# ЛАБОРАТОРНАЯ РАБОТА №6 «Исследование принципов репликации данных»

## Цель работы

Исследовать способы репликации баз данных и их влияние на скорость доступа с данным. Изучить основы репликации данных на примере MySQL.

## Постановка задачи

Используя методику, описанную в приложении, создать простой HTTP

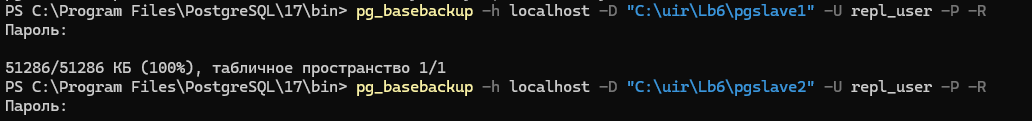
+ PostgreSQL сервер и реализовать репликацию. Вариант таблицы, кол-во SLAVE серверов и номера портов для SLAVE серверов приведены в таблице ниже:

Вариант задания

|  |  |  |  |
| --- | --- | --- | --- |
| № | Название таблицы | Кол-во SLAVE  серверов | Номера портов SLAVE  серверов |
| 3 | cities (id, title, short\_code) | 2 | 3308 3309 |

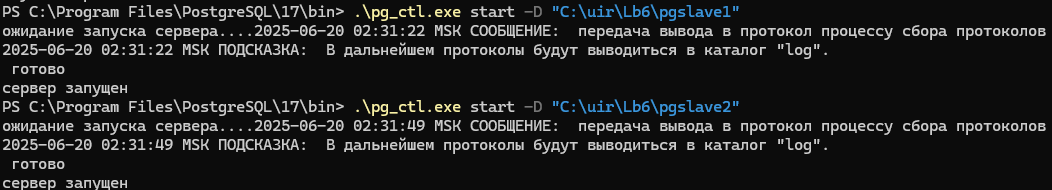
## Ход работы

Была создана реплика с помощью команды pg\_basebackup. Процесс создания реплики показан на рисунке 6.1.



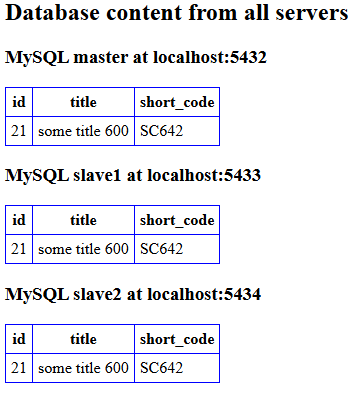
* + - * 1. Создание реплики при помощи pg\_basebackup

После того как реплика была создана, был запущен сервер с репликой. Перед запуском сервера в файле конфигурации для каждой из реплик был задан незанятый порт. Запуск сервера показан на рисунке 6.2.



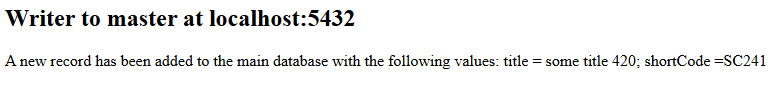
* + - * 1. Запуск SLAVE сервера для репликации данных

Для проверки корректности репликации выполним запрос на вывод данных как с master, так и с SLAVE серверов. Вывод данных показан на рисунке 6.3.



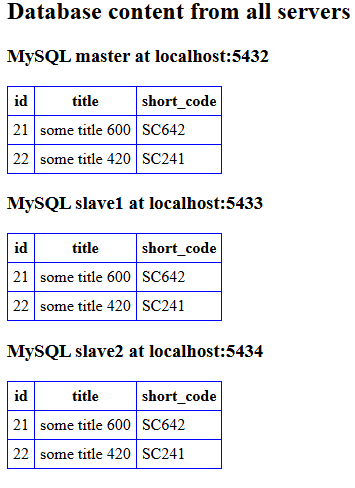
* + - * 1. Вывод данных из базы данных и синхронизация на SLAVE

Для проверки работы репликации добавим новую запись на master сервер. Эта запись будет автоматически реплицирована на все подключенные SLAVE серверы. Генерация записи и добавление её в таблицу показано на рисунке 6.4.



* + - * 1. Добавление новой записи в master

После добавления новой записи на master сервер необходимо проверить, что запись появилась и на SLAVE сервере(рисунок 6.5).



* + - * 1. Результат добавления

## Вывод