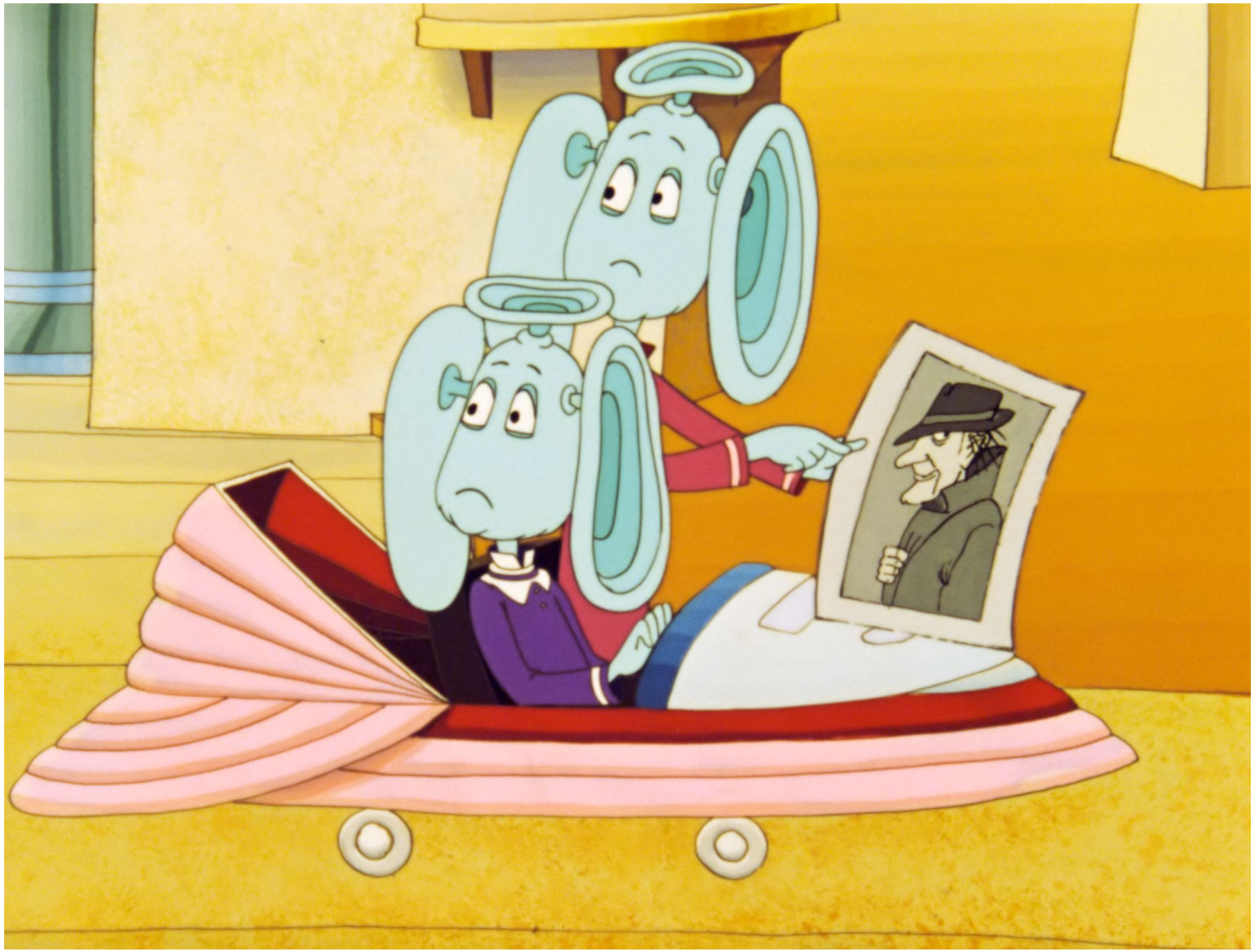
Аристов Евгений

PL/pgSQL в PostgreSQL

за 31 занятие

Серверное программирование в PostgreSQL.
Назначение и основные особенности

https://aristov.tech



Аристов Евгений Николаевич

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 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**

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

https://aristov.tech

Маршрут вебинара

Структура курса

Серверное программирование в PostgreSQL

Назначение и основные особенности

Стенд

https://aristov.tech

Структура курса

https://aristov.tech

Структура курса

Цель курса: помочь участникам понять различия между стандартным SQL и процедурным языком PL/pgSQL, а также научиться эффективно использовать оба инструмента для решения задач в PostgreSQL. Курс направлен на развитие навыков работы с базами данных, включая написание запросов, создание хранимых процедур, функций и триггеров, а также оптимизацию производительности. Является логическим продолжением курса SQL с 0.

Целевая аудитория: ДБА, разработчики баз данных, аналитики данных, системные администраторы, программисты разных направлений.

Формат: онлайн-лекции - в последующем материалы будут выпущены в течении года в виде видеороликов (youtube, rutube, vkvideo) + статей в блоге. Материалы будут публиковаться на странице проекта в github

Продолжительность одной лекции: до 2 часов, количество тем, которые будут разбираться за одну лекцию зависит от скорости прохождения материала (ориентировочно по 4-5 тем).

Курс рассчитан на 5 ~ 7 лекций, исходя из плана на 31 тему.

Серверное программирование

https://aristov.tech

Зачем нужно

- считать данные из базы
- изменить данные
- удалить
- может получить доступ к файловой системе
- может получить доступ к сети, в т.ч. в интернету
- разграничить доступ и возможность изменения/удаления данных разработчиками софта для этой БД

Давайте поподробнее…

Ключевые преимущества

1. Повышение производительности

- **Снижение сетевого трафика:** Вместо пересылки тысяч строк SQL-кода и данных между клиентом и сервером отправляется лишь вызов одной функции. Это значительно уменьшает задержки (latency).
- **Предварительная компиляция и кэширование:** Хранимые процедуры компилируются и оптимизируются один раз при создании, а их планы выполнения кэшируются в памяти, что ускоряет последующие вызовы.
- **Выполнение вблизи данных (Data Locality):** Логика выполняется прямо на том же сервере, где находятся данные, что исключает накладные расходы на передачу и преобразование данных.

https://aristov.tech

Ключевые преимущества

2. Централизация бизнес-логики и обеспечение целостности данных

- **"Единственный источник истины":** Критическая бизнес-логика инкапсулируется внутри БД. Это гарантирует, что все приложения (веб, мобильные, десктопные) будут использовать одни и те же правильные алгоритмы, а не дублировать их у себя.
- **Согласованность данных:** Триггеры автоматически проверяют и поддерживают сложные ограничения целостности, которые невозможно выразить через CHECK или FOREIGN KEY.
- **Безопасность:** Можно предоставлять приложениям права на выполнение процедур, но не на прямые операции INSERT/UPDATE/DELETE к таблицам, скрывая реальную структуру данных.

https://aristov.tech

Ключевые преимущества

3. Упрощение и безопасность

- **Упрощение клиентского кода:** Клиентское приложение становится проще и "тупее". Оно просто вызывает CalculateBonus(employee_id), вместо того чтобы реализовывать сложную логику на Java/C#/python.
- **Защита от SQL-инъекций:** Поскольку параметры передаются в предварительно скомпилированные процедуры, это самый надежный способ защиты от инъекций.
- **Контроль доступа:** Администратор может дать права на выполнение процедуры, но не на изменение данных в таблицах напрямую.

https://aristov.tech

Ключевые преимущества

4. Обслуживание и управляемость

- **Легкость модификации:** Чтобы изменить бизнес-правило, нужно обновить код всего в одном месте — на сервере БД — а не переписывать все клиентские приложения.
- **Версионность и документирование:** Логика в БД может управляться системами контроля версий (Git) так же, как и любой другой код.

https://aristov.tech

Ключевые преимущества

5. Пакетная обработка и сложные операции

- **Эффективность сложных операций:** Операции, требующие множества запросов и промежуточной логики (например, ETL-процессы, сложные отчеты), выполняются максимально эффективно внутри сервера.
- **Транзакционность:** Вся логика внутри процедуры выполняется в рамках одной транзакции, что обеспечивает атомарность (все или ничего).

https://aristov.tech

Типичные сценарии применения

- **Сложные расчеты:** Начисление зарплат, бонусов, расчет налогов.
- **Валидация данных:** Сложные проверки корректности данных перед вставкой.
- **Аудит:** Автоматическое протоколирование изменений с помощью триггеров.
- **ETL-процессы:** Загрузка и преобразование данных.
- **Построение отчетов:** Агрегация больших объемов данных непосредственно в БД.

https://aristov.tech

Недостатки

- **Сложность отладки:** Отладка хранимых процедур может быть сложнее, чем клиентского кода.
- **Привязка к вендору:** SQL-диалекты и особенности procedural language (PL/pgSQL, T-SQL, PL/SQL) переносятся между СУБД (PostgreSQL, Oracle, MS SQL) с большим трудом.
- **Масштабирование:** Слишком сложная логика на сервере БД может стать "бутылочным горлышком". Иногда правильнее вынести логику в отдельный микросервис.
- **Версионность:** Требуются дисциплина и процессы для управления версиями кода БД (миграции, например, с помощью Flyway или Liquibase).

https://aristov.tech

Стенд

Стенд

ВМ - варианты https://aristov.tech/blog/ustanovka-postgresql/

Ubuntu - краткий ликбез по Линукс https://aristov.tech/blog/likbez-po-linux/

Postgres 16-18 - скоро выйдет новая версия

DBeaver + psql

Кто забыл SQL - Курс SQL с нуля и до джуна

Итоги

Итоги

Остались ли вопросы?

Увидимся на следующем занятии

Спасибо за внимание!

Когда дальше и куда?

Аристов Евгений