

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное образовательное
учреждение высшего образования “Национальный
исследовательский университет ИТМО”

**ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И
КОМПЬЮТЕРНОЙ ТЕХНИКИ**

ЛАБОРАТОРНАЯ РАБОТА №1

по дисциплине

“Распределённые системы хранения данных”

вариант 331107

Выполнили:

Иванов Матвей Сергеевич

Шульга Артём Игоревич

группа Р33111

Преподаватель:

Николаев Владимир Вячеславович

г. Санкт-Петербург, 2023

Задание

Используя сведения из системных каталогов, сформировать список таблиц в заданной схеме, хотя бы в одной из строк хотя бы одной из колонок имеющих значение 'NULL'. Полученную информацию представить в виде списка следующего формата:

Схема: s100000

Но. Имя таблицы

- 1 Н_ЛЮДИ
- 2 Н_ХАРАКТЕРИСТИКИ_ВИДОВ_РАБОТ
- 3 Н_УЧЕНИКИ
- 4 Н_ПЛАНЫ
- 5 Н_ОТДЕЛЫ
- ...

Программу оформить в виде процедуры.

Ход выполнения

Ввод схемы:

```
\echo 'Enter scheme: '  
\prompt 'Enter scheme: ' my_scheme_name  
\o /dev/null  
select set_config('psql.my_scheme_name', :my_scheme_name, false);
```

Вспомогательная функция для расчёта отступа:

```
CREATE OR REPLACE FUNCTION get_indent_for_index(ind INT) RETURNS  
VARCHAR(2) AS $$  
DECLARE  
    indent VARCHAR(2);  
BEGIN  
    SELECT REPEAT(' ', GREATEST((2 - floor(log(ind)))::int, 0)) INTO indent;  
    RETURN indent;  
END  
$$ LANGUAGE 'plpgsql';
```

Вспомогательная функция для получения таблиц, к которым возможно сделать SELECT запрос:

```
CREATE OR REPLACE FUNCTION get_tables_with_select(scheme_name
TEXT)
RETURNS TABLE (table_name TEXT) AS $$
BEGIN
    RETURN QUERY
        SELECT DISTINCT p.table_name::text as table_name
        FROM information_schema.table_privileges p
        WHERE
            p.table_schema=scheme_name
            AND p.privilege_type = 'SELECT';
END
$$ LANGUAGE 'plpgsql';
```

Вспомогательная функция для получения таблиц, к которым нельзя сделать SELECT запрос:

```
CREATE OR REPLACE FUNCTION
get_tables_without_select(scheme_name TEXT)
RETURNS TABLE (table_name TEXT) AS $$
BEGIN
    RETURN QUERY
        SELECT
            p.table_name::text as table_name
        FROM information_schema.table_privileges p
        WHERE p.table_schema=scheme_name
        GROUP BY p.table_name
        HAVING COUNT(CASE WHEN p.privilege_type = 'SELECT' THEN 1 END)
        = 0;
END
$$ LANGUAGE 'plpgsql';
```

Основная процедура для вывода таблиц, в которых есть NULL:

```
CREATE OR REPLACE PROCEDURE get_scheme_null_tables()
AS $$
DECLARE
    scheme_name TEXT = current_setting('psql.my_scheme_name');
    cur_table_name TEXT;
    null_rows_cnt INT;
    rec RECORD;
    indent VARCHAR(2);
    row_ind INT = 1;
    header BOOL = FALSE;
BEGIN
    IF NOT scheme_name IN (SELECT nspname FROM
pg_catalog.pg_namespaces)
        THEN
            RAISE NOTICE 'This schema does not exist';
            RETURN;
        END IF;

    IF NOT (SELECT pg_catalog.has_schema_privilege(scheme_name,
'USAGE'))
        THEN
            RAISE NOTICE 'You do not have access to view schema';
            RETURN;
        END IF;

    FOR rec IN
        SELECT * FROM get_tables_with_select(scheme_name)
    LOOP
        IF (NOT header) THEN
            RAISE NOTICE 'No. Имя Таблицы';
            RAISE NOTICE '-----';
            header := TRUE;
        END IF;
        cur_table_name := quote_ident(scheme_name) || '.' ||
quote_ident(rec.table_name);
        EXECUTE 'SELECT count(*) FROM ' || cur_table_name || ' cur_table
WHERE NOT (cur_table IS NOT NULL)' INTO null_rows_cnt;
        IF null_rows_cnt > 0 THEN
            SELECT get_indent_for_index(row_ind) INTO indent;
```

```

        RAISE NOTICE '% %', indent || row_ind::text, rec.table_name;
        row_ind := row_ind + 1;
    END IF;
END LOOP;
row_ind := 1;
header := FALSE;
FOR rec IN
    SELECT * FROM get_tables_without_select(scheme_name)
LOOP
    IF (NOT header) THEN
        RAISE NOTICE 'You do not have access to view tables:~';
        RAISE NOTICE 'Но. Имя Таблицы';
        RAISE NOTICE '-----';
        header := TRUE;
    END IF;
    SELECT get_indent_for_index(row_ind) INTO indent;
    RAISE NOTICE '% %', indent || row_ind::text, rec.table_name;
    row_ind := row_ind + 1;
END LOOP;
END
$$ LANGUAGE 'plpgsql';

```

Вызов процедуры:

```
CALL get_scheme_null_tables();
```

Выводы

В процессе выполнения лабораторной работы мы изучили множество материалов по устройству и организации ввода-вывода в Postgres, а также разобрались в устройстве системных каталогов и информации, которая в них хранится.