1. Министерство образования и науки Российской Федерации
2. Санкт-Петербургский Политехнический Университет Петра Великого
3. —
4. Институт кибербезопасности и защиты информации

**ЛАБОРАТОРНАЯ РАБОТА № 7**

**Логическая репликация в СУБД PostgreSQL**

по дисциплине «Системы управления базами данных»

1. Выполнила
2. студентка гр. 4851003/90801 Кулеева А.Г.

1. Руководитель
2. ассистент Зубков Е.А.
3. Санкт-Петербург
4. 2023

# **Цель**

Получение навыков настройки и управления процессом логической репликации на базе СУБД PostgreSQL.

# Задание

Сформируйте стенд, состоящий из двух виртуальных машин на базе ОС общего назначения.

Выполните настройку параметров функционирования основного сервиса СУБД PostgreSQL согласно логической репликации для каждого компонента стенда. Обоснуйте установленные значения параметров.

Разработайте SQL-скрипты, выполняющие:

* создание и удаление публикаций c поддержкой операций UPDATE и DELETE;
* создание и удаление подписок c поддержкой операций UPDATE и DELETE;
* ознакомление со статусом процесса репликации.

Заполните базу данных в компоненте стенда, предназначенном для публикации, и запустите процесс публикации.

Проведите анализ взаимодействия компонентов стенда:

* передаваемые сетевые пакеты протокола TCP (ПО Wireshark);
* изменения в журналах предзаписи (Write-Ahead Logging, WAL);
* изменения в реплицируемых базах данных.

Выполните модификацию данных в таблицах базы данных компонента стенда, предназначенного для подписок.

Выполните аналогичную модификацию данных в таблицах базы данных компонента стенда, предназначенного для публикации.

Проведите повторный анализ взаимодействия компонентов стенда и сделайте выводы.

# Ход работы

## Настройка PostgreSQL

Для поддержки логической репликации между серверами необходимо изменить несколько параметров конфигурации. Во-первых, нужно настроить Postgres для прослушивания частного сетевого интерфейса. Затем нужно внести соответствующие настройки, чтобы включить репликацию на клиенте.

На публикующем сервере необходимо изменить  главный конфигурационный файл postgresql.conf:

Найти строку:

...  
#listen\_addresses = 'localhost'         # what IP address(es) to listen on;  
...

Раскомментировать ее, удалив #, и указать внутренний ip-адрес публикующего сервера, чтобы включить поддержку соединений по частной сети.

...  
listen\_addresses = 'localhost, 192.168.23.1'  
...

Теперь сервер прослушивает входящие соединения в частной сети и на loopback интерфейсе. Далее необходимо найти строку:

...  
#wal\_level = replica                    # minimal, replica, or logical  
...

Раскомментировать ее и указать в PostgreSQL Write Ahead Log (WAL) уровень logical. Это увеличивает объем записей в логе, добавляя необходимую информацию для извлечения отличий или изменений в конкретных наборах данных:

...  
wal\_level = logical  
...

Записи этого лога будут использоваться сервером-подписчиком, что позволяет реплицировать операции записи высокого уровня с мастера.

Теперь требуется отредактировать pg\_hba.conf, этот файл управляет поддерживаемыми хостами, аутентификацией и доступом к БД. В конец файла нужно добавить строку, которая поддерживает входящие сетевые соединения от подписчика. Используйте внутренний IP-адрес подписчика, сервер должен поддерживать соединения всех пользователей и баз данных:

...  
# TYPE    DATABASE    USER   ADDRESS         METHOD  
...  
host      all         all    192.168.23.138/32      md5

Теперь публикующий сервер поддерживает сетевые соединения от подписчика по парольному хэшу.

Затем требуется изменить настройки межсетевого экрана, чтобы пропустить трафик по порту 5432. Перезапустим сервер PostgreSQL, чтобы изменения вступили в силу.

## Создание ролей, публикаций и подписок

Создадим схему данных на основе варианта из л.р. 1 на обоих серверах. Код представлен в Приложении 1.

На публикующем сервере создадим новую роль пользователя с помощью опции REPLICATION и паролем. Атрибуту REPLICATION нужно присвоить любую роль для репликации, передать полные права на БД и на все таблицы в этой БД новому пользователю.

CREATE ROLE listener WITH REPLICATION LOGIN PASSWORD 'anna';

GRANT ALL PRIVILEGES ON DATABASE lab7 TO listener;

GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO listener;

Публикации — это механизм, который PostgreSQL использует для создания таблиц для репликации. Сервер базы данных будет отслеживать внутреннее состояние соединения и репликации всех серверов-реплик, связанных с данной публикацией. Создадим на сервере публикацию и добавим в нее таблицу.

CREATE PUBLICATION my\_publication;

ALTER PUBLICATION my\_publication ADD TABLE document;

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 1 ― Создание публикации

Теперь можно добавить подписчиков, которые смогут загружать ее данные. Подписка используется PostgreSQL для подключения к существующим публикациям. Публикация может иметь множество подписок на разных серверах-подписчиках, а серверы-подписчики могут также иметь свои собственные публикации с подписчиками. Чтобы получить доступ к данным из таблицы, созданной на публикующем сервере, нужно создать подписку на публикацию my\_publication.

Переходим на сервер подписчик и создем подписку my\_subscription. Команда CREATE SUBSCRIPTION задает имя подписки, а параметр CONNECTION определяет строку для доступа к публикующему серверу. Эта строка включает сведения о соединении главного сервера и учетные данные для входа, включая имя пользователя и пароль, а также имя базы данных.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 2 ― Создание подписки

## Тестирование

Ниже на рисунках 3-6 представлено тестирование запросов и отображение данных на обоих серверах

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок 3 ― Insert (сверху публикующий сервер, снизу подписчик)

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

Рисунок 4 ― Update (сверху публикующий сервер, снизу подписчик)

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание

Рисунок 5 ― Update (сверху подписчик, снизу публикующий сервер)

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

Рисунок 6 — Delete (сверху публикующий сервер, снизу подписчик)

На рисунке 7 представлены сведения о репликации между БД, полученные с помощью запроса:

select \* from pg\_stat\_replication;

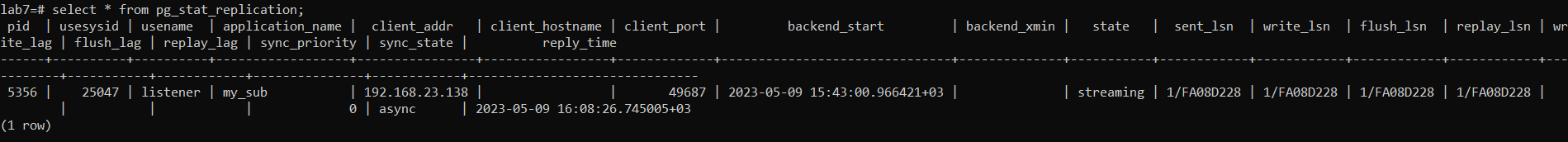


Рисунок 7 ― Результат запроса

На рисунке 8 представлена часть дампа сетевого трафика между двумя машинами-серверами.

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание

Рисунок 8 — Сетевой трафик

# Вывод

В результате выполнения данной лабораторной работы были получены навыки создания публикаций, настройки баз данных-подписчиков, а также управления процессом логической репликации на базе СУБД PostgreSQL. Были проанализированы реплицированные данные при операциях вставки, обновления и удаления, а также набор пакетов, полученный при передаче данных в сети.

ПРИЛОЖЕНИЕ 1

CREATE DATABASE lab7;

\c lab7

CREATE TYPE sec\_lvl AS ENUM ('1', '2', '3');

CREATE TABLE document (

num\_doc BIGSERIAL NOT NULL PRIMARY KEY,

type\_doc VARCHAR(50) NOT NULL UNIQUE,

fio\_respons VARCHAR(100) NOT NULL UNIQUE,

department VARCHAR(50) NOT NULL UNIQUE,

date\_start DATE NOT NULL UNIQUE,

date\_end DATE,

secret sec\_lvl,

content json);

CREATE TYPE oper AS ENUM ('c', 'r', 'u', 'd');

CREATE TABLE change\_log (

num\_doc BIGSERIAL NOT NULL UNIQUE REFERENCES document (num\_doc),

date\_change DATE NOT NULL UNIQUE,

fio VARCHAR(100) NOT NULL UNIQUE,

operation oper,

PRIMARY KEY (num\_doc, date\_change, fio));

//////////////////////////////////////////////////////

INSERT INTO document (

type\_doc, fio\_respons, department, date\_start, date\_end, secret)

VALUES (‘docx’, ‘Ivanov’, ‘IT’, ’2022-01-01’, ’2025-01-01’, ‘2’);

INSERT INTO document (

type\_doc, fio\_respons, department, date\_start, date\_end, secret)

VALUES (‘pdf’, ‘Smirnov’, ‘HR’, ’2020-03-01’, ’2024-01-01’, ‘1’);

update document set date\_end = '2022-04-03' where num\_doc = 2;

delete from document where num\_doc = 1;

///////////////////////////////////////////////////////

CREATE ROLE listener WITH REPLICATION LOGIN PASSWORD 'anna';

GRANT ALL PRIVILEGES ON DATABASE lab7 TO listener;

GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO listener;

CREATE PUBLICATION my\_publication;

ALTER PUBLICATION my\_publication ADD TABLE document;

//////////////////////////////////////////////////

CREATE SUBSCRIPTION my\_subscription CONNECTION 'host=192.168.23.1 port=5432 password=anna user=listener dbname=lab7' PUBLICATION my\_publication;