-- REV - Fetch and execute each line of a cursor.

CREATE OR REPLACE PROCEDURE REV (

P\_CURSOR SYS\_REFCURSOR,

P\_PRINT PLS\_INTEGER DEFAULT 0,

P\_ERROR VARCHAR2 DEFAULT 'ERROR : '

)

AS

L\_Q VARCHAR2(32767);

BEGIN

FETCH P\_CURSOR INTO L\_Q;

WHILE P\_CURSOR%FOUND

LOOP

BEGIN

EXECUTE IMMEDIATE L\_Q;

IF P\_PRINT = 1 THEN

DBMS\_OUTPUT.PUT\_LINE(L\_Q);

END IF;

EXCEPTION WHEN OTHERS THEN

IF P\_PRINT = 1 THEN

DBMS\_OUTPUT.PUT\_LINE(P\_ERROR || ' ' || SQLERRM || ' in ' || SYS\_CONTEXT('USERENV', 'CURRENT\_SCHEMA') || '.');

ELSE

DBMS\_OUTPUT.PUT\_LINE(P\_ERROR || ' ' || L\_Q || '; ' || SQLERRM || ' in ' || SYS\_CONTEXT('USERENV', 'CURRENT\_SCHEMA') || '.');

END IF;

END;

FETCH P\_CURSOR INTO L\_Q;

END LOOP;

END;

/

-- IS\_INTEGER - Return 1 if the input is an integer, 0 if it is not.

CREATE OR REPLACE FUNCTION IS\_INTEGER (theNumber IN VARCHAR2) RETURN NUMBER AS

V\_INT INTEGER;

V\_NUM NUMBER;

V\_CHAR VARCHAR2(30);

BEGIN

V\_INT := TO\_NUMBER(LTRIM(theNumber, ' 0'));

V\_NUM := TO\_NUMBER(LTRIM(theNumber, ' 0'));

V\_CHAR := LTRIM(THENUMBER, ' 0');

IF V\_INT = V\_NUM AND V\_CHAR = TO\_CHAR(V\_INT) THEN

RETURN 1;

ELSE

RETURN 0;

END IF;

RETURN 0;

EXCEPTION

WHEN OTHERS THEN

RETURN 0;

END;

/

-- PON\_CONCAT - Return the text of a single column cursor rolled up into one line and separated by P\_IN\_BETWEEN.

/\*

Returns

1

2

3

as 1, 2, 3.

\*/

CREATE OR REPLACE FUNCTION PON\_CONCAT(P\_CURSOR SYS\_REFCURSOR, P\_IN\_BETWEEN VARCHAR2 DEFAULT ', ')

RETURN VARCHAR2

IS

L\_RETURN VARCHAR2(32767);

L\_TEMP VARCHAR2(32767);

BEGIN

LOOP

FETCH P\_CURSOR INTO L\_TEMP;

EXIT WHEN P\_CURSOR%NOTFOUND;

L\_RETURN := L\_RETURN || P\_IN\_BETWEEN || L\_TEMP;

END LOOP;

RETURN SUBSTR(L\_RETURN, LENGTH(P\_IN\_BETWEEN) + 1, 32767);

CLOSE P\_CURSOR;

END;

/