

**Campus**

Campus Cúmbres

**Materia**

Base de Datos

**Actividad**

Actividad 4

**Maestra**

Edwige Jazzmín Novelo Villegas

**Alumnos**

Juan Porfirio Torres Rojas

#07099471

Jesus David Marroquin Peña

#07101520

# Índice

[**Índice 2**](#_4rqyx4o3a3cc)

[**Instrucciones 3**](#_pxo7xjs5zk01)

[**La base de datos 4**](#_oj3du1je0afg)

[**Datos 6**](#_1uqk9k7rnujp)

[**EJERCICIOS CASE 9**](#_jamwc0d1obs5)

[**EER Diagrama 17**](#_pgnzrpw54yuy)

[**Liga GitHub 17**](#_69vuuy63jn2m)

# Instrucciones

Esta actividad se puede realizar de forma individual o en equipos de hasta tres personas.

Actividad: Realicen los ejercicios de CASE y de Procedimientos que se encuentran en nuestro Drive, el documento se llama Actividad4\_Ejercicios a entregar.

El código SQL de la creación de los CASE y Procedimientos (con su ejecución y comprobación).

Capturas de pantalla de los resultados de la ejecución de cada consulta SQL y la explicación pertinente.

Enlace de Código: Proporcionen un enlace a su código SQL subido en una plataforma como GitHub.

Archivos .sql: Incluyan los archivos .sql.

# 

# 

# 

# 

# 

# 

# La base de datos

CREATE DATABASE IF NOT EXISTS veterinaria;

USE veterinaria;

CREATE TABLE propietarios (

id\_propietario INT PRIMARY KEY AUTO\_INCREMENT,

nombre VARCHAR(100) NOT NULL,

apellido VARCHAR(100) NOT NULL,

telefono VARCHAR(15),

email VARCHAR(100),

direccion TEXT,

fecha\_registro DATE DEFAULT (CURRENT\_DATE)

);

CREATE TABLE mascotas (

id\_mascota INT PRIMARY KEY AUTO\_INCREMENT,

nombre VARCHAR(50) NOT NULL,

especie VARCHAR(30) NOT NULL,

raza VARCHAR(50),

edad INT,

peso DECIMAL(5,2),

color VARCHAR(30),

id\_propietario INT NOT NULL,

fecha\_registro DATE DEFAULT (CURRENT\_DATE),

FOREIGN KEY (id\_propietario) REFERENCES propietarios(id\_propietario)

);

CREATE TABLE veterinarios (

id\_veterinario INT PRIMARY KEY AUTO\_INCREMENT,

nombre VARCHAR(100) NOT NULL,

apellido VARCHAR(100) NOT NULL,

cedula\_profesional VARCHAR(20) UNIQUE,

especialidad VARCHAR(100),

telefono VARCHAR(15),

email VARCHAR(100)

);

CREATE TABLE citas (

id\_cita INT PRIMARY KEY AUTO\_INCREMENT,

fecha\_cita DATE NOT NULL,

hora\_cita TIME NOT NULL,

motivo TEXT,

estado VARCHAR(20) DEFAULT 'Programada',

id\_mascota INT NOT NULL,

id\_veterinario INT NOT NULL,

FOREIGN KEY (id\_mascota) REFERENCES mascotas(id\_mascota),

FOREIGN KEY (id\_veterinario) REFERENCES veterinarios(id\_veterinario)

);

CREATE TABLE tratamientos (

id\_tratamiento INT PRIMARY KEY AUTO\_INCREMENT,

nombre\_tratamiento VARCHAR(100) NOT NULL,

descripcion TEXT,

precio DECIMAL(10,2),

duracion\_minutos INT

);

CREATE TABLE historial\_medico (

id\_historial INT PRIMARY KEY AUTO\_INCREMENT,

fecha\_consulta DATE NOT NULL,

diagnostico TEXT,

observaciones TEXT,

peso\_actual DECIMAL(5,2),

temperatura DECIMAL(4,1),

id\_mascota INT NOT NULL,

id\_veterinario INT NOT NULL,

id\_tratamiento INT,

FOREIGN KEY (id\_mascota) REFERENCES mascotas(id\_mascota),

FOREIGN KEY (id\_veterinario) REFERENCES veterinarios(id\_veterinario),

FOREIGN KEY (id\_tratamiento) REFERENCES tratamientos(id\_tratamiento)

);

CREATE TABLE personas (

id\_persona INT PRIMARY KEY AUTO\_INCREMENT,

nombre VARCHAR(100) NOT NULL,

apellido VARCHAR(100) NOT NULL

);

CREATE TABLE invitados (

id\_invitado INT PRIMARY KEY AUTO\_INCREMENT,

id\_persona INT NOT NULL,

metodo\_invitacion VARCHAR(20),

FOREIGN KEY (id\_persona) REFERENCES personas(id\_persona)

);

CREATE TABLE confirmaciones (

id\_confirmacion INT PRIMARY KEY AUTO\_INCREMENT,

id\_persona INT NOT NULL,

estado VARCHAR(20),

FOREIGN KEY (id\_persona) REFERENCES personas(id\_persona)

);

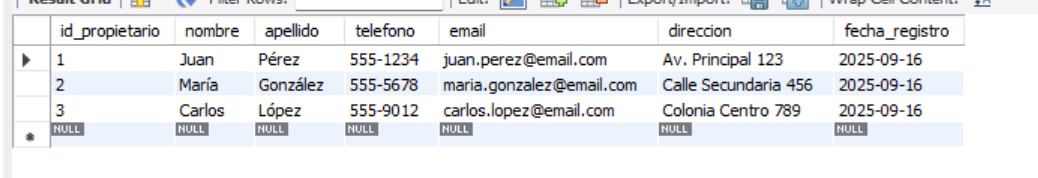
# Datos

INSERT INTO propietarios (nombre, apellido, telefono, email, direccion) VALUES

('Juan', 'Pérez', '555-1234', 'juan.perez@email.com', 'Av. Principal 123'),

('María', 'González', '555-5678', 'maria.gonzalez@email.com', 'Calle Secundaria 456'),

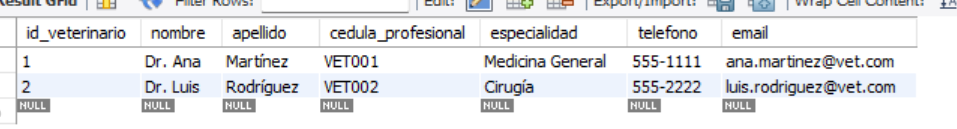
('Carlos', 'López', '555-9012', 'carlos.lopez@email.com', 'Colonia Centro 789');



INSERT INTO veterinarios (nombre, apellido, cedula\_profesional, especialidad, telefono, email) VALUES

('Dr. Ana', 'Martínez', 'VET001', 'Medicina General', '555-1111', 'ana.martinez@vet.com'),

('Dr. Luis', 'Rodríguez', 'VET002', 'Cirugía', '555-2222', '[luis.rodriguez@vet.com](mailto:luis.rodriguez@vet.com)');



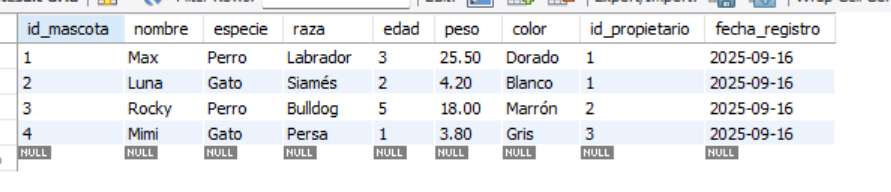
INSERT INTO mascotas (nombre, especie, raza, edad, peso, color, id\_propietario) VALUES

('Max', 'Perro', 'Labrador', 3, 25.5, 'Dorado', 1),

('Luna', 'Gato', 'Siamés', 2, 4.2, 'Blanco', 1),

('Rocky', 'Perro', 'Bulldog', 5, 18.0, 'Marrón', 2),

('Mimi', 'Gato', 'Persa', 1, 3.8, 'Gris', 3);



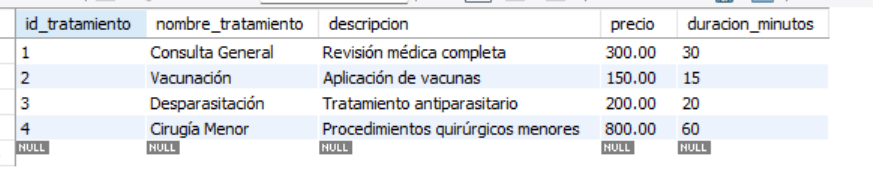
INSERT INTO tratamientos (nombre\_tratamiento, descripcion, precio, duracion\_minutos) VALUES

('Consulta General', 'Revisión médica completa', 300.00, 30),

('Vacunación', 'Aplicación de vacunas', 150.00, 15),

('Desparasitación', 'Tratamiento antiparasitario', 200.00, 20),

('Cirugía Menor', 'Procedimientos quirúrgicos menores', 800.00, 60);



INSERT INTO personas (nombre, apellido) VALUES

('Miguel', 'González'),

('María', 'López'),

('José', 'Martínez'),

('Ana', 'Hernández'),

('Carlos', 'García'),

('Laura', 'Rodríguez'),

('Roberto', 'Pérez'),

('Sofía', 'Sánchez'),

('Diego', 'Ramírez'),

('Valentina', 'Torres');



INSERT INTO invitados (id\_persona, metodo\_invitacion) VALUES

(1, 'WhatsApp'),

(2, 'WhatsApp'),

(3, 'Email'),

(4, 'WhatsApp'),

(5, 'Llamada'),

(6, 'WhatsApp'),

(7, 'En persona'),

(8, 'Email'),

(9, 'WhatsApp'),

(10, 'Email');



INSERT INTO confirmaciones (id\_persona, estado) VALUES

(1, 'Confirma'),

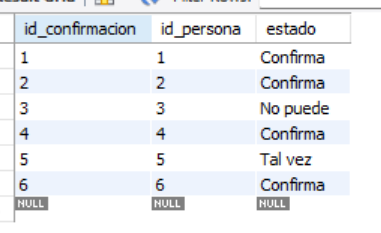
(2, 'Confirma'),

(3, 'No puede'),

(4, 'Confirma'),

(5, 'Tal vez'),

(6, 'Confirma');



# EJERCICIOS CASE

-- Ejercicio 1: CASE SIMPLE

SELECT 

nombre,

apellido,

metodo\_invitacion,

CASE metodo\_invitacion

WHEN 'WhatsApp' THEN '🟢 Mensaje'

WHEN 'Email' THEN '🟣 Correo'

WHEN 'Llamada' THEN '🟤 Teléfono'

WHEN 'En persona' THEN '🟥 Cara a cara'

ELSE '❓ Desconocido'

END as tipo\_invitacion

FROM personas p

JOIN invitados i ON p.id\_persona = i.id\_persona;

-- Ejercicio 2: CASE SEARCHED

SELECT 

p.nombre,

p.apellido,

c.estado as confirmacion,

CASE

WHEN c.estado = 'Confirma' THEN '☑ Viene seguro'

WHEN c.estado = 'No puede' THEN '✗ No puede venir'

WHEN c.estado = 'Tal vez' THEN '🟩 Tal vez viene'

WHEN c.estado IS NULL THEN '🟪 Sin respuesta'

ELSE '💡 Estado raro'

END as situacion

FROM personas p

LEFT JOIN confirmaciones c ON p.id\_persona = c.id\_persona

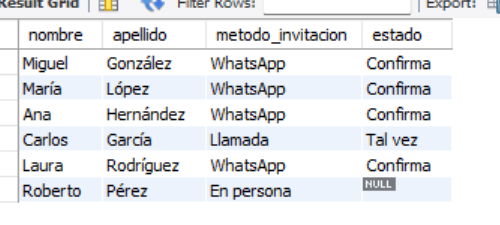
WHERE p.id\_persona IN (SELECT id\_persona FROM invitados);

-- Ejercicio 3: CASE SEARCHED en WHERE

SELECT

p.nombre,

p.apellido,

i.metodo\_invitacion, 

c.estado

FROM personas p

LEFT JOIN invitados i ON p.id\_persona = i.id\_persona

LEFT JOIN confirmaciones c ON p.id\_persona = c.id\_persona

WHERE

CASE

WHEN i.metodo\_invitacion = 'WhatsApp' THEN c.estado = 'Confirma'

WHEN i.metodo\_invitacion = 'Email' THEN c.estado IN ('Confirma', 'Tal vez')

ELSE TRUE

END;

-- PROCEDIMIENTOS ALMACENADOS

-- Ejercicio 4: Procedimiento sin parámetros

DELIMITER //

CREATE PROCEDURE mostrar\_todas\_mascotas()

BEGIN

SELECT

m.id\_mascota as 'ID Mascota',

m.nombre as 'Nombre Mascota',

m.especie as 'Especie',

m.raza as 'Raza',

m.edad as 'Edad',

m.peso as 'Peso (kg)',

CONCAT(p.nombre, ' ', p.apellido) as 'Propietario',

p.telefono as 'Teléfono Propietario'

FROM mascotas m

INNER JOIN propietarios p ON m.id\_propietario = p.id\_propietario

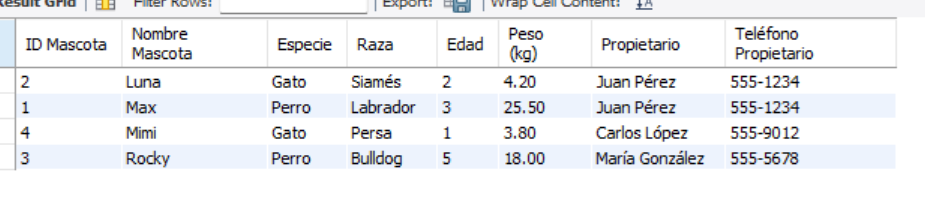
ORDER BY m.nombre;

END //

DELIMITER ;

-- Ejecutar procedimiento sin parámetros (Ejercicio 4)

CALL mostrar\_todas\_mascotas();



-- Ejercicio 5: Procedimiento con IN

DELIMITER //

CREATE PROCEDURE buscar\_mascota\_por\_id(IN p\_id INT)

BEGIN

IF p\_id <= 0 THEN

SELECT 'Error: El ID debe ser mayor a 0' as Mensaje;

ELSE

SELECT

id\_mascota as 'ID',

nombre as 'Nombre',

especie as 'Especie',

raza as 'Raza',

edad as 'Edad (años)',

peso as 'Peso (kg)'

FROM mascotas

WHERE id\_mascota = p\_id;

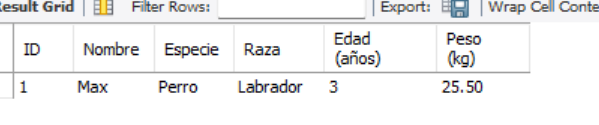
END IF;

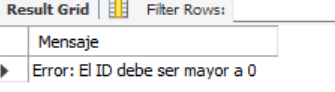
END //

DELIMITER ;

-- Ejecutar procedimiento con IN (Ejercicio 5)

CALL buscar\_mascota\_por\_id(1);

CALL buscar\_mascota\_por\_id(0);



-- Ejercicio 6: Procedimiento con OUT

DELIMITER //

CREATE PROCEDURE contar\_mascotas(OUT p\_total INT)

BEGIN

SELECT COUNT(\*) INTO p\_total FROM mascotas;

END //

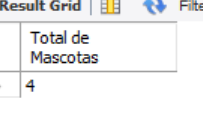
DELIMITER ;

-- Ejecutar procedimiento con OUT (Ejercicio 6)

SET @resultado = 0;

CALL contar\_mascotas(@resultado);

SELECT @resultado as "Total de Mascotas";



-- Ejercicio 7: Procedimiento con IN y OUT

DELIMITER //

CREATE PROCEDURE actualizar\_peso\_mascota(IN p\_id INT, INOUT p\_peso DECIMAL(5,2))

BEGIN

DECLARE peso\_anterior DECIMAL(5,2) DEFAULT 0;

DECLARE mascota\_existe INT DEFAULT 0;

SELECT COUNT(\*) INTO mascota\_existe

FROM mascotas

WHERE id\_mascota = p\_id;

IF mascota\_existe = 0 THEN

SET p\_peso = -1;

ELSE

SELECT peso INTO peso\_anterior

FROM mascotas

WHERE id\_mascota = p\_id;

UPDATE mascotas

SET peso = p\_peso

WHERE id\_mascota = p\_id;

SET p\_peso = peso\_anterior;

END IF;

END //

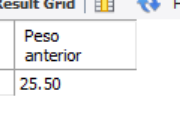
DELIMITER ;

-- Ejecutar procedimiento con INOUT (Ejercicio 7)

SET @nuevo\_peso = 27.8;

CALL actualizar\_peso\_mascota(1, @nuevo\_peso);

SELECT @nuevo\_peso as "Peso anterior";



-- Ejercicio 8: Procedimiento de mantenimiento

DELIMITER //

CREATE PROCEDURE agregar\_nueva\_mascota(

IN p\_nombre VARCHAR(50),

IN p\_especie VARCHAR(30),

IN p\_raza VARCHAR(50),

IN p\_edad INT,

IN p\_peso DECIMAL(5,2),

IN p\_color VARCHAR(30),

IN p\_id\_propietario INT

)

BEGIN

DECLARE propietario\_existe INT DEFAULT 0;

SELECT COUNT(\*) INTO propietario\_existe

FROM propietarios

WHERE id\_propietario = p\_id\_propietario;

IF propietario\_existe = 0 THEN

SELECT 'Error: El propietario no existe' as Mensaje;

ELSE

INSERT INTO mascotas (nombre, especie, raza, edad, peso, color, id\_propietario)

VALUES (p\_nombre, p\_especie, p\_raza, p\_edad, p\_peso, p\_color, p\_id\_propietario);

SELECT 'Mascota agregada exitosamente' as Mensaje;

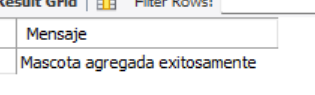
END IF;

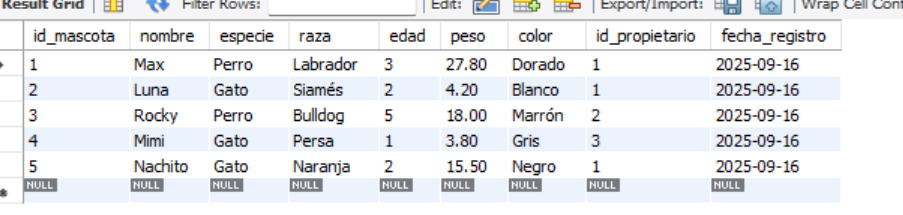
END //

DELIMITER ;

-- Ejecutar procedimiento de mantenimiento (Ejercicio 8)

CALL agregar\_nueva\_mascota('Nachito', 'Gato', 'Naranja', 2, 15.5, 'Negro', 1);





-- Ejercicio 9: Procedimiento de validación

DELIMITER //

CREATE PROCEDURE validar\_edad\_mascota(IN p\_id\_mascota INT, OUT p\_es\_valida VARCHAR(50))

BEGIN

DECLARE v\_edad INT DEFAULT 0;

DECLARE mascota\_existe INT DEFAULT 0;

SELECT COUNT(\*) INTO mascota\_existe

FROM mascotas

WHERE id\_mascota = p\_id\_mascota;

IF mascota\_existe > 0 THEN

SELECT edad INTO v\_edad

FROM mascotas

WHERE id\_mascota = p\_id\_mascota;

END IF;

IF mascota\_existe = 0 THEN

SET p\_es\_valida = 'Error: Mascota no encontrada';

ELSEIF v\_edad <= 0 THEN

SET p\_es\_valida = 'Error: Edad inválida';

ELSE

IF v\_edad <= 25 THEN

SET p\_es\_valida = 'Edad válida';

ELSE

SET p\_es\_valida = 'Edad muy alta (máximo 25 años)';

END IF;

END IF;

END //

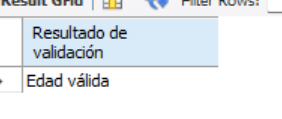
DELIMITER ;

-- Ejecutar procedimiento de validación (Ejercicio 9)

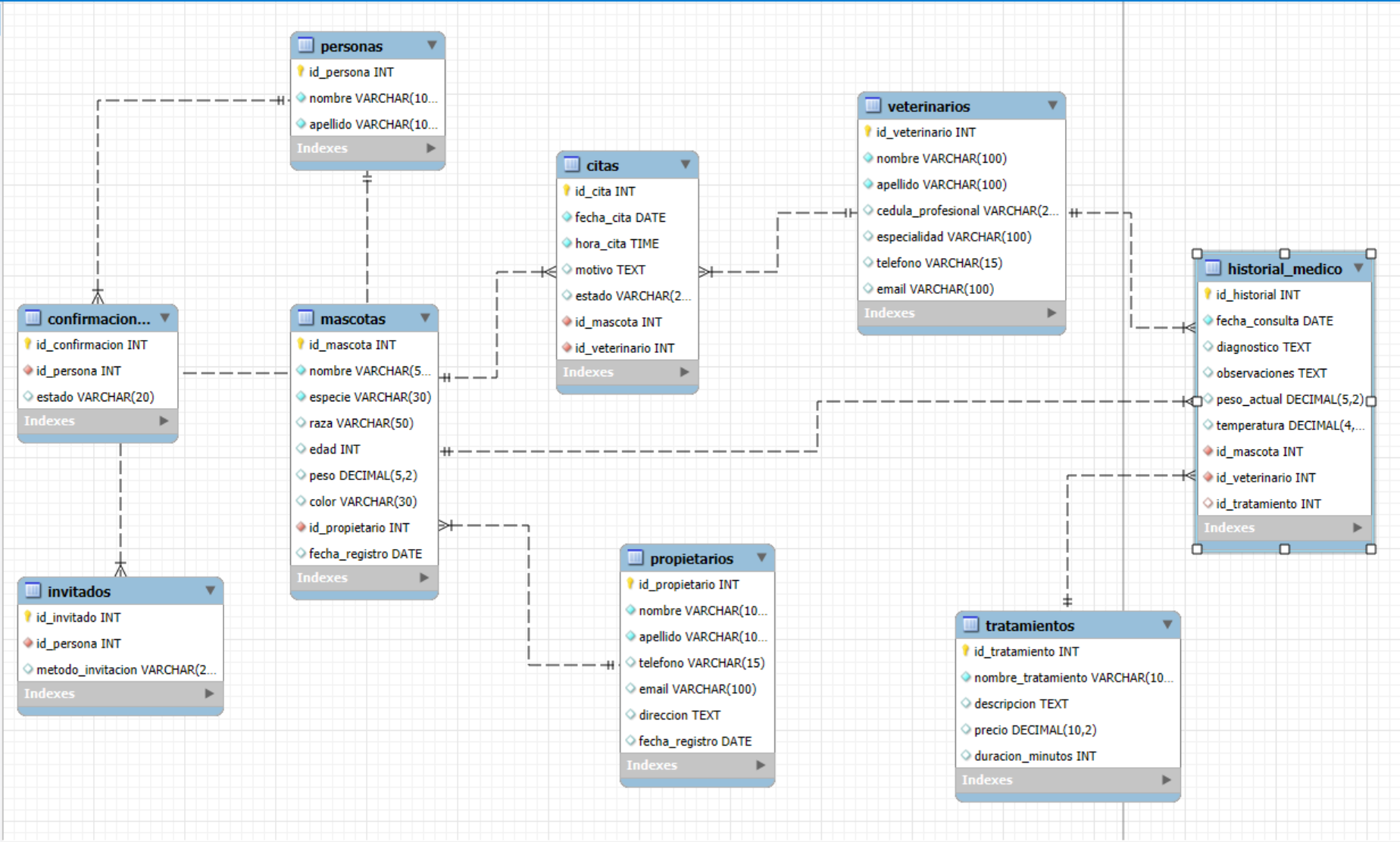
SET @resultado = '';

CALL validar\_edad\_mascota(3, @resultado);

SELECT @resultado as "Resultado de validación";



# EER Diagrama



# Liga GitHub

<https://github.com/enderkai00-png/Actividad-4-Base-de-datos/upload>

# 