

GenTech: Summary / Aug 16, 2023

Базовые термины

- рефакторинг (кода) - переработка (перевыпуск) кода
- стек (программный) - набор технологий (инструментов) для создания проекта
- алгоритм - идея (способ) решения задачи
- программа - реализация алгоритма

Масштабирование (проекта)

- увеличение выч/мощностей по мере роста нагрузки

Виды

1. Горизонтальное (количество)
2. Вертикальное (качество)

```
SELECT * FROM Customers WHERE Country IN ('Germany', 'USA') --  
Country='Germany' -- OR -- Country='USA'
```

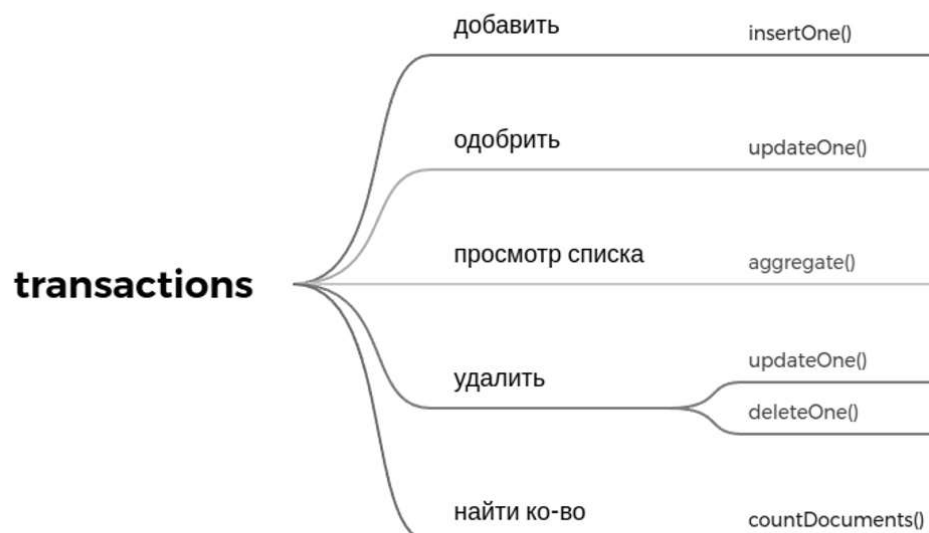
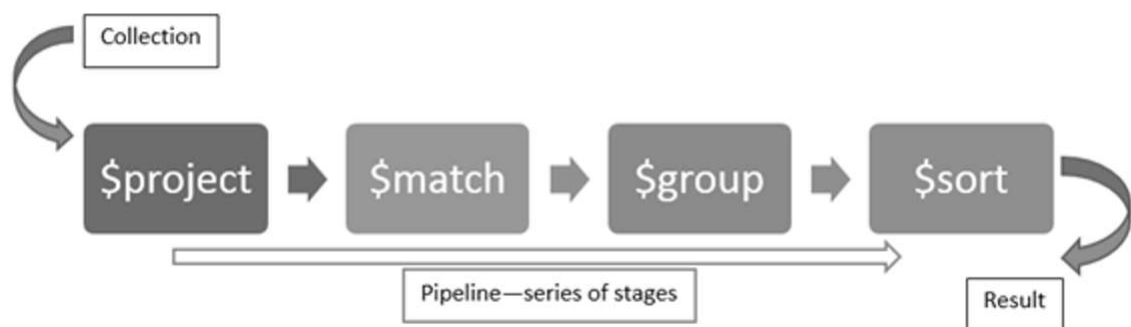
Примеры запросов

```
// Вывести юзеров из USA db.users.aggregate([ { $match: { country: 'USA' } } ])
```

```
// Вывести незаблокированных юзеров db.users.find( { is_blocked: { $ne: true } } // filter )
```

```
// Вывести имена юзеров из China db.users.find( { country: 'China' }, //
filter { fullname: 1, _id: 0 } // projection )
```

```
// Увеличить стоимость на 0.5% для товаров с ценой от 10 до 300 EUR
db.products.updateMany( { price: { $gte: 10, $lte: 300 } }, // filter {
$mul: { price: 1.005 } } // action )
```



Задача. Перечислить поля (англ.) для сущности **ТРАНЗАКЦИЯ**

```
_id sender_id recipient_id amount currency is_completed
```

Задача. Добавить пять транзакций

```
db.transactions.insertMany([ { _id: 1, sender_id: 1, recipient_id: 2,
amount: 20, currency: 'eur', is_completed: true }, { _id: 2, sender_id:
2, recipient_id: 1, amount: 80, currency: 'usd', is_completed: true }, {
_id: 3, sender_id: 3, recipient_id: 2, amount: 10, currency: 'eur' }, {
_id: 4, sender_id: 1, recipient_id: 1, amount: 180, currency: 'usd',
is_completed: true }, { _id: 5, sender_id: 1, recipient_id: 2, amount:
48, currency: 'eur' } ])
```

Задача. Вывести все транзакции

```
db.transactions.aggregate()
```

Задача. Вывести **USD**-транзакции

```
db.transactions.aggregate([ { $match: { currency: 'usd' } } ])
```

Задача. Вывести одну самую большую **EUR**-транзакцию

```
db.transactions.aggregate([ { $match: { currency: 'eur' } }, { $sort: {
amount: -1 } }, { $limit: 1 } ]);
```

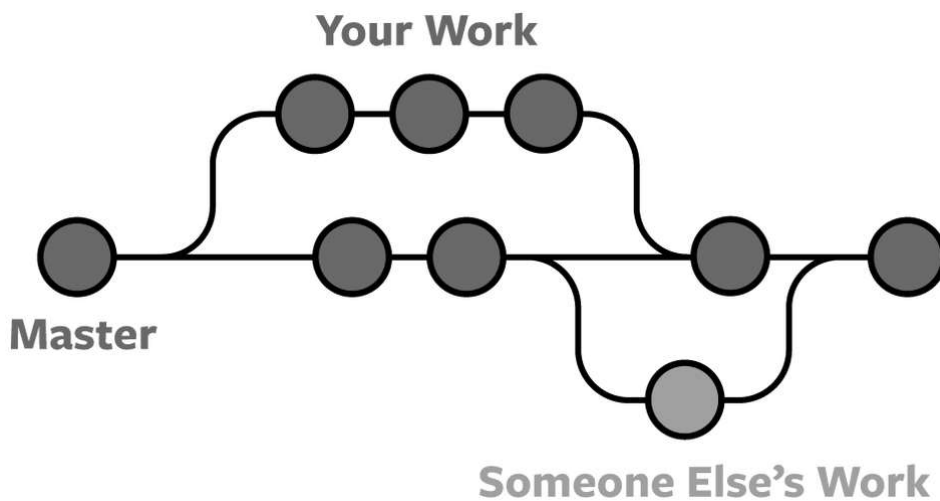
Задача. Вывести транзакцию, которая на втором месте по размеру среди транзакций до **1000** EUR

```
db.transactions.aggregate([ { $match: { currency: 'eur', amount: { $lte:
1000 } } }, { $sort: { amount: -1 } }, { $skip: 1 }, { $limit: 1 } ])
```

Основы командной в Git/GitHub

1. Ветки (ветвление)
 - a. Основные команды
 - b. Стратегии ветвления
2. Модели совместной разработки
 - a. Описание
 - b. Общий порядок работы

Ветвление в Git



Стратегии ветвления

1. GitFlow
 - a. `master/main/stable/production` - long-term
 - b. `develop/current` - long-term
 - c. `bugfix1/issue88/login/payments` - short-term
2. GitHub Flow
 - a. `main`
 - b. short-term

Основные команды

- `git branch название_ветки` создать ветку
- `git branch -m новое_название` переименовать текущую ветку
- `git checkout название_ветки` переключиться на другую ветку
- `git checkout -b новая_ветка` создать и переключиться на новую ветку
- `git branch -D название_ветки` удалить ветку
- `git merge название_ветки` слияние веток
- `git push -u origin branch` выгрузить на GitHub
- `git push origin --delete branch` удалить ветку на GitHub

Модели совместной разработки

1. С доступом на запись к репо (Shared-repo model)

- а. лучше подходит для небольших организованных команд с приватными проектами (репо)

2. Без доступа на запись к репо (Forks and Pull model)

- а. лучше подходит для проектов с открытым исходным кодом и неопределенным кругом разработчиков

Общий порядок (схемы) работы в режиме с доступом на запись к репо

1. Открыть имеющийся репо (или создать новый)
2. Добавить членов команды
3. Клонировать репо (скачать на компьютер)
 - а. `git clone ссылка_на_репо`

4. Открыть репо (как проект в любимом редакторе)