

Nombre de la práctica	OPERACIONES CRUD			No.	01
Asignatura:	Taller de bases de datos	Carrera:	Ingeniería sistemas	en	Duración de la práctica (Hrs)

Nombre del alumno: Jesús Navarrete Martínez  
Grupo: 3501

I. Competencia(s) específica(s):

>=

II. Lugar de realización de la práctica (laboratorio, taller, aula u otro): Aula

III. Material empleado:

- Equipo de computo

IV. Desarrollo de la práctica:

## Problemario para operaciones CRUD

### CREACION DE LA BASE DE DATOS

```
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) NULL  
);
```

### Ejercicios CREATE

1. Inserta un cliente válido en la tabla.

Solución:

```
INSERT INTO clientes (nombre, email, telefono, direccion)  
VALUES ('Juan Pérez', 'juan.perez@example.com', '3334567890', 'Calle Falsa 123');
```

2. Inserta un cliente sin especificar el campo `id_cliente`.

Solución:

```
INSERT INTO clientes (nombre, email, telefono, direccion)
VALUES ('María López', 'maria.lopez@example.com', '3345678901', 'Avenida Siempre
Viva 456');
```

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

**Solución:**

```
INSERT INTO clientes (nombre, email, telefono, direccion)
VALUES ('Carlos González', 'carlos.gonzalezexample.com', '3356789012', 'Boulevard
del Sol 789');
```

4. Inserta múltiples clientes en una sola consulta.

## Solución:

```
INSERT INTO clientes (nombre, email, telefono, direccion)
VALUES
('Luis Martínez', 'luis.martinez@example.com', '3367890123', 'Calle Luna 100'),
('Ana Torres', 'ana.torres@example.com', '3378901234', 'Calle Sol 200'),
('Pedro Sánchez', 'pedro.sanchez@example.com', '3389012345', 'Calle Estrella
300');
```

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

## Solución:

```
INSERT INTO clientes (nombre, email, telefono, direccion)
VALUES ('Julia Ramírez', 'julia.ramirez@example.com', '3456789', 'Calle Cometa
400');
```

## Ejercicios READ

1. Consulta todos los registros de la tabla `clientes`.

## Solución:

```
SELECT * FROM clientes;
```

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

## Solución:

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

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

## Solución:

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

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

## Solución:

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

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

**Solución:**

```
SELECT * FROM clientes ORDER BY nombre ASC;
```

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

**Solución:**

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

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

**Solución:**

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

## Ejercicios UPDATE

1. Actualiza el número de teléfono de un cliente específico.

**Solución:**

```
UPDATE clientes SET telefono = '3398765432' WHERE id_cliente = 1;
```

2. Cambia el `email` de un cliente con un `id_cliente` específico.

**Solución:**

```
UPDATE clientes SET email = 'nuevo.email@example.com' WHERE id_cliente = 2;
```

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

**Solución:**

```
UPDATE clientes SET email = 'juan.perez@example.com' WHERE id_cliente = 3;
```

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

**Solución:**

```
UPDATE clientes SET direccion = 'Nueva Dirección, Calle Actualizada 123' WHERE  
nombre LIKE '%López%';
```

5. Incrementa los `id_cliente` de todos los clientes en 10 (esto es solo un ejercicio teórico).

**Solución:**

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

## Ejercicios DELETE

1. Elimina un cliente específico con un `id_cliente` dado.

**Solución:**

```
DELETE FROM clientes WHERE id_cliente = 1;
```

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

**Solución:**

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

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

**Solución:**

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

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

**Solución:**

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

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

**Solución:**

```
DELETE FROM clientes;
```

## V. Conclusiones:

La implementación de las operaciones CRUD es fundamental para la gestión eficiente de bases de datos en aplicaciones modernas. A lo largo de la práctica, se demostró cómo cada una de estas operaciones permite la interacción con los datos de manera estructurada y organizada. Crear registros, leer información, actualizar datos existentes y eliminar entradas son acciones clave para el ciclo de vida de la información. Además, se evidenció la importancia de validar los datos y asegurar que las operaciones sean realizadas de manera segura y precisa, minimizando el riesgo de errores y asegurando la integridad de la base de datos. En resumen, el dominio de CRUD es esencial para el desarrollo de aplicaciones sólidas y escalables.