

Университет ИТМО
Факультет программной инженерии и компьютерной техники

Распределённые системы хранения данных. Лабораторная работа №1.

Группа: Р33121
Студент: Гиниятуллин Арслан Рафаилович
Преподаватель: Афанасьев Дмитрий Борисович
Вариант: 661

Ключевые слова

База данных, PostgreSQL, системный каталог.

Содержание

1	Цель работы	1
2	Текст задания	1
3	Инициализация объектов	1
4	Реализация скрипта	2
5	Вывод скрипта	2
6	Вывод	3

1 Цель работы

Научиться проектировать базы данных, составлять инфологические и даталогические модели данных, реализовывать их в БД PostgreSQL, научиться выполнять запросы.

2 Текст задания

Используя сведения из системных каталогов, сформировать запрос, реализующий полнотекстовый поиск по исходному коду всех процедур, функций и триггеров в пределах заданной схемы. Поиск должен осуществляться независимо от регистра символов в строке запроса. Программу оформить в виде процедуры.

Пример вывода:

Текст запроса: Н_Люди

No.	Имя объекта	# строки	Текст
1	MyFunction1	16	SELECT * FROM н_люди WHERE
2	MyProcedure1	42	INSERT INTO Н_ЛЮДИ
...			

3 Инициализация объектов

Далее был написан SQL скрипт, создающий таблицу, аналогичную той, что в примере.

```
1 CREATE TABLE emp (  
2     empname          text NOT NULL,  
3     salary            integer  
4 );  
5  
6 CREATE TABLE emp_audit(  
7     operation         char(1)  NOT NULL,  
8     stamp              timestamp NOT NULL,  
9     userid             text     NOT NULL,  
10    empname            text     NOT NULL,  
11    salary              integer  
12 );  
13  
14 CREATE OR REPLACE FUNCTION process_emp_audit() RETURNS TRIGGER AS $emp_audit$  
15 BEGIN  
16     IF (TG_OP = 'DELETE') THEN  
17         INSERT INTO emp_audit SELECT 'D', now(), current_user, OLD.*;
```

```

18         ELSIF (TG_OP = 'UPDATE') THEN
19             INSERT INTO emp_audit SELECT 'U', now(), current_user, NEW.*;
20         ELSIF (TG_OP = 'INSERT') THEN
21             INSERT INTO emp_audit SELECT 'I', now(), current_user, NEW.*;
22         END IF;
23         RETURN NULL;
24     END;
25 $emp_audit$ LANGUAGE plpgsql;
26
27 CREATE TRIGGER emp_audit
28 AFTER INSERT OR UPDATE OR DELETE ON emp
29 FOR EACH ROW EXECUTE FUNCTION process_emp_audit();

```

4 Реализация скрипта

```

1 CREATE
2 OR REPLACE PROCEDURE find_fpt_declarations_like(in string text)
3 LANGUAGE plpgsql AS
4 $$
5 DECLARE
6     col RECORD;
7     index INTEGER = 1;
8 BEGIN
9     RAISE INFO '% % % %', rpad('No.', 4), rpad(' ', 21), rpad('# ', 45);
10    RAISE INFO '% % % %', rpad(repeat('-', 3) || ' ', 4), rpad(repeat('-', 20) || ' ', 45);
11    FOR col IN
12        WITH fpt_declarations AS (SELECT row_number() OVER (PARTITION BY objname) as
13            rownumber, *
14            FROM ((SELECT proname AS objname
15                , string_to_table(pg_get_functiondef(oid)
16                    :: text, E'\n'):: text AS
17                declaration
18                FROM pg_proc
19                WHERE prokind IN ('f', 'p')
20                AND pronamespace = ("current_schema")::
21                regnamespace :: oid))
22            UNION
23            all
24            (SELECT tname AS objname
25                , string_to_table(pg_get_triggerdef(
26                    pg_trigger.oid):: text, E'\n'):: text AS
27                declaration
28                FROM pg_trigger
29                JOIN pg_class
30                ON pg_trigger.tgrelid = pg_class.
31                oid
32                WHERE pg_class.relnamespace = ("current_schema"
33                    ):: regnamespace :: oid))) as t1)
34        SELECT fpt_declarations.rownumber,
35            fpt_declarations.objname,
36            fpt_declarations.declaration
37        FROM fpt_declarations
38        WHERE declaration ILIKE '%' || string || '%'
39        LOOP
40            RAISE INFO '% % % %', rpad(CAST(index AS TEXT), 4), rpad(col.objname, 21),
41                rpad(CAST(col.rownumber AS TEXT), 14), rpad(col.declaration, 45);
42            index = index + 1;
43        END LOOP;
44 END;
45 $$;

```

5 Вывод скрипта

No.	Имя объекта	# строки	Текст
-----	-------------	----------	-------

1	auditlog	1	INSERT INTO AUDIT(EMP_ID, ENTRY_DATE) VAL
2	auditlog	10	CREATE OR REPLACE FUNCTION public.auditlog()
3	example_trigger	1	CREATE TRIGGER example_trigger AFTER INSERT O

6 Вывод

Данная лабораторная работа помогла мне изучить системный каталог PostgreSQL.