

# Лекция 1

## Введение в БД

# Введение в БД и СУБД

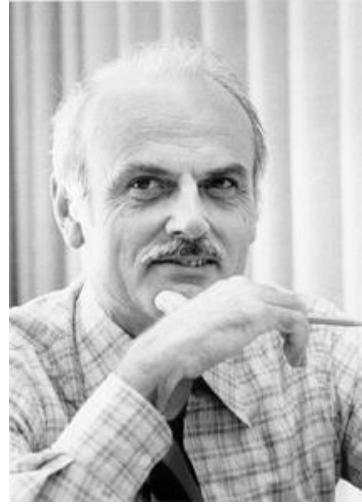


# База данных

- **База данных** – логически структурированная совокупность постоянно хранимых данных, характеризующих актуальное состояние некоторой предметной области и используемых прикладными программными системами какого-либо предприятия
  - Включает также метаданные (словарь данных/системный каталог) и другую служебную информацию.
- В зависимости от способа логической организации (структурирования и представления) данных различают:
  - Реляционные базы данных
  - Нереляционные базы данных (NoSQL)

# Реляционные БД

Концепция впервые была опубликована в 1970  
Эдгаром Франком Коддом



- База данных – совокупность двумерных таблиц, связанных друг с другом с помощью ключей (Primary Key / Foreign Key)
- Каждая таблица хранит информацию об объектах предметной области
- Каждая запись в таблице имеет одинаковую структуру
- Стандартный язык для работы с данными - SQL
- Oracle, MySQL, Microsoft SQL Server, PostgreSQL





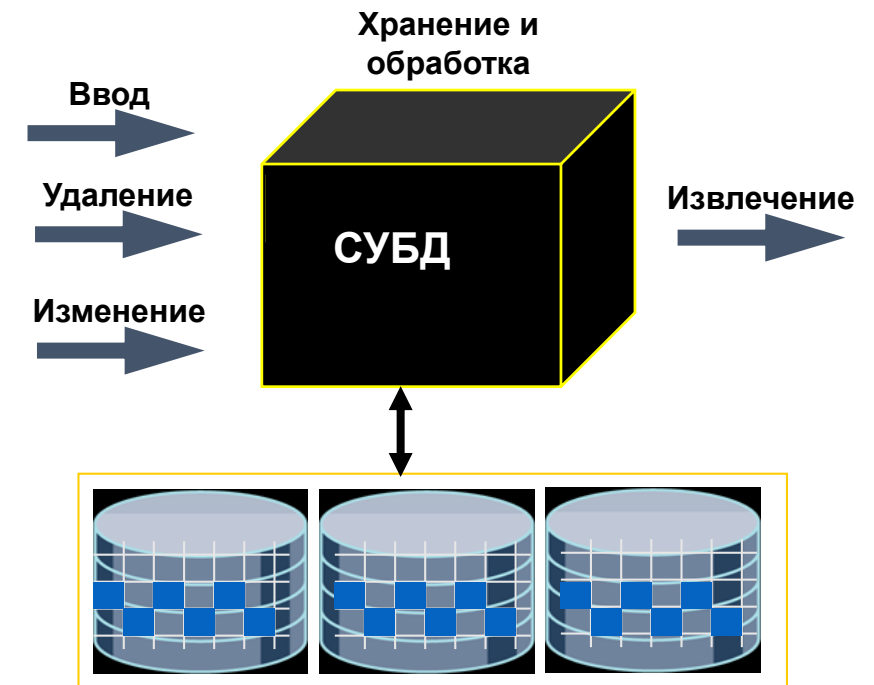
# Нереляционные БД

- **Хранилища ключей и значений.** Простейшая модель. Данные можно легко распределять в кластере. Для сложных запросов не подходят (**Redis, Riak**)
- **Колоночные.** Данные хранятся не по строкам, а по столбцам. Хорошо подходят для BigData (**Hbase, Clickhouse, Vertica**)
- **Документоориентированные.** Хранение коллекций документов с произвольным набором атрибутов (полей) (**CouchDB, Couchbase, MongoDB**)
- **Графовые.** Упор на установление произвольных связей между данными (**OrientDB, Neo4j**)
- <http://nosql-database.org>



# СУБД

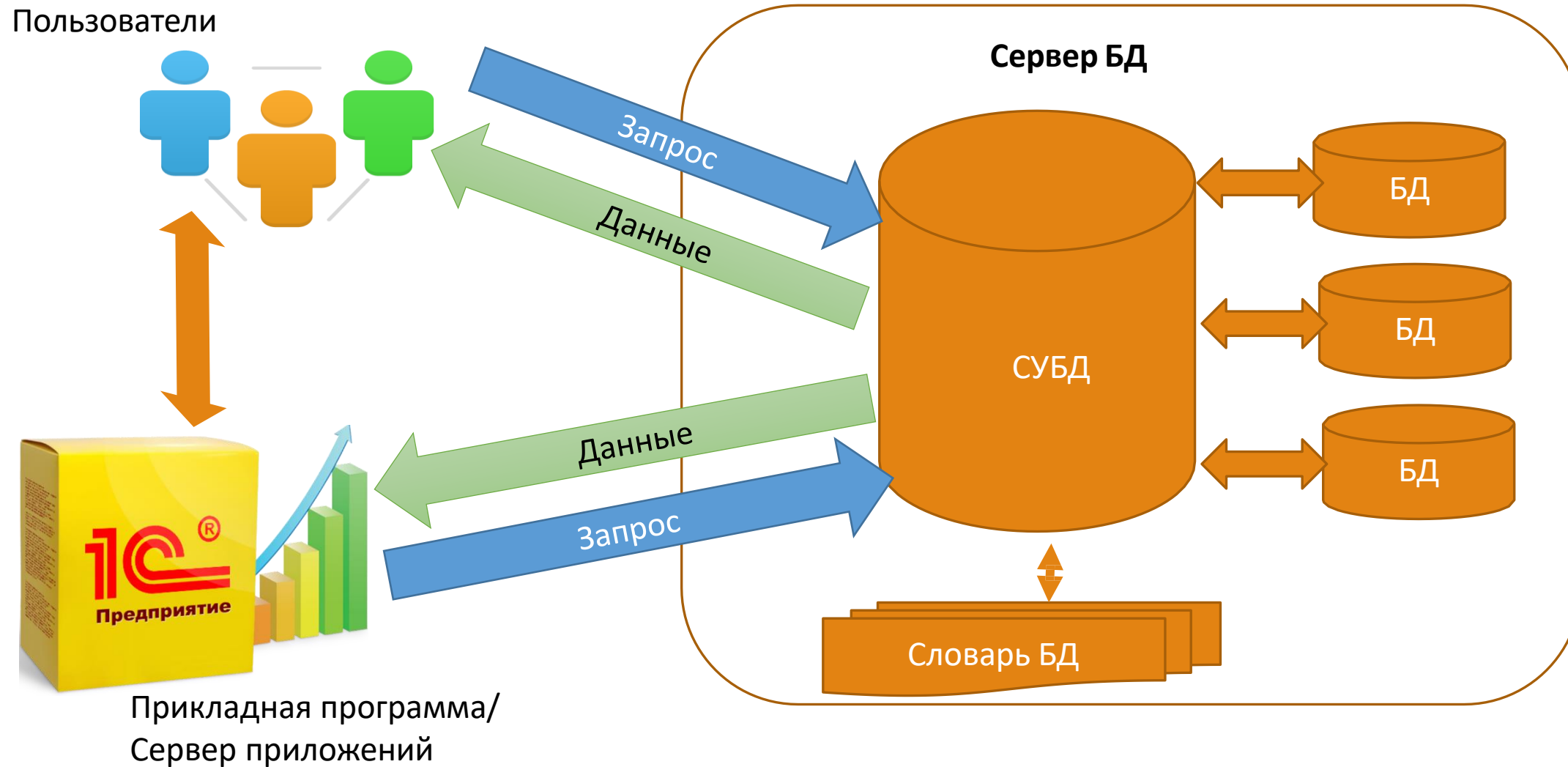
- **Система управления базами данных (СУБД)** – совокупность программных средств, предназначенная для модификации и извлечения из БД необходимых пользователю (прикладной программе) данных, а также для создания баз данных, поддержания их в работоспособном состоянии, обеспечения безопасности баз данных и решения других задач администрирования.
- **СУБД** — это специальное программное обеспечение, предназначенное для управления базами данных





PostgreSQL

# Общее представление СУБД



# Классификация систем баз данных по способу организации (архитектуре)







PostgreSQL

# Наиболее популярные СУБД

Rank			DBMS	Database Model	Score		
Oct 2025	Sep 2025	Oct 2024			Oct 2025	Sep 2025	Oct 2024
1.	1.	1.	Oracle	Relational, Multi-model ⓘ	1212.77	+42.15	-96.67
2.	2.	2.	MySQL	Relational, Multi-model ⓘ	879.66	-12.11	-143.09
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model ⓘ	715.05	-2.27	-87.04
4.	4.	4.	PostgreSQL	Relational, Multi-model ⓘ	643.20	-13.97	-8.96
5.	5.	5.	MongoDB +	Document, Multi-model ⓘ	368.01	-12.49	-37.20
6.	6.	↑ 7.	Snowflake	Relational	198.65	+8.46	+58.05
7.	7.	↓ 6.	Redis	Key-value, Multi-model ⓘ	142.33	-2.84	-7.30
8.	↑ 9.	↑ 14.	Databricks	Multi-model ⓘ	128.80	+4.74	+43.21
9.	↓ 8.	9.	IBM Db2	Relational, Multi-model ⓘ	122.37	-1.81	-0.40
10.	10.	↓ 8.	Elasticsearch	Multi-model ⓘ	116.67	-1.59	-15.18

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

# Основы архитектуры PostgreSQL



PostgreSQL

# PostgreSQL

- PostgreSQL - это система управления базами данных (СУБД) корпоративного класса с открытым исходным кодом, поддерживаемая более 30-лет сообществом разработчиков ПО
- СУБД PostgreSQL является SQL-совместимой и полностью соответствует требованиям ACID (атомарность, согласованность, изолированность, надежность)
- PostgreSQL поддерживает как реляционные (SQL), так и нереляционные (JSON) запросы
- СУБД PostgreSQL используется как основная СУБД для многих веб-приложений, а также мобильных и аналитических приложений



PostgreSQL

# Языковая поддержка

- PostgreSQL поддерживает большинство популярных языков программирования:
  - Python
  - Java
  - C #
  - C / C ++
  - Ruby
  - JavaScript (Node.js)
  - Perl
  - Go
  - Tcl



# Расширяемость

- PostgreSQL спроектирован с расчетом на расширяемость
- Из коробки доступен обширный набор дополнительных модулей – расширений, которые позволяют:
  - работать с внешними источниками данных
  - запускать специальные фоновые процессы для выполнения дополнительных задач
  - выполнять низкоуровневые обращения к системным структурам данных
- Прикладные программисты могут:
  - создавать собственные типы данных на основе уже имеющихся (составные типы, диапазоны, массивы, перечисления)
  - писать хранимые процедуры и функции для обработки данных в БД (в том числе триггеры)
  - писать расширения (языке программирования Си), которые добавляют необходимый функционал и, обычно, могут подключаться даже к работающему серверу
- Если вам не нравится какая-либо часть системы, вы всегда можете разработать собственный плагин



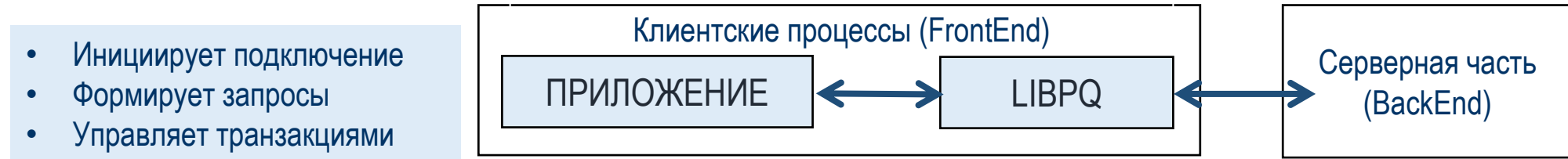
PostgreSQL

# Клиент-серверная СУБД

- PostgreSQL - это СУБД клиент-серверного типа с многопроцессной архитектурой, работающая на одном хосте
- Набор **служебных процессов**, совместно управляющих одним кластером БД, называется «сервером PostgreSQL»
- Один сервер PostgreSQL может управлять несколькими конкурентными клиентскими подключениями
  - Каждое подключение к СУБД имеет свой уникальный идентификатор сессии (pid)
- Текущие активности сервера PostgreSQL - представление **pg\_catalog.pg\_stat\_activity**



# Клиентская часть (FrontEnd)

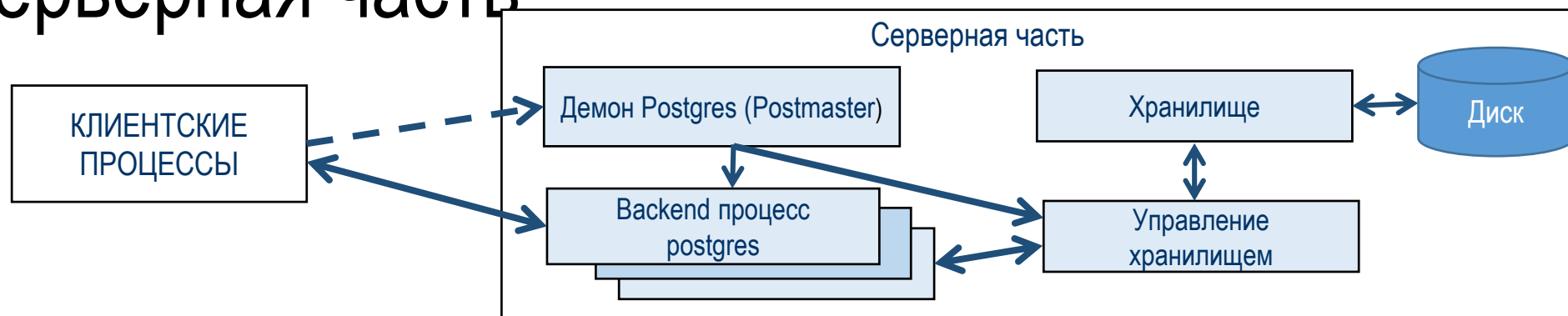


- **FrontEnd** включает в себя:
  - клиентское приложение
  - драйвер, реализующий протокол взаимодействия с сервером (клиенты SQL - библиотека LIBPQ)
- **Библиотека LIBPQ** – основная библиотека, реализующая стандартный протокол взаимодействия
  - входит в состав дистрибутива PostgreSQL и используется всеми штатными утилитами PostgreSQL (в т.ч. psql)
  - содержит набор функций, позволяющих клиентской программе устанавливать соединение с сервером и обмениваться информацией с помощью SQL-запросов
  - для многих языков программирования есть свои библиотеки, которые основаны на libpq
- Для связи с сервером БД используется протокол TCP/IP



PostgreSQL

# Серверная часть



- Серверная часть включает в себя:
  - управляющий серверный процесс-демон postgres (Postmaster)
  - набор служебных процессов
  - хранилище данных
- Хранилище данных использует систему буферизации и является посредником между процессами PostgreSQL и физическим диском
  - на сервере PostgreSQL может существовать только одно хранилище – кластер БД
  - несколько Backend-процессов могут одновременно иметь доступ к информации из хранилища
  - за управление хранилищем отвечают фоновые служебные процессы





# Основные процессы

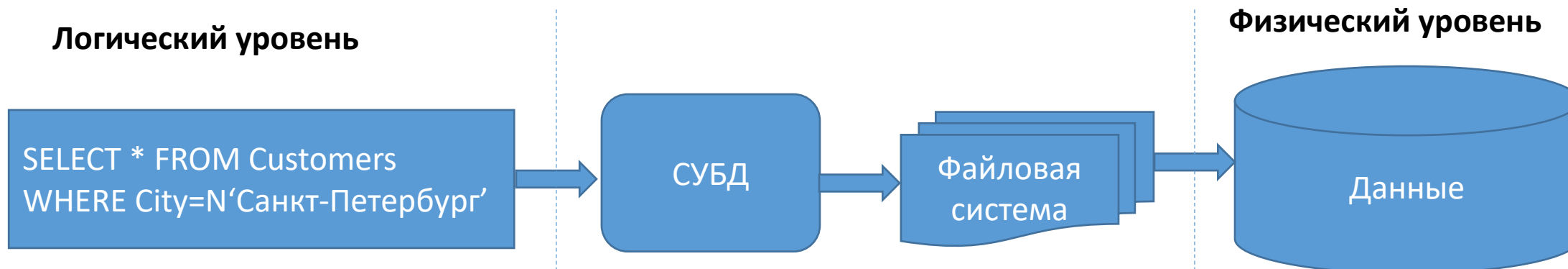
- **FrontEnd процессы** - клиентские приложения:
  - используют PostgreSQL в качестве менеджера баз данных
  - соединение может происходить через TCP/IP или локальные сокеты
- **Демон postmaster** - это основной процесс PostgreSQL:
  - прослушивание через порт/сокет входящих клиентских подключений
  - создание BackEnd процессов и выделение им ресурсов
- **BackEnd процессы:**
  - аутентификация клиентских подключений
  - управление запросами и отправка результатов клиентским приложениям
  - выполнение внутренних задач (служебные процессы)



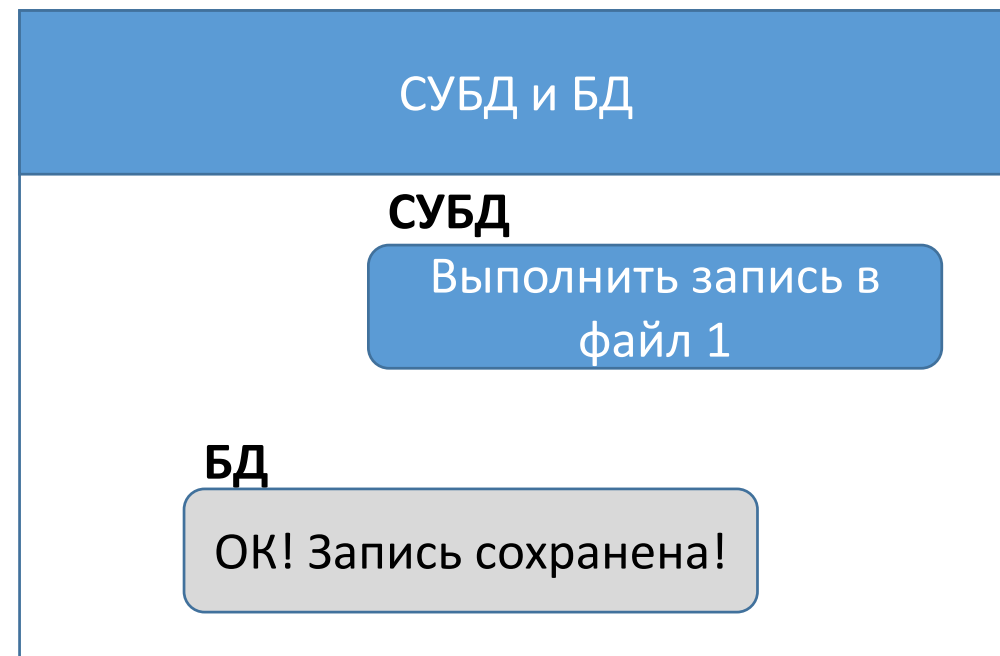
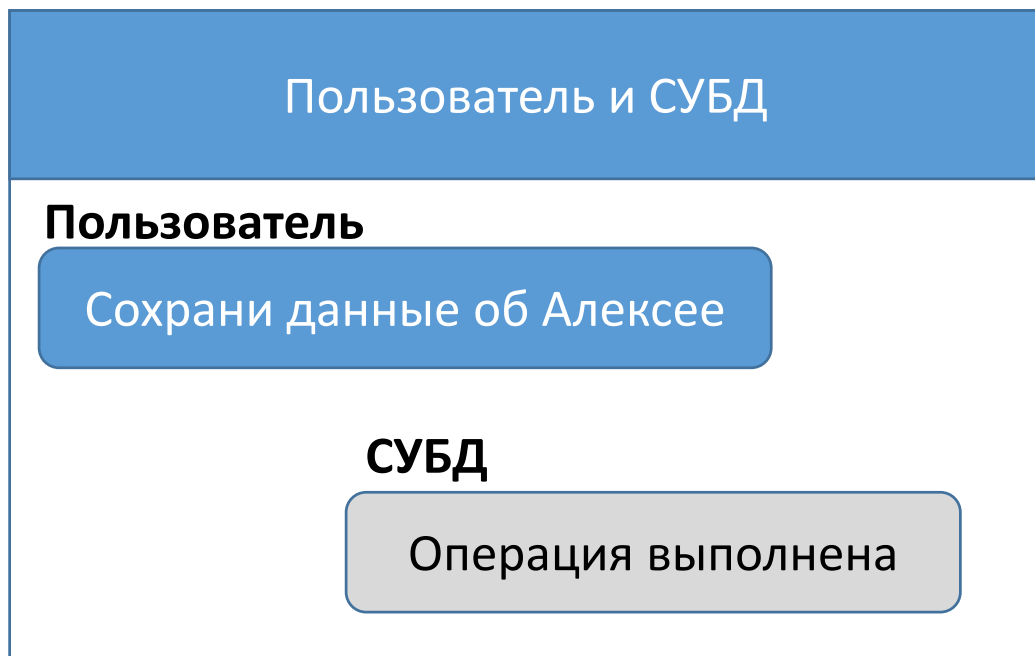
PostgreSQL

# Процесс взаимодействия с БД

- Клиент
  - Используя язык запросов СУБД формулирует требования к результату
  - Опирается на знание логической структуры БД
- СУБД
  - Получает запрос от клиента
  - Анализирует разрешения
  - Анализирует и оптимизирует запрос
  - Создает план выполнения запроса, опираясь на знание физической структуры данных
  - Выполняет запрос
  - Возвращает результат клиенту



# Процесс взаимодействия с БД





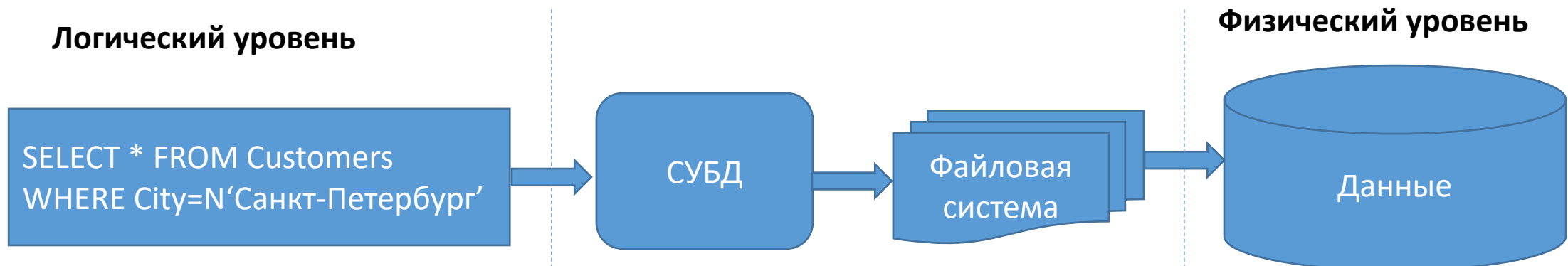
# Взаимодействие с БД

- Для работы с реляционной СУБД существует два основных подхода:
  - работа с **библиотекой**, которая соответствует конкретной СУБД и позволяет использовать для работы с БД **язык БД**
  - работа с **ORM**, которая использует объектно-ориентированный подход для работы с БД и автоматически генерирует код на **языке БД**
    - \*используется программистами клиентских приложений
    - ORM выполняет запросы, которые вы не ожидаете
    - Ваши запросы могут быть не оптимизированы и вы об этом не узнаете



# Высокоуровневые языки запросов

- **Язык БД** - используется для описания структуры БД, управления данными, управления правами доступа к БД и ее объектам, управления транзакциями.
- Современные языки запросов – декларативные языки:
  - Запрос задает требования к результату (что нужно получить), но не алгоритм его получения
  - Декларативные языки позволяют получить высокую эффективность





PostgreSQL

# Подключение к СУБД с использованием клиентской библиотеки

- Строка подключения включает:
  - имя сервера БД (или IP адрес и порт)
  - имя базы данных
  - учетную запись пользователя
  - и другие параметры, необходимые для установки исходного подключения
- Формат: `Key1=Value1;Key2=Value2;Key3=Value3;`



# Подключение к СУБД с использованием клиентской библиотеки

```
pip install psycopg2
```

```
import psycopg2
conn = psycopg2.connect(
    host="vpngw.Avalon.ru",
    port=5432,
    database="dbSQL",
    user="pguser",
    password="Pa$$w0rd")

cur = conn.cursor()
cur.execute('select empid,lastname,firstname
from "HR"."Employees";')
row = cur.fetchone()
while row is not None:
    print(row)
    row = cur.fetchone()
```

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import psycopg2
>>> conn = psycopg2.connect(host = 'vpngw.avalon.ru',
                             port = 5432,
                             database='dbSQL',
                             user='pguser',
                             password='Pa$$w0rd')

>>> conn.info
<psycopg2.extensions.ConnectionInfo object at 0x03FC7DB0>
>>> cur = conn.cursor()
>>> cur.execute('select empid,lastname,firstname from "HR"."Employees";')
>>> print("The number of Employees: ", cur.rowcount)
The number of Employees: 10
>>> row = cur.fetchone()
>>> while row is not None:
    print(row)
    row = cur.fetchone()

(2, 'Funk', 'Don')
(3, 'Lew', 'Judy')
(4, 'Peled', 'Yael')
(8, 'Cameron', 'Maria')
(5, 'Buck', 'Sven')
(6, 'Suurs', 'Paul')
(7, 'King', 'Russell')
(9, 'Dolgopyatova', 'Zoya')
(10, 'Johnson', 'Test 1')
(1, 'yyy', 'Sara')
... ,
```

# Знакомство со средой разработки





PostgreSQL

# Подключение к СУБД

- Для подключения необходимо указать:
  - имя сервера БД (или IP-адрес и порт)
  - имя базы данных
  - учетную запись пользователя
  - и другие параметры, необходимые для установки исходного подключения

# Подключение с использованием pgAdmin

Register - Server

General Connection SSL SSH Tunnel Advanced

Host name/addresses: vpngw.avalon.ru

Port: 5432

Maintenance database: postgres

Username: postgres

Kerberos authentication? ☐

Password: .....

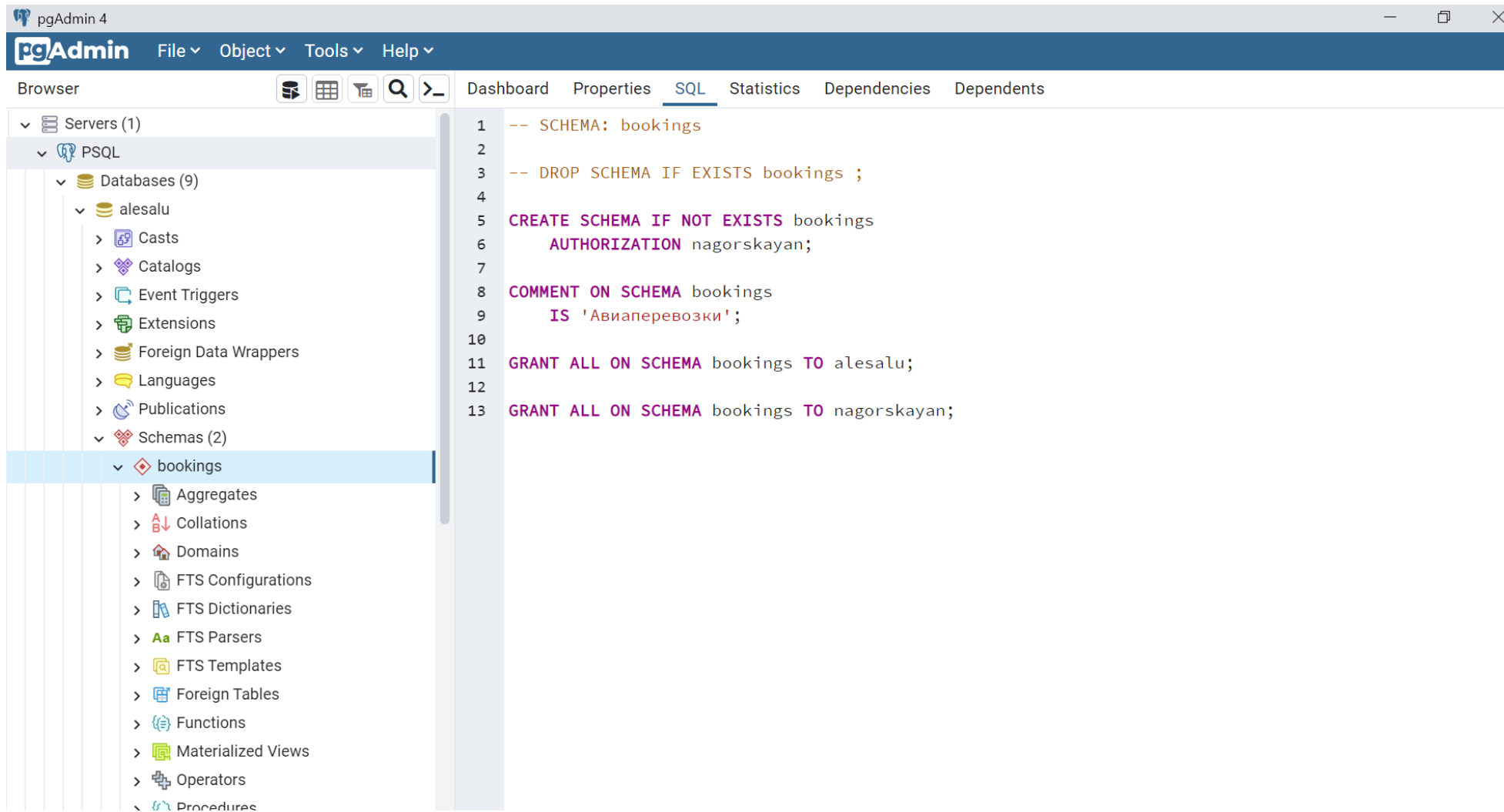
Save password? ☐

Role:

Service:

*i* *?* Close Reset Save

# Интерфейс pgAdmin



The screenshot displays the pgAdmin 4 web interface. The left sidebar shows a tree view of the database structure. Under 'Servers (1)', the 'PSQL' server is expanded, showing 'Databases (9)'. The 'alesalu' database is expanded, and the 'Schemas (2)' section is further expanded, with the 'bookings' schema selected and highlighted in blue. The main panel on the right shows the SQL editor for the 'bookings' schema, with the 'SQL' tab active. The editor contains the following SQL code:

```
1  -- SCHEMA: bookings
2
3  -- DROP SCHEMA IF EXISTS bookings ;
4
5  CREATE SCHEMA IF NOT EXISTS bookings
6    AUTHORIZATION nagorskayan;
7
8  COMMENT ON SCHEMA bookings
9    IS 'Авиаперевозки';
10
11 GRANT ALL ON SCHEMA bookings TO alesalu;
12
13 GRANT ALL ON SCHEMA bookings TO nagorskayan;
```

# Подключение с использованием DBeaver

Создать соединение

**Настройки соединения**

Свойства соединения с PostgreSQL

Главное PostgreSQL Свойства драйвера SSH Proxy SSL

Server

Хост:  Порт:

База данных:

Аутентификация

Аутентификация:

Пользователь:

Пароль:  ☒ Сохранять пароль локально

Advanced

Session role:  Локальный клиент:

Вы можете использовать системные переменные в параметрах.

Драйвер: PostgreSQL



PostgreSQL

# Интерфейс DBeaver

DBeaver 21.2.0 - <postgres> Script-3

Файл Редактирование Навигация Поиск Редактор SQL База данных Окна Справка

SQL Commit Rollback Auto postgres dbo@postgres

Базы данных

Введите часть имени объекта для поиска

- postgres
  - Схемы
    - bookings
    - dbo
    - public
  - Роли
  - Администрирование
  - Расширения
  - Хранилище
  - Системные объекты
    - Методы доступа
    - Кодировки
    - Правила сортировки
    - Оболочки внешних данных
    - Серверы удалённого доступа
  - Языки
    - internal
    - c
    - sql
    - plpgsql
  - Доступные расширения
  - Настройки

```
SELECT *  
FROM dbo."NewTable";
```

NewTable 1

SELECT \* FROM dbo."NewTable" Введите SQL выражение чтобы отфильтровать результаты

id
1



PostgreSQL

# Знакомство с БД



PostgreSQL

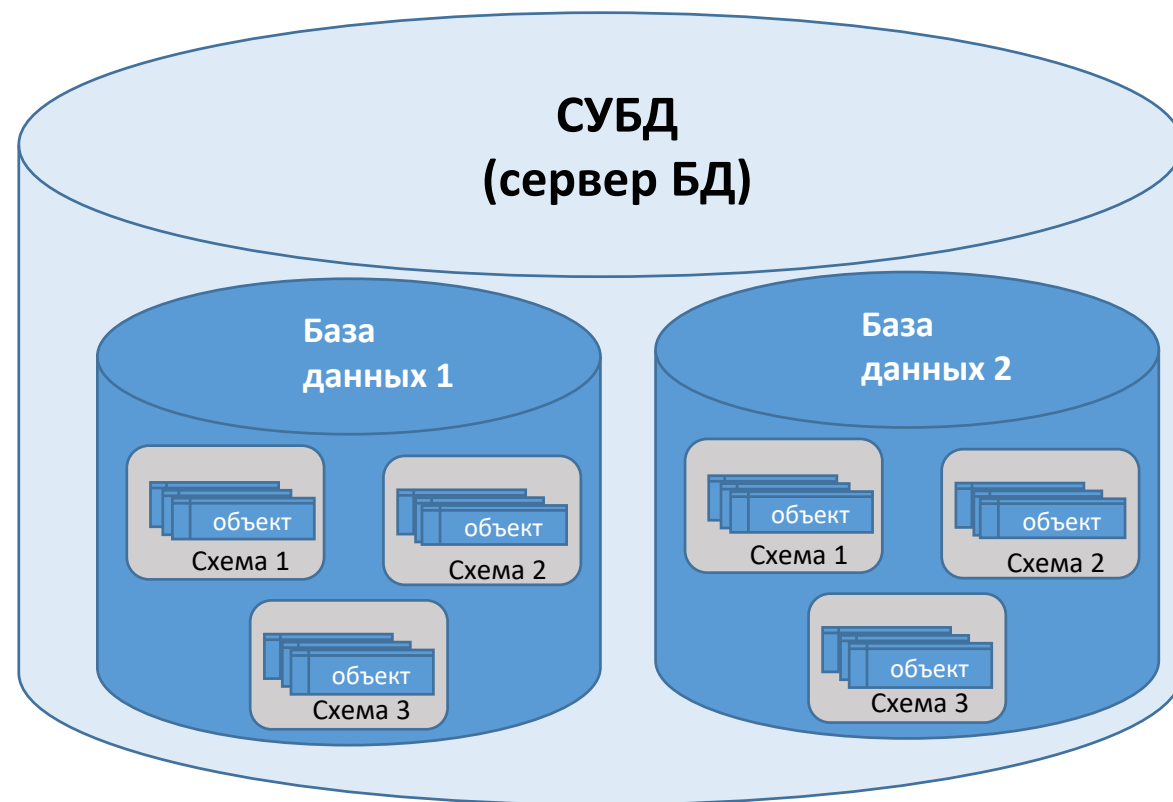
# Именованние объектов БД

Имя\_сервера.Имя\_БД.Имя\_схемы.Имя\_объекта

Пример: Mia-SQL.dbSQL.dbo.Client

- Правила именования объектов
  - Должно начинаться с буквы
  - Может быть длиной до 128 символов\*
  - Должно содержать только символы A–Z, a–z, 0–9, \_, \$ и #
  - Не должно совпадать с именем другого объекта БД
  - Не должно совпадать с зарезервированным словом языка
  - Ограничителем имени является **двойная кавычка** (имена нарушающие правила или чувствительные к регистру)

```
SELECT *  
FROM dbo."NewTable";
```





PostgreSQL

# БД dbSQL

Имя БД

dbSQL

HR

Employees

Production

Categories

Products

Suppliers

Sales

Customers

OrderDetails

Orders

Shippers

Имя  
схемы

Имя  
таблицы

Имя поля

