Mongo DB

Data accessed together should be stored together.

Data Modelling.

(I) Polymorphism → DB ability to store data with diff types.
(II) Easy to work with arrays.
(III) Embedding → Store documents within parent documents to represent relationships

Normalize data → Referencing other documents from a main document.
$lookup storage (Aggregation pipeline)

Schema validation → Define rules & constraints. Executed on the database layer, it also works with polymorphic collections & embedded documents.

To join two table data in mongodb we use $lookup

ex → db.authors.aggregate([

{ Books table with author_id as foreign key & Authors table seperately

```
{
  $match: {
    _id: 20
  }
},
{
  $lookup: {
    from: "books",
    localField: "_id",
    foreignField: "author_id",
    as: "books"
  }
}
]);
```

Same as joins in SQL table.

Update an existing document and store a new schema document along with existing

(I) updateMany

```
db.books.updateMany(
  {} => empty if want to update all else put filter,
  { $set: { type: "paperback" } }
)
```

is same as

```
Alter table books
add column type varchar(50);

update books
set type = "paperback";
```

(II) Find with filter type & name inside document.

```
db.books.find(
  { type: "ebook", "reviews.username": "Bob" },
  { _id: 0, title: 1, "reviews.$": 1 }
)
```

this retrieves the first matching value from the reviews array.

To model a database in MongoDB

(I) Identifying the entities & quantifying the entities no' of records.

(II) Identifying all the read & write queries which user will by to access.

(III) Quantifying all the reads & writes number because write operations are heavier than read ones.

(IV) Identifying the relationships

(I) one to one → should be stored using embedding in single doc

(II) one to many → Referencing is preferred like storing the ids & other informations & large ones in diff doc

(III) many to many.

↓

container embed

or reference

Embedding is preffered for access as single unit. small, fixed data or we want to minimize the db operations.

Referencing is preferred when independent access. large or growing documents.

MongoDB Atlas

(I) Organization → group & define users & teams & grant them access to projects

(II) Projects → Define & organize databases & clusters. Create separate projects for development, testing & production.

M10 → is a dedicated cluster tier that's most suitable for handling large datasets & production deployments.

M0 → is a free cluster for experimentation purpose & it can be upgraded to serverless or dedicated tier if needed.

Serverless → Serverless tier operates on pay as you go, suitable for experimenting CRUD purposes.

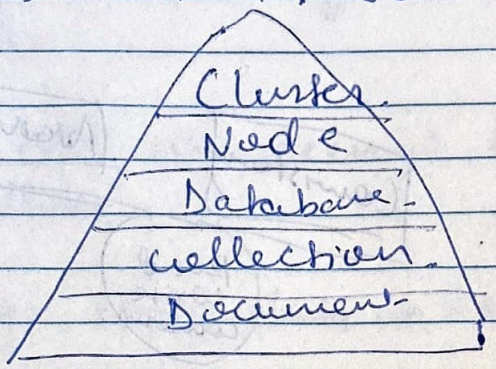
Also always create seperate projects for development, testing & production to isolate the environments.

& Name your organization descriptively to reflects its purpose.

Atlas UI allows to perform CRUD operation build aggregation pipeline & create search indexes which allow to use atlas search & atlas vector search.

Performance Advisor tells what we need to do to improve the performance like creating an index.

MongoDB is a distributed database
Stores data in JSON



(I) Document → object & any related metadata.
field value pair, Data types: strings, numbers, date arrays, objects & more

(II) collections → stores multiple documents into collection
Group of documents that corresponds to an entity, can support multiple entities & different shapes.

(III) Database → A group of collections

(IV) Node → An individual mongoDB instance

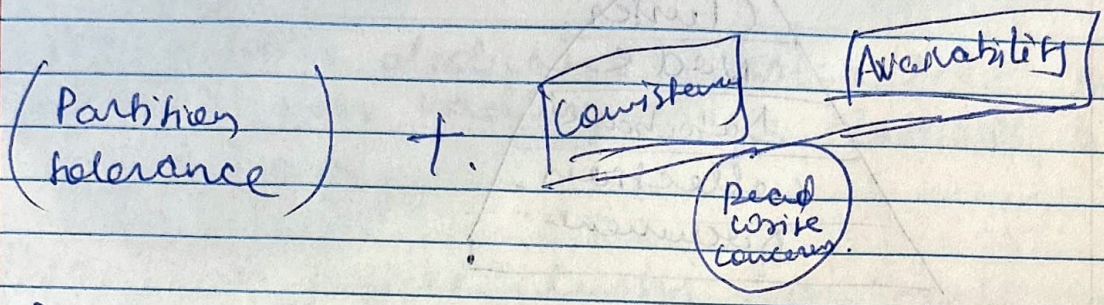
(V) Cluster types → replica sets → high availability
& Sharded clusters → scaling.
group of mongodb nodes.

CAP theorem

It is impossible to simultaneously guarantee consistency, availability & partition tolerance.

A distributed system can almost guarantee two of these.

Mongo DB.



Replica sets → A group of mongod instances that maintain the same data set.

Mongd → is the primary daemon process, it handles data request, manages data access performs background management operations.

Read concern → controls the consistency & isolation of data read from replica sets or sharded clusters if it is always local means it will return the data from the node it was read.

Write concern dictates the number of replica set members that must confirm a write operation. Default is set to majority.

Majority of the replica set confirms that write is successful.

In mongodb sharding provides horizontal scalibility

Atlas is DBaas → Database as service

Mongo data is field value pair & fields must be unique

Data that is accessed together should be stored together.

Mongo stores data in Bson → binary format

Max doc. size → 16 MB.
In a single document maximum of 100 levels of nesting.

In Bson, Numerical value supports two types of integers, Int32 & Int64 & for floating point value Double & Decimal128

Embeded doc is like document inside doc.
Dates are stored as timestamp, Epoch (milliseconds since Jan 1, 1970)