

Database sharding vs partitioning

[closed]

Asked 10 years, 11 months ago Modified 6 months ago

Viewed 180k times



355



Closed. This question is [not about programming or software development](#). It is not currently accepting answers.

💡 This question does not appear to be about [a specific programming problem, a software algorithm, or software tools primarily used by programmers](#). If you believe the question would be on-topic on [another Stack Exchange site](#), you can leave a comment to explain where the question may be able to be answered.

Closed 4 months ago.

The community reviewed whether to reopen this question 4 months ago and left it closed:

Original close reason(s) were not resolved

[Improve this question](#)

I have been reading about scalable architectures recently. In that context, two words that keep on showing up with regards to databases are *sharding* and *partitioning*. I looked up descriptions but still ended up confused.

Could the experts at stackoverflow help me get the basics right?

- What is the difference between *sharding* and *partitioning* ?
- Is it true that '*all sharded databases are essentially partitioned (over different nodes), but all partitioned databases are not necessarily sharded*' ?

mysql

sharding

database-partitioning

Share

Improve this question

Follow

edited Jul 28, 2019 at 12:45



Pavel Ryzhov

5,132 ● 8 ● 53 ● 73

asked Dec 25, 2013 at 9:54



Amit Sharma

6,124 ● 6 ● 28 ● 34

2 digitalocean.com/community/tutorials/... this might help.

– mchawre Feb 16, 2020 at 12:58

9 Answers

Sorted by:

Highest score (default)



280

Partitioning is more a generic term for dividing data across tables or databases. Sharding is one specific type of partitioning, part of what is called horizontal partitioning.



Here you replicate the schema across (typically) multiple instances or servers, using some kind of logic or identifier to know which instance or server to look for the data. An identifier of this kind is often called a "Shard Key".



A common, key-less logic is to use the alphabet to divide the data. A-D is instance 1, E-G is instance 2 etc.

Customer data is well suited for this, but will be somewhat misrepresented in size across instances if the partitioning does not take in to account that some letters are more common than others.

Another common technique is to use a key-synchronization system or logic that ensures unique keys across the instances.

A well known example you can study is how Instagram solved their partitioning in the early days (see link below). They started out partitioned on very few servers, using Postgres to divide the data from the get-go. I believe it was several thousand logical shards on those few physical shards. Read their awesome writeup from 2012 here: [Instagram Engineering - Sharding & IDs](#)

See here as well: <http://www.quora.com/Whats-the-difference-between-sharding-and-partition>

Share Improve this answer

edited Mar 9, 2020 at 12:48

Follow

answered Dec 25, 2013 at 12:22



Canis

4,400 ● 1 ● 25 ● 29

34 Sharding is a *type of HP*. It is not HP. – [NoChance](#) Dec 27, 2013 at 5:58

4 Am I right in thinking horizontal partitioning just means split rows out of a table into several sub-tables (possibly within the same schema or database instance.) While sharding is to horizontally partition, putting the sub-tables into separate schemas within a single database, or into separate database instances on separate machines. Or not? – [Jonathan Hartley](#) Mar 13, 2014 at 20:06

1 Isn't it more accurate to say that sharding is actually >> dividing << a schema not replicating it? – [Adelin](#) Jan 11, 2023 at 14:13

2 @Adelin Actually I would say that is incorrect, as sharding relates to dividing the data across multiple nodes by rows, not columns (horizontal partitioning). If you partition your data vertically, you split the tables by columns. For example putting users with names A-N in DB1 and names O-Z in DB2, where the schema is the same, is sharding (horizontal partitioning), whilst putting usernames in DB1 and emails in DB2, linking them with the `user_id`, would be vertical partitioning. – [Canis](#) Jan 16, 2023 at 9:02

@Canis this is not true, sharding and partitioning are exactly the same thing. There is no difference. Perhaps in 2013 when you wrote the answer and *manual* sharding was the way to go but definitely not now. Partitioning is also splitting data onto multiple servers. – [Geek](#) Apr 20 at 22:34



I've been diving into this as well and although I'm by far the reference on the matter, there are few key facts that



A **partition** is a division of a logical database or its constituent elements into distinct independent parts.



Database **partitioning** is normally done for manageability, performance or availability reasons, as for load balancing.



[https://en.wikipedia.org/wiki/Partition_\(database\)](https://en.wikipedia.org/wiki/Partition_(database)).

Sharding is a type of partitioning, such as **Horizontal Partitioning** (HP)

There is also **Vertical Partitioning** (VP) whereby you split a table into smaller distinct parts. Normalization also involves this splitting of columns across tables, but vertical partitioning goes beyond that and partitions columns even when already normalized.

[https://en.wikipedia.org/wiki/Shard_\(database_architecture\)](https://en.wikipedia.org/wiki/Shard_(database_architecture)).

I really like Tony Baco's answer on Quora where he makes you think in terms of schema (rather than columns and rows). He states that...

"**Horizontal partitioning**", or sharding, is replicating [copying] the schema, and then dividing the data based on a shard key.

"**Vertical partitioning**" involves dividing up the schema (and the data goes along for the ride).

<https://www.quora.com/Whats-the-difference-between-sharding-DB-tables-and-partitioning-them>

Oracle's Database Partitioning Guide has some nice figures. I have copied a few excerpts from the article.

https://docs.oracle.com/cd/B28359_01/server.111/b32024/partition.htm

When to Partition a Table

Here are some suggestions for when to partition a table:

- Tables greater than 2 GB should always be considered as candidates for partitioning.
- Tables containing historical data, in which new data is added into the newest partition. A typical example is a historical table where only the current month's data is updatable and the other 11 months are read only.
- When the contents of a table need to be distributed across different types of storage devices.

Partition Pruning

Partition pruning is the simplest and also the most substantial means to improve performance using partitioning. Partition pruning can often improve query performance by several orders of magnitude. For example, suppose an application contains an Orders table containing a historical record of orders, and that this table has been partitioned by week. A query requesting

orders for a single week would only access a single partition of the Orders table. If the Orders table had 2 years of historical data, then this query would access one partition instead of 104 partitions. This query could potentially execute 100 times faster simply because of partition pruning.

Partitioning Strategies

- Range
- Hash
- List

You can read their text and visualize their images which explain everything pretty well.

And lastly, it is important to understand that databases are extremely resource intensive:

- CPU
- Disk
- I/O
- Memory

Many DBA's will partition on the same machine, where the partitions will share all the resources but provide an improvement in disk and I/O by splitting up the data and/or index.

While other strategies will employ a "shared nothing" architecture where the shards will reside on separate and

distinct computing units (nodes), having 100% of the CPU, disk, I/O and memory to itself. Providing it's own set of advantages and complexities.

https://en.wikipedia.org/wiki/Shared_nothing_architecture

Share Improve this answer

edited Sep 9, 2016 at 13:58

Follow

answered Sep 9, 2016 at 13:53



Jeach

9,002 ● 7 ● 48 ● 60

""Horizontal partitioning", or sharding, is replicating [copying] the schema, and then dividing the data based on a shard key." - this is tautological. – [8bitjunkie](#) Mar 5, 2018 at 15:01

- 1 So there is a mirror, and it is fragmented, hence the etymology. – [mckenzm](#) Jun 11, 2020 at 1:44
-

I wish people would stop conflating Normalization with partitioning/sharding - it just confuses things and obscures the point of normalisation. The resulting physical and logical storage of the data differs, they are done for different reasons and they have different consequences. Add to which Normalization should almost always be performed/be a part of database design whereas partitioning/sharding are only relevant as options in relation to performance and availability at scale. – [Bob](#) Apr 10, 2023 at 11:30



Looks like this answers both your questions:

61



Horizontal partitioning splits one or more tables by row, usually within a single instance of a schema and a database server. It may offer an advantage by reducing index size (and thus search effort) provided that there is some obvious, robust, implicit way to identify in which table a particular row will be found, without first needing to search the index, e.g., the classic example of the 'CustomersEast' and 'CustomersWest' tables, where their zip code already indicates where they will be found.

Sharding goes beyond this: it partitions the problematic table(s) in the same way, but it does this across potentially multiple instances of the schema. The obvious advantage would be that search load for the large partitioned table can now be split across multiple servers (logical or physical), not just multiple indexes on the same logical server.

Source: [Wiki-Shard](#).

Sharding is the process of storing data records across multiple machines and is MongoDB's approach to meeting the demands of data growth. As the size of the data increases, a

single machine may not be sufficient to store the data nor provide an acceptable read and write throughput. Sharding solves the problem with horizontal scaling. With sharding, you add more machines to support data growth and the demands of read and write operations.

Source: [MongoDB](#).

Share Improve this answer

edited Dec 27, 2013 at 5:55

Follow

answered Dec 25, 2013 at 12:34



NoChance

5,752 ● 4 ● 36 ● 48



34

Consider a table in database with 1 million rows and 100 columns. In **Partitioning** you can divide the table into 2 or more tables having properties like:



1. 0.4 million rows (table1), 0.6 million rows (table2)
2. 1 million rows & 60 columns (table1), 1 million rows & 40 columns (table2)



There could be multiple cases like that.

This is general partitioning.

But **Sharding** refers to 1st case only where we are dividing the data on the basis of rows. If we are dividing

the table into multiple tables, we need to maintain multiple similar copies of schemas as now we have multiple tables.

Share Improve this answer

edited May 15 at 15:30

Follow



Prerak Jain

448 ● 1 ● 7 ● 15

answered Aug 22, 2018 at 17:52



vikas bhandari

488 ● 4 ● 6

2 This makes it really clear, thanks! – [Chen Ni](#) Mar 25, 2022 at 7:21

So the bottom line is sharding is row-wise split of table in multiple instance or nodes. whereas partitioning could be row/column-wise split in single instance/node. Did I get correct? – [Md. Saiful Islam](#) Oct 16 at 8:22



26



When talking about partitioning please do not use term replicate or replication. Replication is a different concept and out of scope of this page. When we talk about partitioning then better word is divide and when we talk about sharding then better word is distribute. In partition (normally and in common understanding not always) the rows of large data set table are divided into two or more disjoint (not sharing any row) groups. You can call each group a partition. These groups or all the partitions remain under the control of once RDBMS instance and this is all logical. The base of each group can be a hash or range or etc. If you have ten years data in a table then

you can store each of the year's data in a separate partition and this can be achieved by setting partition boundaries on the basis of a non-null column `CREATE_DATE`. Once you query the db then if you specify a create date between 01-01-1999 and 31-12-2000 then only two partitions will be hit and it will be sequential. I did similar on DB for billion + records and sql time came to 50 millis from 30 seconds using indices etc all. Sharding is that you host each partition on a different node/machine. Now searching inside the partitions/shards can happen in parallel.

Share Improve this answer

Follow

answered Mar 17, 2020 at 7:42



imran

269 ● 3 ● 4

this makes total sense – [Paramvir Singh Karwal](#) Apr 29, 2021 at 10:42



10



Sharding in a special case of **horizontal partitioning**, when partitions spans across multiple database instances. If a database is sharded, it means that it's partitioned by definition.

Share Improve this answer

Follow

answered Jul 28, 2019 at 7:29



Pavel Ryzhov

5,132 ● 8 ● 53 ● 73





Horizontal partition when moved to another *database instance** becomes a **database shard**.

5



Database instance can be on the same machine or on another machine.



Share Improve this answer

answered Jun 21, 2020 at 9:16



Follow



abbas

7,071 ● 3 ● 43 ● 38

It's amazing how all the other answers here struggle to explain what this comment explains in one concise sentence.

– [niryo](#) May 15 at 20:56



Partitioning and Sharding are the same concept. There is no difference between them. From "Designing Data-Intensive Applications - Martin Kleppmann"

0



What we call a partition here is called a shard in MongoDB, Elasticsearch, and SolrCloud; it's known as a region in HBase, a tablet in Bigtable, a vnode in Cassandra and Riak, and a vBucket in Couchbase. However, partitioning is the most established term, so we'll stick with that.

There was a time when mostly RDBMS DBs didn't have concepts of partitioning(Sharding) so the option was to split the DB manually onto different servers. And then application logic use to figure out which partition (Shard)

the application needs to connect. But may be because every Database vendor adopted different terminology led to this confusion but these are same concepts.

Share Improve this answer

answered Apr 20 at 22:45

Follow



Geek

653 ● 1 ● 8 ● 20

Partitioning and sharding are not the same thing. As almost everybody else has correctly noted, sharding is a specific form of partitioning. – [Nicholas Adams](#) Apr 26 at 12:29



0



In short:

1. Sharding to distribute to different machines/server
2. Partitions can be distributed within the same machine or other machines



Share Improve this answer

edited May 23 at 12:19



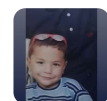
Follow



Alez

2,569 ● 10 ● 26 ● 36

answered May 17 at 17:13



Ankit Kalauni

15 ● 5
