1. Назва та версія СУБД

CУБД – MongoDB- [документоориентированная](https://ru.wikipedia.org/wiki/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%BE%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D0%A1%D0%A3%D0%91%D0%94) [система управления базами](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0%D0%BC%D0%B8_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85" \o "Система управления базами данных)

[данных](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0%D0%BC%D0%B8_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85" \o "Система управления базами данных), не требующая описания схемы таблиц. Считается одним из классических примеров [NoSQL](https://ru.wikipedia.org/wiki/NoSQL" \o "NoSQL)-систем, использует [JSON](https://ru.wikipedia.org/wiki/JSON)-подобные документы и схему базы данных. Применяется в веб-разработке, в частности, в рамках [JavaScript](https://ru.wikipedia.org/wiki/JavaScript" \o ").

Версия – MongoDB 4.4

MongoDB подходит для следующих применений:

* хранение и регистрация событий;
* системы управления документами и контентом;
* электронная коммерция;
* данные мониторинга, датчиков;
* мобильные приложения;
* хранилище операционных данных веб-страниц (например, хранение комментариев, рейтингов, профилей пользователей, сеансы пользователей).

Создание структур данных – JSON-структура – нет чётких требований и стандартов, связанных с его описанием.

Ниже представлен код работы структуры данных:

const userControllr = require("../controllers/post\_controller");

router.get("/posts", userControllr.findAll);

router.post("/add\_post", userControllr.create);

router.get("/post/:id", userControllr.findOne);

router.put("/posts/:id", userControllr.UpdateUser);

router.delete("/posts/:id", userControllr.delete);

const Post = require("../models/post\_model");

exports.create = (req, res) => {

var db = req.db;

var title = req.body.title;

var description = req.body.description;

var video = req.body.video;

var new\_post = new Post({

title: title,

description: description,

video: video

})

new\_post.save(function (error) {

if(error) {

console.log(error)

}

res.send({

success: true

})

})

};

exports.delete = (req, res) => {

var db = req.db;

Post.remove({

\_id: req.params.id

}, function (err, post) {

if (err) {

res.send(err)

}

res.send({

success: true

})

})

};

exports.UpdateUser = (req, res) => {

var db = req.db;

Post.findById(req.params.id, 'title description video', function (error, post) {

if (error) {

console.error(error);

}

post.title = req.body.title

post.description = req.body.description

post.video = req.body.video

post.save(function (error) {

if (error) {

console.log(error)

}

res.send({

success: true

})

})

})

};