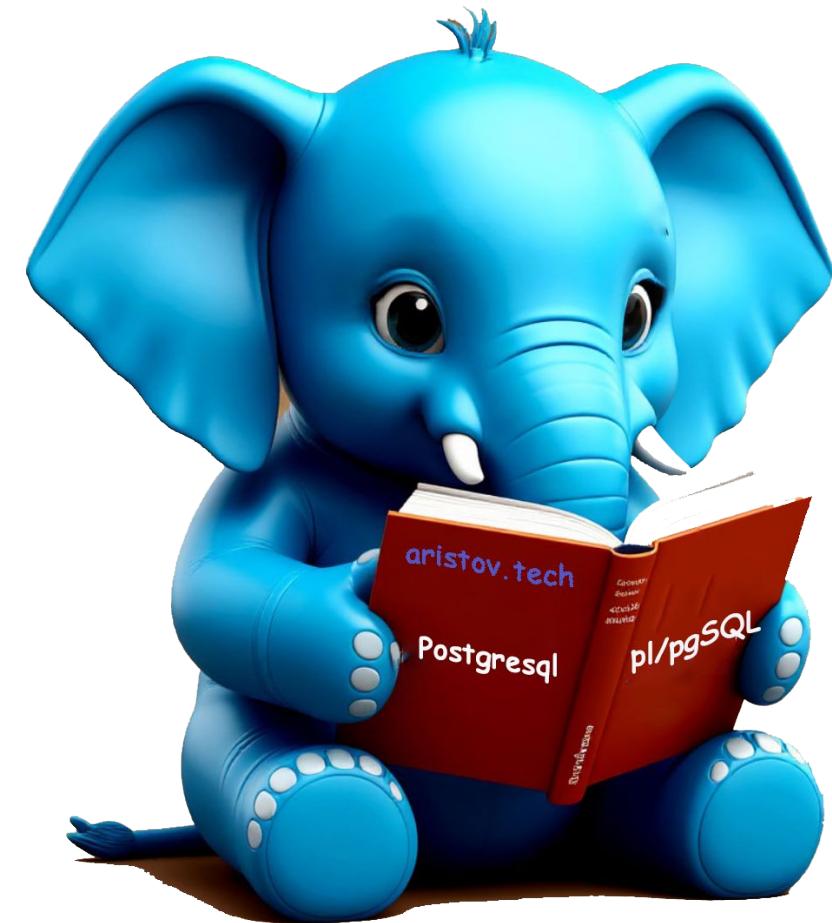


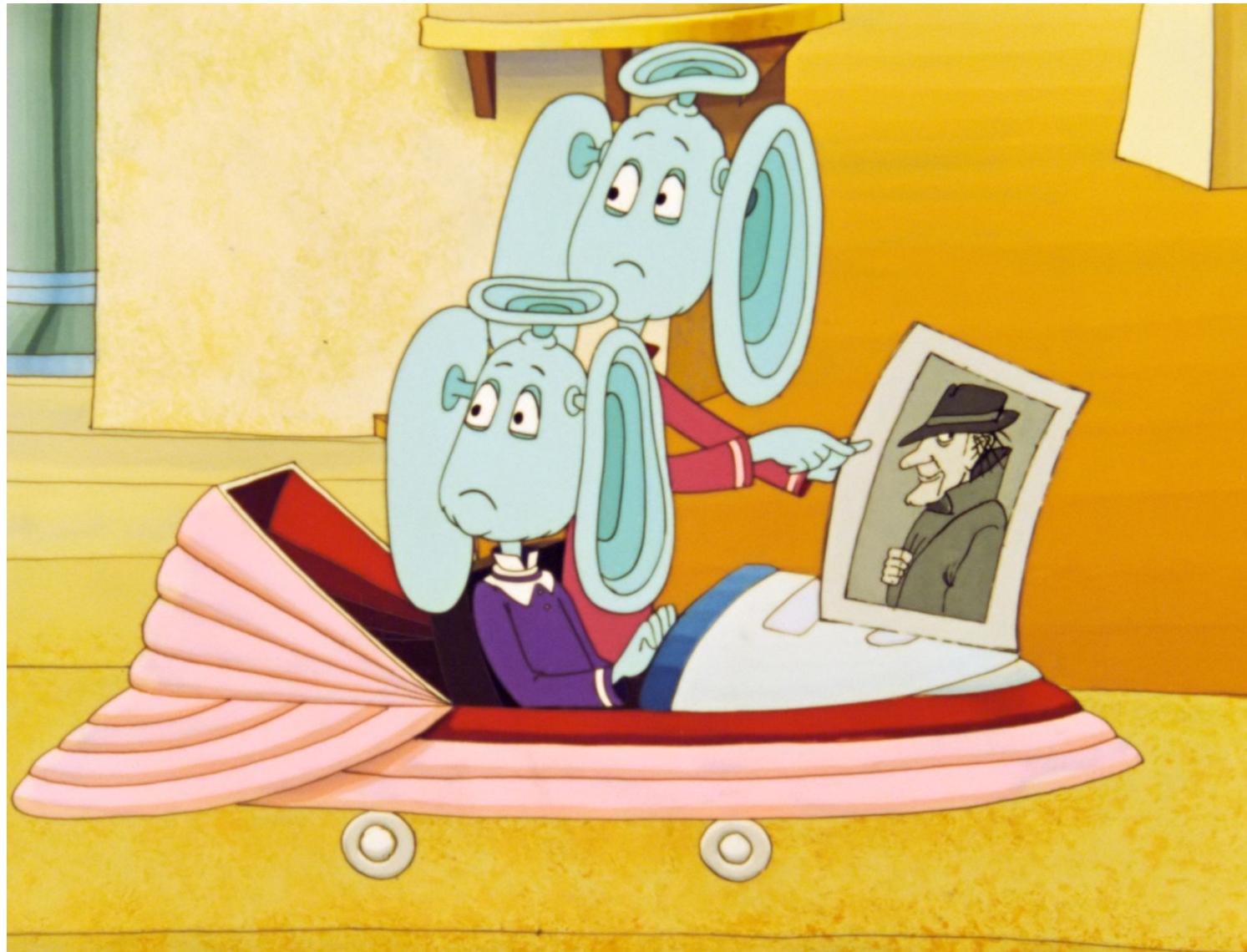
Аристов Евгений

# PL/pgSQL в PostgreSQL

за 31 занятие

Категории изменчивости функций





Аристов  
Евгений  
Николаевич



<https://aristov.tech>

Founder & CEO [aristov.tech](https://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

Варианты кеширования значений функций

Подводные камни

# Категории изменчивости функций

# Категории изменчивости функций

Параметры функции:

- ❖ **IMMUTABLE** - кеширование
- ❖ **STABLE**
- ❖ **VOLATILE (default)**

**IMMUTABLE** показывает, что функция не может модифицировать базу данных и всегда возвращает один и тот же результат при определённых значениях аргументов.

Она не обращается к базе данных и не использует информацию, не переданную ей явно в списке аргументов. Если функция имеет такую характеристику, любой её вызов с аргументами-константами можно немедленно заменить значением функции.

## Категории изменчивости функций

**STABLE** показывает, что функция не может модифицировать базу данных и в рамках одного сканирования таблицы она всегда возвращает один и тот же результат для определённых значений аргументов, но этот результат может быть разным в разных операторах SQL.

Хороший выбор для функций, результаты которых зависят от содержимого базы данных и настраиваемых параметров (например, текущего часового пояса). (Но этот вариант не подходит для триггеров AFTER, желающих прочитать строки, изменённые текущей командой.)

## Категории изменчивости функций

**VOLATILE** показывает, что результат функции может меняться даже в рамках одного сканирования таблицы, так что её вызовы нельзя оптимизировать.

Изменчивы в этом смысле относительно немногие функции баз данных, например: `random()`, `currval()` и `timeofday()`.

Важно, что любая функция с побочными эффектами должна быть классифицирована как изменчивая, даже если её результат вполне предсказуем, чтобы её вызовы не были соптимизированы; пример такой функции: `setval()`.

## Нюансы

**STABLE** и **IMMUTABLE** используют снимок, полученный в начале вызывающего запроса, тогда как функции **VOLATILE** получают свежий снимок в начале каждого запроса, который они выполняют из за особенностей реализации системы MVCC (<https://aristov.tech/blog/mvcc-in-postgresql/> ).

Одна из распространённых ошибок – помечать функцию как **IMMUTABLE**, при том, что её результаты зависят от параметра конфигурации. Например, функция, работающая с временем, может выдавать результаты, зависящие от параметра зоны времени. Для надёжности такие функции следует помечать как **STABLE**.

Давайте разбираться на практике

# Практика

# Итоги

## Итоги

Остались ли вопросы?

Увидимся на следующем занятии

# Спасибо за внимание!

Когда дальше и куда?

Аристов Евгений