**Practical No -6**

**Problem stmt :** Write a Stored Procedure namely proc\_Grade for the categorization of student. If marks scored by students in examination is <=1500 and marks>=990 then student will be placed in distinction category if marks scored are between 989 and900 category is first class, if marks 899 and 825 category is Higher Second Class Write a PL/SQL block for using procedure created with above requirement.

set SERVEROUTPUT ON

create or replace procedure grade\_calc(stud\_id IN number)

is

begin

declare

marks number(10);

--stud\_id number (10):&amp;rollno;

begin

select totalmarks into marks from stud\_marks where rollno=stud\_id;

if marks<=1500 And marks>=990 then

insert into result values(stud\_id,'Distinction');

dbms\_output.put\_line('student is passed with first class distinction');

elsif marks<=989 And marks>=900 then

insert into result values(stud\_id,'First class');

dbms\_output.put\_line('student is passed with first class ');

elsif marks<=899 And marks>=825 then

insert into result values(stud\_id,'Second class');

dbms\_output.put\_line('student is passed with second class');

elsif marks<=824 And marks>=400 then

insert into result values(stud\_id,'pass');

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

else

dbms\_output.put\_line('student is failed ');

insert into result values(stud\_id,'fail');

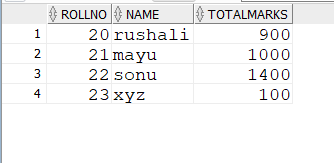
end if;

end ;

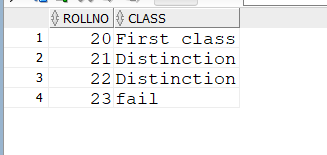
end;

exec grade\_calc(23);

*stud\_marks:*



*Result:*



**Problem stmt :**

Consider following schema for Bank database. Account(Account\_No, Cust\_Name, Balance, NoOfYears) Earned\_Interest(Account\_No,Interest\_Amt) Write a PL/SQL procedure for following requirement. Take as input Account\_No and Interest Rate from User. Calculate the Interest\_Amt as simple interest for the given Account\_No and store it in Earned\_Interest table. Display all the details of Earned\_Interest Table.

create or replace procedure calc\_interest(acc1 number,rate number)

is

begin

declare

bal number(10);

t\_year number(20);

interest\_amt number(20);

begin

select balance into bal from acc where accno=acc1;

select no\_of\_year into t\_year from acc where accno=acc1;

interest\_amt:=( bal\*rate\*t\_year)/100;

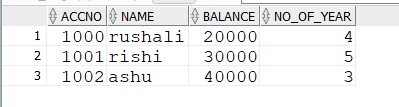
insert into interest\_amount values(acc1,interest\_amt);

end;

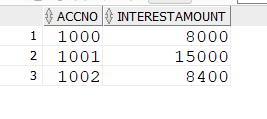
end;

exec calc\_interest(1002,7);

*Acc*



*interest\_amount*



**Problem stmt :**

Write a stored function in PL/SQL for given requirement and use the same in PL/SQL block.

Account no. and branch name will be accepted from user. The same will be searched in table acct\_details. If status of account is active then display appropriate message and also store the account details in active\_acc\_details table, otherwise display message on screen “account is inactive”.

create or replace function active\_acc\_details(acc number)

return varchar

is

begin

declare

st varchar(30);

begin

select status into st from account\_details where accno=acc;

return st;

end;

end;

set SERVEROUTPUT ON;

declare

br varchar(20);

bal number(20);

st\_x varchar(20);

acc1 number(20):=&accountno;

begin

select branch into br from account\_details where accno=acc1;

select balance into bal from account\_details where accno=acc1;

st\_x:=active\_acc\_details(acc1);

if st\_x='active' then

insert into active\_account values(acc1,br,bal);

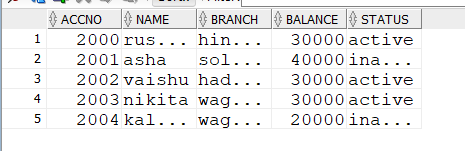
else

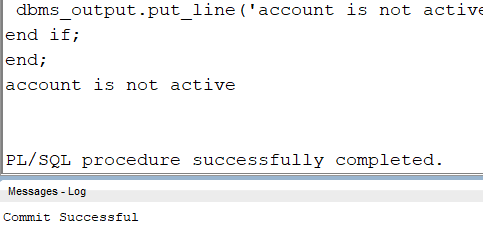
dbms\_output.put\_line('account is not active');

end if;

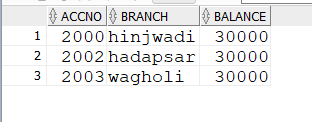
end;

*account\_details*





*active\_account*



**Problem stmt :**

Consider following schema for Sales table Sales (Item\_No int, Quantity int, Unit\_Price number, Status varchar(20)) Status can be “Shipped” or “Delivered” or “Returned” Write a PLSQL function to calculate the total amount of Sales for Items with status “Shipped”. (total\_amount = SUM(Quantity \* Unit\_Price) Display the Total Amount of sales.

CREATE OR REPLACE FUNCTION CalculateShippedSalesTotal RETURN NUMBER IS

total\_amount NUMBER := 0;

BEGIN

SELECT SUM(Quantity \* Unit\_Price)

INTO total\_amount

FROM Sales

WHERE Status = 'Shipped';

RETURN total\_amount;

END;

DECLARE

total NUMBER;

BEGIN

total := CalculateShippedSalesTotal;

DBMS\_OUTPUT.PUT\_LINE('Total Shipped Sales Amount: ' || total);

END;

*Sales*

