БЛОК 10. ОБЗОР РЕЗЕРВНОГО КОПИРОВАНИЯ И РЕПЛИКАЦИИ

ЛОГИЧЕСКАЯ РЕПЛИКАЦИЯ

ЦЕЛЬ

01

Понять, что такое логическая репликация, для чего она нужна

02

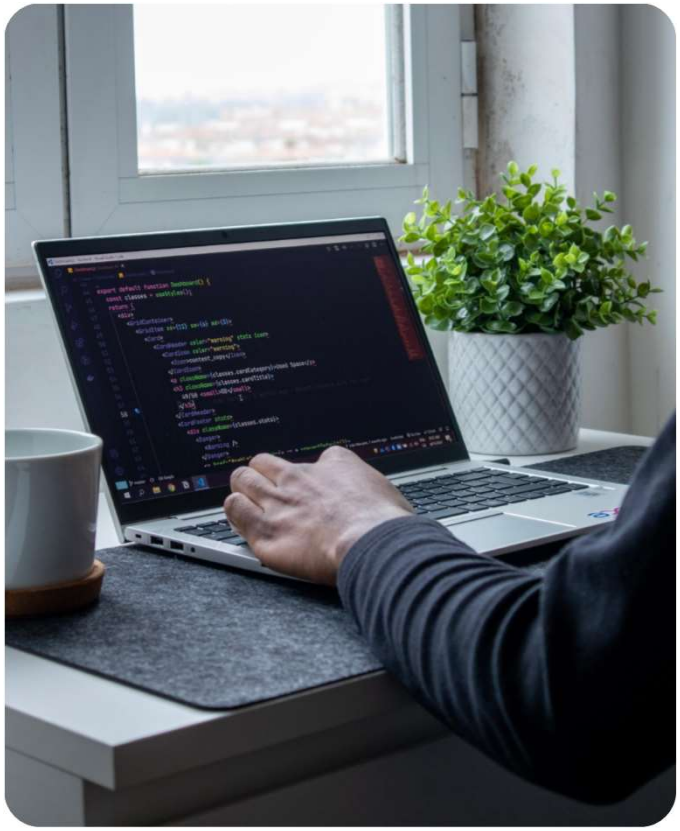
Узнать варианты логической репликации

03

Создать свой логическую реплику

СОДЕРЖАНИЕ УРОКА

1. Логическая репликация
2. Особенности
3. Практика



ЛОГИЧЕСКАЯ РЕПЛИКАЦИЯ

- поставщик-подписчик: поток данных возможен в обе стороны
- информация о строках (уровень журнала logical)
- требуется совместимость на уровне протокола
- репликация между разными основными версиями Postgres
- возможна выборочная репликация отдельных таблиц

ЛОГИЧЕСКАЯ РЕПЛИКАЦИЯ. ВАРИАНТЫ

- Встроенная логическая репликация доступна в версиях PostgreSQL, начиная с 10. Для более ранних версий аналогичный функционал доступен в расширении pg_logical.

- Для передачи логических изменений (на уровне строк) используется протокол репликации. Для работы такой репликации требуется установка уровня журнала **logical**.

- Другой способ организации логической репликации состоит в использовании триггеров для перехвата изменений, помещения этой информации в очередь событий и передача ее на другой сервер. Такой способ, однако, менее эффективен, и уходит в прошлое (**Slony-I**).

- При логической репликации у сервера нет выделенной роли мастера или реплики, что позволяет организовать в том числе и двунаправленную репликацию.

ЛОГИЧЕСКАЯ РЕПЛИКАЦИЯ. ПРИНЦИП РАБОТЫ

Публикующий сервер

- выдает изменения данных построчно в порядке их фиксации
- (реплицируются команды INSERT, UPDATE, DELETE), в 11 версии добавили TRUNCATE
- возможна начальная синхронизация
- всегда используется слот логической репликации
- **DDL не передаются, то есть таблицы-приемники на стороне подписчика надо создавать вручную.**
- Данные последовательностей не реплицируются.
- Реплицировать данные возможно только из базовых таблиц в базовые таблицы. То есть таблицы на стороне публикации и на стороне подписки должны быть обычными, а не представлениями, мат. представлениями, секционированными или сторонними таблицами.
- применение изменений происходит без выполнения команд SQL и связанных с этим накладных расходов на разбор и планирование, что уменьшает нагрузку на подписчика.
- параметр **wal_level = logical**

ЛОГИЧЕСКАЯ РЕПЛИКАЦИЯ. ПРИНЦИП РАБОТЫ

Подписчики

- получают и применяют изменения
- без разбора, трансформаций и планирования — сразу выполнение
- возможны конфликты с локальными данными
- триггеры срабатывают для каждого подписчика отдельно

ЛОГИЧЕСКАЯ РЕПЛИКАЦИЯ. ОСОБЕННОСТИ

Режимы идентификации для изменения и удаления

- столбцы первичного ключа (по умолчанию)
- столбцы указанного уникального индекса с ограничением NOT NULL
- все столбцы
- без идентификации (по умолчанию для системного каталога)

Конфликты — нарушение ограничений целостности

- репликация приостанавливается до устранения конфликта вручную
- либо исправление данных,
- либо пропуск конфликтующей транзакции

ЛОГИЧЕСКАЯ РЕПЛИКАЦИЯ. ПРАКТИКА

Выполним шаги:

1. Будем использовать 2 сервера с предыдущего урока
2. Сделаем promote 2 сервера - получим 2 отдельных сервера!
3. Создадим публикацию таблицы на 1 сервере
4. Подпишемся на эту публикацию на 2 сервере
5. Добавим запись на 1 сервере в эту таблицу и убедимся, что она появилась на 2 сервере

ПОДВЕДЕНИЕ ИТОГОВ

ИТОГИ ЗАНЯТИЯ

01

Поняли, что такое логическая репликация, для чего она нужна

02

Узнали варианты логической репликации

03

Создали свой логическую реплику

ЗАДАНИЕ ДЛЯ САМОПРОВЕРКИ

Цель задания:
Создать свою логическую реплику 1 таблицы

Пошаговый план выполнения:

1. Использовать 2 сервера с предыдущего урока
2. Сделать promote 2 сервера - получить 2 отдельных сервера
3. Включить логический уровень журналирования на 1 сервере
4. Перезагрузить 1 сервер
5. Создать публикацию таблицы на 1 сервере
6. Создать аналогичную таблицу на 2 сервере, если ее там нет
7. Подписаться на эту публикацию на 2 сервере
8. Добавить запись на 1 сервере в эту таблицу
9. Убедиться, что она появилась на 2 сервере
10. Выполните задание самостоятельно. Свое решение вы можете сравнить с эталонным (приложено к уроку)

Задание закончено

СПАСИБО!

На следующем занятии мы рассмотрим тему:

- Best practice

end