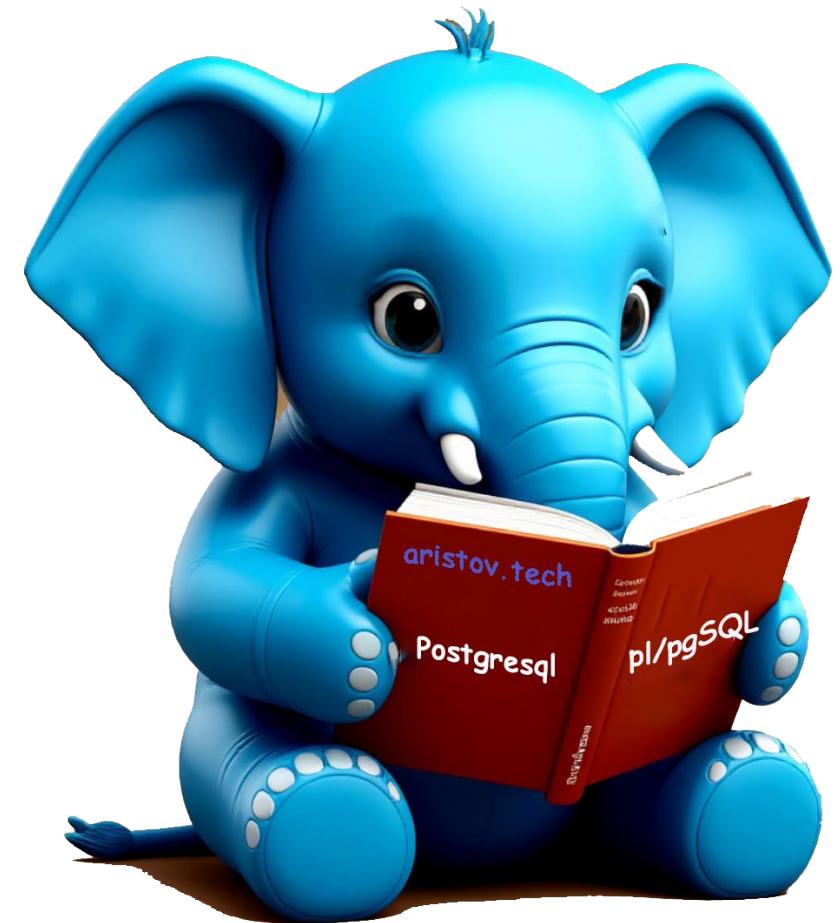


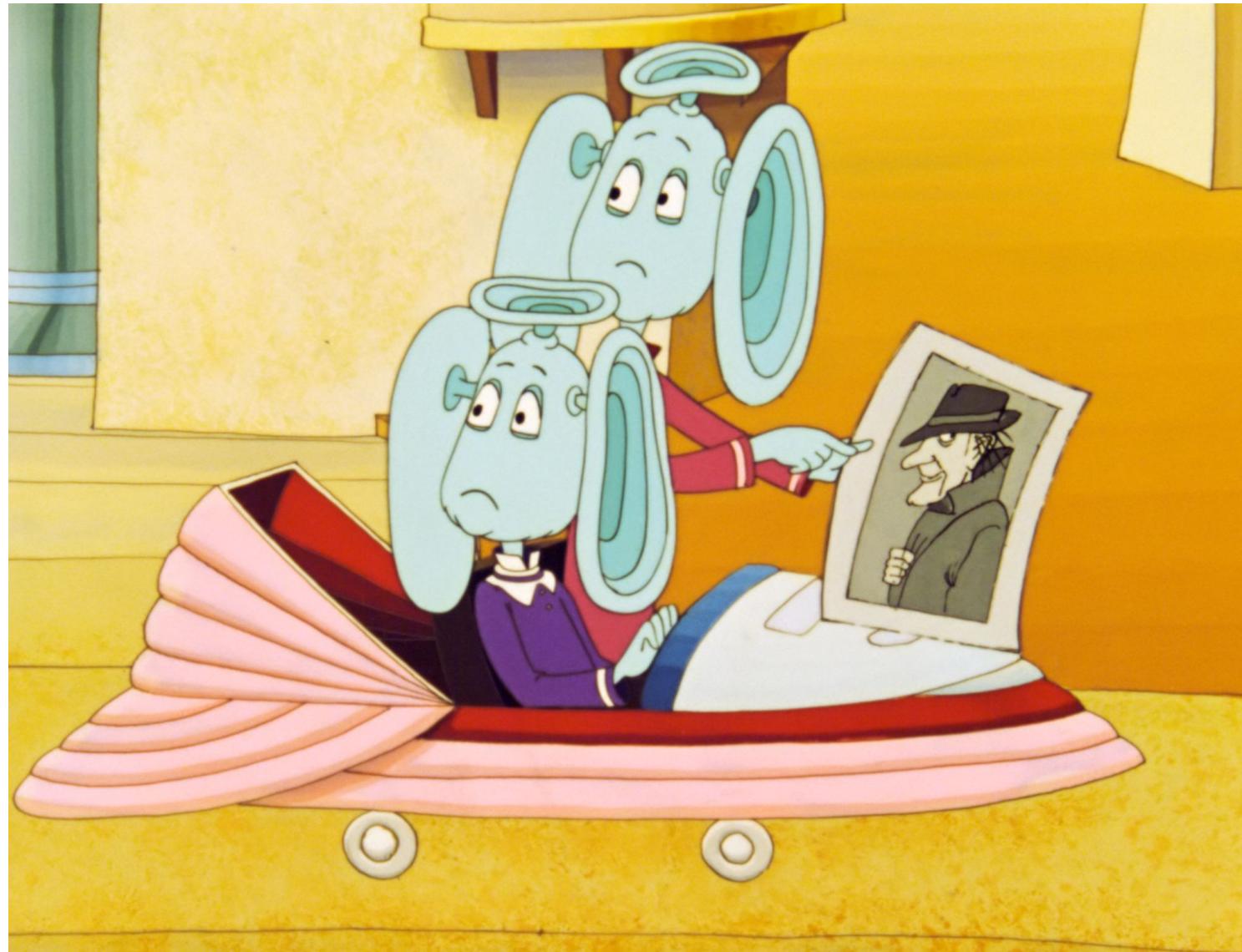
Аристов Евгений

PL/pgSQL в PostgreSQL

за 31 занятие

Перегрузка функций





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, 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

Что такое перегрузка

Зачем она нужна

Ограничения

Best practice

Перегрузка функций

Перегрузка

PostgreSQL допускает **перегрузку** функций, то есть, позволяет использовать **одно имя для нескольких различных функций**, если у них различаются типы входных аргументов.

Независимо от того, используете вы эту возможность или нет, она требует предосторожности при вызове функций в базах данных, где одни пользователи не доверяют другим <https://www.postgresql.org/docs/current/typeconv-func.html>

CREATE FUNCTION func(int) ...

CREATE FUNCTION func(int, int) ...

CREATE FUNCTION func(int, real) ...

CREATE FUNCTION func(real) ...

Перегрузка. Особенности реализации

Две функции считаются совпадающими, если они имеют одинаковые имена и типы входных аргументов, параметры OUT игнорируются. Таким образом, например, эти объявления вызовут **конфликт**:

```
CREATE FUNCTION func(int) ...
CREATE FUNCTION func(int, out text) ...
```

```
select func(1) ???
```

Отсутствие перегрузки по возвращаемому типу с одинаковыми входными параметрами.

Перегрузка. Ограничения

Функции, имеющие разные типы аргументов, не будут считаться конфликтующими в момент создания, но предоставленные для них значения по умолчанию могут вызвать конфликт в момент использования. Например, рассмотрите следующие определения:

CREATE FUNCTION func(int) ...

CREATE FUNCTION func(int, int default 2) ...

Вызов `foo(100)` завершится **ошибкой** из-за неоднозначности в выборе вызываемой функции

Перегрузка. Сложности выбора

Неоднозначность при вызове

```
CREATE FUNCTION process_value(val INTEGER) RETURNS TEXT AS $$ ... $$;
```

```
CREATE FUNCTION process_value(val NUMERIC) RETURNS TEXT AS $$ ... $$;
```

Какую функцию выберет PostgreSQL?

```
SELECT process_value(10); -- INTEGER или NUMERIC?
```

С одинаковым именем должны реализовывать аналогичную бизнес логику!

Перегрузка. Сложности отладки

При ошибке сложно определить, какая именно функция вызвала проблему

ERROR: function process_data(unknown, unknown) does not exist

LINE 1: SELECT process_data('test', 'invalid');

 ^

HINT: No function matches the given name and argument types.

Перегрузка. Итоги

Используйте перегрузку когда:

- ❖ Функции выполняют схожую логику с разными типами данных
- ❖ Хотите создать удобный и интуитивный API
- ❖ Нужна обратная совместимость

Избегайте перегрузки когда:

- ❖ Функции выполняют принципиально разную логику
- ❖ Возможна неоднозначность при вызове
- ❖ Работаете с простыми сценариями

Практика

Итоги

Итоги

Остались ли вопросы?

Увидимся на следующем занятии

Спасибо за внимание!

Когда дальше и куда?

Аристов Евгений