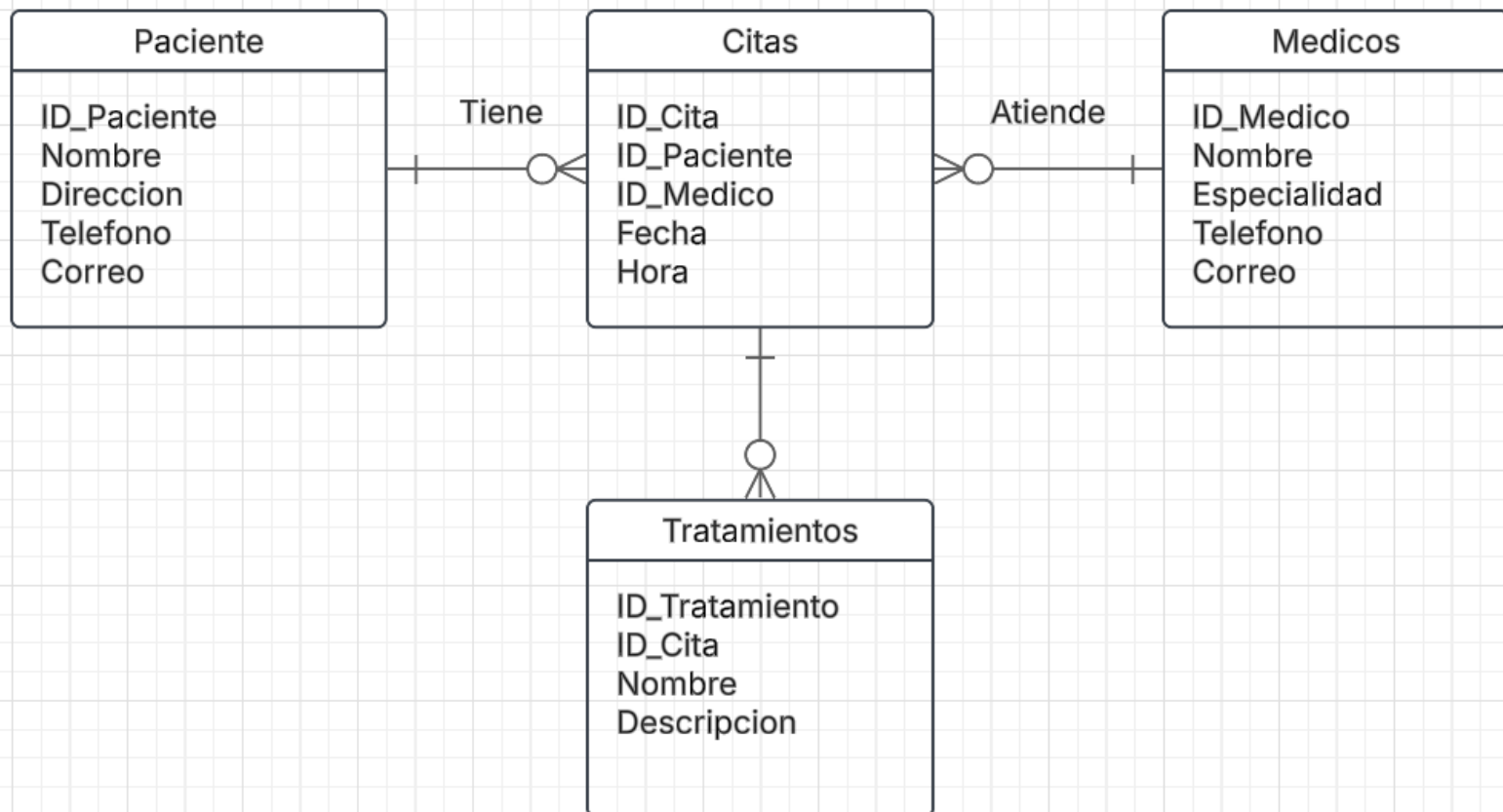


1. SISTEMA GESTION HOSPITALES



CREACIÓN DE TABLAS

```
1  ✓ CREATE TABLE Paciente (  
2    id_paciente SERIAL PRIMARY KEY,  
3    nombre VARCHAR(100) NOT NULL,  
4    direccion VARCHAR(200) NOT NULL,  
5    telefono VARCHAR(20) NOT NULL,  
6    correo VARCHAR(100) UNIQUE NOT NULL  
7  );  
8  
9  ✓ CREATE TABLE Medico (  
10   id_medico SERIAL PRIMARY KEY,  
11   nombre VARCHAR(100) NOT NULL,  
12   especialidad VARCHAR(100) NOT NULL,  
13   telefono VARCHAR(50) NOT NULL,  
14   correo VARCHAR(100) UNIQUE NOT NULL  
15  );  
16  
17  ✓ CREATE TABLE Cita (  
18   id_cita SERIAL PRIMARY KEY,  
19   fecha DATE NOT NULL,  
20   hora TIME NOT NULL,  
21   id_paciente INT NOT NULL,  
22   id_medico INT NOT NULL,  
23   FOREIGN KEY (id_paciente) REFERENCES Paciente (id_paciente),  
24   FOREIGN KEY (id_medico) REFERENCES Medico(id_medico)  
25  );  
26  
27  ✓ CREATE TABLE Tratamiento (  
28   id_tratamiento SERIAL PRIMARY KEY,  
29   descripcion TEXT NOT NULL,  
30   id_cita INT NOT NULL,  
31   FOREIGN KEY (id_cita) REFERENCES Cita(id_cita)  
32  );
```

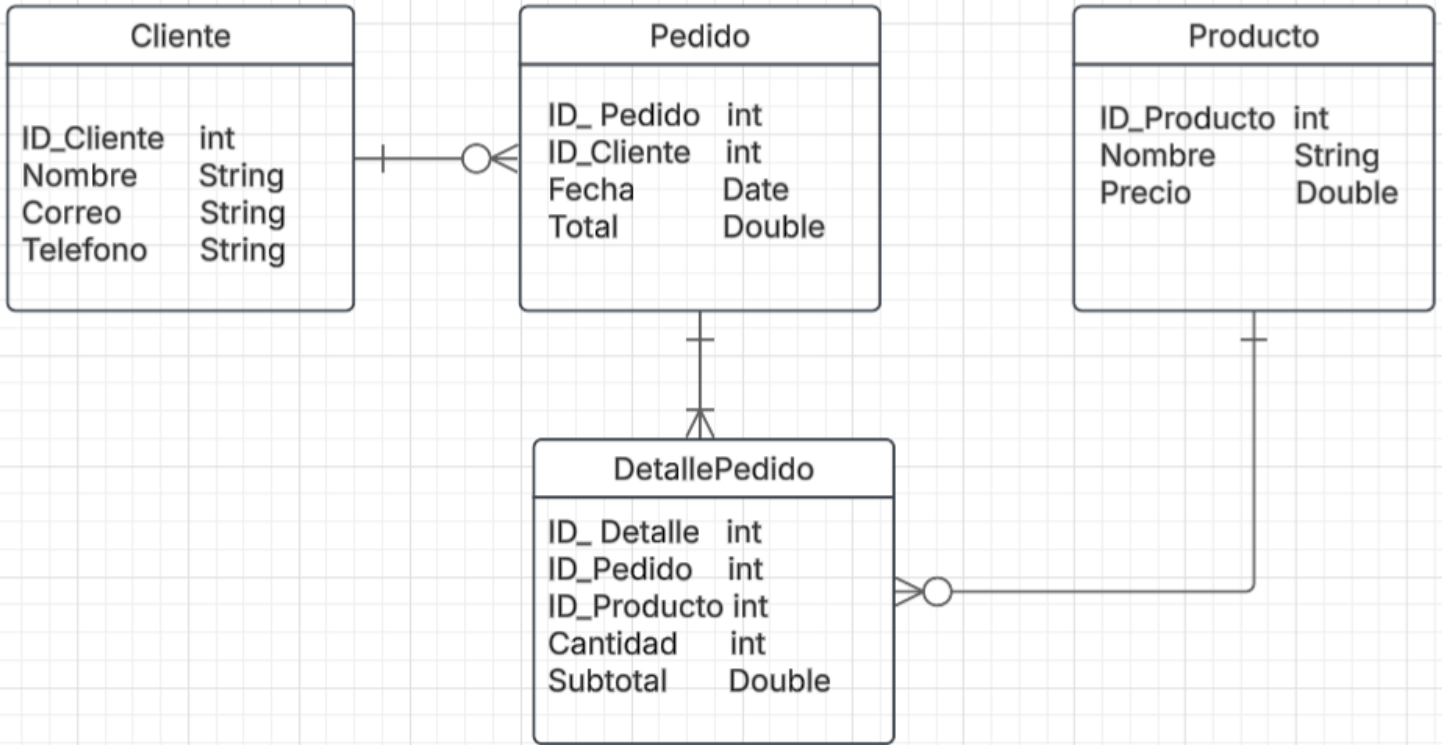
CITAS DE UN PACIENTE ESPECIFICO

```
1 SELECT p.nombre AS paciente,  
2 c.id_cita, c.fecha, c.hora,  
3 t.descripcion AS tratamiento,  
4 m.nombre AS medico  
5 FROM cita c  
6 JOIN tratamiento t ON c.id_cita = t.id_cita  
7 JOIN medico m ON c.id_medico = m.id_medico  
8 JOIN paciente p ON c.id_paciente = p.id_paciente  
9 WHERE c.id_paciente = 1;
```

Data Output Messages Notifications

Showing rows: 1 to						
	paciente character varying (100)	id_cita integer	fecha date	hora time without time zone	tratamiento text	medico character varying (100)
1	Juan Perez	1	2025-03-20	10:00:00	Tratamiento de Hipertension	Dr. Carlos Lopez

2. TIENDA EN LINEA



```
1  ✓ CREATE TABLE Cliente
2  (
3  id_cliente SERIAL PRIMARY KEY,
4  nombre VARCHAR (100) NOT NULL,
5  correo VARCHAR (100) UNIQUE NOT NULL,
6  telefono VARCHAR(20) NOT NULL
7
8  );
9
10 ✓ CREATE TABLE Producto (
11 id_producto SERIAL PRIMARY KEY,
12 nombre VARCHAR(100),
13 precio DOUBLE PRECISION NOT NULL
14 );
15
16 ✓ CREATE TABLE Pedido (
17 id_pedido SERIAL PRIMARY KEY,
18 fecha DATE NOT NULL,
19 total DOUBLE PRECISION NOT NULL,
20 id_cliente INT NOT NULL,
21 FOREIGN KEY (id_cliente) REFERENCES Cliente (id_cliente) ON DELETE CASCADE
22 );
23
24 ✓ CREATE TABLE DetallePedido
25 (
26 id_detalle SERIAL PRIMARY KEY,
27 id_pedido INT NOT NULL,
28 id_producto INT NOT NULL,
29 cantidad INT NOT NULL,
30 subtotal DOUBLE PRECISION NOT NULL,
31 FOREIGN KEY (id_pedido) REFERENCES Pedido(id_pedido) ON DELETE CASCADE,
32 FOREIGN KEY (id_producto) REFERENCES Producto (id_producto) ON DELETE CASCADE
33 );
```

Productos comprados por un cliente especifico

Query Query History

```
1 SELECT c.id_cliente, c.nombre AS cliente, p.id_producto, p.nombre AS producto,
2 p.precio, dp.cantidad, dp.subtotal, pe.fecha
3 FROM DetallePedido dp
4 JOIN Pedido pe ON dp.id_pedido = pe.id_pedido
5 JOIN Producto p ON dp.id_producto = p.id_producto
6 JOIN Cliente c ON pe.id_cliente = c.id_cliente
7 WHERE c.id_cliente = 1;
```

Data Output Messages Notifications

	id_cliente integer	cliente character varying (100)	id_producto integer	producto character varying (100)	precio double precision	cantidad integer	subtotal double precision	fecha date
1	1	Rosario Gonzalez	1	Laptop	1200.5	1	1200.5	2025-03-18
2	1	Rosario Gonzalez	2	Mouse	25.75	1	25.75	2025-03-18

3. Biblioteca Digital



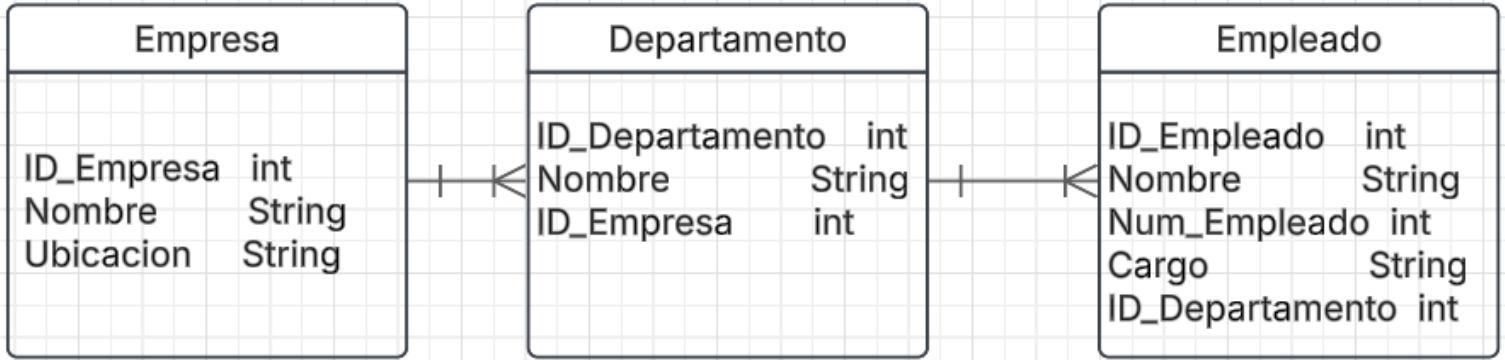
```
1 SELECT u.nombre, l.titulo, p.fecha_inicio, p.fecha_fin, p.estado
2 FROM Prestamo p
3 JOIN Usuario u ON p.id_usuario = u.id_usuario
4 JOIN Libro l ON p.id_libro = l.id_libro
5 WHERE p.estado = 'activo' AND u.id_usuario = 1;
```



SQL

	nombre character varying (100)	titulo character varying (100)	fecha_inicio date	fecha_fin date	estado character varying (10)
1	Ricardo Salazar	La vuelta al mundo en 80 dias	2025-03-01	2025-03-15	activo

4. Sistema de Recursos Humanos





```
1  ✓ CREATE TABLE Empresa
2  (
3  id_empresa SERIAL PRIMARY KEY,
4  nombre VARCHAR(100) NOT NULL,
5  ubicacion VARCHAR(100) NOT NULL
6  );
7
8  ✓ CREATE TABLE Departamento
9  (
10 id_departamento SERIAL PRIMARY KEY,
11 nombre VARCHAR(100) NOT NULL,
12 id_empresa INT NOT NULL,
13 FOREIGN KEY (id_empresa) REFERENCES Empresa(id_empresa) ON DELETE CASCADE
14 );
15
16 ✓ CREATE TABLE Empleado
17 (
18 id_empleado SERIAL PRIMARY KEY,
19 id_departamento INT NOT NULL,
20 nombre VARCHAR(100) NOT NULL,
21 num_empleado INT UNIQUE NOT NULL,
22 cargo VARCHAR(50) NOT NULL,
23 FOREIGN KEY (id_departamento) REFERENCES Departamento(id_departamento) ON DELETE CASCADE
24 );
25
```

Empleados con su departamento

Query Query History

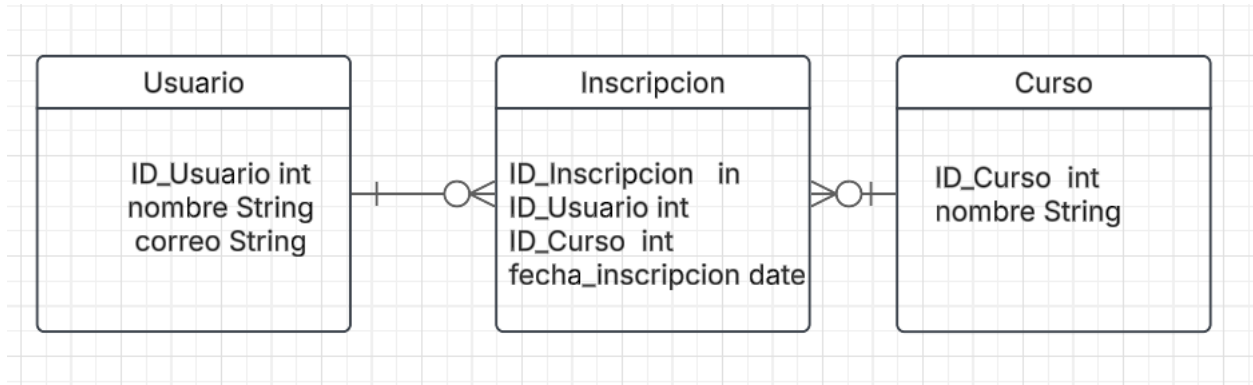
```
1 SELECT e.id_empleado, e.nombre AS empleado, e.num_empleado, e.cargo, d.nombre AS departamento
2 FROM Empleado e
3 JOIN Departamento d ON e.id_departamento = d.id_departamento
4 ORDER BY d.nombre;
```

SQL

Show

	id_empleado integer	empleado character varying (100)	num_empleado integer	cargo character varying (50)	departamento character varying (100)
1	5	Luis Ramirez	1005	Ejecutivo de Atencion	Atencion al Cliente
2	2	Ana Lopez	1002	Desarrollador	Desarrollo
3	4	Maria Garcia	1004	Contadora	Finanzas
4	8	Diana Gomez	1008	Investigadora	Investigacion
5	6	Sofía Fernandez	1006	Coordinadora de Logistica	Logistica
6	3	Carlos Mendoza	1003	Analista de Marketing	Marketing
7	7	Pedro Torres	1007	Supervisor de Produccion	Produccion
8	1	Juan Perez	1001	Gerente de RRHH	Recursos Humanos
9	10	Elena Castro	1010	Administrador de TI	TI
10	9	Ricardo Sanchez	1009	Ejecutivo de Ventas	Ventas

5. Cursos en Linea



Query Query History

```
1  ✓ CREATE TABLE Usuario (  
2    id_usuario SERIAL PRIMARY KEY,  
3    nombre VARCHAR(100) NOT NULL,  
4    correo VARCHAR(100) UNIQUE NOT NULL  
5  );  
6  
7  ✓ CREATE TABLE Curso (  
8    id_curso SERIAL PRIMARY KEY,  
9    nombre VARCHAR(100) NOT NULL  
10 );  
11  
12 ✓ CREATE TABLE Incripcion (  
13   id_incripcion SERIAL PRIMARY KEY,  
14   id_usuario INT NOT NULL,  
15   id_curso INT NOT NULL,  
16   fecha_incripcion DATE,  
17   FOREIGN KEY (id_usuario) REFERENCES Usuario(id_usuario) ON DELETE CASCADE,  
18   FOREIGN KEY (id_curso) REFERENCES Curso(id_curso) ON DELETE CASCADE  
19 );
```

CURSOS INSCRITOS POR UN USUARIO

```
SELECT U.nombre AS usuario, C.nombre AS curso, I.fecha_inscripcion
FROM Inscripcion I
JOIN Usuario U ON I.id_usuario = U.id_usuario
JOIN Curso C ON I.id_curso = C.id_curso
WHERE U.nombre = 'Juan Perez';
```

	usuario character varying (100)	curso character varying (100)	fecha_inscripcion date
1	Juan Perez	Matematicas Avanzadas	2025-03-01
2	Juan Perez	Introduccion a la Programacion	2025-03-02