**PL-SQL (Lab-2)**

**(Procedures and Functions in PL/SQL)**

**1. Create table EMP (eno, ename, bp, da, hra, total). Initially, the ‘total’ column values for**

**all the employees are null. Write a stored procedure that accepts the eno of an employee**

**as an input and computes his total by adding bp + da + hra. Your pl/sql code should send**

**the total to the main block for printing and update it on the emp table.**

**A black background with white text

Description automatically generated**

create or replace procedure totalsal(empno in number, t out number) is

begin

select bp+da+hra into t from emp where eno=empno;

update emp set total=t where empno=eno;

end;

SQL> /

declare

t number;

begin

totalsal(101,t);

dbms\_output.put\_line(t);

end;

/

**A screen shot of a computer

Description automatically generated**

**2. Write a local function that will accept a number as an input, compute it factorial, and**

**send it to the main block for printing.**

declare

n number:=5;

factans number;

function fact(n1 in number) return number as ans number:=1;

begin

for i in 1..n1 loop

ans:=i\*ans;

end loop;

return ans;

end fact;

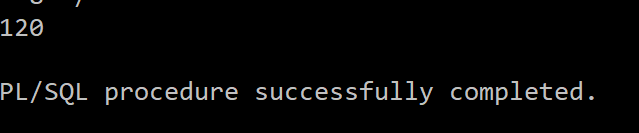
begin

factans:=fact(n);

dbms\_output.put\_line(factans);

end;

/

****

**3. Write a stored procedure that will accept a number, check whether positive, negative or**

**zero and accordingly print the message.**

create or replace procedure fun(n in number) is

begin

if (n>0) then

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

elsif (n<0) then

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

else

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

end if;

end;

SQL> /

A black screen with white text

Description automatically generated

**4. Write a stored function that will find the sum of elements from 1 to n and send the result**

**to the main block. ‘n’ should send as the input to the called function.**

create or replace function sumN(n in number) return number is s number:=0;

begin

for i in 1..n loop

s:=s+i;

end loop;

return s;

end;

/

declare

ans number;

begin

ans:=sumN(5);

dbms\_output.put\_line(ans);

end;

/

**A black background with white text

Description automatically generated**

**5. Write a local procedure that will accept two ranges n1 and n2, find the sum of the average of elements between these ranges and return the result to the main block.**

declare

s number:=0;

ans2 number:=0;

procedure f1(n1 in number, n2 in number, ans out number) is

begin

for i in n1..n2 loop

s:=s+i;

end loop;

ans:=s/(n2-n1+1);

end;

begin

f1(2,5,ans2);

dbms\_output.put\_line(ans2);

end;

/

**A black background with white text

Description automatically generated**

**6. Write a recursive function to display ‘hello world’ 10 times on the screen. The function**

**should accept the string as input from the main block.**

create or replace procedure hw(s in varchar2, n in number) is

begin

if n>0 then

dbms\_output.put\_line(s);

hw(s,n-1);

end if;

end;

/

declare

s varchar2(20):='hello world';

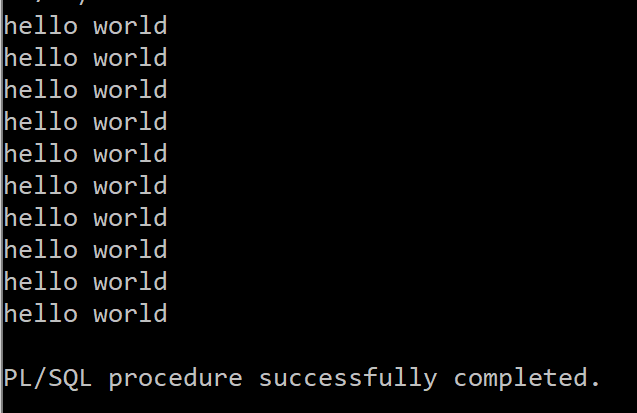
n number:=10;

begin

hw(s,n);

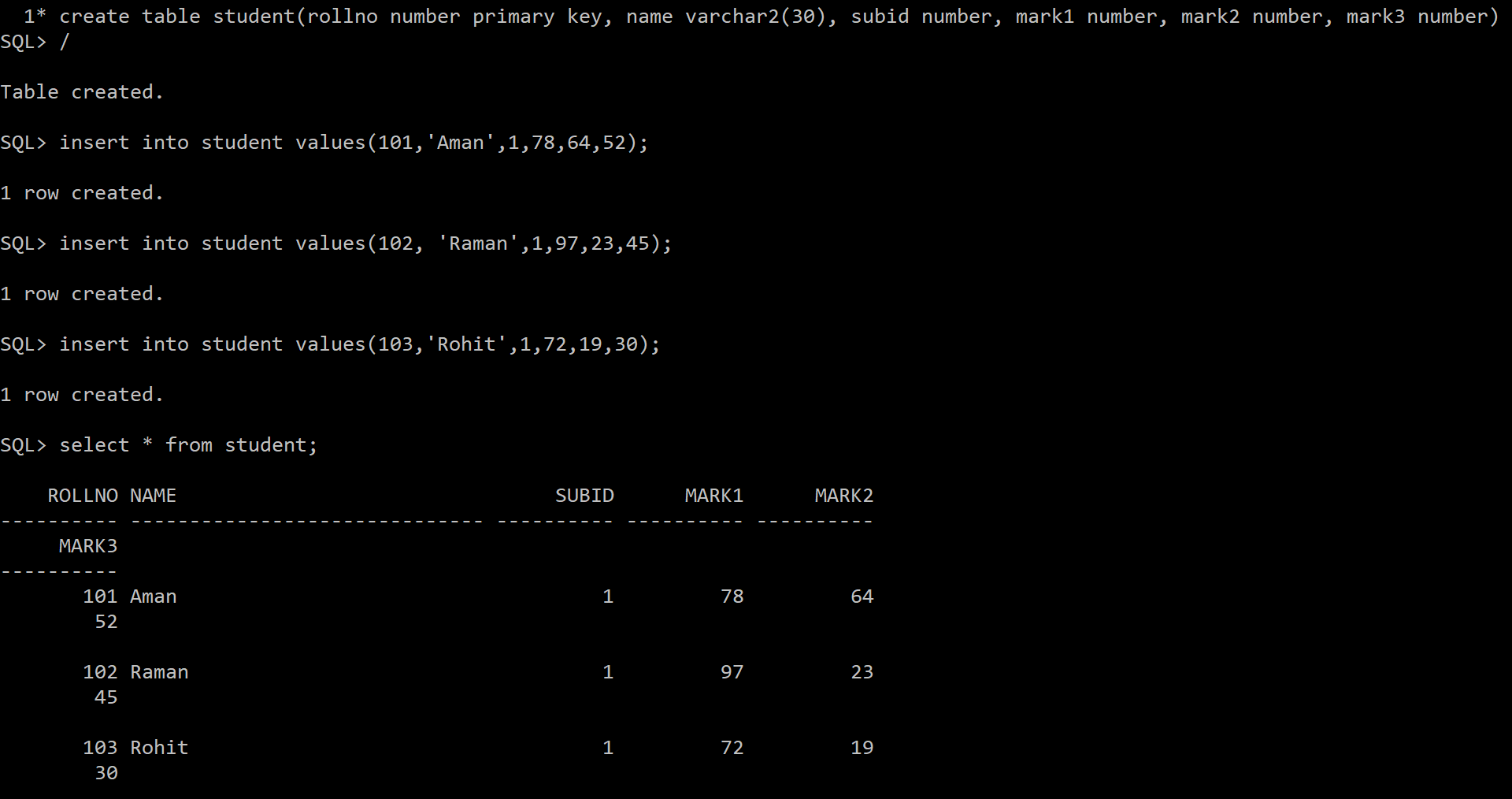
end;

/

****

**Consider a table student (rollno number, name varchar (30), sub-id number, mark1**

**number, mark2 number, mark3 number). Q7-Q10 are based on the student table.**

****

**7. Write a stored procedure ‘p1’, which will accept the rollno of a student as the input, and**

**then find the sum of the three subject marks and keep it in a variable ‘total’. Send this**

**‘total’ to the main block of the program and display it.**

create or replace procedure p1(rno in number, total out number)

is

begin

select mark1+mark2+mark3 into total from student where student.rollno=rno;

end;

/

declare

rno number;

total number;

begin

rno:=&rno;

p1(rno,total);

dbms\_output.put\_line(total);

end;

/

**A black background with white text

Description automatically generated**

**8. Write a local function ‘fun2’, which will call the procedure ‘p1’ of Q7 by passing the**

**rollno of a student. It then finds the average of the total marks and stores it in a variable**

**‘avg’. Send this ‘avg’ to the main block of the program and display it.**

declare

avg2 number;

total number:=0;

function fun2(rno in number) return number is avg1 number;

begin

p1(rno,total);

avg1:=total/3;

return avg1;

end fun2;

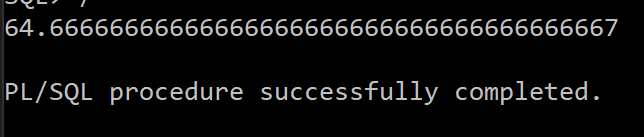
begin

avg2:=fun2(101);

dbms\_output.put\_line(to\_char(avg2));

end;

/

****

**9. Write a stored function ‘fun3’ which will accept the rollno of a student as the input and**

**return the highest marks among the three subjects to the main block.**

create or replace function fun3(rno in number) return number is m number;

begin

select greatest(mark1, mark2, mark3) into m from student where rollno=rno;

return m;

end;

/

declare

m number;

begin

m:=fun3(103);

dbms\_output.put\_line(m);

end;

/

**A black background with white text

Description automatically generated**

**10. Write a local procedure p2, which will call the procedure ‘p1’ of Q7 by passing the rollno**

**of a student. It then received the total marks from the procedure ‘p1’ and sent the**

**student’s three subject marks and the total marks to the main block for printing.**

declare

m1 number;

m2 number;

m3 number;

total number;

rno number;

procedure p2(rno in number, m1 out number, m2 out number, m3 out number, total out number) is

begin

p1(rno,total);

select mark1, mark2, mark3 into m1,m2,m3 from student where rollno=rno;

end;

begin

p2(102,m1,m2,m3,total);

dbms\_output.put\_line('mark1: '||m1);

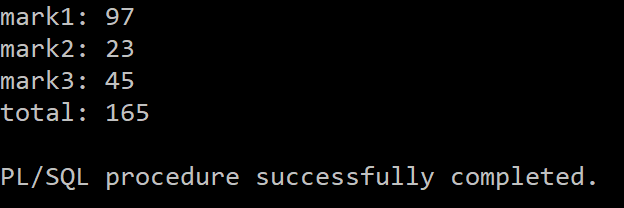
dbms\_output.put\_line('mark2: '||m2);

dbms\_output.put\_line('mark3: '||m3);

dbms\_output.put\_line('total: '||total);

end;

/

****