Week-7

# 6. Write a program to find the largest number from the given three numbers.

SQL>edit largest.sql

declare

a number:=0;

b number:=0;

c number:=0;

begin

a:=&a;

b:=&b;

c:=&c;

if a>b and a>c then

dbms\_output.put\_line('a is large');

else

if b>a and b>c then

dbms\_output.put\_line('b is large');

else

dbms\_output.put\_line('c is large');

end if;

end if;

end;

# 7. Simple programs using loop, while and for iterative control statements.

SQL>edit loop.sql

declare

i number;

begin

i:=1;

loop

dbms\_output.put\_line(i);

i:=i+1;

exit when i>5;

end loop;

end;

SQL>edit while.sql

declare

i number;

begin

i:=1;

while(i<=10)

loop

dbms\_output.put\_line(i);

i:=i+1;

end loop;

end;

SQL>edit for.sql

declare

i number

begin

for i in 1.....15 loop

dbms\_output.put\_line(‘value of i :'||i);

end loop;

end;

# 8. Write a program to check whether the given number is Armstrong or not.

SQL>edit Armstrong.sql

declare

n number;

c number;

m number;

r number;

begin

n:=&n;

c:=0;

m:=n;

while(n!=0)

loop

r:=mod(n,10);

c:=c+(r\*r\*r);

n:=floor(n/10);

end loop;

if (m=c) then

DBMS\_output.put\_line('number is an armstrong');

else

DBMS\_output.put\_line(‘number is not an armstrong’);

End if;

End;

# 9. Write a program to generate all prime numbers below 100.

SQL>editprime.sql

declare

i number;

j number;

c number;

begin

i:=2;

while(i<=100)

loop

j:=1;

c:=0;

while(j<=i)

loop

if (mod(i,j)=0) then

c:=c+1;

end if;

j:=j+1;

end loop;

if(c=2) then

dbms\_output.put\_line(i);

end if;

i:=i+1;

end loop;

end;

week-8

# 10.Write a program to demonstrate the GOTO statement.

SQL>edit goto.sql

declare

i number;

begin

dbms\_output.put\_line('enter i value:');

i:=&i;

if(i>=0) then

goto here;

else

dbms\_output.put\_line(' i is negative');

end if;

goto abc;

<<here>>

dbms\_output.put\_line(' i is positive');

<<abc>>

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

end;

# 11.Write a program to demonstrate %type and %row type attributes.

First of all, create employee table with attributes empid ,empname and insert values in it.Later ,execute the following code.

SQL>edit emp.sql

declare

my\_empid emp.empid%type;

my\_empname emp.name%type;

my\_emprow emp%rowtype;

no number;

begin

no:=&no;

select empid,empname into my\_empid,my\_empname from emp where empid=no;

if (sql%rowcount=1) then

dbms\_output.put\_line('empno is:'||my\_empid||' , '||empname is :'||my\_empname);

else

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

end if;

select \* into my\_emprow from emp where empid=no;

if (sql%rowcount=1) then

dbms\_output.put\_line('empno is:'||my\_empid||' , '||empname is :'||my\_empname);

else

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

end if;

end;

Week-9

# 12.Write a program to demonstrate predefined exceptions.

SQL>edit predefined.sql

declare

a number;

b number;

c number;

begin

a:=&a;

b:=&b;

c:=a/b;

dbms\_output.put\_line('division is '|| c);

Exception

when Zero\_Divide then

dbms\_output.put\_line('b could not be zero');

end;

# 13.Write a program to demonstrate user defined exceptions.

SQL>edit userdefined.sql

declare

a number;

b number;

c number;

divide\_zero exception;

begin

a:=&a;

b:=&b;

if (b=0) then

raise divide\_zero;

else

c:=a/b;

dbms\_output.put\_line('division is '|| c);

end if;

Exception

when divide\_zero then

dbms\_output.put\_line('b could not be zero');

end;

# 14.Create a cursor, which displays all employee numbers and names from the EMP table.

First of all, create employee table with attributes empid ,empname and insert values in it.Later ,execute the following code.

SQL>edit cursor.sql

declare

cursor my\_cursor is select empid,empname from employee;

a employee.empid%type;

b employee.empname%type;

begin

open my\_cursor;

loop

fetch my\_cursor into a,b;

exit when My\_cursor%notfound;

dbms\_output.put\_line('empid is ' || a || ' empname is ' || b);

end loop;

close my\_cursor;

end;

week-10

# 15.Create a cursor, which update the salaries of all employees who works in dept no 10.

First of all, create employee table with attributes empid ,empname ,salary & deptno and insert values in it.Later ,execute the following code.

SQL>edit cur.sql

declare

cursor cur is select empid ,empname,salary,deptno from employee;

a employee.empid%type;

b employee.empname%type;

c employee.salary%type;

d employee.deptno%type;

begin

open cur;

loop

if (deptno=10) then

update employee set salary =slary+2000;

**end if;**

**end loop;**

**close cur;**

**dbms\_output.put\_line('empid is '|| a || ' empname is '|| b||'salary is '|| c||'deptno'|| d);**

**end;**

# **16.Create a cursor, which displays names of employees having salary > 50000.**

First of all, create employee table with attributes empid ,empname ,salary & deptno and insert values in it.Later ,execute the following code.

**SQL>edit cur1.sql**

declare

cursor cur is select empid ,empname,salary,deptno from employee;

a employee.empid%type;

b employee.empname%type;

c employee.salary%type;

d employee.deptno%type;

begin

open cur;

loop

fetch cur into a,b,c,d;

if (c>=50000) then

dbms\_output.put\_line('empid is '|| a || ' empname is '|| b||' salary is '|| c||'deptno'|| d);

end if;

exit when cur%notfound;

end loop;

close cur;

end;

week-11

# 17.Create a procedure to find reverse of a given number.

SQL>edit week11.sql

SUB PROGRAM:

create or replace procedure rev( n in number, rev out number ) as

n1 number;

r number;

begin

n1:=n;

rev:=0;

while(n1!=0)

loop

r:=mod(n1,10);

rev:=rev\*10+r;

n1:=floor(n1/10);

end loop;

end;

**MAIN PROGRAM:**

SQL>declare

a number;

b number;

begin

a:=&a;

rev(a,b);

dbms\_output.put\_line('reverse of number:'||b);

end;

/

# 18.Create a procedure to update the salaries of all employees whose salary is between 25000 to 50000.

First of all, create employee table with attributes empid ,empname ,salary & deptno and insert values in it.Later ,execute the following code.

SQL>edit sal.sql

SUB PROGRAM:

create or replace procedure sal as

cursor cur is select empid ,empname,salary,deptno from employee;

a employee.empid%type;

b employee.empname%type;

c employee.salary%type;

d employee.deptno%type;

e number:=0;

begin

open cur;

loop

fetch cur into a,b,c,d;

if (c>25000 and c<50000) then

update employee set salary =salary+2500;

e:=e+1;

fetch cur into a,b,c,d;

dbms\_output.put\_line('empid is '|| a || ' empname is '|| b||' salary is '|| c||'deptno'|| d);

end if;

exit when cur%notfound;

end loop;

close cur;

dbms\_output.put\_line('count'||e);

end;

**MAIN PROGRAM:**

SQL>declare

begin

sal;

end;

/

**Week-12**

# 19.Create a procedure to demonstrate IN, OUT and INOUT parameters.

SQL>edit week12.sql

SUB PROGRAM:

create or replace procedure findmin(x in number,y in number,z in out number) as

begin

if x<y then

z:=x;

else

z:=y;

end if;

end;

**MAIN PROGRAM:**

SQL>declare

a number;

b number;

c number;

begin

a:=&a;

b:=&b;

findmin(a,b,c);

dbms\_output.put\_line('minimum value:'||c);

end;

/

# 20.Create a function to check whether a given string is palindrome or not.

SQL>edit palindrome.sql

SUB PROGRAM:

create or replace function palin(s1 varchar2) return varchar2 as

s2 varchar2(20);

s3 varchar2(20);

begin

s2:=s1;

select reverse (s2) into s3 from dual;

return s3;

end;

MAIN PROGRAM:

SQL>declare

a varchar2(20);

b varchar2(20);

begin

a:=&a;

b:=palin(a);

if a=b then

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

else

dbms\_output.put\_line('not a palindrome');

end;

/

Week-13

# 21.Create a function to find the sum of salaries of all employees working in depart number 10.

First of all, create employee table with attributes empid ,empname ,salary & deptno and insert values in it.Later ,execute the following code.

SQL>edit salary.sql

SUB PROGRAM:

create or replace function sumsalary(n number) return number as

cursor cur4 is select salary,deptno from employee;

c employee.salary%type;

d employee.deptno%type;

s number:=0;

begin

open cur4;

loop

fetch cur4 into c,d;

if (d=n) then

s:=s+c;

end if;

exit when cur4%notfound;

end loop;

close cur4;

return s;

end;

MAIN PROGRAM:

SQL>declare

n number;

s number;

begin

n:=&n;

s:=sumsalary(n);

if a=b then

dbms\_output.put\_line('sum of salaries of employees belongs to deptno'||n ||'is:'||s);

end;

/

# 22.Create a trigger before/after update on the employee table for each row/statement.

First of all, create employee table with attributes empid ,empname ,salary & deptno and insert values in it.Later ,execute the following code.

SQL>edit trigger.sql

create or replace trigger t1

before update on employee

for each row

begin

dbms\_output.put\_line('row updated');

end;

SQL>update employee set salary =salary+10000 where salary <25000;

Week-14

# 23.Create a trigger before/after delete on the employee table for each row/statement.

First of all, create employee table with attributes empid ,empname ,salary & deptno and insert values in it.Later ,execute the following code.

SQL>edit trigger1.sql

create or replace trigger t5

before delete on emp

for each row

begin

dbms\_output.put\_line('row deleted');

end;

SQL>delete from emp where deptno=5;

# 24.Create a trigger before/after insert on the employee table for each row/statement.

First of all, create employee table with attributes empid ,empname ,salary & deptno and insert values in it.Later ,execute the following code.

SQL>edit trigger2.sql

create or replace trigger t8

before insert on emp

for each row

begin

dbms\_output.put\_line('row inserted');

end;

SQL>insert into emp values(122,'rani',65000,7);