

[Przewodnik administratora](#) > Monitorowanie bazy danych

Niniejszy artykuł zawiera polecenia służące monitorowaniu bazy danych. Większość zapytań wykonać można z poziomu psql lub pgAdmin.

Wersja PostgreSQL

Polecenie zwraca wersję serwera PostgreSQL wraz z danymi serwera, na którym został zainstalowany

```
SELECT version();
```

Rozmiar bazy danych

Polecenie zwraca rozmiar bazy danych. Wykorzystanie funkcji `pg_size_pretty` zwiększa czytelność wyniku polecenia.

```
SELECT pg_size_pretty(pg_database_size('edokumenty'));
```

Zmiana nazwy bazy danych

W przypadku zająścia konieczności zmiany nazwy bazy danych należy wykonać takie polecenie (z poziomu połączenia do innej bazy):

```
ALTER DATABASE edokumenty_2 RENAME TO edokumenty);
```

Sprawdzenie ile pamięci RAM zajmuje Postgres

```
#ps -u postgres o pid= | tr -d ' ' | sed 's#.*#/proc/&/smaps#' | xargs sudo grep ^Pss: | awk '{A+=$2} END{print A}'
```

Wynik podawany jest w kB, na podstawie analizy plików smaps

Sprawdzenie aktualnie wykonywanych zapytań

```
postgres@edokumenty:~$psql

SELECT (now() - pg_stat_activity.xact_start) AS age,
pg_stat_activity.datname, pg_stat_activity.procpid,
pg_stat_activity.username, pg_stat_activity.waiting,
pg_stat_activity.query_start, pg_stat_activity.client_addr,
pg_stat_activity.client_port, pg_stat_activity.current_query
FROM pg_stat_activity
WHERE (pg_stat_activity.xact_start IS NOT NULL)
ORDER BY pg_stat_activity.xact_start;
```

Od wersji postgresql 9.2:

```
postgres@edokumenty:~$psql

SELECT (now() - pg_stat_activity.xact_start) AS age,
pg_stat_activity.datname, pg_stat_activity.pid,
pg_stat_activity.username, pg_stat_activity.waiting,
pg_stat_activity.query_start, pg_stat_activity.client_addr,
pg_stat_activity.client_port, pg_stat_activity.query
FROM pg_stat_activity
WHERE (pg_stat_activity.xact_start IS NOT NULL)
ORDER BY pg_stat_activity.xact_start;
```

Sprawdzenie "poziomu zaśmiecenia" tabel w bazie

```
SELECT psut.relname,
       to_char(psut.last_vacuum, 'YYYY-MM-DD HH24:MI') as last_vacuum,
       to_char(psut.last_autovacuum, 'YYYY-MM-DD HH24:MI') as last_autovacuum,
       to_char(pg_class.reltuples, '9G999G999G999') AS n_tup,
       to_char(psut.n_dead_tup, '9G999G999G999') AS dead_tup,
       to_char(CAST(current_setting('autovacuum_vacuum_threshold') AS bigint)
               + (CAST(current_setting('autovacuum_vacuum_scale_factor') AS numeric)
                  * pg_class.reltuples), '9G999G999G999') AS av_threshold,
CASE
  WHEN CAST(current_setting('autovacuum_vacuum_threshold') AS bigint)
        + (CAST(current_setting('autovacuum_vacuum_scale_factor') AS numeric)
           * pg_class.reltuples) < psut.n_dead_tup
  THEN '*'
  ELSE ''
END AS expect_av
FROM pg_stat_user_tables psut
     JOIN pg_class on psut.relid = pg_class.oid
ORDER BY 1;
```