

# Введение в NOSQL.

## NOSQL vs RDBMS

Курс «NoSQL»



Проверить, идет ли запись

# Меня хорошо видно & слышно?





# Гурьянов Дмитрий

Senior Developer, SKL Group

- Опыт работы в IT – 10 лет
- Опыт преподавания – 4 года
- Специализация – системы автоматизации продаж (CRM)

 @guryanovda

<http://dmgur.ru>  
[students@dmgur.ru](mailto:students@dmgur.ru)

# Правила вебинара



Активно  
участвуем



Задаем вопрос  
в чат или голосом



Вопросы вижу в чате,  
могу ответить не сразу

# Правила вебинара



Off-topic обсуждаем  
в закрытом чате в Telegram

Кто еще не присоединился,  
сканируйте QR код :)



# Маршрут вебинара

Знакомство

О курсе

История развития

NoSQL vs RDBMS

Итоги занятия

Рефлексия



# Цели вебинара

К концу занятия вы сможете

1. Познакомиться с преподавателями и одногруппниками
2. Рассмотреть программу курса
3. Понять, как проходит обучение в OTUS
4. Разобраться в разных типах СУБД



# Знакомство

---



# Команда курса



**Алексей Железной**

ClickHouse



**Гурьянов Дмитрий**

Cassandra,  
Tarantool,  
проектный модуль



**Николай Лапшин**

модуль Document Store



**Константин Войта**

Couchbase,  
etcd,  
Consul,  
RabbitMQ,  
Hadoop File System,  
NATS



**Дмитрий Кириллов**

Redis,  
Elasticsearch,  
Opensearch,  
Prometheus. Grafana



**Наталья Титова**

NoSQL в облаках



**Екатерина Гребцова**

Графовые базы данных (Neo4J)



# Расскажите о себе

Напишите, пожалуйста, в чат или скажите голосом



Как вас зовут?



Какой опыт в IT?



Какие ожидания  
от курса?

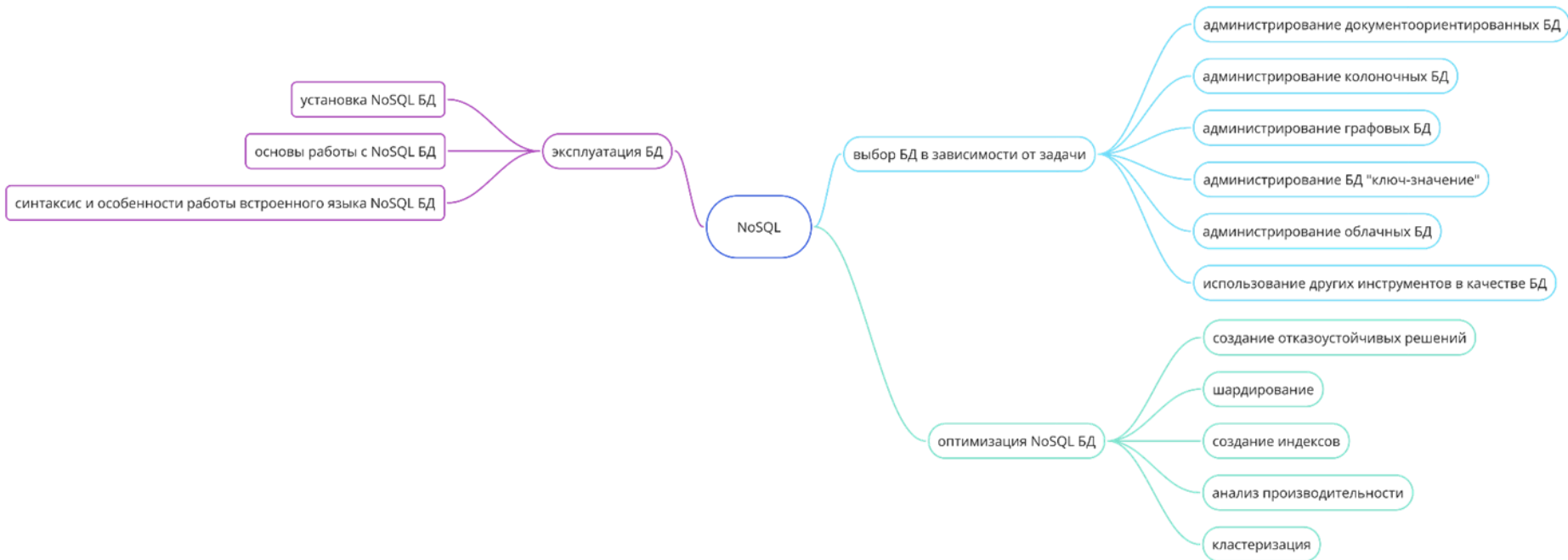


Заполните информацию  
в разделе «О себе»  
в личном кабинете

**0 курсе**

---

# Карта компетенций



# Карта курса



**СТАРТ**

Введение в NOSQL

Document Store

Колоночные БД

Другие NoSQL

Графовые

Key-Value

Другие NoSQL

NoSQL в облаках

Проект

**ФИНИШ**



# Практика на курсе

## 14

домашних заданий

### Инструменты,

с которыми поработаете:

MongoDB, Couchbase, Cassandra,  
ClickHouse, Elasticsearch, Redis, Neo4j, Kafka,  
RabbitMQ, Hadoop, Prometheus, Grafana,  
Tarantool и др.



### Вы научитесь:

- устанавливать и эксплуатировать популярные NoSQL решения от выделенных виртуальных серверов до больших шардированных кластеров и облачных провайдеров
- Сможете различать основные типы NoSQL БД и выбирать оптимальное решение в зависимости от вашей задачи
- Освоите синтаксис и особенности работы встроенного языка каждой NoSQL БД
- Сможете оптимизировать медленные запросы

# Необходимые инструменты и ПО

**Для обучения вам необходимо**

- Личная учетка в <https://github.com/>
- Оформлять ДЗ в readme.md (Markdown) [https://github.com/adam-p/markdown-here/wiki/Markdown- Cheatsheet](https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet)



# Необходимые инструменты и ПО

## Какие инструменты понадобятся?

*Для запуска баз данных можно использовать любое облако и работать по ssh*

- Яндекс Облако <https://cloud.yandex.ru/>
- VKCloud <https://mcs.mail.ru/>
- SberCloud <https://sbercloud.ru/ru>
- TimeWeb
- Selectel <https://selectel.ru/services/cloud/servers/>

ИЛИ

- Docker, docker-compose
- Docker Desktop <https://docs.docker.com/desktop/>
- docker-compose <https://docs.docker.com/compose/install/>





# Результат от курса

## В рамках проектной работы вы...

- Разработаете и протестируете высоконагруженный отказоустойчивый кластер на одной из NoSQL БД;
- Проведете масштабное исследование скорости обработки запросов на нескольких БД;
- Будете работать над личным проектом



# Как будем учиться?



## Вебинары

**Понедельник, четверг, 20:00**

Запись и материалы выкладывают, как правило, на следующий день после вебинара.



## Домашние задания

**1-2 ДЗ в неделю**

Дедлайна нет, кроме окончания курса :)  
Типовой срок проверки: 2-3 дня.



## Чат в Telegram

**Задавайте вопросы, обменивайтесь инсайтами.**



# Лайфхаки



Если не успеваете охватить весь материал, сделайте упор на темы, которые для вас наиболее актуальны



Смело задавайте вопросы, так материал лучше усваивается



Старайтесь применять полученные знания на практике, найдите стажировку, если у вас есть такая возможность



Регулярно выполняйте ДЗ, т.к. наверстать пропуски тяжело



**Больше полезной информации  
в нашем курсе **OTUS Heroes**  
или в [курсе о том, как учатся взрослые](#)**

# Немного о мотивации

Приготовьтесь к волнообразной реакции от «ух ты как интересно» до «куда я попал?»

В такие моменты вспоминайте про цели обучения, которые вас мотивируют.

Это нормально :)



А если вы еще не определились с целью, загляните в [наш гайд по целеполаганию](#)

# Отзывы

Вебинары во многом адаптивны.

Мы наблюдаем за вашим обсуждением в Telegram, берем в работу идеи из ваших отзывов на занятия и домашние задания и становимся лучше!



**Пожалуйста, заполняйте опросы после занятий и ДЗ,  
мы читаем все ваши сообщения :)**



Ссылки на **опросы по занятиям** преподаватели присылают в чат в конце вебинара



Ссылки на **опросы по ДЗ** приходят в чат проверки ДЗ после того, как ваша работа зачтена

# История развития

---



**Какие способы хранения данных вы знаете?**



# Хранение в файле

ПЛЮСЫ

МИНУСЫ

# Хранение в файле

## ПЛЮСЫ

- Простота
- Эффективная запись append
- Дешево с точки зрения ресурсов - минимальные расходы на RAM и CPU

## МИНУСЫ

- Сложно организовать эффективный поиск/удаление/обновление данных
- Необходимо решать проблемы с data races
- Нет поддержки хранения целостности данных и транзакций
- Нет готовых механизмов репликации и шардирования
- При решении этих проблем мы потеряем простоту

# Выбор конца прошлого века - ORACLE

## Нужные навыки для DBA

- **установить настроить Oracle Server из дистрибутива (Solaris)**
  - `init.ora` or `listener.ora`
- **создать**
  - табличные пространства
  - схемы
  - пользователей
  - резервное копирование и восстановление мониторинг
  - борьба с неоптимальными запросами

# Чего не ожидали от DBA

- выбор оптимальной СУБД или другой технологии хранения и обработки данных
- высокая доступность и горизонтальная масштабируемость, соответственно
- репликация и шардирование, балансировщики нагрузки
- docker, k8s
- OLTP, OLAP оптимизации - BI
- знание предметной области и прикладной архитектуры
- хорошее знание инфраструктуры
- знание ОС
- загрузка и выгрузка данных - ETL
- миграция данных между разными СУБД



# NoSQL vs RDBMS

---

# Реляционные базы данных RDBMS (РСУБД)

**Реляционная СУБД** (реляционная система управления базами данных) — это тип базы данных, который основан на **реляционной модели данных**, предложенной Эдгаром Коддом в 1970 году. Название "реляционная" происходит от английского слова "**relation**", что в контексте реляционной модели означает **таблицу** (а не "отношение" в бытовом смысле).

Название "реляционная" происходит от термина "**relation**", который в математике означает **отношение**. В контексте реляционной модели:

- **Таблица** (relation) — это математическое отношение, представляющее собой набор кортежей (строк).
- Каждая строка — это элемент отношения, а столбцы — атрибуты.

Эдгар Кодд использовал математическую теорию множеств и теорию отношений для описания структуры данных, что и дало название "реляционная модель".



# Реляционные базы данных RDBMS (РСУБД)

## Особенности:

- Наличие статичной схемы данных
- Более сложное горизонтальное масштабирование
- Поддержка транзакций и целостности данных ACID
- Поддержка ассоциаций (через вторичные ключи)
- Возможность строить сложные запросы к данным (SQL)





# NoSQL. Немного истории

Первоначальным замыслом были современные системы управления базами данных в веб масштабе

Движение началось в начале 2009 года.

Йохан Оскарссон в то время разработчик last.fm вновь ввёл термин NoSQL, когда он организовал мероприятие для обсуждения “распределённых нереляционных баз данных с открытым исходным кодом”



# Основная идея

Предоставить механизм для хранения и извлечения данных, которые моделируются средствами, отличными от табличных отношений, используемых в реляционных базах данных.

# Основная идея

Предоставить механизм для хранения и извлечения данных, которые моделируется средствами, отличными от табличных отношений, используемых в реляционных базах данных.

Что пытаются достичь этим подходом:

- Простота проектирования
- Горизонтальное масштабирование на кластеры машин
- Преодоление объектно-реляционного разрыва



# Что означает термин NoSQL

- NoSQL = Not SQL

Все что не относится к типам реляционных БД



- Not only SQL

как общее название технологии

# Зачем это нужно?

РСУБД могут не удовлетворять потребностям:

- По гибкости - данные должны быть нужного типа
- По скорости - нельзя просто что-то куда-то

NoSQL – обладает узкой специализацией

- Идеальна для каких то специальных задач
- Проигрывает РСУБД в каких-то общих задачах

# Примеры разных типов баз

## Реляционные БД

- MySQL
- PostgreSQL
- Oracle
- Microsoft SQL Server

## NoSQL БД

- MongoDB (документные)
- Cassandra (столбчатые)
- Redis (ключ-значение)
- Neo4j (графовые)
  
- + Специальные типы

# NoSQL Базы данных

## Особенности:

- Частичная поддержка SQL или отсутствие ее
- Гибкая схема данных или отсутствие жесткой структуры
- Простое горизонтальное масштабирование
- Различные модели данных, такие как ключ- значение, документоориентированные, столбцовые и графовые
- Обычно отсутствует поддержка транзакций и целостности данных
- Подходит для случаев, когда необходимо обрабатывать большие объемы данных с высокой нагрузкой



**Куда все это в итоге развилось?**



<https://db-engines.com/en/ranking>

<https://nosql-database.org/>



# Типы NoSQL баз данных

Ключ-значение — под каждым ключом что-то лежит, пока не запросим — не узнаем что именно

Key	Value
"defaultCity"	"Moscow"
"India"	"(Hindi: Bhārat), officially the Republic of India (Hindi: Bhārat Gaṇarājya)"
"US"	"{population: 332 278 200, capital: Washington, D.C}"

Redis

DynamoDB

Memcached

Tarantool



# Типы NoSQL баз данных

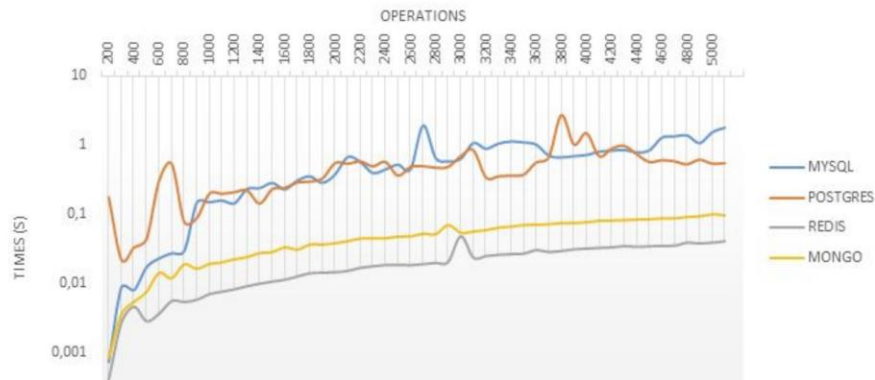
Документно-ориентированная БД – аналог баз типа ключ-значение, но в качестве значений используются «документы», т.е. объекты в определенном формате JSON/XML

mongoDB  
couchDB  
Elasticsearch



# Сравнение с РСУБД

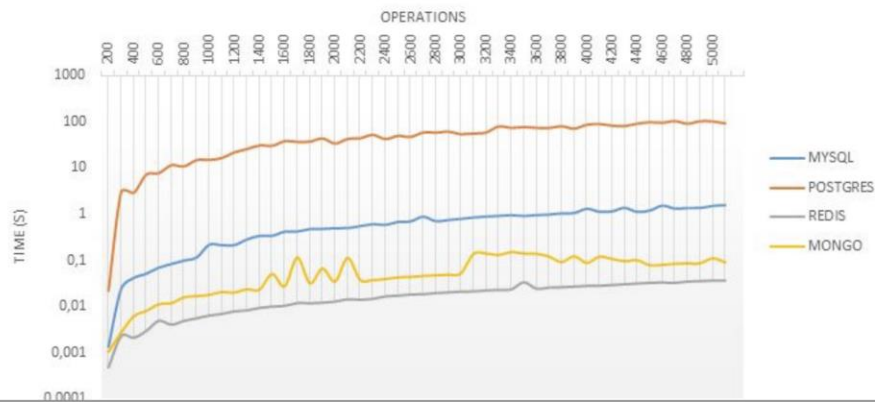
Insert comparison



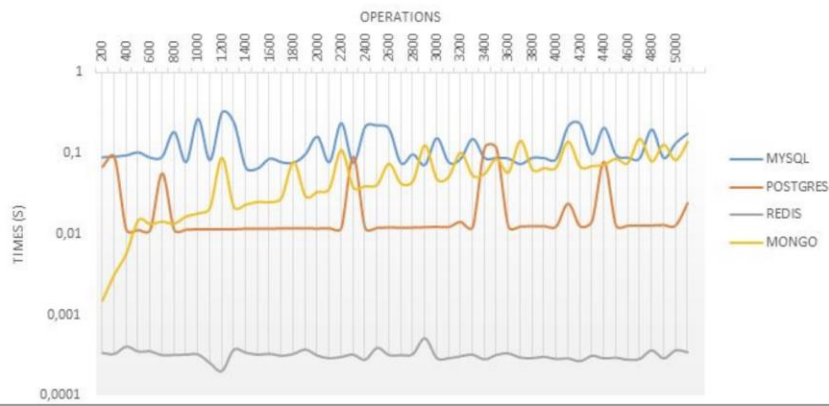
Select comparison



Update comparison



Delete comparison



# Сравнение с РСУБД

Одиночные операции CRUD в базах NoSQL выполняются гораздо быстрее

Но РСУБД может выполнять гораздо больше операций

It depends



# Типы NoSQL баз данных

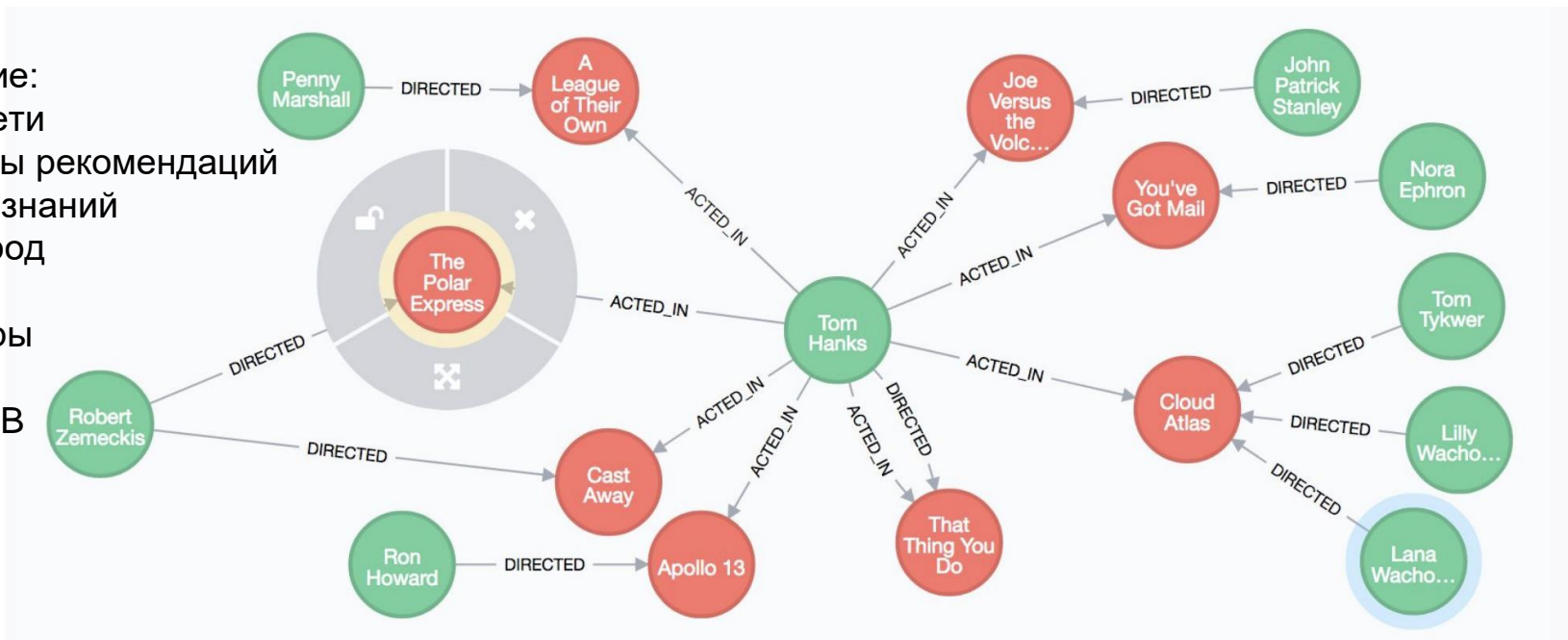
Графовые БД - единицы хранения: узлы и ребра (связи)

Запрос - обход данных от узла к узлу по рёбрам на заданную глубину

Применение:

- Соц. Сети
- Сервисы рекомендаций
- Графы знаний
- АнтиФрод

- Примеры
- Neo4j
- OrientDB
- Titan



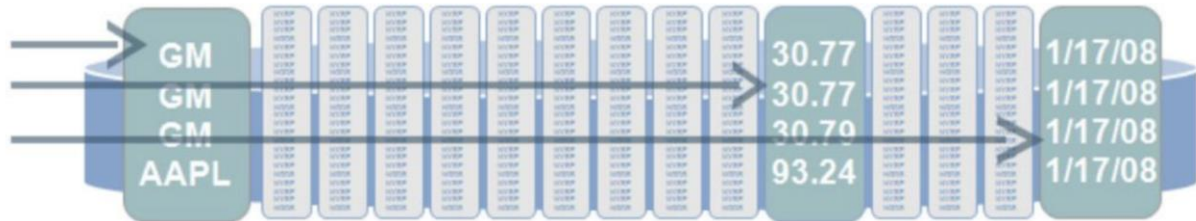
# Типы NoSQL баз данных

Колоночные БД – данные хранятся в ячейках, сгруппированных в колонки

Hbase  
Cassandra  
Google BigTable

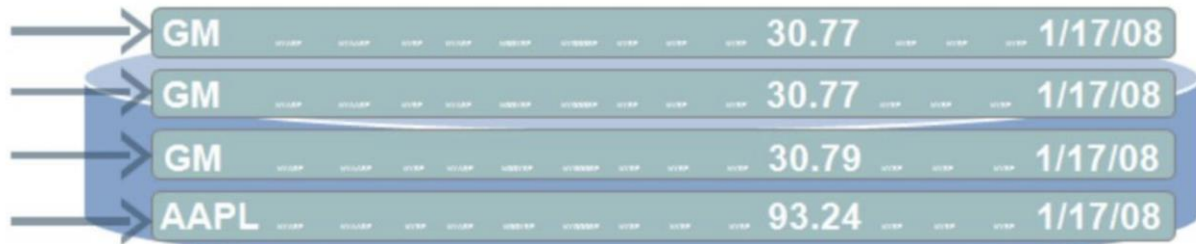
Хранение по  
столбцам

Чтение 3х столбцов



Хранение по  
строкам

Чтение всех столбцов



# Сравнение с РСУБД

	SQL	NoSQL
Способ хранения данных	строгая типизация	можно хранить что угодно
Доступ к данным	SQL запросы (стандарт), ответы в виде плоской таблицы, данных определенного типа	данные по определенному ключу, свои способы работы с данными
Место хранения данных	в идеале на одном сервере, хотя существуют кластеры равноправных серверов	Масса серверов, главное указать принцип, по которому выбираются сервера для хранения данных
Надежность	Все данные надежно связаны между собой	Приоритет по скорости, а не надежности,
Масштабируемость	Вертикальная	Вертикальная/горизонтальная



# Вопросы



если есть вопросы



если вопросов нет

---



# Маршрут вебинара

Знакомство

О курсе

История развития

NoSQL vs RDBMS

Итоги занятия

Рефлексия





**Отправьте в чат эмодзи, который отражает ваше настроение после занятия**



**Продолжите высказывание: теперь я умею/знаю...**



**Что планируете использовать в работе?**

# Приглашаем на следующий вебинар: «САР теорема» – 6.03



Ссылка на вебинар будет  
в ЛК за 15 минут



Материалы  
к занятию в ЛК —  
можно изучать



Обязательный материал  
обозначен красной лентой

# Заполните, пожалуйста, опрос о занятии

---

Мы читаем все ваши сообщения  
и берем их в работу 🖋️ ❤️

**You are our  
OTUS heroes**

