



ОНЛАЙН-ОБРАЗОВАНИЕ

# Онлайн-образование



**Не забыть включить запись!**







# Меня хорошо видно && слышно?

Ставьте ☐, если все хорошо  
Напишите в чат, если есть проблемы







# Тема: Базовые возможности MongoDB, CRUD, фильтры

Курочкин Константин  
Ведущий администратор баз данных  
«Мединдекс»  
[https://t.me/konstantin\\_kurochkin](https://t.me/konstantin_kurochkin)



# Правила вебинара



Активно участвуем



Задаем вопрос в чат или голосом



Off-topic обсуждаем в Slack



Вопросы вижу в чате, могу ответить не сразу

# Маршрут вебинара

Основные понятия



CRUD



find, sort, skip, limit



Рефлексия

# Цели вебинара | После занятия вы сможете

**1** Создать базу данных

**2** Написать CRUD запросы

**3** Использовать фильтры и ограничения в запросах




















# Смысл | зачем вам это уметь

Для работы с популярным  
направлением в БД – NoSQL MongoDB



# Знакомство с MongoDB (db-engines.com)

397 systems in ranking, November 2022

| Rank     |          |                                                                                         | DBMS                                                                                                    | Database Model                                                                                                 | Score    |          |          |
|----------|----------|-----------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------|----------|----------|----------|
| Nov 2022 | Oct 2022 | Nov 2021                                                                                |                                                                                                         |                                                                                                                | Nov 2022 | Oct 2022 | Nov 2021 |
| 1.       | 1.       | 1.                                                                                      | Oracle                 | Relational, Multi-model     | 1241.69  | +5.32    | -31.04   |
| 2.       | 2.       | 2.                                                                                      | MySQL                  | Relational, Multi-model     | 1205.54  | +0.17    | -5.98    |
| 3.       | 3.       | 3.                                                                                      | Microsoft SQL Server  | Relational, Multi-model     | 912.51   | -12.17   | -41.78   |
| 4.       | 4.       | 4.                                                                                      | PostgreSQL             | Relational, Multi-model     | 623.16   | +0.44    | +25.88   |
| 5.       | 5.       | 5.                                                                                      | MongoDB                | Document, Multi-model       | 477.90   | -8.33    | -9.45    |
| 6.       | 6.       | 6.                                                                                      | Redis                  | Key-value, Multi-model      | 182.05   | -1.33    | +10.55   |
| 7.       | 7.       |  8.    | Elasticsearch                                                                                           | Search engine, Multi-model  | 150.32   | -0.74    | -8.76    |
| 8.       | 8.       |  7.  | IBM Db2                                                                                                 | Relational, Multi-model   | 149.56   | -0.10    | -17.96   |
| 9.       | 9.       |  11. | Microsoft Access                                                                                        | Relational                                                                                                     | 135.03   | -3.14    | +15.79   |
| 10.      | 10.      |  9.  | SQLite               | Relational                                                                                                     | 134.63   | -3.17    | +4.83    |

# Где можно использовать ???

- хранение и регистрация событий;
- системы управления документами и контентом;
- электронная коммерция;
- игры;
- данные мониторинга, датчиков;
- мобильные приложения;
- хранилище операционных данных веб-страниц (например, хранение комментариев, рейтингов, профилей пользователей, сеансы пользователей);
- хранение геоданных.



# Стек MEAN

**M** MongoDB

**E** Express.js

**A** Angular.js

**N** Node.js

JavaScript-like синтаксис



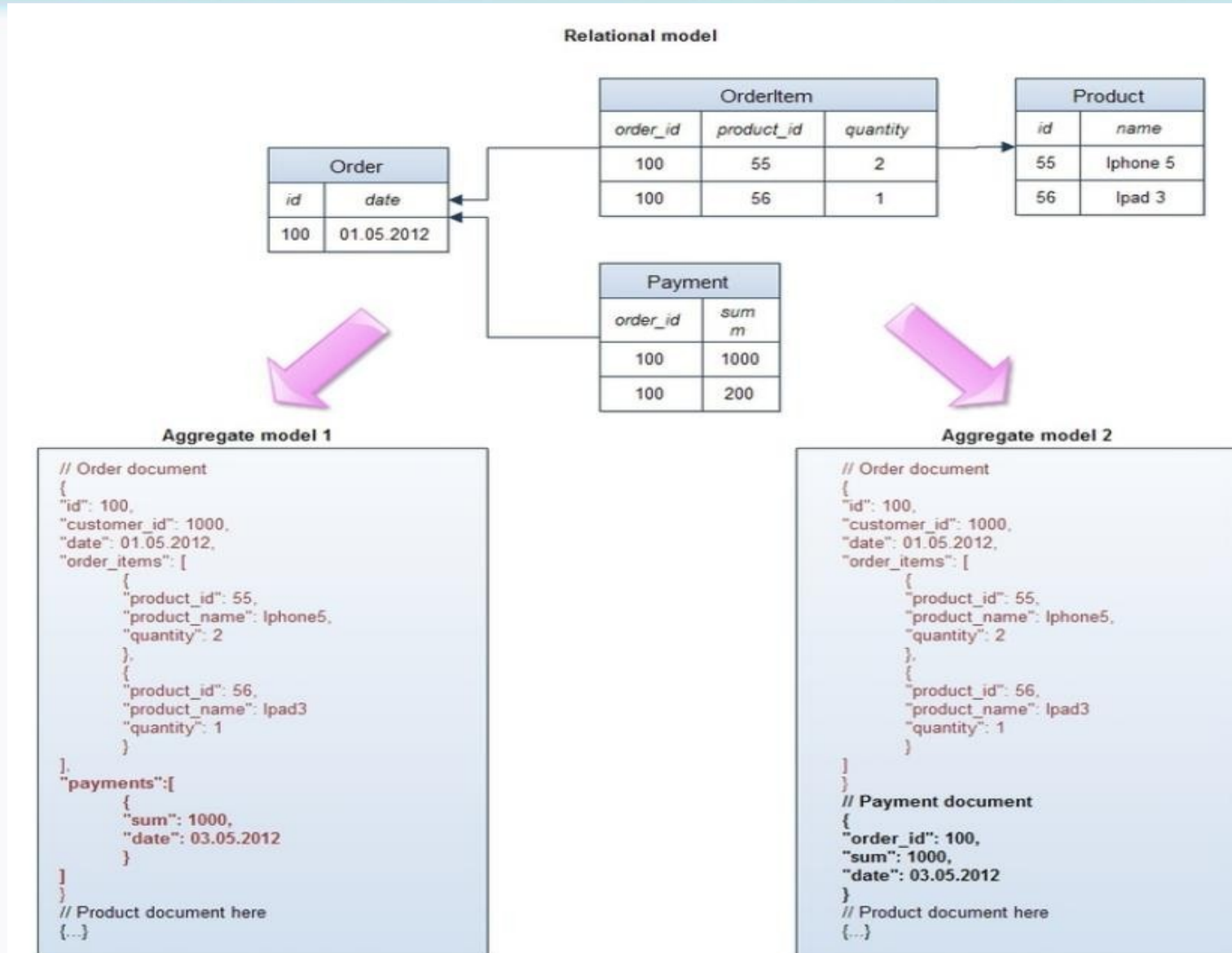
# Варианты использования

## Самый известный пример: интернет магазин

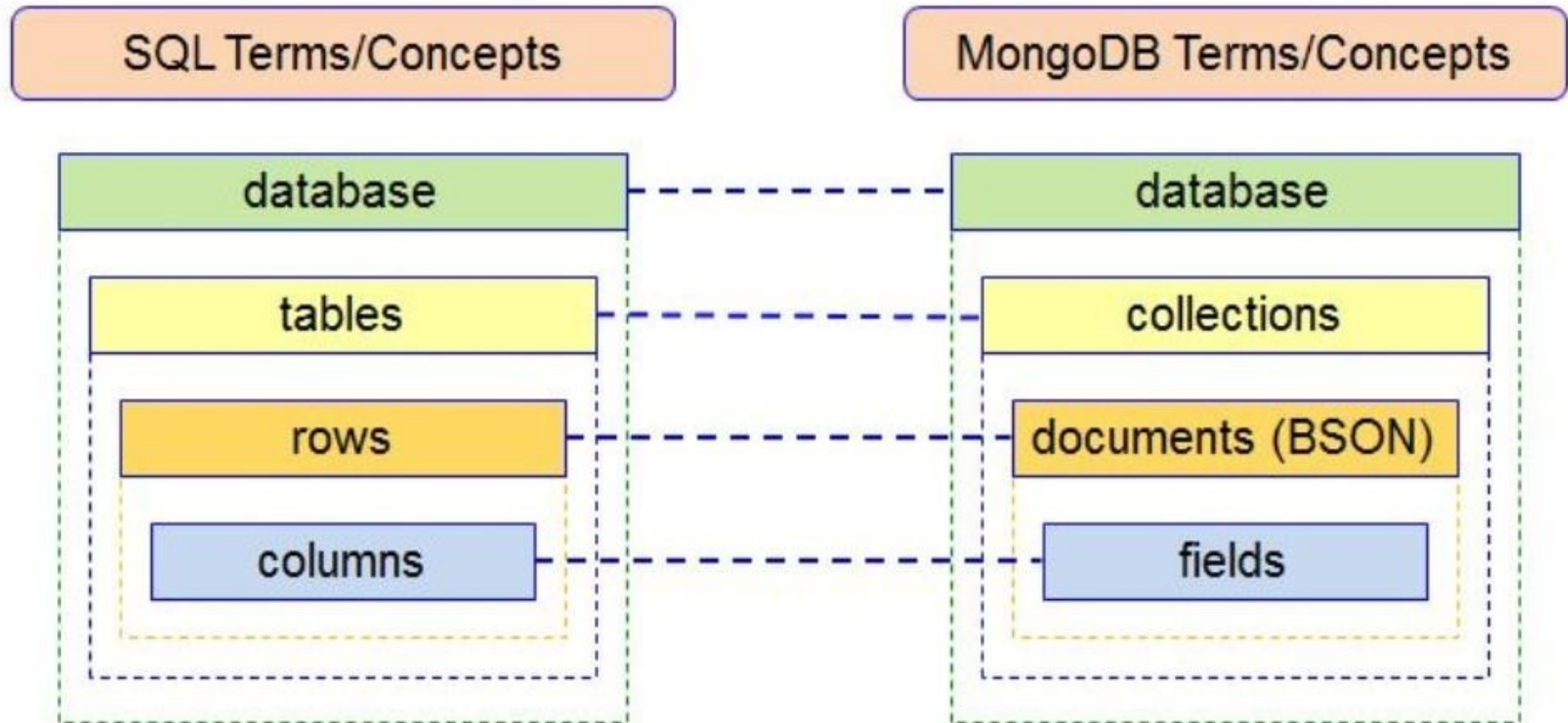
- > 1000 брендов
- > 500 категорий
- > 100 000 товаров
- > у каждого свои характеристики



# SQL / noSQL



# SQL / noSQL

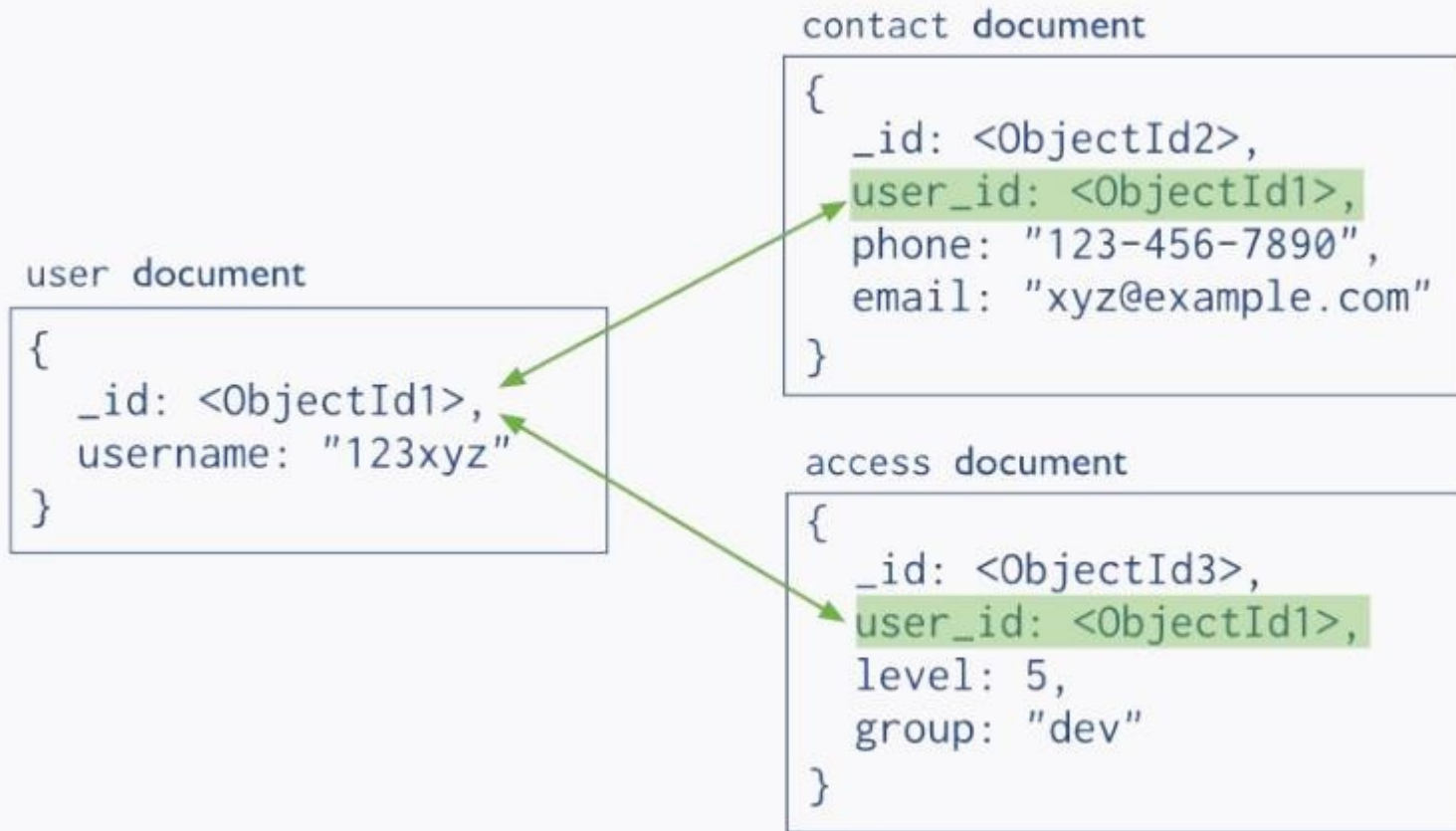




# Основные понятия



# Основные понятия



```
{
  "name": "Bill",
  "surname": "Gates",
  "age": "48",
  "company": {
    "name" : "microsoft",
    "year" : "1974",
    "price" : "300000"
  }
}
```



# Особенности MongoDB

1. Документоориентированная БД.
2. Без схемы базы данных.
3. Документы хранятся в формате BSON (binary JSON).
4. Размер 1 документа – max 16 Мб.
5. В одном документе допускается до 100 уровней вложенностей.
6. Горизонтально масштабируется из коробки.
7. Open-source лицензия.
8. Регистрозависимость:  
    {"age" : "28"}  
    {"Age" : 28}

# Типы данных

- **String**: строковый тип данных, используется кодировка UTF-8)
- **Array**: тип данных для хранения массивов элементов
- **Binary data**: тип для хранения данных в бинарном формате
- **Boolean**: булевый тип данных, хранящий логические значения TRUE или FALSE
- **Date**: хранит дату в формате времени Unix
- **Double**: числовой тип данных для хранения чисел с плавающей точкой
- **Integer**: используется для хранения целочисленных значений
- **JavaScript**: тип данных для хранения кода javascript

# Типы данных

- **Min key/Max key:** используются для сравнения значений с наименьшим/наибольшим элементов BSON
- **Null:** тип данных для хранения значения Null
- **Object:** строковый тип данных, используется для встроенных документов
- **ObjectID:** тип данных для хранения id документа
- **Regular expression:** применяется для хранения регулярных выражений
- **Symbol:** тип данных, идентичный строковому, используется для тех языков, в которых есть специальные символы
- **Timestamp:** применяется для хранения времени



# Полезные материалы

Документация <https://docs.mongodb.com/manual/>

MySQL и MongoDB <https://habr.com/ru/post/322532/>

[Should I Use MongoDB or PostgreSQL?](https://medium.com/geekculture/should-i-use-mongodb-or-postgresql-ba2c1bb8b768)

<https://medium.com/geekculture/should-i-use-mongodb-or-postgresql-ba2c1bb8b768>

# Установка MongoDB

1. Из коробки: <https://docs.mongodb.com/manual/installation/>
2. Docker: <https://phoenixnap.com/kb/docker-mongodb>
3. Kubernetes:  
<https://www.mongodb.com/blog/post/runningmongodb-ops-manager-in-kubernetes>
4. Скрипт от Алексея Цикунова:  
[https://github.com/erlong15/mongo\\_k8s\\_deploy](https://github.com/erlong15/mongo_k8s_deploy)
5. GUI MongoDB Compass:  
<https://www.mongodb.com/try/download/compass>

# CRUD

C - insert  
R - find  
U - update  
D - delete



# Создать базу данных / коллекцию

<https://www.mongodb.com/basics/create-database>

<https://docs.mongodb.com/manual/core/databases-and-collections/>

`show dbs`

`use dbname`

`show collections / show tables`

`db.createCollection("name")`

`db.stats()`

# Insert

<https://docs.mongodb.com/manual/reference/method/db.collection.insert/>

`db.collection.insert()`

`db.collection.insertOne()`

`db.collection.insertMany()`

# Find

<https://docs.mongodb.com/manual/reference/method/db.collection.find/>

```
db.collection.find()
```

```
db.collection.find().pretty()
```

```
db.collection.findOne()
```

Основные операции:

<https://docs.mongodb.com/manual/reference/operator/query/>



# sort, skip, limit

Сортировать:

<https://docs.mongodb.com/manual/reference/method/cursor.sort/index.html>

Игнорировать:

<https://docs.mongodb.com/manual/reference/method/cursor.skip/>

Ограничить:

<https://docs.mongodb.com/manual/reference/method/cursor.limit/>

# Index, aggregate

<https://docs.mongodb.com/manual/indexes/>

```
db.collection.createIndex( { name: 1 | -1 } )
```

<https://docs.mongodb.com/manual/aggregation/>

```
db.collection.aggregate()
```

# Update

<https://docs.mongodb.com/manual/reference/method/db.collection.update/>

`db.collection.update(query, update, options)`



# Delete

<https://docs.mongodb.com/manual/reference/method/db.collection.deleteOne/>

<https://docs.mongodb.com/manual/reference/method/db.collection.deleteMany/>

`db.collection.deleteOne()`

`db.collection.deleteMany()`

`db.collection.drop()`

`db.dropDatabase()`

# Домашнее задание

Необходимо:

1. Установить MongoDB в VM в GCP
2. Заполнить данными в какой-либо предметной области, например интернет-магазин
3. Написать несколько запросов на выборку, обновление и удаление данных

Сдача ДЗ осуществляется в виде миниотчета в markdown в гите.

Задание повышенной сложности\*

Создать индексы и сравнить производительность.

# Рефлексия




1. Как добавить запись
2. Как выбрать значение по фильтру
3. Как отсортировать документы





Заполните, пожалуйста,  
опрос о занятии по ссылке в чате  
<https://otus.ru/polls/52766/>





# **До новых встреч!**

# **Приходите на следующие занятия**

Курочкин Константин  
Ведущий администратор баз данных  
«Мединдекс»  
[https://t.me/konstantin\\_kurochkin](https://t.me/konstantin_kurochkin)