

# Recuperación de Trigger Víctor Gabriel Aróstegui

## Primer ejercicio

```
Query 1 x SQL File 5* x asdasdasd x historico* x
Limit to 400 rows
1 • drop trigger if exists victor_validar;
2 • delimiter //
3 • create trigger victor_validar before insert on persona for each row
4 • begin
5 •   set new.codigo = upper(new.codigo);
6 •   if new.edad = 0 then set new.edad = null;
7 •   end if;
8 •   insert into nuevosDatos (codigo, cuando, tipo)
9 •   values (new.codigo, curdate(), 'I');
10 • end
11 • //
12 • delimiter ;
```

Introducimos nuevos datos.

```
1 • drop trigger if exists victor_validar;
2 • delimiter //
3 • create trigger victor_validar before insert on persona for each row
4 • begin
5 •   set new.codigo = upper(new.codigo);
6 •   if new.edad = 0 then set new.edad = null;
7 •   end if;
8 •   insert into nuevosDatos (codigo, cuando, tipo)
9 •   values (new.codigo, curdate(), 'I');
10 • end
11 • //
12 • delimiter ;
13
14 • select * from nuevosDatos;
15 • select * from persona;
16
17 • insert into persona values ('primero', 'Victor', 0);
18
```

Action Output				
		Time	Action	Message
✓	1	08:51:34	select * from persona LIMIT 0, 400	2 row(s) returned
✓	2	08:52:18	insert into persona values ('primero', 'Victor', 0)	1 row(s) affected

Observamos los cambios en persona.

```
11 //
12 delimiter ;
13
14 • select * from nuevosDatos;
15 • select * from persona;
16
17 • insert into persona values ('primero', 'Victor', 0);
18
```

Result Grid

#	codigo	nombre	edad
1	1	Juan	20
2	p	Pedro	0
3	PRIMERO	Victor	NULL
*	NULL	NULL	NULL

persona 5

Action Output

		Time	Action	Message
✓	1	08:51:34	select * from persona LIMIT 0, 400	2 row(s) returned
✓	2	08:52:18	insert into persona values ('primero', 'Victor', 0)	1 row(s) affected
✓	3	08:53:06	select * from nuevosDatos LIMIT 0, 400	1 row(s) returned
✓	4	08:53:09	select * from persona LIMIT 0, 400	3 row(s) returned

Observamos la inserción en nuevosDatos.

```
1 • drop trigger if exists victor_validar;
2 delimiter //
3 • create trigger victor_validar before insert on persona for each row
4 begin
5   set new.codigo = upper(new.codigo);
6   if new.edad = 0 then set new.edad = null;
7   end if;
8   insert into nuevosDatos (codigo, cuando, tipo)
9   values (new.codigo, curdate(), 'I');
10 end
11 //
12 delimiter ;
13
14 • select * from nuevosDatos;
15 • select * from persona;
16
17 • insert into persona values ('primero', 'Victor', 0);
18
```

Result Grid

#	codigo	cuando	tipo
1	PRIMERO	2019-02-20	I

nuevosDatos 6

Action Output

		Time	Action	Message
✓	2	08:52:18	insert into persona values ('primero', 'Victor', 0)	1 row(s) affected
✓	3	08:53:06	select * from nuevosDatos LIMIT 0, 400	1 row(s) returned
✓	4	08:53:09	select * from persona LIMIT 0, 400	3 row(s) returned
✓	5	08:54:29	select * from nuevosDatos LIMIT 0, 400	1 row(s) returned

## Segundo ejercicio.

```
Query 1 x SQL File 5* x
Limit to 400 rows
1 • drop trigger if exists victor_borrado;
2 • delimiter //
3 • create trigger victor_borrado before delete on persona for each row
4 • begin
5 •   insert into borrados values (old.codigo, old.nombre, old.edad, curdate());
6 • end
7 • //
8 • delimiter ;
9 •
10 • select * from borrados;
```

## Borramos en persona.

```
Query 1 x SQL File 5* x
Limit to 400 rows
1 • drop trigger if exists victor_borrado;
2 • delimiter //
3 • create trigger victor_borrado before delete on persona for each row
4 • begin
5 •   insert into borrados values (old.codigo, old.nombre, old.edad, curdate());
6 • end
7 • //
8 • delimiter ;
9 •
10 • select * from borrados;
11 • select * from persona;
12 •
13 • delete from persona where codigo = 'primero';
```

Result Grid

#	codigo	nombre	edad
1	1	Juan	20
2	p	Pedro	0
*	NULL	NULL	NULL

persona 3 x

Action Output

	Time	Action	Message	
✓	1	09:02:01	select * from persona LIMIT 0, 400	3 row(s) returned
✓	2	09:02:19	delete from persona where codigo = 'primero'	1 row(s) affected
✓	3	09:02:22	select * from persona LIMIT 0, 400	2 row(s) returned

Observamos en borrados.

The screenshot shows a SQL IDE with a query editor and a results pane. The query editor contains the following SQL code:

```
1 • drop trigger if exists victor_borrado;
2 • delimiter //
3 • create trigger victor_borrado before delete on persona for each row
4 • begin
5 •   insert into borrados values (old.codigo, old.nombre, old.edad, curdate());
6 • end
7 • //
8 • delimiter ;
9
10 • select * from borrados;
11 • select * from persona;
12
13 • delete from persona where codigo = 'primero';
```

The results pane shows the output of the queries. The first query, `select * from borrados;`, returned one row:

#	codigo	nombre	edad	fecha
1	PRIMERO	Victor	NULL	2019-02-20

The second query, `select * from persona;`, returned two rows. The third query, `delete from persona where codigo = 'primero';`, affected one row. The fourth query, `select * from borrados;`, returned one row.

The Action Output pane shows the following actions and messages:

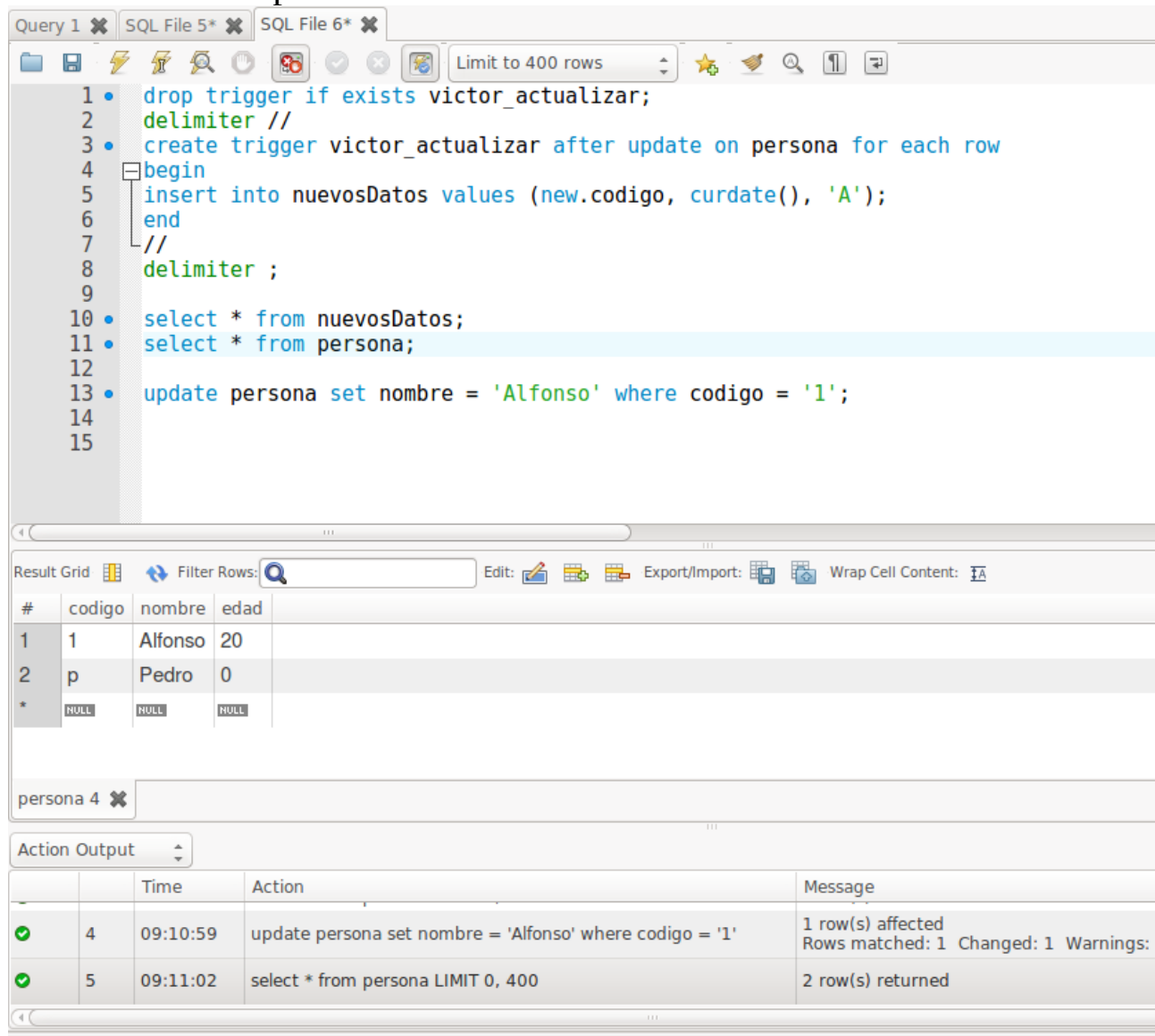
	Time	Action	Message
✓	1 09:02:01	select * from persona LIMIT 0, 400	3 row(s) returned
✓	2 09:02:19	delete from persona where codigo = 'primero'	1 row(s) affected
✓	3 09:02:22	select * from persona LIMIT 0, 400	2 row(s) returned
✓	4 09:03:06	select * from borrados LIMIT 0, 400	1 row(s) returned

Tercer ejercicio.

The screenshot shows a SQL IDE with a query editor. The query editor contains the following SQL code:

```
1 • drop trigger if exists victor_actualizar;
2 • delimiter //
3 • create trigger victor_actualizar after update on persona for each row
4 • begin
5 •   insert into nuevosDatos values (new.codigo, curdate(), 'A');
6 • end
7 • //
8 • delimiter ;
9
10 • select * from nuevosDatos;
```

Actualizamos en persona.



The screenshot shows a SQL IDE interface. The top part is the query editor with the following SQL code:

```
1 • drop trigger if exists victor_actualizar;
2 • delimiter //
3 • create trigger victor_actualizar after update on persona for each row
4 • begin
5 •   insert into nuevosDatos values (new.codigo, curdate(), 'A');
6 •   end
7 • //
8 • delimiter ;
9
10 • select * from nuevosDatos;
11 • select * from persona;
12
13 • update persona set nombre = 'Alfonso' where codigo = '1';
14
15
```

Below the query editor is the Result Grid, which displays the results of the queries. It has columns for row number, codigo, nombre, and edad. The first two rows are highlighted in alternating shades of gray.

#	codigo	nombre	edad
1	1	Alfonso	20
2	p	Pedro	0
*	NULL	NULL	NULL

Below the Result Grid is the Action Output section, which shows the execution log. It has columns for Time, Action, and Message. The first two rows are highlighted in alternating shades of gray.

	Time	Action	Message
✓	4	09:10:59	update persona set nombre = 'Alfonso' where codigo = '1'
✓	5	09:11:02	select * from persona LIMIT 0, 400

Observamos en nuevosDatos. Aparecen dos actualizaciones porque probé el comando varias veces.

Query 1 x SQL File 5\* x SQL File 6\* x

Limit to 400 rows

```

1 • drop trigger if exists victor_actualizar;
2 • delimiter //
3 • create trigger victor_actualizar after update on persona for each row
4 • begin
5 •   insert into nuevosDatos values (new.codigo, curdate(), 'A');
6 • end
7 • //
8 • delimiter ;
9
10 • select * from nuevosDatos;
11 • select * from persona;
12
13 • update persona set nombre = 'Alfonso' where codigo = '1';
14
15

```

Result Grid Filter Rows: Export: Wrap Cell Content:

#	codigo	cuando	tipo
1	PRIMERO	2019-02-20	I
2	1	2019-02-20	A
3	1	2019-02-20	A

nuevosDatos 5 x

Action Output

	Time	Action	Message	
✓	5	09:11:02	select * from persona LIMIT 0, 400	2 row(s) returned
✓	6	09:11:57	select * from nuevosDatos LIMIT 0, 400	3 row(s) returned

## Cuarto ejercicio.

Creamos la tabla nueva.

Query 1 x SQL File 5\* x SQL File 6\* x SQL File 7\* x

Limit to 400 rows

```

1 • select * from persona;
2 • drop table historico_personas;
3 • create table historico_personas (
4 •   codigo varchar(10) primary key,
5 •   nombre varchar(50),
6 •   edad decimal(3),
7 •   fecha date,
8 •   tiempo time
9 • );
10

```



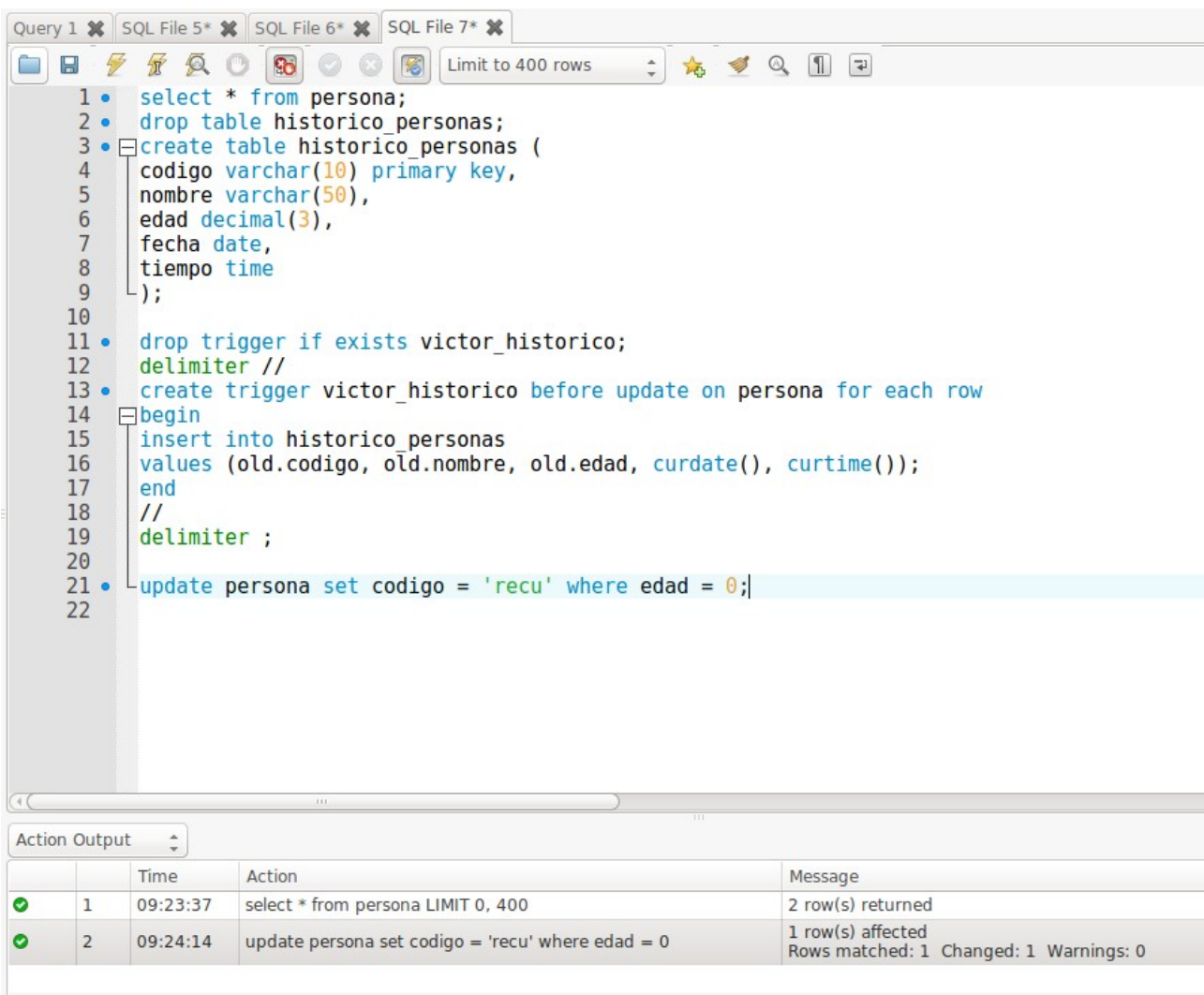
## Trigger del ejercicio.

```
drop trigger if exists victor_historico;
delimiter //
create trigger victor_historico before update on persona for each row
begin
insert into historico_personas
values (old.codigo, old.nombre, old.edad, curdate(), curtime());
end
//
delimiter ;
```

Es el mismo trigger pero con el precedes; aún estando la sintaxis bien, no se crea el disparador.

```
drop trigger if exists victor_historico;
delimiter //
create trigger victor_historico before update on persona for each row precedes victor_actualizar
begin
insert into historico_personas
values (old.codigo, old.nombre, old.edad, curdate(), curtime());
end
//
delimiter ;
```

## Actualización.



The screenshot shows a SQL IDE with a script editor and an action output window. The script editor contains the following SQL code:

```
1 • select * from persona;
2 • drop table historico_personas;
3 • create table historico_personas (
4     codigo varchar(10) primary key,
5     nombre varchar(50),
6     edad decimal(3),
7     fecha date,
8     tiempo time
9 );
10
11 • drop trigger if exists victor_historico;
12 • delimiter //
13 • create trigger victor_historico before update on persona for each row
14 • begin
15 •     insert into historico_personas
16 •     values (old.codigo, old.nombre, old.edad, curdate(), curtime());
17 • end
18 • //
19 • delimiter ;
20
21 • update persona set codigo = 'recu' where edad = 0;
22
```

The action output window shows the following results:

	Time	Action	Message
✓	1 09:23:37	select * from persona LIMIT 0, 400	2 row(s) returned
✓	2 09:24:14	update persona set codigo = 'recu' where edad = 0	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0

Observamos el cambio y la nueva tabla.

The screenshot shows a SQL IDE interface with the following components:

- Query Editor:** Contains the following SQL code:

```
1 • select * from persona;
2 • select * from historico personas;
3 • drop table historico personas;
4 • create table historico_personas (
5     codigo varchar(10) primary key,
6     nombre varchar(50),
7     edad decimal(3),
8     fecha date,
9     tiempo time
10 );
11
12 • drop trigger if exists victor_historico;
13 delimiter //
14 • create trigger victor_historico before update on persona for each row
15 begin
16     insert into historico personas
17     values (old.codigo, old.nombre, old.edad, curdate(), curtime());
18 end
```
- Result Grid:** Displays the results of the last query. It has columns: #, codigo, nombre, edad, fecha, tiempo. The first row shows: 1, p, Pedro, 0, 2019-02-20, 09:24:14. Below it, a row of NULL values is shown.
- historico\_personas 4:** A tab or view label at the bottom of the result grid.
- Action Output:** A log showing the execution of the queries:

		Time	Action	Message	
✖	4	09:26:01	selct * from historico_personas	Error Code: 1064. You have an error in your SQL synta...	0,
✔	5	09:26:08	select * from historico_personas LIMIT 0, 400	1 row(s) returned	0,



Query 1
SQL File 5\*
SQL File 6\*
SQL File 7\*

Limit to 400 rows

```

6      edad decimal(3),
7      fecha date,
8      tiempo time
9  );
10
11 • drop trigger if exists victor_historico;
12 delimiter //
13 • create trigger victor_historico before update on persona for each row
14 begin
15 insert into historico_personas
16 values (old.codigo, old.nombre, old.edad, curdate(), curtime());
17 end
18 //
19 delimiter ;
20
21 • update persona set codigo = 'recu' where edad = 0;
22

```

Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:

#	codigo	nombre	edad
1	1	Alfonso	20
2	recu	Pedro	0
*	NULL	NULL	NULL

persona 3

Action Output

		Time	Action	Message
✓	1	09:23:37	select * from persona LIMIT 0, 400	2 row(s) returned
✓	2	09:24:14	update persona set codigo = 'recu' where edad = 0	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
✓	3	09:25:17	select * from persona LIMIT 0, 400	2 row(s) returned