

-- Procedimientos almacenados para Clientes

DELIMITER //

-- Crear Cliente

```
CREATE PROCEDURE sp_crear_cliente(
    IN p_rut VARCHAR(12),
    IN p_nombre_completo VARCHAR(100),
    IN p_direccion VARCHAR(200),
    IN p_comuna VARCHAR(50),
    IN p_correo_electronico VARCHAR(100),
    IN p_telefono VARCHAR(20)
)
BEGIN
    INSERT INTO clientes (rut, nombre_completo, direccion, comuna,
    correo_electronico, telefono)
    VALUES (p_rut, p_nombre_completo, p_direccion, p_comuna,
    p_correo_electronico, p_telefono);
    SELECT LAST_INSERT_ID() AS id;
END //
```

-- Leer Cliente

```
CREATE PROCEDURE sp_leer_cliente(IN p_id INT)
BEGIN
    SELECT * FROM clientes WHERE id = p_id;
END //
```

-- Actualizar Cliente

```
CREATE PROCEDURE sp_actualizar_cliente(
    IN p_id INT,
    IN p_rut VARCHAR(12),
    IN p_nombre_completo VARCHAR(100),
    IN p_direccion VARCHAR(200),
    IN p_comuna VARCHAR(50),
    IN p_correo_electronico VARCHAR(100),
    IN p_telefono VARCHAR(20)
)
BEGIN
    UPDATE clientes
    SET rut = p_rut,
        nombre_completo = p_nombre_completo,
        direccion = p_direccion,
        comuna = p_comuna,
        correo_electronico = p_correo_electronico,
        telefono = p_telefono
    WHERE id = p_id;
END //
```

-- Eliminar Cliente

```
CREATE PROCEDURE sp_eliminar_cliente(IN p_id INT)
BEGIN
    DELETE FROM clientes WHERE id = p_id;
END //
```

```

-- Procedimientos almacenados para Equipos

-- Crear Equipo
CREATE PROCEDURE sp_crear_equipo(
    IN p_modelo VARCHAR(100),
    IN p_cpu VARCHAR(100),
    IN p_disco_duro INT,
    IN p_ram INT,
    IN p_precio DECIMAL(10, 2),
    IN p_tipo ENUM('desktop', 'laptop'),
    IN p_potencia_fuente INT,
    IN p_factor_forma VARCHAR(20),
    IN p_tamano_pantalla DECIMAL(4, 2),
    IN p_es_touch BOOLEAN,
    IN p_puertos_usb INT
)
BEGIN
    INSERT INTO equipos (modelo, cpu, disco_duro, ram, precio, tipo)
    VALUES (p_modelo, p_cpu, p_disco_duro, p_ram, p_precio, p_tipo);

    SET @last_id = LAST_INSERT_ID();

    IF p_tipo = 'desktop' THEN
        INSERT INTO desktops (id, potencia_fuente, factor_forma)
        VALUES (@last_id, p_potencia_fuente, p_factor_forma);
    ELSE
        INSERT INTO laptops (id, tamano_pantalla, es_touch, puertos_usb)
        VALUES (@last_id, p_tamano_pantalla, p_es_touch, p_puertos_usb);
    END IF;

    SELECT @last_id AS id;
END //

-- Leer Equipo
CREATE PROCEDURE sp_leer_equipo(IN p_id INT)
BEGIN
    SELECT e.*,
           d.potencia_fuente, d.factor_forma,
           l.tamano_pantalla, l.es_touch, l.puertos_usb
    FROM equipos e
    LEFT JOIN desktops d ON e.id = d.id
    LEFT JOIN laptops l ON e.id = l.id
    WHERE e.id = p_id;
END //

```

```

-- Actualizar Equipo
CREATE PROCEDURE sp_actualizar_equipo(
    IN p_id INT,
    IN p_modelo VARCHAR(100),
    IN p_cpu VARCHAR(100),
    IN p_disco_duro INT,
    IN p_ram INT,
    IN p_precio DECIMAL(10, 2),
    IN p_tipo ENUM('desktop', 'laptop'),
    IN p_potencia_fuente INT,
    IN p_factor_forma VARCHAR(20),
    IN p_tamano_pantalla DECIMAL(4, 2),
    IN p_es_touch BOOLEAN,
    IN p_puertos_usb INT
)
BEGIN
    UPDATE equipos
    SET modelo = p_modelo,
        cpu = p_cpu,
        disco_duro = p_disco_duro,
        ram = p_ram,
        precio = p_precio,
        tipo = p_tipo
    WHERE id = p_id;

    IF p_tipo = 'desktop' THEN
        UPDATE desktops
        SET potencia_fuente = p_potencia_fuente,
            factor_forma = p_factor_forma
        WHERE id = p_id;
        DELETE FROM laptops WHERE id = p_id;
    ELSE
        UPDATE laptops
        SET tamano_pantalla = p_tamano_pantalla,
            es_touch = p_es_touch,
            puertos_usb = p_puertos_usb
        WHERE id = p_id;
        DELETE FROM desktops WHERE id = p_id;
    END IF;
END //

-- Eliminar Equipo
CREATE PROCEDURE sp_eliminar_equipo(IN p_id INT)
BEGIN
    DELETE FROM desktops WHERE id = p_id;
    DELETE FROM laptops WHERE id = p_id;
    DELETE FROM equipos WHERE id = p_id;
END //

```

-- Procedimientos almacenados para Ventas

-- Crear Venta

```
CREATE PROCEDURE sp_crear_venta(  
    IN p_cliente_id INT,  
    IN p_equipo_id INT  
)  
BEGIN  
    INSERT INTO ventas (cliente_id, equipo_id, fecha_hora)  
    VALUES (p_cliente_id, p_equipo_id, NOW());  
    SELECT LAST_INSERT_ID() AS id;  
END //
```

-- Leer Venta

```
CREATE PROCEDURE sp_leer_venta(IN p_id INT)  
BEGIN  
    SELECT v.*,  
           c.rut, c.nombre_completo, c.correo_electronico, c.telefono,  
           e.modelo, e.precio, e.tipo  
    FROM ventas v  
    JOIN clientes c ON v.cliente_id = c.id  
    JOIN equipos e ON v.equipo_id = e.id  
    WHERE v.id = p_id;  
END //
```

-- Actualizar Venta

```
CREATE PROCEDURE sp_actualizar_venta(  
    IN p_id INT,  
    IN p_cliente_id INT,  
    IN p_equipo_id INT,  
    IN p_fecha_hora DATETIME  
)  
BEGIN  
    UPDATE ventas  
    SET cliente_id = p_cliente_id,  
        equipo_id = p_equipo_id,  
        fecha_hora = p_fecha_hora  
    WHERE id = p_id;  
END //
```

-- Eliminar Venta

```
CREATE PROCEDURE sp_eliminar_venta(IN p_id INT)  
BEGIN  
    DELETE FROM ventas WHERE id = p_id;  
END //
```

DELIMITER ;