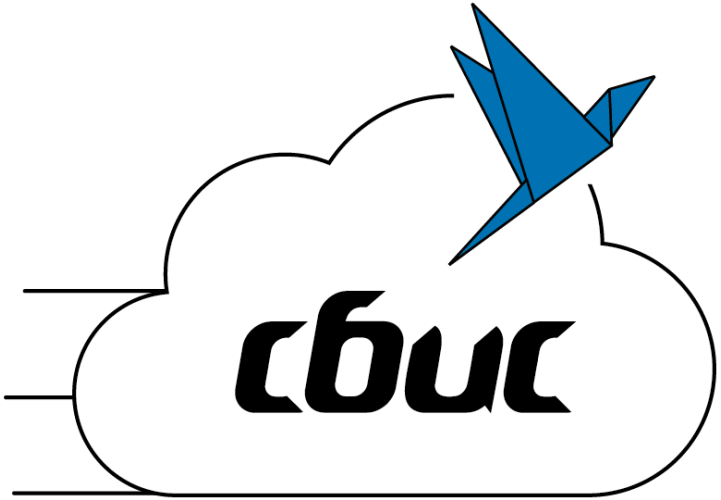


PostgreSQL для начинающих

#1: Основы SQL

Кирилл Боровиков / Компания «Тензор», технический директор / explain.tensor.ru, sbis.ru

SQL – Structured Query Language

SQL (МФА: [ˈɛsˈkjuˈɛl]; аббр. от англ. *Structured Query Language* — «язык структурированных запросов») — декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных.

https://ru.wikipedia.org/wiki/SQL

- язык – декларативный
- цель – управление данными
- база – реляционная

SQL – Structured Query Language

императивные = «как»

декларативные = «что»

SQL – Structured Query Language

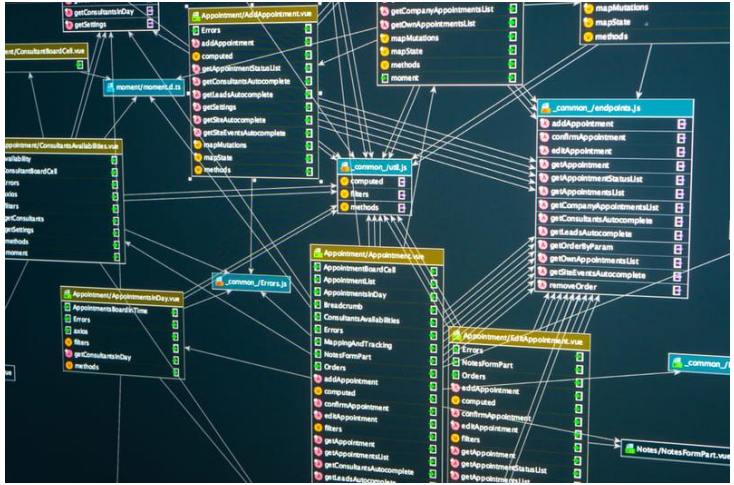
управление операциями

состояния, события

управление данными

строка -> таблица -> база

SQL – Structured Query Language



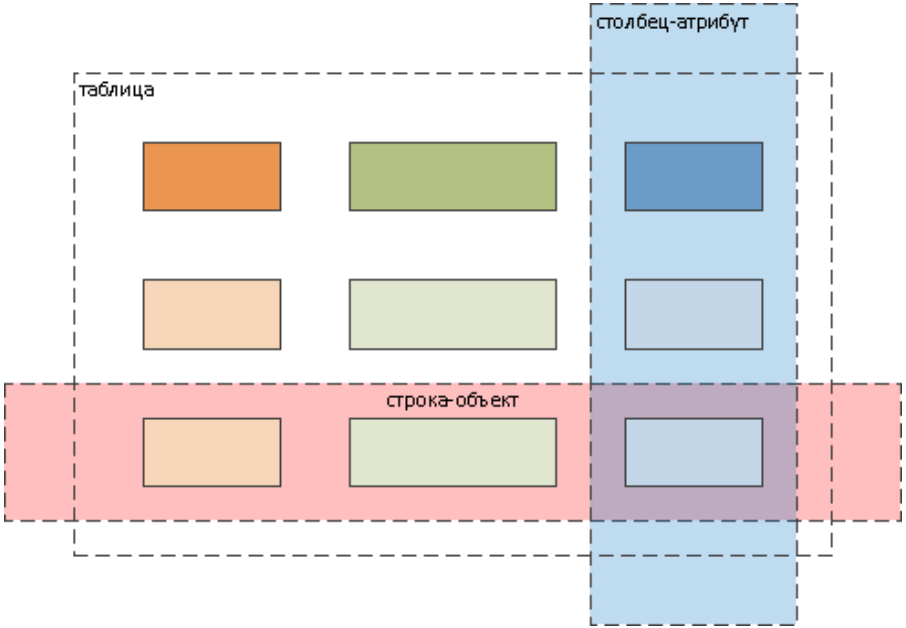
нереляционные

реляционные

redis mongoDB neo4j TARANTOOL elasticsearch KeyDB

PostgreSQL ORACLE DATABASE Microsoft SQL Server MySQL MariaDB

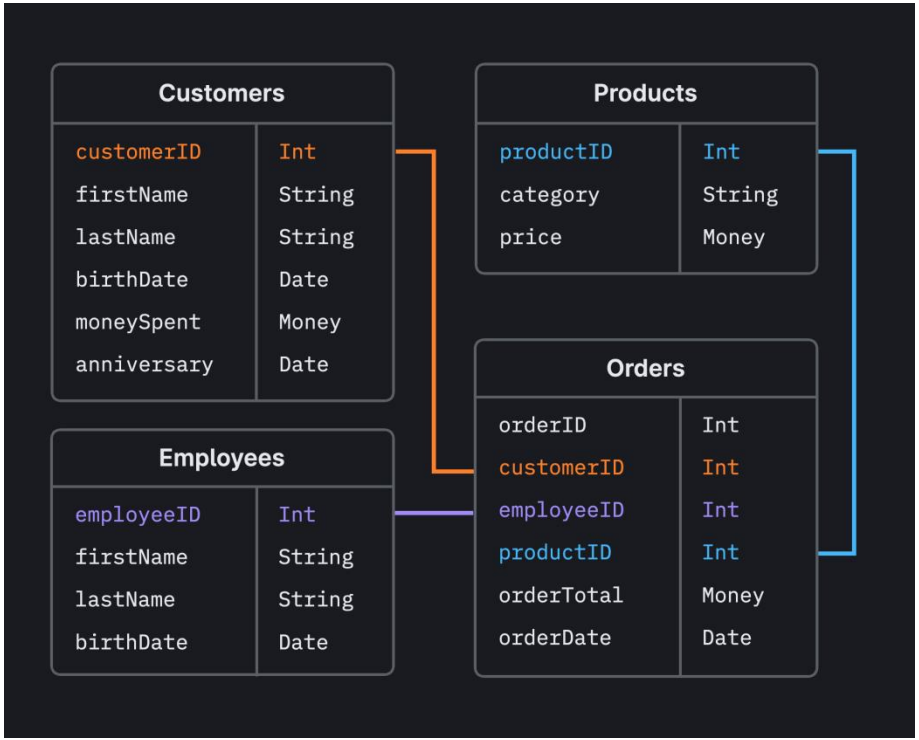
SQL – Structured Query Language



Реляционная БД:

- заранее установленные связи
- таблицы из строк и столбцов
- столбец – тип атрибута
- строка – атрибуты одного объекта
- ячейка (поле записи) – значение

SQL – Structured Query Language



Реляционная БД:

- первичный ключ в таблице (Primary Key, **PK**)

- внешние ключи на другие таблицы (Foreign Keys, **FK**)

- уникальные не-первичные ключи (Unique Keys)

SQL – Structured Query Language

SQL-86 – первый вариант принятый ANSI/ISO

SQL-89 – ... немного поменяли

SQL-92 – ... поменяли уже побольше

SQL:1999 – регулярные выражения, рекурсивные запросы, триггеры, нескалярные типы данных

SQL:2003 – XML, оконные функции и генераторы последовательностей

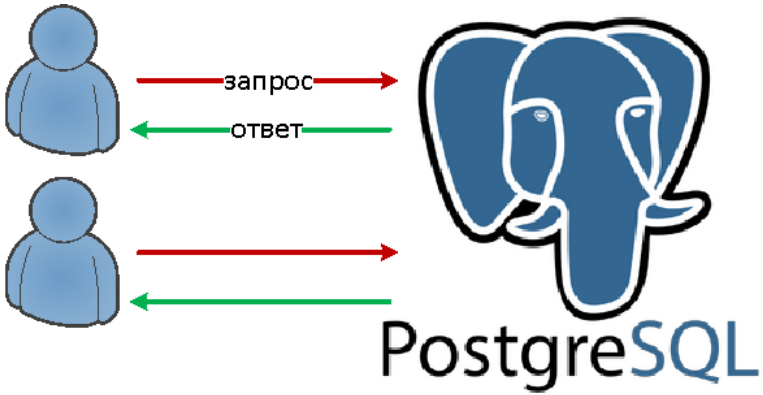
SQL:2006 – XQuery для XML

SQL:2008 – расширены оконные функции

SQL:2011 – поддержка FETCH и PERIOD FOR

SQL:2016 – RLS, JSON, pattern matching

Особенности PostgreSQL



Клиент-серверная архитектура:

- запрос от клиента

- ответ от сервера

- собственный протокол over TCP/IP

- нужно клиентское приложение (или библиотека)

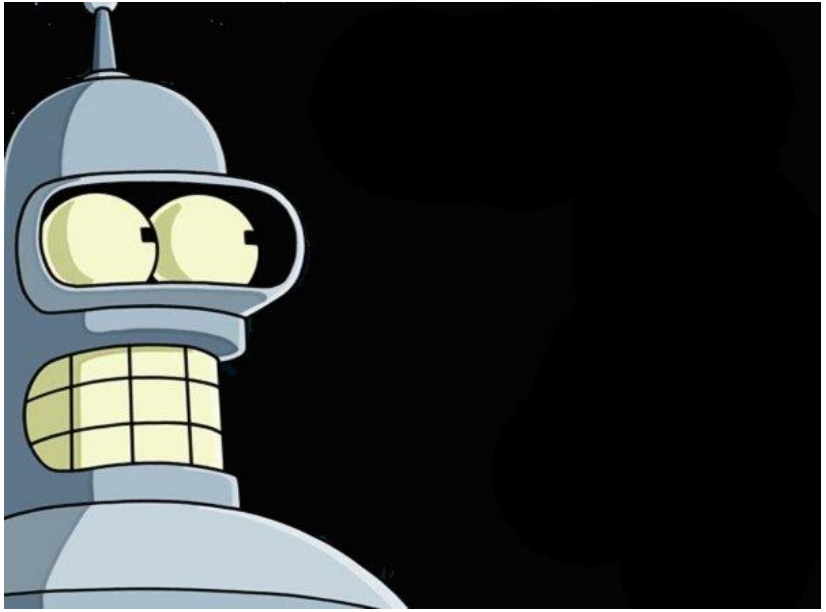
Особенности PostgreSQL

Для человека:

- **psql** (нативная консоль)
- pgAdmin
- DBeaver
- Navicat

https://wiki.postgresql.org/wiki/PostgreSQL_Clients

Особенности PostgreSQL



Для робота:

- libpq (C)

- libpqxx (C++)

- psycopg (Python)

- node-postgres (JavaScript)

https://wiki.postgresql.org/wiki/List_of_drivers

Базовые команды SQL

Создать/изменить/удалить объект СУБД

`CREATE | ALTER | DROP`

Вставить/изменить/удалить записи [таблицы]

`INSERT | UPDATE | DELETE/TRUNCATE`

Получить выборку записей

`SELECT` – **99%, однако**

Базовые команды SQL

«Чтобы ~~продать~~ прочитать что-нибудь ненужное, нужно сначала ~~купить~~ записать что-нибудь ненужное, а у нас ~~денег~~ базы нет.»

Кот Матроскин, «Трое из Простоквашино»

Базовые команды SQL

CREATE

- **CREATE ACCESS METHOD** – создать новый метод доступа
- **CREATE AGGREGATE** – создать агрегатную функцию
- **CREATE CAST** – создать приведение
- **CREATE COLLATION** – создать правило сортировки
- **CREATE CONVERSION** – создать перекодировку
- **CREATE DATABASE** – создать базу данных
- **CREATE DOMAIN** – создать домен
- **CREATE EVENT TRIGGER** – создать событийный триггер
- **CREATE EXTENSION** – установить расширение
- **CREATE FOREIGN DATA WRAPPER** – создать новую обёртку сторонних данных
- **CREATE FOREIGN TABLE** – создать стороннюю таблицу
- **CREATE FUNCTION** – создать функцию
- **CREATE GROUP** – создать роль в базе данных
- **CREATE INDEX** – создать индекс
- **CREATE LANGUAGE** – создать процедурный язык
- **CREATE MATERIALIZED VIEW** – создать материализованное представление
- **CREATE OPERATOR** – создать оператор
- **CREATE OPERATOR CLASS** – создать класс операторов
- **CREATE OPERATOR FAMILY** – создать семейство операторов
- **CREATE POLICY** – создать новую политику защиты на уровне строк для таблицы
- **CREATE PROCEDURE** – создать процедуру
- **CREATE PUBLICATION** – создать публикацию
- **CREATE ROLE** – создать роль в базе данных
- **CREATE RULE** – создать правило перезаписи
- **CREATE SCHEMA** – создать схему
- **CREATE SEQUENCE** – создать генератор последовательности
- **CREATE SERVER** – создать сторонний сервер
- **CREATE STATISTICS** – создать расширенную статистику
- **CREATE SUBSCRIPTION** – создать подписку
- **CREATE TABLE** – создать таблицу
- **CREATE TABLE AS** – создать таблицу из результатов запроса
- **CREATE TABLESPACE** – создать табличное пространство
- **CREATE TEXT SEARCH CONFIGURATION** – создать конфигурацию текстового поиска
- **CREATE TEXT SEARCH DICTIONARY** – создать словарь текстового поиска
- **CREATE TEXT SEARCH PARSER** – создать анализатор текстового поиска
- **CREATE TEXT SEARCH TEMPLATE** – создать шаблон текстового поиска
- **CREATE TRANSFORM** – создать трансформацию
- **CREATE TRIGGER** – создать триггер
- **CREATE TYPE** – создать новый тип данных
- **CREATE USER** – создать роль в базе данных
- **CREATE USER MAPPING** – создать сопоставление пользователя для стороннего сервера
- **CREATE VIEW** – создать представление

```
CREATE DATABASE tst;
```

https://postgrespro.ru/docs/postgresql/15/sql-createdatabase

```
CREATE TABLE tbl(
  k          -- имя поля
    integer -- тип поля
, v
    text
);
```

https://postgrespro.ru/docs/postgresql/15/sql-createtable

Основы SQL

Базовый синтаксис

```
-- это однострочный комментарий
/* а это -
         - многострочный */

fld      -- это поле/столбец
Fld      -- это то же самое поле
FLD      -- ... и это – все оно же (приводится к lower case)

"Fld"    -- а вот это – тоже поле, но совсем другое (кавычки дают регистрозависимость)

'str'    -- это строка

'st''r'  -- это строка с одинарным апострофом
E'st\'r' -- ... и это – она же
$$st'r$$ -- ... и даже вот это
```

https://postgrespro.ru/docs/postgresql/15/sql-syntax-lexical

Основы SQL

Базовые типы данных

- числовые
- символьные
- даты/времени
- логический

https://postgrespro.ru/docs/postgresql/15/datatype

Основы SQL

Базовые типы данных

числовые

Имя	Размер	Описание	Диапазон
smallint	2 байта	целое в небольшом диапазоне	-32768 .. +32767
integer	4 байта	типичный выбор для целых чисел	-2147483648 .. +2147483647
bigint	8 байт	целое в большом диапазоне	-9223372036854775808 .. 9223372036854775807
decimal	переменный	вещественное число с указанной точностью	до 131072 цифр до десятичной точки и до 16383 — после
numeric	переменный	вещественное число с указанной точностью	до 131072 цифр до десятичной точки и до 16383 — после
real	4 байта	вещественное число с переменной точностью	точность в пределах 6 десятичных цифр
double precision	8 байт	вещественное число с переменной точностью	точность в пределах 15 десятичных цифр
smallserial	2 байта	небольшое целое с автоувеличением	1 .. 32767
serial	4 байта	целое с автоувеличением	1 .. 2147483647
bigserial	8 байт	большое целое с автоувеличением	1 .. 9223372036854775807

https://postgrespro.ru/docs/postgresql/15/datatype-numeric

Основы SQL

Базовые типы данных

символьные

Имя	Описание
character varying(*n*), varchar(*n*)	строка ограниченной переменной длины
character(*n*), char(*n*)	строка фиксированной длины, дополненная пробелами
text	строка неограниченной переменной длины

https://postgrespro.ru/docs/postgresql/15/datatype-character

Основы SQL

Базовые типы данных

даты/времени

Имя	Размер	Описание	Наименьшее значение	Наибольшее значение	Точность
`timestamp [ (p) ] [ without time zone ]`	8 байт	дата и время (без часового пояса)	4713 до н. э.	294276 н. э.	1 микросекунда
`timestamp [ (p) ] with time zone`	8 байт	дата и время (с часовым поясом)	4713 до н. э.	294276 н. э.	1 микросекунда
`date`	4 байта	дата (без времени суток)	4713 до н. э.	5874897 н. э.	1 день
`time [ (p) ] [ without time zone ]`	8 байт	время суток (без даты)	00:00:00	24:00:00	1 микросекунда
`time [ (p) ] with time zone`	12 байт	время дня (без даты), с часовым поясом	00:00:00+1559	24:00:00-1559	1 микросекунда
`interval [ поля ] [ (p) ]`	16 байт	временной интервал	-178000000 лет	178000000 лет	1 микросекунда

https://postgrespro.ru/docs/postgresql/15/datatype-datetime

Основы SQL

Базовые типы данных

логический (TRUE, FALSE, … NULL!)

Имя	Размер	Описание
boolean	1 байт	состояние: истина или ложь

https://postgrespro.ru/docs/postgresql/15/datatype-boolean

Основы SQL

Специальные типы данных

двоичные данные

перечисления

геометрические

сетевые адреса

битовые строки

вектора текстового поиска

UUID, XML, JSON

массивы

диапазоны

псевдотипы

https://postgrespro.ru/docs/postgresql/15/datatype

Базовые команды SQL

CREATE

```
CREATE DATABASE tst;
```

https://postgrespro.ru/docs/postgresql/15/sql-createdatabase

```
CREATE TABLE tbl(
  k         -- имя поля
    integer -- тип поля
, v
    text
);
```

https://postgrespro.ru/docs/postgresql/15/sql-createtable

Базовые команды SQL

INSERT

```
INSERT INTO tbl(
  k -- указываем имена вставляемых полей
, v
)
VALUES
  (1, '1st string') -- перечисляем вставляемые строки
, (102, 'another string')
, (3, NULL); -- вовсе не ''
```

https://postgrespro.ru/docs/postgresql/15/sql-insert
https://postgrespro.ru/docs/postgresql/15/sql-values

Базовые команды SQL

UPDATE

```
UPDATE
  tbl
SET
  k = k - 100
, v = '2nd string'
WHERE
  k = 102;
```

https://postgrespro.ru/docs/postgresql/15/sql-update

Базовые команды SQL

DELETE

```
DELETE FROM
  tbl
WHERE
  v IS NULL;
```

https://postgrespro.ru/docs/postgresql/15/sql-delete

Основы SQL

NULL-логика

	T	F	N
NOT	F	T	N

=	T	F	N
T	T	F	N
F	F	T	N
N	N	N	N

и IS NOT DISTINCT FROM

IS	T	F	N
T	T	F	F
F	F	T	F
N	F	F	T

<>	T	F	N
T	F	T	N
F	T	F	N
N	N	N	N

и IS DISTINCT FROM

IS NOT	T	F	N
T	F	T	T
F	T	F	T
N	T	T	F

		IS NULL	IS NOT NULL	IS DISTINCT FROM NULL	IS NOT DISTINCT FROM NULL	IS DISTINCT FROM (NULL,NULL)	IS NOT DISTINCT FROM (NULL,NULL)
N	N	T	F	T	F	F	T
N	T/F	F	F	T	F	T	F
T/F	N	F	F	T	F	T	F
T/F	T/F	F	T	T	F	T	F

```
coalesce(a, <const>) = <const> ~
  a IS NULL OR
  a = <const>
```

```
a IS DISTINCT FROM b ~
  (a IS     NULL AND b IS NOT NULL) OR
  (a IS NOT NULL AND b IS     NULL) OR
  (a IS NOT NULL AND b IS NOT NULL AND a <> b)
```

```
(a,b) IS DISTINCT FROM (c,d) ~
  (a IS DISTINCT FROM c) OR
  (b IS DISTINCT FROM d)
```

https://habr.com/ru/companies/tensor/articles/491366/

Базовые команды SQL

DELETE

```
DELETE FROM
  tbl
WHERE
  v IS NULL
RETURNING *;
```

https://postgrespro.ru/docs/postgresql/15/sql-delete

```
   k     |  v
integer | text
      3 |
```

Базовые команды SQL

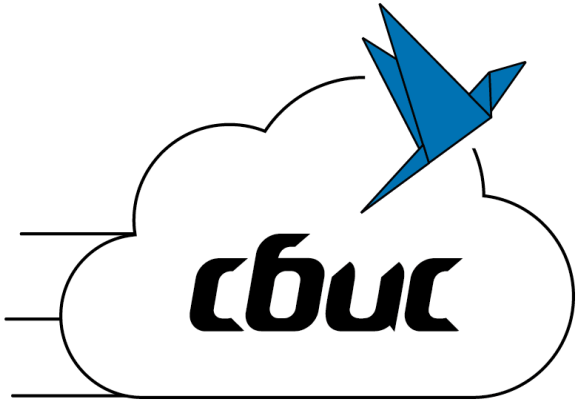
SELECT

```
SELECT
  *
FROM
  tbl;
```

https://postgrespro.ru/docs/postgresql/15/sql-select

```
   k     |  v
integer | text
      1 | 1st string
      2 | 2nd string
```

Базовые команды SQL



Спасибо за внимание!

Боровиков Кирилл

kilor@tensor.ru / https://n.sbis.ru/explain

sbis.ru / tensor.ru