Assignment No. 7
Roll No :- 31384

Assignment No. 7

Title :- PL/SQL Stored procedure & stored function

Problem statement :- write a stored Procedure namely proc-Grade for the categorization of student

objective :- · Understand PL/SQL stored procedure.
- Understand PL/SQL Stord Functia.

sw package :- mysql
PL/SQL

Write a stored procedure namely proc-Grade for the categorization of customer. If purchase by customer is between 9999 & 5000 Category is gold. if purchase between 4999 & 2000 category is silver.

Learning objective :-

- To understand & implement the stored function & stored procedure in PL/SQL.

Theory :-

PL/SQL :-

It stands for procedural language / structured query. PL-SQL offers set of procedural (IF statement, loops, assignment) organized within blocks.

Blocks in PL/SQL :-

A PL/SQL Block defined by the keywords DECLARE, BEGIN, EXCEPTION & END which breaks into three section.

1) Declarative: Statement that declare variable, constant & other code element, which can be used within that block.

2) Executable :- Statements that are run when the block is executed.

3) Exception handling :- A specially structured section you can use to catch or trap.

what is Stored procedure ?

A stored procedure or a simple proc is named PL/SQL block which performs one or more specific task.

This is similar to a procedure in other programming languages.

A procedure has header & a body - The header consists of name of the procedure & parameters or variable passed to the the procedure.

Passing parameters :

i) IN - parameters
ii) OUT - parameter
iii) IN - OUT parameter

Syntax :-

```
Create [or replace] procedure - proc - name [list_parameter]
  is
  Declaration section
  BEGIN
      execution section
  EXCEPTION
      Exception section
END
```

How to execute stored procedure ?

1) Execute proc-name;

2) within another procedure
proc-name;

B) stored function :-

A function is named PL/SQL Block which is similar a procedure. The major difference betn procedure & a function.

a function must always return a value. but a procedure may or may not a return a value.

Create [or replace] function func_name [parameter

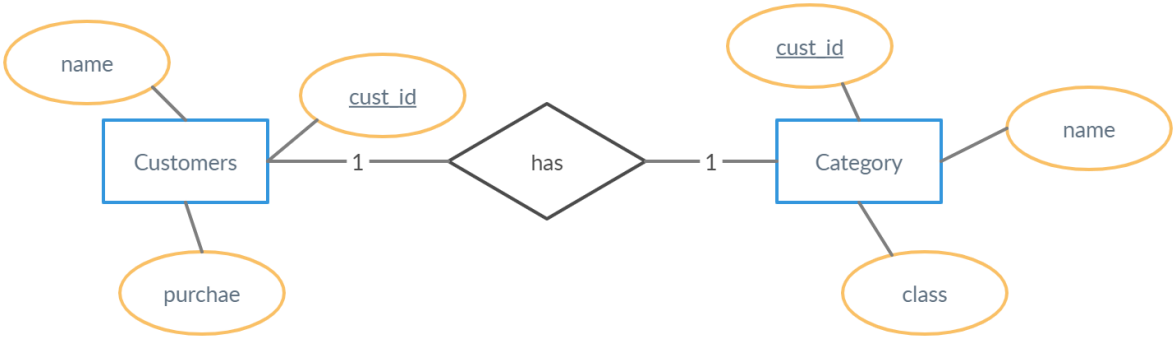
return return_datatype;
is
declaration section
is
declaration section
Begin
Execution section
return return_variable
END;

Return Type: The header section defines the return type of function. The return datatype can be any of oracle datatype like varchar, number etc.

Conclusion :-

Hence, we implemented stored procedure & function.

ER-Diagram :-



Code:-

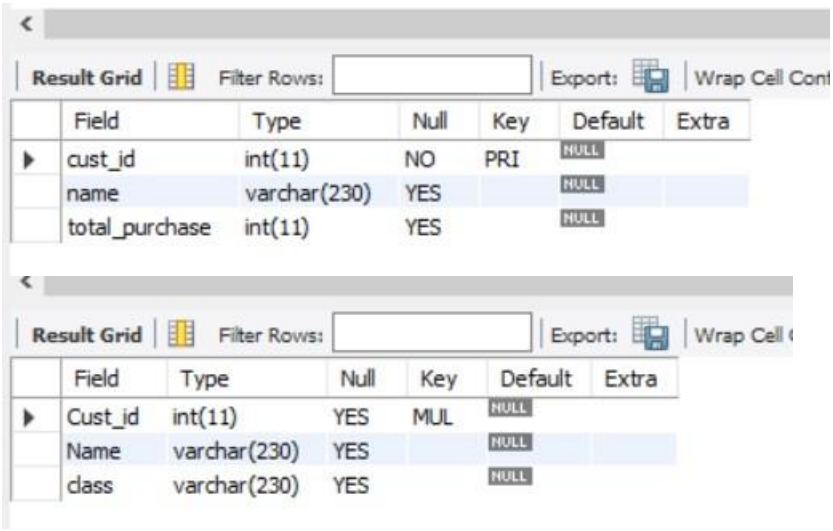
```
use plsql;

create table Customer(cust_id int primary key,name varchar(230),total_purchase int);

describe Customer;

create table Category(Cust_id int,Name varchar(230),class varchar(230), foreign key(Cust_id)
references Customers(Cust_id));

describe Category;


insert into Customer values(1,"Swapnil",11000);

insert into Customer values(2,"Ganesh",6000);

insert into Customer values(3,"Sourabh",3000);




call proc_Grade(11000,1,"Swapnil");

call proc_Grade(6000,2,"Ganesh");

call proc_Grade(3000,3,"Sourabh");


call proc_Grade(4000,4,"rushi");

select * from Customer;

select * from Category;

truncate table Category;
```

```
DELIMITER //
   create procedure proc_Grade(purchase int,cust_id int,name varchar(230))
    begin



        #Exception Handling
        declare continue handler for not found
      begin
```

```
            select 'NOT FOUND';
        end;


            #condition 1
            if( purchase<=20000 && purchase >=10000 )
             then
               insert into Category values(cust_id, name, "Platinum");
            end if;

            #condition2
            if( purchase>=5000 && purchase<=9999)
         then
           insert into Category values(cust_id,name, "gold");
        end if;

     if( purchase>=2000 && purchase<=4999)
         then
           insert into Category values(cust_id,name, "silver");
        end if;


     end;
//

DELIMITER ;

drop procedure proc_Grade;
```

Output Screen shots:-

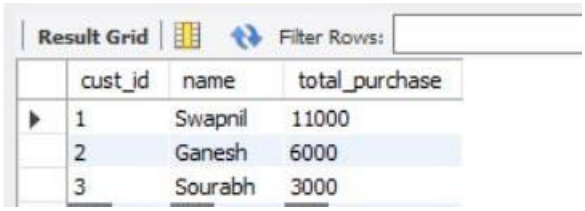
Customer and Category Tables:-



Field	Type	Null	Key	Default	Extra
cust_id	int(11)	NO	PRI	NULL	
name	varchar(230)	YES		NULL	
total_purchase	int(11)	YES		NULL	

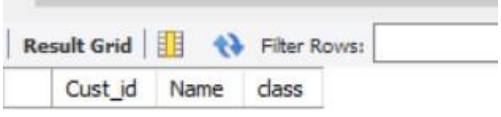
Field	Type	Null	Key	Default	Extra
Cust_id	int(11)	YES	MUL	NULL	
Name	varchar(230)	YES		NULL	
class	varchar(230)	YES		NULL	

Customer Table after inserting 3 values:-



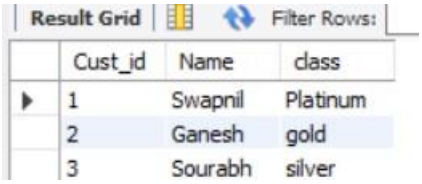
cust_id	name	total_purchase
1	Swapnil	11000
2	Ganesh	6000
3	Sourabh	3000

Category table before calling procedure :-



Cust_id	Name	class

Category Table after calling Procedure :-



Cust_id	Name	class
1	Swapnil	Platinum
2	Ganesh	gold
3	Sourabh	silver