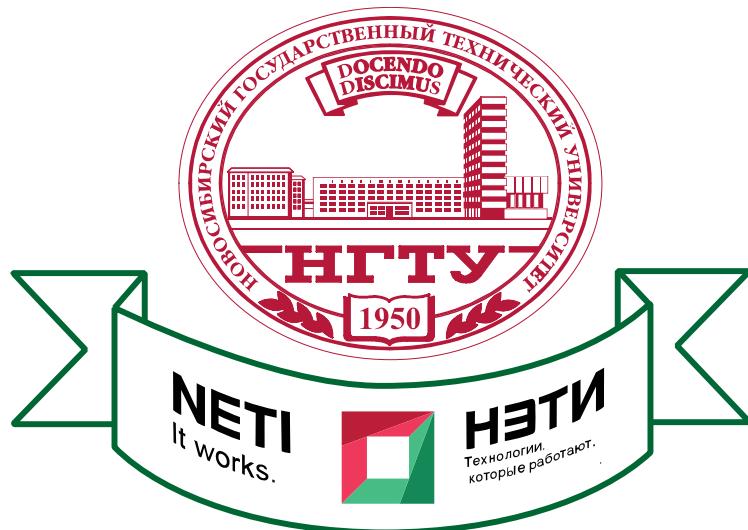


Министерство науки и высшего образования
Российской Федерации

Федеральное государственное бюджетное
образовательное учреждение высшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

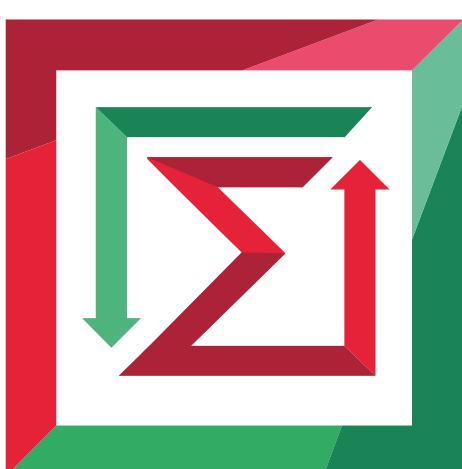


Кафедра теоретической и прикладной информатики

Лабораторная работа № 3

по дисциплине «Администрирование информационных систем»

Транзакции, блокировки, журнал



Факультет: ПМИ
Группа: ПМИ-02
Бригада: 8
Студенты: Сидоров Даниил,
Дюков Богдан
Преподаватель: Аврунев О.Е.

Новосибирск

2026

Цель работы

Ознакомиться с принципами реализации транзакций в PostgreSQL – блокировки, журнал предзаписи (WAL).

Исходные данные

№ Бригады	Таблица
2,8	bookings.tickets

№ Бригады	Количество одновременно запущенных программ
7,8	7

Ход работы

- Получили значения параметров конфигурации, определяющих работу блокировок:
 - Предельное время ожидания снятия блокировки (1 запрос)
 - Запись об ожидании блокировки сверх установленного в журнал сервера (2 запрос)

```
demo=# show deadlock_timeout;
deadlock_timeout
-----
 1s
(1 row)

demo=# show log_lock_waits;
log_lock_waits
-----
 off
(1 row)
```

Включаем параметр `log_lock_waits`, чтобы в случае, если сессия ожидает блокировку дольше, чем `deadlock_timeout`, создавалось сообщение в журнале:

```
demo=# alter system set log_lock_waits=on;
ALTER SYSTEM
demo=# select pg_reload_conf();
 pg_reload_conf
-----
 t
(1 row)

demo=# show log_lock_waits;
 log_lock_waits
-----
 on
(1 row)
```

Получили значения параметров конфигурации, отвечающих за работу WAL:

- Уровень информации, помещаемой в WAL (1 запрос);
- Выполняется ли запись полного образа страницы в WAL при первом изменении, после контрольной точки (2 запрос);
- Выполняется ли сжатие образа страницы (3 запрос);
- Пауза между записью буфера WAL (4 запрос);

```
demo=# show wal_level;
 wal_level
-----
 replica
(1 row)

demo=# show full_page_writes;
 full_page_writes
-----
 on
(1 row)

demo=# show wal_compression;
 wal_compression
-----
 off
(1 row)

demo=# show wal_writer_delay;
 wal_writer_delay
-----
 200ms
(1 row)
```

Получили список сегментов WAL:

```
demo=# select * from pg_ls_waldir();
      name       |   size   |      modification
-----+-----+-----+
000000010000000000000000000013 | 16777216 | 2024-02-29 01:10:35+07
000000010000000000000000000014 | 16777216 | 2024-02-13 05:20:29+07
000000010000000000000000000015 | 16777216 | 2024-02-13 05:20:30+07
000000010000000000000000000016 | 16777216 | 2024-02-13 05:20:30+07
000000010000000000000000000017 | 16777216 | 2024-02-13 05:20:31+07
000000010000000000000000000018 | 16777216 | 2024-02-13 05:20:32+07
000000010000000000000000000019 | 16777216 | 2024-02-13 05:20:33+07
00000001000000000000000000001A | 16777216 | 2024-02-13 05:20:34+07
00000001000000000000000000001B | 16777216 | 2024-02-13 05:20:35+07
00000001000000000000000000001C | 16777216 | 2024-02-13 05:20:36+07
00000001000000000000000000001D | 16777216 | 2024-02-13 05:20:37+07
00000001000000000000000000001E | 16777216 | 2024-02-13 05:20:39+07
00000001000000000000000000001F | 16777216 | 2024-02-13 05:20:39+07
000000010000000000000000000020 | 16777216 | 2024-02-13 05:20:41+07
000000010000000000000000000021 | 16777216 | 2024-02-13 05:21:34+07
000000010000000000000000000022 | 16777216 | 2024-02-13 05:21:35+07
000000010000000000000000000023 | 16777216 | 2024-02-13 05:21:35+07
000000010000000000000000000024 | 16777216 | 2024-02-16 00:52:41+07
(18 rows)
```

```
demo=#
```

Получили текущий идентификатор транзакции и текущий LSN:

```
demo=# select * from txid_current();
txid_current
-----
5912
(1 row)

demo=# select * from pg_current_wal_lsn();
pg_current_wal_lsn
-----
0/13BF41E8
(1 row)
```

Получили идентификатор последней контрольной точки:

```
demo=# select * from pg_control_checkpoint();
checkpoint_lsn | redo_lsn | redo_wal_file | timeline_id | prev_timeline_id | full_page_writes | next_xid | next_oid | next_multixact_id | next_multi_offset | oldest_xid | olde
st_xid_dbid | oldest_active_xid | oldest_multi_xid | oldest_multi_dbid | oldest_commit_ts_xid | newest_commit_ts_xid | checkpoint_time
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
0/13BF37A8 | 0/13BF3770 | 000000010000000000000000000013 | 1 | 1 | t | 0 | 0:5912 | 16541 | 1 | 0 | 722 |
(1 row)

demo=#
```

2. Установить драйвер postgres для python (sudo yum install python-psycopg2):

```
[dba@centos-stream-8 ~]$ sudo yum install python3-psycopg2
[sudo] password for dba:
Last metadata expiration check: 1:19:16 ago on Wed 28 Feb 2024 09:56:20 PM +07.
Dependencies resolved.
=====
 Package          Architecture Version      Repository    Size
=====
Installing:
 python3-psycopg2        x86_64      2.7.5-7.el8   appstream   172 k

Transaction Summary
=====
Install 1 Package

Total download size: 172 k
Installed size: 545 k
Is this ok [y/N]: y
Downloading Packages:
python3-psycopg2-2.7.5-7.el8.x86_64.rpm       416 kB/s | 172 kB   00:00
-----
Total                                         203 kB/s | 172 kB   00:00

Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing :                                                 1/1
  Installing : python3-psycopg2-2.7.5-7.el8.x86_64           1/1
  Running scriptlet: python3-psycopg2-2.7.5-7.el8.x86_64     1/1
  Verifying   : python3-psycopg2-2.7.5-7.el8.x86_64           1/1

Installed:
  python3-psycopg2-2.7.5-7.el8.x86_64

Complete!
[dba@centos-stream-8 ~]$
```

3. Запустили определенное количество python-сценариев, выполняющих в цикле модификацию строк таблицы.

Создаем файл .py:

```
[dba@centos-stream-8 ~]$ vi script.py
```

Записываем в него следующий сценарий (удаление последнего символа из текущего значения passenger_id):

```
import psycopg2
from time import sleep
conn = psycopg2.connect("dbname='demo' user='dba' host='127.0.0.1' password='disidorov2020'")

cur = conn.cursor()
cur.execute("""select a.*
               from bookings.tickets a
              order by a.ticket_no limit 100""")
rows = cur.fetchall()
for i in rows:
    cur.execute("""update bookings.tickets
                  set passenger_id = SUBSTRING(passenger_id, 1, LENGTH(passenger_id) - 1)
                  where ticket_no = %s""", (i[0],))
    sleep(2)
    conn.commit()
```

(установили лимит в получении 100 первых строк таблицы)

Запускаем сценарий 7 раз и убеждаемся в запуске соответствующих процессов:

```
[dba@centos-stream-8 ~]$ python3 script.py & python3 script.py &
[1] 2831
[2] 2832
[3] 2833
[4] 2834
[5] 2835
[6] 2836
[7] 2837
[dba@centos-stream-8 ~]$ ps
  PID TTY      TIME CMD
 2220 pts/2    00:00:00 bash
 2831 pts/2    00:00:00 python3
 2832 pts/2    00:00:00 python3
 2833 pts/2    00:00:00 python3
 2834 pts/2    00:00:00 python3
 2835 pts/2    00:00:00 python3
 2836 pts/2    00:00:00 python3
 2837 pts/2    00:00:00 python3
 2845 pts/2    00:00:00 ps
[dba@centos-stream-8 ~]$
```

4. Во время выполнения получили информацию о блокировках: наименование таблицы, номер страницы, номер кортежа, название режима блокировки, номер процессса:

```
demo=# SELECT
demo-#     relname AS table_name,
demo-#     page,
demo-#     tuple,
demo-#     mode AS lock_mode,
demo-#     pid AS process_id
demo-# FROM
demo-#     pg_locks l
demo-# JOIN
demo-#     pg_stat_all_tables t ON l.relation = t.relid;
table_name | page | tuple | lock_mode | process_id
-----+-----+-----+-----+-----+
pg_namespace |    |    | AccessShareLock | 2900
pg_index |    |    | AccessShareLock | 2900
pg_class |    |    | AccessShareLock | 2900
tickets |    |    | RowExclusiveLock | 2844
tickets |    |    | RowExclusiveLock | 2840
tickets |    |    | RowExclusiveLock | 2839
tickets |    |    | RowExclusiveLock | 2842
tickets |    |    | RowExclusiveLock | 2843
tickets |    |    | RowExclusiveLock | 2838
tickets |    |    | RowExclusiveLock | 2841
(10 rows)
```

Сравнили с информацией, выводимой в панели Dashboard pgAdmin:

PID	Lock type	Target relation	Page	Tuple	vXID (target)	XID (target)	Class	Object ID	vXID (owner)	Mode	Granted?
2572	relation	pg_locks							8/310	AccessShareLock	true
2838	relation	bookings.tickets_pkey							9/230	RowExclusiveLock	true
2838	relation	bookings.tickets							9/230	RowExclusiveLock	true
2839	relation	bookings.tickets_pkey							10/221	RowExclusiveLock	true
2839	relation	bookings.tickets							10/221	RowExclusiveLock	true
2840	relation	bookings.tickets							11/209	RowExclusiveLock	true
2840	relation	bookings.tickets_pkey							11/209	RowExclusiveLock	true
2841	relation	bookings.tickets_pkey							12/114	RowExclusiveLock	true
2841	relation	bookings.tickets							12/114	RowExclusiveLock	true
2842	relation	bookings.tickets_pkey							13/120	RowExclusiveLock	true
2842	relation	bookings.tickets							13/120	RowExclusiveLock	true
2843	relation	bookings.tickets							14/9	RowExclusiveLock	true
2843	relation	bookings.tickets_pkey							14/9	RowExclusiveLock	true
2844	relation	bookings.tickets_pkey							15/14	RowExclusiveLock	true
2844	relation	bookings.tickets							15/14	RowExclusiveLock	true

5. После окончания выполнения получили список сегментов WAL:

```
demo=# select * from pg_ls_waldir();
          name           |   size   |      modification
-----+-----+-----
 0000000100000000000000013 | 16777216 | 2024-02-29 01:14:15+07
 0000000100000000000000014 | 16777216 | 2024-02-13 05:20:29+07
 0000000100000000000000015 | 16777216 | 2024-02-13 05:20:30+07
 0000000100000000000000016 | 16777216 | 2024-02-13 05:20:30+07
 0000000100000000000000017 | 16777216 | 2024-02-13 05:20:31+07
 0000000100000000000000018 | 16777216 | 2024-02-13 05:20:32+07
 0000000100000000000000019 | 16777216 | 2024-02-13 05:20:33+07
 000000010000000000000001A | 16777216 | 2024-02-13 05:20:34+07
 000000010000000000000001B | 16777216 | 2024-02-13 05:20:35+07
 000000010000000000000001C | 16777216 | 2024-02-13 05:20:36+07
 000000010000000000000001D | 16777216 | 2024-02-13 05:20:37+07
 000000010000000000000001E | 16777216 | 2024-02-13 05:20:39+07
 000000010000000000000001F | 16777216 | 2024-02-13 05:20:39+07
 0000000100000000000000020 | 16777216 | 2024-02-13 05:20:41+07
 0000000100000000000000021 | 16777216 | 2024-02-13 05:21:34+07
 0000000100000000000000022 | 16777216 | 2024-02-13 05:21:35+07
 0000000100000000000000023 | 16777216 | 2024-02-13 05:21:35+07
 0000000100000000000000024 | 16777216 | 2024-02-16 00:52:41+07
(18 rows)
```

Текущий идентификатор транзакции и текущий LSN:

```
demo=# select * from txid_current();
txid_current
-----
       6613
(1 row)
```

```
demo=# select * from pg_current_wal_lsn();
 pg_current_wal_lsn
-----
 0/13C1E228
(1 row)
```

Идентификатор последней контрольной точки:

```
demo=# select * from pg_control_checkpoint();
 checkpoint_lsn | redo_lsn | redo_wal_file | timeline_id | prev_timeline_id | full_page_writes | next_xid | next_oid | next_multixact_id | next_multi_offset | oldest_xid | olde
st_xid_dbid | oldest_active_xid | oldest_multi_xid | oldest_dbid | oldest_commit_ts_xid | newest_commit_ts_xid | checkpoint_time
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
0/13C1E118 | 0/13C1E0E0 | 0000001000000000000013 | 1 | 1 | 0 | 0:6613 | 16541 | 1 | 0 | 722 |
(1 row)
demo=#
```

Мы выполнили 100 операций update каждым из 7 потоков, то есть 700 update. Сравнивая идентификаторы транзакций до и после, убедились, что было выполнено 700 транзакций.

Изменили значения конфигурационных параметров:

```
demo=# alter system set wal_level = logical;
ALTER SYSTEM
demo=# alter system set wal_compression=on;
ALTER SYSTEM
demo=#
```

После перезапуска, убедились, что значения параметров изменились:

```
[dba@centos-stream-8 ~]$ systemctl restart postgresql-15
===== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units =====
Authentication is required to restart 'postgresql-15.service'.
Authenticating as: dba
Password:
===== AUTHENTICATION COMPLETE =====
[dba@centos-stream-8 ~]$
```

```
demo=# show wal_level;
 wal_level
-----
 logical
(1 row)

demo=# show wal_compression;
 wal_compression
-----
 pglz
(1 row)
```

6. Повторили запуск сценариев, и получение информации о сегментах журналов.

Информация о сегментах журналов до выполнения:

```
demo=# select * from pg_ls_waldir();
          name           |   size   |      modification
-----+-----+-----
00000001000000000000000013 | 16777216 | 2024-02-29 01:20:29+07
00000001000000000000000014 | 16777216 | 2024-02-13 05:20:29+07
00000001000000000000000015 | 16777216 | 2024-02-13 05:20:30+07
00000001000000000000000016 | 16777216 | 2024-02-13 05:20:30+07
00000001000000000000000017 | 16777216 | 2024-02-13 05:20:31+07
00000001000000000000000018 | 16777216 | 2024-02-13 05:20:32+07
00000001000000000000000019 | 16777216 | 2024-02-13 05:20:33+07
0000000100000000000000001A | 16777216 | 2024-02-13 05:20:34+07
0000000100000000000000001B | 16777216 | 2024-02-13 05:20:35+07
0000000100000000000000001C | 16777216 | 2024-02-13 05:20:36+07
0000000100000000000000001D | 16777216 | 2024-02-13 05:20:37+07
0000000100000000000000001E | 16777216 | 2024-02-13 05:20:39+07
0000000100000000000000001F | 16777216 | 2024-02-13 05:20:39+07
00000001000000000000000020 | 16777216 | 2024-02-13 05:20:41+07
00000001000000000000000021 | 16777216 | 2024-02-13 05:21:34+07
00000001000000000000000022 | 16777216 | 2024-02-13 05:21:35+07
00000001000000000000000023 | 16777216 | 2024-02-13 05:21:35+07
00000001000000000000000024 | 16777216 | 2024-02-16 00:52:41+07
(18 rows)
```

```
demo=#
```

После выполнения:

```
demo=# select * from pg_ls_waldir();
          name           |   size   |      modification
-----+-----+-----
00000001000000000000000013 | 16777216 | 2024-02-29 01:27:22+07
00000001000000000000000014 | 16777216 | 2024-02-13 05:20:29+07
00000001000000000000000015 | 16777216 | 2024-02-13 05:20:30+07
00000001000000000000000016 | 16777216 | 2024-02-13 05:20:30+07
00000001000000000000000017 | 16777216 | 2024-02-13 05:20:31+07
00000001000000000000000018 | 16777216 | 2024-02-13 05:20:32+07
00000001000000000000000019 | 16777216 | 2024-02-13 05:20:33+07
0000000100000000000000001A | 16777216 | 2024-02-13 05:20:34+07
0000000100000000000000001B | 16777216 | 2024-02-13 05:20:35+07
0000000100000000000000001C | 16777216 | 2024-02-13 05:20:36+07
0000000100000000000000001D | 16777216 | 2024-02-13 05:20:37+07
0000000100000000000000001E | 16777216 | 2024-02-13 05:20:39+07
0000000100000000000000001F | 16777216 | 2024-02-13 05:20:39+07
00000001000000000000000020 | 16777216 | 2024-02-13 05:20:41+07
00000001000000000000000021 | 16777216 | 2024-02-13 05:21:34+07
00000001000000000000000022 | 16777216 | 2024-02-13 05:21:35+07
00000001000000000000000023 | 16777216 | 2024-02-13 05:21:35+07
00000001000000000000000024 | 16777216 | 2024-02-16 00:52:41+07
(18 rows)
```

Также проверили наличие в журнале сервера записей об ожидании снятия блокировки:

```
[root@centos-stream-8 log]# cat postgresql-2024-02-29_011942.log
2024-02-29 01:19:42.235 +07 [3114] LOG:  starting PostgreSQL 15.5 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 8.5.0 20210514 (Red Hat 8.5.0-18), 64-bit
2024-02-29 01:19:42.238 +07 [3114] LOG:  listening on IPv6 address "::1", port 5432
2024-02-29 01:19:42.238 +07 [3114] LOG:  listening on IPv4 address "127.0.0.1", port 5432
2024-02-29 01:19:42.242 +07 [3114] LOG:  listening on Unix socket "/tmp/.PGSQL.5432"
2024-02-29 01:19:42.251 +07 [3118] LOG:  database system was shut down at 2024-02-29 01:19:42 +07
2024-02-29 01:19:42.265 +07 [3114] LOG:  database system is ready to accept connections
2024-02-29 01:24:38.701 +07 [3266] LOG:  process 3266 still waiting for ShareLock on transaction 6615 after 1000.180 ms
2024-02-29 01:24:38.701 +07 [3266] DETAIL: Process holding the lock: 3265. Wait queue: 3266, 3267, 3268, 3269, 3270.
2024-02-29 01:24:38.701 +07 [3266] CONTEXT: while rechecking updated tuple (6146,26) in relation "tickets"
2024-02-29 01:24:38.701 +07 [3266] STATEMENT: update bookings.tickets
    set passenger_id = SUBSTRING((passenger_id || '1'), 1, 20)
    where ticket_no = '0005432000987'
2024-02-29 01:24:38.701 +07 [3267] LOG:  process 3267 still waiting for ShareLock on transaction 6615 after 1000.315 ms
2024-02-29 01:24:38.701 +07 [3267] DETAIL: Process holding the lock: 3265. Wait queue: 3266, 3267, 3268, 3269, 3270.
2024-02-29 01:24:38.701 +07 [3267] CONTEXT: while rechecking updated tuple (6146,26) in relation "tickets"
2024-02-29 01:24:38.701 +07 [3267] STATEMENT: update bookings.tickets
    set passenger_id = SUBSTRING((passenger_id || '1'), 1, 20)
    where ticket_no = '0005432000987'
2024-02-29 01:24:38.701 +07 [3269] LOG:  process 3269 still waiting for ShareLock on transaction 6615 after 1000.166 ms
2024-02-29 01:24:38.701 +07 [3269] DETAIL: Process holding the lock: 3265. Wait queue: 3266, 3267, 3268, 3269, 3270.
2024-02-29 01:24:38.701 +07 [3269] CONTEXT: while rechecking updated tuple (6146,26) in relation "tickets"
2024-02-29 01:24:38.701 +07 [3269] STATEMENT: update bookings.tickets
    set passenger_id = SUBSTRING((passenger_id || '1'), 1, 20)
    where ticket_no = '0005432000987'
2024-02-29 01:24:38.701 +07 [3268] LOG:  process 3268 still waiting for ShareLock on transaction 6615 after 1000.297 ms
2024-02-29 01:24:38.701 +07 [3268] DETAIL: Process holding the lock: 3265. Wait queue: 3266, 3267, 3268, 3269, 3270.
2024-02-29 01:24:38.701 +07 [3268] CONTEXT: while rechecking updated tuple (6146,26) in relation "tickets"
2024-02-29 01:24:38.701 +07 [3268] STATEMENT: update bookings.tickets
    set passenger_id = SUBSTRING((passenger_id || '1'), 1, 20)
```

7. Модифицировали сценарий так, чтобы подтверждение транзакции производилось не после каждого update, а после завершения цикла:

```
import psycopg2
from time import sleep
conn = psycopg2.connect("dbname='demo' user='dba' host='127.0.0.1' password='disidorov2020'")

cur = conn.cursor()
cur.execute("""select a.*
               from bookings.tickets a
               order by a.ticket_no limit 100""")
rows = cur.fetchall()
for i in rows:
    cur.execute("""update bookings.tickets
                  set passenger_id = SUBSTRING((passenger_id || '1'), 1, 20)
                  where ticket_no = %s""", (i[0],))
    sleep(2)
conn.commit()
```

Выполнили одновременно два сценария.

LSN и значение идентификатора транзакции до выполнения:

```
demo=# select * from pg_current_wal_lsn();
 pg_current_wal_lsn
-----
 0/13D456D8
(1 row)

demo=# select * from txid_current();
 txid_current
-----
 7334
(1 row)
```

Значение идентификатора транзакции и LSN после выполнения:

```
demo=# select * from txid_current();
 txid_current
-----
      7337
(1 row)

demo=# select * from pg_current_wal_lsn();
 pg_current_wal_lsn
-----
 0/13D64B98
(1 row)

demo=#
```

Ошибка во время выполнения:

```
[dba@centos-stream-8 ~]$ python3 script.py & python3 script.py &
[1] 2101
[2] 2102
[dba@centos-stream-8 ~]$ Traceback (most recent call last):
  File "script.py", line 13, in <module>
    where ticket_no = %s""" , (i[0],))
psycopg2.extensions.TransactionRollbackError: deadlock detected
DETAIL: Process 2103 waits for shareLock on transaction 7336; blocked by process
2104.
Process 2104 waits for shareLock on transaction 7335; blocked by process 2103.
HINT: See server log for query details.
CONTEXT: while updating tuple (492,69) in relation "tickets"
```

Проверили наличие в журнале сервера записей о превышении deadlock_timeout:

```
2024-02-29 03:38:50.799 +07 [2103] ERROR: deadlock detected
2024-02-29 03:38:50.799 +07 [2103] DETAIL: Process 2103 waits for ShareLock on t
ransaction 7336; blocked by process 2104.
```