

## Fun with SQL Dojo IDUG NA – “Fun with SQLPL”

- 1) Create a stored procedure that has the following features:
  - 1) 1 varchar(13) output parameter
  - 2) The stored procedure returns the value of 'Hello World' in the output parameter.

```
CREATE PROCEDURE PROB1 ( OUT OUT_V1 VARCHAR(13))
  DYNAMIC RESULT SETS 0
P1: BEGIN
  SET OUT_V1 = 'Hello World';
END P1
```

- 2) Modify the previous stored procedure with the following features:
- 1) 1 integer input parameter
  - 2) 1 varchar(13) output parameter
  - 3) The procedure will return a value in the output parameter of 'Hello World' if the input parameter is 1, 'Goodbye World' if the input parameter is 2, and 'Invalid Entry' if the input parameter has any other value.

```
CREATE PROCEDURE PROB2 ( IN IN_SET_MSG INT
                        , OUT OUT_V1 VARCHAR(13))
  DYNAMIC RESULT SETS 0
P1: BEGIN
  CASE IN_SET_MSG
    WHEN 1 THEN SET OUT_V1 = 'Hello World';
    WHEN 2 THEN SET OUT_V1 = 'Goodbye World';
    ELSE       SET OUT_V1 = 'Invalid Entry';
  END CASE;
END P1
```

- 3) Modify the previous stored procedure with the following features:
  - 1) Add an input parameter with a datatype of char(3)
  - 2) Add a cursor to the stored procedure that reads the employee table (columns EMPNO, LASTNAME, and SALARY) for a WORKDEPT value equal to the new input parameter
  - 3) Return the cursor to the caller

```
CREATE PROCEDURE PROB3 ( IN  IN_SET_MSG INT
                        , IN  IN_DEPTNO CHAR(03)
                        , OUT OUT_V1 VARCHAR(13))
    DYNAMIC RESULT SETS 1
P1: BEGIN

    DECLARE EMP_C1 CURSOR WITH RETURN FOR
        SELECT E.EMPNO, E.LASTNAME, E.SALARY
        FROM   EMP E
        WHERE  E.WORKDEPT = IN_DEPTNO;

    CASE IN_SET_MSG
        WHEN 1 THEN SET OUT_V1 = 'Hello World';
        WHEN 2 THEN SET OUT_V1 = 'Goodbye World';
        ELSE      SET OUT_V1 = 'Invalid Entry';
    END CASE;

    OPEN EMP_C1;

END P1
```

- 4) Modify the previous stored procedure with the following features:
- 1) Use the department number input parameter to do an existence check against the department table before processing the cursor against the employee table. If the department does not exist then fail the call to the stored procedure with SQLSTATE '38001'. If the department does exist then continue with returning the employee cursor.

```
CREATE PROCEDURE PROB4 ( IN  IN_SET_MSG INT
                        , IN  IN_DEPTNO CHAR(03)
                        , OUT OUT_V1 VARCHAR(13))
    DYNAMIC RESULT SETS 1
P1: BEGIN

    DECLARE WS_DEPTNO CHAR(03) DEFAULT '   ';
    DECLARE SQLCODE INTEGER DEFAULT 0;

    DECLARE EMP_C1 CURSOR WITH RETURN FOR
        SELECT E.EMPNO, E.LASTNAME, E.SALARY
        FROM   EMP E
        WHERE  E.WORKDEPT = IN_DEPTNO;

    CASE IN_SET_MSG
        WHEN 1 THEN SET OUT_V1 = 'Hello World';
        WHEN 2 THEN SET OUT_V1 = 'Goodbye World';
        ELSE       SET OUT_V1 = 'Invalid Entry';
    END CASE;

    SELECT D.DEPTNO INTO WS_DEPTNO
    FROM   DEPT D
    WHERE  D.DEPTNO = IN_DEPTNO;

    IF SQLCODE = 100 THEN SIGNAL SQLSTATE '38001'
        SET MESSAGE_TEXT = 'Department does not exist';
    END IF;

    OPEN EMP_C1;

END P1
```

- 5) If the solution to #4 does not use a condition handler in the existence check processing then modify it to use a condition handler to handle the 'NOT FOUND' condition and fail the call just like in #4.

```
CREATE PROCEDURE PROB5 ( IN  IN_SET_MSG INT
                        , IN  IN_DEPTNO CHAR(03)
                        , OUT OUT_V1 VARCHAR(13))
    DYNAMIC RESULT SETS 1
P1: BEGIN

    DECLARE WS_DEPTNO CHAR(03) DEFAULT '  ';
    DECLARE SQLCODE INTEGER DEFAULT 0;
    DECLARE WS_DEPT_FOUND CHAR(03) DEFAULT 'N';

    DECLARE EMP_C1 CURSOR WITH RETURN FOR
        SELECT E.EMPNO, E.LASTNAME, E.SALARY
        FROM   EMP E
        WHERE  E.WORKDEPT = IN_DEPTNO;

    DECLARE CONTINUE HANDLER FOR NOT FOUND
        H1: BEGIN
            CASE WS_DEPT_FOUND
            WHEN 'N' THEN SIGNAL SQLSTATE '38001'
                        SET MESSAGE_TEXT = 'Department does not exist';
            END CASE;
        END H1;

    CASE IN_SET_MSG
    WHEN 1 THEN SET OUT_V1 = 'Hello World';
    WHEN 2 THEN SET OUT_V1 = 'Goodbye World';
    ELSE      SET OUT_V1 = 'Invalid Entry';
    END CASE;

    SELECT D.DEPTNO INTO WS_DEPTNO
    FROM   DEPT D
    WHERE  D.DEPTNO = IN_DEPTNO;

    SET WS_DEPT_FOUND = 'Y';

    OPEN EMP_C1;

END P1
```

- 6) Modify the previous stored procedure such that after the department existence check and before opening the employee cursor you add a section that gives employees raises. In this section open a cursor against the project table for the input department value. Give the responsible employee for each project a raise of 10%. If an employee is responsible for more than one project the raise is cumulative. In other words, they get 10% more on top of their salary for each project. You will need to code a loop for this. Once raises are issued return the employee cursor.

```
CREATE PROCEDURE PROB6 ( IN IN_SET_MSG INT
                        , IN IN_DEPTNO CHAR(03)
                        , OUT OUT_V1 VARCHAR(13))
    DYNAMIC RESULT SETS 1
P1: BEGIN

    DECLARE WS_DEPTNO CHAR(03) DEFAULT '  ';
    DECLARE SQLCODE INTEGER DEFAULT 0;
    DECLARE WS_DEPT_FOUND CHAR(03) DEFAULT 'N';
    DECLARE EOF_CD CHAR(01) DEFAULT 'N';
    DECLARE WS_RESPEMP CHAR(06) DEFAULT '      ';

    DECLARE EMP_C1 CURSOR WITH RETURN FOR
        SELECT E.EMPNO, E.LASTNAME, E.SALARY
        FROM EMP E
        WHERE E.WORKDEPT = IN_DEPTNO;

    DECLARE PROJ_C1 CURSOR FOR
        SELECT P.RESPEMP
        FROM PROJ P
        WHERE P.DEPTNO = IN_DEPTNO;

    DECLARE CONTINUE HANDLER FOR NOT FOUND
    H1: BEGIN
        CASE WS_DEPT_FOUND
            WHEN 'N' THEN SIGNAL SQLSTATE '38001'
                SET MESSAGE_TEXT = 'Department does not exist';
            ELSE SET EOF_CD = 'Y';
        END CASE;
    END H1;

    CASE IN_SET_MSG
        WHEN 1 THEN SET OUT_V1 = 'Hello World';
        WHEN 2 THEN SET OUT_V1 = 'Goodbye World';
        ELSE SET OUT_V1 = 'Invalid Entry';
    END CASE;
```

```

SELECT D.DEPTNO INTO WS_DEPTNO
FROM   DEPT D
WHERE  D.DEPTNO = IN_DEPTNO;

SET WS_DEPT_FOUND = 'Y';

OPEN PROJ_C1;

FETCH PROJ_C1 INTO WS_RESPEMP;

WHILE EOF_CD = 'N' DO
    UPDATE EMP
        SET SALARY = SALARY * 1.1
        WHERE EMPNO = WS_RESPEMP;
    FETCH PROJ_C1 INTO WS_RESPEMP;
END WHILE;

OPEN EMP_C1;

END P1

```

Another example for a solution to #6:

```

CREATE PROCEDURE PROB6A ( IN   IN_SET_MSG INT
                        , IN   IN_DEPTNO CHAR(03)
                        , OUT  OUT_V1 VARCHAR(13))
    DYNAMIC RESULT SETS 1
P1: BEGIN

    DECLARE WS_DEPTNO CHAR(03) DEFAULT '   ';
    DECLARE SQLCODE INTEGER DEFAULT 0;

    DECLARE EMP_C1 CURSOR WITH RETURN FOR
        SELECT E.EMPNO, E.LASTNAME, E.SALARY
        FROM   EMP E
        WHERE  E.WORKDEPT = IN_DEPTNO;

    CASE IN_SET_MSG
        WHEN 1 THEN SET OUT_V1 = 'Hello World';
        WHEN 2 THEN SET OUT_V1 = 'Goodbye World';
        ELSE       SET OUT_V1 = 'Invalid Entry';
    END CASE;

```

```
BEGIN
  DECLARE EXIT HANDLER FOR NOT FOUND
  BEGIN
    SIGNAL SQLSTATE '38001'
    SET MESSAGE_TEXT = 'Department does not exist';
  END;

  SELECT D.DEPTNO INTO WS_DEPTNO
  FROM   DEPT D
  WHERE  D.DEPTNO = IN_DEPTNO;
END;

FOR VLOOP AS
  SELECT P.RESPEMP AS RESPEMP
  FROM   PROJ P
  WHERE  P.DEPTNO = IN_DEPTNO
  DO
    UPDATE EMP
    SET SALARY = SALARY * 1.1
    WHERE EMPNO = VLOOP.RESPEMP;
END FOR ;

OPEN EMP_C1;

END P1
```