Практика на MongoDB. Виноградов Константин Олегович 23КНТ-2, вариант 23, пункт а)

Изображение выглядит как текст, Шрифт, снимок экрана, белый

Контент, сгенерированный ИИ, может содержать ошибки.

db.doctors.aggregate([

{

$project: {

"taxes": { $subtract: [100, "$received\_percentage"] }

}

},

{

$group: {

"\_id": "$taxes"

}

},

{

$project: {

"taxes": "$\_id",

"\_id": 0

}

}

]);

Пример работы:

Изображение выглядит как текст, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, Шрифт, чек, белый

Контент, сгенерированный ИИ, может содержать ошибки.

db.patients.find(

{

"discount": { $gt: 20 }

},

{

"surname": 1,

"department": 1,

"\_id": 0

}

);

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, Шрифт, снимок экрана, белый

Контент, сгенерированный ИИ, может содержать ошибки.

db.prescriptions.aggregate([

{

$lookup: {

"from": "patients",

"localField": "patient\_id",

"foreignField": "\_id",

"as": "patient"

}

},

{

$unwind: "$patient"

},

{

$lookup: {

"from": "procedures",

"localField": "procedure\_id",

"foreignField": "\_id",

"as": "procedure"

}

},

{

$unwind: "$procedure"

},

{

$project: {

"number": "$\_id",

"surname": "$patient.surname",

"date": 1,

"name": "$procedure.name",

"\_id": 0

}

},

{

$sort: {

"surname": 1

}

}

]);

Изображение выглядит как текст, снимок экрана, дизайн

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Контент, сгенерированный ИИ, может содержать ошибки.

И так далее.

Изображение выглядит как текст, Шрифт, снимок экрана, белый

Контент, сгенерированный ИИ, может содержать ошибки.

db.prescriptions.aggregate([

{

$lookup: {

"from": "doctors",

"localField": "doctor\_id",

"foreignField": "\_id",

"as": "doctor"

}

},

{

$unwind: "$doctor"

},

{

$lookup: {

"from": "procedures",

"localField": "procedure\_id",

"foreignField": "\_id",

"as": "procedure"

}

},

{

$unwind: "$procedure"

},

{

$match: {

$or: [

{ "procedure.name": "ЭКГ" },

{ $expr: { $eq: ["$doctor.department", "$procedure.department"] } }

]

}

},

{

$group: {

"\_id": "$doctor.surname"

}

},

{

$project: {

"surname": "$\_id",

"\_id": 0

}

}

]);

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Уровень 2. Посмотрим, как можно реализовать эти запросы в MongoDB

Изображение выглядит как текст, Шрифт, белый, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

db.patients.aggregate([

{

$lookup: {

"from": "prescriptions",

"localField": "\_id",

"foreignField": "patient\_id",

"as": "prescriptions"

}

},

{

$lookup: {

"from": "doctors",

"localField": "prescriptions.doctor\_id",

"foreignField": "\_id",

"as": "doctors"

}

},

{

$match: {

"doctors.surname": { $nin: ["Гусев"] }

}

},

{

$project: {

"surname": 1,

"\_id": 0

}

}

]);

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, Шрифт, снимок экрана, чек

Контент, сгенерированный ИИ, может содержать ошибки.

db.doctors.aggregate([

{

$sort: {

"received\_percentage": -1

}

},

{

$limit: 1

},

{

$project: {

"department": 1,

"\_id": 0

}

}

]);

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, Шрифт, белый, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

db.patients.aggregate([

{

$project: {

"surname": 1,

"\_id": 0

}

},

{

$unionWith: {

"coll": "doctors",

"pipeline": [

{

$project: {

"surname": 1,

"\_id": 0

}

}

]

}

},

{

$group: {

"\_id": "$surname"

}

},

{

$project: {

"surname": "$\_id",

"\_id": 0

}

}

]);

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Контент, сгенерированный ИИ, может содержать ошибки.

И так далее.

Изображение выглядит как текст, снимок экрана, Шрифт, алгебра

Контент, сгенерированный ИИ, может содержать ошибки.

Вставим данные, чтобы ответ не был пустым:

db.prescriptions.insertMany([

{

"\_id": 43084,

"date": "Суббота",

"patient\_id": 5,

"doctor\_id": 4,

"procedure\_id": 2,

"quantity": 1,

"payment": 0,

"payout": 0

},

{

"\_id": 43085,

"date": "Суббота",

"patient\_id": 5,

"doctor\_id": 2,

"procedure\_id": 3,

"quantity": 1,

"payment": 0,

"payout": 0

},

{

"\_id": 43086,

"date": "Суббота",

"patient\_id": 5,

"doctor\_id": 1,

"procedure\_id": 5,

"quantity": 1,

"payment": 0,

"payout": 0

},

{

"\_id": 43087,

"date": "Суббота",

"patient\_id": 5,

"doctor\_id": 3,

"procedure\_id": 6,

"quantity": 1,

"payment": 0,

"payout": 0

}

]);

db.prescriptions.aggregate([

{

$lookup: {

"from": "procedures",

"localField": "procedure\_id",

"foreignField": "\_id",

"as": "procedure"

}

},

{

$unwind: "$procedure"

},

{

$lookup: {

"from": "patients",

"localField": "patient\_id",

"foreignField": "\_id",

"as": "patient"

}

},

{

$unwind: "$patient"

},

{

$set: {

"payment": {

$multiply: [

"$quantity",

"$procedure.price",

{ $subtract: [1, { $divide: ["$patient.discount", 100] }] }

]

}

}

},

{

$merge: {

"into": "prescriptions",

"whenMatched": "merge",

"whenNotMatched": "discard"

}

}

]);

db.prescriptions.aggregate([

{

$lookup: {

"from": "doctors",

"localField": "doctor\_id",

"foreignField": "\_id",

"as": "doctor"

}

},

{

$unwind: "$doctor"

},

{

$set: {

"payout": {

$multiply: [

"$payment",

{ $divide: ["$doctor.received\_percentage", 100] }

]

}

}

},

{

$merge: {

"into": "prescriptions",

"whenMatched": "merge",

"whenNotMatched": "discard"

}

}

]);

db.patients.aggregate([

{

$match: {

"discount": { $gt: 0 }

}

},

{

$lookup: {

"from": "procedures",

"pipeline": [

{

$group: {

"\_id": null,

"procedureIds": { $push: "$\_id" }

}

}

],

"as": "allProcedures"

}

},

{

$unwind: "$allProcedures"

},

{

$lookup: {

"from": "prescriptions",

"localField": "\_id",

"foreignField": "patient\_id",

"as": "patientPrescriptions"

}

},

{

$unwind: {

"path": "$patientPrescriptions",

"preserveNullAndEmptyArrays": true

}

},

{

$match: {

$or: [

{ "patientPrescriptions": { $exists: false } },

{ "patientPrescriptions.date": { $ne: "Понедельник" } }

]

}

},

{

$group: {

"\_id": "$\_id",

"surname": { $first: "$surname" },

"allProcedures": { $first: "$allProcedures.procedureIds" },

"procedureIds": { $addToSet: "$patientPrescriptions.procedure\_id" }

}

},

{

$match: {

$expr: {

$eq: [

{ $size: "$allProcedures" },

{ $size: { $setIntersection: ["$allProcedures", "$procedureIds"] } }

]

}

}

},

{

$project: {

"surname": 1,

"\_id": 0

}

}

]);

Изображение выглядит как текст, снимок экрана, Шрифт, Графика

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, Шрифт, снимок экрана, белый

Контент, сгенерированный ИИ, может содержать ошибки.

db.procedures.aggregate([

{

$group: {

"\_id": null,

"avgPrice": { $avg: "$price" },

"procedures": { $push: "$$ROOT" }

}

},

{

$unwind: "$procedures"

},

{

$match: {

$expr: { $gt: ["$procedures.price", "$avgPrice"] }

}

},

{

$project: {

"name": "$procedures.name",

"\_id": 0

}

}

]);

Изображение выглядит как текст, снимок экрана, Шрифт, Графика

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, Шрифт, снимок экрана, белый

Контент, сгенерированный ИИ, может содержать ошибки.

db.prescriptions.aggregate([

{

$lookup: {

"from": "patients",

"localField": "patient\_id",

"foreignField": "\_id",

"as": "patient"

}

},

{

$unwind: "$patient"

},

{

$match: {

"patient.department": "Кардиология"

}

},

{

$group: {

"\_id": "$patient.surname",

"totalPayment": { $sum: "$payment" }

}

},

{

$project: {

"surname": "$\_id",

"totalPayment": 1,

"\_id": 0

}

}

]);

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Контент, сгенерированный ИИ, может содержать ошибки.