Ref - https://www.youtube.com/watch?v=UFVFIKduXpo

Mongo DB - Managing HuMONGOus data

It is based on Document Data Model which is very flexible

It provides the consistency and partition tolerance in the CAP theorem (Mongo and HBase).Since it has to deal with bigdata partition tolerance is something it hust has to do and it chooses favour to consistency over availability.

It has a single master, primary database that you have to talk to all the time to ensure consistency but if master goes down we would be unavailable.

Basically any JSON blob data you can shove intoa document in MongoDB it does not have to be structured you don't have to have the same schema across each document. You put whatever you want.

1.No real schema is enforced in mongodb. You can have difference fields in every document if you want.

2.No single key as in other database.

- But you can create indices on any field you want or even combinations of fields

- If you want to shard then you must do so on some index (shard - horizontally partitioning it, different range of data would be available in different servers then you need to have some unique index to do that sharding on)

3.Results in a lot of flexibility - but with great power comes with great responsibility

4.As it is noSQL database you can't use join efficiently so you have to denormalize the schema as much as you can.

Terminology

1.Database (databases)

2.Collections (tables)

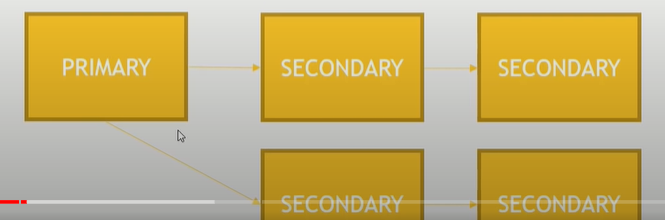
3.Documents (rows)

Note: the restriction is you can't move data between collections across different databases. So if you do need to reference data between different collections they do need to be on same database.

Replication sets:

1. Single master
2. Maintains backup copies of your database instance

* Secondaries can elect a new primary within seconds if your primary goes down
* But make sure your operation log is long enough to give you time to recover the primary when it comes back



<https://www.youtube.com/watch?v=UFVFIKduXpo>