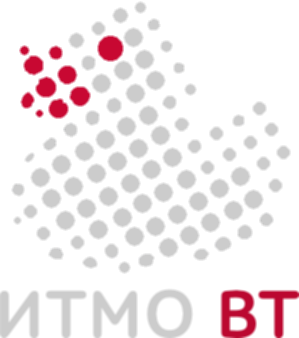
Санкт-Петербургский Национальный Исследовательский Университет Информационных Технологий, Механики и Оптики

ПИиКТ

Лабораторная работа №1

по дисциплине

«Распределенные системы хранения данных»



Выполнил: Студенты группы P33112

Почикалин Владислав

Чарный Никита

Преподаватель: Николаев Владимир Вячеславович

Санкт-Петербург

2021 г.

**Вариант 31168:**

**Изображение выглядит как текст

Автоматически созданное описание**

**Выполнение:**

create or replace PROCEDURE *get\_columns\_info*(x in varchar2 , y in varchar2)  
 IS  
 -- variables part  
 tablename VARCHAR2(40) := y;  
 schemaname Varchar2(40) := x;  
 colNo VARCHAR2(128) := 'No.';  
 colName VARCHAR2(128) := 'Имя столбца';  
 colAttr VARCHAR2(128) := 'Атрибуты';  
 noCounter NUMBER := 0;  
 noLen NUMBER := 3;  
 colLen NUMBER := 15;  
 attrLen NUMBER := 60;  
 attrNameLen NUMBER := 7;  
 dataType VARCHAR2(128);  
 constr VARCHAR2(128);  
 name varchar2(128) := '';  
 surname varchar2(128) := '';  
 constrName varchar2(128) := '';  
 countReservedWords number := 0;  
 countParamsExisting number := 0;  
  
 -- exception part  
 schemaNotFound Exception;  
 tableNotFound Exception;  
 schemaContainsReservedWords Exception;  
 tableContainsReservedWords Exception;  
  
  
  
 cursor CONST is  
 select DISTINCT acc.OWNER, acc.CONSTRAINT\_NAME, acc.COLUMN\_NAME, ac.CONSTRAINT\_TYPE, atc.DATA\_TYPE, atc.DATA\_PRECISION from ALL\_CONSTRAINTS ac  
 JOIN ALL\_CONS\_COLUMNS acc on ac.CONSTRAINT\_NAME = acc.CONSTRAINT\_NAME  
 JOIN ALL\_TAB\_COLUMNS atc on acc.COLUMN\_NAME = atc.COLUMN\_NAME and acc.TABLE\_NAME = atc.TABLE\_NAME  
 where atc.OWNER = schemaname and acc.TABLE\_NAME = tablename;  
   
 cursor FIO is  
 select ФАМИЛИЯ, ИМЯ from Н\_ЛЮДИ where ИД =  
 (select ЧЛВК\_ИД from Н\_УЧЕНИКИ where ИД = (select user\_id from ALL\_USERS where USERNAME = schemaname));  
  
  
  
 cursor FK\_curs is  
 select acc2.CONSTRAINT\_NAME as fk\_constrName, acc2.COLUMN\_NAME as fk\_name, acc.TABLE\_NAME as ref\_tab, acc.COLUMN\_NAME as ref\_name from ALL\_CONSTRAINTS ac  
 join ALL\_CONS\_COLUMNS acc on ac.R\_CONSTRAINT\_NAME = acc.CONSTRAINT\_NAME  
 join ALL\_CONS\_COLUMNS acc2 on ac.CONSTRAINT\_NAME = acc2.CONSTRAINT\_NAME  
 where ac.OWNER = schemaname and ac.TABLE\_NAME = tablename and ac.CONSTRAINT\_TYPE = 'R';  
  
 BEGIN  
 -- validation part  
 select count(\*) into countReservedWords from V$RESERVED\_WORDS where KEYWORD=schemaname;  
 if countReservedWords >0 then  
 raise schemaContainsReservedWords;  
 end if;  
   
 select count(\*) into countReservedWords from V$RESERVED\_WORDS where KEYWORD=tablename;  
 if countReservedWords >0 then  
 raise tableContainsReservedWords;  
 end if;  
   
 select count(\*) into countParamsExisting from DBA\_USERS where USERNAME=schemaname;  
 IF schemaname IS NULL or countParamsExisting = 0 then  
 raise schemaNotFound;  
 end if;  
   
 select count(\*) into countParamsExisting from DBA\_TABLES where TABLE\_NAME=tablename;  
 if tablename is NULL or countParamsExisting = 0 then  
 raise tableNotFound;  
 end if;  
   
 -- set user's info  
 FOR fi IN FIO loop  
 name := fi.ИМЯ;  
 surname := fi.ФАМИЛИЯ;  
 end loop;  
  
 -- set result table  
 DBMS\_OUTPUT.*PUT\_LINE*('Пользователь: ' || surname || ' ' || name || ' ' || '(' || schemaname || ')');  
 DBMS\_OUTPUT.*PUT\_LINE*('Таблица: ' || tableName);  
 DBMS\_OUTPUT.*PUT\_LINE*('');  
 DBMS\_OUTPUT.*PUT\_LINE*(RPAD(colNo, noLen) || ' ' || RPAD(colName, colLen) || ' ' || RPAD(colAttr, attrLen));  
 DBMS\_OUTPUT.*PUT\_LINE*(RPAD('-', noLen, '-') || ' ' || RPAD('-', colLen, '-') || ' ' || RPAD('-', attrLen, '-'));  
 FOR ROW IN CONST loop  
 noCounter := noCounter + 1;  
 colName := ROW.COLUMN\_NAME;  
 colAttr := ROW.DATA\_TYPE;  
 constr := ROW.CONSTRAINT\_TYPE;  
 constrName := ROW.CONSTRAINT\_NAME;  
 dataType := RPAD('Type: ', attrNameLen) || ROW.DATA\_TYPE;  
 IF ROW.DATA\_PRECISION IS NOT NULL THEN  
 dataType := dataType || '(' || ROW.DATA\_PRECISION || ')';  
 END IF;  
 DBMS\_OUTPUT.*PUT\_LINE*(RPAD(noCounter, noLen, ' ') || ' ' || RPAD(colName, colLen, ' ') || ' ' || dataType);  
 IF constr = 'R' THEN  
 for line in FK\_curs loop  
 if line.fk\_constrName = constrName then  
 DBMS\_OUTPUT.*PUT\_LINE*(RPAD(' ', noLen + colLen + 2) || RPAD('Constr: ', 8) ||'"' || line.fk\_name || '"' || ' References ' || line.ref\_tab || '(' || line.ref\_name || ')');  
 end if;  
 end loop;  
 end if;  
 IF constr = 'U' then  
 DBMS\_OUTPUT.*PUT\_LINE*(RPAD(' ', noLen + colLen + 2) || RPAD('Constr: ', 8) ||'"' || 'UNIQUE'|| '"');  
 end if;  
 IF constr = 'C' then  
 DBMS\_OUTPUT.*PUT\_LINE*(RPAD(' ', noLen + colLen + 2) || RPAD('Constr: ', 8) ||'"' || 'NOT NULL'|| '"');  
 end if;  
 IF constr = 'P' then  
 DBMS\_OUTPUT.*PUT\_LINE*(RPAD(' ', noLen + colLen + 2) || RPAD('Constr: ', 8) ||'"' || 'PRIMARY KEY'|| '"');  
 end if;  
 end loop;  
  
 EXCEPTION  
 WHEN schemaNotFound THEN raise\_application\_error(- 20001, 'No schema with that name');  
 WHEN tableNotFound THEN raise\_application\_error(- 20001, 'No table with that name');  
 WHEN schemaContainsReservedWords THEN raise\_application\_error(- 20001, 'Schema name contains reserved words');  
 WHEN tableContainsReservedWords THEN raise\_application\_error(- 20001, 'Table name contains reserved words');  
  
 end *get\_columns\_info*;

/

SET LIN 400;

SET SERVEROUT ON SIZE UNLIMITED;

accept sch PROMPT 'Enter schema name: ';

ACCEPT t PROMPT 'Enter table name: ';

BEGIN

get\_columns\_info(&sch,&t);

end;

/

**Вывод**

В ходе выполнения лабораторной работы были изучены основные конструкции языка PL / SQL, а также структура словаря данных, который является основным источником мета информации для базы данных.