

UNIVERSIDAD DE ORIENTE
 NUCLEO E ANZOATEGUI
 TALLER DE SISTEMA DE BASE DE DATOS

Taller de Sistemas de Base de Datos
 10 PTS

Dado el siguientes ejemplar de base de datos

Debe crear utilizando XAMPP

La base de datos del ejemplar en MYSQL de XAMPP el nombre será su primer Apellido y la inicial del nombre en mi caso sería carrasquerom

Cuatro disparadores

Cuatro Eventos

Seis procedimientos almacenados

Deben entregar la base de datos

Y en un archivo en Word cada uno de los script asociados con los disparadores, eventos y procedimientos almacenados

propietario				empleado				
cedulap	nombre	direccion		cedulae	nombre	rif	cargo	sueldo
100	pedro	plc		1000	maria	j100	cajera	80
200	juan	bna		2000	luis	j100	panadero	85
300	Chan	plc		3000	antonio	j100	vendedor	50
400	jose	guanta		4000	carmen	j200	cajero	70
500	joan	fecheria		5000	teresa	j200	pastelero	120
				6000	ana	j300	ayudante	70
NEGOCIO				7000	Carolina	j300	sastre	150
rif	cedulap	nombren	Direccion	8000	hector	j400	vendedor	60
J100	100	Pasteleria XXX	plc	9000	luis	j400	cajero	70
J200	100	Pasteleria YYY	bna	9500	manuel	j500	almacen	90
j300	200	Sastreria AAA	plc	9600	jacson	j500	cajero	80
j400	300	Quincalla PPP	guanta	9700	andres	j500	vendedor	70
j500	400	Libreria CCC	bna	9800	gerardo	j600	encargado	100
j600	400	cyber NNN	bna	8100	miguel	j600	tecnico	80
j700	500	cyber OOO	lecheria	8200	sonia	j700	tecnico	90
				8500	dario	j700	tecnico	90

Asignación #01 Eulises Brazon

Creando las tablas de datos con sus respectivos atributos

```
1 -- -- creando las tablas de datos --
2 CREATE TABLE propietario (cedulap int(4) DEFAULT NULL, nombre varchar(10) DEFAULT NULL, direccion varchar(10) DEFAULT NULL);
3 CREATE TABLE negocio (rif varchar(4) DEFAULT NULL, cedula int(4) DEFAULT NULL,
4     nombren varchar(15) DEFAULT NULL, direccion varchar(10) DEFAULT NULL);
5 CREATE TABLE empleado (cedulae int(5) DEFAULT NULL, nombre varchar(10) DEFAULT NULL, rif varchar(5) DEFAULT NULL,
6     cargo varchar(10) DEFAULT NULL, sueldo int(4) DEFAULT NULL );
```

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0008 segundos.)

```
-- -- creando las tablas de datos -- CREATE TABLE propietario (cedulap int(4) DEFAULT NULL, nombre varchar(10) DEFAULT NULL, direccion varchar(10)
DEFAULT NULL);
```

[[Editar en línea](#)] [[Editar](#)] [[Crear código PHP](#)]

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0008 segundos.)

```
CREATE TABLE negocio (rif varchar(4) DEFAULT NULL, cedula int(4) DEFAULT NULL, nombren varchar(15) DEFAULT NULL, direccion varchar(10) DEFAULT
NULL);
```

[[Editar en línea](#)] [[Editar](#)] [[Crear código PHP](#)]

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0006 segundos.)

```
CREATE TABLE empleado (cedulae int(5) DEFAULT NULL, nombre varchar(10) DEFAULT NULL, rif varchar(5) DEFAULT NULL, cargo varchar(10) DEFAULT NULL,
sueldo int(4) DEFAULT NULL );
```

[[Editar en línea](#)] [[Editar](#)] [[Crear código PHP](#)]

Cargando los datos de la tabla propietario

```
1 -- -- cargando datos tabla propietario--
2
3 INSERT INTO propietario (cedulap, nombre, direccion ) VALUES
4 (100 , 'pedro', 'plc'),
5 (200 , 'juan', 'bna'),
6 (300 , 'chan', 'plc'),
7 (400 , 'jose', 'guanta'),
8 (500 , 'joan', 'lecheria')|
```

✓ 5 filas insertadas. (La consulta tardó 0,0525 segundos.)

```
-- -- cargando datos tabla propietario-- INSERT INTO propietario (cedulap, nombre, direccion ) VALUES (100 , 'pedro', 'plc'), (200 , 'juan', 'bna'),
(300 , 'chan', 'plc'), (400 , 'jose', 'guanta'), (500 , 'joan', 'lecheria');
```

[[Editar en línea](#)] [[Editar](#)] [[Crear código PHP](#)]

Cargando los datos de la tabla negocio

```
1 -- -- cargando datos de la tabla negocio --
2 INSERT INTO negocio (rif, cedulap, nombren, direccion) values
3 ('J100', 100, 'Pasteleria XXX', 'plc'),
4 ('J200', 100, 'Pasteleria YYY', 'bna'),
5 ('J300', 200, 'Sastreria AAA', 'plc'),
6 ('J400', 300, 'Quincalla PPP', 'guanta'),
7 ('J500', 400, 'Libreria CCC', 'bna'),
8 ('J600', 400, 'vyber NNN', 'bna'),
9 ('J700', 500, 'cyber 000', 'lecheria');
```

✓ 7 filas insertadas. (La consulta tardó 0,0500 segundos.)

```
-- -- cargando datos de la tabla negocio -- INSERT INTO negocio (rif, cedulap, nombren, direccion) values ('J100', 100, 'Pasteleria XXX', 'plc'),
('J200', 100, 'Pasteleria YYY', 'bna'), ('J300', 200, 'Sastreria AAA', 'plc'), ('J400', 300, 'Quincalla PPP', 'guanta'), ('J500', 400, 'Libreria
CCC', 'bna'), ('J600', 400, 'vyber NNN', 'bna'), ('J700', 500, 'cyber 000', 'lecheria');
```

[\[Editar en línea \]](#) [\[Editar \]](#) [\[Crear código PHP \]](#)

Agregando los datos de la tabla empleado

```
1 -- -- insertando datos de la tabla empleado --
2 INSERT INTO empleado (cedulae, nombre, rif, cargo, sueldo) VALUES
3 (1000, 'maria', 'J100', 'cajera', 80),
4 (2000, 'luis', 'J100', 'panadero', 85),
5 (3000, 'antonio', 'J100', 'vendedor', 50),
6 (4000, 'carmen', 'J200', 'cajero', 70),
7 (5000, 'teresa', 'J200', 'pastelero', 120),
8 (6000, 'ana', 'J300', 'ayudante', 70),
9 (7000, 'carolina', 'J300', 'sastre', 150),
10 (8000, 'hector', 'J400', 'vendedor', 60),
11 (9000, 'luis', 'J400', 'cajero', 70),
12 (9500, 'manuel', 'J500', 'almecen', 90),
13 (9600, 'jacson', 'J500', 'cajero', 80),
14 (9700, 'andres', 'J500', 'vendedor', 70),
15 (9800, 'gerardo', 'J600', 'encargado', 100),
16 (8100, 'miguel', 'J600', 'tecnico', 80),
17 (8200, 'sonia', 'J700', 'tecnico', 90),
18 (8500, 'dario', 'J700', 'tecnico', 90);
```

✓ 16 filas insertadas. (La consulta tardó 0,0008 segundos.)

```
-- -- insertando datos de la tabla empleado -- INSERT INTO empleado (cedulae, nombre, rif, cargo, sueldo) VALUES (1000, 'maria', 'J100', 'cajera'
, 80), (2000, 'luis', 'J100', 'panadero', 85), (3000, 'antonio', 'J100', 'vendedor', 50), (4000, 'carmen', 'J200', 'cajero', 70), (5000,
'teresa', 'J200', 'pastelero', 120), (6000, 'ana', 'J300', 'ayudante', 70), (7000, 'carolina', 'J300', 'sastre', 150), (8000, 'hector',
'J400', 'vendedor', 60), (9000, 'luis', 'J400', 'cajero', 70), (9500, 'manuel', 'J500', 'almecen', 90), (9600, 'jacson', 'J500', 'cajero'
, 80), (9700, 'andres', 'J500', 'vendedor', 70), (9800, 'gerardo', 'J600', 'encargado', 100), (8100, 'miguel', 'J600', 'tecnico', 80),
(8200, 'sonia', 'J700', 'tecnico', 90), (8500, 'dario', 'J700', 'tecnico', 90);
```

[\[Editar en línea \]](#) [\[Editar \]](#) [\[Crear código PHP \]](#)

Creando una tabla de registro de empleados eliminados

```
1 CREATE TABLE reg_empleado_D (cedulae varchar(5), nombren varchar(10), fecha_D datetime);
```

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0009 segundos.)

```
CREATE TABLE reg_empleadoD (cedulae varchar(5), nombren varchar(10), fecha_D datetime);
```

[Editar en línea] [Editar] [Crear código PHP]

Para posteriormente crear **un disparador** que guarde los datos de los empleados antes de ser eliminados de la tabla empleado

```
1 CREATE TRIGGER emplado_BD BEFORE DELETE ON empleado FOR EACH ROW
2 INSERT INTO reg_empleado_D (cedulae, nombren, fecha_D) VALUES (old.cedulae, old.nombren, now());
```

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0879 segundos.)

```
CREATE TRIGGER emplado_BD BEFORE DELETE ON empleado FOR EACH ROW INSERT INTO reg_empleado_D (cedulae, nombren, fecha_D) VALUES (old.cedulae, old.nombren, now());
```

[Editar en línea] [Editar] [Crear código PHP]

Creando una tabla para guardar el registro de los nuevos propietarios que van ingresando al sistema

```
1 CREATE TABLE new_propietario (cedulap int(4), nombre varchar(10), fecha_I datetime);
2
```

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0009 segundos.)

```
CREATE TABLE new_propietario (cedulap int(4), nombre varchar(10), fecha_I datetime);
```

[Editar en línea] [Editar] [Crear código PHP]

Creando un **segundo trigger** que registra la fecha en la que se a ingresado un nuevo propietario dentro de la tabla new propietario.

```
1 CREATE TRIGGER reg_propietario_I BEFORE INSERT ON propietario FOR EACH ROW
2 INSERT INTO new_propietario (cedulap, nombre, fecha_I) VALUES (new.cedulap, new.nombre, now());
```

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,2625 segundos.)

```
CREATE TRIGGER reg_propietario_I BEFORE INSERT ON propietario FOR EACH ROW INSERT INTO new_propietario (cedulap, nombre, fecha_I) VALUES (new.cedulap, new.nombre, now());
```

[Editar en línea] [Editar] [Crear código PHP]

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0019 segundos.)

```
drop TRIGGER reg_propietario_I;
```

[Editar en línea] [Editar] [Crear código PHP]

Es mejor programarlo para que se ejecute after (después)

```
1 CREATE TRIGGER reg_propietario_I AFTER INSERT ON propietario FOR EACH ROW
2 INSERT INTO new_propietario (cedulap, nombre, fecha_I) VALUES (new.cedulap, new.nombre, now());
```

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0820 segundos.)

```
CREATE TRIGGER reg_propietario_I AFTER INSERT ON propietario FOR EACH ROW INSERT INTO new_propietario (cedulap, nombre, fecha_I) VALUES (new.cedulap, new.nombre, now());
```

[[Editar en línea](#)] [[Editar](#)] [[Crear código PHP](#)]

Creando el **tercer trigger** utilizado para validar que al actualizar el sueldo del empleado, no debe de ser negativo, de ser así no se guarda la modificación y sigue conservando su valor anterior

```
1 DELIMITER //
2 CREATE TRIGGER validar_SUELDO BEFORE UPDATE ON empleado FOR EACH ROW
3 BEGIN
4     IF (NEW.sueldo < 0) THEN
5         SET NEW.sueldo = old.sueldo;
6     END IF;
7 END; //
```

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,1548 segundos.)

```
CREATE TRIGGER validar_SUELDO BEFORE UPDATE ON empleado FOR EACH ROW BEGIN IF (NEW.sueldo < 0) THEN SET NEW.sueldo = old.sueldo; END IF; END;;
```

[[Editar en línea](#)] [[Editar](#)] [[Crear código PHP](#)]

Creando el **cuarto trigger** se utiliza para evitar que se logre cargar un valor negativo en el campo de cedula de propietario

```
1 DELIMITER //
2 CREATE TRIGGER validar_cedula_p BEFORE INSERT ON propietario FOR EACH ROW
3 BEGIN
4     IF(new.cedulap<0) THEN
5         set new.cedulap=0;
6     END IF;
7 END; //
```

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0782 segundos.)

```
CREATE TRIGGER validar_cedula_p BEFORE INSERT ON propietario FOR EACH ROW BEGIN IF(new.cedulap<0) THEN set new.cedulap=0; END IF; END;;
```

[[Editar en línea](#)] [[Editar](#)] [[Crear código PHP](#)]

Asignación #01 Eulises Brazon

Activando los eventos

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0005 segundos.)

```
SET GLOBAL event_scheduler = ON;
```

[Editar en línea](#) [[Editar](#)] [[Crear código PHP](#)]

Creando el **primer evento** para que inserte un elemento de prueba en la tabla de propietario

```
1 CREATE EVENT insertion_event
2 ON SCHEDULE AT '2022-05-20 11:00:00'
3 DO INSERT INTO propietario (cedulap, nombre, direccion) VALUES (1234, 'prueba', 'insertando');
```

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0244 segundos.)

```
CREATE EVENT insertion_event ON SCHEDULE AT '2022-05-20 11:00:00' DO INSERT INTO propietario (cedulap, nombre, direccion) VALUES (1234, 'prueba', 'insertando');
```

[[Editar en línea](#)] [[Editar](#)] [[Crear código PHP](#)]

Segundo evento que modifica el campo de la dirección con el valor actualizado para la tupla donde la cédula es 1234

```
1 CREATE EVENT UPDATE_event
2 ON SCHEDULE AT '2022-05-20 11:01:00'
3 ON COMPLETION PRESERVE ENABLE DO UPDATE propietario set direccion = 'actualizando' where cedulap = 1234;
```

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0312 segundos.)

```
CREATE EVENT UPDATE_event ON SCHEDULE AT '2022-05-20 11:01:00' ON COMPLETION PRESERVE ENABLE DO UPDATE propietario set direccion = 'actualizando' where cedulap = 1234;
```

[[Editar en línea](#)] [[Editar](#)] [[Crear código PHP](#)]

Creando un **tercer evento** para modificar el campo de nombre en la tupla donde cedula es 1234

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0292 segundos.)

```
CREATE EVENT UPDATE_nombrep_event ON SCHEDULE AT '2022-05-20 11:02:00' ON COMPLETION PRESERVE ENABLE DO UPDATE propietario set nombrep = 'modificado' where cedulap = 1234;
```

[[Editar en línea](#)] [[Editar](#)] [[Crear código PHP](#)]

```
1 CREATE EVENT UPDATE_nombrep_event
2 ON SCHEDULE AT '2022-05-20 11:02:00'
3 ON COMPLETION PRESERVE ENABLE DO UPDATE propietario set nombrep = 'modificado' where cedulap = 1234;
```

Por último, un **cuarto evento** que elimine esta persona ficticio que se uso para la prueba

```
1 CREATE EVENT DELETE_event
2 ON SCHEDULE AT '2022-05-20 11:03:00'
3 ON COMPLETION PRESERVE ENABLE DO DELETE FROM propietario where cedulap = 1234;
```

Asignación #01 Eulises Brazon

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0426 segundos.)

```
CREATE EVENT DELETE_event ON SCHEDULE AT '2022-05-20 11:03:00' ON COMPLETION PRESERVE ENABLE DO DELETE FROM propietario where cedulap = 1234;
```

[Editar en línea] [Editar] [Crear código PHP]

Procedimiento uno creado para retornar los datos del empleado que coincida con la cedula buscada

```
1 CREATE PROCEDURE buscar_empleado (cedula_B int(5)) SELECT * FROM empleado WHERE cedulae= cedula_B;
```

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0845 segundos.)

```
CREATE PROCEDURE buscar_empleado (cedula_B int(5)) SELECT * FROM empleado WHERE cedulae= cedula_B;
```

[Editar en línea] [Editar] [Crear código PHP]

Procedimiento dos actualizar sueldo de un empleado

```
1 CREATE PROCEDURE actualizar_sueldo (cedula_e int(4),NewSueldo int(4))
2 UPDATE empleado SET sueldo=NewSueldo WHERE cedulae=cedula_e;
```

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0765 segundos.)

```
CREATE PROCEDURE actualizar_sueldo (cedula_e int(4),NewSueldo int(4)) UPDATE empleado SET sueldo=NewSueldo WHERE cedulae=cedula_e;
```

[Editar en línea] [Editar] [Crear código PHP]

Procedimiento tres retorna los datos de un negocio buscado por su rif

```
1 CREATE PROCEDURE buscar_negocio (rif_B varchar(4)) SELECT * FROM negocio WHERE rif= rif_B;
```

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0885 segundos.)

```
CREATE PROCEDURE buscar_negocio (rif_B varchar(4)) SELECT * FROM negocio WHERE rif= rif_B;
```

[Editar en línea] [Editar] [Crear código PHP]

Procedimiento cuatro una búsqueda del dato de un propietario según su cedula

```
1 CREATE PROCEDURE buscar_propietario (cedula_p int(4)) SELECT * FROM propietario WHERE cedulap= cedula_p;
```

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0884 segundos.)

```
CREATE PROCEDURE buscar_propietario (cedula_p int(4)) SELECT * FROM propietario WHERE cedulap= cedula_p;
```

[Editar en línea] [Editar] [Crear código PHP]

Procedimiento cinco modificar el cargo de una cedula

```
1 CREATE PROCEDURE actualizar_cargo (cedula_e int(5), NewCargo varchar(10))
2 UPDATE empleado SET cargo=NewCargo WHERE cedulae= cedula_e;
```

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0754 segundos.)

```
CREATE PROCEDURE actualizar_cargo (cedula_e int(5), NewCargo varchar(10)) UPDATE empleado SET cargo=NewCargo WHERE cedulae= cedula_e;
```

[\[Editar en línea \]](#) [\[Editar \]](#) [\[Crear código PHP \]](#)

Procedimiento seis dada una ubicación retorna los negocios que se encuentran en esa zona

```
1 CREATE PROCEDURE buscar_negociosPorUbicacion (direccion_B varchar(10)) SELECT * FROM negocio WHERE direccion= direccion_B;
```

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0867 segundos.)

```
CREATE PROCEDURE buscar_negociosPorUbicacion (direccion_B varchar(10)) SELECT * FROM negocio WHERE direccion= direccion_B;
```

[\[Editar en línea \]](#) [\[Editar \]](#) [\[Crear código PHP \]](#)