

Triggers et Events

```
DELIMITER $$
CREATE TRIGGER payments_after_insert
AFTER INSERT ON payments
FOR EACH ROW
BEGIN
    UPDATE invoices
    SET payment_total = payment_total + NEW.amount
    WHERE invoice_id = NEW.invoice_id;
END$$
DELIMITER ;

insert into payments
(payment_id,
client_id,
invoice_id,
date,
amount,
payment_method
)
VALUES
(default, 5, 5, NOW(), 10,1);

SHOW TRIGGERS;
DROP TRIGGER payments_after_insert;
```

The screenshot shows a database management tool interface. On the left, the 'Navigator' pane displays the database schema, including tables like 'clients', 'invoices', 'payments', and 'invoices_archived'. The 'Triggers' section is expanded, showing the trigger 'payments_after_insert'. The main pane displays the SQL query for creating the trigger and inserting data into the 'payments' table. The 'Result Grid' at the bottom shows the execution results, including the trigger definition and the insertion of a new payment record.

Trigger	Event	Table	Statement	Timing	Created	sql_mode	Definer	character_set_client	collation_connection	Database Collation
payments_after_insert	INSERT	payments	BEGIN UPDATE invoices SET pa...	AFTER	2022-05-06 18:59:54.56	STRICT_TRANS_TABLES,NO_ENGINE...	root@localhost	utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0900_ai_ci

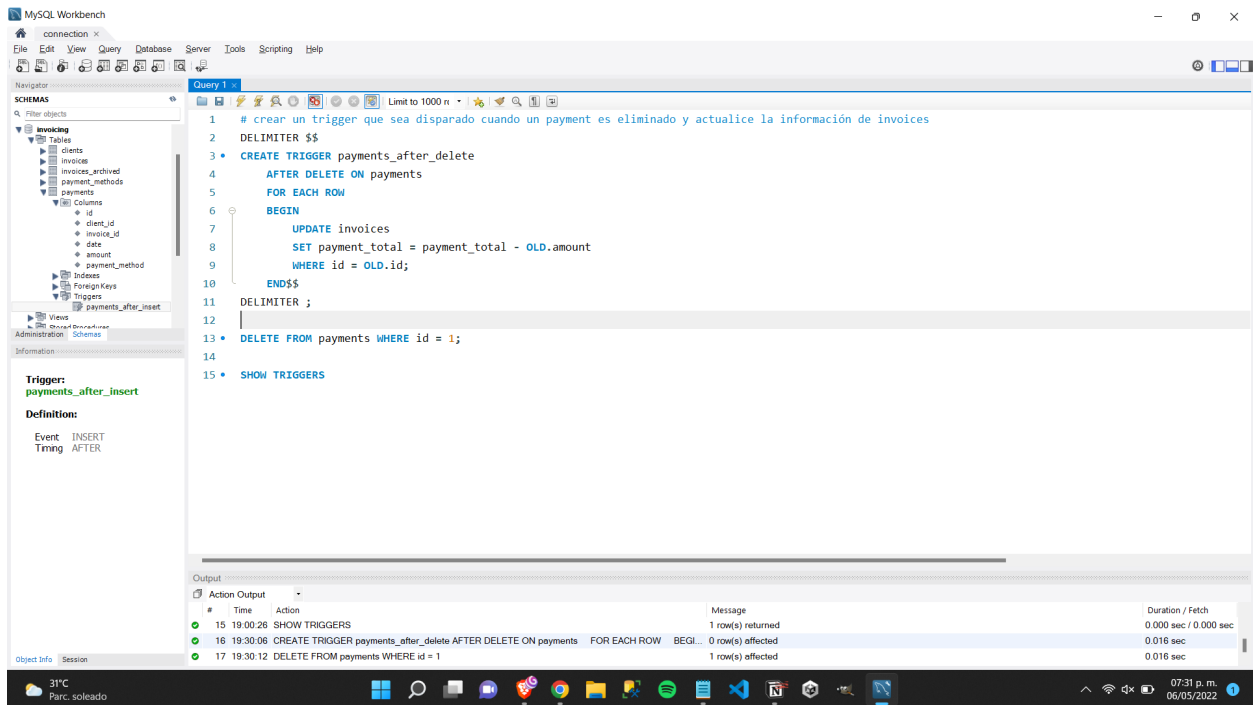
#	Time	Action	Message	Duration / Fetch
13	18:59:54	CREATE TRIGGER payments_after_insert AFTER INSERT ON payments FOR EACH ROW BEGIN...	0 row(s) affected	0.000 sec
14	19:00:08	insert into payments (id, client_id, invoice_id, date, amount, payment_method) VALUES (default, 5, 5, ...	1 row(s) affected, 1 warning(s): 1292 Incorrect date value: '2022-05-06 19:00:08' for column 'date' at ro...	0.016 sec
15	19:00:26	SHOW TRIGGERS	1 row(s) returned	0.000 sec / 0.000 sec

- Créer un trigger que sea disparado cuando un payment es eliminado y actualice la información de invoices

```
DELIMITER $$
CREATE TRIGGER payments_after_delete
AFTER DELETE ON payments
FOR EACH ROW
BEGIN
    UPDATE invoices
    SET payment_total = payment_total - OLD.amount
    WHERE id = OLD.id;
END$$
```

```
DELIMITER ;

DELETE FROM payments WHERE id = 1;
```



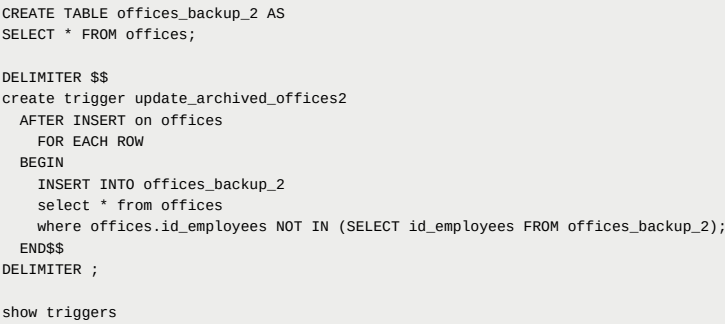
- Invoicing
 - Que permita respaldar los nuevos datos en la tabla respaldo después de
 - Hr
 - store
- CUSTOMERS -> CUSTOMERS_ARCHIVED
- ORDER -> ORDER_ARCHIVED
- ORDER_ITEMS -> ORDER_ITEMS_ARCHIVED



```
CREATE TABLE offices_backup AS
SELECT * FROM offices;

DELIMITER $$
create trigger update_archived_offices
AFTER INSERT on offices
FOR EACH ROW
BEGIN
    INSERT INTO offices_backup
    select * from offices
    where offices.id_office NOT IN (SELECT id_office FROM offices_backup);
END$$
DELIMITER ;

show triggers
```



MySQL Workbench

connection x

File Edit View Query Database Server Tools Scripting Help

Navigator

Schemas

Filter objects

classimodes

hr

employees

Columns

id_employees

first_name

last_name

job_title

salary

reports_to

office_id

Indexes

Foreign Keys

Triggers

offices

Views

Stored Procedures

Functions

Administration Schemas

Information

Schema: store

Query 1

```

1 CREATE TABLE offices_backup_2 AS
2 SELECT * FROM offices;
3
4 DELIMITER $$
5 create trigger update_archived_offices2
6 AFTER INSERT on offices
7 FOR EACH ROW
8 BEGIN
9     INSERT INTO offices_backup_2
10     select * from offices
11     where offices.id_employees NOT IN (SELECT id_employees FROM offices_backup_2);
12 END$$
13 DELIMITER ;
14
15 show triggers

```

Result Grid

Trigger	Event	Table	Statement	Timing	Created	sql_mode	Definer	character_set_client	collation_connection	Database Collation
update_archived_offices	INSERT	offices	BEGIN INSERT INTO offices_backup_2 AS SELECT * FROM offices	AFTER	2022-05-06 20:41:00.64	STRICT_TRANS_TABLES,NO_ENGINE...	root@localhost	utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0900_ai_ci
update_archived_offices2	INSERT	offices	BEGIN INSERT INTO offices_backup_2 AS SELECT * FROM offices	AFTER	2022-05-06 20:45:30.89	STRICT_TRANS_TABLES,NO_ENGINE...	root@localhost	utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0900_ai_ci

Result 5 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
26	20:45:30	CREATE TABLE offices_backup_2 AS SELECT * FROM offices	10 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.063 sec
27	20:45:30	create trigger update_archived_offices2 AFTER INSERT on offices FOR EACH ROW BEGIN IN...	0 row(s) affected	0.015 sec
28	20:45:36	show triggers	2 row(s) returned	0.016 sec / 0.000 sec

Object Info Session

31°C Parc. soleado

08:46 p. m. 06/05/2022

```

CREATE TABLE inventory_backup AS
SELECT * FROM products;

DELIMITER $$
create trigger update_inventory1
AFTER INSERT on products
FOR EACH ROW
BEGIN
    INSERT INTO inventory_backup
    select * from products
    where products.id NOT IN (SELECT id FROM inventory_backup);
END$$
DELIMITER ;

show triggers

```

MySQL Workbench

connection x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHMAS

Filter objects

classimodels

hr

inventory

products

Columns:

id

name

quantity_in_stock

unit_price

Indexes

ForeignKeys

Triggers

Views

Stored Procedures

Functions

invoicing

columnsone

sakila

Administration

Schemas

Information

Tables: products

Columns:

id int AI PK

name varchar(45)

quantity_in_stock int

unit_price float

Query 1

Find

Limit to 1000 n

Done

```

1 • CREATE TABLE inventory_backup AS
2 SELECT * FROM products;
3
4 DELIMITER $$
5 • create trigger update_inventory1
6 AFTER INSERT on products
7 FOR EACH ROW
8 BEGIN
9 INSERT INTO inventory_backup
10 select * from products
11 where products.id NOT IN (SELECT id FROM inventory_backup);
12 END$$
13 DELIMITER ;
14
15 • show triggers

```

Result Grid

Trigger	Event	Table	Statement	Timing	Created	sql_mode	Definer	character_set_client	collation_connection	Database Collation
update_inventory1	INSERT	products	INSERT INTO inventory_ba...	AFTER	2022-05-06 20:49:11.79	STRICT_TRANS_TABLES,NO_ENGINE...	root@localhost	utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0900_ai_ci

Output

Action Output

#	Time	Action	Message	Duration / Fetch
34	20:49:11	CREATE TABLE inventory_backup AS SELECT * FROM products	Error Code: 1050. Table 'inventory_backup' already exists	0.000 sec
35	20:49:11	create trigger update_inventory1 AFTER INSERT on products FOR EACH ROW BEGIN INSERT...	0 row(s) effected	0.016 sec
36	20:49:14	show triggers	1 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

31°C Parc. soleado

08:49 p. m. 06/05/2022

```

CREATE TABLE invoicing_backup AS
SELECT * FROM clients;

DELIMITER $$
create trigger update_invoicing1
AFTER INSERT on clients
FOR EACH ROW
BEGIN
INSERT INTO invoicing_backup
select * from clients
where clients.id NOT IN (SELECT id FROM invoicing_backup);
END$$
DELIMITER ;

show triggers

```

MySQL Workbench

connection x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHMAS

Filter objects

classimodes
hr
inventory
invoicing
Tables
clients
Columns
id
name
address
city
state
phone
Indexes
ForeignKeys
Triggers
invoices
invoices_archived
payment_methods
payment

Administration Schemas

Information

Tables: products

Columns:

id int AI PK
name varchar(45)
quantity_in_stock int
unit_price float

Query 1

Find offices

```

3
4 DELIMITER $$
5 • create trigger update_invoicing1
6   AFTER INSERT on clients
7   FOR EACH ROW
8   BEGIN
9     INSERT INTO invoicing_backup
10    select * from clients
11    where clients.id NOT IN (SELECT id FROM invoicing_backup);
12  END$$
13 DELIMITER ;
14
15 • show triggers
16
17

```

Result Grid

Trigger	Event	Table	Statement	Timing	Created	sql_mode	Definer	character_set_client	collation_connection	Database Collation
update_invoicing1	INSERT	clients	BEGIN INSERT INTO invoicing_bac...	AFTER	2022-05-06 20:50:57.94	STRICT_TRANS_TABLES,NO_ENGINE...	root@localhost	utf8mb4	utf8mb4_0900_ai_o	utf8mb4_0900_ai_o
payments_after_insert	INSERT	payments	BEGIN UPDATE invoices SET pa...	AFTER	2022-05-06 18:59:54.56	STRICT_TRANS_TABLES,NO_ENGINE...	root@localhost	utf8mb4	utf8mb4_0900_ai_o	utf8mb4_0900_ai_o
payments_after_delete	DELETE	payments	BEGIN UPDATE invoices SET pa...	AFTER	2022-05-06 19:30:06.38	STRICT_TRANS_TABLES,NO_ENGINE...	root@localhost	utf8mb4	utf8mb4_0900_ai_o	utf8mb4_0900_ai_o

Output

Action Output

#	Time	Action	Message	Duration / Fetch
38	20:50:57	CREATE TABLE invoicing_backup AS SELECT * FROM clients	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.062 sec
39	20:50:57	create trigger update_invoicing1 AFTER INSERT on clients FOR EACH ROW BEGIN INSERT IN...	0 row(s) affected	0.015 sec
40	20:51:04	show triggers	3 row(s) returned	0.015 sec / 0.000 sec

Object Info Session

31°C Parc, soleado

08:51 p. m. 06/05/2022

```

CREATE TABLE invoicing_backup2 AS
SELECT * FROM invoices;

DELIMITER $$
create trigger update_invoicing2
  AFTER INSERT on invoices
  FOR EACH ROW
  BEGIN
    INSERT INTO invoicing_backup2
    select * from invoices
    where invoices.id NOT IN (SELECT id FROM invoicing_backup2);
  END$$
DELIMITER ;

show triggers

```

MySQL Workbench

connection x

File Edit View Query Database Server Tools Scripting Help

Navigator

Schemas

Filter objects

classimodes

hr

inventory

invoicing

Tables

clients

invoices

Columns

id int PK

number int

client_id varchar(20)

invoice_total float

payment_total float

invoice_date date

due_date date

payment_date date

Indexes

ForeignKeys

Triggers

Administration

Schemas

Information

Tables: invoicing

Columns:

id int PK

number int

client_id varchar(20)

invoice_total float

payment_total float

invoice_date date

due_date date

payment_date date

Query 1

Find

Limit to 1000 n

Done

```

5 • create trigger update_invoicing2
6   AFTER INSERT on invoices
7   FOR EACH ROW
8   BEGIN
9     INSERT INTO invoicing_backup2
10    select * from invoices
11    where invoices.id NOT IN (SELECT id FROM invoicing_backup2);
12  END$$
13 DELIMITER ;
14
15 • show triggers
16
17
18

```

Result Grid

Trigger	Event	Table	Statement	Timing	Created	sql_mode	Definer	character_set_client	collation_connection	Database Collation
update_invoicing1	INSERT	clients	BEGIN INSERT INTO invoicing_bac...	AFTER	2022-05-06 20:50:57.94	STRICT_TRANS_TABLES,NO_ENGINE...	root@localhost	utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0900_ai...
update_invoicing2	INSERT	invoices	BEGIN INSERT INTO invoicing_bac...	AFTER	2022-05-06 20:52:00.27	STRICT_TRANS_TABLES,NO_ENGINE...	root@localhost	utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0900_ai...
payments_after_insert	INSERT	payments	BEGIN UPDATE invoices SET pa...	AFTER	2022-05-06 19:59:54.56	STRICT_TRANS_TABLES,NO_ENGINE...	root@localhost	utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0900_ai...
payments_after_delete	DELETE	payments	BEGIN UPDATE invoices SET pa...	AFTER	2022-05-06 19:30:06.38	STRICT_TRANS_TABLES,NO_ENGINE...	root@localhost	utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0900_ai...

Result 9 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
41	20:52:00	CREATE TABLE invoicing_backup2 AS SELECT * FROM invoices	16 row(s) affected Records: 16 Duplicates: 0 Warnings: 0	0.047 sec
42	20:52:00	create trigger update_invoicing2 AFTER INSERT on invoices FOR EACH ROW BEGIN INSERT L...	0 row(s) affected	0.015 sec
43	20:52:05	show triggers	4 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

31°C Parc. soleado

08:52 p. m. 06/05/2022

```

CREATE TABLE invoicing_backup3 AS
SELECT * FROM invoices;

DELIMITER $$
create trigger update_invoicing3
  AFTER INSERT on invoices_archived
  FOR EACH ROW
  BEGIN
    INSERT INTO invoicing_backup3
    select * from invoices_archived
    where invoices_archived.id NOT IN (SELECT id FROM invoicing_backup3);
  END$$
DELIMITER ;

show triggers

```

MySQL Workbench

connection x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHMAS

Filter objects

classicmodels
hr
inventory
invoicing
Tables
clients
invoices
invoices_archived
payment_methods
payments
Views
Stored Procedures
Functions
pokemons
salle
store
sys
world

Administration Schemas

Information

Tables: invoicing

Columns:

id	number	int PK
client_id	varchar(20)	int
invoice_total	float	
payment_total	float	
invoice_date	date	
due_date	date	
payment_date	date	

Query 1

Find offices

```

3
4 DELIMITER $$
5 • create trigger update_invoicing3
6   AFTER INSERT on invoices_archived
7   FOR EACH ROW
8   BEGIN
9     INSERT INTO invoicing_backup3
10    select * from invoices_archived
11    where invoices_archived.id NOT IN (SELECT id FROM invoicing_backup3);
12  ENDS$
13 DELIMITER ;
14
15 • show triggers
16
17

```

Result Grid

Trigger	Event	Table	Statement	Timing	Created	sql_mode	Definer	character_set_client	collation_connection	Database
update_invoicing1	INSERT	clients	BEGIN INSERT INTO invoicing_bac...	AFTER	2022-05-06 20:50:57.94	STRICT_TRANS_TABLES,NO_ENGINE...	root@localhost	utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0
update_invoicing2	INSERT	invoices	BEGIN INSERT INTO invoicing_bac...	AFTER	2022-05-06 20:52:00.27	STRICT_TRANS_TABLES,NO_ENGINE...	root@localhost	utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0
update_invoicing3	INSERT	invoices_archived	BEGIN INSERT INTO invoicing_bac...	AFTER	2022-05-06 20:53:20.54	STRICT_TRANS_TABLES,NO_ENGINE...	root@localhost	utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0
payments_after_insert	INSERT	payments	BEGIN UPDATE invoices SET pa...	AFTER	2022-05-06 18:59:54.56	STRICT_TRANS_TABLES,NO_ENGINE...	root@localhost	utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0
payments_after_delete	DELETE	payments	BEGIN UPDATE invoices SET pa...	AFTER	2022-05-06 19:30:06.38	STRICT_TRANS_TABLES,NO_ENGINE...	root@localhost	utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0

Result 10 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
44	20:53:20	CREATE TABLE invoicing_backup3 AS SELECT * FROM invoices	16 row(s) affected Records: 16 Duplicates: 0 Warnings: 0	0.078 sec
45	20:53:20	create trigger update_invoicing3 AFTER INSERT on invoices_archived FOR EACH ROW BEGIN ...	0 row(s) affected	0.016 sec
46	20:53:23	show triggers	5 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

29°C Muy soleado

08:53 p. m. 06/05/2022

```

CREATE TABLE invoicing_backup4 AS
SELECT * FROM invoices;

DELIMITER $$
create trigger update_invoicing4
  AFTER INSERT on invoices_archived
  FOR EACH ROW
  BEGIN
    INSERT INTO invoicing_backup4
    select * from invoices_archived
    where invoices_archived.id NOT IN (SELECT id FROM invoicing_backup4);
  ENDS$
DELIMITER ;

show triggers

```


MySQL Workbench

connection x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHMAS

Filter objects

classicmodels
hr
inventory
invoicing
Tables
clients
invoices
invoices_archived
payment_methods
payments
Views
Stored Procedures
Functions
pokemons
salle
store
sys
world

Administration Schemas

Information

Tables: invoicing

Columns:

id	number	int PK
client_id	varchar(20)	int
invoice_total	float	float
payment_total	float	float
invoice_date	date	date
due_date	date	date
payment_date	date	date

Query 1

Find

Limit to 1000 n

Done

```

5 • create trigger update_invoicing4
6   AFTER INSERT on invoices_archived
7   FOR EACH ROW
8   BEGIN
9     INSERT INTO invoicing_backup4
10    select * from invoices_archived
11    where invoices_archived.id NOT IN (SELECT id FROM invoicing_backup4);
12  END$$
13 DELIMITER ;
14
15 • show triggers
16
17
18

```

Result Grid

Trigger	Event	Table	Statement	Timing	Created	sql_mode	Definer	character_set_client	collation_connection	Database
update_invoicing1	INSERT	clients	BEGIN INSERT INTO invoicing_bac...	AFTER	2022-05-06 20:50:57.94	STRICT_TRANS_TABLES,NO_ENGINE...	root@localhost	utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0
update_invoicing2	INSERT	invoices	BEGIN INSERT INTO invoicing_bac...	AFTER	2022-05-06 20:52:00.27	STRICT_TRANS_TABLES,NO_ENGINE...	root@localhost	utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0
update_invoicing3	INSERT	invoices_archived	BEGIN INSERT INTO invoicing_bac...	AFTER	2022-05-06 20:53:20.54	STRICT_TRANS_TABLES,NO_ENGINE...	root@localhost	utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0
update_invoicing4	INSERT	invoices_archived	BEGIN INSERT INTO invoicing_bac...	AFTER	2022-05-06 20:54:41.09	STRICT_TRANS_TABLES,NO_ENGINE...	root@localhost	utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0
payments_after_insert	INSERT	payments	BEGIN UPDATE invoices SET pa...	AFTER	2022-05-06 18:59:54.56	STRICT_TRANS_TABLES,NO_ENGINE...	root@localhost	utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0
payments_after_delete	DELETE	payments	BEGIN UPDATE invoices SET pa...	AFTER	2022-05-06 19:30:06.38	STRICT_TRANS_TABLES,NO_ENGINE...	root@localhost	utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0

Result 11 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
47	20:54:41	CREATE TABLE invoicing_backup4 AS SELECT * FROM invoices	16 row(s) affected Records: 16 Duplicates: 0 Warnings: 0	0.047 sec
48	20:54:41	create trigger update_invoicing4 AFTER INSERT on invoices_archived FOR EACH ROW BEGIN	0 row(s) affected	0.000 sec
49	20:54:43	show triggers	6 row(s) returned	0.016 sec / 0.000 sec

Object Info Session

29°C Muy soleado

08:54 p. m. 06/05/2022

```

CREATE TABLE invoicing_backup5 AS
SELECT * FROM invoices;

DELIMITER $$
create trigger update_invoicing5
  AFTER INSERT on payment_methods
  FOR EACH ROW
  BEGIN
    INSERT INTO invoicing_backup5
    select * from payment_methods
    where payment_methods.id NOT IN (SELECT id FROM invoicing_backup5);
  END$$
DELIMITER ;

show triggers

```

MySQL Workbench

connection x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHMAS

Filter objects

classicmodels
hr
inventory
invoicing
Tables
clients
invoices
invoices_archived
payment_methods
payments
Views
Stored Procedures
Functions
pokemons
salle
store
sys
world

Administration Schemas

Information

Tables: invoicing

Columns:

id int PK
number int
client_id varchar(20)
invoice_total float
invoice_date date
due_date date
payment_date date

Query 1

Find

Limit to 1000 rows

3
4 DELIMITER \$\$
5 • create trigger update_invoicing5
6 AFTER INSERT on payment_methods
7 FOR EACH ROW
8 BEGIN
9 INSERT INTO invoicing_backup5
10 select * from payment_methods
11 where payment_methods.id NOT IN (SELECT id FROM invoicing_backup5);
12 END\$\$
13 DELIMITER ;
14
15 • show triggers
16
17

Result Grid

Trigger	Event	Table	Statement	Timing	Created	sql_mode	Definer	character_set_client	collation_connection	Datat
update_invoicing1	INSERT	clients	BEGIN INSERT INTO invoicing_bac...	AFTER	2022-05-06 20:50:57.94	STRICT_TRANS, TABLES, NO_ENGINE...	root@localhost	utf8mb4	utf8mb4_0900_ai_ci	utf8m
update_invoicing2	INSERT	invoices	BEGIN INSERT INTO invoicing_bac...	AFTER	2022-05-06 20:52:00.27	STRICT_TRANS, TABLES, NO_ENGINE...	root@localhost	utf8mb4	utf8mb4_0900_ai_ci	utf8m
update_invoicing3	INSERT	invoices_archived	BEGIN INSERT INTO invoicing_bac...	AFTER	2022-05-06 20:53:20.54	STRICT_TRANS, TABLES, NO_ENGINE...	root@localhost	utf8mb4	utf8mb4_0900_ai_ci	utf8m
update_invoicing4	INSERT	invoices_archived	BEGIN INSERT INTO invoicing_bac...	AFTER	2022-05-06 20:54:41.09	STRICT_TRANS, TABLES, NO_ENGINE...	root@localhost	utf8mb4	utf8mb4_0900_ai_ci	utf8m
update_invoicing5	INSERT	payment_methods	BEGIN INSERT INTO invoicing_bac...	AFTER	2022-05-06 20:55:39.35	STRICT_TRANS, TABLES, NO_ENGINE...	root@localhost	utf8mb4	utf8mb4_0900_ai_ci	utf8m
payments_after_insert	INSERT	payments	BEGIN UPDATE invoices SET pa...	AFTER	2022-05-06 18:59:54.56	STRICT_TRANS, TABLES, NO_ENGINE...	root@localhost	utf8mb4	utf8mb4_0900_ai_ci	utf8m
payments after delete	DELETE	payments	BEGIN UPDATE invoices SET pa...	AFTER	2022-05-06 19:30:06.38	STRICT_TRANS, TABLES, NO_ENGINE...	root@localhost	utf8mb4	utf8mb4_0900 ai ci	utf8m

Result 12 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
50	20:55:39	CREATE TABLE invoicing_backup6 AS SELECT * FROM invoices	16 row(s) affected Records: 16 Duplicates: 0 Warnings: 0	0.031 sec
51	20:55:39	create trigger update_invoicing5 AFTER INSERT on payment_methods FOR EACH ROW BEGIN ...	0 row(s) affected	0.016 sec
52	20:55:41	show triggers	7 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

29°C
Muy soleado

08:55 p. m.
06/05/2022

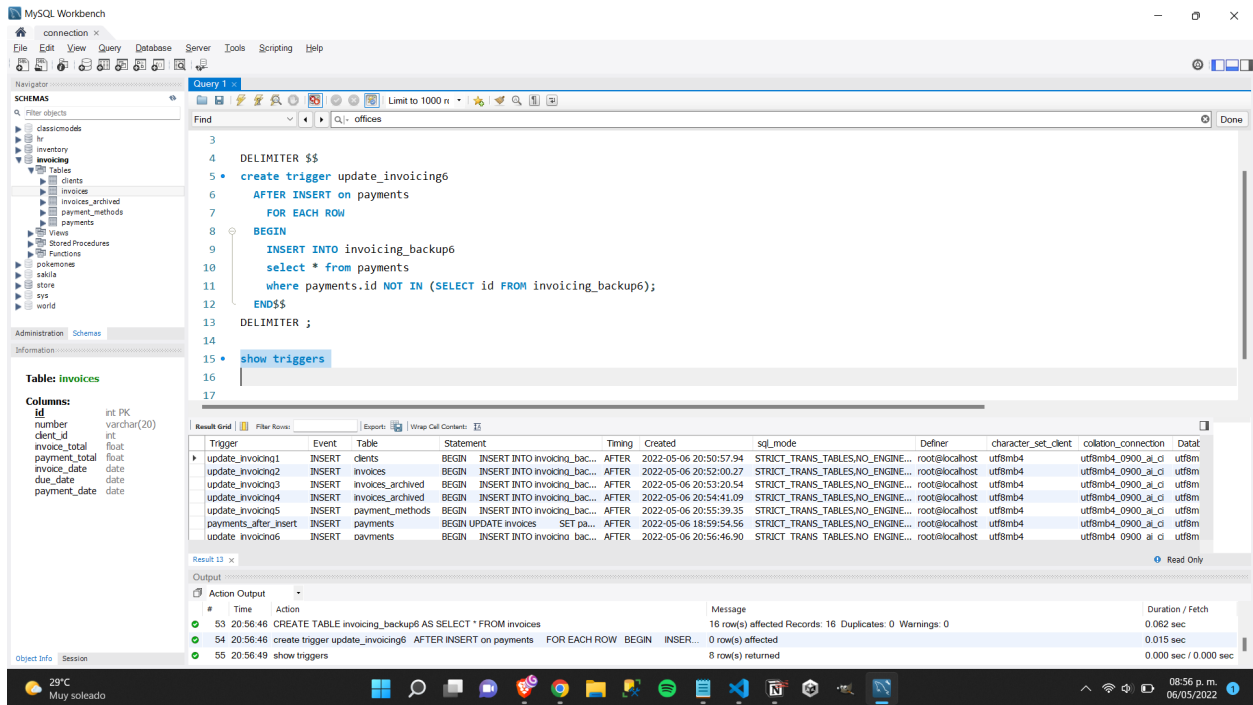
```

CREATE TABLE invoicing_backup6 AS
SELECT * FROM invoices;

DELIMITER $$
create trigger update_invoicing6
  AFTER INSERT on payments
  FOR EACH ROW
  BEGIN
    INSERT INTO invoicing_backup6
    select * from payments
    where payments.id NOT IN (SELECT id FROM invoicing_backup6);
  END$$
DELIMITER ;

show triggers

```



- Crear un trigger que sea disparado cuando un payment es eliminado y actualice la información de invoices

```

create table if not exists sql_hr.employees_backup as select * from sql_hr.employees;
delimiter $$
DROP TRIGGER IF EXISTS sql_hr.employees _after_insert$$
create trigger sql_hr.employees _after_insert
after insert on sql_hr.employees
for each row
begin
insert into sql_hr.employees_backup (select * from sql_hr.employees where employee_id not in (select * from sql_hr.employees_backup));
end $$
delimiter ;
create table if not exists sql_hr.offices_backup as select * from sql_hr.offices;
delimiter $$
DROP TRIGGER IF EXISTS sql_hr.offices _after_insert$$
create trigger sql_hr.offices _after_insert
after insert on sql_hr.offices
for each row
begin
insert into sql_hr.offices_backup (select * from sql_hr.offices where office_id not in (select * from sql_hr.offices_backup));
end $$
delimiter ;
create table if not exists sql_inventory.products_backup as select * from sql_inventory.products;
delimiter $$
DROP TRIGGER IF EXISTS sql_inventory.products _after_insert$$
create trigger sql_inventory.products _after_insert
after insert on sql_inventory.products
for each row
begin
insert into sql_inventory.products_backup (select * from sql_inventory.products where product_id not in (select * from sql_inventory.products_backup));
end $$
delimiter ;
create table if not exists sql_invoicing.clients_backup as select * from sql_invoicing.clients;
delimiter $$
DROP TRIGGER IF EXISTS sql_invoicing.clients _after_insert$$
create trigger sql_invoicing.clients _after_insert

```

```

        after insert on sql_invoicing.clients
        for each row
        begin
            insert into sql_invoicing.clients_backup (select * from sql_invoicing.clients where client_id not in (select * from
        end $$
delimiter ;
create table if not exists sql_invoicing.invoices_backup as select * from sql_invoicing.invoices;
delimiter $$
DROP TRIGGER IF EXISTS sql_invoicing.invoices_after_insert$$
create trigger sql_invoicing.invoices_after_insert
        after insert on sql_invoicing.invoices
        for each row
        begin
            insert into sql_invoicing.invoices_backup (select * from sql_invoicing.invoices where invoice_id not in (select * fr
        end $$
delimiter ;
create table if not exists sql_invoicing.payment_methods_backup as select * from sql_invoicing.payment_methods;
delimiter $$
DROP TRIGGER IF EXISTS sql_invoicing.payment_methods_after_insert$$
create trigger sql_invoicing.payment_methods_after_insert
        after insert on sql_invoicing.payment_methods
        for each row
        begin
            insert into sql_invoicing.payment_methods_backup (select * from sql_invoicing.payment_methods where payment_method_id
        end $$
delimiter ;
create table if not exists sql_invoicing.payments_backup as select * from sql_invoicing.payments;
delimiter $$
DROP TRIGGER IF EXISTS sql_invoicing.payments_after_insert$$
create trigger sql_invoicing.payments_after_insert
        after insert on sql_invoicing.payments
        for each row
        begin
            insert into sql_invoicing.payments_backup (select * from sql_invoicing.payments where payment_id not in (select * fr
        end $$
delimiter ;
create table if not exists sql_store.customers_backup as select * from sql_store.customers;
delimiter $$
DROP TRIGGER IF EXISTS sql_store.customers_after_insert$$
create trigger sql_store.customers_after_insert
        after insert on sql_store.customers
        for each row
        begin
            insert into sql_store.customers_backup (select * from sql_store.customers where customer_id not in (select * from s
        end $$
delimiter ;
create table if not exists sql_store.order_item_notes_backup as select * from sql_store.order_item_notes;
delimiter $$
DROP TRIGGER IF EXISTS sql_store.order_item_notes_after_insert$$
create trigger sql_store.order_item_notes_after_insert
        after insert on sql_store.order_item_notes
        for each row
        begin
            insert into sql_store.order_item_notes_backup (select * from sql_store.order_item_notes where order_item_note_id not
        end $$
delimiter ;
create table if not exists sql_store.order_items_backup as select * from sql_store.order_items;
delimiter $$
DROP TRIGGER IF EXISTS sql_store.order_items_after_insert$$
create trigger sql_store.order_items_after_insert
        after insert on sql_store.order_items
        for each row
        begin
            insert into sql_store.order_items_backup (select * from sql_store.order_items where order_item_id not in (select * f
        end $$
delimiter ;
create table if not exists sql_store.order_statuses_backup as select * from sql_store.order_statuses;
delimiter $$
DROP TRIGGER IF EXISTS sql_store.order_statuses_after_insert$$
create trigger sql_store.order_statuses_after_insert
        after insert on sql_store.order_statuses
        for each row
        begin
            insert into sql_store.order_statuses_backup (select * from sql_store.order_statuses where order_status_id not in (se
        end $$
delimiter ;
create table if not exists sql_store.orders_backup as select * from sql_store.orders;
delimiter $$
DROP TRIGGER IF EXISTS sql_store.orders_after_insert$$
create trigger sql_store.orders_after_insert
        after insert on sql_store.orders

```

```

        for each row
        begin
            insert into sql_store.orders_backup (select * from sql_store.orders where order_id not in (select * from sql_store.
        end $$
delimiter ;
create table if not exists sql_store.products_backup as select * from sql_store.products;
delimiter $$
DROP TRIGGER IF EXISTS sql_store.products_after_insert$$
create trigger sql_store.products_after_insert
after insert on sql_store.products
for each row
begin
    insert into sql_store.products_backup (select * from sql_store.products where product_id not in (select * from sql_
end $$
delimiter ;

```
