|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ  Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |

Институт комплексной безопасности и специального приборостроения

Кафедра КБ-14 «Интеллектуальные системы информационной безопасности»

### Политики безопасности баз данных

### Практическая работа № 4

# ОТЧЁТ

# Выполнил студент группы БСБО-07-20 Любовский С.В.

# Выполнение задания 1.

1. Создать новую базу данных с именем wal\_db.

CREATE DATABASE wal\_db;

1. Создать таблицу orders с колонками:
   1. id: уникальный идентификатор заказа (целое число, первичный ключ, автоинкремент)
   2. customer\_name: имя клиента (varchar(255))
   3. order\_date: дата размещения заказа (date)
   4. total\_amount: общая сумма заказа (numeric)

CREATE TABLE orders (

id serial primary key,

customer\_name varchar(255),

order\_date date,

total\_amount numeric(16, 2)

);

1. Вставить в таблицу orders несколько примеров данных.

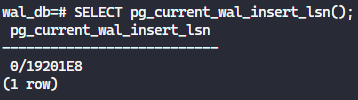
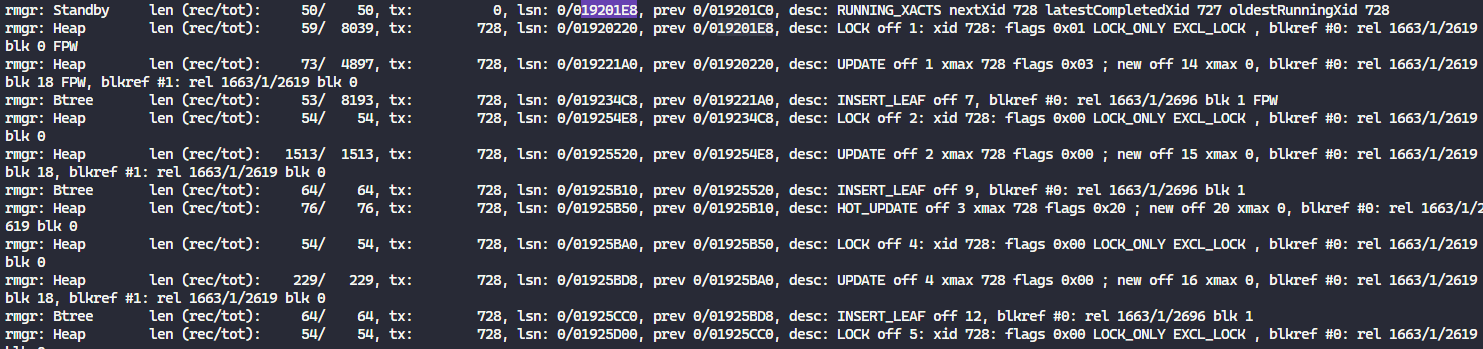
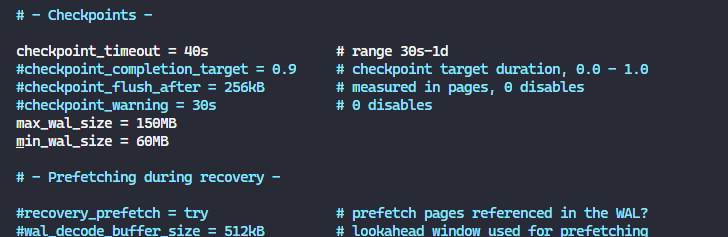
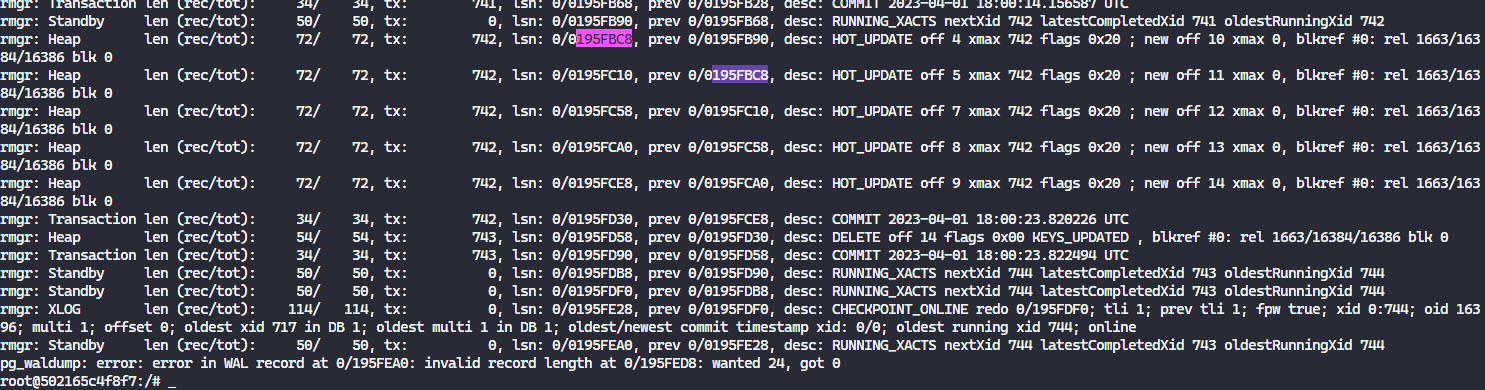
INSERT INTO orders (customer\_name, order\_date, total\_amount)

VALUES

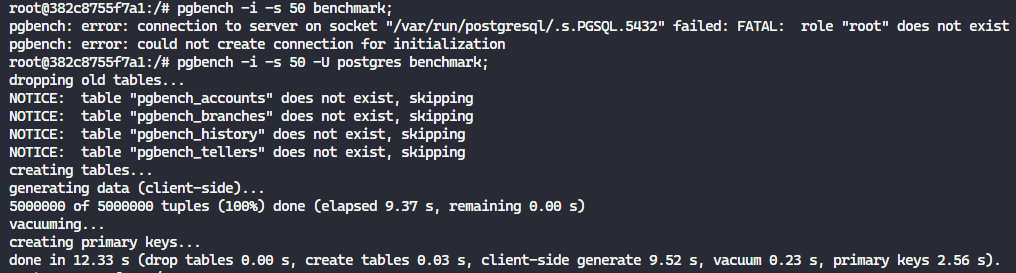
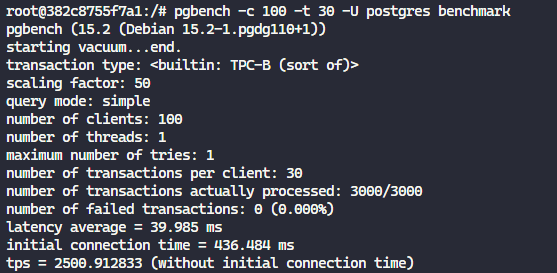
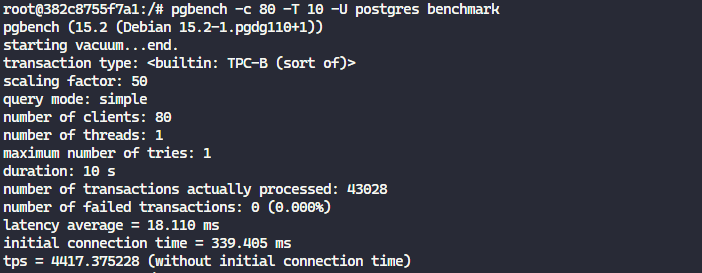
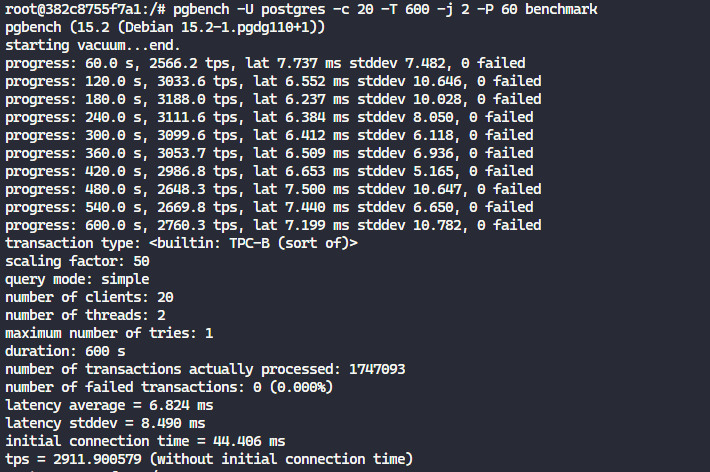
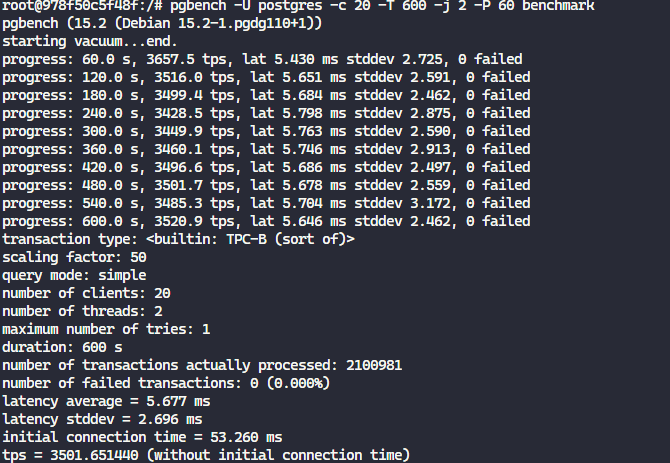
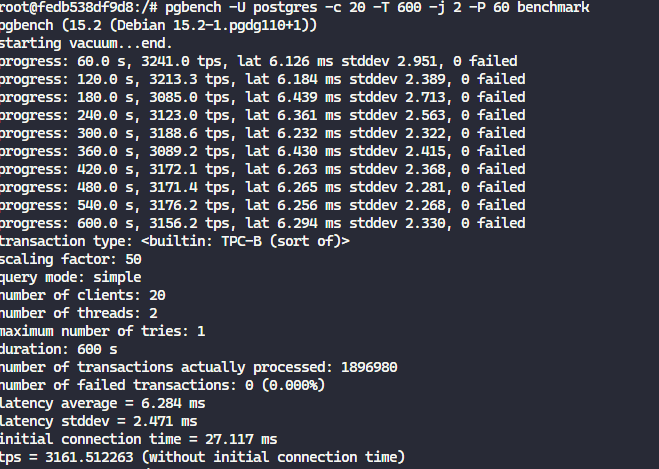
('ivan', '02-02-2022'::date, 1010.55),

('petr', '03-03-2022'::date, 999.99),

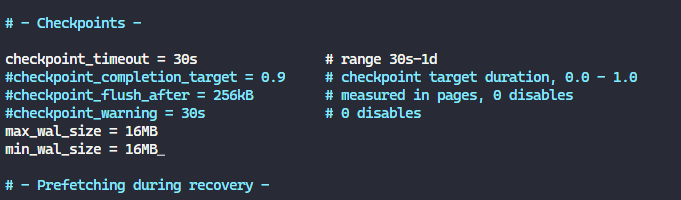
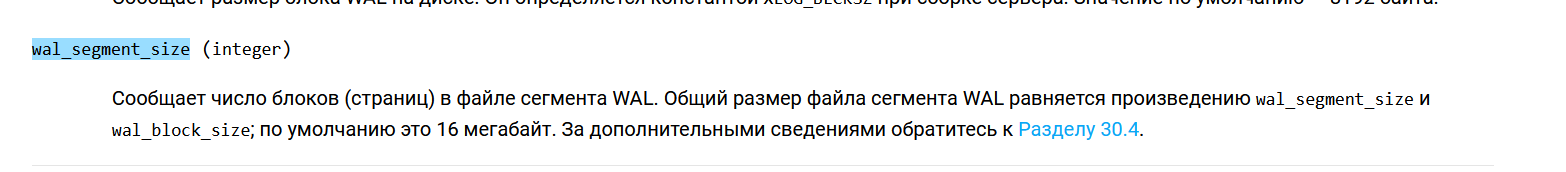
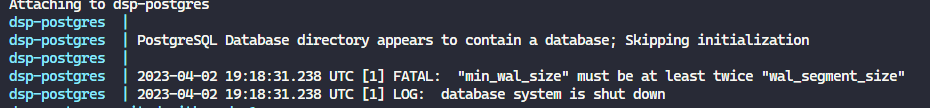
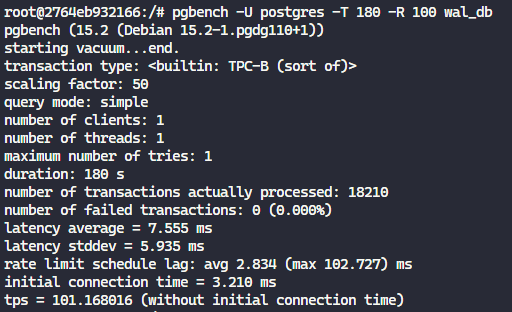
('nastya', '04-04-2022'::date, 949.49);

1. Узнать сколько байт занимают сгенерированные журнальные записи  
     
   SELECT pg\_current\_wal\_insert\_lsn();  
   
2. Изменить некоторые из существующих записей в таблице orders  
     
   UPDATE orders SET total\_amount = total\_amount + 500 WHERE order\_date > '01-03-2022'::date;
3. Удалить несколько записей из таблицы orders  
     
   DELETE FROM orders WHERE customer\_name = 'nastya';
4. Проверить содержимое файлов журнала WAL, чтобы увидеть записи операций, выполненных в шагах 4-6 (Для проверки содержимого файлов журнала WAL можно воспользоваться утилитой pg\_waldump).  
     
   
5. Настроить параметры, связанные с WAL, такие как max\_wal\_size и checkpoint\_timeout в postgresql.conf.  
     
   
6. Перезапустить сервер PostgreSQL, чтобы применить изменения в конфигурации
7. Повторить шаги 4-7, чтобы наблюдать изменения в файлах WAL из-за новых параметров конфигурации.  
     
   

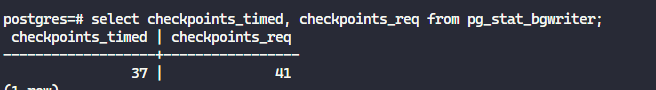
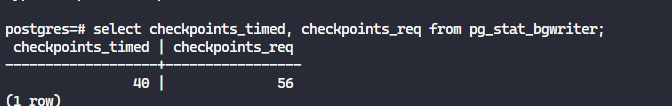
# Выполнение задания 2.

1. Создайте новую базу данных benchmark в PostgreSQL, которая будет использоваться для тестирования производительности с помощью pgbench.  
     
   CREATE DATABASE benchmark;
2. Настройте базу данных benchmark используя утилиту pgbench с параметрами -i и -s 50 (объясните что делают эти параметры)  
     
   Ключ -i отвечает за инициализацию тестовых таблиц  
   Ключ -s отвечает за масштабирование тестов, умножает количество генерируемых строк на этот коэффициент  
     
   
3. Запустите команду pgbench на созданной таблице с разными параметрами, такими как количество клиентов, количество транзакций и длительность тестирования.  
     
   Запуск с 100 клиентов и 30 транзакциями  
     
     
   Запуск с 80 клиентов и 10 секундами продолжительности тестов  
   
4. Проанализируйте результаты тестирования, смотрите на показатели, такие как количество транзакций в секунду, время выполнения транзакций и общее время работы  
     
   В первом случае тесты были запущены для фиксированного количества транзакций (30 транзакций на клиент – 3000 в сумме). Среднее время на транзакцию у одного клиента – 39мс. Количество транзакций в секунду – 2500  
     
   Во втором случае тесты были запущенны на фиксированное время и 80 клиентами. Среднее время одной транзакции составило – 18мс, количество в секунду – 4417.
5. Попробуйте изменить параметры pgbench и повторить тестирование, чтобы понять, как они влияют на производительность базы данных. Используйте различные значения для параметров, а также изменяйте конфигурационные параметры PostgreSQL, такие как shared\_buffers и work\_mem, чтобы увидеть, как это влияет на производительность.  
     
   Проведем тестирование на стандартной конфигурации. Будем использовать 20 подключений 2 потока и тест продолжительностью 10 минут, чтобы получить +- воспроизводимые результаты.  
   Получили 2911 tps. Теперь увеличим объем памяти который postgresql использует для кэширования. За это отвечает параметр shared\_buffers. По умолчанию он равен 128MB но сейчас большинство компьютеров имеют объемы памяти большие на порядки. Установим значения 512MB и посмотрим что получится.  
   Полученный результат на ~600 транзакций в секунду больше. Попробуем увеличить еще в 2 раза до 1024MB  
   В этот раз результаты оказались неожиданными, tps просел на 400. Возможно это связано с тем, что запуск производился на ноутбуке в WSL и условия могут меняться от количества зарядки и других факторов. Но в целом дальнейшее увеличение этого параметра не сильно улучшит ситуацию.

# Выполнение задания 3.

1. Для базы данных из первого задания настройте выполнение контрольной точки раз в 30 секунд. Установите параметры min\_wal\_size и max\_wal\_size в 16 МБ.  
     
     
   min\_wal\_size и max\_wal\_size должны быть минимум в два раза больше чем wal\_segment\_size. Получаем ошибку.  
     
     
     
     
   Установим 32MB
2. Несколько минут c помощью утилиты pgbench подавайте нагрузку 100 транзакций/сек.  
     
   
3. Измерьте, какой объем журнальных файлов был сгенерирован за это время. Оцените, какой объем приходится в среднем на одну контрольную точку.  
     
     
     
   При весе каждого дампа 16мб суммарный их объем будет равен 32мб.

Контрольные точки срабатывают каждые 30 секунд. Учитывая время работы бенчмарка равного 3 минуты, то выходит, за одну минуту приходится объем равный 16мб, тогда за 30 секунд: 16мб / 2, что будет равно 8мб.

1. Проверьте данные статистики: все ли контрольные точки выполнялись по расписанию? Как можно объяснить полученный результат?  
     
     
   данные в таблице до бенчмарка  
     
     
   после  
     
     
   Видно что за время выполнения бенчмарка было запрошено и выполнено 15 чекпоинтов, по времени было выполнено 3. Вероятно, все дело в том, что система не успевает собирать их в дамп вовремя из-за небольшого размера max\_wal\_size.
2. Сбросьте настройки к значениям по умолчанию.  
   