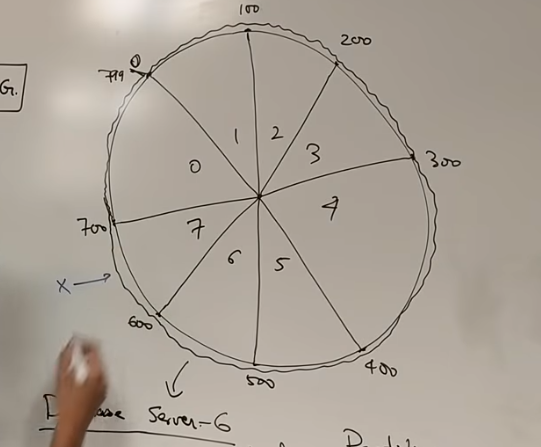
Database Sharding

Sharding a database is a common scalability strategy used when designing server side systems. The server side system architecture uses concepts like sharding to make systems more scalable, reliable and performant.

Sharding is horizontal partitioning of data according to a shard key. This shard key determines which database the entry to be persisted is sent to. Some common strategies for this are reverse proxies.

Eg. Sharding based on key as location as in Tinder.



**Problems:**

* Joins across shards. (across networks and hence costly).
* Shards are inflexible. Once sharded cannot be changed. One method to overcome is consistent hashing(creating virtual servers by hashing k times). Eg. Memcached is flexible by using application logic above database. This inflexibility of sharding can be overcome by dynamically by breaking each shard into smaller pieces and have a manager to map requests to the further slices of a shard.(hierarchical sharding)

**Best Practices:**

* Create index on the shards. The index can be on diff attribute compared to the shard key.
* In case a shard fails.eg. loss of electricity, we can have a master slave architecture.
* Practical implementation is touch. Consider indexing/ NoSql database etc.