

1) Structured Data (Tabular)

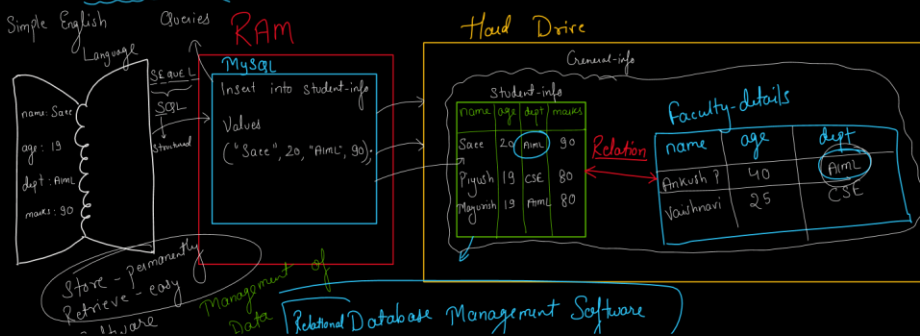
name	age	dept	marks	year
Kunal	23	AIML	72	2009
Shreyas	20	COML	70	2010
Mayuri	20	AIML	65	2021
Rohan	19	AIML	81	2025

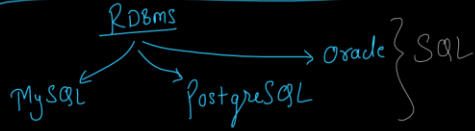
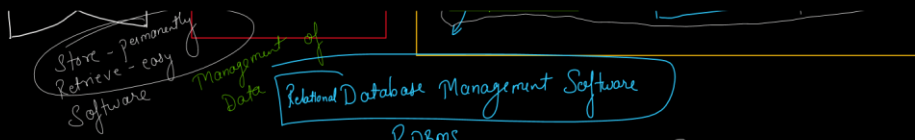
2) Semi-Structured (document)

```

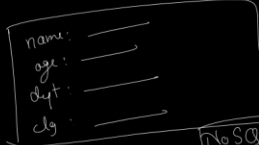
name: Kunal
age: 23
dept: AIML
marks: 72
year: 2009
  
```

3) Unstructured

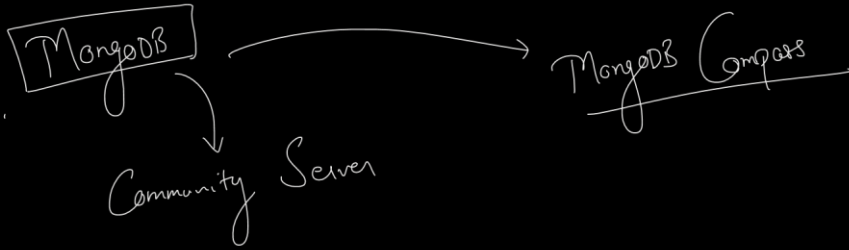
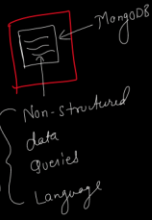




Semi-Structured (Document)



NoSQL

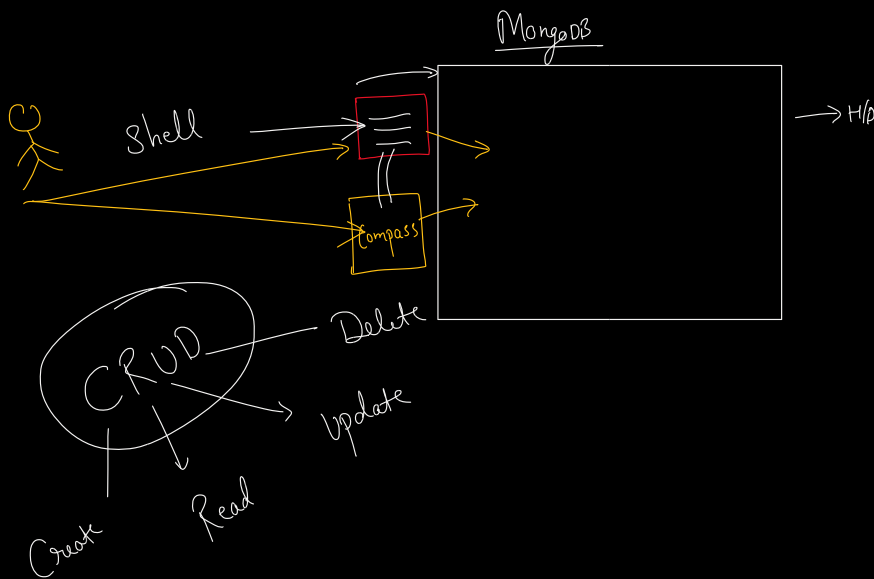


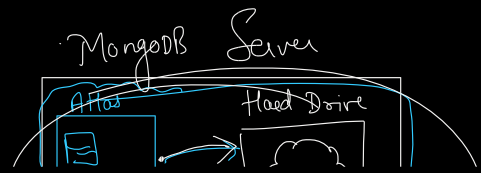
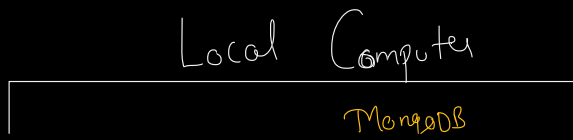
MySQL

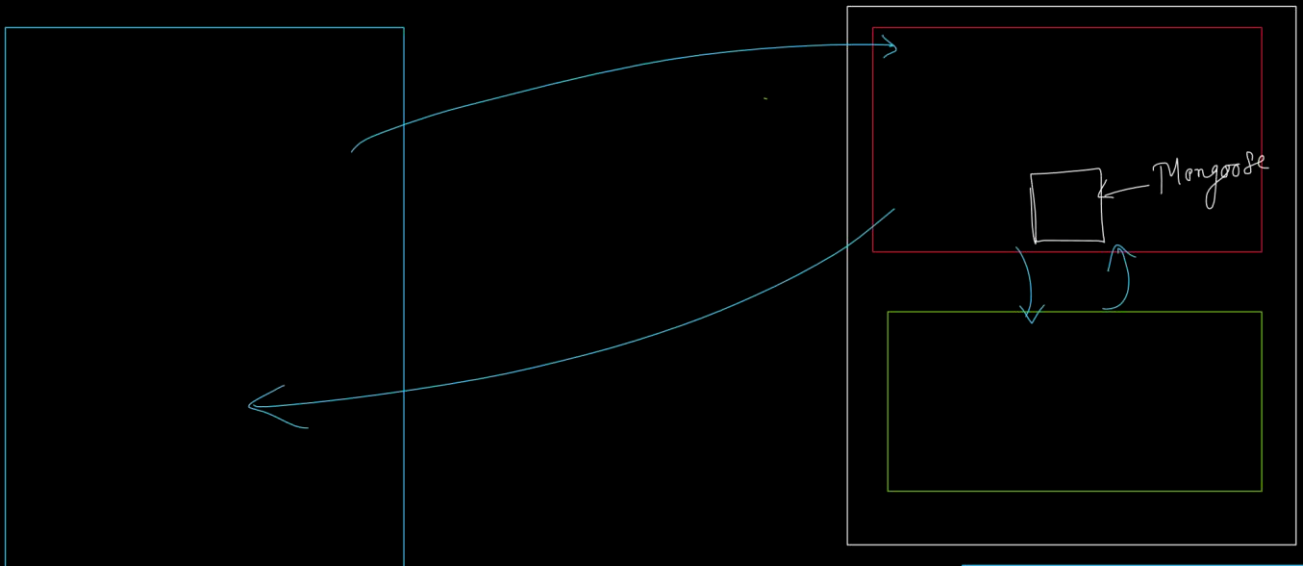
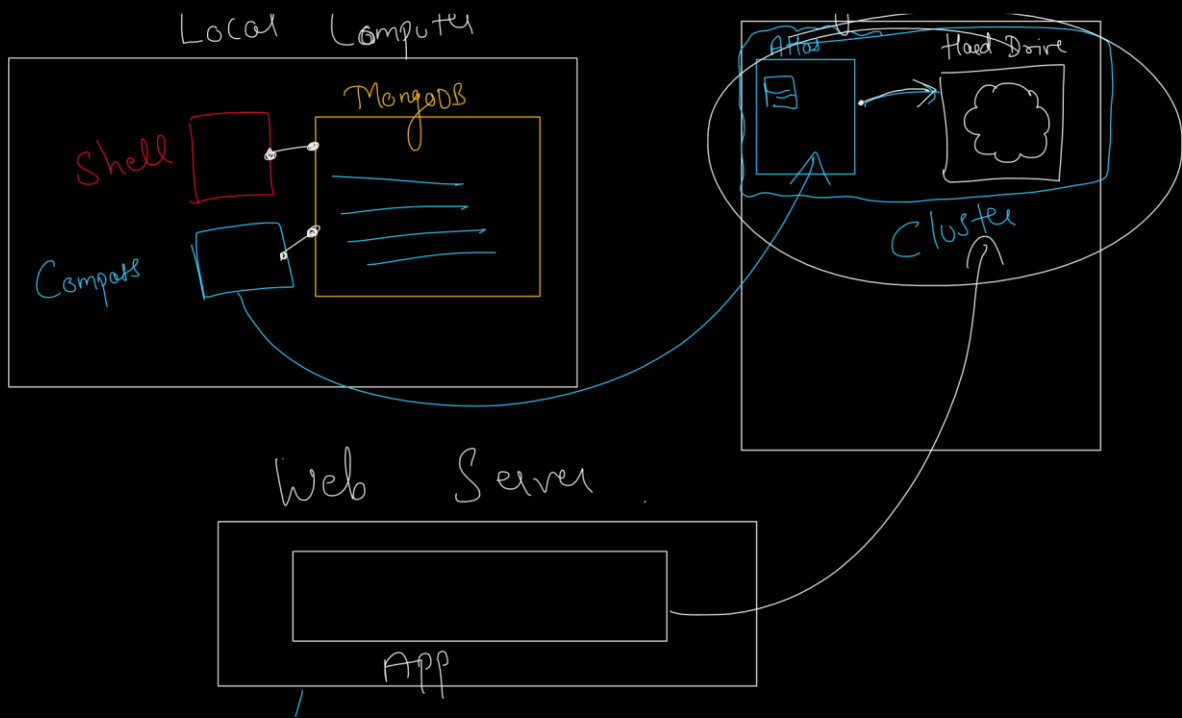
- Database
- Tables
- Row

MongoDB

- Database
- Collections
- Documents







```
File Edit Selection View Go Run ... Mongoose-Tutorial
```

EXPLORER

- server.js
- index.ejs
- style.css
- UserProfile.js

OPEN EDITORS

- server.js
- index.ejs
- style.css
- UserProfile.js

MONGOOSE-TUTORIAL

- models
- node_modules
- public/css
- style.css
- views
- index.ejs
- package-lock.json
- package.json
- server.js

```
37 // 5. Basic Route: Home page (list all user profiles)
38 app.get('/', async (req, res) => {
39   try {
40     const profiles = await UserProfile.find({});
41     // Render 'index.ejs' with an array of profiles
42     res.render('index', profiles);
43   } catch (error) {
44     res.status(500).send(error.message);
45   }
46 });
47
48 // 6. Create (POST) a new user profile
49 app.post('/users', async (req, res) => {
50   try {
51     const newUser = await UserProfile.create(req.body);
52     // Redirect user to home page or return 301
53     res.redirect('/');
54   } catch (error) {
55     res.status(400).send('Error creating user: ' + error.message);
56   }
57 });
58
59 // 7. Start the server
```

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

rahul@Romeo-Alpha09 MINGW64 /r/MERN Stack VAP - Jan 25 - Course Resources/BackEnd-Tutorial/mongoose-Tutorial (main)

```
rahul@Romeo-Alpha09 MINGW64 /r/MERN Stack VAP - Jan 25 - Course Resources/BackEnd-Tutorial/mongoose-Tutorial
(main)
$
```

