

MTC True Tech

Talents



MTC

SQL. ИСПОЛЬЗОВАНИЕ НА ПРАКТИКЕ



MTC Teta



Меня хорошо видно и слышно?



Проверить, идет ли запись



Ставим «+», если все хорошо
«-», если есть проблемы

МТС Тета





Дмитрий Гаврилов

Руководитель Центра компетенций Аналитики

- В данный момент руководитель Центра компетенций Аналитики
- 5 лет опыта работы дата аналитиком в Big Data MTC
- С 2003 года работаю с различными СУБД

МТС Тета



ПОЧЕМУ SQL?

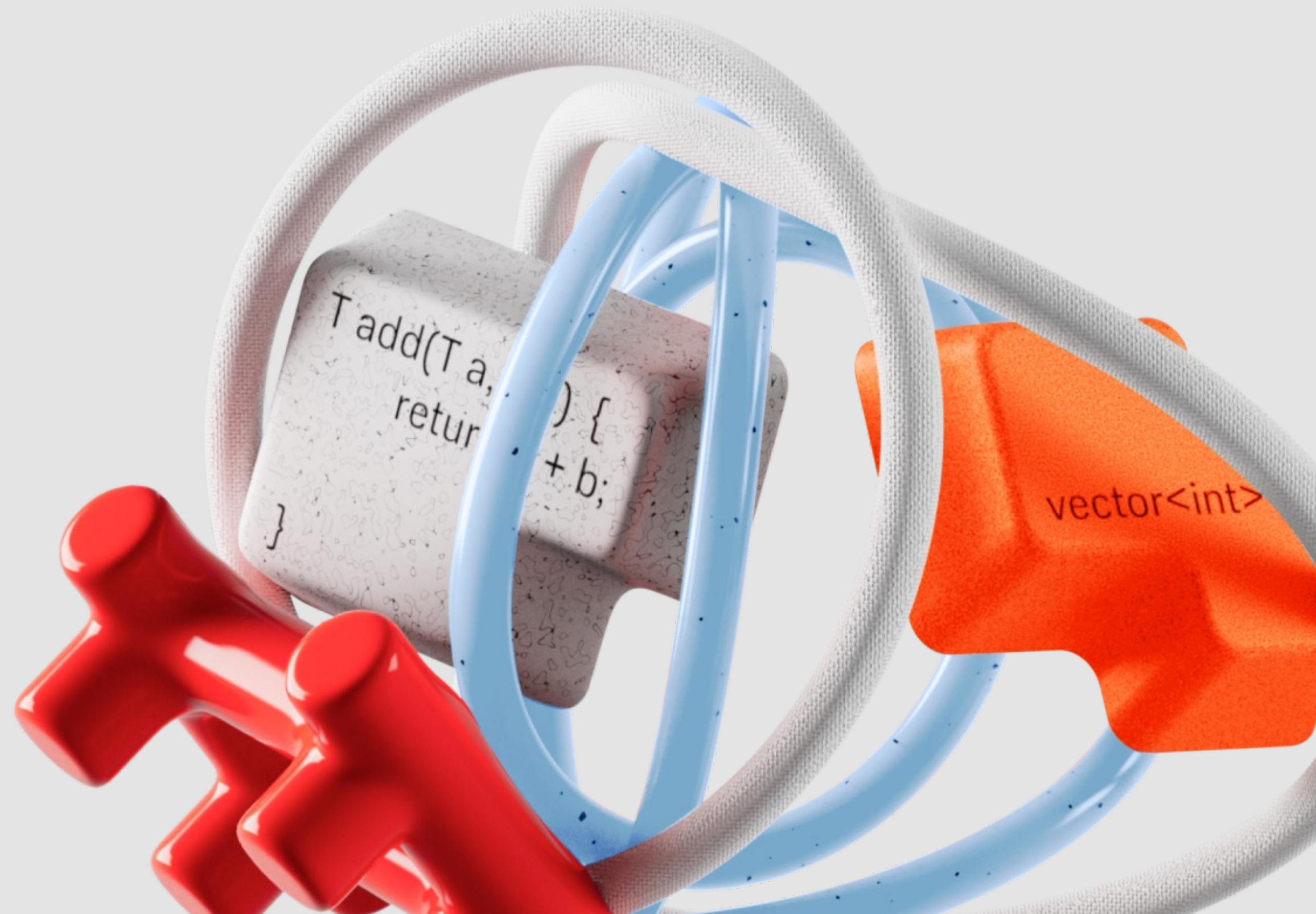
- Без данных нет ML и DS
- SQL — язык для работы с данными

План:

- Где применяется SQL
- SQL и его разновидности (HiveQL, Spark SQL, T-SQL)
- Подключение к онлайн БД для решения задач
- Создание таблиц для задачи, наполнение данными (DDL)
- Задачи на соединения таблиц, фильтрация данных
- Обновление, удаление записей в таблице (DML)
- Обработка NULL, использование Distinct, UNION, VIEW
- Применение агрегатов группировка, сортировка, HAVING
- Обобщённое табличное выражение (Common Table Expression) в SQL
- Задача на применение оконной функции и CTE
- Подзапросы
- Неявное объединение таблиц
- План запроса, оптимизация

ГДЕ ПРИМЕНЯЕТСЯ SQL?

М Т
С



МТС Тета



СУБД, фреймворки

СУБД (Система Управления Базами Данных) — это программное обеспечение для работы с базами данных



Разновидности SQL

T-SQL — дополнен, например, возможностью использовать переменные, функции, выражения:

```
USE TestDatabase
-- Объявление переменных
DECLARE @EmpID int, @EmpName varchar(40)
-- Задание значения переменной @EmpID
SET @EmpID = 1
-- Задание значения переменной @EmpName
SELECT @EmpName = UserName FROM Users WHERE UserID = @EmpID
-- Вывод переменной @EmpName в результат запроса
SELECT @EmpName AS [Employee Name]
GO
```

HiveQL — Hive Query Language, адаптированная версия SQL для фреймворка Hive

- в HiveQL нет некоторых функций, операций и операторов SQL (UPDATE и DELETE statements, INSERT для отдельных строк)
- HiveQL позволяет вставлять пользовательский код для ситуаций, которые не вписываются в типовой SQL, предоставляя соответствующие инструменты для обработки входа и выхода — определенные пользователем функции: User Defined Function (UDF), User Defined Aggregate Function (UDAF), User Defined Tabular Function (UDTF)
- HiveQL не поддерживает типы данных даты и времени, т.к. они рассматриваются как строки.

YQL — Yandex Query Language. Версия SQL, обогащенная различными наборами библиотек для внутренних продуктов компании Яндекс.

Разновидности SQL

Spark SQL — это модуль фреймворка Spark для структурированной обработки распределенных данных, позволяющий выполнять запросы на языке SQL (Structured Query Language). Спарк SQL использует датафреймы (dataframe) для работы с данными.

Датафрейм — это структура данных, представляющая собой таблицу с данными, упорядоченными по именованным столбцам. Спарк SQL поддерживает загрузку данных из различных внешних источников (например, Hive, JSON, Clickhouse,) для формирования датафреймов.

```
spark.read.json("s3n://...")  
  .registerTempTable("json")  
results = spark.sql(  
  """SELECT *  
    FROM people  
    JOIN json ...""")
```

```
results = spark.sql(  
  "SELECT * FROM people")  
names = results.map(lambda p: p.name)
```


DDL

DDL (Data Definition Language) — группа команд, используемых для создания и изменения всех объектов баз данных

Основные команды:

- CREATE DATABASE — создать базу данных
- CREATE TABLE — создать таблицу
- ALTER TABLE — изменить таблицу
- TRUNCATE — удалить все строки, очистить таблицу
- DROP TABLE — удалить таблицу
- DROP DATABASE — удалить базу данных

```
-- создать базу данных Database_Name
CREATE DATABASE school_db;

-- создать таблицу Student
CREATE TABLE student (
    roll_no int ,
    first_name varchar (20) ,
    last_name varchar (20) ,
    st_age int ,
    marks Int);

-- добавить в таблицу Student поле Fathers_Name
ALTER TABLE student ADD fathers_Name varchar(60);
-- удалить из таблицы student колонки st_age, marks
ALTER TABLE student DROP st_age, marks;
-- изменить тип данных колонки student на varchar(25)
ALTER TABLE student MODIFY ( last_Name varchar(25));

-- очистить таблицу student
TRUNCATE TABLE student;

-- удалить таблицу student
DROP TABLE student;
-- удалить базу данных school_db
DROP DATABASE school_db;
```

DML

DML (Data Manipulation Language) — группа операторов, позволяющих извлекать, вставлять и изменять записи в таблицах

Команды:

- SQL — извлечь строки из таблицы
- INSERT — вставить строки в таблицу
- UPDATE — обновить строки в таблице
- DELETE — удалить строки из таблицы

```
-- выбрать все строки из таблицы employees
select *
from employees
```

id	emp_name	salary	dept_id	manager_id
1	Idris	1000	1	1
2	Aweda	2000	2	2
3	Zubair	3000	3	2
4	Young	4000	10	3
5	Babu	5000	1	3
6	John	1000	8	1
7	Ivan	5000	2	2

```
-- удалить всех сотрудников с именем Ivan
delete from employees
where emp_name = 'Ivan'
```

```
-- выбрать все строки из таблицы employees
select *
from employees
```

id	emp_name	salary	dept_id	manager_id
1	Idris	1000	1	1
2	Aweda	2000	2	2
3	Zubair	3000	3	2
4	Young	4000	10	3
5	Babu	5000	1	3
6	John	1000	8	1

TCL (Transaction Control Language) — используется для управления транзакциями

Транзакция — это применение одного или нескольких изменений в БД

Транзакция начинается в момент, когда сделано первое изменение в БД

После всех изменений в БД транзакция:

- Либо откатывается (ROLLBACK) — возвращается к состоянию до транзакции
- Либо применяется (COMMIT), то есть все изменения вступают в силу без возможности откатить (сделать ROLLBACK)

TCL

Основные команды TCL:

- COMMIT — применяет все изменения, сделанные транзакцией, без возможности отката
- ROLLBACK — откатывает все сделанные транзакцией изменения

```
select * from employees;
```

id	emp_name	salary	dept_id	manager_id
1	Idris	1000	1	1
2	Aweda	2000	2	2

```
update employees set salary = salary + 1000;
```

```
rollback
```

```
select * from employees;
```

id	emp_name	salary	dept_id	manager_id
1	Idris	1000	1	1
2	Aweda	2000	2	2

```
update employees set salary = salary + 1000;
```

```
commit;
```

```
rollback;
```

```
select * from employees;
```

id	emp_name	salary	dept_id	manager_id
1	Idris	2000	1	1
2	Aweda	3000	2	2

DCL (Data Control Language) — команды, используемые для предоставления и отзыва привилегий пользователя базы данных

Некоторые объекты, тесно связанные с DCL:

- Пользователь (User) — учетная запись пользователя базы данных, имеющая логин и пароль.
- Роль (Role) — группа пользователей, имеющих схожие необходимые права. Например Роль «Администраторы» или «Разработчики»
- Схема (Schema) — группа таблиц, которых может объединять какой-то общий проект или бизнес-подразделение пользователей

DCL (Data Control Language) — команды, используемые для предоставления и отзыва привилегий пользователя базы данных

Основные команды DCL:

- Grant — выдать права на доступ к объектам БД пользователю или роли
- Revoke — отозвать права на доступ к объектам БД у пользователя или роли

```
-- создаем роль admins
create role admins;

-- даем роли admins права на чтение, изменение,
-- вставку строк в таблицу employees
grant select, update, insert on employees to admins;

-- создаем двух новых пользователей user1, user2
create user user1 IDENTIFIED by password1;
create user user2 IDENTIFIED by password2;

-- добавляем user1 в роль admins
-- user1 будет иметь доступы, которые есть у admins
grant admins to user1;

-- даем пользователю user2 права на чтение employees
-- права выдаются не через роль, а непосредственно
grant select on employees to user2;

-- отзываем все права на таблицу employees у роли admins
-- user1 тоже автоматически потеряет права на employees
revoke all on employees from admins;
```

SQL: Команда SELECT

Основные команды DCL:

- Во фразе SELECT записываются названия необходимых колонок таблицы. Если нужны все — то символ * (звездочка)
- Во фразе FROM указывается название таблицы, из которой требуется выборка
- Во фразе WHERE записывается фильтр на строки, если он нужен.

Фразы SELECT и FROM обязательны в каждом SQL-запросе

Простейший синтаксис:

```
SELECT  column1, ...., column2  
FROM    table  
WHERE   conditions
```

ПЕРЕРЫВ 5 МИНУТ

М Т
С

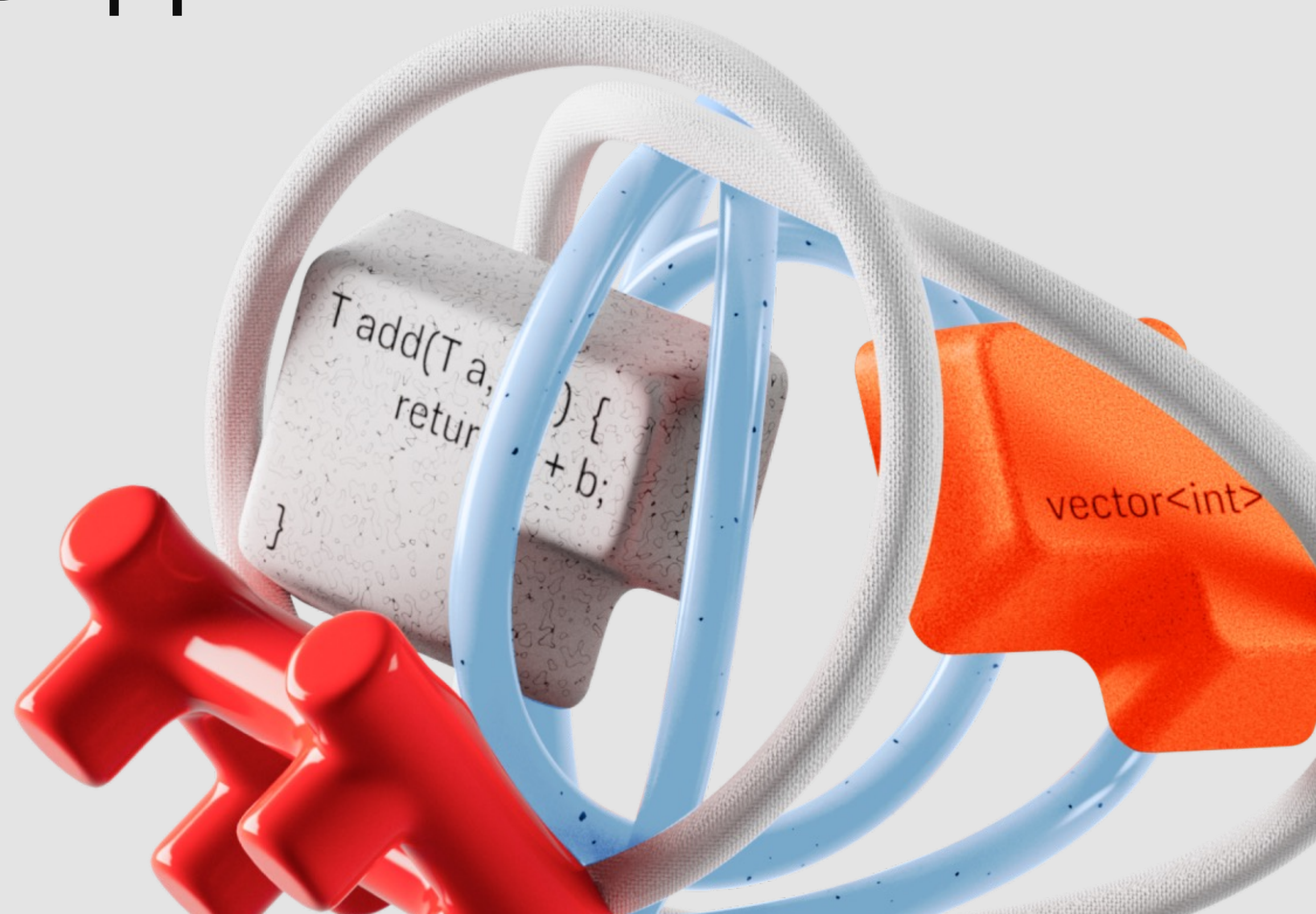


МТС Тета



ПОДКЛЮЧЕНИЕ К ОНЛАЙН БД ДЛЯ РЕШЕНИЯ ЗАДАЧ

М Т
С



МТС Тета



Онлайн тренажер DB-Fiddle



Database: PostgreSQL v17

Run

Update

Fork

Load Example

Star

PRO

Embed

PRO

Collaborate

Sign in

Have any feedback?

Fiddle Title

Практика по SQL

35 characters remaining.

Fiddle Description

300 characters remaining.

Private Fiddle

PRO

☐

This setting cannot be modified after saving the fiddle.

Upgrade to PRO

50% OFF for Early Adopters

Show Keyboard Shortcuts

Build your website for just \$3.88/mth. More value and performance with Namecheap.

ads via Carbon

Schema SQL

```
1 -- Создание таблиц для задачи, наполнение данными (DDL)
2 -- *****
3 create table a (a int);
4 insert into a (a) values (1),(2),(3);
5 create table b (b int);
6 insert into b (b) values (3),(4),(5);
7
8 -- Таблица пользователей
9 create table users (
10     user_id int, -- id пользователя
11     name text, -- имя
12     sex text -- пол
13 );
14
15 insert into users (user_id,name,sex) values
16     (111,'Дмитрий','М'),
17     (122,'Ирина','Ж'),
18     (133,'Егор','М');
19
20 -- Таблица абонентов
21 create table abonents (
22     user_id int, -- id пользователя
23     msisdn text -- номер телефона
24 );
25
26 insert into abonents (user_id,msisdn) values
27     (111,'+79162005990'),
28     (122,'+79262171187'),
29     (144,'+79165104810');
30
31 -- Таблица истории изменения балансов
32 create table balance_history (
33     msisdn text, -- номер телефона
34     ts timestamp, -- дата и время изменения баланса
35     balance float -- баланс (деньги на телефоне)
36 );
37
38 insert into balance_history (msisdn,ts,balance) values
39     ('+79162005990','2024-01-01 00:37:56', 250.01),
40     ('+79162005990','2024-01-01 01:00:36', 243.47),
41     ('+79162005990','2024-01-10 15:37:01', 200.02),
42     ('+79162005990','2024-02-07 10:37:28', 450.02),
43     ('+79262171187','2024-01-08 02:04:49', 200.07),
44     ('+79262171187','2024-01-17 08:47:37', 180.56),
45     ('+79262171187','2024-01-18 09:03:45', 250.47),
46     ('+79262171187','2024-02-01 18:36:50', 10.00),
47     ('+79262171187','2024-03-10 00:37:56', 400.00),
48     ('+79165104810','2024-01-04 04:05:37', 218.09),
49     ('+79165104810','2024-01-06 08:47:37', 100.45);
```

Query SQL

1


DB Fiddle – Crafted with ♥ by Status200 in the United Kingdom.

Terms of Use • Privacy / Cookie Policy • Status200 Ltd © 2014



Создание таблиц

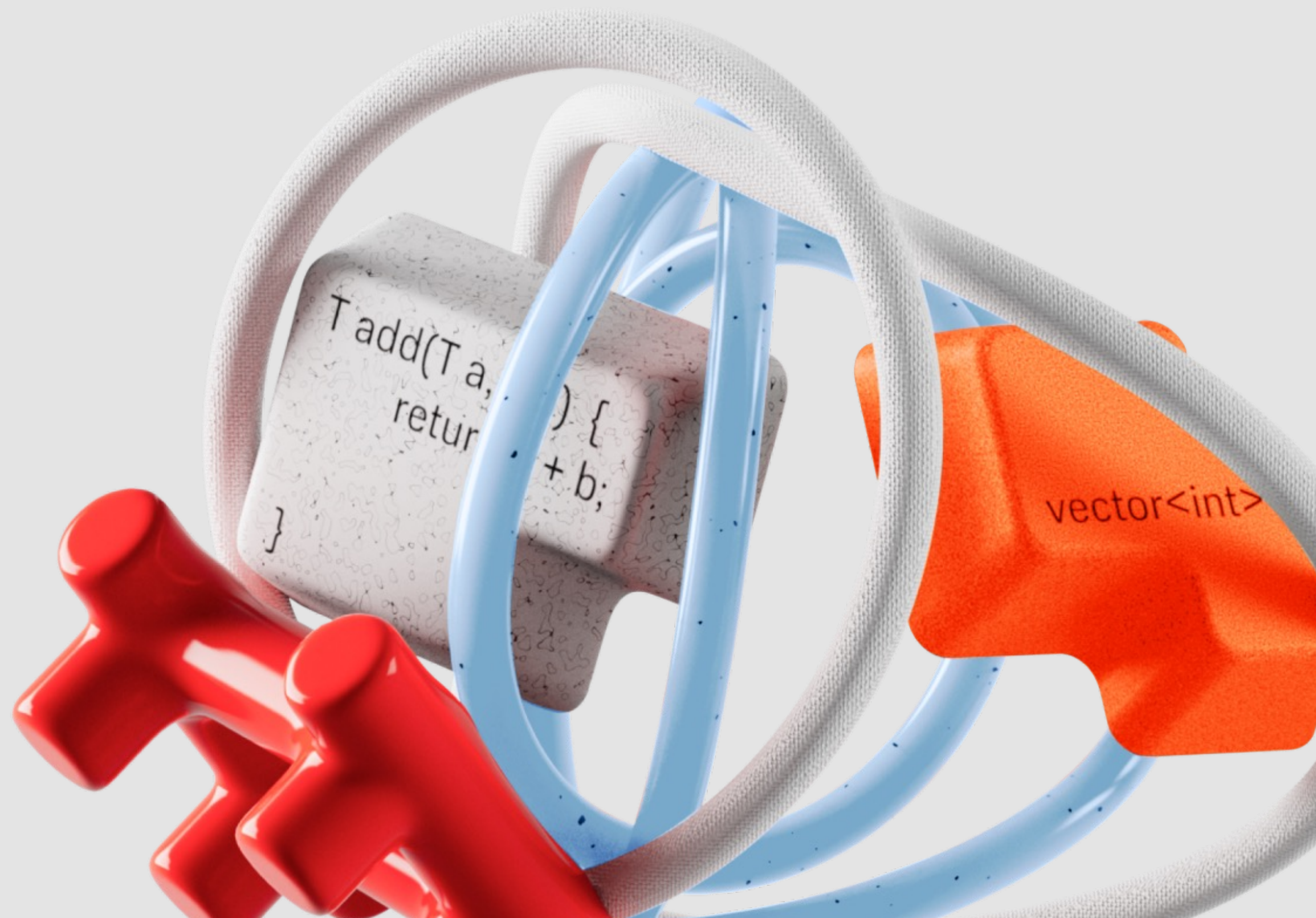


SQL Fiddle  PostgreSQL 9.6 ▾ [View Sample Fiddle](#) [Clear](#) [Text to DDL](#)

```
1 create table users (  
2     user_id int,  
3     name text,  
4     sex text  
5 );  
6  
7 insert into users (user_id,name,sex) values  
8     (111,'Дмитрий','М'),  
9     (122,'Ирина','Ж'),  
10    (133,'Егор','М');  
11  
12 create table abonents (  
13     user_id int,  
14     msisdn bigint  
15 );  
16  
17 insert into abonents (user_id,msisdn) values  
18     (111,9162005990),  
19     (132,9162001187),  
20     (123,9165104810);  
21  
22 create table balance_history (  
23     msisdn bigint,  
24     ts timestamp,
```

[Build Schema](#) [Edit Fullscreen](#) [Browser](#) [\[;\]](#)

ЗАДАЧИ НА СОЕДИНЕНИЯ ТАБЛИЦ, ФИЛЬТРАЦИЯ ДАННЫХ



MTC Teta



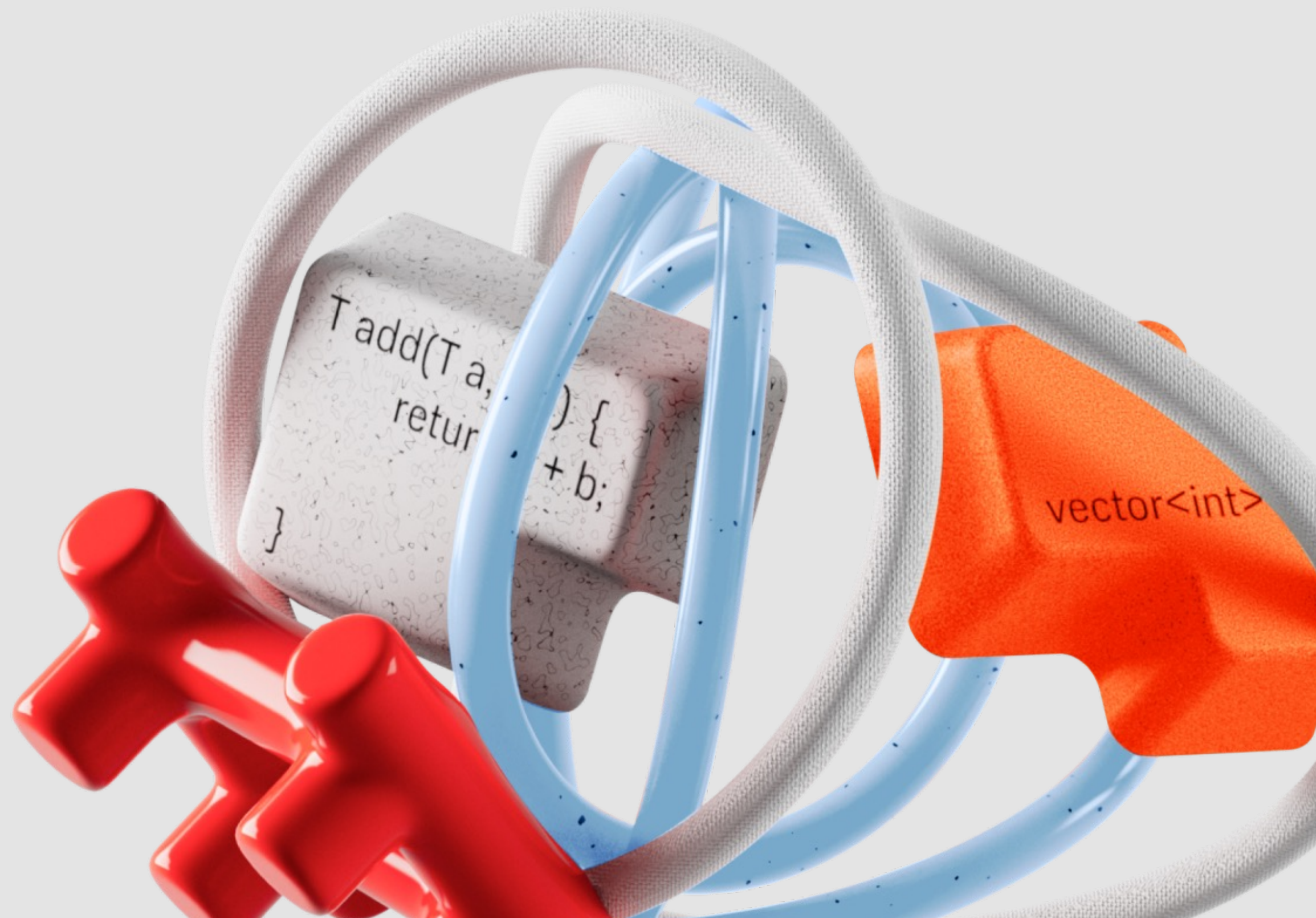
Создание таблиц

Donate About

```
1 select
2   name,
3   ab.msisdn
4 from
5   users as a
6   join abonents as ab
7     on a.user_id = ab.user_id;
8
9 select
10  name,
11  ab.msisdn
12 from
13  users as a
14  left join abonents as ab
15    on a.user_id = ab.user_id;
16
17
```

Run SQL ▶ Edit Fullscreen ↗ [:] ▾

ОБНОВЛЕНИЕ, УДАЛЕНИЕ ЗАПИСЕЙ В ТАБЛИЦЕ (DML)



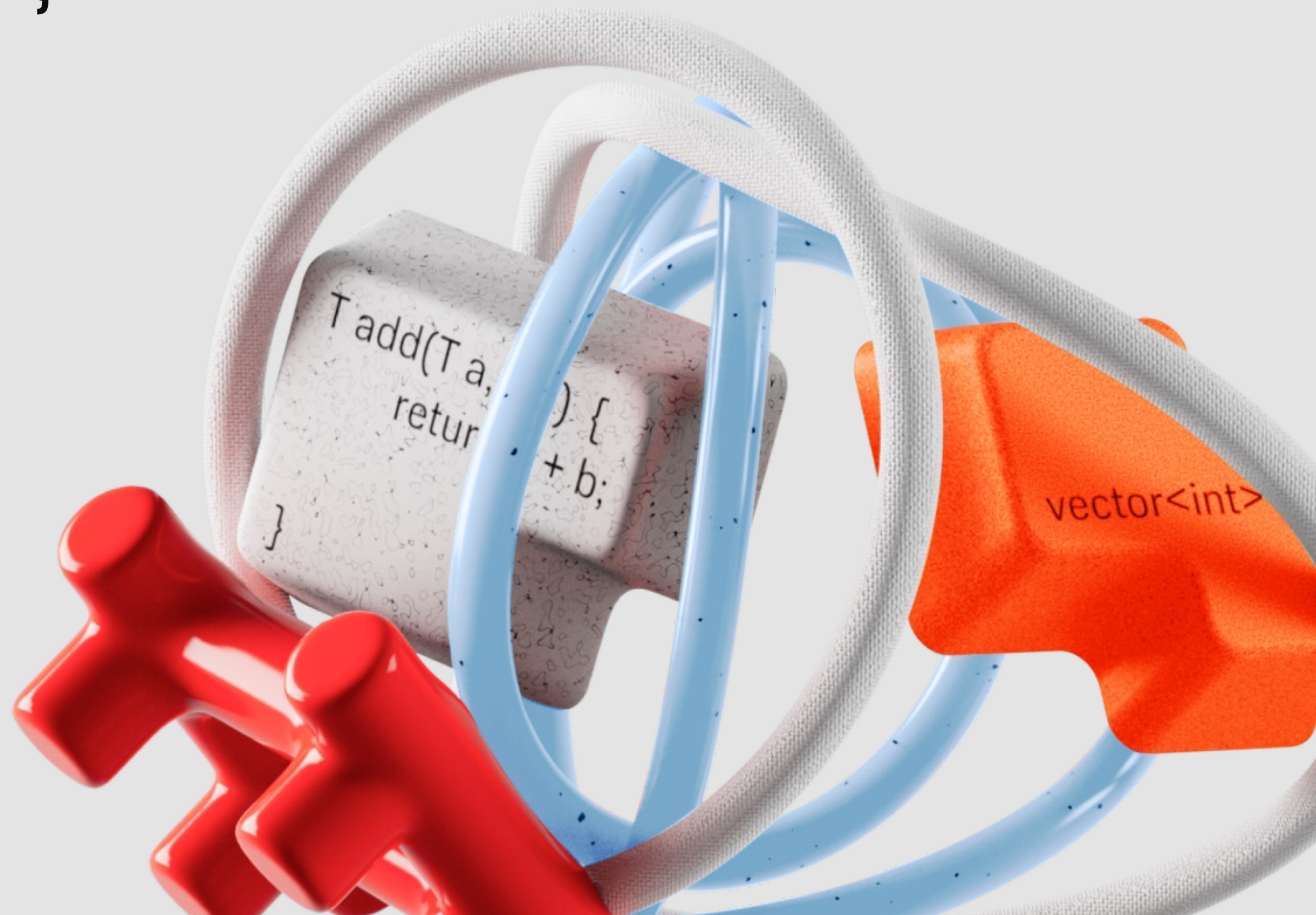
Обновление, удаление записей



MTC Тета



ОБРАБОТКА NULL, ИСПОЛЬЗОВАНИЕ DISTINCT, UNION, VIEW



MTC Тета



NULL, Distinct, UNION

```
1 select
2   name,
3   coalesce(cast(ab.msisdn as text), 'нет номера') as msisdn
4 from
5   users as a
6   left join abonents as ab
7     on a.user_id = ab.user_id;
8
9 select
10  name
11 from
12  users as a
13  left join abonents as ab
14    on a.user_id = ab.user_id
15 where ab.msisdn is null;
16
17 select
18  name
19 from
20  users as a
21  left join abonents as ab
22    on a.user_id = ab.user_id
23     and ab.msisdn is null;
24
```

Run SQL ▶

Edit Fullscreen ↗

[;] ▼

View

- View (вью, вьюха) — это хранимый в базе данных SQL-запрос.
- После создания вьюхи, к ней можно обращаться с помощью SELECT FROM как к обычной таблице.
- Цель создания вью — хранить часто используемый в работе SQL-запрос без необходимости писать его заново. Также вью сокращает количество SQL — кода и делает его понятней

```
1 create or replace view all_users as
2 select
3     *
4 from
5     users
6 union
7     select
8         133, 'Егор', 'М'
9 union all
10    select
11        144, 'Макар', 'М'
12 union all
13    select
14        155, 'Татьяна', 'Ж';
15
16 select * from all_users;
```

Run SQL ▶

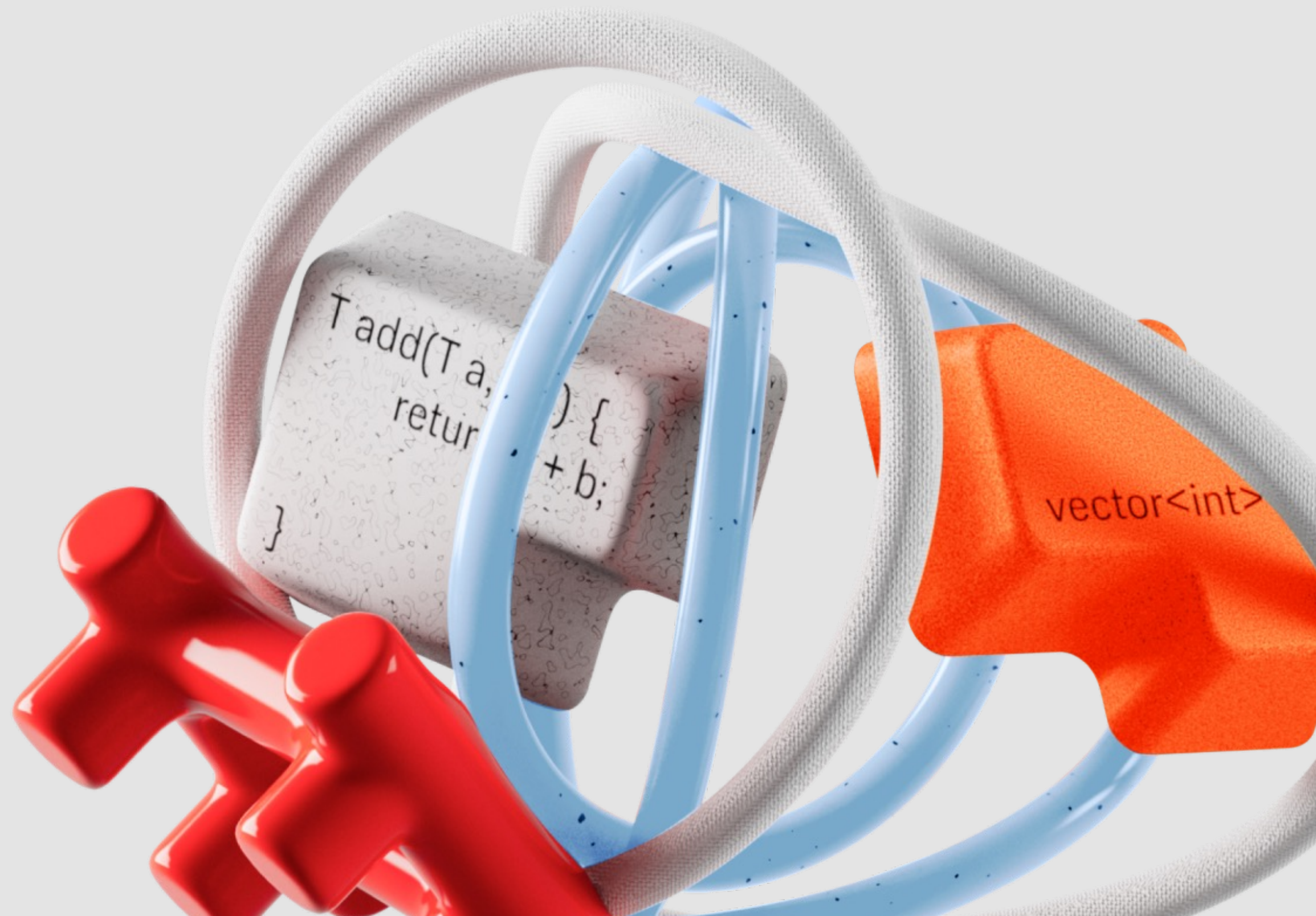
Edit Fullscreen ↗

[;] ▼

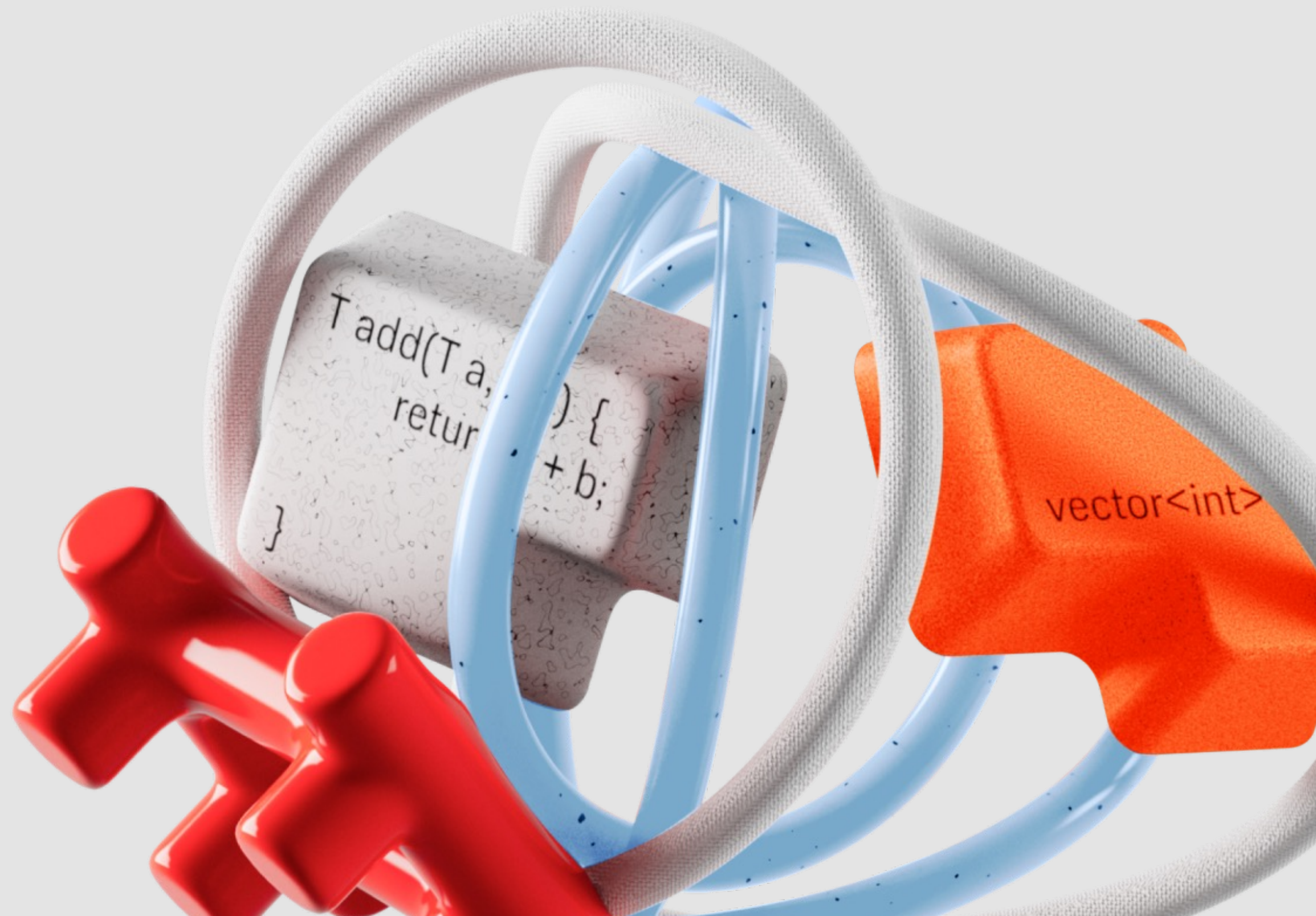
ПРИМЕНЕНИЕ АГРЕГАТОВ ГРУППИРОВКА, СОРТИРОВКА, HAVING



ОБОБЩЁННОЕ ТАБЛИЧНОЕ ВЫРАЖЕНИЕ (COMMON TABLE EXPRESSION) В SQL



ЗАДАЧИ НА ПРИМЕНЕНИЕ ОКОННЫХ ФУНКЦИЙ И СТЕ

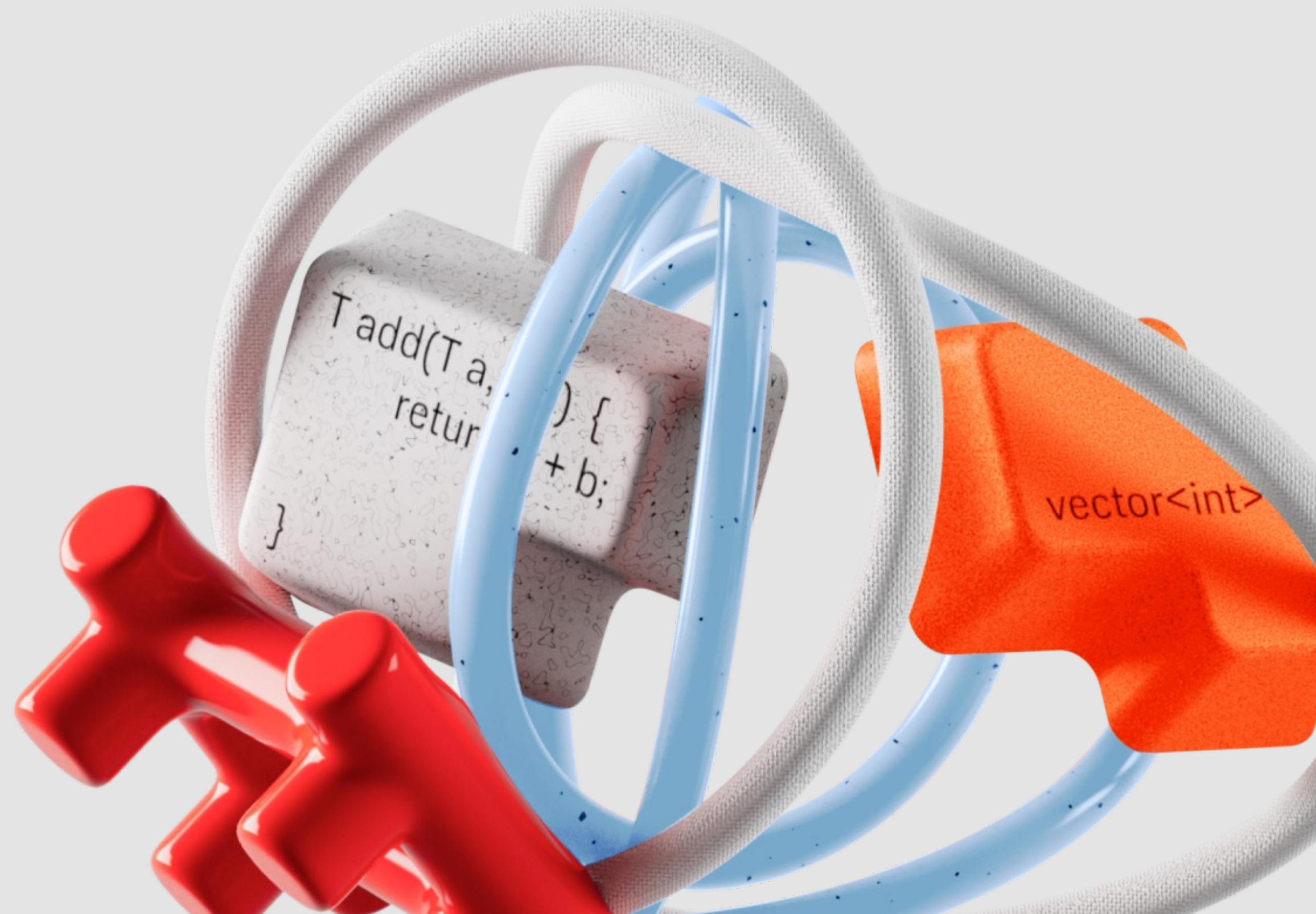


MTC Тета



ПОДЗАПРОСЫ

М Т
С



MTC Тета

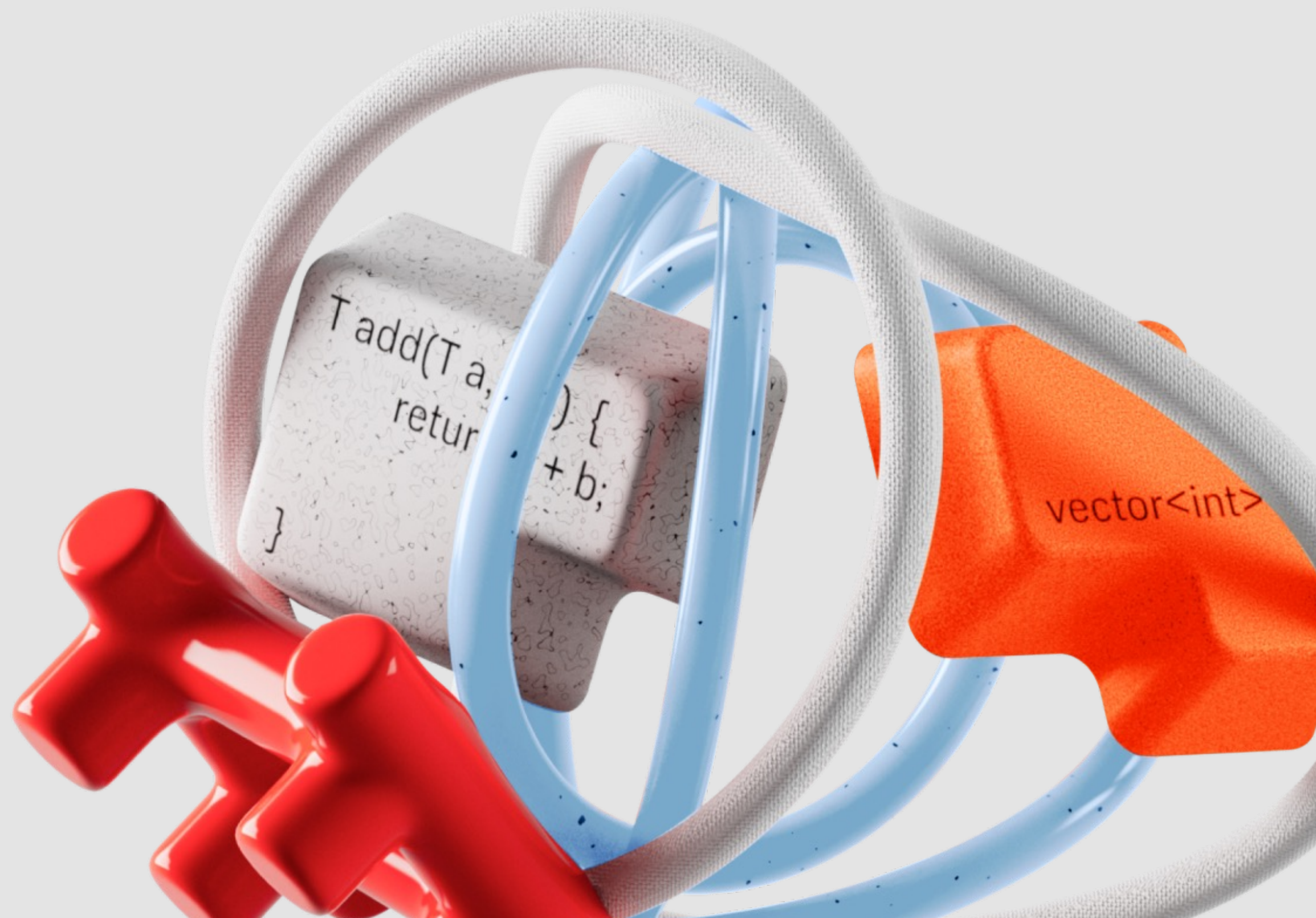


Subqueries

- Когда мы обсуждали синтаксис SQL-запросов, то говорили, что во фразе `SELECT` указываются названия колонок, во фразе `FROM` названия таблиц, а во фразе `WHERE` также участвуют колонки.
- Иногда бывает удобно и это разрешает делать синтаксис SQL, вместо названий колонок и таблиц в `SELECT FROM` и `WHERE` ставить другие SQL-запросы. Такие запросы называют вложенными или подзапросами.
- Очень часто (а может и всегда) SQL-запрос можно переписать так, чтобы в нем не было ни одного подзапроса. Это делается, например, за счет добавления еще одной таблицы в `join`.

НЕЯВНОЕ ОБЪЕДИНЕНИЕ ТАБЛИЦ

М Т
С

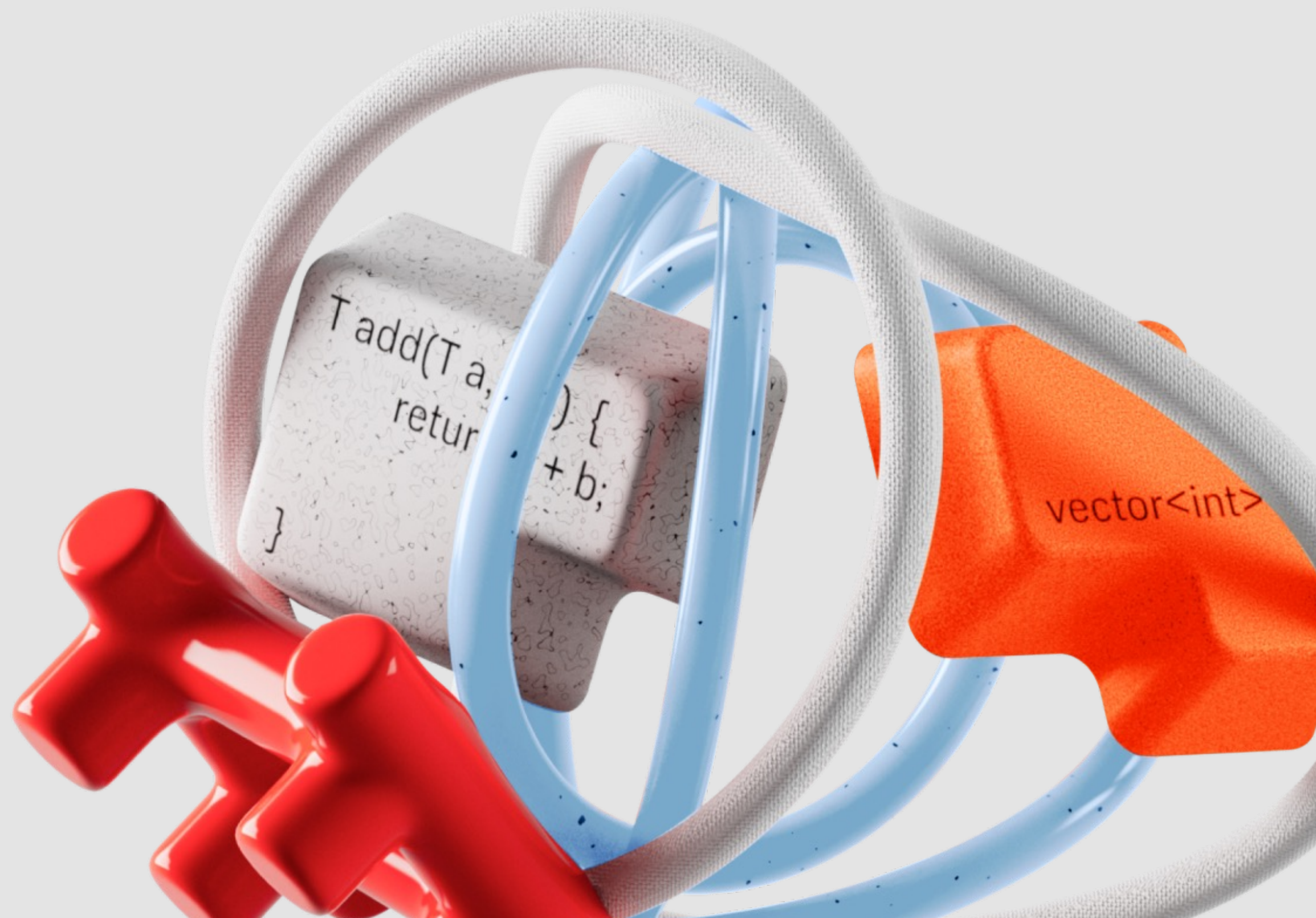


МТС Тета



ПЛАН ЗАПРОСА, ОПТИМИЗАЦИЯ

М Т
С



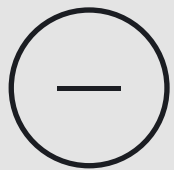
МТС Тета



Вопросы?



Ставим «+»,
если есть вопросы



Ставим «-»,
если нет вопросов

МТС Тета



MTC True Tech

Talents



M T
C

СПАСИБО ЗА ВНИМАНИЕ

Гаврилов Дмитрий

Руководитель Центра компетенций Аналитики

MTC Teta

