Ако е с unique key в колоната username.
→ дава грешка, ако опитаме да добавим нещо, което вече имаме

Пишем си insert.
Ако даде err

```
insert into users (username, password) values (?,?)
select username from users where username like "?%"
```

↳ гледаме там, където има още подобни имена за да дадем незаето подобно.

или просто пише ново име докато стане

релационни бази данни: лесно и бързо прави join
↳ MySQL е такава
↳ пазим информацията в таблици

MongoDB → NoSQL
сваляме community версия
↳ пази информацията в collectionи
↓
в него има обекти

API - application programming ~~[illegible]~~ interface
за език
↳ библиотека, с която достъпваме функции от езика

У може да имат различни имена на параметрите

вместо con.query("sql code") имаме:

db.collection("име на collection").insertOne
↓
инсертва 1 елемент

db.collection("име на collection").insertOne({username: pesho, password: "1234"})
→ JSON

→ самия код дава грешка, а не само базата данни в err
- по-бързо записване на данни
=> гъвкавост на данните.
пр.: имаме готов проект, но решаваме че за всеки потребител да пазим и имейл, т.е. трябва да променим какво пазим.
→ MongoDB: променяме кода и готово
→ SQL: добавяме колона, празни клетки или нещо и т.н. (по-трудно)
много компресирани колони → пазим само тези, които ни касаят
недостатъци: значително по-бавно четене и търсене на данни
по-трудна работа със свързани данни (няма foreign key, join и т.н.)

MongoDB Compass → подобно на DBearer
↳ и него инсталираме за MacOS

- влизаме на някакъв server и там има бази данни

Create Database
→ трябва поне един collection

Add data → import file
↳ insert one doc.

имената на параметрите трябва да са в кавички

npm install mongodb
socketio
express

var MongoClient = require("mongodb").MongoClient;
↳ зареждане API на MongoDB

MongoClient.connect("mongodb://localhost:27017/", (err, con) => {
//кодът, който се изпълнява след свързването с MongoDB
let db = con.db("името на базата данни");
↳ свързване към една база данни
след това чак стартираме сървъра
})

let coll = db.collection("име на collection") → дава конкретния collection

coll.insertOne(json, (err,res)=>{})
↳ код след изпълнение на insert.

coll.~~insert~~ insertMany([json1, json2, ...], (err,res)=>{})
insert на много неща

coll.findOne(json, (err,res)=>{})
↳ съдържа първия обект от collection-а, който има полетата, описани в json-а

пример: coll.findOne({username:"pesho"}, (err,res) => { })
↳ res ще даде първия обект, който има поле username със стойност "pesho"

съответно find повече от едно...

unique key в MongoDB:

Create index
↳ index definition, options
(името на параметъра) ↳ unique index

ако имаме {username:username}
можем само {username}

alert("text"); -> прозорче с "text"

* {useUnifiedTopology=true} -> добре е да го има
конзолата иначе дава warning, който не е проблем

има updateOne -> ъпдейтва данните