

Специальная цена

# PostgreSQL для администраторов баз данных и разработчиков

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

Длительность курса: 118 академических часов

## 1 Реляционные базы, история и место в современном мире

### Цели занятия:

научиться отличать SQL и NoSQL;  
научиться представлять сложность выбора той или иной технологии СУБД;  
чуть лучше понимать, чем мы будем заниматься на этом курсе.

### Краткое содержание:

где мы были и где сейчас;  
выбор и PostgreSQL;  
случай из жизни;  
программа курса.

---

## 2 SQL и реляционные СУБД. Введение в PostgreSQL

### Цели занятия:

объяснить основу реляционной модели данных;  
объяснить назначение языка SQL и его основные конструкции;  
иметь представление об основных реляционных СУБД;  
рассмотреть разницу в уровнях изоляции транзакций

### Краткое содержание:

реляционная модель и SQL;  
OLTP, ACID, MVCC, ARIES;  
уровни изоляции транзакций;  
современные РСУБД;  
введение в PostgreSQL и практика.

### Домашние задания

#### 1 Работа с уровнями изоляции транзакции в PostgreSQL

Цель: - научиться работать с Google Cloud Platform на уровне Google Compute Engine (IaaS)  
- научиться управлять уровнем изоляции транзакции в PostgreSQL и понимать особенность работы уровней read committed и repeatable read

- создать новый проект в Google Cloud Platform, например postgres2022-<ууууммдд>, где ууууммдд год, месяц и день вашего рождения (имя проекта должно быть уникально на уровне GCP)
- дать возможность доступа к этому проекту пользователю ifti@yandex.ru с ролью Project Editor
- далее создать инстанс виртуальной машины Compute Engine с дефолтными параметрами
- добавить свой ssh ключ в GCE metadata
- зайти удаленным ssh (первая сессия), не забывая про ssh-add
- 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?  
ДЗ сдаем в виде миниотчета в markdown в гите

---

### 3 Установка PostgreSQL

#### Цели занятия:

развернуть кластер PostgreSQL различными способами  
объяснить возможности и ограничения каждого способа  
выбрать наиболее оптимальный способ в зависимости от задачи

уметь обновить кластер на новую версию

#### Краткое содержание:

ставим из пакетов

Google Cloud SQL

Google Cloud Marketplace

Docker и Kubernetes

обновим кластер на новую версию

#### Домашние задания

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

Цель: - установить PostgreSQL в Docker контейнере

- настроить контейнер для внешнего подключения

- сделать в GCE инстанс с Ubuntu 20.04
  - поставить на нем Docker Engine
  - сделать каталог /var/lib/postgres
  - развернуть контейнер с PostgreSQL 14 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/компьютера извне инстансов GCP
  - 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
-

4 **DDL: создание, изменение и удаление объектов в PostgreSQL**

**Цели занятия:**

создавать и редактировать объекты БД в СУБД PostgreSQL.

**Краткое содержание:**

create и alter

- базы данных

- тейблспейсов

- пользователей и ролей

- различных видов таблиц, в том числе временных и внешних

- работа с последовательностями (sequences)

- представления и материализованные представления

---

5 **DML: вставка, обновление, удаление, выборка данных**

**Цели занятия:**

использовать различные способы выборки данных из одной и нескольких таблиц;

использовать нужный тип JOIN и различные условия в WHERE.

**Краткое содержание:**

SELECT;

различные варианты JOIN;

условия WHERE.

---

6 **Физический уровень PostgreSQL**

**Цели занятия:**

поговорить об устройстве PostgreSQL;

рассмотреть процессы PostgreSQL и структуру памяти;

обсудить как PostgreSQL работает с данными на физическом уровне.

**Краткое содержание:**

работа с PostgreSQL;

серверные процессы и память;

физическая структура данных;  
практика.

## Домашние задания

### 1 Установка и настройка PostgreSQL

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

- создайте виртуальную машину с Ubuntu 20.04 LTS (bionic) в GCE типа e2-medium в default VPC в любом регионе и зоне, например us-central1-a

- поставьте на нее PostgreSQL 14 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 14 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/14` в `/mnt/data` - `mv /var/lib/postgresql/14 /mnt/data`

- попытайтесь запустить кластер - `sudo -u postgres pg_ctlcluster 14 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 14 main start  
- напишите получилось или нет и почему  
- зайдите через 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.

---

## 7 Логический уровень PostgreSQL

### Цели занятия:

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

### Краткое содержание:

логический уровень psql и системный каталог;  
работа с данными;  
практика.

### 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 создайте новый кластер PostgreSQL 13 (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 как убрать эти права? 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

## 1 MVCC, vacuum и autovacuum.

### Цели занятия:

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

### Краткое содержание:

вспоминаем ACID;  
MVCC в PostgreSQL;  
Vacuum & Autovacuum;  
практика.

### Домашние задания

#### 1 Настройка autovacuum с учетом оптимальной производительности

Цель: - запустить нагрузочный тест pgbench с профилем нагрузки DWH  
- настроить параметры autovacuum для достижения максимального уровня устойчивой производительности

- создать GCE инстанс типа e2-medium и standard disk 10GB
  - установить на него PostgreSQL 13 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
-

## Цели занятия:

настроить журналирование;  
корректно настроить схему контрольных точек.

## Краткое содержание:

буферный кеш;  
журнал предзаписи;  
контрольная точка;  
настройки журнала.

## 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?
-

### Цели занятия:

объяснить как работают блокировки;  
научиться находить проблемные места.

### Краткое содержание:

блокировки объектов;  
блокировки строк;  
блокировки в памяти.

### 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.
-

## 4 Семинар

### **Цели занятия:**

ответы на вопросы слушателей  
поговорим о выборе проектов

### **Краткое содержание:**

выбор проектной работы  
более подробный разбор непонятных моментов

## 1 Настройка PostgreSQL

### Цели занятия:

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

### Краткое содержание:

настройка сервера;  
работа с параметрами;  
значения параметров;  
практика.

### Домашние задания

#### 1 Нагрузочное тестирование и тюнинг PostgreSQL

Цель: - сделать нагрузочное тестирование PostgreSQL

- настроить параметры PostgreSQL для достижения максимальной производительности

- сделать проект <firstname>-<lastname>-<ууууммдд>-10
  - сделать инстанс Google Cloud Engine типа e2-medium с ОС Ubuntu 20.04
  - поставить на него PostgreSQL 13 из пакетов собираемых postgres.org
  - настроить кластер PostgreSQL 13 на максимальную производительность не обращая внимание на возможные проблемы с надежностью в случае аварийной перезагрузки виртуальной машины
  - нагрузить 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**

**Цели занятия:**

настроить бэкапы;  
восстановить информацию после сбоя.

**Краткое содержание:**

бэст практис;  
варианты логического бэкапа;  
физический бэкап;  
а как же в облаках?

---



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

#### Цели занятия:

настроить репликацию;  
выбрать правильный план репликации.

#### Краткое содержание:

задачи репликации;  
физическая репликация;  
логическая репликация;  
практика применения.

#### Домашние задания

##### 1 Репликация

Цель: реализовать свой миникластер на 3 ВМ.

На 1 ВМ создаем таблицы test для записи, test2 для запросов на чтение. Создаем публикацию таблицы test и подписываемся на публикацию таблицы test2 с ВМ №2. На 2 ВМ создаем таблицы test2 для записи, test для запросов на чтение. Создаем публикацию таблицы test2 и подписываемся на публикацию таблицы test1 с ВМ №1. 3 ВМ использовать как реплику для чтения и бэкапов (подписаться на таблицы из ВМ №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.

---

## 4 Кластер Patroni

### Цели занятия:

построить отказоустойчивый кластер с помощью patroni.

### Краткое содержание:

patron;  
consul;  
pgbouncer;  
HA cluster.

**1** **Виды индексов.  
Работа с индексами  
и оптимизация  
запросов**

**Цели занятия:**

создать индексы;  
научиться использовать различные виды индексов;  
прочитать вывод команду explain;  
оптимизировать скорость работы запросов.

**Краткое содержание:**

виды индексов;  
типы использования индекса в explain;  
зачем он вообще нужен;  
принцип работы индексов.

---

**2** **Различные виды  
join'ов. Применение  
и оптимизация**

**Цели занятия:**

объяснить зачем нужны join;  
объяснить чем они отличаются;  
использовать join для правильного соединения данных из разных таблиц.

**Краткое содержание:**

виды join;  
отличия и варианты применения;  
практика.

---

**3** **Сбор и  
использование  
статистики**

**Цели занятия:**

настроить сбор статистики;  
научиться пользоваться системным представлением со статистикой;  
научиться пользоваться расширением pg\_stat\_statements.

**Краткое содержание:**

сбор статистики;  
системные представления статистики;  
расширение 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

---

4 **Оптимизация  
производительности.  
Профилирование.  
Мониторинг**

**Цели занятия:**

определить узкие места базы, оптимизировать и профилировать запросы

**Краткое содержание:**

EXPLAIN

ANALYZE

оптимизация: объекты, подходы;

методики мониторинга и профайлинга

---

## 5 Секционирование

### Цели занятия:

сделать партиционирование для своих таблиц в PostgreSQL;  
поработать с pg\_partman;  
научиться обслуживать партицированные таблицы.

### Краткое содержание:

рассмотрим партиционирование в общем;  
изучим решения, которые использовались (используются) в PostgreSQL;  
применим их на практике.

### Домашние задания

#### 1 Секционирование таблицы

Цель: научиться секционировать таблицы.

Секционировать большую таблицу из демо базы flights

---

## 6 Хранимые функции и процедуры часть 1

### Цели занятия:

получить представление о способах реализации бизнес-логики на стороне сервера БД, научиться разрабатывать простые функции на «чистом» SQL

### Краткое содержание:

Функции: аргументы, передача значений, перегрузка и полиморфизм, основные атрибуты.  
Язык PL/PGSQL: основы – переменные, составные типы, возвращаемые значения и выходные параметры.

---

## 7 Хранимые функции и процедуры часть 2

### Цели занятия:

освоить язык PL/ PGSQL, научиться работать с курсорами.

### Краткое содержание:

Язык PL/PGSQL (продолжение): основные операторы, управляющие конструкции, динамический SQL.

Обработка ошибок.

Процедуры. Управление транзакциями в процедурах.

Курсоры.

---

## Цели занятия:

научиться разрабатывать DML-триггеры и событийные триггеры.

## Краткое содержание:

Триггеры: DML-триггеры, триггерные функции. Переменные контекста. Типизация триггеров. Плюсы и минусы использования триггеров. Событийные триггеры.

## Домашние задания

### 1 Триггеры, поддержка заполнения витрин

Цель: Создать триггер для поддержки витрины в актуальном состоянии.

Скрипт и развернутое описание задачи – в ЛК (файл hw\_triggers.sql) или по ссылке: <https://disk.yandex.ru/d/l70AvknAepIJXQ>

В БД создана структура, описывающая товары (таблица goods) и продажи (таблица sales).

Есть запрос для генерации отчета – сумма продаж по каждому товару.

БД была денормализована, создана таблица (витрина), структура которой повторяет структуру отчета.

Создать триггер на таблице продаж, для поддержки данных в витрине в актуальном состоянии (вычисляющий при каждой продаже сумму и записывающий её в витрину)

Подсказка: не забыть, что кроме INSERT есть еще UPDATE и DELETE

\* Чем такая схема (витрина+триггер) предпочтительнее отчета, создаваемого "по требованию" (кроме производительности)?  
Подсказка: В реальной жизни возможны изменения цен.



## 1 Выбор темы и организация 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 Проект

Цель: Создать свой проект

1. Выбрать тему проектной работ
  2. Согласовать ее с руководителем курса
  3. Реализовать свой проект на 60%+
  4. Сделать презентацию своего проекта и защитить его на защите
- 

## 2 Консультация по дипломному проекту

### 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  
проектных  
работ**

**Цели занятия:**

защитить проект и получить рекомендации экспертов.

**Краткое содержание:**

презентация проектов перед комиссией;  
вопросы и комментарии по проектам.

---

**4 Итоги 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:**

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