



Nombre de la práctica	Operaciones CRUD 3			No.	1
Asignatura:	Taller de Base de datos	Carrera:	Ingeniería en Sistemas Computacionales	Duración de la práctica (Hrs)	

NOMBRE DEL ALUMNO: Vanesa Hernández Martínez  
GRUPO: 3501

II. Lugar de realización de la práctica (laboratorio, taller, aula u otro):  
Actividades en aula de clases y en equipo personal

III. Material empleado:

- Laptop
- Navicat

### Creación de las tabla

```
CREATE TABLE clientes (
    id_cliente INT PRIMARY KEY AUTO_INCREMENT,
    nombre VARCHAR(100) NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL CHECK (email
    LIKE '%_@%._%'),
    telefono VARCHAR(15) CHECK (LENGTH(telefono) >= 10),
    direccion VARCHAR(255) NOT NULL
);
```

### Ejercicios CREATE

1. Inserta un cliente válido en la tabla.

```
-- 1)
INSERT INTO clientes (nombre, email, telefono, direccion)
VALUES (1, 'Juan Pérez', 'juan.perez@gmail.com', '1234567890', 'Calle 123,
Ciudad de Mexico');
```

id_cliente	nombre	email	telefono	direccion
# int	varchar(100)	varchar(100)	varchar(15)	varchar(255)
1	Juan Pérez	juan.perez@gmail	1234567890	Calle 123, Ciudad

- Inserta un cliente sin especificar el campo id\_cliente.

```
-- 2)
INSERT INTO clientes (nombre, email, telefono, direccion)
VALUES ('Ana García', 'ana.garcia@example.com', '0987654321', 'Avenida 456, Ciudad ABC');
```

id_cliente # int	nombre nvarchar(100)	email nvarchar(100)	telefono nvarchar(15)	direccion nvarchar(255)
1	Juan Pérez	juan.perez@gmail	1234567890	Calle 123, Ciudad
2	Ana García	ana.garcia@exam	0987654321	Avenida 456, Ciud

- Intenta insertar un cliente con un formato de correo incorrecto (debería fallar).

```
-- 3)
INSERT INTO clientes (nombre, email, telefono, direccion) VALUES ('Carlos Ruiz', 'carlos.ruiz@incorrecto', '1112223334', 'Plaza 789, Ciudad DEF');
```

Query	Message	Query Time	Fetch Time
INSERT INTO clientes (nombre, email, telefono, direccion) VALUES ('Carlos Ruiz', 'carlos.ruiz@incorrecto', '1112223334', 'Plaza 789, Ciudad DEF')	3819 - Check constraint 'clientes_chk_1' is violated.	0.000s	0.000s

- Inserta múltiples clientes en una sola consulta.

```
-- 4)
INSERT INTO clientes (nombre, email, telefono, direccion)
VALUES
('Laura Martínez', 'laura.martinez@gmail.com', '1234509876', 'Calle 101, Ciudad GHI'),
('Pedro López', 'pedro.lopez@hotmail.com', '9876543210', 'Avenida 202, Ciudad JKL'),
('Marta Sánchez', 'marta.sanchez@mail.com', '4567890123', 'Plaza 303, Ciudad MNO');
```

id_cliente # int	nombre nvarchar(100)	email nvarchar(100)	telefono nvarchar(15)	direccion nvarchar(255)
1	Juan Pérez	juan.perez@gmail	1234567890	Calle 123, Ciudad
2	Ana García	ana.garcia@exam	0987654321	Avenida 456, Ciud
3	Laura Martínez	laura.martinez@g	1234509876	Calle 101, Ciudad
4	Pedro López	pedro.lopez@hoti	9876543210	Avenida 202, Ciud
5	Marta Sánchez	marta.sanchez@rr	4567890123	Plaza 303, Ciudad

5. Inserta un cliente con un número de teléfono de menos de 10 caracteres (debería fallar).

```
-- 5)
```

```
INSERT INTO clientes (nombre, email, telefono, direccion)
VALUES ('Mario Gómez', 'mario.gomez@hotmail.com', '123456', 'Calle 404,
Ciudad PQR');
```

Query	Message	Query Time	Fetch Time
INSERT INTO clientes (nombre, email, telefono, direccion) VALUES ('Mario Gómez', 'mario.gomez@hotmail.com', '123456',	3819 - Check constraint 'clientes_chk_2' is violated.	0.001s	0.000s

## Ejercicios READ

1. Consulta todos los registros de la tabla clientes.

```
-- 1)
```

```
SELECT * FROM clientes;
```

id_cliente	nombre	email	telefono	direccion
1	Juan Pérez	juan.perez@gmail.com	1234567890	Calle 123, Ciudad de Mexi
2	Ana García	ana.garcia@example.cc	0987654321	Avenida 456, Ciudad ABC
3	Laura Martínez	laura.martinez@gmail.c	1234509876	Calle 101, Ciudad GHI
4	Pedro López	pedro.lopez@hotmail.c	9876543210	Avenida 202, Ciudad JKL
5	Marta Sánchez	marta.sanchez@mail.co	4567890123	Plaza 303, Ciudad MNO

2. Consulta el nombre y email de todos los clientes.

```
-- 2)
```

```
SELECT nombre, email FROM clientes;
```

nombre	email
Juan Pérez	juan.perez@gmail.com
Ana García	ana.garcia@example.cc
Laura Martínez	laura.martinez@gmail.c
Pedro López	pedro.lopez@hotmail.c
Marta Sánchez	marta.sanchez@mail.co

3. Consulta los clientes cuyo número de teléfono empiece con "555".

```
-- 3)
SELECT * FROM clientes WHERE telefono LIKE '555%';
```

id_cliente	nombre	email	telefono	direccion
(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

4. Consulta los clientes cuyo nombre contenga "López".

```
-- 4)
SELECT * FROM clientes WHERE nombre LIKE '%López%';
```

id_cliente	nombre	email	telefono	direccion
4	Pedro López	pedro.lopez@hotmail.c	9876543210	Avenida 202, Ciudad JKL

5. Consulta los clientes ordenados por nombre en orden ascendente.

```
-- 5)
SELECT * FROM clientes ORDER BY nombre ASC;
```

id_cliente	nombre	email	telefono	direccion
2	Ana García	ana.garcia@example.cc	0987654321	Avenida 456, Ciudad ABC
1	Juan Pérez	juan.perez@gmail.com	1234567890	Calle 123, Ciudad de Mexi
3	Laura Martínez	laura.martinez@gmail.c	1234509876	Calle 101, Ciudad GHI
5	Marta Sánchez	marta.sanchez@mail.co	4567890123	Plaza 303, Ciudad MNO
4	Pedro López	pedro.lopez@hotmail.c	9876543210	Avenida 202, Ciudad JKL

6. Consulta el email de los clientes cuyo id sea par.

```
-- 6)
SELECT email FROM clientes WHERE id_cliente % 2 = 0;
```

email
ana.garcia@example.cc
pedro.lopez@hotmail.c

- Consulta los clientes con direcciones que contengan más de 10 caracteres.

-- 7)

```
SELECT * FROM clientes WHERE LENGTH(direccion) > 10;
```

id_cliente	nombre	email	telefono	direccion
1	Juan Pérez	juan.perez@gmail.com	1234567890	Calle 123, Ciudad de Mexi
2	Ana García	ana.garcia@example.cc	0987654321	Avenida 456, Ciudad ABC
3	Laura Martínez	laura.martinez@gmail.c	1234509876	Calle 101, Ciudad GHI
4	Pedro López	pedro.lopez@hotmail.c	9876543210	Avenida 202, Ciudad JKL
5	Marta Sánchez	marta.sanchez@mail.co	4567890123	Plaza 303, Ciudad MNO

## Ejercicios UPDATE

- Actualiza el número de teléfono de un cliente específico.

-- 1)

```
UPDATE clientes SET telefono = '9876543210' WHERE id_cliente = 1;
```

- Cambia el email de un cliente con un id\_cliente específico.8

-- 2)

```
UPDATE clientes SET email = 'nuevo.email@gmail.com'
WHERE id_cliente = 8;
```

- Intenta actualizar el correo de un cliente a un email que ya existe (debería fallar).

-- 3)

```
UPDATE clientes SET email = 'ana.garcia@example.com'
WHERE id_cliente = 3;
```

Query	Message	Query Time	Fetch Time
UPDATE clientes SET email = 'ana.garcia@example.com' WHERE id_cliente = 3	1062 - Duplicate entry 'ana.garcia@example.com' for key 'clientes.email'	0.033s	0.000s

- Actualiza la dirección de todos los clientes cuyos nombres contengan "López".

-- 4)

```
UPDATE clientes SET direccion = 'Nueva Dirección, Ciudad XYZ'
WHERE nombre LIKE '%López%';
```

5. Incrementa los id\_cliente de todos los clientes en 10 (esto es solo un ejercicio teórico).

-- 5)

```
UPDATE clientes SET id_cliente = id_cliente + 10;
```

## Tabla después de los ejercicios de update

id_cliente # int	nombre nvarchar(100)	email nvarchar(100)	telefono nvarchar(15)	direccion nvarchar(255)
11	Juan Pérez	juan.perez@gmail.com	9876543210	Calle 123, Ciudad
12	Ana García	ana.garcia@example.com	0987654321	Avenida 456, Ciudad
13	Laura Martínez	laura.martinez@gmail.com	1234509876	Calle 101, Ciudad
14	Pedro López	pedro.lopez@hotmail.com	9876543210	Nueva Dirección, Ciudad
15	Marta Sánchez	marta.sanchez@mail.com	4567890123	Plaza 303, Ciudad

## Ejercicios DELETE

1. Elimina un cliente específico con un id\_cliente dado.

-- 1)

```
DELETE FROM clientes WHERE id_cliente = 11;
```

id_cliente # int	nombre nvarchar(100)	email nvarchar(100)	telefono nvarchar(15)	direccion nvarchar(255)
12	Ana García	ana.garcia@example.com	0987654321	Avenida 456, Ciudad
13	Laura Martínez	laura.martinez@gmail.com	1234509876	Calle 101, Ciudad
14	Pedro López	pedro.lopez@hotmail.com	9876543210	Nueva Dirección, Ciudad
15	Marta Sánchez	marta.sanchez@mail.com	4567890123	Plaza 303, Ciudad

2. Elimina todos los clientes que tengan un número de teléfono que empiece con "555".

-- 2)

```
DELETE FROM clientes WHERE telefono LIKE '555%';
```

3. Elimina todos los clientes cuyo nombre contenga "Gómez".

-- 3)

```
DELETE FROM clientes WHERE nombre LIKE '%Gómez%';
```

4. Elimina todos los clientes con direcciones que contengan menos de 10 caracteres.

-- 4)

```
DELETE FROM clientes WHERE LENGTH(direccion) < 10;
```

5. Elimina todos los registros de la tabla clientes (¡CUIDADO!).

-- 5)

```
DELETE FROM clientes;
```

id_cliente	nombre	email	telefono	direccion
# int	varchar(100)	varchar(100)	varchar(15)	varchar(255)
(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

## Conclusión

Las operaciones básicas de manipulación de bases de datos como las que hemos realizado (inserciones, actualizaciones, consultas y eliminaciones) son fundamentales para el manejo eficiente de la información en cualquier sistema. A través de estas consultas, hemos aprendido cómo agregar y validar datos, realizar actualizaciones selectivas, y eliminar información según criterios específicos, manteniendo siempre la integridad de los datos mediante restricciones como UNIQUE, CHECK, y la longitud de los campos.

Es importante tener cuidado al realizar operaciones de eliminación y actualización, especialmente cuando se trata de claves primarias o grandes conjuntos de datos, ya que un error puede impactar seriamente en la integridad de la base de datos. Así, el trabajo cuidadoso con consultas SQL garantiza que las bases de datos funcionen de manera óptima y confiable, permitiendo un acceso rápido y preciso a la información.