

Домашнее задание к занятию "6.4. PostgreSQL"

Задача 1

Используя docker поднимите инстанс PostgreSQL (версию 13). Данные БД сохраните в volume.

Подключитесь к БД PostgreSQL используя psql.

Воспользуйтесь командой \? для вывода подсказки по имеющимся в psql управляющим командам.

Найдите и приведите управляющие команды для:

вывода списка БД

подключения к БД

вывода списка таблиц

вывода описания содержимого таблиц

выхода из psql

Ответ

вывод с терминала

```
hachiko@hachik-O:~$ sudo -i
```

```
[sudo] пароль для hachiko:
```

```
root@hachik-O:~# docker run --rm --name postgresql-docker -e
```

```
POSTGRES_PASSWORD=12345678 -v bd_data:/var/lib/postgresql/data -p 5432:5432 -d  
postgres:13
```

```
Unable to find image 'postgres:13' locally
```

```
13: Pulling from library/postgres
```

```
a603fa5e3b41: Pull complete
```

```
02d7a77348fd: Pull complete
```

```
16b62ca80c8f: Pull complete
```

```
fbd795da1fe1: Pull complete
```

```
9c68de39d930: Pull complete
```

```
2e441a95082c: Pull complete
```

```
1c97f440fe14: Pull complete
```

```
87a3f78bc5d1: Pull complete
```

```
1314749d2f36: Pull complete
```

```
5e2f643b4476: Pull complete
```

```
74014bfe15af: Pull complete
```

```
568e6d0b637d: Pull complete
```

```
91ce450159ab: Pull complete
```

```
Digest: sha256:3c6a77caf1ef2ae91ef1a2cdc2ae219e65e9ea274fbfa0d44af3ec0fccef0d8d
```

```
Status: Downloaded newer image for postgres:13
```

```
0116f21aedfa5603a4f95026ccde4c096cfa23a531247c1509f6a4c50cf10bfa
```

Подключитесь к БД PostgreSQL используя psql.

```
root@hachik-O:~# docker exec -it postgresql-docker bash
```

```
root@0116f21aedfa:~# psql -U postgres
```

```
psql (13.9 (Debian 13.9-1.pgdg110+1))
```

```
Type "help" for help.
```

Воспользуйтесь командой \? для вывода подсказки по имеющимся в psql управляющим командам.

Найдите и приведите управляющие команды для:

вывода списка БД

postgres=# \?

postgres=# \+

List of databases

| Name | Owner | Encoding | Collate | Ctype | Access privileges | Size |
|---|------------|--|------------|------------|-----------------------|---------|
| Tablespace | | Description | | | | |
| -----+-----+-----+-----+-----+-----+----- | | | | | | |
| ---+-----+-----+-----+-----+-----+----- | | | | | | |
| postgres | postgres | UTF8 | en_US.utf8 | en_US.utf8 | | 7901 |
| kB | pg_default | default administrative connection database | | | | |
| template0 | postgres | UTF8 | en_US.utf8 | en_US.utf8 | =c/postgres | + 7753 |
| kB | pg_default | unmodifiable empty database | | | | |
| | | | | | postgres=CTc/postgres | |
| | | | | | | |
| template1 | postgres | UTF8 | en_US.utf8 | en_US.utf8 | =c/postgres | + 7753 |
| kB | pg_default | default template for new databases | | | | |
| | | | | | postgres=CTc/postgres | |
| | | | | | | |

(3 rows)

postgres=# \conninfo

You are connected to database "postgres" as user "postgres" via socket in "/var/run/postgresql" at port "5432".

postgres=#

вывода списка таблиц

postgres=# \dtS

List of relations

| Schema | Name | Type | Owner |
|-------------------------|--------------------|-------|----------|
| -----+-----+-----+----- | | | |
| pg_catalog | pg_aggregate | table | postgres |
| pg_catalog | pg_am | table | postgres |
| pg_catalog | pg_amop | table | postgres |
| pg_catalog | pg_amproc | table | postgres |
| pg_catalog | pg_attrdef | table | postgres |
| pg_catalog | pg_attribute | table | postgres |
| pg_catalog | pg_auth_members | table | postgres |
| pg_catalog | pg_authid | table | postgres |
| pg_catalog | pg_cast | table | postgres |
| pg_catalog | pg_class | table | postgres |
| pg_catalog | pg_collation | table | postgres |
| pg_catalog | pg_constraint | table | postgres |
| pg_catalog | pg_conversion | table | postgres |
| pg_catalog | pg_database | table | postgres |
| pg_catalog | pg_db_role_setting | table | postgres |
| pg_catalog | pg_default_acl | table | postgres |
| pg_catalog | pg_depend | table | postgres |

| | | |
|-------------------|--------------------------------|------------------|
| <i>pg_catalog</i> | <i>pg_description</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_enum</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_event_trigger</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_extension</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_foreign_data_wrapper</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_foreign_server</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_foreign_table</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_index</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_inherits</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_init_privs</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_language</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_largeobject</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_largeobject_metadata</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_namespace</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_opclass</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_operator</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_opfamily</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_partitioned_table</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_policy</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_proc</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_publication</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_publication_rel</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_range</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_replication_origin</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_rewrite</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_seclabel</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_sequence</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_shdepend</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_shdescription</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_shseclabel</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_statistic</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_statistic_ext</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_statistic_ext_data</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_subscription</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_subscription_rel</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_tablespace</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_transform</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_trigger</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_ts_config</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_ts_config_map</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_ts_dict</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_ts_parser</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_ts_template</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_type</i> | table postgres |
| <i>pg_catalog</i> | <i>pg_user_mapping</i> | table postgres |

(62 rows)

postgres=#

вывода описания содержимого таблиц

postgres=# \dtS+

List of relations

| Schema | Name | Type | Owner | Persistence | Size | Description |
|--------|------|------|-------|-------------|------|-------------|
|--------|------|------|-------|-------------|------|-------------|

| | | | | | | |
|------------|-------------------------|-------|----------|-----------|------------|--|
| pg_catalog | pg_aggregate | table | postgres | permanent | 56 kB | |
| pg_catalog | pg_am | table | postgres | permanent | 40 kB | |
| pg_catalog | pg_amop | table | postgres | permanent | 80 kB | |
| pg_catalog | pg_amproc | table | postgres | permanent | 64 kB | |
| pg_catalog | pg_attrdef | table | postgres | permanent | 8192 bytes | |
| pg_catalog | pg_attribute | table | postgres | permanent | 456 kB | |
| pg_catalog | pg_auth_members | table | postgres | permanent | 40 kB | |
| pg_catalog | pg_authid | table | postgres | permanent | 48 kB | |
| pg_catalog | pg_cast | table | postgres | permanent | 48 kB | |
| pg_catalog | pg_class | table | postgres | permanent | 136 kB | |
| pg_catalog | pg_collation | table | postgres | permanent | 240 kB | |
| pg_catalog | pg_constraint | table | postgres | permanent | 48 kB | |
| pg_catalog | pg_conversion | table | postgres | permanent | 48 kB | |
| pg_catalog | pg_database | table | postgres | permanent | 48 kB | |
| pg_catalog | pg_db_role_setting | table | postgres | permanent | 8192 bytes | |
| pg_catalog | pg_default_acl | table | postgres | permanent | 8192 bytes | |
| pg_catalog | pg_depend | table | postgres | permanent | 488 kB | |
| pg_catalog | pg_description | table | postgres | permanent | 376 kB | |
| pg_catalog | pg_enum | table | postgres | permanent | 0 bytes | |
| pg_catalog | pg_event_trigger | table | postgres | permanent | 8192 bytes | |
| pg_catalog | pg_extension | table | postgres | permanent | 48 kB | |
| pg_catalog | pg_foreign_data_wrapper | table | postgres | permanent | 8192 bytes | |
| pg_catalog | pg_foreign_server | table | postgres | permanent | 8192 bytes | |
| pg_catalog | pg_foreign_table | table | postgres | permanent | 8192 bytes | |
| pg_catalog | pg_index | table | postgres | permanent | 64 kB | |
| pg_catalog | pg_inherits | table | postgres | permanent | 0 bytes | |
| pg_catalog | pg_init_privs | table | postgres | permanent | 56 kB | |
| pg_catalog | pg_language | table | postgres | permanent | 48 kB | |
| pg_catalog | pg_largeobject | table | postgres | permanent | 0 bytes | |
| pg_catalog | pg_largeobject_metadata | table | postgres | permanent | 0 bytes | |
| pg_catalog | pg_namespace | table | postgres | permanent | 48 kB | |
| pg_catalog | pg_opclass | table | postgres | permanent | 48 kB | |
| pg_catalog | pg_operator | table | postgres | permanent | 144 kB | |
| pg_catalog | pg_opfamily | table | postgres | permanent | 48 kB | |
| pg_catalog | pg_partitioned_table | table | postgres | permanent | 8192 bytes | |
| pg_catalog | pg_policy | table | postgres | permanent | 8192 bytes | |
| pg_catalog | pg_proc | table | postgres | permanent | 688 kB | |
| pg_catalog | pg_publication | table | postgres | permanent | 0 bytes | |
| pg_catalog | pg_publication_rel | table | postgres | permanent | 0 bytes | |
| pg_catalog | pg_range | table | postgres | permanent | 40 kB | |
| pg_catalog | pg_replication_origin | table | postgres | permanent | 8192 bytes | |

| | | | | | | |
|------------|-----------------------|-------|----------|-----------|------------|--|
| pg_catalog | pg_rewrite | table | postgres | permanent | 656 kB | |
| pg_catalog | pg_seclabel | table | postgres | permanent | 8192 bytes | |
| pg_catalog | pg_sequence | table | postgres | permanent | 0 bytes | |
| pg_catalog | pg_shdepend | table | postgres | permanent | 40 kB | |
| pg_catalog | pg_shdescription | table | postgres | permanent | 48 kB | |
| pg_catalog | pg_shseclabel | table | postgres | permanent | 8192 bytes | |
| pg_catalog | pg_statistic | table | postgres | permanent | 248 kB | |
| pg_catalog | pg_statistic_ext | table | postgres | permanent | 8192 bytes | |
| pg_catalog | pg_statistic_ext_data | table | postgres | permanent | 8192 bytes | |
| pg_catalog | pg_subscription | table | postgres | permanent | 8192 bytes | |
| pg_catalog | pg_subscription_rel | table | postgres | permanent | 0 bytes | |
| pg_catalog | pg_tablespace | table | postgres | permanent | 48 kB | |
| pg_catalog | pg_transform | table | postgres | permanent | 0 bytes | |
| pg_catalog | pg_trigger | table | postgres | permanent | 8192 bytes | |
| pg_catalog | pg_ts_config | table | postgres | permanent | 40 kB | |
| pg_catalog | pg_ts_config_map | table | postgres | permanent | 56 kB | |
| pg_catalog | pg_ts_dict | table | postgres | permanent | 48 kB | |
| pg_catalog | pg_ts_parser | table | postgres | permanent | 40 kB | |
| pg_catalog | pg_ts_template | table | postgres | permanent | 40 kB | |
| pg_catalog | pg_type | table | postgres | permanent | 120 kB | |
| pg_catalog | pg_user_mapping | table | postgres | permanent | 8192 bytes | |

(62 rows)
postgres=#

выхода из psql

postgres=# \q

root@0116f21aedfa:/#

Задача 2

Используя psql создайте БД test_database.

Изучите бэкап БД.

Восстановите бэкап БД в test_database.

Перейдите в управляющую консоль psql внутри контейнера.

Подключитесь к восстановленной БД и проведите операцию ANALYZE для сбора статистики по таблице.

Используя таблицу pg_stats, найдите столбец таблицы orders с наибольшим средним значением размера элементов в байтах.

Приведите в ответе команду, которую вы использовали для вычисления и полученный результат.

Ответ

вывод с терминала

root@0116f21aedfa:/# **psql -U postgres**

psql (13.9 (Debian 13.9-1.pgdg110+1))

Type "help" for help.

```
postgres=# CREATE DATABASE test_database;  
CREATE DATABASE  
postgres=#
```

Восстановите бэкап БД в test_database

```
root@hachik-O:~# docker cp test_dump.sql postgresql-docker:/tmp  
root@hachik-O:~#  
root@hachik-O:~# docker exec -it postgresql-docker bash  
root@0116f21aedfa:/# psql -U postgres -f /tmp/test_dump.sql test_database  
SET  
SET  
SET  
SET  
SET  
set_config  
-----
```

(1 row)

```
SET  
SET  
SET  
SET  
SET  
SET  
CREATE TABLE  
ALTER TABLE  
CREATE SEQUENCE  
ALTER TABLE  
ALTER SEQUENCE  
ALTER TABLE  
COPY 8  
setval  
-----
```

8

(1 row)

```
ALTER TABLE  
root@0116f21aedfa:/#
```

Перейдите в управляющую консоль psql внутри контейнера

```
root@0116f21aedfa:/# psql -U postgres  
psql (13.9 (Debian 13.9-1.pgdg110+1))  
Type "help" for help.
```

Подключитесь к восстановленной БД и проведите операцию ANALYZE для сбора статистики по таблице.

```
postgres=# \c test_database
```

You are now connected to database "test_database" as user "postgres".

```
test_database=# \dt
```

List of relations

| Schema | Name | Type | Owner |
|--------|--------|-------|----------|
| public | orders | table | postgres |

(1 row)

```
test_database=# ANALYZE VERBOSE public.orders;
```

INFO: analyzing "public.orders"

INFO: "orders": scanned 1 of 1 pages, containing 8 live rows and 0 dead rows; 8 rows in sample, 8 estimated total rows

ANALYZE

Используя таблицу pg_stats, найдите столбец таблицы orders с наибольшим средним значением размера элементов в байтах.

Приведите в ответе команду, которую вы использовали для вычисления и полученный результат.

```
test_database=# SELECT avg_width FROM pg_stats WHERE tablename='orders';
```

```
avg_width
```

```
-----  
4  
16  
4
```

(3 rows)

```
test_database=#
```

Задача 3

Архитектор и администратор БД выяснили, что ваша таблица orders разрослась до невиданных размеров и поиск по ней занимает долгое время. Вам, как успешному выпускнику курсов DevOps в нетологии предложили провести разбиение таблицы на 2 (шардировать на orders_1 - price>499 и orders_2 - price<=499).

Предложите SQL-транзакцию для проведения данной операции.

Можно ли было изначально исключить "ручное" разбиение при проектировании таблицы orders?

Ответ

Предложите SQL-транзакцию для проведения данной операции

```
CREATE TABLE orders_1 (CHECK (price > 499)) INHERITS (orders);  
CREATE TABLE orders_2 (CHECK (price <= 499)) INHERITS (orders);
```

```
CREATE RULE rule_orders_1 AS ON INSERT TO orders WHERE (price > 499)  
DO INSTEAD INSERT INTO orders_1 VALUES (NEW.*);
```

```
CREATE RULE rule_orders_2 AS ON INSERT TO orders WHERE (price <= 499)  
DO INSTEAD INSERT INTO orders_2 VALUES (NEW.*);
```

```
INSERT INTO orders_1 (title, price) (select title, price from orders where price > 499);  
INSERT INTO orders_2 (title, price) (select title, price from orders where price <= 499);
```

Можно ли было изначально исключить "ручное" разбиение при проектировании таблицы orders?

Спроектировать разбиение таблицы путем декларативного партиционирования. Основная таблица в таком случае определяется как "секционированная таблица" и не заполняется данными. Данные пишутся сразу в заранее подготовленные партии.

Задача 4

Используя утилиту pg_dump создайте бэкап БД test_database.

Как бы вы доработали бэкап-файл, чтобы добавить уникальность значения столбца title для таблиц test_database?

Ответ

Используя утилиту pg_dump создайте бэкап БД test_database

```
test_database=# export PGPASSWORD=12345678 && pg_dump -h localhost -U  
postgres test_database > /tmp/test_database_backup.sql  
test_database=#
```

Как бы вы доработали бэкап-файл, чтобы добавить уникальность значения столбца title для таблиц test_database?

Для уникальности можно добавить индекс или первичный ключ.

```
test_database=# CREATE INDEX ON orders ((lower(title)))  
test_database=#
```