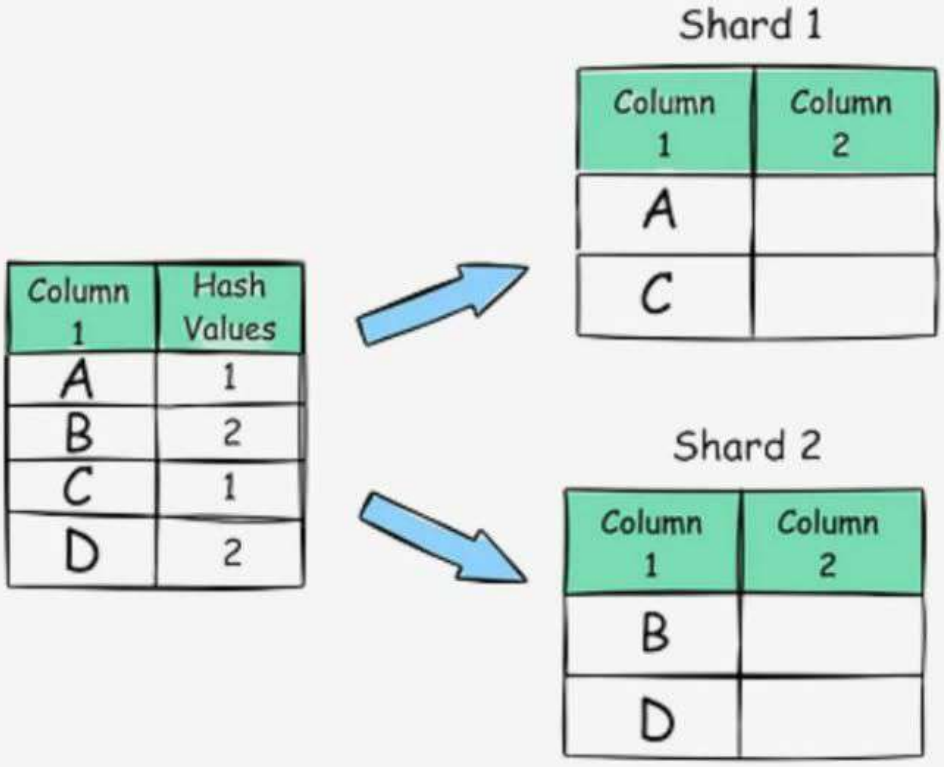
Database Sharding

swipe

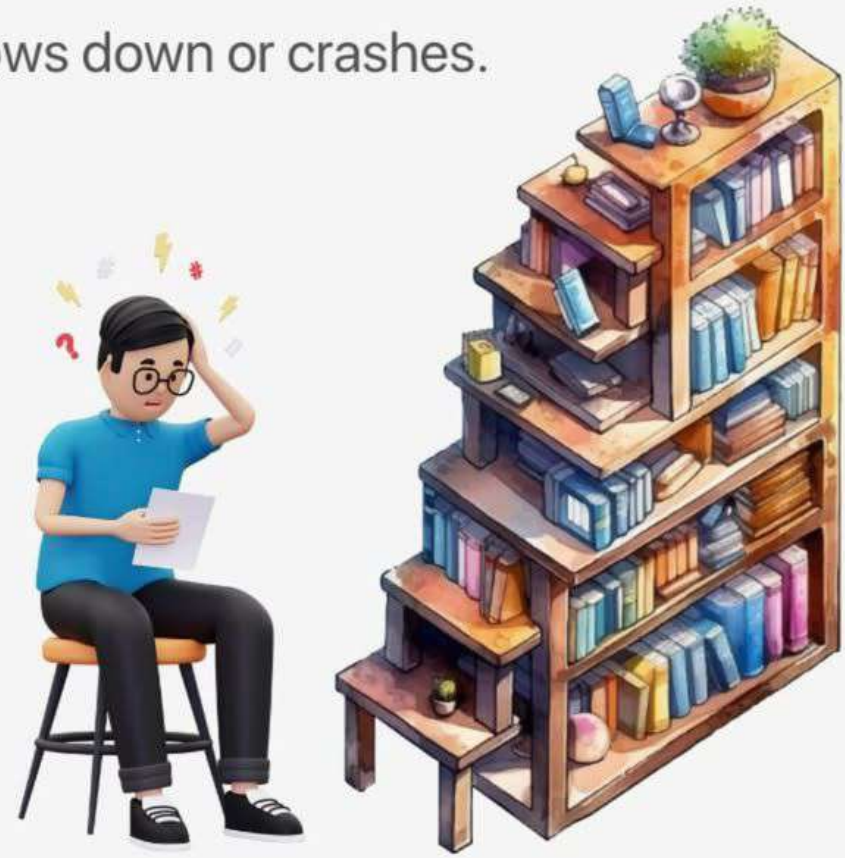


Defining the problem

Imagine a **huge library** with **all books** on **one shelf**. Everyone crowds there, making it **slow** to find books and stressing the librarian (your database).

Problems:

- Too many requests overwhelm the database.
- Searching takes too long (all books on same shelf).
- The system slows down or crashes.



The Solution: Sharding

Split the books across shelves (servers):

- A-M on one shelf (Server 1).
- N-Z on another shelf (Server 2).

Now:

- Searching is faster (smaller shelves).
- Fewer people crowd each shelf.
- The librarian (database) stays efficient.

Logical vs Physical Sharding

Logical shards are pieces of your data divided horizontally using a specific strategy, such as grouping by user ID, region or maybe price.

These shards are stored **on the same database instance** or server.

PRODUCT	PRICE
WIDGET	$118
GIZMO	$88
TRINKET	$37
THINGAMAJIG	$18
DOODAD	$60
TCHOTCHKE	$999

($0-$49.99)

PRODUCT	PRICE
TRINKET	$37
THINGAMAJIG	$18

($50-$99.99)

PRODUCT	PRICE
GIZMO	$88
DOODAD	$60

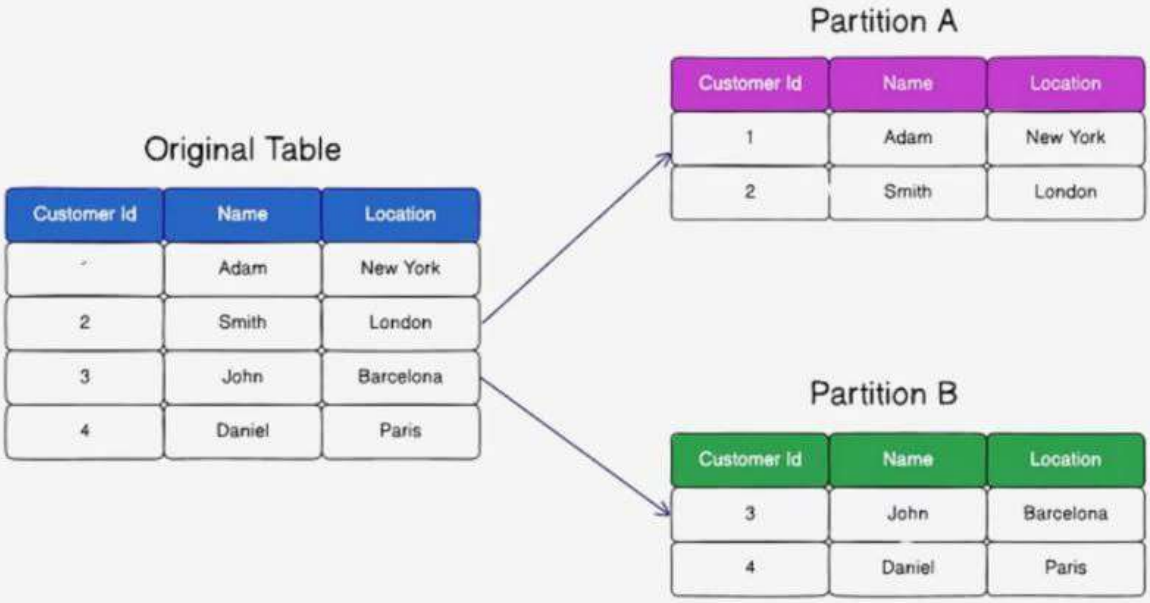
($100+)

PRODUCT	PRICE
WIDGET	$118
TCHOTCHKE	$999

What is Sharding?

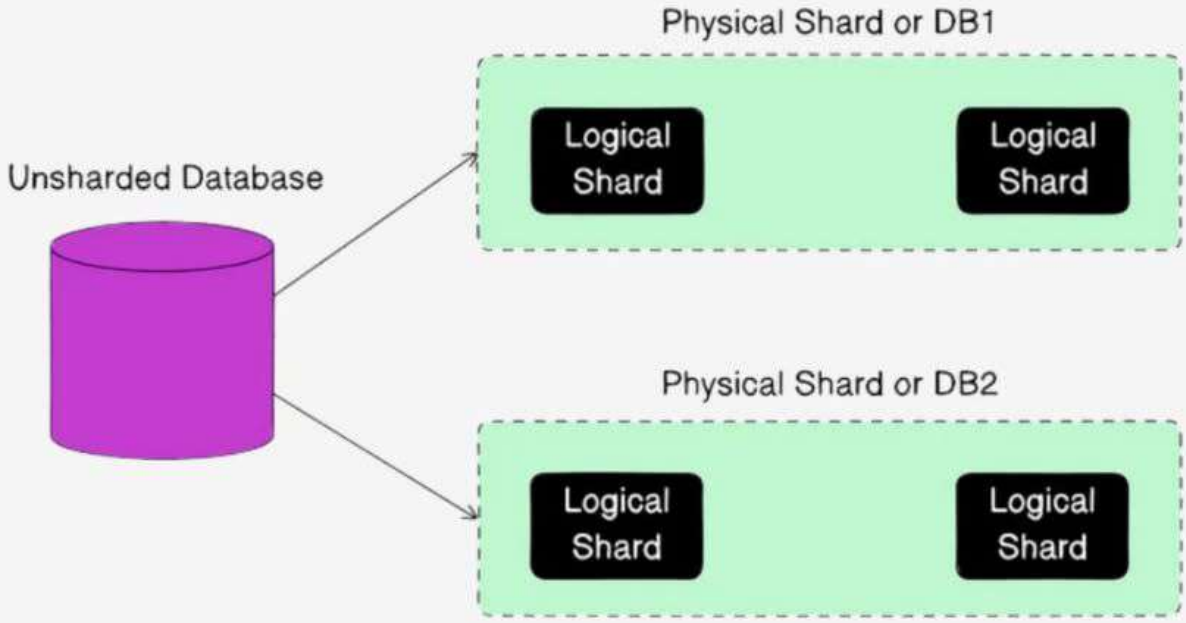
Sharding is a strategy to **horizontally scale** the database.

It evolves out of horizontal partitioning in which you separate the rows of one table into multiple different tables, known as **partitions**



Physical Sharding:

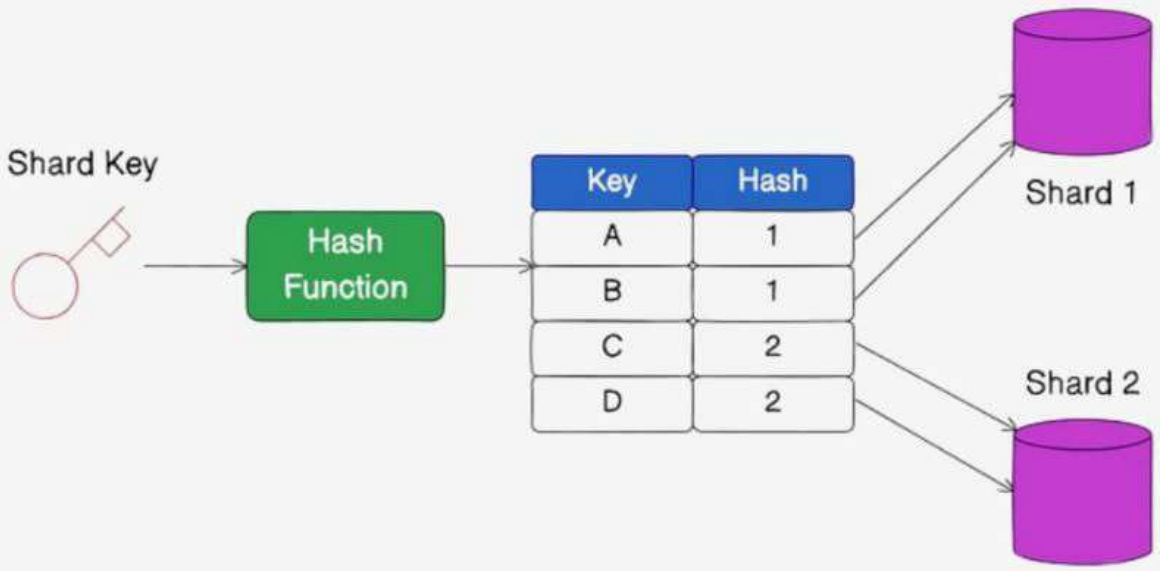
On the contrary, in **physical sharding** each shard is **hosted on a different node** or server instance. However, a physical shard can contain one or more logical shards.



👉Remember that the goal of sharding is to help your application deal with fewer data for a given request.

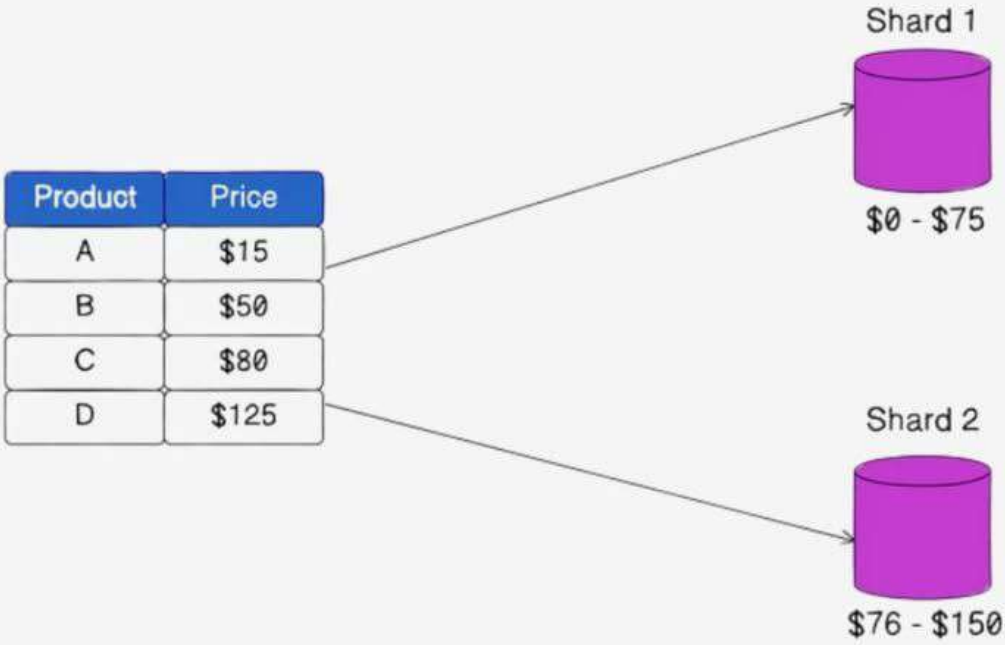
Sharding Strategies

Key-Based Sharding: partitions the data based on a **hash function** applied to one or more columns in the table. That's why this type of sharding is also known as **hash-based sharding**.



Sharding Strategies

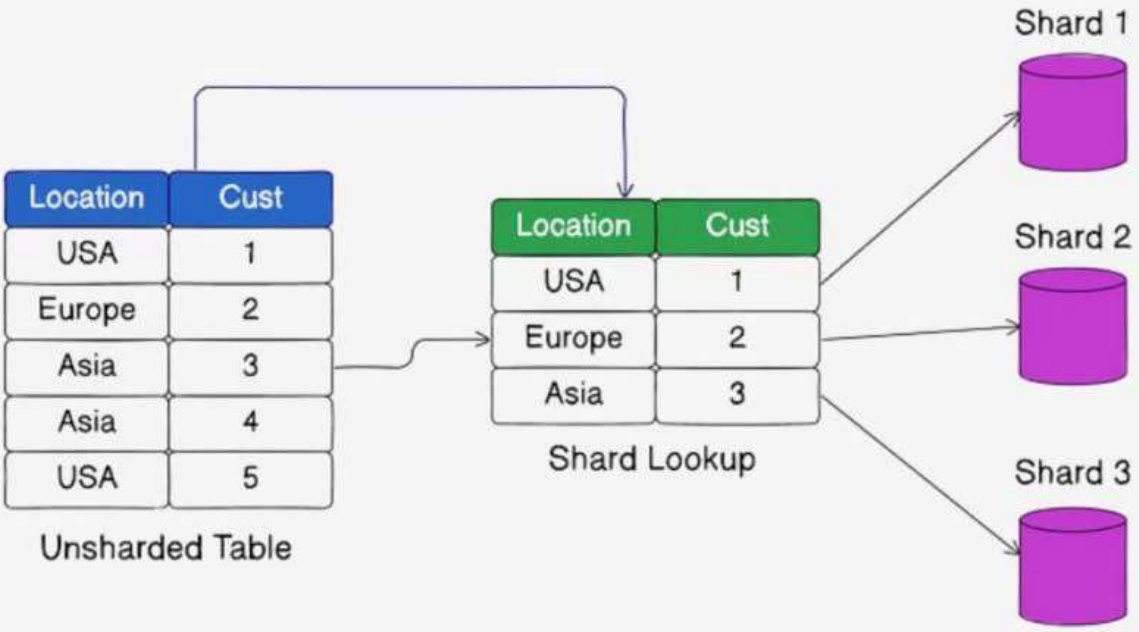
Range-Based Sharding: partitions the data based on a **specific range of value**s in a particular column.



Sharding Strategies

Directory-Based Sharding: you perform sharding based on a **lookup table**.

Like a directory (similar to an address book) that holds the relation between the data and the specific shard where you can find it.



Benefits of Sharding

- **Improved Performance:** Smaller data sets make queries faster.
- **Scalability**: Easy to add more servers as data grows.
- **Reduced Load**: Distributes traffic across multiple shards.
- **Fault Tolerance**: Issues in one shard don't affect others.
- **Cost Efficiency**: Use smaller, less expensive servers.