# -- Creación de la base de datos "estacionamientos".

CREATE DATABASE IF NOT EXISTS estacionamientos;

USE estacionamientos;

# -- Creación de las tablas.

-- Creación de la tabla Usuarios

CREATE TABLE IF NOT EXISTS usuarios (

id INTEGER PRIMARY KEY AUTO\_INCREMENT,

nombre TEXT NOT NULL,

contrasenya TEXT NOT NULL

);

-- Creación de la tabla Estacionamientos

CREATE TABLE IF NOT EXISTS estacionamientos (

id INTEGER PRIMARY KEY AUTO\_INCREMENT,

estacionamiento TEXT NOT NULL,

ubicacion TEXT NOT NULL,

telefono VARCHAR(10) NOT NULL,

tiempo\_tolerancia INTEGER NOT NULL,

administrador\_id INTEGER NOT NULL,

FOREIGN KEY (administrador\_id) REFERENCES usuarios (id)

);

-- Creación de la tabla Vehiculos

CREATE TABLE IF NOT EXISTS vehiculos (

id INTEGER PRIMARY KEY AUTO\_INCREMENT,

hora\_ingreso TIMESTAMP NOT NULL,

hora\_salida TIMESTAMP,

pago REAL,

estacionamiento\_id INTEGER NOT NULL,

tiempo\_estacionado INTEGER,

cobro\_id INTEGER,

registrado TINYINT,

transaccion\_completada TINYINT DEFAULT 0, -- Nueva columna para controlar la transacción

FOREIGN KEY (estacionamiento\_id) REFERENCES estacionamientos (id)

);

-- Creación de la tabla Cobros

CREATE TABLE IF NOT EXISTS cobros (

id INTEGER PRIMARY KEY AUTO\_INCREMENT,

estacionamiento\_id INTEGER NOT NULL,

tiempo\_min INTEGER NOT NULL,

tiempo\_max INTEGER NOT NULL,

costo REAL NOT NULL,

FOREIGN KEY (estacionamiento\_id) REFERENCES estacionamientos (id)

);

# -- Creación de los triggers & procedimiento.

-- Creación del procedimiento almacenado sp\_actualizar\_vehiculo

DELIMITER //

CREATE PROCEDURE sp\_actualizar\_vehiculo(

IN p\_id INTEGER,

IN p\_hora\_salida TIMESTAMP

)

BEGIN

DECLARE v\_registrado TINYINT;

DECLARE v\_hora\_ingreso TIMESTAMP;

DECLARE v\_cobro\_id INTEGER;

DECLARE v\_tiempo\_estacionado INTEGER;

DECLARE v\_costo REAL;

-- Obtener el estado de 'registrado' y 'hora\_ingreso' del vehículo actual

SELECT registrado, hora\_ingreso, cobro\_id INTO v\_registrado, v\_hora\_ingreso, v\_cobro\_id FROM vehiculos WHERE id = p\_id;

-- Si el vehículo está registrado y la transacción ha sido completada, no permitir cambios en la hora de ingreso

IF v\_registrado = 1 AND p\_hora\_salida IS NOT NULL AND p\_hora\_salida <> v\_hora\_ingreso AND transaccion\_completada = 1 THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'No se permite editar la hora de ingreso una vez que se ha registrado y completado la transacción del vehículo.';

ELSE

-- Realizar la actualización de la hora de salida del vehículo

UPDATE vehiculos

SET hora\_salida = p\_hora\_salida

WHERE id = p\_id;

-- Calcular el tiempo estacionado y el costo si el vehículo ha sido registrado y la hora de salida no es nula

IF v\_registrado = 1 AND p\_hora\_salida IS NOT NULL THEN

SELECT tiempo\_min, tiempo\_max, costo INTO v\_tiempo\_estacionado, v\_cobro\_id FROM cobros WHERE id = v\_cobro\_id;

SET v\_tiempo\_estacionado = TIMESTAMPDIFF(MINUTE, v\_hora\_ingreso, p\_hora\_salida);

SET v\_costo = v\_tiempo\_estacionado \* (v\_cobro\_id = 1 AND v\_tiempo\_estacionado <= v\_tiempo\_max) + (v\_cobro\_id = 2 AND v\_tiempo\_estacionado > v\_tiempo\_max);

UPDATE vehiculos

SET tiempo\_estacionado = v\_tiempo\_estacionado, pago = v\_costo, cobro\_id = v\_cobro\_id

WHERE id = p\_id;

-- Marcar la transacción como completada

UPDATE vehiculos

SET transaccion\_completada = 1

WHERE id = p\_id;

END IF;

END IF;

END;

//

DELIMITER ;

# -- Inserts

INSERT INTO usuarios (nombre, contrasenya)

VALUES ('admin', 'contrasena\_admin');

INSERT INTO estacionamientos (estacionamiento, ubicacion, telefono, tiempo\_tolerancia, administrador\_id)

VALUES ('Estacionamiento ABC', 'Ubicación XYZ', '1234567890', 15, 1);

-- Insertar tarifas en la tabla Cobros

-- Tolerancia de 15 minutos

INSERT INTO cobros (estacionamiento\_id, tiempo\_min, tiempo\_max, costo)

VALUES (1, 0, 15, 0);

-- Primeras 2 horas $20

INSERT INTO cobros (estacionamiento\_id, tiempo\_min, tiempo\_max, costo)

VALUES (1, 16, 120, 20);

-- Hora adicional $20

INSERT INTO cobros (estacionamiento\_id, tiempo\_min, tiempo\_max, costo)

VALUES (1, 121, 1440, 20);

-- Pensión por día (24 hrs) $200

INSERT INTO cobros (estacionamiento\_id, tiempo\_min, tiempo\_max, costo)

VALUES (1, 1441, 10080, 200);

-- Pensión por semana (168 hrs) $1000

INSERT INTO cobros (estacionamiento\_id, tiempo\_min, tiempo\_max, costo)

VALUES (1, 10081, 43200, 1000);

-- Pensión por mes (720 hrs) $4000

INSERT INTO cobros (estacionamiento\_id, tiempo\_min, tiempo\_max, costo)

VALUES (1, 43201, 259200, 4000);