



**TECNOLÓGICO
NACIONAL DE MÉXICO**



Tecnológico Nacional de México, Campus
Mexicali
Ingeniería en Sistemas Computacionales

Alumno:
Emanuel Padilla Valencia
23490363

Materia: Fundamentos de Bases de Datos
Docente: José Ramón Bogarin Valenzuela
Tarea 3 Unidad 2 Consultas SQL
Mexicali Baja California, al 18 de marzo
del 2025.

1. Sistema de Gestión de Hospitales.

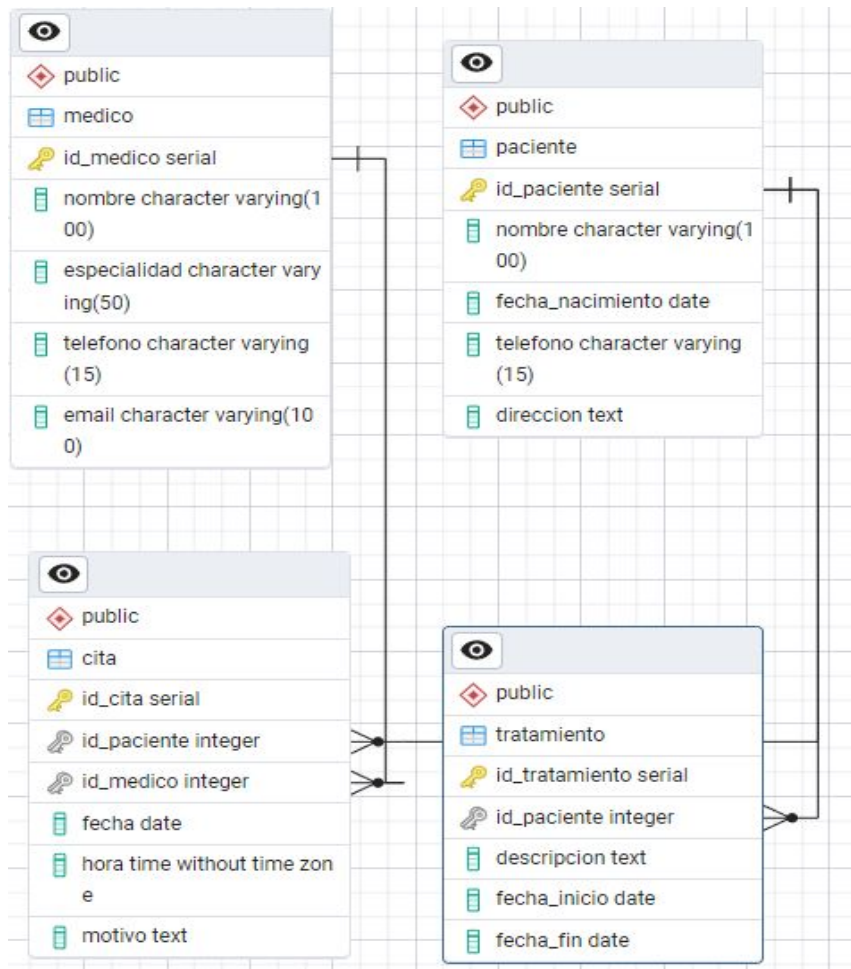
Identificación de entidades claves:

- **Paciente:** Identifica a los pacientes que reciben atención.
- **Médico:** Representa a los médicos del hospital.
- **Cita:** Registra las consultas médicas de los pacientes.
- **Tratamiento:** Representa los tratamientos que recibe un paciente.

Especificaciones:

- Relación **Paciente - Cita** (Uno a Muchos): Un paciente puede tener varias citas.
- Relación **Médico - Cita** (Uno a Muchos): Un médico puede atender muchas citas.
- Relación **Cita - Tratamiento** (Uno a Muchos): Cada cita puede tener tratamientos asociados.
- Se debe normalizar la base de datos para evitar redundancias.
- Implementación de claves primarias y foráneas para mantener integridad referencial.
- Consultas SQL necesarias para obtener citas de un paciente específico.

Diagrama E-R con sus relaciones y atributos



Esquema relacional con tablas y columnas

Paciente: (id_paciente **(PK)**, nombre, fecha_nacimiento, telefono, direccion).

Médico: (id_medico **(PK)**, nombre, especialidad, telefono,email).

Cita: (id_cita **(PK)**, id_paciente **(FK)**, id_medico **(FK)**, fecha, hora, motivo).

Tratamiento: (id_tratamiento **(PK)**, id_paciente **(FK)**, descripcion, fecha_inicio, fecha_fin).


Implementación de la base de datos en SQL

```
CREATE TABLE Paciente (  
  id_paciente SERIAL PRIMARY KEY,  
  nombre VARCHAR(100) NOT NULL,  
  fecha_nacimiento DATE NOT NULL,  
  telefono VARCHAR(15),  
  direccion TEXT  
);  
  
CREATE TABLE Medico (  
  id_medico SERIAL PRIMARY KEY,  
  nombre VARCHAR(100) NOT NULL,  
  especialidad VARCHAR(50) NOT NULL,  
  telefono VARCHAR(15),  
  email VARCHAR(100)  
);  
  
CREATE TABLE Cita (  
  id_cita SERIAL PRIMARY KEY,  
  id_paciente INT NOT NULL,  
  id_medico INT NOT NULL,  
  fecha DATE NOT NULL,  
  hora TIME NOT NULL,  
  motivo TEXT,  
  FOREIGN KEY (id_paciente) REFERENCES Paciente(id_paciente) ON DELETE CASCADE,  
  FOREIGN KEY (id_medico) REFERENCES Medico(id_medico) ON DELETE CASCADE  
);  
  
CREATE TABLE Tratamiento (  
  id_tratamiento SERIAL PRIMARY KEY,  
  id_paciente INT NOT NULL,  
  descripcion TEXT NOT NULL,  
  fecha_inicio DATE NOT NULL,  
  fecha_fin DATE,  
  FOREIGN KEY (id_paciente) REFERENCES Paciente(id_paciente) ON DELETE CASCADE  
);
```

Consulta específica de un paciente

```
1 SELECT P.nombre AS paciente, C.id_cita, C.fecha, C.hora, C.motivo, M.nombre
2 FROM Cita C
3 JOIN Medico M ON C.id_medico = M.id_medico
4 JOIN Paciente P ON C.id_paciente = P.id_paciente
5 WHERE P.nombre = 'Juan Pérez';
6
```

Data Output Mensajes Notificaciones

Showing rows: 1 to 1  Page No: 1 of 1 						
	paciente character varying (100) 	id_cita integer 	fecha date 	hora time without time zone 	motivo text 	medico character varying (100) 
1	Juan Pérez	1	2025-03-10	08:30:00	Revisión cardiológica	Dr. José Martínez

2. Tienda en Línea

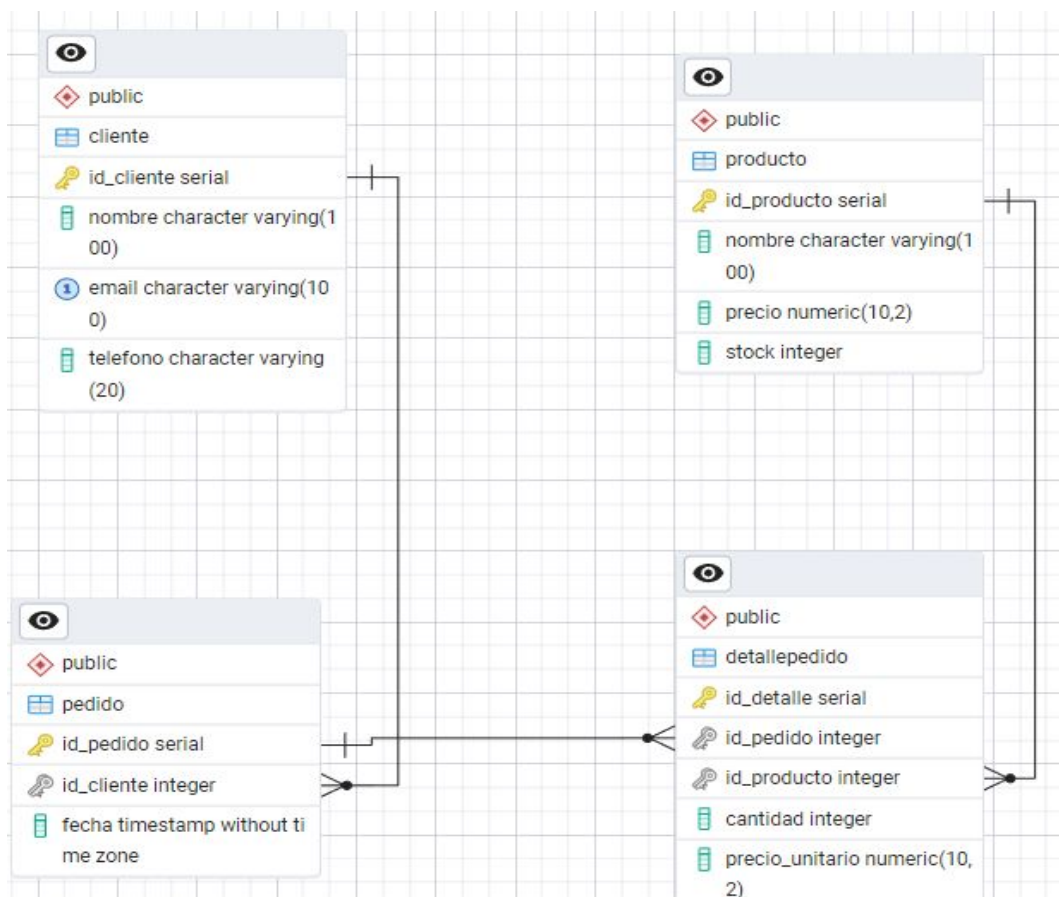
Identificación de entidades clave:

- **Cliente:** Representa a los clientes que realizan compras en la tienda en línea.
- **Producto:** Contiene la información de los artículos disponibles para la venta.
- **Pedido:** Registra las compras realizadas por los clientes.
- **DetallePedido:** Contiene los productos específicos dentro de cada pedido.

Especificaciones:

- **Relación Cliente - Pedido (Uno a Muchos):** Un cliente puede realizar varios pedidos.
- **Relación Pedido - DetallePedido (Uno a Muchos):** Un pedido puede incluir múltiples productos.
- **Relación Producto - DetallePedido (Uno a Muchos):** Un producto puede aparecer en varios pedidos

Diagrama E-R con sus relaciones y atributos



Esquema relacional con tablas y columnas

1. **Cliente** (id_cliente **PK**, nombre, email **UNIQUE**, telefono)
2. **Producto** (id_producto **PK**, nombre, precio, stock)
3. **Pedido** (id_pedido **PK**, id_cliente **FK**, fecha)
4. **DetallePedido** (id_detalle **PK**, id_pedido **FK**, id_producto **FK**, cantidad, precio_unitario)

Implementación de la base de datos en SQL

```
CREATE TABLE Cliente (  
    id_cliente SERIAL PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    email VARCHAR(100) UNIQUE NOT NULL,  
    telefono VARCHAR(20)  
);  
  
CREATE TABLE Producto (  
    id_producto SERIAL PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    precio DECIMAL(10,2) NOT NULL,  
    stock INT NOT NULL CHECK (stock >= 0)  
);  
  
CREATE TABLE Pedido (  
    id_pedido SERIAL PRIMARY KEY,  
    id_cliente INT REFERENCES Cliente(id_cliente),  
    fecha TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);  
  
CREATE TABLE DetallePedido (  
    id_detalle SERIAL PRIMARY KEY,  
    id_pedido INT REFERENCES Pedido(id_pedido),  
    id_producto INT REFERENCES Producto(id_producto),  
    cantidad INT NOT NULL CHECK (cantidad > 0),  
    precio_unitario DECIMAL(10,2) NOT NULL  
);
```

Consulta específica de un cliente

```
1 SELECT D.nombre AS departamento, E.nombre AS empleado, E.puesto, E.salario
2 FROM Empleado E
3 JOIN Departamento D ON E.id_departamento = D.id_departamento
4 WHERE D.nombre = 'Recursos Humanos';
5
```

Data Output Mensajes Notificaciones

≡+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

Showing rows: 1 to 1

✎

Page No: 1

	departamento character varying (100) 🔒	empleado character varying (100) 🔒	puesto character varying (100) 🔒	salario numeric (10,2) 🔒
1	Recursos Humanos	Juan Pérez	Gerente de Recursos Humanos	45000.00

3. Biblioteca Digital

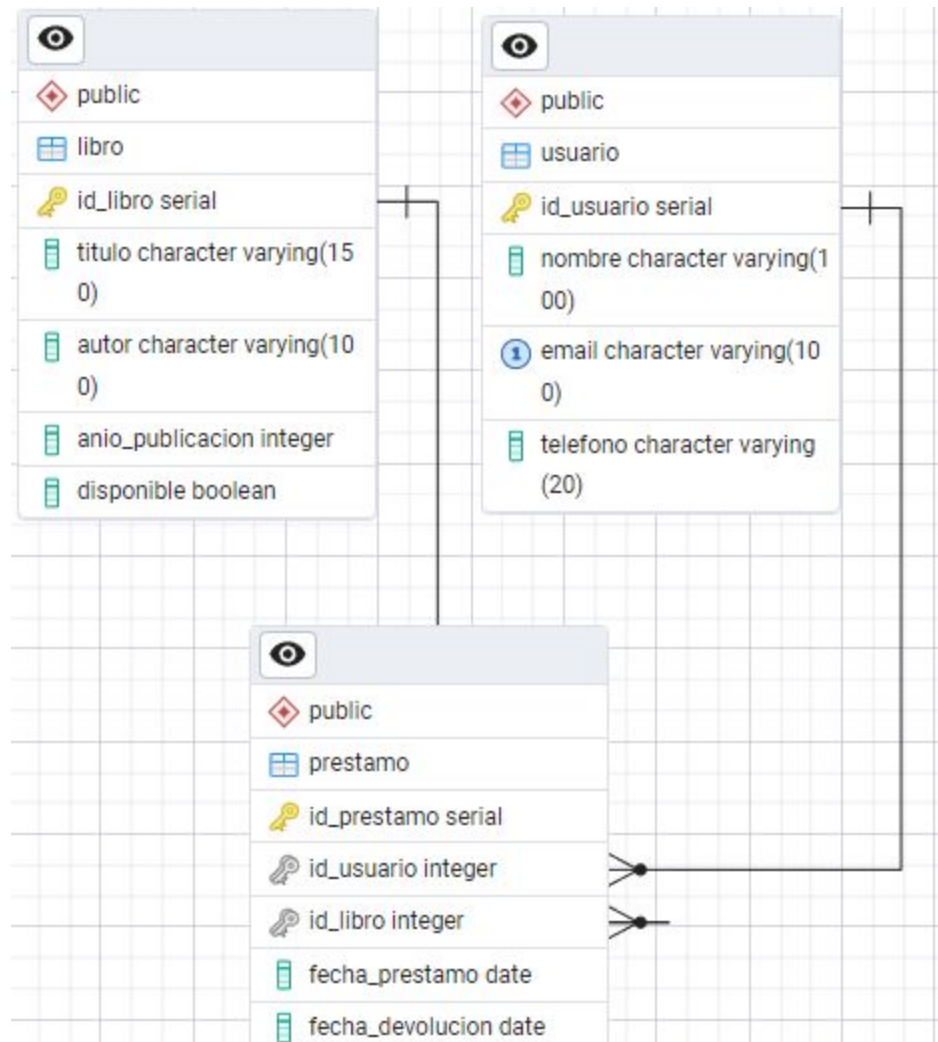
Identificación de entidades clave:

- **Usuario:** Representa a los usuarios que pueden solicitar préstamos de libros.
- **Libro:** Contiene la información de los libros disponibles en la biblioteca digital.
- **Préstamo:** Registra los préstamos de libros realizados por los usuarios.

Especificaciones:

- **Relación Usuario - Préstamo (Uno a Muchos):** Un usuario puede tener múltiples préstamos activos.
- **Relación Libro - Préstamo (Uno a Muchos):** Un libro puede estar prestado a diferentes usuarios en distintas fechas.

Diagrama E-R con sus relaciones y atributos



Esquema Relacional

1. **Usuario** (id_usuario **PK**, nombre, email **UNIQUE**, telefono)
2. **Libro** (id_libro **PK**, titulo, autor, anio_publicacion, disponible)
3. **Préstamo** (id_prestamo **PK**, id_usuario **FK**, id_libro **FK**, fecha_prestamo, fecha_devolucion)

Implementación de la base de datos en SQL

```
CREATE TABLE Usuario (  
    id_usuario SERIAL PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    email VARCHAR(100) UNIQUE NOT NULL,  
    telefono VARCHAR(20)  
);  
  
CREATE TABLE Libro (  
    id_libro SERIAL PRIMARY KEY,  
    titulo VARCHAR(150) NOT NULL,  
    autor VARCHAR(100) NOT NULL,  
    anio_publicacion INT NOT NULL,  
    disponible BOOLEAN DEFAULT TRUE  
);  
  
CREATE TABLE Prestamo (  
    id_prestamo SERIAL PRIMARY KEY,  
    id_usuario INT REFERENCES Usuario(id_usuario),  
    id_libro INT REFERENCES Libro(id_libro),  
    fecha_prestamo DATE NOT NULL DEFAULT CURRENT_DATE,  
    fecha_devolucion DATE  
);
```

Consulta específica de un cliente

```
1  SELECT U.nombre AS usuario, L.titulo AS libro, P.fecha_prestamo
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 U.nombre = 'Juan Pérez' AND P.fecha_devolucion IS NULL;
6
```

Data Output Mensajes Notificaciones

Showing rows: 1 to 1 			
	usuario character varying (100) 	libro character varying (150) 	fecha_prestamo date 
1	Juan Pérez	Cien años de soledad	2025-03-10

4. Sistema de Recursos Humanos

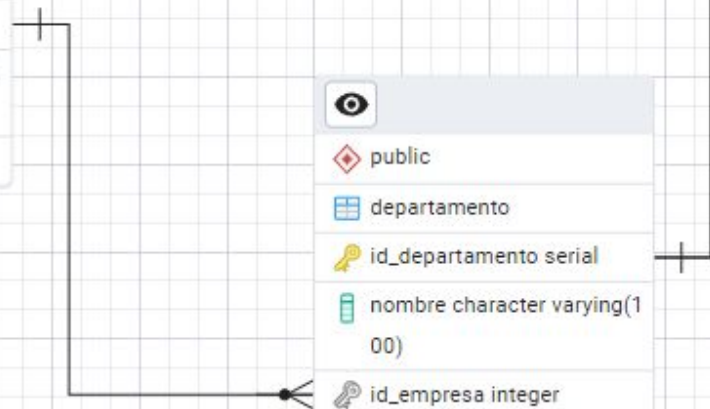
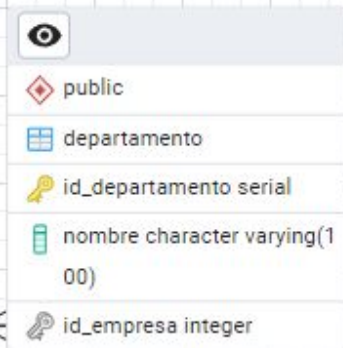
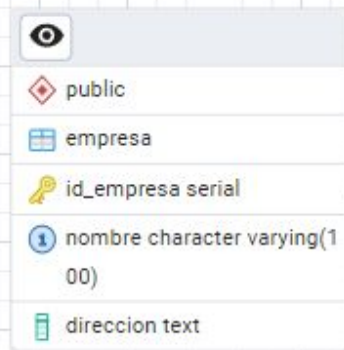
Identificación de entidades clave:

- **Empresa:** Representa la organización que agrupa los departamentos y empleados.
- **Departamento:** Define las diferentes áreas dentro de la empresa.
- **Empleado:** Contiene la información de los trabajadores de la empresa.

Especificaciones:

- **Relación Empresa - Departamento (Uno a Muchos):** Una empresa puede tener múltiples departamentos, pero cada departamento pertenece a una sola empresa.
- **Relación Departamento - Empleado (Uno a Muchos):** Un departamento puede tener varios empleados, pero cada empleado pertenece a un solo departamento.

Diagrama E-R con sus relaciones y atributos



Esquema Relacional

1. **Empresa** (id_empresa **PK**, nombre **UNIQUE**, direccion)
2. **Departamento** (id_departamento **PK**, nombre, id_empresa **FK**)
3. **Empleado** (id_empleado **PK**, nombre, puesto, salario, id_departamento **FK**)

Implementación de la base de datos en SQL

```
CREATE TABLE Empresa (  
    id_empresa SERIAL PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL UNIQUE,  
    direccion TEXT NOT NULL  
);  
  
CREATE TABLE Departamento (  
    id_departamento SERIAL PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    id_empresa INT REFERENCES Empresa(id_empresa)  
);  
  
CREATE TABLE Empleado (  
    id_empleado SERIAL PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    puesto VARCHAR(100) NOT NULL,  
    salario DECIMAL(10,2) NOT NULL CHECK (salario > 0),  
    id_departamento INT REFERENCES Departamento(id_departamento)  
);
```

Consulta a empleado por departamento en específico

```
1  SELECT D.nombre AS departamento, E.nombre AS empleado, E.puesto, E.salario
2  FROM Empleado E
3  JOIN Departamento D ON E.id_departamento = D.id_departamento
4  WHERE D.nombre = 'Desarrollo de Software';
5
```

Data Output Mensajes Notificaciones

	departamento character varying (100) 🔒	empleado character varying (100) 🔒	puesto character varying (100) 🔒	salario numeric (10,2) 🔒
1	Desarrollo de Software	María Gómez	Desarrollador Senior	60000.00

5. Plataforma de Cursos en Línea

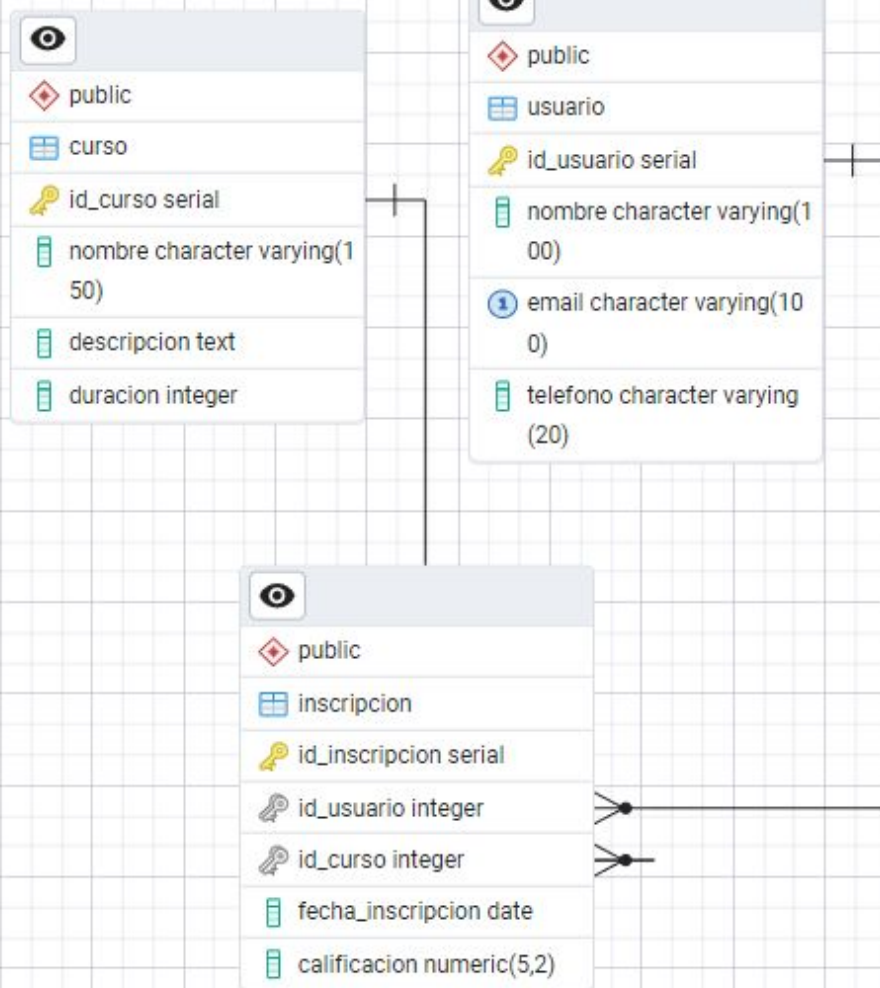
Identificación de entidades clave:

- **Usuario:** Representa a los estudiantes que se inscriben en cursos en línea.
- **Curso:** Contiene la información de los cursos disponibles en la plataforma.
- **Inscripción:** Registra la relación entre los usuarios y los cursos en los que están inscritos.

Especificaciones:

- **Relación Usuario - Inscripción (Uno a Muchos):** Un usuario puede inscribirse en múltiples cursos, pero cada inscripción pertenece a un solo usuario.
- **Relación Curso - Inscripción (Uno a Muchos):** Un curso puede tener varios estudiantes inscritos, pero cada inscripción pertenece a un solo curso.

Diagrama E-R con sus relaciones y atributos



Esquema Relacional

1. **Usuario** (id_usuario **PK**, nombre, email **UNIQUE**, telefono)
2. **Curso** (id_curso **PK**, nombre, descripcion, duracion)
3. **Inscripción** (id_inscripcion **PK**, id_usuario **FK**, id_curso **FK**, fecha_inscripcion, calificacion)

Implementación de la base de datos en SQL

```
CREATE TABLE Usuario (  
    id_usuario SERIAL PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    email VARCHAR(100) UNIQUE NOT NULL,  
    telefono VARCHAR(20)  
);  
  
CREATE TABLE Curso (  
    id_curso SERIAL PRIMARY KEY,  
    nombre VARCHAR(150) NOT NULL,  
    descripcion TEXT,  
    duracion INT NOT NULL CHECK (duracion > 0) -- Duración en horas  
);  
  
CREATE TABLE Inscripcion (  
    id_inscripcion SERIAL PRIMARY KEY,  
    id_usuario INT REFERENCES Usuario(id_usuario),  
    id_curso INT REFERENCES Curso(id_curso),  
    fecha_inscripcion DATE NOT NULL DEFAULT CURRENT_DATE,  
    calificacion DECIMAL(5,2) CHECK (calificacion BETWEEN 0 AND 100)  
);
```

Consulta SQL
para obtener
los cursos
inscritos por
cada usuario

```
1 SELECT U.nombre AS usuario, C.nombre AS curso, I.fecha_inscripcion, I.calificacion
2 FROM Inscripcion I
3 JOIN Usuario U ON I.id_usuario = U.id_usuario
4 JOIN Curso C ON I.id_curso = C.id_curso
5 ORDER BY U.nombre, I.fecha_inscripcion;
6
```

Data Output Mensajes Notificaciones

Showing rows: 1 to 10 Page No: 1 of 1

	usuario character varying (100)	curso character varying (150)	fecha_inscripcion date	calificacion numeric (5,2)
1	Ana Torres	Desarrollo Web con React	2025-03-13	75.20
2	Carlos Sánchez	Python para Ciencia de Datos	2025-03-12	[null]
3	Elena Medina	Diseño UX/UI	2025-03-19	78.90
4	Fernando López	Gestión de Proyectos Ágiles	2025-03-18	80.70
5	Gabriela Castillo	Marketing Digital	2025-03-17	[null]
6	Juan Pérez	SQL Básico	2025-03-10	85.50
7	Lucía Fernández	Administración de Servidores Linux	2025-03-15	88.30
8	María Gómez	Java Avanzado	2025-03-11	90.00
9	Pedro Ramírez	Ciberseguridad	2025-03-14	[null]
10	Roberto Díaz	Machine Learning	2025-03-16	91.00