Полезные запросы PostgreSQL

```
--Активные запросы по БД
```

```
SELECT client_addr,usename,datname,now()-query_start AS dur,pid,query,waiting
FROM pg_stat_activity
WHERE datname ILIKE 'vte11'
ORDER BY dur DESC;
```

```
--Статистика по таблице
```

```
SELECT relname, n_tup_ins, n_tup_upd, n_tup_del, n_live_tup, n_dead_tup, last_vacuum,
last_autovacuum, last_analyze, last_autoanalyze
FROM pg_stat_user_tables
WHERE relname = 'as_database_state_list';
```

```
--Блокировки
```

```
SELECT l.pid, l.mode, granted, fastpath, c.relname
FROM pg_locks l
JOIN pg_class c on c.oid = l.relation
ORDER BY 2,5;
```

```
--Висят подготовленные транзакции?
```

```
--получаем gid подготовленной транзакции
SELECT gid, * FROM pg_prepared_xacts;
--откатываем подготовленную транзакцию по gid
ROLLBACK PREPARED gid;
```

```
--PG top
```

```
SELECT now(),tablename, writes, n_tup_ins, n_tup_upd, n_tup_del, n_tup_idx_upd,
idx_scan, seq_scan, seq_tup_read, locks, reltuples
FROM pg_stat_user_tables U
LEFT JOIN (SELECT case WHEN schemaname = 'public' THEN relname ELSE schemaname || '.'
|| relname END tablename,
            schemaname AS T_schemaname, relname AS T_relname, current_database()
dbname,
            (n_tup_ins + n_tup_upd + n_tup_del) writes, (n_tup_upd - n_tup_hot_upd)
n_tup_idx_upd
            FROM pg_stat_user_tables) T ON (T_relname = relname AND T_schemaname =
U.schemaname)
LEFT JOIN (SELECT relname AS L_relname, nspname AS L_schemaname, reltuples, case WHEN
L_locks > 0 THEN L_locks ELSE 0 END locks
            FROM pg_class C
            LEFT JOIN (SELECT relation, COUNT(*) L_locks
                        FROM pg_locks
                        WHERE NOT granted
                        GROUP BY relation) L ON L.relation = C.oid
            LEFT JOIN pg_namespace n ON n.oid = C.relnamespace) L ON (L_relname =
relname AND L_schemaname = U.schemaname)
ORDER BY 12,2;
```

```
--Список всех таблиц
```

```
SELECT * FROM pg_tables WHERE tableowner != 'postgres';
```

```
--Список всех таблиц имеющих записи, но не имеющих индексы
```

```
SELECT c.relname, c.reltuples
FROM pg_class c
LEFT JOIN pg_namespace n ON n.oid = c.relnamespace
LEFT JOIN pg_tablespace t ON t.oid = c.reltablespace
WHERE c.relkind = 'r'
AND pg_get_userbyid(c.relowner) != 'postgres'
AND c.relhasindex IS FALSE
AND c.reltuples != 0
ORDER BY c.reltuples DESC;
```

```
--Список представлений с исходниками
```

```
SELECT table_name,view_definition
FROM information_schema.views
WHERE table_schema NOT IN ('pg_catalog', 'information_schema');
```

```
--Все поля всех таблиц с типами
```

```
SELECT
table_name,column_name,column_default,is_nullable,udt_name||COALESCE('('||character_m
aximum_length||')','') AS type
FROM information_schema.columns
WHERE table_schema NOT IN ('pg_catalog', 'information_schema')
ORDER BY table_name,ordinal_position;
```

```
--Ограничения
```

```
SELECT tc.constraint_name,
tc.constraint_type,
tc.table_name,
kcu.column_name,
tc.is_deferrable,
tc.initially_deferred,
rc.match_option AS match_type,
rc.update_rule AS on_update,
rc.delete_rule AS on_delete,
ccu.table_name AS references_table,
ccu.column_name AS references_field
FROM information_schema.table_constraints tc
LEFT JOIN information_schema.key_column_usage kcu ON tc.constraint_catalog =
kcu.constraint_catalog AND tc.constraint_schema = kcu.constraint_schema AND
tc.constraint_name = kcu.constraint_name
LEFT JOIN information_schema.referential_constraints rc ON tc.constraint_catalog =
rc.constraint_catalog AND tc.constraint_schema = rc.constraint_schema AND
tc.constraint_name = rc.constraint_name
LEFT JOIN information_schema.constraint_column_usage ccu ON
rc.unique_constraint_catalog = ccu.constraint_catalog AND rc.unique_constraint_schema
= ccu.constraint_schema AND rc.unique_constraint_name = ccu.constraint_name
ORDER BY table_name;
```

```
--Список всех функций
```

```
SELECT routine_name
FROM information_schema.routines
WHERE specific_schema NOT IN ('pg_catalog', 'information_schema')
AND type_udt_name != 'trigger';
```

```
--Исходный код всех функций
```

```
SELECT p.proname AS procedure_name,
p.pronargs AS num_args,
t1.typname AS return_type,
l.lanname AS language_type,
p.proargtypes AS argument_types_oids,
prosrc AS body
FROM pg_proc p
LEFT JOIN pg_type t1 ON p.prorettype=t1.oid
LEFT JOIN pg_authid a ON p.proowner=a.oid
LEFT JOIN pg_language l ON p.prolang=l.oid
WHERE a.rolname != 'postgres';
```

```
--Вывести параметры конфигурации отличающиеся от дефолтных
```

```
SELECT p.name, p.source, p.setting,p.short_desc FROM pg_settings p WHERE p.source !=
'default';
```

```
--Вывести индексы которые не используются
```

```
SELECT relname, indexrelname
FROM pg_stat_user_indexes
WHERE idx_scan = 0;
```

```
--Где лежат файлы?
```

```
SELECT current_setting('data_directory') UNION
SELECT current_setting('hba_file') UNION
SELECT current_setting('ident_file')UNION
SELECT current_setting('config_file') UNION
SELECT current_setting('krb_server_keyfile');
```

```
--Сколько места на HDD занимает база?
```

```
SELECT pg_size_pretty(pg_database_size(current_database()));
```

```
--Сколько места на HDD занимает таблица?
```

```
SELECT pg_size_pretty(pg_relation_size('as_database_state_list'));
```

```
--10 самых больших таблиц
```

```
SELECT table_name,pg_size_pretty(pg_relation_size(table_name)) as size
FROM information_schema.tables
WHERE table_schema NOT IN ('information_schema','pg_catalog')
ORDER BY pg_relation_size(table_name) DESC
LIMIT 10;
```

```
--Перечень полей содержащих более 99 процентов значений NULL
```

```
SELECT pgs.tablename,pgs.attname,pgc.reltuples AS "rows",pgs.null_frac * 100 as "% of
NULLs"
FROM pg_catalog.pg_stats pgs
JOIN (SELECT c.relname, c.reltuples,n.nspname
FROM pg_class c
LEFT JOIN pg_namespace n ON n.oid = c.relnamespace
LEFT JOIN pg_tablespace t ON t.oid = c.reltablespace
WHERE c.relkind = 'r'
AND pg_get_userbyid(c.relowner) != 'postgres'
AND c.reltuples != 0) pgc ON pgc.relname = pgs.tablename AND pgs.schemaname =
pgc.nspname
WHERE pgs.null_frac > 0.99
ORDER BY 3 DESC,1;
```

```
--Объем всех баз сервера (например для бэкапа)
```

```
SELECT pg_size_pretty(sum(pg_database_size(datname))) FROM pg_database;
```