



Jonas Marasus

Senior Database Engineer

War Story: How Big Is Too Big For a Schema

8 May 2025

Copyright © by Thinkproject. All rights reserved.

How Big is Too Big?


This is probably too big:

```
postgres=# SELECT count(*) FROM pg_class;  
count  
-----  
      100000000  
(1 row)
```



What Actually Breaks?

- VACUUM and ANALYZE meet OOM-Killer
 - Postgres' relcache takes 4kb memory per relation visited by VACUUM or ANALYZE. (regardless of maintenance_work_mem)
 - Band aid: Use vacuumdb

A screenshot of a terminal window displaying the output of the 'top' command. The header row is green and contains the following fields: PID, USER, PRI, NI, VIRT, RES, SHR, S, CPU%, MEM%, TIME+, and Command. The MEM% column has a small downward arrow icon next to it. The first data row shows a postgres user process with PID 1808505. Its VIRT memory is 42.4G, RES is 41.7G, and SHR is 17.8G. The CPU% is 72.8 and MEM% is 64.8. The command is 'postgres: postgres db 127.0.0.1(55962) VACUUM'.

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
1808505	postgres	20	0	42.4G	41.7G	17.8G	D	72.8	64.8	8:31.35	postgres: postgres db 127.0.0.1(55962) VACUUM

- pg_dump --schema-only takes hours.
 - Putting dumped objects in the right order requires identifying and fixing dependency cycles, this algorithm is non-linear.
- "out of shared memory" when accessing all tables in a transaction (e.g. pg_dump)
 - Band aid: increase max_locks_per_transaction.

- For dynamic schemas, consider using jsonb instead.
- When partitioning, avoid large numbers of small partitions.
- Anticipate some changes in usage patterns
- Avoid unused indexes even on empty tables

Thank you!