Мамонов Антон 2ИСиП-19-1

Блокировки строк

Смоделируйте ситуацию обновления одной и той же строки тремя командами UPDATE в разных сеансах. Изучите возникшие блокировки в представлении pg_locks и убедитесь, что все они понятны.

Первый терминал

```
postgres=# CREATE DATABASE locks_rows;
CREATE DATABASE
postgres=#
postgres=#  \c locks_rows
Вы подключены к базе данных "locks_rows" как пользователь "postgres".
locks_rows=# CREATE TABLE accounts(acc_no integer, amount numeric);
CREATE TABLE
locks_rows=#  INSERT INTO accounts VALUES (1,1000.00),(2,2000.00),(3,3000.00);
INSERT 0 3
locks_rows=# CREATE VIEW locks AS
locks_rows-# SELECT pid,
locks_rows-#        locktype,
locks_rows-#        CASE locktype
locks_rows-#          WHEN 'relation' THEN relation::REGCLASS::text
locks_rows-#          WHEN 'virtualxid' THEN virtualxid::text
locks_rows-#          WHEN 'transactionid' THEN transactionid::text
locks_rows-#          WHEN 'tuple' THEN relation::REGCLASS::text||':'||tuple::text
locks_rows-#        END AS lockid,
locks_rows-#        mode,
locks_rows-#        granted
locks_rows-# FROM pg_locks;
CREATE VIEW
locks_rows=#  BEGIN;
BEGIN
locks_rows=#  SELECT txid_current(), pg_backend_pid();
 txid_current | pg_backend_pid
--------------+----------------
          914 |          10204
(1 строка)


locks_rows=# UPDATE accounts SET amount = amount + 100.00 WHERE acc_no = 1;
UPDATE 1
locks_rows=# SELECT * FROM locks WHERE pid = 10204;
  pid  |   locktype    |  lockid  |       mode       | granted
-------+---------------+----------+------------------+---------
 10204 | relation      | pg_locks | AccessShareLock  | t
 10204 | relation      | locks    | AccessShareLock  | t
 10204 | relation      | accounts | RowExclusiveLock | t
 10204 | virtualxid    | 4/7      | ExclusiveLock    | t
 10204 | transactionid | 914      | ExclusiveLock    | t
(5 строк)


locks_rows=# ROLLBACK;
ROLLBACK
```

Второй терминал

```
postgres=# \c locks_rows
Вы подключены к базе данных "locks_rows" как пользователь "postgres".
locks_rows=#  BEGIN;
BEGIN
locks_rows=# SELECT txid_current(), pg_backend_pid();
 txid_current | pg_backend_pid
--------------+----------------
          915 |           9672
(1 строка)


locks_rows=#  UPDATE accounts SET amount = amount + 100.00 WHERE acc_no = 1;
UPDATE 1
locks_rows=#
locks_rows=#
locks_rows=#
locks_rows=#
locks_rows=#
locks_rows=#
locks_rows=#
locks_rows=#
locks_rows=# SELECT * FROM locks WHERE pid = 9672;
 pid  |   locktype    |  lockid  |       mode       | granted
------+---------------+----------+------------------+---------
 9672 | relation      | pg_locks | AccessShareLock  | t
 9672 | relation      | locks    | AccessShareLock  | t
 9672 | relation      | accounts | RowExclusiveLock | t
 9672 | virtualxid    | 5/2      | ExclusiveLock    | t
 9672 | transactionid | 915      | ExclusiveLock    | t
(5 строк)


locks_rows=# ROLLBACK;
ROLLBACK
```

Третий терминал

```
postgres=# \c locks_rows
Вы подключены к базе данных "locks_rows" как пользователь "postgres".
locks_rows=# BEGIN;
BEGIN
locks_rows=# SELECT txid_current(), pg_backend_pid();
 txid_current | pg_backend_pid
--------------+----------------
          916 |           8404
(1 строка)


locks_rows=# UPDATE accounts SET amount = amount + 100.00 WHERE acc_no = 1;
UPDATE 1
locks_rows=# SELECT * FROM locks WHERE pid = 8404;
 pid  |   locktype    |  lockid  |       mode       | granted
------+---------------+----------+------------------+---------
 8404 | relation      | pg_locks | AccessShareLock  | t
 8404 | relation      | locks    | AccessShareLock  | t
 8404 | relation      | accounts | RowExclusiveLock | t
 8404 | virtualxid    | 6/2      | ExclusiveLock    | t
 8404 | transactionid | 916      | ExclusiveLock    | t
(5 строк)


locks_rows=# SELECT pid, wait_event_type, wait_event, pg_blocking_pids(pid)
locks_rows-# FROM pg_stat_activity
locks_rows-# WHERE backend_type = 'client backend';
  pid  | wait_event_type | wait_event | pg_blocking_pids
-------+-----------------+------------+------------------
 10204 | Client          | ClientRead | {}
  9672 | Client          | ClientRead | {}
  8404 |                 |            | {}
(3 строки)


locks_rows=# ROLLBACK;
ROLLBACK
```

Воспроизведите взаимоблокировку трех транзакций. Можно ли разобраться в ситуации постфактум, изучая журнал сообщений?

```
postgres=# \c locks_rows
Вы подключены к базе данных "locks_rows" как пользователь "postgres".
locks_rows=# BEGIN;
BEGIN
locks_rows=# UPDATE accounts SET amount = amount - 100.00 WHERE acc_no = 1;
UPDATE 1
locks_rows=#  UPDATE accounts SET amount = amount + 100.00 WHERE acc_no = 2;
UPDATE 1
locks_rows=#
locks_rows=#
locks_rows=#
locks_rows=# commit;
COMMIT
```

```
postgres=# \c locks_rows
Вы подключены к базе данных "locks_rows" как пользователь "postgres".
locks_rows=# BEGIN;
BEGIN
locks_rows=# UPDATE accounts SET amount = amount - 100.00 WHERE acc_no = 2;
UPDATE 1
locks_rows=# UPDATE accounts SET amount = amount + 100.00 WHERE acc_no = 3;
UPDATE 1
locks_rows=# commit;
COMMIT
locks_rows=#
```

```
postgres=# \c locks_rows
Вы подключены к базе данных "locks_rows" как пользователь "postgres".
locks_rows=# BEGIN;
BEGIN
locks_rows=# UPDATE accounts SET amount = amount - 100.00 WHERE acc_no = 3;
UPDATE 1
locks_rows=# UPDATE accounts SET amount = amount + 100.00 WHERE acc_no = 1;
ОШИБКА:  обнаружена взаимоблокировка
ПОДРОБНОСТИ:  Процесс 4892 ожидает в режиме ShareLock блокировку "транзакция 920"; заблокирован процессом 1796.
Процесс 1796 ожидает в режиме ShareLock блокировку "транзакция 921"; заблокирован процессом 2776.
Процесс 2776 ожидает в режиме ShareLock блокировку "транзакция 922"; заблокирован процессом 4892.
ПОДСКАЗКА:  Подробности запроса смотрите в протоколе сервера.
КОНТЕКСТ:  при изменении кортежа (0,1) в отношении "accounts"
locks_rows=# commit;
ROLLBACK
```

Могут ли две транзакции, выполняющие единственную команду UPDATE одной и той же таблицы, заблокировать друг друга? Попробуйте воспроизвести такую ситуацию.

```
locks_rows=# BEGIN;
BEGIN
locks_rows=# DECLARE c2 CURSOR FOR
locks_rows-# SELECT * FROM accounts ORDER BY acc_no DESC -- в обратную сторо
ну
locks_rows-# FOR UPDATE;
DECLARE CURSOR
locks_rows=# FETCH c2;
 acc_no | amount
--------+---------
      3 | 3100.00
(1 строка)


locks_rows=# FETCH c2;
 acc_no | amount
--------+---------
      2 | 2000.00
(1 строка)


locks_rows=# COMMIT;
COMMIT
```

```
locks_rows=#
locks_rows=# BEGIN;
BEGIN
locks_rows=# DECLARE c1 CURSOR FOR
locks_rows-# SELECT * FROM accounts ORDER BY acc_no
locks_rows-# FOR UPDATE;
DECLARE CURSOR
locks_rows=# FETCH c1;
 acc_no | amount
--------+---------
      1 | 900.00
(1 строка)
```