

Организация данных

Алексей Федин

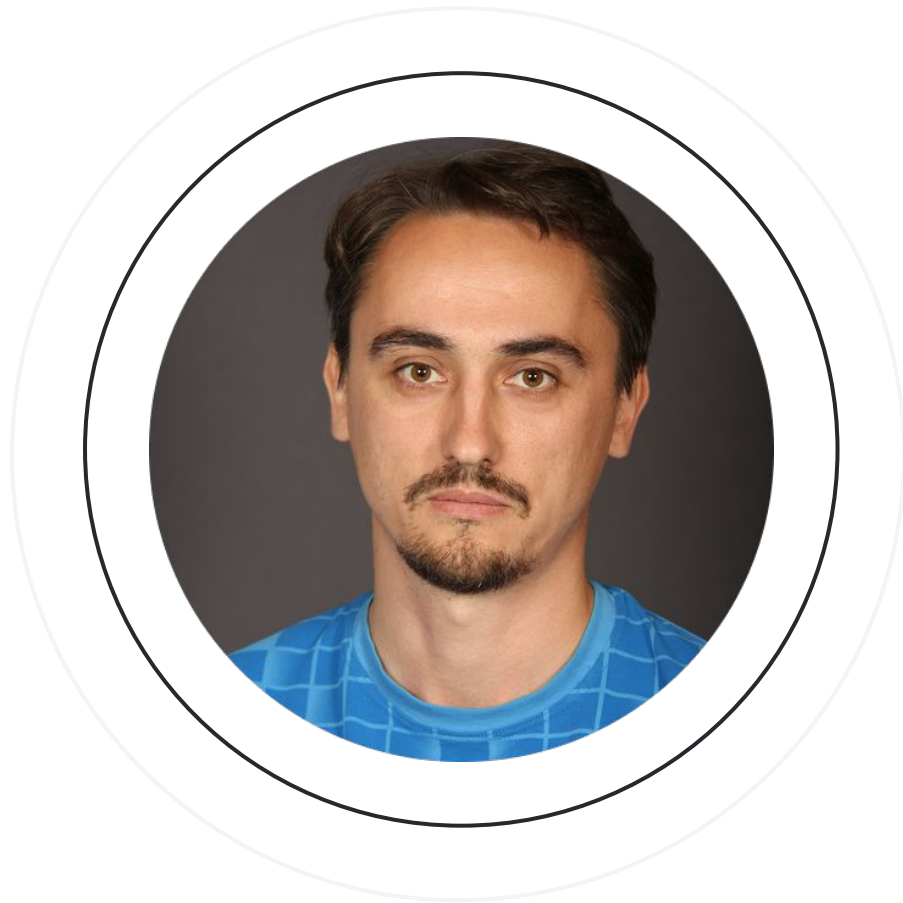
Системный аналитик в ООО «Открытые решения»



Алексей Федин

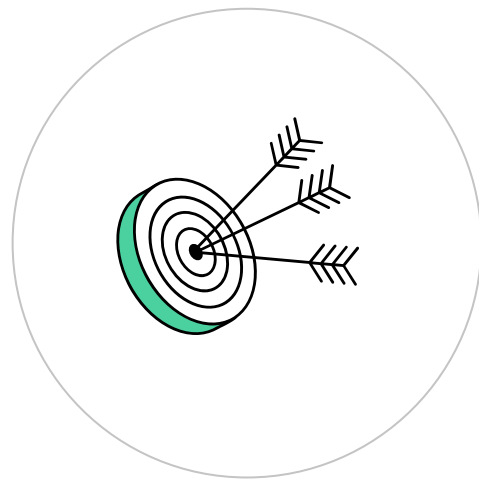
О спикере:

- системный аналитик
- работает в IT с 2002 года
- опыт администрирования и работы с сетями более 10 лет
- с 2008 года занимается проектированием и наладкой решений информационной безопасности в промышленности
- работал в «PTСофт», Positive Technologies, iGrids, ElcomSoft



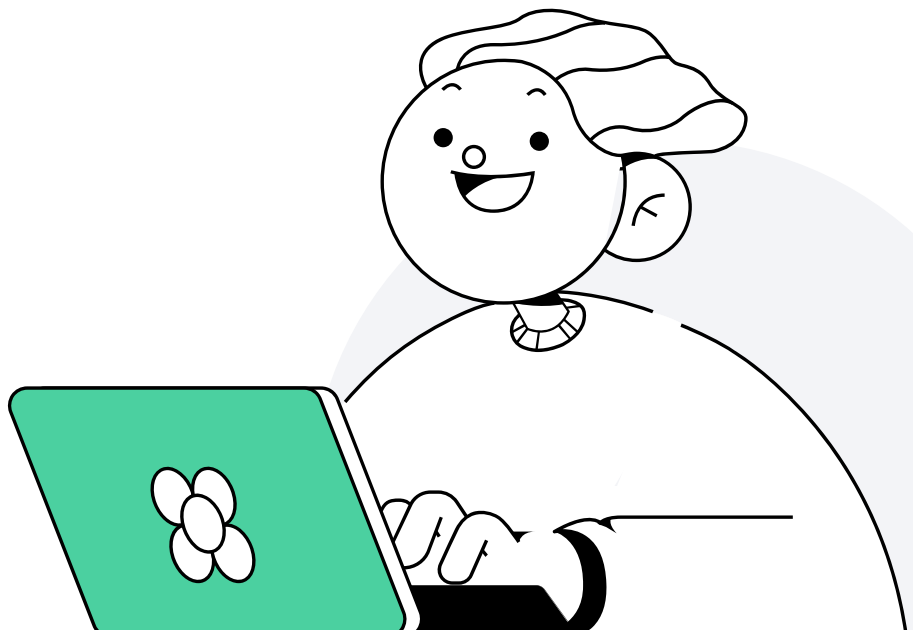
Цели занятия

- Повторить, что такое модели данных
- Узнать, как хранятся данные в PostgreSQL
- Научиться работать с системными таблицами и журналами



План занятия

- 1 Модели данных
- 2 Хранение данных
- 3 Хранение данных в PostgreSQL
- 4 Системные таблицы
- 5 Системные журналы
- 6 Настройки WAL
- 7 Итоги
- 8 Домашнее задание



Модели данных



1



Модель данных — система взаимосвязанных понятий и правил, которая описывает структуру и свойства данных, используемых в БД

Состав модели данных

- Базовые типы данных
- Правила построения составных типов данных
- Правила взаимосвязей между объектами данных
- Правила задания ограничений целостности
- Операции, используемые в модели данных

Реляционная модель данных

С точки зрения математики:

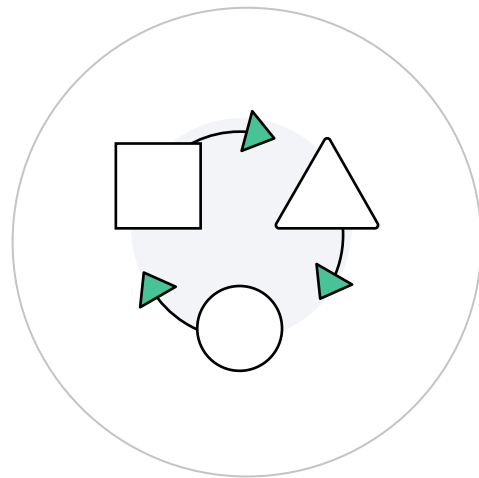
- домен — множество значений, обладающих общими свойствами
- отношение — предикат для n -аргументов
- атрибут — аргумент отношения
- кортеж — значение атрибутов отношения

С точки зрения IT:

- домен — тип данных
- отношение — таблица
- атрибут — столбец таблицы
- кортеж — строка таблицы

Типы связей

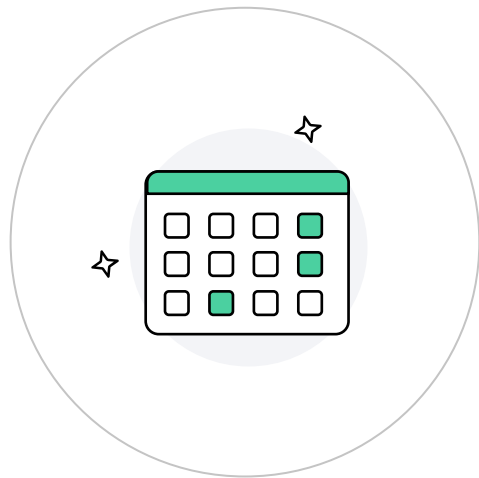
- Один к одному
- Один ко многим
- Многие ко многим



Ключи

Ключ — столбец или группа столбцов, значения которых уникальным образом идентифицируют строку таблицы

Внешний ключ — ключ для связи между таблицами



Целостность данных

Целостность — набор правил, согласно которым данные в БД находятся в согласованном состоянии.

Виды целостности для реляционных БД:

- целостность доменов
- целостность сущностей
- ссылочная целостность

Нормализация данных

Нормализация — набор правил, позволяющих избежать избыточности данных и облегчающих выполнение операций над ними.

Помогает бороться с некорректными операциями:

- вставка
- удаление
- редактирование

Нормальные формы

Нормализация реляционных данных состоит в выполнении 5 этапов (форм) преобразования данных: 1NF ... 5NF.

Все нормальные формы должны быть выполнены последовательно — **пропускать этапы нормализации нельзя**

Нормальные формы

- 1 Все поля в таблицах неделимы и не содержат повторяющихся групп
- 2 Все неключевые поля в таблицах зависят от первичного ключа
- 3 В таблицах отсутствуют избыточные неключевые поля
- 4 В таблицах устранены многозначные зависимости

ACID

ACID — набор требований к транзакционной системе, обеспечивающих её наиболее надёжную и предсказуемую работу:

- атомарность — atomicity
- согласованность — consistency
- изолированность — isolation
- устойчивость — durability

Хранение данных

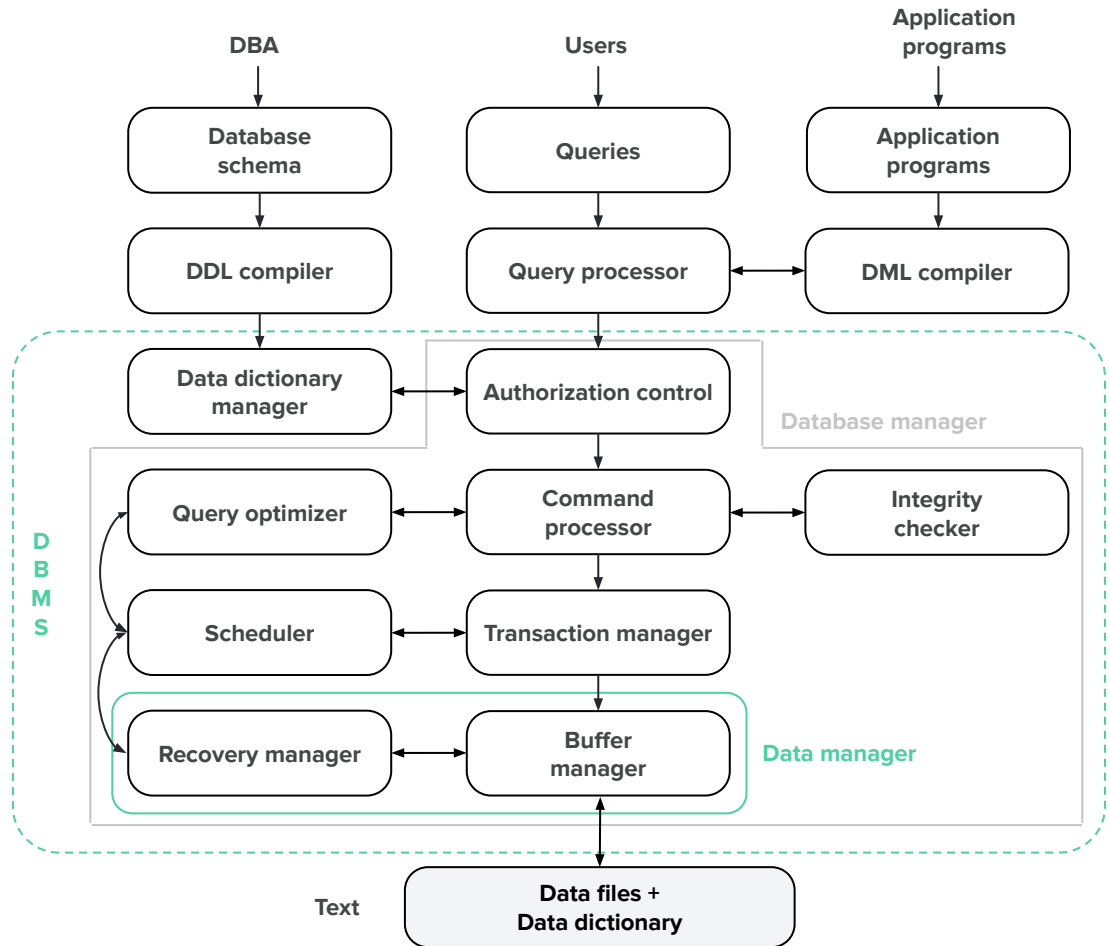


2

СУБД

СУБД (система управления базами данных, database management system, DBMS) — специальный подход к организации хранения и обработки данных, который представляет собой совокупность программных и специальных языковых средств

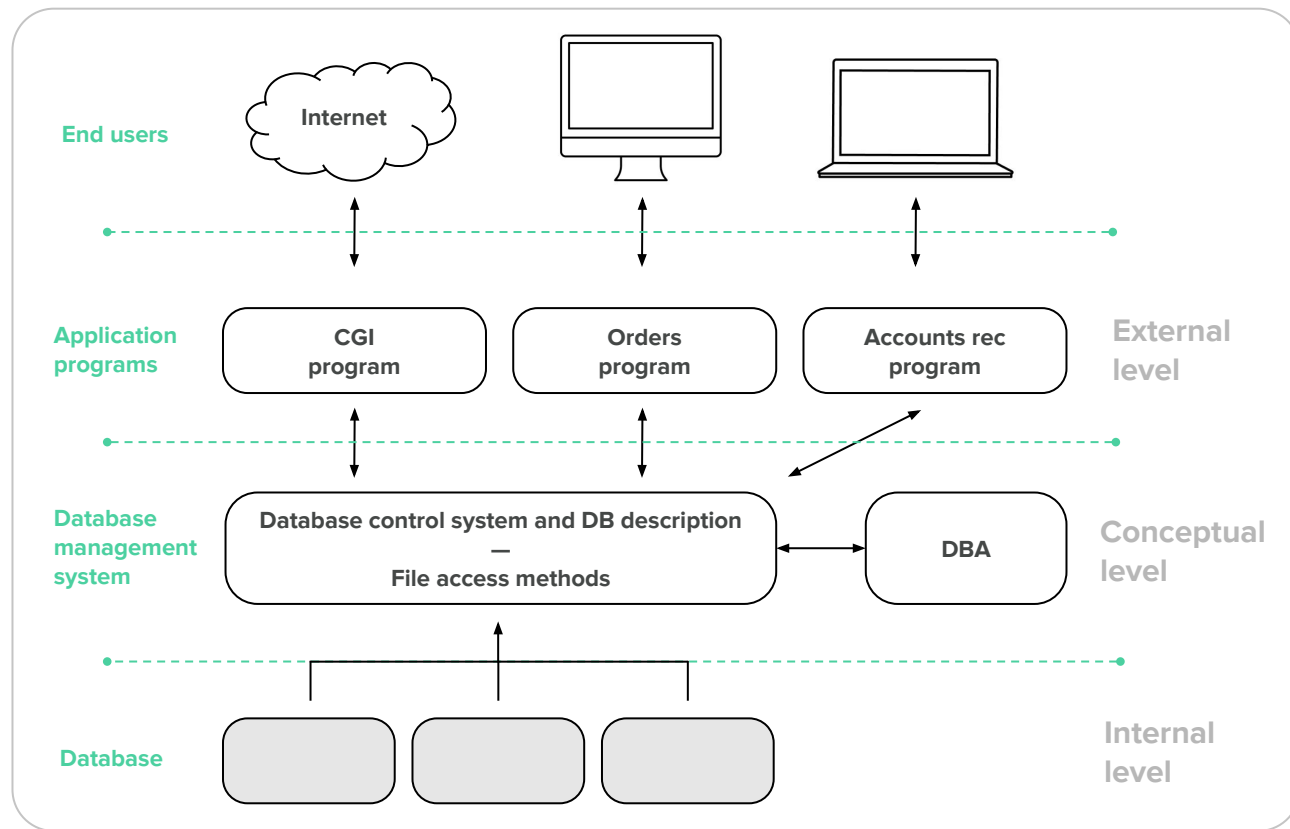
База данных (database) — совокупность взаимосвязанных данных, управляемых СУБД



Архитектура СУБД

Для разделения представления пользовательских данных и их физической организации в системе СУБД обычно выделяют 3 уровня:

1. Внешний
2. Концептуальный
3. Внутренний



Архитектура СУБД: внешний уровень

Внешний уровень — представление данных с точки зрения пользователя.

Сами представления могут различаться для каждого пользователя и отображаться в наиболее удобном виде:

- технологический процесс
- рецепт
- экранная форма
- запись в журнале и т. д.

Архитектура СУБД: концептуальный уровень

На **концептуальном уровне** описывается, какие данные будут храниться в БД и как они связаны между собой.

Можно сказать, что это **логическая структура БД**:

- атрибуты
- связи
- ограничения данных и связей
- атрибуты безопасности и т. д.

Архитектура СУБД: внутренний уровень

На **внутреннем уровне** описывается физическое хранение данных в системе.

Основная цель — достижение максимальной производительности системы.

На этом уровне описывается **взаимодействие СУБД и ОС**:

- хранение данных
- хранение индексов
- сжатие данных
- шифрование и т. д.

Хранение данных в PostgreSQL



3

Табличные пространства

Табличное пространство — каталог файловой системы, в котором хранятся файлы БД.

При формировании кластера создаются 2 табличных пространства:

- pg_default — табличное пространство по умолчанию (\$PGDATA/base)
- pg_global — пространство для общих объектов системного каталога (\$PGDATA/global)

Шаблоны табличных пространств можно изменить

PGDATA

PGDATA — переменная среды, указывающая на каталог, в котором хранятся файлы конфигурации и файлы данных, используемые кластером БД.

Просмотр PGDATA:

```
show data_directory;
```


Создание табличного пространства

1. Создание каталога в файловой системе:

```
\! mkdir /home/db/test_dir
```

2. Назначение прав пользователю PostgreSQL
3. Создание табличного пространства:

```
CREATE TABLESPACE test_ts LOCATION '/home/db/test_dir';
```

4. Создание БД в табличном пространстве:

```
CREATE DATABASE test_sb TABLESPACE test_ts;
```

5. Просмотр табличных пространств:

```
\db
```

Операции с табличным пространством

1. Удаление табличного пространства:

```
DROP TABLESPACE test_ts;
```

2. Перемещение таблицы в другое табличное пространство:

```
ALTER TABLE table1 SET TABLESPACE test_ts;
```

3. Перемещение индекса в другое табличное пространство:

```
ALTER INDEX i_table1 SET TABLESPACE test_ts;
```

Слои

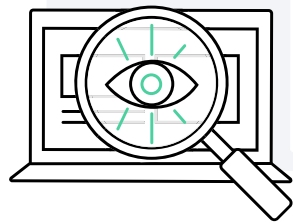
Файл БД на диске занимает не больше 1 ГБ, он кратен 8 КБ. По этой причине если размер таблицы более 1 ГБ, то она хранится в нескольких файлах. Файлы состоят из 8 КБ страниц, которые при необходимости помещаются в буферный кеш.

Существуют следующие слои:

- main — содержит данные объекта
- init — содержит пустую копию таблицы (только для нежурналируемых таблиц)
- fsm — хранит информацию о том, где внутри файлов есть свободное пространство
- vm — применяется для оптимизации, чтобы VACUUM знал, какие страницы чистить уже не нужно

Демонстрация работы

Работа с табличными пространствами



Системные таблицы



4

Системный каталог

Системный каталог (информационная схема) — объект, содержащий информацию об объектах в базе данных.

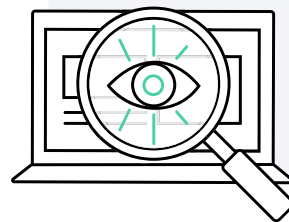
Системный каталог — это набор таблиц, расположенных в `pg_catalog`

Системные представления

Системные представления содержат некоторые популярные запросы к системным каталогам или предоставляют доступ к внутреннему состоянию сервера

Демонстрация работы

Устройство системного каталога



Системные журналы



5

Журналы

Журналы — текстовые файлы, в которых записывается информация о текущей активности в БД:

- пользователи, осуществляющие доступ к объектам
- ошибки
- изменения в настройках
- выполнение запросов
- выполнение транзакций и др.

Расположение журналов

Debian:

```
/var/log/postgresql/postgresql-x.x.main.log.X.x.
```

Red Hat:

```
/var/lib/pgsql/data/pg_log
```

Windows:

```
C:\Program Files\PostgreSQL\<version>\data\pg_log
```

Основные журналы

Журнал запросов — хранит все запросы, которые были сделаны на сервере. Должен быть включён `log_statement`

Журнал транзакций — хранит записи всех событий, происходящих с базой данных. Используется для восстановления после сбоев

Журнал соединений — хранит все подключения к серверу. Должен быть включён `log_connections` в файле `postgresql.conf`

Журнал ошибок — хранит сообщения о работе сервера. Уровнем важности регистрации сообщений об ошибках управляет `log_in_error_statement`

Журнал аудита — хранит изменения, внесённые в базу данных, и данные о выполненных запросах: кто, когда и какой именно запрос выполнял

Уровни логирования

DEBUG1...5	Избыточно подробная информация — более интересна для разработчиков
INFO	Подробная информация
NOTICE	Полезная информация
WARNING	Предупреждения о вероятных проблемах
ERROR	Ошибки
LOG	Данные для логов
FATAL	Ошибки, приводящие к прерыванию текущего сеанса работы
PANIC	Ошибки, приводящие к прерыванию всех сеансов

Уровни логирования

log_rotation_age	Автоматическая ротация логов (минуты)
log_rotation_size	Автоматическая ротация логов (килобайты)
log_truncate_on_rotation	Перезапись данных в логах

pgbadger

pgbadger — утилита для анализа журналов PostgreSQL. Она отображает:

- статистику БД
- запросы, находящиеся в ожидании дольше всего
- запросы, создающие больше всего вре́менных файлов
- самые медленные запросы
- самые частые запросы
- самые частые ошибки
- гистограммы и отчёты о производительности и т. п.

<https://github.com/darold/pgbadger>

Настройки WAL



6

WAL

WAL (write-ahead log, журнал предзаписи) — это стандартный метод обеспечения целостности данных в PostgreSQL.

Работа механизма WAL состоит в том, что изменения в файлах с данными записываются только после их занесения в журнал.

Таким образом, любые изменения, которые не применялись к страницам с данными, могут быть воссозданы из записей журнала

Checkpoint

Checkpoint — точка, означающая, что состояние всех страниц на определённый момент времени было записано на диск.

В случае восстановления БД достаточно работать с записями WAL только с последнего checkpoint

Уровни WAL

Minimal — гарантирует только восстановление после сбоя:

```
wal_level = minimal
```

Replica — гарантирует возможность восстановления из резервной копии и возможность физической репликации:

```
wal_level = replica
```

Logical — гарантирует логическое декодирование и логическую репликацию:

```
wal_level = logical
```

Итоги

- Повторили, что такое модели данных
- Узнали, как хранятся данные в PostgreSQL
- Научились работать с системными таблицами и журналами



Домашнее задание

Давайте посмотрим ваше домашнее задание

- 1 Вопросы по домашней работе задавайте в чате группы
- 2 Задачи можно сдавать по частям
- 3 Зачёт по домашней работе ставят после того, как приняты все задачи



Задавайте вопросы и пишите отзыв о лекции

Алексей Федин

Системный аналитик в ООО «Открытые решения»

