



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

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

**ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ
И КОМПЬЮТЕРНОЙ ТЕХНИКИ**



РСХД.

0000—0?

ЛАБОРАТОРНАЯ РАБОТА №1

по дисциплине
“Распределённые системы хранения данных”

Вариант: 374799.

Работу выполнил:

Студент группы Р3311

Болорболд Аригуун

Преподаватель:

Николаев Владимир Владиславович

ПИИКТ

И-ой семестр

20/21

1	Текст задания	3
2	Выполнение	3
3	Пример исполнения анонимного блока	5
4	Вывод	5

1 Текст задания

Используя сведения из системных каталогов получить информацию о любой таблице: Номер по порядку, Имя столбца, Атрибуты (в атрибуты столбца включить тип данных и внешние ключи).

Таблица: н_характеристики_видов_работ

№. 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 анонимного блока.

2 Выполнение

Код анонимного блока

Listing 1: lab1_script.sql

```
set custom.table_name = : 'custom_table_name';
set custom.table_schema = : 'custom_table_schema';
set custom.table_databs = : 'custom_table_databs';

do
$$
declare
    schema_name text;
    tab_name text;
    databs_name text;
    column_record record;
    table_id oid;
    column_number int;
    my_column_name text;
    column_type text;
    column_constr text;
    result text;
begin
    tab_name := current_setting('custom.table_name', true);

    begin
        if position('.') in tab_name > 0 then
            if array_length(string_to_array(tab_name, '.'), 1) = 2 then
                databs_name := NULL;
                schema_name := split_part(tab_name, '.', 1);
                tab_name := split_part(tab_name, '.', 2);
            elsif array_length(string_to_array(tab_name, '.'), 1) = 3 then
                databs_name := split_part(tab_name, '.', 1);
                schema_name := split_part(tab_name, '.', 2);
                tab_name := split_part(tab_name, '.', 3);
            end if;
        end if;
    end;
```

```

        end if;
    end if;
exception
    when others then
        databs_name := NULL;
        schema_name := NULL;
        tab_name := NULL;
end;

if tab_name is null or tab_name = '' then
    raise exception '                .';
end if;
if schema_name is null or schema_name = '' then
    execute 'select current_schema()' into schema_name;
end if;
if databs_name is null or databs_name = '' then
    execute 'select current_database()' into databs_name;
end if;

select c.oid into table_id from pg_catalog.pg_class c
join pg_catalog.pg_namespace n on c.relnamespace = n.oid
where c.relname = tab_name and n.nspname = schema_name;

if table_id is null then
    raise exception '                .', databs_name, schema_name, tab_name;
end if;

raise notice '                : %.%.%', databs_name, schema_name, tab_name;
raise notice 'No.                ';
raise notice '-----';

for column_record in
    select attnum, attname, atttypid
    from pg_catalog.pg_attribute
    where attrelid = table_id and attnum > 0
    order by attnum
loop
    column_number := column_record.attnum;
    my_column_name := column_record.attname;
    select typename into column_type
    from pg_catalog.pg_type
    where oid = column_record.atttypid;

    select string_agg(
        format('%s" References %s.%s.%s(%s)', conname, databs_name, n.nspname,
            confrelid::regclass, confkey[1]),
        E'\n'
    )
    into column_constr
    from pg_constraint con
    join pg_catalog.pg_namespace n on con.connamespace = n.oid
    where con.conrelid = table_id and con.contype = 'f'
    and column_record.attnum = any(con.conkey);

    select format('%-3s %-14s %-8s %-2s %s', column_number, my_column_name, 'Type',
        ':', column_type)
    into result;
    raise notice '%', result;

    if column_constr is not null then
        select format('%-18s %-8s %-2s %s', ' ', 'Constr', ':', column_constr) into
            result;
    end if;
end loop;

```

```

        raise notice '%', result;
    end if;
end loop;
end;
$$ LANGUAGE plpgsql;

```

3 Пример исполнения анонимного блока

Вызов анонимного блока из терминала:

В базе данных studs:

```

studs=> \i lab1_script.sql
SET
psql:lab1_script.sql:66: NOTICE: Таблица: s372799.crew
psql:lab1_script.sql:66: NOTICE:
psql:lab1_script.sql:66: NOTICE:
psql:lab1_script.sql:66: NOTICE:
psql:lab1_script.sql:66: NOTICE: 1 id Type : int4
psql:lab1_script.sql:66: NOTICE: Constr : "crew_id_fkey" References crewmember(1)
psql:lab1_script.sql:66: NOTICE: 2 spaceshipid Type : int4
psql:lab1_script.sql:66: NOTICE: Constr : "crew_spaceshipid_fkey" References spaceship(1)
psql:lab1_script.sql:66: NOTICE: 3 teamwork Type : text
DO
studs=> |

```

В ucheb:

```

[s372799@helios ~/ddbs/lab1]$ psql -h pg -d ucheb -v custom_table_name="Н_ХАРАКТЕРИСТИКИ_ВИДОВ_РАБОТ" -f lab1_script.sql
SET
SET
SET
psql:lab1_script.sql:95: NOTICE: ??????: ucheb.public.Н_ХАРАКТЕРИСТИКИ_ВИДОВ_РАБОТ
psql:lab1_script.sql:95: NOTICE:
psql:lab1_script.sql:95: NOTICE:
psql:lab1_script.sql:95: NOTICE:
psql:lab1_script.sql:95: NOTICE: 1 СВР_ИД Type : int4
psql:lab1_script.sql:95: NOTICE: 2 ВР_ИД Type : int4
psql:lab1_script.sql:95: NOTICE: 3 КТО_СОЗДАЛ Type : varchar
psql:lab1_script.sql:95: NOTICE: 4 КОГДА_СОЗДАЛ Type : timestamp
psql:lab1_script.sql:95: NOTICE: 5 КТО_ИЗМЕНИЛ Type : varchar
psql:lab1_script.sql:95: NOTICE: 6 КОГДА_ИЗМЕНИЛ Type : timestamp
DO
[s372799@helios ~/ddbs/lab1]$

```

4 Вывод

В ходе лабораторной работы я изучил системные каталоги и методы обращения с ними. Этот процесс также включает в себе повторение основ PLpgSQL и работа с ролями и правами доступа.