

Practica 4.1

P



Ricardo sorin alamajn

Contenido

[1. Función y Procedimiento 2](#_Toc166532849)

[o Función 2](#_Toc166532850)

[o Procedimiento 4](#_Toc166532851)

[2. Función y Procedimiento 6](#_Toc166532852)

[o Función 6](#_Toc166532853)

[o Procedimiento 7](#_Toc166532854)

[3. Función y Procedimiento 9](#_Toc166532855)

[o Función 9](#_Toc166532856)

[o Procedimiento 9](#_Toc166532857)

[4. Procedimiento Libre 9](#_Toc166532858)

[5. Procedimiento Libre 9](#_Toc166532859)

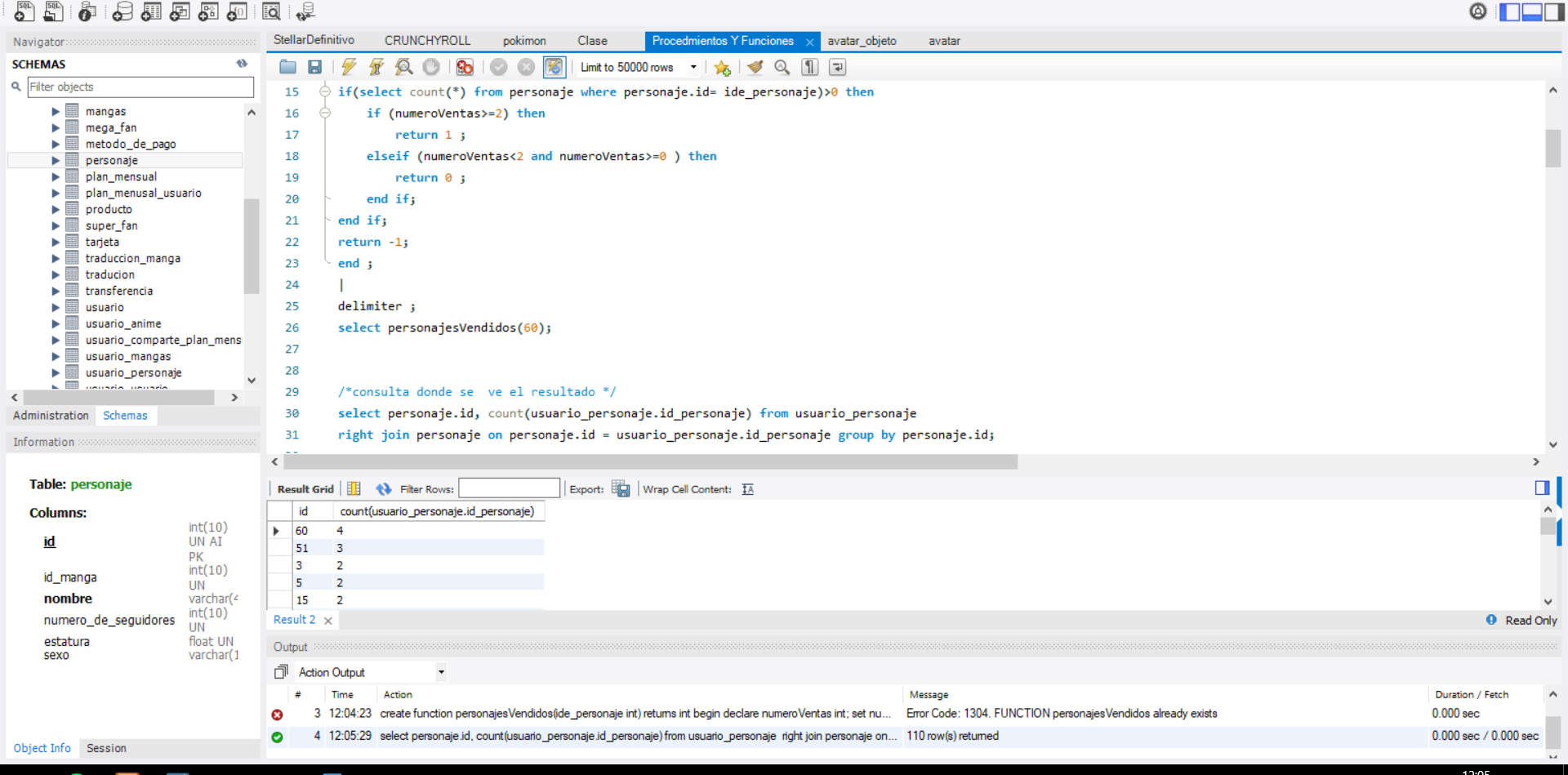
# Función y Procedimiento

## Función

/\*personajes de los cuales se hayan comprado 2 o más figuras (1) o que no tenga ninguna figara vendida (0) o que no existan -1 \*/

select personaje.id, count(usuario\_personaje.id\_personaje) from usuario\_personaje

right join personaje on personaje.id = usuario\_personaje.id\_personaje group by personaje.id;

**

drop function personajesVendidos;

delimiter |

create function personajesVendidos(ide\_personaje int)

returns int

begin

declare numeroVentas int;

set numeroVentas = (select count(usuario\_personaje.id\_personaje) from usuario\_personaje

right join personaje on personaje.id = usuario\_personaje.id\_personaje where ide\_personaje= id\_personaje group by personaje.id);

if(select count(\*) from personaje where personaje.id= ide\_personaje)>0 then

if (numeroVentas>=2) then

return 1 ;

elseif (numeroVentas<2 and numeroVentas>=0 ) then

return 0 ;

end if;

end if;

return -1;

end ;

|

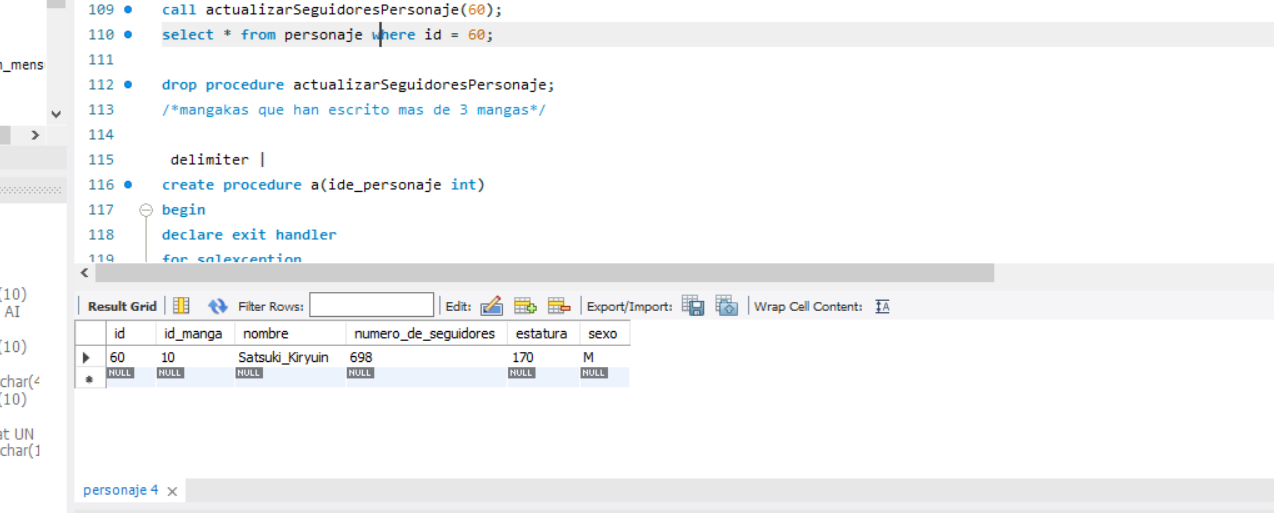
delimiter ;

select personajesVendidos(60);



## Procedimiento

*/\*dada una id de personaje funcion(personajesVendidos), si no se ha vendido ninguna vez le vamos a subir los seguidores en 1000 y si tiene mas de 2 figuras vendidas le aumentaremos el numero de seguidores en +100\*/*

**

*delimiter |*

*create procedure actualizarSeguidoresPersonaje(ide\_personaje int)*

*begin*

*declare numeroSeguidoresPersonaje int;*

*set numeroSeguidoresPersonaje = (select numero\_de\_seguidores from personaje where id = ide\_personaje);*

*if(select personajesVendidos(ide\_personaje)) =1 then*

*update personaje set numero\_de\_seguidores = numeroSeguidoresPersonaje+100 where id = ide\_personaje;*

*elseif(select personajesVendidos(ide\_personaje))=0 then*

*update personaje set numero\_de\_seguidores = numeroSeguidoresPersonaje+1000 where id = ide\_personaje;*

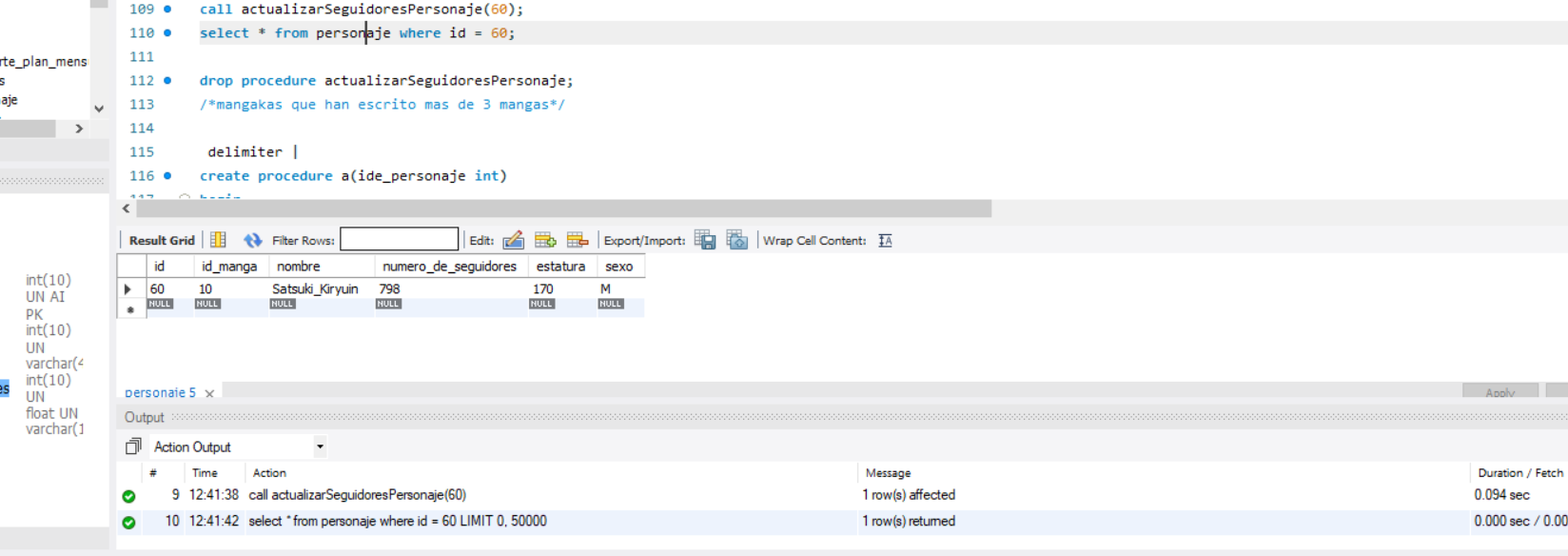
*end if;*

*end ;*

*|*

*delimiter ;*

llamada al procedimiento:



call actualizarSeguidoresPersonaje(60);

select \* from personaje where id = 60;

# Función y Procedimiento

## Función

*/\*Funcion que calcula el tiempo que lleva un usiario dada su ide con el servicio\*/*

drop function tiempoConaLaPlataforma;

delimiter |

create function tiempoConaLaPlataforma(ide\_usuario int)

returns int

begin

/\*sacamos la variable que nos interesa \*/

declare fecha\_inicio1 date;

declare fecha\_final1 date;

set fecha\_inicio1=(select fecha\_inicio from plan\_mensual

join plan\_menusal\_usuario on plan\_menusal\_usuario.id\_plan\_mensual =plan\_mensual.id

join usuario on usuario.id = plan\_menusal\_usuario.id\_usuario

where usuario.id =ide\_usuario);

set fecha\_final1=(select fecha\_final from plan\_mensual

join plan\_menusal\_usuario on plan\_menusal\_usuario.id\_plan\_mensual =plan\_mensual.id

join usuario on usuario.id = plan\_menusal\_usuario.id\_usuario

where usuario.id = ide\_usuario);

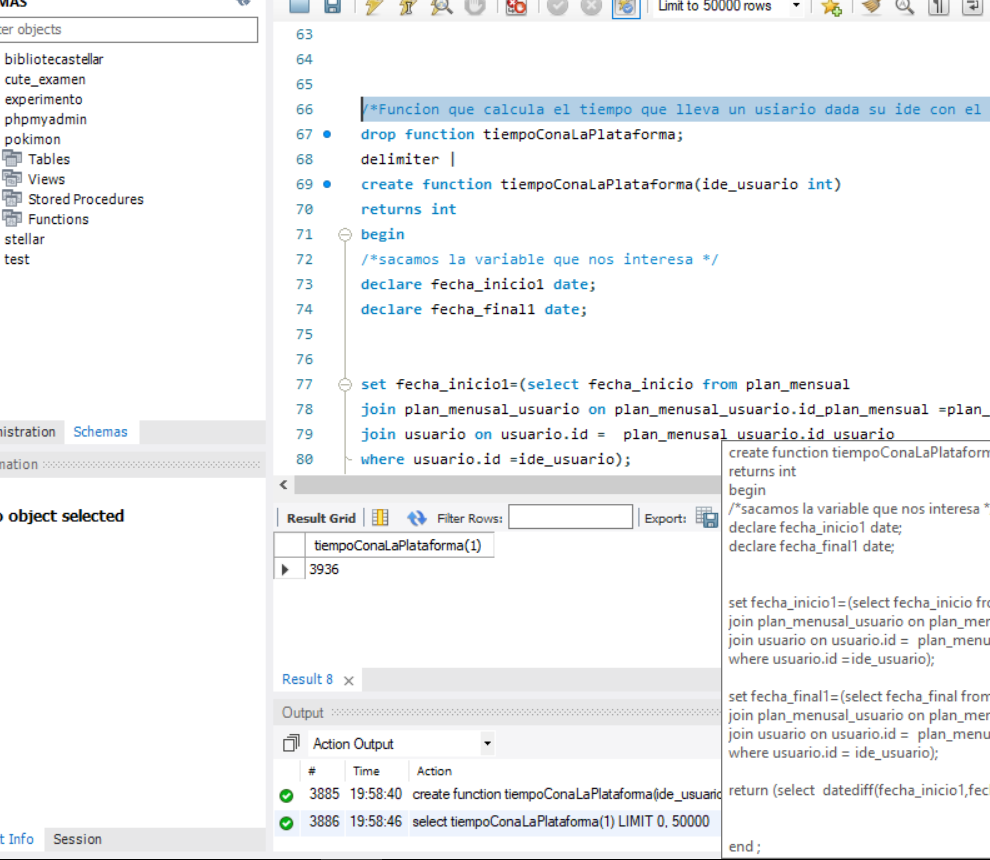
return (select datediff(fecha\_inicio1,fecha\_final1));

end ;

|

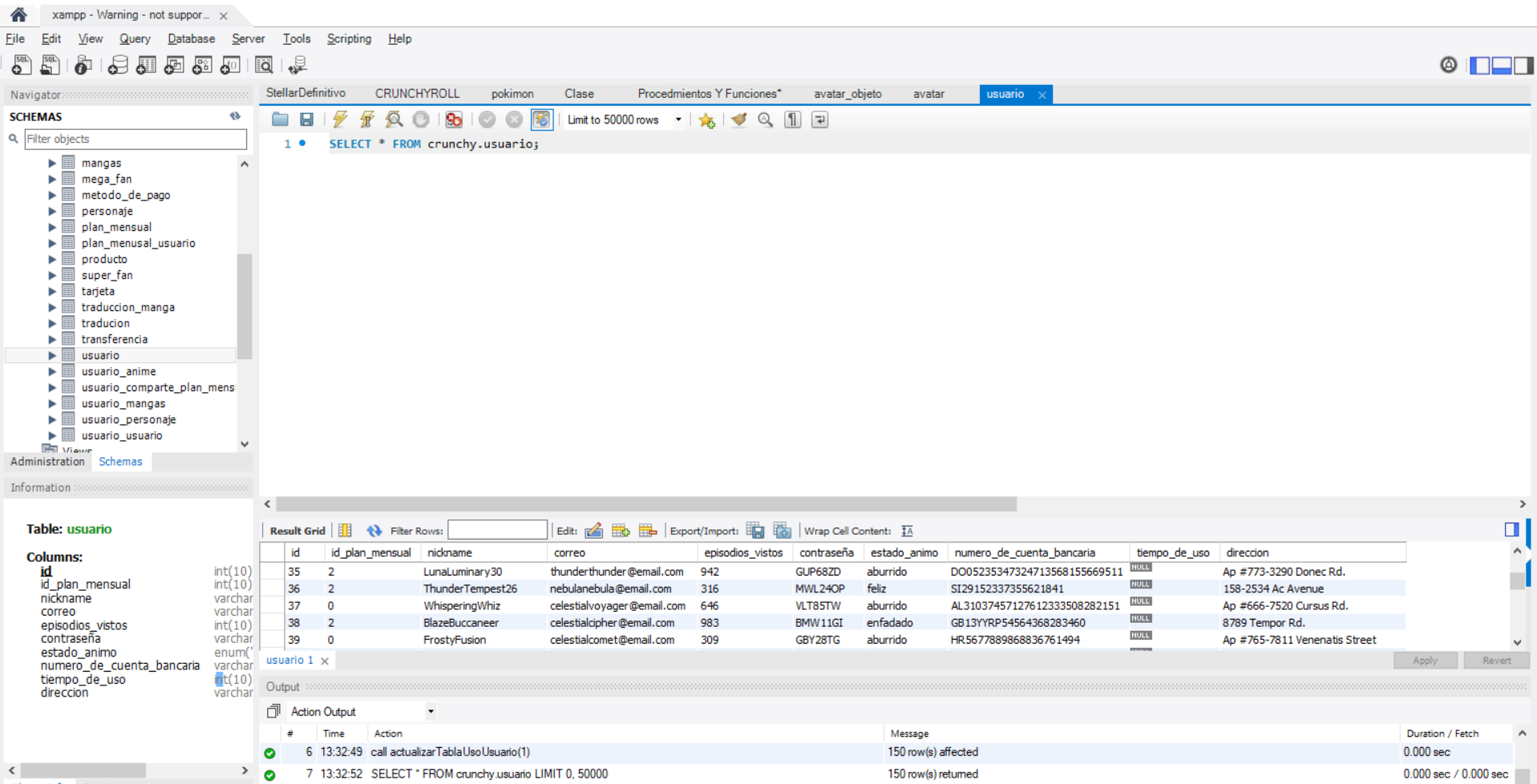
delimiter ;

select tiempoConaLaPlataforma(1);



## Procedimiento

*/\*actualizaremos la columna en la tabla usuario donde le asignamos el tiempo que llevan en la plataforma !!!! puede haber alguno negativo ya que esto no estuvo controlado al hacer la base\*/*

**

**Función**

*delimiter |*

*create procedure actualizarTablaUsoUsuario(ide\_usuario int)*

*begin*

*declare exit handler*

*for sqlexception*

*begin*

*rollback;*

*end;*

*start transaction;*

*if( (select id from usuario where id=ide\_usuario)) >0 then*

*update usuario*

*set tiempo\_de\_uso = (select tiempoConaLaPlataforma(ide\_usuario) );*

*end if;*

*commit;*

*end ;*

*|*

*delimiter ;*

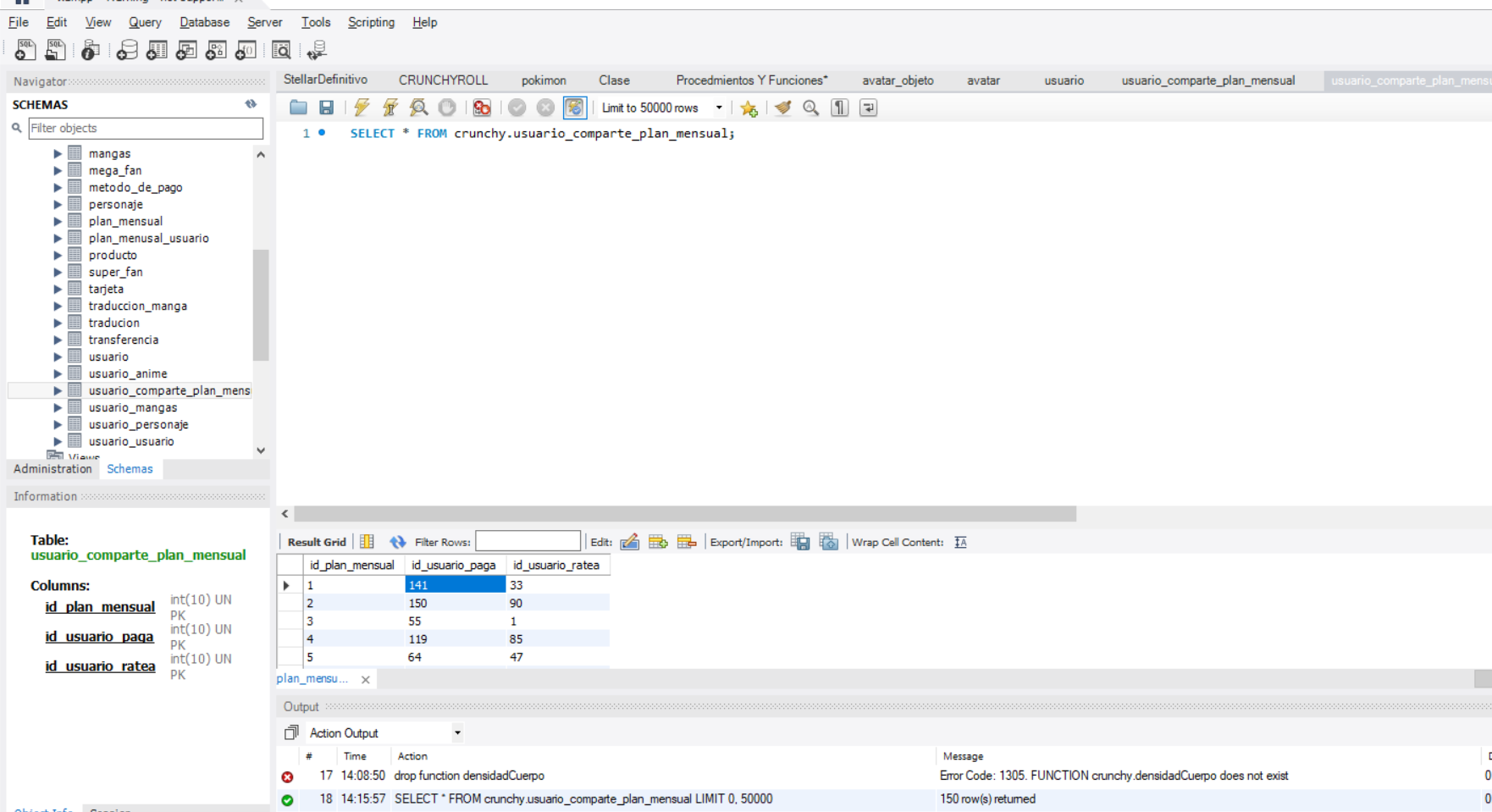
*call actualizarTablaUsoUsuario(1);*

*drop procedure actualizarTablaUsoUsuario;*

# Función y Procedimiento

## Función

/\*usuarios que comparten su suscripción con otro le entra 2 id de usuario (devuelve 1 si los dos comparen el plan menusual,0 si no y -1 si no existe)\*/



delimiter |

create function usuarioComparte(ide\_usuario int,ide\_usario\_recibe int)

returns int

begin

/\*comprobamos is existen los dos ususarios\*/

declare id\_plan\_usuario1 int;

declare id\_plan\_usuario2 int ;

if((select id from usuario where id=ide\_usuario)) >0 then

set id\_plan\_usuario1 = (select id\_plan\_mensual from usuario\_comparte\_plan\_mensual where id\_usuario\_paga= ide\_usuario );

if((select id from usuario where id=ide\_usario\_recibe)) >0 then

set id\_plan\_usuario2 = (select id\_plan\_mensual from usuario\_comparte\_plan\_mensual where id\_usuario\_ratea= ide\_usario\_recibe );

/\*vemos si comparten los dos el plan menusal \*/

if(id\_plan\_usuario1 = id\_plan\_usuario2)>0 then

return 1;

else

return 0;

end if;

end if;

end if;

end ;

|

delimiter ;

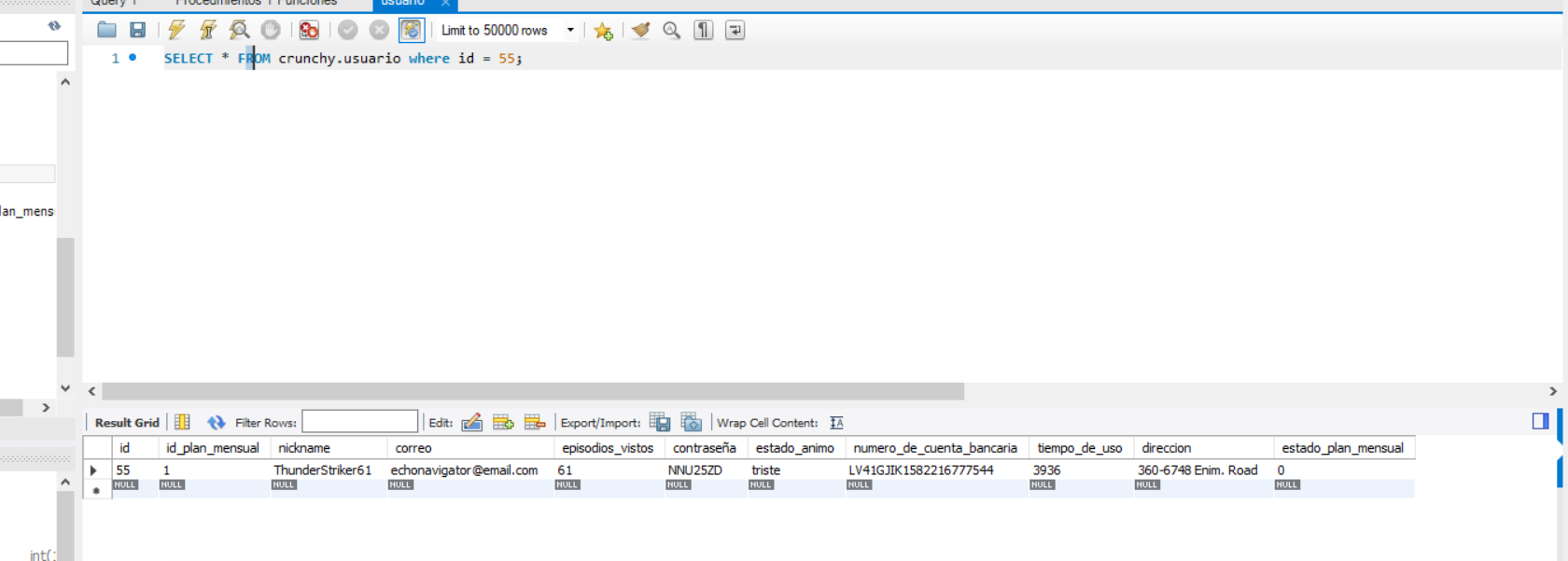
select usuarioComparte(55,1);

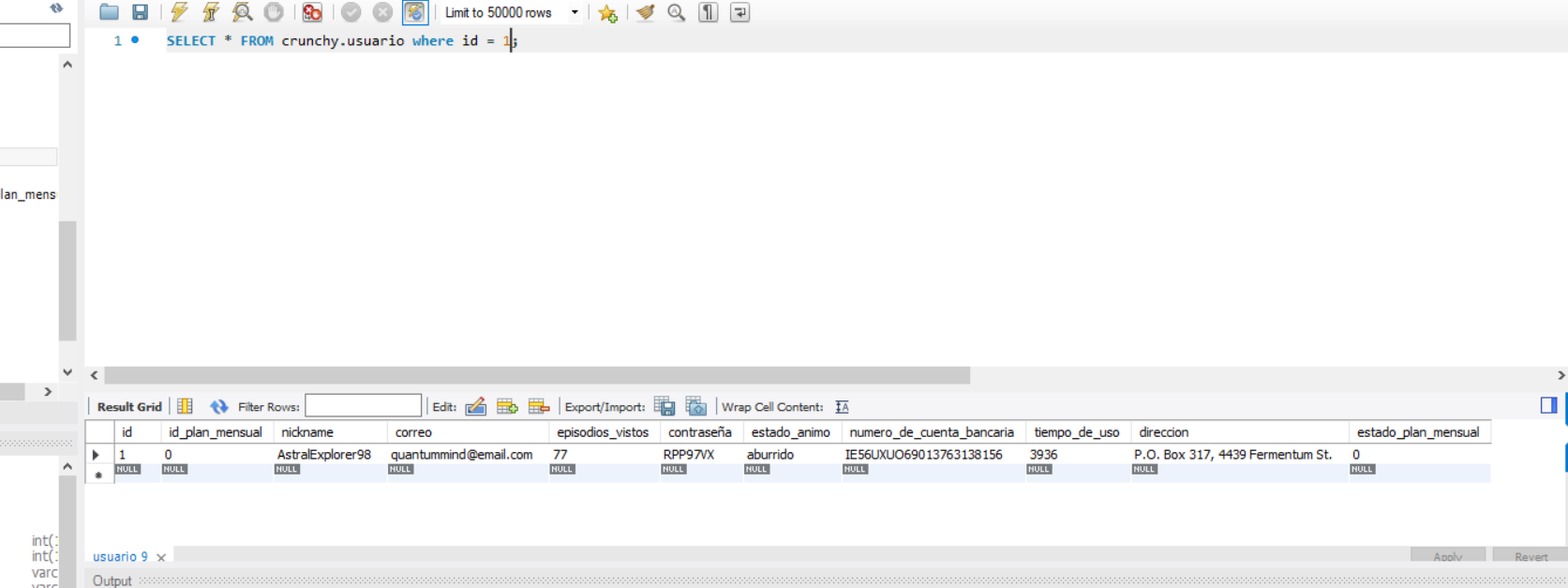
drop function usuarioComparte;

## Procedimiento

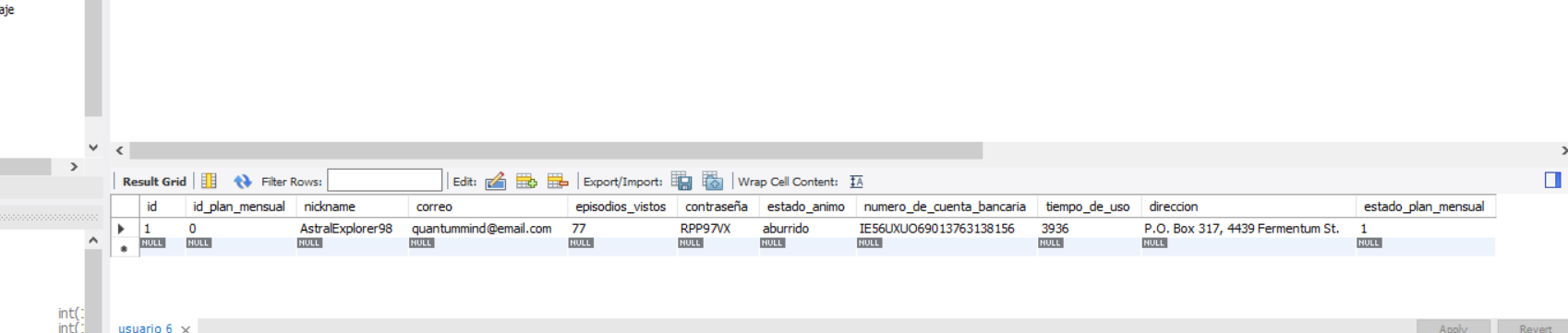
/\*procedimientop que añade una columna a la tbala usuario en la que pondra estado\_plan\_mneusal y si lo compare pondra 1 y si no pondra no 0\*/

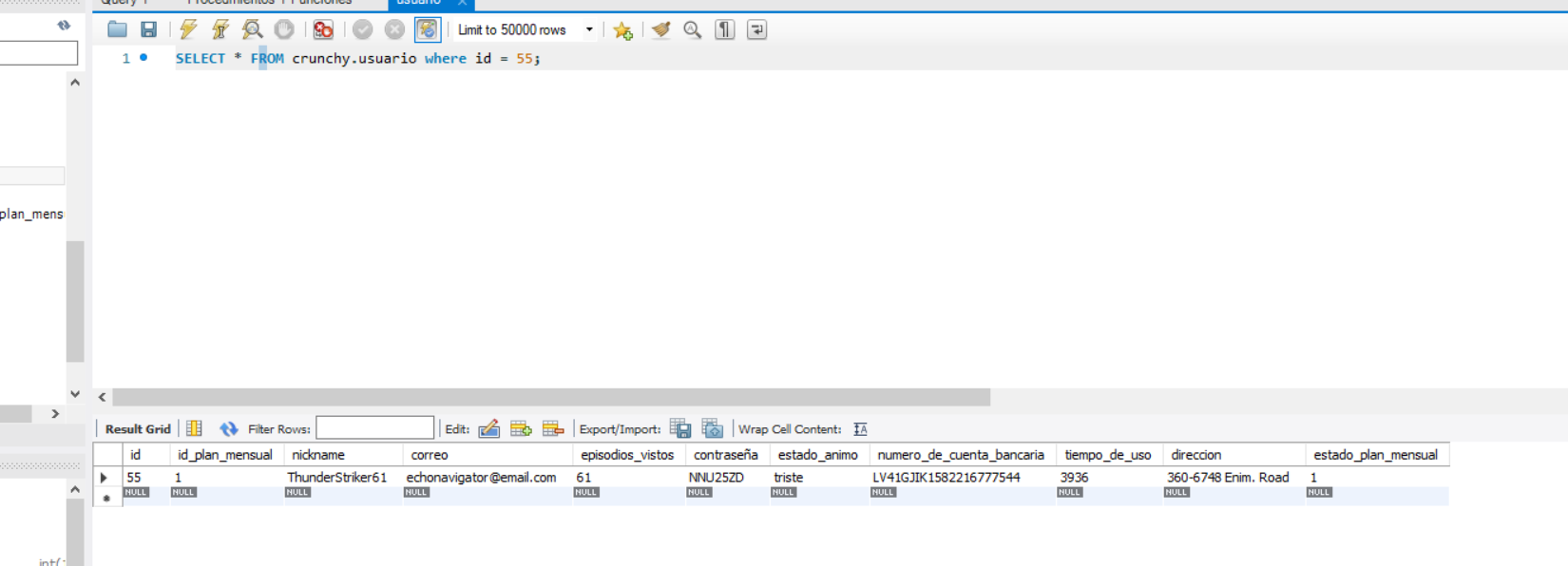
Antes:





Despues:





begin

declare exit handler

for sqlexception

begin

rollback;

end;

start transaction;

if( (select id from usuario where id=ide\_usuario)) >0 then

if( (select id from usuario where id=ide\_usario\_recibe)) >0 then

if(usuarioComparte(ide\_usuario,ide\_usario\_recibe) = 1 or usuarioComparte(ide\_usuario,ide\_usario\_recibe) = 0 ) then

update usuario

set estado\_plan\_mensual = (select 1 from usuario where id = ide\_usuario );

update usuario

set estado\_plan\_mensual =(select 1 from usuario where id = ide\_usario\_recibe );

end if;

end if;

end if;

commit;

end ;

|

delimiter ;

call compartenPlanMenusal(5,1);

drop procedure compartenPlanMenusal;

# Procedimiento Libre

# Procedimiento Libre