1)

db.transactions.insertMany( [ { 'transaction\_id': 't001', 'account\_id': 'a001', 'amount': 50.00, 'currency': 'usd', 'transaction\_type': 'debit', 'description': 'Grocery Shopping', 'created\_at': new Date() }, { 'transaction\_id': 't002', 'account\_id': 'a002', 'amount': 150.00, 'currency': 'usd', 'transaction\_type': 'debit', 'description': 'Grocery Shopping', 'created\_at': new Date() }, { 'transaction\_id': 't003', 'account\_id': 'a003', 'amount': 250.00, 'currency': 'usd', 'transaction\_type': 'debit', 'description': 'Grocery Shopping', 'created\_at': new Date() }, { 'transaction\_id': 't004', 'account\_id': 'a004', 'amount': 50.00, 'currency': 'usd', 'transaction\_type': 'debit', 'description': 'Grocery Shopping', 'created\_at': new Date() }, { 'transaction\_id': 't005', 'account\_id': 'a003', 'amount': 50.00,

'currency': 'usd',

'transaction\_type': 'debit',

'description': 'Grocery Shopping',

'created\_at': new Date()

}

]

)

Starta University кому Все 12:27

db.transactions.insertOne({  
 'id': 't001',  
 'created\_at': new Date(),  
 'amount': 1005,  
 'currency': 'eur', // eur, usd  
 'sender\_id': 'u001',  
 'recipient\_id': 'u002'  
})

2) db.transactions.findOne(  
 {  
 'id': 't001'  
 },// filter  
 {'amount': 1, 'currency': 1, '\_id': 0}  
)

3) А) Задача   
 - вывести всех пользователей, кроме тех, что из Germany и Spain

Б) Задача   
 - вывести всех пользователей, кроме тех, что из Germany

db.users.find(  
 {  
 'country': {'$ne': 'Germany'}  
 } // filter  
)

db.users.find(  
 {  
 'country': {'$nin': ['Germany', 'Spain']}  
 } // filter  
)

4) Задача   
 - удалить транзакции, где сумма перевода равно нулю (0)

5) Задача   
 - отметить завершенными две транзакции с ID 't001', 't101'  
 - is\_completed (true)

db.transactions.updateMany(  
 {  
 'id': {'$in': ['t001', 't101']}  
 }, // filter  
 {  
 '$set': {  
 'is\_completed': true  
 }  
 } // action  
)

6) db.calls.insertOne({  
 'id': 'c001',  
 'created\_at': new Date(),  
   
 'duration\_secs': 3600,  
 'user\_id': 'u001',  
 'worker\_id': 'w001',  
 'direction': 'in' // out  
})

7) db.calls.updateMany(  
 {}, // filter  
 {  
 '$push': {  
 'topic': 'credit'  
 }  
 }  
)

8) db.calls.updateMany(  
 {}, // filter  
 {  
 '$pull': {  
 'topic': 'credit'  
 }  
 }  
)

9) db.calls.updateMany(  
 {}, // filter  
 {  
 '$addToSet': {  
 'topic': 'credit'  
 }  
 }  
)

10) db.calls.updateMany( // credit + premium  
 {}, // filter  
 {  
 '$addToSet': {  
 'topic': {'$each': ['credit', 'premium']}  
 }  
 }  
)

11) db.calls.updateMany( // credit + premium  
 {}, // filter  
 {  
 '$pullAll': {  
 'topic': ['credit', 'premium']  
 }  
 }  
)

12) db.calls.find(   
 {  
 'topic': 'debt'  
 }, // filter  
 {'topic': 1, '\_id': 0}  
)

13) Задача   
 - для всех входящих звонков от 5 мин  
 (1) добавить темы общения 'credit', 'debt'  
 (2) установить дата/время завершения звонка (finished\_at)  
   
   
 ОТВЕТ: прямое сообщение в ZOOM

db.calls.updateMany(   
 {  
 'direction': 'in',  
 'duration\_secs': {'$gte': 300}  
 }, // filter  
 {  
 '$set': {  
 'finished\_at': new Date()  
 },  
 '$addToSet': {  
 'topic': {'$each': ['credit', 'debt']}  
 }  
 } // action  
)

14) Задача   
 - для всех входящих и исходящих звонков до 1 часа  
 (1) увеличить счетчик "ПРОСЛУШАНО\_РАЗ" на 1  
 (2) убрать тему общения 'debt'

db.calls.updateMany(   
 {  
 'direction': {'$in': ['in', 'out']},  
 'duration\_secs': {'$lte': 3600}  
 }, // filter  
 {  
 '$inc': {  
 'listen\_count': 1  
 },  
 '$pull': {

15) Задача   
 - для всех пользователей из China и Германии и балансом больше нуля  
 (1) установить поле is\_active: true  
 (2) и добавить произвольный email-адрес  
 (email-адресов может быть несколько)

16) Задача   
 - для всех пользователей из China и Германии и балансом больше нуля  
 (1) установить поле is\_active: true  
 (2) и добавить произвольный email-адрес  
 (email-адресов может быть несколько)

Задача  
 - разблокировать всех пользователей не из China

db.users.updateMany(  
 {  
 'country': {'$in': ['China', 'Germany']},  
 'balance': {'$gt': 0}  
 }, // filter  
 {

'$set': {  
 'is\_active': true  
 },  
 '$addToSet': {  
 'email': 'user1@example.org'  
 }  
 } // action  
)

17) db.users.updateMany(  
 {  
 'country': {'$ne': 'China'}  
 }, //   
 {  
 '$unset': {  
 'is\_blocked': null  
 }  
 }  
)

18) db.users.find( // все пользователи или из Китая или нулевой баланс  
 {  
 '$or': [  
 {'country': 'China'},  
 {'balance': 0}  
 ]  
 } // filter  
)

19) Задача (1)  
  
  
 - найти пользователей из Германии  
 -- с балансом от 100  
 ИЛИ  
 -- со значением is\_active: true

**GenTech / Feb 28, 2023**

**Что такое MongoDB?**

* СУБД
* Прикладное ПО

**Что собой представляет документная модель?**

* Предлагает описывать сущности с помощью документов (объектов, ассоц/массивов)

**Принципиальные отличия MongoDB от RDBMS?**

1. Использует другой язык запросов (Query API - NoSQL)
2. Динамическая схема данных (или без схемы)
3. Позволяет хранить не только скалярные значения

‘email@example.org’

[1, 2, 3]

**Самая распространенная архитектура хранения**

ОДНА СУЩНОСТЬ (ОБЪЕКТ) == ОДИН ДОКУМЕНТ

**Обязательные поля (мета) для любой сущности (объекта)**

1. Первичные ключ
   1. Ненулевой
   2. Уникальный
   3. Можно использовать UUID
2. Дата/время добавления

**Коллекция (документов)** - аналог таблицы - способ объединения документов по категориям

$in проверка принадлежности

$nin проверка отсутствия

**Основные операторы работы с массивами и множествами**

$push добавить элемент в **массив**

$pull удалить элемент из массива

$addToSet добавить элемент в **множество**

$each (для $addToSet и $push) добавить список значений в массив или множество

**Логические операторы**

$and логическое И

$or логическое ИЛИ

$nor отрицание ИЛИ

$not логическое НЕТ (отрицание)

**Справочная информация**

* webhooks