**PROCEDIMIENTOS EN WORKBENCH**

1. **Crearemos una tabla lo más sencilla posible**

* create table t (s1 int) ;
* insert into t values (5) ;
* select \* from t ;

1. **Crearemos un delimitador**

delimiter //

drop procedure if exists p1 ;

create procedure p1() select \* from t ; //

delimiter ;

→ create procedure funciona para insertar datos

→ select \* from t lo usaremos como el body

1. **Con esto podremos llamar a la consulta mediante call**

call p1 ;

1. **Modificaremos el cuerpo para añadir un argumento**

delimiter //

drop procedure if exists p1 ;

create procedure p1(arg1 int) select \* from t where s1 < arg1; //

delimiter ;

1. **Si quisiéramos llamar de nuevo a la consulta deberíamos de aplicar un valor en call**

* **call p1(4) →** No recibiremos ningún valor
* **call p1(6) →** Nos mostrará el valor 5

**EJEMPLO PRÁCTICO:**

**Crear un procedimiento llamado p2 que reciba un argumento de tipo INT y que su body lo que haga es un insert en la tabla t con el valor proporcionado**

1. ***Escribiremos la función correspondiente***

delimiter //

drop procedure if exists p2 ;

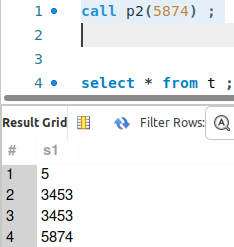
create procedure p2(arg1 int) insert into t values(arg1) ; //

delimiter ;

1. ***Si pulsamos en funcionar, nos dará error, para ello primero llamaremos a p2 con cualquier valor***

call p2(5874) ;

1. ***Volveremos a ejecutar la primera función para que ahora si este correcta***



**CASO PRÁCTICO**

**Crear un procedimiento llamado p3 que reciba un argumento de tipo INT y que su body lo que haga es un insert en la tabla t con el valor proporcionado y luego nos muestra todas las filas con s1 inferior o igual al argumento indicado**

1. ***Primer paso***

drop procedure if exists p3 ;

delimiter //

create procedure p3(arg1 int)

begin

insert into t values(arg1) ;

select \* from t where s1 <= arg1 ;

end ; //

delimiter ;

1. ***Segundo paso***

call p3(6000) ;

1. ***Tercer paso***

select \* from t ;

**CASO PRÁCTICO**

**Crear un procedimiento llamado p4 que reciba un argumento de tipo INT y que su body lo que haga es un insert en la tabla t con el valor proporcionado y luego nos muestra todas las filas con s1 inferior o igual al argumento indicado**

drop procedure if exists p4 ;

delimiter //

create procedure p4(arg1 int)

begin

declare v1 int ;

set v1 = arg1 \* 2 ;

insert into t values(v1) ;

select \* from t where s1 <= v1 ;

end ; //

delimiter ;

**Con esta función multiplicaremos X2 el valor que pulsemos en p4**

call p4(23) ;

**CASO PRÁCTICO**

**Crear un procedimiento que recibe arg1**

* **Si arg1 > 10 se inserta el doble del valor**
* **Si arg1 <= 10 se inserta el triple del valor**
* **En ambos casos, se devuelve todas las filas con valor inferior o igual al insertado**

**if arg > 10 then**

**sentencia1 ;**

**sentencia2 ;**

**else**

**sentencia\_a**

**sentencia\_b**

drop procedure if exists p5 ;

delimiter //

create procedure p5(arg1 int)

begin

declare va1 int ;

if arg1 > 10 then

set va1 = arg1 \* 2 ;

else

set va1 = arg1 \* 3 ;

end if ;

insert into t values(va1) ;

select \* from t ;

end ;

//

delimiter ;

***EJEMPLO CON WHILE:***

drop procedure if exists p6 ;

delimiter //

create procedure p6(arg1 int)

begin

declare v1 int ;

set v1 = 0 ;

while v1 < arg1 do

insert into t values(v1) ;

set v1 = v1 + 1 ;

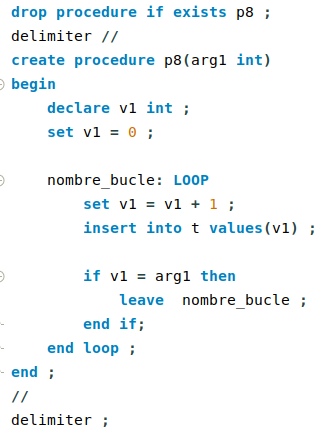
end while ;

end ;

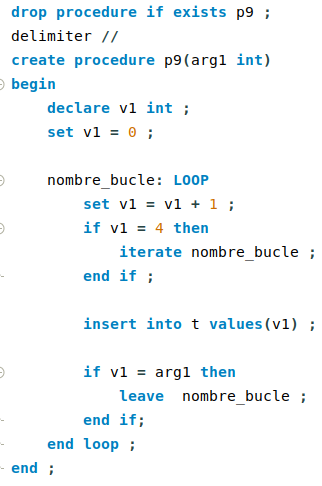
//

delimiter ;

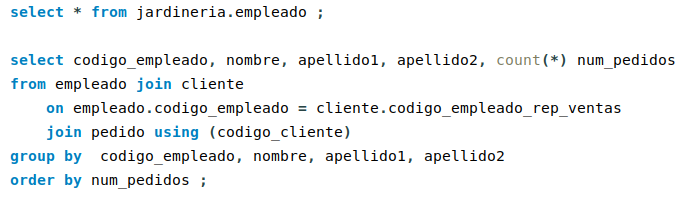
**BUCLES LOOP**



**BUCLES LOOP CON ITERATE**



**NUEVO**



-- Procedimiento que recibe un codigo\_empleado

-- Usa select count(\*) into variable para obtener el número de pedidos que tiene el empleado

-- SI el número de pedidos > 15, el puesto le pone delante "SUPERJEFE"

-- SI el número de pedidos > 5, el puesto le pone delante "JEFE"

-- En otro caso, al puesto le pone delante "PONTE LAS PILAS"

delimiter //

create procedure revisar\_puesto(codigo int)

begin

declare numero\_pedidos int ;

declare nuevo\_puesto varchar(45) ;

select count(\*)

into numero\_pedidos

from cliente join pedido using (codigo\_cliente)

where codigo\_empleado\_rep\_ventas = a\_codigo ;

if numero\_pedidos > 15 then

set nuevo\_puesto= "SUPERJEFE" ;

elseif numero\_pedidos > 5 then

set nuevo\_puesto= "JEFE" ;

else

set nuevo\_puesto= "CUIDADIN" ;

end if;

update empleado

set puesto = concat(nuevo\_puesto, puesto)

where codigo\_empleado = a\_codigo ;

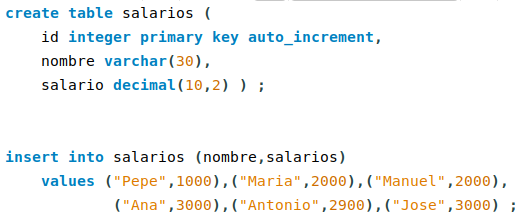
end ;

//

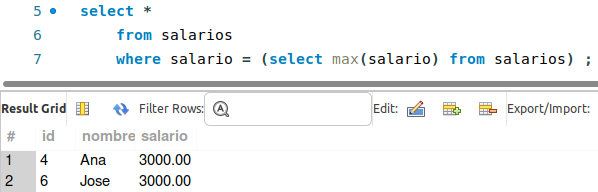
delimiter ;

**PROCEDIMIENTO PARA FUNCIONES EN SUBCONSULTAS:**

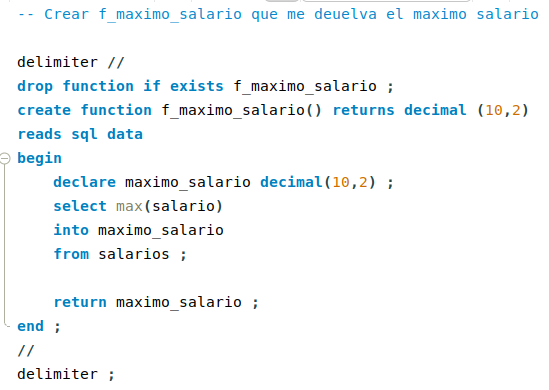
**CASO PARA AVERIGUAR EL SALARIO MÁXIMO**



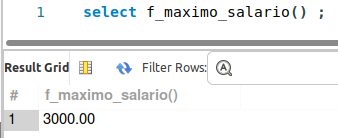
Empezaremos on una con consulta normal y sus datos



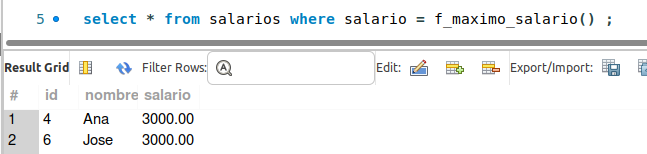
Crearemos nuestra propia subconsulta



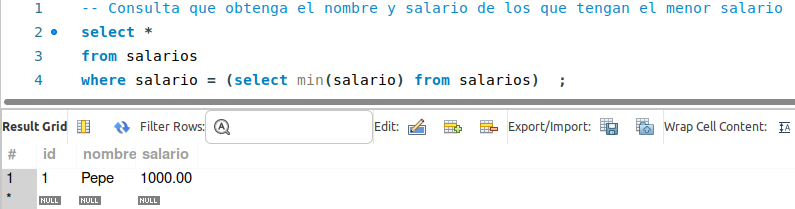
Hecha la función la llamaremos

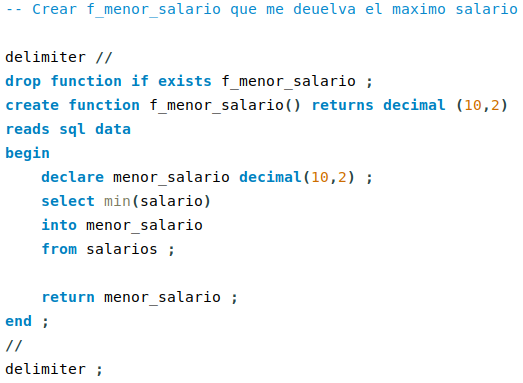


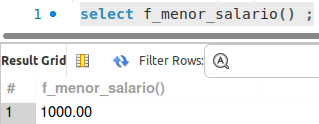
Y volveremos a la subconsulta para cambiarla por la función anteriormente creada

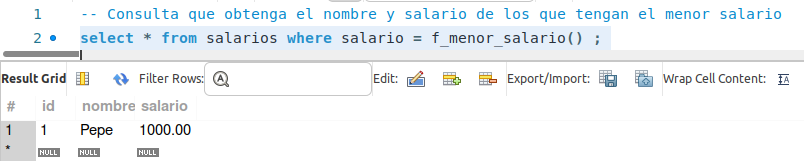


**OTRO CASO PARA AVERIGUAR EL MENOR SALARIO:**









**OTRO CASO PARA AVERIGUAR EL PROMEDIO:**

