

# Unlocking the Postgres Lock Manager

BRUCE MOMJIAN



This talk explores all aspects of locking in Postgres by showing queries and their locks; covered lock types include row, table, shared, exclusive, and advisory lock types.

*<https://momjian.us/presentations>*



*Creative Commons Attribution License*

*Last updated: June 2024*

# No One Likes Locking But it Is Necessary for Proper Database Operation



# Outline

1. Locking Introduction
2. Transaction Identifiers
3. Lock Types
4. Lock Examples

# 1. Locking Introduction



<https://www.flickr.com/photos/54409200@N04/>

# What an Adventure! Xyzzy

Little maze of twisting passages

Little maze of twisty passages

Little twisty maze of passages

Maze of little twisting passages

Maze of little twisty passages

Maze of twisting little passages

Maze of twisty little passages

Twisting little maze of passages

Twisting maze of little passages

Twisty little maze of passages

Twisty maze of little passages

# The Real Postgres Lock Types

ACCESS SHARE

ROW SHARE

ROW EXCLUSIVE

SHARE UPDATE EXCLUSIVE

SHARE

SHARE ROW EXCLUSIVE

EXCLUSIVE

ACCESS EXCLUSIVE

# Share/Exclusive Types

ACCESS **SHARE**

ROW **SHARE**

ROW **EXCLUSIVE**

**SHARE** UPDATE **EXCLUSIVE**

**SHARE**

**SHARE** ROW **EXCLUSIVE**

**EXCLUSIVE**

ACCESS **EXCLUSIVE**

# Row/Access Types

ACCESS SHARE

ROW SHARE

ROW EXCLUSIVE

SHARE UPDATE EXCLUSIVE

SHARE

SHARE ROW EXCLUSIVE

EXCLUSIVE

ACCESS EXCLUSIVE



# MVCC

*Multiversion Concurrency Control (MVCC) allows Postgres to offer high concurrency even during significant database read/write activity. MVCC specifically offers behavior where "readers never block writers, and writers never block readers".*

While Multiversion Concurrency Control (MVCC) reduces locking requirements, it does not eliminate locking.

## 2. Transaction Identifiers



<https://www.flickr.com/photos/grendelkhan/>

## Keep Your Eye on the Red (Text)



<https://www.flickr.com/photos/alltheaces/>

# What Is Our Process Identifier (PID)?

```
SELECT pg_backend_pid();  
pg_backend_pid  
-----  
11306
```

## What Is Our Virtual XID (VXID)

```
SELECT virtualtransaction AS vxid, transactionid::text
FROM    pg_locks
WHERE    pid = pg_backend_pid()
ORDER BY 1, 2
LIMIT 1;
```

```
vxid | transactionid
-----+-----
2/7  |
```

2 is the backend id, and 7 is the virtual transaction id for this backend, i.e., backend id/backend-local xid.

# What Is Our Backend Id?

```
SELECT *  
FROM   pg_stat_get_backend_idset() AS t(id)  
WHERE  pg_stat_get_backend_pid(id) = pg_backend_pid();  
      id
```

----

2

The maximum backend id is set by *max\_connections*.

*Query courtesy of Phil Sorber.*

# The VXID Increments

```
SELECT virtualtransaction AS vxid, transactionid::text
FROM   pg_locks
WHERE  pid = pg_backend_pid()
ORDER BY 1, 2
LIMIT 1;
 vxid | transactionid
-----+-----
 2/10 |
```

```
SELECT virtualtransaction AS vxid, transactionid::text
FROM   pg_locks
WHERE  pid = pg_backend_pid()
ORDER BY 1, 2
LIMIT 1;
 vxid | transactionid
-----+-----
 2/11 |
```

# Getting a Real/External/Non-Virtual XID

```
BEGIN WORK;
```

```
SELECT virtualtransaction AS vxid, transactionid::text
FROM   pg_locks
WHERE  pid = pg_backend_pid()
ORDER BY 1, 2
LIMIT 1;
```

```
vxid | transactionid
-----+-----
2/12 |
```

```
ANALYZE pg_language;
```



## Getting a Real/External/Non-Virtual XID

```
SELECT virtualtransaction AS vxid, transactionid::text
FROM   pg_locks
WHERE  pid = pg_backend_pid()
ORDER BY 1, 2
LIMIT 1;
 vxid | transactionid
```

```
-----+-----
 2/12 | 674
```

```
SELECT txid_current();
 txid_current
```

```
-----
      674
```

```
COMMIT;
```

Transaction identifiers range from 3 to 4 billion ( $2^{32}$ ). Zero(0) is an invalid transaction id, and 1 and 2 are used for setting frozen xids (*committed* and *aborted*).

## Requesting Your XID Assigns One

```
BEGIN WORK;
```

```
SELECT virtualtransaction AS vxid, transactionid::text
FROM   pg_locks
WHERE  pid = pg_backend_pid()
ORDER BY 1, 2
LIMIT 1;
```

```
 vxid | transactionid
-----+-----
 2/13 |
```

*-- this will assign a non-virtual xid if not already assigned*

```
SELECT txid_current();
 txid_current
-----
```

```
675
```

## Requesting Your XID Assigns One

```
SELECT virtualtransaction AS vxid, transactionid::text
FROM    pg_locks
WHERE    pid = pg_backend_pid()
ORDER BY 1, 2
LIMIT 1;
 vxid | transactionid
-----+-----
 2/13 | 675

COMMIT;
```

### 3. Lock Types



<https://www.flickr.com/photos/proimos/>

## Setup: Create View *lockview*

```
-- cannot be a temporary view because other sessions must see it
CREATE VIEW lockview AS
SELECT  pid, virtualtransaction AS vxid, locktype AS lock_type,
        mode AS lock_mode, granted,
        CASE
            WHEN virtualxid IS NOT NULL AND transactionid IS NOT NULL
            THEN    virtualxid || ' ' || transactionid
            WHEN virtualxid::text IS NOT NULL
            THEN    virtualxid
            ELSE    transactionid::text
        END AS xid_lock, relname,
        page, tuple, classid, objid, objsubid
FROM    pg_locks LEFT OUTER JOIN pg_class ON (pg_locks.relation = pg_class.oid)
WHERE   -- do not show our view's locks
        pid != pg_backend_pid() AND
        -- no need to show self-vxid locks
        virtualtransaction IS DISTINCT FROM virtualxid
-- granted is ordered earlier
ORDER BY 1, 2, 5 DESC, 6, 3, 4, 7;
```

## Create View *lockview1*

```
CREATE VIEW lockview1 AS
SELECT  pid, vxid, lock_type, lock_mode,
        granted, xid_lock, relname
FROM    lockview
-- granted is ordered earlier
ORDER BY 1, 2, 5 DESC, 6, 3, 4, 7;
```

## Create View *lockview2*

```
CREATE VIEW lockview2 AS
SELECT  pid, vxid, lock_type, page,
        tuple, classid, objid, objsubid
FROM    lockview
-- granted is first
-- add non-display columns to match ordering of lockview
ORDER BY 1, 2, granted DESC, vxid, xid_lock::text, 3, 4, 5, 6, 7, 8;
```

## Create and Populate Table *lockdemo*

```
CREATE TABLE lockdemo (col int);
```

```
INSERT INTO lockdemo VALUES (1);
```



# Explicit ACCESS SHARE Locking

```
BEGIN WORK;
```

```
LOCK TABLE lockdemo IN ACCESS SHARE MODE;
```

```
-- force future psql commands to use the current database
```

```
\setenv PGDATABASE :DBNAME
```

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11306	2/23	relation	AccessShareLock	t		lockdemo

# Explicit ACCESS SHARE Locking

```
\! psql -e -c 'SELECT * FROM lockview2;' | sed 's/^/\t/g'
SELECT * FROM lockview2;
  pid | vxid | lock_type | page | tuple | classid | objid | objsubid
-----+-----+-----+-----+-----+-----+-----+-----
 11306 | 2/23 | relation |      |      |         |      |
```

COMMIT;

Future slides will only show *lockview2* if it contains useful information.

# Implicit ACCESS SHARE Locking

```
BEGIN WORK;
```

```
SELECT * FROM lockdemo;
```

```
col
```

```
-----
```

```
1
```

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
-----	-----	-----	-----	-----	-----	-----
11306	2/24	relation	AccessShareLock	t		lockdemo

```
COMMIT;
```

# Multi-Table ACCESS SHARE Locking

```
BEGIN WORK;
```

```
SELECT  pg_class.oid  
FROM    pg_class JOIN pg_namespace ON (relnamespace = pg_namespace.oid)  
        JOIN pg_attribute ON (pg_class.oid = pg_attribute.attrelid)
```

```
LIMIT 1;
```

```
oid
```

```
-----
```

```
112
```

# Multi-Table ACCESS SHARE Locking

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11306	2/25	relation	AccessShareLock	t		pg_attribute
11306	2/25	relation	AccessShareLock	t		pg_attribute_relid_attnam_index
11306	2/25	relation	AccessShareLock	t		pg_attribute_relid_attnum_index
11306	2/25	relation	AccessShareLock	t		pg_class
11306	2/25	relation	AccessShareLock	t		pg_class_oid_index
11306	2/25	relation	AccessShareLock	t		pg_class_relname_nsp_index
11306	2/25	relation	AccessShareLock	t		pg_namespace
11306	2/25	relation	AccessShareLock	t		pg_namespace_nspname_index
11306	2/25	relation	AccessShareLock	t		pg_namespace_oid_index

```
COMMIT;
```

# Explicit ROW SHARE Locking

```
BEGIN WORK;
```

```
LOCK TABLE lockdemo IN ROW SHARE MODE;
```

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11306	2/26	relation	RowShareLock	t		lockdemo

```
COMMIT;
```

# Implicit ROW SHARE Locking

```
BEGIN WORK;
```

```
SELECT * FROM lockdemo FOR SHARE;
```

```
col
```

```
-----
```

```
1
```

```
SELECT txid_current();
```

```
txid_current
```

```
-----
```

```
681
```

# Implicit ROW SHARE Locking

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11306	2/27	transactionid	ExclusiveLock	t	681	
11306	2/27	relation	RowShareLock	t		lockdemo

```
COMMIT;
```

When a transaction is assigned a transaction id, it **self-locks** its transaction id so other sessions can wait on that lock for the transaction to finish.



# Explicit ROW EXCLUSIVE Locking

```
BEGIN WORK;
```

```
LOCK TABLE lockdemo IN ROW EXCLUSIVE MODE;
```

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11306	2/28	relation	RowExclusiveLock	t		lockdemo

```
COMMIT;
```

# Implicit ROW EXCLUSIVE Locking

```
BEGIN WORK;
```

```
DELETE FROM lockdemo;
```

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11306	2/29	transactionid	ExclusiveLock	t	682	
11306	2/29	relation	RowExclusiveLock	t		lockdemo

```
ROLLBACK WORK;
```

# Explicit SHARE UPDATE EXCLUSIVE Locking

```
BEGIN WORK;
```

```
LOCK TABLE lockdemo IN SHARE UPDATE EXCLUSIVE MODE;
```

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11306	2/30	relation	ShareUpdateExclusiveLock	t		lockdemo

```
COMMIT;
```

# Implicit SHARE UPDATE EXCLUSIVE Locking

```
BEGIN WORK;
```

```
ANALYZE lockdemo;
```

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11306	2/31	transactionid	ExclusiveLock	t	683	
11306	2/31	relation	ShareUpdateExclusiveLock	t		lockdemo

```
ROLLBACK WORK;
```

# Explicit SHARE Locking

```
BEGIN WORK;
```

```
LOCK TABLE lockdemo IN SHARE MODE;
```

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11306	2/32	relation	ShareLock	t		lockdemo

```
COMMIT;
```

# Implicit SHARE Locking

```
BEGIN WORK;
```

```
CREATE UNIQUE INDEX i_lockdemo on lockdemo(col);
```

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11306	2/33	transactionid	ExclusiveLock	t	684	
11306	2/33	relation	AccessExclusiveLock	t		
11306	2/33	relation	AccessShareLock	t		lockdemo
11306	2/33	relation	ShareLock	t		lockdemo

```
COMMIT;
```

# Explicit SHARE ROW EXCLUSIVE Locking

```
BEGIN WORK;
```

```
LOCK TABLE lockdemo IN SHARE ROW EXCLUSIVE MODE;
```

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11306	2/34	relation	ShareRowExclusiveLock	t		lockdemo

```
COMMIT;
```

# Implicit SHARE ROW EXCLUSIVE Locking

```
BEGIN WORK;
```

```
CREATE TRIGGER lockdemo_trigger
```

```
BEFORE UPDATE ON lockdemo
```

```
FOR EACH ROW EXECUTE FUNCTION suppress_redundant_updates_trigger();
```

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11306	2/35	transactionid	ExclusiveLock	t	685	
11306	2/35	relation	ShareRowExclusiveLock	t		lockdemo

```
ROLLBACK WORK;
```



# Explicit EXCLUSIVE Locking

```
BEGIN WORK;
```

```
LOCK TABLE lockdemo IN EXCLUSIVE MODE;
```

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11306	2/36	relation	ExclusiveLock	t		lockdemo

```
COMMIT;
```

This lock mode is not automatically used by any Postgres SQL commands.

# Explicit ACCESS EXCLUSIVE Locking

```
BEGIN WORK;
```

```
LOCK TABLE lockdemo IN ACCESS EXCLUSIVE MODE;
```

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11306	2/37	relation	AccessExclusiveLock	t		lockdemo

```
COMMIT;
```

ACCESS EXCLUSIVE is the default mode for the LOCK command.

# Implicit ACCESS EXCLUSIVE Locking

```
BEGIN WORK;
```

```
CLUSTER lockdemo USING i_lockdemo;
```

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11306	2/38	transactionid	ExclusiveLock	t	686	
11306	2/38	object	AccessExclusiveLock	t		
11306	2/38	object	AccessExclusiveLock	t		
11306	2/38	relation	AccessExclusiveLock	t		i_lockdemo
11306	2/38	relation	AccessExclusiveLock	t		lockdemo
11306	2/38	relation	AccessExclusiveLock	t		
11306	2/38	relation	AccessShareLock	t		i_lockdemo
11306	2/38	relation	ShareLock	t		lockdemo

# Implicit ACCESS EXCLUSIVE Locking

```
\! psql -e -c 'SELECT * FROM lockview2;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview2;
```

pid	vxid	lock_type	page	tuple	classid	objid	objsubid
11306	2/38	transactionid					
11306	2/38	object			1247	16409	0
11306	2/38	object			1247	16410	0
11306	2/38	relation					
11306	2/38	relation					
11306	2/38	relation					
11306	2/38	relation					
11306	2/38	relation					

```
COMMIT;
```

1247 is the *pg\_class* entry for *pg\_type*. 16409 and 16410 are used as temporary file names.

## 4. Lock Examples



*Ponte Milvio*

<https://www.flickr.com/photos/pricey/>

# Row Locks Are Not Visible in *pg\_locks*

```
DELETE FROM lockdemo;
```

```
BEGIN WORK;
```

```
INSERT INTO lockdemo VALUES (1);
```

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11306	2/40	transactionid	ExclusiveLock	t	688	
11306	2/40	relation	RowExclusiveLock	t		lockdemo

## Two Rows Are Similarly Invisible

```
INSERT INTO lockdemo VALUES (2), (3);
```

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11306	2/40	transactionid	ExclusiveLock	t	688	
11306	2/40	relation	RowExclusiveLock	t		lockdemo

```
COMMIT;
```

# Update Also Causes an Index Lock

```
BEGIN WORK;
```

```
UPDATE lockdemo SET col = 1 WHERE col = 1;
```

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11306	2/41	transactionid	ExclusiveLock	t	689	
11306	2/41	relation	RowExclusiveLock	t		i_lockdemo
11306	2/41	relation	RowExclusiveLock	t		lockdemo



# Two Row Updates Are Similar

```
UPDATE lockdemo SET col = 2 WHERE col = 2;
```

```
UPDATE lockdemo SET col = 3 WHERE col = 3;
```

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11306	2/41	transactionid	ExclusiveLock	t	689	
11306	2/41	relation	RowExclusiveLock	t		i_lockdemo
11306	2/41	relation	RowExclusiveLock	t		lockdemo

```
COMMIT;
```

# Delete of One Row Is Similar

```
BEGIN WORK;
```

```
DELETE FROM lockdemo WHERE col = 1;
```

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11306	2/42	transactionid	ExclusiveLock	t	690	
11306	2/42	relation	RowExclusiveLock	t		i_lockdemo
11306	2/42	relation	RowExclusiveLock	t		lockdemo

# Delete of Two Rows Is Similar

```
DELETE FROM lockdemo;
```

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11306	2/42	transactionid	ExclusiveLock	t	690	
11306	2/42	relation	RowExclusiveLock	t		i_lockdemo
11306	2/42	relation	RowExclusiveLock	t		lockdemo

```
ROLLBACK WORK;
```

# Explicit Row Locks Are Similar

```
BEGIN WORK;
```

```
SELECT * FROM lockdemo WHERE col = 1 FOR UPDATE;
```

```
col
```

```
-----
```

```
1
```

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11306	2/43	transactionid	ExclusiveLock	t	691	
11306	2/43	relation	RowShareLock	t		i_lockdemo
11306	2/43	relation	RowShareLock	t		lockdemo

# Three Explicit Row Locks Are Similar

```
SELECT * FROM lockdemo FOR UPDATE;
```

```
col
```

```
-----
```

```
1
```

```
2
```

```
3
```

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11306	2/43	transactionid	ExclusiveLock	t	691	
11306	2/43	relation	RowShareLock	t		i_lockdemo
11306	2/43	relation	RowShareLock	t		lockdemo

```
COMMIT;
```

# Explicit Shared Row Locks Are Similar

```
BEGIN WORK;
```

```
SELECT * FROM lockdemo WHERE col = 1 FOR SHARE;
```

```
col
```

```
-----
```

```
1
```

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11306	2/44	transactionid	ExclusiveLock	t	692	
11306	2/44	relation	RowShareLock	t		i_lockdemo
11306	2/44	relation	RowShareLock	t		lockdemo

# Three Explicit Shared Row Locks Are Similar

```
SELECT * FROM lockdemo FOR SHARE;
```

```
col
```

```
-----
```

```
1
```

```
2
```

```
3
```

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11306	2/44	transactionid	ExclusiveLock	t	692	
11306	2/44	relation	RowShareLock	t		i_lockdemo
11306	2/44	relation	RowShareLock	t		lockdemo

```
COMMIT;
```

## Restore Table *Lockdemo*

```
DELETE FROM lockdemo;
```

```
INSERT INTO lockdemo VALUES (1);
```



# UPDATE Is Not Blocked by SELECT

```
BEGIN WORK;
```

```
SELECT ctid, xmin, * FROM lockdemo;
```

ctid	xmin	col
(0,8)	694	1

# UPDATE Is Not Blocked by SELECT

```
SELECT pg_backend_pid();
```

```
pg_backend_pid
```

```
-----
```

```
11306
```

```
SELECT txid_current();
```

```
txid_current
```

```
-----
```

```
695
```

# UPDATE Is Not Blocked by SELECT

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
SELECT * FROM lockview1;
  pid | vxid | lock_type | lock_mode | granted | xid_lock | relname
-----+-----+-----+-----+-----+-----+-----
 11306 | 2/47 | transactionid | ExclusiveLock | t       | 695      |
 11306 | 2/47 | relation      | AccessShareLock | t       |          | i_lockdemo
 11306 | 2/47 | relation      | AccessShareLock | t       |          | lockdemo
```

```
\! psql -e -c 'UPDATE lockdemo SET col = 2; \
SELECT pg_sleep(0.200); \
SELECT ctid, xmin, * FROM lockdemo;' | \
sed 's/^/\t/g' &
```

Backslashes are used to illustrate continued lines, though *psql* does not support the use of backslashes in this way; \! commands must be on a single line.

# UPDATE Is Not Blocked by SELECT

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11306	2/47	transactionid	ExclusiveLock	t	695	
11306	2/47	relation	AccessShareLock	t		i_lockdemo
11306	2/47	relation	AccessShareLock	t		lockdemo
11557	3/110	transactionid	ExclusiveLock	t	696	
11557	3/110	relation	RowExclusiveLock	t		i_lockdemo
11557	3/110	relation	RowExclusiveLock	t		lockdemo

ctid	xmin	col
(0,9)	696	2

```
COMMIT WORK;
```

## Restore Table *Lockdemo*

```
DELETE FROM lockdemo;
```

```
INSERT INTO lockdemo VALUES (1);
```

## Two Concurrent Updates Show Locking

```
BEGIN WORK;
```

```
SELECT ctid, xmin, * FROM lockdemo;
```

ctid	xmin	col
(0,10)	698	1

```
UPDATE lockdemo SET col = 2;
```

## Two Concurrent Updates Show Locking

```
SELECT ctid, xmin, * FROM lockdemo;
```

ctid	xmin	col
(0,11)	699	2

```
SELECT pg_backend_pid();
```

pg_backend_pid
11306

```
SELECT txid_current();
```

txid_current
699

# Two Concurrent Updates Show Locking

```
\! psql -e -c 'BEGIN WORK; UPDATE lockdemo SET col = 3; \
SELECT pg_sleep(0.200); \
COMMIT;' | \
sed 's/^/\t/g' &
```

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11306	2/51	transactionid	ExclusiveLock	t	699	
11306	2/51	relation	AccessShareLock	t		i_lockdemo
11306	2/51	relation	AccessShareLock	t		lockdemo
11306	2/51	relation	RowExclusiveLock	t		i_lockdemo
11306	2/51	relation	RowExclusiveLock	t		lockdemo
11575	3/112	transactionid	ExclusiveLock	t	700	
11575	3/112	relation	RowExclusiveLock	t		i_lockdemo
11575	3/112	relation	RowExclusiveLock	t		lockdemo
11575	3/112	tuple	ExclusiveLock	t		lockdemo
11575	3/112	transactionid	ShareLock	f	699	



## Two Concurrent Updates Show Locking

```
\! psql -e -c 'SELECT * FROM lockview2;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview2;
```

pid	vxid	lock_type	page	tuple	classid	objid	objsubid
11306	2/51	transactionid					
11306	2/51	relation					
11306	2/51	relation					
11306	2/51	relation					
11306	2/51	relation					
11575	3/112	transactionid					
11575	3/112	relation					
11575	3/112	relation					
11575	3/112	tuple	0	10			
11575	3/112	transactionid					

```
COMMIT;
```

## Two Concurrent Updates Show Locking

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11575	3/112	transactionid	ExclusiveLock	t	700	
11575	3/112	relation	RowExclusiveLock	t		i_lockdemo
11575	3/112	relation	RowExclusiveLock	t		lockdemo

# Three Concurrent Updates Show Locking

```
CREATE VIEW lockinfo_hierarchy AS
WITH RECURSIVE lockinfo1 AS (
    SELECT pid, vxid, granted, xid_lock, lock_type, relname, page, tuple
    FROM lockview
    WHERE xid_lock IS NOT NULL AND
           relname IS NULL AND
           granted
    UNION ALL
    SELECT lockview.pid, lockview.vxid, lockview.granted, lockview.xid_lock,
           lockview.lock_type, lockview.relname, lockview.page, lockview.tuple
    FROM lockinfo1 JOIN lockview ON (lockinfo1.xid_lock = lockview.xid_lock)
    WHERE lockview.xid_lock IS NOT NULL AND
           lockview.relname IS NULL AND
           NOT lockview.granted AND
           lockinfo1.granted),
```

# Three Concurrent Updates Show Locking

```
lockinfo2 AS (  
    SELECT pid, vxid, granted, xid_lock, lock_type, relname, page, tuple  
    FROM lockview  
    WHERE lock_type = 'tuple' AND  
           granted  
    UNION ALL  
    SELECT lockview.pid, lockview.vxid, lockview.granted, lockview.xid_lock,  
           lockview.lock_type, lockview.relname, lockview.page, lockview.tuple  
    FROM lockinfo2 JOIN lockview ON (  
        lockinfo2.lock_type = lockview.lock_type AND  
        lockinfo2.relname = lockview.relname AND  
        lockinfo2.page = lockview.page AND  
        lockinfo2.tuple = lockview.tuple)  
    WHERE lockview.lock_type = 'tuple' AND  
           NOT lockview.granted AND  
           lockinfo2.granted  
)  
SELECT * FROM lockinfo1  
UNION ALL  
SELECT * FROM lockinfo2;
```

# Three Concurrent Updates Show Locking

```
BEGIN WORK;
```

```
SELECT ctid, xmin, * FROM lockdemo;
```

ctid	xmin	col
(0,12)	700	3

```
UPDATE lockdemo SET col = 4;
```

## Three Concurrent Updates Show Locking

```
SELECT ctid, xmin, * FROM lockdemo;
```

ctid	xmin	col
(0,13)	702	4

```
SELECT pg_backend_pid();
```

pg_backend_pid
11306

```
SELECT txid_current();
```

txid_current
702

# Three Concurrent Updates Show Locking

```
\! psql -e -c 'BEGIN WORK; UPDATE lockdemo SET col = 5; \
    SELECT pg_sleep(0.200); \
    COMMIT;' | \
    sed 's/^/\t/g' &
\! psql -e -c 'BEGIN WORK; UPDATE lockdemo SET col = 6; \
    SELECT pg_sleep(0.200); \
    COMMIT;' | \
    sed 's/^/\t/g' &
\! psql -e -c 'BEGIN WORK; UPDATE lockdemo SET col = 7; \
    SELECT pg_sleep(0.200); \
    COMMIT;' | \
    sed 's/^/\t/g' &
```

# Three Concurrent Updates Show Locking

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11306	2/54	transactionid	ExclusiveLock	t	702	
11306	2/54	relation	AccessShareLock	t		i_lockdemo
11306	2/54	relation	AccessShareLock	t		lockdemo
11306	2/54	relation	RowExclusiveLock	t		i_lockdemo
11306	2/54	relation	RowExclusiveLock	t		lockdemo
11596	3/114	transactionid	ExclusiveLock	t	703	
11596	3/114	relation	RowExclusiveLock	t		i_lockdemo
11596	3/114	relation	RowExclusiveLock	t		lockdemo
11596	3/114	tuple	ExclusiveLock	t		lockdemo
11596	3/114	transactionid	ShareLock	f	702	
11600	4/14	transactionid	ExclusiveLock	t	704	
11600	4/14	relation	RowExclusiveLock	t		i_lockdemo
11600	4/14	relation	RowExclusiveLock	t		lockdemo
11600	4/14	tuple	ExclusiveLock	f		lockdemo
11604	5/2	transactionid	ExclusiveLock	t	705	
11604	5/2	relation	RowExclusiveLock	t		i_lockdemo
11604	5/2	relation	RowExclusiveLock	t		lockdemo
11604	5/2	tuple	ExclusiveLock	f		lockdemo



# Three Concurrent Updates Show Locking

```
\! psql -e -c 'SELECT * FROM lockview2;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview2;
```

pid	vxid	lock_type	page	tuple	classid	objid	objsubid
11306	2/54	transactionid					
11306	2/54	relation					
11306	2/54	relation					
11306	2/54	relation					
11306	2/54	relation					
11596	3/114	transactionid					
11596	3/114	relation					
11596	3/114	relation					
11596	3/114	tuple	0	12			
11596	3/114	transactionid					
11600	4/14	transactionid					
11600	4/14	relation					
11600	4/14	relation					
11600	4/14	tuple	0	12			
11604	5/2	transactionid					
11604	5/2	relation					
11604	5/2	relation					
11604	5/2	tuple	0	12			

# Three Concurrent Updates Show Locking

```
\! psql -e -c 'SELECT * FROM lockinfo_hierarchy;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockinfo_hierarchy;
```

pid	vxid	granted	xid_lock	lock_type	relname	page	tuple
11306	2/54	t	702	transactionid			
11596	3/114	t	703	transactionid			
11600	4/14	t	704	transactionid			
11604	5/2	t	705	transactionid			
11596	3/114	f	702	transactionid			
11596	3/114	t		tuple	lockdemo	0	12
11600	4/14	f		tuple	lockdemo	0	12
11604	5/2	f		tuple	lockdemo	0	12

# Three Concurrent Updates Show Locking

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11596	3/114	transactionid	ExclusiveLock	t	703	
11596	3/114	relation	AccessShareLock	t		i_lockdemo
11596	3/114	relation	AccessShareLock	t		lockdemo
11596	3/114	relation	RowExclusiveLock	t		i_lockdemo
11596	3/114	relation	RowExclusiveLock	t		lockdemo
11600	4/14	transactionid	ExclusiveLock	t	704	
11600	4/14	relation	RowExclusiveLock	t		i_lockdemo
11600	4/14	relation	RowExclusiveLock	t		lockdemo
11600	4/14	relation	AccessExclusiveLock	t		lockdemo
11600	4/14	transactionid	ShareLock	f	703	
11604	5/2	transactionid	ExclusiveLock	t	705	
11604	5/2	relation	RowExclusiveLock	t		i_lockdemo
11604	5/2	relation	RowExclusiveLock	t		lockdemo
11604	5/2	relation	AccessExclusiveLock	f		lockdemo
11604	5/2	transactionid	ExclusiveLock	t	703	
11604	5/2	transactionid	RowExclusiveLock	t		i_lockdemo
11604	5/2	transactionid	RowExclusiveLock	t		lockdemo
11604	5/2	transactionid	AccessExclusiveLock	f		lockdemo

# Deadlocks

```
DELETE FROM lockdemo;
```

```
INSERT INTO lockdemo VALUES (50), (80);
```

# Deadlocks

```
BEGIN WORK;
```

```
UPDATE lockdemo SET col = 50 WHERE col = 50;
```

```
SELECT pg_backend_pid();  
pg_backend_pid
```

```
-----
```

```
11306
```

```
SELECT txid_current();  
txid_current
```

```
-----
```

```
710
```

# Deadlocks

```
\! psql -e -c 'BEGIN WORK; \  
    UPDATE lockdemo SET col = 81 WHERE col = 80; \  
    UPDATE lockdemo SET col = 51 WHERE col = 50; \  
    COMMIT;' | \  
    sed 's/^\t/g' &  
  
SELECT pg_sleep(0.200);
```

# Deadlocks

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11306	2/61	transactionid	ExclusiveLock	t	710	
11306	2/61	relation	RowExclusiveLock	t		i_lockdemo
11306	2/61	relation	RowExclusiveLock	t		lockdemo
11642	3/116	transactionid	ExclusiveLock	t	711	
11642	3/116	relation	RowExclusiveLock	t		i_lockdemo
11642	3/116	relation	RowExclusiveLock	t		lockdemo
11642	3/116	tuple	ExclusiveLock	t		lockdemo
11642	3/116	transactionid	ShareLock	f	710	

# Deadlocks

```
\! psql -e -c 'SELECT * FROM lockview2;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview2;
```

pid	vxid	lock_type	page	tuple	classid	objid	objsubid
11306	2/61	transactionid					
11306	2/61	relation					
11306	2/61	relation					
11642	3/116	transactionid					
11642	3/116	relation					
11642	3/116	relation					
11642	3/116	tuple	0	18			
11642	3/116	transactionid					



# Deadlocks

```
-- show lockview while waiting for deadlock_timeout
\! psql -e -c 'SELECT pg_sleep(0.200); \
    SELECT * FROM lockview1;' | \
    sed 's/^/\t/g' &
\! psql -e -c 'SELECT pg_sleep(0.400); \
    SELECT * FROM lockview2;' | \
    sed 's/^/\t/g' &

-- the next line hangs waiting for deadlock_timeout
UPDATE lockdemo SET col = 80 WHERE col = 80;
```

# Deadlocks

```
SELECT pg_sleep(0.200); SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11306	2/61	transactionid	ExclusiveLock	t	710	
11306	2/61	relation	RowExclusiveLock	t		i_lockdemo
11306	2/61	relation	RowExclusiveLock	t		lockdemo
11306	2/61	tuple	ExclusiveLock	t		lockdemo
11306	2/61	transactionid	ShareLock	f	711	
11642	3/116	transactionid	ExclusiveLock	t	711	
11642	3/116	relation	RowExclusiveLock	t		i_lockdemo
11642	3/116	relation	RowExclusiveLock	t		lockdemo
11642	3/116	tuple	ExclusiveLock	t		lockdemo
11642	3/116	transactionid	ShareLock	f	710	

# Deadlocks

```
SELECT pg_sleep(0.400); SELECT * FROM lockview2;
```

pid	vxid	lock_type	page	tuple	classid	objid	objsubid
11306	2/61	transactionid					
11306	2/61	relation					
11306	2/61	relation					
11306	2/61	tuple	0	19			
11306	2/61	transactionid					
11642	3/116	transactionid					
11642	3/116	relation					
11642	3/116	relation					
11642	3/116	tuple	0	18			
11642	3/116	transactionid					

# Deadlocks

ERROR: deadlock detected

DETAIL: Process 11306 waits for ShareLock on transaction 711; blocked by process 11642.  
Process 11642 waits for ShareLock on transaction 710; blocked by process 11306.

HINT: See server log for query details.

CONTEXT: while updating tuple (0,18) in relation "lockdemo"

COMMIT;

The Postgres server log will report the queries involved in the deadlock.

# Three-Way Deadlocks

```
DELETE FROM lockdemo;
```

```
INSERT INTO lockdemo VALUES (40), (60), (80);
```

# Three-Way Deadlocks

```
BEGIN WORK;
```

```
UPDATE lockdemo SET col = 40 WHERE col = 40;
```

```
SELECT pg_backend_pid();  
pg_backend_pid  
-----  
11306
```

```
SELECT txid_current();  
txid_current  
-----  
714
```

# Three-Way Deadlocks

```
\! psql -e -c 'BEGIN WORK; \  
    UPDATE lockdemo SET col = 61 WHERE col = 60; \  
    UPDATE lockdemo SET col = 41 WHERE col = 40; \  
    COMMIT;' | \  
    sed 's/^\t/g' &  
\! psql -e -c 'BEGIN WORK; \  
    UPDATE lockdemo SET col = 81 WHERE col = 80; \  
    UPDATE lockdemo SET col = 61 WHERE col = 60; \  
    COMMIT;' | \  
    sed 's/^\t/g' &  
  
SELECT pg_sleep(0.200);
```

## Three-Way Deadlocks

```
\! psql -e -c 'SELECT pg_sleep(0.200); \
    SELECT * FROM lockview1;' | \
    sed 's/^\t/g' &
\! psql -e -c 'SELECT pg_sleep(0.400); \
    SELECT * FROM lockview2;' | \
    sed 's/^\t/g' &
```

*-- the next line hangs waiting for deadlock timeout*

```
UPDATE lockdemo SET col = 80 WHERE col = 80;
```



# Three-Way Deadlocks

```
SELECT pg_sleep(0.200); SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11306	2/65	transactionid	ExclusiveLock	t	714	
11306	2/65	relation	RowExclusiveLock	t		i_lockdemo
11306	2/65	relation	RowExclusiveLock	t		lockdemo
11306	2/65	tuple	AccessExclusiveLock	t		lockdemo
11306	2/65	transactionid	ShareLock	f	716	
11662	3/118	transactionid	ExclusiveLock	t	715	
11662	3/118	relation	RowExclusiveLock	t		i_lockdemo
11662	3/118	relation	RowExclusiveLock	t		lockdemo
11662	3/118	tuple	AccessExclusiveLock	t		lockdemo
11662	3/118	transactionid	ShareLock	f	714	
11666	4/22	transactionid	ExclusiveLock	t	716	
11666	4/22	relation	RowExclusiveLock	t		i_lockdemo
11666	4/22	relation	RowExclusiveLock	t		lockdemo
11666	4/22	tuple	AccessExclusiveLock	t		lockdemo
11666	4/22	transactionid	ShareLock	f	715	

# Three-Way Deadlocks

```
SELECT pg_sleep(0.400); SELECT * FROM lockview2;
```

pid	vxid	lock_type	page	tuple	classid	objid	objsubid
11306	2/65	transactionid					
11306	2/65	relation					
11306	2/65	relation					
11306	2/65	tuple	0	24			
11306	2/65	transactionid					
11662	3/118	transactionid					
11662	3/118	relation					
11662	3/118	relation					
11662	3/118	tuple	0	22			
11662	3/118	transactionid					
11666	4/22	transactionid					
11666	4/22	relation					
11666	4/22	relation					
11666	4/22	tuple	0	23			
11666	4/22	transactionid					

# Three-Way Deadlocks

ERROR: deadlock detected

DETAIL: Process 11662 waits for ShareLock on transaction 714; blocked by process 11306.

Process 11306 waits for ShareLock on transaction 716; blocked by process 11666.

Process 11666 waits for ShareLock on transaction 715; blocked by process 11662.

HINT: See server log for query details.

CONTEXT: while updating tuple (0,22) in relation "lockdemo"

COMMIT;

## Restore Table *Lockdemo*

```
DELETE FROM lockdemo;
```

```
INSERT INTO lockdemo VALUES (1);
```

# Serializable

```
BEGIN WORK;
```

```
SELECT * FROM lockdemo;
```

```
col
```

```
-----
```

```
1
```

# Serializable

```
SELECT pg_backend_pid();  
pg_backend_pid  
-----  
11306
```

```
SELECT txid_current();  
txid_current  
-----  
719
```

# Serializable

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11306	2/70	transactionid	ExclusiveLock	t	719	
11306	2/70	relation	AccessShareLock	t		i_lockdemo
11306	2/70	relation	AccessShareLock	t		lockdemo

```
COMMIT;
```

# Serializable

```
BEGIN WORK;
```

```
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
```

```
SELECT * FROM lockdemo;
```

```
col
```

```
-----
```

```
1
```



# Serializable

```
SELECT pg_backend_pid();  
pg_backend_pid  
-----  
11306
```

```
SELECT txid_current();  
txid_current  
-----  
720
```

# Serializable

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11306	2/71	transactionid	ExclusiveLock	t	720	
11306	2/71	relation	AccessShareLock	t		i_lockdemo
11306	2/71	relation	AccessShareLock	t		lockdemo
11306	2/71	relation	SIReadLock	t		lockdemo

```
COMMIT;
```

# Unique Insert Locking

```
\d lockdemo
```

```
Table "public.lockdemo"
```

```
Column | Type   | Modifiers
```

```
-----+-----+-----  
col     | integer |
```

```
Indexes:
```

```
    "i_lockdemo" UNIQUE, btree (col) CLUSTER
```

# Unique Insert Locking

```
BEGIN WORK;
```

```
INSERT INTO lockdemo VALUES (2);
```

```
SELECT pg_backend_pid();  
pg_backend_pid
```

```
-----  
11306
```

```
SELECT txid_current();  
txid_current
```

```
-----  
721
```

# Unique Insert Locking

```
\! PGOPTIONS='-c statement_timeout=200' \  
psql -e -c 'INSERT INTO lockdemo VALUES (2);' | \  
sed 's/^/\t/g' &  
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
```

SELECT * FROM lockview1;						
pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11306	2/78	transactionid	ExclusiveLock	t	721	lockdemo
11306	2/78	relation	RowExclusiveLock	t		
11696	3/128	transactionid	ExclusiveLock	t	722	i_lockdemo lockdemo
11696	3/128	relation	RowExclusiveLock	t		
11696	3/128	relation	RowExclusiveLock	t		
11696	3/128	transactionid	ShareLock	f	721	

```
SELECT pg_sleep(0.400);  
ERROR: canceling statement due to statement timeout
```

```
ROLLBACK WORK;
```

# Subtransactions

```
BEGIN WORK;
```

```
UPDATE lockdemo SET col = 1;
```

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11306	2/79	transactionid	ExclusiveLock	t	723	
11306	2/79	relation	RowExclusiveLock	t		i_lockdemo
11306	2/79	relation	RowExclusiveLock	t		lockdemo

# Subtransactions

```
SAVEPOINT lockdemo1;
```

```
UPDATE lockdemo SET col = 2;
```

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^\t/g'
```

```
SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11306	2/79	transactionid	ExclusiveLock	t	723	
11306	2/79	transactionid	ExclusiveLock	t	724	
11306	2/79	relation	RowExclusiveLock	t		i_lockdemo
11306	2/79	relation	RowExclusiveLock	t		lockdemo

# Subtransactions

```
ROLLBACK WORK TO SAVEPOINT lockdemo1;
```

```
UPDATE lockdemo SET col = 3;
```

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11306	2/79	transactionid	ExclusiveLock	t	723	
11306	2/79	transactionid	ExclusiveLock	t	725	
11306	2/79	relation	RowExclusiveLock	t		i_lockdemo
11306	2/79	relation	RowExclusiveLock	t		lockdemo

```
COMMIT;
```



# Advisory Locks

```
BEGIN WORK;
```

```
SELECT pg_advisory_lock(col) FROM lockdemo;  
pg_advisory_lock
```

```
\! psql -e -c 'SELECT * FROM lockview1;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview1;
```

pid	vxid	lock_type	lock_mode	granted	xid_lock	relname
11306	2/80	advisory	ExclusiveLock	t		
11306	2/80	relation	AccessShareLock	t		i_lockdemo
11306	2/80	relation	AccessShareLock	t		lockdemo

# Advisory Locks

```
\! psql -e -c 'SELECT * FROM lockview2;' | sed 's/^/\t/g'
```

```
SELECT * FROM lockview2;
```

pid	vxid	lock_type	page	tuple	classid	objid	objsubid
11306	2/80	advisory			0	3	1
11306	2/80	relation					
11306	2/80	relation					

```
SELECT pg_advisory_unlock(col) FROM lockdemo;
```

```
pg_advisory_unlock
```

```
-----  
t
```

```
COMMIT;
```

## Joining *Pg\_locks* and *Pg\_stat\_activity*

```
-- cannot be a temporary view because other sessions must see it
CREATE VIEW lock_stat_view AS
SELECT  pg_stat_activity.pid AS pid,
        query, wait_event, vxid, lock_type,
        lock_mode, granted, xid_lock
FROM    lockview JOIN pg_stat_activity ON (lockview.pid = pg_stat_activity.pid);
```

## Joining *Pg\_locks* and *Pg\_stat\_activity*

```
BEGIN WORK;
```

```
UPDATE lockdemo SET col = 1;
```

```
SELECT pg_backend_pid();  
pg_backend_pid  
-----  
11306
```

```
SELECT txid_current();  
txid_current  
-----  
727
```

# Joining *Pg\_locks* and *Pg\_stat\_activity*

```
\! psql -e -c 'UPDATE lockdemo SET col = 2;' | sed 's/^/\t/g' &
\! psql -e -c 'UPDATE lockdemo SET col = 3;' | sed 's/^/\t/g' &
\! psql -e -c 'SELECT * FROM lock_stat_view;' | sed 's/^/\t/g'
```

pid	query	wait_event	vxid	lock_type	lock_mode	granted	xid_lock
-----+-----+-----+-----+-----+-----+-----+-----							
11306	SELECT txid_current();	ClientRead	2/83	transactionid	ExclusiveLock	t	727
11306	SELECT txid_current();	ClientRead	2/83	relation	RowExclusiveLock	t	
11306	SELECT txid_current();	ClientRead	2/83	relation	RowExclusiveLock	t	
11740	UPDATE lockdemo SET col = 2;	transactionid	3/146	transactionid	ExclusiveLock	t	728
11740	UPDATE lockdemo SET col = 2;	transactionid	3/146	relation	RowExclusiveLock	t	
11740	UPDATE lockdemo SET col = 2;	transactionid	3/146	relation	RowExclusiveLock	t	
11740	UPDATE lockdemo SET col = 2;	transactionid	3/146	tuple	ExclusiveLock	t	
11740	UPDATE lockdemo SET col = 2;	transactionid	3/146	transactionid	ShareLock	f	727
11748	UPDATE lockdemo SET col = 3;	tuple	4/30	transactionid	ExclusiveLock	t	729
11748	UPDATE lockdemo SET col = 3;	tuple	4/30	relation	RowExclusiveLock	t	
11748	UPDATE lockdemo SET col = 3;	tuple	4/30	relation	RowExclusiveLock	t	
11748	UPDATE lockdemo SET col = 3;	tuple	4/30	tuple	ExclusiveLock	f	

```
SELECT pg_blocking_pids(11740);
pg_blocking_pids
```

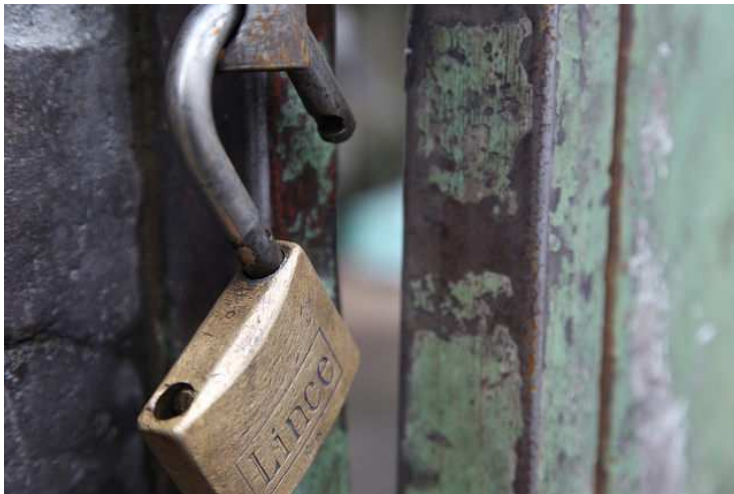
```
-----
{11306}
```

```
SELECT pg_blocking_pids(11748);
pg_blocking_pids
```

```
-----
{11740}
```

```
COMMIT;
```

# Conclusion



<https://momjian.us/presentations>

<https://www.flickr.com/photos/denisgarcia/>