**Class Test 14**

*To solve the following use the scott schema*

1. Create a cursor that can output all the data from salgrade table.
2. Create a record that can output the id, name, job, and hiredate of the employee whose Id is 7654.
3. Create a trigger in such a way that whenever a new row is inserted into the salgrade table an output ‘New Row Added’ is generated.
4. Create a package that contains a procedure which can display the grade whose losal and hisal is passed as its parameter.

**Answer:**

1.

**DECLARE**

**CURSOR** salgrade\_cursor **IS**

**SELECT** \* **FROM** salgrade;

*-- Declare variables to hold salgrade columns*

grade NUMBER;

losal NUMBER;

hisal NUMBER;

**BEGIN**

**OPEN** salgrade\_cursor;

LOOP

**FETCH** salgrade\_cursor **INTO** grade, losal, hisal;

EXIT **WHEN** salgrade\_cursor%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('Grade: ' || grade || ', Low Salary: ' || losal || ', High Salary: ' || hisal);

**END** LOOP;

**CLOSE** salgrade\_cursor;

**END**;

/

2.

**DECLARE**

emp\_id NUMBER := 7654;

emp\_rec scott.emp%ROWTYPE;

emp\_name VARCHAR2(50);

emp\_job VARCHAR2(50);

emp\_hiredate DATE;

**BEGIN**

**SELECT** ename, job, hiredate **INTO** emp\_name, emp\_job, emp\_hiredate

**FROM** emp

**WHERE** empno = emp\_id;

DBMS\_OUTPUT.PUT\_LINE('Employee ID: ' || emp\_id || ', Name: ' || emp\_name ||

', Job: ' || emp\_job || ', Hire Date: ' || emp\_hiredate);

**END**;

/

3.

**DECLARE**

**CURSOR** salgrade\_cursor **IS**

**SELECT** \* **FROM** salgrade;

*-- Declare variables to hold salgrade columns*

grade NUMBER;

losal NUMBER;

hisal NUMBER;

**BEGIN**

**OPEN** salgrade\_cursor;

LOOP

**FETCH** salgrade\_cursor **INTO** grade, losal, hisal;

EXIT **WHEN** salgrade\_cursor%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('Grade: ' || grade || ', Low Salary: ' || losal || ', High Salary: ' || hisal);

**END** LOOP;

**CLOSE** salgrade\_cursor;

**END**;

/

4.

**CREATE** **OR** **REPLACE** PACKAGE grade\_package **AS**

**PROCEDURE** display\_grade(losal\_in **IN** NUMBER, hisal\_in **IN** NUMBER);

**END** grade\_package;

/

**CREATE** **OR** **REPLACE** PACKAGE BODY grade\_package **AS**

**PROCEDURE** display\_grade(losal\_in **IN** NUMBER, hisal\_in **IN** NUMBER) **IS**

grade\_val NUMBER;

**BEGIN**

**SELECT** grade **INTO** grade\_val

**FROM** salgrade

**WHERE** losal = losal\_in **AND** hisal = hisal\_in;

DBMS\_OUTPUT.PUT\_LINE('Grade for Low Salary ' || losal\_in || ' and High Salary ' || hisal\_in || ' is: ' || grade\_val);

EXCEPTION

**WHEN** NO\_DATA\_FOUND **THEN**

DBMS\_OUTPUT.PUT\_LINE('No matching grade found for provided salary range.');

**END** display\_grade;

**END** grade\_package;

/