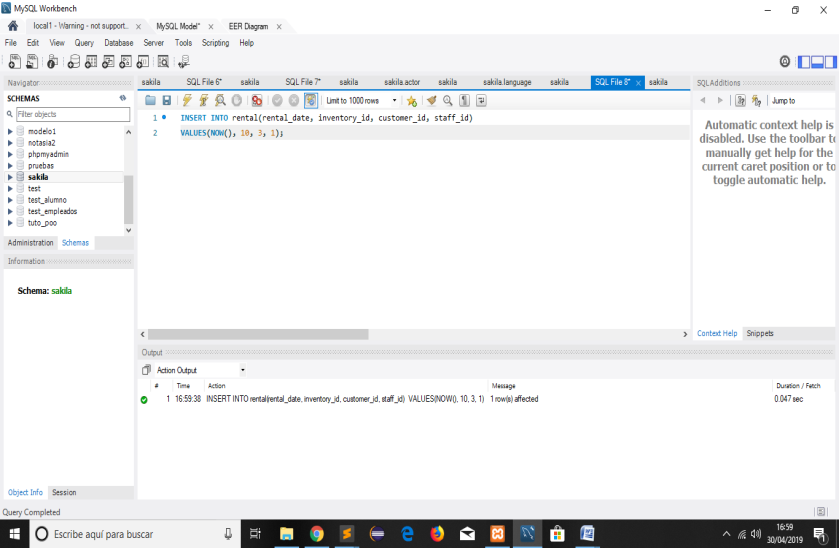
**1.- ASEGURARSE QUE TODAS LAS TABLES TIENEN CONTENIDO**

a.- Si no tienen contenido Insertar al menos 5 registros

a.1 INSERT INTO rental(rental\_date,inventory\_id,customer\_id, staff\_id)

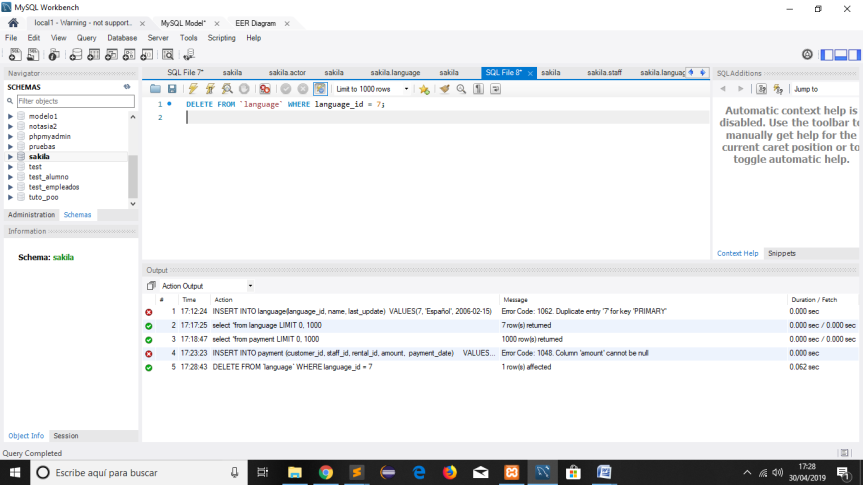
VALUES (NOW(),10,3,1);



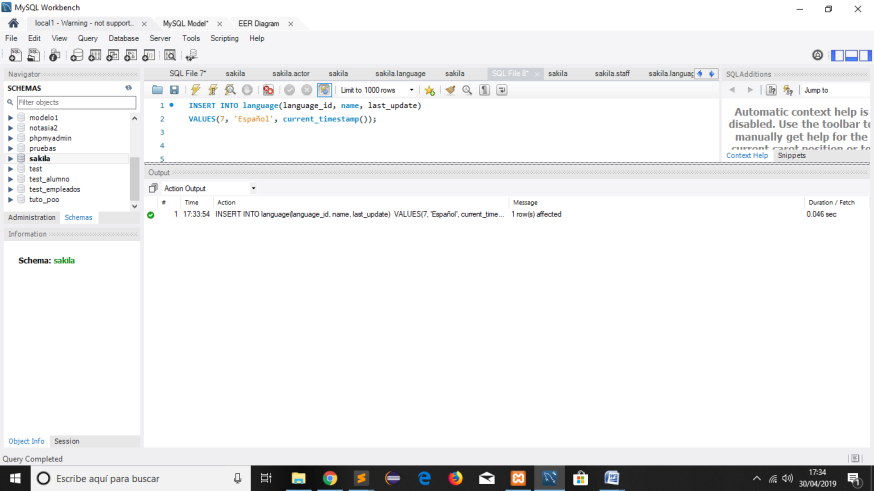
a.2INSERT INTO language(language\_id, name, last\_update)

VALUES(7, 'Español', CORRENT\_TIMESTAMP);

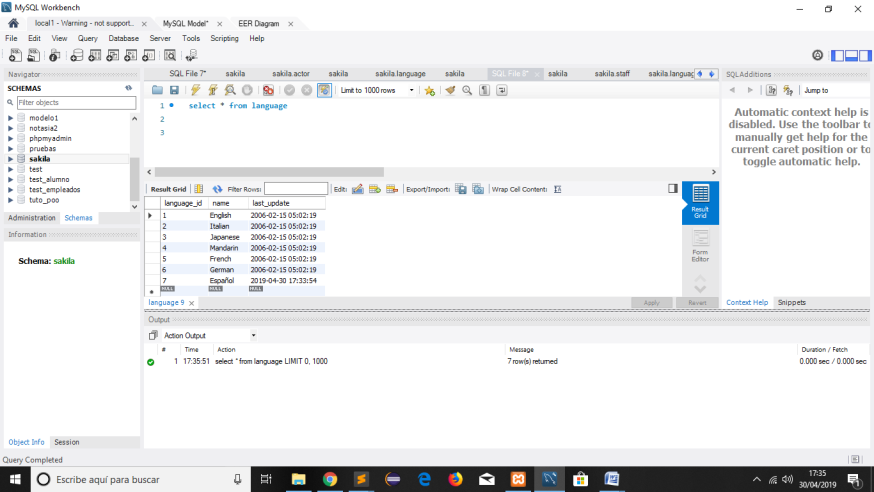
Primero borramos el registro 7 dentro de lenguaje que es el español



Luego ejecutamos la instrucción Insert y tenemos



Observamos se incorporo a la tabla



b.- Crear una Tabla e ingresar 5 registros ademas actualizar el nombre del primer registro

CREATE TABLE director ( director\_id INT PRIMARY KEY, nombre VARCHAR(20),apellido VARCHAR(50), last\_update timestamp );

INSERT INTO director VALUES ( 1, 'Will','Smith', CURRENT\_TIMESTAMP);

INSERT INTO director VALUES ( 2, 'Marry', 'Perez',CURRENT\_TIMESTAMP );

INSERT INTO director VALUES ( 3, 'Dean','rojas', CURRENT\_TIMESTAMP );

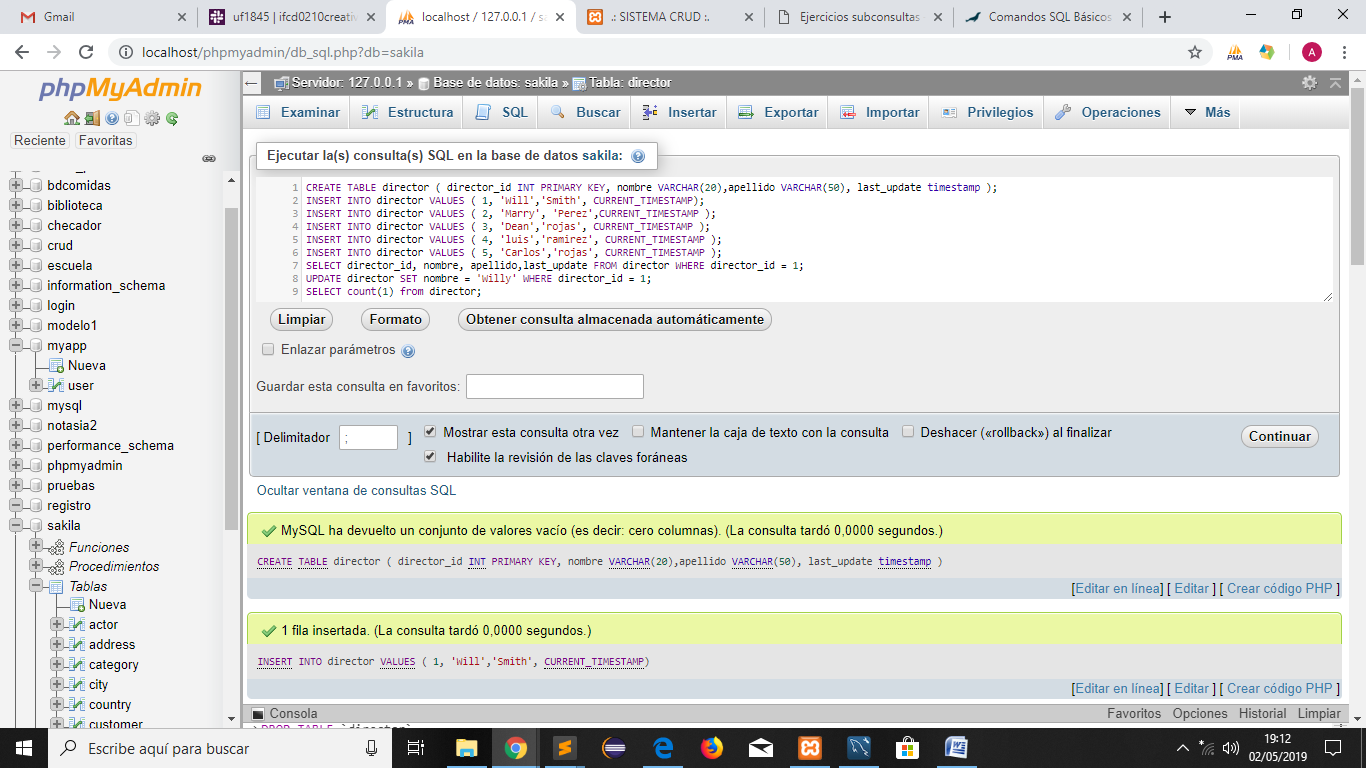
INSERT INTO director VALUES ( 4, 'luis','ramirez', CURRENT\_TIMESTAMP );

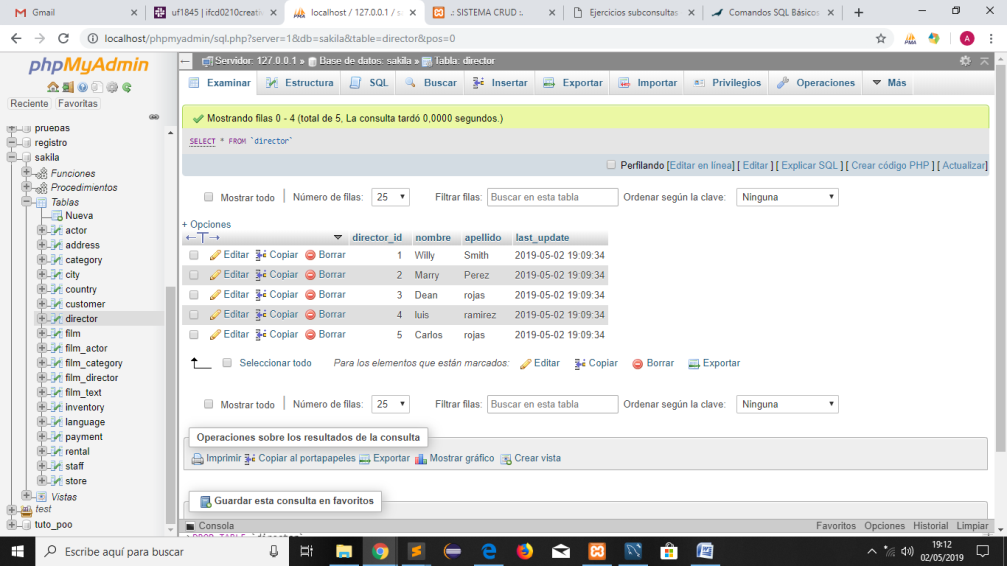
INSERT INTO director VALUES ( 5, 'Carlos','rojas', CURRENT\_TIMESTAMP );

SELECT director\_id, nombre, apellido,last\_update FROM director WHERE director\_id = 1;

UPDATE director SET nombre = 'Willy' WHERE director\_id = 1;

SELECT count(1) from director;





b.1 Crear una tabla

CREATE TABLE `customer\_log` (

`idcustomer\_log` int(11) NOT NULL AUTO\_INCREMENT,

`customer\_id` int(11) DEFAULT NULL,

`user` varchar(45) DEFAULT NULL,

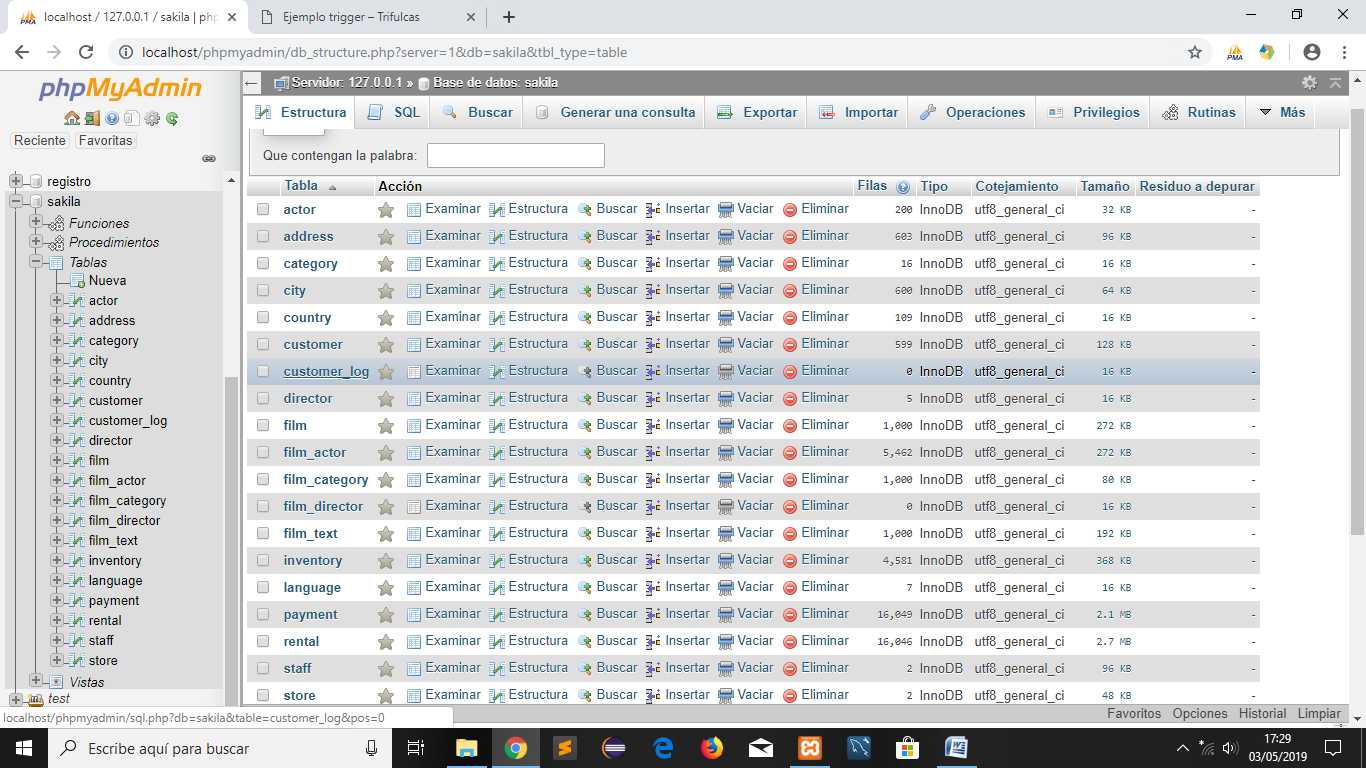
`fecha` datetime DEFAULT NULL,

`nombre` varchar(45) DEFAULT NULL,

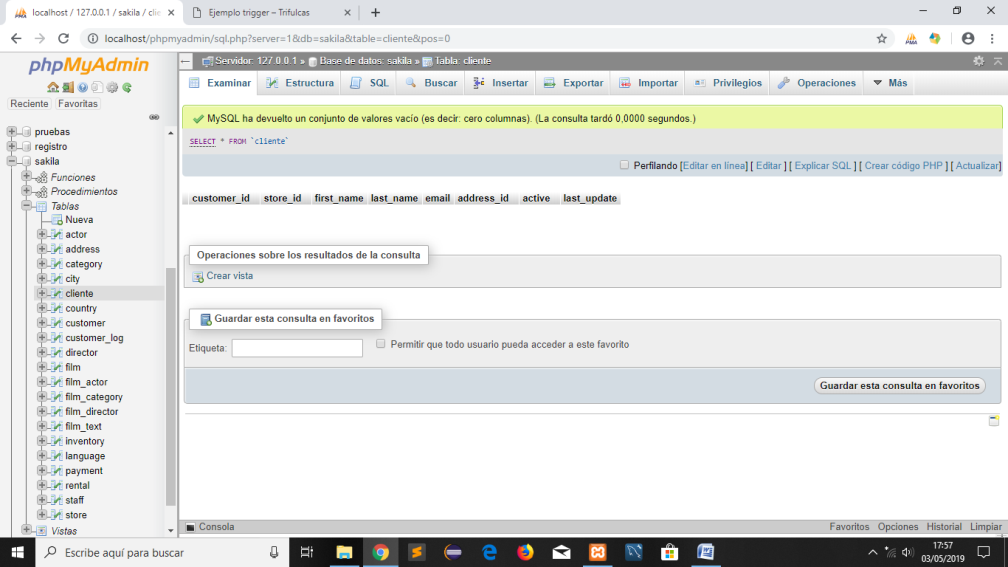
`nombre\_ant` varchar(45) DEFAULT NULL,

PRIMARY KEY (`idcustomer\_log`)

)



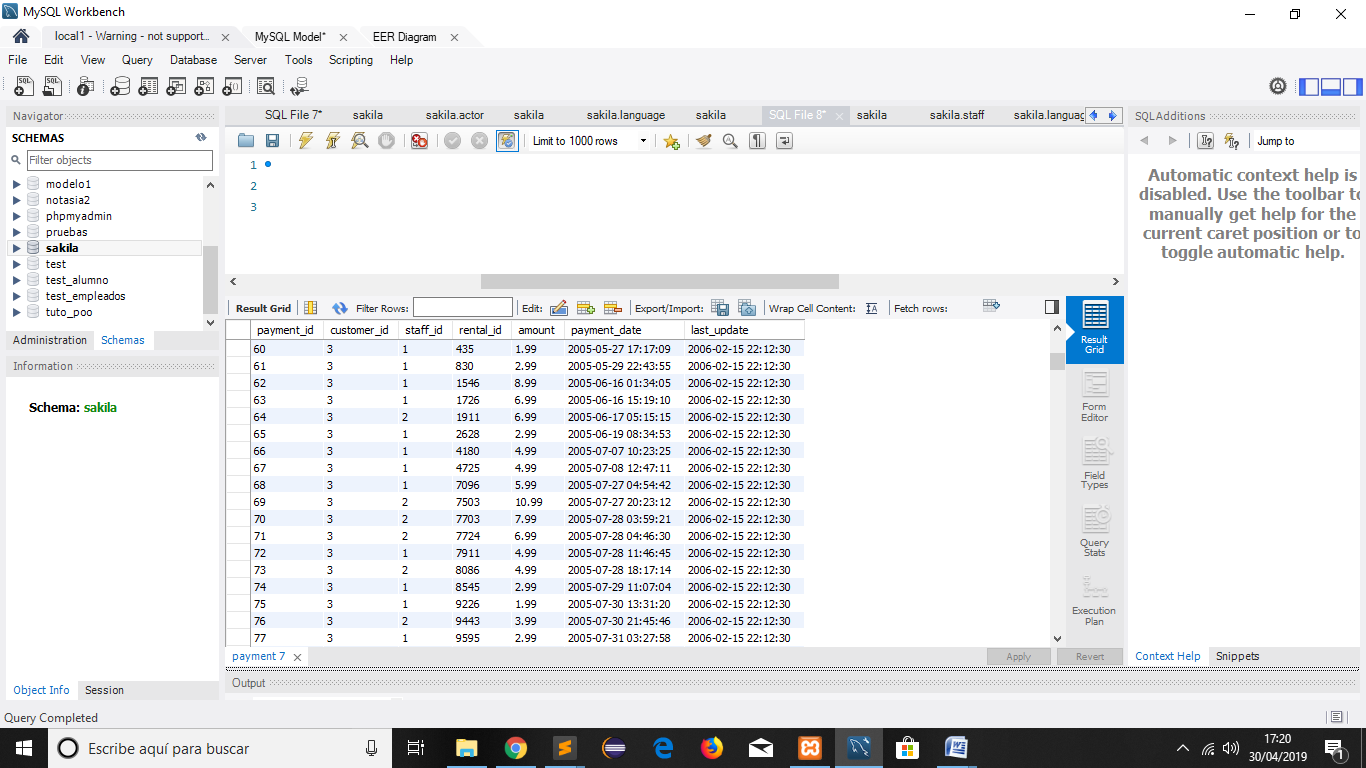
b.2. [CREATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/create-table.html) [Table](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/create-table.html) `cliente` ( `customer\_id` smallint (5) [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_not) NULL AUTO\_INCREMENT, `store\_id` [tinyint](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/numeric-types.html) (3) [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_not) NULL , `first\_name` [varchar](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/string-types.html) (45) [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_not) NULL , `last\_name` [varchar](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/string-types.html) (45) [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_not) NULL ,`email` [varchar](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/string-types.html) (50) [DEFAULT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/miscellaneous-functions.html#function_default) NULL , `address\_id` smallint (5) [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_not) NULL , `active` [tinyint](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/numeric-types.html) (1) [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_not) NULL [DEFAULT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/miscellaneous-functions.html#function_default) '1', `last\_update` [timestamp](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/date-and-time-types.html) [not](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_not) null, PRIMARY KEY (`customer\_id`) )



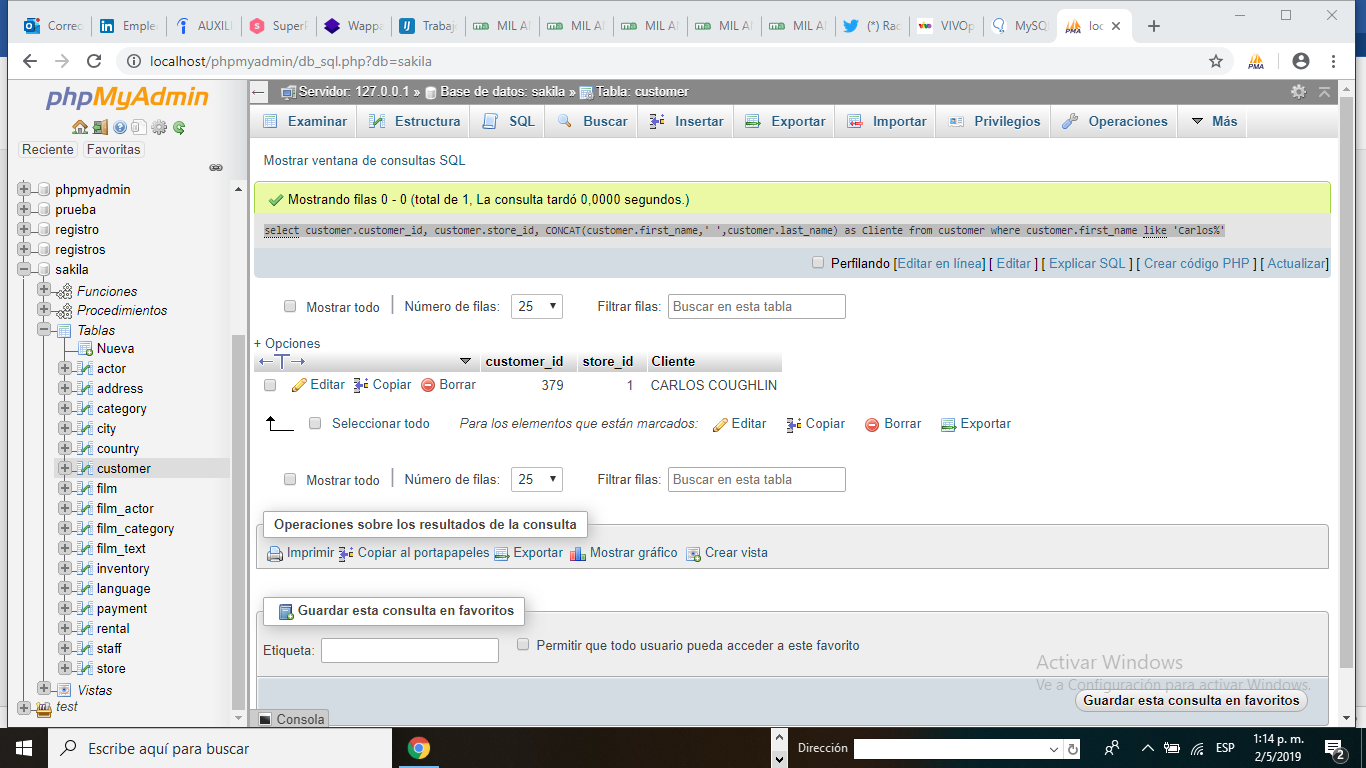
**2.-Realizar 5 consultas**

a. SQL (SELECT,FROM,WHERE,ORDER BY)

SELECT \* FROM payment

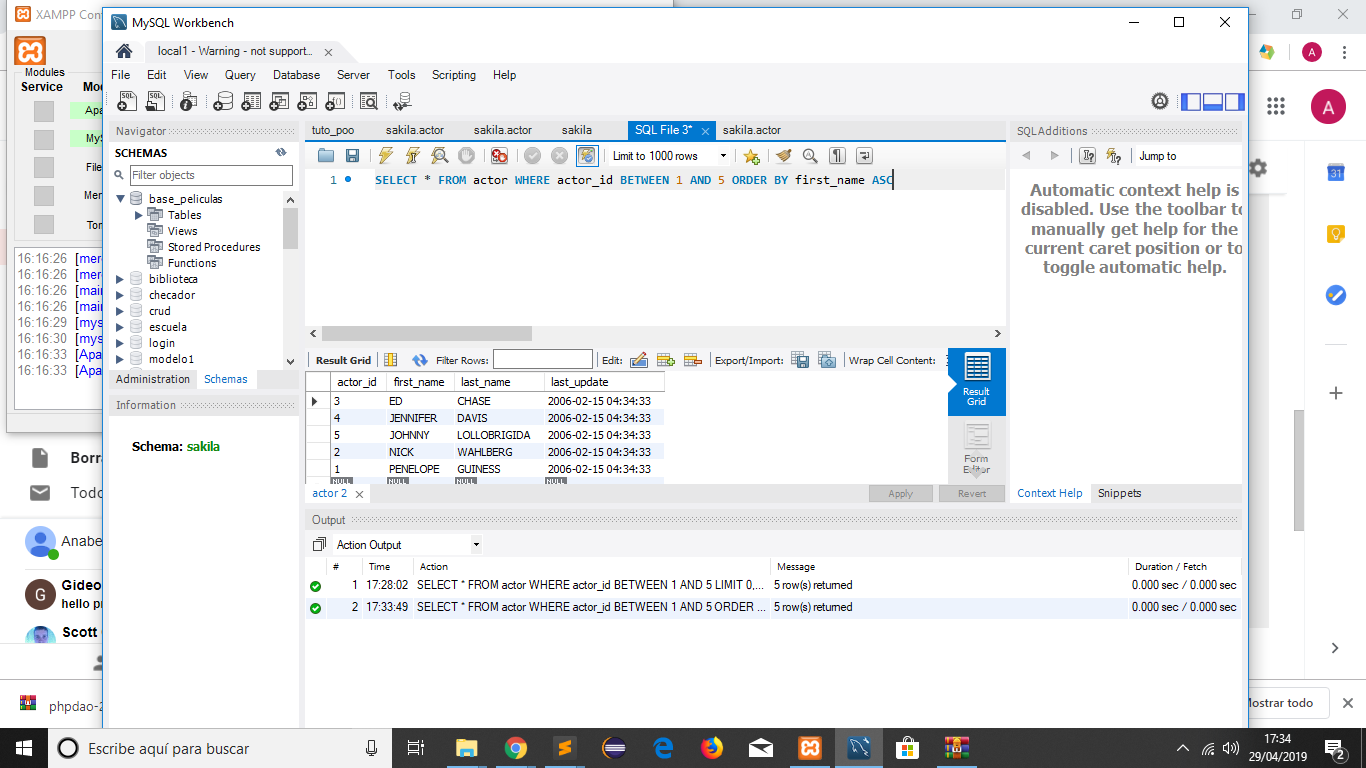


a.1 [select](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) customer.customer\_id, customer.store\_id, CONCAT(customer.first\_name,' ',customer.last\_name) as Cliente from customer where customer.first\_name [like](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/string-comparison-functions.html#operator_like) 'Carlos%'



Selecciona todos los campos de la tabla actor donde el actor\_id este entre el 1 y 5 y ordenalo alfabeticamente en ascenso por el first\_name

a.2 SELECT \* FROM actor WHERE actor\_id BETWEEN 1 AND 5 ORDER BY first\_name ASC

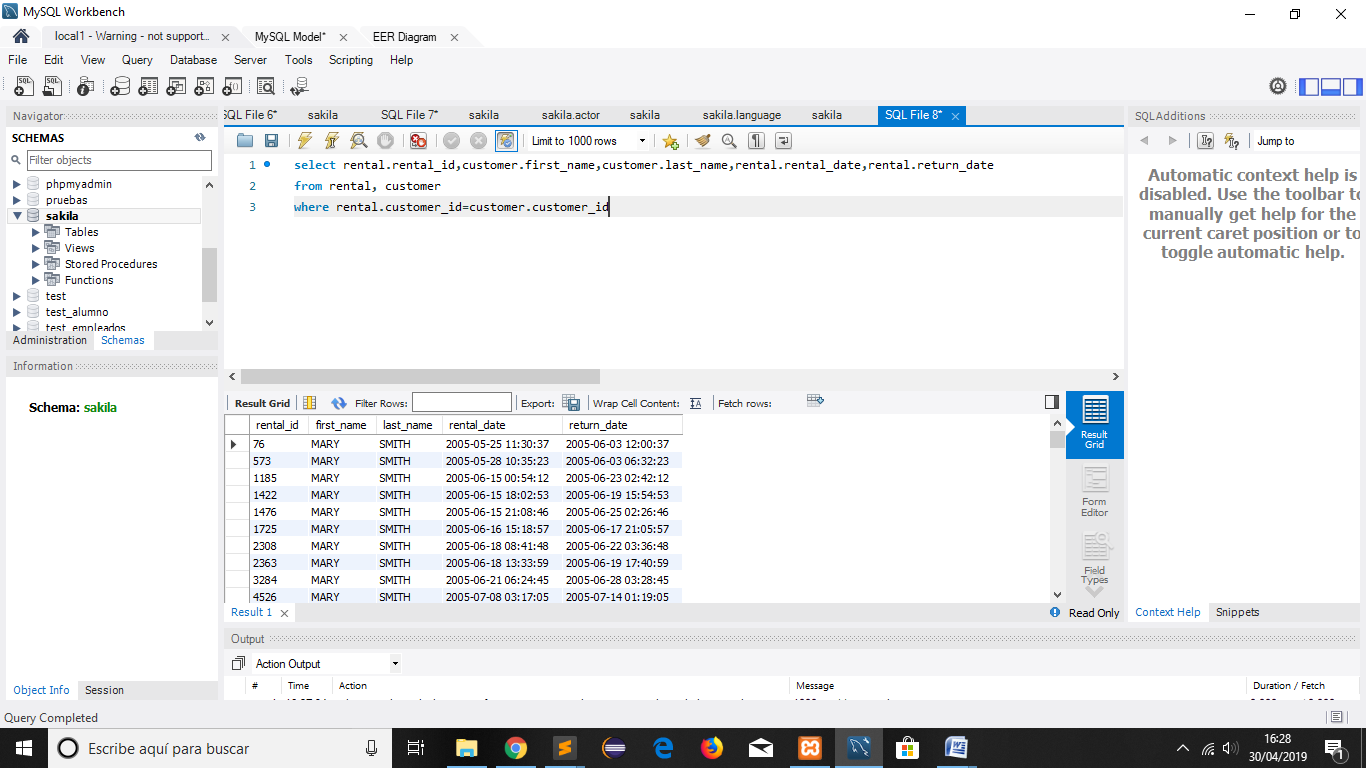


a.3 Busca en la tabla rental el cliente y la fecha de renta de películas donde el cliente sea cliente en customer

SELECT rental.rental\_id, customer.first\_name, customer\_last\_name, rental.rental\_date, rental.return\_date

FROM rental,customer

WHERE rental.customer\_id =customer.customer\_id



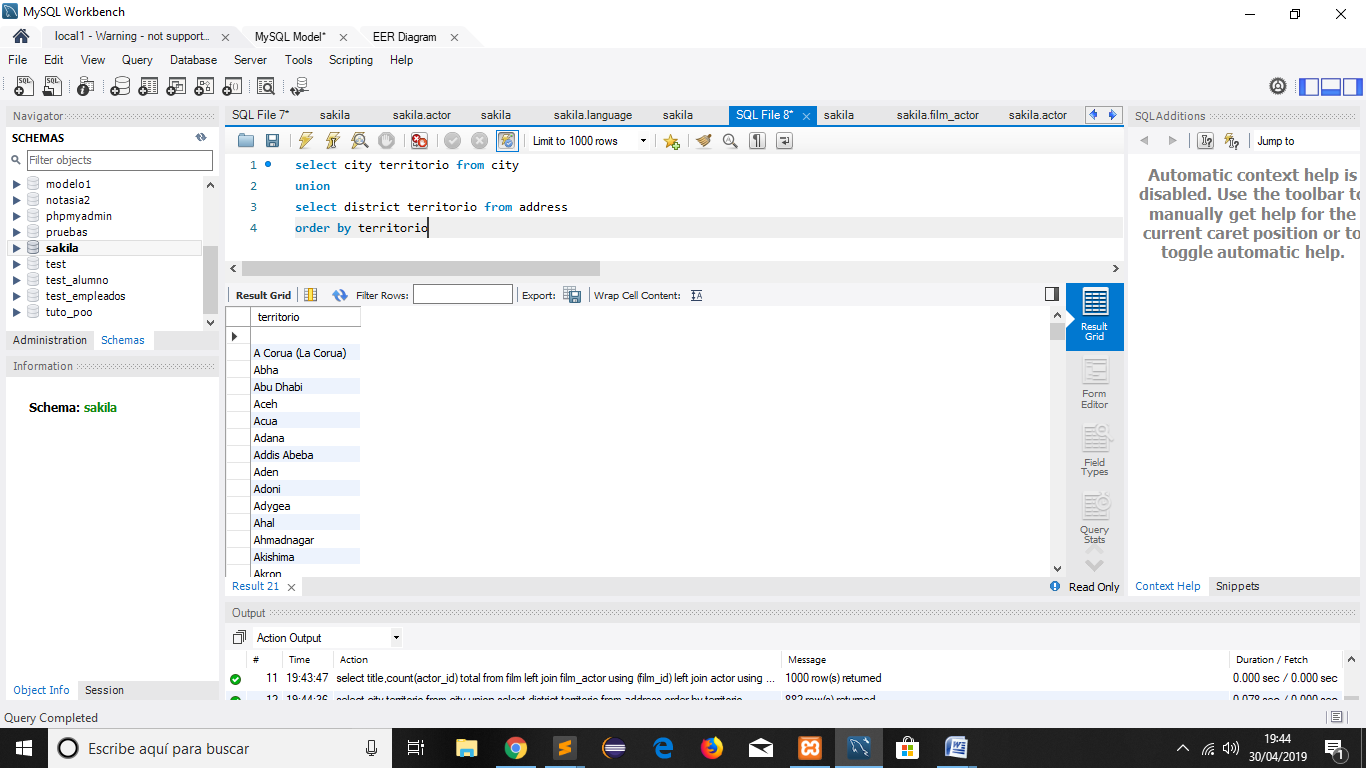
a.4 busca en la tabla city los campos city territorio y esos registros y únelo con los registros district territorio de la tabla address y muéstralo ordenado por territorrio

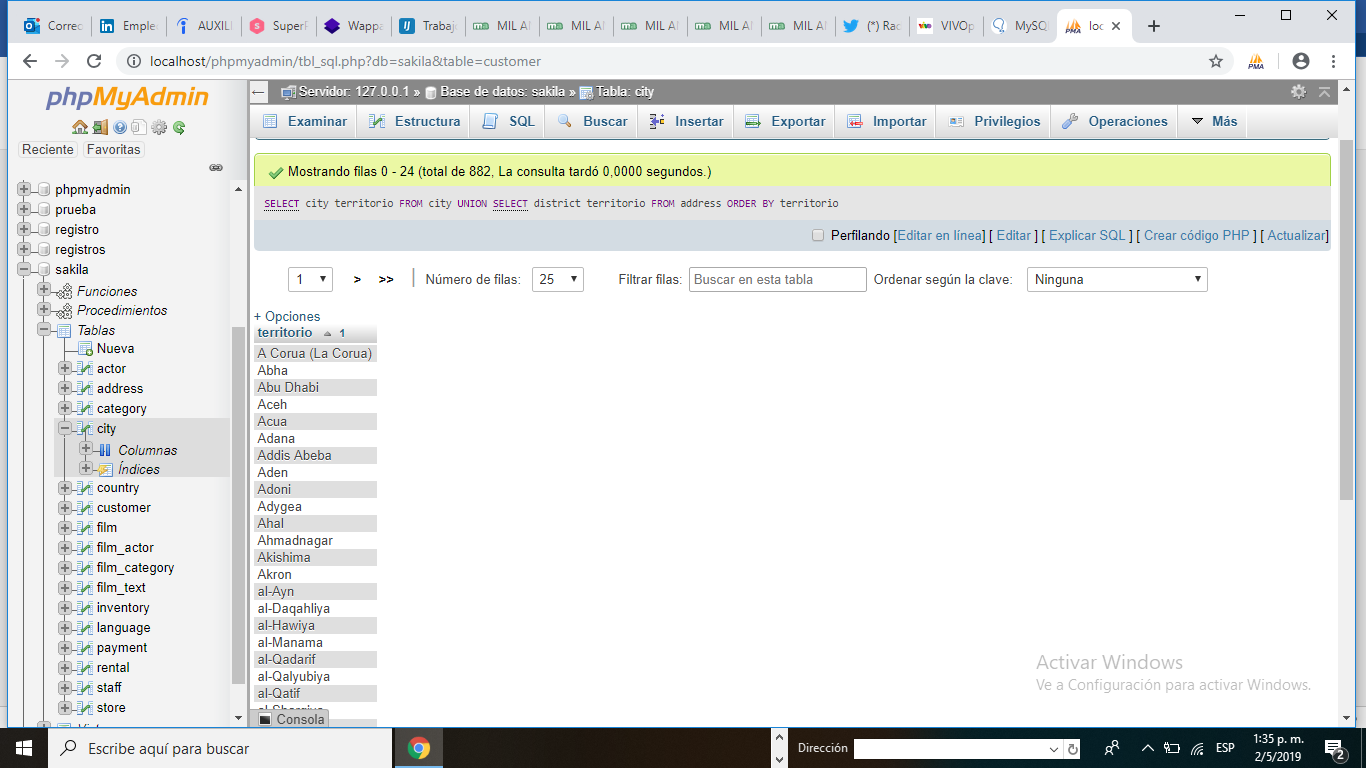
SELECTcity territorio FROM city

UNION

SELECT district territorio FROM address

ORDER BY territorio





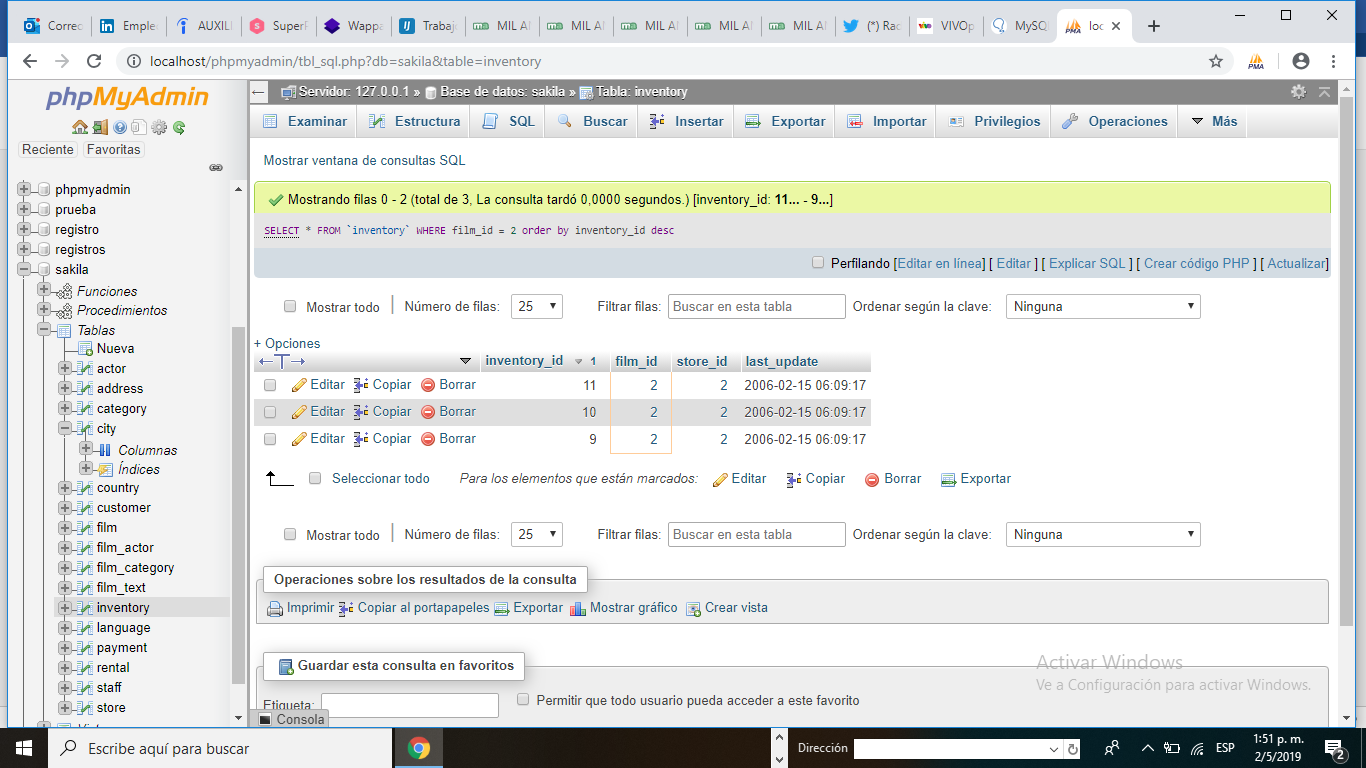
a.5Selecciona todos los campon de la tabla inventory donde el film\_id sea igual a 2 y ordenalo descendentemente por el campo inventory\_id

SELECT \*

FROM `inventory`

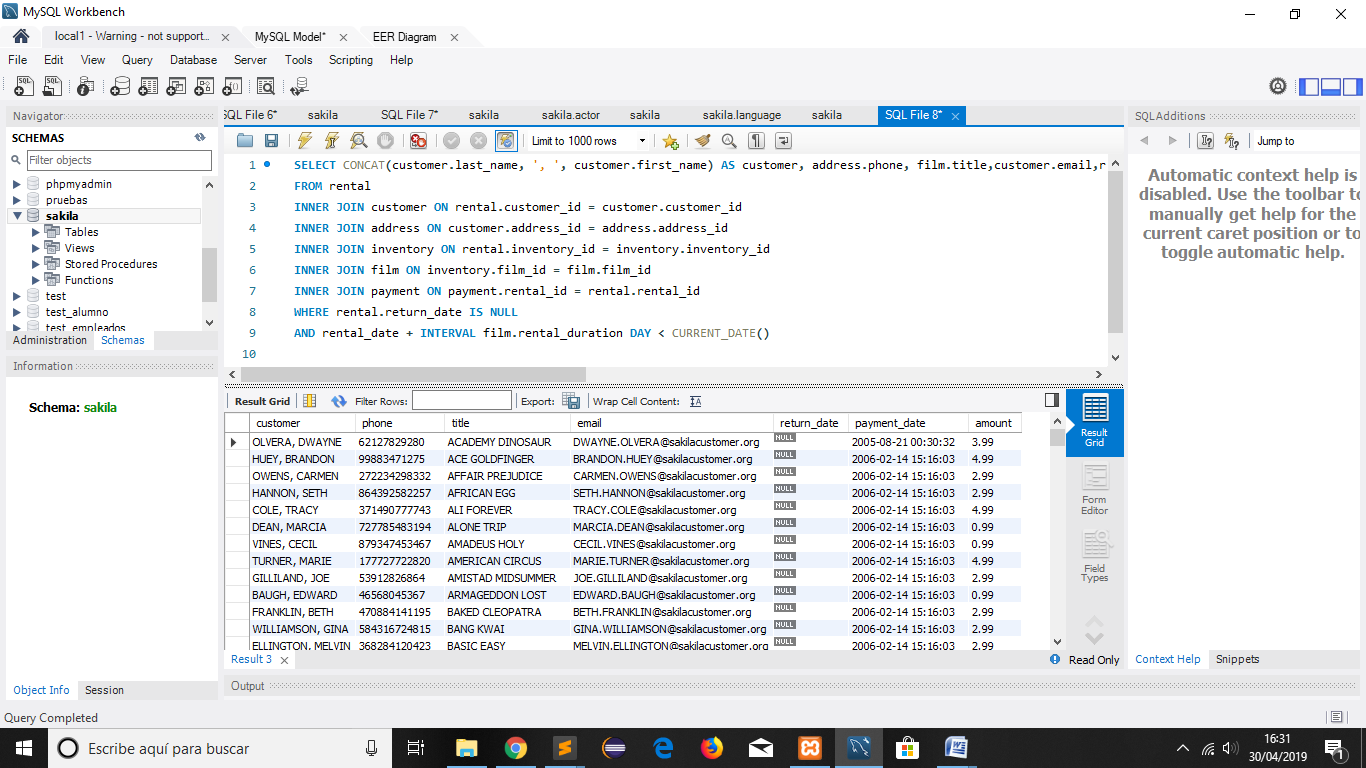
WHERE film\_id = 2

ORDER BY inventory\_idDES



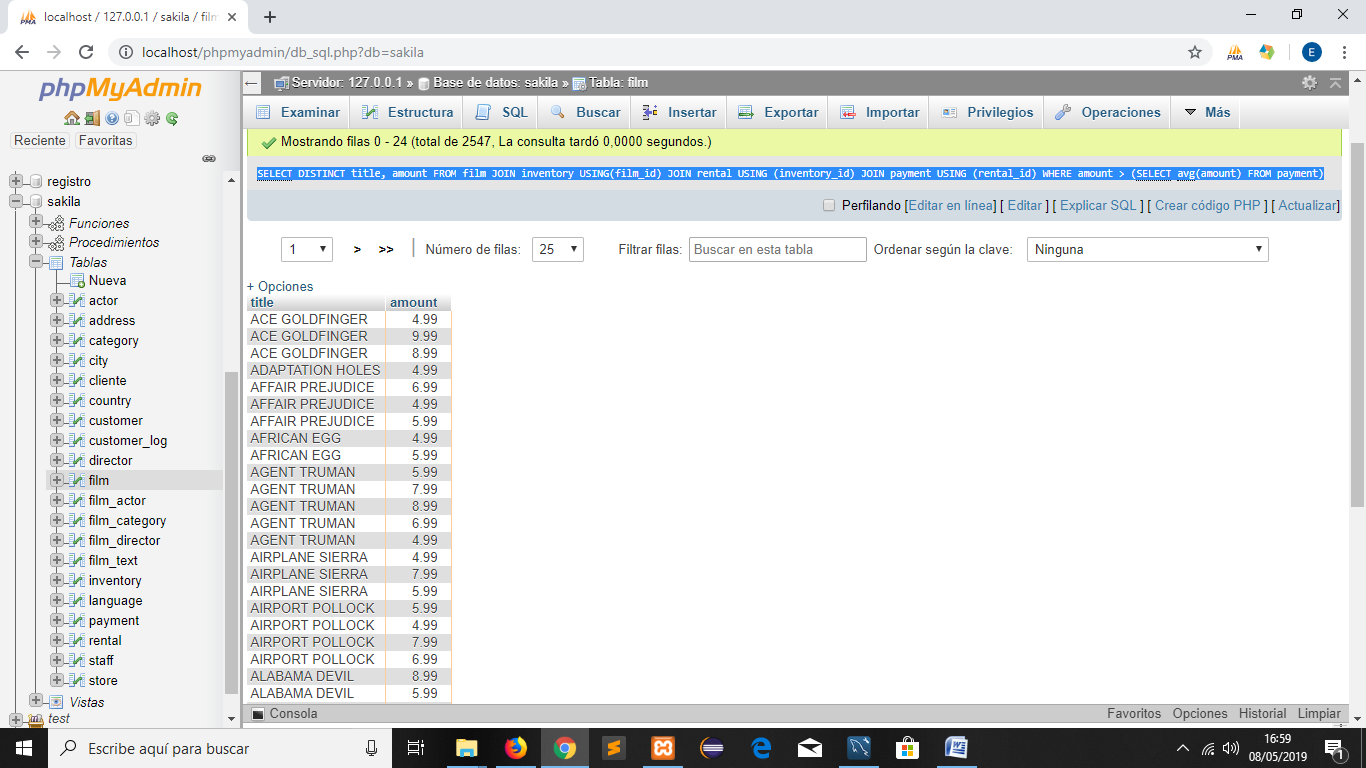
**B. SQL JOINS**

B1.- Join que genera el listado de alquileres de películas por clientes atrasados, también se mostrara si la película esta paga, en qué fecha y cuanto pago, así como, el cliente que mas películas alquilo



b.2 Alquileres por encima de la media

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) DISTINCT title, amount FROM film JOIN inventory USING(film\_id) JOIN rental USING (inventory\_id) JOIN payment USING (rental\_id) WHERE amount > ([SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) [avg](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html#function_avg)(amount) FROM payment)



b.3 Clientes con más películas alquiladas que alguno de los que empiezan por ‘A’

Count es una función que en este caso cuenta el número de veces que una película fue alquilada que cumpla con la condición de que el nombre comience por A

Group by agrupa los registros por el campo indicado**having total>**any(aquí donde indicamos que alguno empieza por A, si fuereasí**having total>all,** acá diríamos donde todos empiezan por a)

Select first\_name,count(rental\_id) total from

Customer join rental using (customer\_id)

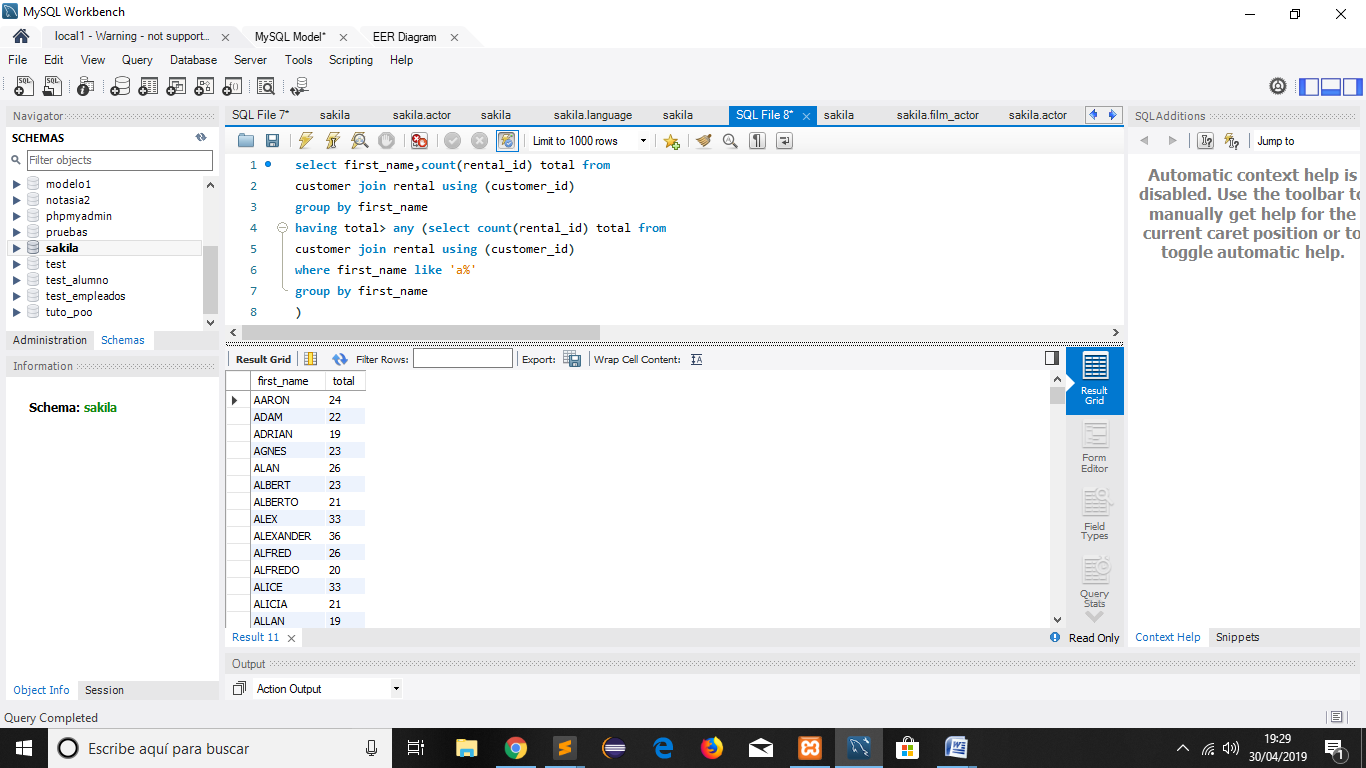
Groupby first\_name

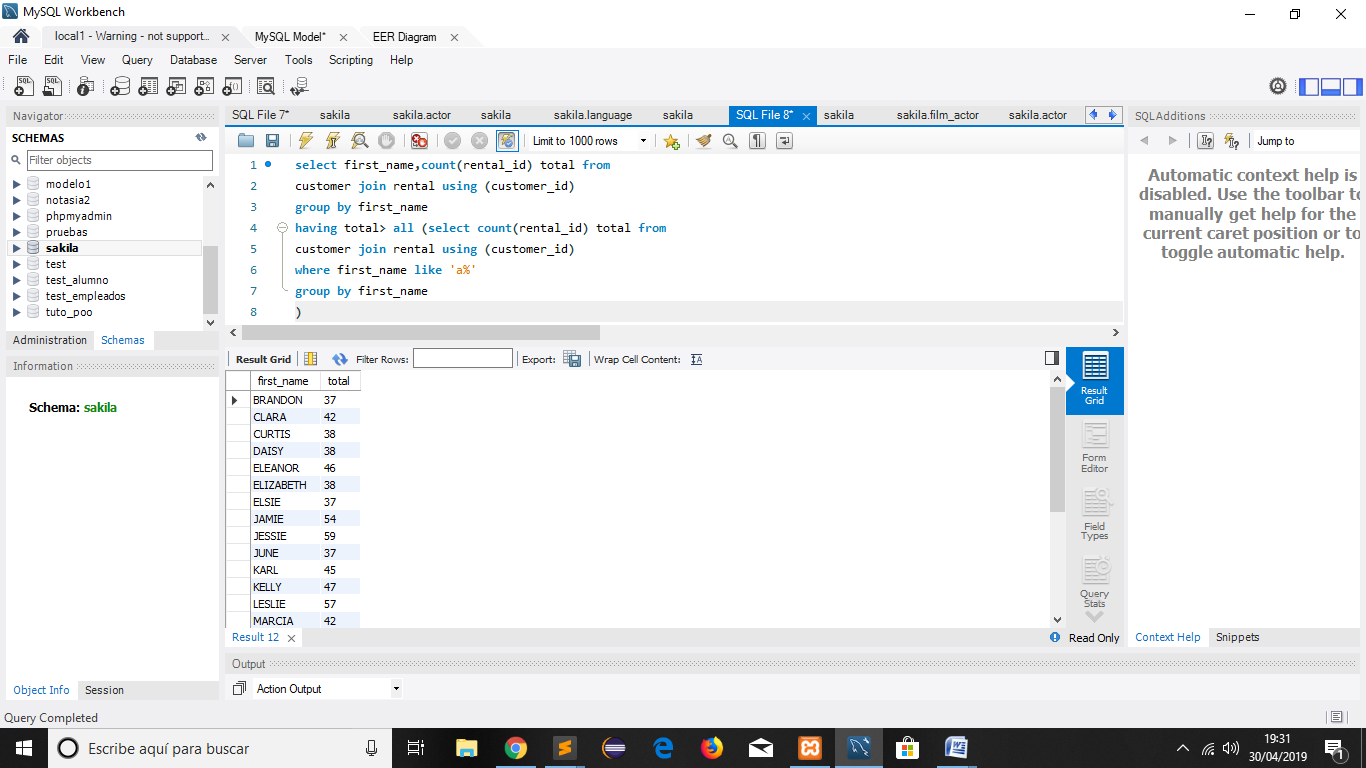
**having total>any** (selectcount(rental\_id) total from

customer join renta lusing (customer\_id)

where first\_name like 'a%'

group by first\_name)





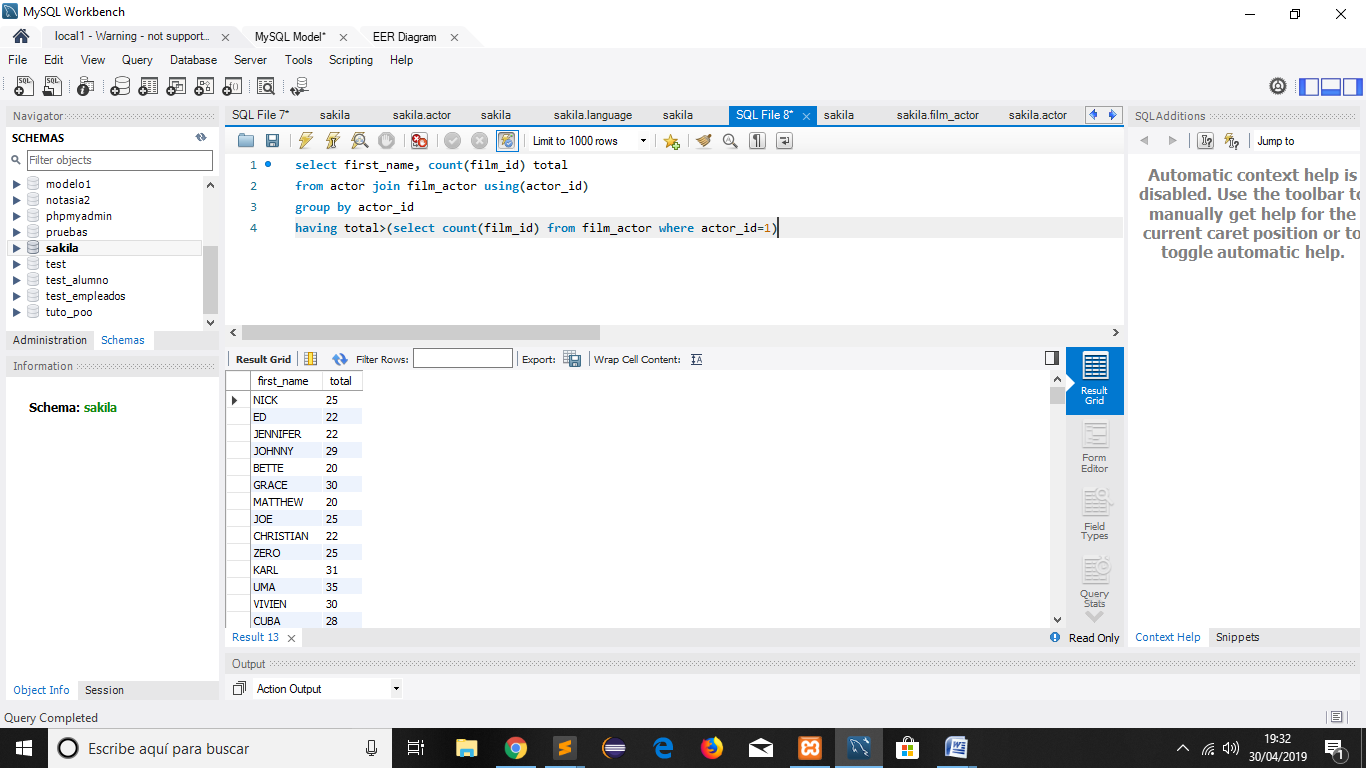
b.4 Actores con más películas que el actor de id 1

Select first\_name, count(film\_id) total

from actor join film\_actor using(actor\_id)

group by actor\_id

having total>(select count(film\_id) from film\_acto rwhere actor\_id=1)



b.5 Actores que trabajan en películas con rating ‘R’

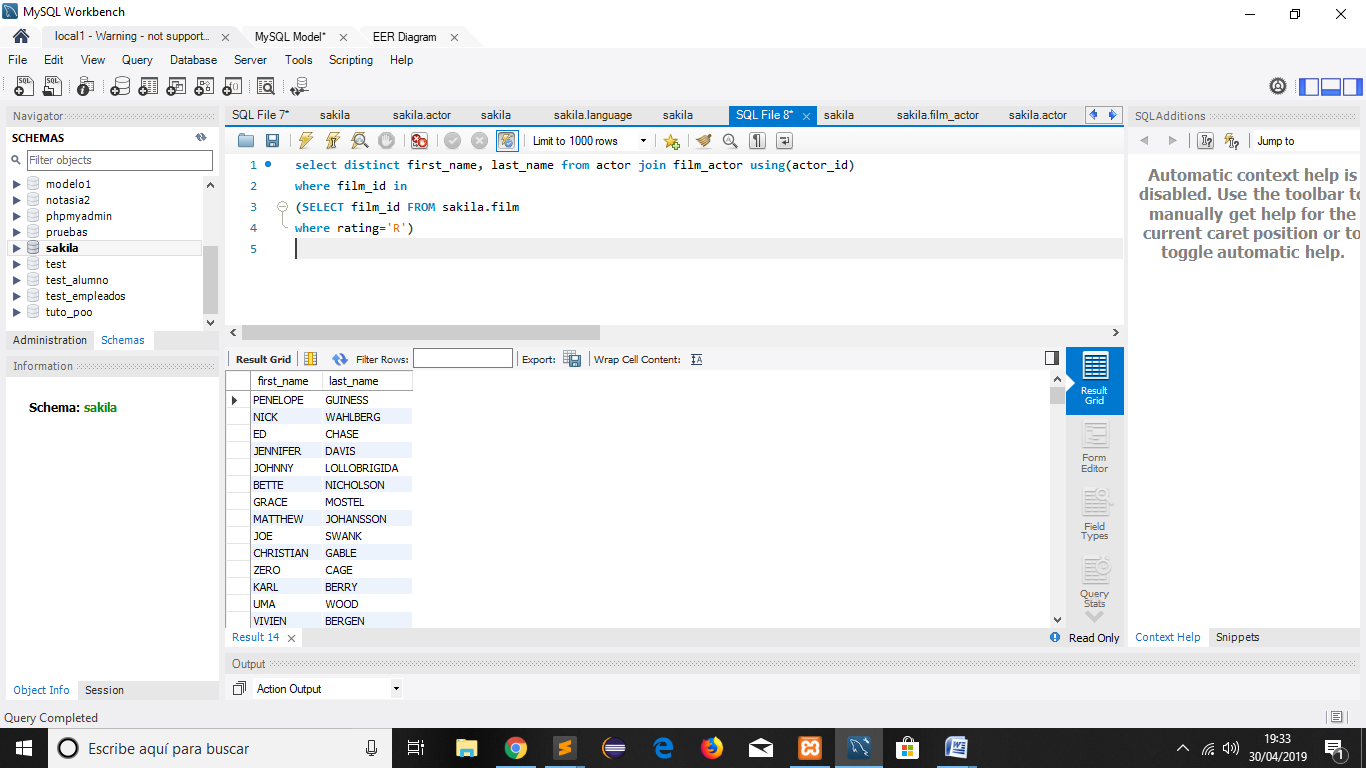
Distinct, con este comando te permite evitar las repeticiones de registro en esta consulta se traen los nombres y apellidos de los actores con mas rating.

Select distinct first\_name, last\_name from actor join film\_actorusing(actor\_id)

Where film\_id in

(SELECT film\_id FROM sakila.film

where rating='R')

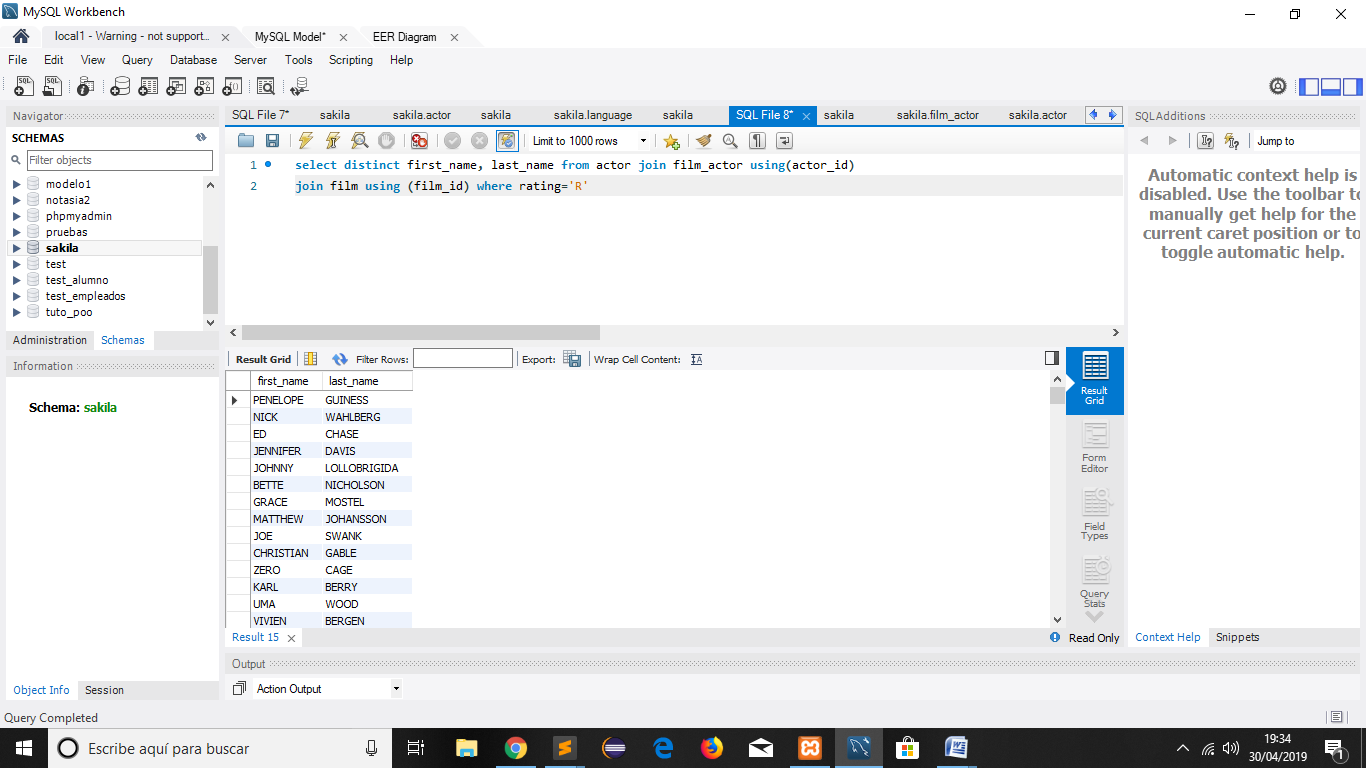


b.5

ó

Select distinct first\_name, last\_name from actor join film\_acto rusing(actor\_id)

join film using (film\_id) where rating='R'



b.6 Actores que no hayan trabajado en películas de acción(con not in, indico que no estén incluidos)

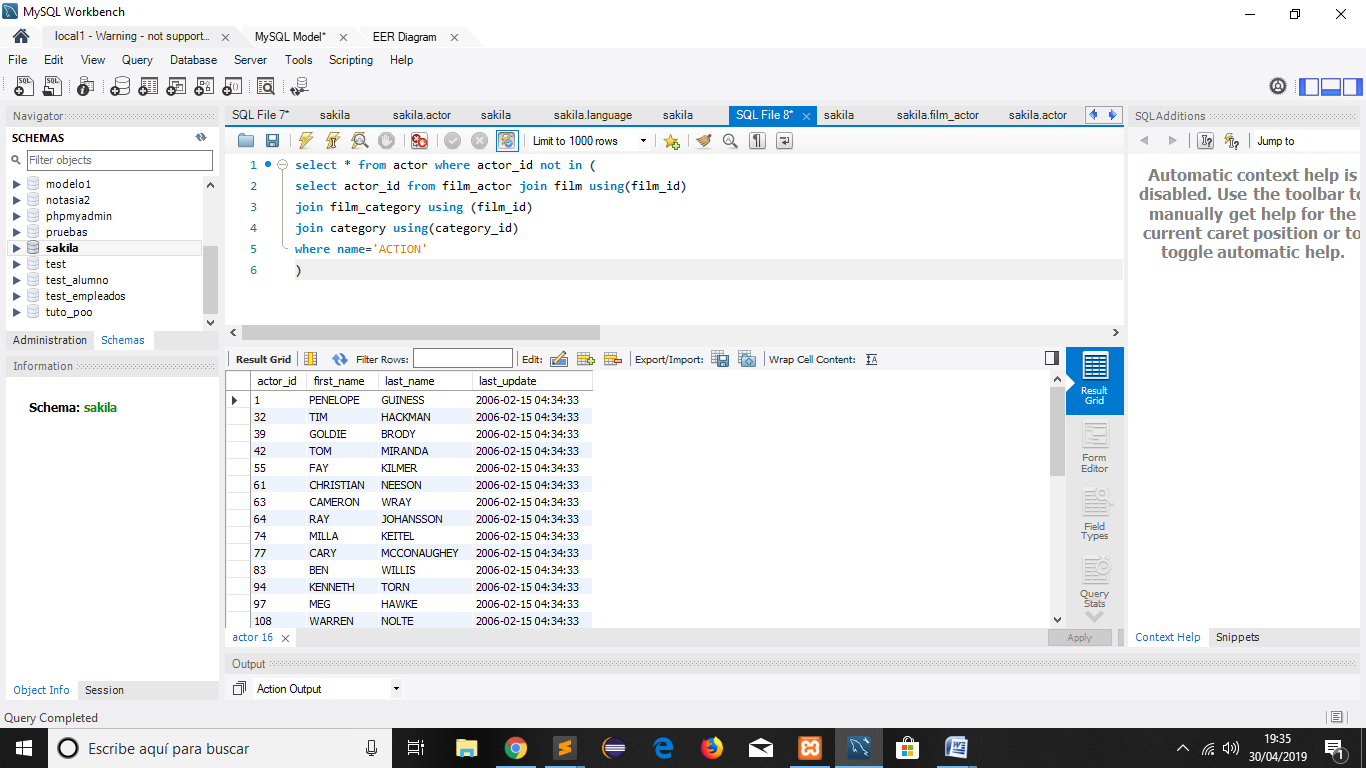
select \* from actor whereactor\_id not in (

select actor\_id fromfilm\_acto rjoin film using(film\_id)

join film\_categoryusing (film\_id)

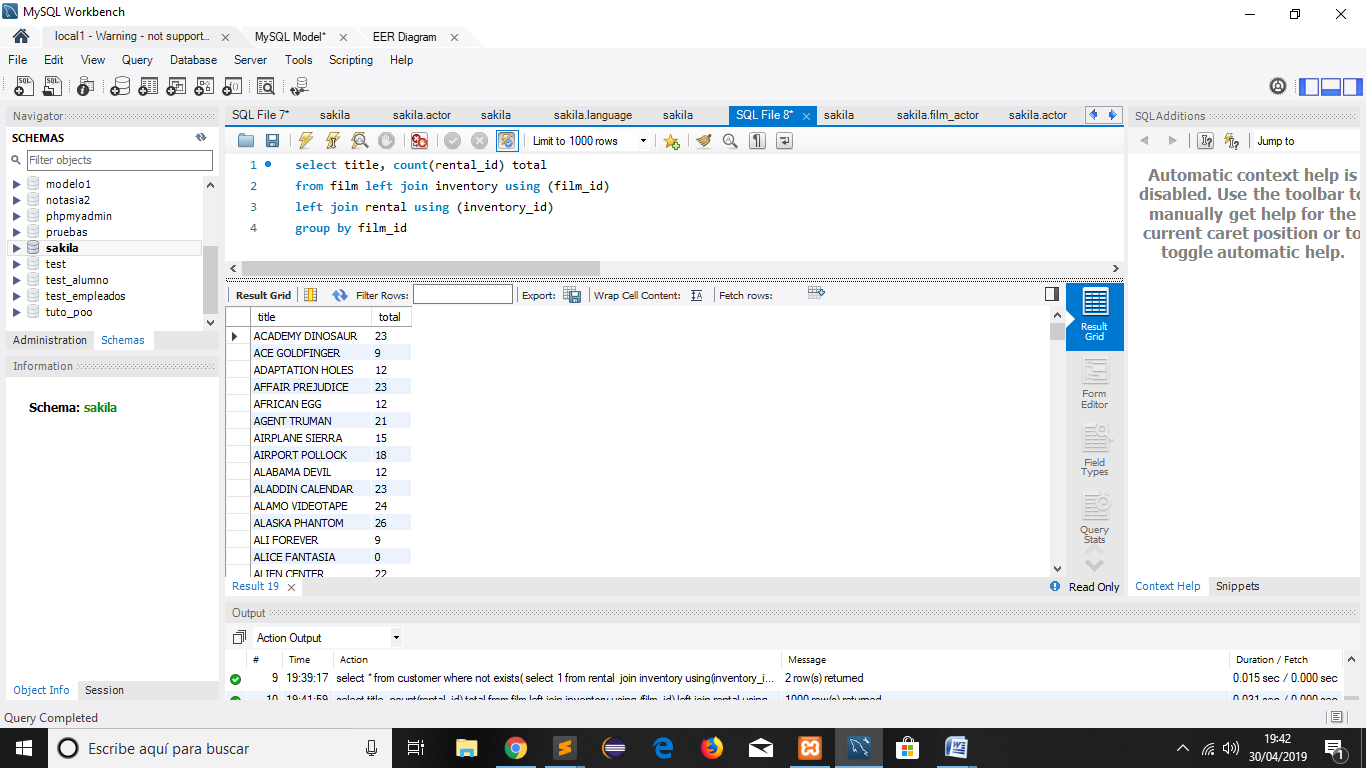
join categoryusing(category\_id)

where name='ACTION')



b.7 Número de veces que se ha alquilado una película, incluyendo las que no se han alquilado nunca

Select title, count(rental\_id) total from film left joininventory using (film\_id)left joinrental using (inventory\_id)group byfilm\_id



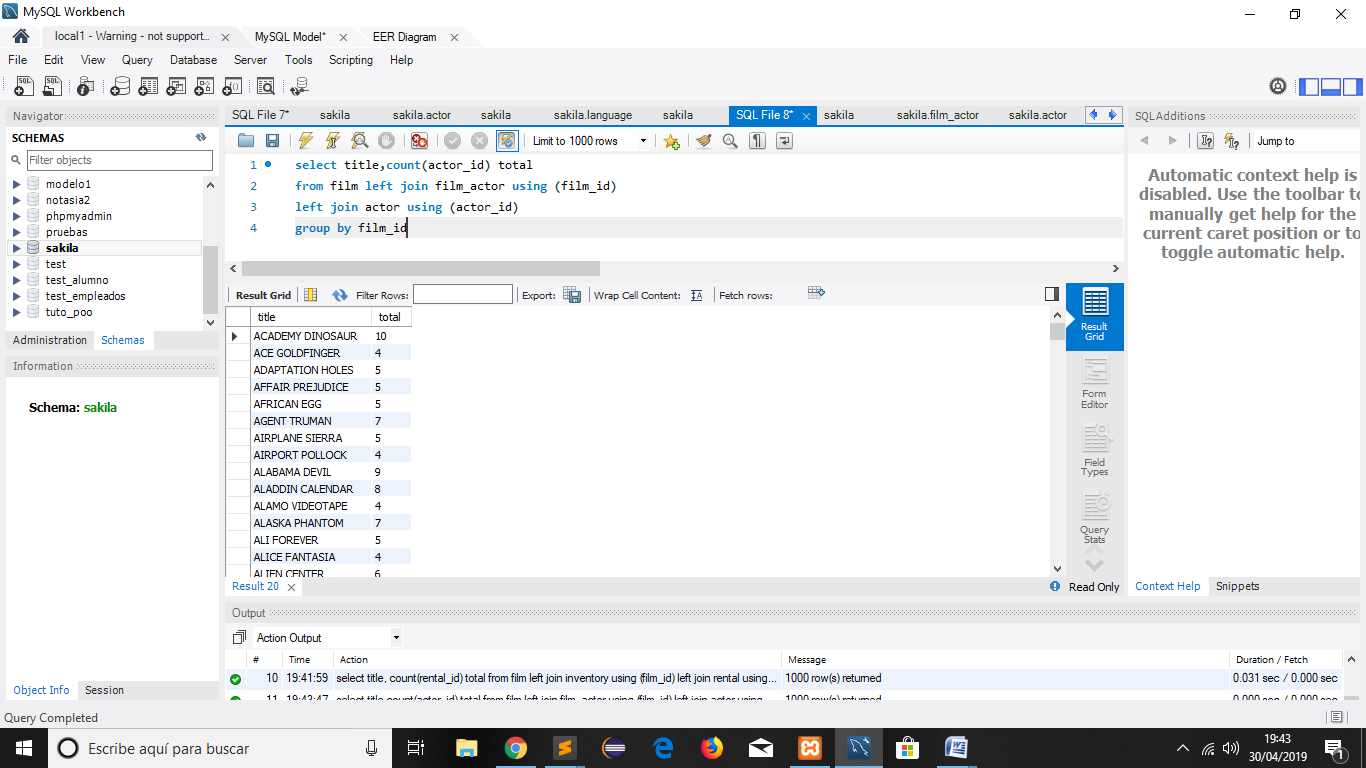
b.8 Número de actores por película, incluyendo las películas que no tengan actores

Select title,count(actor\_id) total

From film left join film\_actor using (film\_id)

Left join actor using (actor\_id)

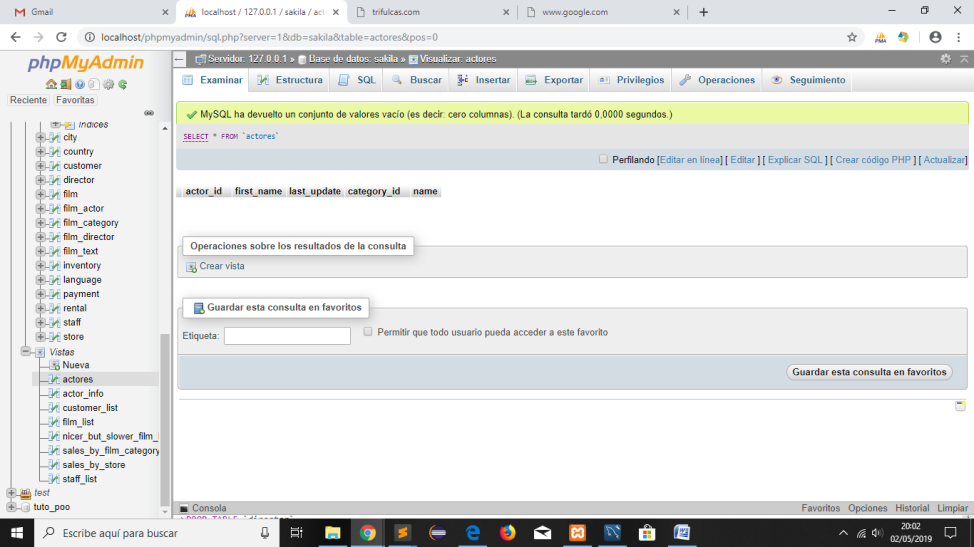
Group by film\_id



**3- Realizar al menos una sub-consulta de su elección que genere al menos una Vistas**

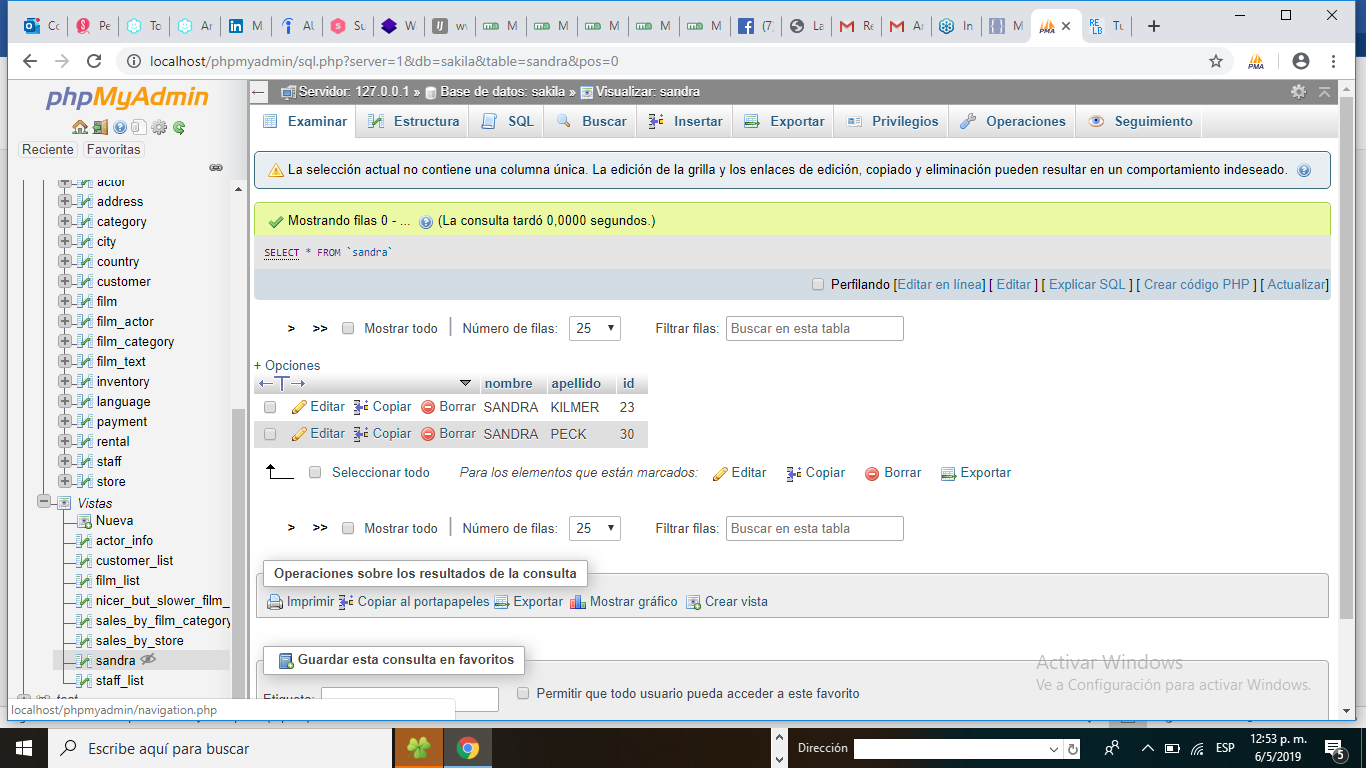
Vista

[CREATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/create-view.html) [VIEW](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/create-view.html) ACTORES AS [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) a.actor\_id, a.first\_name, a.last\_update, c.category\_id, c.name FROM actor a JOIN category c on a.last\_update=c.last\_update

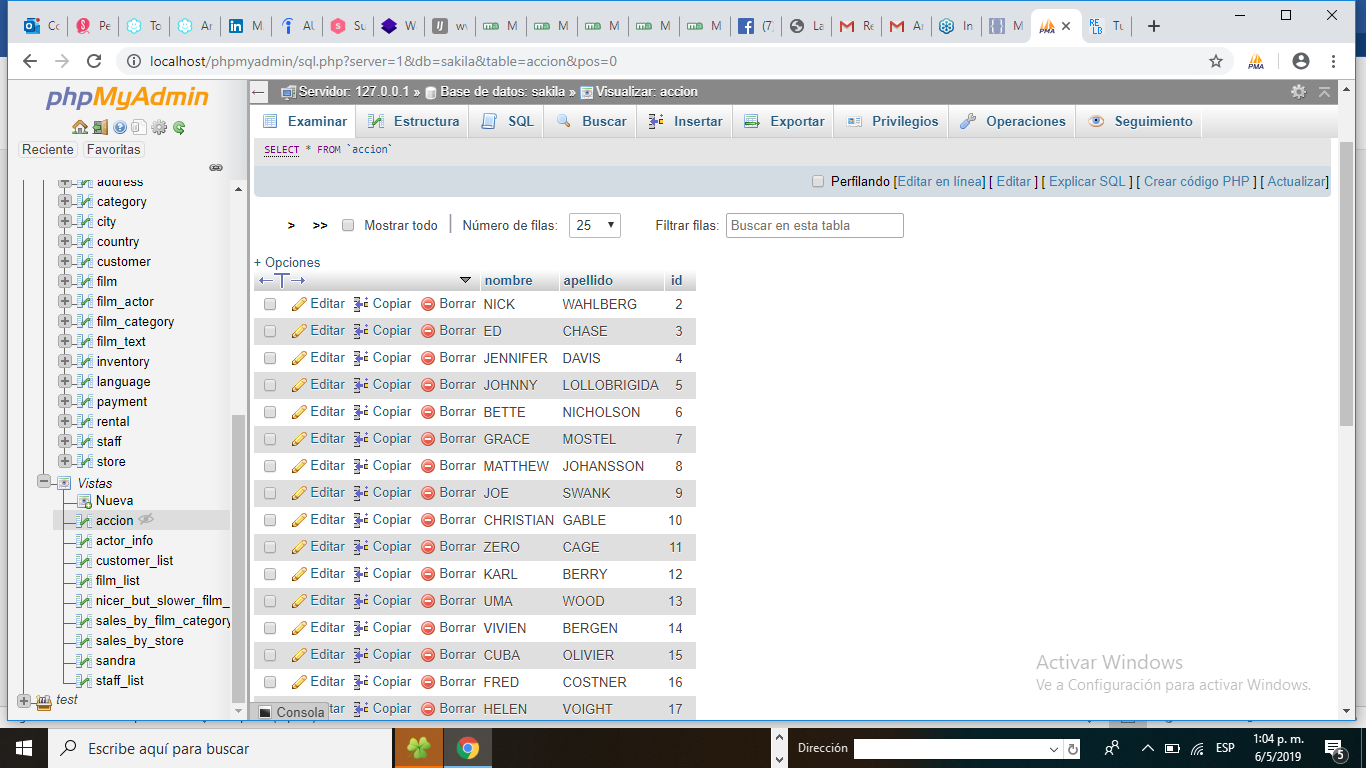


3.2.- Sub-consulta con vista

[CREATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/create-view.html) [VIEW](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/create-view.html) sandra AS [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) first\_name as nombre, last\_name as apellido, actor\_id as id FROM actor WHERE first\_name = 'SANDRA'

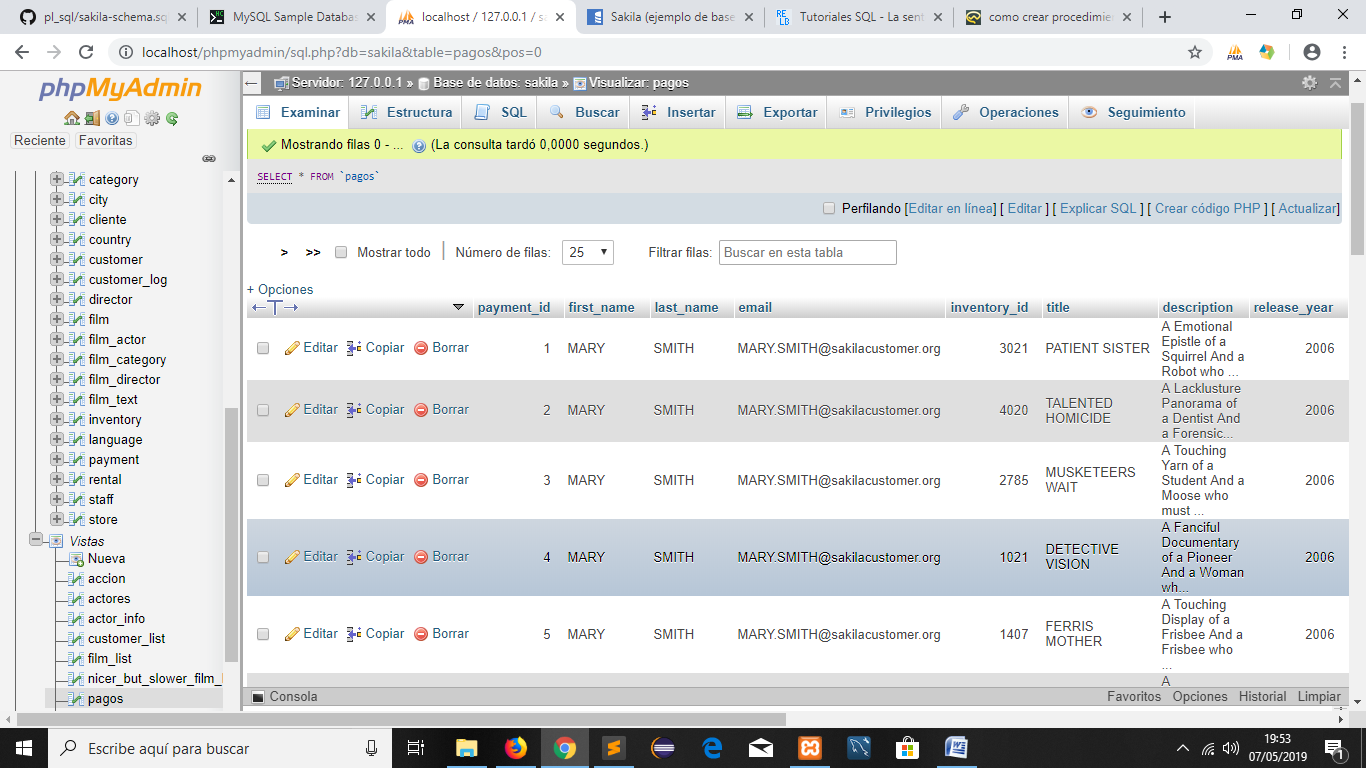


3.3[CREATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/create-view.html) [VIEW](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/create-view.html) ACCION AS [select](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) first\_name as nombre,last\_name as apellido, actor\_id as id from actor where actor\_id [in](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/comparison-operators.html#function_in)( [select](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) actor\_id as id from film\_actor join film using(film\_id)join film\_category using (film\_id) join category using(category\_id) where name='ACTION')



**3.4** Crea una vista llamada pagos donde los campos se rellenaran con los datos provenientes de la conjunción de las tablas payment, rental, inventory y film

[CREATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/create-view.html) [VIEW](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/create-view.html) pagos AS [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) payment.payment\_id, customer.first\_name, customer.last\_name, customer.email, inventory.inventory\_id, film.title, film.description, film.release\_year,film.rental\_duration, film.rental\_rate, film.length, film.rating, film.special\_features, rental.rental\_date, rental.return\_date, payment.amount, payment.payment\_date FROM payment [left](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/string-functions.html#function_left)join customer on ( payment.customer\_id = customer.customer\_id) [left](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/string-functions.html#function_left) join rental on ( payment.rental\_id = rental.rental\_id) [left](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/string-functions.html#function_left) join inventory on ( rental.inventory\_id =inventory.inventory\_id) [left](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/string-functions.html#function_left) join film on ( inventory.film\_id = film.film\_id)



**4. TRIGGERS (hacer 2)**

**4.1** DELIMITER //

CREATE TRIGGER cliente\_actualizado BEFORE UPDATE ON customer

FOR EACH ROW BEGIN

DECLARE vUser varchar(50);

-- Almacena el nombre del usuario después de insertarlo en la columna de la misma tabla

SELECT USER() INTO vUser;

-- Inserta el récor de un cliente

INSERT INTO customer\_log

( customer\_id,

fecha,

user,nombre,nombre\_ant)

VALUES

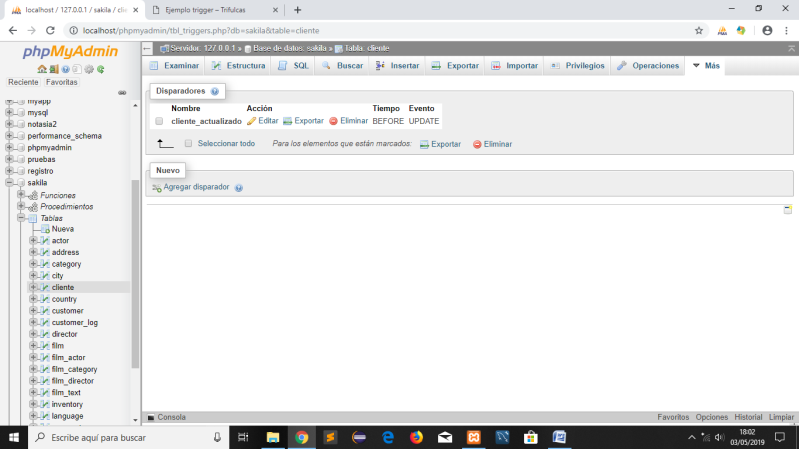
( NEW.customer\_id,

SYSDATE(),

vUser,NEW.first\_name, OLD.first\_name );

END;//

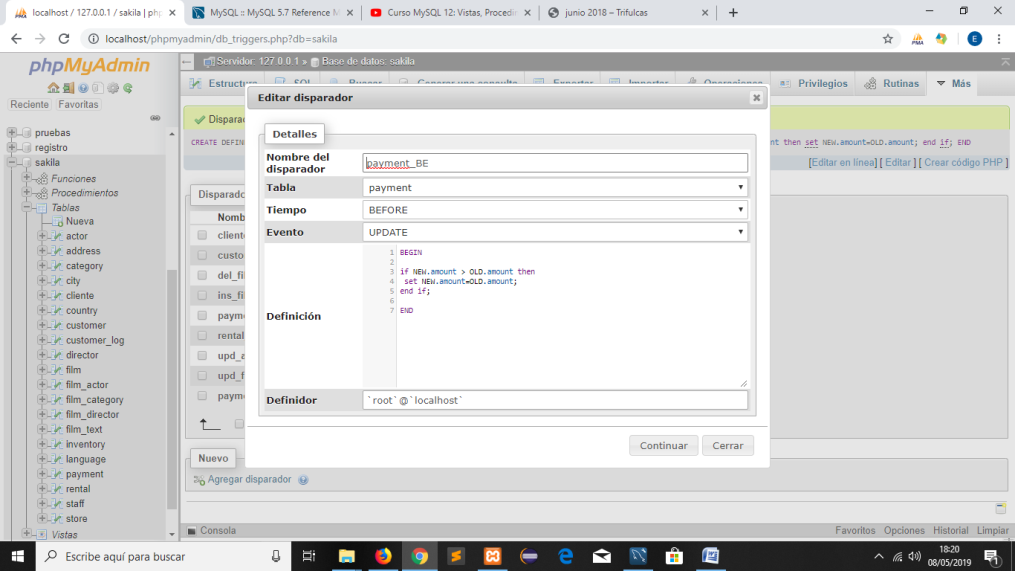
DELIMITER ;



4.2 [DROP](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/drop-trigger.html) [TRIGGER](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/drop-trigger.html) [IF](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/control-flow-functions.html#function_if) EXISTS `upd\_film`;CREATE DEFINER=`root`@`localhost` TRIGGER `upd\_actor` AFTER UPDATE ON `actor` FOR EACH ROW BEGIN [IF](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/control-flow-functions.html#function_if) (old.first\_name != new.first\_name) [OR](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_or) (old.last\_name!= new.last\_name) [OR](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_or) (old.actor\_id != new.actor\_id) THEN [UPDATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/update.html) film\_actor [SET](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/set.html) first\_name=new.first\_name, last\_name=new.last\_name, actor\_id=new.actor\_id WHERE actor\_id=old.actor\_id; END[IF](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/control-flow-functions.html#function_if); END



4.3 CREATE DEFINER=```root```@```localhost``` TRIGGER `payment\_BE` BEFORE UPDATE ON `payment` FOR EACH ROW BEGIN [if](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/control-flow-functions.html#function_if) NEW.amount > OLD.amount then [set](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/set.html) NEW.amount=OLD.amount; end [if](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/control-flow-functions.html#function_if); END



**5. FUNCIONES**

5.1 SELECT

CASE dayofweek(payment\_date)

when 1 then 'Domingo'

when 2 then 'Lunes'

when 3 then 'Martes'

when 4 then 'Miércoles'

when 5 then 'Jueves'

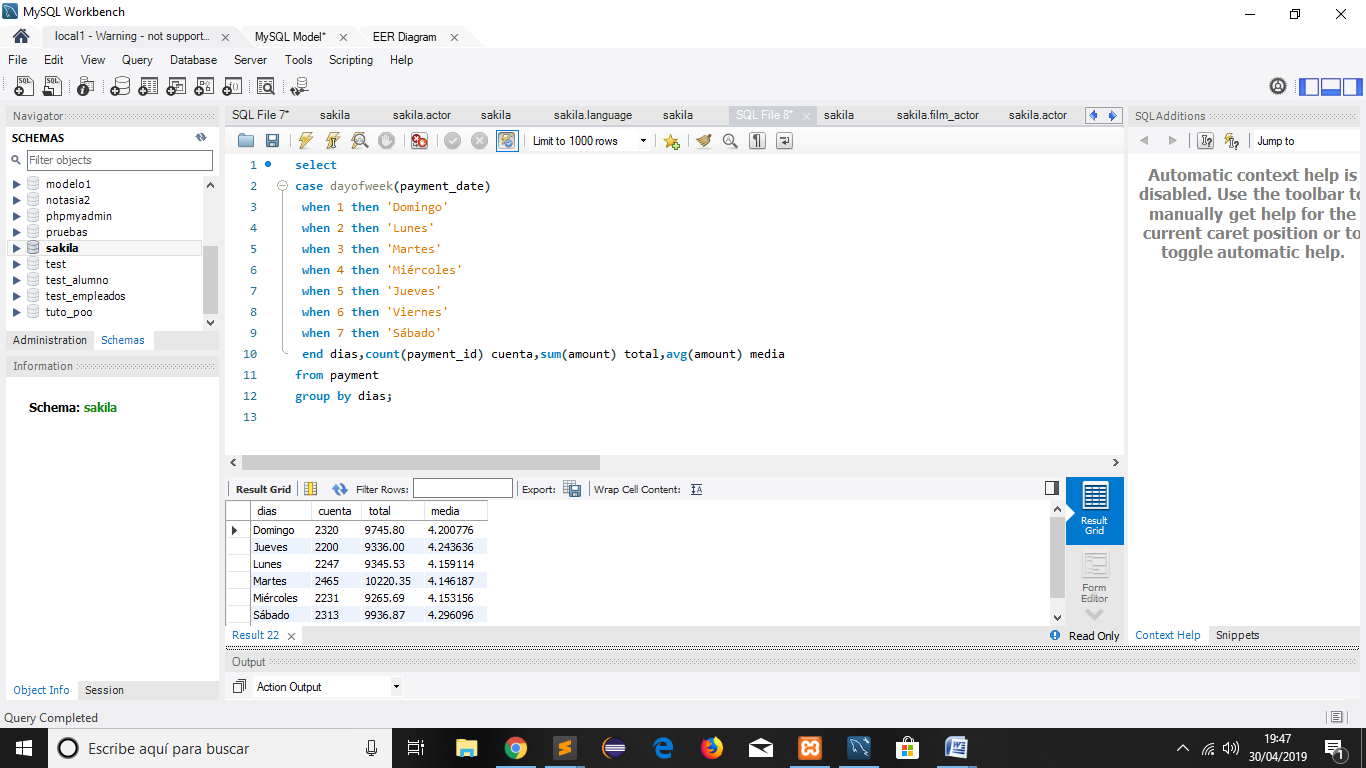
when 6 then 'Viernes'

when 7 then 'Sábado'

end dias, count(payment\_id) cuenta, sum(amount) total, avg(amount) media

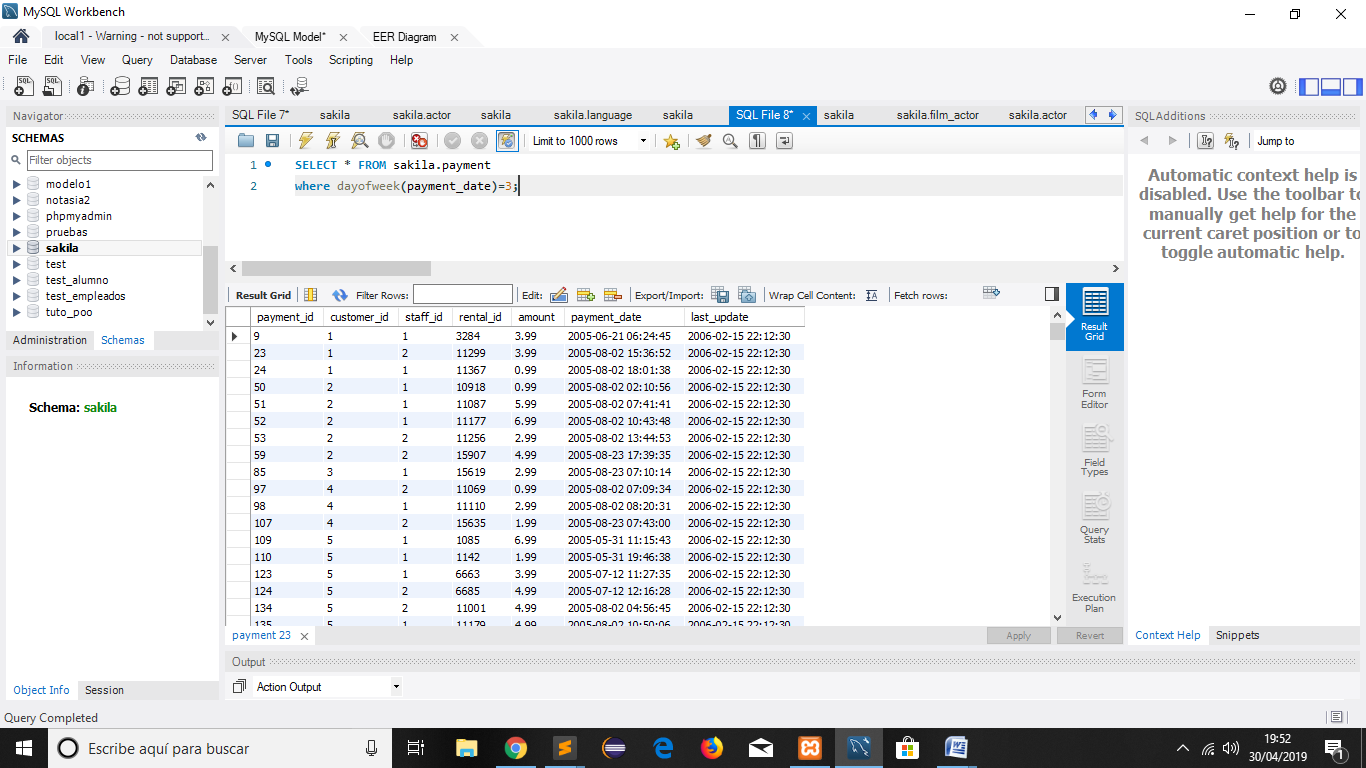
from payment

group by dias;

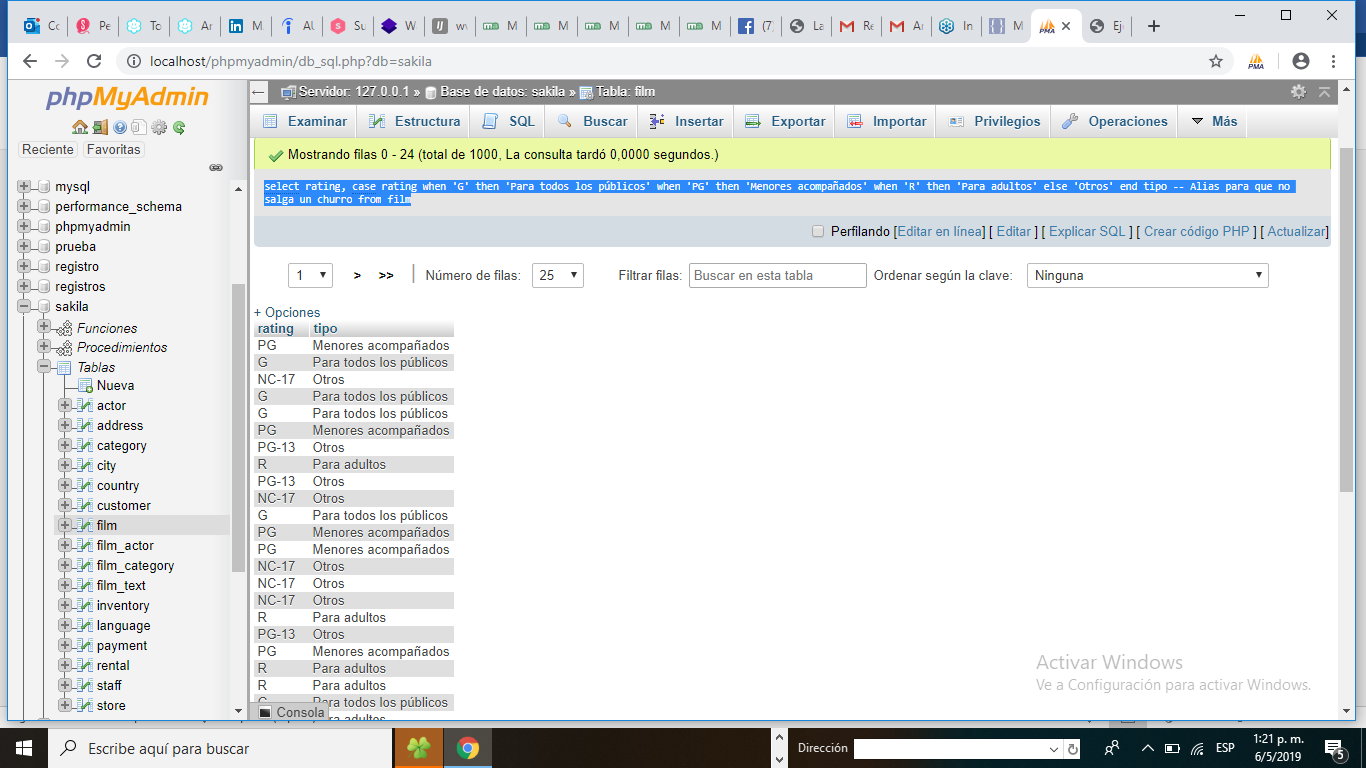


5.2 SELECT \* FROM sakila.payment

WHERE dayofweek (payment\_date)=3;



5.3 [select](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) rating, [case](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/control-flow-functions.html#operator_case) rating when 'G' then 'Para todos los públicos' when 'PG' then 'Menores acompañados' when 'R' then 'Para adultos' else 'Otros' end tipo – Alias  from film



5.4 Select concat(first\_name,' ',last\_name) cliente,

concat\_ws(' - ',first\_name, last\_name,email) nombre ,

length(first\_name) longitud,

left(first\_name,2) izq,

right(first\_name,2) der,

locate('@',email) pos,

substr(last\_name,4,2) subcadena,

substr(email,1,locate('@',email)-1) lmail,

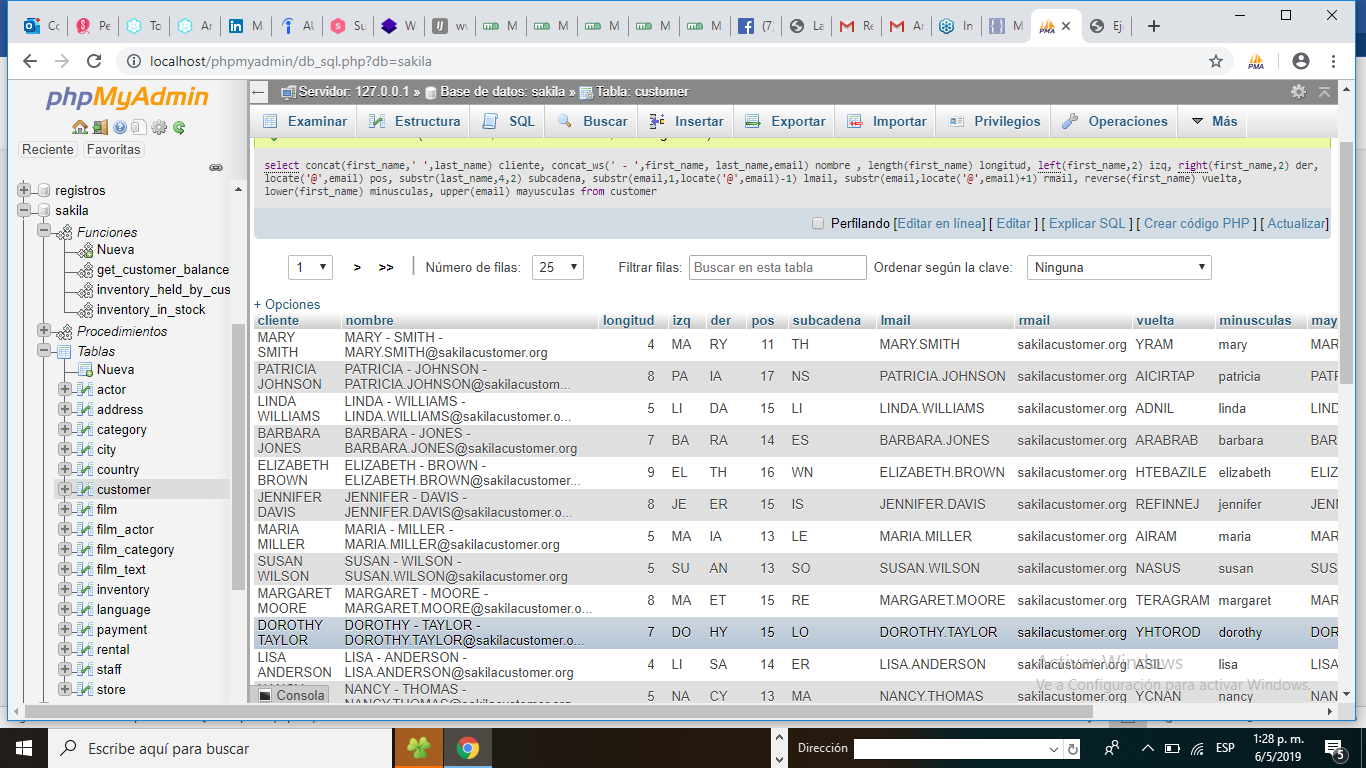
substr(email,locate('@',email)+1) rmail,

reverse(first\_name) vuelta,

lower(first\_name)  minusculas,

upper(email) mayusculas

from customer



**5.5 FUNCIONES**

FUNCTION inventory\_in\_stock (p\_inventory\_id( INT ))

BEGIN DECLARE v\_rentals INT;

DECLARE v\_out INT;

SELECT COUNT(\*) INTO v\_rentals

FROM rental

WHERE inventory\_id = p\_inventory\_id;

IF v\_rentals = 0 THEN

RETURN TRUE;

END IF;

SELECT COUNT(rental\_id) INTO v\_out

FROM inventory LEFT JOIN rental USING(inventory\_id)

WHERE inventory.inventory\_id = p\_inventory\_id

AND rental.return\_date IS NULL;

IF v\_out > 0 THEN

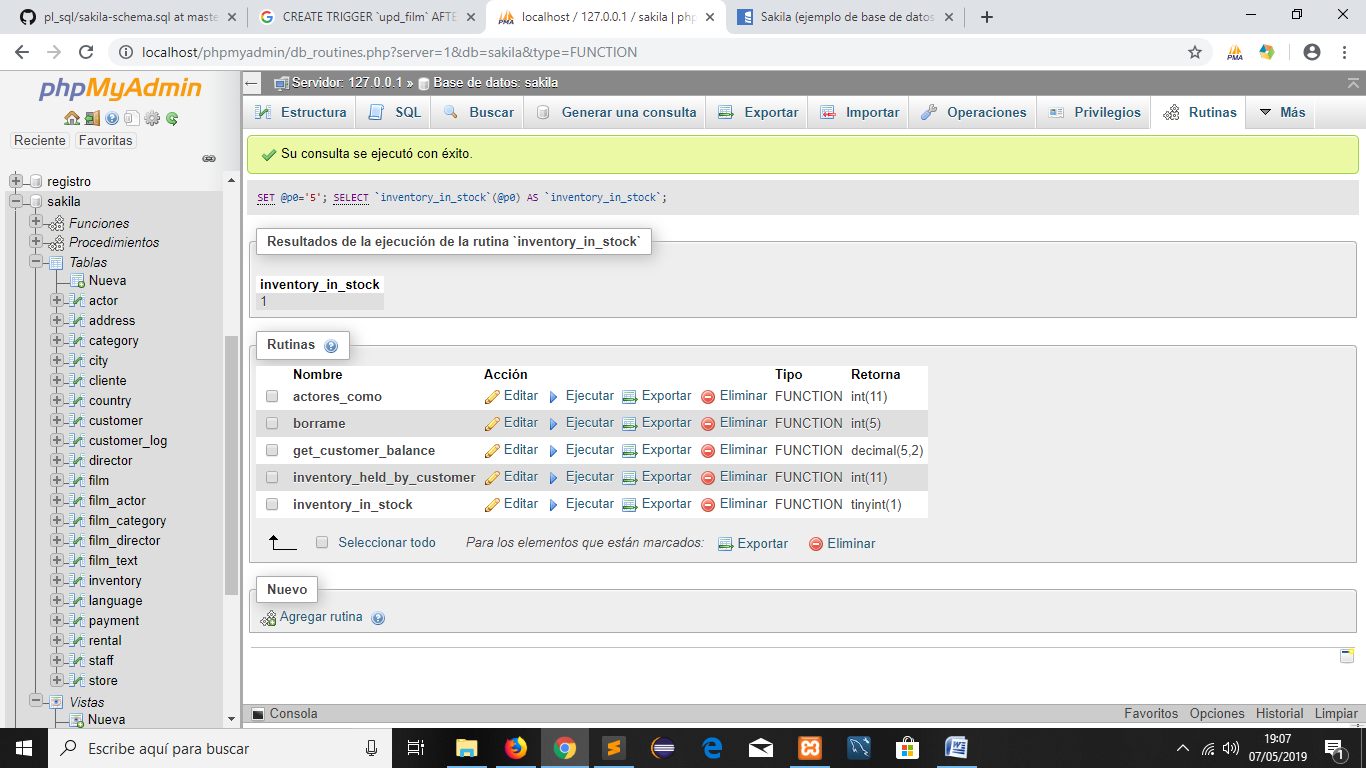
RETURN FALSE;

ELSE

RETURN TRUE;

END IF;

END



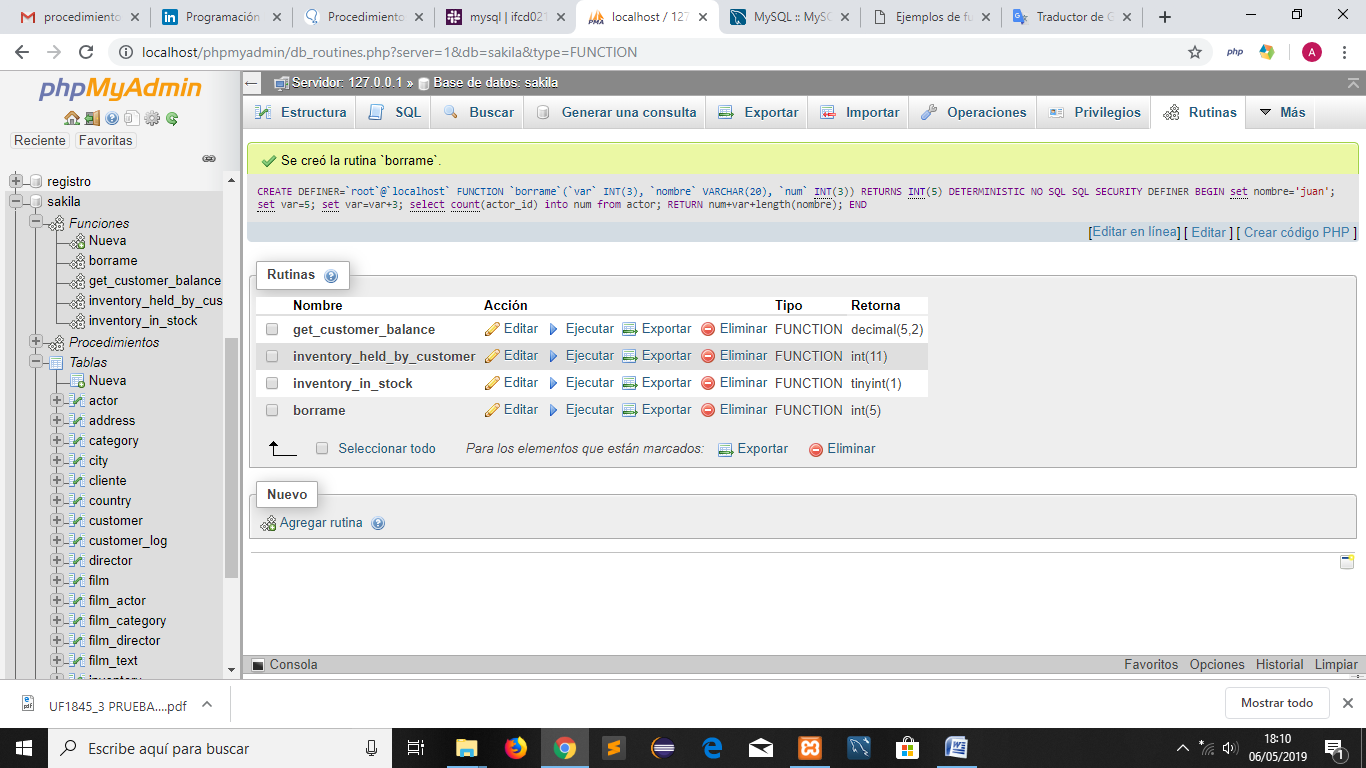
5.6

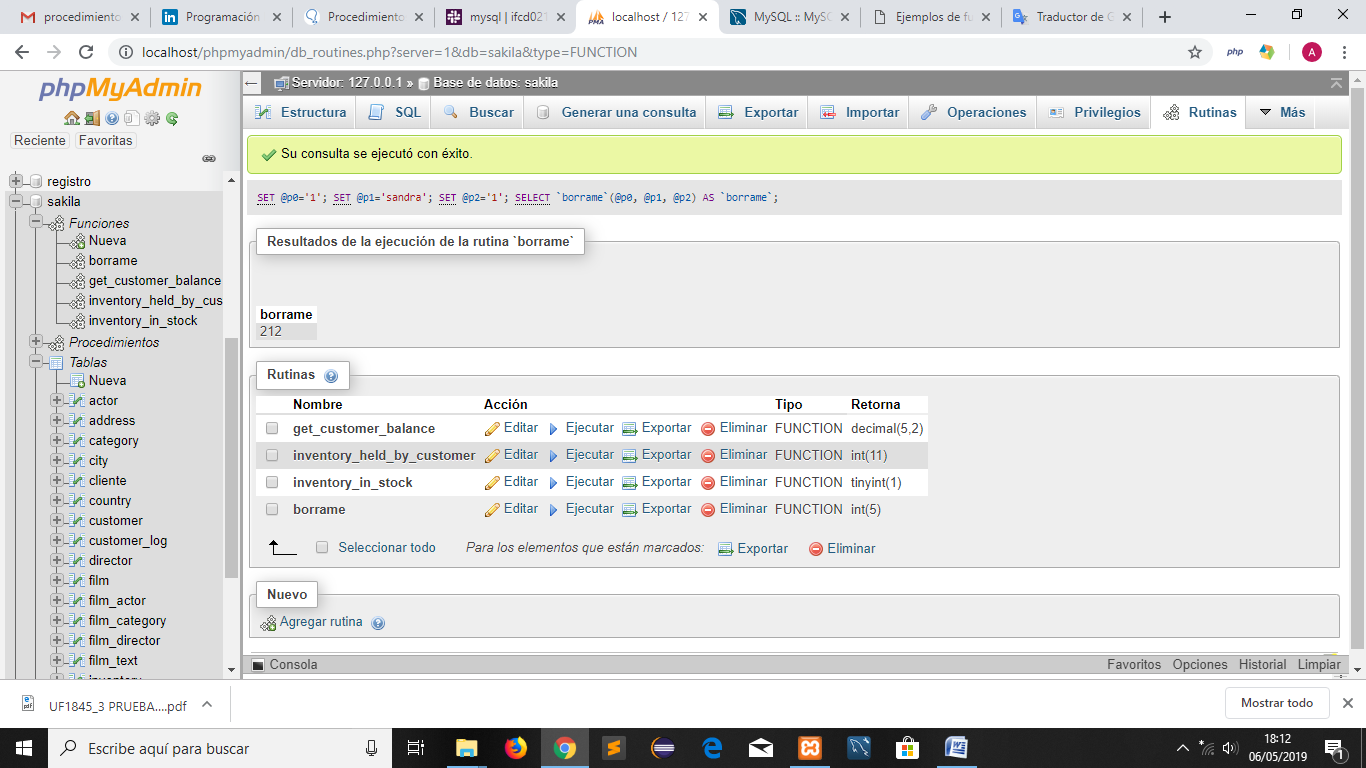
CREATE DEFINER=`root`@`localhost` FUNCTION `borrame`(`var` INT(3), `nombre` VARCHAR(20), `num` [INT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/numeric-types.html)(3)) RETURNS [INT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/numeric-types.html)(5) DETERMINISTIC NO SQL SQL SECURITY DEFINER

BEGIN [set](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/set.html) nombre='juan';[set](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/set.html) var=5; [set](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/set.html) var=var+3;

[select](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html)  [count](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html#function_count)(actor\_id) into num from actor;

RETURN num+var+length(nombre); END

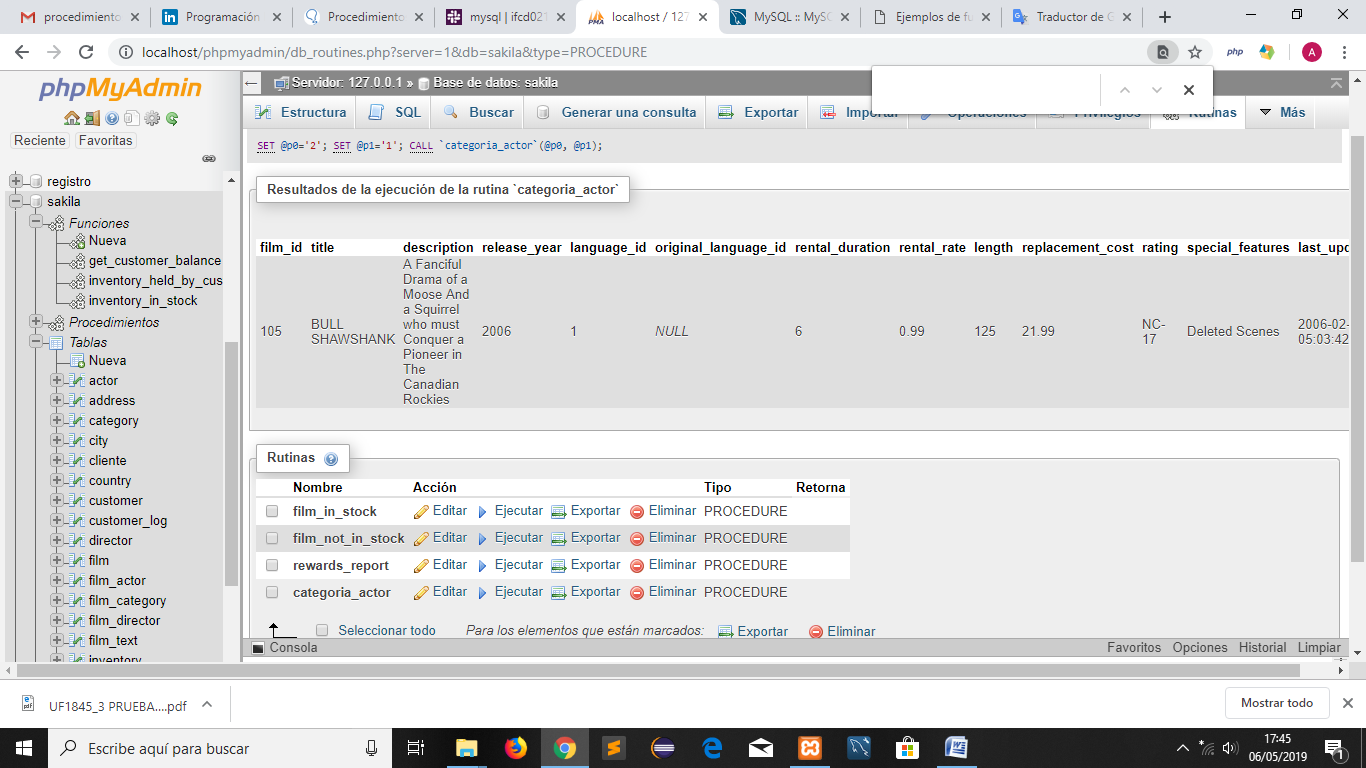




**6.- CREAR UN PROCEDIMIENTO**

Procedimiento que visualiza  todos los actores que hayan trabajado en películas cuya categoría sea la indicada en la consulta (Acción).

6.1 CREATE DEFINER=`root`@`localhost` PROCEDURE `categoria\_actor`() [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_not) DETERMINISTIC NO SQL SQL SECURITY DEFINER BEGIN [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) \* FROM film WHERE film\_id [in](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/comparison-operators.html#function_in) ([SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) film\_id FROM film\_actor WHERE actor\_id=actor) [and](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_and) film\_id [in](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/comparison-operators.html#function_in)  ([select](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) film\_id  from film\_category  where category\_id =categ); END



6.2 Procedimiento que visualiza  todas las películas cuyo costo de reemplazo sea superior  a un valor que se pasará como parámetro de entrada. ¿Cuántas películas tienen un costo de reemplazo superior a 10€?

CREATE DEFINER=`root`@`localhost ` PROCEDURE `costes`(IN `coste` [INT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/numeric-types.html)(11)) [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_not) DETERMINISTIC NO SQL SQL SECURITY DEFINER BEGIN [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) \* FROM film WHERE replacement\_cost > coste; END

