***PL/SQL LAB 3***

**1)**CREATE OR REPLACE PROCEDURE proc1(deptno number)

AS

sal NUMBER(6);

BEGIN

SELECT MAX(staff\_sal) INTO sal FROM staff\_master

WHERE dept\_code=deptno;

DBMS\_OUTPUT.PUT\_LINE('Maximum Salary is :'||sal);

END proc1;

**--Calling proc1**

SQL> exec proc1(10);

Maximum Salary is :32000

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**2) --Creating Age Function**

CREATE OR REPLACE FUNCTION

AGE\_FUNC(dob DATE)

RETURN NUMBER

AS

Age NUMBER;

BEGIN

Age:=ROUND((sysdate-dob)/365);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE(‘Please provide age ‘);

RETURN Age;

END AGE\_FUNC;

**--Calling Age Function**

SELECT AGE\_FUNC(‘18-AUG-1993’) FROM DUAL;

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**3.** **-- creating procedure proc2**

CREATE OR REPLACE PROCEDURE proc2(staffcode number)

AS

**-- Declaring Variables**

name staff\_master.staff\_name%TYPE;

NO\_DATA EXCEPTION;

BEGIN

SELECT staff\_name INTO name FROM staff\_master WHERE staff\_code=staffcode;

IF(name IS NULL) THEN

RAISE NO\_DATA;

END IF;

UPDATE staff\_master SET staff\_name=UPPER(staff\_name) WHERE staff\_code=staffcode;

**-- Exception Block**

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE(‘staff name not found with the given staff number’);

WHEN NO\_DATA THEN

DBMS\_OUTPUT.PUT\_LINE(‘staff name is NULL’);

END proc2;

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**4. -- creating procedure proc3**

CREATE OR REPLACE PROCEDURE

Manager\_details (scode NUMBER)

AS

**-- declaring variables**

Code staff\_master.staff\_code%TYPE;

name staff\_master.staff\_name%TYPE;

dcode department\_master.dept\_code%TYPE;

name2 staff\_master.staff\_name%TYPE;

BEGIN

SELECT

s.staff\_code,s.staff\_name,s.dept\_code,m.staff\_name INTO code,name,dcode,name2

FROM staff\_master s,staff\_master m

WHERE s.mgr\_code=m.staff\_code AND s.staff\_code=scode;

DBMS\_OUTPUT.PUT\_LINE(‘staff\_code’|| code);

DBMS\_OUTPUT.PUT\_LINE(‘staff\_name’|| name);

DBMS\_OUTPUT.PUT\_LINE(‘staff\_dept\_code’|| dcode);

DBMS\_OUTPUT.PUT\_LINE(‘Manager Name’|| name2);

**-- Exception Block**

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE(‘No manager Found with the given staff code’);

END Manager\_details;

**-- Declaring**

DECLARE

scode NUMBER(10);

BEGIN

Scode:=&staff\_code;

Manager\_details(scode);

END;

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**5 . -- creating function compute**

CREATE OR REPLACE FUNCTION compute(scode NUMBER)

RETURN NUMBER

AS  
**-- Declare Variables and Cursor**

name varchar2(10);

salary NUMBER(6);

hdate DATE;

service NUMBER(2);

ctc NUMBER(7);

counter NUMBER(3);

NO\_EMP\_DATA EXCEPTION;

CURSOR eCur is

SELECT staff\_name,staff\_sal,hiredate

FROM staff\_master

WHERE staff\_code=scode;

BEGIN

OPEN eCur;

DBMS\_OUTPUT.PUT\_LINE(‘------------------------‘);

LOOP

FETCH eCur INTO name,salary,hdate;

EXIT WHEN eCur%NOTFOUND;

Salary:=salary+salary\*(0.15+0.2+0.08);

Service:=ROUND((sysdate-hdate)/365);

DBMS\_OUTPUT.PUT\_LINE(‘ ‘||name||’ ’||salary||’ ’||service);

**-- Checking service**

IF(service<1) THEN

ctc:=salary;

ELSIF(service>=1 AND service<2) THEN

ctc:=salary+(salary\*0.1);

ELSIF(service>=2 AND service<4) THEN

ctc:=salary+(salary\*0.2);

ELSIF(service>4) THEN

ctc:=salary+(salary\*0.3);

END IF;

END LOOP;

DBMS\_OUTPUT.PUT\_LINE(‘------------------------‘);

counter:=eCur%ROWCOUNT;

IF(counter=0) THEN

RAISE NO\_EMP\_DATA;

END IF;

CLOSE eCur;

RETURN ctc;

**-- Exception Block**

EXCEPTION

WHEN NO\_EMP\_DATA THEN

DBMS\_OUTPUT.PUT\_LINE(‘No Employee Details Found’);

RETURN 0;

END compute;

**-- Calling the Function**

DECLARE

Ctc NUMBER(7);

Scode NUMBER(6);

BEGIN

Scode:=&staffcode;

Ctc:=compute(scode);

DBMS\_OUTPUT.PUT\_LINE(‘Salary of’||scode||’is’||ctc);

END;

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**6. -- creating procedure proc6 for getting status**

CREATE OR REPLACE PROCEDURE proc6

AS

**-- Declaring Variables and Cursor**

CODE staff\_master.staff\_code%TYPE;

NAME staff\_master.staff\_name%TYPE;

DNAME department\_master.dept\_name%TYPE;

DESIGNATION designation\_master.design\_name%TYPE;

SALARY staff\_master.staff\_sal%TYPE;

DCODE staff\_master.dept\_code%TYPE;

Avgsal NUMBER(5);

NO\_STAFF\_DETAILS EXCEPTION;

Status varchar(7);

CURSOR C is SELECT s.staff\_name,d.dept\_name,de.design\_name,s.staff\_sal,s.dept\_code

FROM staff\_master s,department\_master d,designation\_master de

WHERE s.dept\_code=d.dept\_code AND s.design\_code=de.design\_code;

BEGIN

OPEN C;

FETCH C INTO NAME,DNAME,DESIGNATION, SALARY,DCODE;

**-- Raising Exception when no details found**

IF(C%ROWCOUNT=0) THEN

RAISE NO\_STAFF\_DETAILS;

END IF;

IF(C%ROWCOUNT>0) THEN

DBMS\_OUTPUT.PUT\_LINE(C%ROWCOUNT);

ELSE

DBMS\_OUTPUT.PUT\_LINE(‘NOT EXISTS’);

END IF;

WHILE C%FOUND

LOOP

FETCH C INTO name,dname,designation,salary,dcode;

SELECT AVG(staff\_sal) INTO avgsal

FROM staff\_master

WHERE dept\_code=dcode

GROUP BY dept\_code;

IF(avgSal>salary) THEN

STATUS:=’LESSER’;

ELSIF(avgSal<salary) THEN

STATUS:=’GREATER’;

ELSE

STATUS:=’EQUAL’;

END IF;

DBMS\_OUTPUT.PUT\_LINE(‘\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*’);

DBMS\_OUTPUT.PUT\_LINE(‘staffname:-’||name); DBMS\_OUTPUT.PUT\_LINE(‘department:-’|| DNAME);

DBMS\_OUTPUT.PUT\_LINE(‘designation:-’|| designation);

DBMS\_OUTPUT.PUT\_LINE(‘Average Sal:-’|| avgsal);

DBMS\_OUTPUT.PUT\_LINE(‘Status:-’||status);

DBMS\_OUTPUT.PUT\_LINE(‘\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*’);

END LOOP;

CLOSE C;

**-- Exception Block**

EXCEPTION  
 WHEN NO\_STAFF\_DETAILS THEN

DBMS\_OUTPUT.PUT\_LINE(‘No staff details found’);

END proc6;

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**7. --Creating table staff\_master\_back**

CREATE TABLE staff\_master\_back AS SELECT \* FROM staff\_master;

**-- creating procedure update\_sal to update the salaries**

CREATE OR REPLACE PROCEDURE update\_sal(scode IN NUMBER) AS

Salary NUMBER;

Exp NUMBER;

Hire DATE;

New\_Sal NUMBER;

BEGIN

SELECT staff\_sal INTO salary FROM staff\_master WHERE staff\_code=scode;

Exp:=(sysdate-hire)/365;

IF exp<2 THEN

New\_sal:=salary+0;

ELSIF exp>=2 AND exp<5 THEN

New\_Sal:=salary+(0.2\*salary);

ELSE

New\_Sal:=salary+(0.25\*salary);

END IF;

UPDATE staff\_master SET staff\_sal=new\_sal WHERE staff\_code=scode;

**-- Exception Block**

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE(‘No salary found with the given staff code’);

END update\_sal;

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**8)-- creating procedure update\_sal**

CREATE OR REPLACE PROCEDURE books(bcode In NUMBER)

AS

SCODE number;

Stud\_code NUMBER;

Sname varchar2(20);

Stud\_name varchar2(20);

Idate date;

Design varchar2(20);

Exp\_Ret\_Date DATE;

Act\_Ret\_Date DATE;

CURSOR cName IS SELECT staff\_code,student\_code,book\_issue\_date,book\_expected\_return\_date,book\_actual\_return\_date

FROM book\_transaction

WHERE book\_code=bcode;

BEGIN

IF cname%ISOPEN THEN

NULL;

ELSE

OPEN cName;

END IF;

FETCH cName INTO scode,stud\_code,idate,exp\_ret\_date,act\_ret\_date;

WHILE cName%FOUND

LOOP

IF(act\_ret\_date IS NULL) THEN

IF(scode IS NULL) THEN

SELECT student\_name INTO stud\_name FROM student\_master WHERE student\_code=stud\_code;

DBMS\_OUTPUT.PUT\_LINE(‘Student- ‘||stud\_code||’ ‘||stud\_name||’ ‘||idate||’ ‘||exp\_ret\_date);

END IF;

IF(stud\_code IS NULL) THEN

SELECT s.staff\_name,d.design\_name INTO sname,design FROM staff\_master s,designation\_master d WHERE

staff\_code=scode AND s.design\_code=d.design\_code;

DBMS\_OUTPUT.PUT\_LINE(‘Staff-‘||scode||’ ‘||sname||’ ‘||idate||’ ‘||design||’ ‘||exp\_ret\_Date);

END IF;

END IF;

FETCH cname INTO scode,stud\_code,idate,exp\_ret\_date,act\_ret\_date;

END LOOP;

CLOSE cName;

END books;

-- Calling The Procedure

BEGIN

Books(&code);

END;

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**9) --Function for years of experience calculation**

CREATE OR REPLACE FUNCTION cal(hiredate1 date)

RETURN NUMBER

AS

Exp NUMBER(6);

BEGIN

Exp:=ROUND((SYSDATE-hiredate1)/365);

RETURN exp;

END cal;

**--Procedure for additional allowance for the staff based on experience**

CREATE OR REPLACE PROCEDURE proc9(hd date)

AS

Exp1 number(5);

Aa number(9); **--additional allowance variable**

Id NUMBER(9);

BEGIN

SELECT staff\_code INTO id FROM staff\_master WHERE hiredate=hd;

Exp1:=cal(hd); **--Calling Function cal()**

Aa:=exp1\*3000;

DBMS\_OUTPUT.PUT\_LINE(‘staff id:-’||id);

DBMS\_OUTPUT.PUT\_LINE(‘date of joining:-’||hd);

DBMS\_OUTPUT.PUT\_LINE(‘experience:-’||exp1);

DBMS\_OUTPUT.PUT\_LINE(‘additional allowance:-’||aa);

INSERT INTO staff\_allowance\_table values(id,hd,exp1,aa);

END proc9;

**--Procedure Executing**

Exec proc9(’17-jan-02’);