МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И ПРОГРАММНОЙ ИНЖЕНЕРИИ

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

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

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №12 |
| Манипулирование данными в документной базе данных. |
| по курсу: Проектирование баз данных |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4932 |  |  |  | Н.С. Иванов |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2022

#### Текст 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)

#### Вариант

8. оборудование театра: роль, спектакль, реквизит (для роли /как часть костюма), название костюма, деталь костюма, размер, автор модели, дата разработки (учесть, чтобы в спектаклях идущих одновременно не использовались одинаковые объекты)

а. спектакли, в которых используются костюмы, имеющие в названии слово шут

б. костюм, в котором есть и куртка и штаны

в. автор, разработавший самый старый из костюмов

г. спектаклю, к которому разработано наибольшее число костюмов

д. автор, не разрабатывавший костюмы к «Золушке»

#### Запросы

// а. спектакли, в которых используются костюмы, имеющие в названии слово шут  
db.performance.find()  
  
db.performance.aggregate([  
 {  
 $lookup: {  
 from: 'clothes',  
 localField: 'roles.clothes.clothes\_id',  
 foreignField: '\_id',  
 as: "clothes"  
 }  
 },  
 {  
 $match: {  
 'clothes.name' : /шут/i  
 }  
 }  
])  
  
// б. костюм, в котором есть и куртка и штаны  
db.clothes.aggregate([  
 {  
 $match: {  
 $and: [  
 { 'detail.name': 'Куртка'},  
 { 'detail.name': 'Штаны'}  
 ]  
 }  
 }  
])  
  
// в. автор, разработавший самый старый из костюмов  
db.author.aggregate([  
 {  
 $lookup: {  
 from: 'clothes',  
 localField: 'clothes.clothes\_id',  
 foreignField: '\_id',  
 as: "clothes"  
 }  
 },  
 {  
 $project: {  
 oldest\_author\_clothes: { $min: "$clothes.date" },  
 name: 1  
 }  
 },  
 {  
 $group: {  
 \_id: null,  
 oldest\_clothes: { $min: '$oldest\_author\_clothes' },  
 author\_data: {  
 $push: '$$ROOT'  
 }  
 }  
 },  
 {  
 $unwind: "$author\_data"  
 },  
 {  
 $project:  
 {  
 'author\_data.name': 1,  
 'oldest\_clothes': 1,  
 'author\_data.oldest\_author\_clothes': 1,  
 is\_eq: {  
 $eq: [  
 '$author\_data.oldest\_author\_clothes',  
 '$oldest\_clothes'  
 ]  
 }  
 }  
 },  
 {  
 $match: {  
 is\_eq: true  
 }  
 },  
 {  
 $project: {  
 'author\_data.name': 1,  
 'author\_data.oldest\_author\_clothes': 1  
 }  
 }  
])  
  
db.author.find()  
  
// г. спектаклю, к которому разработано наибольшее число костюмов  
db.performance.aggregate([  
 {  
 $lookup: {  
 from: 'clothes',  
 localField: 'roles.clothes.clothes\_id',  
 foreignField: '\_id',  
 as: "clothes"  
 }  
 },  
 {  
 $project: {  
 count\_clothes: {  
 $size: '$clothes'  
 },  
 name: 1  
 }  
 },  
 {  
 $group: {  
 \_id: null,  
 max\_count: { $max: '$count\_clothes' },  
 performance\_data: {  
 $push: '$$ROOT'  
 }  
 }  
 },  
 {  
 $unwind: "$performance\_data"  
 },  
 {  
 $project: {  
 'max\_count': 1,  
 'performance\_data.name': 1,  
 'performance\_data.count\_clothes': 1,  
 'is\_eq': {  
 $eq: [  
 '$performance\_data.count\_clothes',  
 '$max\_count'  
 ]  
 }  
  
 }  
 },  
 {  
 $match: {  
 "is\_eq": true  
 }  
 }  
])  
  
// д. автор, не разрабатывавший костюмы к «Золушке»  
db.author.aggregate([  
 {  
 $lookup: {  
 from: 'performance',  
 localField: 'clothes.clothes\_id',  
 foreignField: 'roles.clothes.clothes\_id',  
 as: "performance"  
 }  
 },  
 {  
 $project: {  
 name: 1,  
 "performance.name": 1,  
 zolushka: {  
 $filter: {  
 input: "$performance.name",  
 as: "p\_name",  
 cond: {  
 $eq: [  
 '$$p\_name',  
 'Золушка'  
 ]  
 }  
 }  
 }  
 }  
 },  
 {  
 $match: {  
 zolushka: {  
 $size: 0  
 }  
 }  
 }  
])