ГУАП

КАФЕДРА № 43

ОТЧЕТ

ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

старший преподаватель М. В. Величко

должность, уч. степень, звание подпись, дата инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

Лабораторная работа №12   
Манипулирование данными в документной базе данных

по курсу: «Проектирование баз данных»

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. № Z1431 М.Д..Быстров

подпись, дата инициалы, фамилия

Санкт-Петербург 2024

# ЗАДАНИЕ

Привести пример обновления и удаления данных из базы

Выполнить запросы на выборку по варианту задания

**Вариант** 4:

парк: статуи, фонтаны, деревья ,породы, дата высадки, дата обрезки, расположение, аллеи

1. аллеи, на которых встречаются разные виды кленов (клен в названии)
2. аллеи, на которых есть и статуи и фонтаны
3. дерево, которое было посажено позже всех
4. порода, деревьев которой больше всего
5. аллея, на которой нет фонтанов

**Модель БД**

* + - 1. Аллеи

{

"id": 2,

"num": 2

}

* + - 1. Элементы парковой композиции

{

"alley\_id": 3,

"id": 6,

"name": "Клен",

"specs": {

"cut\_date": "2024-10-28T00:00:00",

"plant\_date": "2024-09-28T00:00:00",

"species": "клен канадский"

},

"type": "tree"

}

* + - 1. Породы деревьев

{

"name": "клен канадский"

}

**Пример обновления и удаления данных**

Обновление данных:

db.alleys.update({id: 7}, {$set:{id: 7, num: 6}})

{

acknowledged: true,

insertedId: null,

matchedCount: 1,

modifiedCount: 1,

upsertedCount: 0

}

Удаление данных:

db.alleys.deleteOne({id: 8})

**{**

**acknowledged: true,**

**deletedCount: 1**

**}**

**Текст запросов и наборы данных**

1. аллеи, на которых встречаются разные виды кленов (клен в названии)

db.getCollection('alleys').aggregate(

[

{

$lookup:

{

from: "items",

localField: "id",

foreignField: "alley\_id",

pipeline: [

{

$match: {

name: /Клен/

}

}

],

as: "maples"

}

},

{

$match:

{

$expr: {

$gt: [

{

$size: "$maples"

},

0

]

}

}

},

{

$unwind:

{

path: "$maples"

}

},

{

$group:

{

\_id: {

\_id: "$\_id",

id: "$id",

num: "$num"

},

unique\_species: {

$addToSet: "$maples.specs.species"

}

}

},

{

$match:

{

$expr: {

$gt: [

{

$size: "$unique\_species"

},

1

]

}

}

},

{

$project:

{

\_id: "$\_id.\_id",

id: "$\_id.id",

num: "$\_id.num"

}

}

]);

Результат выполнения запроса:

{

"\_id": {

"$oid": "672d0be72b729721dbd1b349"

},

"id": 4,

"num": 4

}

2. аллеи, на которых есть и статуи и фонтаны

db.getCollection('alleys').aggregate([

{

$lookup:

{

from: "items",

localField: "id",

foreignField: "alley\_id",

pipeline: [

{

$match: {

type: "statue"

}

}

],

as: "statues"

}

},

{

$lookup:

{

from: "items",

localField: "id",

foreignField: "alley\_id",

pipeline: [

{

$match: {

type: "fountain"

}

}

],

as: "fountains"

}

},

{

$match:

{

$expr: {

$and: [

{

$gt: [

{

$size: "$fountains"

},

0

]

},

{

$gt: [

{

$size: "$statues"

},

0

]

}

]

}

}

},

{

$project:

{

id: 1,

num: 1

}

}

]);

Результат выполнения запроса:

{

"\_id": {

"$oid": "672d0be72b729721dbd1b34b"

},

"id": 6,

"num": 6

}

3. дерево, которое было посажено позже всех

Запрос:

db.getCollection('items').aggregate( [

{

$group:

{

\_id: **null**,

max\_plant\_date: {

$max: "$specs.plant\_date"

}

}

},

{

$lookup:

{

from: "items",

localField: "max\_plant\_date",

foreignField: "specs.plant\_date",

as: "result"

}

},

{

$unwind:

{

path: "$result"

}

},

{

$replaceWith:

"$result"

},

{

$limit:

1

}

]);

Результат выполнения запроса:

{

"\_id": {

"$oid": "672d14e52b729721dbd1b36f"

},

"alley\_id": 3,

"id": 6,

"name": "Клен",

"specs": {

"cut\_date": "2024-10-28T00:00:00",

"plant\_date": "2024-09-28T00:00:00",

"species": "клен канадский"

},

"type": "tree"

}

* + - 1. порода, деревьев которой больше всего

db.getCollection('species').aggregate([

{

$lookup:

{

from: "items",

localField: "name",

foreignField: "specs.species",

as: "trees"

}

},

{

$set: {

count: {

$size: "$trees"

}

}

},

{

$sort:

{

count: -1

}

},

{

$limit:

1

},

{

$unset:

["count", "trees"]

}

]);

Результат выполнения запроса:

{

"\_id": {

"$oid": "672d0eb52b729721dbd1b366"

},

"name": "красный клен"

}

* + - 1. аллея, на которой нет фонтанов

db.getCollection('alleys').aggregate([

{

$lookup:

{

from: "items",

localField: "id",

foreignField: "alley\_id",

pipeline: [

{

$match: {

type: "fountain"

}

}

],

as: "fountains"

}

},

{

$set:

{

fountains\_count: {

$size: "$fountains"

}

}

},

{

$match:

{

$expr: {

$eq: ["$fountains\_count", 0]

}

}

},

{

$unset:

["fountains", "fountains\_count"]

}

]);

Результат выполнения запроса:

[{

"\_id": {

"$oid": "672d0be72b729721dbd1b348"

},

"id": 3,

"num": 3

},

{

"\_id": {

"$oid": "672d0be72b729721dbd1b349"

},

"id": 4,

"num": 4

},

{

"\_id": {

"$oid": "672d0be72b729721dbd1b34a"

},

"id": 5,

"num": 5

},

{

"\_id": {

"$oid": "672d0be72b729721dbd1b34c"

},

"id": 7,

"num": 7

}]

**Вывод**

В ходе выполнения 12 лабораторной работы были созданы запросы к документной базе данных в соответствии с вариантом.

Приведена физическая модель БД в формате JSON.

Все запросы выполнены с помощью конвейеров агрегации, выполняющих последовательные преобразования выборки документов коллекции.

Результаты запросов к тестовым наборам данных совпали с ожидаемыми.