

ГУАП

КАФЕДРА № 43

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

старший преподаватель

должность, уч. степень, звание

М. В. Величко

инициалы, фамилия

подпись, дата

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

Лабораторная работа №12
Манипулирование данными в
документной базе данных

по курсу: «Проектирование баз данных»

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. №

Z1431

подпись, дата
фамилия

М.Д..Быстров
инициалы,

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

- а. 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
- д. аллея, на которой нет фонтанов

Модель БД

1. Аллеи - alleys
2. Элементы парковой композиции - items
3. Породы деревьев - species

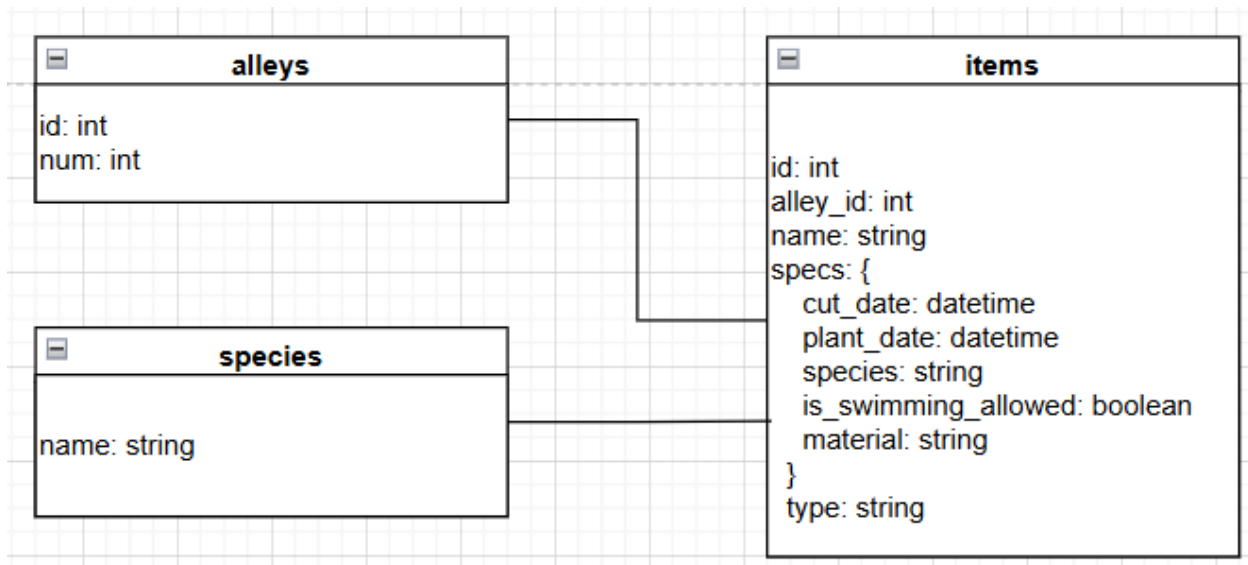


Рисунок 1 Физическая схема БД

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"
  }
]);
```

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"
      }
    }
  },
  {
    $group:
    {
      _id: "$count",
      species: {
        $push: "$$ROOT"
      }
    }
  },
  {
    $sort: {
      _id: -1
    }
  },
  {
    $limit:
    1
  },
  {
    $unwind:
    {
      path: "$species"
    }
  },
  {
    $replacewith:
    "$species"
  },
  {
    $unset: ["count", "trees"]
  }
]);

```

Результат выполнения запроса:

```

{
  "_id": {

```

```

    "$oid": "672d0eb52b729721dbd1b366"
  },
  "name": "красный клен"
}

```

2. 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.