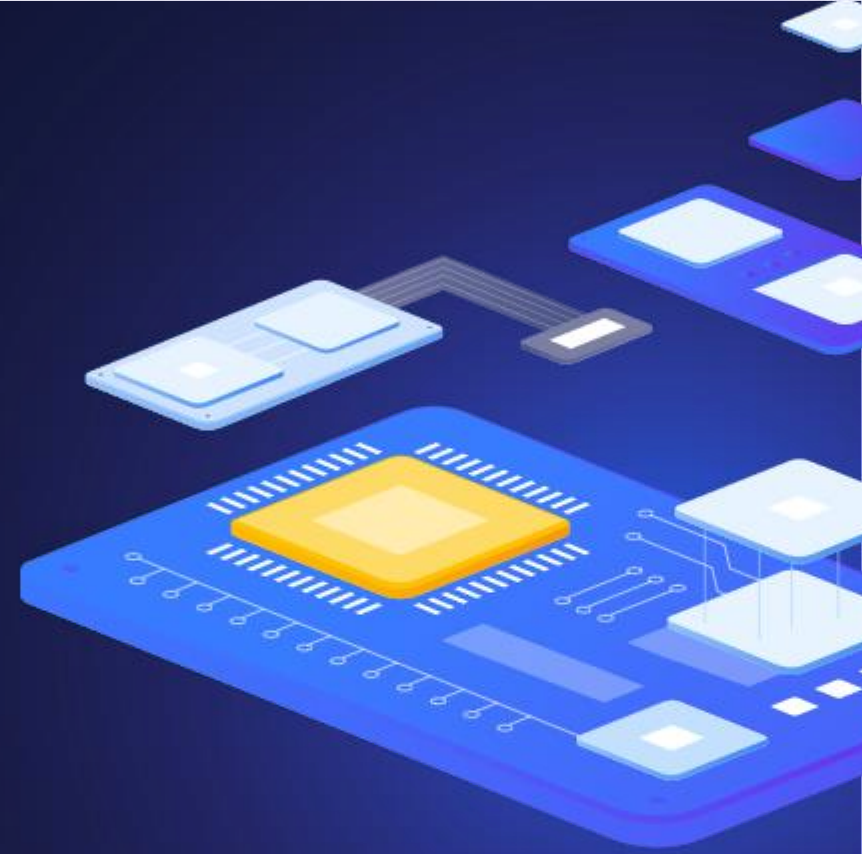




# What's new in PostgreSQL 16

1

Improvements in data processing in  
large environments



# Performance improvements

## › **SELECT DISTINCT statement**

Added the ability for aggregates that have ORDER BY or DISTINCT to use pre-sorted data from an index, if it exists. Previously Postgres would sort the aggregates every time and even if indexes with pre-sorted data existed. For these scenarios, aggregates are now faster.

## › **Reduce memory overhead and increase performance of palloc**

Every time we allocate memory in Postgres via palloc, this feature reduces the size of the palloc header. Postgres can use less memory to do more stuff in query execution.

## What's new in PostgreSQL 16

# Parallel Query Enhancements

- ▶ Parallel Hash Full Join
- ▶ Parallel query is now supported for RIGHT OUTER JOIN and FULL OUTER JOIN syntax.
- ▶ Full JOINS are commonly used to find the differences between 2 tables. Prior to Postgres 16, parallelism was not implemented for full hash JOINS, which made them slow.



Figure 1: FULL JOIN, 1 process

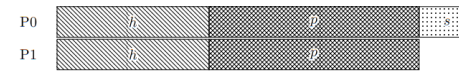


Figure 2: FULL JOIN, 2 processes

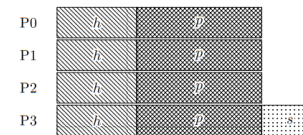


Figure 3: FULL JOIN, 4 processes

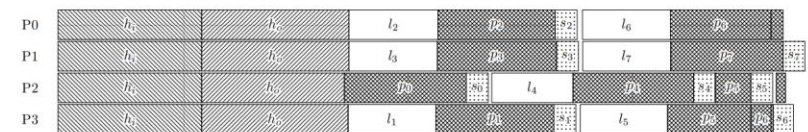


Figure 4: FULL JOIN, 4 processes, 8 batches

# Page-level FREEZE

- ▶ The FREEZE processing can now be executed on a page-level. This will perform WAL deduplication and reduce the number of writes.
- ▶ After a quick benchmark, the amount of WAL records for freezing 1 million tuples reduced to about one-fifth (1.2GB vs 250MB). Great.

# Improved batch processing performance

## ▶ **Faster COPY**

Concurrent copy into a single relation is 3X faster (with 16 clients) in Postgres 16 due to the “refactoring relation extension” work. Even single threaded copy is faster. How? New infrastructure was added to make the process of extending tables more efficient. (“Side effect” of Andreas Freund work on AIO)

- ▶ It is now possible to batch insert multiple records with the COPY statement for a foreign table using the postgres\_fdw module.

# 2

Improvements in server  
administration



# Regex for hostname and username

- ▶ The `pg_hba.conf` file now supports regular expressions for database names and usernames.
- ▶ Usernames and database names starting with a slash (/) are considered regular expressions.

```
# TYPE      DATABASE      USER          ADDRESS      METHOD
host       /^devdb[1-5] dev           10.0.1.0/24  scram-sha-256
host       test         /^jakub.*$   10.0.1.0/24  scram-sha-256
```



# Include another file

- ▶ The `pg_hba.conf` and the `pg_ident.conf` files can now include other files just as the `postgresql.conf` file can.

Syntax	Description
<b>include</b> file_name	Include the file
<b>include_if_exists</b> file_name	Include the file if exists
<b>include_dir</b> dir_name	Include all *.conf files in given directory

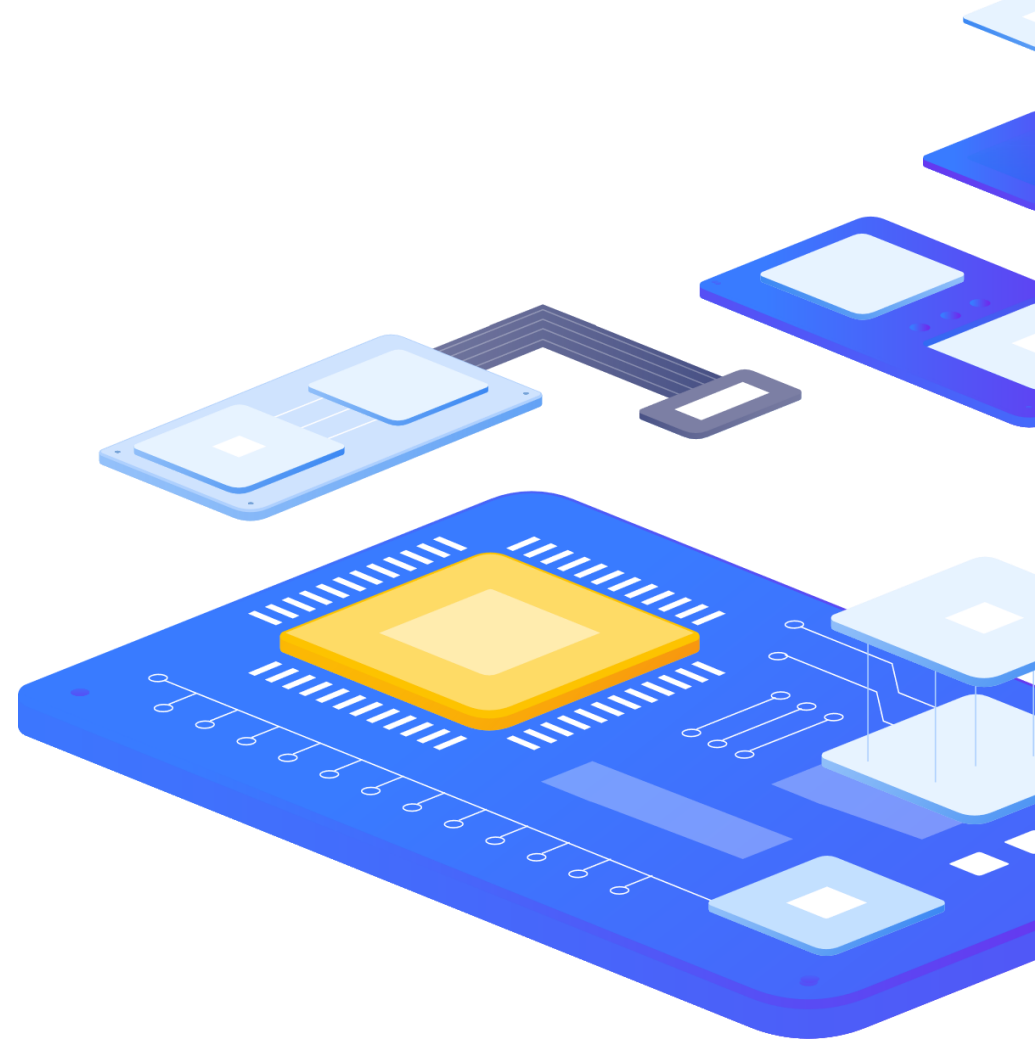
# Logging

## › Checkpoint

The LSN(Log Sequence Number) information is added to the log when the parameter `log_checkpoints` is set to 'on'.

## › Auto vacuum

Add frozen table information to auto vacuum logs.



# Kerberos Credential Delegation

- ▶ PostgreSQL 16 now supports GSSAPI/Kerberos authentication delegated by the client to the server.
- ▶ This enables `postgres_fdw` to connect to another server and authenticate as the same user who connects to it.
- ▶ Because no password is needed when querying foreign table via Kerberos delegation, the "password\_required" option in user mapping must be set to false by a superuser.
- ▶ Other than this, it should work with normal user.

## What's new in PostgreSQL 16

# pg\_stat\_io

- ▶ Providing more detailed IO statistics
- ▶ It doesn't tell us which tables, or which backends are responsible for that IO, but at the least we know when the IO is related to clients, autovacuum, checkpoints, etc.
- ▶ Its just a beginning, right now there are lots of IO not tracked(WAL, „bypass io“, per conn. IO)
- ▶ Parameter **track\_io\_timing** should be set to 'on' to output read\_time column, write\_time column, extend\_time column and fsync\_time column.
- ▶ Execute the **pg\_stat\_reset\_shared** function to reset the contents of the view.

## What's new in PostgreSQL 16

# pg\_stat\_io

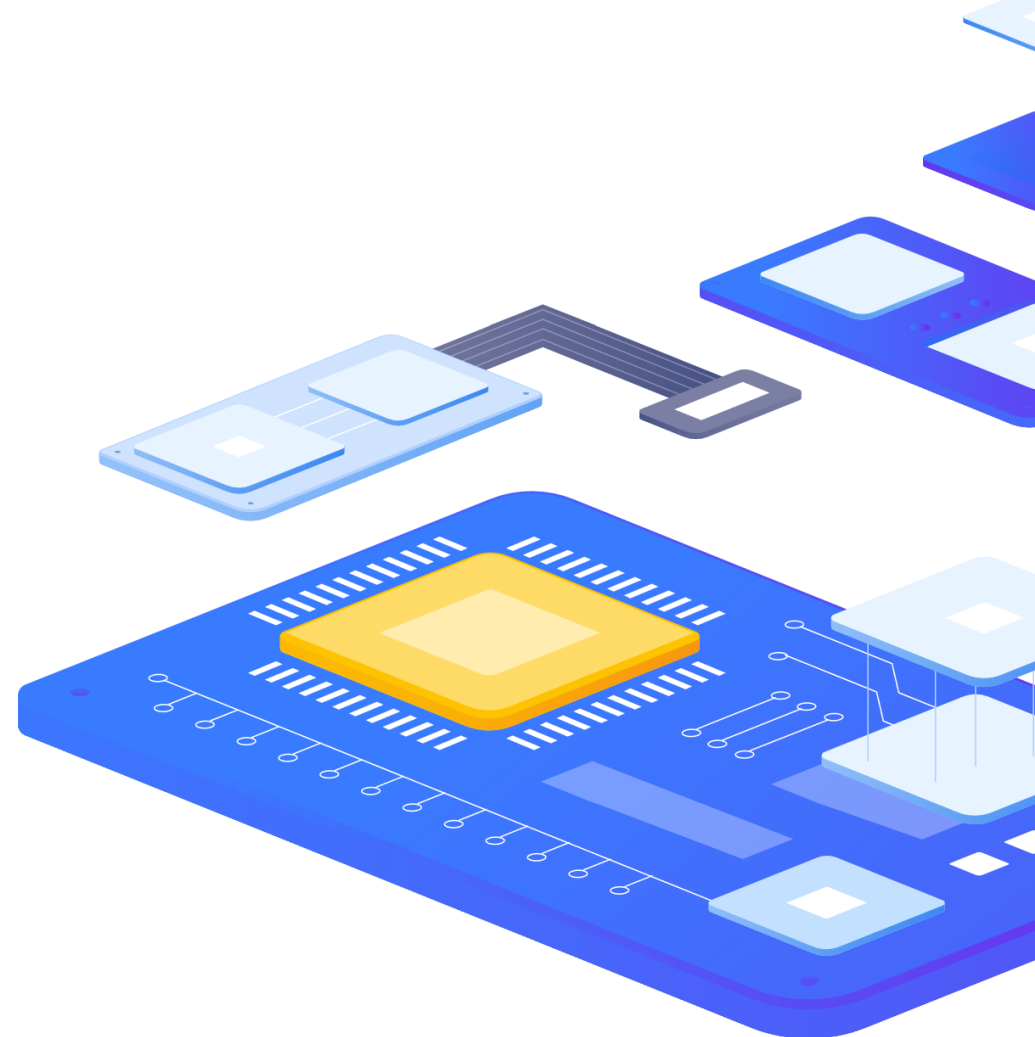
› The context of an I/O operation. Possible values are

› **Normal**

The default or standard context for a type of I/O operation. For example, by default, relation data is read into and written out from shared buffers. Thus, reads and writes of relation data to and from shared buffers are tracked in context normal.

› **Vacuum**

I/O operations performed outside of shared buffers while vacuuming and analyzing permanent relations. Temporary table vacuums use the same local buffer pool as other temporary table I/O operations and are tracked in context normal.



## What's new in PostgreSQL 16

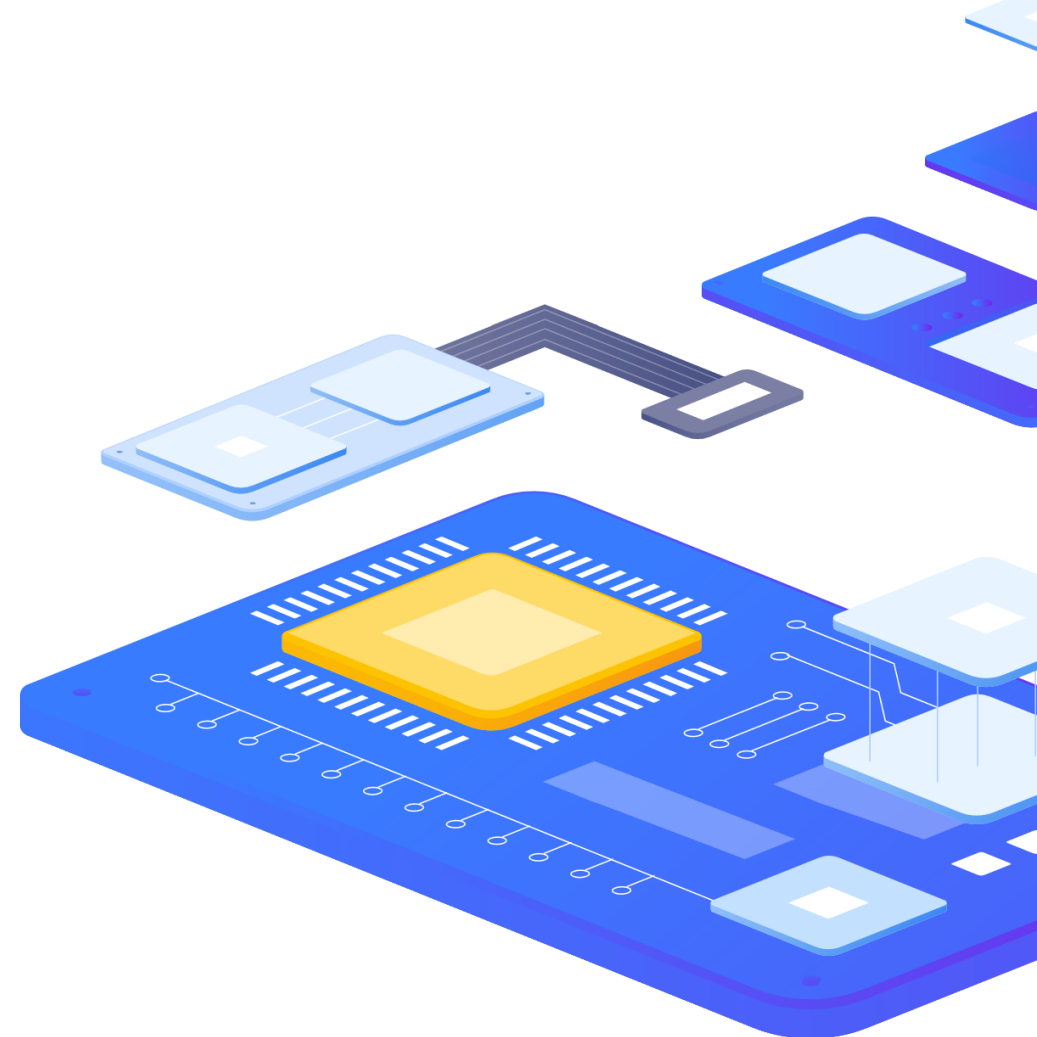
# pg\_stat\_io

### › Bulkread

Certain large read I/O operations done outside of shared buffers, for example, a sequential scan of a large table.

### › Bulkwrite

Certain large write I/O operations done outside of shared buffers, such as COPY.



## What's new in PostgreSQL 16

# pg\_stat\_io

- ▶ `SELECT * FROM pg_stat_io WHERE reads <> 0 OR writes <> 0 OR extends <> 0;`
- ▶ Add a new Backend Type for standalone backends, `B_STANDALONE_BACKEND`.
- ▶ Both the bootstrap backend and single user mode backends will have Backend Type `B_STANDALONE_BACKEND`.

```
postgres=# SELECT * FROM pg_stat_io WHERE reads <> 0 OR writes <> 0 OR extends <> 0;
```

backend_type	object	context	reads	read_time	writes	write_time	writebacks	writeback_time	extends	extend_time	op_bytes	hits	evictions	reuses	fsyncs	fsync_time	stats_reset
client backend	relation	normal	71	0.973	0	0	0	0	0	0	8192	1545	0	0	0	0	2023-09-03 16:46:00.053436+02
standalone backend	relation	bulkwrite	0	0	0	0	0	0	8	0	8192	7	0	0	0	0	2023-09-03 16:46:00.053436+02
standalone backend	relation	normal	535	0	1012	0	1012	0	666	0	8192	85801	0	0	0	0	2023-09-03 16:46:00.053436+02
standalone backend	relation	vacuum	10	0	0	0	0	0	0	0	8192	910	0	0	0	0	2023-09-03 16:46:00.053436+02

(4 rows)

```
postgres=# SELECT * FROM pg_stat_io WHERE reads <> 0 OR writes <> 0 OR extends <> 0;
```

backend_type	object	context	reads	read_time	writes	write_time	writebacks	writeback_time	extends	extend_time	op_bytes	hits	evictions	reuses	fsyncs	fsync_time	stats_reset
autovacuum worker	relation	normal	32	0.5750000000000001	0	0	0	0	10	0.449	8192	10369	0	0	0	0	2023-09-03 16:46:00.053436+02
autovacuum worker	relation	vacuum	13	0.189	0	0	0	0	0	0	8192	248	0	0	0	0	2023-09-03 16:46:00.053436+02
client backend	relation	bulkread	815523	38605.352	0	0	0	0	0	0	8192	1100	0	814709	0	0	2023-09-03 16:46:00.053436+02
client backend	relation	bulkwrite	0	0	1636817	14349.515	0	0	1639364	5667.984	8192	1613728	0	1637371	0	0	2023-09-03 16:46:00.053436+02
client backend	relation	normal	118	1.6460000000000001	0	0	0	0	526	5.066	8192	116527	0	0	0	0	2023-09-03 16:46:00.053436+02
client backend	relation	vacuum	29960	616.403	0	0	0	0	0	0	8192	199	0	29928	0	0	2023-09-03 16:46:00.053436+02
background worker	relation	bulkread	822679	38879.966	0	0	0	0	0	0	8192	976	0	822647	0	0	2023-09-03 16:46:00.053436+02
checkpointer	relation	normal	0	0	3736	43.262	3792	21.001	0	0	8192	0	0	411	86566.306	0	2023-09-03 16:46:00.053436+02
standalone backend	relation	bulkwrite	0	0	0	0	0	0	8	0	8192	7	0	0	0	0	2023-09-03 16:46:00.053436+02
standalone backend	relation	normal	535	0	1012	0	1012	0	666	0	8192	85801	0	0	0	0	2023-09-03 16:46:00.053436+02
standalone backend	relation	vacuum	10	0	0	0	0	0	0	0	8192	910	0	0	0	0	2023-09-03 16:46:00.053436+02

(11 rows)

## What's new in PostgreSQL 16

# pg\_stat\_io

- ▶ You want client backend normal writes close to zero.
- ▶ If client backend normal writes and reads are roughly the same, you maybe need to increase shared buffers.
- ▶ New and better way of calculating cache hit ratio.
- ▶ `SELECT (hits/(hits + reads)::float) * 100 FROM pg_stat_io WHERE backend_type = 'client backend' AND object = 'relation' AND context = 'normal';`

**99,9%**

- ▶ `select (sum(blks_hit) / (sum(blks_read) + sum(blks_hit))::float) * 100 FROM pg_stat_database;`

**45%**



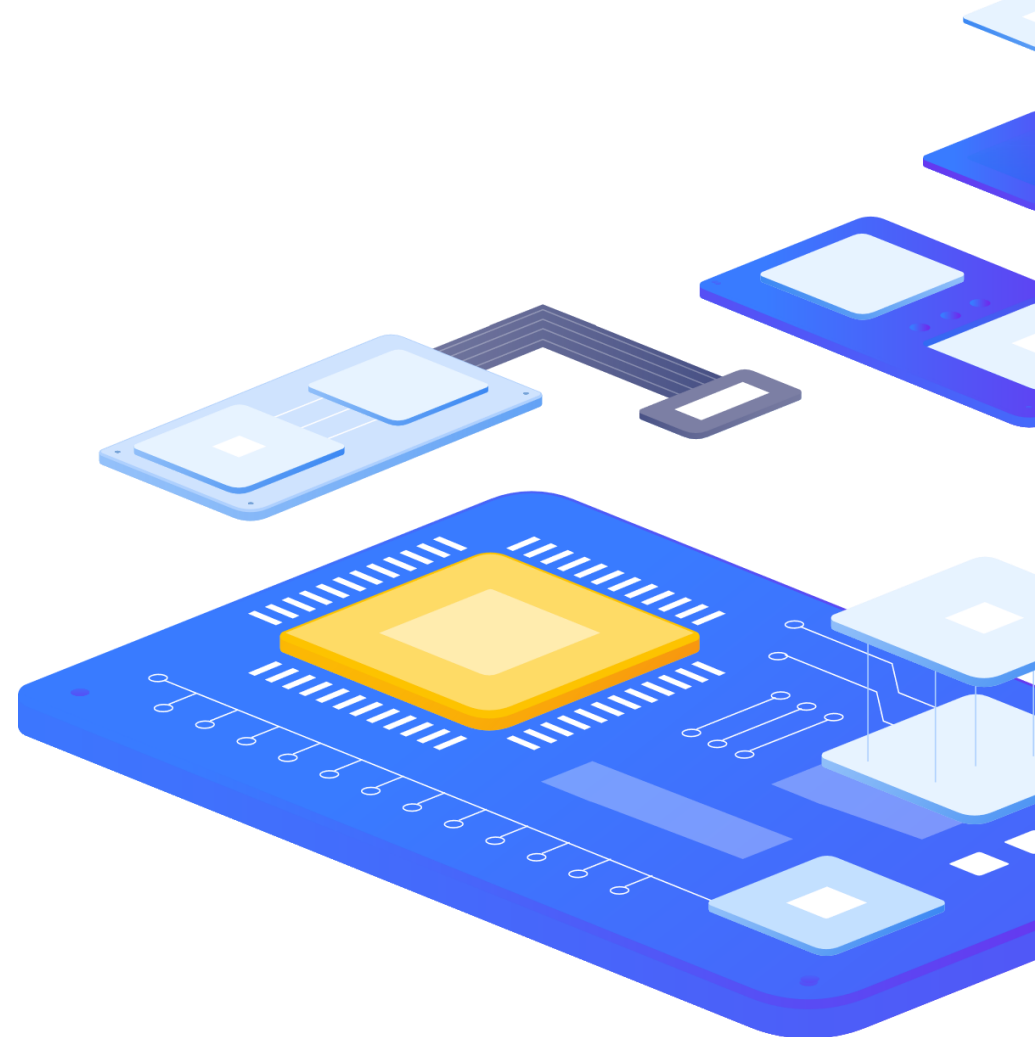
What's new in PostgreSQL 16

# Logical Replication

## › Parallel apply

In previous versions, the large transactions were applied after being saved to a temporary file. PostgreSQL 16 now allows transaction deltas to be applied directly by multiple worker processes.

To enable this feature, specify '**parallel**' for the option streaming in the CREATE SUBSCRIPTION statement. The maximum number of worker processes a subscription uses is determined by the parameter `max_parallel_apply_workers_per_subscription`.



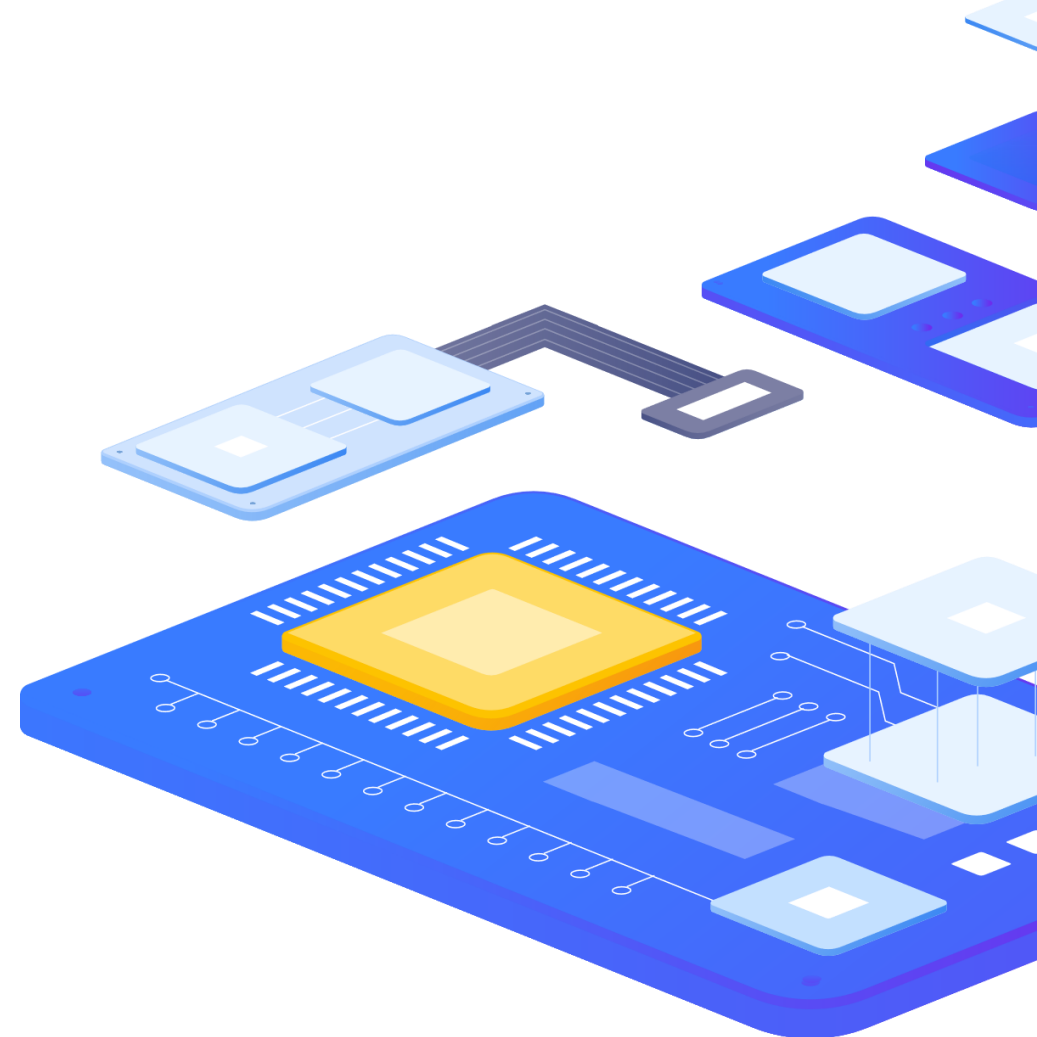
What's new in PostgreSQL 16

# Logical Replication

## › Origin option for SUBSCRIPTION

The CREATE /ALTER SUBSCRIPTION statements now accept the 'origin' option. This parameter specifies the type of change to request to PUBLICATION. A suborigin column has been added to the pg\_subscription catalog to store this option information.

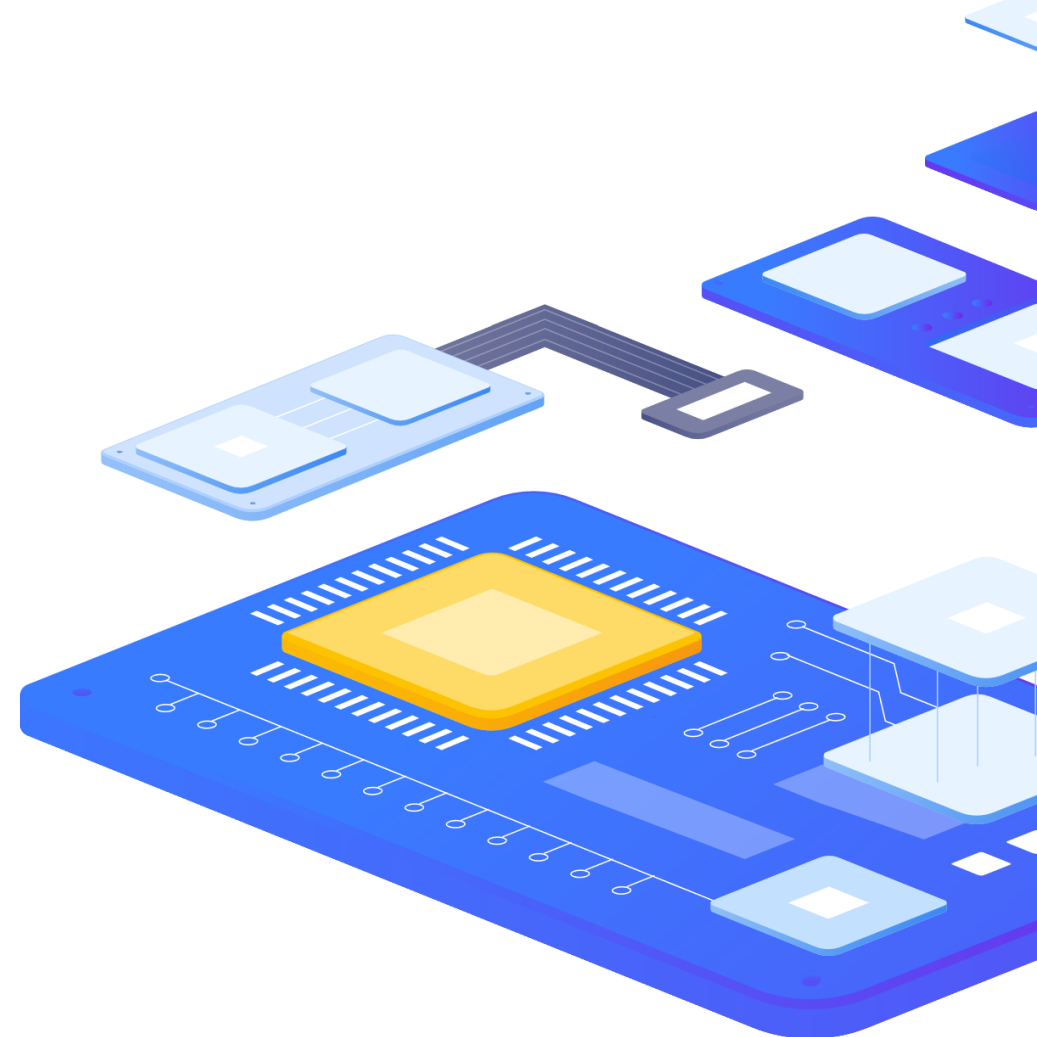
In logical replication, currently Walsender sends the data that is generated locally and the data that are replicated from other instances. This results in infinite recursion in circular logical replication setup.



## What's new in PostgreSQL 16

# Logical Replication

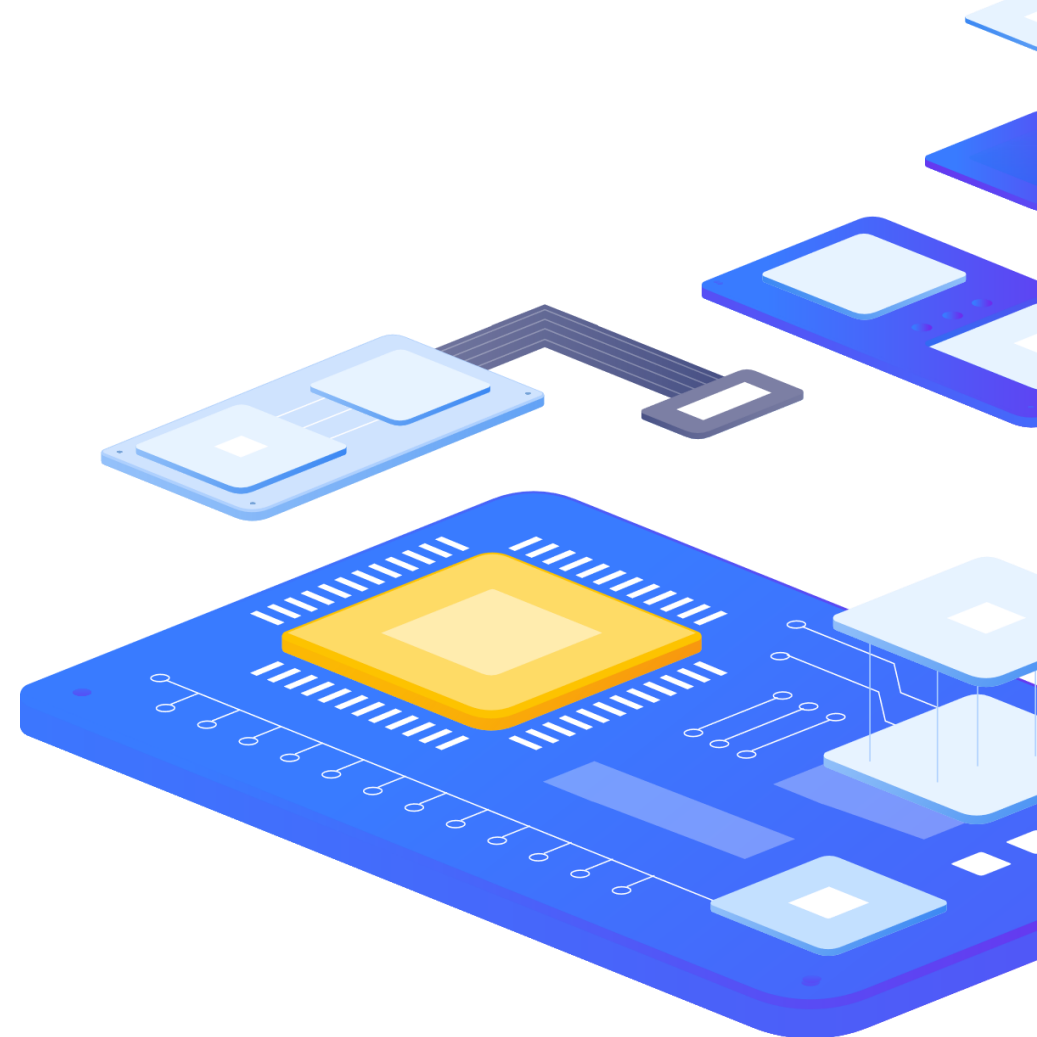
- ▶ **Run\_as\_owner option for SUBSCRIPTION**
- ▶ In previous versions of logical replication, SUBSCRIPTION updated the table with the privileges of the SUBSCRIPTION owner.
- ▶ By default, in PostgreSQL 16, apply is done with the privileges of the table owner.
- ▶ Setting the run\_as\_owner option to 'true' allows the same behavior as in previous versions.
- ▶ The subrunasowner column has been added to the pg\_subscription catalog to store the information of this option.



What's new in PostgreSQL 16

# Logical Replication

- ▶ **SUBSCRIPTION creation role**
- ▶ General users who are granted the `pg_create_subscription` role will be able to create SUBSCRIPTIONs.
- ▶ However, the password is required in the CONNECTION clause.
- ▶ Only users with the SUPERUSER attribute can turn off the `password_required` attribute.
- ▶ The `subpasswordrequired` column has been added to the `pg_subscription` catalog with this option.



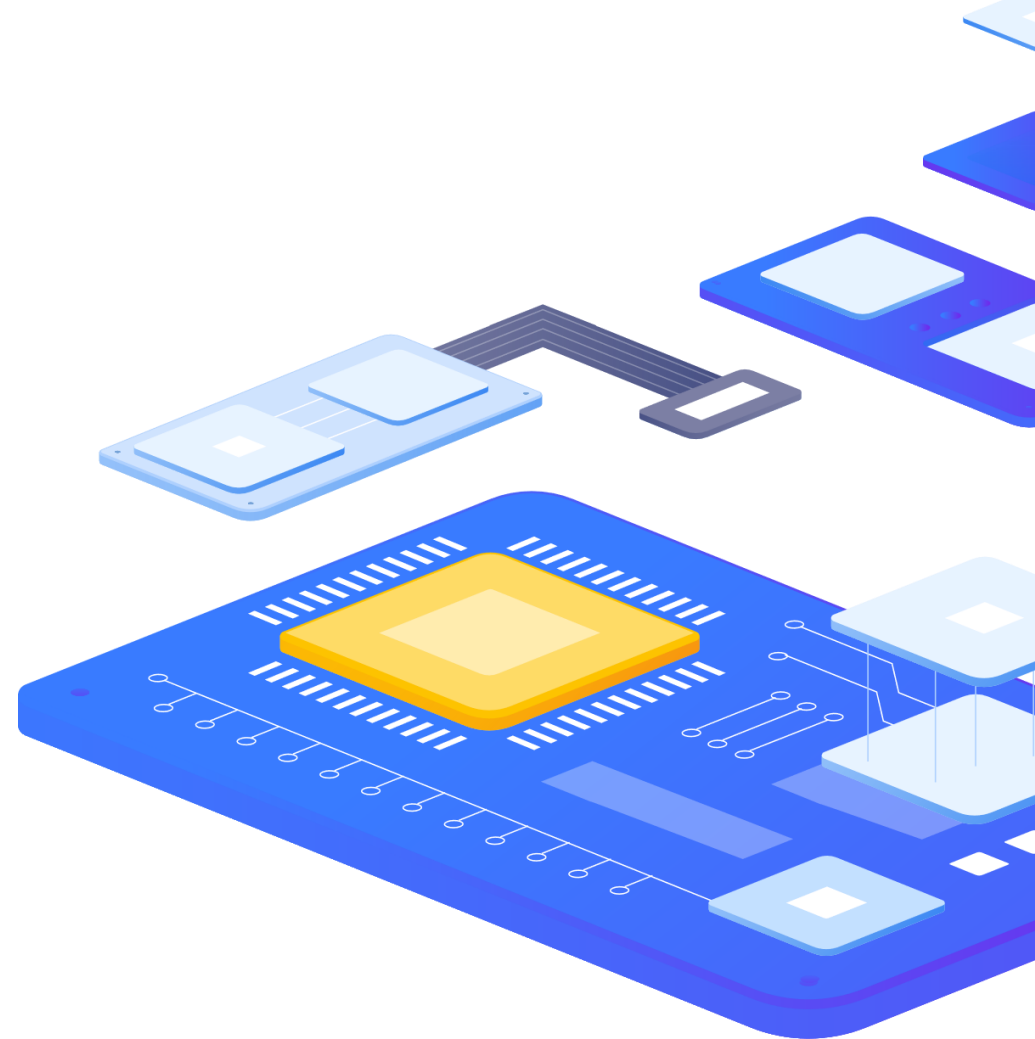
What's new in PostgreSQL 16

# Logical Replication

- ▶ **BINARY transfer for initial data synchronization**
- ▶ When both PUBLICATION / SUBSCRIPTION are PostgreSQL 16 or higher and binary option of the SUBSCRIPTION is enabled, initial data is now transferred in binary format.
- ▶ The following example is the COPY statement log executed for initial data synchronization on the PUBLICATION side.

master (**no binary copy**): **20007.7948** ms

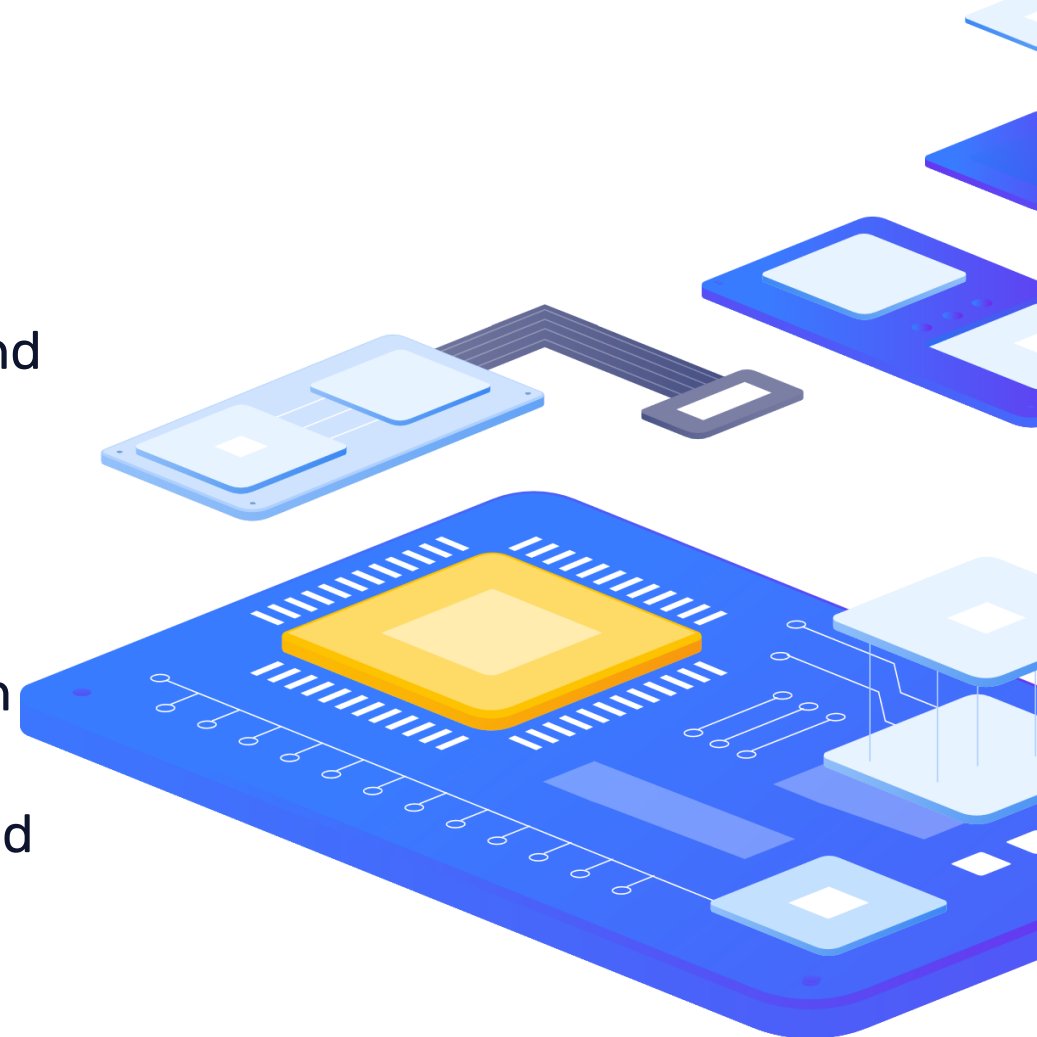
the patch (allows **binary copy**): **8874,869** ms



# Logical Replication

## › Index with **REPLICA IDENTITY FULL**

- › Even in an environment where REPLICA IDENTITY and primary keys cannot be used at the publisher, the subscriber can use indexes other than primary keys and REPLICA IDENTITY.
- › The indexes that can be used are B-tree indexes, not partial indexes, and require at least one column reference.
- › Pgbench\_accounts(bid) with the index, the replication catches up around ~5 seconds.
- › When the index is dropped, the replication takes around ~300 seconds.



# Logical Replication

## ▶ **Logical decoding on standby instance**

- ▶ Logical decoding can now be executed on the standby instance.  
The `pg_create_logical_replication_slot` function can be executed on the standby instance.
- ▶ Creation of a logical slot requires information about all the currently running transactions.
- ▶ On the primary, this information is available directly, but on a standby, this information has to be obtained from primary.
- ▶ Thus, slot creation may need to wait for some activity to happen on the primary.
- ▶ If the primary is idle, creating a logical slot on standby may take noticeable time.
- ▶ This can be sped up by calling the `pg_log_standby_snapshot` function on the primary.

## What's new in PostgreSQL 16

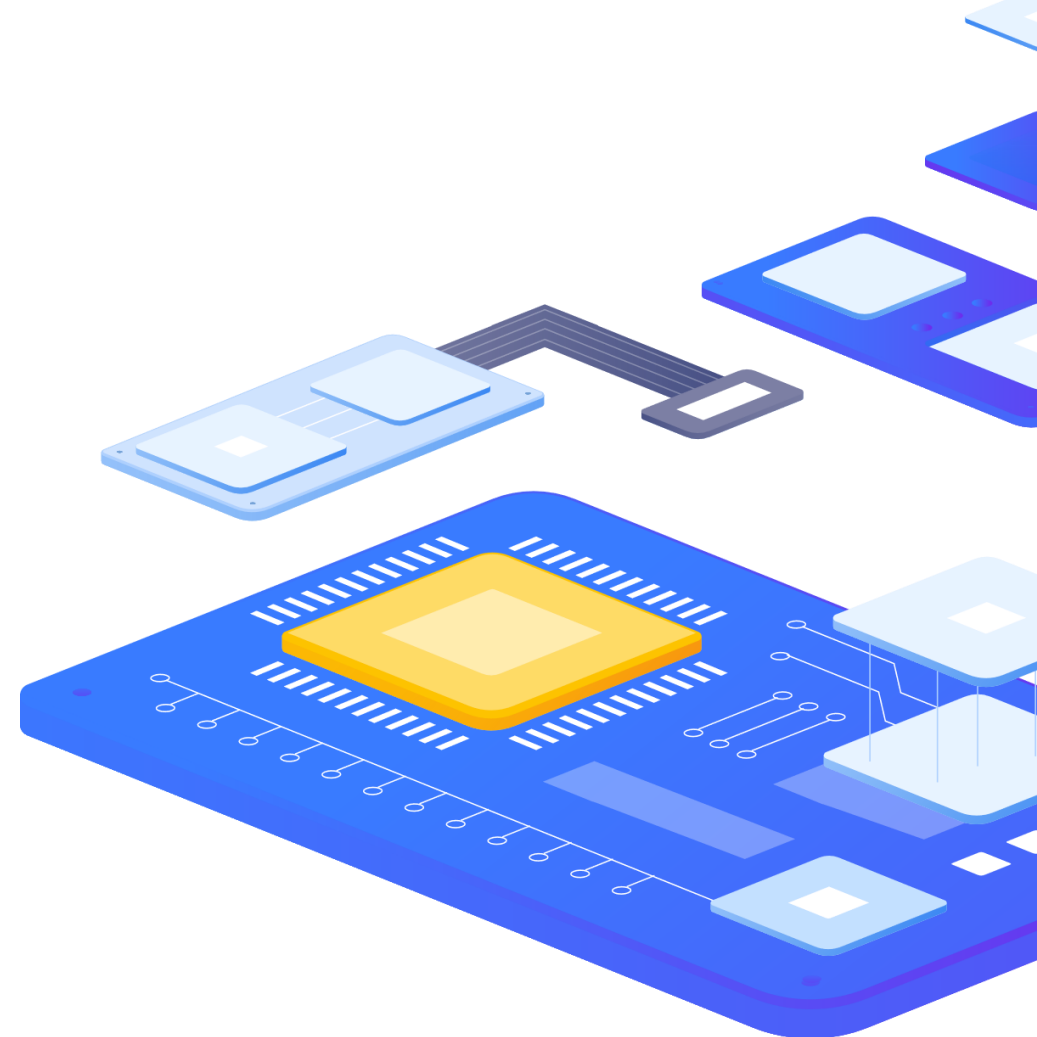
# Libpg

### › **Connection string `require_auth`**

Add a connection string `require_auth` that specifies the list of required authentication methods. If there are no authentication methods that the server can provide, the connection will fail.

### › **Connection string `load_balance_hosts`**

The connection string `load_balance_hosts` automatically selects a specific instance from the destination list. Possible values are `random` and `disable`(default)



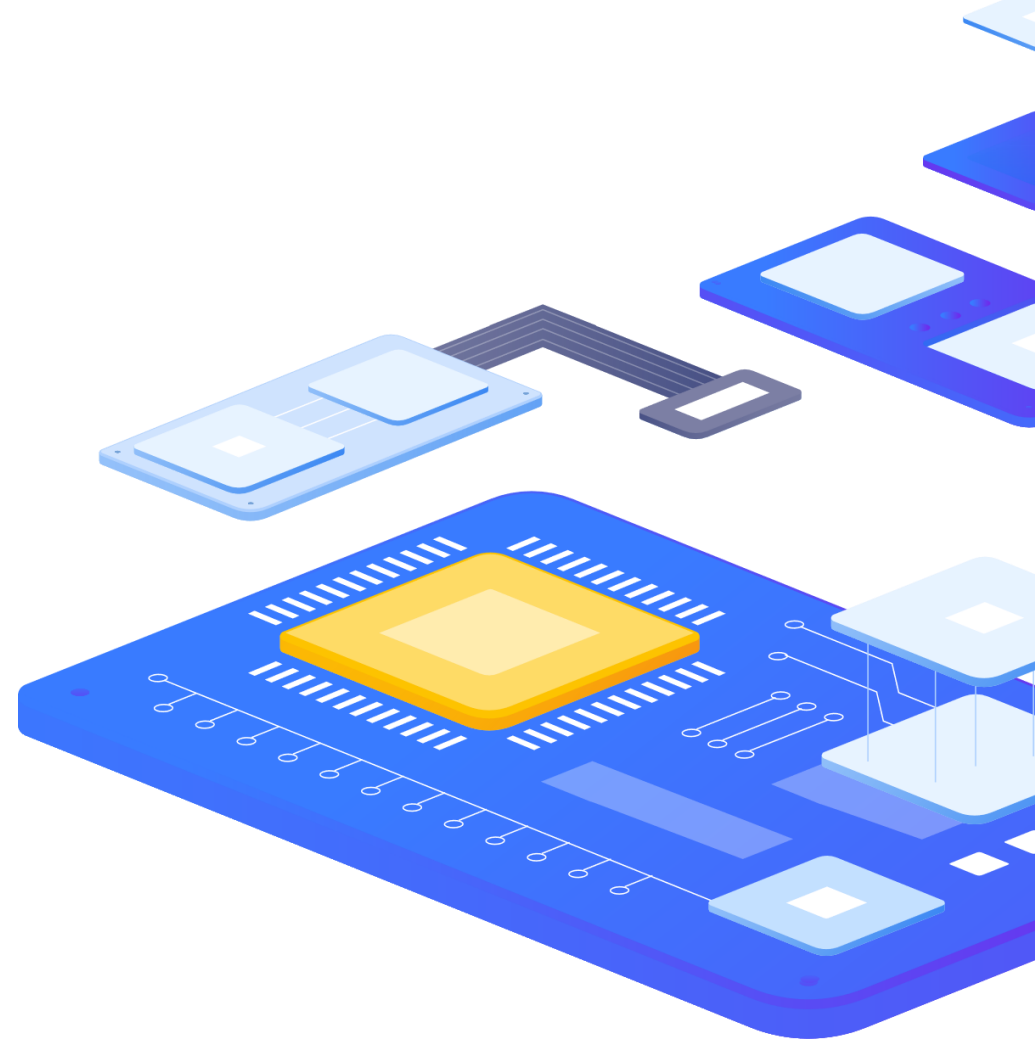


What's new in PostgreSQL 16

# Libpg

## › Connection string sslrootcert

The connection string `sslrootcert` can now specify 'system' as its configuration value. This configuration value loads the system's trusted CA root certificate for certificate validation.



3

Incompatibility



# Promotion Standby

- ▶ Database promotion by trigger file creation is no longer supported.
- ▶ The parameter `promote_trigger_file` has been removed accordingly.
- ▶ Use the `pg_ctl promote` command or `pg_promote` function to execute promotion.

```
pg_ctl promote -D /datadir/path
```

# Bootstrap user attribute

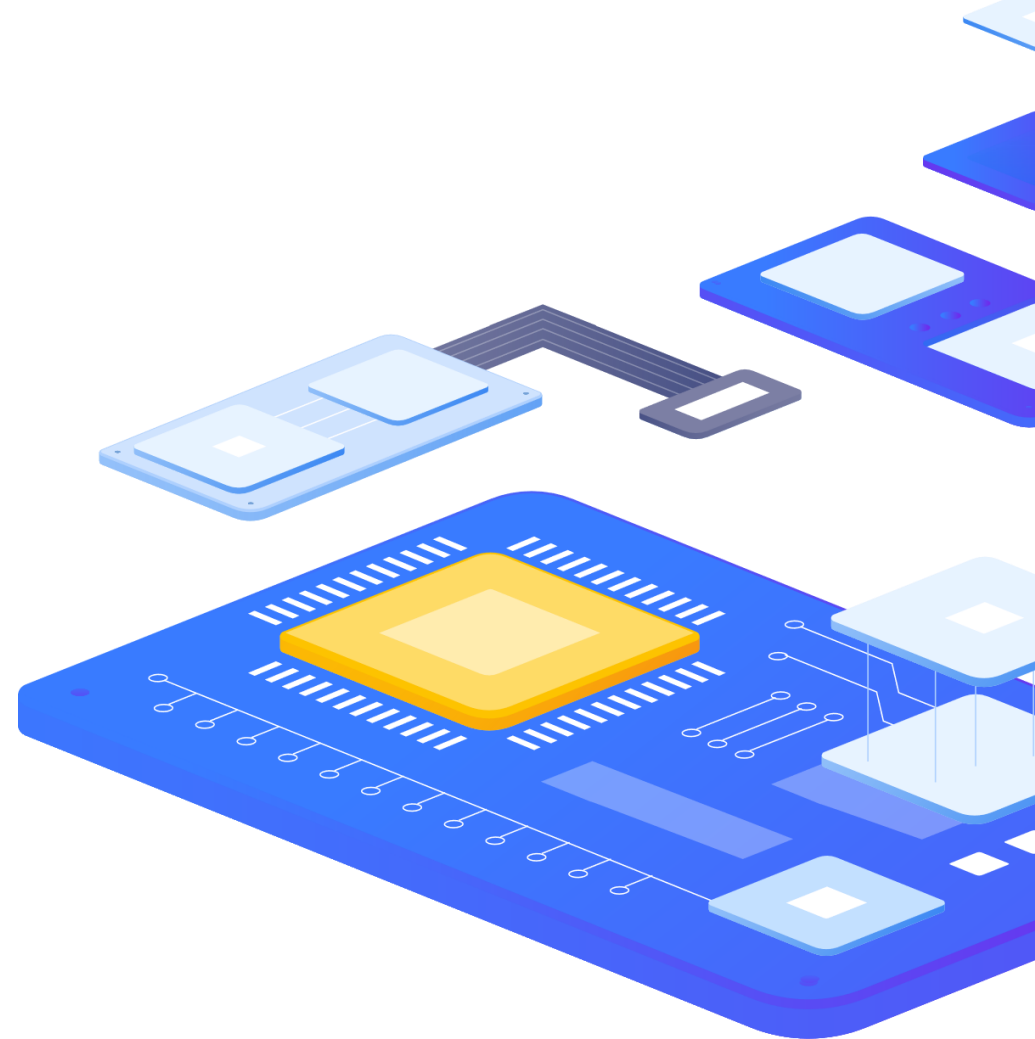
- ▶ The SUPERUSER attribute can no longer be removed from the bootstrap user.

```
postgres=# ALTER USER postgres NOSUPERUSER;  
ERROR: permission denied to alter role  
DETAIL: The bootstrap user must have the SUPERUSER attribute.
```

## What's new in PostgreSQL 16

# WAL Archive

- ▶ Parameter settings the `archive_command` parameters and the `archive_library` can no longer be set at the same time.
- ▶ In PostgreSQL 15 `archive_library` was preferred. If set both at the same time.



# Modified Parameters

The following parameters have been changed.

Parameter name	Changes
archive_command	It can no longer be set at the same time as archive_library.
archive_library	It can no longer be set at the same time as archive_command.
wal_sync_method	It is now possible to specify fdatasync on NTFS in Windows environment.
shared_preload_libraries	Now also handled in single user mode.
debug_parallel_query	Renamed from force_parallel_mode.

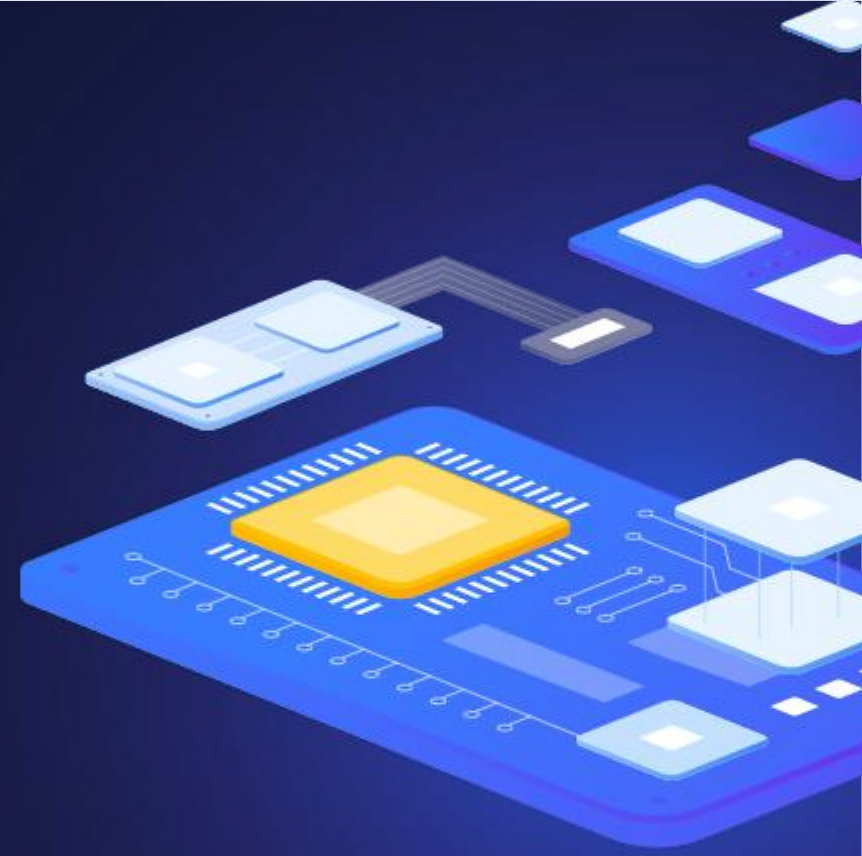
# Removed parameters

The following parameters have been removed

Parameter name	Reason
<code>vacuum_defer_cleanup_age</code>	Removed due to difficulty in determining a reasonable setpoint and the availability of alternatives
<code>promote_trigger_file</code>	Removed because process startup every 5 seconds was deemed useless and trigger files are no longer supported

4

SQL

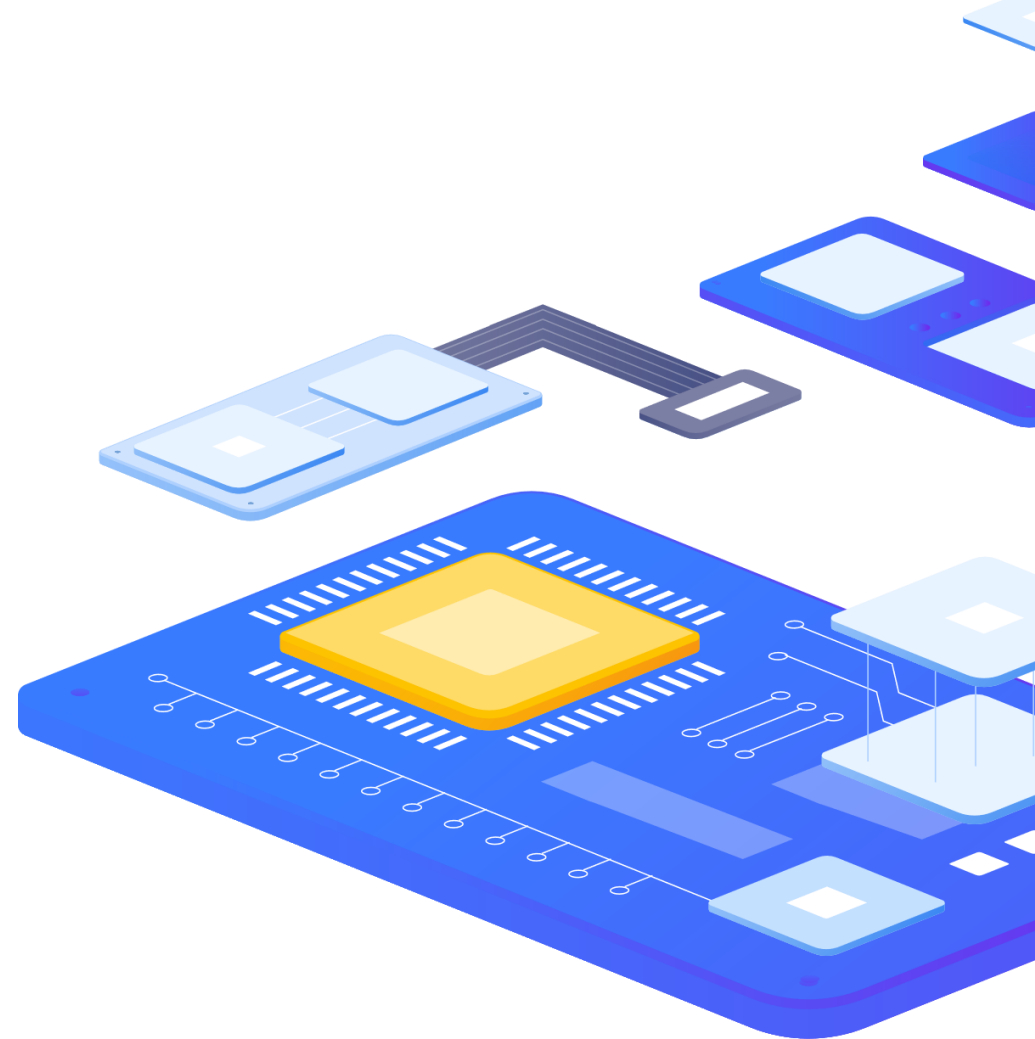




## What's new in PostgreSQL 16

# COPY

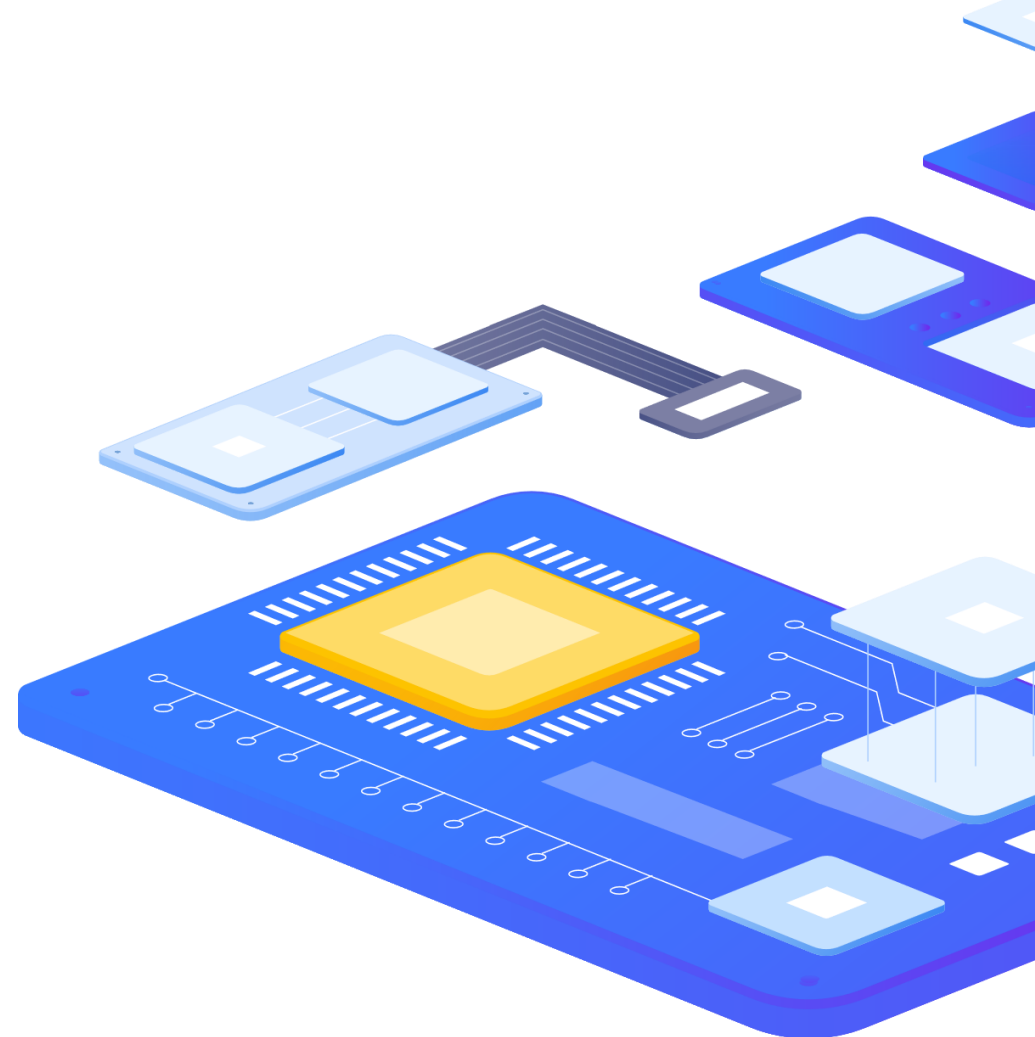
- › The option DEFAULT have been added to the COPY FROM statement.
- › If the value specified for this option matches the input value, then the DEFAULT value for the column will be entered.



## What's new in PostgreSQL 16

# JSON

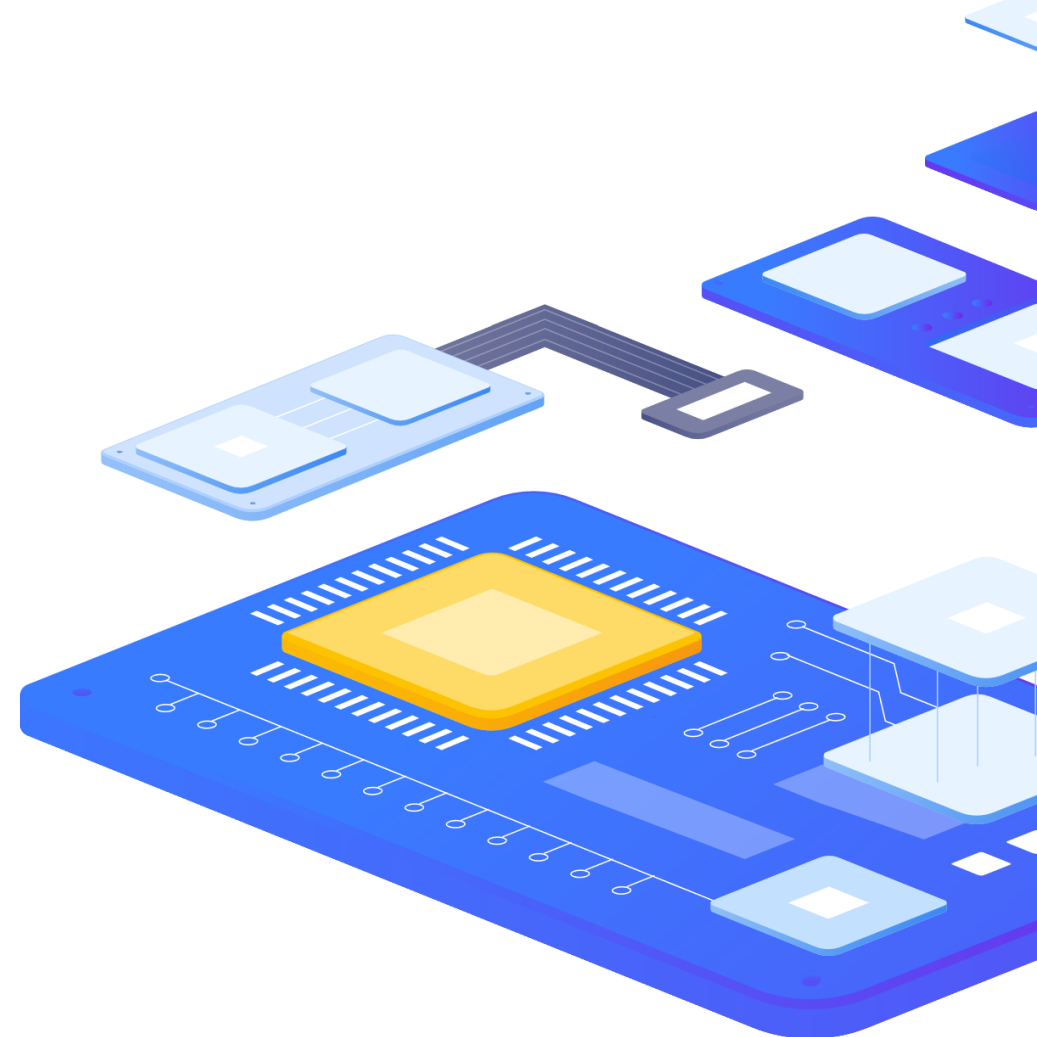
- › JSON type constructors that are compliant with the SQL/JSON standard have been added. The following functions are available:
- › **json\_array**
- › **json\_arrayagg**
- › **json\_object**
- › **json\_objectagg**



## What's new in PostgreSQL 16

# Subquery

- ▶ It is no longer necessary to specify an alias name for subqueries in the FROM clause.
- ▶ This allows aliases for sub-SELECTs and VALUES clauses in the FROM clause to be omitted.
- ▶ This is an extension of the SQL standard, supported by some other database systems, and so eases the transition from such systems, as well as removing the minor inconvenience caused by requiring these aliases.



## What's new in PostgreSQL 16

# Functions

- ▶ Few new functions have been added/enhanced.

For more detailed information consult release notes.

- ▶ **pg\_split\_walfile\_name**

This function returns the sequence number and timeline id from the WAL file name.

```
postgres=# SELECT timeline_id FROM pg_split_walfile_name(pg_walfile_name(pg_current_wal_lsn()));
 timeline_id 
-----
            1
(1 row)
```

5

Misc



# Compression

## › **pg\_basebackup**

It is now possible to specify 'long' for the level of the Zstandard compression method. This option improves the compression ratio for large data but increases resource usage.

**--compress=zstd:long**

## › **pg\_dump**

The --compress option now accepts a compression method and compression level separated by a colon (:). The compression method can be one of none, gzip, lz4 or zstd.

## What's new in PostgreSQL 16

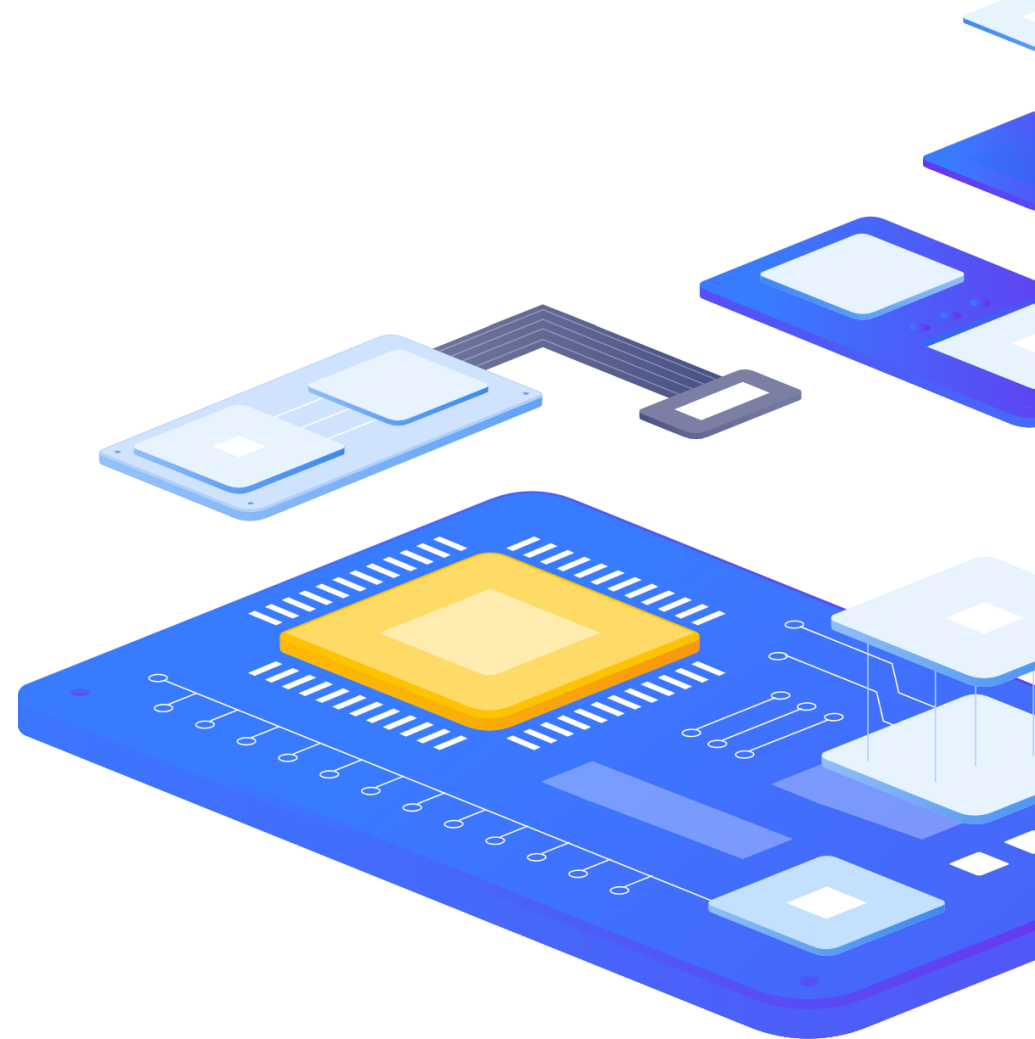
# ICU

- › Build ICU support by default

This removes build flag **--with-icu** and adds flag **--without-icu**.

- › Specify custom rules

Custom rules can now be specified for the ICU locales.



# New Predefined Roles

Role Name	Description
pg_create_subscription	Allows creation of SUBSCRIPTION
pg_maintain	Allows execution of maintenance statements on all tables
pg_use_reserved_connections	Connections assigned to parameter reserved_connections can be used



6

Newly added parameters



# List of newly added parameters

Parameter name	Description (context)	Default value
debug_io_direct	Determine where direct I/O is used (postmaster)	"
createrole_self_grant	Options automatically specified in the GRANT statement (user)	"
enable_presorted_aggregate	Use the sort processing optimization function (user)	on
gss_accept_delegation	Whether to accept GSSAPI authentication delegation from the client (sighup)	off
icu_validation_level	ICU locale check log output level (user)	error
logical_replication_mode	Setting the transfer method for logical replication (user)	buffered

# List of newly added parameters






Parameter name	Description (context)	Default value
max_parallel_apply_workers_per_subscription	Maximum number of workers per subscription (sighup)	2
reserved_connections	Number of connections reserved for general users (postmaster)	0
scram_iterations	SCRAM authentication iteration count (user)	4096
send_abort_for_crash	Send a SIGABRT signal when the backend crashes (sighup)	off
send_abort_for_kill	Send SIGABRT signal when child process is stuck (sighup)	off
vacuum_buffer_usage_limit	Default shared buffer size used by VACUUM (user)	256kB

## What's new in PostgreSQL 16

# Certified training and Support

### Základní certifikovaná školení




### Pokročilá certifikovaná školení

Základní certifikovaná školení	Základní certifikovaná školení	Základní certifikovaná školení	Pokročilá certifikovaná školení	Pokročilá certifikovaná školení
 <p><b>Základy SQL</b></p> <p>Tento kurz byl navržen pro lidi, kteří se chtějí seznámit s SQL. Na konkrétních příkladech z každodenní praxe se naučíte jak SQL funguje a jak správně psát SQL dotazy.</p> <p>4 DNY</p> <p><b>48 000,-</b> Bez DPH</p>	 <p><b>Úvod do PostgreSQL</b></p> <p>Seznámení se s PostgreSQL, popis základních i pokročilých SQL příkazů potřebných nejen pro každodenní práci s PostgreSQL, a to na praktických příkladech.</p> <p>4 DNY</p> <p><b>48 000,-</b> Bez DPH</p>	 <p><b>PostgreSQL Professional</b></p> <p>Tento kurz poskytuje hluboký vhled do pokročilých témat PostgreSQL, jako je indexování, parametry úložiště, optimalizace, replikace, monitorování a mnoho dalších.</p> <p>3 DNY</p> <p><b>36 000,-</b> Bez DPH</p>	 <p><b>PostgreSQL – správa a ladění výkonu</b></p> <p>Tento kurz je určen převážně pro databázové adminy a systémové správce. Zabývat se zde budeme tématy konfigurace, správy, provozu a ladění výkonu PostgreSQL.</p> <p>4 DNY</p> <p><b>48 000,-</b> Bez DPH</p>	 <p><b>PostgreSQL – vysoká dostupnost a Patroni</b></p> <p>Kurz je určený pro pokročilé uživatele PostgreSQL, které zajímá provozování PostgreSQL v režimu plně automatizované vysoké dostupnosti.</p> <p>3 DNY</p> <p><b>36 000,-</b> Bez DPH</p>
<p>Požadavky: Žádné Možnost absolvovat online kurz: Ano Certifikace: Ano</p>	<p>Požadavky: Žádné Možnost absolvovat online kurz: Ano Certifikace: Ano</p>	<p>Požadavky: Žádné Možnost absolvovat online kurz: Ano Certifikace: Ano</p>	<p>Požadavky: základní znalost PostgreSQL a základní znalost OS Linux Možnost absolvovat online kurz: Ano Certifikace: Ano</p>	<p>Požadavky: pokročilá znalost PostgreSQL a základní znalosti OS Linux Možnost absolvovat online kurz: Ano Certifikace: Ano</p>
<p>Termíny kurzu:</p> <p>Na dotaz 9.–12. 10. 23 5.–8. 2. 24</p>	<p>Termíny kurzu:</p> <p>Na dotaz 23.–26. 10. 23 19.–22. 2. 24</p>	<p>Termíny kurzu:</p> <p>Na dotaz 6.–8. 11. 23 4.–6. 3. 24</p>	<p>Termíny kurzu:</p> <p>Na dotaz 20.–23. 11. 23 18.–21. 3. 24</p>	<p>Termíny kurzu:</p> <p>Na dotaz 4.–6. 12. 23 8.–10. 4. 24</p>
<p><a href="#">REGISTROVAT</a></p> <p><a href="#">Více info a termínů</a></p>	<p><a href="#">REGISTROVAT</a></p> <p><a href="#">Více info a termínů</a></p>	<p><a href="#">REGISTROVAT</a></p> <p><a href="#">Více info a termínů</a></p>	<p><a href="#">REGISTROVAT</a></p> <p><a href="#">Více info a termínů</a></p>	<p><a href="#">REGISTROVAT</a></p> <p><a href="#">Více info a termínů</a></p>

What's new in PostgreSQL 16

# Certified training and Support

Expertní certifikovaná školení

PostgreSQL v Kubernetes	Úvod do PostGIS	Migrace na PostgreSQL
		
<b>PostgreSQL v Kubernetes</b>	<b>Úvod do PostGIS</b>	<b>Migrace na PostgreSQL</b>
Tento kurz obsahuje úvod do samotného Kubernetes a jeho zdrojů, které jsou potřebné ke správě PostgreSQL, a veškerá témata týkající se produkčního provozování PostgreSQL na této platformě.	Tento kurz nabízí úvod do PostGIS a jeho nejdůležitějších funkcí a možností, od typů prostorových dat až po prostorová spojení. A to vše nejen teoreticky, ale i s reálnými datovými soubory.	Toto školení bylo navrženo pro lidi, kteří chtějí přejít na PostgreSQL.
3 DNY	3 DNY	4 DNY
<b>36 000,-</b> Bez DPH	<b>36 000,-</b> Bez DPH	<b>48 000,-</b> Bez DPH
Požadavky: pokročilá znalost PostgreSQL a OS Linux Možnost absolvovat online kurz: Ano Certifikace: Ano	Požadavky: pokročilá znalost PostgreSQL Možnost absolvovat online kurz: Ano Certifikace: Ano	Požadavky: pokročilá znalost PostgreSQL Možnost absolvovat online kurz: Ano Certifikace: Ano
Termíny kurzu: Na dotaz 11.–13. 12. 23 22.–24. 4. 24	Termíny kurzu: Na dotaz 8.–10. 1. 24 13.–15. 5. 24	Termíny kurzu: Na dotaz 22.–25. 1. 24 27.–30. 5. 24
<a href="#">REGISTROVAT</a>	<a href="#">REGISTROVAT</a>	<a href="#">REGISTROVAT</a>
<a href="#">Více info a termínů</a>	<a href="#">Více info a termínů</a>	<a href="#">Více info a termínů</a>

# Certified training and Support



## DBA – PostgreSQL

PostgreSQL je výkonný open-source systém pro správu objektově-relačních databází (ORDBMS). Využívá se pro vývoj aplikací, datové sklady, analýzy a další úlohy náročné na práci s daty. Mezi klíčové vlastnosti PostgreSQL patří výkonný engine, podpora pokročilých datových typů a metod indexování a podpora uložených procedur a triggerů napsaných v různých programovacích jazycích, včetně PL/pgSQL, Tcl a Pythonu. Kromě toho PostgreSQL podporuje spolupráci více verzí (MVCC), což umožňuje současný přístup více uživatelů ke stejným datům bez konfliktů, a nabízí robustní podporu integrity a zabezpečení dat.

[MÁM ZÁJEM O TUTO SLUŽBU →](#)[VIDEO →](#)

Precizní správa databází podložená zkušenostmi



Spolehlivý monitoring a systém notifikací



Stabilita, dostupnost a škálovatelnost



Questions?



## Contact us:

Phone:



+420 800 244 442

Web:



<https://www.initmax.cz>

Email:



[tomas.hermanek@initmax.cz](mailto:tomas.hermanek@initmax.cz)

LinkedIn:



<https://www.linkedin.com/company/initmax>

Twitter:



<https://twitter.com/initmax>

Tomáš Heřmánek:



+420 732 447 184