

Assignment_PLSQL_2

Neelesh Pandey

Employee ID : 91904

Ans -1.

```
set serveroutput on;

declare

dept_no number;

updated number;

begin

dept_no := &dept_no;

update employee set salary = salary+(salary*10/100) where depart_id = dept_no;

updated := SQL%ROWCOUNT;

dbms_output.put_line('Total numbers of record updated are ' || updated);

end;
```

```
UPDATED
OLD SALARY : 102
NEW SALARY : 112
SALARY DIFFERENCE : 10
Total numbers of record updated are 1

PL/SQL procedure successfully completed.
```

Ans -2.

SIMPLE LOOP

```
set serveroutput on;

declare

cursor records is select ename, sal from emp fetch first 5 rows only;
```

```

sal_grade number;

results records%rowtype;

begin open records;

loop fetch records into results;

exit when records%notfound;

select grade into sal_grade from salgrade where losal<= results.sal and hisal >= results.sal;

dbms_output.put_line('name '||results.ename||' sal is '||results.sal||' grade id '||sal_grade);

end loop;

close records;

end;

/

01403. 00000 - "no data found"
*Cause:      No data was found from the objects.
*Action:     There was no data from the objects which may be due to end of fetch.

```

WHILE LOOP

```

set serveroutput on;

declare cursor records is select ename, sal from emp fetch first 5 rows only;

sal_grade number;

results records%rowtype;

begin

open records;

fetch records into results;

while records%found loop

dbms_output.put_line('name '||results.ename||' sal is '||results.sal||' grade id '||sal_grade);

fetch records into results;

```

```

select grade into sal_grade from salgrade where losal<= results.sal and hisal >= results.sal;

end loop;

close records;

end;

/

```

FOR LOOP

```

set serveroutput on;

declare cursor records is select ename, sal from emp fetch first 5 rows only;

sal_grade number;

results records%rowtype;

begin

for results in records loop select grade into sal_grade from salgrade where losal<= results.sal and hisal
>= results.sal;

dbms_output.put_line('name '||results.ename||' sal is '||results.sal||' grade id '||sal_grade);

end loop;

end;

/

```

Ans -3.

```

SET SERVEROUTPUT ON;

DECLARE

CURSOR DEP_ALL IS

SELECT ENAME, SAL FROM EMP WHERE JOB_DESCRIPTION = 'CLERK';

V_ENAME EMP.ENAME%TYPE;

V_SAL EMP.SAL%TYPE;

BEGIN

OPEN DEP_ALL;

```

```

LOOP

FETCH DEP_ALL INTO V_ENAME,V_SAL;

EXIT WHEN DEP_ALL%NOTFOUND;

DBMS_OUTPUT.PUT_LINE(V_ENAME||' SALARY IS '||V_SAL);

END LOOP;

CLOSE DEP_ALL;

END;

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

/

```

Ans-4.

```

DECLARE

CURSOR EMP_CUR(EMP_JOB EMP.JOB_DESCRIPTION%TYPE) IS SELECT ENAME,DEPTNO,SAL FROM EMP
WHERE JOB_DESCRIPTION = EMP_JOB;

EMP_RECORD EMP_CUR%ROWTYPE;

BEGIN

OPEN EMP_CUR('&EMP_JOB');

LOOP

FETCH EMP_CUR INTO EMP_RECORD;

EXIT WHEN EMP_CUR%NOTFOUND;

dbms_output.put_line(EMP_RECORD.ENAME||' '||EMP_RECORD.DEPTNO||' '||EMP_RECORD.SAL);

END LOOP;

CLOSE EMP_CUR;

END;

```

/

Rohan 15 650000

PL/SQL procedure successfully completed.

Ans-5.

DECLARE

EMP_DEPTNO EMP.DEPTNO%TYPE ;

EMP_SAL EMP.SAL%TYPE;

EMP_JOB EMP.JOB_DESCRIPTION%TYPE;

CURSOR EMP_CUR IS SELECT SAL FROM EMP;

BEGIN

OPEN EMP_CUR;

LOOP

FETCH EMP_CUR INTO EMP_SAL;

EXIT WHEN EMP_CUR%NOTFOUND;

IF (EMP_JOB = 'CLERK' AND EMP_DEPTNO = 10) THEN UPDATE EMP SET SAL = (SAL + (SAL*20/100));

DBMS_OUTPUT.PUT_LINE(EMP_SAL);

ELSIF (EMP_JOB = 'MANAGER' AND EMP_DEPTNO = 20) THEN UPDATE EMP SET SAL = (SAL + (SAL*5/100));

DBMS_OUTPUT.PUT_LINE(EMP_SAL);

ELSE UPDATE EMP SET SAL = (SAL + (SAL*10/100));

DBMS_OUTPUT.PUT_LINE(EMP_SAL);

```

END IF;

END LOOP;

CLOSE EMP_CUR;

END;

/

```

```

40000
30000
250000
350000
650000
20000

PL/SQL procedure successfully completed.

```

Ans-6.

```

SET SERVEROUTPUT ON;

DECLARE

CURSOR DEPT_CUR IS SELECT DEPTNO, DNAME FROM DEPT; CURSOR EMP_CUR(DEPT_NO NUMBER) IS
SELECT * FROM EMP WHERE DEPTNO = DEPT_NO;

BEGIN

FOR I IN DEPT_CUR LOOP

DBMS_OUTPUT.PUT_LINE('DEPT_NO = ' || I.DEPTNO || ' ' || 'DEPT_NAME = ' || I.DNAME);

DBMS_OUTPUT.PUT_LINE('-----');

FOR J IN EMP_CUR(I.DEPTNO) LOOP DBMS_OUTPUT.PUT_LINE('NAME = ' || J.ENAME || ' ' || 'EMP_NO = '
|| J.EMPNO || ' ' || 'SALARY = ' || J.SAL || ' ' || 'JOB = ' || J.JOB);

DBMS_OUTPUT.PUT_LINE('-----');

END LOOP;

END LOOP;

END;

```

/

Ans-7.

DECLARE

EMP_SAL EMP.SAL%TYPE := '&SAL';

EMP_NAME EMP.ENAME%TYPE;

BEGIN

SELECT ENAME INTO EMP_NAME FROM EMP WHERE SAL = EMP_SAL;

DBMS_OUTPUT.PUT_LINE(EMP_NAME || ' HAS SALARY ' || EMP_SAL);

EXCEPTION

WHEN TOO_MANY_ROWS

THEN

DBMS_OUTPUT.PUT_LINE('MORE THAN ONE EMPLOYEE WITH SALARY ' || EMP_SAL);

INSERT INTO MESSAGES(ERR_MESSAGE) VALUES ('MORE THAN ONE EMPLOYEE WITH SALARY OF ' || EMP_SAL);

WHEN NO_DATA_FOUND THEN DBMS_OUTPUT.PUT_LINE('NO EMPLOYEE WITH SALARY ' || EMP_SAL);

INSERT INTO MESSAGES(ERR_MESSAGE) VALUES ('NO EMPLOYEE WITH SALARY OF ' || EMP_SAL);

WHEN OTHERS THEN DBMS_OUTPUT.PUT_LINE('SOME OTHER ERROR OCCURED');

INSERT INTO MESSAGES(ERR_MESSAGE) VALUES ('SOME OTHER ERROR OCCURED');

END;

/

```
Abhishek HAS SALARY 53147
```

```
PL/SQL procedure successfully completed.
```

Ans-8.

DECLARE

DEPT_NO DEPT.DEPTNO%TYPE:=&DNO;

DEPT_NAME DEPT.DNAME%TYPE:='&DNAME';

DEPT_LOC DEPT.LOC%TYPE:='&LOC';

DUPLICATE_DEPT EXCEPTION;

PRAGMA EXCEPTION_INIT(DUPLICATE_DEPT,-00001);

BEGIN

INSERT INTO DEPT VALUES(DEPT_NO,DEPT_NAME,DEPT_LOC);

DBMS_OUTPUT.PUT_LINE('RECORD INSERTED SUCCEESFULLY');

EXCEPTION

WHEN DUPLICATE_DEPT

THEN

DBMS_OUTPUT.PUT_LINE('DUPLICATE DEPARTMENT FOUND | INSERTION FAILED');

END;

/

Ans-9.

SET SERVEROUTPUT ON;

DECLARE

DUPLICATE_PRESIDENT EXCEPTION;

JOB_COUNT NUMBER;


```

CURSOR EMP_CUR IS SELECT COUNT(*) FROM EMP WHERE JOB = 'PRESIDENT' GROUP BY JOB;

BEGIN

OPEN EMP_CUR;

FETCH EMP_CUR INTO JOB_COUNT;

DBMS_OUTPUT.PUT_LINE(JOB_COUNT);

IF JOB_COUNT > 2 THEN RAISE DUPLICATE_PRESIDENT;

END IF;

DBMS_OUTPUT.PUT_LINE('NO DUPLICATE VALUES');

EXCEPTION WHEN DUPLICATE_PRESIDENT THEN DBMS_OUTPUT.PUT_LINE('MORE THAN ONE
PRESIDENT');

CLOSE EMP_CUR;

END;

/

NO DUPLICATE VALUES

PL/SQL procedure successfully completed.

```

Ans-10.

```

SET SERVEROUTPUT ON;

DECLARE

EMPNO1 NUMBER := &EMPNO1;

EMPNO2 NUMBER := &EMPNO2;

SAL1 NUMBER;

SAL2 NUMBER;

BEGIN

```

```
SELECT SAL INTO SAL1 FROM EMP WHERE EMPNO = EMPNO1;

UPDATE EMP SET SAL = SAL + (SAL * 10/100) WHERE EMPNO = EMPNO1;

EXCEPTION WHEN NO_DATA_FOUND

THEN

DBMS_OUTPUT.PUT_LINE('SORRY, THERE IS NO EMPLOYEE WITH EMPLOYEE NUMBER ' || EMPNO1);

END;
```

```
SELECT SAL INTO SAL2 FROM EMP WHERE EMPNO = EMPNO2;

UPDATE EMP SET SAL = SAL + (SAL * 20/100) WHERE EMPNO = EMPNO2;

EXCEPTION WHEN NO_DATA_FOUND

THEN

DBMS_OUTPUT.PUT_LINE('SORRY, THERE IS NO EMPLOYEE WITH EMPLOYEE NUMBER ' || EMPNO2);

END;
```

/