Procedures in Mysql:

DELIMITER &&

**CREATE** **PROCEDURE** procedure\_name [[IN | **OUT** | INOUT] parameter\_name datatype [, parameter datatype]) ]

**BEGIN**

    Declaration\_section

    Executable\_section

**END** &&

DELIMITER;

With a Simple SELECT QUERY

Let’s now see how we can CREATE it for selecting data from the table.

DELIMITER $$

CREATE PROCEDURE GetusersData1()

BEGIN

SELECT \* FROM users;

END$$

DELIMITER ;

#call the procedure

call GetusersData1();

#Creating a Procedure with Input Parameters

DELIMITER $$

CREATE PROCEDURE GetDetailsById(IN usersid INT)

BEGIN

SELECT \* FROM sample1 where id =usersid;

END $$

DELIMITER ;

call GetDetailsById(1);

#Creating Procedure with Output Parameters

DELIMITER &&

CREATE PROCEDURE spGetAverageSalary(OUT avgSalary DECIMAL(10,2))

BEGIN

SELECT AVG(salary) INTO avgSalary FROM employee;

END &&

DELIMITER

select \* from employee;

call spGetAveragesalary(@avgSalary);

select @avgSalary;

#Creating Procedure with INOUT Parameters

DELIMITER &&

CREATE PROCEDURE sp\_findTotSal1(INOUT salary INT, IN commission INT)

BEGIN

SET salary = salary + commission;

END &&

DELIMITER

After execution of a procedure run like the below format:

set @salary = 20000;

call sp\_findTotSal1(@salary,3000);

select @salary;

set @salary=15000;

call sp\_findTotSal(@salary,2000);

#update table data using stored procedure

DELIMITER &&

Create Procedure Update\_employee ( IN id INT, IN state varchar(30))

BEGIN

UPDATE employee

SET

state = state WHERE empid = id;

END &&

DELIMITER ;

call Update\_employee(1,'kerala');

#Delete procedure

drop procedure Update\_employee;

#if statement in procedure

DELIMITER $$

CREATE PROCEDURE GetCustomerLevel(

IN orderNumber INT,

OUT CustomerLevel VARCHAR(20))

BEGIN

DECLARE Totalprice int DEFAULT 0;

SELECT qty\*price

INTO Totalprice

FROM orderdetails

WHERE oId = orderNumber;

IF Totalprice > 1000 THEN

SET CustomerLevel = 'Regular';

END IF;

END $$

DELIMITER ;

Call GetCustomerLevel()

#case statement in procedure

#Case Stmt

CREATE PROCEDURE GetCustomerShipping(

IN pCustomerNUmber INT,

OUT pShipping VARCHAR(50)

)

BEGIN

DECLARE customerCountry VARCHAR(100);

SELECT

country

INTO customerCountry FROM

customers

WHERE

customerNumber = pCustomerNUmber;

CASE customerCountry

WHEN 'USA' THEN

SET pShipping = '2-day Shipping';

WHEN 'Canada' THEN

SET pShipping = '3-day Shipping';

ELSE

SET pShipping = '5-day Shipping';

END CASE;

END$$

DELIMITER ;

Call GetCustomershipping(1,@pShipping);

Select @pShipping.

#Example2 for case stmt

# To find a grade based on the emp salary

DELIMITER $$

Create procedure sp\_salGrade4(in id varchar(20),out grade varchar(10))

begin

declare empsal int default 0;

select salary into empsal

from pms\_employee\_details

where Eno=id;

case empsal

when empsal>50000 then

set grade = 'A';

when empsal>30000 and empsal<=50000 then

set grade='B';

when empsal>20000 and empsal<=30000 then

set grade='C';

else

set grade='D';

end case;

end $$

DELIMITER ;

call sp\_salGrade4('E24',@grade);

select @grade;

#Loop Example

DELIMITER $$

CREATE PROCEDURE LoopDemo()

BEGIN

DECLARE x INT;

DECLARE str VARCHAR(255);

SET x = 1;

SET str = '';

loop\_label: LOOP

IF x > 10 THEN

LEAVE loop\_label;

END IF;

SET x = x + 1;

IF (x mod 2) THEN

ITERATE loop\_label;

ELSE

SET str = CONCAT(str,x,',');

END IF;

END LOOP;

SELECT str;

END$$

DELIMITER ;

call LoopDemo();

#while Loop Example: Sum of digit

CREATE PROCEDURE sumofdigit()

begin

declare num int;

declare sum int;

declare rem int ;

set @num=2345;

set @sum=0;

while(@num>0)

do

set @rem=@num%10; #2345%10=5,234%10=4,23%10=3,2;

set @sum=@sum+@rem; #0+5=5=4=9+3=12+2=14;

set @num=@num/10; #2345/10=234,234/10=23,23/10=2,2/10=0;

end while;

#set output = 'The sum of digit '+cast(@sum as decimal);

#set output1 = concat('The sum of digit ',@sum);

select @sum;

#select output1;

End

#repeat Loop

DELIMITER $$

CREATE PROCEDURE RepeatDemo()

BEGIN

DECLARE counter INT DEFAULT 1;

DECLARE result VARCHAR(100) DEFAULT '';

REPEAT

SET result = CONCAT(result,counter,',');

SET counter = counter + 1;

UNTIL counter >= 10

END REPEAT;

-- display result

SELECT result;

END &&

DELIMITER ;

CALL RepeatDemo();