|  |  |  |
| --- | --- | --- |
|  | | |
|  | **Hurlingham**, Agosto del 2021  Buenos Aires, Argentina | |
| **ESTRATEGIAS DE PERSISTENCIA (759\_45)**  **TP\_CLASE2\_PRACTICA**  Profesor: Lic. Pablo Marcelli  Alumno: Raul A. Gutierrez  correo electronico: [correo@raulgutierrez.com.ar](mailto:correo@raulgutierrez.com.ar)  [raul.gutierrez@estudiantes.unahur.edu.ar](mailto:raul.gutierrez@estudiantes.unahur.edu.ar)  Entrega: Lunes 23 de Agosto 2021 | |  |

**Construccion Tabla 1**

CREATE DATABASE IF NOT EXISTS `deposito`;

USE DATABASE `deposito`;

CREATE TABLE IF NOT EXISTS `FABRICANTES` (

`Codigo` INT NOT NULL,

`Nombre` VARCHAR(100),

PRIMARY KEY (`Codigo`)

);

CREATE TABLE IF NOT EXISTS `ARTICULOS` (

`Codigo` INT NOT NULL,

`Nombre` VARCHAR(100),

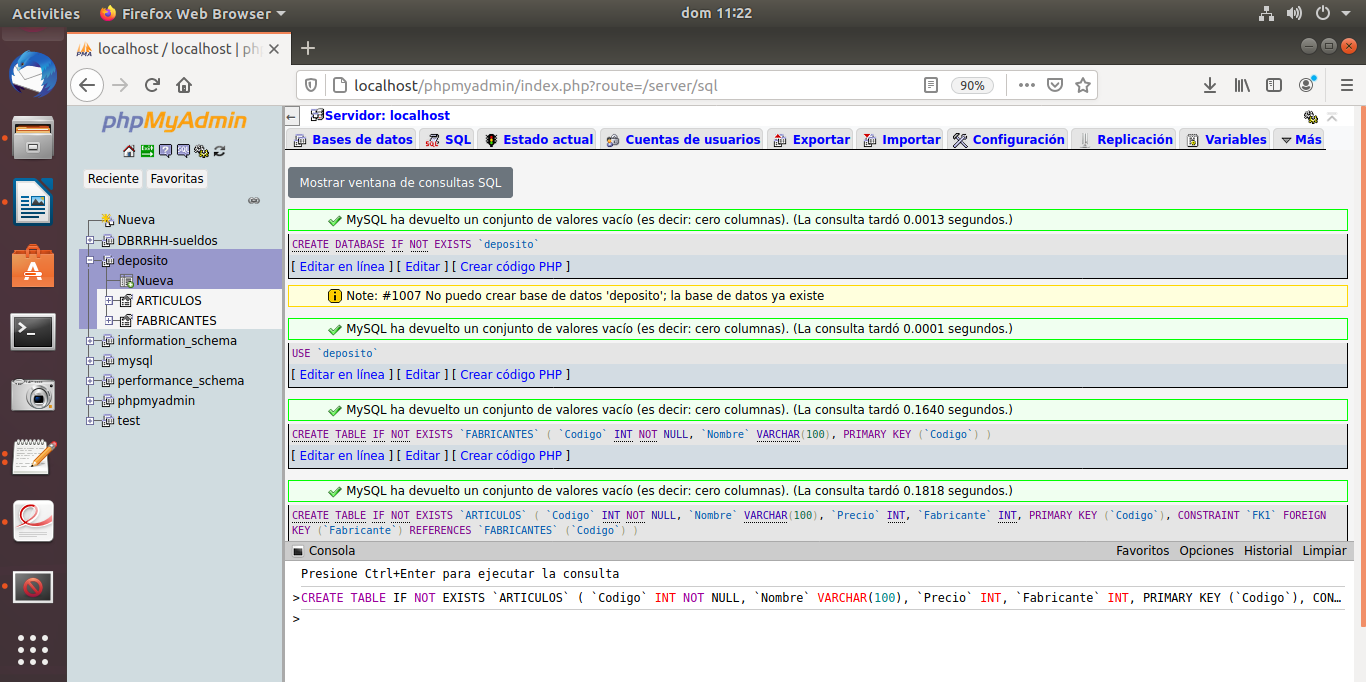
`Precio` INT,

`Fabricante` INT,

PRIMARY KEY (`Codigo`),

CONSTRAINT `FK1` FOREIGN KEY (`Fabricante`) REFERENCES `FABRICANTES` (`Codigo`)

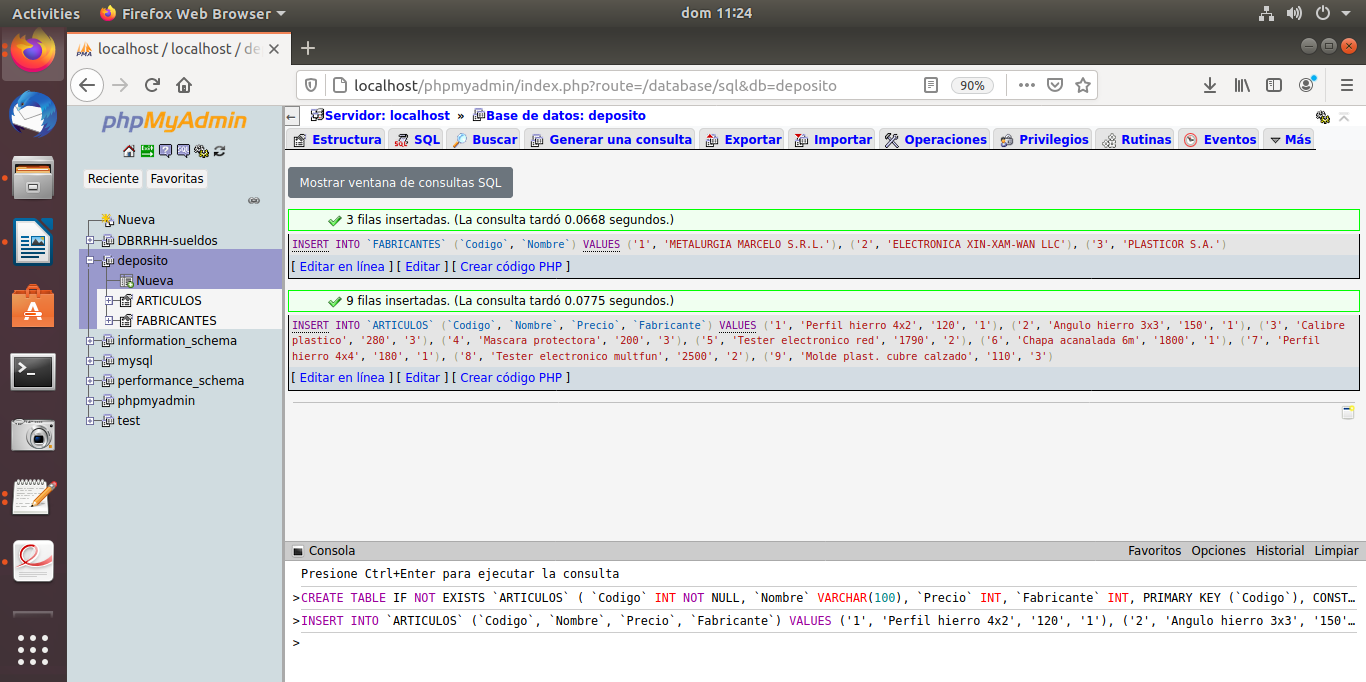
);



Agregando datos tabla1

INSERT INTO `FABRICANTES` (`Codigo`, `Nombre`) VALUES ('1', 'METALURGIA MARCELO S.R.L.'), ('2', 'ELECTRONICA XIN-XAM-WAN LLC'), ('3', 'PLASTICOR S.A.');

INSERT INTO `ARTICULOS` (`Codigo`, `Nombre`, `Precio`, `Fabricante`) VALUES ('1', 'Perfil hierro 4x2', '120', '1'), ('2', 'Angulo hierro 3x3', '150', '1'), ('3', 'Calibre plastico', '280', '3'), ('4', 'Mascara protectora', '200', '3'), ('5', 'Tester electronico red', '1790', '2'), ('6', 'Chapa acanalada 6m', '1800', '1'), ('7', 'Perfil hierro 4x4', '180', '1'), ('8', 'Tester electronico multfun', '2500', '2'), ('9', 'Molde plast. cubre calzado', '110', '3');



**Construccion Tabla 2**

CREATE DATABASE IF NOT EXISTS `departamentos`;

USE `departamentos`;

CREATE TABLE IF NOT EXISTS `DEPARTAMENTOS` (

`Codigo` INT NOT NULL,

`Nombre` VARCHAR(100),

`Presupuesto` INT,

PRIMARY KEY (`Codigo`)

);

CREATE TABLE IF NOT EXISTS `EMPLEADOS` (

`DNI` VARCHAR(8) NOT NULL,

`Nombre` VARCHAR(100),

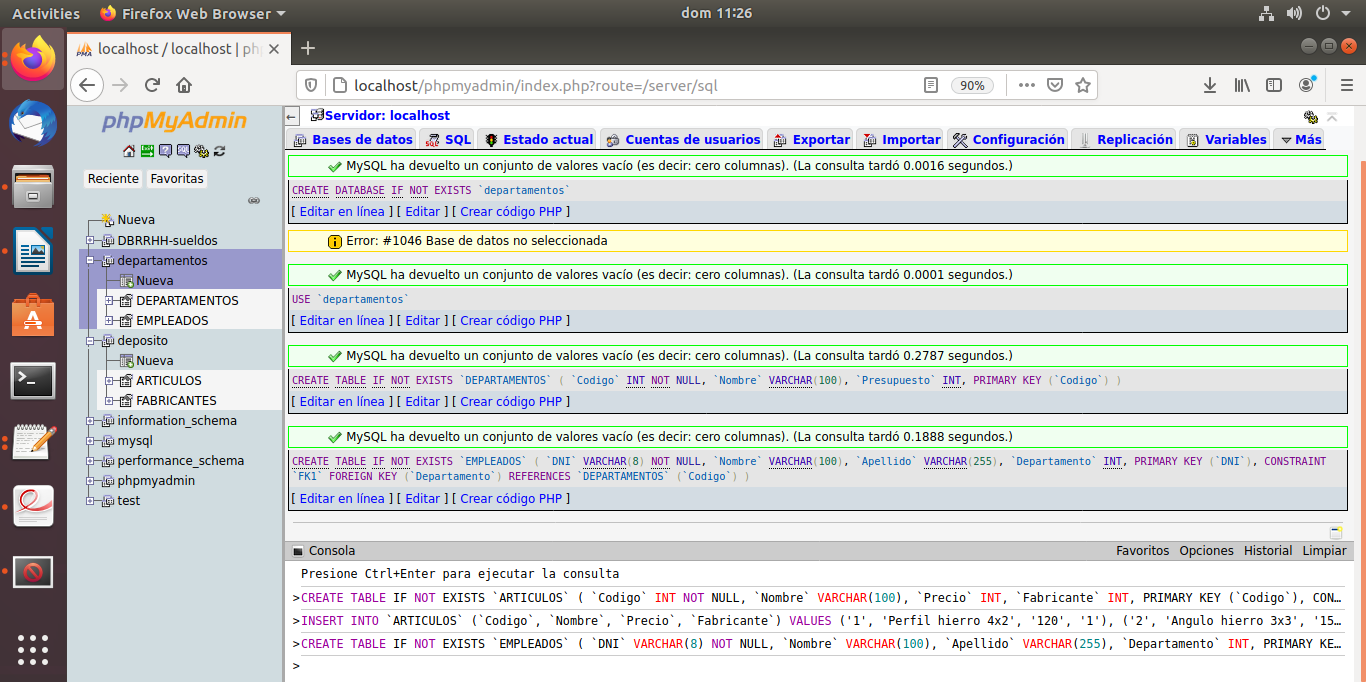
`Apellido` VARCHAR(255),

`Departamento` INT,

PRIMARY KEY (`DNI`),

CONSTRAINT `FK1` FOREIGN KEY (`Departamento`) REFERENCES `DEPARTAMENTOS` (`Codigo`)

);



Agregando datos Tabla 2

INSERT INTO `DEPARTAMENTOS` (`Codigo`, `Nombre`, `Presupuesto`)

VALUES

('1', 'INFRAESTRUCTURA', '195000'),

('2', 'DESARROLLO','25000'),

('3', 'DISEÑO', '55000'),

('4', 'ADMINISTRACION', '45000');

INSERT INTO `EMPLEADOS` (`DNI`, `Nombre`, `Apellido`, `Departamento`)

VALUES

('28657983', 'Gabriel', 'Vega', '3'),

('34453984', 'Angel', 'Barrientos', '2'),

('27342904', 'Robert', 'Roja', '1'),

('44345235', 'Marcela', 'Lopez', '3'),

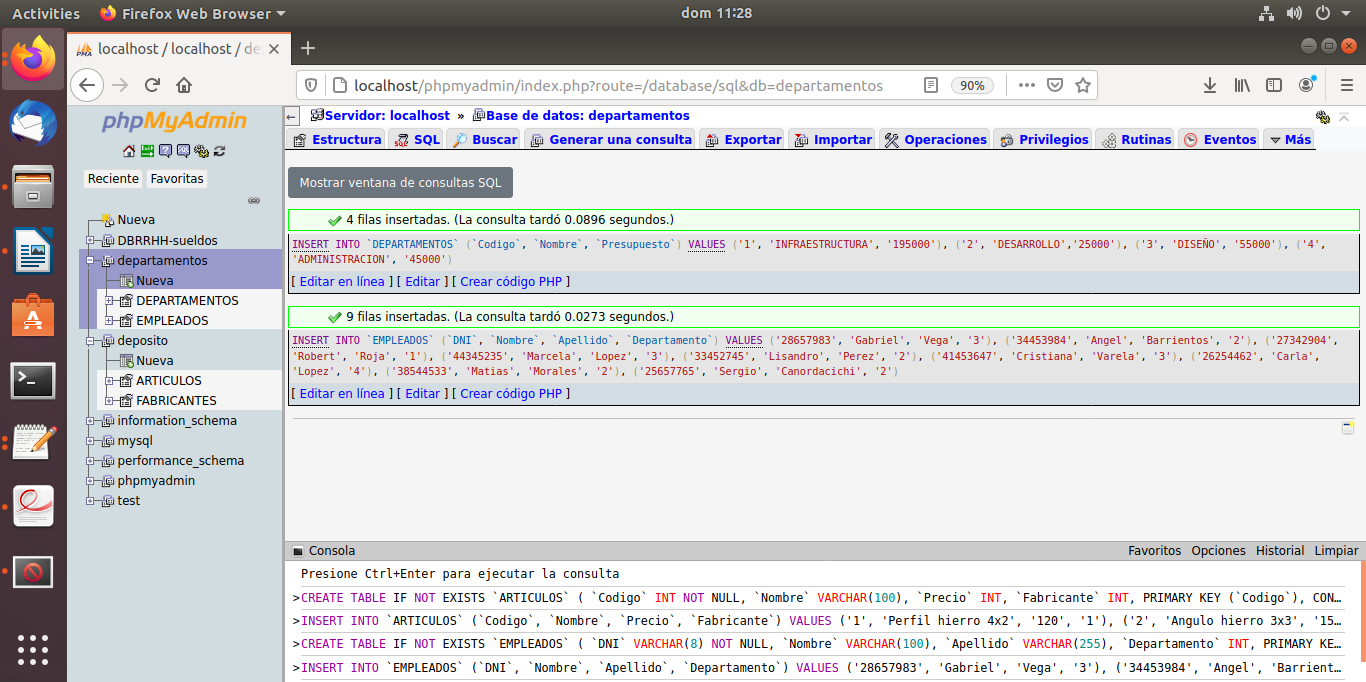
('33452745', 'Lisandro', 'Perez', '2'),

('41453647', 'Cristiana', 'Varela', '3'),

('26254462', 'Carla', 'Lopez', '4'),

('38544533', 'Matias', 'Morales', '2'),

('25657765', 'Sergio', 'Canordacichi', '2');



**Construccion Tabla 3**

CREATE DATABASE IF NOT EXISTS `almacen`;

USE `almacen`;

CREATE TABLE IF NOT EXISTS `ALMACENES` (

`Codigo` INT NOT NULL,

`Lugar` VARCHAR(100),

`Capacidad` INT,

PRIMARY KEY (`Codigo`)

);

CREATE TABLE IF NOT EXISTS `CAJAS` (

`NumRefencia` CHAR(5),

`Contenido` VARCHAR(100),

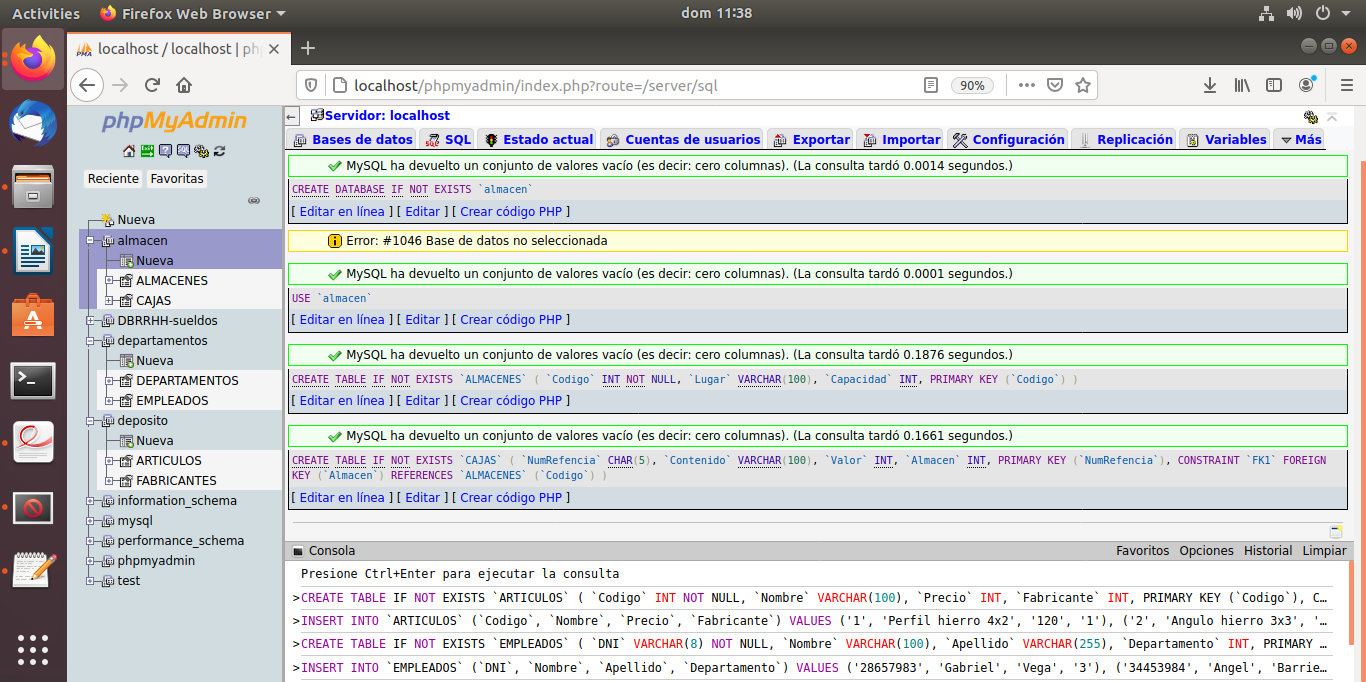
`Valor` INT,

`Almacen` INT,

PRIMARY KEY (`NumRefencia`),

CONSTRAINT `FK1` FOREIGN KEY (`Almacen`) REFERENCES `ALMACENES` (`Codigo`)

);



Agregando datos Tabla 2

INSERT INTO `ALMACENES` (`Codigo`, `Lugar`, `Capacidad`)

VALUES

('1', 'Pilar', '5'),

('2', 'Villa Crespo', '3'),

('3', 'Rosario', '10');

INSERT INTO `CAJAS` (`NumRefencia`, `Contenido`, `Valor`, `Almacen`)

VALUES

('00001', 'Huevos', '220', '2'),

('00002', 'Repuestos para Bicis', '7500', '1'),

('00003', 'Componentes electronicos', '5000', '3'),

('00004', 'Mascaras protectoras', '1200', '3'),

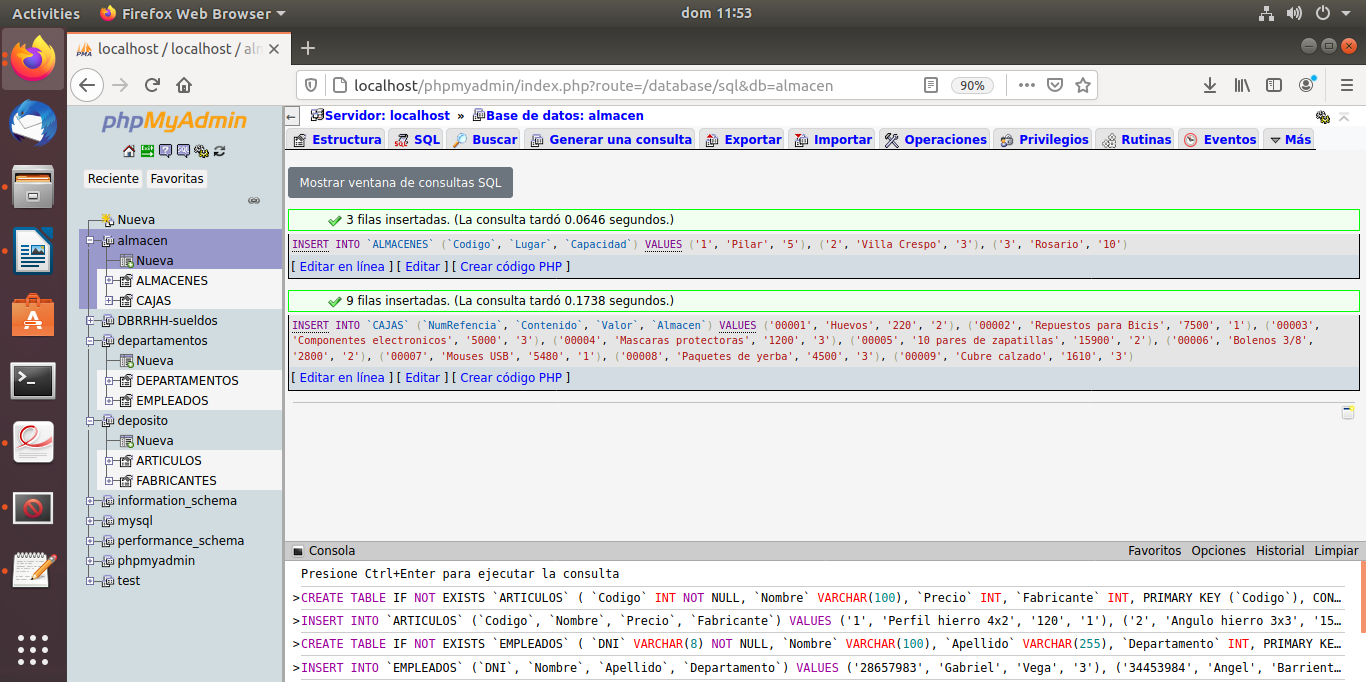
('00005', '10 pares de zapatillas', '15900', '2'),

('00006', 'Bolenos 3/8', '2800', '2'),

('00007', 'Mouses USB', '5480', '1'),

('00008', 'Paquetes de yerba', '4500', '3'),

('00009', 'Cubre calzado', '1610', '3');



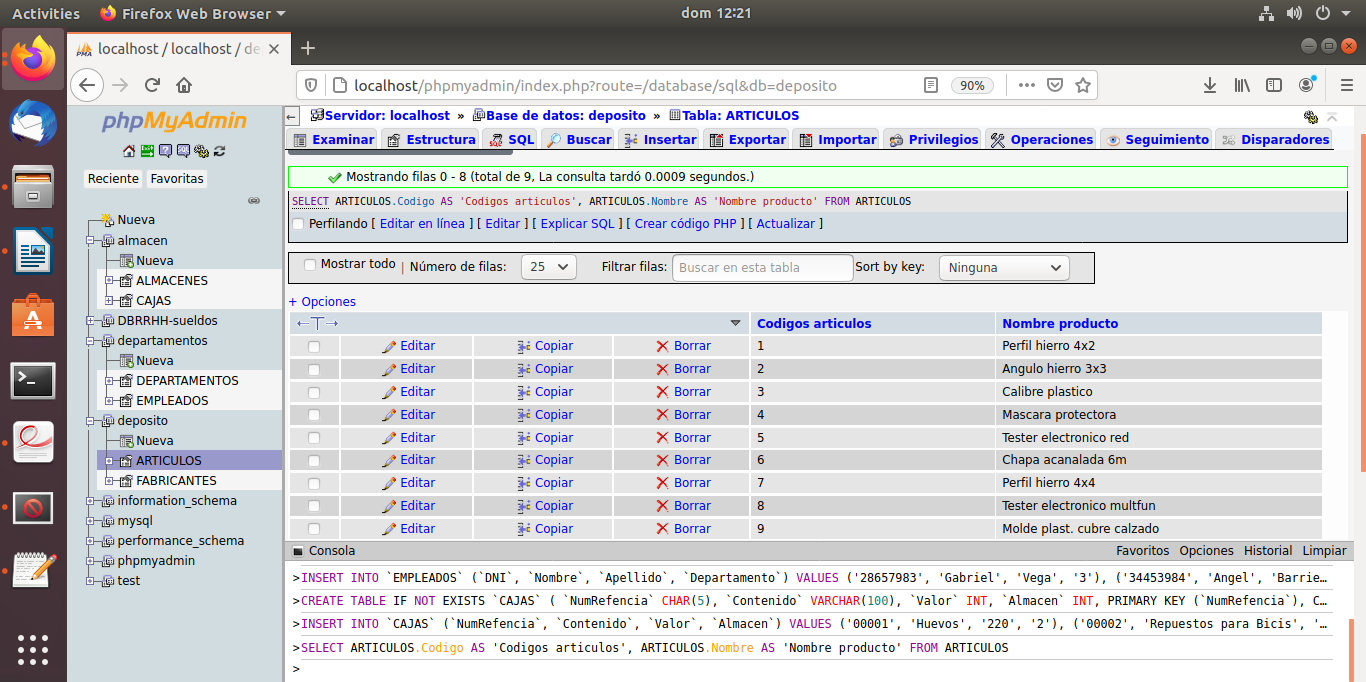
**Ejercicios**

/\* 1.- Obtener los nombres de los productos de la tienda. \*/

SELECT ARTICULOS.Codigo AS 'Codigos articulos',

ARTICULOS.Nombre AS 'Nombre producto'

FROM ARTICULOS



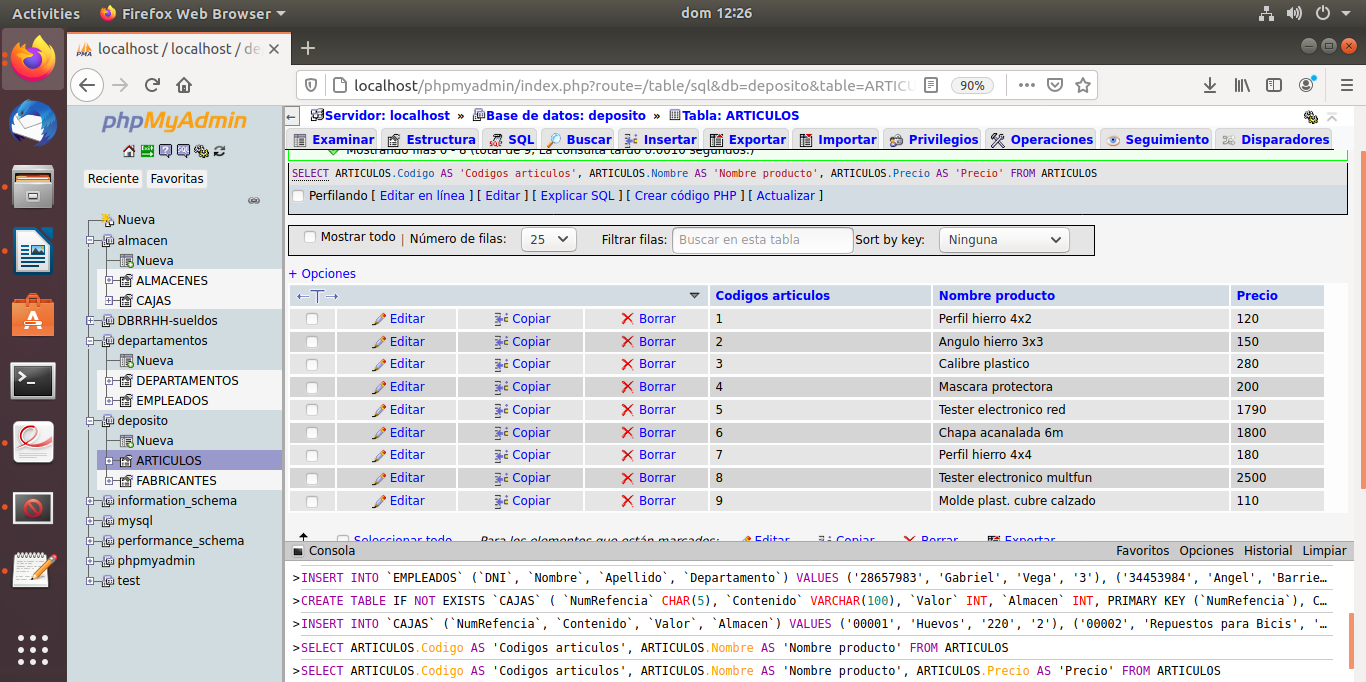
/\* 2.- Obtener los nombres y los precios de los productos de la tienda. \*/

SELECT ARTICULOS.Codigo AS 'Codigos articulos',

ARTICULOS.Nombre AS 'Nombre producto',

ARTICULOS.Precio AS 'Precio'

FROM ARTICULOS



/\* 3.- Obtener el nombre de los productos cuyo precio sea menor o igual a 200. \*/

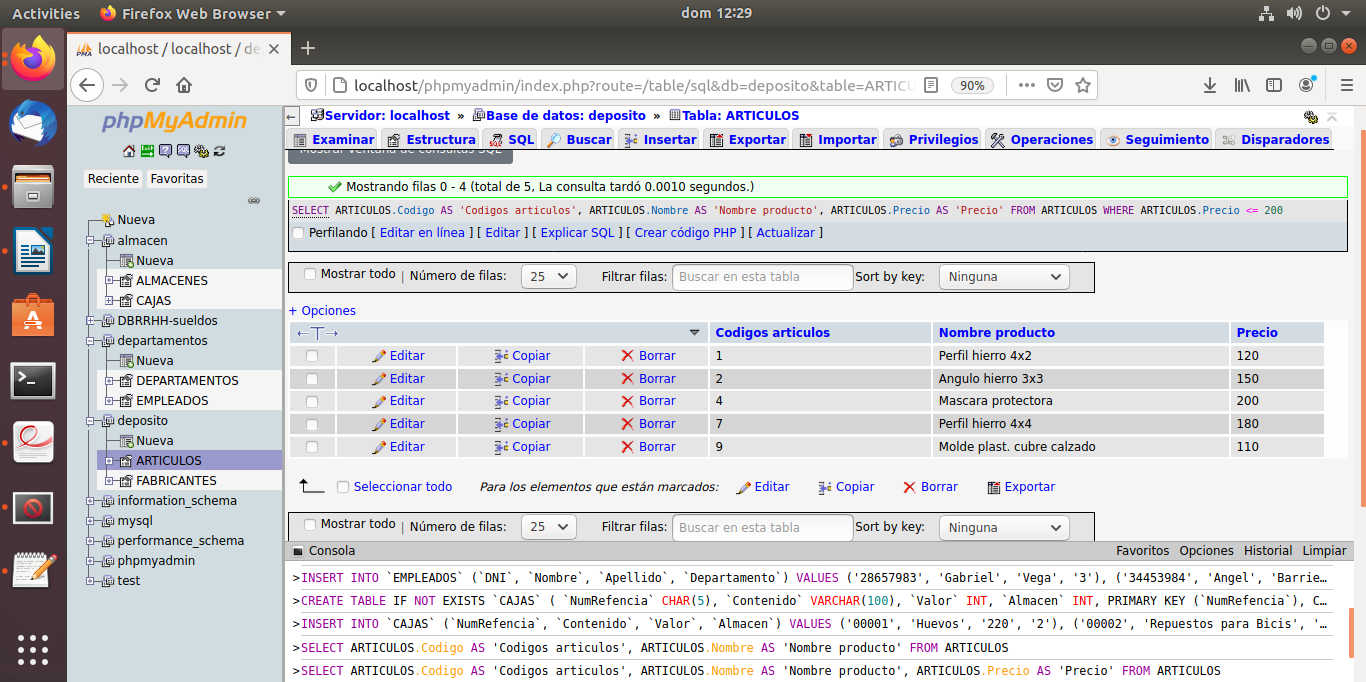
SELECT ARTICULOS.Codigo AS 'Codigos articulos',

ARTICULOS.Nombre AS 'Nombre producto',

ARTICULOS.Precio AS 'Precio'

FROM ARTICULOS

WHERE ARTICULOS.Precio <= 200



/\*4. Obtener todos los datos de los artículos cuyo precio esté entre los 60 $ y los 120 $

(ambas cantidades incluidas). \*/

Nota: Se agregan articulos para mejorar el resultado.

INSERT INTO `ARTICULOS` (`Codigo`, `Nombre`, `Precio`, `Fabricante`)

VALUES

('10', 'Perfil hierro 2x2', '80', '1'),

('11', 'Angulo hierro 1x1', '50', '1'),

('12', 'Regla plastica', '40', '3'),

('13', 'Barbijo', '60', '3')

;

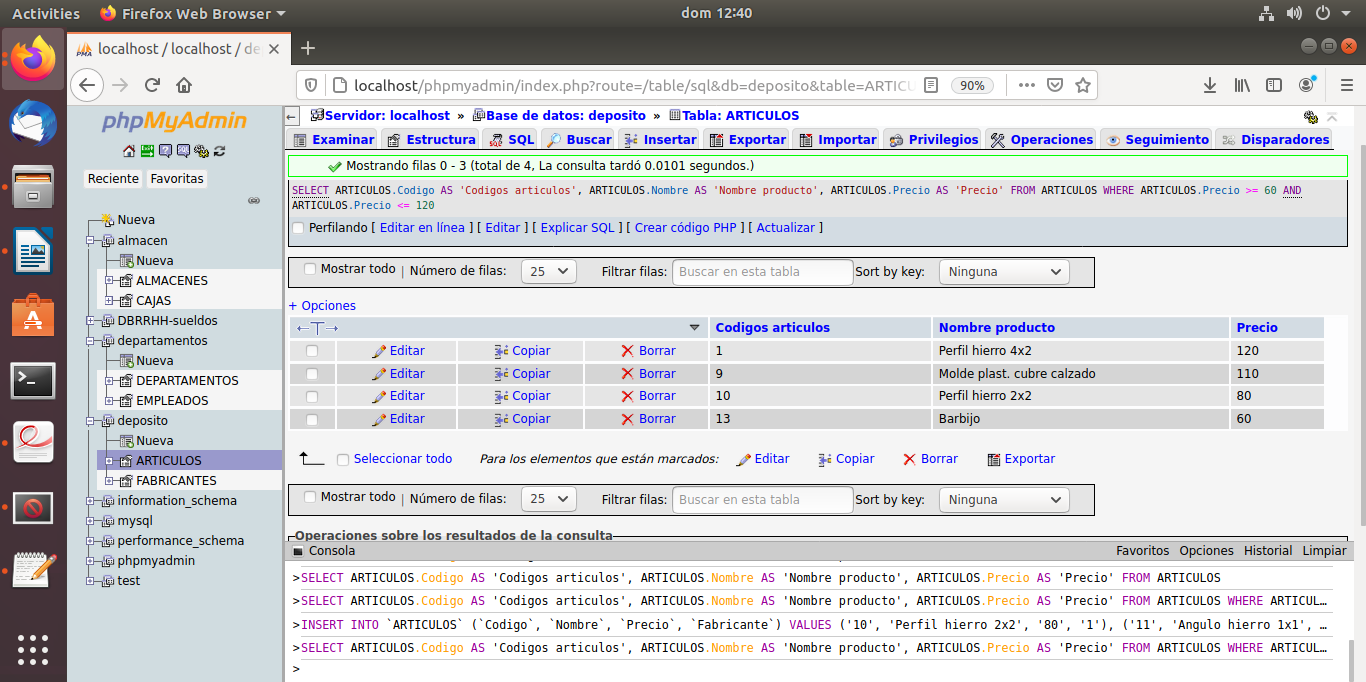
SELECT ARTICULOS.Codigo AS 'Codigos articulos',

ARTICULOS.Nombre AS 'Nombre producto',

ARTICULOS.Precio AS 'Precio'

FROM ARTICULOS

WHERE ARTICULOS.Precio >= 60 AND ARTICULOS.Precio <= 120

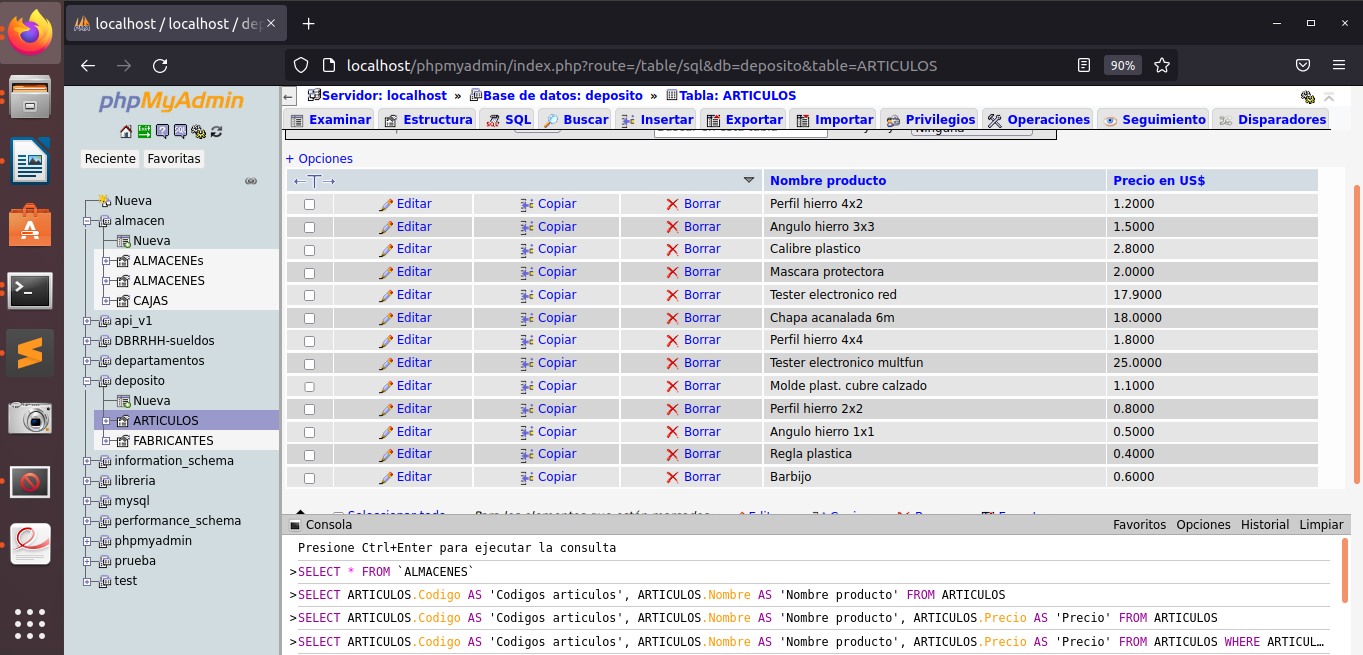


/\* 5. Obtener el nombre y el precio en dolares (es decir, el precio multiplicado por 85). \*/

SELECT ARTICULOS.Nombre AS 'Nombre producto',

ARTICULOS.Precio / 100 AS 'Precio en US$'

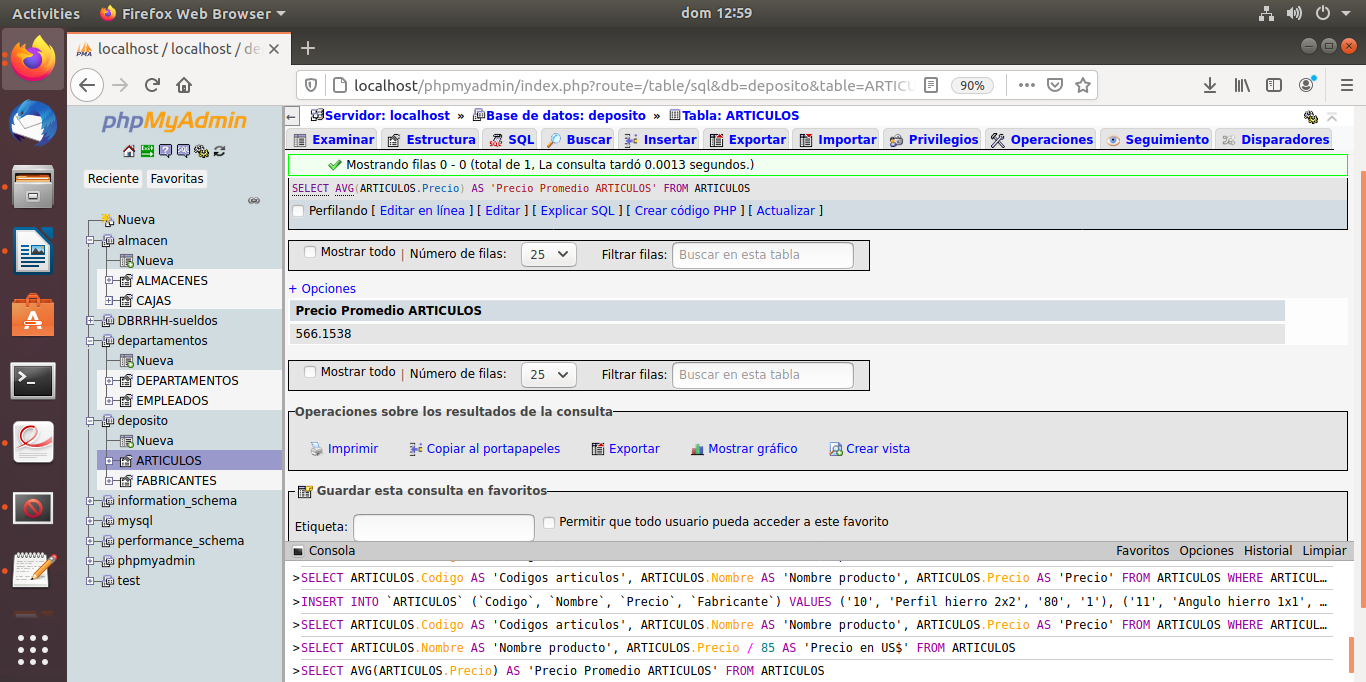
FROM ARTICULOS



/\* 6. Seleccionar el precio medio de todos los productos. \*/

SELECT AVG(ARTICULOS.Precio) AS 'Precio Promedio ARTICULOS'

FROM ARTICULOS



/\* 7. Obtener un listado de artículos, incluyendo el nombre del artículo, su precio, y el

nombre de su fabricante. \*/

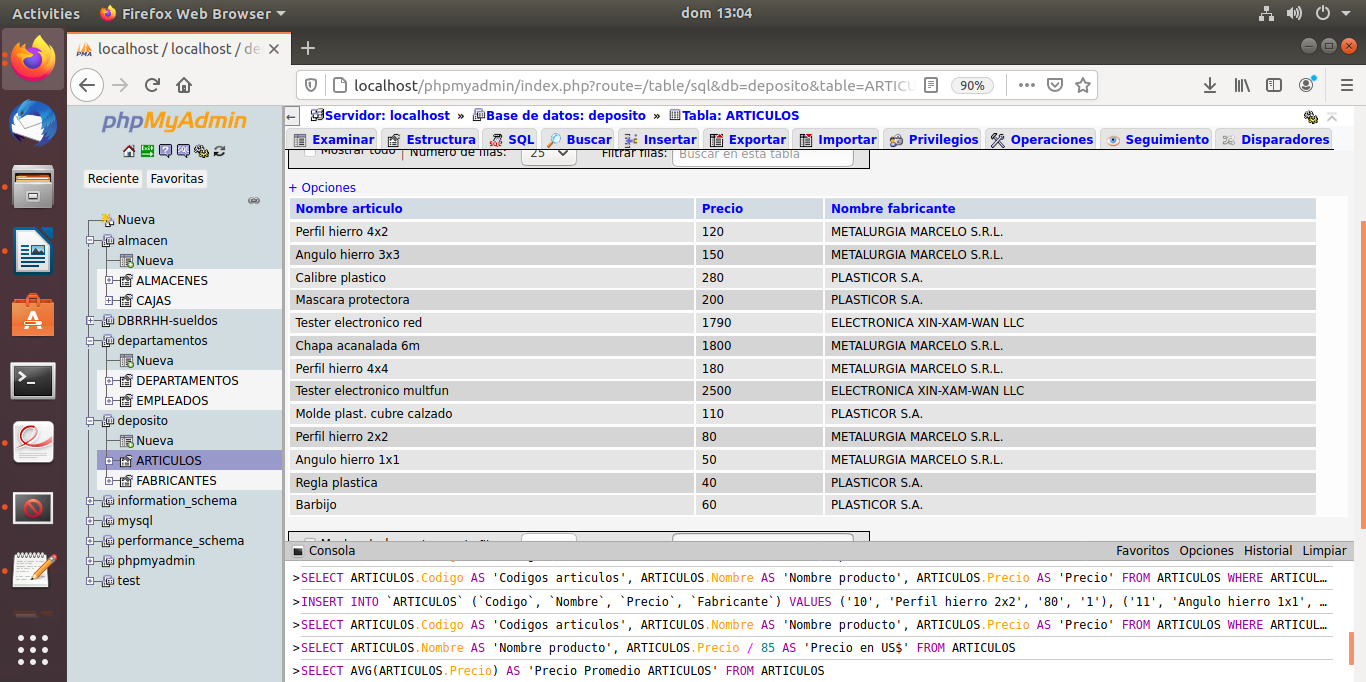
SELECT ARTICULOS.Nombre AS 'Nombre articulo',

ARTICULOS.Precio AS 'Precio',

FABRICANTES.Nombre AS 'Nombre fabricante'

FROM ARTICULOS, FABRICANTES

WHERE ARTICULOS.Fabricante = FABRICANTES.Codigo



/\* 8. Obtener todos los datos de los empleados que se apellidan Lopez y los que se apellidan Perez. \*/

SELECT EMPLEADOS.Nombre AS 'Nombre empleado',

EMPLEADOS.Apellido AS 'Apellido empleado',

EMPLEADOS.DNI AS 'DNI',

DEPARTAMENTOS.Nombre AS 'Departamento'

FROM EMPLEADOS

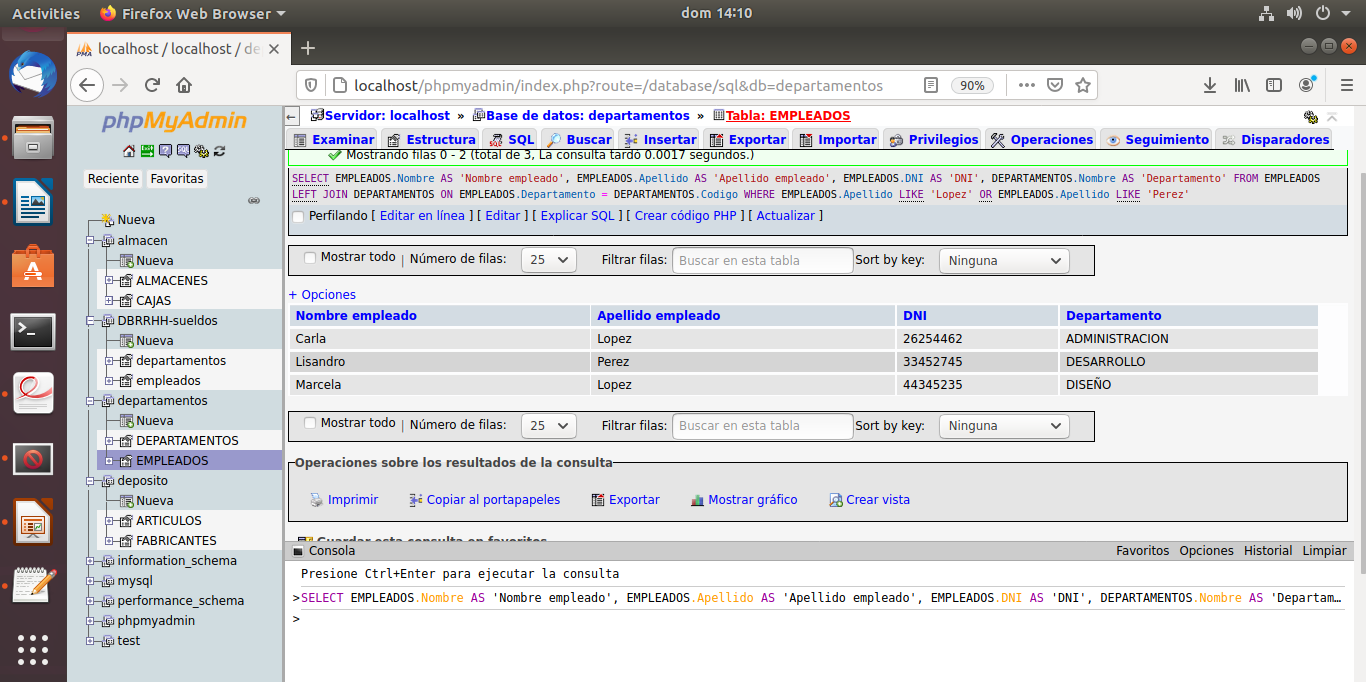
LEFT JOIN DEPARTAMENTOS ON EMPLEADOS.Departamento = DEPARTAMENTOS.Codigo

WHERE

EMPLEADOS.Apellido LIKE 'Lopez'

OR

EMPLEADOS.Apellido LIKE 'Perez'



/\* 9. Obtener todos los datos de los empleados que trabajan para el departamento 37 y para el departamento 77. \*/

Nota: se agregan registros para brindar resultado.

INSERT INTO `DEPARTAMENTOS` (`Codigo`, `Nombre`, `Presupuesto`)

VALUES

('37', 'COBRANZA', '28000'),

('77', 'VENTAS','65000');

INSERT INTO `EMPLEADOS` (`DNI`, `Nombre`, `Apellido`, `Departamento`)

VALUES

('32657983', 'Gabriela', 'Vega', '37'),

('38453984', 'Angela', 'Barrientos', '77'),

('38342904', 'Robertico', 'Vera Cruz de los Mares', '77')

;

SELECT EMPLEADOS.Nombre AS 'Nombre empleado',

EMPLEADOS.Apellido AS 'Apellido empleado',

EMPLEADOS.DNI AS 'DNI',

DEPARTAMENTOS.Nombre AS 'Departamento'

FROM EMPLEADOS

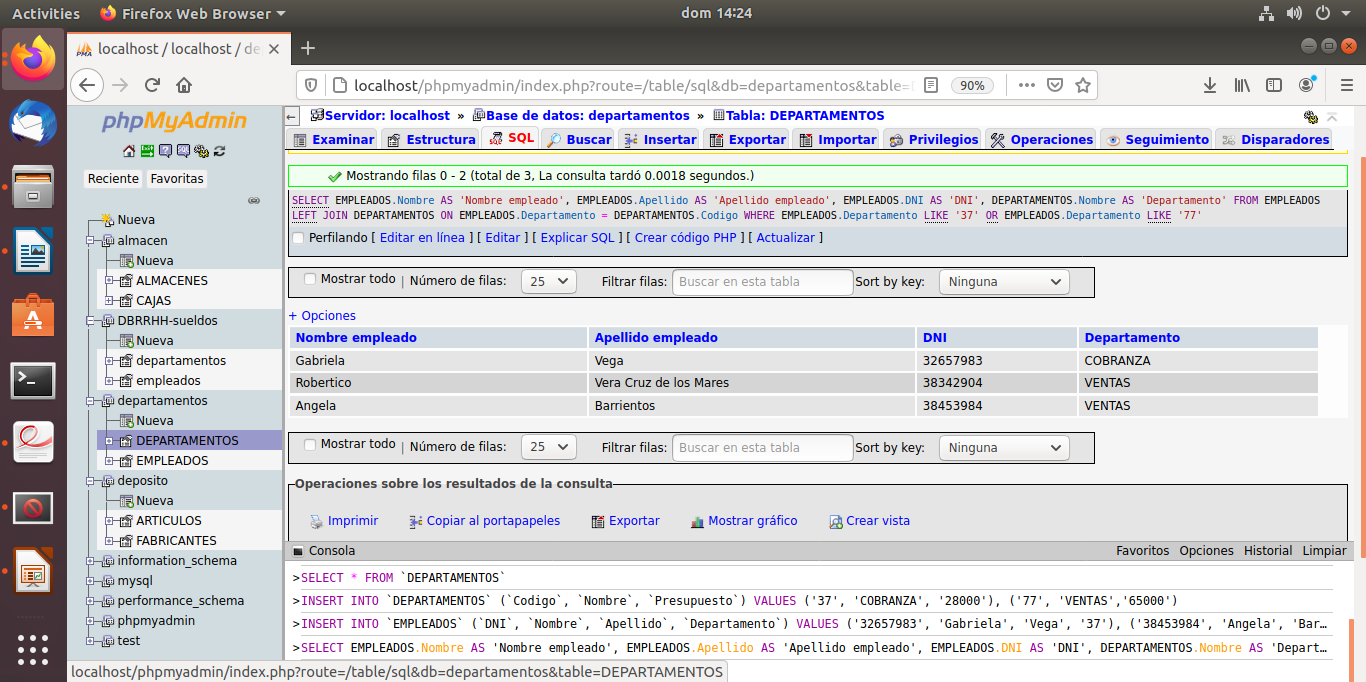
LEFT JOIN DEPARTAMENTOS ON EMPLEADOS.Departamento = DEPARTAMENTOS.Codigo

WHERE

EMPLEADOS.Departamento LIKE '37'

OR

EMPLEADOS.Departamento LIKE '77'



/\* 10. Obtener un listado completo de empleados, incluyendo el nombre y apellidos del empleado junto al nombre y presupuesto de su departamento. \*/

SELECT EMPLEADOS.Nombre AS 'Nombre empleado',

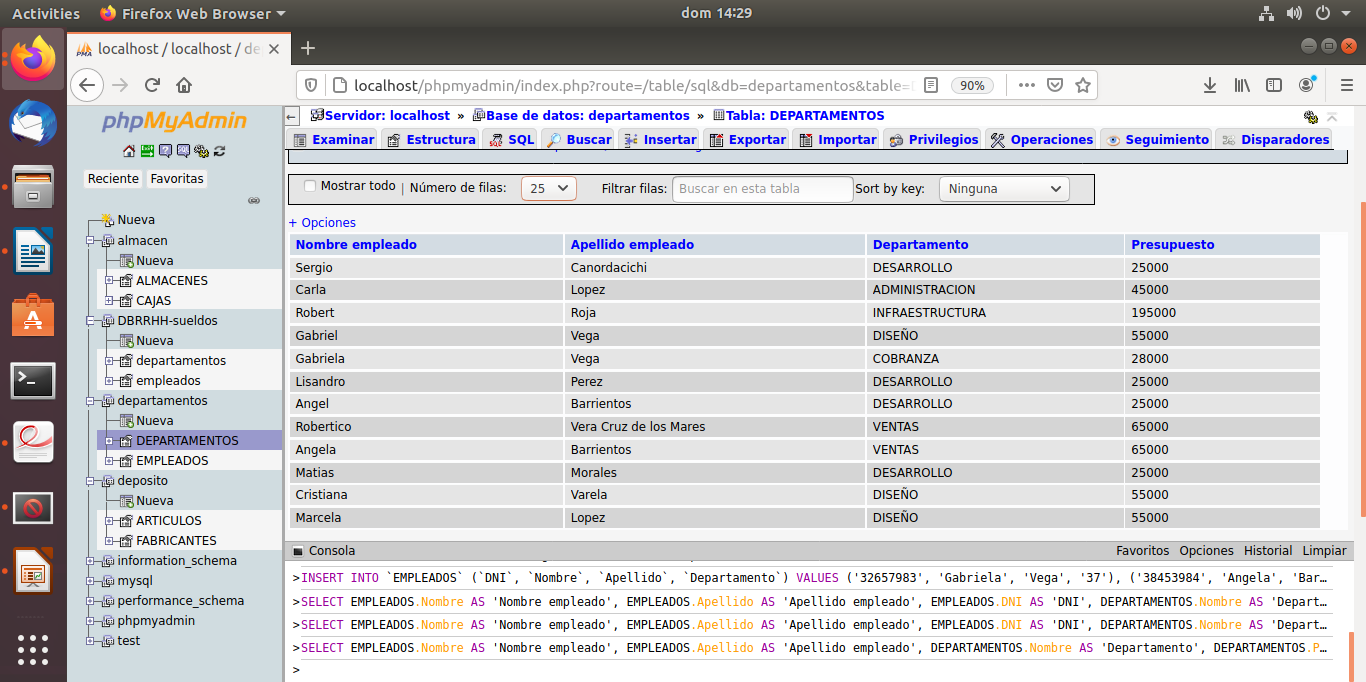
EMPLEADOS.Apellido AS 'Apellido empleado',

DEPARTAMENTOS.Nombre AS 'Departamento',

DEPARTAMENTOS.Presupuesto AS 'Presupuesto'

FROM EMPLEADOS

LEFT JOIN DEPARTAMENTOS ON EMPLEADOS.Departamento = DEPARTAMENTOS.Codigo



/\* 11. Obtener los nombres (unicamente los nombres) de los departamentos que tienen mas de dos empleados. \*/

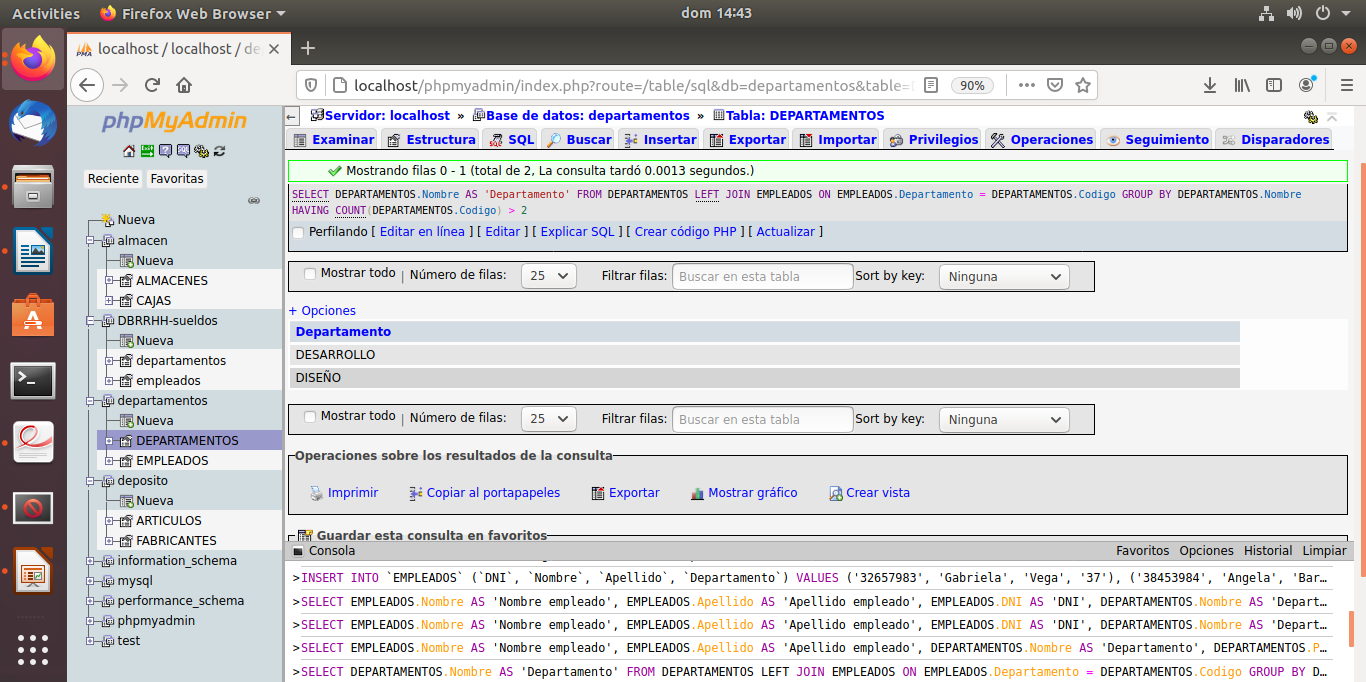
SELECT DEPARTAMENTOS.Nombre AS 'Departamento'

FROM DEPARTAMENTOS

LEFT JOIN EMPLEADOS ON EMPLEADOS.Departamento = DEPARTAMENTOS.Codigo

GROUP BY DEPARTAMENTOS.Nombre

HAVING COUNT(DEPARTAMENTOS.Codigo) > 2



/\* 12. Añadir un nuevo departamento: ‘Calidad’, con presupuesto de 40.000 $ y código. Añadir un empleado vinculado al departamento reci ́en creado: Esther Vazquez, DNI: 89267109 \*/

INSERT INTO `DEPARTAMENTOS` (`Codigo`, `Nombre`, `Presupuesto`)

VALUES

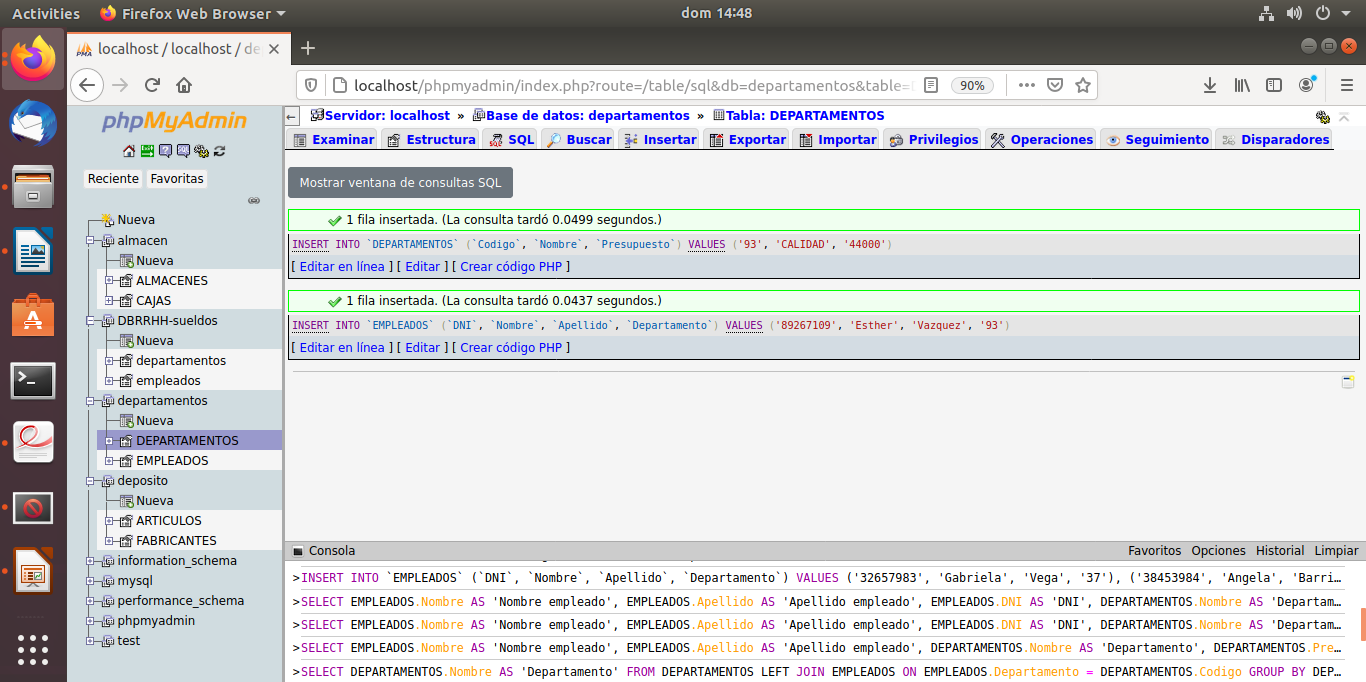
('93', 'CALIDAD', '44000');

INSERT INTO `EMPLEADOS` (`DNI`, `Nombre`, `Apellido`, `Departamento`)

VALUES

('89267109', 'Esther', 'Vazquez', '93')

;



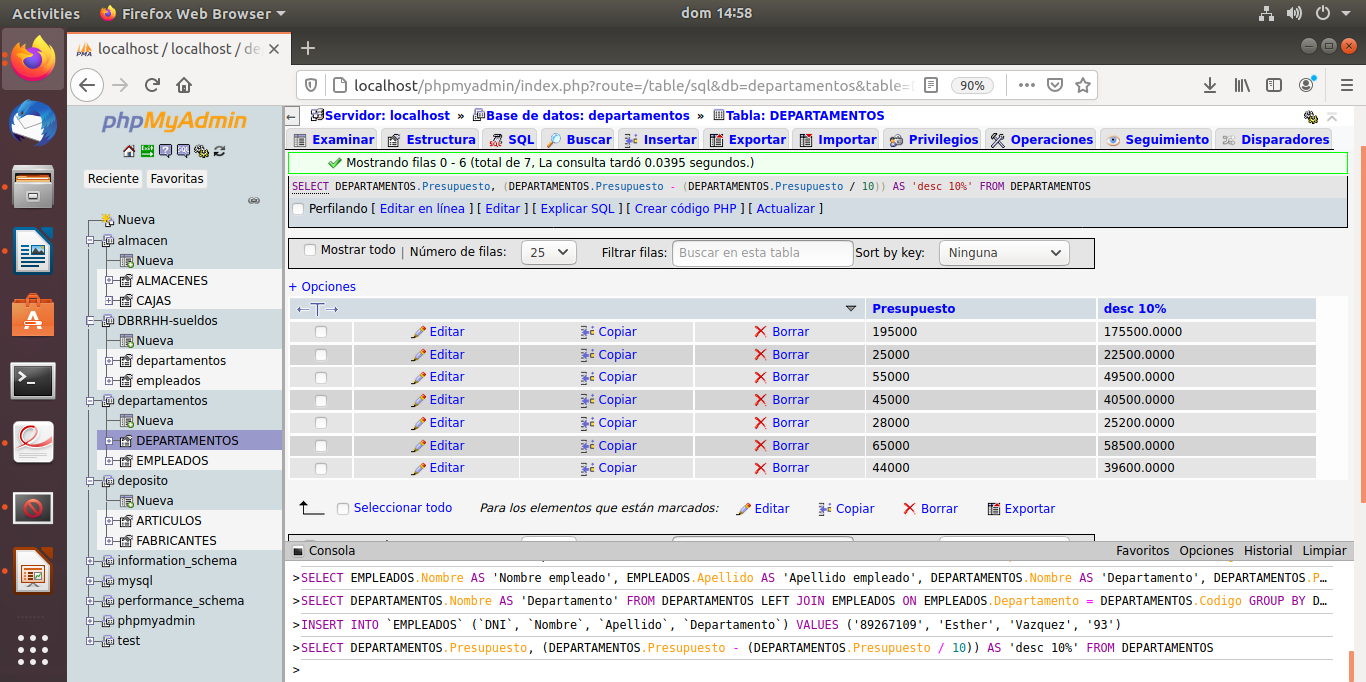
/\* 13. Aplicar un recorte presupuestario del 10 % a todos los departamentos. \*/

/\* Test para previsualizar el resultado\*/

SELECT DEPARTAMENTOS.Presupuesto,

(DEPARTAMENTOS.Presupuesto - (DEPARTAMENTOS.Presupuesto / 10)) AS 'desc 10%'

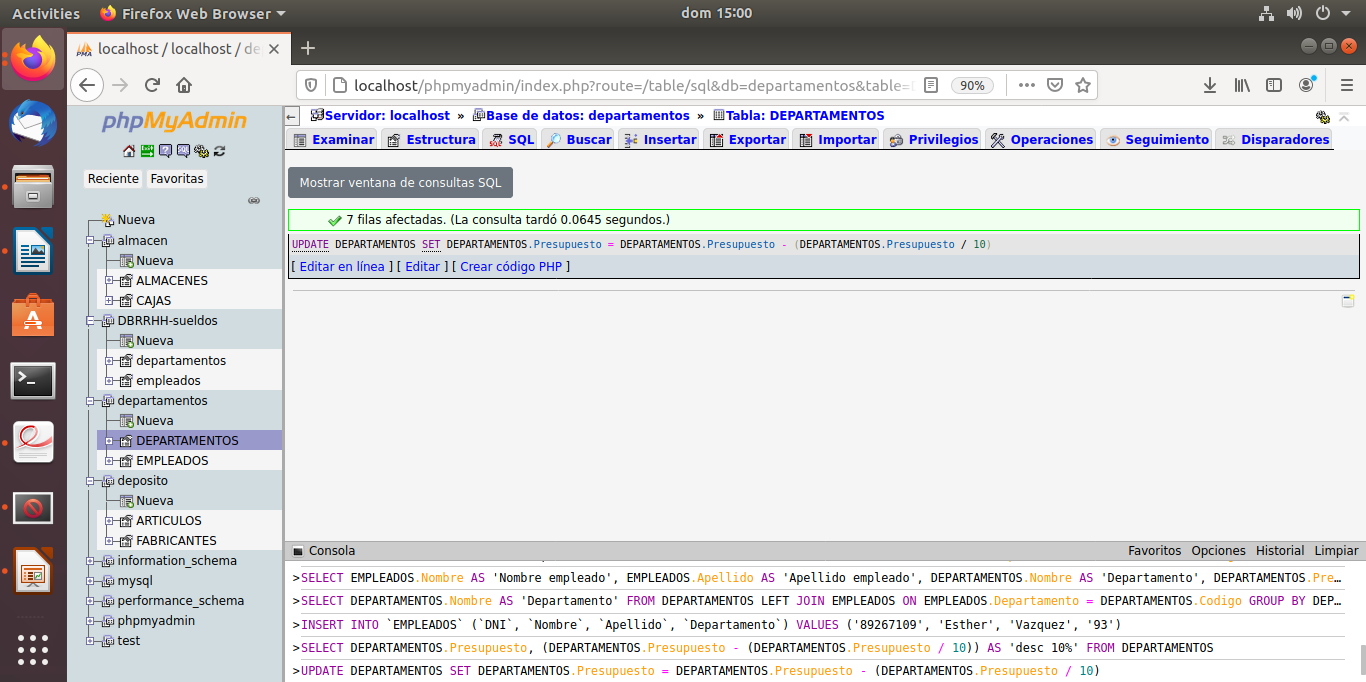
FROM DEPARTAMENTOS



/\* Update \*/

UPDATE DEPARTAMENTOS

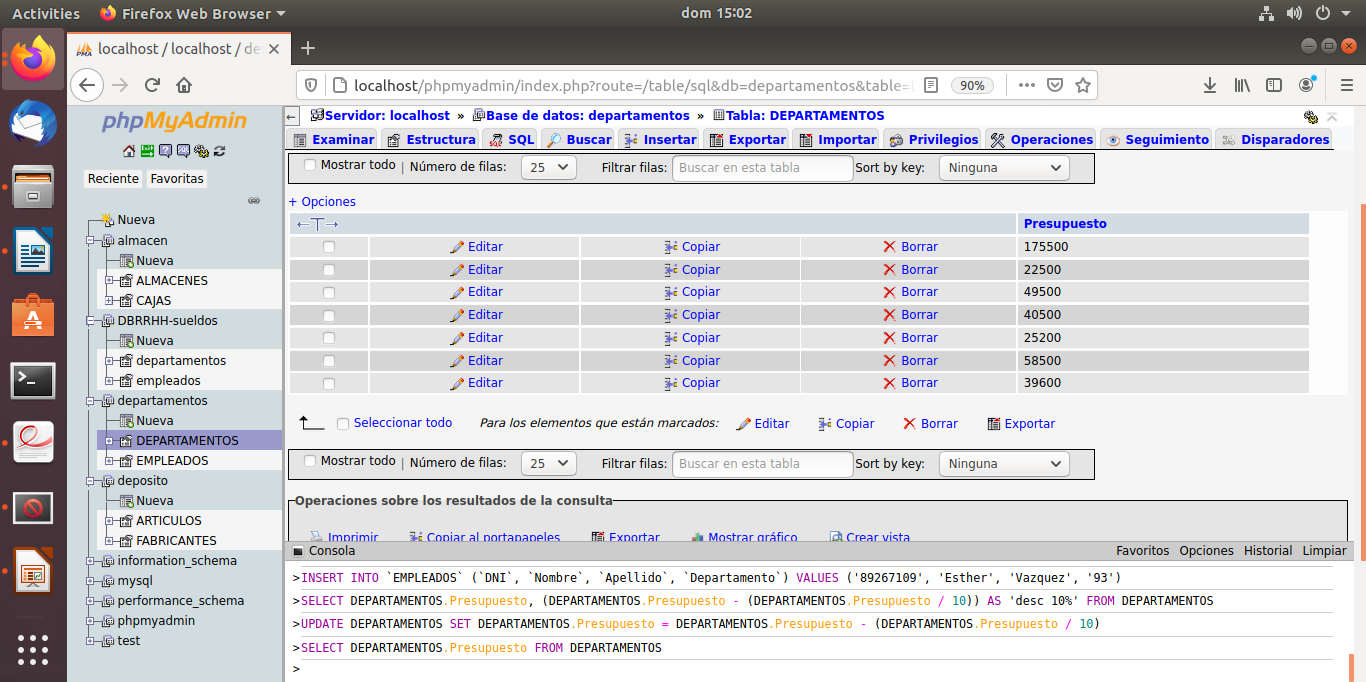
SET DEPARTAMENTOS.Presupuesto = DEPARTAMENTOS.Presupuesto - (DEPARTAMENTOS.Presupuesto / 10)



/\* Consulta \*/

SELECT DEPARTAMENTOS.Presupuesto

FROM DEPARTAMENTOS



/\* 14.- Obtener el numero de cajas que hay en cada almac ́en. \*/

SELECT ALMACENES.Lugar AS 'Almacen',

COUNT(CAJAS.NumRefencia) AS 'Cant.Cajas'

FROM ALMACENES, CAJAS

WHERE CAJAS.Almacen = ALMACENES.Codigo

GROUP BY ALMACENES.Lugar



/\* Vaciar el contenido de los almacenes que esta ́n saturados. \*/

/\* Determinar almacenes saturados \*/

SELECT ALMACENES.Lugar AS 'Almacen',

ALMACENES.Codigo AS 'Codigo',

ALMACENES.Capacidad AS 'Capacidad del almacen',

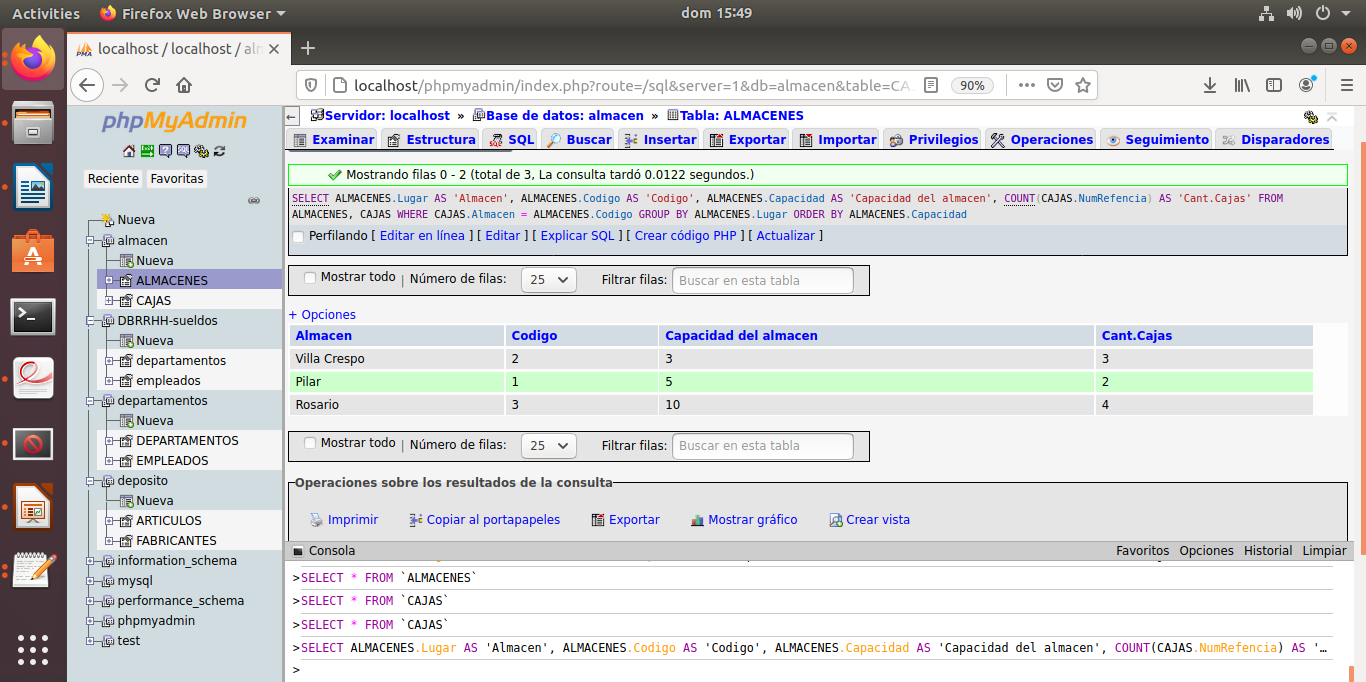
COUNT(CAJAS.NumRefencia) AS 'Cant.Cajas'

FROM ALMACENES, CAJAS

WHERE CAJAS.Almacen = ALMACENES.Codigo

GROUP BY ALMACENES.Lugar

ORDER BY ALMACENES.Capacidad



/\* Encontrado el almacen, se lo vacia de todas las cajas que tiene \*/

/\* Test = Ver las cajas que se sacaran del almacen \*/

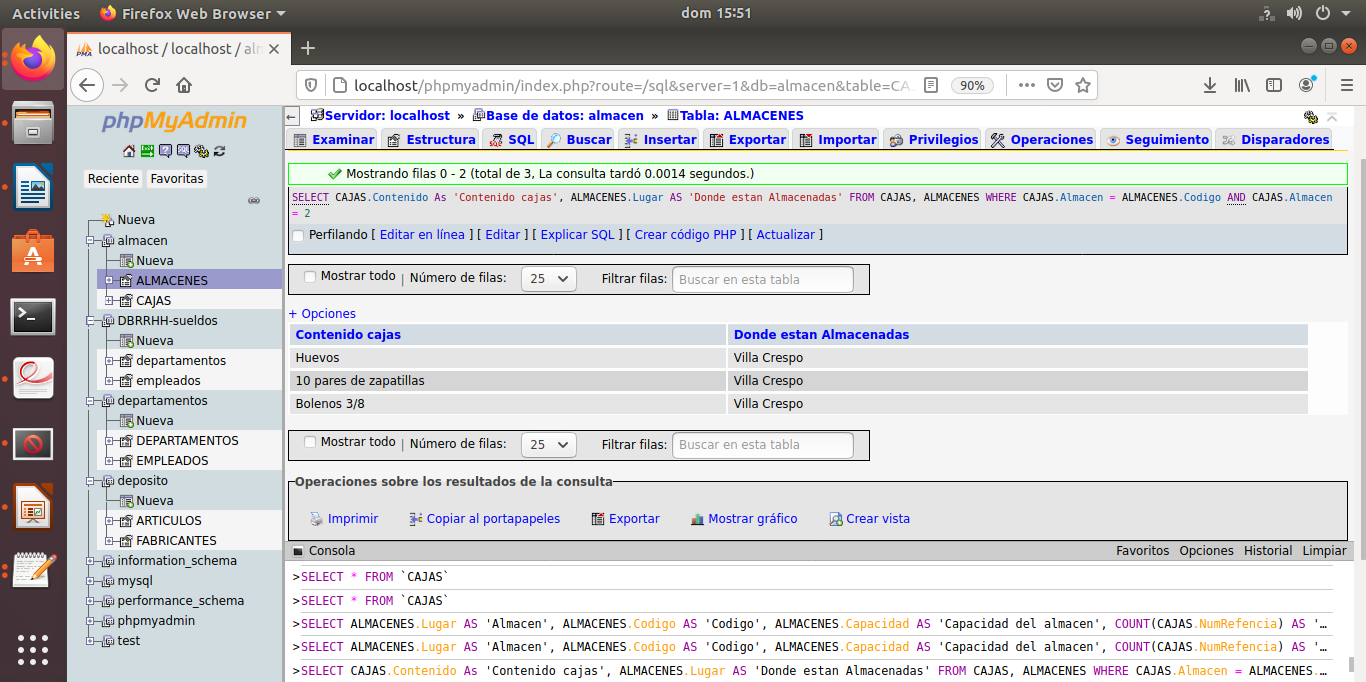
SELECT CAJAS.Contenido As 'Contenido cajas',

ALMACEN.Lugar AS 'Donde estan Almacenadas'

FROM CAJAS, ALMACENES

WHERE CAJAS.Almacen = ALMACENES.Codigo

AND CAJAS.Almacen = 3

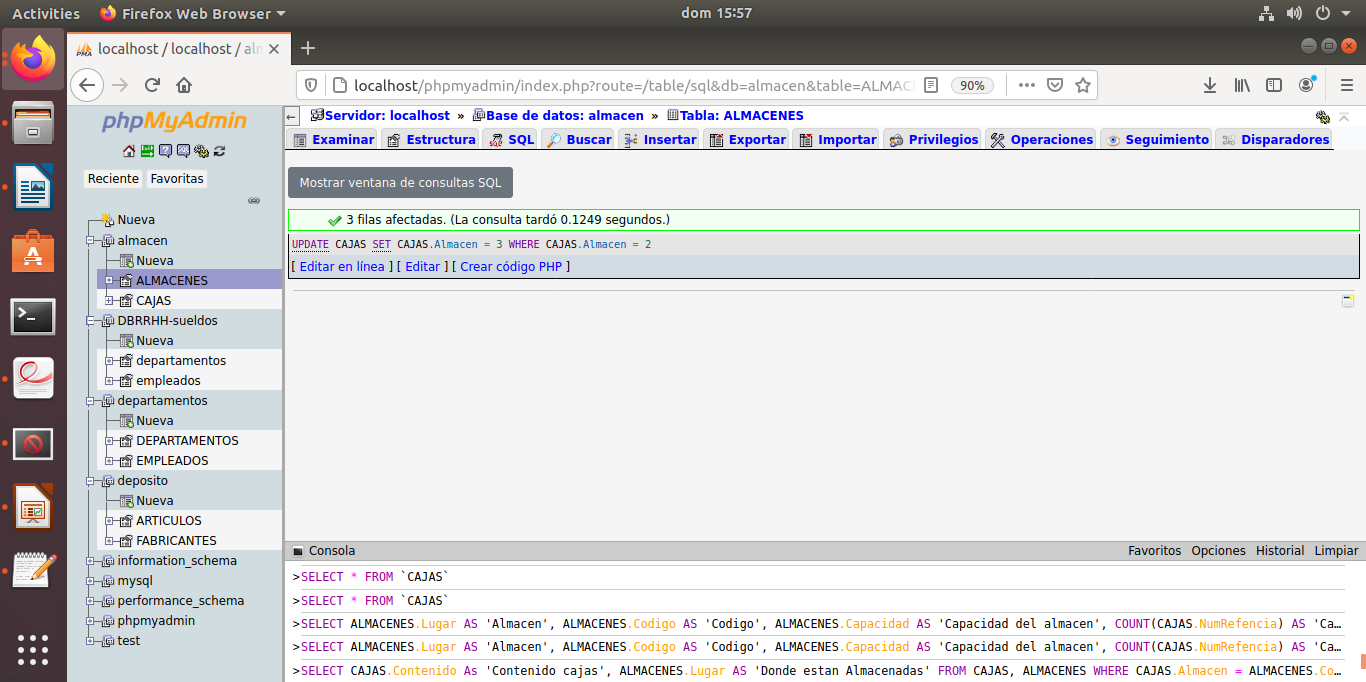


/\* Update -> Se vacia el almacen de saturado. Plus -> se envian las cajas al almacen con mas espacio \*/

UPDATE CAJAS

SET CAJAS.Almacen = 3

WHERE CAJAS.Almacen = 2



/\* Se lista de nuevo los almacenes con los cambios realizados\*/

SELECT ALMACENES.Lugar AS 'Almacen',

ALMACENES.Codigo AS 'Codigo',

ALMACENES.Capacidad AS 'Capacidad del almacen',

COUNT(CAJAS.NumRefencia) AS 'Cant.Cajas'

FROM ALMACENES, CAJAS

WHERE CAJAS.Almacen = ALMACENES.Codigo

GROUP BY ALMACENES.Lugar

ORDER BY ALMACENES.Capacidad

