How to implement sharding

How Does Sharding Work?

The sharding process involves several key components including:

1. **Sharding Key**: The shard key is a unique identifier used to determine which shard a particular piece of data belongs to. It can be a single column or a combination of columns.
2. **Data Partitioning**: Partitioning involves splitting the data into shards based on the shard key. Each shard represents a portion of the total data set. Common strategies to partition database are **range-based, hash-based, and directory-based sharding.**
3. **Shard Mapping**: Creating a mapping of shard keys to shard locations.
4. **Shard Management**: The shard manager oversees the distribution of data across shards, ensuring data consistency and integrity.
5. **Query Routing**: The query router intercepts incoming queries and directs them to the appropriate shard(s) based on the shard key.

Or implement Consistent Hashing

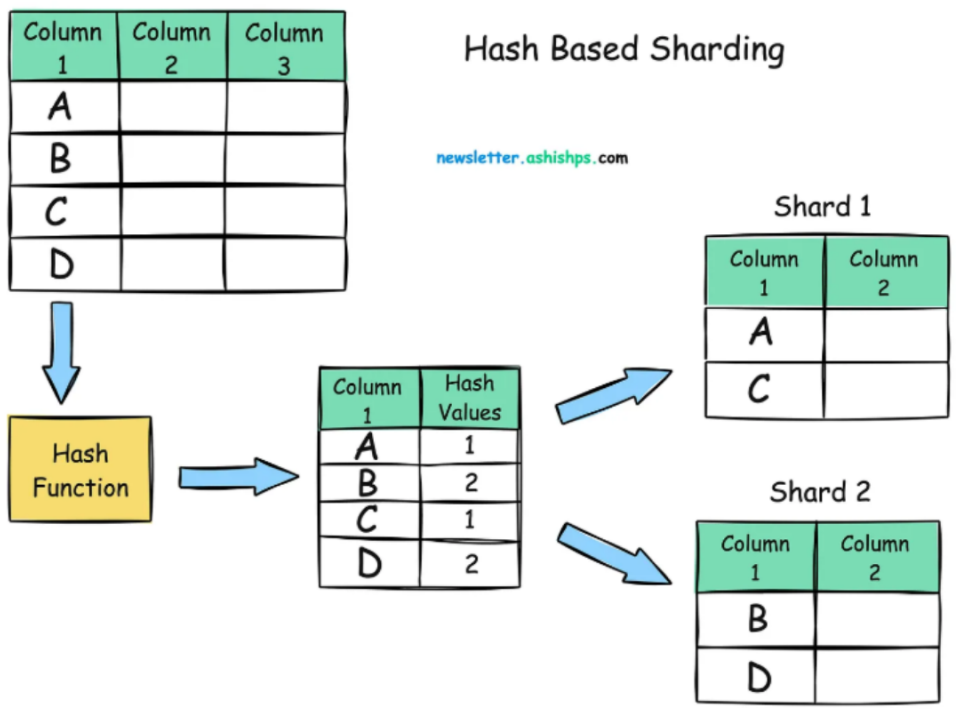
How Does Sharding Work?

The sharding process involves several key components including:

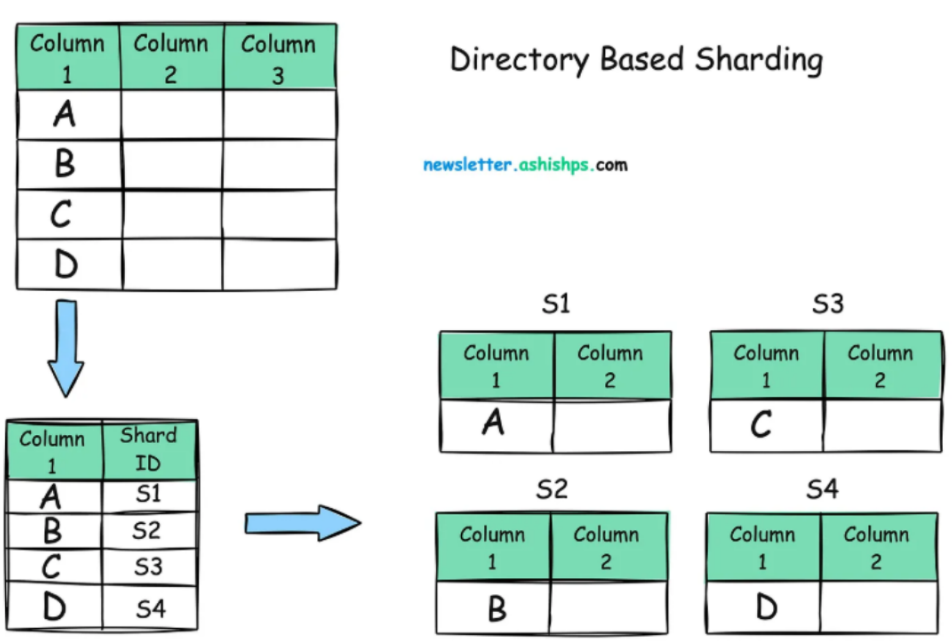
1. **Sharding Key**: The shard key is a unique identifier used to determine which shard a particular piece of data belongs to. It can be a single column or a combination of columns.
2. **Data Partitioning**: Partitioning involves splitting the data into shards based on the shard key. Each shard represents a portion of the total data set. Common strategies to partition database are **range-based, hash-based, and directory-based sharding.**
3. **Shard Mapping**: Creating a mapping of shard keys to shard locations.
4. **Shard Management**: The shard manager oversees the distribution of data across shards, ensuring data consistency and integrity.
5. **Query Routing**: The query router intercepts incoming queries and directs them to the appropriate shard(s) based on the shard key.

Sharding Strategies

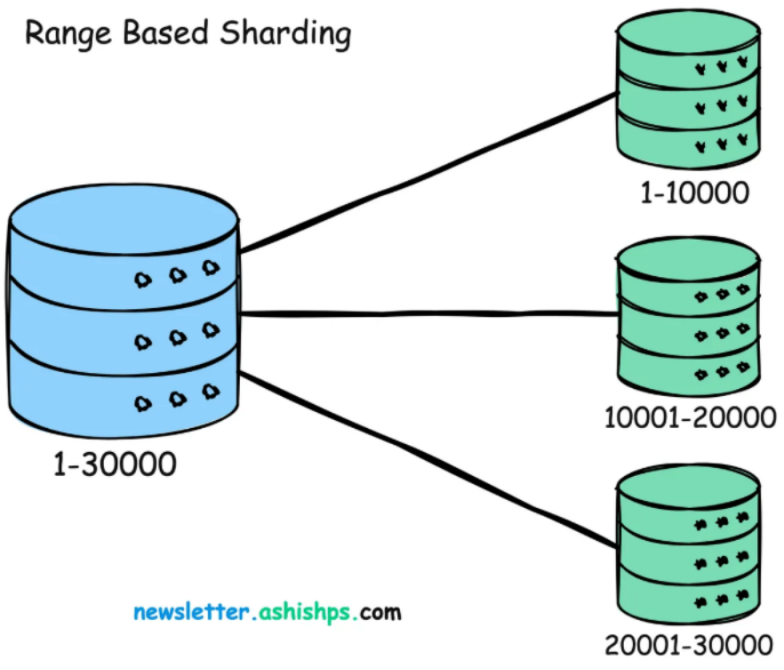
- **Hash-Based Sharding**: Data is distributed using a hash function, which maps data to a specific shard.
    - **Example:** `Hash(user_id) % 2` determines the shard number for a user, distributing users evenly across 2 shards.



- **Directory-Based Sharding**: Maintains a lookup table that directly maps specific keys to specific shards.



- **Range-Based Sharding**: Data is distributed based on a range of values, such as dates or numbers.
    - **Example:** Shard 1 contains records with IDs from 1 to 10000, Shard 2 contains records with IDs from 10001 to 20000, and so on.



- **Geo-Based Sharding**: Data is distributed based on geographic location.
    - **Example:** Shard 1 serves users in North America, Shard 2 serves users in Europe, Shard 3 serves users in Asia.