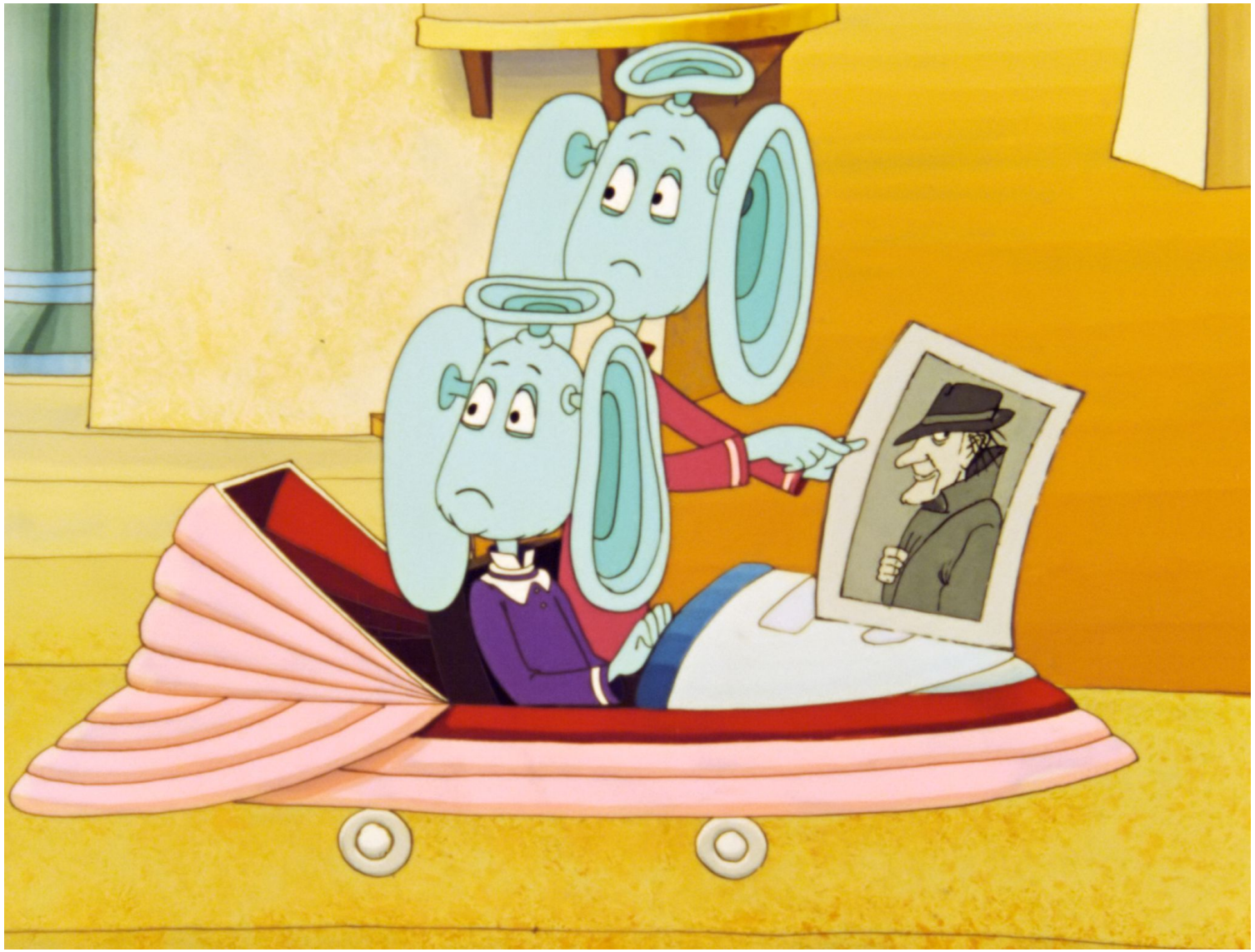
Аристов Евгений

PL/pgSQL в PostgreSQL

за 31 занятие

Составные типы данных и вычисляемые поля

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

Составные типы данных

Вычисляемые поля

Составные типы данных

Составные типы данных

Несмотря на то, что базовых типов 300+, довольно часто необходимо создавать свои типы данных - чем то похоже на объекты и наследование как в ООП.

```
CREATE TYPE currency AS (
    amount numeric,
    code   text
);

CREATE TABLE transactions(
    account_id   integer,
    debit        currency,
    credit       currency,
    date_entered date DEFAULT current_date
);
```

Составные типы данных

Доступ ко вложенным объектам осуществляется как обычно - через точку:

```
CREATE FUNCTION multiply(factor numeric, cur currency) RETURNS currency AS $$
    SELECT ROW(factor * cur.amount, cur.code)::currency;
$$ IMMUTABLE LANGUAGE SQL;
```

https://aristov.tech

Составные типы данных. Ограничения

- добавлять/удалять поля через alter type add/remove field, что может накладывать ограничения на модификацию существующих данных
- Ограниченная поддержка индексов
- Сложности с миграциями
- Не все клиентские библиотеки хорошо поддерживают

Составные типы отлично подходят для сложных структур данных, где нужна строгая типизация и логическая группировка полей!

https://aristov.tech

Виртуальные колонки

Виртуальные колонки

С недавних пор у нас появилась возможность создавать и использовать генерируемые колонки - GENERATED ALWAYS. Кроме дополнительных вычислений минусом является необходимость их хранить рядом с данными.

Если это нам не подходит, то на помощь приходит механизм виртуальных колонок, где на вход мы подаём всю строку и на основании неё генерируем нужное нам значение.

Например Фамилия И.О. на основании ФИО.

Так бы нам пришлось каждый раз повторять формулу - а в данном случае мы просто указываем имя дополнительного поля по имени функции:

```
CREATE FUNCTION textcurrency(cur currency) RETURNS text AS $$
    SELECT cur.amount || cur.code;
$$ IMMUTABLE LANGUAGE SQL;

SELECT t.*, t.textcurrency FROM transactions t;
```

Виртуальные колонки. Минусы

Необходимо вычислять такую колонку каждый раз при обращении, что может быть дорого.

Практика

Итоги

Итоги

Остались ли вопросы?

Увидимся на следующем занятии

Спасибо за внимание!

Когда дальше и куда?

Аристов Евгений