CON BUCLE LOOP

CREATE OR REPLACE PROCEDURE cursor\_demo()

LANGUAGE PLPGSQL... (Tiempo restante: 4 KB)

Expandir

message.txt4 KB

﻿

CURSORES

CON BUCLE LOOP

CREATE OR REPLACE PROCEDURE cursor\_demo()

LANGUAGE PLPGSQL

AS $$

DECLARE

ent1 int;

ent2 int;

cadena varchar (15);

cur1 cursor FOR SELECT id, dato FROM t1;

cur2 cursor FOR SELECT i FROM t2;

BEGIN

open cur1;

open cur2;

loop

fetch cur1 INTO ent1,cadena; --Extrae datos (con fetch) y los vuelca en las variables

fetch cur2 INTO ent2;

exit when not found; --Cuando ya no encuentra nada

if ent1 < ent2 then

INSERT INTO t3 VALUES (ent1,cadena );

else

INSERT INTO t3 VALUES (ent2,cadena);

end if;

end loop;

close cur1;

close cur2;

END;

$$

BUCLE WHILE

CREATE OR REPLACE PROCEDURE cursor\_demo()

LANGUAGE PLPGSQL

AS $$

DECLARE

ent1 int;

ent2 int;

cadena varchar (15);

cur1 cursor FOR SELECT id, dato FROM t1;

cur2 cursor FOR SELECT i FROM t2;

BEGIN

open cur1;

open cur2;

fetch cur1 INTO ent1,cadena; --en While se pone antes y despues

fetch cur2 INTO ent2;

WHILE (FOUND) LOOP --Cuando ya no encuentra nada

if ent1 > ent2 then

INSERT INTO t3 VALUES (ent1,cadena );

else

INSERT INTO t3 VALUES (ent2,cadena);

end if;

fetch cur1 INTO ent1,cadena; --Extrae datos (con fetch) y los vuelca en las variables

fetch cur2 INTO ent2;

end loop;

close cur1;

close cur2;

END;

$$

MÁS EJEMPLOS DE CURSOR

BUCLE LOOP

DO

$$

DECLARE

cur1 CURSOR FOR SELECT ID,dato FROM t1;

cur2 CURSOR FOR SELECT i FROM t2;

num1 int;

num2 int;

cadena varchar (16);

BEGIN

OPEN cur1;

OPEN cur2;

LOOP

FETCH cur1 INTO num1, cadena;

FETCH cur2 INTO num2;

EXIT WHEN NOT FOUND;

IF num1 < num2 THEN

RAISE NOTICE '% (t1) es menor que % (t2)',num1,num2;

ELSEIF num1 > num2 THEN

RAISE NOTICE '% (t1) es mayor que % (t2)',num1,num2;

ELSE

RAISE NOTICE '% (t1) es igual que % (t2)',num1,num2;

END IF;

END LOOP;

CLOSE cur1;

CLOSE cur2;

END;

$$

BUCLE WHILE

DO

$$

DECLARE

cur1 CURSOR FOR SELECT ID,dato FROM t1;

cur2 CURSOR FOR SELECT i FROM t2;

num1 int;

num2 int;

cadena varchar (16);

BEGIN

OPEN cur1;

OPEN cur2;

--LOOP

FETCH cur1 INTO num1, cadena;

FETCH cur2 INTO num2;

WHILE (FOUND) LOOP

--FETCH cur1 INTO num1, cadena;

--FETCH cur2 INTO num2;

--EXIT WHEN NOT FOUND;

IF num1 < num2 THEN

RAISE NOTICE '% (t1) es menor que % (t2)',num1,num2;

ELSEIF num1 > num2 THEN

RAISE NOTICE '% (t1) es mayor que % (t2)',num1,num2;

ELSE

RAISE NOTICE '% (t1) es igual que % (t2)',num1,num2;

END IF;

FETCH cur1 INTO num1, cadena;

FETCH cur2 INTO num2;

END LOOP;

CLOSE cur1;

CLOSE cur2;

END;

$$

BUCLE FOR

DO

$$

DECLARE

cur1 CURSOR FOR SELECT ID FROM t1;

cur2 CURSOR FOR SELECT i FROM t2;

num2 int;

BEGIN

open cur2;

FOR num1 IN cur1 LOOP

FETCH cur2 INTO num2;

RAISE NOTICE '% (t1) i % (t2)',num1,num2;

END LOOP;

CLOSE cur2;

END;

$$

SEGUNDO EJEMPLO DO

DO

$$

DECLARE

cur1 CURSOR FOR SELECT \* FROM t1;

cur2 CURSOR FOR SELECT i FROM t2;

num2 int;

BEGIN

open cur2;

FOR num1 IN cur1 LOOP

FETCH cur2 INTO num2;

IF num1.id < num2 THEN

RAISE NOTICE '% (t1) es menor que % (t2)',num1.id,num2;

ELSEIF num1.id > num2 THEN

RAISE NOTICE '% (t1) es mayor que % (t2)',num1.id,num2;

ELSE

RAISE NOTICE '% (t1) es igual que % (t2)',num1.id,num2;

END IF;

END LOOP;

CLOSE cur2;

END;

$$