**Task 22 (Shared Buffers)**

Папка задания в репозитории:

<https://github.com/artemgur/AdvancedPostgreSQL/tree/master/Task%2022%20(Shared%20Buffers)>

Flyway-скрипты хранятся в следующей папке:

<https://github.com/artemgur/AdvancedPostgreSQL/tree/master/flyway/sql>

**Создание таблиц**

| CREATE TABLE sb\_1000 (  a int,  b int,  c int );  CREATE TABLE sb\_10000 (  a int,  b int,  c int );  CREATE TABLE sb\_100000 (  a int,  b int,  c int ); |
| --- |

**Заполнение таблиц**

| INSERT INTO sb\_1000  SELECT i, round(random() \* 1000)::int, round(random() \* 1000)::int  FROM generate\_series(1, 1000) AS k(i);  INSERT INTO sb\_10000  SELECT i, round(random() \* 1000)::int, round(random() \* 1000)::int  FROM generate\_series(1, 10000) AS k(i);  INSERT INTO sb\_100000  SELECT i, round(random() \* 1000)::int, round(random() \* 1000)::int  FROM generate\_series(1, 100000) AS k(i); |
| --- |

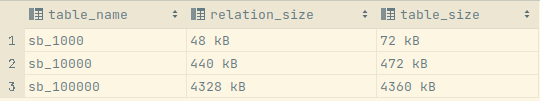
**Размер таблиц**

Чтобы узнать необходимый размер shared\_buffers, рассмотрим размер таблиц.

| SELECT table\_name,  pg\_size\_pretty(pg\_relation\_size(table\_name)) AS relation\_size,  pg\_size\_pretty(pg\_table\_size(table\_name)) AS table\_size FROM unnest(ARRAY['sb\_1000', 'sb\_10000', 'sb\_100000']) as k(table\_name) |
| --- |

Я рассматривал как table\_size, так и relation\_size. Это понадобится в дальнейшем.

Были получены следующие результаты:



**Используемый размер shared\_buffers**

Далее необходимо узнать, какой объем shared\_buffers используется в данный момент.

Для этого я подключил расширение pg\_buffercache и создал VIEW на основе запроса из презентации.

| CREATE EXTENSION IF NOT EXISTS pg\_buffercache;  CREATE VIEW buffer\_view AS SELECT c.relname,  d.datname,  count(\*) \* current\_setting('block\_size')::integer AS buffered,  round(100.0 \* count(\*) /  (SELECT setting  FROM pg\_settings  WHERE name = 'shared\_buffers')::integer, 1) AS buffer\_percent,  round(100.0 \* count(\*) \* current\_setting('block\_size')::integer /  pg\_relation\_size(c.oid), 1) AS percent\_of\_relation,  round(100.0 \* count(\*) \* current\_setting('block\_size')::integer /  pg\_table\_size(c.oid), 1) AS percent\_of\_table FROM pg\_class c  INNER JOIN pg\_buffercache b  ON b.relfilenode = pg\_relation\_filenode(c.oid)  INNER JOIN pg\_database d ON b.reldatabase = d.oid GROUP BY c.oid, c.relname, d.datname; |
| --- |

К запросу из презентации я добавил столбец, соответствующий relation\_size. Также VIEW содержит не отформатированный с помощью pg\_size\_pretty размер использованного таблицей shared\_buffers. Это понадобится для того, чтобы можно было получить используемый объем shared\_buffers, просуммировав значения столбца.

Для получения использованного объема shared\_buffers я использовал следующий запрос:

| SELECT pg\_size\_pretty(sum(buffered)) AS buffered\_total FROM buffer\_view |
| --- |

Был получен следующий результат:



**Максимальный размер shared\_buffers**

Чтобы узнать заданный в postgresql.conf максимальный размер shared\_buffers, можно использовать следующий запрос:

| SELECT current\_setting('shared\_buffers') AS shared\_buffers\_max\_size |
| --- |

В моем случае заданный в postgresql.conf размер shared\_buffers равен значению по умолчанию — 128 МБ. Этого достаточно, чтобы разместить в shared\_buffers рассматриваемые таблицы.

**Задание максимального размера shared\_buffers**

Максимальный размер shared\_buffers можно изменить в postgresql.conf. Для этого нужно изменить значение параметра shared\_buffers и перезапустить PostgreSQL.

Для определения оптимального размера shared\_buffers можно использовать PGTune.

**Prewarm**

Для загрузки таблиц в shared\_buffers я подключил расширение pg\_prewarm и запустил prewarm для таблиц.

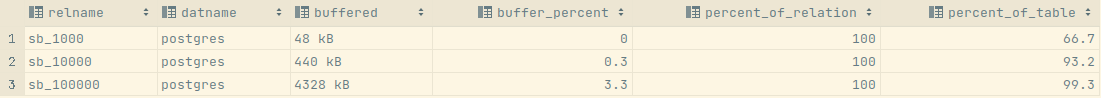
| CREATE EXTENSION IF NOT EXISTS pg\_prewarm;  SELECT pg\_prewarm('sb\_1000'); SELECT pg\_prewarm('sb\_10000'); SELECT pg\_prewarm('sb\_100000'); |
| --- |

**Проверка загрузки таблиц в shared\_buffers**

Чтобы проверить, загрузились ли таблицы в shared\_buffers, я использовал следующий запрос, использующий созданный ранее VIEW.

| SELECT relname, datname,   pg\_size\_pretty(buffered) AS buffered,   buffer\_percent, percent\_of\_relation, percent\_of\_table FROM buffer\_view WHERE relname LIKE 'sb\_1000%' |
| --- |

Были получены следующие результаты:



Таким образом, размер буфера этих таблиц после prewarm соответствует не table\_size, а relation\_size.