**LAB CYCLE 2**

1. Write a PL/SQL code to accept the text and reverse the given text. Check the text is palindrome or not.

**PL/SQL CODE:-**

DECLARE

s VARCHAR2(10) := 'malayalam';

l VARCHAR2(20);

t VARCHAR2(10);

BEGIN

FOR i IN REVERSE 1..Length(s) LOOP

l := Substr(s, i, 1);

t := t||''||l;

END LOOP;

IF t = s THEN

dbms\_output.Put\_line(t ||''||' is palindrome');

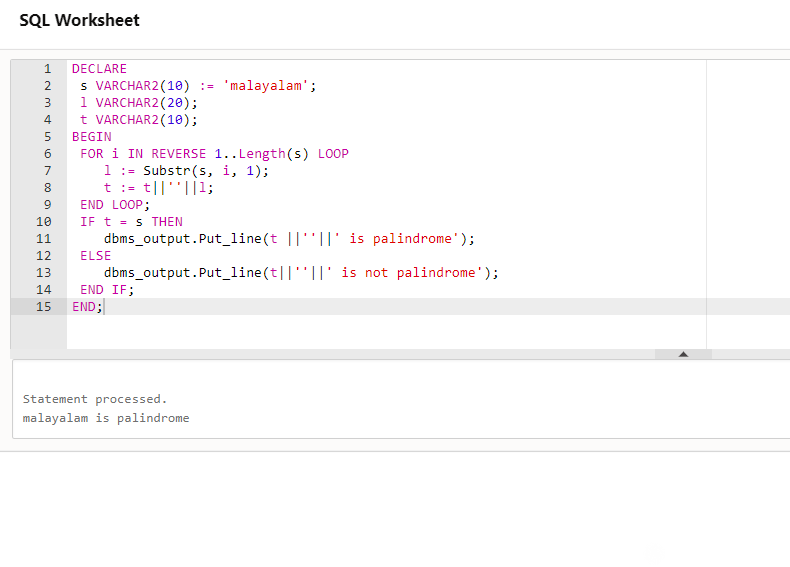
ELSE

dbms\_output.Put\_line(t||''||' is not palindrome');

END IF;

END;

**OUTPUT:-**



1. Write a program to read two numbers; If the first no > 2nd no, then swap the numbers; if the first number is an odd number, then find its cube; if first no < 2nd no then raise it to its power; if both the numbers are equal, then find its sqrt.

**PL/SQL CODE:-**

DECLARE

a INTEGER:=10;

b INTEGER:=7;

temp INTEGER:=0;

c INTEGER;

cube INTEGER;

BEGIN

IF a > b THEN

temp:=a;

a:=b;

b:=temp;

DBMS\_OUTPUT.PUT\_LINE('After swapping the a value is '||a ||' and b value is '||b);

IF MOD(b,2) !=0 THEN

cube:=a \* a \* a;

DBMS\_OUTPUT.PUT\_LINE('Cube is :'||cube);

ELSE

DBMS\_OUTPUT.PUT\_LINE('first number is even');

END IF;

ELSIF a < b THEN

c:=a \*\*b;

DBMS\_OUTPUT.PUT\_LINE('Power is :'||c);

ELSIF a=b THEN

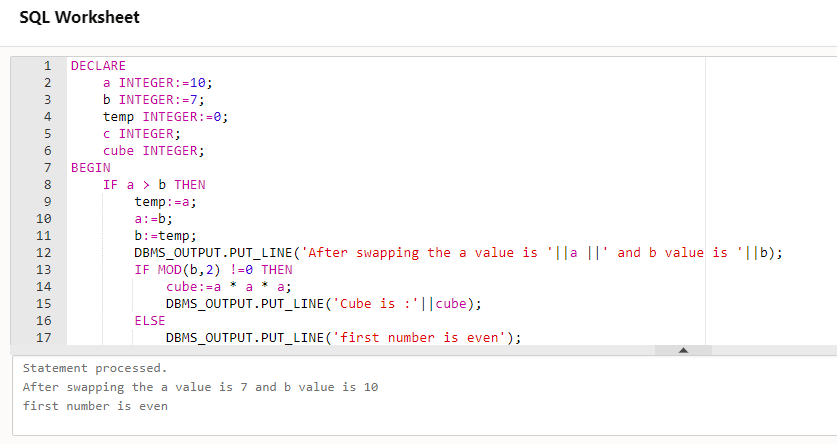
DBMS\_OUTPUT.PUT\_LINE('Square root of a is :'||(SQRT(a)));

DBMS\_OUTPUT.PUT\_LINE('Square root of b is :'||(SQRT(b)));

END IF;

END;

**OUTPUT:-**

****

1. Write a program to generate first 10 terms of the Fibonacci series

**PL/SQL CODE:-**

DECLARE

a NUMBER:=0;

b NUMBER:=1;

c NUMBER;

BEGIN

DBMS\_OUTPUT.PUT(a||' '||B||' ');

FOR I IN 3..10 LOOP

c:=a+b;

DBMS\_OUTPUT.PUT(c||' ');

a:=b;

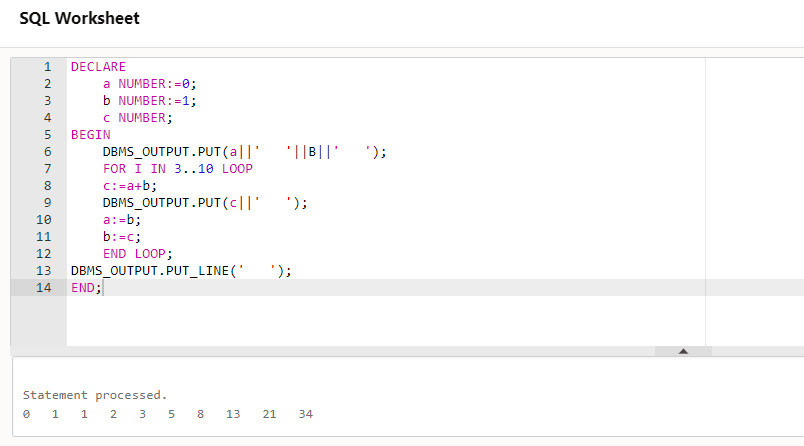
b:=c;

END LOOP;

DBMS\_OUTPUT.PUT\_LINE(' ');

END;

**OUTPUT:-**



1. Write a PL/SQL program to find the salary of an employee in the EMP table (Get the empno from the user). Find the employee drawing minimum salary. If the minimum salary is less than 7500, then give an increment of 15%. Also create an emp %rowtype record. Accept the empno from the user, and display all the information about the employee.

**PL/SQL CODE:-**







Declare

emno employee.emp\_no%type;

salary employee.emp\_salary%type;

emp\_rec employee%rowtype;

begin

emno:=104;

select emp\_salary into salary from employee where emp\_no=emno;

if salary<7500 then

update employee set emp\_salary=emp\_salary \* 15/100 where

emp\_no=emno;

else

dbms\_output.put\_line('No more increment');

end if;

select \* into emp\_rec from employee where emp\_no=emno;

dbms\_output.put\_line('Employee num: '||emp\_rec.emp\_no);

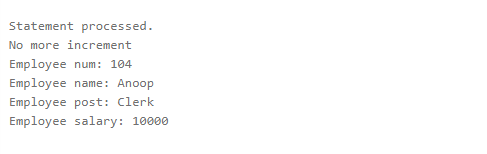
dbms\_output.put\_line('Employee name: '||emp\_rec.emp\_name);

dbms\_output.put\_line('Employee post: '||emp\_rec.emp\_post);

dbms\_output.put\_line('Employee salary: '||emp\_rec.emp\_salary);

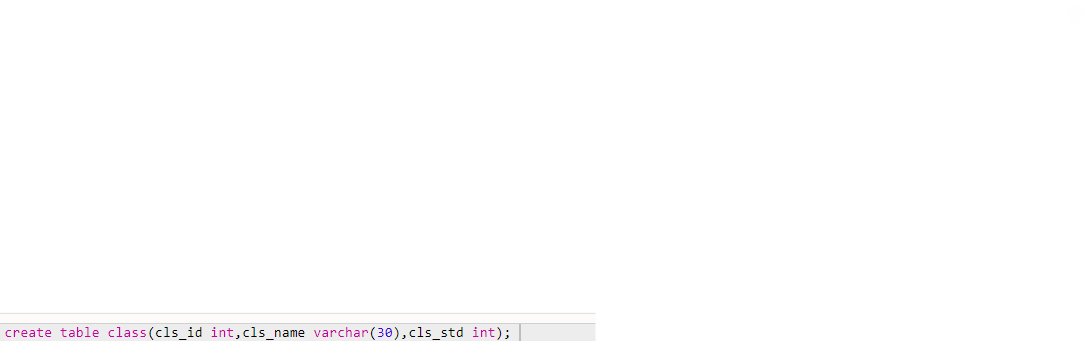
end;

**OUTPUT:-**

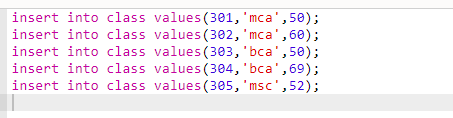


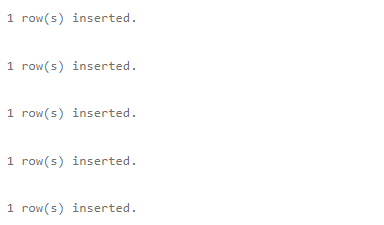
1. Write a PL/SQL function to find the total strength of students present in different classes of the MCA department using the table Class(ClassId, ClassName, Strength);

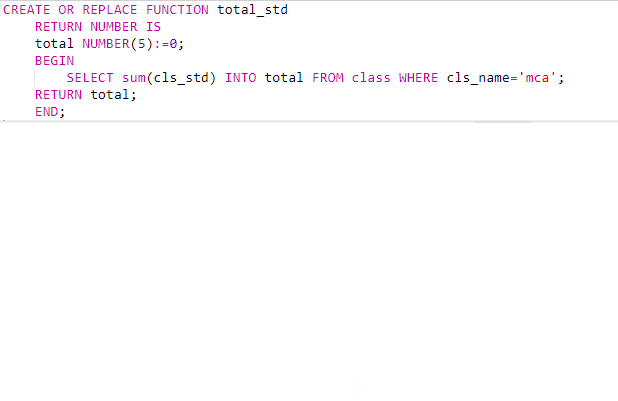
**PL/SQL CODE:-**



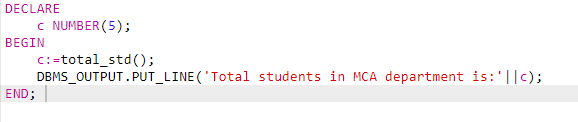


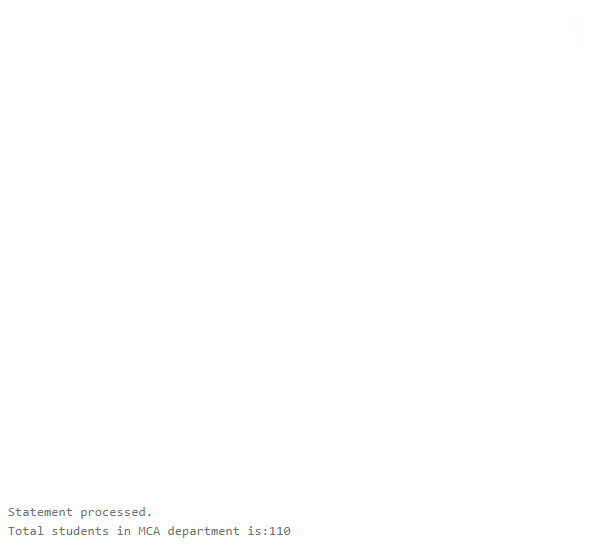








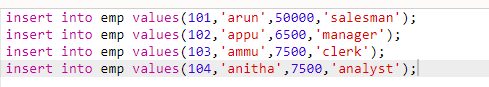


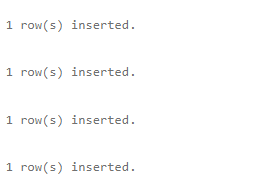


1. Write a PL/SQL procedure to increase the salary for the specified employee. Using empno in the employee table based on the following criteria: increase the salary by 5% for clerks, 7% for salesman, 10% for analyst and 20 % for manager. Activate using PL/SQL block.









**Procedure:**

CREATE OR REPLACE PROCEDURE increSalary

IS

emp1 emp%rowtype;

sal emp.salary%type;

dpt emp.emp\_dpt%type;

BEGIN

SELECT salary,emp\_dpt INTO sal,dpt FROM emp WHERE emp\_no = 104;

IF dpt ='clerk' THEN

UPDATE emp SET salary = salary+salary\* 5/100 ;

ELSIF dpt = 'salesman' THEN

UPDATE emp SET salary = salary+salary\* 7/100 ;

ELSIF dpt = 'analyst' THEN

UPDATE emp SET salary = salary+salary\* 10/100 ;

ELSIF dpt = 'manager' THEN

UPDATE emp SET salary = salary+salary\* 20/100 ;

ELSE

DBMS\_OUTPUT.PUT\_LINE ('NO INCREMENT');

END IF;

SELECT \* into emp1 FROM emp WHERE emp\_no = 104;

DBMS\_OUTPUT.PUT\_LINE ('Name: '||emp1.emp\_name);

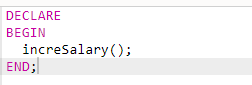
DBMS\_OUTPUT.PUT\_LINE ('employee number: '||emp1.emp\_no);

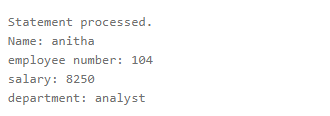
DBMS\_OUTPUT.PUT\_LINE ('salary: '|| emp1.salary);

DBMS\_OUTPUT.PUT\_LINE ('department: '|| emp1.emp\_dpt);

END;







1. Create a **cursor** to modify the salary of ‘president’ belonging to all departments by 50%

**PL/SQL Code:-**





insert into Employee values(100,'Joseph','manager',56000,'sales')

1 row(s) inserted.

insert into Employee values(101,'Ravi','clerk',23000,'sales')

1 row(s) inserted.

insert into Employee values(102,'Paul','execute',48000,'HR')

1 row(s) inserted.

insert into Employee values(103,'Rani','president',50000,'HR')

1 row(s) inserted.

insert into Employee values(104,'Antony','president',48000,'marketing')

1 row(s) inserted.

insert into Employee values(105,'Rose','accountant',45000,'marketing')

1 row(s) inserted.

insert into Employee values(106,'Lovely','president',49000,'purchase')

1 row(s) inserted.

insert into Employee values(107,'Babu','supervisor',32000,'purchase')

1 row(s) inserted.

DECLARE

total\_rows number(2);

emp1 Employee%rowtype;

BEGIN

UPDATE Employee SET emp\_salary=emp\_salary+emp\_salary \* 50/100 where emp\_post='president';

IF sql%notfound THEN

dbms\_output.put\_line(' no employee updated');

ELSIF sql%found THEN

total\_rows := sql%rowcount;

dbms\_output.put\_line( total\_rows ||' employee updated');

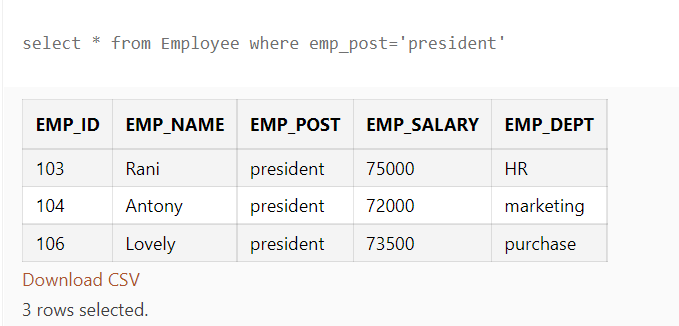
end if;

SELECT \* into emp1 FROM Employee WHERE (emp\_id=104 and emp\_post='president');

END;

Statement processed.

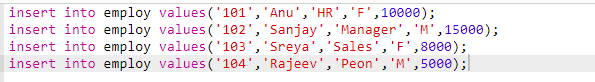
3 employee updated

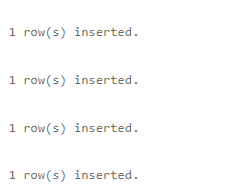


1. Write a cursor to display list of Male and Female employees whose name starts with S.









**Code:-**

DECLARE

CURSOR emp1 IS

SELECT emp\_id,emp\_name,emp\_post,emp\_salary FROM employ where emp\_name like ('S%') ;

emp2 emp1%ROWTYPE;

BEGIN

OPEN emp1;

LOOP

FETCH emp1 INTO emp2;

EXIT WHEN emp1%NOTFOUND;

dbms\_output.Put\_line('Employee\_ID: ' ||emp2.emp\_id);

dbms\_output.Put\_line('Employee\_Name: ' ||emp2.emp\_name);

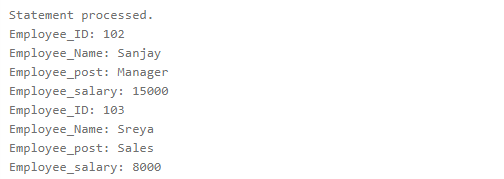
dbms\_output.Put\_line('Employee\_post: ' ||emp2.emp\_post);

dbms\_output.Put\_line('Employee\_salary: '||emp2.emp\_salary);

END LOOP;

CLOSE emp1;

END;

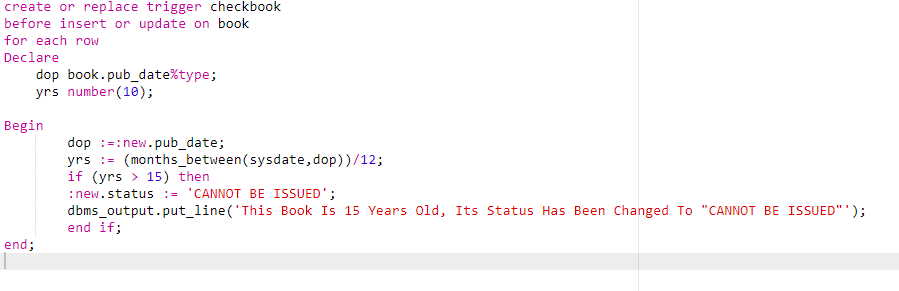


1. Create the following tables for Library Information System: Book : (accession-no, title, publisher, publishedDate, author, status). Status could be issued, present in the library, sent for binding, and cannot be issued. Write a **trigger** which sets the status of a book to "cannot be issued", if it is published 15 years back.

**PL/SQL CODE:-**

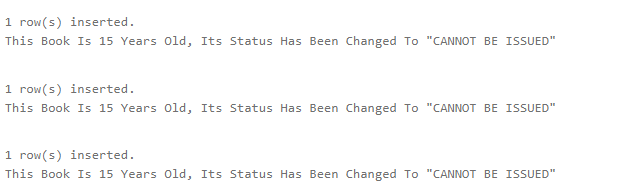
















1. Create a table Inventory with fields pdtid, pdtname, qty and reorder\_level. Create a **trigger** control on the table for checking whether qty<reorder\_level while inserting values.

**PL/SQL Code:-**



****

