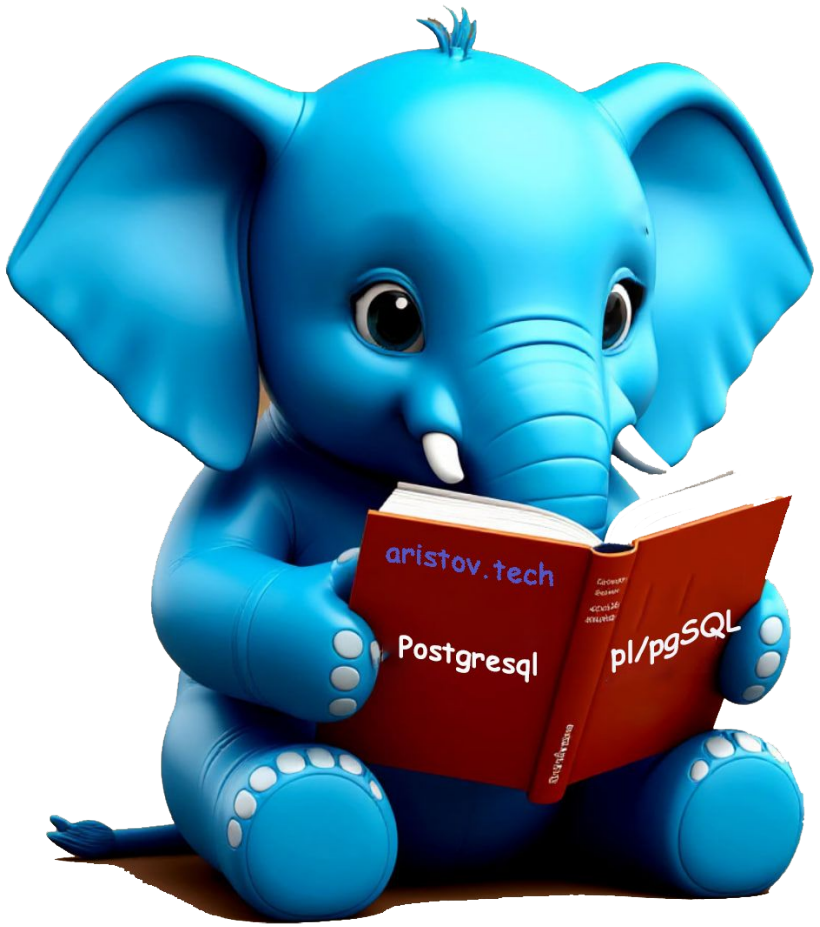
Аристов Евгений

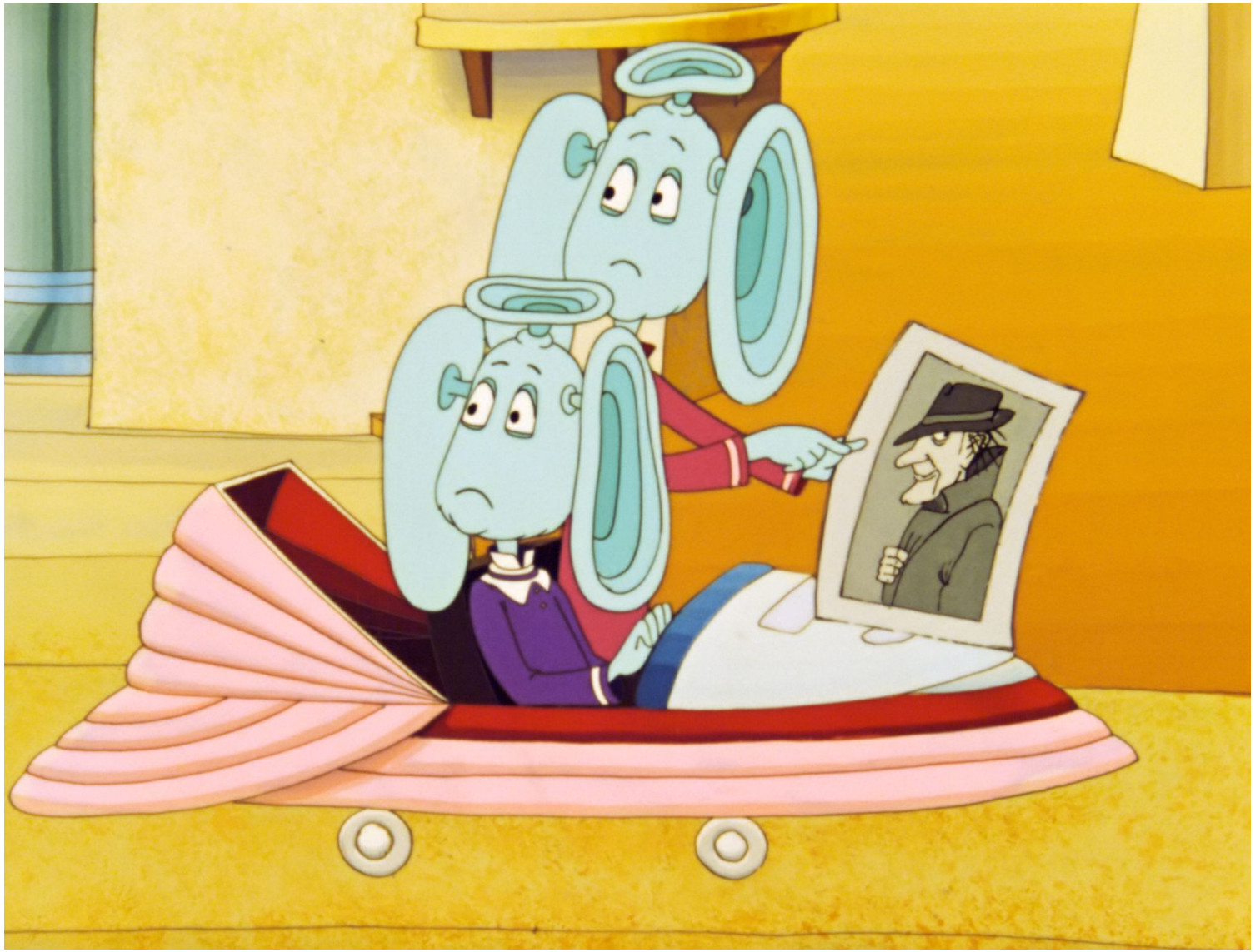
PL/pgSQL в PostgreSQL

за 31 занятие



Использование операторов

https://aristov.tech



Аристов Евгений Николаевич

Founder & CEO aristov.tech
25 лет занимаюсь разработкой БД и ПО

Архитектор высоконагруженных баз данных и инфраструктуры

Спроектировал и разработал более ста проектов для финансового сектора, сетевых магазинов, фитнес-центров, отелей.

Сейчас решаю актуальные для бизнеса задачи: аудит и оптимизация БД и инфраструктуры, миграция на PostgreSQL, обучение сотрудников.

Автор более 10 практических курсов по PostgreSQL, MySQL, Mongo и др..

Автор книг по PostgreSQL. Новинка **PostgreSQL 16: лучшие практики оптимизации**

Правила вебинара

Задаем вопрос в чат

Вопросы вижу, отвечу в момент логической паузы

Если есть вопрос голосом - поставьте знак ? в чат

Если остались вопросы, можно их задать на следующем занятии или в комментариях к записи

https://aristov.tech

Маршрут вебинара

Переопределение оператора - причины, синтаксис, примеры

https://aristov.tech

Переопределение оператора

Оператор

Переопределение операторов в PostgreSQL — это мощный механизм, позволяющий изменять поведение стандартных операторов (+, -, =, >, < и др.) для пользовательских типов данных. Это особенно полезно при работе с сложными структурами данных.

Создадим тип для комплексных чисел:

```
CREATE TYPE complex AS (
    real DOUBLE PRECISION,
    imag DOUBLE PRECISION
);
```

https://aristov.tech

Оператор

Напишем функцию сложения двух комплексных чисел:

```
CREATE FUNCTION complex_add(complex, complex)
RETURNS complex AS $$
BEGIN
    RETURN ROW(
        $1.real + $2.real,
        $1.imag + $2.imag
    )::complex;
END;
$$ LANGUAGE plpgsql IMMUTABLE;
```

https://aristov.tech

Оператор

-- Связываем функцию с оператором +

```
CREATE OPERATOR + (
    LEFTARG = complex,
    RIGHTARG = complex,
    PROCEDURE = complex_add,
    COMMUTATOR = +
);
```

Теперь можно использовать оператор сложения в просто sql:

```
SELECT ROW(3, 4)::complex + ROW(1, 2)::complex;
```

https://aristov.tech

Операторы. Ключевые преимущества

- **Естественный синтаксис** - a + b вместо add(a, b)
- **Интеграция с SQL** - работают в WHERE, ORDER BY, JOIN
- **Поддержка индексов** - через классы операторов
- **Полиморфизм** - один оператор для разных типов (17 тема)
- **Оптимизация** - планировщик учитывает пользовательские операторы

https://aristov.tech

Операторы. Ограничения

- Нельзя переопределить встроенные операторы для базовых типов
- Требует тщательного тестирования
- Может усложнить понимание кода

https://aristov.tech

Практика

Итоги

Итоги

Остались ли вопросы?

Увидимся на следующем занятии

Спасибо за внимание!

Когда дальше и куда?

Аристов Евгений