**FOR LOOP**

Iterates over range of values or data coming from a table.

FOR counter\_name IN start\_value .. end\_value BY stepping

LOOP

Statements

END LOOP;

--Sum odd values up to a max number

CREATE OR REPLACE FUNCTION fn\_for\_test(max\_num int)

RETURNS int AS

$body$

--Put variables here

DECLARE

tot\_sum INT DEFAULT 0;

BEGIN

FOR i IN 1 .. max\_num BY 2

LOOP

tot\_sum := tot\_sum + i;

END LOOP;

RETURN tot\_sum;

END;

$body$

LANGUAGE plpgsql

SELECT fn\_for\_test(5);

You can also count in reverse with FOR i IN REVERSE max\_num .. 1 BY 2

**For Loops with Result Sets Blocks and Raise Notice**

--Use a bloc to test this

DO

$$

DECLARE

rec record;

BEGIN

FOR rec IN

SELECT first\_name, last\_name

FROM sales\_person

LIMIT 5

LOOP

--Outputs info to Messages

RAISE NOTICE '%, %', rec.first\_name, rec.last\_name;

END LOOP;

END;

$$

LANGUAGE plpgsql

**For Each and Arrays**

FOREACH var IN ARRAY array\_name

-- Print all values in the array

DO

$body$

DECLARE

arr1 int[] := array[1,2,3];

i int;

BEGIN

FOREACH i IN ARRAY arr1

LOOP

RAISE NOTICE '%', i;

END LOOP;

END;

$body$

LANGUAGE plpgsql

**While Loop**

-- Sums values as long as a condition is true

DO

$body$

DECLARE

j INT DEFAULT 1;

tot\_sum INT DEFAULT 0;

BEGIN

WHILE j <= 10

LOOP

tot\_sum := tot\_sum + j;

j := j + 1;

END LOOP;

RAISE NOTICE '%', tot\_sum;

END;

$body$

LANGUAGE plpgsql

**Continue**

--Prints the odd numbers from 1 to 10

DO

$$

DECLARE

i int DEFAULT 1;

BEGIN

LOOP

i := i + 1;

EXIT WHEN i > 10;

CONTINUE WHEN MOD(i, 2) = 0;

RAISE NOTICE 'Num : %', i;

END LOOP;

END;

$$

LANGUAGE plpgsql

**Return Inventory Value by Supplier**

CREATE OR REPLACE FUNCTION fn\_get\_supplier\_value(the\_supplier varchar)

RETURNS varchar AS

$body$

DECLARE

supplier\_name varchar;

price\_sum numeric;

BEGIN

SELECT product.supplier, SUM(item.price)

INTO supplier\_name, price\_sum

FROM product, item

WHERE product.supplier = the\_supplier

GROUP BY product.supplier;

RETURN CONCAT(supplier\_name, ' Inventory Value : $', price\_sum);

END;

$body$

LANGUAGE plpgsql

SELECT fn\_get\_supplier\_value('Nike');