

```

-- Database: licencias_conducir

-- DROP DATABASE IF EXISTS licencias_conducir;

CREATE DATABASE licencias_conducir
    WITH
        OWNER = postgres
        ENCODING = 'UTF8'
        LC_COLLATE = 'Spanish_Spain.1252'
        LC_CTYPE = 'Spanish_Spain.1252'
        LOCALE_PROVIDER = 'libc'
        TABLESPACE = pg_default
        CONNECTION LIMIT = -1
        IS_TEMPLATE = False;

CREATE TABLE ciudadanos (
    id SERIAL PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    apellido VARCHAR(100) NOT NULL,
    fecha_nacimiento DATE NOT NULL,
    direccion TEXT NOT NULL,
    telefono VARCHAR(15) UNIQUE,
    correo VARCHAR(100) UNIQUE
);

-- Insertando datos en ciudadanos
INSERT INTO ciudadanos (nombre, apellido, fecha_nacimiento, direccion,
telefono, correo) VALUES
('Juan', 'Perez', '1985-07-20', 'Calle 123, Ciudad', '5551234567',
'juan.perez@example.com'),
('Maria', 'Lopez', '1990-03-15', 'Avenida 456, Ciudad', '5559876543',
'maria.lopez@example.com'),
('Carlos', 'Gomez', '1978-11-02', 'Boulevard 789, Ciudad', '5555678912',
'carlos.gomez@example.com');

SELECT * FROM ciudadanos;

CREATE TABLE licencias (
    id SERIAL PRIMARY KEY,
    ciudadano_id INT NOT NULL,
    numero_licencia VARCHAR(20) UNIQUE NOT NULL,
    categoria VARCHAR(10) NOT NULL,
    fecha_expedicion DATE NOT NULL,
    fecha_vencimiento DATE NOT NULL,
    FOREIGN KEY (ciudadano_id) REFERENCES ciudadanos(id) ON DELETE
CASCADE
);

-- Insertando datos en licencias
INSERT INTO licencias (ciudadano_id, numero_licencia, categoria,
fecha_expedicion, fecha_vencimiento) VALUES
(1, 'LIC12345', 'A', '2022-06-10', '2027-06-10'),

```

```
(2, 'LIC67890', 'B', '2023-01-20', '2028-01-20'),
(3, 'LIC54321', 'C', '2021-09-05', '2026-09-05');
```

```
SELECT * FROM licencias;
```

```
CREATE TABLE citas (
    id SERIAL PRIMARY KEY,
    ciudadano_id INT NOT NULL,
    fecha TIMESTAMP NOT NULL UNIQUE,
    tipo VARCHAR(50) NOT NULL, -- Ejemplo: "Expedición", "Renovación"
    estado VARCHAR(20) NOT NULL DEFAULT 'Pendiente', -- "Pendiente",
    "Pagado", "Cancelado"
    FOREIGN KEY (ciudadano_id) REFERENCES ciudadanos(id) ON DELETE
    CASCADE
);
```

```
-- Insertando datos en citas
INSERT INTO citas (ciudadano_id, fecha, tipo, estado) VALUES
(1, '2025-04-15 10:00:00', 'Renovación', 'Pendiente'),
(2, '2025-04-16 11:30:00', 'Expedición', 'Pendiente'),
(3, '2025-04-17 09:00:00', 'Renovación', 'Pendiente');
```

```
SELECT * FROM citas;
```

```
CREATE TABLE pagos (
    id SERIAL PRIMARY KEY,
    cita_id INT NOT NULL UNIQUE,
    monto DECIMAL(10,2) NOT NULL,
    metodo_pago VARCHAR(50) NOT NULL, -- "Tarjeta", "Transferencia"
    fecha_pago TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    estado VARCHAR(20) NOT NULL DEFAULT 'Completado', -- "Completado",
    "Fallido"
    FOREIGN KEY (cita_id) REFERENCES citas(id) ON DELETE CASCADE
);
```

```
-- Insertando datos en pagos
INSERT INTO pagos (cita_id, monto, metodo_pago, estado) VALUES
(1, 500.00, 'Tarjeta', 'Completado'),
(2, 450.00, 'Transferencia', 'Completado'),
(3, 400.00, 'Tarjeta', 'Completado');
```

```
SELECT * FROM pagos;
```

```
-- Intentando insertar una cita en una fecha y hora ya ocupada
-- Mismo horario que la cita de Juan Pérez
DO $$
BEGIN
```

```
INSERT INTO citas (ciudadano_id, fecha, tipo, estado) VALUES
(2, '2025-04-15 10:00:00', 'Expedición', 'Pendiente');
EXCEPTION
    WHEN unique_violation THEN
        RAISE NOTICE 'Error: La fecha y hora seleccionadas ya están
ocupadas. Intente con otro horario.';
END $$;
```

```
-- Consulta para mostrar nombre completo, tipo de licencia, fecha de
cita, monto y método de pago
SELECT CONCAT(c.nombre, ' ', c.apellido) AS nombre_completo, l.categoria
AS tipo_licencia, ci.fecha AS fecha_cita, p.monto, p.metodo_pago
FROM ciudadanos c
JOIN licencias l ON c.id = l.ciudadano_id
JOIN citas ci ON c.id = ci.ciudadano_id
JOIN pagos p ON ci.id = p.cita_id;
```