DECLARE

-- Define variables to hold the cursor parameters and record data

v\_dept\_id N\_RollCall.dept\_id%TYPE; -- Example parameter, could be any column

v\_rollcall\_id N\_RollCall.rollcall\_id%TYPE;

v\_rollcall\_time N\_RollCall.rollcall\_time%TYPE;

-- Define a cursor to fetch records based on the parameter

CURSOR c\_rollcall (p\_dept\_id IN N\_RollCall.dept\_id%TYPE) IS

SELECT rollcall\_id, rollcall\_time

FROM N\_RollCall

WHERE dept\_id = p\_dept\_id;

-- Record to hold data fetched from the cursor

v\_rollcall\_record c\_rollcall%ROWTYPE;

BEGIN

-- Initialize the dept\_id parameter (this can be set dynamically based on user input)

v\_dept\_id := 101; -- Example department ID for merging records

-- Open the cursor with the parameter value

OPEN c\_rollcall(v\_dept\_id);

-- Loop through all records returned by the cursor

LOOP

FETCH c\_rollcall INTO v\_rollcall\_record;

EXIT WHEN c\_rollcall%NOTFOUND; -- Exit loop when no more records

-- Check if the record already exists in O\_RollCall

SELECT COUNT(\*) INTO v\_rollcall\_id

FROM O\_RollCall

WHERE rollcall\_id = v\_rollcall\_record.rollcall\_id;

-- If the record doesn't exist, insert it into O\_RollCall

IF v\_rollcall\_id = 0 THEN

INSERT INTO O\_RollCall (rollcall\_id, rollcall\_time, dept\_id)

VALUES (v\_rollcall\_record.rollcall\_id, v\_rollcall\_record.rollcall\_time, v\_dept\_id);

DBMS\_OUTPUT.PUT\_LINE('Record inserted for RollCall ID: ' || v\_rollcall\_record.rollcall\_id);

ELSE

DBMS\_OUTPUT.PUT\_LINE('Record with RollCall ID: ' || v\_rollcall\_record.rollcall\_id || ' already exists. Skipping.');

END IF;

END LOOP;

-- Close the cursor after processing all records

CLOSE c\_rollcall;

DBMS\_OUTPUT.PUT\_LINE('Merge process completed.');

EXCEPTION

WHEN OTHERS THEN

-- Exception handling for unexpected errors

DBMS\_OUTPUT.PUT\_LINE('Error: ' || SQLERRM);

END;

/