.

11. The method of claim 7, wherein said determining that the scalable table should be partitioned comprises:

determining that the one or more items cannot be stored in a single partition of the scalable table.

12. The method of claim 7,

wherein the primary key comprises a single attribute that is designated as a hash key attribute;

wherein each of the items stored in the scalable table comprises a respective value for the hash key attribute; and

wherein programmatically partitioning the scalable table comprises dividing the items in the scalable table into two or more partitions dependent on a hash of their respective hash key attribute values.

13. The method of claim 7,

wherein the primary key comprises an attribute that is designated as a hash key attribute and another attribute designated as a range key attribute;

wherein each of the items stored in the scalable table comprises a respective value for the hash key attribute and a respective value for the range key attribute;

wherein the range key attribute value uniquely identifies a particular item from among all items in the scalable table having a same hash key attribute value, wherein the items having the same hash key are ordered according to their respective range key attribute values; and

wherein programmatically partitioning the scalable table comprises dividing items in the scalable table having a same hash key attribute value into two or more partitions dependent on their respective range key attribute values.

14. The method of claim 7, wherein said programmatically partitioning the scalable table comprises dividing the scalable table into two or more partitions, and storing each of the two or more partitions on different ones of the plurality of computing nodes.

15. The method of claim 7, wherein said programmatically partitioning the scalable table comprises moving or splitting a partition of the scalable table while one or more replicas of the partition continue to accept and process requests.

16. The method of claim 7, wherein said programmatically partitioning the scalable table is performed in response to receiving an increasing number of requests targeting items in the scalable table, items in a particular partition of the scalable table, or items stored on a same computing node as at least a subset of the items in the scalable table, or in response to receiving a decreasing number of requests targeting items in the scalable table, items in a particular partition of the scalable table, or items stored on a same computing node as at least a subset of the items in the scalable table.

17. The method of claim 7,

wherein the request to create a table further comprises an indication of one or more user preferences, wherein the one or more user preferences comprise a preferred service request throughput level or a service request throughput level for which a guarantee is requested

wherein said programmatically performing at least one of sizing or partitioning the scalable table is performed in response to detecting that one or more of the user preferences is not being met.

18. The method of claim 7, further comprising:

receiving a request to retrieve one or more items from the scalable table, wherein the request comprises an indication of a preferred consistency model; and

in response to receiving the request to retrieve the one or more items, returning the one or more items in a manner that is consistent with the preferred consistency model.

19. The method of claim 7, wherein said creating comprises:

generating metadata associated with the scalable table; and

storing the metadata in another scalable table in the non-relational data store.

20. A non-transitory, computer-readable storage medium storing program instructions that when executed on one or more computers cause the one or more computers to perform:

receiving a service request to create a table on behalf of a storage service client, wherein the service request specifies an identifier of the table and a primary key by which to index items stored in the table;

in response to said receiving:

creating a scalable table in a non-relational data store, wherein the scalable table is configured to store a plurality of items each of which comprises a value for the primary key, and wherein the scalable table does not have a pre-determined size limit; and

in response to receiving one or more service requests to store a plurality of items in the scalable table:

determining whether the plurality of items can be stored in a single partition of the scalable table; and

in response to determining that the plurality of items cannot be stored in a single partition of the scalable table, partitioning the scalable table.

21. The storage medium of claim 20, wherein when executed on the one or more computers, the program instructions further cause the one or more computers to perform:

subsequent to partitioning the scalable table, performing at least one of sizing or re-partitioning the scalable table in response to receiving one or more service requests to store, retrieve, modify, or delete items in the scalable table.

22. The storage medium of claim 21, wherein said re-partitioning the scalable table is performed in response to receiving an increasing number of requests targeting items in the scalable table, items in a particular partition of the scalable table, or items stored on a same

computing node as at least a subset of the items in the scalable table, or in response to receiving a decreasing number of requests targeting items in the scalable table, items in a particular partition of the scalable table, or items stored on a same computing node as at least a subset of the items in the scalable table.

23. The storage medium of claim 20, wherein creating the scalable table comprises:

gathering information reflecting the status of one or more of the plurality of computing nodes;

determining one or more computing nodes on which to create the scalable table dependent on the gathered information; and

creating a respective partition for storing at least a subset of the scalable table on each of the one or more computing nodes.

24. The storage medium of claim 20,

wherein creating the scalable table comprises initiating performance of an asynchronous table creation workflow; and

wherein in response to receiving a service request to describe one or more tables maintained in the non-relational data store, the program instructions further cause the one or more computers to perform returning information about the one or more tables, wherein the information about the one or more tables comprises information indicating whether the scalable table has been successfully created by the asynchronous table creation workflow.

25. The storage medium of claim 20,

wherein the primary key comprises a single attribute that is designated as a hash key attribute;

wherein each of the items stored in the scalable table comprises a respective value for the hash key attribute; and

wherein said partitioning the scalable table comprises dividing the items in the scalable table into two or more partitions dependent on a hash of their respective hash key attribute values.

26. The storage medium of claim 20, wherein said partitioning the scalable table comprises dividing the scalable table into two or more partitions, and storing each of the two or more partitions on different ones of the plurality of computing nodes.

27. The storage medium of claim 20,

wherein the primary key comprises an attribute that is designated as a hash key attribute and another attribute designated as a range key attribute;

wherein each of the items stored in the scalable table comprises a respective value for the hash key attribute and a respective value for the range key attribute;

wherein the range key attribute value uniquely identifies a particular item from among all items in the scalable table having a same hash key attribute value, wherein the items having the same hash key are ordered according to their respective range key attribute values; and

wherein partitioning the scalable table comprises dividing items in the scalable table having a same hash key attribute value into two or more partitions dependent on their respective range key attribute values.

28. A system, comprising:

a plurality of computing nodes, each comprising at least one processor and memory, wherein the plurality of computing nodes is configured to implement a data storage service;

wherein in response to receiving a service request through a web services interface to create a table on behalf of a storage service, and wherein the service request specifies an identifier of the table and a primary key by which to index items stored in the table, the data storage service is configured to create a scalable table in a non-relational data store;

wherein the scalable table is configured to store a plurality of items each of which comprises a value for the primary key;

wherein the scalable table does not have a pre-determined size limit;

wherein creating the scalable table comprises initiating performance of an asynchronous table creation workflow; and

wherein the data storage service is further configured to:

receive one or more requests to store a plurality of items in the scalable table;

in response to said receiving one or more requests, determine whether the scalable table should be partitioned; and

in response to determining that the scalable table should be partitioned, partition the scalable table.

29. The system of claim 28,

wherein in response to receiving a service request to describe one or more tables maintained by the data storage service, the data storage service is configured to return information about the one or more tables; and

wherein the information about the one or more tables comprises information indicating whether the scalable table has been successfully created by the asynchronous table creation workflow.

30. The system of claim 28, wherein the data storage service is further configured to programmatically perform at least one of sizing or partitioning the scalable table in response to receiving one or more service requests to store, retrieve, modify, or delete items in the scalable table.

31. The system of claim 28,

wherein to determine that the scalable table should be partitioned, the data storage service is configured to:

determine that the plurality of items cannot be stored in a single partition of the scalable table.

32. The system of claim 28, wherein the data storage service is further configured to programmatically partitioning the scalable table in response to receiving an increasing number of requests targeting items in the scalable table, items in a particular partition of the scalable table, or items stored on a same computing node as at least a subset of the items in the scalable table, or in response to receiving a decreasing number of requests targeting items in the scalable table, items in a particular partition of the scalable table, or items stored on a same computing node as at least a subset of the items in the scalable table.

33. The system of claim 28,

wherein the primary key comprises a single attribute that is designated as a hash key attribute and another attribute designated as a range key attribute;

wherein each of the items stored in the scalable table comprises a respective value for the hash key attribute and a respective value for the range key attribute; and

wherein the range key attribute value uniquely identifies a particular item from among all items in the scalable table having a same hash key attribute value, wherein the items having the same hash key are ordered according to their respective range key attribute values; and

wherein the data storage service is further configured to divide items in the scalable table having a same hash key attribute value into two or more partitions dependent on their respective range key attribute values.

34. The system of claim 28, wherein said creating the scalable table further comprises:

gathering information reflecting the status of one or more of the plurality of computing nodes; and

determining one or more computing nodes on which to create the scalable table dependent on the gathered information; and

creating a respective partition for storing at least a subset of the scalable table on each of the one or more computing nodes.

35. The system of claim 28,

wherein in response to receiving a request to retrieve one or more items from the scalable table, and wherein the request comprises an indication of a preferred consistency model, the data storage service is configured to return the one or more items in a manner that is consistent with the preferred consistency model.