ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Старший преподаватель |  |  |  | Н.В Путилова |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №2  Манипулирование данными в документной базе данных. |
| **по дисциплине: Проектирование баз данных** |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. | 4134к |  |  |  | Костяков Н.А. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

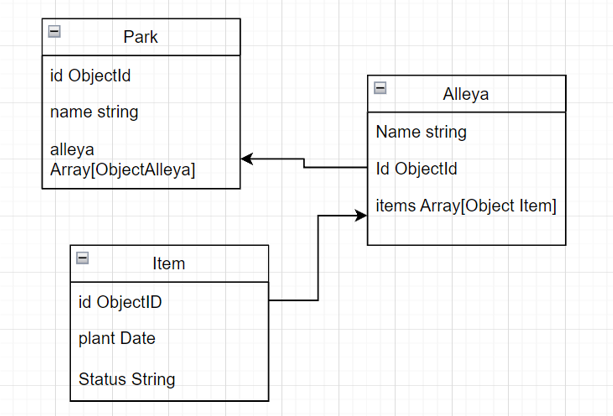
Санкт-Петербург

2023

**Описание задания**

1)Привести пример обновления и удаления данных из документной базы данных, разработанной в работе 11. (по 2 примера: один из вложенного документа, другой из документа коллекции верхнего уровня)  
2)Выполнить запросы на выборку из базы данных в соответствии с вариантом задания .  
(Запрос на поиск подстроки.+  
Запрос на использование одного документа несколько раз.  
Запрос с условием на агрегатную функцию.+  
Запрос получением агрегатной функции от агрегатной функции.+  
Запрос на разность 2 запросов)

Модель бд



**Вариант** 4 Парки и аллеи

**Листинг запросов**

//a)аллеи, на которых встречаются разные виды кленов (клен в названии)

db.alleya.aggregate([

  {

    $unwind: "$items" // Развернуть массив items

  },

  {

    $match: {"items.status": "tree", "items.id": {$in: db.item.distinct("\_id", {"kind": /Клен/i})}} // Фильтр для кленов

  },

  {

    $group: {

      \_id: "$\_id",

      name: {$first: "$name"}, // Получить имя аллеи

      count: {$sum: 1} // Подсчитать количество кленов на аллее

    }

  }

,

  {

    $match: {count: {$gte: 2}} // Фильтр для аллеи с 2 и более кленами

  },

  {

    $project: {name: 1} // Вывести только имя аллеи

  }

])

// б)аллеи, на которых есть и статуи и фонтаны

db.alleya.aggregate([

  {

    $match: {

      "items.status": { $all: ["statue", "fountain"] } // Находим аллеи с элементами статуй или фонтанов

    }

  },

  {

    $group: {

      \_id: "$\_id",

      name: { $first: "$name" }, // Получаем имя аллеи

      items: { $push: "$items.status" } // Создаем массив статусов элементов на аллее

    }

  }

])

// в)дерево, которое было посажено позже всех

db.item.aggregate([

  {

    $match: { "status": "tree" } // Находим только элементы с типом "tree"

  },

  {

    $group: {

      \_id: null,

      latestPlantDate: { $max: "$plant" } // Находим максимальную дату посадки

    }

  },

  {

    $lookup: {

      from: "item",

      let: { latestPlantDate: "$latestPlantDate" },

      pipeline: [

        {

          $match: {

            $expr: {

              $eq: ["$plant", "$$latestPlantDate"] // Находим элемент с датой посадки равной максимальной

            },

            "status": "tree" // Убеждаемся, что это дерево

          }

        }

      ],

      as: "latestTree" // Сохраняем результат в поле "latestTree"

    }

  },

  {

    $unwind: "$latestTree" // Развертываем результат

  },

  {

    $project: {

      \_id: "$latestTree.\_id",

      kind: "$latestTree.kind",

      plantDate: "$latestTree.plant"

    }

  }

])

// г)порода, деревьев которой больше всего //вывод с одинаковым числом

db.item.aggregate([

  {

    $match: { "status": "tree" } // фильтруем только документы со статусом "tree"

  },

  {

    $group: {

      \_id: "$kind", // группируем по полю "kind"

      count: { $sum: 1 } // считаем количество документов в каждой группе

    }

  },

  {

    $group: {

      \_id: null,

      maxCount: { $max: "$count" }, // находим максимальное значение count

      trees: { $push: { kind: "$\_id", count: "$count" } } // сохраняем все породы деревьев и их count

    }

  },

  {

    $project: {

      \_id: 0,

      maxTrees: {

        $filter: {

          input: "$trees",

          as: "tree",

          cond: { $eq: ["$$tree.count", "$maxCount"] } // фильтруем только породы с максимальным значением count

        }

      }

    }

  }

])

// д)аллея, на которой нет фонтанов -------------------------------------------------

db.collection.aggregate([

  {

    $project: {

      name: 1,

      fountainCount: {

        $size: {

          $filter: {

            input: "$items",

            as: "item",

            cond: { $eq: ["$$item.status", "fountain"] }

          }

        }

      }

    }

  },

  {

    $match: { fountainCount: { $eq: 0 } }

  },

  {

    $project: { \_id: 0, name: 1 }

  }

])

// update

db.alleya.update\_one(

  {"\_id" : ObjectId("65c9f3d329b337b832d3d4")},

   { "$push": { "items": {"\_id":  ObjectId("65c9f3d329b337b832d3f4"), "status":"statue"} } })

db.item.updateOne(

  { "\_id": ObjectId("601c11b3d8e31506c64b07d4") },

  { "$set": { "kind": "Клен" } }

);

// delete

db.item.deleteOne({ "\_id": ObjectId("601c11b3d8e31506c64b07d4") });

db.alleya.updateMany(

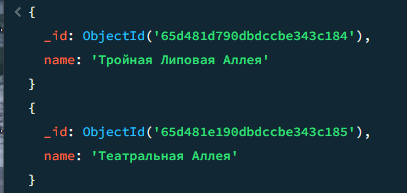
  {}, // Пустой фильтр, чтобы применить обновление ко всем документам коллекции

  { $pull: { "items": { "id": ObjectId("601c11b3d8e31506c64b07d4") } } }

)

Результаты работы

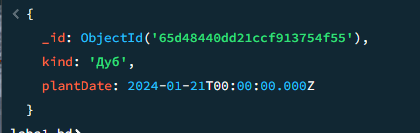
//a)аллеи, на которых встречаются разные виды кленов (клен в названии)



 б)аллеи, на которых есть и статуи и фонтаны



В) дерево, которое было посажено позже всех



Г) порода, деревьев которой больше всего //вывод с одинаковым числом



Д) аллея, на которой нет фонтанов

