Mastering Database Partitioning and Sharding

educative

Database Partitioning

Divides a large dataset into smaller, more manageable pieces (sub-tables) within the same database instance to optimize performance.

User Table

User ID	Name	Age	Location
1001	Alice	17	America
1002	Bob	23	Europe
1003	Charlie	31	America
1004	David	15	Asia
1005	Eve	20	Europe
1006	Frank	40	Asia
1007	Grace	36	Asia
1008	Heidi	29	Europe

Note: We'll use the user table as a reference to define different types of partitioning and sharding.

Horizontal Partitioning refers to dividing data into smaller sub-tables horizontally, keeping the schema as it is.

Partition 1

User ID	Name	Age	Location
1001	Alice	17	America
1002	Bob	23	Europe
1003	Charlie	31	America
1004	David	15	Asia

Partition 2

User ID	Name	Age	Location
1005	Eve	20	Europe
1006	Frank	40	Asia
1007	Grank	36	Asia
1008	Heidi	29	Europe

Vertical Partitioning refers to changing the table schema by vertically dividing or partitioning the data.

Partition 1

User ID	Name
1001	Alice
1002	Bob
1003	Charlie
1004	David
1005	Eve
1006	Frank
1007	Grace
1008	Heidi

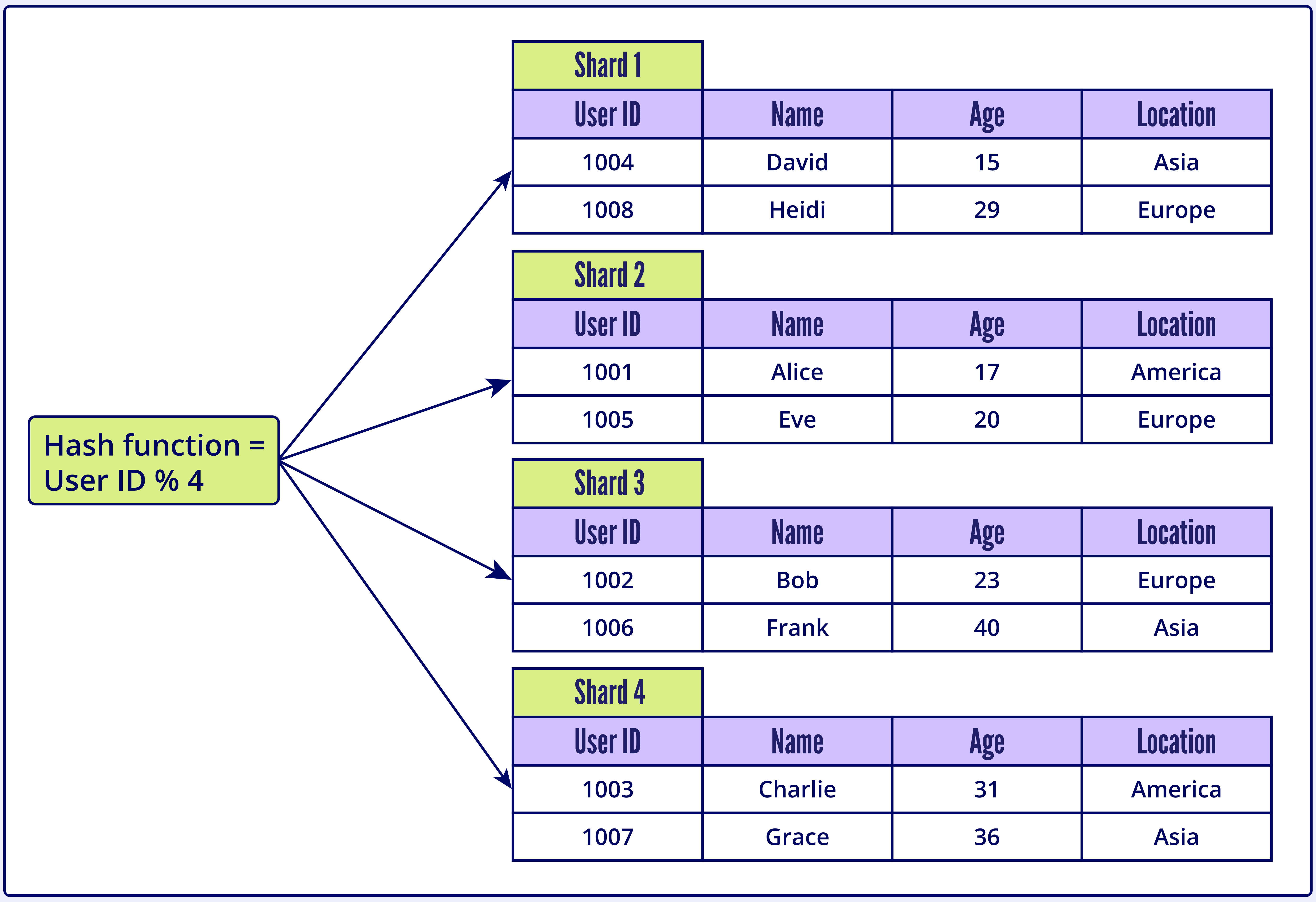
Partition 2

User ID	Location
1001	America
1002	Europe
1003	America
1004	Asia
1005	Europe
1006	Asia
1007	Asia
1008	Europe

Database Sharding

A type of database partitioning, involves splitting a database into smaller pieces (shards) and distributing them across multiple servers to improve scalability and performance.

Hash Sharding is distributing data across shards using a hash function to divide and store it evenly.



Shard 1

User ID	Name	Age	Location
1004	David	15	Asia
1008	Heidi	29	Europe

Shard 2

User ID	Name	Age	Location
1001	Alice	17	America
1005	Eve	20	Europe

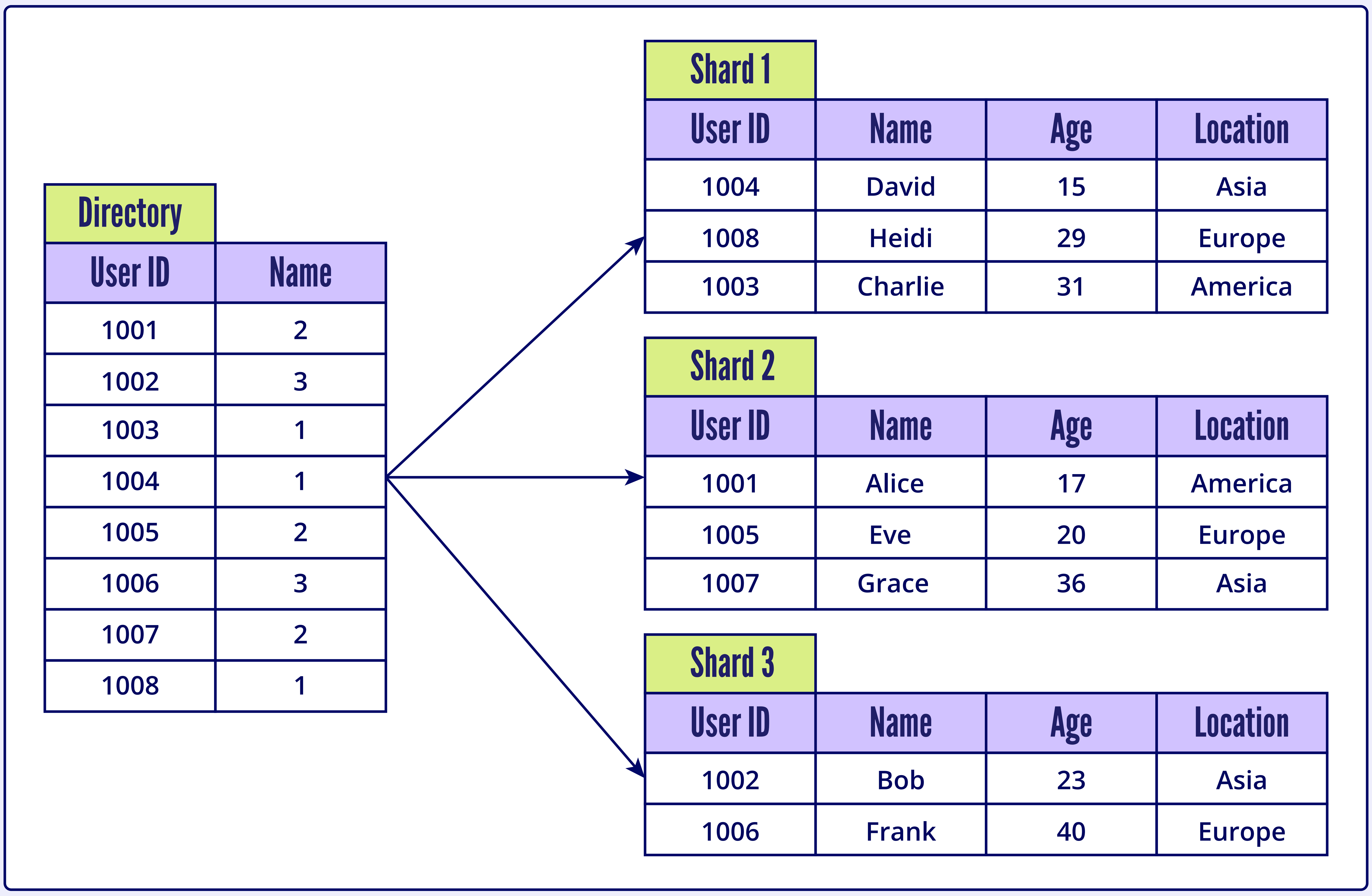
Shard 3

User ID	Name	Age	Location
1002	Bob	23	Europe
1006	Frank	40	Asia

Shard 4

User ID	Name	Age	Location
1003	Charlie	31	America
1007	Grace	36	Asia

Directory-Based Sharding uses a central directory (or lookup table) to map each data item to its corresponding shard.



Directory

User ID	Name
1001	2
1002	3
1003	1
1004	1
1005	2
1006	3
1007	2
1008	1

Shard 1

User ID	Name	Age	Location
1004	David	15	Asia
1008	Heidi	29	Europe
1003	Charlie	31	America

Shard 2

User ID	Name	Age	Location
1001	Alice	17	America
1005	Eve	20	Europe
1007	Grace	36	Asia

Shard 3

User ID	Name	Age	Location
1002	Bob	23	Asia
1006	Frank	40	Europe

Geographic Sharding distributes data across multiple shards based on the physical location or region of the data's origin.

America

User ID	Name	Age	Location
1001	Alice	17	America
1003	Charlie	31	America

Asia

User ID	Name	Age	Location
1004	David	15	Asia
1006	Frank	40	Asia
1007	Grace	36	Asia

Europe

User ID	Name	Age	Location
1002	Bob	23	Europe
1005	Eve	20	Europe
1008	Heidi	29	Europe

Range-Based Sharding divides data into shards based on predefined ranges of a key attribute, such as date, age, or user ID.

Age < 20

User ID	Name	Age	Location
1001	Alice	17	America
1003	David	15	America

20 ≥ Age < 30

User ID	Name	Age	Location
1002	Bob	23	Europe
1005	Eve	20	Europe
1008	Heidi	29	Europe

Age ≥ 30

User ID	Name	Age	Location
1003	Charlie	31	America
1006	Frank	40	Asia
1007	Grace	36	Asia

Entity-Based Sharding divides the database into shards based on specific entities or objects by grouping related data according to their type, such as users, products, or orders.

User Shard

User ID	Name	Location	Email
1001	Alice	America	alice@example.com
1002	Charlie	America	charlie@example.com

Product Shard

Product ID	Product	Category	Price
101	Smartphone	Electronics	$399
102	Running shoes	Sports	$111

Order Shard

Order ID	Product	Order Date	Status
201	Smartphone	2024-01-05	Shipped
202	Running shoes	2024-01-12	Pending

Partitioning and sharding are essential in System Design to distribute data across multiple nodes. This ensures scalability, high availability, and optimized performance by preventing any single server from becoming a bottleneck as the system grows.

To learn more about database partitioning and sharding, you can take help from the following course: