

The Best of Bruce's Postgres Slides

BRUCE MOMJIAN



This talk has the best slides from my 25+ Postgres presentations.

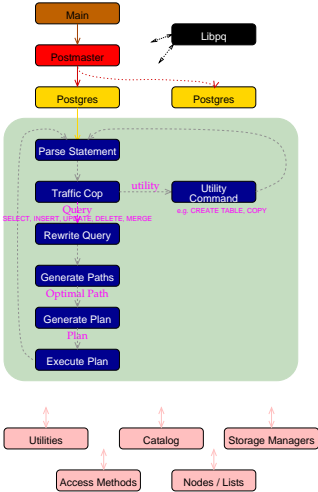
<https://momjian.us/presentations>



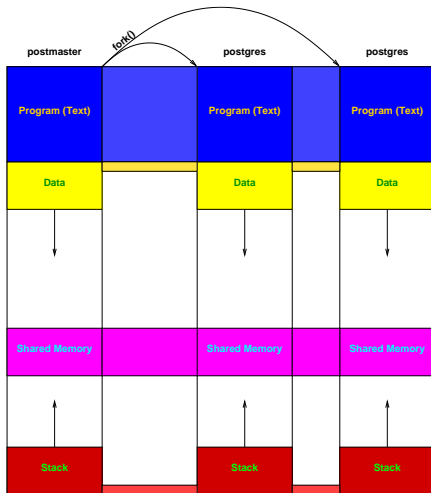
Creative Commons Attribution License

Last updated: October 2024

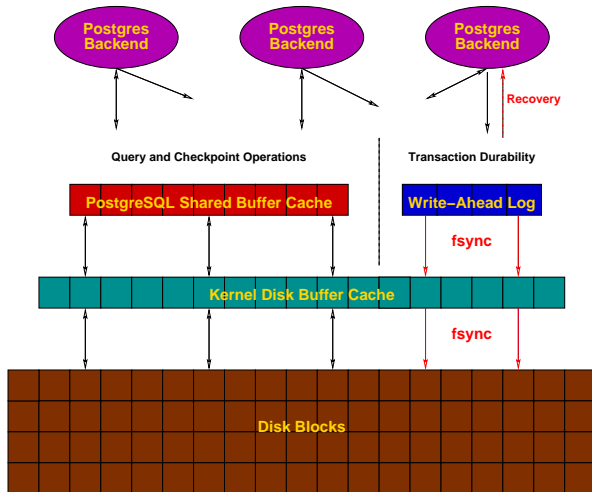
Postgres System Architecture



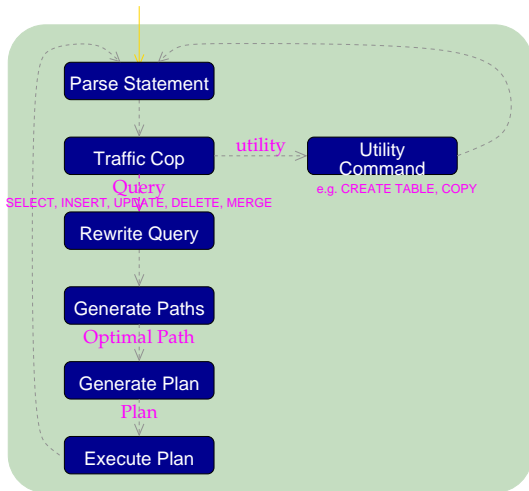
Shared Memory Creation



Shared Buffers and WAL



Backend Flowchart — Magnified



Query Processing

```
FindExec: found "/var/local/postgres/bin/postmaster" using argv[0]
./bin/postmaster: BackendStartup: pid 3320 user postgres db test socket 5
./bin/postmaster child[3320]: starting with (postgres -d99 -F -d99 -v131072 -p test )
FindExec: found "/var/local/postgres/bin/postgres" using argv[0]
DEBUG: connection: host=[local] user=postgres database=test
DEBUG: InitPostgres
DEBUG: StartTransactionCommand
DEBUG: query: SELECT firstname
        FROM friend
        WHERE age = 33;
DEBUG: parse tree: { QUERY :command 1 :utility <> :resultRelation 0 :into <> :isPortal false :isBinary false :isTemp false :hasAggs false :hasSubLinks false :rtable ({ RTE :relname friend :reloid 26912 :subquery <> :alias <> :eref { ATTR :relname friend :attrs ( "firstname" "lastname" "city" "state" "age" ) :inh true :inFromCl true :checkForRead true :checkForWrite false :checkAsUser 0 } :jointree { FROMEXPR :fromlist ({ RANGETBLREF 1 }) :quals { EXPR :typeOid 16 :opType op :oper { OPER :opno 96 :opid 0 :opresul type 16 } :args ({ VAR :varno 1 :varattno 5 :vartype 23 :vartypmod -1 :varlevelsup 0 :varnoold 1 :varoattno 5 } { CONST :consttype 23 :constlen 4 :constbyval true :constisnull false :constvalue 4 [ 33 0 0 0 ] }) :rowMarks () :targetList ({ TARGETENTRY :resdom { RESDOM :resno 1 :restype 1042 :restypmod 19 :resname firstname :reskey 0 :reskeyop 0 :ressortgroupref 0 :resjunk false } :expr { VAR :varno 1 :varattno 1 :vartype 1042 :vartypmod 19 :varlevelsup 0 :varnoold 1 :varoattno 1 } :groupClause <> :havingQual <> :distinctClause <> :sortClause <> :limitOffset <> :limitCount <> :setOperations <> :resultRelations () }
DEBUG: rewritten parse tree:
DEBUG: { QUERY :command 1 :utility <> :resultRelation 0 :into <> :isPortal false :isBinary false :isTemp false :hasAggs false :hasSubLinks false :rtable ({ RTE :relname friend :reloid 26912 :subquery <> :alias <> :eref { ATTR :relname friend :attrs ( "firstname" "lastname" "city" "state" "age" ) :inh true :inFromCl true :checkForRead true :checkForWrite false :checkAsUser 0 } :jointree { FROMEXPR :fromlist ({ RANGETBLREF 1 }) :quals { EXPR :typeOid 16 :opType op :oper { OPER :opno 96 :opid 0 :opresulttype 16 } :args ({ VAR :varno 1 :varattno 5 :vartype 23 :vartypmod -1 :varlevelsup 0 :varnoold 1 :varoattno 5 } { CONST :consttype 23 :constlen 4 :constbyval true :constisnull false :constvalue 4 [ 33 0 0 0 ] }) :rowMarks () :targetList ({ TARGETENTRY :resdom { RESDOM :resno 1 :restype 1042 :restypmod 19 :resname firstname :reskey 0 :reskeyop 0 :ressortgroupref 0 :resjunk false } :expr { VAR :varno 1 :varattno 1 :vartype 1042 :vartypmod 19 :varlevelsup 0 :varnoold 1 :varoattno 1 } :groupClause <> :havingQual <> :distinctClause <> :sortClause <> :limitOffset <> :limitCount <> :setOperations <> :resultRelations () }
DEBUG: plan: { SEQSCAN :startup_cost 0.00 :total_cost 22.50 :rows 10 :width 12 :qptargetlist ({ TARGETENTRY :resdom { RESDOM :resno 1 :restype 1042 :restypmod 19 :resname firstname :reskey 0 :reskeyop 0 :ressortgroupref 0 :resjunk false } :expr { VAR :varno 1 :varattno 1 :vartype 1042 :vartypmod 19 :varlevelsup 0 :varnoold 1 :varoattno 1 } :qpqual ({ EXPR :typeOid 16 :opType op :oper { OPER :opno 96 :opid 65 :opresulttype 16 } :args ({ VAR :varno 1 :varattno 5 :vartype 23 :vartypmod -1 :varlevelsup 0 :varnoold 1 :varoattno 5 } { CONST :consttype 23 :constlen 4 :constbyval true :constisnull false :constvalue 4 [ 33 0 0 0 ] }) :lefttree <> :righttree <> :extprn () :loprn () :initplan <> :nprn 0 :scanreloid 1 }
DEBUG: ProcessQuery
DEBUG: CommitTransactionCommand
DEBUG: proc_exit(0)
DEBUG: shm_exit(0)
DEBUG: exit(0)
./bin/postmaster: reaping dead processes...
./bin/postmaster: CleanupProc: pid 3320 exited with status 0
```

EXPLAIN with Constants of Various Frequencies

l	count	lookup_letter
p	199	Seq Scan on sample (cost=0.00..13.16 rows=199 width=2)
s	9	Seq Scan on sample (cost=0.00..13.16 rows=9 width=2)
c	8	Seq Scan on sample (cost=0.00..13.16 rows=8 width=2)
r	7	Seq Scan on sample (cost=0.00..13.16 rows=7 width=2)
t	5	Bitmap Heap Scan on sample (cost=4.29..12.76 rows=5 width=2)
f	4	Bitmap Heap Scan on sample (cost=4.28..12.74 rows=4 width=2)
v	4	Bitmap Heap Scan on sample (cost=4.28..12.74 rows=4 width=2)
d	4	Bitmap Heap Scan on sample (cost=4.28..12.74 rows=4 width=2)
a	3	Bitmap Heap Scan on sample (cost=4.27..11.38 rows=3 width=2)
_	3	Bitmap Heap Scan on sample (cost=4.27..11.38 rows=3 width=2)
u	3	Bitmap Heap Scan on sample (cost=4.27..11.38 rows=3 width=2)
e	2	Index Scan using i_sample on sample (cost=0.00..8.27 rows=1 width=2)
i	1	Index Scan using i_sample on sample (cost=0.00..8.27 rows=1 width=2)
k	1	Index Scan using i_sample on sample (cost=0.00..8.27 rows=1 width=2)

Explaining the Postgres Query Optimizer

Deadlocks

```
SELECT pg_sleep(0.500); 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	

Unlocking the Postgres Lock Manager

MVCC Behavior

Cre	40
Exp	

INSERT

Cre	40
Exp	47

DELETE

Cre	64
Exp	78

old (delete)

UPDATE

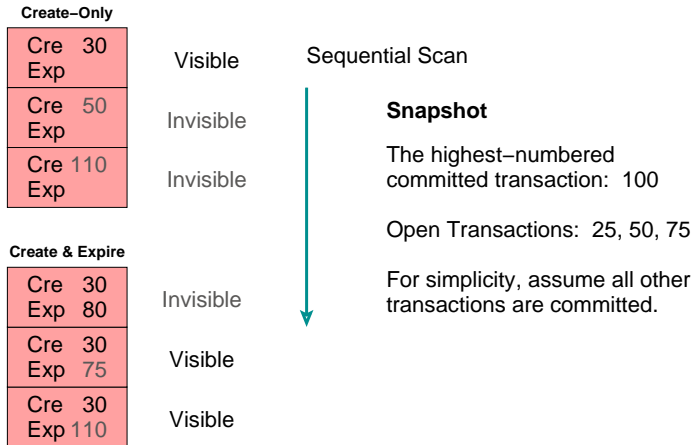
Cre	78
Exp	

new (insert)

UPDATE is effectively a DELETE and an INSERT.

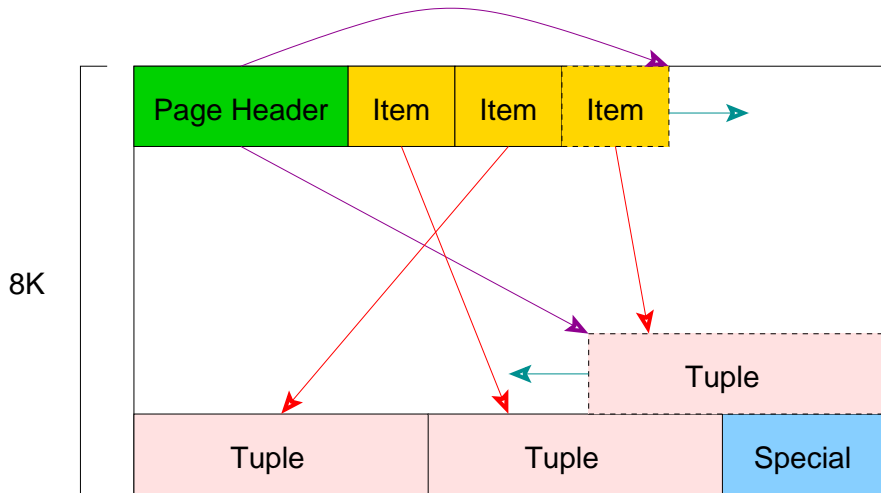
MVCC Unmasked

MVCC Examples

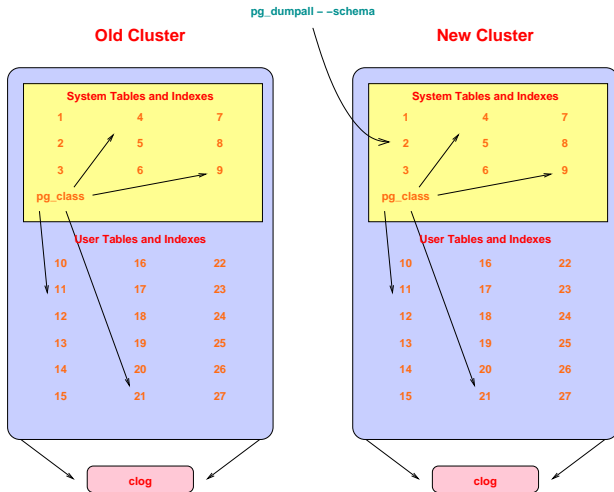


Internally, the creation xid is stored in the system column 'xmin', and expire in 'xmax'.

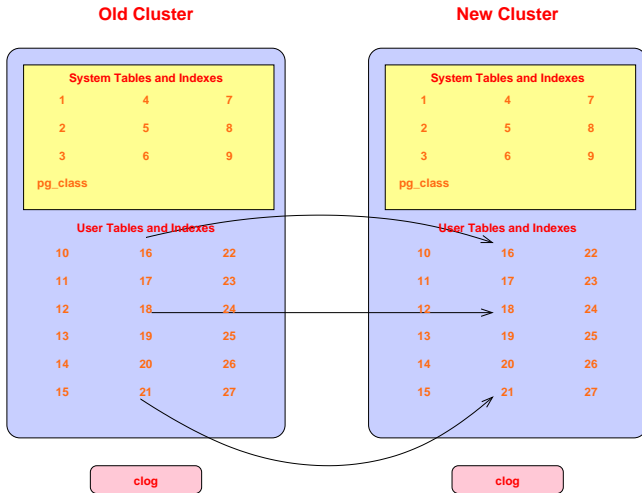
Heap Page Structure



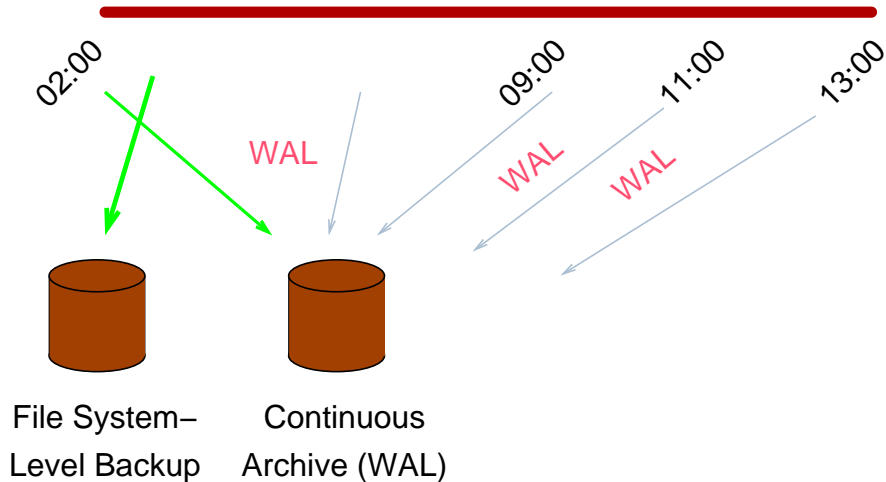
Pg_upgrade: Restore Schema In New Cluster



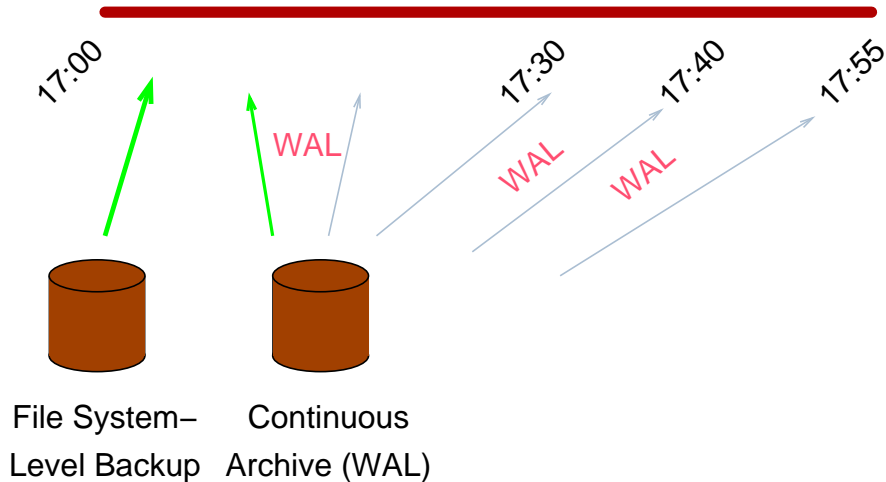
Pg_upgrade: Copy User Heap/Index Files



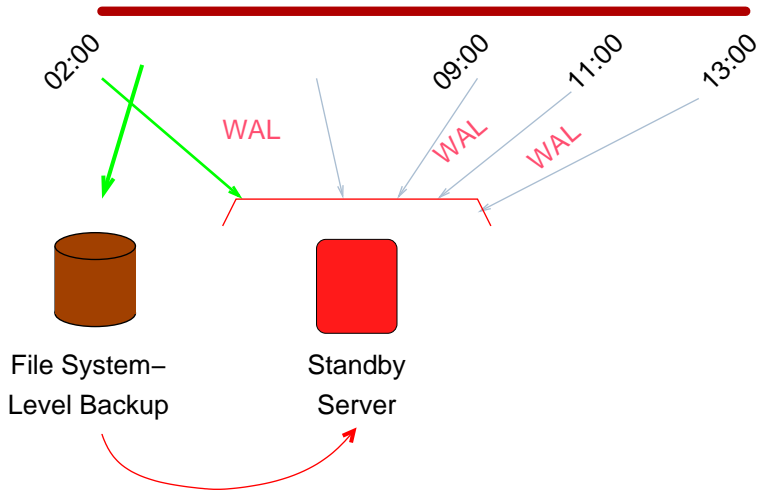
Continuous Archiving



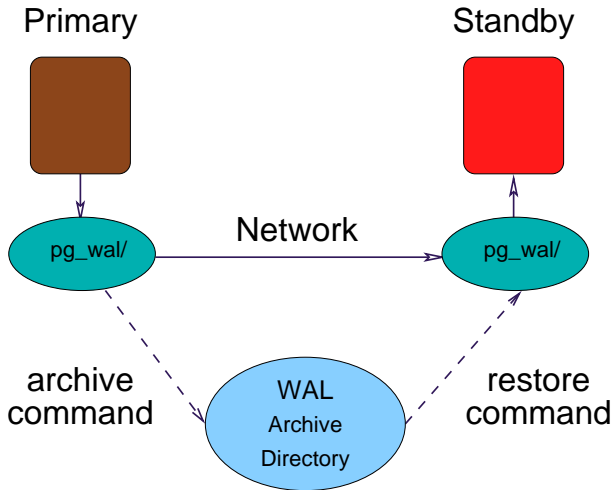
Point-in-Time Recovery



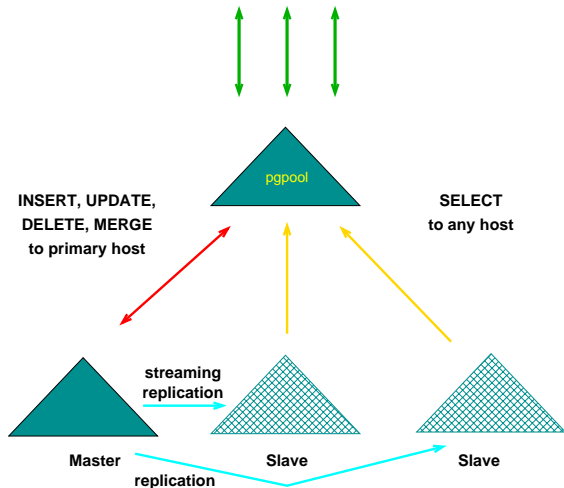
Streaming Replication Setup



Streaming Replication in Operation

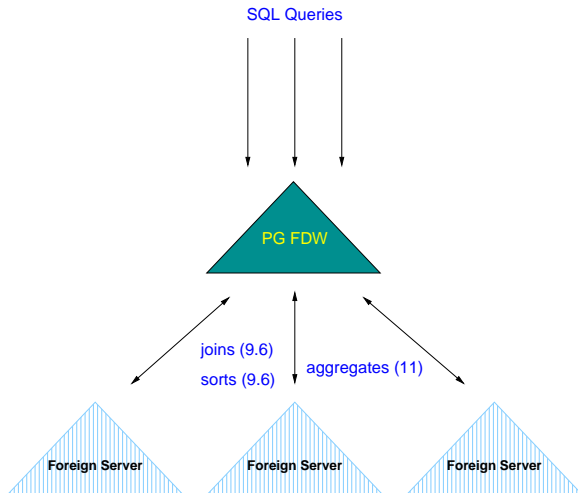


Read Scaling Using Pgpool & Streaming Replication

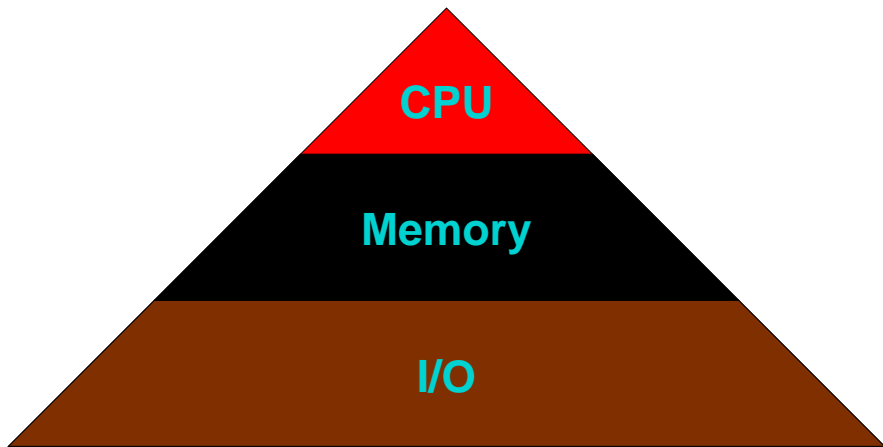


A full copy of the data exists on every node.

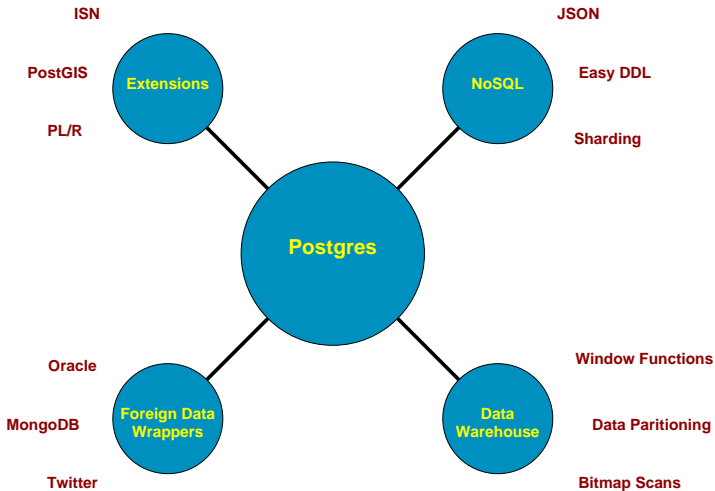
Write Scaling Using FDW-Based Sharding



Database Server Hardware Priorities



Postgres's Central Role



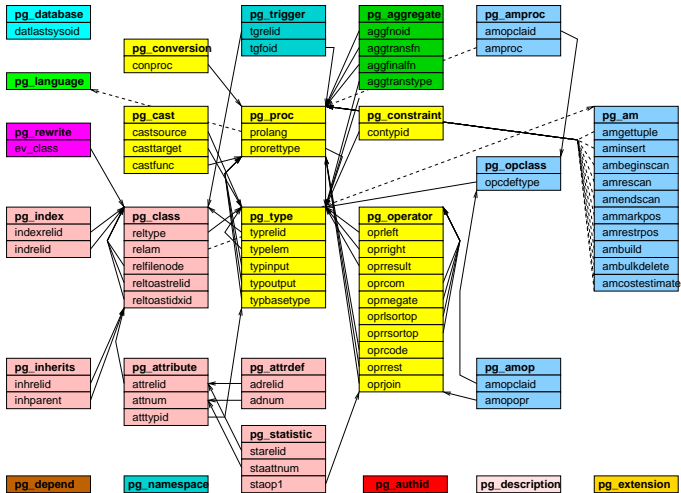
Use of the Contains Operator @>

\do @>

List of operators

Schema	Name	Left arg type	Right arg type	Result type	Description
pg_catalog	@>	aclitem[]	aclitem	boolean	contains
pg_catalog	@>	anyarray	anyarray	boolean	contains
pg_catalog	@>	anyrange	anyelement	boolean	contains
pg_catalog	@>	anyrange	anyrange	boolean	contains
pg_catalog	@>	box	box	boolean	contains
pg_catalog	@>	box	point	boolean	contains
pg_catalog	@>	circle	circle	boolean	contains
pg_catalog	@>	circle	point	boolean	contains
pg_catalog	@>	jsonb	jsonb	boolean	contains
pg_catalog	@>	path	point	boolean	contains
pg_catalog	@>	polygon	point	boolean	contains
pg_catalog	@>	polygon	polygon	boolean	contains
pg_catalog	@>	tsquery	tsquery	boolean	contains

Postgres System Tables



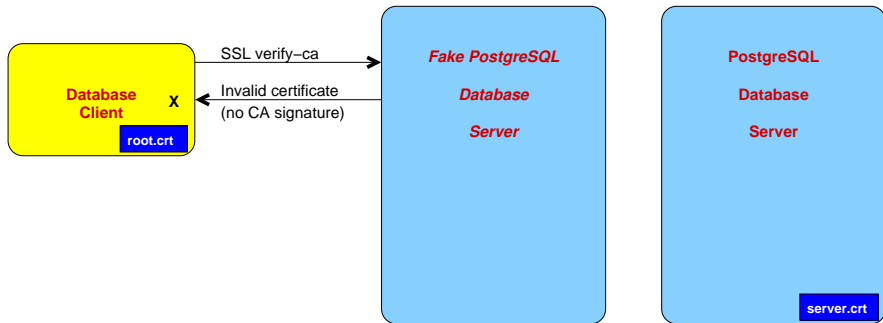
CTEs: Mixing Modification Commands

```
CREATE TEMPORARY TABLE old_orders (order_id INTEGER);

WITH source (order_id) AS (
    DELETE FROM orders WHERE name = 'my order' RETURNING order_id
), source2 AS (
    DELETE FROM items USING source WHERE source.order_id = items.order_id
)
INSERT INTO old_orders SELECT order_id FROM source;
```

Programming the SQL Way with Common Table Expressions

SSL 'Verify-Ca' Is Secure From Spoofing



Conclusion



<https://momjian.us/presentations>