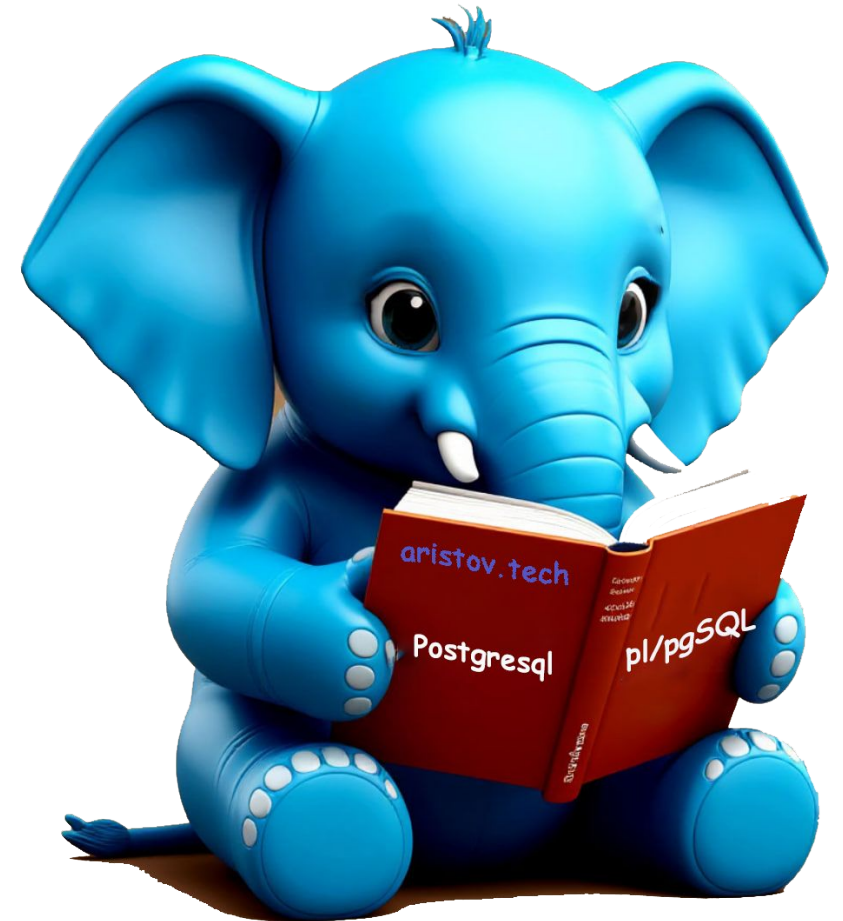


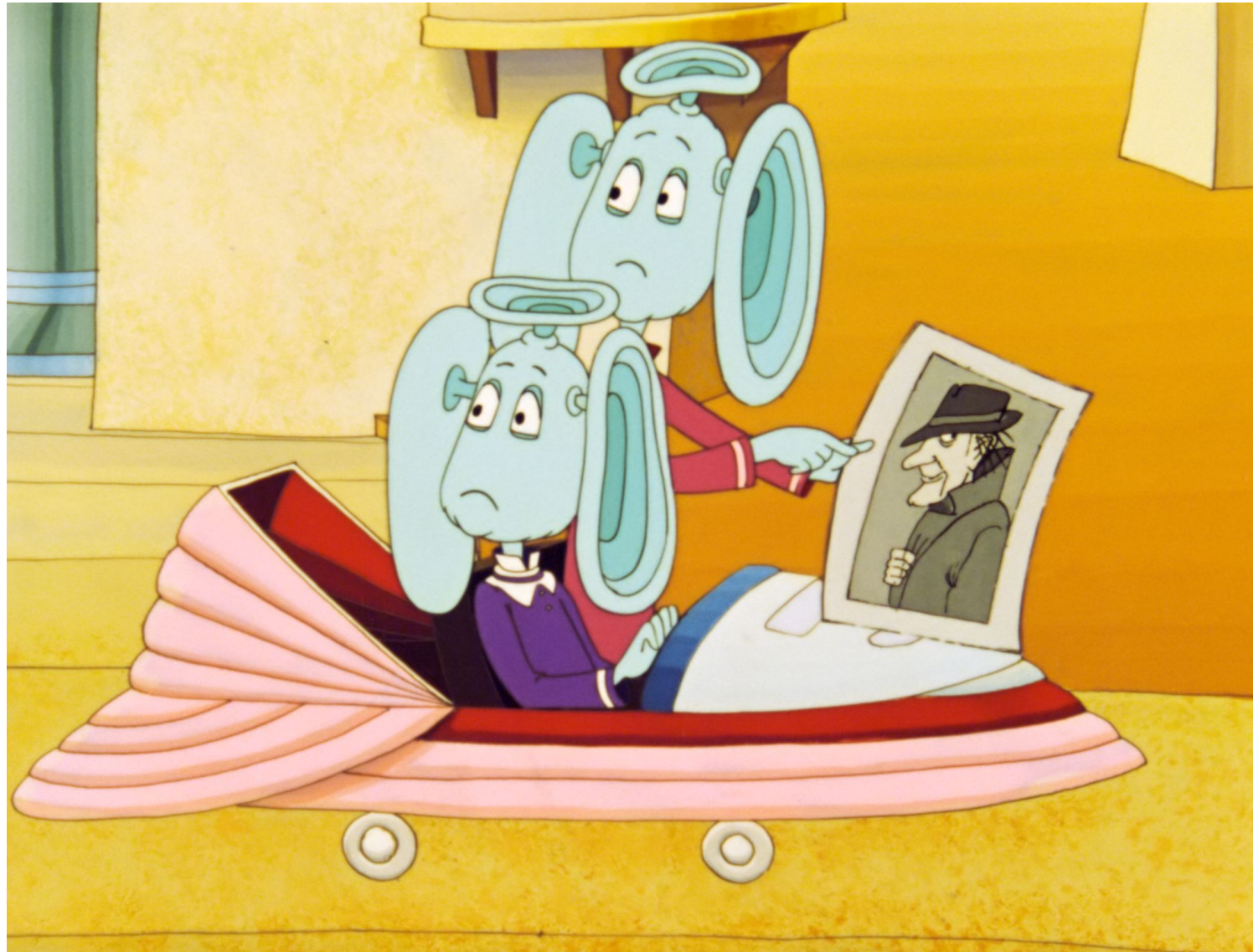
Аристов Евгений

PL/pgSQL в PostgreSQL

за 31 занятие

Понятие блока





**Аристов
Евгений
Николаевич**



<https://aristov.tech>

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](#)

<https://aristov.tech>

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

Маршрут вебинара

Понятие блока

Вложенность блоков

Видимость переменных

Использование специальных меток

Блоки. Понятие и варианты реализации

Понятие блока

PL/pgSQL это блочно-структурированный язык. Текст тела функции должен быть *блоком*.
Структура блока:

```
[ <<метка>> ]  
[ DECLARE  
  объявления ]  
BEGIN  
  операторы  
END [ метка ];
```

Каждое объявление и каждый оператор в блоке должны завершаться символом ";"(точка с запятой). Блок, вложенный в другой блок, должен иметь точку с запятой после END, как показано выше. Однако финальный END, завершающий тело функции, не требует точки с запятой.

Метка

Метка требуется только тогда, когда нужно идентифицировать блок в операторе EXIT, или дополнить имена переменных, объявленных в этом блоке. Если метка указана после END, то она должна совпадать с меткой в начале блока.

Вложенность

Любой оператор в выполняемой секции блока может быть **вложенным блоком**.

Вложенные блоки используются:

- ❖ для логической группировки нескольких операторов
- ❖ локализации области действия переменных для группы операторов
 - автоматическое уничтожение локальных переменных при выходе из блока
 - нельзя извне обратиться к внутренней переменной
- ❖ для возможно ВЫХОДИТЬ из внутреннего блока на ЛЮБОЙ уровень выше
- ❖ *не забывайте про отступы - чтобы было легче читать и понимать, когда закончится вложенный блок*

Во время выполнения вложенного блока переменные, объявленные в нём, **скрывают переменные внешних блоков с такими же именами**. Чтобы получить доступ к внешним переменным, нужно дополнить их имена меткой блока.

Посмотрим пример из документации

```
CREATE FUNCTION somefunc() RETURNS integer AS $$
<< out_block >>
DECLARE
    quantity integer := 300;
BEGIN
    RAISE NOTICE 'Сейчас quantity = %', quantity; -- Выводится 300
    quantity := 500;
    --
    -- Вложенный блок
    --
    DECLARE
        quantity integer := 800;
    BEGIN
        RAISE NOTICE 'Сейчас quantity = %', quantity; -- Выводится 800
        RAISE NOTICE 'Во внешнем блоке quantity = %', out_block.quantity; -- Выводится 500
    END;

    RAISE NOTICE 'Сейчас quantity = %', quantity; -- Выводится 500

    RETURN quantity;
END;
$$ LANGUAGE plpgsql;
```

Практика

Итоги

Итоги

Остались ли вопросы?

Увидимся на следующем занятии

Спасибо за внимание!

Когда дальше и куда?

Аристов Евгений