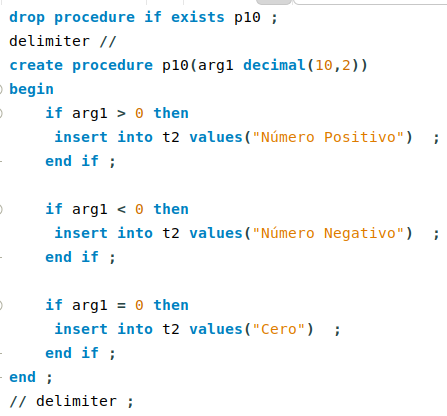
1. **Escribe una función que reciba un número decimal de entrada y que devuelva un texto indicando "número positivo", "número negativo" o "cero", según el valor recibido.**





1. **Escribe una función que reciba un número entero y devuelva un texto, según la nota de un alumno: "Insuficiente" entre 0 y 5, "suficiente" entre 5 y 6, "Bien" entre 6 y 7, "Notable" entre 7 y 9, "Sobresaliente" entre 9 y 10. "incorrecto" en otro caso**

DROP FUNCTION IF EXISTS nota\_alumno;

DELIMITER //

CREATE FUNCTION nota\_alumno(nota INT) RETURNS VARCHAR(50)

BEGIN

DECLARE resultado VARCHAR(50);

IF nota >= 0 AND nota < 5 THEN

SET resultado = 'Insuficiente';

ELSEIF nota >= 5 AND nota < 6 THEN

SET resultado = 'Suficiente';

ELSEIF nota >= 6 AND nota < 7 THEN

SET resultado = 'Bien';

ELSEIF nota >= 7 AND nota < 9 THEN

SET resultado = 'Notable';

ELSEIF nota >= 9 AND nota <= 10 THEN

SET resultado = 'Sobresaliente';

ELSE

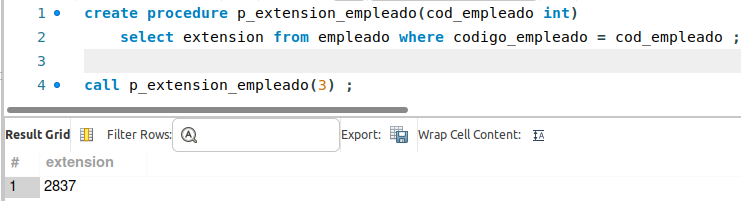
SET resultado = 'Incorrecto';

END IF;

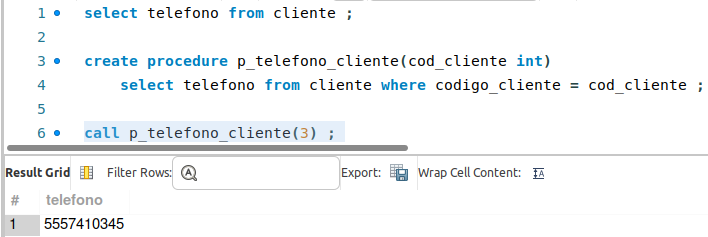
RETURN resultado;

END;

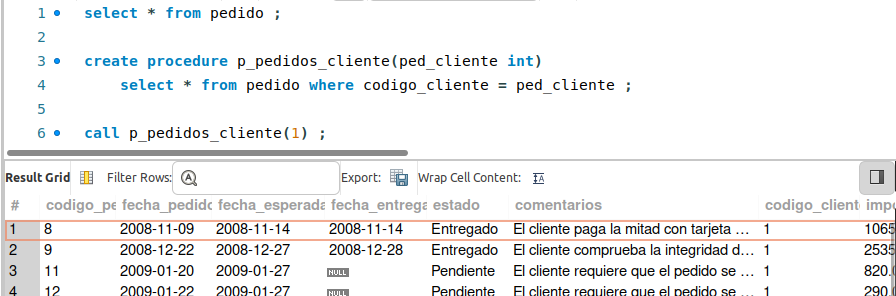
1. **Crea un procedimiento llamado p\_extension\_empleado que reciba como argumento el código de empleado y devuelva su extensión.**



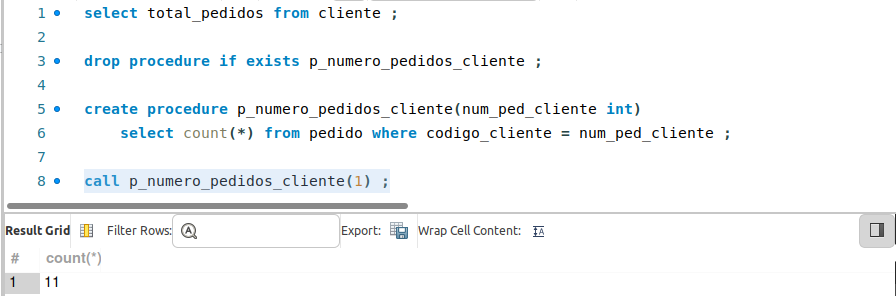
1. **Crea un procedimiento llamado p\_telefono\_cliente que reciba como argumento el código de cliente y devuelva su teléfono.**



1. **Crea un procedimiento llamado p\_pedidos\_cliente que reciba como argumento el código de cliente y devuelva los datos de los pedidos que ha realizado (select ... from pedidos where ...)**



1. **Crea un procedimiento llamado p\_numero\_pedidos\_cliente que reciba como argumento el código de cliente y devuelva el número total de pedidos que ha realizado (select count ...)**



1. **Crea un procedimiento llamado p\_productos\_gama que reciba como argumento una 'gama' y devuelva los productos de dicha gama.**

# **Base de Datos y Consultas de Películas (ahora con funciones)**

1. **Número de películas estrenadas en cada año con el tema “suspense”.**

select distinct p.anioestreno, f\_num\_peliculas\_tema\_ano(t.coditema, p.anioestreno)

from pelicula p, tema t

where t.nombre = "suspense"

order by p.anioestreno ;

* 1. **Crear una función f\_num\_peliculas\_tema\_ano(a\_tema, a\_ano) que devuelva el número de películas del tema y año indicado**
  2. **Escribir la consulta usando la función**

drop function if exists f\_num\_peliculas\_tema\_ano ;

delimiter //

create function f\_num\_peliculas\_tema\_ano(a\_coditema int, a\_ano int) returns int

reads sql data

begin

declare num\_peliculas int ;

select count(distinct p.codipelicula)

into num\_peliculas

from pelicula p join tema\_pelicula t using (codipelicula)

where p.anioestreno = a\_ano and t.coditema = a\_coditema ;

return num\_peliculas ;

end ;

//

delimiter ;

1. **La película más corta de cada año**

select f\_duracion\_peli\_mas\_corta(1979) ;

select \*

from pelicula

where duracion=f\_duracion\_peli\_mas\_corta(anioestreno)

order by anioestreno, titulo ;

* 1. **Crear una función f\_duracion\_peli\_mas\_corta(a\_ano) que devuelva la duración de la película más corta del año indicado**
  2. **Escribir la consulta usando la función**

drop function if exists f\_duracion\_peli\_mas\_corta ;

delimiter //

create function f\_duracion\_peli\_mas\_corta(a\_ano int) returns int

reads sql data

begin

declare min\_duracion int ;

select min(duracion)

into min\_duracion

from pelicula

where anioestreno=a\_ano ;

return min\_duracion ;

end ;

//

delimiter ;

1. **La película más larga de cada año**
   1. **Crear una función f\_duracion\_peli\_mas\_larga(a\_ano) que devuelva la duración de la película más corta del año indicado**
   2. **Escribir la consulta usando la función**
2. **Mostrar la duración media de las películas de la productora que más películas tiene.**

select avg(duracion)

from pelicula

where codiproductora=f\_productora\_mas\_peliculas()

* 1. **Crear una función f\_numero\_peliculas(a\_productora) que devuelva el número de películas que tiene una productora**

delimiter //

create function f\_numero\_peliculas(a\_productora int) returns int

reads sql data

begin

declare num\_peli int ;

select count(\*)

into num\_peli

from pelicula

where codiproductora=a\_productora ;

return num\_peli ;

end ;

//

delimiter ;

* 1. **Crear una función f\_productora\_mas\_peliculas() que devuelva la productora que más películas tiene, usando la función anterior.**
     1. **Escribe la consulta usando la función anterior.**

drop function if exists f\_productora\_mas\_peliculas ;

delimiter //

create function f\_productora\_mas\_peliculas() returns int

reads sql data

begin

declare v\_codiproductora int ;

declare v\_np int ;

select codiproductora, f\_numero\_peliculas(codiproductora) np

into v\_codiproductora, v\_np

from productora

order by np desc

limit 1 ;

return v\_codiproductora;

end ;

//

delimiter ;

# **Funciones y consultas sobre la BBDD Reyes**

1. **f\_numero\_pedidos(a\_id\_juguete, a\_ano\_carta) que devuelva cuántas veces se ha pedido un juguete en un determinado año**

drop function if exists f\_numero\_pedidos ;

delimiter //

create function f\_numero\_pedidos(a\_id\_juguete int, a\_ano\_carta int) returns int

reads sql data

begin

declare veces\_pedido int ;

select count(id\_juguete)

into veces\_pedido

from cartas\_reyes join juguetes\_carta using(id\_carta)

where a\_id\_juguete = id\_juguete

and a\_ano\_carta = ano\_carta

group by id\_juguete ;

return veces\_pedido ;

end ;

// delimiter ;

* 1. **Usando la función anterior, escribe la consulta: Listado con el nombre de los juguetes y de los países en los que las existencias de dichos juguetes sean menores a la cantidad de juguetes pedidos por las familias de dicho país en el año 2007.**

1. **f\_numero\_pedidos\_tipo\_juguete(a\_cod\_tipo, a\_ano\_carta) que devuelve el número de juguetes de un determinado tipo que se han pedido en el año indicado.**

delimiter //

create function f\_numero\_pedidos\_tipo\_juguete(a\_cod\_tipo int, a\_ano\_carta int)

returns int

reads sql data

begin

declare v\_num\_pedidos int;

select count(\*)

into v\_num\_pedidos

from juguetes j join juguetes\_carta jc using (id\_juguete)

join cartas\_reyes c using (id\_carta)

where j\_cod\_tipo=a\_cod\_tipo and c\_ano\_carta=a\_ano\_carta ;

return v\_num\_pedidos ;

end ;

//

* 1. **Usando la función anterior, escribe la consulta: Nombre del país o países en el que las personas, en este año 2007, hayan pedido mayor número de juguetes bélicos (con tipos\_juguete.descripcion=”BELICOS”).**

1. f\_comportamiento(a\_id\_persona, a\_ano) que devuelve el comportamiento de la persona en el año indicado
   1. Usando esa función, escribe la consulta: Listado con el nombre y apellidos de los hijos que hayan tenido mejor comportamiento en el año 2007 que en el 2006, ordenado alfabéticamente por apellidos y luego por nombre.
2. f\_num\_familias\_residentes(a\_cod\_pais) que devuelve el número de familias que residen en el pais indicado
   1. Con esa función, escribe la consulta: Listado con el número de familias residentes en cada país, ordenado por número de familias residentes y por nombre del país.
3. f\_importe\_carta(id\_carta) que devuelve el importe total de los juguetes incluidos en la carta
   1. Con esa función, escribe la consulta: Listado con la familia (id\_familia, apellidos) y el año, con el importe de los juguetes que ha pedido dicho año.