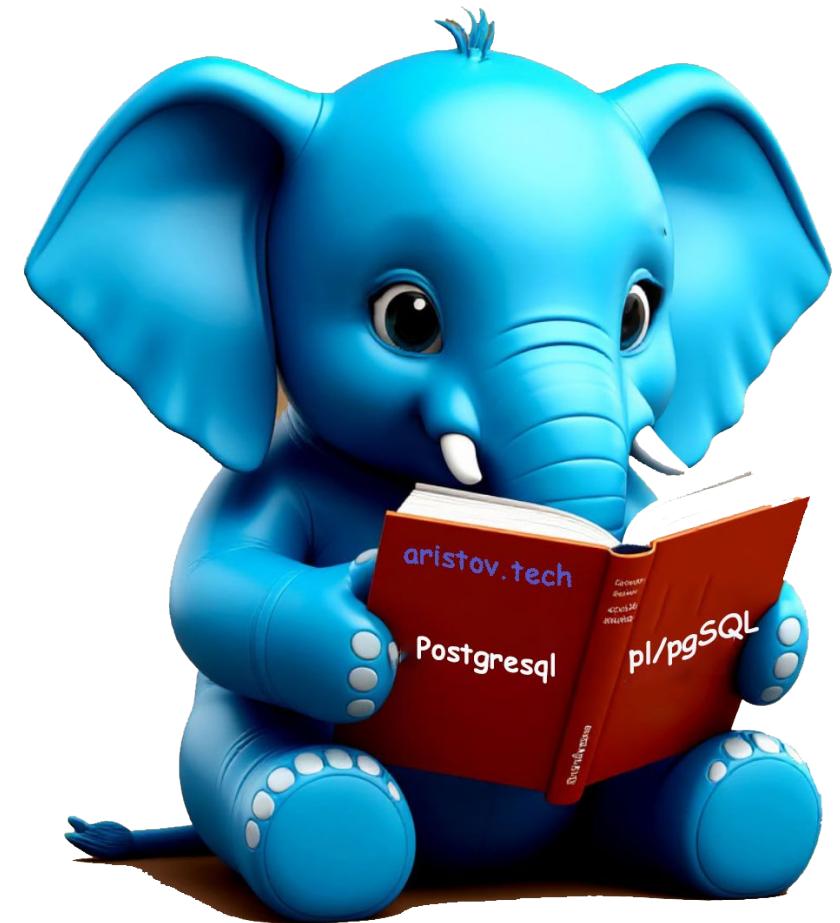


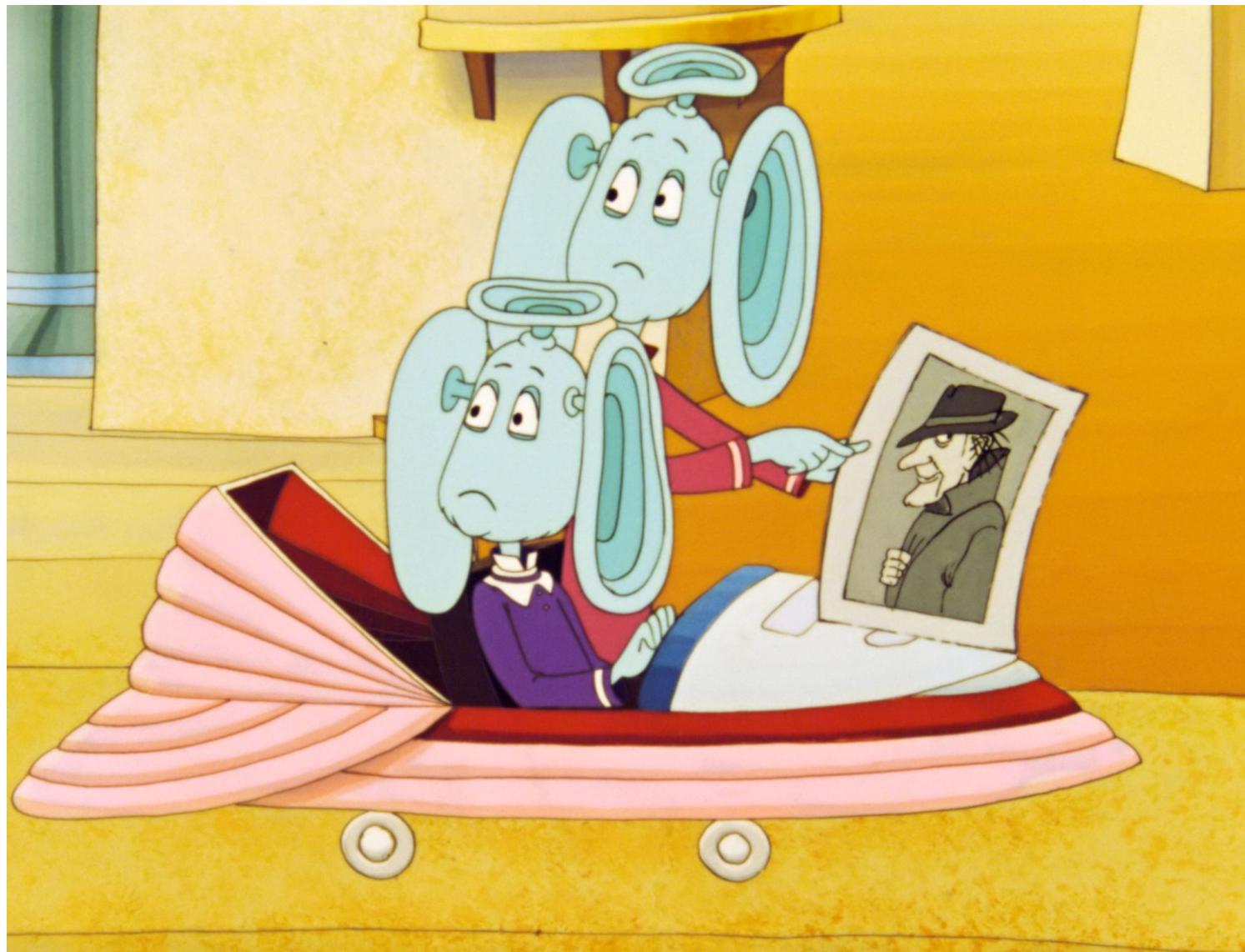
Аристов Евгений

# PL/pgSQL в PostgreSQL

за 31 занятие

Транзакции в серверном  
программировании





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

Причины использования транзакций в процедурах

Особенности реализации

Savepoint

Ограничения

# Транзакции

# Транзакции

Управлять транзакциями (`begin`, `commit`, `rollback`, `savepoint`) можно только в процедурах (в т.ч. анонимных).

Процедура как и функция выполняется в атомарной транзакции, но в процедуре возможно завершить текущую транзакцию (`commit/rollback`) и автоматически начнётся новая.

Также можно сохранить текущие изменения (`savepoint`), чтобы при определенных обстоятельствах можно было вернуться на эту сохраненную точку. Также можно обрабатывать ошибки и строить кастомную логику, основанную на цепочке транзакций.

Вложенных транзакций в PostgreSQL, к сожалению, нет.

# Транзакции. Итоги

## Ключевые моменты управления транзакциями:

- ❖ Используйте **SAVEPOINT** для гранулярного контроля
- ❖ Обрабатывайте ошибки на разных уровнях
- ❖ Минимизируйте время блокировок
- ❖ Документируйте поведение транзакций в процедурах

## Процедуры с транзакциями идеальны для:

- ❖ Сложных бизнес-процессов
- ❖ Операций, требующих атомарности
- ❖ Систем с высокими требованиями к целостности данных
- ❖ Распределенных операций

# Практика

# Итоги

## Итоги

Остались ли вопросы?

Увидимся на следующем занятии

# Спасибо за внимание!

Когда дальше и куда?

Аристов Евгений