



PostgreSQL 19: What to Expect from the Upcoming Release

A practical preview of PostgreSQL 19: the features shipping, the ones just pulled, and what DBAs and developers should check before upgrading.

18 · Stable

19 Beta 4 · Sep 24, 2026

GA targeted end of Oct 2026



Pranav Ghotekar

Consulting Lead, Mydbops

Friday, 25 September 2026

3:30 PM – 4:15 PM IST · Live Webinar



Pranav Ghotekar



Consulting Lead

Leads PostgreSQL consulting engagements: performance tuning, replication topologies, and high availability.



Focus for this session

A grounded look at PostgreSQL 19: what shipped, what changed its mind mid-beta, and what breaks on upgrade.



About Mydbops

ISO-certified managed database services and consulting for MySQL, MariaDB, PostgreSQL, and TiDB.

MYDBOPS AT A GLANCE

10+

Years of open-source database expertise

6000+

Servers managed

300+

Clients



What this session covers



A preview, not a launch

PostgreSQL 19 is in beta, this is a guided look at where it stands today, not a finished feature list.



The features that matter operationally

REPACK, parallel autovacuum, WAIT FOR, logical replication, and the new plan-advice tooling.



What just got pulled

Five features were reverted, one by one, during the beta cycle. We'll walk through them in order and what that means for you.



A pre-upgrade checklist

PostgreSQL 18 vs 19 comparison and breaking changes worth testing before 19 reaches general availability.



What we will walk through

01 Release status & timeline

02 PostgreSQL 18 vs 19, at a glance

03 Smarter SQL & developer experience

04 Maintenance you don't dread

05 Replication & observability

06 Five features that got reverted

07 Breaking changes to check

08 Key takeaways, sources & Q&A



SECTION 1

Release status & timeline

PostgreSQL 19 has not shipped yet. Here's exactly where it stands as we speak, and what that beta status actually means for you.





How we got here - and what's next

JUST ANNOUNCED (SEP 3, 2026): Beta 4 ships September 24 - the day before this webinar. Details below.

- Apr 8, 2026** ● **Feature freeze**
No new features accepted; the project moves to stabilization and bug fixing.
- Jun 4, 2026** ● **Beta 1 released**
First public preview. SQL/PGQ, ON CONFLICT DO SELECT, REPACK, parallel autovacuum all included.
- Jul 16, 2026** ● **Beta 2 released**
Fixes for temporal SQL, SQL/PGQ, logical decoding, and autovacuum regressions.
- Aug 13, 2026** ● **Beta 3 released**
28 security vulnerabilities and over 110 bugs fixed, across all supported versions.
- Jul 17 – Sep 16** ● **Five features reverted**
GROUP BY ALL, partitions, SQL/PGQ, temporal DML & checksums - details in Section 06.
- Sep 24, 2026** ● **Beta 4 released [NEW]**
Confirmed Sept 3 via the official pgsql-hackers list - the newest build, released the day before today's session.
- End of Oct 2026** ● **General availability (target)**
Slipped from the original Sep/Oct window due to a higher volume of reported issues this cycle; RCs come first.



PostgreSQL 19 at a glance

A practical release. Fewer flashy new engines, more of the things DBAs fight with every week.



REPACK

Online table rebuilds without VACUUM FULL's exclusive lock



Parallel autovacuum

Multi-worker vacuum plus a scoring system for what runs first



IGNORE NULLS

lead(), lag(), and friends can now skip NULLs automatically



Sequences replicate

Logical replication finally carries sequence values



WAIT FOR

Standbys can wait for a chosen LSN - read-your-writes made easy



ON CONFLICT DO SELECT

Atomic get-or-create in a single statement



COPY TO ... JSON

Native JSON output for COPY, including a single-array mode



pg_plan_advice

An official, in-core way to stabilize planner decisions



PostgreSQL 19, quantified

Every figure here comes straight from the official PostgreSQL release notes and announcements.

4

Beta releases so far

Beta 1 (Jun 4), Beta 2 (Jul 16), Beta 3 (Aug 13),
Beta 4 (Sep 24)

28

Security vulnerabilities fixed

In the Aug 13, 2026 update, across all supported
versions

110+

Bugs fixed

In that same Aug 13 update round

4

Major new system views

pg_stat_lock, pg_stat_recovery,
pg_stat_autovacuum_scores,
pg_dsm_registry_allocations

5

Notable features reverted pre-GA

GROUP BY ALL, partitions, SQL/PGQ, FOR
PORTION OF, checksums

~7

Months, freeze to target GA

Apr 8 feature freeze → end of Oct 2026 target
release



PostgreSQL 18 → 19: what actually changes

Capability	PostgreSQL 18	PostgreSQL 19
Online table rebuild	VACUUM FULL / CLUSTER - exclusive lock	REPACK CONCURRENTLY - reads & writes continue
Autovacuum	One worker per table, age-based order	Parallel workers + a priority scoring system
Data checksums	Toggle only offline, via pg_checksums	Reverted for 19 - still offline-only
Logical replication scope	Tables only	Tables + sequences
Enabling logical replication	Requires raising wal_level and a restart	Can switch on automatically, no restart
Standby read-your-writes	No built-in primitive	WAIT FOR LSN
Query plan stability	Third-party extensions only	Official pg_plan_advice / pg_stash_advice
INSERT ... ON CONFLICT	DO NOTHING / DO UPDATE	Adds DO SELECT for atomic get-or-create
Temporal UPDATE / DELETE	Manual range-splitting in app code	Reverted for 19 - still manual
JIT compilation default	Enabled, cost-based activation	Disabled by default



SECTION 2

Smarter SQL & developer experience

Three additions that trim boilerplate SQL applications have been working around for years.





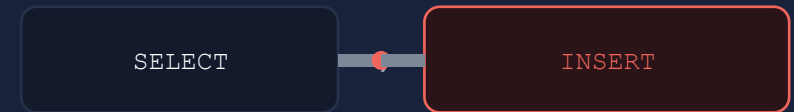
ON CONFLICT DO SELECT - atomic get-or-create

INSERT ... ON CONFLICT DO SELECT ... RETURNING

A single statement that inserts a new row, or - on a conflicting key - selects and optionally locks the existing one with FOR UPDATE / FOR SHARE, all atomically.

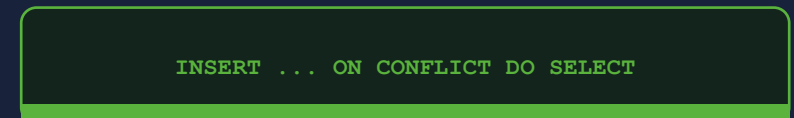
- ☑ Replaces the classic SELECT-then-INSERT-on-failure race condition
- ☑ No more relying on ON CONFLICT DO NOTHING followed by a second SELECT
- ☑ Row-level locking options make it safe under concurrent access

BEFORE



A gap between the two statements - another session can insert the same key first.

NOW



One atomic statement - no window for a second session to slip in between.



FOR PORTION OF - reverted Sept 15, 2026



UPDATE / DELETE ... FOR PORTION OF valid_at - pulled before Beta 4

This would have updated or deleted just a slice of a temporal range, with PostgreSQL automatically splitting the surrounding period so history outside the edit stayed intact. Peter Eisentraut reverted it on September 15, 2026 - 23 commits pulled, no reason stated in the revert itself.

Off the table for 19

Would have closed the last gap in the SQL:2011 temporal-table standard - retargeted at PostgreSQL 20.

Still app-side splitting

On 19, same as 18, you still hand-roll the range split across two or three statements in application code.

Watch this space for v20

The design built on PostgreSQL's existing range and multirange types - worth re-testing once it resurfaces in a future release.



Smaller changes worth knowing



Window functions skip NULLs

`lead()`, `lag()`, `first_value()`, `last_value()`, and `nth_value()` gain an `IGNORE NULLS / RESPECT NULLS` clause.



More jsonpath string methods

`lower()`, `upper()`, `initcap()`, `replace()`, `split_part()`, and the trim family arrive in jsonpath expressions.



GROUP BY handles more subqueries

Target-list subqueries with expressions referencing outer columns are now processed correctly, plus a `GROUPING()` alias fix.



COPY TO can emit JSON

Native JSON output for COPY, including a single-array mode via the `FORCE_ARRAY` option.



SECTION 3

Maintenance you don't dread

The operations that used to mean a maintenance window now run online, in parallel, and with far better visibility.





REPACK replaces VACUUM FULL & CLUSTER

```
REPACK table_name [ USING INDEX index_name ] [ CONCURRENTLY ]
```

One command, one clear name, for reclaiming disk space and reorganizing table contents - the functionality VACUUM FULL and CLUSTER always had, minus the confusing naming. Both old commands still work for compatibility.

Without CONCURRENTLY

- Behaves like today's VACUUM FULL / CLUSTER
- Takes an ACCESS EXCLUSIVE lock
- Fastest option when downtime is acceptable

With CONCURRENTLY

- Rebuilds the table without blocking reads or writes
- Uses replication slots to track concurrent changes
- New max_repack_replication_slots setting governs it



What REPACK CONCURRENTLY actually buys you

BEFORE · VACUUM FULL



Table locked, end to end



Every read and write queues up behind an ACCESS EXCLUSIVE lock until the rebuild finishes on a large table, that's a scheduled maintenance window.



NOW · REPACK CONCURRENTLY



Application keeps running



PostgreSQL builds a fresh copy of the table in the background, using a replication slot to catch up on changes, then swaps it in with only a brief final lock.



Autovacuum gets parallel and prioritized



Parallel workers

Autovacuum can now use multiple parallel worker processes to vacuum a table's indexes at once, instead of working through them one at a time.

`autovacuum_max_parallel_workers`

Server-wide cap; per-table override via the `autovacuum_parallel_workers` storage parameter.

New: `pg_stat_autovacuum_scores` reports the per-table scoring details, and `VACUUM (VERBOSE)` now logs memory usage and parallelism.



A scoring system for what runs first

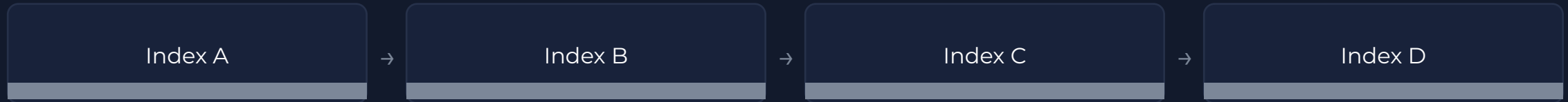
Instead of a rough age-based ordering, autovacuum now scores every table and works on whichever needs it most.

- `autovacuum_freeze_score_weight`
- `autovacuum_multixact_freeze_score_weight`
- `autovacuum_vacuum_score_weight`
- `autovacuum_vacuum_insert_score_weight`
- `autovacuum_analyze_score_weight`



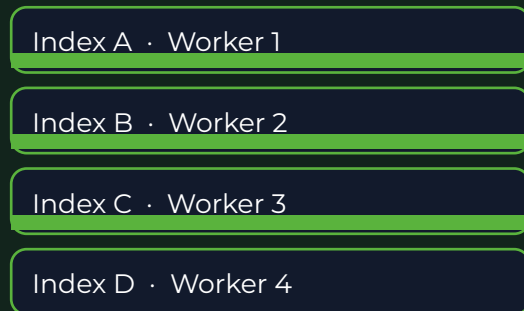
Serial vs. parallel, side by side

POSTGRESQL 18 - ONE WORKER, ONE INDEX AT A TIME



Total time = A + B + C + D, run one after another

POSTGRESQL 19 - MULTIPLE WORKERS, IN PARALLEL



All four run at once - total time $\approx$ the slowest single index, not the sum of all four.



Online checksums - reverted Sept 16, 2026



Previously, checksum status could only change while the cluster was offline.

PostgreSQL 19 would have let you toggle checksums on a running server. It shipped through Beta 3, then was reverted Sept 16, 2026.

Still offline-only

On 19, same as 18, turning checksums on or off still means a maintenance window with `pg_checksums`.

Work continues for v20

The worker, state machine, and tests stay intact on master, retargeted at PostgreSQL 20, not abandoned.

Check your rollout plan

If your 19 plan assumed online checksum toggling, pull it out and budget a maintenance window instead.



COPY grows up



JSON output for COPY TO

New JSON format, plus a `FORCE_ARRAY` option to emit a single JSON array instead of one object per line.



Works on partitioned tables

COPY TO can now read straight from a partitioned table, previously you needed COPY (SELECT ...) as a workaround.



Multiple header lines

COPY FROM can skip more than one header line, not just the first.



Bad values become NULL

The new `ON_ERROR SET_NULL` option converts invalid input into NULL instead of aborting the load.



SECTION 4

Replication & observability

Logical replication closes a long-standing gap, and the server finally tells you what it's waiting on.





Logical replication catches up on two fronts



Sequences finally replicate

- Publications can now include ALL SEQUENCES
- CREATE SUBSCRIPTION and ALTER SUBSCRIPTION ... REFRESH SEQUENCES sync values
- pg_get_sequence_data() lets you inspect sync status directly



Enable it without a restart

- When wal_level = replica, logical replication can now switch on automatically as needed
- The new effective_wal_level setting reports what's actually active
- One less restart standing between you and a logical subscriber

Also new: retain_dead_tuples on a publication keeps conflict-resolution data around, with max_retention_duration to cap how long - useful for bidirectional replication setups.



What replicates now, that didn't before

PUBLISHER



orders table



orders_id_seq

NEW in 19: sequence now travels with its table

table →

sequence →

SUBSCRIBER



orders table



orders_id_seq

In PostgreSQL 18, this value was NOT kept in sync, a common surprise after failover.



WAIT FOR - read-your-writes on standbys



```
WAIT FOR LSN 'lsn_value' WITH ( MODE 'standby_replay', TIMEOUT '5s', NO_THROW )
```

A standby can now block until it has replayed changes up to a chosen point, so a client that just wrote to the primary can query a replica and be sure its own change is visible.

1

Client writes to primary

Primary returns the LSN of the commit.

2

Client issues WAIT FOR on the standby

Connection blocks until that LSN is replayed, or the timeout fires.

3

Client reads from the standby

The write is guaranteed visible, no more polling or artificial delays.



New eyes on the planner and the server



pg_plan_advice & pg_stash_advice

An official, in-core contrib module for stabilizing and controlling planner decisions, no more relying on third-party extensions to pin a query plan. `pg_stash_advice` stores per-query-id advice and applies it automatically the next time that query runs.

Also: EXPLAIN gains an IO option for async I/O activity, and Memoize cache/lookup estimates are now shown in plans.

NEW SYSTEM VIEWS

pg_stat_lock

Per-lock-type statistics, server-wide.

pg_stat_recovery

Recovery status and progress for standbys.

pg_stat_autovacuum_scores

The per-table scoring behind autovacuum's queue.

pg_dsm_registry_allocations

Dynamic shared memory usage by registry.



SECTION 5

Reverted, one by one

PostgreSQL 19 hasn't shipped and it kept getting smaller during the beta cycle. Here's exactly what changed, and when.





The reverts, one by one

Same pattern, five times over thirteen weeks - a bug or design gap found too late to fix safely.

1

GROUP BY ALL · Jul 17, 2026

Returned wrong results for columns that also appear in ORDER BY with non-default equality semantics.

2

Partition MERGE / SPLIT · Aug 27, 2026

14 commits pulled, unresolved design issues surfaced too late in the cycle to fix safely.

3

SQL/PGQ - property graph queries · Sep 7, 2026

The release's headline feature. 47 commits pulled after locking and prerequisite-check gaps were found.

4

UPDATE / DELETE ... FOR PORTION OF · Sep 15, 2026

23 commits pulled, right before Beta 4's commit freeze, no reason stated in the revert itself.

5

Online data-checksum toggling · Sep 16, 2026

Pulled from the stable branch over concerns more issues could surface after GA. Work continues on master for v20.

Worth knowing: [postgresql.org's release-19.html](https://www.postgresql.org/release-19.html) lags the commit history, it still lists SQL/PGQ, FOR PORTION OF, and checksum toggling as shipping. Beta 3/4 and the commit record confirm all five are out.



The three headline reverts - two more just before Beta 4

REVERTED JUL 17, 2026

GROUP BY ALL

Auto-grouped every non-aggregate column, saving you from re-listing the SELECT list in GROUP BY. Reverted after a post-commit review found it returned wrong results for columns that also appear in ORDER BY with non-default equality semantics.

REVERTED AUG 27, 2026

ALTER TABLE ... MERGE/SPLIT PARTITIONS

Would have let you merge or split partitions in place. 14 commits were pulled after unresolved design issues surfaced too late in the cycle to fix safely.

REVERTED SEP 7, 2026

SQL/PGQ - property graph queries

Was set to be the release's headline feature: CREATE PROPERTY GRAPH and GRAPH_TABLE pattern-matching over relational tables. 47 commits pulled, the feature and every fix built on it since. (Two more reverts followed Sept 15-16 - see previous slide.)



What this means for your rollout



Earliest return is PostgreSQL 20

All five reverted features are expected to be retried in the next annual release, around September 2027 - not a guarantee, but the stated intent.



Don't trust Beta-1 blog posts alone

Plenty of write-ups from June still describe SQL/PGQ and GROUP BY ALL as PostgreSQL 19 features. Cross-check anything you read against Beta 3 or later.



Re-check any 2026 plan built on these

If a migration, a training deck, or an app design assumed graph queries, in-place partition splitting, temporal FOR PORTION OF, or online checksum toggling in 19, that assumption needs to be removed now.



Test against Beta 4, not earlier betas

Beta 4 (Sep 24, 2026) is the newest build - announced Sep 3 and released the day before this webinar. It's the closest preview yet of what will actually ship.



Defaults and behaviors that will change



JIT is now disabled by default

The cost-based activation logic was unreliable. Analytical-query-heavy sites must now opt back in manually.



TOAST compression defaults to lz4

Changed from pglz for better compression performance out of the box.



max_locks_per_transaction default doubles: 64 → 128

Lock allocation changed internally, so existing tuned values effectively need doubling to match prior capacity.



RADIUS authentication removed

It only ever supported UDP, which cannot be made secure.



standard_conforming_strings forced on

Pre-19 dumps made with it off will not load, regenerate dumps with a current pg_dump.



Default inet/cidr index opclass changes to GiST

pg_upgrade will refuse to upgrade a cluster with an index on the old btree_gist opclasses, they could silently miss matching rows.



Key takeaways

1

It's a practical release, not a flashy one

REPACK, parallel autovacuum, WAIT FOR, and plan advice solve everyday operational pain.

2

Verify your roadmap against Beta 4, not Beta 1

GROUP BY ALL, SQL/PGQ, partition merge/split, FOR PORTION OF, and online checksums were all pulled after beta, check any plan that assumed them.

3

Budget time for the default changes

JIT off, lz4 TOAST, and the doubled max_locks_per_transaction default are easy to miss and easy to test now.

4

Logical replication is much closer to complete

Sequence replication and restart-free enablement remove two long-standing operational headaches.

5

Start testing on Beta 4 today

GA has slipped slightly to end of October 2026, a beta cluster now turns launch day into a formality, not a fire drill.



Official sources

Every fact, date, and figure in this deck traces back to these official PostgreSQL Global Development Group pages.



postgresql.org/docs/19/release-19.html

Official PostgreSQL 19 release notes - features, incompatibilities, and full change list. Recheck before airtime.



postgresql.org/about/news/postgresql-19-beta-1-released

Beta 1 announcement - June 4, 2026



postgresql.org/about/news/postgresql-19-beta-2-released

Beta 2 announcement - July 16, 2026



postgresql.org/about/news/...-19-beta-3-released

Beta 3 announcement - August 13, 2026, with the full CVE and bug-fix list



lists.postgresql.org - [pgsql-hackers](https://pgsql-hackers.org)

Official mailing list archive - Beta 4 date (Sep 24), revised end-of-Oct GA target, and the Sept 15-16 revert threads



postgresql.org

Main project site - current release status and news archive



commitfest.postgresql.org

Official patch tracker - status of individual features, including reverted ones



Questions?

Thank you for joining MyWebinar #55.

Reach out any time if you'd like a second pass once PostgreSQL 19 reaches general availability.

 www.mydbops.com  www.mydbops.com/contact  linkedin.com/company/mydbops