

# PostgreSQL

Курс предназначен для разработчиков и администраторов, готовых освоить работу с PostgreSQL

Длительность курса: 148 академических часов

## 1 Введение

- |       |   |   |
|-------|---|---|
| 1     | <b>Реляционные базы, история и место в современном мире</b> | <p>отличать SQL и NoSQL;<br/>представлять сложность выбора той или иной технологии СУБД;<br/>чуть лучше понимать, чем мы будем заниматься на этом курсе.</p>  |
| <hr/> |   |   |
| 2     | <b>SQL и реляционные СУБД. Введение в PostgreSQL</b>        | <p>объяснить основу реляционной модели данных;<br/>объяснить назначение языка SQL и его основные конструкции;<br/>иметь представление о основных реляционных СУБД;<br/>понимать разницу в уровнях изоляции транзакций</p> <p>Домашние задания</p> <p>1 Работа с уровнями изоляции транзакции в PostgreSQL</p> <p>Цель: - научиться работать с Google Cloud Platform на уровне Google Compute Engine (IaaS)<br/>- научиться управлять уровнем изоляции транзакции в PostgreSQL и понимать особенность работы уровней read committed и repeatable read</p> <p>- создать новый проект в Google Cloud Platform, например postgres2020-&lt;ууууммдд&gt;, где ууууммдд год, месяц и день вашего рождения (имя проекта должно быть уникально на уровне GCP)<br/>- дать возможность доступа к этому проекту пользователю postgres202005@gmail.com с ролью Project Editor<br/>- далее создать инстанс виртуальной машины Compute Engine с дефолтными параметрами<br/>- добавить свой ssh ключ в GCE metadata<br/>- зайти удаленным ssh (первая сессия), не забывая про ssh-add</p> |

- поставить PostgreSQL
- зайти вторым ssh (вторая сессия)
- запустить везде psql из под пользователя postgres
- выключить auto commit
- сделать в первой сессии новую таблицу и наполнить ее данными
- create table persons(id serial, first\_name text, second\_name text);
- insert into persons(first\_name, second\_name) values('ivan', 'ivanov');
- insert into persons(first\_name, second\_name) values('petr', 'petrov');
- commit;
- посмотреть текущий уровень изоляции: show transaction isolation level
- начать новую транзакцию в обеих сессиях с дефолтным (не меняя) уровнем изоляции
- в первой сессии добавить новую запись
- insert into persons(first\_name, second\_name) values('sergey', 'sergeev');
- сделать select \* from persons во второй сессии
- видите ли вы новую запись и если да то почему?
- завершить первую транзакцию - commit;
- сделать select \* from persons во второй сессии
- видите ли вы новую запись и если да то почему?
- завершите транзакцию во второй сессии
- начать новые но уже repeatable read транзакции - set transaction isolation level repeatable read;
- в первой сессии добавить новую запись
- insert into persons(first\_name, second\_name) values('sveta', 'svetova');
- сделать select \* from persons во второй сессии
- видите ли вы новую запись и если да то почему?
- завершить первую транзакцию - commit;
- сделать select \* from persons во второй сессии
- видите ли вы новую запись и если да то почему?
- завершить вторую транзакцию
- сделать select \* from persons во второй сессии
- видите ли вы новую запись и если да то почему?
- остановите виртуальную машину но не удаляйте ее

### 3 **Физический уровень PostgreSQL**

иметь представление об устройстве PostgreSQL;  
 знать процессы PostgreSQL и структуру памяти;  
 знать как PostgreSQL работает с данными на физическом уровне.

### 4 **Установка PostgreSQL**

разворачивать кластер PostgreSQL различными способами;  
 объяснить возможности и ограничения каждого способа;  
 выбирать наиболее оптимальный способ в зависимости от задачи.

Домашние задания

#### 1 Установка и настройка PostgreSQL в контейнере Docker

Цель: - создавать дополнительный диск для уже существующей виртуальной машины, размечать его и делать на нем файловую систему

- переносить содержимое базы данных PostgreSQL на дополнительный диск
- переносить содержимое БД PostgreSQL между виртуальными машинами
- установить PostgreSQL в Docker контейнере
- настроить контейнер для внешнего подключения

#### 1 вариант:

- создайте виртуальную машину с Ubuntu 18.04 LTS (bionic) в GCE типа n1-standard-1 в default VPC в любом регионе и зоне, например us-central1-a
- поставьте на нее PostgreSQL через `sudo apt`
- проверьте что кластер запущен через `sudo -u postgres pg_lsclusters`
- зайдите из под пользователя postgres в psql и сделайте произвольную таблицу с произвольным содержимым
 

```
postgres=# create table test(c1 text);
postgres=# insert into test values('1');
\q
```
- остановите postgres например через `sudo -u postgres pg_ctlcluster 10 main stop`
- создайте новый standard persistent диск GKE через Compute Engine -> Disks в том же регионе и зоне что GCE инстанс размером например 10GB
- добавьте свежесозданный диск к виртуальной машине - надо зайти в режим ее редактирования и дальше выбрать пункт `attach existing disk`
- проинициализируйте диск согласно инструкции и подмонтировать файловую систему, только не забывайте менять имя диска на актуальное, в вашем случае это скорее всего будет `/dev/sdb` - <https://www.digitalocean.com/community/tutorials/how-to-partition-and-format-storage-devices-in-linux>
- сделайте пользователя postgres владельцем `/mnt/data` - `chown -R postgres:postgres /mnt/data/`
- перенесите содержимое `/var/lib/postgres/10` в `/mnt/data` - `mv /var/lib/postgresql/10 /mnt/data`
- попытайтесь запустить кластер - `sudo -u postgres pg_ctlcluster 10 main start`
- напишите получилось или нет и почему
- задание: найти конфигурационный параметр в файлах расположенных в `/etc/postgresql/10/main` который надо поменять и поменяйте его
- напишите что и почему поменяли
- попытайтесь запустить кластер - `sudo -u postgres pg_ctlcluster 10 main start`
- напишите получилось или нет и почему
- зайдите через psql и проверьте содержимое ранее созданной таблицы
- задание со звездочкой: не удаляя существующий GCE инстанс сделайте новый, поставьте на его PostgreSQL, удалите файлы с данными из `/var/lib/postgres`, перемонтируйте внешний диск который сделали ранее от первой виртуальной машины ко второй и запустите PostgreSQL на второй машине так чтобы он работал с данными на внешнем диске, расскажите как вы это сделали и что в итоге получилось.

#### 2 вариант:

- сделать в GCE инстанс с Ubuntu 18.04
- поставить на нем Docker Engine

- сделать каталог /var/lib/postgres
- развернуть контейнер с PostgreSQL 11 смонтировав в него /var/lib/postgres
- развернуть контейнер с клиентом postgres
- подключится из контейнера с клиентом к контейнеру с сервером и сделать таблицу с парой строк
- подключится к контейнеру с сервером с ноутбука
- удалить контейнер с сервером
- создать его заново
- подключится снова из контейнера с клиентом к контейнеру с сервером
- проверить, что данные остались на месте
- оставляйте в ЛК ДЗ комментарии что и как вы делали и как боролись с проблемами

**1 Логический уровень PostgreSQL**

иметь представление об логическом устройстве PostgreSQL;  
изучить работу утилиты psql;  
знать как PostgreSQL работает с данными на логическом уровне.

## Домашние задания

**1** Работа с базами данных, пользователями и правами

Цель: - создание новой базы данных, схемы и таблицы  
- создание роли для чтения данных из созданной схемы созданной базы данных  
- создание роли для чтения и записи из созданной схемы созданной базы данных

1 вариант:

1 создайте новый кластер PostgreSQL 11 (на выбор - GCE, CloudSQL)

2 зайдите в созданный кластер под пользователем postgres

3 создайте новую базу данных testdb

4 зайдите в созданную базу данных под пользователем postgres

5 создайте новую схему testnm

6 создайте новую таблицу t1 с одной колонкой c1 типа integer

7 вставьте строку со значением c1=1

8 создайте новую роль readonly

9 дайте новой роли право на подключение к базе данных testdb

10 дайте новой роли право на использование схемы testnm

11 дайте новой роли право на select для всех таблиц схемы testnm

12 создайте пользователя testread с паролем test123

13 дайте роль readonly пользователю testread

14 зайдите под пользователем testread в базу данных testdb

15 сделайте select \* from t1;

16 получилось? (могло если вы делали сами не по шпаргалке и не упустили один существенный момент про который позже)

17 напишите что именно произошло в тексте домашнего задания

18 у вас есть идеи почему? ведь права то дали?

19 посмотрите на список таблиц

20 подсказка в шпаргалке под пунктом 20

21 а почему так получилось с таблицей (если делали сами и без шпаргалки то может у вас все нормально)

22 вернитесь в базу данных testdb под пользователем postgres

23 удалите таблицу t1

24 создайте ее заново но уже с явным указанием имени схемы testnm

25 вставьте строку со значением c1=1

26 зайдите под пользователем testread в базу данных testdb

27 сделайте select \* from testnm.t1;

28 получилось?

29 есть идеи почему? если нет - смотрите шпаргалку

30 как сделать так чтобы такое больше не повторялось?  
если нет идей - смотрите шпаргалку  
31 сделайте `select * from testnm.t1;`  
32 получилось?  
33 есть идеи почему? если нет - смотрите шпаргалку  
31 сделайте `select * from testnm.t1;`  
32 получилось?  
33 ура!  
34 теперь попробуйте выполнить команду `create table t2(c1 integer); insert into t2 values (2);`  
35 а как так? нам же никто прав на создание таблиц и `insert` в них под ролью `readonly`?  
36 есть идеи? если нет - смотрите шпаргалку  
37 если вы справились сами то расскажите что сделали и почему, если смотрели шпаргалку - объясните что сделали и почему выполнив указанные в ней команды  
38 теперь попробуйте выполнить команду `create table t3(c1 integer); insert into t2 values (2);`  
39 расскажите что получилось и почему

---

## 2 **MVCC, vacuum и autovacuum.**

объяснить работу механизма многоверсионности в PostgreSQL; знать и уметь использовать `vacuum` и `autovacuum`; понимать если ли у вас проблемы с наличием или отсутствием `autovacuum`.

Домашние задания

### 1 Настройка `autovacuum` с учетом оптимальной производительности

Цель: - запустит нагрузочный тест `pgbench` с профилем нагрузки DWH

- настроит параметры `autovacuum` для достижения максимального уровня устойчивой производительности

- создать GCE инстанс типа `n1-standard-1` (1 vCPU, 3.75 GB memory) и `standard disk 10GB`

- установить на него PostgreSQL 11 с дефолтными настройками

- применить параметры настройки PostgreSQL из прикрепленного к материалам занятия файла

- выполнить `pgbench -i postgres`

- запустить `pgbench -c8 -P 60 -T 3600 -U postgres postgres`

- дать отработать до конца

- зафиксировать среднее значение `tps` в последней 1/6 части работы

- а дальше настроить `autovacuum` максимально эффективно

- так чтобы получить максимально ровное значение `tps` на горизонте часа

---

## 3 **Журналы**

настраивать журналирование; корректно настроить схему контрольных точек.

Домашние задания

### 1 Работа с журналами

Цель: - уметь работать с журналами и контрольными точками

- уметь настраивать параметры журналов

1. Настройте выполнение контрольной точки раз в 30 секунд.
2. 10 минут с помощью утилиты `rgbench` подавайте нагрузку.
3. Измерьте, какой объем журнальных файлов был сгенерирован за это время. Оцените, какой объем приходится в среднем на одну контрольную точку.
4. Проверьте данные статистики: все ли контрольные точки выполнялись точно по расписанию. Почему так произошло?
5. Сравните `tps` в синхронном/асинхронном режиме утилитой `rgbench`. Объясните полученный результат.
6. Создайте новый кластер с включенной контрольной суммой страниц. Создайте таблицу. Вставьте несколько значений. Выключите кластер. Измените пару байт в таблице. Включите кластер и сделайте выборку из таблицы. Что и почему произошло? как проигнорировать ошибку и продолжить работу?

## 4 Блокировки

объяснить как работают блокировки;  
уметь находить проблемные места.

Домашние задания

### 1 Механизм блокировок

Цель: понимать как работает механизм блокировок объектов и строк

1. Настройте сервер так, чтобы в журнал сообщений сбрасывалась информация о блокировках, удерживаемых более 200 миллисекунд. Воспроизведите ситуацию, при которой в журнале появятся такие сообщения.
2. Смоделируйте ситуацию обновления одной и той же строки тремя командами `UPDATE` в разных сеансах. Изучите возникшие блокировки в представлении `pg_locks` и убедитесь, что все они понятны. Пришлите список блокировок и объясните, что значит каждая.
3. Воспроизведите взаимоблокировку трех транзакций. Можно ли разобраться в ситуации постфактум, изучая журнал сообщений?
4. Могут ли две транзакции, выполняющие единственную команду `UPDATE` одной и той же таблицы (без `where`), заблокировать друг друга?  
\* Попробуйте воспроизвести такую ситуацию.

## 1 Настройка PostgreSQL

работать с параметрами конфигурации PostgreSQL;  
объяснить в чем разница между различными группами параметров;  
объяснить выбор оптимального значения для параметров.

Домашние задания

### 1 Нагрузочное тестирование и тюнинг PostgreSQL

Цель: - делать нагрузочное тестирование PostgreSQL  
- настраивать параметры PostgreSQL для достижения максимальной производительности

- сделать проект <firstname>-<lastname>-<yyyymmdd>-05
- сделать инстанс Google Cloud Engine типа n1-standard-1 с ОС Ubuntu 18.04
- поставить на него PostgreSQL 11 из пакетов собираемых postgres.org
- настроить кластер PostgreSQL 11 на максимальную производительность не обращая внимание на возможные проблемы с надежностью в случае аварийной перезагрузки виртуальной машины
- загрузить кластер через утилиту <https://github.com/Percona-Lab/sysbench-tpcc> (требуется установка <https://github.com/akopytov/sysbench>)
- написать какого значения tps удалось достичь, показать какие параметры в какие значения устанавливали и почему

## 2 Резервное копирование и восстановление

настраивать бэкапы;  
восстанавливать информацию после сбоя.

## 3 Виды и устройство репликации в PostgreSQL. Практика применения

настраивать репликацию;  
выбрать правильный план репликации.

Домашние задания

### 1 Репликация

Цель: реализовать свой миникластер на 3 VM.

На 1 VM создаем таблицы test для записи, test2 для запросов на чтение. Создаем публикацию таблицы test и подписываемся на публикацию таблицы test2 с VM №2. На 2 VM создаем таблицы test2 для записи, test для запросов на чтение. Создаем публикацию таблицы test2 и подписываемся на публикацию таблицы test1 с VM №1. 3 VM использовать как реплику для чтения и бэкапов (подписаться на таблицы из VM №1 и №2 ). Небольшое описание, того, что получилось.



\* реализовать горячее реплицирование для высокой доступности на 4ВМ. Источником должна выступать ВМ №3. Написать с какими проблемами столкнулись.

- 1 **Работа с большим объемом реальных данных**
- объяснить природу больших данных и знать места их обитания;  
загружать большие данные в PostgreSQL;  
объяснить как работать с большими данными.
- Домашние задания
- 1 Разворачиваем и настраиваем БД с большими данными
- Цель: - знать различные механизмы загрузки данных  
- уметь пользоваться различными механизмами загрузки данных
- Необходимо провести сравнение скорости работы запросов на различных СУБД
- 1) Выбрать одну из СУБД
  - 2) Загрузить в неё данные (10 Гб)
  - 3) Сравнить скорость выполнения запросов на PostgreSQL и выбранной СУБД
  - 4) Описать что и как делали и с какими проблемами столкнулись
- 
- 2 **Виды индексов. Работа с индексами и оптимизация запросов**
- создавать индексы;  
использовать различные виды индексов;  
читать вывод команду explain;  
оптимизировать скорость работы запросов.
- 
- 3 **Различные виды join'ов. Применение и оптимизация**
- объяснить зачем нужны join;  
объяснить чем они отличаются;  
использовать join для правильного соединения данных из разных таблиц.
- 
- 4 **Сбор и использование статистики**
- настраивать сбор статистики;  
пользоваться системным представлением со статистикой;  
пользоваться расширением pg\_stat\_statements.
- Домашние задания
- 1 Работа с индексами, join'ами, статистикой
- Цель: - знать и уметь применять основные виды индексов PostgreSQL  
- строить и анализировать план выполнения запроса  
- уметь оптимизировать запросы для с использованием индексов  
- знать и уметь применять различные виды join'ов  
- строить и анализировать план выполнения запроса  
- оптимизировать запрос  
- уметь собирать и анализировать статистику для таблицы
- 1 вариант:

Создать индексы на БД, которые ускорят доступ к данным.

В данном задании тренируются навыки:

- определения узких мест
- написания запросов для создания индекса
- оптимизации

Необходимо:

- 1) Создать индекс к какой-либо из таблиц вашей БД
- 2) Прислать текстом результат команды explain, в которой используется данный индекс
- 3) Реализовать индекс для полнотекстового поиска
- 4) Реализовать индекс на часть таблицы или индекс на поле с функцией
- 5) Создать индекс на несколько полей
- 6) Написать комментарии к каждому из индексов
- 7) Описать что и как делали и с какими проблемами столкнулись

2 вариант:

В результате выполнения ДЗ вы научитесь пользоваться различными вариантами соединения таблиц.

В данном задании тренируются навыки:

- написания запросов с различными типами соединений

Необходимо:

- 1) Реализовать прямое соединение двух или более таблиц
- 2) Реализовать левостороннее (или правостороннее) соединение двух или более таблиц
- 3) Реализовать кросс соединение двух или более таблиц
- 4) Реализовать полное соединение двух или более таблиц
- 5) Реализовать запрос, в котором будут использованы разные типы соединений
- 6) Сделать комментарии на каждый запрос
- 7) К работе приложить структуру таблиц, для которых выполнялись соединения

\* Придумайте 3 своих метрики на основе показанных представлений, отправьте их через ЛК, а так же поделитесь с коллегами в слаке

---

5 **Оптимизация производительности. Профилирование. Мониторинг**

определять узкие места базы, оптимизировать и профилировать запросы

---

6 **Pentaho**

научиться использовать кубы данных

---

7 **Секционирование**

сделать партиционирование для своих таблиц в PostgreSQL  
Работать с pg\_partman  
Обслуживать партицированные таблицы

Домашние задания

- 1 Секционирование таблицы

Секционировать большую таблицу из демо базы flights

## 5 Высокая доступность и горизонтальная масштабируемость

1	<b>Введение в Kubernetes. Часть 1</b>	научиться работать в Kubernetes
2	<b>Введение в Kubernetes. Часть 2</b>	научиться работать в Kubernetes
3	<b>Варианты кластеров высокой доступности для PostgreSQL</b>	выбрать оптимальный вариант высокой доступности для PostgreSQL; проанализировать варианты классических HA кластеров для PostgreSQL; настраивать классический PostgreSQL HA кластер.
4	<b>Работа с кластером высокой доступности</b>	выбрать оптимальный вариант высокой доступности для PostgreSQL; проанализировать варианты классических HA кластеров для PostgreSQL; настраивать классический PostgreSQL HA кластер.  Домашние задания  1 Развернуть HA кластер  Цель: - уметь развернуть высокодоступный кластер PostgreSQL собственными силами - уметь развернуть высокодоступный сервис на базе PostgreSQL на базе одного из 3-ки ведущих облачных провайдеров - AWS, GCP и Azure  Выбрать один из вариантов и развернуть кластер. Описать что и как делали и с какими проблемами столкнулись Вариант 1 • How to Deploy PostgreSQL for High Availability Вариант 2 • Introducing pg_auto_failover: Open source extension for automated failover and high-availability in PostgreSQL Для гурманов • Настройка Active/Passive PostgreSQL Cluster с использованием Pacemaker, Corosync, и DRBD (CentOS 5,5)  * Создать два кластера GKE в разных регионах Установить на первом Patroni HA кластер Установить на втором Patroni Standby кластер Настроить TCP LB между регионами Сделать в каждом регионе по клиентской VM Проверить как ходит трафик с клиентской VM Описать что и как делали и с какими проблемами столкнулись

5 **Способы горизонтального масштабирования PostgreSQL** проанализировать горизонтальное масштабирование PostgreSQL; объяснить, когда стоит использовать горизонтальное масштабирование, а когда нет; настраивать параллельный кластер.

---

6 **Работа с горизонтально масштабируемым кластером** разворачивать мультимастер кластера; настраивать мультимастер кластера; настраивать геокластер.

Домашние задания

1 Multi master

Цель: - уметь развернуть multi master кластер PostgreSQL своими руками  
- уметь развернуть PostgreSQL like географически распределенный сервис от одного из 3-х крупнейших облачных провайдеров - AWS, GCP и Azure

1 вариант:

Развернуть CockroachDB в GKE или GCE  
Потесировать dataset с чикагскими такси  
Или залить 10Гб данных и протестировать скорость запросов в сравнении с 1 экземпляром PostgreSQL  
Описать что и как делали и с какими проблемами столкнулись

2 вариант:

Переносим тестовую БД 10 Гб в географически распределенный PostgreSQL like сервис  
Описать что и как делали и с какими проблемами столкнулись

- PostgreSQL и Kubernetes**

развертыванию HA и мультимастер PostgreSQL кластера в k8s.

Домашние задания

  - Работа с PostgreSQL в Kubernetes

Цель: - уметь запускать HA и multi master PostgreSQL кластер в Kubernetes

Запустить HA и multi master PostgreSQL кластер в Kubernetes. Описать что и как делали и с какими проблемами столкнулись

---
- PostgreSQL и AWS**

работать в AWS;  
воспользоваться PostgreSQL based сервисами в AWS;  
воспользоваться PostgreSQL like сервисами в AWS.

---
- PostgreSQL и Google Cloud Platform**

работать в GCP;  
воспользоваться PostgreSQL based сервисами в GCP;  
воспользоваться PostgreSQL like сервисами в GCP.

---
- PostgreSQL и Azure**

работать в Azure;  
воспользоваться PostgreSQL based сервисами в Azure;  
воспользоваться PostgreSQL like сервисами в Azure.

Домашние задания

  - PostgreSQL и Azure, GCP, AWS

Цель: - уметь пользоваться PostgreSQL based и like сервисами в Azure, GCP, AWS

1 вариант:  
Воспользоваться PostgreSQL based и like сервисами в AWS. Описать что и как делали и с какими проблемами столкнулись

2 вариант:  
Воспользоваться PostgreSQL based и like сервисами в GCP. Описать что и как делали и с какими проблемами столкнулись

3 вариант:  
Воспользоваться PostgreSQL based и like сервисами в Azure. Описать что и как делали и с какими проблемами столкнулись

---
- Подведение итогов курса**

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.





- |   |  |
|---|--|
| <b>1</b> <b>Выбор темы и организация проектной работы</b> | выбрать и обсудить тему проектной работы;<br>спланировать работу над проектом;<br>ознакомиться с регламентом работы над проектом.  |
|   | Домашние задания   |
|   | <b>1</b> Проект<br><br>Цель: Создать свой проект<br><br>1. Выбрать тему проектной работ<br>2. Согласовать ее с руководителем курса<br>3. Реализовать свой проект на 60%+<br>4. Сделать презентацию своего проекта и защитить его на защите   |
| <b>2</b> <b>Самый простой back-end на NodeJS</b>          | научиться разрабатывать самый простой back-end;<br>писать простейшие приложения на JS;<br>пользоваться платформой NodeJS;<br>научиться разрабатывать простой back-end для Web-приложений на NodeJS.  |
| <b>3</b> <b>Веб-приложение с UI и БД</b>                  | научиться писать простейший UI;<br>подключать приложение к БД.   |
|   | Домашние задания   |
|   | <b>1</b> Разработка собственного приложения для работы с кластером<br><br>Цель: - уметь создавать Фронтенд+Бэкенд для работы с кластером<br><br>Написать API для работы с физ.лицами.<br>Простая обёртка<br>Формат JSON физ.лица `{id:1, name: 'Ivan', age: 18}`<br>`GET /person` - получить всех/поиск<br>`GET /person/:id` - получение фил.лица по id<br>`POST /persons` - создание физ.лица<br>`DELETE /person` - удаление физ.лица<br>Нагрузить приложение через яндекс танк |
| <b>4</b> <b>Консультация по дипломному проекту</b>        | получить ответы на вопросы по проекту, ДЗ и по курсу.  |
| <b>5</b> <b>Защита проектных работ</b>                    | защитить проект и получить рекомендации экспертов.   |