

ПРОГРАММА ПОВЫШЕНИЯ КВАЛИФИКАЦИИ «PostgreSQL: Уровень 2. Продвинутые возможности»

Состоит из 11 тем:

- Тема 1. Пользовательские функции и процедуры
- Тема 2. Триггеры как специальные процедуры в СУБД
- Тема 3. Обеспечение хранения атрибутов большого размера
- Тема 4. Пользовательские типы данных
- Тема 5. Слабоструктурированные типы данных
- Тема 6. Курсоры
- Тема 7. Правила
- Тема 8. Работа с иерархическими структурами
- Тема 9. Основы полнотекстового поиска
- Тема 10. Секционирование (партиционирование) таблиц
- Тема 11. Обертки внешних данных

Тема 1. Пользовательские функции и процедуры

Языки создания функций: возможность создания процедур и функций на различных языках в PostgreSQL – расширение возможностей по написанию бизнес-логики, выбор наиболее эффективного инструмента для каждой задачи. SQL — базовый язык для написания функций, который позволяет использовать встроенные SQL-запросы и конструкции для реализации функциональности. PL/pgSQL — процедурный язык PostgreSQL, возможности для поддержки переменных, условных конструкций (IF, CASE), циклов (FOR, WHILE), обработки ошибок и других элементов структурного программирования. PL/pgSQL для написания сложной серверной логики, триггеров и процедур. Сторонние языки (PL/Python, PL/Perl, PL/Tcl и др.) — обеспечение возможности подключения и использования нескольких языков программирования через механизм расширений (PL – Procedural Languages). Создание функций на языках- Python, Perl, Tcl, JavaScript: использование богатых библиотек соответствующего языка, возможность реализации сложных вычислений и алгоритмов, интеграция с внешними системами и API. Функции на SQL и PL/pgSQL: от функций с SQL-кодом к полноценной среде для разработки логики «на сервере». Процедуры (procedures): выполнение операции транзакционного управления (COMMIT и ROLLBACK). Создание функций на ненативных языках и расширение возможностей PostgreSQL.

Тема 2. Триггеры как специальные процедуры в СУБД

Назначение триггеров в СУБД: автоматизация обработки данных при операциях INSERT (вставка), UPDATE (обновление) и DELETE (удаление). Обеспечение целостности данных, ведение аудита изменений, реализация сложной бизнес-логики, которая не может быть полностью инкапсулирована

средствами приложений. Виды триггеров по времени выполнения: BEFORE (до события), AFTER (после события), INSTEAD OF (вместо события). Процедура создания триггеров и триггерных функций. Вариативность применения триггеров.

Тема 3. Обеспечение хранения атрибутов большого размера

Потребность в хранении больших данных — большие тексты, двоичные файлы, изображения, документы и т.п. Возможности PostgreSQL для работы с такими большими объектами (large objects) и средства их реализации. Сравнительный анализ возможных подходов к хранению больших данных (преимущества и недостатки): технология TOAST, специальная системная таблица pg_largeobject, дополнительное расширение lo (large object), которое облегчает работу с large objects через SQL-интерфейс. Они обеспечивают эффективное хранение и управление большими атрибутами с высоким уровнем производительности и удобства.

Тема 4. Пользовательские типы данных

Понятие и назначение перечислений. Перечисления. Диапазонные типы. Базовые типы данных: особенности хранения и представления в памяти. Преобразования между базовыми типами. Домены данных: сравнение домена с типом данных, валидация на уровне домена. Интеграция доменов с системой типов и обработкой ошибок. Роль доменов в обеспечении качества данных и поддержке бизнес-логики. Операторы, классы операторов, семейства операторов: роль в работе с типами данных, использование операторов для обеспечения интуитивного интерфейса пользовательских типов, Определение и переопределение операторов в пользовательских типах (операторные методы).

Тема 5. Слабоструктурированные типы данных

Слабоструктурированные типы: хранение динамических, изменяемых структурой данных. XML (eXtensible Markup Language) как стандарт для обмена структурированными данными с гибкой схемой: хранение, запрос и манипулирование XML-документами. PostgreSQL-тип xml для хранения XML-документов: хранения валидных XML-документов, функции для извлечения информации, преобразования, проверки схем; XPath и XQuery для выборки данных. Тип JSON/JSONB для формата обмена данными для представления объектов с произвольной структурой: поддержка индексов (GIN, GiST), возможность частичного обновления документов. Тип hstore: специализированный тип данных в PostgreSQL для хранения пар ключ-значение в виде хеш-таблицы.

Тема 6. Курсоры

Построчная обработка данных в базе данных. Использование курсоров в SQL и PLPgSQL коде. Передача курсора клиенту. Преимущества

использования курсоров: экономия памяти, удобство обработки, эффективность. Подходит для работы с большими наборами данных.

Тема 7. Правила

Создание пользовательских правил трансформации. Модификация запроса перед выполнением для реализации сложных бизнес-логик или оптимизации запросов. Принципы работы правил трансформации: **синтаксический разбор запроса, семантический анализ, трансформация, планирование и выполнение.**

Тема 8. Работа с иерархическими структурами

Основные паттерны представления иерархических структур в реляционных БД и их характеристики: Adjacency List и рекурсивные запросы (Materialized Path (предоставление каждому элементу конкретного пути и выделение родительских и дочерних элементов) и использование расширения ltree (способ работы, поддержка операции с путями, такими как поиск, фильтрация и сортировка)

Тема 9. Основы полнотекстового поиска

Документ в контексте полнотекстового поиска. Запросы FTS: **ключевые слова:** поиск по точным словам или фразам; **операторы:** использование логических операторов (AND, OR, NOT) для уточнения поиска; **подстановочные символы:** использование символов, таких как * или ?, для поиска слов с похожими окончаниями. Конфигурации. Индексы.

Тема 10. Секционирование (партиционирование) таблиц

Задачи, решаемые с помощью секционирования: обеспечение безопасности, изоляция секций, снижение рисков, удобство обслуживания. Виды секционирования. Обслуживание секций.

Тема 11. Обертки внешних данных

Назначение оберток. Использование postgres_fdw. Использование file_fdw. Возможности использования.

Директор
Института открытого образования



Т.В. Бутова