



# What's new in PostgreSQL 15

1

Improvements in data processing in  
large environments

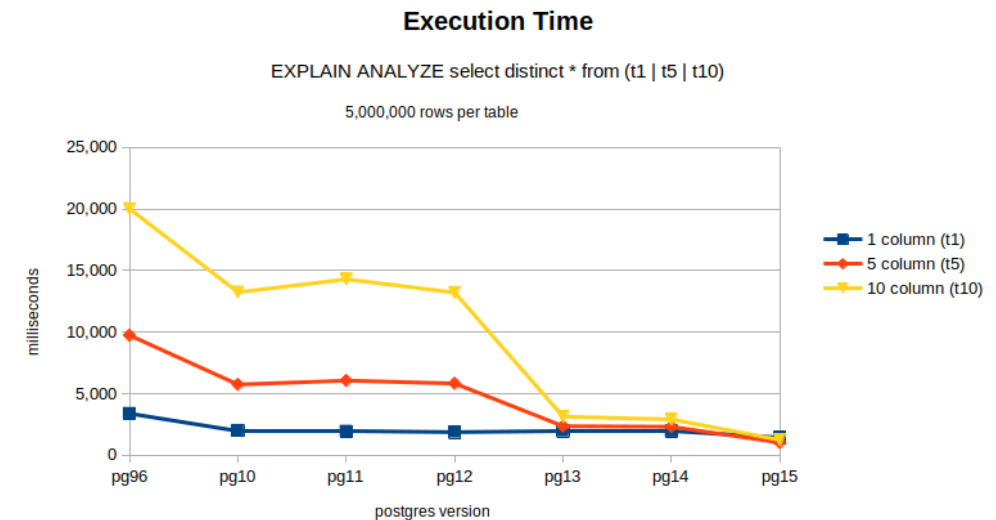


## What's new in PostgreSQL 15

# Performance improvements

Mainly heavily loaded systems will benefit from the following features:

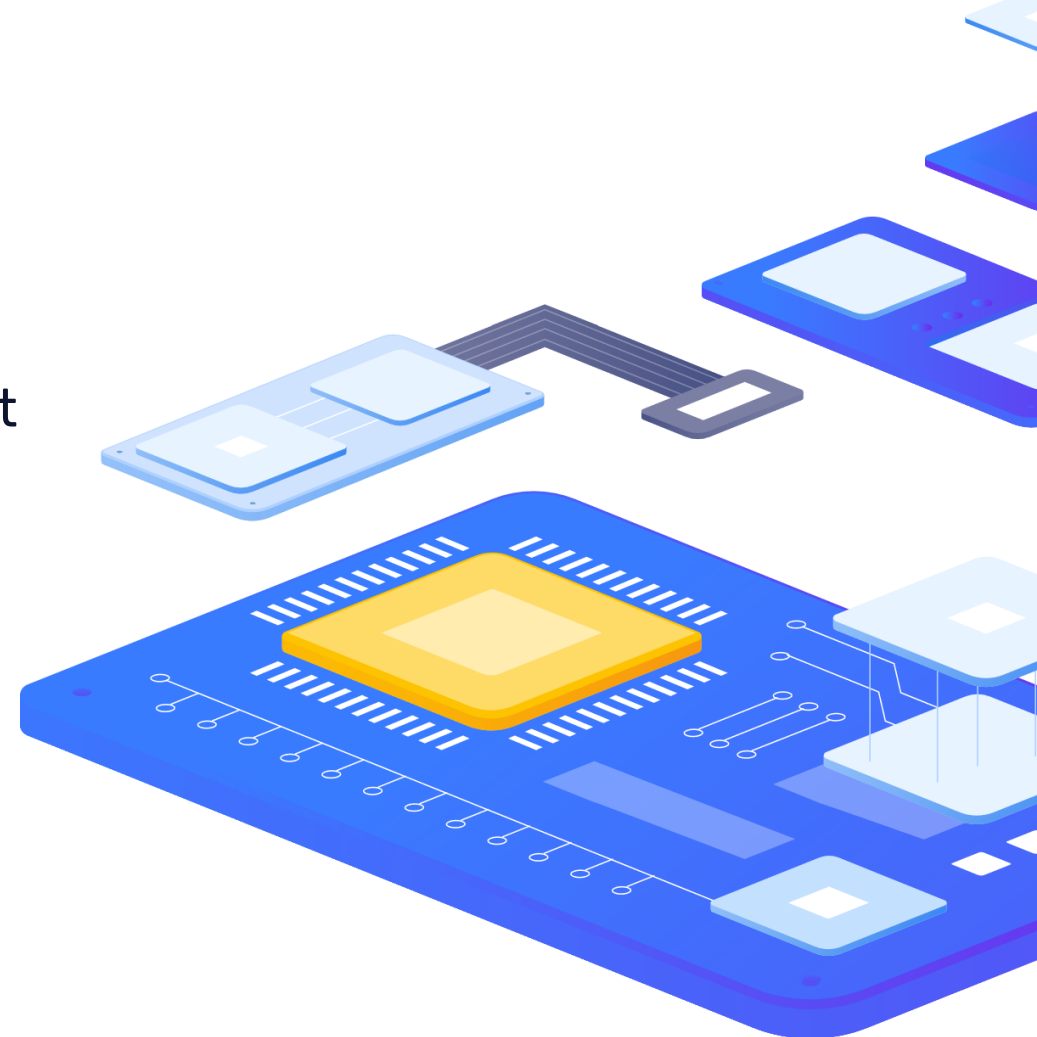
- ▶ The stats collector(`pg_stat_*` views) process has been removed, and performance statistics collection is no longer done over UDP using a single process, but is handled using shared memory.
- ▶ This will make it possible to monitor more statistics in the future, which the previous implementation could no longer handle efficiently.
- ▶ Parallel processing can now be used for SELECT DISTINCT queries.
- ▶ Acceleration of various types of operations + reduction of memory requirements - for example ORDER BY, GROUP BY, Merge Join...



## What's new in PostgreSQL 15

# Replication log reduction

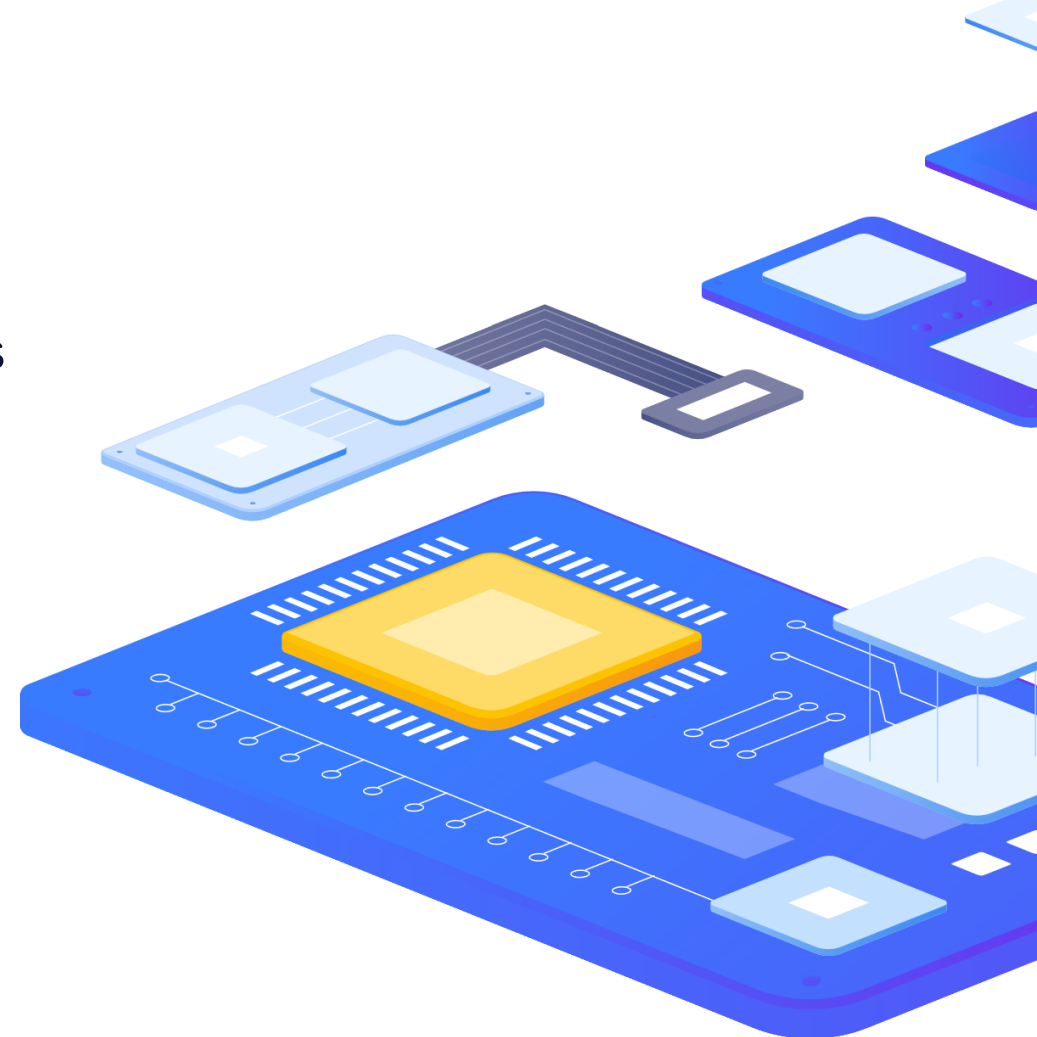
- ▶ PostgreSQL 15 improves crash recovery and physical replication performance for very busy databases by minimizing I/O "jams". How?
- ▶ This is due to a change that allows using GUC `maintenance_io_concurrency` (default value 10) to set how many I/O operations it can start at once.
- ▶ Instead of doing only one I/O operation on a random read.
- ▶ `maintenance_io_concurrency=16`



## What's new in PostgreSQL 15

# Faster WAL archiving

- ▶ The `pgarch_readyXlog()` function scans all files in the `pg_wal/archive_status/` folders to find the next candidate for archiving.
- ▶ This means that for archiving every single file it makes a complete scan of the directory.
- ▶ Newly, a list of 64 items is now loaded (can only be changed at compile time) and the function then uses this "buffer".
- ▶ This leads to a reduction in the number of full scans. Up to a 20x speedup has been reported from the community.



# New compression methods: Zstandard and LZ4

- › Version 14 added support for LZ4 for TOAST compression
- › Zstandard and LZ4 can now be used for WAL compression
  - › `wal_compression=lz4`
- › And also in `pg_basebackup`. Both the compression method and the compression level can be specified in the `--compress` argument.
- › We enter the compression level in the form "level=number" after (:), which is after the name of the compression method.
- › For Zstandard compression, you can also set the "workers=number" option to set up parallel processing.

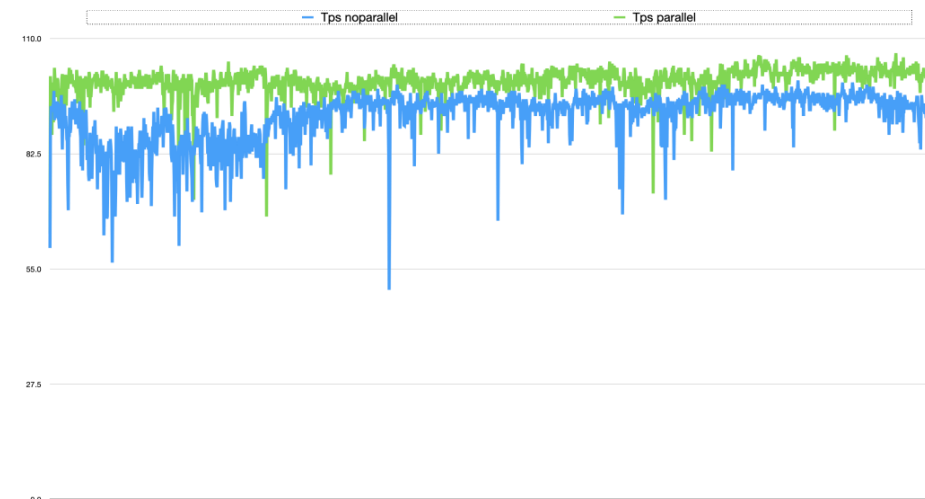
```
pg_basebackup --compress=zstd:level=8 --format=tar -D data
```

## What's new in PostgreSQL 15

# Postgres\_fdw Parallel\_commit

- ▶ Added the `parallel_commit` option, which sets how remote transaction writes are performed.
- ▶ By setting this option to 'on', we enable writing of remote transactions in parallel.
- ▶ The default setting is 'off' which causes transactions to be written serially.

```
ALTER SERVER loopback OPTIONS (ADD parallel_commit 'true');
```



2

Server management improvements

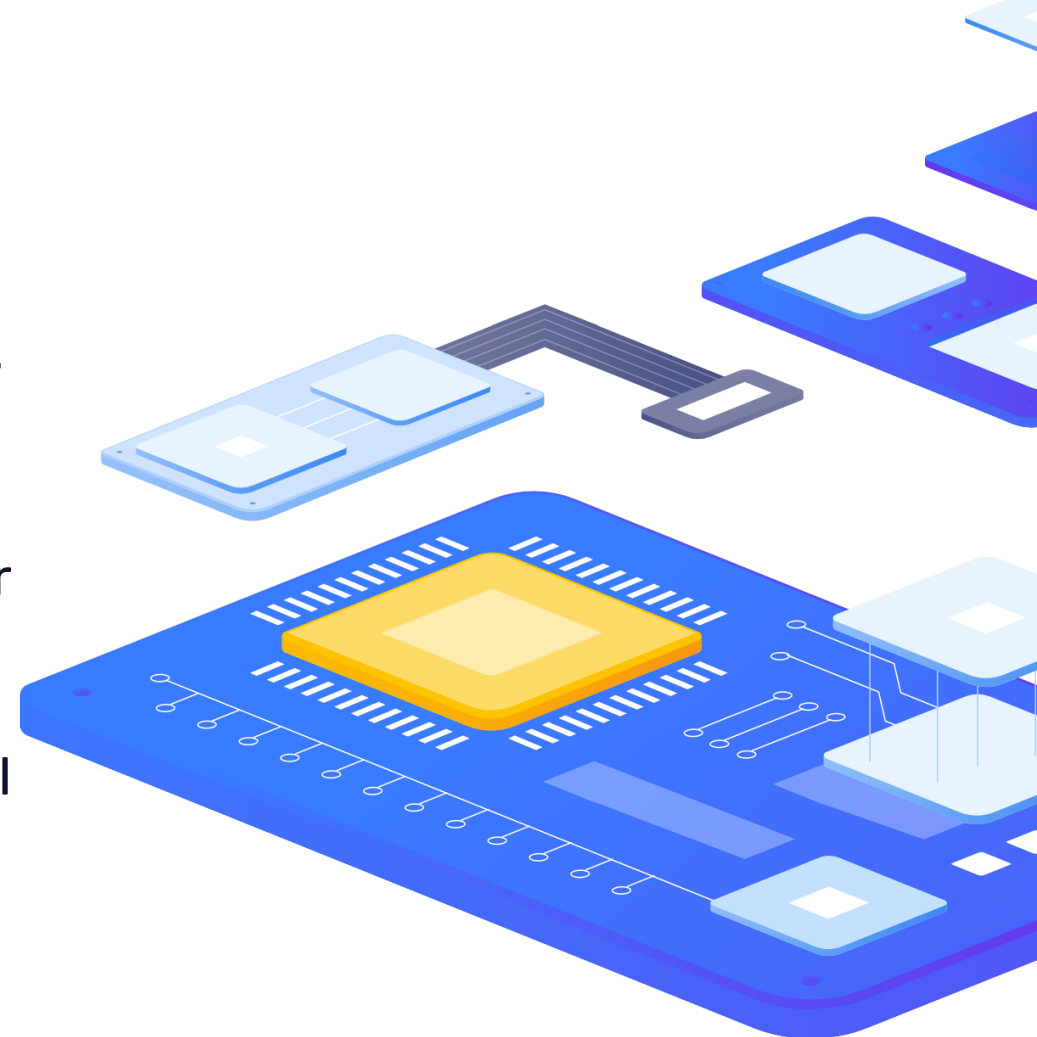




## What's new in PostgreSQL 15

# Library for WAL archiving

- ▶ WAL archiving can now be done using a shared library.
- ▶ It is therefore possible to write an extension to Postgres that will be called for each WAL file to be archived, but the module will always be loaded in RAM.
- ▶ No more forking and shell loading which is insanely expensive.
- ▶ GUC `archive_library` was added for this purpose, so far only the `basic_archive` reference module is implemented (it can only copy a file).
- ▶ We will see different modules in the future, personally I expect one of the first ones to be for `pgBackRest`.



# Global Locale Provider

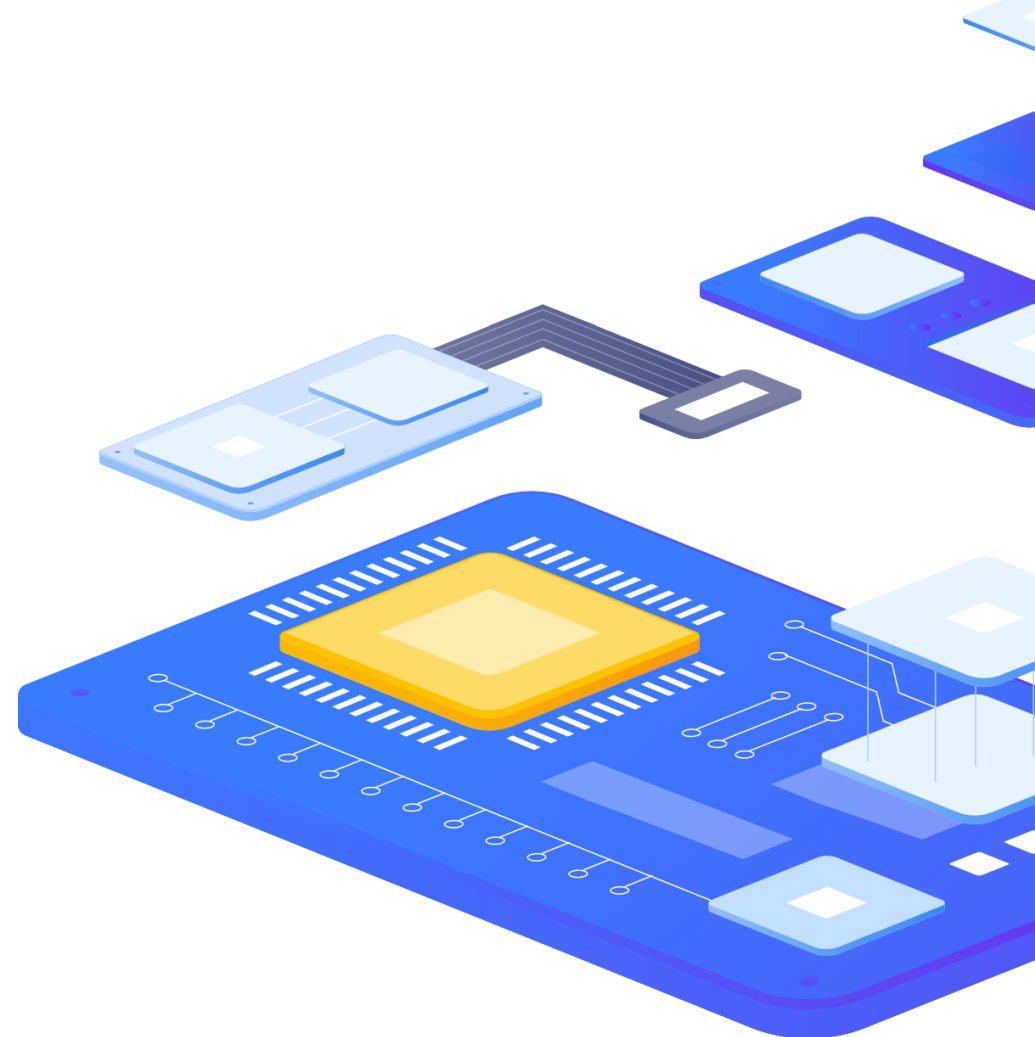
- ▶ The ICU (International Components for Unicode) locale can now be defined globally.
- ▶ It can therefore be defined, for example, for the entire database or even a cluster.
- ▶ The `--locale-provider` argument can now be used in the `initdb` and `createdb` commands.
- ▶ We set the name of the ICU locale with the `--icu-locale` argument.

```
$ initdb --locale-provider=icu --icu-locale=cs-x-icu --encoding=utf8 -D data  
CREATE DATABASE iculocale LOCALE_PROVIDER=icu ICU_LOCALE=cs-x-icu TEMPLATE=template0;
```

## What's new in PostgreSQL 15

# Logging

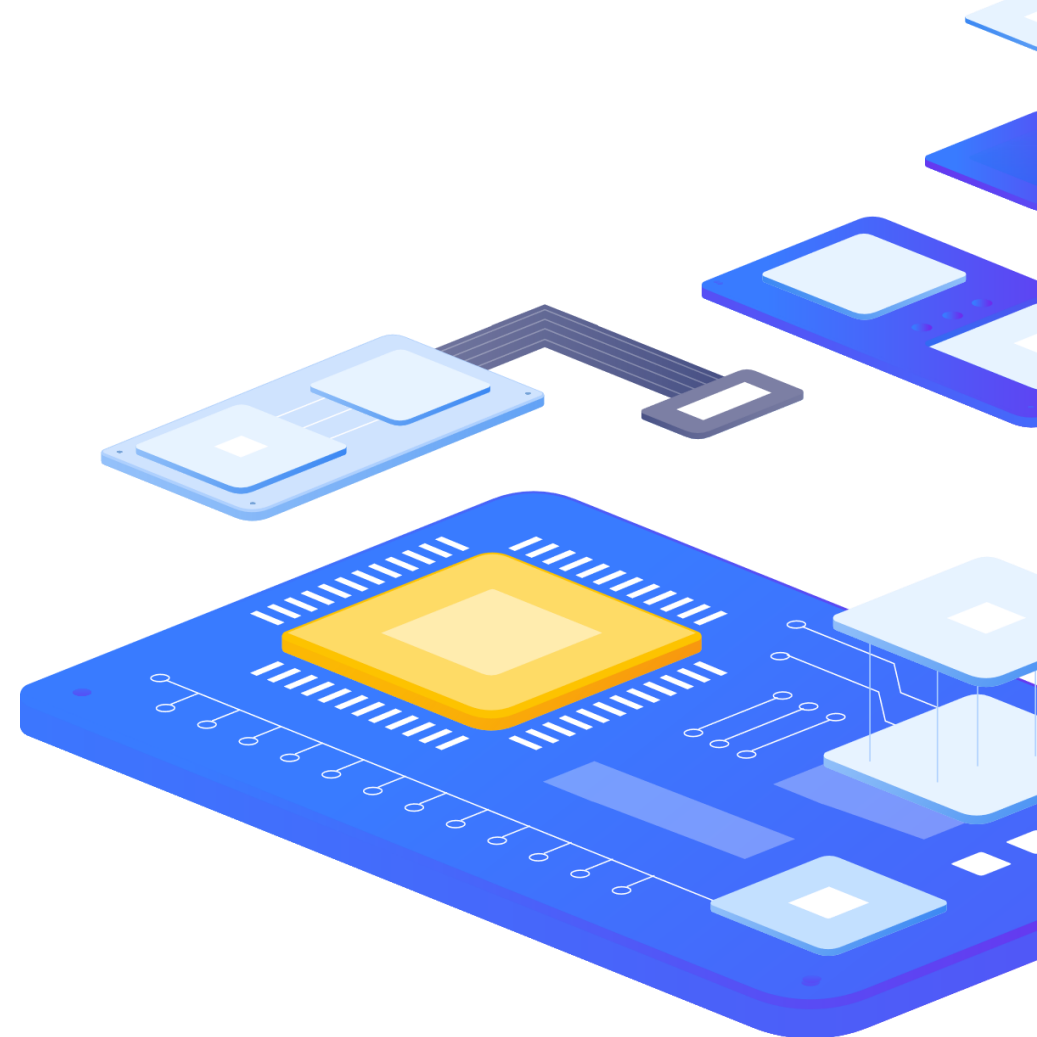
- ▶ Logging can now be done in JSON format. Please note that logging in JSON is much trickier than logging in the standard log format.
- ▶ But if you basically log only errors, then you don't have to worry about it.
- ▶ Checkpoint logging is now enabled by default.
- ▶ Slow autovacuum is now logged by default.
  - ▶ `log_autovacuum_min_duration`



## What's new in PostgreSQL 15

# pg\_stat\_statements

- ▶ The `pg_stat_statements` view has been extended with the following information:
- ▶ Adds information about I/O times when working with temporary files.
- ▶ Another field with information about JIT compilation.
  - ▶ `jit_functions`
  - ▶ `jit_generation_time`
  - ▶ `jit_inlining_count`
  - ▶ `jit_inlining_time`
  - ▶ `jit_optimization_count`
  - ▶ `jit_optimization_time`
  - ▶ `jit_emission_count`
  - ▶ `jit_emission_time`



## What's new in PostgreSQL 15

# pg\_checkpoint role

- ▶ A new predefined role `pg_checkpoint` has been added.
- ▶ The account to which it is assigned allows you to run the `CHECKPOINT` command, which until now was reserved only for the superuser.

```
postgres=# CREATE USER test PASSWORD 'test';
postgres=# \connect postgres test
postgres=> CHECKPOINT;
ERROR: must be superuser or have privileges of pg_checkpoint to do CHECKPOINT
postgres=> \connect postgres postgres
postgres=# GRANT pg_checkpoint TO test;
postgres=# \connect postgres test
postgres=> CHECKPOINT;
CHECKPOINT
```

# Logical replication

- ▶ You can now terminate the SUBSCRIPTION if an error occurs.
- ▶ The `disable_on_error` option can be specified and is disabled by default. This makes it possible not to repeat the same mistake over and over again.
- ▶ The `pg_stat_subscription_stats` view has been added, so it is possible to monitor the status of errors that occur in the subscription worker during logical replication.
- ▶ Logical Replication now enables selective (partial) replication.
- ▶ The conditions for filtering rows can be defined in the WHERE clause of the CREATE PUBLICATION or ALTER PUBLICATION statement.

# Logical replication

- ▶ Only specific columns in the table can be replicated.
- ▶ They can be specified with `CREATE PUBLICATION` or `ALTER PUBLICATION`.
- ▶ All tables in the schema can easily be added to the `PUBLICATION`.
- ▶ This is done by using the `FOR TABLES IN SCHEMA` clause in the `CREATE PUBLICATION` or `ALTER PUBLICATION` statement. Can only be performed by the super user.
- ▶ When changing `PUBLICATION`, do not forget to run `REFRESH PUBLICATION` on the subscriber.
- ▶ `TWO_PHASE` option added to `CREATE_REPLICATION_SLOT` command.
- ▶ `PREPARE TRANSACTION`, `COMMIT PREPARED`, `ROLLBACK PREPARED`, etc. can now be decoded.

## What's new in PostgreSQL 15

# Wait Events

In `pg_stat_activity` we can now come across these wait events.

Wait Event name	Type	Description
ArchiveCleanupCommand	IPC	Waiting for the <code>archive_cleanup_command</code> command to complete
ArchiveCommand	IPC	Waiting for the <code>archive_command</code> command to complete
BaseBackupSync	IO	Waiting for storage synchronization of base backup
BaseBackupWrite	IO	Waiting for base backup write
RecoveryEndCommand	IPC	Waiting for the <code>recovery_end_command</code> command to complete
RestoreCommand	IPC	Waiting for the <code>restore_command</code> command to complete
VersionFileWrite	IO	Waiting for the version file to be written while creating a database
VacuumTruncate	Timeout	Waiting to acquire an exclusive lock to truncate off any empty pages



3

Incompatibility



## What's new in PostgreSQL 15

# Revoke PUBLIC CREATE

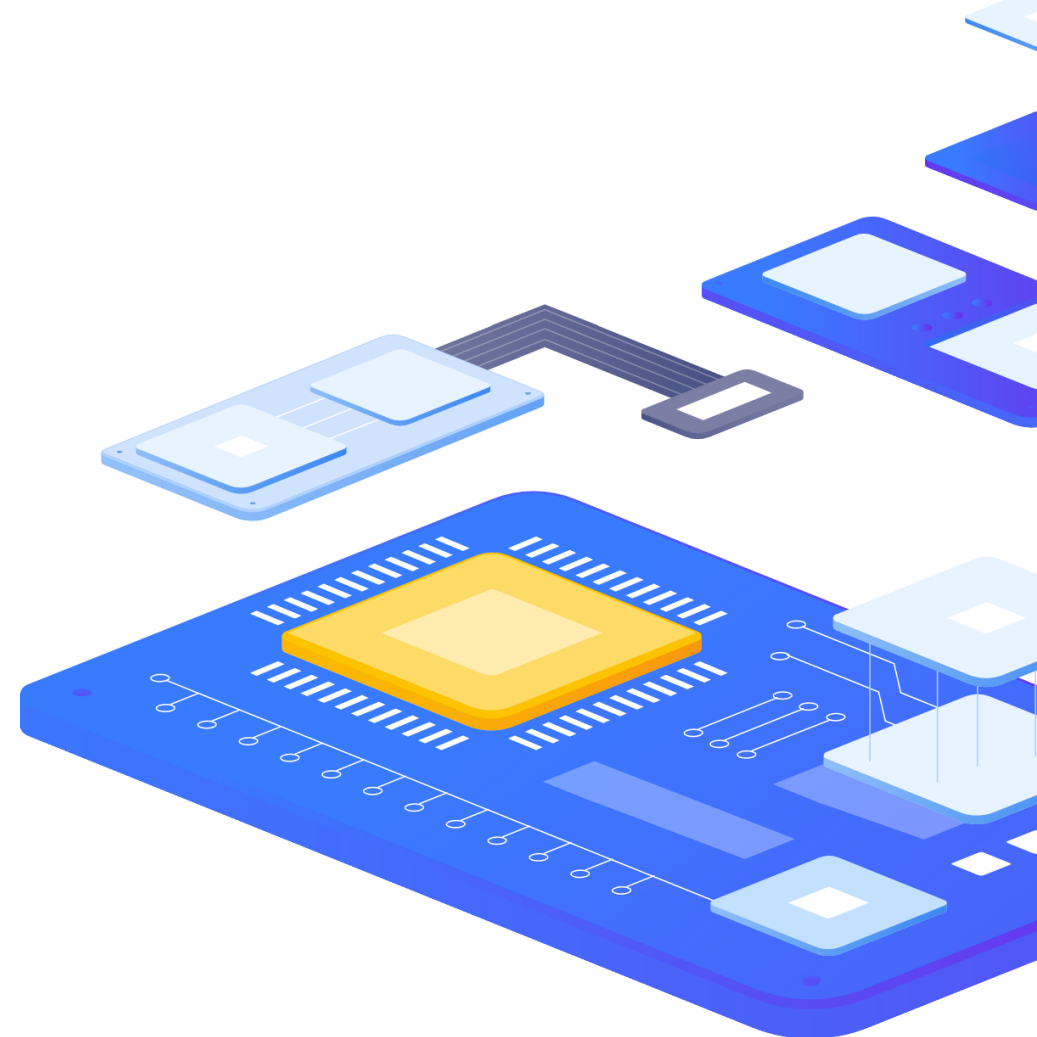
- ▶ Removing the permissions of non-superuser accounts to create tables in the public schema of databases they do not own.

```
$ create database x;  
$ create user test;  
$ create database test with owner test;  
psql -U test -d x -c 'create table a (b int)'  
ERROR: permission denied for schema public  
LINE 1: create table a (b int)  
  
psql -U test -d test -c 'create table a (b int)'  
CREATE TABLE
```

## What's new in PostgreSQL 15

# Miscellaneous

- ▶ the ANALYZE command now uses `maintenance_io_concurrency` instead of `effective_io_concurrency`
- ▶ `pg_dump` / `pg_dumpall` / `pg_upgrade` – does not support an upgrade from versions prior to 9.2
- ▶ Python 2 support has been removed
- ▶ Removed deprecated exclusive backup mode



# Default values change

- ▶ The following parameters have their default values changed

<b>Parameter name</b>	<b>PostgreSQL 14</b>	<b>PostgreSQL 15</b>
hash_mem_multiplier	1.0	2.0
log_autovacuum_min_duration	-1	10min
log_checkpoints	off	on

# Removed parameters

- ▶ The following parameters have been removed

<b>Parameter name</b>	<b>Reason</b>
stats_temp_directory	It was removed due to the shared memory of statistics.

4

SQL

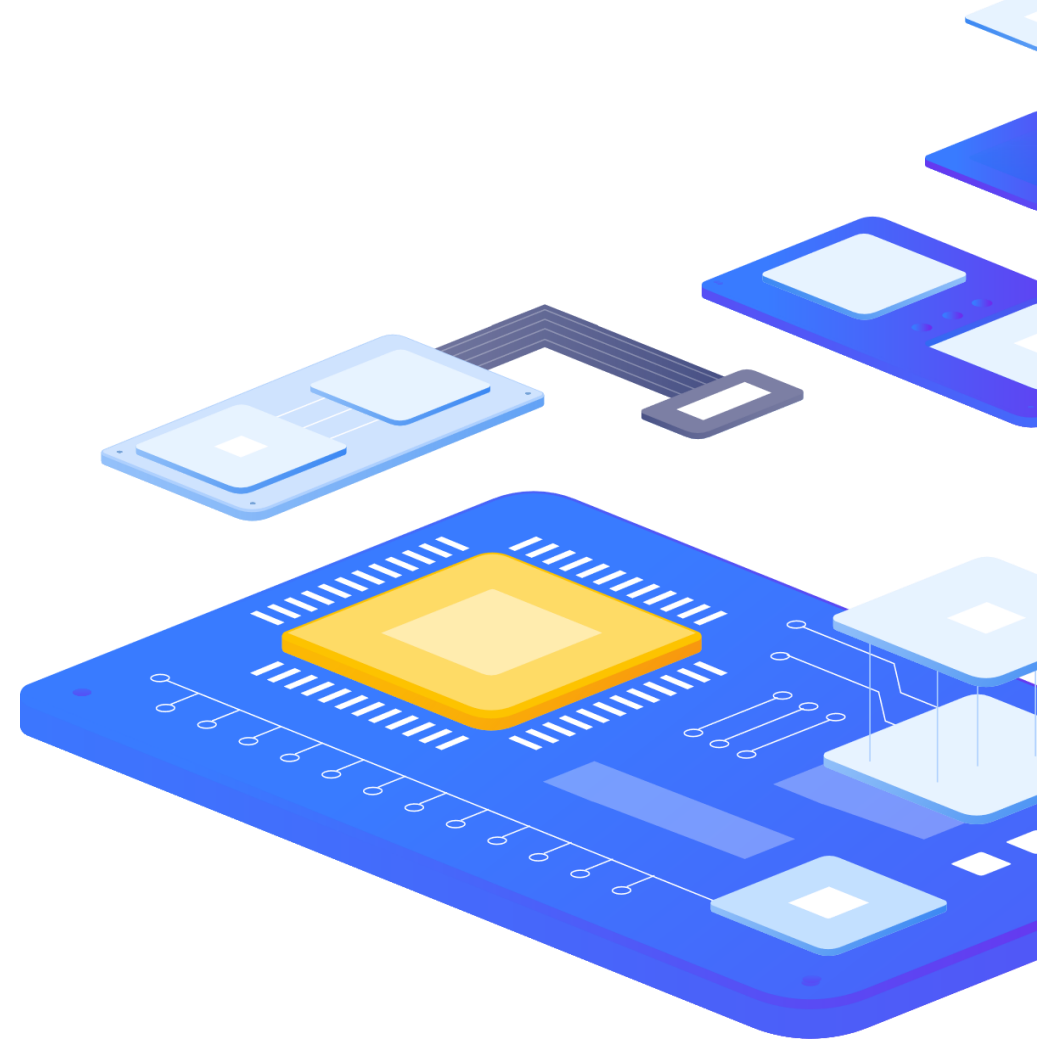


## What's new in PostgreSQL 15

# MERGE

- ▶ The MERGE command now allows you to execute INSERT / DELETE / UPDATE commands at the same time.
- ▶ The implementation of this command will in turn bring Postgres a step closer to commercial databases and hopefully also allow more migration from these commercial databases to Postgres.

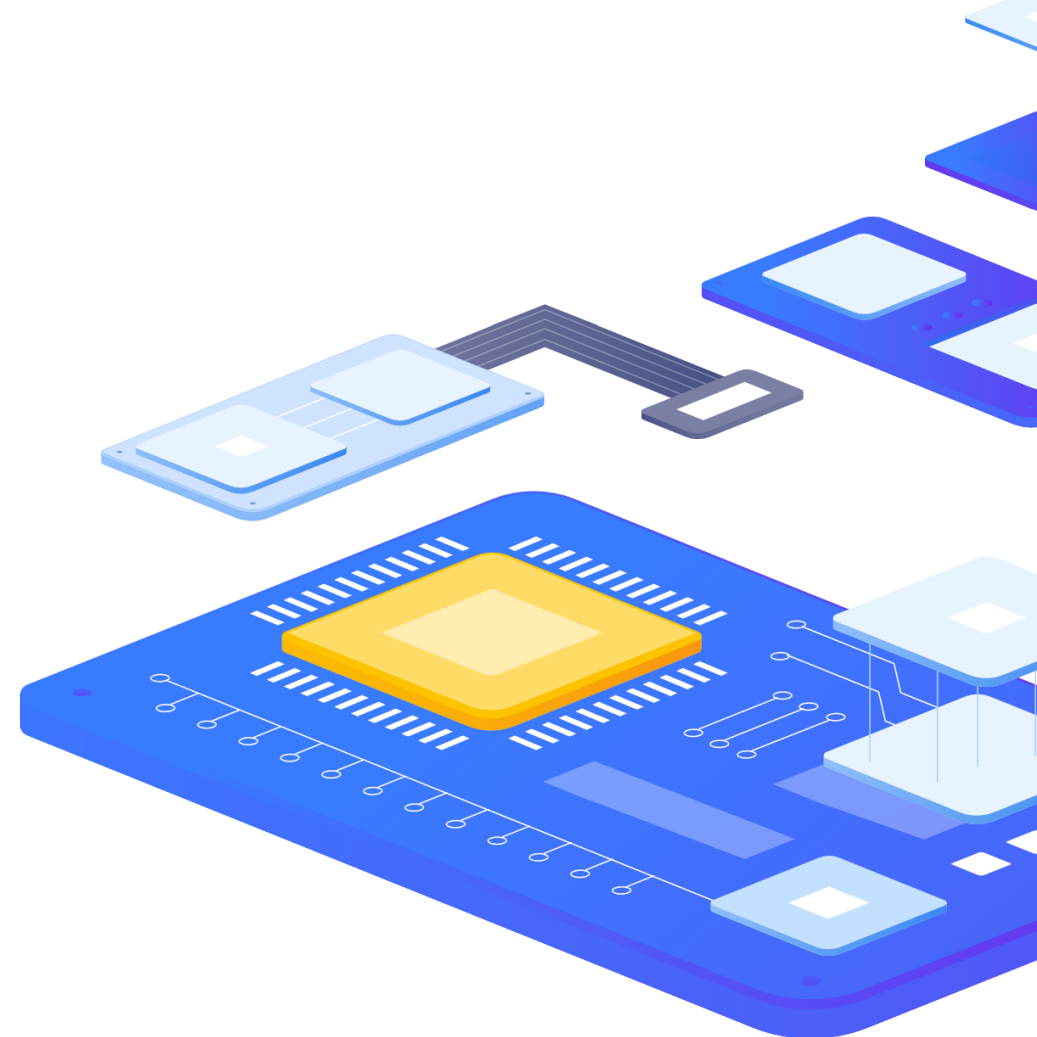
```
MERGE INTO customer_account ca
USING recent_transactions t
ON t.customer_id = ca.customer_id
WHEN MATCHED THEN
  UPDATE SET balance = balance + transaction_value
WHEN NOT MATCHED THEN
  INSERT (customer_id, balance)
VALUES (t.customer_id, t.transaction_value);
```



## What's new in PostgreSQL 15

# CREATE VIEW

- ▶ The `security_invoker` option has also been added for the `CREATE VIEW` command.
- ▶ By default, access to the table on which the view is created is done within the account permissions of the view owner.
- ▶ If the `security_invoker` option is set to `True`, the permissions of the user accessing the view are used when accessing the table.
- ▶ This setting can also be specified in the `ALTER VIEW` statement.



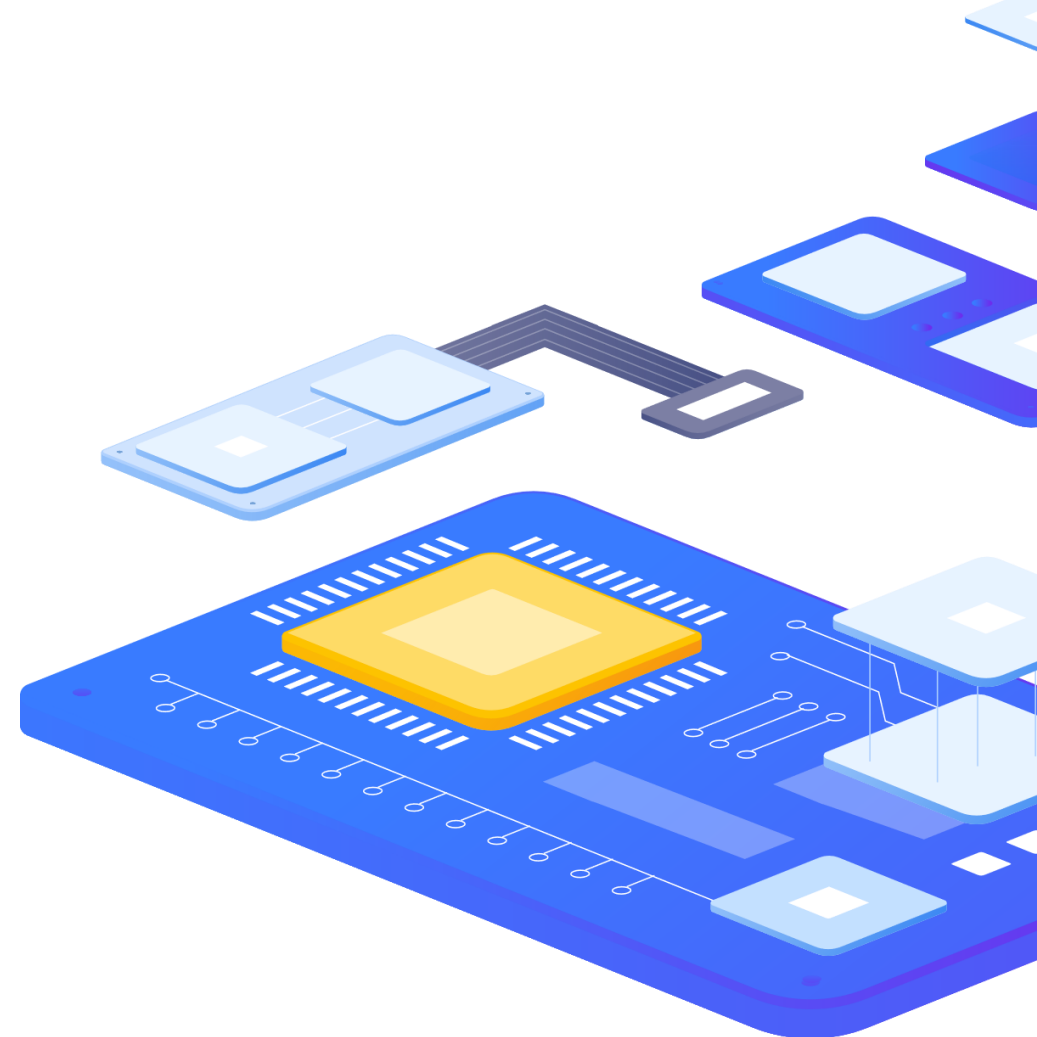


## What's new in PostgreSQL 15

# GRANT

- ▶ Roles can newly be assigned the permission to change a specific GUC using the ALTER SYSTEM command.

```
GRANT ALTER SYSTEM ON PARAMETER log_statement TO test;
```

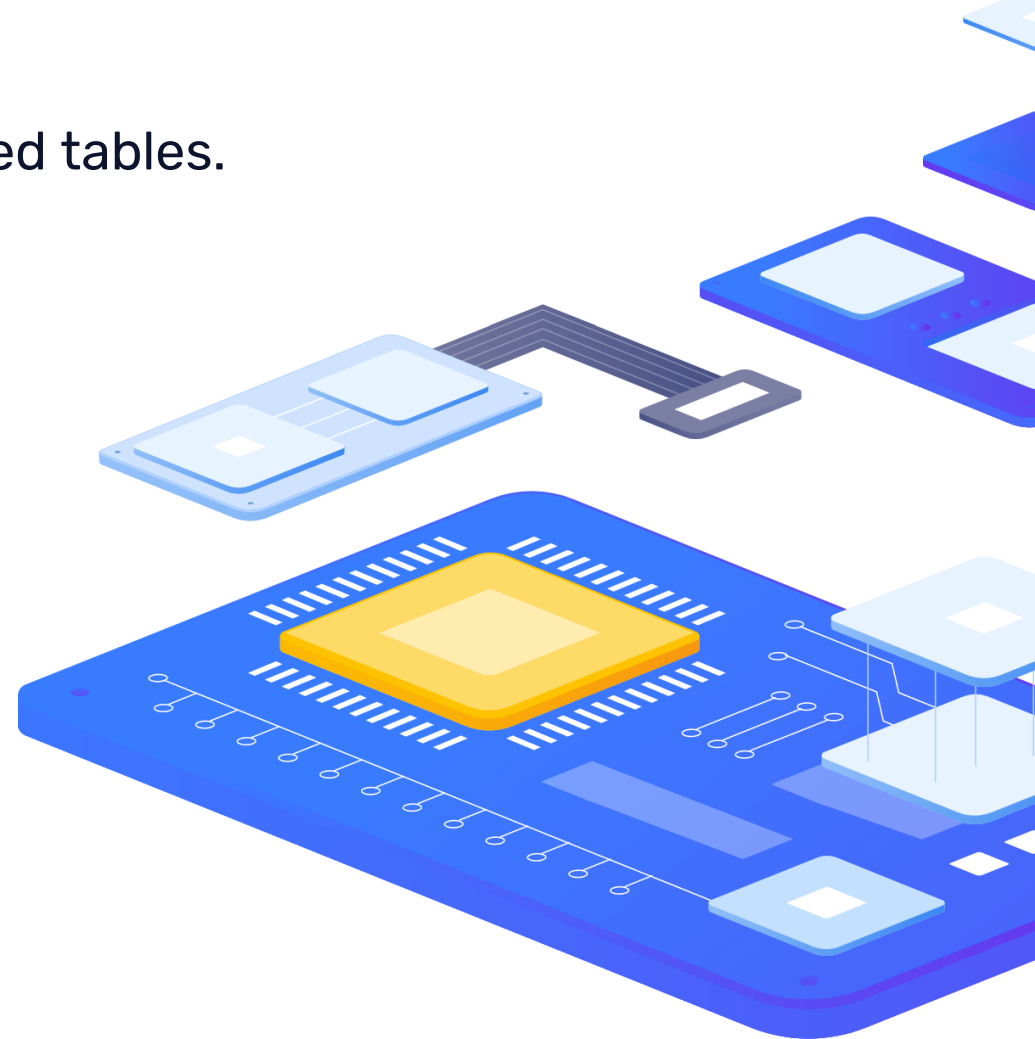


## What's new in PostgreSQL 15

# CLUSTER

- ▶ The CLUSTER command can now be used on partitioned tables.

```
postgres=> CREATE TABLE part1 (c1 INT PRIMARY KEY, c2  
VARCHAR(10)) PARTITION BY LIST(c1);  
CREATE TABLE  
postgres=> CLUSTER part1 USING part1_pkey;  
CLUSTER
```



5

Newly added parameters



## What's new in PostgreSQL 15

# List of new parameters

Parameter name	Description (context)	Default value
<code>allow_in_place_tablespaces</code>	Developer parameters to create a tablespace in the <code>pg_tblspc</code> directory (superuser)	off
<code>archive_library</code>	WAL archive library name (sighup)	-
<code>log_startup_progress_interval</code>	Time before startup process outputs log if the long-running process occurs (sighup)	10s
<code>recovery_prefetch</code>	Whether WAL prefetching is performed during recovery (sighup)	try
<code>recursive_worktable_factor</code>	Multiplier to determine work table size for recursive queries (user)	10
<code>shared_memory_size</code>	The calculated size of main shared memory (internal)	-
<code>shared_memory_size_in_huge_pages</code>	Number of Huge Pages pages used for shared memory (internal)	-
<code>stats_fetch_consistency</code>	Determines the behavior when cumulative statistics are accessed multiple times (user)	cache
<code>wal_decode_buffer_size</code>	Buffer size for WAL decoding (postmaster)	512kB



Questions?



What's new in PostgreSQL 15

## Kontaktujte nás:

Telefon:



+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