

Защита 5 лабораторной работы

1. В чем отличие DBref от FK?

DBref служит для ссылки на другой документ не обязательно нормализованной базы данных и, как правило, содержит `_id` документа, имя коллекции, имя базы данных

FK характерен для нормализованных баз данных, указывает (представляет собой значение) на соответствующую запись в другой таблице (как правило совпадает с первичным ключом записи в другой таблице)

4. find. Синтаксис и примеры. Как ограничить выборку?

`db.collection_name.find({параметры поиска}, {параметры ограничения выборки}).опциональная_функция()`.

Пример: выберем из коллекции автомобилей только те, что стоят более 10000 у.е, без параметра `_id`, первые 3

`db.autos.find({price:{$gt : 10000}}, {_id : 0}).limit(3)`

7. \$in, \$all. Как и для чего они используются? Привести примеры.

\$in - логический оператор, который определяет массив возможных выражений и ищет те ключи, значение которых имеется в массиве.

Пример выборки автомобилей как минимум с одним из двух цветов.

`db.autos.find({color:{$in : ["white", "purple"]}})`

\$all - логический оператор, который ищет документы, чей набор значений, в отличие от \$in, полностью совпадает с данным множеством

Пример выборки автомобилей бело-фиолетового корпуса

`db.autos.find({color:{$all : ["white", "purple"]}})`

10. Создайте новую базу данных MongoDB с именем "task10db". В этой базе данных создайте коллекцию "products". Добавьте не менее 500 записей о продуктах в коллекцию (название и цена), используйте цикл. Название не обязательно должно быть осмысленным.

Осуществите запросы поиска минимальной цены, максимальной цены и сортировки продуктов по цене.

Создайте индекс для цены.

Выполните те же запросы. Сравните производительность до и после создания индексов.

use task10db

db.createCollection("products")

*for(i = 0; i < 500; i++){db.products.insert({name:i,
price:Math.floor(Math.random()*200)}}}*

db.products.find({}).sort({price : -1}).limit(1) - максимальное значение

db.products.find({}).sort({price : 1}).limit(1) - минимальное значение

db.products.find({}).sort({price : 1}) - сортировка по цене

db.products.ensureIndex({"price":1})

db.products.explain("executionStats").find({}).sort({price : 1}).limit(1)

*Данная операция при индексации показала 1мс, без - 2мс. На более
большом массиве значений превосходство индексации было бы более
очевидно*