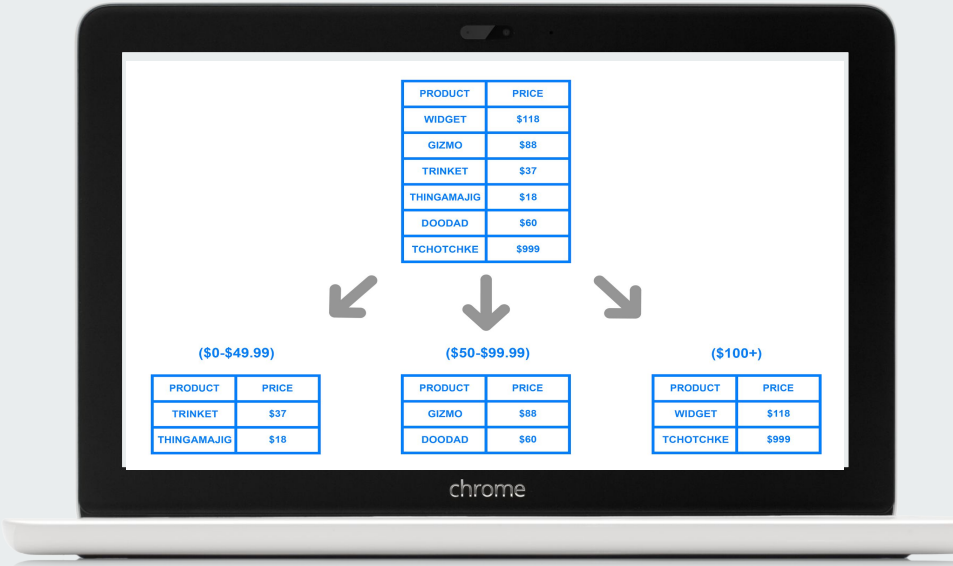
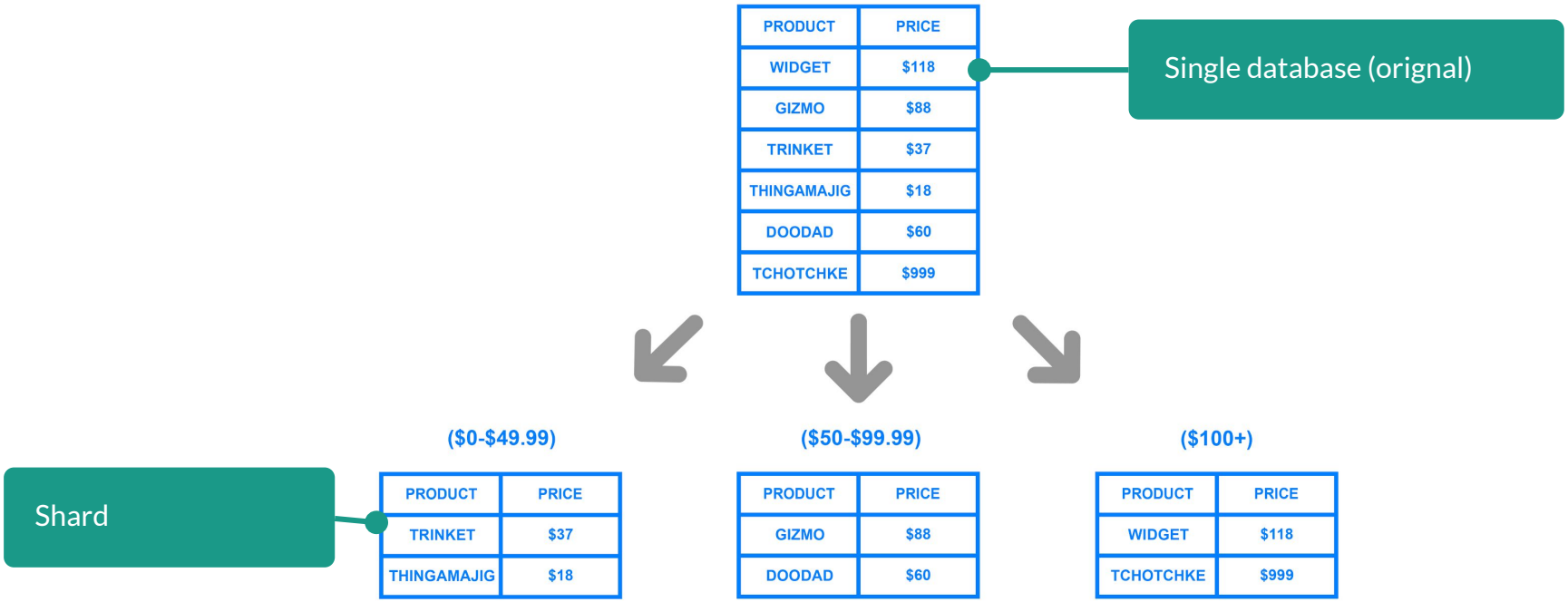
Database Sharding

Team 11
Jasshan 2018101014
Manish 2019201052
Rutwik 20171125



What is Database Sharding ?

- Sharding is partitioning the data from a single data source to multiple partitions in which the structure of each partition is identical to others.
- Individual partition is also referred to as a shard.
- Shards can be placed in the same server or different servers.


PRODUCT
PRICE
WIDGET
$118
GIZMO
$88
TRINKET
$37
THINGAMAJIG
$18
DOODAD
$60
TCHOTCHKE
$999
Single database (orignal)
($0-$49.99)
($50-$99.99)
($100+)
Shard
PRODUCT
PRICE
TRINKET
$37
THINGAMAJIG
$18
PRODUCT
PRICE
GIZMO
$88
DOODAD
$60
PRODUCT
PRICE
WIDGET
$118
TCHOTCHKE
$999

Why do we need Database Sharding ?

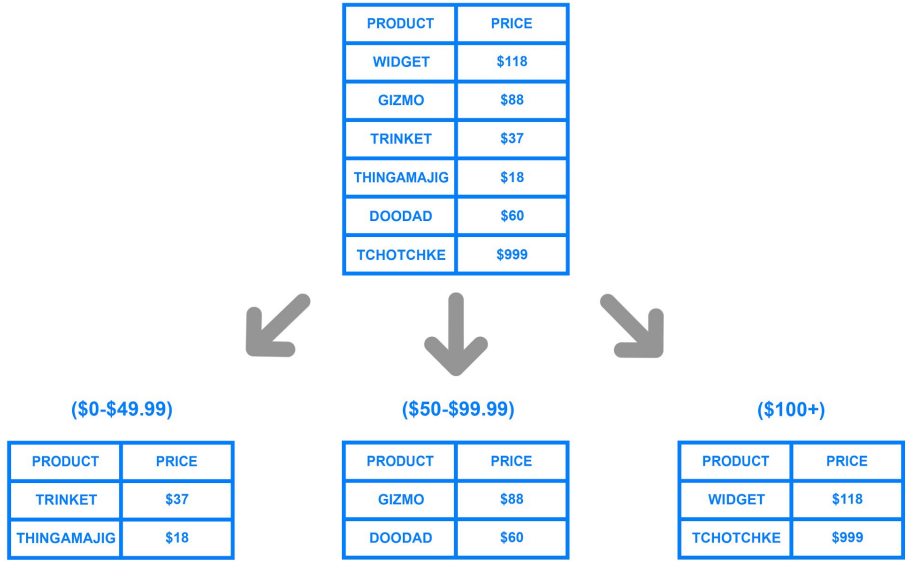
- If we read everything from one table, response time might increase with an increase in load and the database can also go down.
- Sharding splits the traffic from a single table to multiple tables

How is Sharding done ?

- Range based partitioning
  - In this case, data is sharded (partitioned) based on the range of a key. The data within the same range falls in the same partition.
- Hash partitioning
  - In this case, we pick up a key and pass to a hash function and get the partition, i.e. the hash function can be considered a map from key to partition.

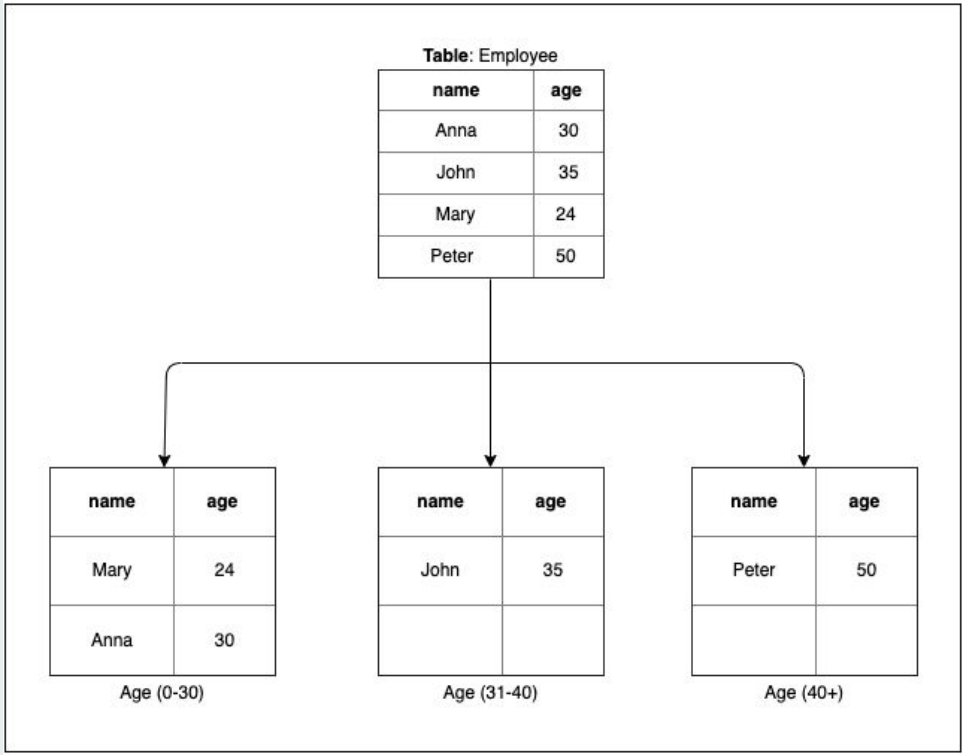
Range based partitioning

Example 1



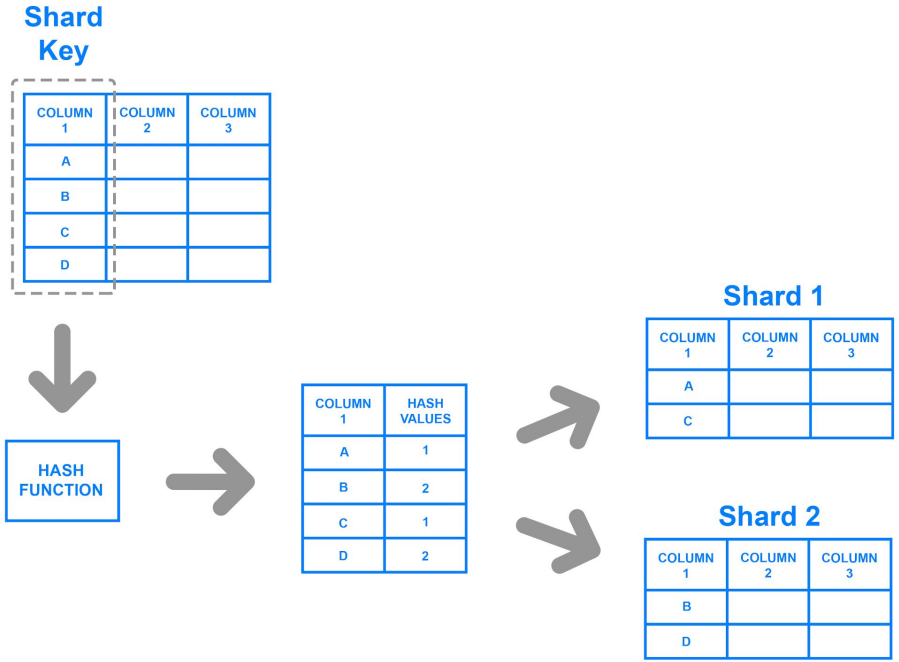
Range based partitioning

Example 2



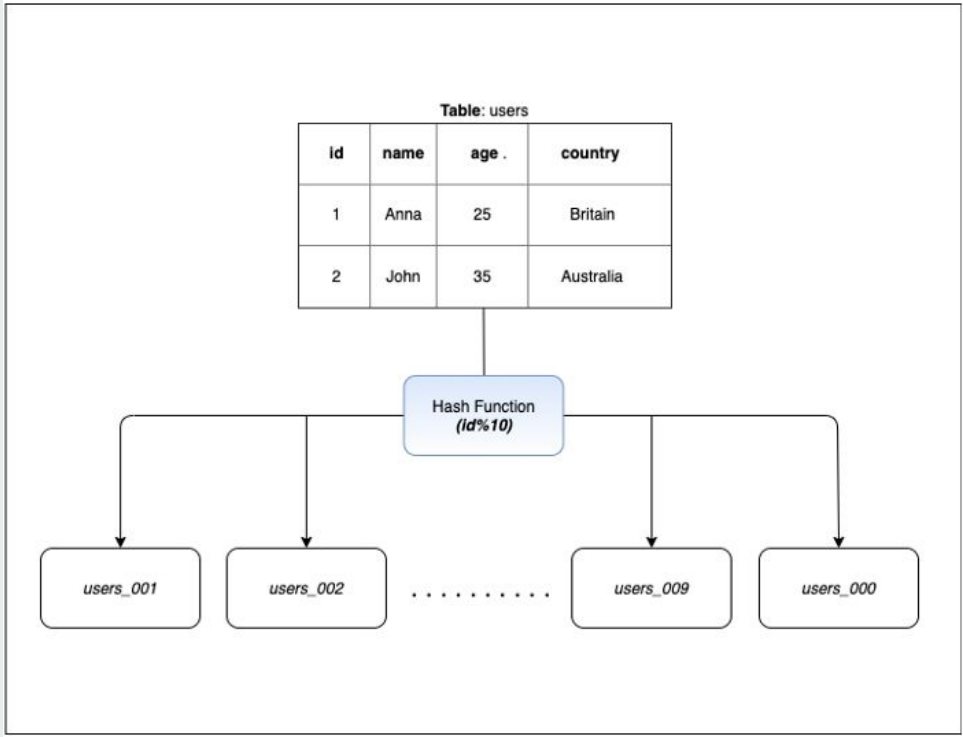
Hash partitioning

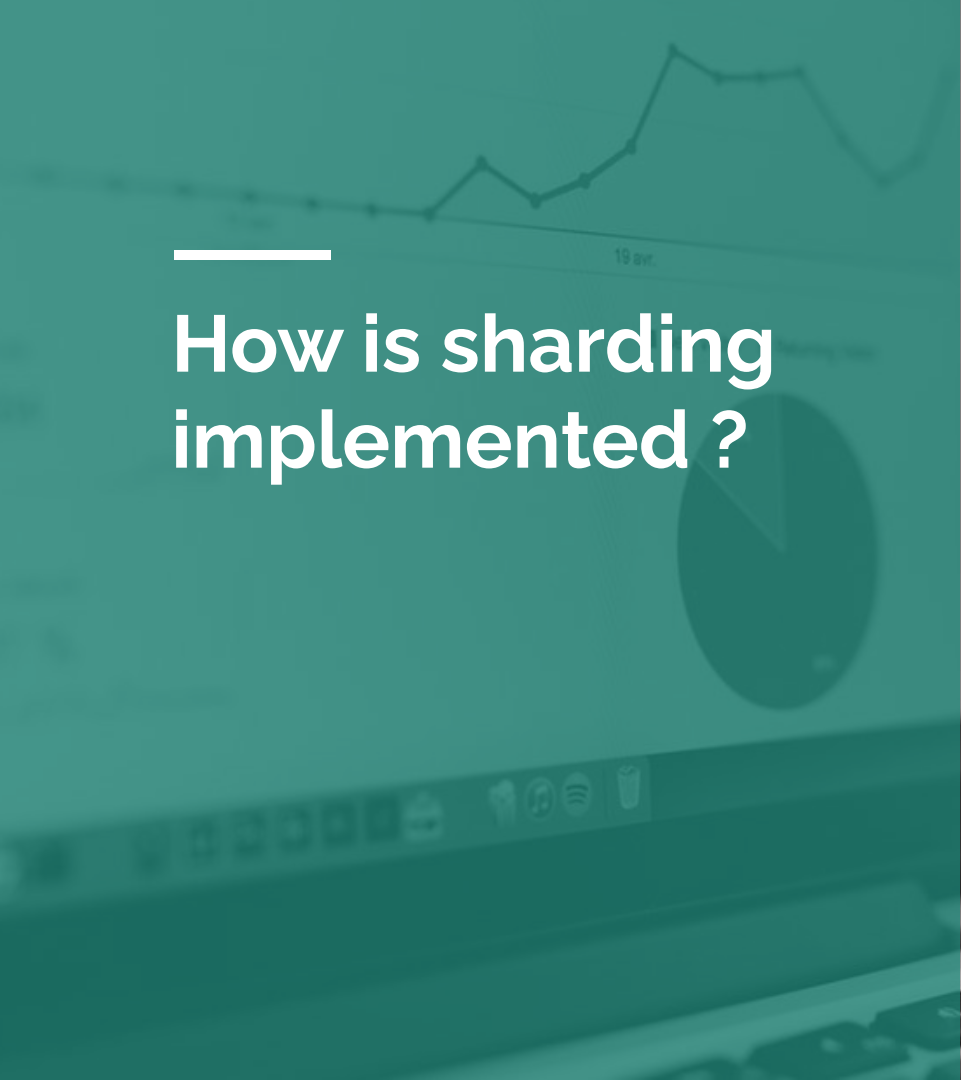
Example 1 (hash table)



Hash partitioning

Example 2 (hash function)



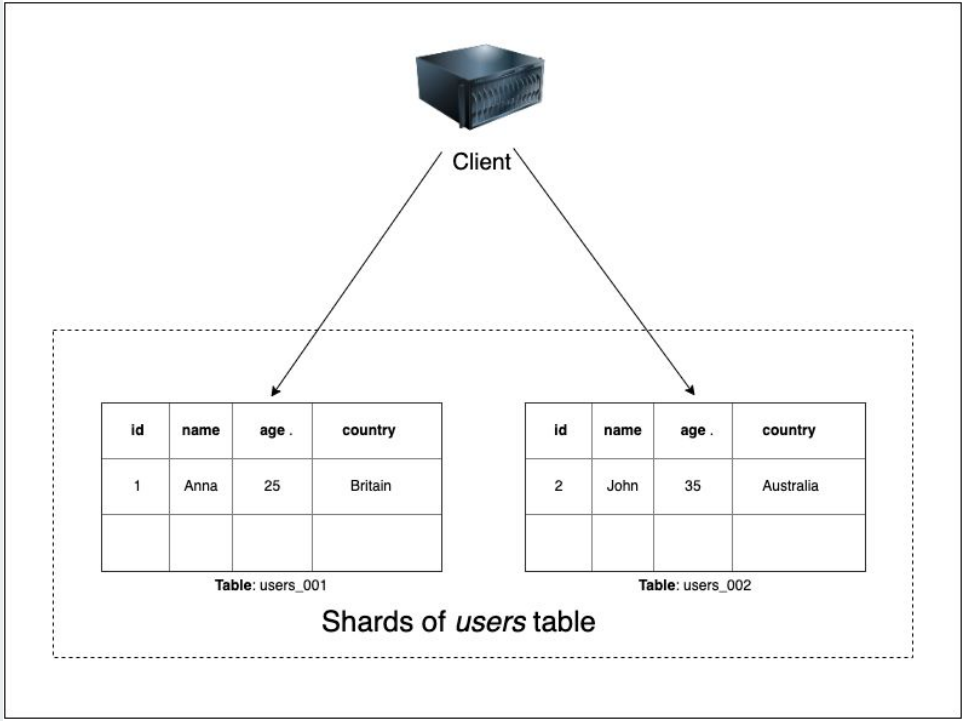


How is sharding implemented ?

- Client side partitioning
- Proxy assisted partitioning

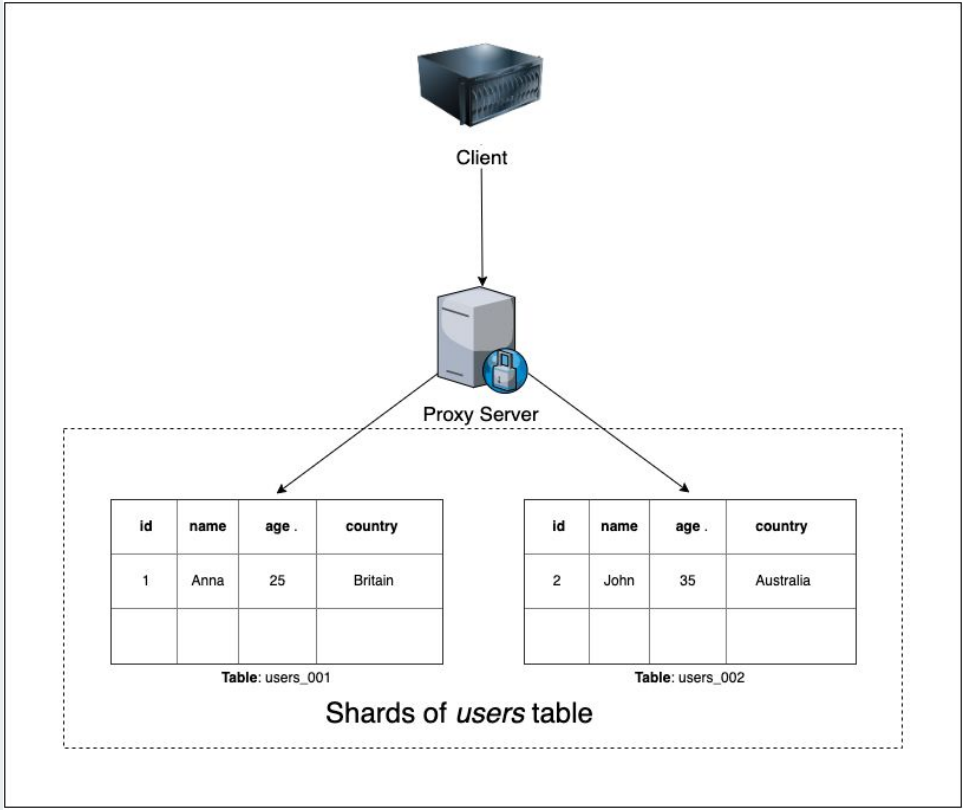
Client side partitioning

The clients know how data is partitioned and directly select the partitions for reading and writing the data.



Proxy assisted partitioning

In proxy-assisted partitioning, instead of making a direct call to a shard, clients make a request to a proxy server. The proxy server forwards this request to the right shard according to the schema of sharding.



How is sharding different from horizontal partitioning ?

- ➔ Horizontal partitioning splits one or more tables by row, usually within a *single* instance of a schema and a database server. It will over an advantage by reducing the search effort
- ➔ 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. Thus the data can now be split across multiple servers (logical or physical)

Conclusion

- ➔ Sharding can be a great solution for a database with a large amount of data. It helps to split the load from a single node to multiple nodes. But, it adds a lot of complexity to the application.
- ➔ Sharding can be necessary in some cases, but one need to exhaust other options like adding caching or migrating to a larger server before adding sharding as the time to create and the maintenance costs might outweigh the benefits of sharding.