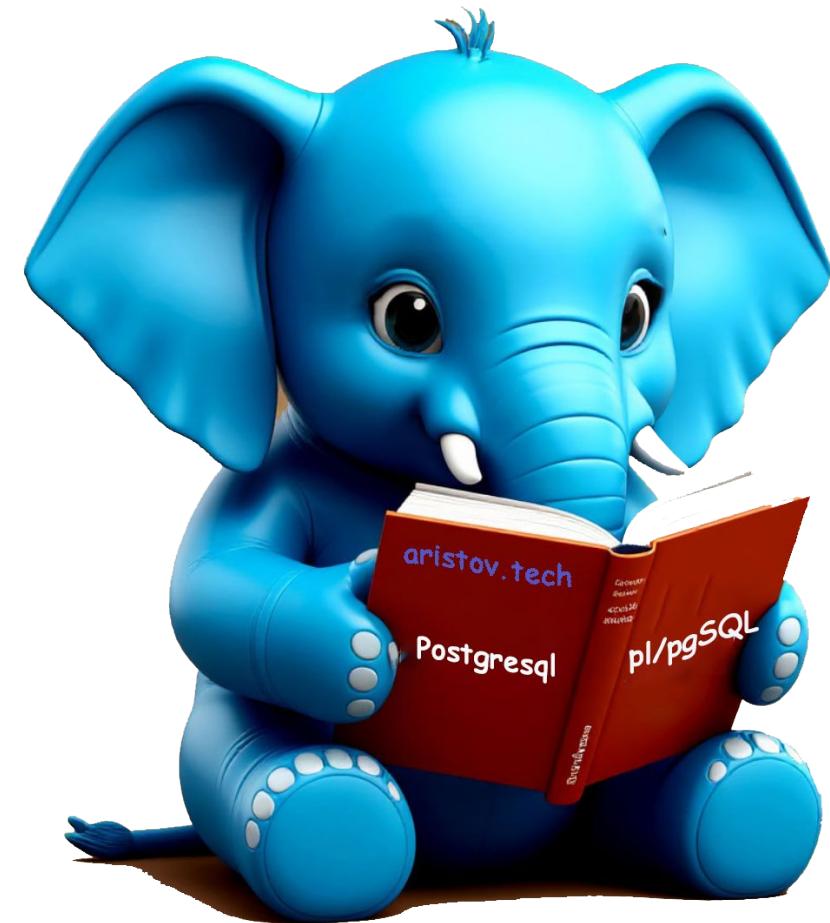


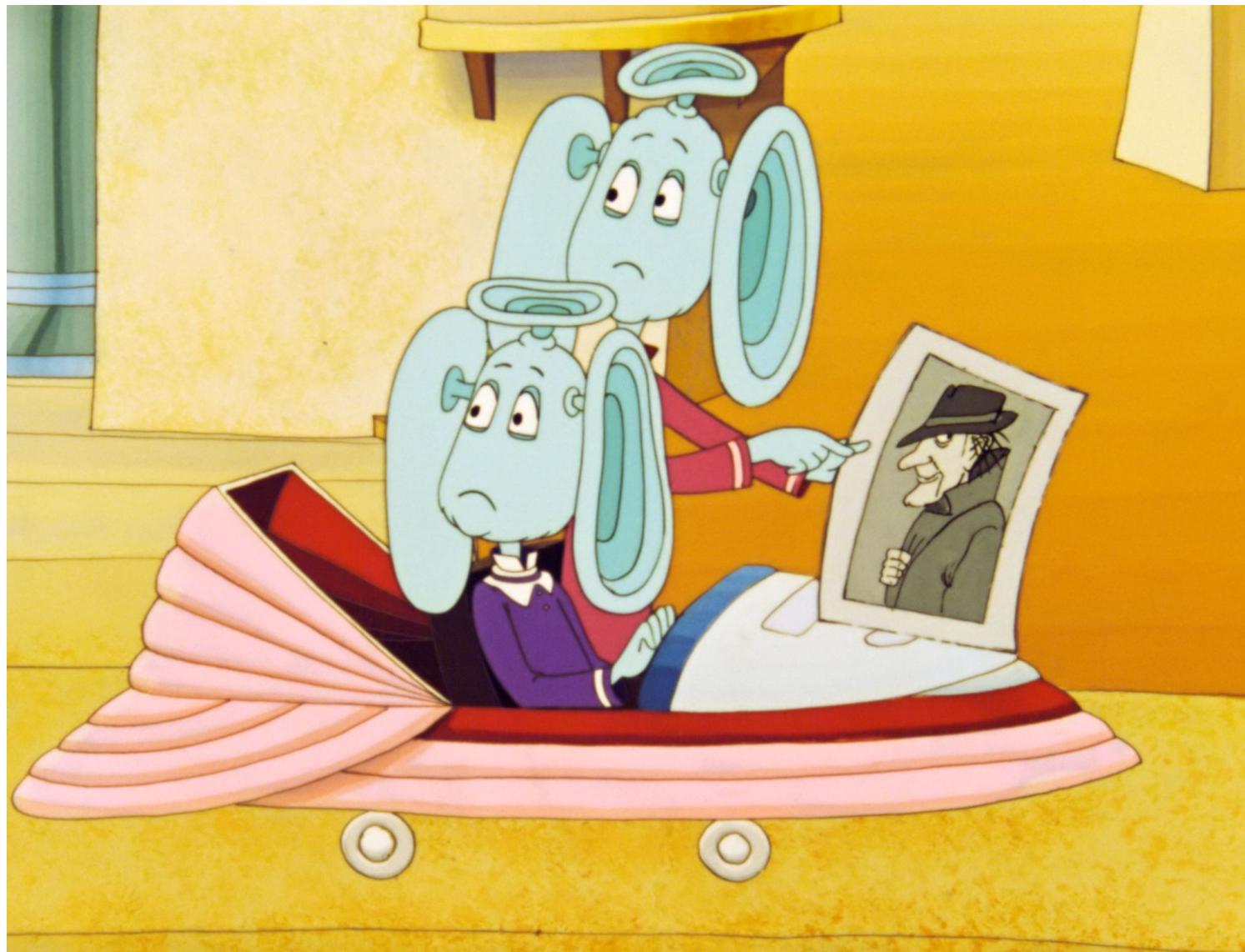
Аристов Евгений

PL/pgSQL в PostgreSQL

за 31 занятие

Использование входных и выходных
параметров, передача массива





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

Автор более 10 практических курсов по PostgreSQL, MySQL, MongoDB 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

Маршрут вебинара

Продвинутый возврат значений из функции

Варианты и их отличия

Особенности применения

Входные выходные параметры

FUNCTION ([режим] параметр 1, [режим] параметр 2 ...)

RETURN тип_результата

Режимы:

- ❖ **IN** (входной) (default)
- ❖ **OUT** (выходной)
- ❖ **INOUT** (входной и выходной)
- ❖ **VARIADIC** (переменный входной)

Особенности:

- ❖ За **единственным** аргументом VARIADIC могут следовать только аргументы OUT.
- ❖ Кроме того, аргументы OUT и INOUT нельзя использовать с предложением **RETURNS [TABLE]**
- ❖ очень удобно, что теперь можно использовать ИМЕНА переменных для возврата

Объявление параметров функции

Когда функция на PL/pgSQL объявляется с **выходными** параметрами, им по аналогии выдаются цифровые идентификаторы \$n и для них можно создавать псевдонимы точно таким же способом, как и для обычных входных параметров.

Выходной параметр это фактически переменная, стартующая с NULL и которой присваивается значение во время выполнения функции. Возвращается последнее присвоенное значение. Например, функция sales может быть переписана так:

```
CREATE FUNCTION sales(total real, OUT tax real) AS $$  
BEGIN  
    tax := total * 0.04;  
END;  
$$ LANGUAGE plpgsql;
```

Возвращение значения

!!!Самое важное!!!

Мы:

- ❖ **или** пишем RETURNS тип/множество/таблицу
- ❖ **или** указываем OUT/INOUT переменные (при этом можно указать RETURNS и тот же тип ОДНОЙ переменной - но это масло масляное)
- ❖ одновременно НЕ работает

Возвращение значения

Внутри PL/pgSQL можно использовать без изменений практически любые команды SQL, не возвращающие результат:

INSERT, UPDATE, DELETE, CREATE, DROP...

Команды INSERT, UPDATE, DELETE могут возвращать результат с помощью фразы **RETURNING**.

Их можно использовать в PL/pgSQL точно так же, как SELECT, добавив фразу INTO.

Итоги

Используйте VARIADIC когда:

- ❖ Количество однотипных аргументов переменное
- ❖ Хотите удобный синтаксис вызова
- ❖ Аргументы логически однородны

Избегайте VARIADIC когда:

- ❖ Нужно несколько списков аргументов
- ❖ Аргументы имеют разные типы
- ❖ Требуется строгая проверка на этапе компиляции

Практика

Итоги

Итоги

Остались ли вопросы?

Увидимся на следующем занятии

Спасибо за внимание!

Когда дальше и куда?

Аристов Евгений