

федеральное государственное автономное образовательное учреждение высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

ОТЧЕТ

по лабораторной работе №1

по дисциплине **«Распределенные системы хранения данных»**

Вариант 312536

Автор: Кулаков Н. В.

Факультет: ПИиКТ

Группа: Р33312

Преподаватель: Шешуков Д. М.



УНИВЕРСИТЕТ ИТМО

Санкт-Петербург 2023

Задание:

Используя сведения из системных каталогов получить информацию о любой таблице:
Номер по порядку, Имя столбца, Атрибуты (в атрибуты столбца включить тип данных, ограничение типа CHECK, комментарий?).

Таблица: Н_ЛЮДИ

Но.	Имя столбца	Атрибуты
1	ИД	Type : NUMBER(9) NOT NULL Comment : 'Уникальный номер человека'
2	ФАМИЛИЯ	Type : VARCHAR2(25) NOT NULL Comment : 'Фамилия человека'
3	ИМЯ	Type : VARCHAR2(2000) NOT NULL Comment : 'Имя человека'
4	ОТЧЕСТВО	Type : VARCHAR2(20) Comment : 'Отчество человека'
5	ДАТА_РОЖДЕНИЯ	Type : DATE NOT NULL Comment : 'Дата рождения человека'
6	ПОЛ	Type : CHAR(1) NOT NULL Constr : "AVCON_378561_ПОЛ_000" CHECK (ПОЛ IN ('М', 'Ж')) Constr : "AVCON_388176_ПОЛ_000" CHECK (ПОЛ IN ('М', 'Ж')) Comment : 'Пол человека'

Программу оформить в виде процедуры.

Решение:

```
create or replace procedure rshd_perform_lab1(table_name name)
language plpgsql
as $$
declare
    table_oid oid;
    trec record;
    buffer text;
    rbuf text;
begin
    -- Get table oid to use in future
    select table_name::regclass::oid into table_oid;

    buffer := format(E'\n\nТаблица: %s\n\n', table_name);
    buffer := buffer || format(E'%-5s %-20s %-50s\n',
        'No.', 'Имя столбца', 'Атрибуты');
    buffer := buffer || format(E'%-5s %-20s %-50s',
        '---', '-----', '-----');
```

```

-- Get information about table columns (common table expression)
FOR trec in (
  with from_pg_attr as (
    select
      attrelid, -- table identifier
      attname,
      attnum,
      atttypid,
      nullif(information_schema._pg_char_max_length(
        atttypid, atttypmod), -1) as maxlen,
      attnotnull
    from pg_catalog.pg_attribute
    where attrelid = table_oid and attnum > 0
  ),
  -- Get information about column types
  from_pg_type as (
    select distinct
      oid as typeoid,
      typname
    from pg_catalog.pg_type
  join
    from_pg_attr on from_pg_attr.atttypid = pg_catalog.pg_type.oid
  ),
  -- Get information about column constraints
  from_pg_constr as (
    select
      -- oid, -- row identifier
      conrelid, -- table
      unnest(conkey) unnested_key, -- column number
      conname,
      -- contype,
      pg_get_constraintdef(oid) constraintdef
    from pg_catalog.pg_constraint
    where
      conrelid = table_oid and contype = 'c' -- Select only checks
    order by unnested_key
  ),
  -- Get description
  from_pg_descr as (
    select
      objoid, -- refers to the table
      objsubid, -- column
      description
    from pg_catalog.pg_description
    where objoid = table_oid
  )
  -- Combine received result
  -- select * from from_pg_type
  select
    -- attrelid,
    attnum,
    attname,

```

```

-- atttypid,
typename,
maxlen,
attnotnull,
array_agg((conname, constraintdef)) conlist,
-- array_agg(conname) conname_list,
-- array_agg(constraintdef) constraintdef_list,
description
from
    from_pg_attr pgattr
join
    from_pg_type pgtype on pgattr.atttypid = pgtype.typeoid
left join
    from_pg_constr pgconstr on
        pgattr.attrelid = pgconstr.conrelid and
        pgattr.attnum = pgconstr.unnested_key
left join
    from_pg_descr pgdescr on
        pgattr.attrelid = pgdescr.objoid and
        pgattr.attnum = pgdescr.objsubid
group by attnum, attname, typename, maxlen, attnotnull, description
order by attnum
)
-- Loop for each table_record
LOOP
    rbuf := format('%-5s %-20s', trec.attnum, trec.attname);
    -- Type
    rbuf := rbuf || format(E' %-10s: %s', 'Type', trec.typename);
    if trec.maxlen is not null then
        rbuf := rbuf || format('(%s)', trec.maxlen);
    end if;
    -- not null
    if trec.attnotnull is true then
        rbuf := rbuf || ' NOT NULL';
    end if;

    -- Constraints
    declare
        conname_len int4 := array_length(trec.conlist, 1);
        name_def record;
    begin
        if conname_len > 0 then
            foreach name_def in array trec.conlist loop
                if name_def.f1 is not NULL then
                    rbuf := rbuf || format(E'\n%-5s %-20s %-10s: "%s" %s',
                        ", ", 'Constr', name_def.f1, name_def.f2);
                end if;
            end loop;
        end if;
    end;

    -- Comments

```

```
    if trec.description is not null then
        rbuf := rbuf || format(E'\n%-5s %-20s %-10s: %s',
            ", ", 'Descript', trec.description);
    end if;

    buffer := buffer || E'\n' || rbuf;
end LOOP;

raise notice '%', buffer;
end;
$$;
```

Вывод:

При выполнении лабораторной работы я изучил системные каталоги СУДБ PostgreSQL, и использовал их для получения информации об атрибутах таблицы. Кроме того, узнал больше о типе record, анонимных recordax.