TUT 8 DSW PL-SQL

1)

DELIMITER $$

create procedure factorial(in n int,out result int)

begin

declare i int;

set i=1;

set result=1;

while i<=n

do

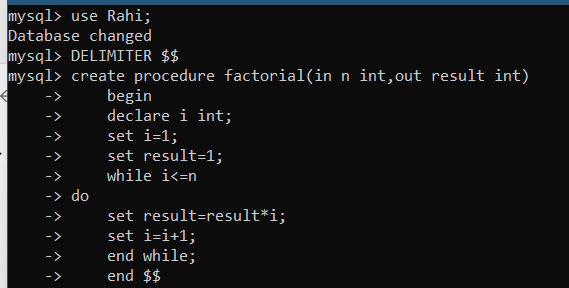
set result=result\*i;

set i=i+1;

end while;

end $$

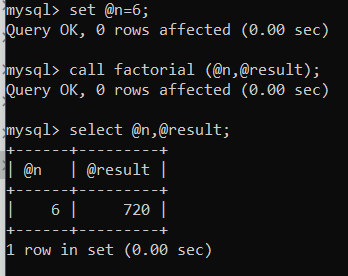
delimiter ;



set @n=6;

call factorial (@n,@result);

select @n,@result;



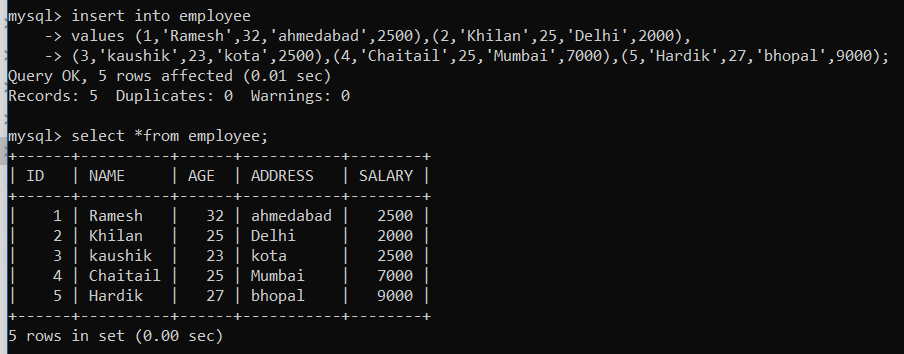
//first of all we will create table

create table employee (ID int,NAME varchar(30),AGE int,ADDRESS varchar(50) ,SALARY int);

insert into employee

values (1,'Ramesh',32,'ahmedabad',2500),(2,'Khilan',25,'Delhi',2000),

(3,'kaushik',23,'kota',2500),(4,'Chaitail',25,'Mumbai',7000),(5,'Hardik',27,'bhopal',9000);



2)

delimiter //

create procedure increment\_salary ()

begin

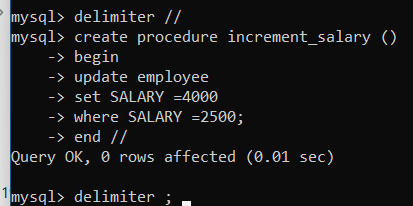
update employee

set SALARY =4000

where SALARY =2500;

end //

delimiter ;



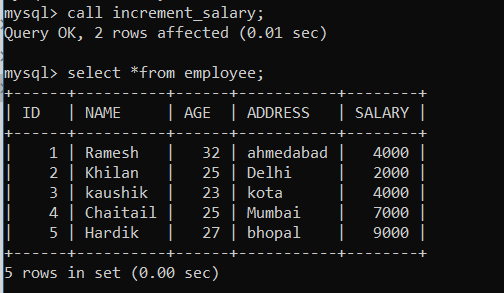
// updation laane ke liye phele call karna zaruri hota hai;

//uske baad select option ka use hoga;

Calling Funtion

call increment\_salary;

select \*from employee;



3)

DELIMITER $$

CREATE FUNCTION fibonacci(num INT)

RETURNS varchar(255)

DETERMINISTIC

BEGIN

DECLARE fib1 INT DEFAULT 0;

DECLARE fib2 INT DEFAULT 1;

DECLARE fib3 INT DEFAULT 0;

DECLARE str VARCHAR(255) DEFAULT '0,1,';

WHILE num >1 DO

SET fib3 = fib1 + fib2;

SET fib1 = fib2;

SET fib2 = fib3;

SET num = num - 1;

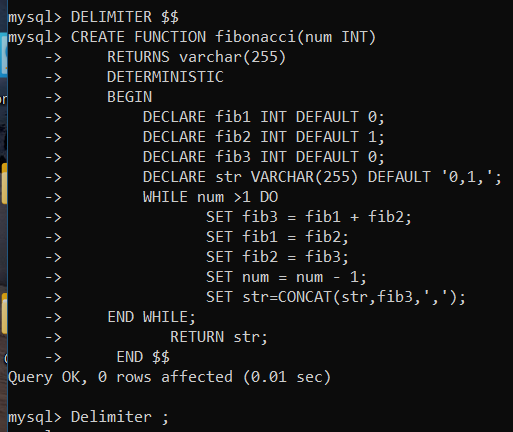
SET str=CONCAT(str,fib3,',');

END WHILE;

RETURN str;

END $$

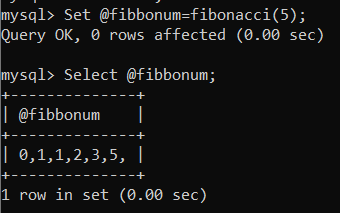
Delimiter ;



Calling function

Set @fibbonum=fibonacci(5);

Select @fibbonum;



4)

delimiter $$

create function EmpSalary()

returns int

deterministic

begin

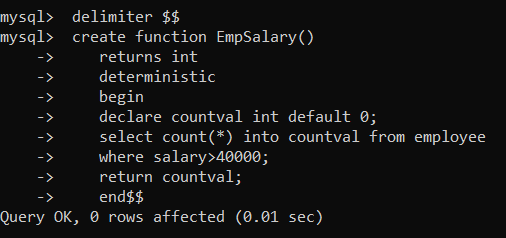
declare countval int default 0;

select count(\*) into countval from employee

where salary>40000;

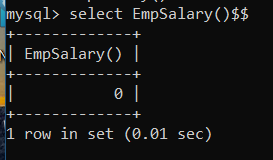
return countval;

end$$



to call

select EmpSalary()$$



5)

delimiter //

create procedure employee()

begin

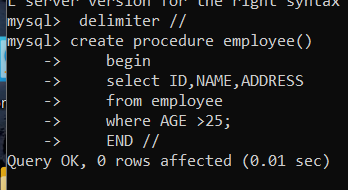
select ID,NAME,ADDRESS

from employee

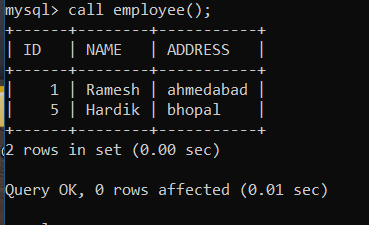
where AGE >25;

END //

delimiter ;



call employee();



6)

create table product(

ID int primary key auto\_increment,

Code varchar(10),

Price int);

create table PriceLogs(

Code varchar(10),

Price int);

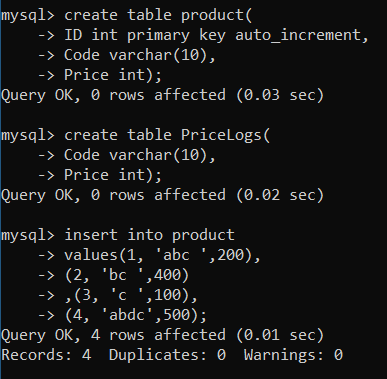
insert into product

values(1, 'abc ',200),

(2, 'bc ',400)

,(3, 'c ',100),

(4, 'abdc',500);



DELIMITER $$

CREATE TRIGGER price\_changes

BEFORE UPDATE

ON product FOR EACH ROW

BEGIN

INSERT INTO PriceLogs VALUES(OLD.code, OLD.price);

END $$

DELIMITER ;

Update product set price=600 where code='abc';