**//1**

CREATE OR REPLACE FUNCTION IS\_PRIME(NUM IN NUMBER)

RETURN BOOLEAN

IS

I NUMBER;

BEGIN

IF NUM = 2 THEN

RETURN TRUE;

ELSIF NUM < 2 THEN

RETURN FALSE;

ELSIF MOD(NUM,2) = 0 THEN

RETURN FALSE;

END IF;

FOR I IN 3..FLOOR(SQRT(NUM))

LOOP

IF MOD(NUM, I) = 0 THEN

RETURN FALSE;

END IF;

END LOOP;

RETURN TRUE;

END IS\_PRIME;

SET SERVEROUTPUT ON;

DECLARE

result BOOLEAN;

BEGIN

result := IS\_PRIME(13);

DBMS\_OUTPUT.PUT\_LINE('Is 13 prime? ' || CASE WHEN result THEN 'Yes' ELSE 'No' END);

END;

**//2**

CREATE OR REPLACE FUNCTION FACTORIAL\_NUM(NUM IN NUMBER)

RETURN NUMBER

IS

RESULT NUMBER := 1;

A NUMBER;

BEGIN

FOR A IN 2..FLOOR(NUM)

LOOP

RESULT := RESULT\*A;

END LOOP;

RETURN RESULT;

END FACTORIAL\_NUM;

SET SERVEROUTPUT ON;

DECLARE

result NUMBER;

BEGIN

result := FACTORIAL\_NUM(4);

DBMS\_OUTPUT.PUT\_LINE('FACTORIAL NUM OF 4 = ' || result );

END;

**//3**

CREATE OR REPLACE FUNCTION CALCULATE\_BONUS(SALARY IN NUMBER, JOIN\_YEAR IN NUMBER)

RETURN NUMBER

IS

BONUS NUMBER;

YEARS\_OF\_SERVICE NUMBER := EXTRACT(YEAR FROM SYSDATE) - JOIN\_YEAR;

BEGIN

IF YEARS\_OF\_SERVICE < 1 THEN

BONUS := 0;

ELSIF YEARS\_OF\_SERVICE < 5 THEN

BONUS := SALARY \* 0.05;

ELSIF YEARS\_OF\_SERVICE < 10 THEN

BONUS := SALARY \* 0.10;

ELSE

BONUS := SALARY \* 0.15;

END IF;

RETURN BONUS;

END CALCULATE\_BONUS;

SET SERVEROUTPUT ON;

DECLARE

BONUS\_SALARY NUMBER;

BEGIN

BONUS\_SALARY := CALCULATE\_BONUS(2000, 7);

DBMS\_OUTPUT.PUT\_LINE('THIS EMPLOYEE BONUS SALARY IS ' || BONUS\_SALARY);

END;

**//4**

CREATE OR REPLACE PROCEDURE CALCULATE\_AVERAGE\_SALARY(DEPT\_ID IN NUMBER, AVG\_SALARY OUT NUMBER)

IS

BEGIN

SELECT AVG(SALARY)

INTO AVG\_SALARY

FROM HR.EMPLOYEES

WHERE DEPARTMENT\_ID = DEPT\_ID;

IF AVG\_SALARY IS NULL THEN

AVG\_SALARY := 0;

END IF;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

AVG\_SALARY := 0;

WHEN OTHERS THEN

RAISE;

END CALCULATE\_AVERAGE\_SALARY;

SET SERVEROUTPUT ON;

DECLARE

AVERAGE\_SALARY NUMBER;

BEGIN

CALCULATE\_AVERAGE\_SALARY(20, AVERAGE\_SALARY);

DBMS\_OUTPUT.PUT\_LINE('THIS DEPARTMENT AVERAGE SALARY IS ' || AVERAGE\_SALARY);

END;

**//5**

CREATE OR REPLACE PROCEDURE COUNT\_TOTAL\_EMPLOYEES(J\_ID IN VARCHAR2, TOTAL\_EMPLOYEES OUT NUMBER)

IS

BEGIN

SELECT COUNT(\*)

INTO TOTAL\_EMPLOYEES

FROM HR.EMPLOYEES

WHERE JOB\_ID = J\_ID;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

TOTAL\_EMPLOYEES := 0;

WHEN OTHERS THEN

RAISE;

END COUNT\_TOTAL\_EMPLOYEES;

SET SERVEROUTPUT ON;

DECLARE

TOTAL\_EMPLOYEES NUMBER;

BEGIN

COUNT\_TOTAL\_EMPLOYEES('AD\_VP', TOTAL\_EMPLOYEES);

DBMS\_OUTPUT.PUT\_LINE('THIS JOB TITLE TOTAL EMPLOYEES ARE ' || TOTAL\_EMPLOYEES);

END;

**//6**

CREATE OR REPLACE PROCEDURE UPDATE\_EMPLOYEE\_SALARY(P\_EMPLOYEE\_ID IN NUMBER, P\_PERFORMANCE\_RATING IN VARCHAR2)

IS

V\_NEW\_SALARY NUMBER;

BEGIN

IF P\_PERFORMANCE\_RATING = 'Excellent' THEN

SELECT SALARY \* 1.10 INTO V\_NEW\_SALARY

FROM HR.EMPLOYEES

WHERE EMPLOYEE\_ID = P\_EMPLOYEE\_ID;

ELSIF P\_PERFORMANCE\_RATING = 'Good' THEN

SELECT SALARY \* 1.05 INTO V\_NEW\_SALARY

FROM HR.EMPLOYEES

WHERE EMPLOYEE\_ID = P\_EMPLOYEE\_ID;

ELSIF P\_PERFORMANCE\_RATING = 'Average' THEN

SELECT SALARY INTO V\_NEW\_SALARY

FROM HR.EMPLOYEES

WHERE EMPLOYEE\_ID = P\_EMPLOYEE\_ID;

ELSE

RAISE\_APPLICATION\_ERROR(-20001, 'Invalid performance rating.');

END IF;

UPDATE HR.EMPLOYEES

SET SALARY = V\_NEW\_SALARY

WHERE EMPLOYEE\_ID = P\_EMPLOYEE\_ID;

COMMIT;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('No employee found with ID: ' || P\_EMPLOYEE\_ID);

WHEN OTHERS THEN

RAISE;

END UPDATE\_EMPLOYEE\_SALARY;

SET SERVEROUTPUT ON;

BEGIN

UPDATE\_EMPLOYEE\_SALARY(101, 'Excellent');

END;