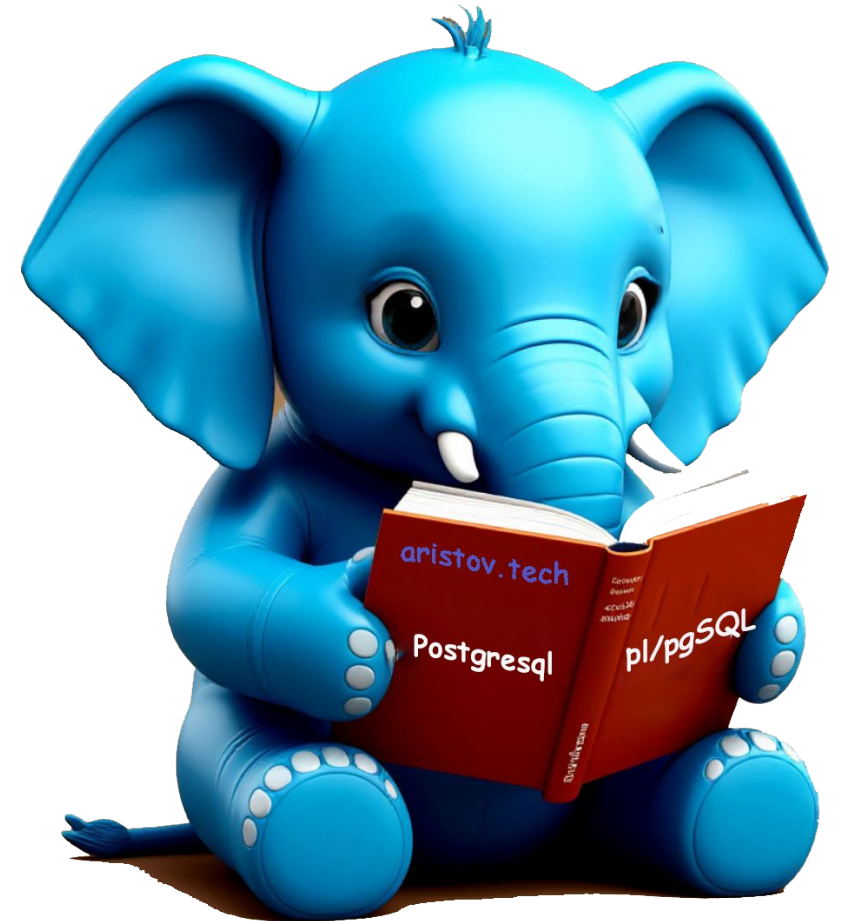


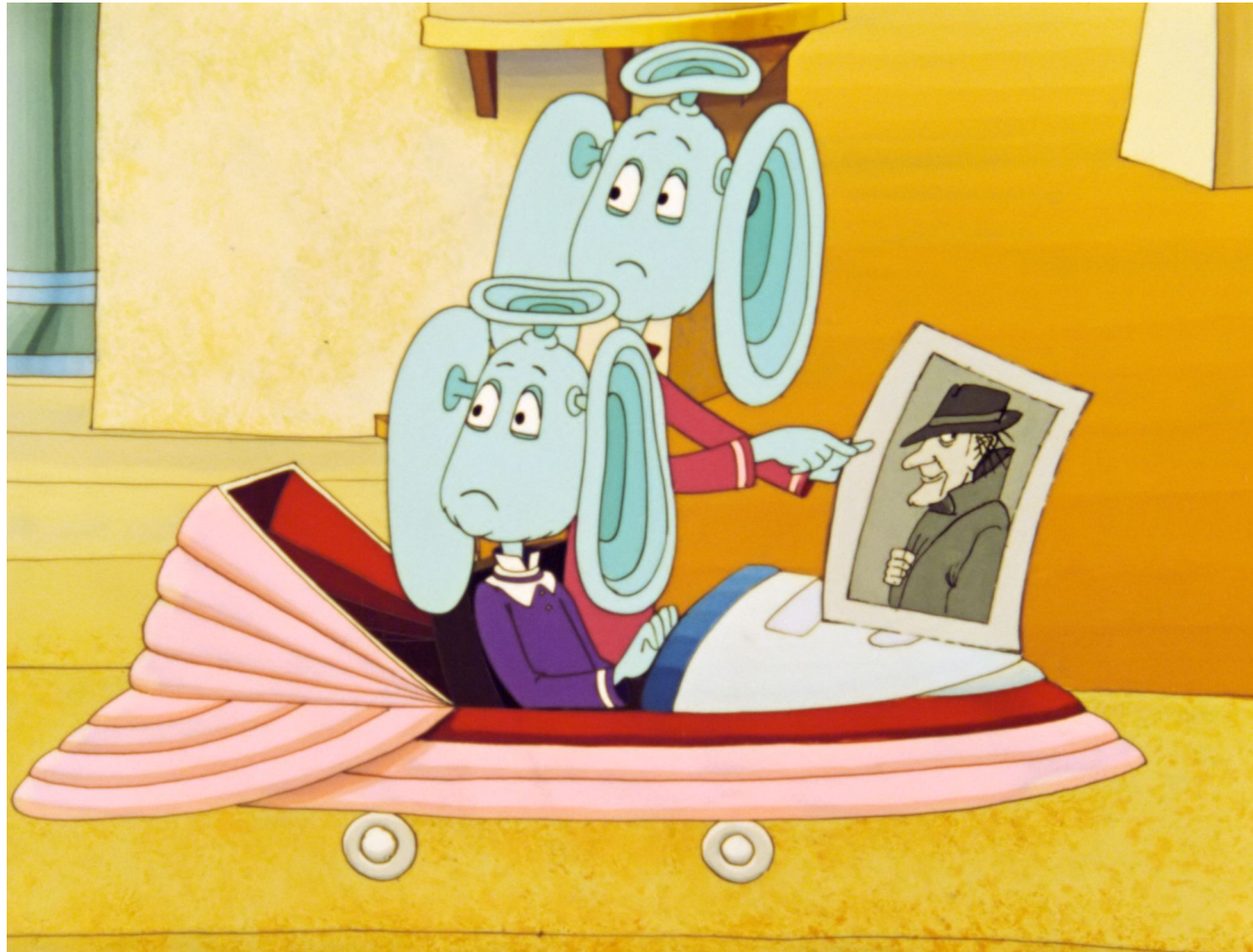
**Аристов Евгений**

# **PL/pgSQL в PostgreSQL**

**за 31 занятие**

**DDL триггеры**





**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 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

DDL триггеры и отличие от DML

Кейсы использования

# DDL триггеры

# DDL триггеры - событийные

DDL триггеры в PostgreSQL (триггеры на операции CREATE, ALTER, DROP) позволяют перехватывать и контролировать **изменения схемы базы данных**. Они работают на уровне событий, а не строк.



# DDL триггеры - событийные

<https://www.postgresql.org/docs/current/event-triggers.html>

как это выглядело до 17 версии:

Command Tag	ddl_command_start	ddl_command_end	sql_drop	table_rewrite	Notes
ALTER AGGREGATE	X	X	-	-	
ALTER COLLATION	X	X	-	-	
ALTER CONVERSION	X	X	-	-	
ALTER DOMAIN	X	X	-	-	
ALTER DEFAULT PRIVILEGES	X	X	-	-	
ALTER EXTENSION	X	X	-	-	
ALTER FOREIGN DATA WRAPPER	X	X	-	-	
ALTER FOREIGN TABLE	X	X	X	-	



# Особенности реализации

Теперь есть список команд для перехвата:

[38.1.1. login](#)

[38.1.2. ddl\\_command\\_start](#)

[38.1.3. ddl\\_command\\_end](#)

[38.1.4. sql\\_drop](#)

[38.1.5. table\\_rewrite](#)

# И на каждую команду список действий.

## Пример `ddl_command_start`:

<https://aristov.tech>

- ❖ CREATE
- ❖ ALTER
- ❖ DROP
- ❖ COMMENT
- ❖ GRANT
- ❖ IMPORT FOREIGN SCHEMA
- ❖ REINDEX
- ❖ REFRESH MATERIALIZED VIEW
- ❖ REVOKE
- ❖ SECURITY LABEL

До 17 версии была матрица - <https://www.postgresql.org/docs/17/event-trigger-matrix.html>

<https://aristov.tech>

# Ограничения

- ❖ Не могут отменить DDL операцию (только бросить исключение)
- ❖ Не работают для некоторых системных операций
- ❖ Не все команды отрабатывают как в доке (drop, ряд других DDL команд) Могут влиять на миграции базы данных
- ❖ Не можете изменять ещё не существующий объект (comment и т.д.)

# Задачи

DDL триггеры идеально подходят для:

- ❖ Аудита изменений схемы базы данных
- ❖ Защиты критических объектов от удаления
- ❖ Автоматизации назначения прав
- ❖ Валидации соглашений об именовании
- ❖ Синхронизации с внешними системами
- ❖ При удалении старой секции автоматизированное создание новой

# DDL самописный триггер

Пример на языке C:

<https://www.postgresql.org/docs/current/event-trigger-example.html>

# Практика

# Итоги



# Итоги

Остались ли вопросы?

Увидимся на следующем занятии

# Спасибо за внимание!

Когда дальше и куда?

Аристов Евгений