

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: Распределенные системы хранения данных

Лабораторная работа 1

Вариант 312471

Выполнил:

Гурьянов Кирилл Алексеевич

Группа: Р33302

Преподаватель:

Шешуков Дмитрий Михайлович

Санкт-Петербург

2024

Задание

Введите вариант: **312471**

Используя сведения из системных каталогов, получить информацию обо всех столбцах заданной таблицы, находящейся в схеме заданного пользователя. Полученную информацию представить в следующем формате:

Пользователь: Ivan Ivanov (s1000000)

Таблица: н_характеристики_видов_работ

Но. столбца	Имя столбца	Атрибуты
1	свр_ид	Type : NUMBER(9) Constr : "хвр_свр_fk" References н_свойства_вр(ид)
2	вр_ид	Type : NUMBER(9) Constr : "хвр_вр_fk" References н_виды_работ(ид)
2	кто_создал	Type : Date
3	когда_создал	Type : Date
4	кто_изменил	Type : Date
5	когда_изменил	Type : Date

Программу оформить в виде процедуры.

Цель

Познакомиться с системными каталогами Postgresql, научиться составлять запросы к системным каталогам с целью получения информации о таблицах, создать процедуру, которая по заданному названию таблицы и пользователю получит информацию обо всех столбцах таблицы.

Разработанная процедура

В ходе выполнения лабораторной работы была написана процедура, код который приложен ниже.

```
CREATE OR REPLACE PROCEDURE get_table_columns_info(p_table_name text,
```

```

p_user_name text) AS
$BODY$
DECLARE
    v_schema_name record;
    v_table_name record;
    v_attname record;
    v_column_info record;
    v_constraint record;
    v_max_legth int;
    v_column_num integer := 0;
    not_have_len bool = false;
    result text;
    constraint_res text;
    number_of_failed_attempts int = 0;
    number_of_scheme int = 0;
BEGIN
    FOR v_schema_name IN
        SELECT nspname, oid
        FROM pg_namespace
        WHERE has_schema_privilege(p_user_name, nspname, 'USAGE')
    LOOP
        number_of_scheme = number_of_scheme + 1;
        IF (SELECT relname
            FROM pg_class
            WHERE relnamespace = v_schema_name.oid
            AND relkind = 'r' AND relname = p_table_name) IS NULL THEN
            number_of_failed_attempts = number_of_failed_attempts + 1;
        ELSE
            FOR v_table_name IN
                SELECT relname, oid
                FROM pg_class
                WHERE relnamespace = v_schema_name.oid
                AND relkind = 'r' AND relname = p_table_name
            LOOP
                RAISE NOTICE 'Пользователь: % (%)', p_user_name,
v_schema_name.nspname;
                RAISE NOTICE 'Таблица: %', v_table_name.relname;
                RAISE NOTICE 'Но. Имя столбца Атрибуты';
                RAISE NOTICE '--- -----';
                RAISE NOTICE '-----';

                FOR v_attname IN (
                    SELECT attname, atttypid, atttypmod
                    FROM pg_attribute
                    WHERE attrelid = v_table_name.oid AND attnum > 0
                )
                LOOP
                    SELECT typname, typlen, typnotnull INTO v_column_info
                    FROM pg_type
                    WHERE oid = v_attname.atttypid;
                    v_column_num := v_column_num + 1;
                END LOOP;
            END LOOP;
        END IF;
    END LOOP;
END
$BODY$

```

```

not_have_len = false;
CASE
    WHEN v_column_info.typname = 'varchar' THEN
        v_max_legth = v_attname.atttypmod - 4;
    WHEN v_column_info.typname = 'text' OR
v_column_info.typname = 'date' OR v_column_info.typname = 'int4' THEN
        not_have_len = true;
    WHEN v_column_info.typname = 'numeric' THEN
        SELECT numeric_precision INTO v_max_legth FROM
information_schema.columns
        WHERE table_name = p_table_name AND column_name =
v_attname.attname;
    ELSE
        v_max_legth = v_column_info.typlen;
END CASE;

IF not_have_len THEN
    SELECT FORMAT('%-3s %-20s Type : %-10s', v_column_num,
v_attname.attname, v_column_info.typname) INTO result;
    RAISE NOTICE '%', result;
ELSE
    SELECT FORMAT('%-3s %-20s Type : %s(%s)', v_column_num,
v_attname.attname, v_column_info.typname, v_max_legth) INTO result;
    RAISE NOTICE '%', result;

END IF;

FOR v_constraint IN (
    SELECT c.conname, c.conrelid, c.confrelid, a.attname,
conf.relname as conf_table, a2.attname as conf_column
    FROM pg_constraint c
    JOIN pg_attribute a ON a.attnum = ANY(c.conkey) AND
a.attrelid = c.conrelid
    JOIN pg_class conf ON conf.oid = c.confrelid
    JOIN pg_attribute a2 ON a2.attnum = ANY(c.confkey) AND
a2.attrelid = c.confrelid
    WHERE a.attrelid = v_table_name.oid AND a.attname =
v_attname.attname
)
LOOP
    SELECT FORMAT('
References %s(%s)', v_constraint.conname, v_constraint.conf_table,
v_constraint.conf_column) INTO constraint_res;
    RAISE NOTICE '%', constraint_res;
END LOOP;

FOR v_constraint IN (
    SELECT c.conname, c.conrelid, c.confrelid, a.attname,
c.conrelid, c.confkey, c.contype, pg_get_constraintdef(c.oid) as check_condition
    FROM pg_constraint c
    JOIN pg_attribute a ON a.attnum = ANY(c.conkey) AND

```

```

a.attrelid =          c.conrelid
                        WHERE a.attrelid = v_table_name.oid AND c.contype =
'c'AND a.attname = v_attname.attname
                        )
                        LOOP
                            SELECT FORMAT('                Constr : %s %s',
v_constraint.conname, v_constraint.check_condition) INTO constraint_res;
                            RAISE NOTICE '%', constraint_res;
                        END LOOP;
                    END LOOP;
                END LOOP;
            END IF;
        END LOOP;
    IF (number_of_failed_attempts = number_of_scheme) THEN
        RAISE NOTICE 'Данная таблица не найдена';
    END IF;
END;
$BODY$
LANGUAGE plpgsql;

```

Результат работы процедуры

```

studs=> call get_table_columns_info ('driver', 's312471');
NOTICE: Пользователь: s312471 (s312471)
NOTICE: Таблица: driver
NOTICE: No.  Имя столбца  Атрибуты
NOTICE: ---  -----
NOTICE: 1    id              Type : int4
NOTICE: 2    person_id       Type : int4
NOTICE:                Constr : driver_person_id_fkey References person(id)
NOTICE: 3    passport        Type : varchar(10)
NOTICE:                Constr : driver_passport_check CHECK (((passport)::text ~ '^[0-9]{10}$'::text))
NOTICE: 4    bank_card_number Type : text
CALL
studs=> call get_table_columns_info ('cargo', 's312471');
NOTICE: Пользователь: s312471 (s312471)
NOTICE: Таблица: cargo
NOTICE: No.  Имя столбца  Атрибуты
NOTICE: ---  -----
NOTICE: 1    id              Type : int4
NOTICE: 2    weight          Type : float8
NOTICE:                Constr : cargo_weight_check CHECK ((weight <= (25000)::double precision))
NOTICE: 3    width            Type : float8
NOTICE:                Constr : cargo_width_check CHECK ((width <= (2.5)::double precision))
NOTICE: 4    height           Type : float8
NOTICE:                Constr : cargo_height_check CHECK ((height <= (4)::double precision))
NOTICE: 5    length           Type : float8
NOTICE:                Constr : cargo_length_check CHECK ((length <= (15)::double precision))
NOTICE: 6    order_id         Type : int4
NOTICE:                Constr : cargo_order_id_fkey References orders(id)
NOTICE: 7    cargo_type       Type : cargo_type(4)
CALL

```

```

NOTICE: Пользователь: s312471 (s312471)
NOTICE: Таблица: loading_unloading_agreement
NOTICE: No. Имя столбца Атрибуты
NOTICE: -----
NOTICE: 1 order_id Type : int4
NOTICE: Constr : loading_unloading_agreement_order_id_fkey References orders(id)
NOTICE: 2 driver_id Type : int4
NOTICE: Constr : loading_unloading_agreement_driver_id_fkey References driver(id)
NOTICE: 3 departure_point Type : int4
NOTICE: Constr : loading_unloading_agreement_departure_point_fkey References storage_point(address_id)
NOTICE: Constr : loading_unloading_agreement_check CHECK ((departure_point <> delivery_point))
NOTICE: 4 delivery_point Type : int4
NOTICE: Constr : loading_unloading_agreement_delivery_point_fkey References storage_point(address_id)
NOTICE: Constr : loading_unloading_agreement_check CHECK ((departure_point <> delivery_point))
NOTICE: 5 sender_id Type : int4
NOTICE: Constr : loading_unloading_agreement_sender_id_fkey References person(id)
NOTICE: 6 receiver_id Type : int4
NOTICE: Constr : loading_unloading_agreement_receiver_id_fkey References person(id)
NOTICE: 7 unloading_time Type : time(8)
NOTICE: Constr : loading_unloading_agreement_check1 CHECK (((unloading_time >= '00:01:00'::time without time zone) AND (unloading_time < '24:00:00'::time without t
ime zone) AND (loading_time >= '00:01:00'::time without time zone) AND (loading_time < '24:00:00'::time without time zone)))
NOTICE: 8 loading_time Type : time(8)
NOTICE: Constr : loading_unloading_agreement_check1 CHECK (((unloading_time >= '00:01:00'::time without time zone) AND (unloading_time < '24:00:00'::time without t
ime zone) AND (loading_time >= '00:01:00'::time without time zone) AND (loading_time < '24:00:00'::time without time zone)))

```

```

lab1=# set search_path = 'my_schema1';
SET
lab1=# \dt
          Список отношений
  Схема   |  Имя   |  Тип   | Владелец
-----+-----+-----+-----
my_schema1 | table1 | таблица | postgres
(1 строка)

lab1=# set search_path = 'my_schema2';
SET
lab1=# \dt
          Список отношений
  Схема   |  Имя   |  Тип   | Владелец
-----+-----+-----+-----
my_schema2 | table1 | таблица | postgres
(1 строка)

lab1=# call get_table_columns_info ('table1', 'postgres');
ЗАМЕЧАНИЕ: Пользователь: postgres (my_schema1)
ЗАМЕЧАНИЕ: Таблица: table1
ЗАМЕЧАНИЕ: No. Имя столбца Атрибуты
ЗАМЕЧАНИЕ: ---
ЗАМЕЧАНИЕ: 1 first_col Type : int4
ЗАМЕЧАНИЕ: 2 second_col Type : int4
ЗАМЕЧАНИЕ: Пользователь: postgres (my_schema2)
ЗАМЕЧАНИЕ: Таблица: table1
ЗАМЕЧАНИЕ: No. Имя столбца Атрибуты
ЗАМЕЧАНИЕ: ---
ЗАМЕЧАНИЕ: 3 first_col2 Type : int4
ЗАМЕЧАНИЕ: 4 second_col2 Type : varchar(12)
CALL

```

Если пользователь, запустивший процедуру, имеет права доступа к нескольким схемам и в этих схемах имеются таблицы с одинаковыми именами, то процедура "обрабатывает" два раза и выводит информацию по обеим таблицам.

Объяснение процедуры

```

FOR v_schema_name IN
    SELECT nspname, oid
    FROM pg_namespace
    WHERE has_schema_privilege(p_user_name, nspname, 'USAGE')

```

```

LOOP
...
END LOOP;

```

Достаем имя пространства имен и oid тех схем, к которым у переданного пользователя имеются права доступа.

```

FOR v_table_name IN
    SELECT relname, oid
    FROM pg_class
    WHERE relnamespace = v_schema_name.oid
    AND relkind = 'r' AND relname = p_table_name
LOOP
    RAISE NOTICE 'Пользователь: % (%)', p_user_name,
v_schema_name.nspname;
    RAISE NOTICE 'Таблица: %', v_table_name.relname;
    RAISE NOTICE 'No. Имя столбца  Атрибуты';
    RAISE NOTICE '--- -----';
    -----';
    ...
END LOOP;

```

Здесь для каждой схемы мы находим имена и oid таблиц, которые имеются имя, которое передается в процедуру. Условие relkind = 'r' необходимо, чтобы найти именно таблицы.

```

FOR v_attname IN (
    SELECT attname, atttypid,
    FROM pg_attribute
    WHERE attrelid = v_table_name.oid AND attnum > 0
)
LOOP
...
END LOOP;

```

Здесь мы достаем информацию о столбцах таблицы из pg_attribute. Нам нужны имена и тип данных столбцов, у которых attrelid = v_table_name.oid (проверка, что колонка принадлежит нужной таблице) и attnum > 0 (проверка, что столбец не является служебным, т.к. системные столбцы имеют отрицательные номера).

```

SELECT typename, typelen, typnotnull INTO v_column_info
    FROM pg_type
    WHERE oid = v_attname.atttypid;
v_column_num := v_column_num + 1;
not_have_len = false;
CASE
    WHEN v_column_info.typname = 'varchar' THEN
        v_max_legth = v_attname.atttypmod - 4;
    WHEN v_column_info.typname = 'text' OR v_column_info.typname = 'date' OR
v_column_info.typname = 'int4' OR v_column_info.typname = 'float8' THEN
        not_have_len = true;
    WHEN v_column_info.typname = 'numeric' THEN
        SELECT numeric_precision INTO v_max_legth
        FROM information_schema.columns
        WHERE table_name = p_table_name AND column_name = v_attname.attname;
    ELSE
        v_max_legth = v_column_info.typelen;
END CASE;

```

Здесь мы достаём детализированную информацию по форматам данных, которые находятся в колонках таблицы, а именно имя типа данных, его typelen, которое показывает число байт во внутреннем представлении типа. Затем идет проверка на определенные типы данных, которые должны иметь определенное или не иметь определенную длину или точность. Например для varchar достается atttypmod, в котором записан максимальный размер столбца varchar.

```

IF not_have_len THEN
    SELECT FORMAT('%-3s %-20s Type : %-10s', v_column_num,
v_attname.attname, v_column_info.typname) INTO result;
    RAISE NOTICE '%', result;
ELSE
    SELECT FORMAT('%-3s %-20s Type : %s(%s)', v_column_num,
v_attname.attname, v_column_info.typname, v_max_legth) INTO result;
    RAISE NOTICE '%', result;
END IF;

```

Здесь происходит вывод в зависимости от наличия специальных условий у типа данных.

```

FOR v_constraint IN (
    SELECT c.conname, c.conrelid, c.confrelid, a.attname,
conf.relname as conf_table, a2.attname as conf_column
    FROM pg_constraint c
    JOIN pg_attribute a ON a.attnum = ANY(c.conkey) AND
a.attrelid = c.conrelid
    JOIN pg_class conf ON conf.oid = c.confrelid

```



```

                                JOIN pg_attribute a2 ON a2.attnum = ANY(c.confkey) AND
a2.attrelid = c.confrelid
                                WHERE a.attrelid = v_table_name.oid AND a.attname =
v_attname.attname
                                )
                                LOOP
                                SELECT FORMAT('                                Constr : %s
References %s(%s)', v_constraint.conname, v_constraint.conf_table,
v_constraint.conf_column) INTO constraint_res;
                                RAISE NOTICE '%', constraint_res;
                                END LOOP;

```

Здесь происходит поиск и вывод ограничений, которые имеются на колонке, мы выводим внешние ключи, которые ссылаются на другую таблицу с указанием колонки, на которую они ссылаются. Мы достаем имя ограничения, oid таблицы, для которой установлено это ограничение, oid таблицы, на которую ссылается это ограничение, имя столбца, который является внешним ключом, имя таблицы, на которое ссылается ограничение, имя столбца, на который ссылается ограничение.

Вывод

В ходе выполнения лабораторной работы я познакомился с системными каталогами Postgresql, научился составлять запросы к системным каталогам с целью получения информации о таблицах, создал процедуру, которая по заданному названию таблицы и пользователю получит информацию обо всех столбцах таблицы.