Procedural Parameter Modes

```
        ┌─────────────────────────┐
        │ Procedure               │
        │ □  IN parameter         │
Calling │ □  OUT parameter        │
environment □  IN  OUT parameter   │
        │                         │
        │ ┌─────────────────────┐ │
        │ │ (DECLARE)           │ │
        │ │ ┌─────────────────┐ │ │
        │ │ └─────────────────┘ │ │
        │ │ BEGIN               │ │
        │ │ ┌─────────────────┐ │ │
        │ │ └─────────────────┘ │ │
        │ │ EXCEPTION           │ │
        │ │ ┌─────────────────┐ │ │
        │ │ └─────────────────┘ │ │
        │ │ END;                │ │
        │ └─────────────────────┘ │
        └─────────────────────────┘
```
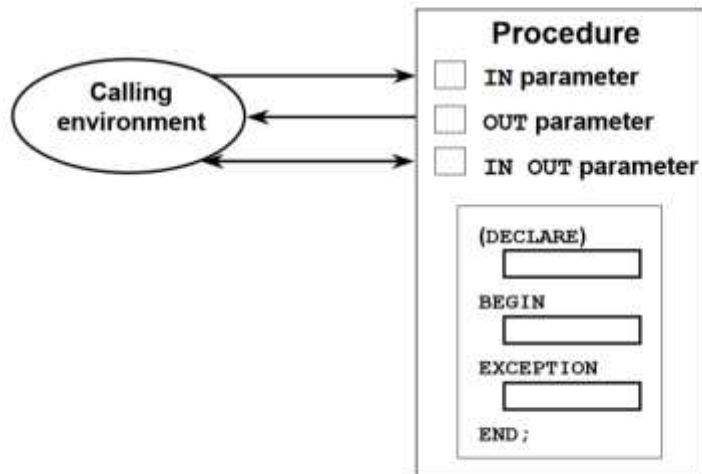
IN Parameters

1.CREATE OR REPLACE PROCEDURE raise_salary

(p_id IN employees.employee_id%TYPE)

IS

BEGIN

UPDATE employees

SET salary = salary * 1.10

WHERE employee_id = p_id;

END raise_salary;

OUT Parameters

2.CREATE OR REPLACE PROCEDURE query_emp

(p_id IN employees.employee_id%TYPE,

p_name OUT employees.last_name%TYPE,

p_salary OUT employees.salary%TYPE,

p_comm OUT employees.commission_pct%TYPE)

IS

BEGIN

SELECT last_name, salary, commission_pct

INTO p_name, p_salary, p_comm

FROM employees

WHERE employee_id = p_id;

END query_emp;

Viewing OUT Parameters

3.VARIABLE g_name VARCHAR2(25)

VARIABLE g_sal NUMBER

VARIABLE g_comm NUMBER

EXECUTE query_emp(171, :g_name, :g_sal, :g_comm)

PRINT g_name

IN OUT Parameters

4.CREATE OR REPLACE PROCEDURE format_phone

(p_phone_no IN OUT VARCHAR2)

IS

BEGIN

p_phone_no := '(' || SUBSTR(p_phone_no,1,3) ||

')' || SUBSTR(p_phone_no,4,3) ||

'-' || SUBSTR(p_phone_no,7);

END format_phone;

5.VARIABLE g_phone_no VARCHAR2(15)

BEGIN

:g_phone_no := '8006330575';

END;

PRINT g_phone_no

EXECUTE format_phone (:g_phone_no)

PRINT g_phone_no

DEFAULT Option for Parameters

6.CREATE OR REPLACE PROCEDURE add_dept

(p_name IN departments.department_name%TYPE

DEFAULT 'unknown',

```
      p_loc IN departments.location_id%TYPE
      DEFAULT 1700)
IS
BEGIN
INSERT INTO departments(department_id,
department_name, location_id)
VALUES (departments_seq.NEXTVAL, p_name, p_loc);
END add_dept;
PASSING VALUE
BEGIN
add_dept;
add_dept ('TRAINING', 2500);
add_dept ( p_loc => 2400, p_name =>'EDUCATION');
add_dept ( p_loc => 1200) ;
END;
```

Handled Exceptions

```
7.CREATE PROCEDURE p2_ins_dept(p_locid NUMBER) IS
v_did NUMBER(4);
BEGIN
DBMS_OUTPUT.PUT_LINE('Procedure p2_ins_dept started');
INSERT INTO departments VALUES (5, 'Dept 5', 145, p_locid);
SELECT department_id INTO v_did FROM employees
WHERE employee_id = 999;
END;


CREATE PROCEDURE p1_ins_loc(p_lid NUMBER, p_city VARCHAR2)
IS
v_city VARCHAR2(30); v_dname VARCHAR2(30);
BEGIN
DBMS_OUTPUT.PUT_LINE('Main Procedure p1_ins_loc');
```

```
INSERT INTO locations (location_id, city) VALUES (p_lid, p_city);

SELECT city INTO v_city FROM locations WHERE location_id = p_lid;

DBMS_OUTPUT.PUT_LINE('Inserted city '||v_city);

DBMS_OUTPUT.PUT_LINE('Invoking the procedure p2_ins_dept ...');

p2_ins_dept(p_lid);

EXCEPTION

WHEN NO_DATA_FOUND THEN

DBMS_OUTPUT.PUT_LINE('No such dept/loc for any employee');

END;
```