Министерство образования и науки Российской Федерации

федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

УНИВЕРСИТЕТ ИТМО»

Факультет среднего профессионального образования

ОТЧЁТ

ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 7

по теме: Знакомство с MongoDB

по дисциплине: Основы проектирования баз данных

Специальность 09.02.07 Информационные системы и программирование

Выполнил:

студент группы Y2336

\_\_\_\_\_\_\_\_ Кропотов Н.Д.

Проверил:

\_\_\_\_\_\_\_\_ Говоров А. И.

Дата: «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2020г.

Оценка \_\_\_\_\_\_\_\_\_\_\_

Санкт-Петербург 2020

Цель работы

Цель практической работы №7:овладеть практическими навыками и умениями реализации баз данных в MongoDB.

Задание

Для выполнения работы Вам потребуется:

1. Попытаться уменьшить размер модели реляционной базы данных, с которой Вы работали.
2. Реализовать Вашу модель в MongoDB.
3. Заполнить модель данными, настолько, чтобы все запросы реализуемые Вами в пункте 2.d выдавали непустые строки.
4. Реализовать минимум 40% Ваших запросов, сделанных в работе по реляционным базам данных в MongoDB.

ПЕРЕЧЕНЬ ЗАПРОСОВ

Необходимо реализовать некоторое количество SQL-запросов.

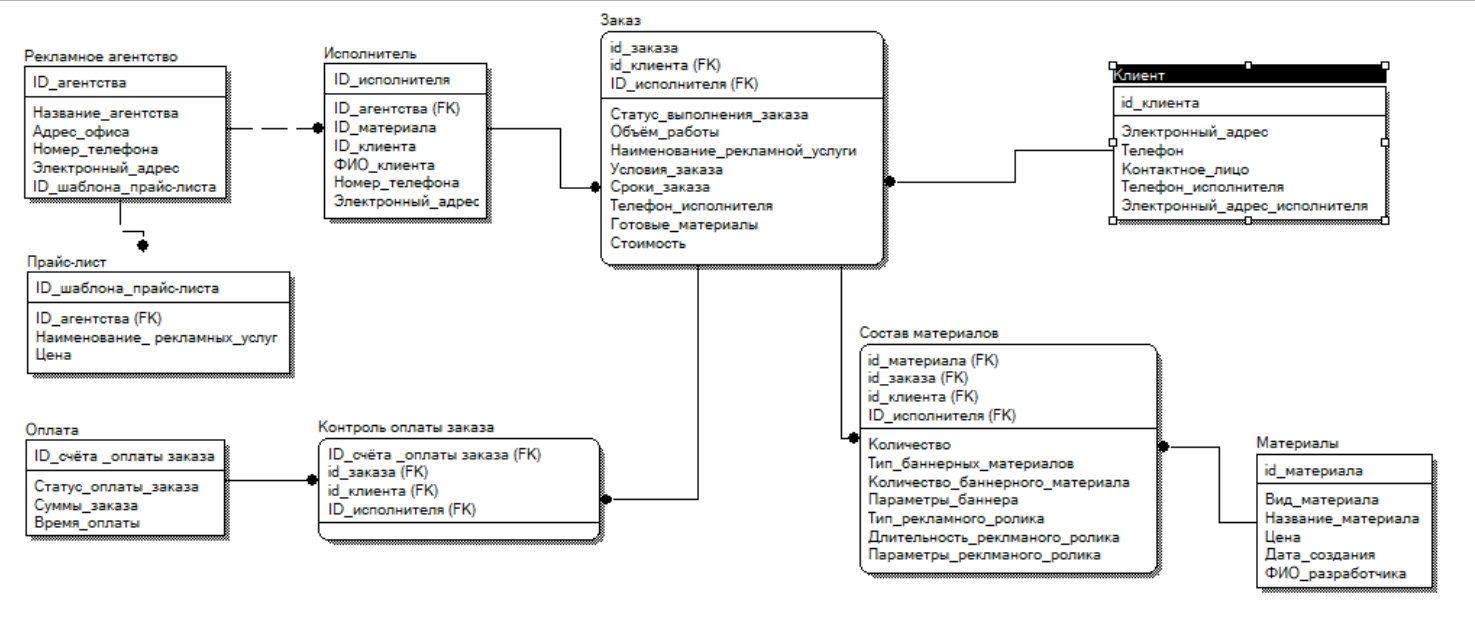
Примерный набор требуемых запросов:

* выбор значений заданных атрибутов из более, чем двух таблиц, с сортировкой – от 1 балла;
* использование условий WHERE, состоящих из более, чем одного условия – от 1 балла;
* использование функций для работы с датами – от 2 баллов;
* использование строковых функций – от 3 баллов;
* запрос с использованием подзапросов – от 2 баллов (многострочный подзапрос - от 3 баллов);
* вычисление групповой (агрегатной) функции – от 1 балла (с несколькими таблицами – от 3 баллов);
* вычисление групповой (агрегатной) функции с условием HAVING – от 2 баллов; использование предикатов EXISTS, ALL, SOME и ANY - от 4 баллов; использование запросов с операциями реляционной алгебры (объединение, пересечение и т.д.) - от 3 баллов;
* использование объединений запросов (inner join и т.д.) - от 3 баллов.

Запросов должно быть не менее 10 (30 баллов на 5, 25 баллов на 4, 20 баллов на 3) (5, 4, 3 – это оценки, а не суммы баллов в журнале).

выполнение

1. Схема физической модели БД, спроектированной с использованием CA Erwin Data Modeler представлена на рисунке №1.



*Рисунок 1 Схема инфологической модели данных БД, выполненная в среде* ***CA ERwin Process Modeler***

1. **Логи реализации модели:**

**> use reklama**

**switched to db reklama**

**> db.createCollection("agency")**

**{ "ok" : 1 }**

**> db.agency.insert({row\_agency: 1, agency\_id: 123, name\_agency: 'SAD', executor\_id: 11111, contact: {phone\_agency: 12345, mail\_agency: 'sad@gmail.com', adress\_agency: 'England'}, prise\_list\_id: 123})**

**WriteResult({ "nInserted" : 1 })**

**> db.agency.insert({row\_agency: 2, agency\_id: 456, name\_agency: 'WES', executor\_id: 22222, contact: {phone\_agency: 37389, mail\_agency: 'wes@gmail.com', adress\_agency: 'France'}, prise\_list\_id: 456})**

**WriteResult({ "nInserted" : 1 })**

**> db.agency.insert({row\_agency: 3, agency\_id: 123, name\_agency: 'SAD', executor\_id: 55555, contact: {phone\_agency: 12345, mail\_agency: 'sad@gmail.com', adress\_agency: 'England'}, prise\_list\_id: 123})**

**WriteResult({ "nInserted" : 1 })**

**> db.agency.insert({row\_agency: 4, agency\_id: 456, name\_agency: 'WES', executor\_id: 44444, contact: {phone\_agency: 56789, mail\_agency: 'wes@gmail.com', adress\_agency: 'France'}, prise\_list\_id: 456})**

**WriteResult({ "nInserted" : 1 })**

**> db.agency.insert({row\_agency: 5, agency\_id: 456, name\_agency: 'WES', executor\_id: 66666, contact: {phone\_agency: 76432, mail\_agency: 'wes@gmail.com', adress\_agency: 'France'}, prise\_list\_id: 456})**

**WriteResult({ "nInserted" : 1 })**

**> db.agency.insert({row\_agency: 6, agency\_id: 789, name\_agency: 'NAS', executor\_id: 33333, contact: {phone\_agency: 34567, mail\_agency: 'nas@gmail.com', adress\_agency: 'Russia'}, prise\_list\_id: 789})**

**WriteResult({ "nInserted" : 1 })**

**> show collections**

**agency**

**> db.createCollection("client")**

**{ "ok" : 1 }**

**> db.client.insert({client\_id: 65823, contact\_client: {person: 'Itzan Escamilla', client\_phone: 648472, client\_mail: 'itzan@gmail.com'}, contact\_executor: {executor\_phone: 423412, executor\_mail: 'connerwalton@gmail.com'}})**

**WriteResult({ "nInserted" : 1 })**

**> db.client.insert({client\_id: 31345, contact\_client: {person: 'Miguel Bernardeau', client\_phone: 313452, client\_mail: 'miguel@gmail.com'}, contact\_executor: {executor\_phone: 429884, executor\_mail: 'jeffreymason@gmail.com'}})**

**WriteResult({ "nInserted" : 1 })**

**> db.client.insert({client\_id: 44627, contact\_client: {person: 'Aron Piper', client\_phone: 424985, client\_mail: 'aron@gmail.com'}, contact\_executor: {executor\_phone: 463745, executor\_mail: 'marahaney@gmail.com'}})**

**WriteResult({ "nInserted" : 1 })**

**> db.client.insert({client\_id: 12345, contact\_client: {person: 'Omar Ayuso', client\_phone: 539866, client\_mail: 'omar@gmail.com'}, contact\_executor: {executor\_phone: 864395, executor\_mail: 'dominikadyer@gmail.com'}})**

**WriteResult({ "nInserted" : 1 })**

**> db.client.insert({client\_id: 90624, contact\_client: {person: 'Ester Exposito', client\_phone: 243567, client\_mail: 'ester@gmail.com'}, contact\_executor: {executor\_phone: 968445, executor\_mail: 'taylermccarthny@gmail.com'}})**

**WriteResult({ "nInserted" : 1 })**

**> db.client.insert({client\_id: 318405, contact\_client: {person: 'Danna Paola', client\_phone: 450386, client\_mail: 'danna@gmail.com'}, contact\_executor: {executor\_phone: 768356, executor\_mail: 'brontekim@gmail.com'}})**

**WriteResult({ "nInserted" : 1 })**

**> db.createCollection("executor")**

**{ "ok" : 1 }**

**> db.executor.insert({executor\_id: 11111, agency\_id: 123, client\_info: {client\_id: 65823, person: 'Itzan Escamilla'}, contact\_executor:{executor\_phone: 423412, executor\_mail: 'connerwalton@gmail.com'}, material\_id: 57394})**

**WriteResult({ "nInserted" : 1 })**

**> db.executor.insert({executor\_id: 22222, agency\_id: 456, client\_info: {client\_id: 31345, person: 'Miguel Bernardeau'}, contact\_executor:{executor\_phone: 429884, executor\_mail: 'jeffreymason@gmail.com'}, material\_id: 9483})**

**WriteResult({ "nInserted" : 1 })**

**> db.executor.insert({executor\_id: 44444, agency\_id: 456, client\_info: {client\_id: 44627, person: 'Aron Piper'}, contact\_executor:{executor\_phone: 463745, executor\_mail: 'marahaney@gmail.com'}, material\_id: 31486})**

**WriteResult({ "nInserted" : 1 })**

**> db.executor.insert({executor\_id: 55555, agency\_id: 123, client\_info: {client\_id: 12345, person: 'Omar Ayuso'}, contact\_executor:{executor\_phone: 864395, executor\_mail: 'dominikadyer@gmail.com'}, material\_id: 43824})**

**WriteResult({ "nInserted" : 1 })**

**> db.executor.insert({executor\_id: 33333, agency\_id: 789, client\_info: {client\_id: 90624, person: 'Ester Exposito'}, contact\_executor:{executor\_phone: 968445, executor\_mail: 'taylermccarthny@gmail.com'}, material\_id: 3174})**

**WriteResult({ "nInserted" : 1 })**

**> db.executor.insert({executor\_id: 66666, agency\_id: 456, client\_info: {client\_id: 318405, person: 'Danna Paola'}, contact\_executor:{executor\_phone: 768356, executor\_mail: 'brontekim@gmail.com'}, material\_id: 32423})**

**WriteResult({ "nInserted" : 1 })**

**> db.createCollection("prise\_list")**

**{ "ok" : 1 }**

**> db.prise\_list.insert({prise\_list\_id: 123, agency\_id: 123, service\_name: ['banner creation','video creation'], service\_price: ['300','600']})**

**WriteResult({ "nInserted" : 1 })**

**> db.prise\_list.insert({prise\_list\_id: 456, agency\_id: 456, service\_name: ['banner creation','video creation'], service\_price: ['250','700']})**

**WriteResult({ "nInserted" : 1 })**

**> db.prise\_list.insert({prise\_list\_id: 789, agency\_id: 789, service\_name: ['banner creation','video creation'], service\_price: ['400','580']})**

**WriteResult({ "nInserted" : 1 })**

**> db.createCollection("materials")**

**{ "ok" : 1 }**

**> db.material.insert({material\_id: 57394, material\_info:{type\_material: 'banner', date\_create\_material: new Date('May 12, 2019'), developer: 'Jaylen Davidson'}, price\_material: 100})**

**WriteResult({ "nInserted" : 1 })**

**> db.material.insert({material\_id: 64533, material\_info:{type\_material: 'video', date\_create\_material: new Date('March 02, 2019'), developer: 'Nella Reyna'}, price\_material: 200})**

**WriteResult({ "nInserted" : 1 })**

**> db.material.insert({material\_id: 9483, material\_info:{type\_material: 'video', date\_create\_material: new Date('August 07, 2019'), developer: 'Rayn Kelly'}, price\_material: 300})**

**WriteResult({ "nInserted" : 1 })**

**> db.material.insert({material\_id: 31486, material\_info:{type\_material: 'video', date\_create\_material: new Date('June 07, 2020'), developer: 'Nikita Kropotov'}, price\_material: 500})**

**WriteResult({ "nInserted" : 1 })**

**> db.material.insert({material\_id: 43824, material\_info:{type\_material: 'video', date\_create\_material: new Date('December 23, 2019'), developer: 'Sidney Forster'}, price\_material: 230})**

**WriteResult({ "nInserted" : 1 })**

**> db.material.insert({material\_id: 3174, material\_info:{type\_material: 'video', date\_create\_material: new Date('December 03, 2019'), developer: 'Diana Olson'}, price\_material: 160})**

**WriteResult({ "nInserted" : 1 })**

**> db.material.insert({material\_id: 32323, material\_info:{type\_material: 'banner', date\_create\_material: new Date('September 15, 2019'), developer: 'Aniela Joyce'}, price\_material: 190})**

**WriteResult({ "nInserted" : 1 })**

**> db.createCollection("payment")**

**{ "ok" : 1 }**

**> db.payment.insert({payment\_id: 543, payment\_state: 'not paid', paument\_price: 500, pay\_time: '-'})**

**WriteResult({ "nInserted" : 1 })**

**> db.payment.insert({payment\_id: 133, payment\_state: 'not paid', paument\_price: 700, pay\_time: '-'})**

**WriteResult({ "nInserted" : 1 })**

**> db.payment.insert({payment\_id: 453, payment\_state: 'not paid', paument\_price: 1200, pay\_time: '-'})**

**WriteResult({ "nInserted" : 1 })**

**> db.payment.insert({payment\_id: 756, payment\_state: 'paid', paument\_price: 1000, pay\_time: new Date('February 13, 2019')})**

**WriteResult({ "nInserted" : 1 })**

**> db.payment.insert({payment\_id: 774, payment\_state: 'paid', paument\_price: 600, pay\_time: new Date('August 15, 2019')})**

**WriteResult({ "nInserted" : 1 })**

**> db.createCollection("order")**

**{ "ok" : 1 }**

**> db.order.insert({order\_id: 30, client\_id: 31345, order\_status: 'waiting', workload: '2 weeks', service\_name: 'video creation', executor\_id: 22222, payment\_id: 756, material\_id: 9483, cost: 1000})**

**WriteResult({ "nInserted" : 1 })**

**> db.order.insert({order\_id: 40, client\_id: 44627, order\_status: 'pause', workload: '2 month', service\_name: 'video creation', executor\_id: 33333, payment\_id: 133, material\_id: 3174, cost: 1200})**

**WriteResult({ "nInserted" : 1 })**

**> db.order.insert({order\_id: 50, client\_id: 12345, order\_status: 'waiting', workload: '1 week', service\_name: 'banner creation', executor\_id: 55555, payment\_id: 453, material\_id: 43824, cost: 500})**

**WriteResult({ "nInserted" : 1 })**

**> db.order.insert({order\_id: 60, client\_id: 65823, order\_status: 'over', workload: '3 weeks', service\_name: 'banner creation', executor\_id: 66666, payment\_id: 774, material\_id: 32423, cost: 1100})**

**WriteResult({ "nInserted" : 1 })**

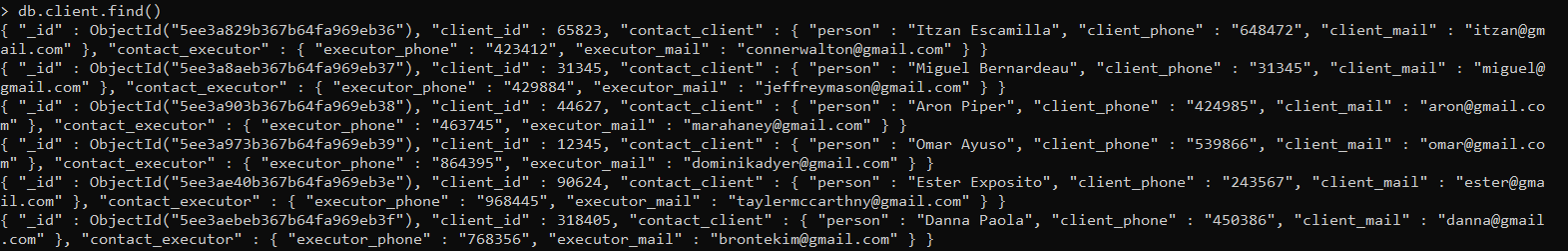
**> db.order.insert({order\_id: 20, client\_id: 90624, order\_status: 'pause', workload: '3 weeks', service\_name: 'banner creation', executor\_id: 11111, payment\_id: 543, material\_id: 57394, cost: 700})**

**WriteResult({ "nInserted" : 1 })**

1. **Запросы:**

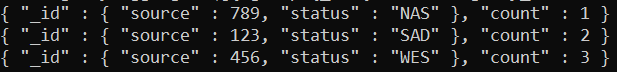
**Получить всех клиентов:**

**db.client.find()**



**Вывести все агентства и сгруппировать их по названию:**

**db.agency.aggregate([{$group:{\_id:{source:"$agency\_id",status: "$name\_agency"}, count:{$sum:1}}},{$sort:{"\_id.status": 1}}])**



**Вывести клиента с номером 65823:**

**db.client.find({client\_id: 65823})**



**Вывести ID платёжного счёта оплаченного «2019-02-13»:**

**db.payment.find({pay\_time: new Date("February 13, 2019")},{\_id:0})**



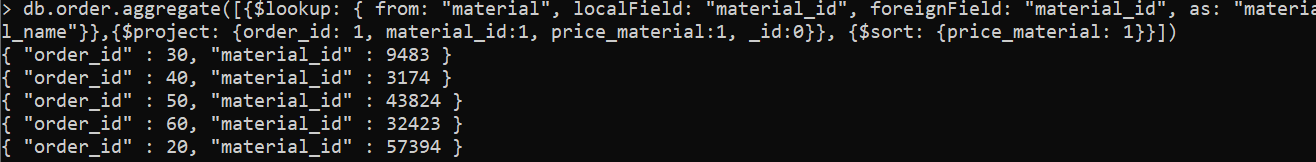
**Вывести заказ использующий материал с ID\_material = 9483:**

**db.order.find({material\_id: "9483"},{order\_id:1, material\_id:1, \_id:0})**



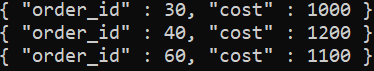
**Вывести ID заказа, ID материала:**

**db.order.aggregate([{$lookup: { from: "material", localField: "material\_id", foreignField: "material\_id", as: "material\_name"}},{$project: {order\_id: 1, material\_id:1, price\_material:1, \_id:0}}, {$sort: {price\_material: 1}}])**



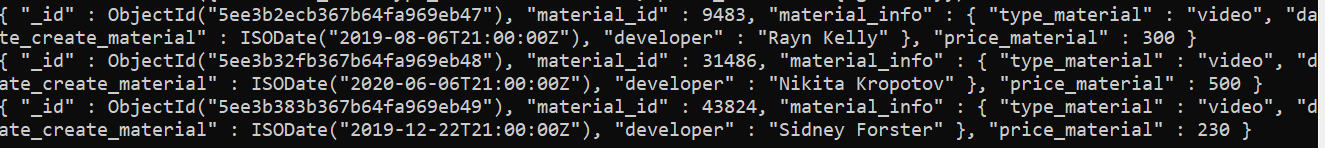
**Вывести ID заказов максимальная цена которых больше 1000:**

**db.order.find({"cost" : {$gte: 1000}}, {order\_id: 1, cost:1, \_id:0})**



**Вывести ID материала с типом «видео» И ценой больше 210:**

**db.material.find({'material\_info.type\_material': 'video', price\_material: {$gt: 210}})**



**ВЫВОД**

**В лабораторной работе №7 были получены практические навыки создания базы данных в MongoDB, заполнение их рабочими данными и выполнение запросов.**