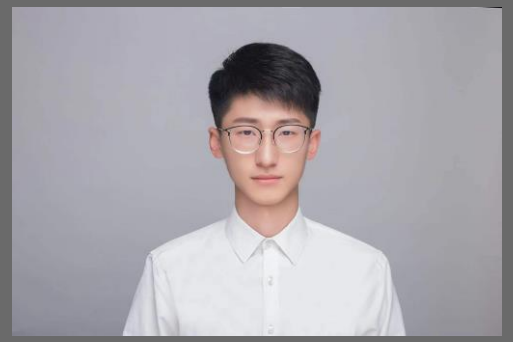
PostgreSQL-15beta1 系统表、视图及等待事件改动

墨天轮

墨天轮 | PostgreSQL中文社区

个人介绍



阎书利

云和恩墨PG技术顾问，曾负责电信、新能源等行业数据库运维，
曾在某银行分布式核心系统项目中负责数据库上线维护工作，最后系统全面投产上线。

2021年获PostgreSQL ACE Partner
2021年获PGFans社区MVP
2021年获Gauss 松鼠会优秀会员之一

PG分会认证讲师、恩墨学院讲师

目前主要从事银行金融行业PostgreSQL数据库、openGauss/MogDB的运维
以及去O迁移等相关工作。

01

PostgreSQL15系统视图改动

System view changes in PostgreSQL15

系统视图改动

视图名	更改方式	视图描述
pg_ident_file_mappings	新增	提供 pg_ident.conf 内容摘要的系统视图
pg_stat_recovery_prefetch	新增	动态统计视图，提供有关恢复期间 WAL 预取的信息
pg_stat_subscription_stats	新增	动态统计视图，在反映到订阅时提供诸如错误数之类的信息
pg_stats_ext	调整	增加继承列
pg_stats_ext_exprs	调整	增加继承列

1.新增pg_ident_file_mappings

pg_ident.conf

```
postgres=# select * from pg_ident_file_mappings;
 line_number | map_name  | sys_name | pg_username | error
-------------+-----------+----------+-------------+-------
          43 | map_xgs_1 | xgs      | ysl         |
(1 row)
```

```
# MAPNAME       SYSTEM-USERNAME         PG-USERNAME
map_xgs_1       xgs                     ysl
"pg_ident.conf" 43L, 1678C
```

```
postgres=# \d pg_ident_file_mappings
        View "pg_catalog.pg_ident_file_mappings"
   Column     |  Type   | Collation | Nullable | Default
-------------+---------+-----------+----------+---------
 line_number | integer |           |          |
 map_name    | text    |           |          |
 sys_name    | text    |           |          |
 pg_username | text    |           |          |
 error       | text    |           |          |
```

类似于 pg_hba_file_rules视图

```
postgres=# select * from pg_hba_file_rules;
 line_number | type  |   database    | user_name |  address  |                 netmask                 | auth_method | options | error
-------------+-------+---------------+-----------+-----------+-----------------------------------------+-------------+---------+-------
          89 | local | {all}         | {all}     |           |                                         | trust       |         |
          91 | host  | {all}         | {all}     | 127.0.0.1 | 255.255.255.255                         | trust       |         |
          93 | host  | {all}         | {all}     | ::1       | ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff | trust       |         |
          96 | local | {replication} | {all}     |           |                                         | trust       |         |
          97 | host  | {replication} | {all}     | 127.0.0.1 | 255.255.255.255                         | trust       |         |
          98 | host  | {replication} | {all}     | ::1       | ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff | trust       |         |
(6 rows)
```

```
CREATE VIEW pg_ident_file_mappings AS
    SELECT * FROM pg_ident_file_mappings() AS A;

REVOKE ALL ON pg_ident_file_mappings FROM PUBLIC;
REVOKE EXECUTE ON FUNCTION pg_ident_file_mappings() FROM PUBLIC;
```

src/backend/catalog/system_views.sql
包含初始化SQL脚本

```
postgres=# select * from pg_views where viewname='pg_ident_file_mappings';
-[ RECORD 1 ]-----------------------------------------------------------------------------------
schemaname | pg_catalog
viewname   | pg_ident_file_mappings
viewowner  | postgres
definition |  SELECT a.line_number,                                                            +
           |     a.map_name,                                                                   +
           |     a.sys_name,                                                                   +
           |     a.pg_username,                                                                +
           |     a.error                                                                       +
           |    FROM pg_ident_file_mappings() a(line_number, map_name, sys_name, pg_username, error);
```

pg_views

pg_ident_file_mappings()
函数

pg_ident_file_mappings
视图

```
postgres=# SELECT * FROM pg_ident_file_mappings();
 line_number | map_name  | sys_name | pg_username | error
-------------+-----------+----------+-------------+-------
          43 | map_xgs_1 | xgs      | ysl         |
(1 row)

postgres=# select * from pg_ident_file_mappings;
 line_number | map_name  | sys_name | pg_username | error
-------------+-----------+----------+-------------+-------
          43 | map_xgs_1 | xgs      | ysl         |
(1 row)
```

墨天轮 PostgreSQL中文社区

src/backend/catalog/system_views.sql

```
CREATE VIEW pg_hba_file_rules AS
    SELECT * FROM pg_hba_file_rules() AS A;

REVOKE ALL ON pg_hba_file_rules FROM PUBLIC;
REVOKE EXECUTE ON FUNCTION pg_hba_file_rules() FROM PUBLIC;

CREATE VIEW pg_ident_file_mappings AS
    SELECT * FROM pg_ident_file_mappings() AS A;

REVOKE ALL ON pg_ident_file_mappings FROM PUBLIC;
REVOKE EXECUTE ON FUNCTION pg_ident_file_mappings() FROM PUBLIC;
```

```
fill_hba_line(Tuplestorestate *, TupleDesc, int, HbaLin...
foreach(line, hba_lines)
MemoryContextDelete(linecxt) 声明
MemoryContextSwitchTo(oldcxt) 声明
MemoryContextDelete(hbacxt) 声明
pg_hba_file_rules(PG_FUNCTION_ARGS)
NUM_PG_IDENT_FILE_MAPPINGS_ATTS
fill_ident_line(Tuplestorestate *, TupleDesc, int, IdentL...
foreach(line, ident_lines)
MemoryContextDelete(linecxt) 声明
MemoryContextSwitchTo(oldcxt) 声明
MemoryContextDelete(identcxt) 声明
pg_ident_file_mappings(PG_FUNCTION_ARGS)
```

src/backend/utils/adt/hbafuncs.c

功能实现部分

pg_ident_file_mappings()

└── fill_ident_view()

其他部分读取pg_ident.conf文件，并用视图记录填充tuplestore

└── fill_ident_line()

构建一行pg_ident_file_mappings视图 → *tuplestore

2.新增pg_stat_recovery_prefetch

pg_stat_get_recovery_prefetch()

新增 recovery_prefetch 参数

官方手册：

recovery_prefetch (enum)

Whether to try to prefetch blocks that are referenced in the WAL that are not yet in the buffer pool, during recovery. Valid values are off, on and try (the default). The setting try enables prefetching only if the operating system provides the posix_fadvise function, which is currently used to implement prefetching. Note that some operating systems provide the function, but it doesn't do anything.

Prefetching blocks that will soon be needed can reduce I/O wait times during recovery with some workloads. See also the wal_decode_buffer_size and maintenance_io_concurrency settings, which limit prefetching activity.

在恢复期间是否尝试预取 WAL 中引用的尚未在缓冲池中的块。有效值为off,on和try（默认值）。该设置try仅在操作系统提供该posix_fadvise()功能时才启用预取，目前用于实现预取。

posix_fadvise()是linux上对文件进行预取的系统调用。预先声明文件数据的访问模式程序，可以使用posix_fadvise()来宣布意图以后以特定的模式访问文件数据，因此允许内核执行适当的优化。

NAME top

posix_fadvise - predeclare an access pattern for file data

SYNOPSIS top

```
#include <fcntl.h>

int posix_fadvise(int fd, off_t offset, off_t len, int advice);
```

Feature Test Macro Requirements for glibc (see feature_test_macros(7)):

posix_fadvise():
_POSIX_C_SOURCE >= 200112L

DESCRIPTION top

Programs can use posix_fadvise() to announce an intention to access file data in a specific pattern in the future, thus allowing the kernel to perform appropriate optimizations.

The *advice* applies to a (not necessarily existent) region starting at *offset* and extending for *len* bytes (or until the end of the file if *len* is 0) within the file referred to by *fd*. The *advice* is not binding; it merely constitutes an expectation on behalf of the application.

Permissible values for *advice* include:

POSIX_FADV_NORMAL
Indicates that the application has no advice to give about its access pattern for the specified data. If no advice

https://man7.org/linux/man-pages/man2/fadvise64.2.html

```
#include <fcntl.h>

int posix_fadvise(int fd, off_t offset, off_t len, int advice);

Feature Test Macro Requirements for glibc (see
feature_test_macros(7)):

    posix_fadvise():
        _POSIX_C_SOURCE >= 200112L
```

POSIX_FADV_NORMAL
表示该应用程序没有建议提供有关其指定的数据访问模式。如果没有意见，给出了一个打开的文件，这是默认的假设。
POSIX_FADV_SEQUENTIAL
该应用程序需要访问指定的数据顺序（与以前高的人读低偏移）。
POSIX_FADV_RANDOM
将指定的数据将会以随机顺序进行访问。
POSIX_FADV_NOREUSE
将指定的数据将只访问一次。
POSIX_FADV_WILLNEED
将指定的数据将在不久的将来访问。
POSIX_FADV_DONTNEED
指定的数据不会在短期内被访问。

Permissible values for *advice* include:

POSIX_FADV_NORMAL
Indicates that the application has no advice to give about its access pattern for the specified data. If no advice is given for an open file, this is the default assumption.

POSIX_FADV_SEQUENTIAL
The application expects to access the specified data sequentially (with lower offsets read before higher ones).

POSIX_FADV_RANDOM
The specified data will be accessed in random order.

POSIX_FADV_NOREUSE
The specified data will be accessed only once.

In kernels before 2.6.18, **POSIX_FADV_NOREUSE** had the same semantics as **POSIX_FADV_WILLNEED**. This was probably a bug; since kernel 2.6.18, this flag is a no-op.

POSIX_FADV_WILLNEED
The specified data will be accessed in the near future.

POSIX_FADV_WILLNEED initiates a nonblocking read of the specified region into the page cache. The amount of data read may be decreased by the kernel depending on virtual memory load. (A few megabytes will usually be fully satisfied, and more is rarely useful.)

POSIX_FADV_DONTNEED
The specified data will not be accessed in the near future.

POSIX_FADV_DONTNEED attempts to free cached pages associated with the specified region. This is useful, for example, while streaming large files. A program may periodically request the kernel to free cached data that has already been used, so that more useful cached pages are not discarded instead.

https://man7.org/linux/man-pages/man2/fadvise64.2.html

```
postgres=# select * from pg_stat_recovery_prefetch;
         stats_reset          | prefetch | hit | skip_init | skip_new | skip_fpw | skip_rep | wal_distance | block_distance | io_depth
------------------------------+----------+-----+-----------+----------+----------+----------+--------------+----------------+----------
 2022-06-07 14:07:43.01979+08 |        0 |   0 |         0 |        0 |        0 |        0 |            0 |              0 |        0
(1 row)

postgres=# select * from pg_stat_get_recovery_prefetch();
         stats_reset          | prefetch | hit | skip_init | skip_new | skip_fpw | skip_rep | wal_distance | block_distance | io_depth
------------------------------+----------+-----+-----------+----------+----------+----------+--------------+----------------+----------
 2022-06-07 14:07:43.01979+08 |        0 |   0 |         0 |        0 |        0 |        0 |            0 |              0 |        0
(1 row)
```

pg_stat_recovery_prefetch视图的列:

列名	类型	描述
stats_reset	timestamp with time zone	带时区的时间戳
prefetch	bigint	因为不在缓冲池中而预取的块的数量
hit	bigint	不预取的块的数量，因为它们已经在缓冲池中
skip_init	bigint	不预取的块的数量，因为它们会被初始化为0
skip_new	bigint	未预取的块的数量，因为它们还不存在
skip_fpw	bigint	由于WAL中包含了全页映像而没有预取的块的数量
skip_rep	bigint	不预取的块的数量，因为它们最近已经被预取了
wal_distance	integer	预取器提前查看的字节数
block_distance	integer	预取器正在查找前面的多少块
io_depth	integer	已经启动了多少次预取但尚未完成

```
/*
 * Counters exposed in shared memory for pg_stat_recovery_prefetch.
 */
typedef struct XLogPrefetchStats
{
    pg_atomic_uint64 reset_time;    /* Time of last reset. */
    pg_atomic_uint64 prefetch;  /* Prefetches initiated. */
    pg_atomic_uint64 hit;       /* Blocks already in cache. */
    pg_atomic_uint64 skip_init; /* Zero-inited blocks skipped. */
    pg_atomic_uint64 skip_new;  /* New/missing blocks filtered. */
    pg_atomic_uint64 skip_fpw;  /* FPWs skipped. */
    pg_atomic_uint64 skip_rep;  /* Repeat accesses skipped. */

    /* Dynamic values */
    int         wal_distance;   /* Number of WAL bytes ahead. */
    int         block_distance; /* Number of block references ahead. */
    int         io_depth;       /* Number of I/Os in progress. */
} XLogPrefetchStats;
```

3. 新增pg_stat_subscription_stats

```
postgres=# select * from pg_stat_subscription_stats;
 subid | subname | apply_error_count | sync_error_count | stats_reset
-------+---------+-------------------+------------------+-------------
 24607 | sub     |                 2 |                0 |
 24615 | sub2    |                 2 |             2520 |
(2 rows)
```

对于每个订阅，pg_stat_subscription_stats视图将包含一行

ALTER SUBSCRIPTION ... SKIP
disable_on_error

4.原视图改动

pg_stats_ext和pg_stats_ext_exprs视图都各自在视图里加了一个‘inherited’（继承）列

```
postgres=# \d pg_stats_ext
                View "pg_catalog.pg_stats_ext"
         Column         |        Type        | Collation | Nullable | Default
------------------------+--------------------+-----------+----------+---------
 schemaname             | name               |           |          |
 tablename              | name               |           |          |
 statistics_schemaname  | name               |           |          |
 statistics_name        | name               |           |          |
 statistics_owner       | name               |           |          |
 attnames               | name[]             |           |          |
 exprs                  | text[]             |           |          |
 kinds                  | "char"[]           |           |          |
 n_distinct             | pg_ndistinct       | C         |          |
 dependencies           | pg_dependencies    | C         |          |
 most_common_vals       | text[]             |           |          |
 most_common_val_nulls  | boolean[]          |           |          |
 most_common_freqs      | double precision[] |           |          |
 most_common_base_freqs | double precision[] |           |          |

postgres=#
```

```
postgres=#  \d pg_stats_ext
                View "pg_catalog.pg_stats_ext"
         Column         |        Type        | Collation | Nullable | Default
------------------------+--------------------+-----------+----------+---------
 schemaname             | name               |           |          |
 tablename              | name               |           |          |
 statistics_schemaname  | name               |           |          |
 statistics_name        | name               |           |          |
 statistics_owner       | name               |           |          |
 attnames               | name[]             |           |          |
 exprs                  | text[]             |           |          |
 kinds                  | "char"[]           |           |          |
 inherited              | boolean            |           |          |
 n_distinct             | pg_ndistinct       | C         |          |
 dependencies           | pg_dependencies    | C         |          |
 most_common_vals       | text[]             |           |          |
 most_common_val_nulls  | boolean[]          |           |          |
 most_common_freqs      | double precision[] |           |          |
 most_common_base_freqs | double precision[] |           |          |
```

```
postgres=# \d pg_stats_ext_exprs
            View "pg_catalog.pg_stats_ext_exprs"
         Column          |   Type   | Collation | Nullable | Default
-------------------------+----------+-----------+----------+---------
 schemaname              | name     |           |          |
 tablename               | name     |           |          |
 statistics_schemaname   | name     |           |          |
 statistics_name         | name     |           |          |
 statistics_owner        | name     |           |          |
 expr                    | text     |           |          |
 null_frac               | real     |           |          |
 avg_width               | integer  |           |          |
 n_distinct              | real     |           |          |
 most_common_vals        | anyarray |           |          |
 most_common_freqs       | real[]   |           |          |
 histogram_bounds        | anyarray |           |          |
 correlation             | real     |           |          |
 most_common_elems       | anyarray |           |          |
 most_common_elem_freqs  | real[]   |           |          |
 elem_count_histogram    | real[]   |           |          |
```

```
postgres=# \d pg_stats_ext_exprs
            View "pg_catalog.pg_stats_ext_exprs"
         Column          |   Type   | Collation | Nullable | Default
-------------------------+----------+-----------+----------+---------
 schemaname              | name     |           |          |
 tablename               | name     |           |          |
 statistics_schemaname   | name     |           |          |
 statistics_name         | name     |           |          |
 statistics_owner        | name     |           |          |
 expr                    | text     |           |          |
 inherited               | boolean  |           |          |
 null_frac               | real     |           |          |
 avg_width               | integer  |           |          |
 n_distinct              | real     |           |          |
 most_common_vals        | anyarray |           |          |
 most_common_freqs       | real[]   |           |          |
 histogram_bounds        | anyarray |           |          |
 correlation             | real     |           |          |
 most_common_elems       | anyarray |           |          |
 most_common_elem_freqs  | real[]   |           |          |
 elem_count_histogram    | real[]   |           |          |
```

pg_stats_ext和pg_stats_ext_exprs视图通过
src/backend/catalog/system_views.sql
可以发现,inherited列是依照pg_statistic_ext_data系统表的stxdinherit列而来的。这个列也是pg_statistic_ext_data系统表在PG-15beta1新增的一列。是一个boolean型，如果为真，则统计信息包括继承子列。

pg_statistic_ext_data.stxdinherit

↓

pg_stats_ext.inherited

pg_stats_ext_exprs.inherited

```
CREATE VIEW pg_stats_ext WITH (security_barrier) AS
    SELECT cn.nspname AS schemaname,
           c.relname AS tablename,
           sn.nspname AS statistics_schemaname,
           s.stxname AS statistics_name,
           pg_get_userbyid(s.stxowner) AS statistics_owner,
           ( SELECT array_agg(a.attname ORDER BY a.attnum)
             FROM unnest(s.stxkeys) k
                  JOIN pg_attribute a
                       ON (a.attrelid = s.stxrelid AND a.attnum = k)
           ) AS attnames,
           pg_get_statisticsobjdef_expressions(s.oid) as exprs,
           s.stxkind AS kinds,
           sd.stxdinherit AS inherited,
           sd.stxdndistinct AS n_distinct,
           sd.stxddependencies AS dependencies,
           m.most_common_vals,
           m.most_common_val_nulls,
           m.most_common_freqs,
           m.most_common_base_freqs
    FROM pg_statistic_ext s JOIN pg_class c ON (c.oid = s.stxrelid)
         JOIN pg_statistic_ext_data sd ON (s.oid = sd.stxoid)
```

02

PostgreSQL15系统表改动

System table changes in PostgreSQL15

系统表改动

表名	更改方式	视图描述
pg_parameter_acl	新增	一个系统表，记录一个或多个角色被授权的配置参数
pg_publication_namespace	新增	记录数据库中模式和发布之间的映射的系统表
pg_collation	调整	更改 collcollate 列和 collctype 列的类型（name → text）；添加 colliculocale 列
pg_constraint	调整	添加 confdelsetcols 列
pg_database	调整	更改 datcollate 列、datctype 列的类型（name → text）；添加 datcollversion 列、daticulocale 列、datlocprovider 列；删除 datlastsysoid 列
pg_index	调整	添加 indnullsnotdistinct 列
pg_publication_rel	调整	添加 prattrs 列和 prqual 列
pg_statistic_ext_data	调整	retype stxdexpr 列（pg_statistic[] 相同，但类型oid 不同）；添加 stxdinherit 列
pg_subscription	调整	添加 subdisableonerr 列、subskiplsn 列、subtwophasestate 列

1.新增pg_parameter_acl

参数可授权给普通用户
alter system set去修改

相应增加视图去记录赋予了哪些用户可以修改哪些参数

```
postgres=# \c postgres ysl
You are now connected to database "postgres" as user "ysl".
postgres=> ALTER SYSTEM SET log_statement = 'all' ;
ERROR:  permission denied to set parameter "log_statement"
postgres=> \c postgres postgres
You are now connected to database "postgres" as user "postgres".
postgres=# GRANT ALTER SYSTEM ON PARAMETER log_statement TO ysl;
GRANT
postgres=# SELECT * FROM pg_parameter_acl ;
  oid  |    parname    |                paracl
-------+---------------+---------------------------------------
 16425 | log_statement | {postgres=sA/postgres,ysl=A/postgres}
(1 row)

postgres=# \c postgres ysl
You are now connected to database "postgres" as user "ysl".
postgres=> ALTER SYSTEM SET log_statement = 'all' ;
ALTER SYSTEM
postgres=> \c postgres postgres
You are now connected to database "postgres" as user "postgres".
postgres=# revoke alter system on parameter log_statement FROM ysl;
REVOKE
postgres=# SELECT * FROM pg_parameter_acl ;
 oid | parname | paracl
-----+---------+--------
(0 rows)

postgres=# \c postgres ysl
You are now connected to database "postgres" as user "ysl".
postgres=> ALTER SYSTEM SET log_statement = 'all' ;
ERROR:  permission denied to set parameter "log_statement"
```

2.新增pg_publication_namespace

15版本新特性——模式发布，支持
FOR ALL TABLE IN SCHEMA语法

当在CREATE PUBLICATION语句中指定FOR ALL TABLE IN SCHEMA子句时，会把信息存储下来，提供publication和schema之间的映射。

```
postgres=# CREATE PUBLICATION pubaa FOR ALL TABLES IN SCHEMA yslaa;
CREATE PUBLICATION
postgres=# select * from pg_publication_namespace ;
  oid  | pnpubid | pnnspid
-------+---------+---------
 24625 |   24624 |   24623
(1 row)
```

```
postgres=# select * from pg_publication_namespace;
  oid  | pnpubid | pnnspid
-------+---------+---------
 24625 |   24624 |   24623
(1 row)

postgres=# select oid,nspname from pg_namespace where nspname='yslaa';
  oid  | nspname
-------+---------
 24623 | yslaa
(1 row)

postgres=# select oid,pubname from  pg_publication where pubname='pubaa';
  oid  | pubname
-------+---------
 24624 | pubaa
(1 row)
```

3.调整 pg_collation

pg_collation：描述可用的排序规则

```
postgres=# \d pg_collation
                Table "pg_catalog.pg_collation"
       Column        |  Type   | Collation | Nullable | Default
---------------------+---------+-----------+----------+---------
 oid                 | oid     |           | not null |
 collname            | name    |           | not null |
 collnamespace       | oid     |           | not null |
 collowner           | oid     |           | not null |
 collprovider        | "char"  |           | not null |
 collisdeterministic | boolean |           | not null |
 collencoding        | integer |           | not null |
 collcollate         | name    |           | not null |
 collctype           | name    |           | not null |
 collversion         | text    | C         |          |
Indexes:
    "pg_collation_oid_index" PRIMARY KEY, btree (oid)
    "pg_collation_name_enc_nsp_index" UNIQUE CONSTRAINT, btree (collname, collencoding, collnamespac
e)
```

```
                Table "pg_catalog.pg_collation"
       Column        |  Type   | Collation | Nullable | Default
---------------------+---------+-----------+----------+---------
 oid                 | oid     |           | not null |
 collname            | name    |           | not null |
 collnamespace       | oid     |           | not null |
 collowner           | oid     |           | not null |
 collprovider        | "char"  |           | not null |
 collisdeterministic | boolean |           | not null |
 collencoding        | integer |           | not null |
 collcollate         | text    | C         |          |
 collctype           | text    | C         |          |
 colliculocale       | text    | C         |          |
 collversion         | text    | C         |          |
Indexes:
    "pg_collation_oid_index" PRIMARY KEY, btree (oid)
    "pg_collation_name_enc_nsp_index" UNIQUE CONSTRAINT, btree (collname, collencoding, collnamespace)
```

更改 collcollate 列和 collctype 列的类型（name → text）；添加 colliculocale 列，用来表示ICU排序规则名

ICU (International Components for Unicode) 是为软件应用提供Unicode和全球化支持的一套成熟、广泛使用的C/C++、Java和.NET 类库集

```
-[ RECORD 16 ]------+----------------------
oid                 | 12352
collname            | af_ZA.utf8
collnamespace       | 11
collowner           | 10
collprovider        | c
collisdeterministic | t
collencoding        | 6
collcollate         | af_ZA.utf8
collctype           | af_ZA.utf8
colliculocale       |
collversion         | 2.17
```

更改 collcollate 列和 collctype 列的类型（name → text）；

name:定长字符串类型,供系统内部使用的，name类型长度当前定为64字节（63个可用字符加上结束符）。
Text：可以存储任意长度的字符串，最大长度没有限制。

Typcategory：是一种任意的数据类型分类，它被分析器用来决定哪种隐式转换“更好”。S表示字符串类型。
Typispreferred：如果此类型在它的typcategory中是一个更好的转换目标，此列为真。

所以text的类型被分析器认为text是字符串类型里的首选。

```
postgres=# select oid,typname,typlen,typtype,typcategory,typispreferred from pg_type where typcategory = 'S';
  oid  |    typname     | typlen | typtype | typcategory | typispreferred
-------+----------------+--------+---------+-------------+----------------
    19 | name           |     64 | b       | S           | f
    25 | text           |     -1 | b       | S           | t
  1042 | bpchar         |     -1 | b       | S           | f
  1043 | varchar        |     -1 | b       | S           | f
 13338 | character_data |     -1 | d       | S           | f
 13340 | sql_identifier |     64 | d       | S           | f
 13348 | yes_or_no      |     -1 | d       | S           | f
(7 rows)
```

4.调整 pg_constraint

墨天轮 PostgreSQL中文社区

pg_constraint存储表上的检查、主键、唯一、外键和排他约束

```
postgres=# \d pg_constraint
                Table "pg_catalog.pg_constraint"
     Column     |     Type     | Collation | Nullable | Default
----------------+--------------+-----------+----------+---------
 oid            | oid          |           | not null |
 conname        | name         |           | not null |
 connamespace   | oid          |           | not null |
 contype        | "char"       |           | not null |
 condeferrable  | boolean      |           | not null |
 condeferred    | boolean      |           | not null |
 convalidated   | boolean      |           | not null |
 conrelid       | oid          |           | not null |
 contypid       | oid          |           | not null |
 conindid       | oid          |           | not null |
 conparentid    | oid          |           | not null |
 confrelid      | oid          |           | not null |
 confupdtype    | "char"       |           | not null |
 confdeltype    | "char"       |           | not null |
 confmatchtype  | "char"       |           | not null |
 conislocal     | boolean      |           | not null |
 coninhcount    | integer      |           | not null |
 connoinherit   | boolean      |           | not null |
 conkey         | smallint[]   |           |          |
 confkey        | smallint[]   |           |          |
 conpfeqop      | oid[]        |           |          |
 conppeqop      | oid[]        |           |          |
 conffeqop      | oid[]        |           |          |
 confdelsetcols | smallint[]   |           |          |
 conexclop      | oid[]        |           |          |
 conbin         | pg_node_tree | C         |          |
Indexes:
    "pg_constraint_oid_index" PRIMARY KEY, btree (oid)
    "pg_constraint_conname_nsp_index" btree (conname, connamespace)
    "pg_constraint_conparentid_index" btree (conparentid)
    "pg_constraint_conrelid_contypid_conname_index" UNIQUE CONSTRAINT, btree (conrelid, contypid, conname
)
    "pg_constraint_contypid_index" btree (contypid)

postgres=#
```

添加 confdelsetcols 列，外键具有ON DELETE子句的列。

```
CREATE TABLE fktable1 (
 tid INT,
 id INT,
 fk_id_del_set_null INT,
 fk_id_del_set_default INT DEFAULT 0,
 FOREIGN KEY (tid, fk_id_del_set_null) REFERENCES pktable1
ON DELETE SET NULL (fk_id_del_set_null),
 FOREIGN KEY (tid, fk_id_del_set_default) REFERENCES pktable1
ON DELETE SET DEFAULT (fk_id_del_set_default)
) ;
```

```
postgres=# select * from pg_constraint where conname like '%fktable1%';
  oid  |                 conname                 | connamespace | contype | condeferrable | condeferre
d | convalidated | conrelid | contypid | conindid | conparentid | confrelid | confupdtype | confdeltyp
e | confmatchtype | conislocal | coninhcount | connoinherit | conkey | confkey | conpfeqop | conppeqop
 | conffeqop | confdelsetcols | conexclop | conbin
-------+-----------------------------------------+--------------+---------+---------------+----------
--+--------------+----------+----------+----------+-------------+-----------+-------------+----------
--+---------------+------------+-------------+--------------+--------+---------+-----------+---------
-+-----------+---------------+-----------+--------
 24630 | fktable1_tid_fk_id_del_set_null_fkey    |         2200 | f       | f             | f
  | t            |    24626 |        0 |    24624 |           0 |     24621 | a           | n
  | s            | t          |           0 | t            | {1,3}  | {1,2}   | {96,96}   | {96,96}
 | {96,96}    | {3}          |           |
 24635 | fktable1_tid_fk_id_del_set_default_fkey |         2200 | f       | f             | f
  | t            |    24626 |        0 |    24624 |           0 |     24621 | a           | d
  | s            | t          |           0 | t            | {1,4}  | {1,2}   | {96,96}   | {96,96}
 | {96,96}    | {4}          |           |
(2 rows)
```

```
postgres=# \d fktable1
                    Table "public.fktable1"
        Column         |  Type   | Collation | Nullable | Default
-----------------------+---------+-----------+----------+---------
 tid                   | integer |           |          |
 id                    | integer |           |          |
 fk_id_del_set_null    | integer |           |          |
 fk_id_del_set_default | integer |           |          | 0
Foreign-key constraints:
    "fktable1_tid_fk_id_del_set_default_fkey" FOREIGN KEY (tid, fk_id_del_set_default) REFERENCES pkta
ble1(tid, id) ON DELETE SET DEFAULT (fk_id_del_set_default)
    "fktable1_tid_fk_id_del_set_null_fkey" FOREIGN KEY (tid, fk_id_del_set_null) REFERENCES pktable1(t
id, id) ON DELETE SET NULL (fk_id_del_set_null)

postgres=# select * from fktable1;
 tid | id | fk_id_del_set_null | fk_id_del_set_default
-----+----+--------------------+-----------------------
(0 rows)
```

5.调整pg_database

墨天轮 PostgreSQL中文社区

```
postgres=# \d pg_database
                Table "pg_catalog.pg_database"
     Column      |   Type    | Collation | Nullable | Default
-----------------+-----------+-----------+----------+---------
 oid             | oid       |           | not null |
 datname         | name      |           | not null |
 datdba          | oid       |           | not null |
 encoding        | integer   |           | not null |
 datcollate      | name      |           | not null |
 datctype        | name      |           | not null |
 datistemplate   | boolean   |           | not null |
 datallowconn    | boolean   |           | not null |
 datconnlimit    | integer   |           | not null |
 datlastsysoid   | oid       |           | not null |
 datfrozenxid    | xid       |           | not null |
 datminmxid      | xid       |           | not null |
 dattablespace   | oid       |           | not null |
 datacl          | aclitem[] |           |          |
Indexes:
    "pg_database_oid_index" PRIMARY KEY, btree (oid), tablespace "pg_global"
    "pg_database_datname_index" UNIQUE CONSTRAINT, btree (datname), tablespace "pg_global"
Tablespace: "pg_global"
```

删除

```
postgres=# \d pg_database
                Table "pg_catalog.pg_database"
     Column      |   Type    | Collation | Nullable | Default
-----------------+-----------+-----------+----------+---------
 oid             | oid       |           | not null |
 datname         | name      |           | not null |
 datdba          | oid       |           | not null |
 encoding        | integer   |           | not null |
 datlocprovider  | "char"    |           | not null |
 datistemplate   | boolean   |           | not null |
 datallowconn    | boolean   |           | not null |
 datconnlimit    | integer   |           | not null |
 datfrozenxid    | xid       |           | not null |
 datminmxid      | xid       |           | not null |
 dattablespace   | oid       |           | not null |
 datcollate      | text      | C         | not null |
 datctype        | text      | C         | not null |
 daticulocale    | text      | C         |          |
 datcollversion  | text      | C         |          |
 datacl          | aclitem[] |           |          |
Indexes:
    "pg_database_oid_index" PRIMARY KEY, btree (oid), tablespace "pg_global"
    "pg_database_datname_index" UNIQUE CONSTRAINT, btree (datname), tablespace "pg_global"
Tablespace: "pg_global"
```

更改 datcollate 列、datctype 列的类型（name → text）；
添加 datcollversion 列、daticulocale 列、datlocprovider 列；
删除 datlastsysoid 列（数据库中最后一个系统OID）

函数 pg_database_collation_actual_version() 报告底层操作系统排序规则版本，
并且 ALTER DATABASE ... REFRESH 设置数据库以匹配操作系统排序规则版本。

6.调整 pg_index

墨天轮 PostgreSQL中文社区

UNIQUE NULLS NOT DISTINCT
允许唯一约束和索引将 NULL 值视为不同的

添加 indnullsnotdistinct 列，

此值仅用于唯一索引。
如果为false，这个唯一索引将认为null值是不同的
(因此索引可以在一个列中包含多个null值，这是PostgreSQL的默认行为)。

如果为真，则将null值视为相等
(因此索引在一列中只能包含一个null值)。

```
postgres=# \d pg_index
                      Table "pg_catalog.pg_index"
       Column         |     Type     | Collation | Nullable | Default
----------------------+--------------+-----------+----------+---------
 indexrelid           | oid          |           | not null |
 indrelid             | oid          |           | not null |
 indnatts             | smallint     |           | not null |
 indnkeyatts          | smallint     |           | not null |
 indisunique          | boolean      |           | not null |
 indnullsnotdistinct  | boolean      |           | not null |
 indisprimary         | boolean      |           | not null |
 indisexclusion       | boolean      |           | not null |
 indimmediate         | boolean      |           | not null |
 indisclustered       | boolean      |           | not null |
 indisvalid           | boolean      |           | not null |
 indcheckxmin         | boolean      |           | not null |
 indisready           | boolean      |           | not null |
 indislive            | boolean      |           | not null |
 indisreplident       | boolean      |           | not null |
 indkey               | int2vector   |           | not null |
 indcollation         | oidvector    |           | not null |
 indclass             | oidvector    |           | not null |
 indoption            | int2vector   |           | not null |
 indexprs             | pg_node_tree | C         |          |
 indpred              | pg_node_tree | C         |          |
Indexes:
    "pg_index_indexrelid_index" PRIMARY KEY, btree (indexrelid)
    "pg_index_indrelid_index" btree (indrelid)
```

7.调整 pg_publication_rel

```
postgres=# \d pg_publication_rel
          Table "pg_catalog.pg_publication_rel"
 Column  |     Type     | Collation | Nullable | Default
---------+--------------+-----------+----------+---------
 oid     | oid          |           | not null |
 prpubid | oid          |           | not null |
 prrelid | oid          |           | not null |
 prqual  | pg_node_tree | C         |          |
 prattrs | int2vector   |           |          |
Indexes:
    "pg_publication_rel_oid_index" PRIMARY KEY, btree (oid)
    "pg_publication_rel_prpubid_index" btree (prpubid)
    "pg_publication_rel_prrelid_prpubid_index" UNIQUE CONSTRAINT, btree (prrelid, prpubid)
```

添加prqual 列和 prattrs列

Prqual:关系的发布限定条件的表达式树(以nodeToString()表示形式)。如果没有发布限定条件，则为空。

Prattrs:这是一个值数组，指示哪些表列是发布的一部分。例如，值为3意味着发布表的第三列。空值表示发布所有列。

```
postgres=# CREATE PUBLICATION pub1 FOR TABLE logical_tab1 where ( id>1);
CREATE PUBLICATION
postgres=# \d logical_tab1
                   Table "public.logical_tab1"
 Column |         Type          | Collation | Nullable | Default
--------+-----------------------+-----------+----------+---------
 id     | integer               |           | not null |
 name   | character varying(20) |           |          |
Indexes:
    "logical_tab1_pkey" PRIMARY KEY, btree (id)
Publications:
    "pub1" WHERE (id > 1)
    "pub_II"
```

```
postgres=# select * from pg_publication_rel;
-[ RECORD 1 ]-------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------
------------
oid     | 24622
prpubid | 24621
prrelid | 24616
prqual  |
prattrs |
-[ RECORD 2 ]-------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------
------------
oid     | 32809
prpubid | 32808
prrelid | 24616
prqual  | {OPEXPR :opno 521 :opfuncid 147 :opresulttype 16 :opretset false :opcollid 0 :inputco
llid 0 :args ({VAR :varno 1 :varattno 1 :vartype 23 :vartypmod -1 :varcollid 0 :varlevelsup 0 :
varnosyn 1 :varattnosyn 1 :location 55} {CONST :consttype 23 :consttypmod -1 :constcollid 0 :co
nstlen 4 :constbyval true :constisnull false :location 58 :constvalue 4 [ 1 0 0 0 0 0 0 0 ]}) :
location 57}
prattrs |
```

8.调整 pg_statistic_ext_data

修改列的type。stxdexpr 列（pg_statistic[] 相同，但类型oid 不同）；
添加stxdinherit 列 。是一个boolean型，如果为真，则统计信息包括继承子列。

```
postgres=# \d pg_statistic_ext_data
              Table "pg_catalog.pg_statistic_ext_data"
      Column       |       Type        | Collation | Nullable | Default
-------------------+-------------------+-----------+----------+---------
 stxoid            | oid               |           | not null |
 stxdinherit       | boolean           |           | not null |
 stxdndistinct     | pg_ndistinct      | C         |          |
 stxddependencies  | pg_dependencies   | C         |          |
 stxdmcv           | pg_mcv_list       | C         |          |
 stxdexpr          | pg_statistic[]    |           |          |
Indexes:
    "pg_statistic_ext_data_stxoid_inh_index" PRIMARY KEY, btree (stxoid, stxdinherit)
```

```
postgres=# select * from pg_type  where typname = 'pg_statistic';
-[ RECORD 1 ]--+-------------
oid            | 12029
typname        | pg_statistic
typnamespace   | 11
typowner       | 10
typlen         | -1
typbyval       | f
typtype        | c
typcategory    | C
typispreferred | f
typisdefined   | t
typdelim       | ,
```

```
postgres=# select * from pg_type  where typname = 'pg_statistic';
-[ RECORD 1 ]--+-------------
oid            | 10029
typname        | pg_statistic
typnamespace   | 11
typowner       | 10
typlen         | -1
typbyval       | f
typtype        | c
typcategory    | C
typispreferred | f
typisdefined   | t
typdelim       | ,
```

9.调整 pg_subscription

```
postgres=# \d pg_subscription
             Table "pg_catalog.pg_subscription"
      Column       |  Type   | Collation | Nullable | Default
-------------------+---------+-----------+----------+---------
 oid               | oid     |           | not null |
 subdbid           | oid     |           | not null |
 subname           | name    |           | not null |
 subowner          | oid     |           | not null |
 subenabled        | boolean |           | not null |
 subbinary         | boolean |           | not null |
 substream         | boolean |           | not null |
 subtwophasestate  | "char"  |           | not null |
 subdisableonerr   | boolean |           | not null |
 subskiplsn        | pg_lsn  |           | not null |
 subconninfo       | text    | C         | not null |
 subslotname       | name    |           |          |
 subsynccommit     | text    | C         | not null |
 subpublications   | text[]  | C         | not null |
Indexes:
    "pg_subscription_oid_index" PRIMARY KEY, btree (oid), tablespace "pg_global"
    "pg_subscription_subname_index" UNIQUE CONSTRAINT, btree (subdbid, subname), tablespace "pg
_global"
Tablespace: "pg_global"
```

添加 subdisableonerr 列、subskiplsn 列、subtwophasestate 列

```
-[ RECORD 2 ]-----+-----------------------------------------------------------------------
oid               | 24615
subdbid           | 5
subname           | sub2
subowner          | 10
subenabled        | t
subbinary         | f
substream         | f
subtwophasestate  | d
subdisableonerr   | f
subskiplsn        | 0/0
subconninfo       | dbname=postgres host=172.20.10.6 port=6000 user=ysla  password=1qaz!QAZ
subslotname       | sub2
subsynccommit     | off
subpublications   | {pub2}
```

03

PostgreSQL15新增等待事件

Added wait event in PostgreSQL15

PostgreSQL数据库里，有着9类等待事件，PostgreSQL15-beta1里种类也没有发生变化

```
/* ---------- * Wait Classes * ---------- */
#define PG_WAIT_LWLOCK 0x01000000U /* 等待LWLock */
#define PG_WAIT_LOCK 0x03000000U /* 等待Lock */
#define PG_WAIT_BUFFER_PIN 0x04000000U /* 等待访问数据缓冲区 */
#define PG_WAIT_ACTIVITY 0x05000000U /* 服务器进程处于空闲状态 */
#define PG_WAIT_CLIENT 0x06000000U /* 等待应用客户端程序在套接字中进行操作 */
#define PG_WAIT_EXTENSION 0x07000000U /* 等待扩展模块中的操作 */
#define PG_WAIT_IPC 0x08000000U /* 等待进程间通信 */
#define PG_WAIT_TIMEOUT 0x09000000U /* 等待达到超时时间 */
#define PG_WAIT_IO 0x0A000000U /* 等待IO操作完成 */
```

主要在IO，IPC（进程间通信），TIMEOUT这三类等待事件上进行了增加。
IO大类新增了三个事件。IPC增加了四个。TIMEOUT类增加了一个。

Wait Event name	更改方式	Type	Description
BaseBackupSync	新增	IO	等待basebackup的存储同步
BaseBackupWrite	新增	IO	正在等待basebackup写入
VersionFileWrite	新增	IO	在创建数据库时等待写入版本文件。
ArchiveCleanupCommand	新增	IPC	等待archive_cleanup_command命令完成
ArchiveCommand	新增	IPC	等待archive_command命令完成
RecoveryEndCommand	新增	IPC	等待recovery_end_command命令完成
RestoreCommand	新增	IPC	等待restore_command命令完成
VacuumTruncate	新增	Timeout	等待获取排他锁以截断任何空类型

新增本地shell命令相关等待事件

墨天轮 观看视频回放

墨天轮

PostgreSQL中文社区