1. **Write a PL/SQL statement that will print your name on the screen**

begin

dbms\_output.put\_line('Syed Nouman');

end;

Syed Nouman

Statement processed.

1. **Write a PL/SQL statement that will declare & Initialize a variable and print its value.**

declare

a number(5);

BEGIN

a:= 27;

dbms\_output.put\_line(' a = '||a);

end;

a = 27

Statement processed.

1. **Write a PL/SQL statement that will take a number at run time. Print “Pass” if number is greater than or equal to 50 or “Fail” otherwise.**

declare

a number(5);

BEGIN

a:=:a;

if (a>=50) then

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

else

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

END IF;

end;

**output:**

**If I enter 40**

Fail

Statement processed.

1. **Write a procedure which will take 2 numbers as parameters and display their product.**

i)

create procedure product(

a number ,

b number

) is

BEGIN

dbms\_output.put\_line(a\*b);

END;

ii)

BEGIN

product(5,6);

END;

30

Statement processed.

1. **Write a procedure which will insert a row in dept table**

i)

create or replace procedure insert\_dept(

dno dept.deptno%type,

dname dept.dname%type,

loc dept.loc%type

) is

BEGIN

insert into dept values(dno,dname,loc);

END;

ii)

BEGIN

insert\_dept(50,'Liberary','Lahore');

END;

iii)

select \* from dept

|  |  |  |
| --- | --- | --- |
| **DEPTNO** | **DNAME** | **LOC** |
| 10 | ACCOUNTING | NEW YORK |
| 20 | RESEARCH | DALLAS |
| 30 | SALES | CHICAGO |
| 40 | OPERATIONS | BOSTON |
| 50 | Liberary | Lahore |

1. **Write a function which will take a year as a parameter.it should return “leap year” if the year is a leap year or “Not leap year “otherwise.**

i)

create or replace function isLeapYear

(

n number

)

return varchar is

begin

if(mod(n,400)=0) then

return 'Leap Year';

elsif (mod(n,100)=0) then

return 'Not Leap Year';

elsif (mod(n,4)=0) then

return 'Leap Year';

else

return 'Not Leap Year';

end if;

end;

ii)

select isLeapYear(2000) from dual;

|  |
| --- |
| **ISLEAPYEAR(2000)** |
| Leap Year |

1. **Write a function which takes an integer as a parameter and returns “Even Number” if the number is even and “Odd Number ”if the number is odd. You are not allowed to use MOD() function.**

i)

create or replace function EvenOrOdd(n number)

return varchar is

num number;

begin

num:=n;

loop

if(num<=1)

then exit;

end if;

num:=num-2;

end loop;

if (num=0) then

return 'Even Number';

else

return 'Odd Number';

end if;

end;

ii)

select EvenOrOdd(10) from dual

|  |
| --- |
| **EVENORODD(10)** |
| Even Number |

1. **Write a function which will take empno (emp table) as a parameter and returns the salary of the employee**

i)

create or replace function getSalary(eno emp.empno%type)

return number is

num number;

begin

select sal into num from emp where eno=empno;

return num;

end;

ii)

select getSalary(7876) as sal from dual

|  |
| --- |
| **SAL** |
| 1100 |

1. **Write a procedure which will take empno as a parameter and display the ename and salary of the employee.**

i)

create or replace procedure display(eno emp.empno%type)

is

enam emp.ename%type;

sal emp.sal%type;

begin

select ename,sal into enam,sal from emp where eno=empno;

dbms\_output.put\_line('name='||enam);

dbms\_output.put\_line('sal='||sal);

end;

ii)

begin

display(7902);

end;

name=FORD

sal=3000

Statement processed.

1. **Write a PL/SQL block to differenciate between CHAR and VARCHAR2 datatype.**

declare

ch char(5):='hello';

vch varchar2(10):='h';

begin

dbms\_output.put\_line(length(ch));

dbms\_output.put\_line(length(vch));

end;

5

1

Statement processed.

1. **Write a PL/SQL block to calculate the sum of comm and salary of an employee getting input of it’s empno.**

i)

create or replace function sumSalComm(eno emp.empno%type)

return number is

comm emp.comm%type;

sal emp.sal%type;

begin

select sal,comm into sal,comm from emp where eno=empno;

return sal+nvl(comm,0);

end;

ii)

select sumSalComm(7499) from dual;

|  |
| --- |
| **SUMSALCOMM(7499)** |
| 1900 |

1. **Write a code to insert data by getting user input into customer table you created in last lab.**

i)

create or replace procedure insert\_customer(

id customer.CUSTOMER\_ID%type,

name customer.CUST\_NAME%type,

city customer.CITY%type,

grade customer.GRADE%type,

SalID customer.SALESMAN\_ID%type

) is

BEGIN

insert into customer values (id,name,city,grade,SalID);

END;

ii)

begin

insert\_customer(3050,'Nouman','Lahore',400,5008);

end;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CUSTOMER\_ID** | **CUST\_NAME** | **CITY** | **GRADE** | **SALESMAN\_ID** |
| 3002 | Nick Rimando | New York | 100 | 5001 |
| 3003 | jozy Altidor | Moscow | 200 | 5007 |
| 3009 | Geoff Cameron | Berlin | 100 | 5003 |
| 3004 | Fabian Johnson | Paris | 300 | 5006 |
| 3007 | Brad Davis | New York | 200 | 5001 |
| 3005 | Graham Zusi | California | 200 | 5002 |
| 3008 | Julian Green | London | 300 | 5002 |
| 3050 | Nouman | Lahore | 400 | 5008 |

1. **Write a SQL query to get the last record of the table emp**

create or replace procedure last\_record()

is

empno emp.empno%type,

ename emp.ename%type,

job emp.job%type,

mgr emp.mgr%type,

hiredate emp.hiredate%type,

sal emp.sal%type,

comm emp.comm%type,

deptno emp.deptno%type

BEGIN

select max(empno),ename,job,mgr,hiredate,sal,comm,deptno

into empno,ename,job,mgr,hiredate,sal,comm,deptno from emp

where empno=empno;

dbms\_output.put\_line(empno);

dbms\_output.put\_line(ename);

dbms\_output.put\_line(job);

dbms\_output.put\_line(mgr);

dbms\_output.put\_line(hiredate);

dbms\_output.put\_line(sal);

dbms\_output.put\_line(comm);

dbms\_output.put\_line(deptno);

END;

1. **Write a PL/SQL program to convert a temperature in scale Fahrenheit to Celsius and vice versa.**
2. **Write a function to get the marks of a student as input and showing Grade as output. (<50 F, <60 E, <70 D, <75 C, <80 B, >=85 A)**

i)

create or replace function findGrade(marks number)

return char is

begin

if(marks<50) then return 'F';

elsif(marks<60) then return 'E';

elsif(marks<70) then return 'D';

elsif(marks<75) then return 'C';

elsif(marks<80) then return 'B';

elsif(marks>=85) then return 'A';

end if;

end;

ii)

select findGrade(72)from dual

|  |
| --- |
| **FINDGRADE(72)** |
| C |