# SNS COLLEGE OF ENGINEERING

Kurumbapalayam(Po), Coimbatore – 641 107
Accredited by NAAC-UGC with 'A' Grade
Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

## Department of Artificial Intelligence and Data Science

### Course Name – 19AD501 Big Data Analytics

### III Year / V Semester

### Unit 5 – Big Data Database

### Topic – Sharding

# Sharding

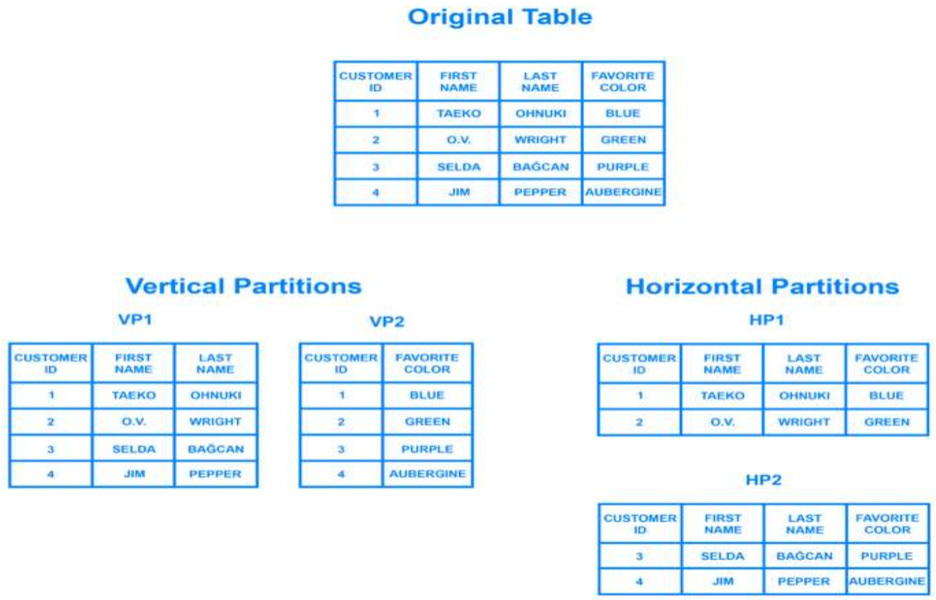
- Sharding is a database architecture pattern related to horizontal partitioning or horizontal scaling.
- It divides large datasets and distributed over multiple servers or shards. Each shard is independent database and collectively they would constitute a logical database.
- Database sharding is a type of horizontal partitioning that splits large databases into smaller components, which are faster and easier to manage.
- A shard is an individual partition that exists on separate database server instance to spread load.
- Auto sharding or data sharding is needed when a dataset is too big to be stored in a single database.

# Sharding

Why Sharding?

- As both the database size and number of transactions increase, so does the response time for querying the database. Costs associated with maintaining a huge database can also skyrocket due to the number and quality of computers you need to manage your workload.

- Data shards, on the other hand, have fewer hardware and software requirements and can be managed on less expensive servers.



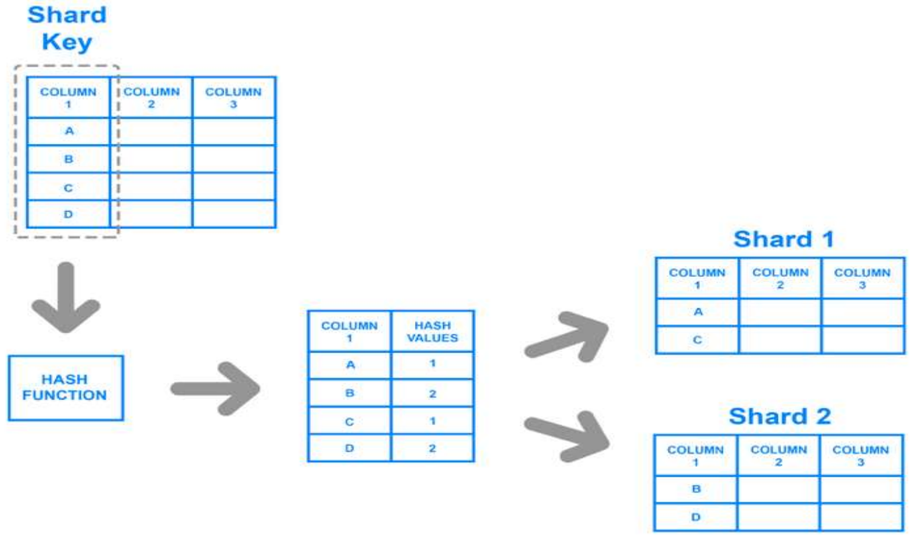
# Sharding

Sharding Architectures

Key Based Sharding

Key based sharding, also known as hash based sharding, involves using a value taken from newly written data — such as a customer's ID number, a client application's IP address, a ZIP code, etc. — and plugging it into a hash function to determine which shard the data should go to.

A hash function is a function that takes as input a piece of data (for example, a customer email) and outputs a discrete value, known as a hash value



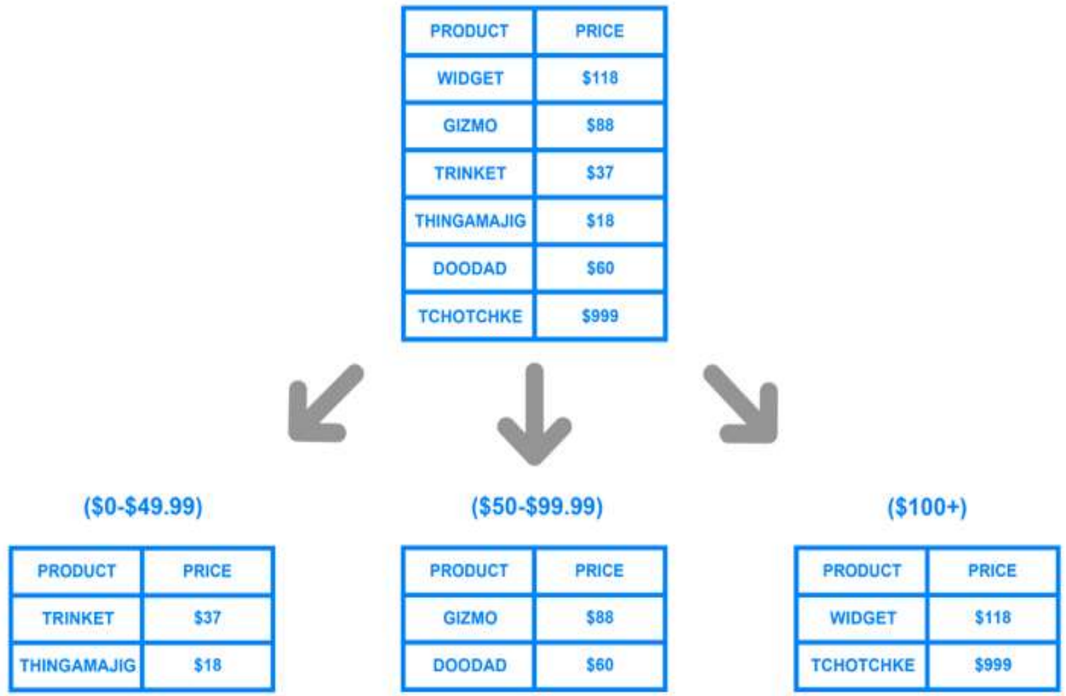
# Sharding

Range Based Sharding

Range based sharding involves sharding data based on ranges of a given value.

The main benefit of range based sharding is that it's relatively simple to implement.

Every shard holds a different set of data but they all have an identical schema as one another, as well as the original database.



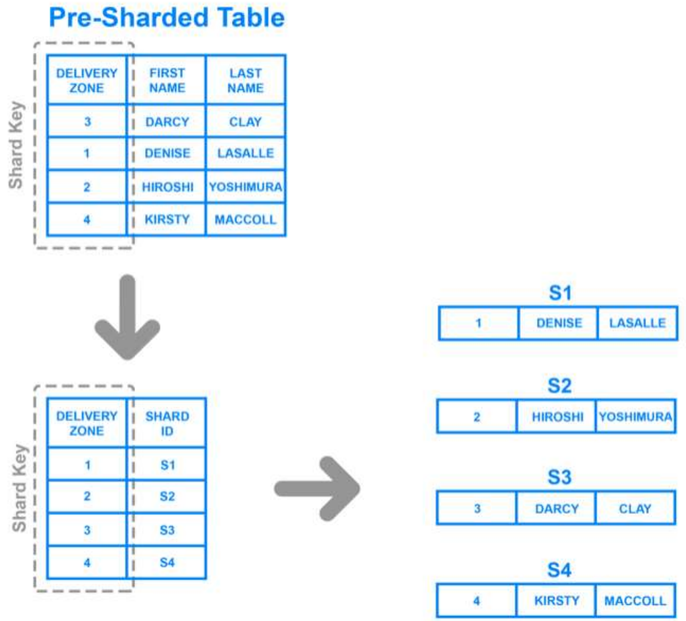
# Sharding

**Directory Based Sharding**

To implement directory based sharding, one must create and maintain a lookup table that uses a shard key to keep track of which shard holds which data.

The main appeal of directory based sharding is its flexibility. Range based sharding architectures limit you to specifying ranges of values, while key based ones limit you to using a fixed hash function which, as mentioned previously, can be exceedingly difficult to change later on.

Directory based sharding, on the other hand, allows you to use whatever system or algorithm you want to assign data entries to shards, and it's relatively easy to dynamically add shards using this approach.



# THANK YOU