**CURSORS**

Cursors are used to step backwards or forwards through rows of data. They can be pointed at a row and then select, update or delete. Cursor gets data, pushes it to another language for processing operations that add, edit, or delete.

Cursors are first declared defining the selection options to be used. It is then opened so it retrieves the data. Then individual rows can be fetched. After use the cursor is closed freeing memory. When needed the cursor can be used as needed.

**Example with Cursors**

DO

$body$

DECLARE

msg text DEFAULT '';

rec\_customer record;

-- Declare cursor with customer data

cur\_customers CURSOR

FOR

SELECT \* FROM customer;

BEGIN

-- Open cursor

OPEN cur\_customers;

LOOP

-- Fetch records from cursor

FETCH cur\_customers INTO rec\_customer;

-- Loop until nothing more is found

EXIT WHEN NOT FOUND;

-- Concatenates all customer names together

msg := msg || rec\_customer.first\_name || ' ' || rec\_customer.last\_name || ', ';

END LOOP;

RAISE NOTICE 'Customers : %', msg;

END;

$body$

**Using Cursors with Functions**

-- Cursurs & Functions

-- Function returns a list of all customers in provided state

CREATE OR REPLACE FUNCTION fn\_get\_cust\_by\_state(c\_state varchar)

RETURNS text

LANGUAGE PLPGSQL

AS

$body$

DECLARE

cust\_names text DEFAULT '';

rec\_customer record;

cur\_cust\_by\_state CURSOR (c\_state varchar)

FOR

SELECT

first\_name, last\_name, state

FROM customer

WHERE state = c\_state;

BEGIN

-- Open cursor and pass the parameter

OPEN cur\_cust\_by\_state(c\_state);

LOOP

-- Move row of data to rec\_customer

FETCH cur\_cust\_by\_state INTO rec\_customer;

-- Loop until nothing more is found

EXIT WHEN NOT FOUND;

-- Concat customer name for each row

cust\_names := cust\_names || rec\_customer.first\_name || ' ' || rec\_customer.last\_name || ', ';

END LOOP;

-- Close cursor

CLOSE cur\_cust\_by\_state;

RETURN cust\_names;

END;

$body$

SELECT fn\_get\_cust\_by\_state('CA');