
*Escuela Superior de Computo*

*Grupo: 2CV2*

*Bases de datos*

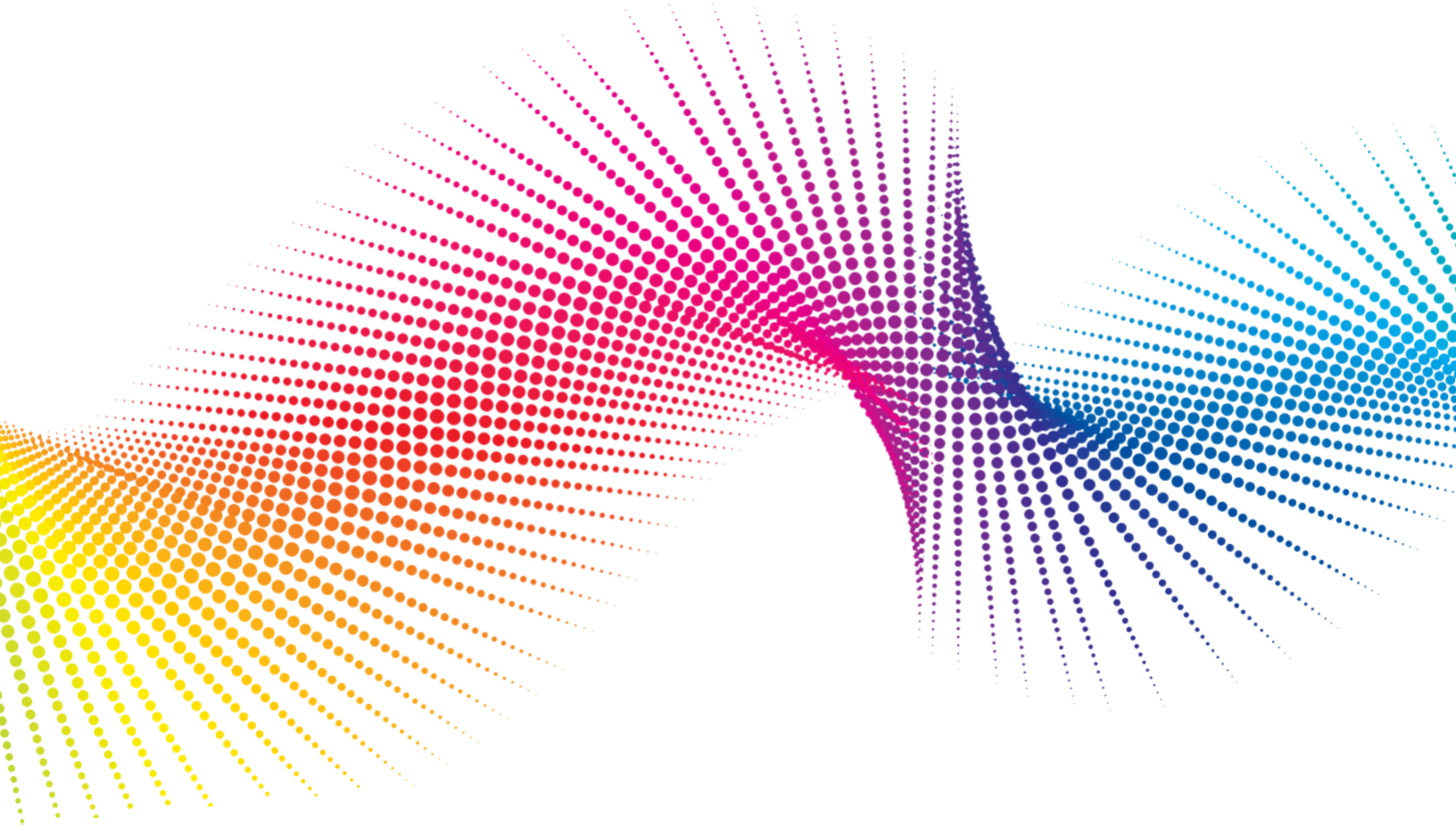
*M. en C. Euler Hernández Contreras*

*3° Parcial*

*Viernes 18 de Mayo de 2018*

*Reporte de 8° practica*

*Alumno: Aaron Antonio Garcia Gonzalez*



Índice

## Marco Teórico

### Sentencias de insercion, actualizacion y eliminacion de tuplas

#### INSERT

Su finalidad es la de insertar datos en las tablas de una base de datos.

Es posible insertar una nueva fila en una tabla de dos formas distintas:

- INSERT INTO nombre_tabla
  VALUES (valor1, valor2, valor3, .)
- INSERT INTO nombre_tabla (columna1, columna2, columna3,.)
  VALUES (valor1, valor2, valor3, .)

#### UPDATE

UPDATE actualiza columnas de filas existentes de una tabla con nuevos valores. La cláusula SET indica las columnas a modificar y los valores que deben tomar. La cláusula WHERE, si se da, especifica qué filas deben ser actualizadas. Si no se especifica, serán actualizadas todas ellas. Si se especifica la cláusula ORDER BY, las filas se modificarán en el orden especificado. La cláusula LIMIT establece un límite al número de filas que se pueden actualizar.

La sentencia UPDATE soporta los modificadores siguientes:

Si se usa la palabra LOW_PRIORITY, la ejecución de UPDATE se retrasará hasta que no haya otros clientes haciendo lecturas de la tabla.

Si se especifica IGNORE, la sentencia UPDATE no se abortará si se producen errores durante la actualización. Las filas con conflictos de claves duplicadas no se actualizarán. Las filas para las que la actualización de columnas se puedan producir errores de conversión se actualizarán con los valores válidos más próximos.

#### DELETE

DELETE elimina columnas desde "table_name" que satisfagan la condición dada por la "where_definition", y devuelve el número de registros borrados.

Si se usa una sentencia DELETE sin cláusula *WHERE*, todas las filas serán borradas. Una forma más rápida de hacer esto, cuando no se necesita conocer el número de filas eliminadas, es usar .

#### TRUNCATE

TRUNCATE TABLE vacía una tabla por completo. Lógicamente, esto es equivalente a una sentencia que borre todas las filas, pero existen diferencias prácticas bajo algunas circunstancias.

Para InnoDB, TRUNCATE TABLE es mapeado a , de modo que no hay diferencia. para otros motores de almacenamiento, TRUNCATE TABLE difiere de en lo siguiente, desde MySQL 4.0:

El truncado trabaja suprimiendo y recreando la tabla, que es mucho más rápido que borrar filas una a una.

TRUNCATE no es seguro a nivel de transacción; se puede obtener un error si se tienen transacciones activas o un bloqueo de tabla activo.

No se devuelve el número de filas eliminadas.

Mientras el fichero de definición de tabla 'table_name.frm' sea válido, la tabla puede ser recreada como una tabla vacía con TRUNCATE TABLE, aunque los ficheros de datos o índices estén corruptos.

El manipulador de tabla no recordará el último valor usado para AUTO_INCREMENT, pero empezará a contar desde el principio. Esto es cierto incluso para tablas MyISAM, que generalmente no reutiliza los valores de secuencia.

## Diferencias entre MySQL truncate, delete y drop

Estas sentencias MySQL sirven para eliminar diferentes tipos de contenido de una base de datos

MySQL TRUNCATE TABLE

Con esta sentencia podemos vaciar todo el contenido una tabla de una base de datos. La estructura de dicha tabla permace intacta.

MySQL DELETE

Esta sentencia elimina una serie de registros/filas de una tabla, si no se especifica la claúsula WHERE se eliminarán todos los registros de la tabla del mismo modo que TRUNCATE.

MySQL DROP

Esta sentencia es más completa que las anteriores, puede borrar tablas, vistas o bases de datos.

## CREATE TRIGGER

Vamos a repasar la sentencia de MySQL CREATE TRIGGER, su funcionalidad es la de detectar ciertos eventos asociados a una tabla de la base de datos y ejecutar una serie de acciones tras dicha detección. Estas acciones también son conocidas como disparadores.
Estos eventos son:

- INSERT: Inserción de datos.
- UPDATE: Actualización de datos.
- DELETE: Eliminación de datos.

Con DROP TRIGGER eliminamos el disparador.

Sintaxis de la sentencia de MySQL CREATE TRIGGER

```
CREATE
    [DEFINER = { user | CURRENT_USER }]
    TRIGGER trigger_name
    trigger_time trigger_event
    ON tbl_name FOR EACH ROW
    [trigger_order]
```

trigger_body

DEFINER: Especifica el usuario de BBDD con privilegios para desencadenar un TRIGGER. Por defecto es el usuario que crear el TRIGGER.

trigger_name: Nombre del TRIGGER/disparador.

trigger_time: Especifica cuando se ha de ejecutar el TRIGGER, antes o después del evento detectado.

trigger_event: Evento que activa el TRIGGER –> INSERT, UPDATE y DELETE.

tbl_name: Nombre de la tabla en la que detectaremos el trigger_event.

trigger_order: Una tabla puede tener asociados varios TRIGGER, por defecto la ejecución de cada uno es el orden en el que fuero creados. Para alterar ese orden podemos definirlos con FOLLOWS (depués de nombre trigger) y con PRECEDE (antes de nombre_trigger).

trigger_body: Código del TRIGGER.

Por ejemplo:

- Primero crearemos la tabla para la que se establecerá el trigger:

```
mysql> CREATE TABLE people (age INT, name varchar(150));
```

- A continuación definiremos el trigger. Se ejecutará antes de cada sentencia INSERT para la tabla people:

```
mysql> delimiter //
mysql> CREATE TRIGGER agecheck BEFORE INSERT ON people FOR EACH ROW IF NEW.age < 0 THEN SET NEW.age
= 0; END IF;//
Query OK, 0 rows affected (0.00 sec)
mysql> delimiter ;
```

- Insertaremos dos registros para comprobar la funcionalidad del trigger.

```
mysql> INSERT INTO people VALUES (-20, 'Sid'), (30, 'Josh');
Query OK, 2 rows affected (0.00 sec)
Records: 2 Duplicates: 0 Warnings: 0
```

- Al final comprobaremos el resultado.

```
mysql> SELECT * FROM people;
+——-+——-+
| age | name |
+——-+——-+
| 0 | Sid |
| 30 | Josh |
+——-+——-+
2 rows in set (0.00 sec)
```

Instrucciones:

1. Crear una base de datos
   Create database elektra;
2. Usar la base de datos creada en el paso anterior
   Use elektra;
3. Cargar el script de la base de datos
   source elektra.sql;

- Instrucción a realizar
- Instrucción
- Notas u observaciones

Operaciones de escritura

1. Describimos la relacion producto
   Desc producto;
   Describimos la relacion subcategoria
   Desc subcategoria;

Esto con el objetivo de insertar datos en el paso 2, ya que ya conoceremos las definiciones de los datos que se insertaran.

2. Agregamos un nuevo producto

```
insert into producto
values (3000,"Cilantro","Lo que sea",3.45,
"verdura rica", (select idSubcategoria
from subcategoria where nombre like "Prepara% de Ali%"));
```

Se incluye una subconsulta que devuelve la subcategoria de la relacion subcategoria, para realizar lo anterior, el select se tiene que delimitar por un parentesis

2.1. Consultamos lo anterior

```
select * from producto where idproducto=3000;
```

3. Realizar dos inserciones en una misma sentencia

```
insert into producto
values (3001,"Zanahoria","Lo que sea",5.78,
"verdura rica", (select idSubcategoria
from subcategoria where nombre like "Cocin%")),
(3002,"Cebolla","Lo que sea",2.99,
"verdura rica", (select idSubcategoria
from subcategoria where nombre like "Cocin%"));
```

Para hacer dos inserciones en una misma sentencia, marcamos cada grupo de valores (tuplas), separados por comas y delimitados por un parentesis

3.1. Consultamos lo anterior

```
select * from producto where idproducto between 3000 and 3500;
```

4. Realizar inserciones especificas

```
insert into producto (idproducto, nombre, precioUnitario)
values (30003, "PA", 45.99), (3004, "PB", 60), (3005, "PC", 3000);
```

Al solo insertar datos especificos de una tupla, tenemos que verificar que la definicion de la relacion lo permita, esto es, por ejemplo que no haya NOT NULL, ademas en la sentencia antes del "values" se declaran que datos seran ingresado, todo esto delimitado por un parentesis y se procede despues del "values" ingresando los datos respectivos que se indicaron, los datos que no se ingresaron quedan nulos en la tupla.

4.1 Consultar lo anterior

```
select * from producto where idproducto between 3000 and 3500;
```

5. A todas las mujeres aumentar su salario en un 10%

Primero que nada, consultamos el slario original previo al aumento

```
select idcliente, nombre, salario from cliente
sexo where sexo="F";
```

Volvemos a consultar y los cambios se veran reflejados en todas las feminas

```
update cliente
set salario=salario*1.1
where sexo="F";

select idcliente, nombre, salario from cliente
sexo where sexo like "F";
```

6. Actualizar el correo electronico de todos los que se apelliden "Hernandez"

Primero que nada consultamos cuantos y cuales son los clientes que se apellidan asi

```
select idcliente, nombre, apPaterno, email
from cliente where apPaterno like "Hern_ndez";
```

Ahora si realizamos la actualizacion

```
update cliente set email="hernandezx@escom.com"
where apPaterno like "Hern_nde%";
```

7. Cambiar la fecha de pago para aquellos clientes que ganan mas de $10,000 y poner la fecha de hoy

Consultamos los clientes que cumplen las caracteristicas de este inciso

```
select c.apPaterno, c.apMaterno, c.nombre, c.salario,
p.fechaPago from cliente c, pago p
where c.idcliente=p.idcliente
and c.salario>12000
order by salario;
```

Ahora si, procedemos a realizar la actualizacion

```
update cliente c, pago p set
```

```
p.fechapago= now()
where c.idcliente=p.idcliente
and c.salario>12000;
```

Consultamos nuestra actualizacion

```
select c.apPaterno, c.apMaterno, c.nombre, c.salario,
p.fechaPago from cliente c, pago p
where c.idcliente=p.idcliente
and c.salario>12000
order by salario;
```

8. Actualizar el telegono de las siguientes sucursales

Lo que haremos en cada una, sera consultar, actualizar y consultar

8.1. Apa calvillo

```
select nombre, tel from tienda
where nombre like "Apa Calv%";

update tienda set tel= "00-00-00-00-00"
where nombre like "Mega Tiju% Cent%";

select nombre, tel from tienda
where nombre like "Apa Calv%";
```

8.2. mega Tijuana Centro

```
select nombre, tel from tienda
where nombre like "mega Tij% Cent%";

update tienda set tel= "11-11-11-11-11"
where nombre like "Mega Tiju% Cent%";

select nombre, tel from tienda
where nombre like "mega Tij% Cent%";
```

8.3. Mega Muzquiz

```
select nombre, tel from tienda
where nombre like "Mega Muzqu%";

update tienda set tel= "01-23-05-67-89"
where nombre like "Mega Muzqu%";

select nombre, tel from tienda
where nombre like "Mega Muzqu%";
```

9. Eliminar los productos que valen menos de $1,000

Lo que haremos en cada una, sera consultar, actualizar y consultar

```
select count(*) from producto
where preciounitario<1000;

delete from producto
where precioUnitario<1000;

select count(*) from producto
where preciounitario<1000;
```

10. Eliminamos los clientes que hicieron pagos el dia "2010-03-19"

Primero que nada, veremos cuantos clientes cumplen con la condicion previa, asi como los clientes totales

```
select count(*) from cliente c, pago p
where c.idcliente=p.idcliente
and p.fechapago="2010-03-19";

select count(*) from cliente;
```

Ahora si realizamos la consulta

```
delete c from cliente c, pago p
where c.idcliente=p.idcliente
and p.fechapago="2010-03-19";
```

Volvemos a consultar, para asegurnos que nuestra eliminacion se hizo correctamente

```
select count(*) from cliente c, pago p
where c.idcliente=p.idcliente
and p.fechapago="2010-03-19";

select count(*) from cliente;
```

## Capturas de Pantalla

```
mysql> desc producto;
+---------------+--------------+------+-----+---------+-------+
| Field         | Type         | Null | Key | Default | Extra |
+---------------+--------------+------+-----+---------+-------+
| idproducto    | int(11)      | NO   | PRI | NULL    |       |
| nombre        | varchar(100) | YES  |     | NULL    |       |
| descripcion   | varchar(400) | YES  |     | NULL    |       |
| precioUnitario | double      | YES  |     | NULL    |       |
| marca         | varchar(45)  | YES  |     | NULL    |       |
| idsubcategoria | int(11)     | YES  | MUL | NULL    |       |
+---------------+--------------+------+-----+---------+-------+
6 rows in set (0.01 sec)
```

```
mysql> insert into producto
    -> values (3000,"Cilantro","Lo que sea",3.45,
    -> "verdura rica", (select idSubcategoria
    -> from subcategoria where nombre like "Prepara% de Ali%"));
Query OK, 1 row affected (0.01 sec)

mysql> select * from producto where idproducto=3000;
+------------+----------+-------------+----------------+-------------+----------------+
| idproducto | nombre   | descripcion | precioUnitario | marca       | idsubcategoria |
+------------+----------+-------------+----------------+-------------+----------------+
|       3000 | Cilantro | Lo que sea  |           3.45 | verdura rica |             22 |
+------------+----------+-------------+----------------+-------------+----------------+
1 row in set (0.01 sec)
```

```
mysql> select * from producto where idproducto between 3000 and 3500;
+------------+----------+-------------+----------------+-------------+----------------+
| idproducto | nombre   | descripcion | precioUnitario | marca       | idsubcategoria |
+------------+----------+-------------+----------------+-------------+----------------+
|       3000 | Cilantro | Lo que sea  |           3.45 | verdura rica |             22 |
|       3005 | PC       | NULL        |           3000 | NULL        |           NULL |
+------------+----------+-------------+----------------+-------------+----------------+
2 rows in set (0.00 sec)
```

```
+------------+-----------+-------------+----------------+-------------+----------------+
| idproducto | nombre    | descripcion | precioUnitario | marca       | idsubcategoria |
+------------+-----------+-------------+----------------+-------------+----------------+
|       3000 | Cilantro  | Lo que sea  |           3.45 | verdura rica |             22 |
|       3001 | Zanahoria | Lo que sea  |           5.78 | verdura rica |             41 |
|       3002 | Cebolla   | Lo que sea  |           2.99 | verdura rica |             41 |
|       3005 | PC        | NULL        |           3000 | NULL        |           NULL |
+------------+-----------+-------------+----------------+-------------+----------------+
4 rows in set (0.00 sec)
```

```
mysql> select idcliente, nombre, salario from cliente
    -> sexo where sexo like "F";
+-----------+----------------------+--------------------+
| idcliente | nombre               | salario            |
+-----------+----------------------+--------------------+
|         1 | NURIA DEL CARMEN     | 7425.000000000001 |
|        15 | ELIZABETH DEL CARMEN | 7425.000000000001 |
|        75 | JESSICA MONSERRAT    | 7425.000000000001 |
|       135 | NADIA LUCERO         | 7425.000000000001 |
|       136 | KARINA JANET         | 7425.000000000001 |
|       139 | RUTH HEIDI           | 7425.000000000001 |
|       149 | GRISSEL XHARENY      | 7425.000000000001 |
|       157 | DAYANA MARISOL       | 7425.000000000001 |
+-----------+----------------------+--------------------+
8 rows in set (0.00 sec)
```

```
mysql> update cliente set email="hernandezx@escom.com"
    -> where apPaterno like "Hern_nde%";
Query OK, 5 rows affected (0.00 sec)
Rows matched: 5  Changed: 5  Warnings: 0

mysql> select idcliente, nombre, apPaterno, email
    -> from cliente where apPaterno like "Hern_ndez";
+-----------+--------------------+-----------+----------------------+
| idcliente | nombre             | apPaterno | email                |
+-----------+--------------------+-----------+----------------------+
|        12 | PEDRO EVERARDO     | HERNANDEZ | hernandezx@escom.com |
|        40 | JOSE ROGELIO       | HERNANDEZ | hernandezx@escom.com |
|        41 | JESUS ALEJANDRO    | HERNANDEZ | hernandezx@escom.com |
|        42 | LUIS CARLOS        | HERNANDEZ | hernandezx@escom.com |
|        79 | MAURICIO FRANCISCO | HERNANDEZ | hernandezx@escom.com |
+-----------+--------------------+-----------+----------------------+
5 rows in set (0.00 sec)
```

```
mysql> select c.apPaterno, c.apMaterno, c.nombre, c.salario,
    -> p.fechaPago from cliente c, pago p
    -> where c.idcliente=p.idcliente
    -> and c.salario>12000
    -> order by salario;
+-----------+-----------+--------------------+---------+------------+
| apPaterno | apMaterno | nombre             | salario | fechaPago  |
+-----------+-----------+--------------------+---------+------------+
| HERNANDEZ | LOPEZ     | JOSE ROGELIO       |   12500 | 2010-03-20 |
| LOPEZ     | MARTINEZ  | JONATHAN           |   12500 | 2010-03-22 |
| MARTINEZ  | BEDOLLA   | MARCO ANTONIO      |   12500 | 2010-03-08 |
| MARTINEZ  | PE¥A      | JESUS ADRIAN       |   12500 | 2010-03-23 |
| LANDERO   | REYES     | MAURICIO           |   12500 | 2010-03-22 |
| GARCIA    | SOTO      | ELIAS ENRIQUE      |   12500 | 2010-03-04 |
| HERNANDEZ | OYARZABAL | MAURICIO FRANCISCO |   12500 | 2010-03-09 |
| ESCARCEGA | JAIME     | ANGEL OMAR         |   12500 | 2010-03-04 |
| HERNANDEZ | MEJIA     | LUIS CARLOS        |   12500 | 2010-03-21 |
| MARTINEZ  | LOPEZ     | YOSHIO ALEXIS      |   12500 | 2010-03-23 |
| HERNANDEZ | MEJIA     | JESUS ALEJANDRO    |   12500 | 2010-03-20 |
| ESTRADA   | PAVIA     | JOSE ANTONIO       |   12500 | 2010-03-02 |
| MARTINEZ  | LUNA      | NICOLAS            |   12500 | 2010-03-23 |
| LOPEZ     | MARRON    | RICARDO NESTOR     |   12500 | 2010-03-20 |
+-----------+-----------+--------------------+---------+------------+
14 rows in set (0.00 sec)
```

```
mysql> select nombre, tel from tienda
    -> where nombre like "Apa Calv%";
+--------------+----------------+
| nombre       | tel            |
+--------------+----------------+
| APA CALVILLO | 01-01-01-01-01 |
+--------------+----------------+
1 row in set (0.00 sec)

mysql>
mysql> update tienda set tel= "00-00-00-00-00"
    -> where nombre like "Mega Tiju% Cent%";
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select nombre, tel from tienda
    -> where nombre like "Apa Calv%";
+--------------+----------------+
| nombre       | tel            |
+--------------+----------------+
| APA CALVILLO | 01-01-01-01-01 |
+--------------+----------------+
1 row in set (0.00 sec)

mysql> select nombre, tel from tienda
    -> where nombre like "mega Tij% Cent%";
+---------------------+----------------+
| nombre              | tel            |
+---------------------+----------------+
| MEGA TIJUANA CENTRO | 00-00-00-00-00 |
+---------------------+----------------+
1 row in set (0.00 sec)

mysql>
mysql> update tienda set tel= "11-11-11-11-11"
    -> where nombre like "Mega Tiju% Cent%";
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
mysql> update tienda set tel= "11-11-11-11-11"
    -> where nombre like "Mega Tiju% Cent%";
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1  Changed: 0  Warnings: 0

mysql> select nombre, tel from tienda
    -> where nombre like "mega Tij% Cent%";
+---------------------+----------------+
| nombre              | tel            |
+---------------------+----------------+
| MEGA TIJUANA CENTRO | 11-11-11-11-11 |
+---------------------+----------------+
1 row in set (0.00 sec)
```

```
mysql> select nombre, tel from tienda
    -> where nombre like "Mega Muzqu%";
+--------------+----------------+
| nombre       | tel            |
+--------------+----------------+
| MEGA MUZQUIZ | 01-23-05-67-89 |
+--------------+----------------+
1 row in set (0.00 sec)

mysql>
mysql> update tienda set tel= "01-23-05-67-89"
    -> where nombre like "Mega Muzqu%";
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1  Changed: 0  Warnings: 0

mysql>
mysql> select nombre, tel from tienda
    -> where nombre like "Mega Muzqu%";
+--------------+----------------+
| nombre       | tel            |
+--------------+----------------+
| MEGA MUZQUIZ | 01-23-05-67-89 |
+--------------+----------------+
1 row in set (0.00 sec)
```

```
mysql> delete c from cliente c, pago p
    -> where c.idcliente=p.idcliente
    -> and p.fechapago="2010-03-19";
Query OK, 1 row affected (0.00 sec)

mysql> select count(*) from cliente c, pago p
    -> where c.idcliente=p.idcliente
    -> and p.fechapago="2010-03-19";
+----------+
| count(*) |
+----------+
|        0 |
+----------+
1 row in set (0.00 sec)

mysql>
mysql> select count(*) from cliente;
+----------+
| count(*) |
+----------+
|      125 |
+----------+
1 row in set (0.00 sec)
```

## Conclusiones

A mi parecer, esta practica ha sido la mas sencilla de todas las que hemos realizado, estuvo muy chido la parte en la que dentro de una actualizacion se realizo una subconsulta, probablemente fue algo que otros porfes no enseñan a sus grupos, el tipo de actualizaciones que se realizaron se sintieron muy reales, como si fueren en ls vida real que nnos espera.

Para finalizar,  me gustaria decir que al ser la ultima practica tematica programada de la unidad de aprendizaje, me quede con ganas de aprender a meter datos por ejemplo desde un excel a la base de datos, truquillos extras, pero en general, ya tenemos las bases para seguir aprendiendo por nuestra cuenta.

## Referencias

R. Elmasri and S. Navathe, Fundamentos de sistemas de bases de datos. Pearson Education, 2007. [Online]. Available: https://books.google.com.mx/books?id=NT3uPQAACAAJ

C. Ricardo, Bases de datos. McGraw-Hill Interamericana, 2000. [Online]. Available: https://books.google.com.mx/books?id=BmVHAQAACAAJ

(2017) Mysql :: Mysql documentation. [Online]. Available: https://dev.mysql.com/doc/

https://www.anerbarrena.com/tipos-dato-mysql-5024/
https://advenis.wordpress.com/2010/04/21/tipos-de-datos-en-mysql/
https://es.wikipedia.org/wiki/MySQL
https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/que-son-los-metadatos-y-cual-es-su-utilidad