Need of stored procedure or function:

=====================================

Reusability of code(query)

table: emplyoee

id name sal dept doj ....

=> select \* from employee where id=4;

=> select \* from employee where id=8;

=> select \* from employee where id=6;

query or set of queries that are required repeatedly are

stored in a procedure or function

when it ia required, a call is given to stored procedure or

function and it executes the query.

conclusion: write a query once, use many times

delimiter:

===========

which states the completion of an instruction or query in

a code

DBMS --> delimiter

delimiter // or $$

delimiter ;

syntax of stored procedure

==========================

delimiter //

create procedure procedure\_name(parameter)

BEGIN

body of procedure;

END //

delimiter ;

call procedure

=================

syntax:

call procedure\_name();

drop procedure

================

drop procedure procedure\_name;

parameters in store procedure

=============================

syntax:

mode variable\_name datatype(size)

mode: IN OUT INOUT

Parameter only accepting value from the call of stored

procedure has to be declared with the mode IN.

if parameter is returning value from stored procedure to call

then that parameter has to be declared as OUT.

accept + return => INOUT

declaring a variable inside stored procedure

==============================================

=> DECLARE variable datatype(size) => DECLARE z int

to set a value to a variable

============================

SET variable\_name=value;

x=10, y=20

y=x+y

=>30

IF..ELSE in stored procedure

LOOPs in stored procedure

create database db;

use db;

create table employee(id int primary key auto\_increment,name varchar(50), dept varchar(50),sal float,doj date);

insert into employee(name,dept,sal,doj)

values('harry','HR',45000,'2022-09-09'),

('mac','IT',60000,'2023-09-08');

insert into employee(name,dept,sal,doj)

values('hari','production',40000,'2021-09-09'),

('shree','IT',65000,'2023-09-01');

select \* from employee where id=3;

delimiter //

create procedure abc()

begin

select \* from employee;

end//

show procedure status where db="db";

call abc();

drop procedure abc;

delimiter //

create procedure abc1(IN x int,IN y int)

begin

declare z int;

set z=x+y;

select z as result;

end //

delimiter ;

call abc1(10,20);

SET @y=20;

select @y;

call abc(10,@y);

Decision control instruction

============================

1)IF

2)IF..ELSE

3)ELSEIF

1)IF statement

syntax:

IF condition THEN

if body;

END IF;

2)IF..ELSE statment

syntax:

IF condition THEN

if body;

ELSE

else body;

END IF;

3)ELSEIF syntax:

----------------

IF condition THEN

if body;

ELSEIF condition THEN

else if body;

ELSEIF condition THEN

else if body;

ELSE

else body;

END IF;

delimiter //

create procedure emp\_grade(IN x int)

begin

declare temp float;

select sal into temp from employee where id=x;

if temp>=50000 then

select 'platinum employee grade' as grade;

else

select 'gold employee grade' as grade;

end if;

end //

call emp\_grade(2);

drop procedure emp\_grade;

delimiter //

create procedure emp\_grade(IN x int)

BEGIN

DECLARE temp float;

select sal INTO temp from employee where id=x;

IF temp>=60000 THEN

select 'Platinum employee' as grade;

ELSEIF temp>=45000 and temp<60000 THEN

select 'GOLD employee' as grade;

ELSE

select 'silver employee' as grade;

END IF;

END //

call emp\_grade(4);

Loop Control Instructions in stored procedure

==============================================

While loop

============

syntax:

initialization;

WHILE condition DO

while body;

increment / decrement;

END WHILE;

initialization

=================

DECLARE i int;

SET i=1;

increment:

==========

SET i=i+1;

REPEAT

========

syntax:

REPEAT

statements;

UNTIL condition

END REPEAT;

delimiter //

create procedure loop1(IN x int)

begin

declare i int;

set i=1;

while i<=x DO

select i;

set i=i+1;

end while;

end //

call loop1(5);

select \* from employee;

delimiter //

create procedure loop2()

begin

declare i int;

declare temp float;

set i=1;

while i<=4 do

select sal into temp from employee where id=i;

if temp>=60000 then

select 'platinum employee as grade';

else

select 'gold employee as grade ';

end if;

set i=i+1;

end while;

end //

call loop2();

stored functions

=================

used to achieve reusability of code as in stored

prcedure

Difference between stored procedure and stored function

=======================================================

1)stored procedure can not return value but

stored function returns value.

2)you can not use a stored procedure in SQL query, but you

can use stored function in sql query.

3)In stored procedure, there are three types of parameters

(modes) IN, OUT,INOUT, but in stored function,

all parameters have IN mode,

so here you don't need to specify IN mode for a parameter in

function

delimiter //

create function myfun(x float)

returns varchar(20)

reads sql data

deterministic

begin

declare res varchar(20);

if x>=60000 then

set res='paltinum';

else

set res='gold';

end if;

return (res);

end //

select name,dept,sal,myfun(sal) from employee;