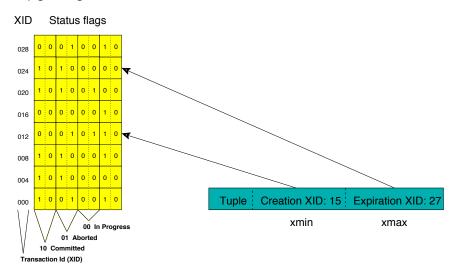
Lock Modes

Mode	Used
Access Share Lock	SELECT
Row Share Lock	SELECT FOR UPDATE
Row Exclusive Lock	INSERT, UPDATE, DELETE
Share Lock	CREATE INDEX
Share Row Exclusive Lock	EXCLUSIVE MODE but allows ROW SHARE LOCK
Exclusive Lock	Blocks ROW SHARE LOCK and SELECTFOR UPDA
Access Exclusive Lock	ALTER TABLE, DROP TABLE, VACUUM, and unquali
	!

Transaction Status

pg_clog



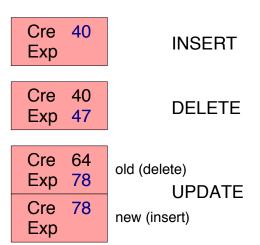
Multi-Version Concurrency Control

- ► Each query sees only transactions completed before it started
- ▶ On query start, PostgreSQL records:
 - the transaction counter
 - all transaction id's that are in-process
- ► In a multi-statement transaction, a transaction's own previous queries are also visible
- ▶ The above assumes the default read committed isolation level

MVCC Tuple Requirements

- ▶ Visible tuples must have a creation transaction id that:
 - is a committed transaction
 - ▶ is less than the transaction counter stored at guery start *and*
 - was not in-process at query start
- ▶ Visible tuples must *also* have an expire transaction id that:
 - ▶ is blank or aborted or
 - ▶ is greater than the transaction counter stored at query start *or*
 - was in-process at query start

MVCC Behavior



UPDATE is effectively a DELETE and an INSERT.

MVCC Examples

Create-Only

Cre 30 Exp Cre 50 Exp

Visible

Invisible

Invisible

Sequential Scan

Snapshot

The highest–numbered committed transaction: 100

Open Transactions: 25, 50, 75

For simplicity, assume all other transactions are committed.

Create & Expire

Exp

Cre 30 Exp 80 Cre 30 Exp 75 Cre 30 Exp 110

Invisible

Visible

Visible

System Tables

