

PRACTICA 4.1

P



RICARDO SORIN ALAMAJN

Contenido

- 1. Función y Procedimiento 2
 - Función 2
 - Procedimiento 4
- 2. Función y Procedimiento 6
 - Función 6
 - Procedimiento 7
- 3. Función y Procedimiento 9
 - Función 9
 - Procedimiento 11
- 4. Procedimiento Libre 13
- 5. Procedimiento Libre 15

1. Función y Procedimiento

o Función

/*personajes de los cuales se hayan comprado 2 o más figuras (1) o que no tenga ninguna figura 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;
```

The screenshot shows a SQL IDE interface with a schema explorer on the left, a central editor with a function definition, and a results pane at the bottom.

Schema Explorer: Lists various tables including mangas, mega_fan, metodo_de_pago, personaje, plan_mensual, plan_mensual_usuario, producto, super_fan, tarjeta, traduccion_manga, traduccion, transferencia, usuario, usuario_anime, usuario_comparte_plan_mens, usuario_mangas, and usuario_personaje.

Function Definition:

```
15 if(select count(*) from personaje where personaje.id= ide_personaje)>0 then  
16 if (numeroVentas>=2) then  
17 return 1 ;  
18 elseif (numeroVentas<2 and numeroVentas>=0 ) then  
19 return 0 ;  
20 end if;  
21 end if;  
22 return -1;  
23 end ;  
24 |  
25 delimiter ;  
26 select personajesVendidos(00);  
27  
28  
29 /*consulta donde se ve el resultado */  
30 select personaje.id, count(usuario_personaje.id_personaje) from usuario_personaje  
31 right join personaje on personaje.id = usuario_personaje.id_personaje group by personaje.id;
```

Result Grid:

id	count(usuario_personaje.id_personaje)
60	4
51	3
3	2
5	2
15	2

Action Output:

#	Time	Action	Message	Duration / Fetch
3	12:04:23	create function personajesVendidos(ide_personaje int) returns int begin declare numeroVentas int; set nu...	Error Code: 1304. FUNCTION personajesVendidos already exists	0.000 sec
4	12:05:29	select personaje.id, count(usuario_personaje.id_personaje) from usuario_personaje right join personaje on...	110 row(s) returned	0.000 sec / 0.000 sec

```

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 ;

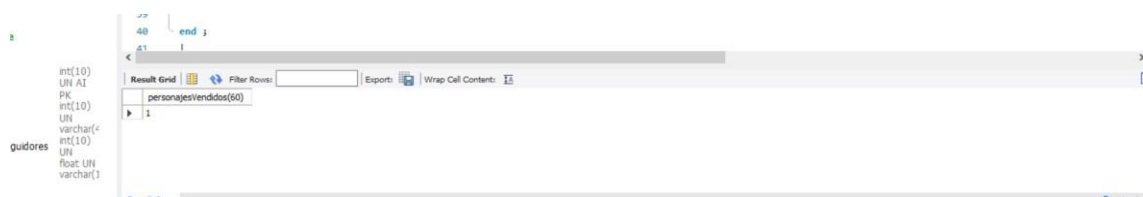
end if;

end if; return -1; end ; |

delimiter ; select

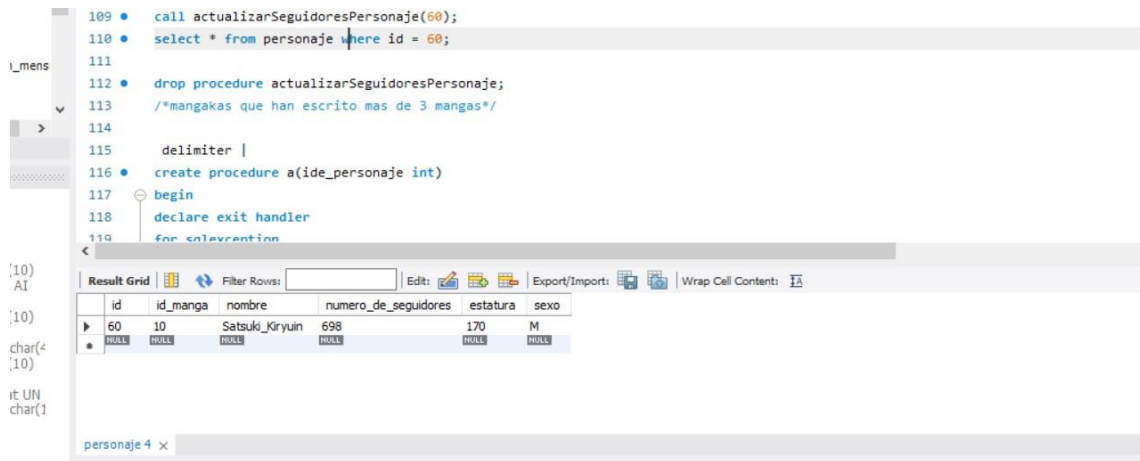
personajesVendidos(60);

```



○ 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:

109 • call actualizarSeguidoresPersonaje(60);
110 • select * from personaje where id = 60;
111
112 • drop procedure actualizarSeguidoresPersonaje;
113 /*mangakas que han escrito mas de 3 mangas*/
114
115 delimiter |
116 • create procedure a(ide_personaje int)
117

Result Grid

	id	id_manga	nombre	numero_de_seguidores	estatura	sexo
▶	60	10	Satsuki_Kiryuin	798	170	M
•	DATE	DATE	DATE	DATE	DATE	DATE

personaje 5 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
9	12:41:38	call actualizarSeguidoresPersonaje(60)	1 row(s) affected	0.094 sec
10	12:41:42	select * from personaje where id = 60 LIMIT 0, 50000	1 row(s) returned	0.000 sec / 0.00

call actualizarSeguidoresPersonaje(60); select

* from personaje where id = 60;

2. Función y Procedimiento

○ Función

```
/*Funcion que calcula el tiempo que lleva un usuario 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;

if( (select count(*) from usuario where id=ide_usuario)) >0 then
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 if;

end ;

|
delimiter ;
```

select tiempoConaLaPlataforma(1);

The screenshot shows a MySQL IDE interface. On the left, a sidebar lists database objects: 'er objects', 'bibliotecastellar', 'cute_examen', 'experimento', 'phpmyadmin', 'pokimon', 'Tables', 'Views', 'Stored Procedures', 'Functions', 'stellar', and 'test'. The main editor displays SQL code for creating a function named 'tiempoConaLaPlataforma'. The code includes comments in Spanish and SQL syntax for dropping the function, creating it with parameters, and using 'declare', 'set', 'join', and 'where' clauses. Below the editor, a 'Result Grid' shows a single row with the value '3936'. The 'Output' pane at the bottom shows a log of actions, including the creation of the function and the execution of the select statement with a limit of 50,000 rows.

```

63
64
65
66  /*Funcion que calcula el tiempo que lleva un usuario dada su ide con el
67  • drop function tiempoConaLaPlataforma;
68  delimiter |
69  • create function tiempoConaLaPlataforma(ide_usuario int)
70  returns int
71  begin
72  /*sacamos la variable que nos interesa */
73  declare fecha_inicio1 date;
74  declare fecha_final1 date;
75
76
77  set fecha_inicio1=(select fecha_inicio from plan_mensual
78  join plan_mensual_usuario on plan_mensual_usuario.id_plan_mensual =plan_
79  join usuario on usuario.id = plan_mensual_usuario.id usuario
80  where usuario.id =ide_usuario);

```

Result Grid

	tiempoConaLaPlataforma(1)
▶	3936

Result 8 x

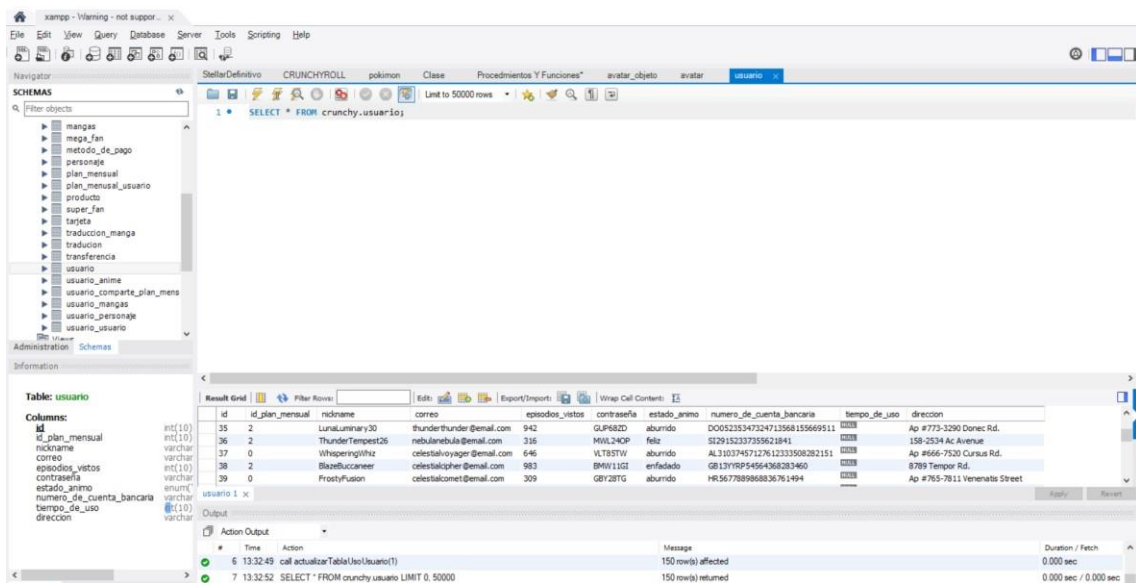
Output

Action Output

#	Time	Action
3885	19:58:40	create function tiempoConaLaPlataforma(ide_usuario
3886	19:58:46	select tiempoConaLaPlataforma(1) LIMIT 0, 50000

o 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*/*



Procedimiento

delimiter |

create procedure actualizarTablaUsoUsuario(id_usuario int)

begin

declare exit handler for sqlexception begin rollback;

end;

start transaction;

if((select count(*) from usuario where id=id_usuario)) >0 then

update usuario

set tiempo_de_uso = (select

tiempoConaLaPlataforma(id_usuario))

where id = id_usuario; end if;

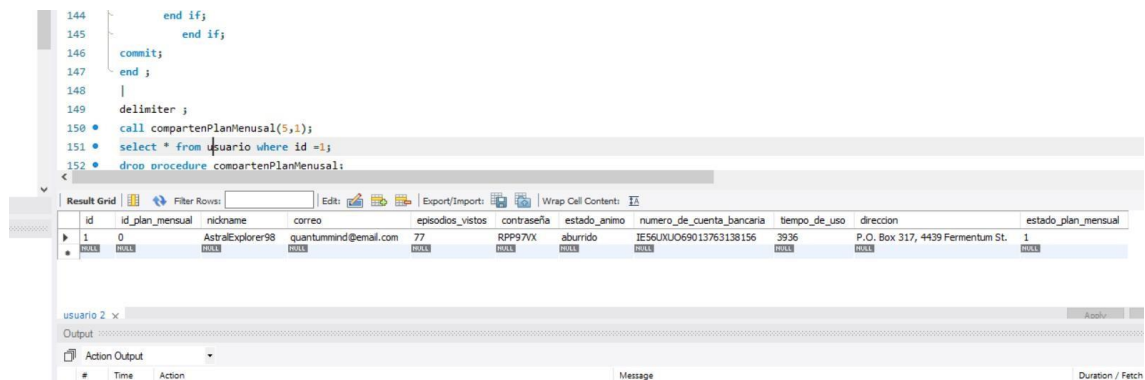
commit; end

;

| delimiter ; call

actualizarTablaUsoUsuario(1); select *

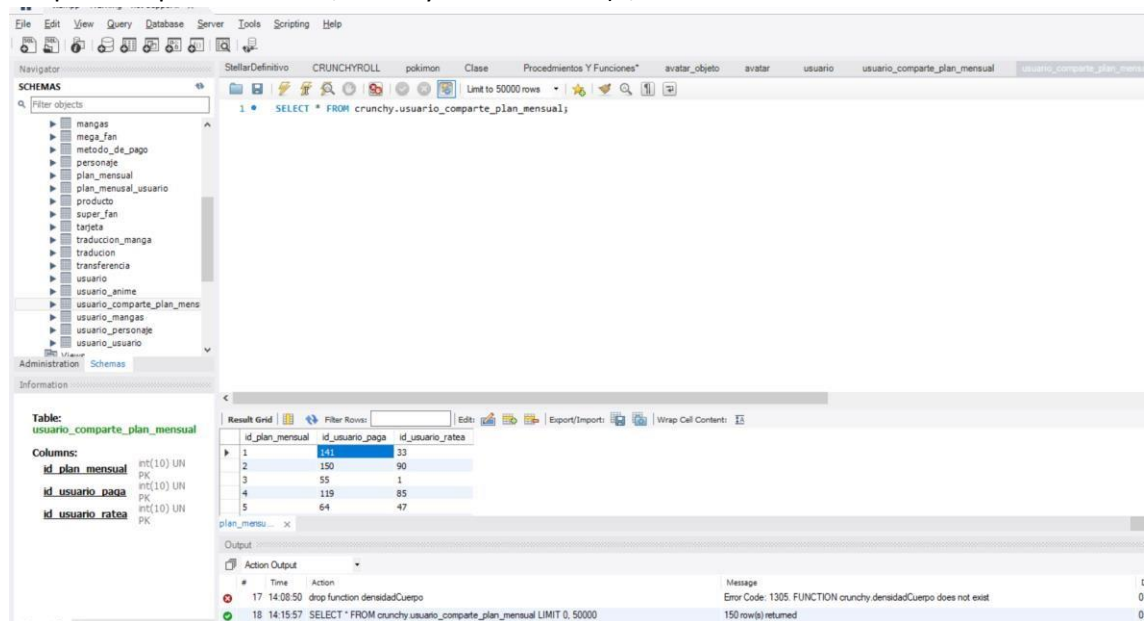
from usuario where id =1; drop procedure
 actualizarTablaUsoUsuario; select * from
 usuario ;



3. Función y Procedimiento

o Función

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



```

delimiter |

create function usuarioComparte(ide_usuario int,ide_usuario_recibe
int)
returns int
begin
/*comprobamos si existen los dos usuarios*/

declare id_plan_usuario1 int;
declare id_plan_usuario2 int ;

if((select count(*) 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 count(*) from usuario where id=ide_usuario_recibe)) >0 then
set id_plan_usuario2 = (select id_plan_mensual from usuario_comparte_plan_mensual
where id_usuario_ratea= ide_usuario_recibe );
/*vemos si comparten los dos el plan mensual */
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;

```

o Procedimiento

/*procedimientop que añade una columna a la tballa usuario en la que pondra estado_plan_mneusal y si lo compare pondra 1 y si no pondra no 0*/

Antes:

Query 1: `SELECT * FROM crunchy.usuario where id = 55;`

id	id_plan_mensual	nickname	correo	episodios_vistos	contraseña	estado_animo	numero_de_cuenta_bancaria	tiempo_de_uso	direccion	estado_plan_mensual
55	1	ThunderStriker61	echonavigator@email.com	61	NNU25ZD	triste	LV41GJ0K1582216777544	3936	360-6748 Enem. Road	0

Query 2: `SELECT * FROM crunchy.usuario where id = 1;`

id	id_plan_mensual	nickname	correo	episodios_vistos	contraseña	estado_animo	numero_de_cuenta_bancaria	tiempo_de_uso	direccion	estado_plan_mensual
1	0	AstraExplorer98	quantummind@email.com	77	RPP97YX	aburrido	IE56UJ0O69013763138156	3936	P.O. Box 317, 4439 Fermentum St.	0

Despues:

Result Grid

Filter Rows

Edit

Export/Import

Wrap Cell Content

id	id_plan_mensual	nickname	correo	episodios_vistos	contraseña	estado_animo	numero_de_cuenta_bancaria	tiempo_de_uso	direccion	estado_plan_mensual
1	0	AstraExplorer98	quantummind@gmail.com	77	RPP97VX	aburrido	IE56JAU069013763138156	3936	P.O. Box 317, 4439 Fermentum St.	1

usuario 6

Limit to 50000 rows

1

SELECT * FROM crunchy.usuario where id = 55;

Result Grid

Filter Rows

Edit

Export/Import

Wrap Cell Content

id	id_plan_mensual	nickname	correo	episodios_vistos	contraseña	estado_animo	numero_de_cuenta_bancaria	tiempo_de_uso	direccion	estado_plan_mensual
55	1	ThunderStriker61	echonavigator@gmail.com	61	NNU25ZD	triste	LV41GJIK1582216777544	3936	360-6748 Erin. Road	1

delimiter |

create procedure compartenPlanMenusal(ide_usuario int,ide_usuario_recibe
int)

begin

declare exit handler for sqlexception begin rollback;

end;

start transaction;

if((select count(*) from usuario where id=ide_usuario)) >0 then

if((select count(*) from usuario where id=ide_usuario_recibe))

>0 then

if(usuarioComparte(ide_usuario,ide_usuario_recibe) = 1 or
usuarioComparte(ide_usuario,ide_usuario_recibe) = 0) then

update usuario

set estado_plan_mensual = (select 1 from usuario where id = ide_usuario)

where id = ide_usuario; update usuario

set estado_plan_mensual =(select 1 from usuario where id
=ide_usuario_recibe) where

id = ide_usuario_recibe;

end if;

end if;

end if;

commit;

end ; |

delimiter

; call

comparte

nPlanMe

```

nusal(5,1)

; select *

from

usuario

where id

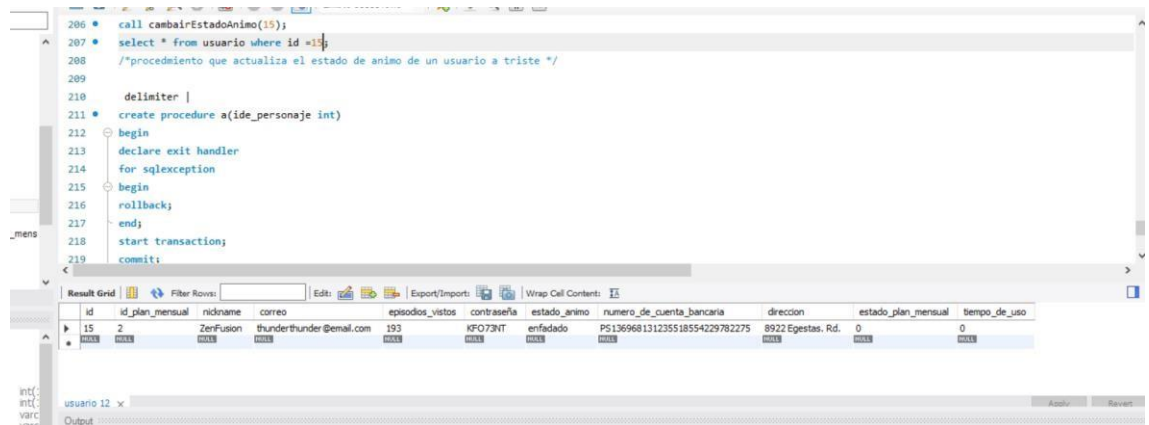
=1;

select * from usuario where id =5;
drop procedure compartenPlanMenuasal;

```

4. Procedimiento Libre

/*procedimiento que cambia el estado de animo de un usuario a feliz*/



```

delimiter | create procedure
cambiarEstadoAnimo(ide_usuario int) begin declare
exit handler for sqlexception begin rollback; end; start
transaction;

if( (select count(*)_from usuario where id=ide_usuario)) >0 then

if(((select estado_animo from usuario where id = ide_usuario )= 'feliz'
))>0 then

select 'ya es feliz';

else      update usuario

set estado_animo = 'feliz'

where id = ide_usuario;

end if;

```

```

end if;

commit;

end ; |

delimiter

;

drop procedure cambiarEstadoAnimo; call

cambiarEstadoAnimo(15);

select * from usuario where id =15;

```

The screenshot shows a SQL IDE with a script execution window. The script contains the following SQL commands:

```

214 for sqlexception
215 begin
216 rollback;
217 end;
218 start transaction;
219 commit;

```

The output window displays the results of the execution, including a table of user data and an action log.

id	id_plan_mensual	nickname	correo	episodios_vistos	contraseña	estado_animo	numero_de_cuenta_bancaria	direccion	estado_plan_mensual	tiempo_de_uso
15	2	ZenFusion	thunderthunder@email.com	193	KFO73NT	felz	PS136968131235518554229782275	8922 Egestas. Rd.	0	0

The action log shows the following actions:

#	Time	Action	Message	Duration / Fetch
31	13:34:03	alter table usuario drop tiempo_de_uso	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.047 sec
32	13:34:05	alter table usuario add tiempo_de_uso int default 0	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.063 sec
33	13:34:36	select * from usuario where id =12 LIMIT 0, 50000	1 row(s) returned	0.000 sec / 0.000 sec
34	13:34:46	select * from usuario where id =15 LIMIT 0, 50000	1 row(s) returned	0.000 sec / 0.000 sec
35	13:35:14	call cambiarEstadoAnimo(15)	1 row(s) affected	0.062 sec

5. Procedimiento Libre

/*procedimeinto que añade una fecha de creacion a los mangas*/

```
alter table mangas add fecha_creacion date default '2000-03-13' ;
```

The screenshot shows a SQL IDE with a script execution window. The script contains the following SQL commands:

```

219 if((select id from mangas where id=ide_manga)) >0 then
220 update mangas
221 set fecha_creacion = fecha_creacion_manga
222 where id = ide_manga;
223 else
224 select 'fecha no existe';
225 end if;
226 commit;
227 end ;
228 |
229 delimiter ;
230 call fechaCreacionManga(5,109);
231 SELECT * FROM crunchy.mangas;

```

The output window displays the results of the execution, including a table of manga data and an action log.

id	id_genero	emision	nombre	mangaka	paginas	fecha_creacion
1	77	0	Botchan (Manga)	Boy's Abyss	146	2000-03-13
2	94	0	Hakkyuu	Adolf (Nueva Edición)	28	2000-03-13
3	7	0	Ana, la de Avoriale	Children of the Whales	127	2000-03-13
4	5	1	Hunter_x_hunter	de	250	2000-03-13
5	109	1	Sota_Sota	Chitose_etc.	179	2000-03-13

The action log shows the following actions:

#	Time	Action	Message	Duration / Fetch
219	13:34:03	if((select id from mangas where id=ide_manga)) >0 then	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.047 sec
220	13:34:05	update mangas	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.063 sec
221	13:34:36	set fecha_creacion = fecha_creacion_manga	1 row(s) returned	0.000 sec / 0.000 sec
222	13:34:46	where id = ide_manga;	1 row(s) returned	0.000 sec / 0.000 sec
223	13:35:14	select 'fecha no existe';	1 row(s) affected	0.062 sec

```

delimiter | create procedure fechaCreacionManga(ide_manga
int,fecha_creacion_manga date) begin declare exit handler for sqlexception begin

```


rollback; end; start transaction; if((select id from mangas where id=ide_manga)) >0

then

update mangas

set fecha_creacion = fecha_creacion_manga

where id = ide_manga;

else

select 'fecha no existe';

end if;

commit; end ; | delimiter ; call

fechaCreacionManga(1,curdate());

SELECT * FROM crunchy.mangas;

The screenshot shows a MySQL IDE interface with a SQL script in the main editor and its execution results in the bottom panels.

SQL Script:

```
219 if( (select id from mangas where id=ide_manga)) >0 then
220     update mangas
221     set fecha_creacion = fecha_creacion_manga
222     where id = ide_manga;
223 else
224     select 'fecha no existe';
225 end if;
226 commit;
227 end ;
228 |
229 delimiter ;
230 * call fechaCreacionManga(1,curdate());
231 * SELECT * FROM crunchy.mangas;
232
```

Result Grid:

id	id_genero	emision	nombre	mangaka	paginas	fecha_creacion
1	77	0	Botchan_Manga	Boy's_Abyss	146	2024-05-14
2	94	0	Hakkyuu	Adolf_(Nueva_Edición)	28	2000-03-13
3	7	0	Ana_la_de_Avonlea	Children_of_the_Whales	127	2000-03-13
4	5	1	Hunter_x_Hunter	de	250	2000-03-13
5	109	1	Bola_Bola	Chitose_etc.	179	2000-03-13

Action Output:

#	Time	Action	Message	Duration / Fetch
7	16:07:50	alter table mangas add fecha_creacion date default '2000-03-13'	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.063 sec
8	16:09:45	create procedure fechaCreacionManga(ide_manga int, fecha_creacion_manga date) begin declare exit ha...	0 row(s) affected	0.000 sec
9	16:10:00	SELECT * FROM crunchy.mangas LIMIT 0, 50000	300 row(s) returned	0.000 sec / 0.000 sec
10	16:10:31	create procedure fechaCreacionManga(ide_manga int, fecha_creacion_manga date) begin declare exit ha...	Error Code: 1304. PROCEDURE fechaCreacionManga already exists	0.000 sec
11	16:10:34	call fechaCreacionManga(1,curdate())	1 row(s) affected	0.000 sec
12	16:10:36	SELECT * FROM crunchy.mangas LIMIT 0, 50000	300 row(s) returned	0.000 sec / 0.000 sec