



**TECNOLÓGICO
NACIONAL DE MÉXICO**



Tecnológico Nacional de México, Campus Mexicali
Ingeniería en Sistemas Computacionales

Alumno:
Clara Andrea Martinez Valdez
23490379

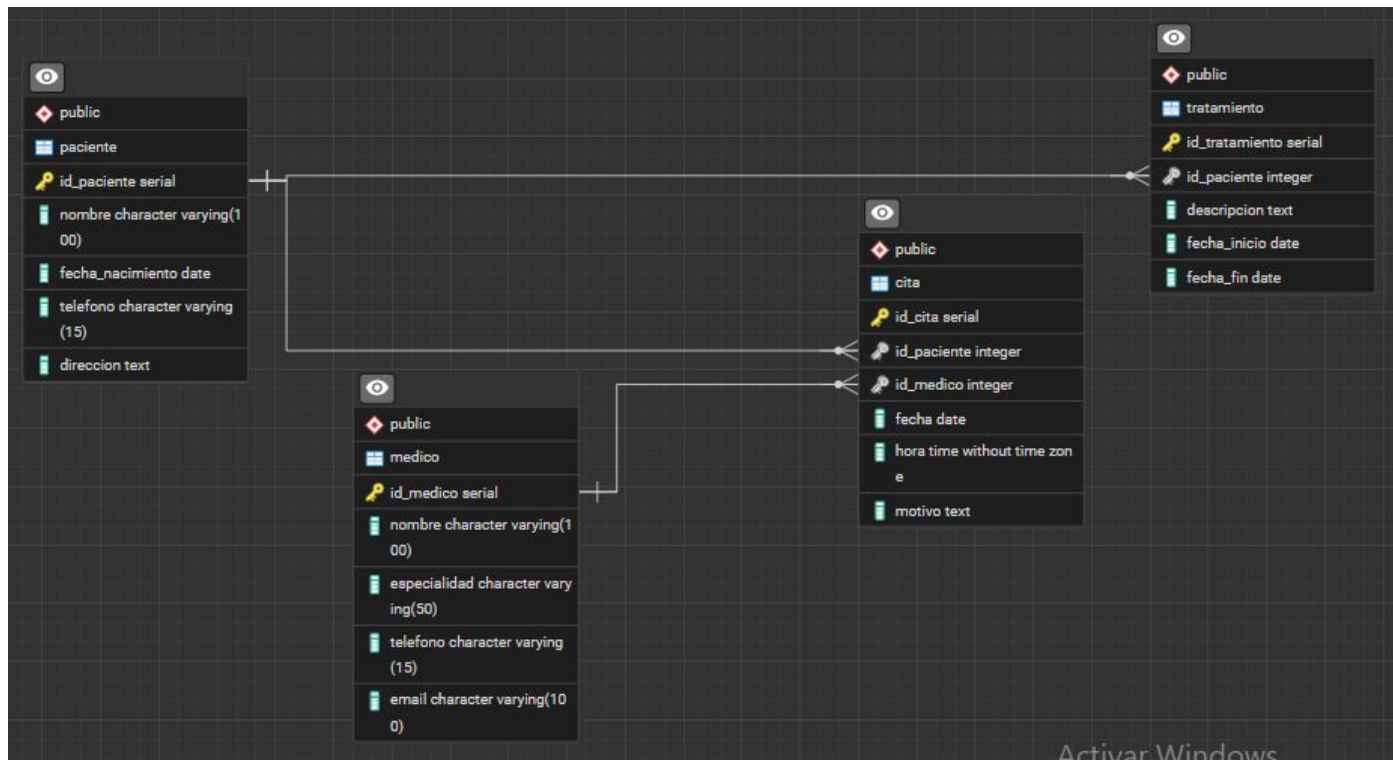
Materia: Fundamentos de Bases de Datos
Docente: José Ramón Bogarin Valenzuela
Tarea2 Unidad 2 Consultas SQL
Mexicali Baja California, al 18 de marzo del 2025.

1. Sistema de Gestión de Hospitales.

Identificación de entidades claves:

- **Paciente:** Representa a los pacientes que reciben atención en el hospital.
- **Médico:** Representa a los médicos que atienden a los pacientes.
- **Cita:** Registro de las consultas médicas programadas entre pacientes y médicos.
- **Tratamiento:** Representa los tratamientos prescritos a los pacientes.

Diagrama E-R con sus relaciones y atributos principales



Esquema Relacional.

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 SQL en la base de datos

```
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  
);
```







Consultar las citas de un paciente en específico

Consulta Historial de Consultas

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

Data Output Mensajes Notificaciones



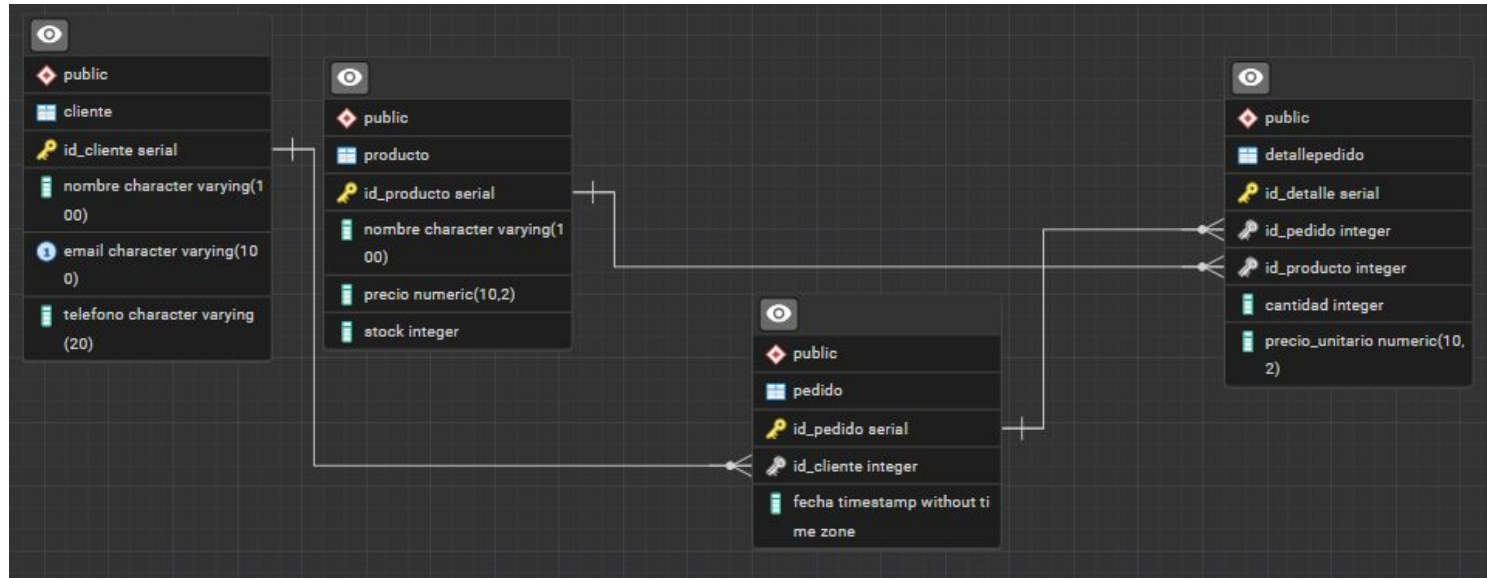
	paciente character varying (100) 	id_cita integer 	fecha date 	hora time without time zone 	medico character varying (100) 	motivo text 
1	Luis Gómez	3	2025-04-05	09:00:00	Dr. Miguel Sanchez	Consulta dermatológica

2. Tienda en Línea

Definición de entidades.

- **Cliente:** Representa a los usuarios que realizan compras.
- **Producto:** Representa los artículos disponibles para la venta.
- **Pedido:** Contiene la información de cada compra realizada.
- **DetallePedido:** Relaciona los productos con los pedidos, especificando la cantidad y precio.

Diagrama E-R con sus relaciones y atributos principales



Esquema relacional

Tabla Cliente

```
CREATE TABLE Cliente (  
    id_cliente SERIAL PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    email VARCHAR(100) UNIQUE NOT NULL,  
    telefono VARCHAR(20)  
);
```

Tabla Producto

```
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 NOT NULL  
);
```

Tabla Pedido

```
CREATE TABLE Pedido (  
    id_pedido SERIAL PRIMARY KEY,  
    id_cliente INT REFERENCES Cliente(id_cliente),  
    fecha TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

Tabla DetallePedido

```
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  
);
```

Implementación de 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  
);
```

Activar Windows

Ve a Configuración para activar

Consultar los productos comprados por un cliente específico usando SQL

```
1 SELECT C.nombre AS Cliente, P.nombre AS Producto, DP.cantidad, DP.precio_unitario
2 FROM Cliente C
3 JOIN Pedido Pe ON C.id_cliente = Pe.id_cliente
4 JOIN DetallePedido DP ON Pe.id_pedido = DP.id_pedido
5 JOIN Producto P ON DP.id_producto = P.id_producto
6 WHERE C.id_cliente = 5;
```

Data Output Mensajes Notificaciones



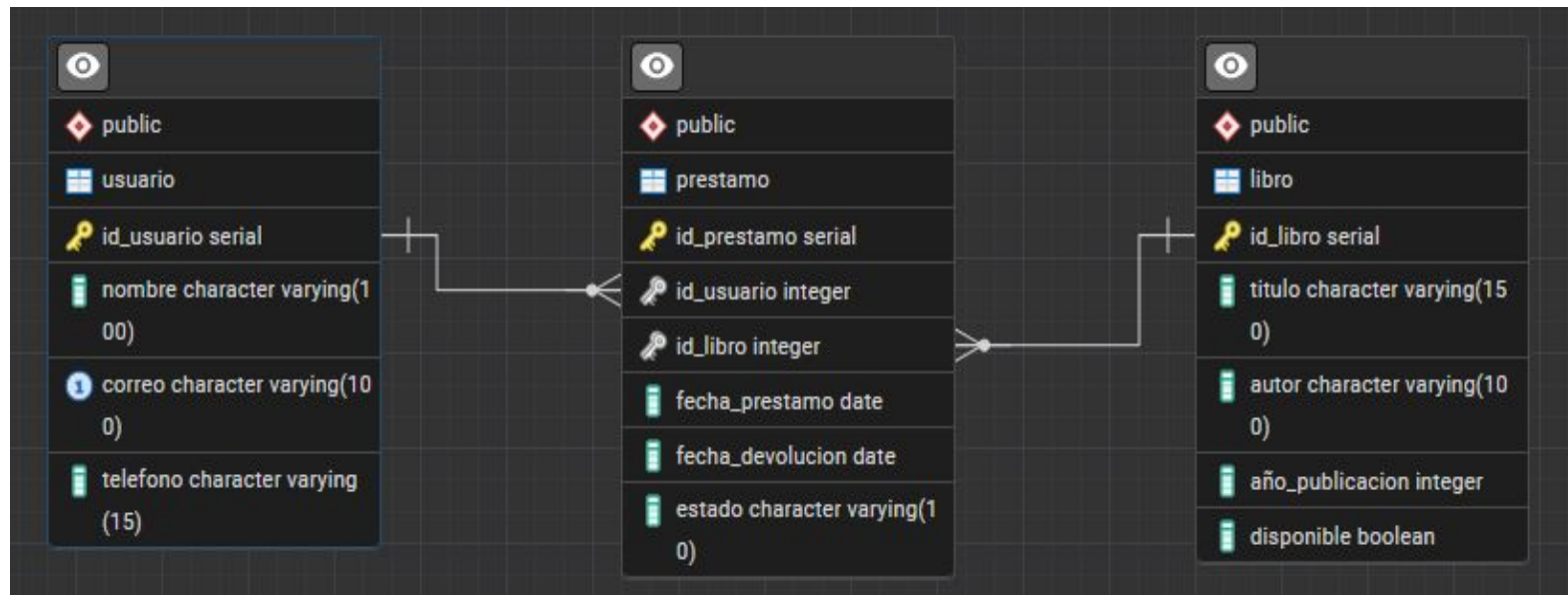
	cliente character varying (100) 🔒	producto character varying (100) 🔒	cantidad integer 🔒	precio_unitario numeric (10,2) 🔒
1	Luis Torres	Escritorio de oficina	1	5000.00

3. Biblioteca Digital

Identificación de entidades clave.

- **Usuario:** Representa a los lectores registrados en la biblioteca.
- **Libro:** Representa los libros disponibles en la biblioteca.
- **Préstamo:** Relación entre un usuario y un libro, indicando que ha sido prestado.

Diseño de modelo E-R



Transformación del modelo en un conjunto de tablas relacionales




Tabla Usuario

	id_usuario [PK] integer	nombre character varying (100)	correo character varying (100)	telefono character varying (15)
1	1	Ana Pérez	ana@example.com	555-1234
2	2	Carlos Gómez	carlos@example.com	555-5678
3	3	María López	maria@example.com	555-8765
4	4	Juan Torres	juan@example.com	555-4321
5	5	Elena Ramírez	elena@example.com	555-6789
6	6	Luis Herrera	luis@example.com	555-9876
7	7	Gabriela Rojas	gabriela@example.com	555-5432
8	8	Ricardo Díaz	ricardo@example.com	555-2345
9	9	Andrea Vargas	andrea@example.com	555-7654
10	10	Fernando Morales	fernando@example.com	555-3210

Tabla Libro

	id_libro [PK] integer 	titulo character varying (150) 	autor character varying (100) 	año_publicacion integer 	disponible boolean 
1	1	El Principito	Antoine de Saint-Exupéry	1943	true
2	2	1984	George Orwell	1949	true
3	3	Cien Años de Soledad	Gabriel García Márquez	1967	true
4	4	Don Quijote de la Mancha	Miguel de Cervantes	1605	true
5	5	Crimen y Castigo	Fiódor Dostoievski	1866	true
6	6	Los Juegos del Hambre	Suzanne Collins	2008	true
7	7	Orgullo y Prejuicio	Jane Austen	1813	true
8	8	Harry Potter y la Piedra Filosofal	J.K. Rowling	1997	true
9	9	El Código Da Vinci	Dan Brown	2003	true
10	10	La Sombra del Viento	Carlos Ruiz Zafón	2001	true

Tabla Prestamos

	Id_prestamo [PK] integer 	Id_usuario integer 	Id_libro integer 	fecha_prestamo date 	fecha_devolucion date 	estado character varying (10) 
1	1	1	1	2025-03-10	[null]	Activo
2	2	2	2	2025-03-12	2025-03-20	Finalizado
3	3	3	3	2025-03-15	[null]	Activo
4	4	4	4	2025-03-16	2025-03-22	Finalizado
5	5	5	5	2025-03-18	[null]	Activo
6	6	6	6	2025-03-19	2025-03-25	Finalizado
7	7	7	7	2025-03-20	[null]	Activo
8	8	8	8	2025-03-21	2025-03-27	Finalizado
9	9	9	9	2025-03-22	[null]	Activo
10	10	10	10	2025-03-23	2025-03-29	Finalizado

Implementación de la base de datos

```
1 CREATE TABLE Usuario (  
2     ID_Usuario SERIAL PRIMARY KEY,  
3     Nombre VARCHAR(100) NOT NULL,  
4     Correo VARCHAR(100) UNIQUE NOT NULL,  
5     Telefono VARCHAR(15)  
6 );  
7  
8 CREATE TABLE Libro (  
9     ID_Libro SERIAL PRIMARY KEY,  
10    Titulo VARCHAR(150) NOT NULL,  
11    Autor VARCHAR(100) NOT NULL,  
12    Año_Publicacion INT NOT NULL,  
13    Disponible BOOLEAN DEFAULT TRUE  
14 );  
15  
16 CREATE TABLE Prestamo (  
17     ID_Prestamo SERIAL PRIMARY KEY,  
18     ID_Usuario INT,  
19     ID_Libro INT,  
20     Fecha_Prestamo DATE NOT NULL,  
21     Fecha_Devolucion DATE,  
22     Estado VARCHAR(10) CHECK (Estado IN ('Activo', 'Finalizado')) DEFAULT 'Activo',  
23     FOREIGN KEY (ID_Usuario) REFERENCES Usuario(ID_Usuario),  
24     FOREIGN KEY (ID_Libro) REFERENCES Libro(ID_Libro)  
25 );  
26
```

Consultas SQL para obtener préstamos activos de un usuario

```
1 SELECT P.ID_Prestamo, U.Nombre, L.Titulo, 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 P.ID_Usuario = 3 AND P.Estado = 'Activo';
6
```

Data Output Mensajes Notificaciones

≡+ 📄 ▼ 📋 ▼ 🗑️ 🗄️ ⬇️ 📈 SQL

	id_prestamo integer	nombre character varying (100)	titulo character varying (150)	fecha_prestamo date
1	3	María López	Cien Años de Soledad	2025-03-15

4. Sistema De Recursos Humanos

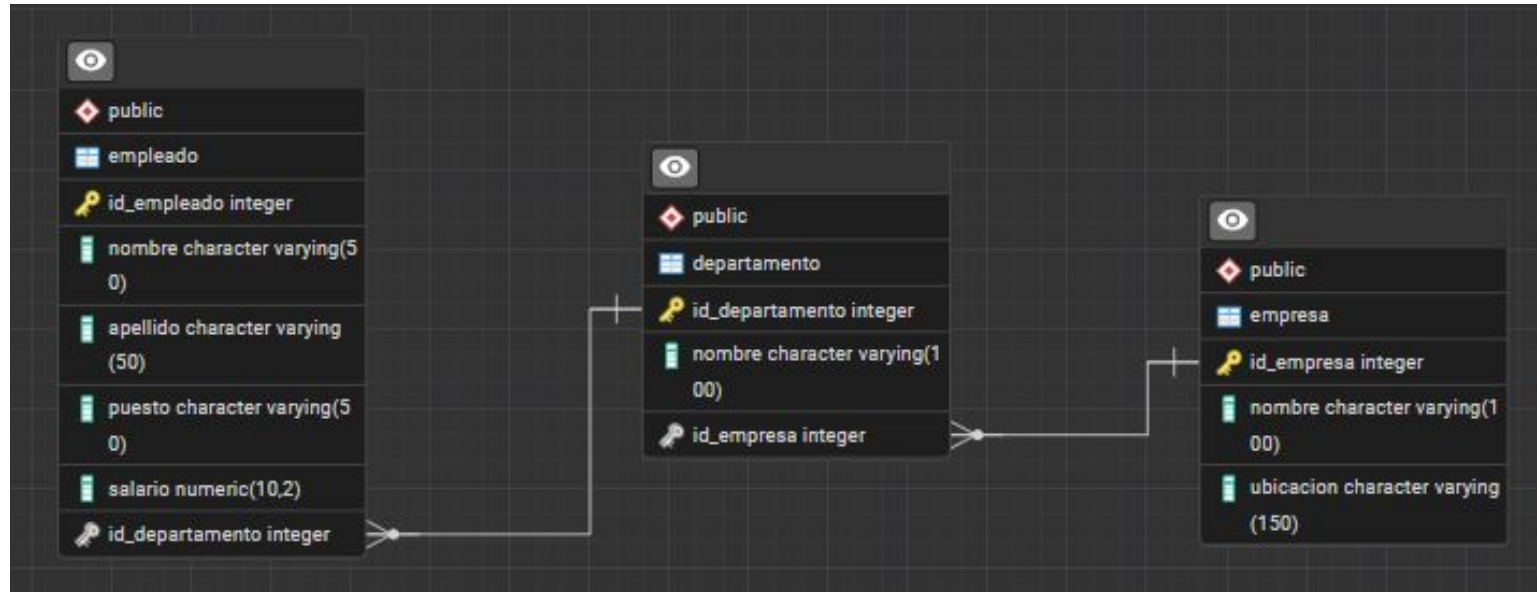
Definición de entidades

Empleado (ID, Nombre, Apellido, Puesto, Salario, ID_Departamento)

Departamento (ID_Departamento, Nombre)

Empresa (ID_Empresa, Nombre, Ubicación)

Diagrama modelo E-R



Esquema Relacional

Empresa (ID_Empresa, Nombre, Ubicacion)

- ID_Empresa → **Clave Primaria**

Departamento (ID_Departamento, Nombre, ID_Empresa)

- ID_Departamento → **Clave Primaria**
- ID_Empresa → **Clave Foránea** → Empresa(ID_Empresa)

Empleado (ID_Empleado, Nombre, Apellido, Puesto, Salario, ID_Departamento)

- ID_Empleado → **Clave Primaria**
- ID_Departamento → **Clave Foránea** → Departamento(ID_Departamento)

Implementación de la base de datos en SQL

```
CREATE TABLE Empresa (  
    ID_Empresa INT PRIMARY KEY,  
    Nombre VARCHAR(100) NOT NULL,  
    Ubicacion VARCHAR(150) NOT NULL  
);  
  
CREATE TABLE Departamento (  
    ID_Departamento INT PRIMARY KEY,  
    Nombre VARCHAR(100) NOT NULL,  
    ID_Empresa INT,  
    FOREIGN KEY (ID_Empresa) REFERENCES Empresa(ID_Empresa)  
);  
  
CREATE TABLE Empleado (  
    ID_Empleado INT PRIMARY KEY,  
    Nombre VARCHAR(50) NOT NULL,  
    Apellido VARCHAR(50) NOT NULL,  
    Puesto VARCHAR(50) NOT NULL,  
    Salario DECIMAL(10,2) NOT NULL,  
    ID_Departamento INT,  
    FOREIGN KEY (ID_Departamento) REFERENCES Departamento(ID_Departamento)  
);
```

Consultar empleados por departamento mediante SQL

```
1 SELECT Empleado.ID_Empleado, Empleado.Nombre, Empleado.Apellido, Empleado.Puesto,  
2     Empleado.Salario, Departamento.Nombre AS Departamento  
3 FROM Empleado  
4 JOIN Departamento ON Empleado.ID_Departamento = Departamento.ID_Departamento  
5 WHERE Departamento.Nombre = 'Desarrollo';  
6
```

Data Output Mensajes Notificaciones

SQL

Showing results

	id_empleado integer	nombre character varying (50)	apellido character varying (50)	puesto character varying (50)	salario numeric (10,2)	departamento character varying (100)
1	1	Juan	Pérez	Desarrollador	45000.00	Desarrollo
2	4	María	López	Analista	50000.00	Desarrollo
3	8	Elena	Torres	Diseñadora UX	48000.00	Desarrollo

5. Plataforma de Cursos en Línea

Identificación de entidades clave

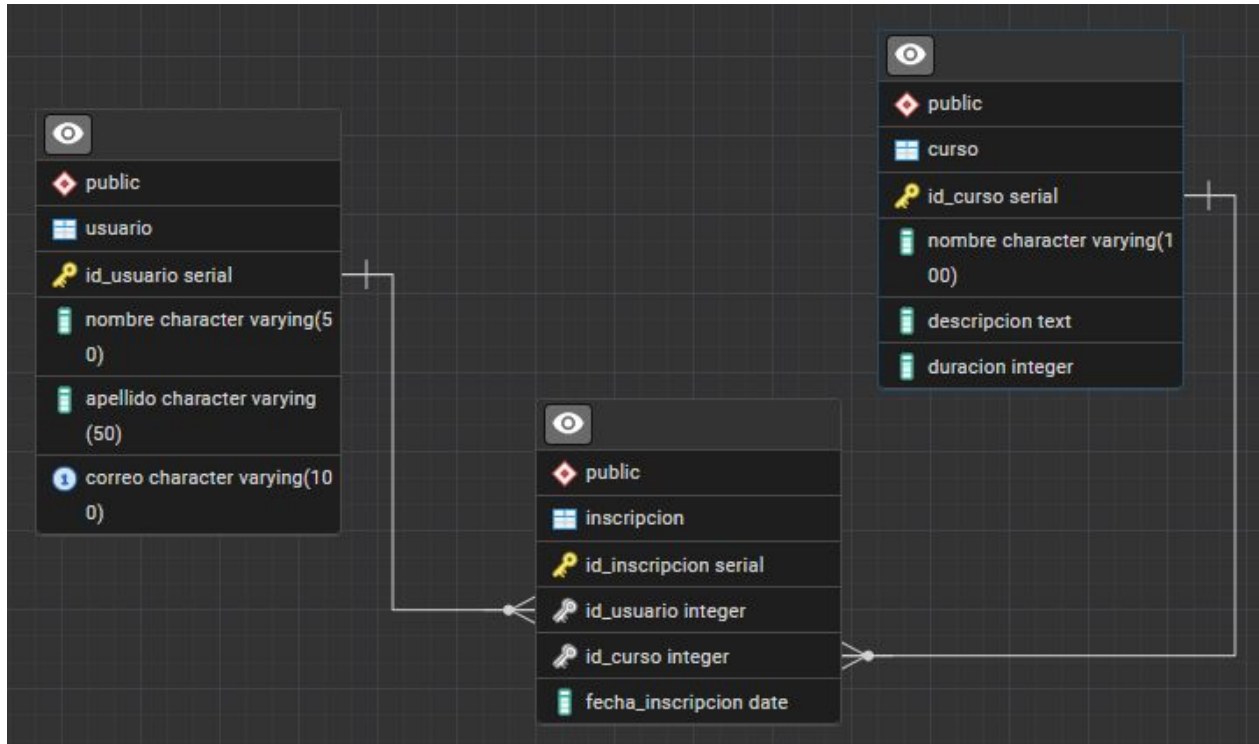
Las entidades clave en el sistema son:

- **Usuario** (ID_Usuario, Nombre, Apellido, Correo)
- **Curso** (ID_Curso, Nombre, Descripción, Duración)
- **Inscripción** (ID_Inscripción, ID_Usuario, ID_Curso, Fecha_Inscripción)

Relaciones:

- Un **usuario** puede inscribirse en varios **cursos**.
- Un **curso** puede tener varios **usuarios** inscritos.

Diseño de modelo E-R



Transformación de modelo en un conjunto de tablas relacionales

Tabla Usuario

	id_usuario [PK] integer	nombre character varying (50)	apellido character varying (50)	correo character varying (100)
1	1	Juan	Pérez	juan.perez@email.com
2	2	Ana	Gómez	ana.gomez@email.com
3	3	Luis	Martínez	luis.martinez@email.com
4	4	María	López	maria.lopez@email.com
5	5	Pedro	Ramírez	pedro.ramirez@email.com
6	6	Sofía	Fernández	sofia.fernandez@email.com
7	7	Carlos	Santos	carlos.santos@email.com
8	8	Elena	Torres	elena.torres@email.com
9	9	Ricardo	Vargas	ricardo.vargas@email.com
10	10	Gabriela	Jiménez	gabriela.jimenez@email.com

Tabla Curso

	id_curso [PK] integer 	nombre character varying (100) 	descripcion text 	duracion integer 
1	1	SQL Básico	Curso introductorio de SQL	10
2	2	Python para Análisis de Datos	Curso de análisis de datos con Python	15
3	3	Desarrollo Web con HTML y CSS	Curso práctico sobre HTML y CSS	12
4	4	JavaScript Avanzado	Curso avanzado de JavaScript	20
5	5	Bases de Datos Relacionales	Conceptos clave sobre bases de datos	18
6	6	Inteligencia Artificial	Introducción a IA y Machine Learning	25
7	7	Gestión de Proyectos	Fundamentos de gestión y metodologías ágiles	14
8	8	Ciberseguridad	Principios básicos de seguridad informática	16
9	9	Big Data y Analytics	Curso sobre análisis de grandes volúmenes de datos	22
10	10	Desarrollo Móvil con Android	Curso sobre desarrollo de apps en Android	19

Tabla Inscripcion

	Id_Inscripcion [PK] integer 	Id_usuario integer 	Id_curso integer 	fecha_inscripcion date 
1	1	1	1	2024-03-01
2	2	2	3	2024-03-02
3	3	3	2	2024-03-03
4	4	4	4	2024-03-04
5	5	5	5	2024-03-05
6	6	6	6	2024-03-06
7	7	7	7	2024-03-07
8	8	8	8	2024-03-08
9	9	9	9	2024-03-09
10	10	10	10	2024-03-10

Implementación de la base de datos en SQL

```
CREATE TABLE Usuario (  
    ID_Usuario SERIAL PRIMARY KEY,  
    Nombre VARCHAR(50) NOT NULL,  
    Apellido VARCHAR(50) NOT NULL,  
    Correo VARCHAR(100) UNIQUE NOT NULL  
);  
  
CREATE TABLE Curso (  
    ID_Curso SERIAL PRIMARY KEY,  
    Nombre VARCHAR(100) NOT NULL,  
    Descripcion TEXT,  
    Duracion INT NOT NULL -- Duración en horas  
);  
  
CREATE TABLE Inscripcion (  
    ID_Inscripcion SERIAL PRIMARY KEY,  
    ID_Usuario INT NOT NULL,  
    ID_Curso INT NOT NULL,  
    Fecha_Inscripcion DATE NOT NULL,  
    FOREIGN KEY (ID_Usuario) REFERENCES Usuario(ID_Usuario) ON DELETE CASCADE,  
    FOREIGN KEY (ID_Curso) REFERENCES Curso(ID_Curso) ON DELETE CASCADE  
);
```

Consultar los cursos inscritos por un usuario en la base de datos

```
SELECT u.Nombre AS Usuario, u.Apellido, c.Nombre AS Curso, c.Duracion, 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.Correo = 'gabriela.jimenez@email.com';
```

data Output Mensajes Notificaciones

SQL Show

usuario	apellido	curso	duracion	fecha_inscripcion
character varying (50)	character varying (50)	character varying (100)	integer	date
Gabriela	Jiménez	Desarrollo Móvil con Android	19	2024-03-10