

Факультет программной инженерии и компьютерной техники

Распределённые системы хранения данных

Лабораторная работа №1

Вариант №31139

Преподаватель: Николаев Владимир Вячеславович

Выполнил: Кульбако Артемий Юрьевич, Сараев Владислав Витальевич Р33112

# Задание

Используя сведения из представлений словаря данных, получить информацию обо всех триггерах, назначенных на указанную таблицу схемы.

COLUMN NAME TRIGGER NAME

----------------------- ------------------------

COLUMN1 TRIGGER1

COLUMN2 TRIGGER2

...

Программу оформить в виде процедуры.

# Выполнение

*CREATE OR REPLACE PACKAGE* triggers\_util *AS  
 PROCEDURE* get\_filtered(tbl\_name *IN VARCHAR2*, scheme\_name *IN VARCHAR2*, trg\_cursor *OUT SYS\_REFCURSOR*);  
 *PROCEDURE* print\_format(rc *IN SYS\_REFCURSOR*);  
*END*;  
/  
  
*CREATE OR REPLACE PACKAGE BODY* triggers\_util *AS  
 -- получить курсор с именами колонок и назначенных на них триггеров для запрошенного сочетания таблицы и схемы  
 PROCEDURE* get\_filtered(tbl\_name *IN VARCHAR2*, scheme\_name *IN VARCHAR2*, trg\_cursor *OUT SYS\_REFCURSOR*) *AS  
 BEGIN  
 OPEN* trg\_cursor *FOR  
 SELECT* column\_name, trigger\_name *FROM* dba\_trigger\_cols  
 *WHERE* table\_name *LIKE* tbl\_name *AND* trigger\_owner *LIKE* scheme\_name *AND* table\_owner *LIKE* scheme\_name;  
 *END*;  
  
 *-- вывести курсор с колонками и триггерами  
 PROCEDURE* print\_format(rc *IN SYS\_REFCURSOR*) *AS* names *VARCHAR2*(32);  
 *triggers VARCHAR2*(32);  
 *BEGIN* DBMS\_OUTPUT.PUT\_LINE(RPAD('COLUMN NAME', 23) || ' ' || RPAD('TRIGGER NAME', 23));  
 DBMS\_OUTPUT.PUT\_LINE(RPAD('-', 23, '-') || ' ' || RPAD('-', 23, '-'));  
 *FETCH* rc *INTO* names, *triggers*;  
 *IF* rc%*FOUND THEN  
 LOOP  
 FETCH* rc *INTO* names, *triggers*;  
 *EXIT WHEN* rc%*NOTFOUND*;  
 DBMS\_OUTPUT.PUT\_LINE(RPAD(names, 23) || ' ' || RPAD(*triggers*, 23));  
 *END LOOP*;  
 *ELSE* DBMS\_OUTPUT.PUT\_LINE('triggers for requested table and schema combination are did not exists');  
 *END IF*;  
 *END*;  
*END*;  
/  
  
*-- проверить, что искомый объект существует в базе данных  
CREATE OR REPLACE FUNCTION* is\_exist(where\_ *IN VARCHAR2*, what\_ *IN VARCHAR2*, field\_ *IN VARCHAR2*) *RETURN BOOLEAN AS* accepted\_results *NUMBER*;  
*BEGIN  
 EXECUTE IMMEDIATE* 'SELECT COUNT(\*) FROM ' || where\_ ||  
 ' WHERE ' || field\_ || q'[ = ']' || what\_ || q'[']'  
 *INTO* accepted\_results;  
 *IF* accepted\_results = 0 *THEN RETURN FALSE*;  
 *ELSE RETURN TRUE*;  
 *END IF*;  
*END*;  
/  
  
*SET SERVEROUTPUT ON*;  
*SET FEEDBACK OFF*;  
*SET VERIFY OFF*;  
*ACCEPT s CHAR PROMPT* 'Type scheme name: '  
*ACCEPT* t *CHAR PROMPT* 'Type table name: '  
*DECLARE* result\_cursor *SYS\_REFCURSOR*;  
 scheme\_name *VARCHAR2*(32) := '&s';  
 table\_name *VARCHAR2*(32) := '&t';  
 scheme\_input\_exception *EXCEPTION*;  
 table\_input\_exception *EXCEPTION*;  
*BEGIN  
 IF* scheme\_name *IS NULL THEN RAISE* scheme\_input\_exception; *END IF*;  
 *IF* table\_name *IS NULL THEN RAISE* table\_input\_exception; *END IF*;  
 *IF* is\_exist('dba\_users', scheme\_name, 'username') = *TRUE THEN  
 IF* is\_exist('dba\_tables', table\_name, 'table\_name') = *TRUE THEN* triggers\_util.get\_filtered(table\_name, scheme\_name, result\_cursor);  
 triggers\_util.print\_format(result\_cursor);  
 *ELSE* DBMS\_OUTPUT.PUT\_LINE('Typed table didnt exist in database');  
 *END IF*;  
 *ELSE* DBMS\_OUTPUT.PUT\_LINE('Typed schema didnt exist in database');  
 *END IF*;  
*EXCEPTION  
 WHEN* scheme\_input\_exception *THEN* DBMS\_OUTPUT.PUT\_LINE('Typed schema name is empty');  
 *WHEN* table\_input\_exception *THEN* DBMS\_OUTPUT.PUT\_LINE('Typed table name is empty');  
*END*;  
/  
  
*SHOW ERRORS*;

# Вывод

Основная сложность работы – освоение достаточно сложного после pgSQL языка PL/SQL ввиду его возраста:

1. работа с курсорами сложнее чем автораспаковка столбцов таблицы, но занимает меньше места в коде
2. Присутствие deprecated типов данных (VARCHAR)
3. Наличие процедур с «возвращаемым» значением, концептуальный смысл которых не велик
4. Не самая удобная стандартная командная оболочка ( - ; / )

Тем не менее, наличие огромного количества представлений для всех возможный случаев делает Oracle удобным для настройки и администрирования, что видимо и позволило ей стать одной из самых используемых СУБД, а к недостаткам синтаксиса можно привыкнуть и/или списать на долгую историю продукта.