1. **Write an anonymous pl/sql block to insert, update and delete data from EMP table.**

set serveroutput on;

begin

DBMS\_OUTPUT.PUT\_LINE('Insert data in a employee table');

insert into employee(empno,ename,job,mgr,hiredate,sal,deptno)

values(null,null,null,null,null,null,null);

DBMS\_OUTPUT.PUT\_LINE('update data in a employee table');

update employee set empno=empno+100;

DBMS\_OUTPUT.PUT\_LINE('update data in a employee table');

Delete from student where empno=4713;

END;

/

1. **Write a pl/sql block to select the employee’s name from EMP table in a variable using %TYPE attribute and print it.**
2. **Write a PL/SQL block for the below logic using both IF-ELSE statement.**
   1. **If employee salary is less than 2000, then print Underpaid**
   2. **If employee salary is between 2000 and 4000, then print Perfectly paid.**
   3. **If employee salary is greater than 4000, then print Highly paid.**

set serveroutput on;

declare

a number(10):=3000;

begin

if(a<2000) then

dbms\_output.put\_line('Underpaid');

elsif(a>=2000 and a<=4000) THEN

dbms\_output.put\_line('perfectly paid');

elsif(a<4000) then

dbms\_output.put\_line('Highly paid');

else

dbms\_output.put\_line('None of the values is matching');

END IF;

END;

/

1. **Write a PL/SQL block for the below logic using both CASE statement.**
   1. **If employee hire date is before 1982, then print old employee**
   2. **If employee hire date is between 1982 and 1986, then print recently joined**
   3. **If employee hire date is after 1986, then print new employee.**

DECLARE

BEGIN

CASE TRUE

WHEN date<1982 THEN

DBMS\_OUTPUT.PUT\_LINE(‘Old employee’);

WHEN date>=1982 and date<=1986 then

DBMS\_OUTPUT.PUT\_LINE(‘recently joined);

WHEN date>=1986 then

DBMS\_OUTPUT.PUT\_LINE(‘new employee’);

ELSE

END CASE;

s

1. **Using for loop, print numbers from 1 to 10.**

set serveroutput on;

DECLARE

n number:= 10;

BEGIN

DBMS\_OUTPUT.PUT\_LINE ('The first '||n||' numbers are: ');

for i in 1..n loop

dbms\_output.put(i||' ');

END LOOP;

dbms\_output.new\_line;

END;

/

1. **Using while loop, print numbers 1 to 10 in reverse order.**

set serveroutput on;

Declare

I number:=10;

Begin

While (I>=1)

loop

Dbms\_output.put\_line(I);

I:=I-1;

End loop;

End

1. **Using a record, write pl/sql block to insert a new employee data.**

set serveroutput on;

begin

DBMS\_OUTPUT.PUT\_LINE('Insert new employee data into employee table');

insert into employee(empno,ename,job,mgr,hiredate,sal,deptno)

values(11,’Raj’,’Soft\_En’,23,to\_date(‘12/06/1989’),3000,12);

END;

/

1. **Fetch and print all the employees name using cursors.**

set serveroutput on;

DECLARE

z\_empid employees.employee\_id%TYPE;

z\_empname employees.first\_name%TYPE;

z\_salary employees.salary%TYPE;

CURSOR employee\_cursor IS

SELECT employee\_id,

first\_name,

salary

FROM employees;

BEGIN

OPEN employee\_cursor;

LOOP

FETCH employee\_cursor

INTO z\_empid,

z\_empname,

z\_salary;

EXIT

WHEN employee\_cursor%NOTFOUND;

IF (z\_salary > 8000) THEN

dbms\_output.Put\_line(z\_empid

|| ' '

|| z\_empname

|| ' '

|| z\_salary);

ELSE

dbms\_output.Put\_line(z\_empname

|| ' salary is less then 8000');

END IF;

END LOOP;

CLOSE employee\_cursor;

END;

/