Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Дисциплина: Администрирование систем управления базами данных

Лабораторная работа №1

Вариант 332042

Выполнил:
Серебренникова В. В.

Группа:
P33202

Проверил:
Николаев В. В.

Санкт-Петербург
2024

Оглавление

Описание задания

Вариант 332042

Используя сведения из системных каталогов, получить информацию обо всех таблицах, на которые текущий пользователь может выдать права доступа другому заданному пользователю. Полученную информацию представить в виде списка следующего формата:

Текущий пользователь: s100000
Кому выдаём права доступа: s111111

No. Имя таблицы
--- ------------------------
  1 TABLE1
  2 TABLE2
  3 TABLE3
...

Программу оформить в виде процедуры.

Отчет

По заданию необходимую нам информацию (существующие роли, права) необходимо узнать из системных каталогов. Для поиска информации о существующих таблицах для последующего анализа возможной выдачи прав мы можем обратиться к каталогу pg_catalog, а именно к его представлению pg_tables. На рис. 1 показан примерный вывод информации из этого каталога.

```
studs=> select * from pg_tables;
     schemaname     |              tablename              | tableowner | tablespace | hasindexes | hasrules | hastriggers | rowsecurity
--------------------+-------------------------------------+------------+------------+------------+----------+-------------+-------------
 juli               | a                                   | juli       |            | f          | f        | f           | f
 s311718            | my_auth_members                     | s311718    |            | f          | f        | f           | f
 s295964            | planets                             | s295964    |            | t          | f        | t           | f
 s336551            | users                               | s336551    |            | t          | f        | f           | f
 pg_catalog         | pg_statistic                        | postgres   |            | t          | f        | f           | f
 s295964            | expeditions                         | s295964    |            | t          | f        | t           | f
 s295964            | spaceships                          | s295964    |            | t          | f        | t           | f
 s295964            | crews                               | s295964    |            | t          | f        | t           | f
 s295964            | people                              | s295964    |            | t          | f        | t           | f
 s336551            | cities                              | s336551    |            | t          | f        | f           | f
```

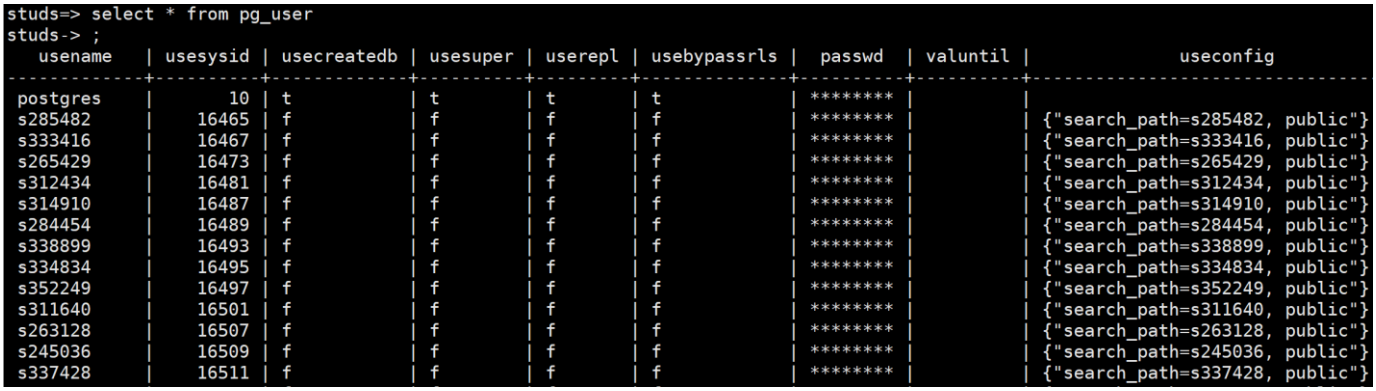
Рис. 1 – pg_tables

Самое полезное для данной задачи представление находится в каталоге information_schema, и это представление - information_schema.role_table_grants. В нем содержится информация обо всех выданных правах на таблицы в формате “кто выдал - кому выдал - каталог таблицы - схема - имя таблицы - тип привилегии (SELECT, INSERT, DELETE, TRUNCATE, UPDATE и так далее) - может ли передать роль - с иерархией”. Пример выведенной информации из представления приведен на рис. 2.

```
studs=> select * from information_schema.role_table_grants;
 grantor | grantee | table_catalog | table_schema |   table_name   | privilege_type | is_grantable | with_hierarchy
---------+---------+---------------+--------------+----------------+----------------+--------------+----------------
 s338828 | s338828 | studs         | s338828      | lightning      | INSERT         | YES          | NO
 s338828 | s338828 | studs         | s338828      | lightning      | SELECT         | YES          | YES
 s338828 | s338828 | studs         | s338828      | lightning      | UPDATE         | YES          | NO
 s338828 | s338828 | studs         | s338828      | lightning      | DELETE         | YES          | NO
 s338828 | s338828 | studs         | s338828      | lightning      | TRUNCATE       | YES          | NO
 s338828 | s338828 | studs         | s338828      | lightning      | REFERENCES     | YES          | NO
 s338828 | s338828 | studs         | s338828      | lightning      | TRIGGER        | YES          | NO
 s338828 | s338828 | studs         | s338828      | rooms          | INSERT         | YES          | NO
 s338828 | s338828 | studs         | s338828      | rooms          | SELECT         | YES          | YES
 s338828 | s338828 | studs         | s338828      | rooms          | UPDATE         | YES          | NO
 s338828 | s338828 | studs         | s338828      | rooms          | DELETE         | YES          | NO
 s338828 | s338828 | studs         | s338828      | rooms          | TRUNCATE       | YES          | NO
 s338828 | s338828 | studs         | s338828      | rooms          | REFERENCES     | YES          | NO
 s338828 | s338828 | studs         | s338828      | rooms          | TRIGGER        | YES          | NO
 s338828 | s338828 | studs         | s338828      | people         | INSERT         | YES          | NO
 s338828 | s338828 | studs         | s338828      | people         | SELECT         | YES          | YES
 s338828 | s338828 | studs         | s338828      | people         | UPDATE         | YES          | NO
 s338828 | s338828 | studs         | s338828      | people         | DELETE         | YES          | NO
 s338828 | s338828 | studs         | s338828      | people         | TRUNCATE       | YES          | NO
 s338828 | s338828 | studs         | s338828      | people         | REFERENCES     | YES          | NO
```

Рис. 2 - information_schema.role_table_grants

Так же нам пригодится информация из представления pg_user каталога pg_catalog, содержащего информацию о пользователях в нашей базе данных. Примерное содержимое представления представлено на рис. 3.



```
studs=> select * from pg_user
studs-> ;
   usename   | usesysid | usecreatedb | usesuper | userepl | usebypassrls |  passwd  | valuntil |              useconfig
-------------+----------+-------------+----------+---------+--------------+----------+----------+-------------------------------
 postgres    |       10 | t           | t        | t       | t            | ******** |          |
 s285482     |    16465 | f           | f        | f       | f            | ******** |          | {"search_path=s285482, public"}
 s333416     |    16467 | f           | f        | f       | f            | ******** |          | {"search_path=s333416, public"}
 s265429     |    16473 | f           | f        | f       | f            | ******** |          | {"search_path=s265429, public"}
 s312434     |    16481 | f           | f        | f       | f            | ******** |          | {"search_path=s312434, public"}
 s314910     |    16487 | f           | f        | f       | f            | ******** |          | {"search_path=s314910, public"}
 s284454     |    16489 | f           | f        | f       | f            | ******** |          | {"search_path=s284454, public"}
 s338899     |    16493 | f           | f        | f       | f            | ******** |          | {"search_path=s338899, public"}
 s334834     |    16495 | f           | f        | f       | f            | ******** |          | {"search_path=s334834, public"}
 s352249     |    16497 | f           | f        | f       | f            | ******** |          | {"search_path=s352249, public"}
 s311640     |    16501 | f           | f        | f       | f            | ******** |          | {"search_path=s311640, public"}
 s263128     |    16507 | f           | f        | f       | f            | ******** |          | {"search_path=s263128, public"}
 s245036     |    16509 | f           | f        | f       | f            | ******** |          | {"search_path=s245036, public"}
 s337428     |    16511 | f           | f        | f       | f            | ******** |          | {"search_path=s337428, public"}
```

Рис. 3 – pg_user

Для подключения к базе введем следующую команду: psql -h pg -d studs .

Ниже приведен код процедуры, созданной в процессе выполнения лабораторной работы.

```
CREATE OR REPLACE PROCEDURE can_assign_role(
first_user text,
second_user text
)
AS
$$
DECLARE
rec record;
num int := 1;
BEGIN
IF NOT EXISTS(
SELECT *
FROM pg_user
WHERE usename = first_user
)
THEN RAISE INFO 'Пользователь % не найден', first_user;
RETURN;
END IF;
IF NOT EXISTS(
SELECT *
FROM pg_user
WHERE usename = second_user
)
THEN RAISE INFO 'Пользователь % не найден', second_user;
RETURN;
END IF;
RAISE INFO 'Текущий пользователь: %', first_user;
RAISE INFO 'Кому выдаём права доступа: %', second_user;
RAISE INFO 'No. Имя таблицы';
RAISE INFO '--- ------------------------------';
FOR rec IN
SELECT DISTINCT table_name as name_of_table
FROM information_schema.role_table_grants as grants
JOIN pg_tables as tables
```

```
ON grants.table_schema = tables.schemaname
AND grants.table_name = tables.tablename
WHERE grantor = first_user
AND is_grantable = 'YES'
LOOP
RAISE INFO '% %', num, rec.name_of_table;
num := num + 1;
END LOOP;
END
$$ LANGUAGE plpgsql;
```

Ниже представлены результаты вызова процедуры с допустимыми пользователями (рис. 4) и недопустимыми (рис. 5).

```
studs=> call can_assign_role('s338828', 's339057');
INFO:  Текущий пользователь: s338828
INFO:  Кому выдаём права доступа: s339057
INFO:  No. Имя таблицы
INFO:  --- ------------------------------
INFO:  1 doors
INFO:  2 doors_in_rooms
INFO:  3 lightning
INFO:  4 people
INFO:  5 rooms
CALL
studs=>
```

Рис. 4 - Результат работы процедуры с допустимыми данными

```
studs=> call can_assign_role('s338828', 'ahsfghjsaf');
INFO:  Пользователь ahsfghjsaf не найден
CALL
studs=>
```

Рис. 5 - Результат работы процедуры с недопустимыми данными

По варианту задания использование information_schema было сокращено. Вместо information_schema были использованы представления, которыми можно воспользоваться только через системный каталог pg_catalog. Следующий код был получен с использованием развернутых запросов для представлений из information_schema.views, не содержащих информационной схемы:

```
CREATE OR REPLACE PROCEDURE can_assign_role(
  first_user text,
  second_user text
)
AS
$$
DECLARE
  rec record;
  num int := 1;
```

```
BEGIN
  IF NOT EXISTS(
    SELECT *
    FROM pg_user
    WHERE usename = first_user
  )
    THEN RAISE INFO 'Пользователь % не найден', first_user;
    RETURN;
  END IF;
  IF NOT EXISTS(
    SELECT *
    FROM pg_user
    WHERE usename = second_user
  )
    THEN RAISE INFO 'Пользователь % не найден', second_user;
    RETURN;
  END IF;
  RAISE INFO 'Текущий пользователь: %', first_user;
  RAISE INFO 'Кому выдаём права доступа: %', second_user;
  RAISE INFO 'No. Имя таблицы';
  RAISE INFO '--- ------------------------------';
  FOR rec IN
  with instead_of_IS as (SELECT u_grantor.rolname AS grantor,
    grantee.rolname AS grantee,
    current_database() AS table_catalog,
    nc.nspname AS table_schema,
    c.relname AS table_name,
    c.prtype AS privilege_type,
    CASE WHEN (pg_has_role(grantee.oid, c.relowner, 'USAGE'::text) OR c.grantable)
  THEN 'YES'::text
  ELSE 'NO'::text END AS is_grantable,
    CASE WHEN (c.prtype = 'SELECT'::text)
  THEN 'YES'::text
  ELSE 'NO'::text END AS with_hierarchy

   FROM ( SELECT pg_class.oid,
            pg_class.relname,
            pg_class.relnamespace,
            pg_class.relkind,
            pg_class.relowner,
            (aclexplode(COALESCE(pg_class.relacl,  acldefault('r'::"char",  pg_class.relowner)))).grantor  AS
grantor,
            (aclexplode(COALESCE(pg_class.relacl,  acldefault('r'::"char",  pg_class.relowner)))).grantee  AS
grantee,
            (aclexplode(COALESCE(pg_class.relacl,  acldefault('r'::"char",  pg_class.relowner)))).privilege_type
AS privilege_type,
            (aclexplode(COALESCE(pg_class.relacl, acldefault('r'::"char", pg_class.relowner)))).is_grantable AS
is_grantable
           FROM pg_class)
        as
        c(oid,
```

```
      relname,
      relnamespace,
      relkind,
      relowner,
      grantor,
      grantee,
      prtype,
      grantable),

    pg_namespace as nc,
    pg_authid as u_grantor,

    ( SELECT pg_authid.oid,
   pg_authid.rolname
     FROM pg_authid
   UNION ALL
   SELECT (0)::oid AS oid, 'PUBLIC'::name)

   as grantee(oid, rolname)

WHERE ((c.relnamespace = nc.oid)
     AND (c.relkind = ANY (ARRAY['r'::"char", 'v'::"char", 'f'::"char", 'p'::"char"]))
     AND (c.grantee = grantee.oid)
     AND (c.grantor = u_grantor.oid)
     AND (c.prtype = ANY (ARRAY['INSERT'::text, 'SELECT'::text, 'UPDATE'::text, 'DELETE'::text,
'TRUNCATE'::text, 'REFERENCES'::text, 'TRIGGER'::text]))
     AND (pg_has_role(u_grantor.oid, 'USAGE'::text)
      OR pg_has_role(grantee.oid, 'USAGE'::text)
      OR (grantee.rolname = 'PUBLIC'::name)))
)
    SELECT DISTINCT table_name as name_of_table
    FROM instead_of_IS as grants
    JOIN pg_tables as tables
      ON grants.table_schema = tables.schemaname
      AND grants.table_name = tables.tablename
    WHERE grantor = first_user
    AND is_grantable = 'YES'
    LOOP
      RAISE INFO '% %', num, rec.name_of_table;
      num := num + 1;
    END LOOP;
END
$$ LANGUAGE plpgsql;
asdasdasd
```

Несмотря на то, что представления из этого запроса и находились в представлении information_schema.role_table_grants, которым я пользовалась в прошлом коде, на этот раз я не смогу использовать созданную процедуру, так как теперь я для работы с именем пользователя напрямую обращаюсь к каталогам, к которым у моего пользователя s338828 нет доступа (pg_authid). Однако, я

проверила код в PGAdmin4, и работает он идентично прошлому, поэтому задачу можно считать решенной.

Ниже приведена ошибка, с которой я столкнулась в базе данных studs:

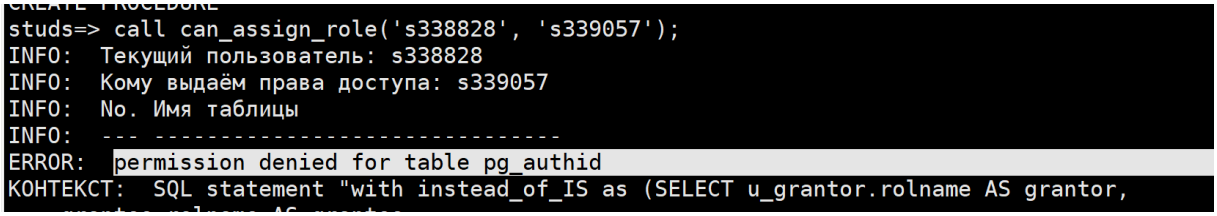


Рис. 6 - Ошибка

Выводы

В ходе лабораторной работе я освежила свои знания работы с базами данных, полученные в прошлом семестре, а также научилась работать с различными системными каталогами, содержащимися в них представлениями и использовать их для выполнения своих задач.