

# NoSQL

Длительность курса: 136 академических часов

# 1 Введение в NOSQL

## 1 Введение в NOSQL. NOSQL vs RDBMS

### Цели занятия:

объяснить разницу между NoSQL и RDBMS решениями;  
сравнить NoSQL и RDBMS;  
проанализировать историю NoSQL;  
рассмотреть программу курса;  
развернем VM в GCP

### Краткое содержание:

NoSQL и RDBMS;  
история NoSQL;  
программа курса;  
работа в облаке Google

---

## 2 CAP теорема

### Цели занятия:

объяснить глубину дилеммы на примере проблемы византийских генералов;  
описать алгоритм консенсуса.

### Краткое содержание:

CAP теорема;  
CP системы. RAFT, PAXOS;  
PA системы. GOSSIP, CasPAXOS.

### Домашние задания

#### 1 CAP теорема

Цель: В результате выполнения ДЗ вы научитесь работать с гитом.

Необходимо написать к каким системам по CAP теореме относятся перечисленные БД и почему: MongoDB, MSSQL, Cassandra.

ДЗ сдается ссылкой на гит, где расположен миниотчет в маркдауне.

### 1 Базовые возможности mongodb

#### Цели занятия:

сможет развернуть БД mongo разными способами; писать простейшие запросы к данным.

#### Краткое содержание:

понятие коллекции, документов и полей;  
CRUD-операции;  
фильтрация по полям;  
sort, skip, limit;  
запросы по документам.

#### Домашние задания

##### 1 MongoDB

Цель: В результате выполнения ДЗ вы научитесь разворачивать MongoDB, заполнять данными и делать запросы.

Необходимо:

- установить MongoDB одним из способов: VM, докер;
- заполнить данными;
- написать несколько запросов на выборку и обновление данных

Сдача ДЗ осуществляется в виде миниотчета.

---

---

---

---

Задание повышенной сложности\*  
создать индексы и сравнить производительность.

---

2 **Кластерные возможности mongodb**

**Цели занятия:**

шардировать в mongodb: mongos;  
правильно выбрать ключ шардирования.

**Краткое содержание:**

варианты репликации;  
Replica set;  
концепция кворума;  
балансировка;  
подходы к шардированию в БД.

---

3 **Map-reduce & aggregation framework**

**Цели занятия:**

строить сложные запросы;  
писать свои функции.

**Краткое содержание:**

концепция map-reduce;  
Зачем нужен map-reduce в mongodb;  
концепция pipeline;  
структура и синтаксис агрегации;  
\$match;  
стадия \$group;  
стадия \$lookup;  
работа с геоданными.

---

## 4 Оптимизация производительности mongodb

### Цели занятия:

построить индексы;  
оптимизировать CRUD;  
профилирование запросов

### Краткое содержание:

виды индексов в mongodb;  
профилирование;  
дисковые движки;  
GridFS;  
производительность кластеров.

### Домашние задания

#### 1 MongoDB 2

Цель: В результате выполнения ДЗ вы настроите реплицирование и шардирование, аутентификацию в кластере и проверите отказоустойчивость.

Необходимо:

- построить шардированный кластер из 3 кластерных нод( по 3 инстанса с репликацией) и с кластером конфига(3 инстанса);
- добавить балансировку, загрузить данными, выбрать хороший ключ шардирования, посмотреть как данные перебалансируются между шардами;
- поронять разные инстансы, посмотреть, что будет происходить, поднять обратно. Описать что произошло.

\* настроить аутентификацию и многоуровневый доступ;

Формат сдачи - readme с описанием алгоритма действий, результатами и проблемами.

---

## Цели занятия:

объяснить основы работы с Couchbase, архитектуру и возможности.

## Краткое содержание:

Основы архитектуры  
Отказоустойчивый кластер  
Анализ производительности

## Домашние задания

### 1 Кластер Couchbase

Цель: Построение отказоустойчивого кластера Couchbase

Развернуть кластер Couchbase  
Создать БД, наполнить небольшими тестовыми данными  
Проверить отказоустойчивость

---

## Цели занятия:

объяснить основы работы с Elasticsearch, архитектуру и возможности.

## Краткое содержание:

основные модели использования Elasticsearch; архитектура и компоненты Elasticsearch; происхождение "ELK стек" и его актуальность; установка и настройка Elasticsearch; создание и настройка индексов; типы запросов Elasticsearch и особенности их применения; Kibana - основы интерфейса и использования.

## Домашние задания

### 1 Знакомство с Elasticsearch

Цель: Научитесь разворачивать ES в AWS и использовать полнотекстовый нечеткий поиск

Развернуть Instance ES – желательно в AWS  
Создать в ES индекс, в нём должно быть обязательное поле text типа string  
Создать для индекса pattern  
Добавить в индекс как минимум 3 документа желательно со следующим содержанием:  
«моя мама мыла посуду а кот жевал сосиски»  
«рама была отмыта и вылизана котом»  
«мама мыла раму»  
Написать запрос нечеткого поиска к этой коллекции документов ко ключу «мама ела сосиски»  
Расшарить коллекцию postman (желательно сдавать в таком формате)  
Прислать ссылку на коллекцию

## 1 Clickhouse

### Цели занятия:

описать основы работы с Clickhouse, архитектура и возможности.

### Краткое содержание:

когда Clickhouse полезен, а когда не стоит его использовать;  
основные движки таблиц/БД Clickhouse и их особенности;  
движок MergeTree и его специфика;  
использование совместно с Kafka и PostgreSQL  
протоколы доступа (http, tcp и базовые библиотеки);  
установка и настройка

### Домашние задания

#### 1 ClickHouse

Необходимо, используя tutorial  
<https://clickhouse.tech/docs/ru/getting-started/tutorial/> :  
- развернуть БД;  
- выполнить импорт тестовой БД;  
- выполнить несколько запросов и оценить скорость выполнения.

\* развернуть дополнительно одну из тестовых БД  
<https://clickhouse.com/docs/en/getting-started/example-datasets> , протестировать скорость запросов

\*\* развернуть Кликхаус в кластерном исполнении, создать распределенную таблицу, заполнить данными и протестировать скорость по сравнению с 1 инстансом

Дз сдается в виде миниотчета.

---

2 **Cassandra:  
Distributed Key  
Value,  
Architecture**

**Цели занятия:**

описать основы работы с Cassandra, архитектуру и возможности.

**Краткое содержание:**

CAP

Data Partition

Data Replication

Consistency Level

Inconsistency resolution

System architecture diagram

---

3 **Cassandra:  
System  
Components**

**Цели занятия:**

компоненты Cassandra.

**Краткое содержание:**

Coordinator

Token ring

Handling failures

Write path

Read Path

Primary Key

Column Family

---

## 4 Cassandra: CQL and Java, Scala API

### Цели занятия:

использование языка Кассандры + Scala API

### Краткое содержание:

Keyspace  
DDL: Create/Drop table  
Inserts  
Schema Evolution  
Spark Cassandra  
DataStax

### Домашние задания

- 1 Масштабирование и отказоустойчивость Cassandra. Часть 1. - Домашнее Задание

Цель: Подготовить среду и развернуть Cassandra кластер для дальнейшего изучения возможностей масштабирования и восстановления Cassandra кластеров.

Необходимо:

- развернуть Kubernetes кластер в облаке или локально (используя "ПРЕРЕКВИЗИТЫ.docx" из материалов);
  - поднять 3 узловой Cassandra кластер на Kubernetes (используя "How to Run Cassandra on Azure Kubernetes Service (AKS), part1.pdf" из материалов);
  - нагрузить кластер при помощи Cassandra Stress Tool (используя "How to use Apache Cassandra Stress Tool.pdf" из материалов).
-

5 **Cassandra:  
monitoring and  
problem solving**

**Цели занятия:**

мониторинг Кассандры

**Краткое содержание:**

Graphana  
Prometheus  
Influx  
JMX  
Beans

**Домашние задания**

- 1 Масштабирование и отказоустойчивость Cassandra. Часть 2 - Домашнее Задание

Цель: В результате выполнения ДЗ вы изучите сторонние возможности восстановления Cassandra кластеров, замерите производительность вашего кода.

Необходимо:

- забэкапить и восстановить используя 3DSnap ваш 3 узловой Cassandra кластер на Kubernetes из прошлого ДЗ(используя "How to Run Cassandra on Azure Kubernetes Service (AKS), part2.pdf" из материалов);
  - выбрать два на ваш вкус драйвера к Cassandra и сравнить их по производительности и потреблению ресурсов (используя "Apache Cassandra with JPA.pdf" из материалов).
- 

6 **Cassandra:  
company cases**

**Цели занятия:**

реальные кейсы использования Кассандры

**Краткое содержание:**

company cases  
NVidia  
Ozon  
Russian Post

## 1 Redis. Часть 1

### Цели занятия:

охарактеризовать преимущества и недостатки NoSQL баз данных по сравнению с реляционными;  
составить стратегии кеширования RDMBS с использованием Redis.

### Краткое содержание:

области применения;  
внутренние структуры;  
очереди, транзакции.

---

## 2 Redis. Часть 2

### Цели занятия:

охарактеризовать преимущества и недостатки NoSQL баз данных по сравнению с реляционными;  
описать случаи использования NoSQL;  
составить стратегии кеширования RDMBS с использованием Redis;  
применять Redis.

### Краткое содержание:

кластеризация;  
redis sentinel;  
redis cluster.

### Домашние задания

#### 1 Redis

Необходимо:

- сохранить большой жсон (~20МБ) в виде разных структур - строка, hset, zset, list;
- протестировать скорость сохранения и чтения;
- предоставить отчет.

---

---

---

Задание повышенной сложности\*  
настроить редис кластер на 3х нодах с отказоустойчивостью, затюнить таймоуты

---

## 3 etcd

### Цели занятия:

познакомиться с etcd и понять спектр ее применения.

### Краткое содержание:

алгоритм Raft;  
отличия etcd от прочих аналогичных решений;  
основные моменты установки и настройки кластера etcd;  
бэкап и репликация кластера.

---

## 4 Consul

### **Цели занятия:**

настроить DNS с healthcheck на базе консула.

### **Краткое содержание:**

установка;  
настройка;  
кластеризация;  
федерация;  
бэкап;  
service discovery;  
DNS.

---

## 5 Foundation db

### **Цели занятия:**

научиться работать с foundation db

### **Краткое содержание:**

foundation db - обзор и архитектура

## 1 Нео4j, часть 1

### Краткое содержание:

особенности графовых баз данных;  
использование графовых баз данных в продакшене;  
тюнинг графовой БД.

### Домашние задания

#### 1 Использование neo4j

Цель: 1. Научиться представлять данные в neo4j  
2. Научиться использовать Cypher для манипуляции данными в neo4j

1. Взять 4-5 популярных туроператора.
  2. Каждый туроператор должен быть представлен в виде ноды neo4j
  3. Взять 10-15 направлений, в которые данные операторы предоставляют путевки.
  4. Представить направления в виде связки нод: страна - конкретное место
  5. Взять ближайшие к туристическим локациям города, в которых есть аэропорты или вокзалы и представить их в виде нод
  6. Представить маршруты между городами в виде связей. Каждый маршрут должен быть охарактеризован видом транспорта, который позволяет переместиться между точками.
  7. Написать запрос, который бы выводил направление (со всеми промежуточными точками), который можно осуществить только наземным транспортом.
-

### Краткое содержание:

основные юзкейсы для графовой БД;  
архитектура Neo4J.

### Домашние задания

#### 1 Оптимизация запросов

Цель: 1. Научиться смотреть план запросов  
2. Создание индексов

1. Составить план запроса из предыдущего домашнего задания
2. Добавить индексы для оптимизации запроса
3. Еще раз посмотреть план запроса и убедиться, что индексы позволили оптимизировать запрос

## 1 Tarantool

### Цели занятия:

строить высоконапряженные решения на базе tarantool.

### Краткое содержание:

основы работы с СУБД tarantool;  
архитектура, возможности.

### Домашние задания

#### 1 Работа с tarantool

Необходимо написать на тарантуле биллинг реального времени облачной системы. Должны быть хранимые процедуры:

- добавление денег на баланс;
- изменение расхода денег в секунду.

Когда баланс становится равным нулю, тарантул по http должен сделать GET-запрос на какой-либо внешний url, где передать userID пользователя, у которого кончились деньги (запрос на отключение виртуальных машин). Этот вызов должен происходить как можно быстрее после окончания денег на счете.

Для реализации рекомендуется использовать библиотеку expirationd.

Использовать шардинг на основе vshard.

---

## 2 RabbitMQ

### Цели занятия:

конфигурировать rabbitMQ кластер

### Краткое содержание:

протокол AMQP;  
задача и функциональность брокера сообщений;  
особенности кластера rabbitMQ;  
запуск кластера;  
проверка отказоустойчивости.

---

## 3 Kafka

### Цели занятия:

Объяснить архитектуру kafka, плюсы и минусы;  
Сравнить в другими брокерами;  
Познакомиться с Kafka APIs;

### Краткое содержание:

kafka - обзор и архитектура;  
kafka Топики. Партишены и Реплики;  
основные операции на кластере;  
понимание кластерных метрик и того как их интерпретировать;  
понимание Producer API Kafka;  
понимание Consumer API Kafka;  
понимание Streams API Kafka;  
возможность интеграции Kafka с внешними системами данных с использованием Kafka Connect.

---

4 s3

**Цели занятия:**

понять, что такое S3  
увидеть реализации S3  
рассмотреть дополнительные возможности версий S3

**Краткое содержание:**

протокол S3;  
CERN;  
фичи и ограничения s3: версионирование и другие.

---

5 Hadoop File System

**Цели занятия:**

Рассмотрим основную идею и архитектуру  
распределенной файловой системы  
Форматы данных и их особенности + знакомство с  
фичами Hive SQL доступа

**Краткое содержание:**

Hadoop File System  
форматы данных  
SQL + Hive  
bash + Java/Scala

---

6 Prometheus.  
Grafana

**Цели занятия:**

prometheus;  
grafana;  
alertManager;  
promQL.

**Краткое содержание:**

метрики Prometheus;  
PromQL;  
service Monitors;  
demo.

## 1 Google BigQuery

### Цели занятия:

описать характеристики bigQuery - уникального аналитического хранилища данных, которое доступно только в Google Cloud.

### Краткое содержание:

обзор NoSQL сервисов управления структурированными данными в Google Cloud Platform; демонстрация уникальных возможностей Google BigQuery.

---

## Цели занятия:

Объяснить архитектуру Cosmos DB;  
Научиться пользоваться Cosmos DB APIs;  
Показать основные фишки Cosmos DB;

## Краткое содержание:

azure Cosmos DB - обзор и архитектура;  
azure Cosmos DB - глобальное распределение данных и доступность;  
azure Cosmos DB - пропускная способность, масштабируемость, RU/sec;  
обзор SQL API в Azure Cosmos DB;  
обзор Cassandra API в Azure Cosmos DB;  
обзор MongoDB API в Azure Cosmos DB;  
обзор Gremlin API в Azure Cosmos DB;  
обзор Table API в Azure Cosmos DB;  
выбор подходящего API (SQL, Cassandra, MongoDB, Gremlin, Table) для Azure Cosmos DB;  
добавление данных в Azure Cosmos DB и доступ к ним;  
храняемые процедуры, транзакции и отслеживание изменений;  
оптимизация производительности Azure Cosmos DB с помощью стратегий секционирования и индексирования.

## Домашние задания

### 1 Облака

Необходимо:

- одну из облачных БД заполнить данными (любыми из предыдущих дз);
- протестировать скорость запросов.

---

---

---

Задание повышенной сложности\*

сравнить 2-3 облачных NoSQL по скорости загрузки данных и времени выполнения запросов.

---

### 3 AWS DynamoDB

#### **Цели занятия:**

изучить основы работы AWS DynamoDB

#### **Краткое содержание:**

основы работы с AW DynamoDB;  
архитектура и возможности.

## 1 Выбор проектной работы

### Цели занятия:

выбрать и обсудить тему проектной работы;  
спланировать работу над проектом;  
ознакомиться с регламентом работы над проектом.

### Краткое содержание:

правила работы над проектом и специфика проведения итоговой защиты;  
требования к результату проекта и итоговой документации.

### Домашние задания

#### 1 Проектная работа

Цель: защитить проект и получить рекомендации экспертов.

презентация проектов перед комиссией;  
вопросы и комментарии по проектам.

студенты без собственного проекта могут присутствовать на защите и принять участие в обсуждении проектов своих коллег.

---

## 2 Консультация по проектной работе

### Цели занятия:

получить ответы на вопросы по проекту, ДЗ и по курсу.

### Краткое содержание:

вопросы по улучшению и оптимизации работы над проектом;  
затруднения при выполнении ДЗ;  
вопросы по программе.

---

**3 Защита  
проектной  
работы**

**Цели занятия:**

защитить проект и получить рекомендации экспертов.

**Краткое содержание:**

презентация проектов перед комиссией;  
вопросы и комментарии по проектам.

---

**4 Подведение  
итогов курса**

**Цели занятия:**

узнать, как получить сертификат об окончании курса, как взаимодействовать после окончания курса с OTUS и преподавателями, какие вакансии и позиции есть для выпускников (опционально - в России и за рубежом) и на какие компании стоит обратить внимание.

**Краткое содержание:**

1. организационные вопросы;
2. рынок вакансий по направлению;
3. статистика курса и вопросы по курсу.