TAREA UNIDAD 2

Fundamentos de base de datos

15 DE MARZO DE 2025

DAMIAN ALEXANDER LUGO AGUILAR- 23490386
Prof. Jose Ramon Bogarin Valenzuela

Indice

Sistema de gestion de hospitales 3

Tienda en linea 5

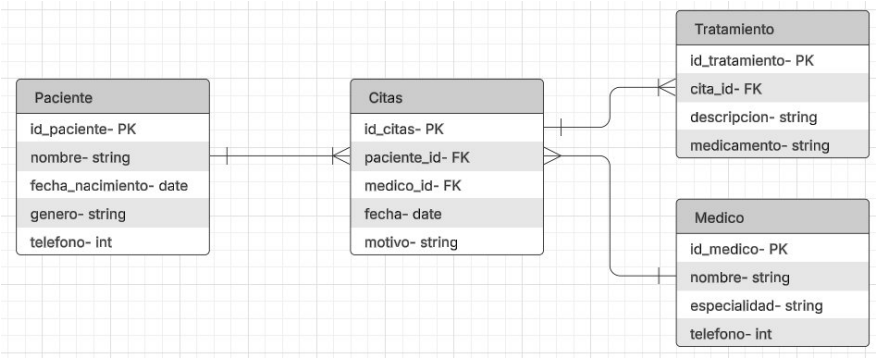
Bibliotecta digital 7

Sistema de recursos humanos 9

Plataforma de cursos en linea 11

Sistema de gestion de hospitales

Teniendo las siguientes entidades: pacientes, citas, medicos y tratamiento. Podemos hacer el diagrama de entidad-relacion



Con esto podemos crear nuestras tablas en query

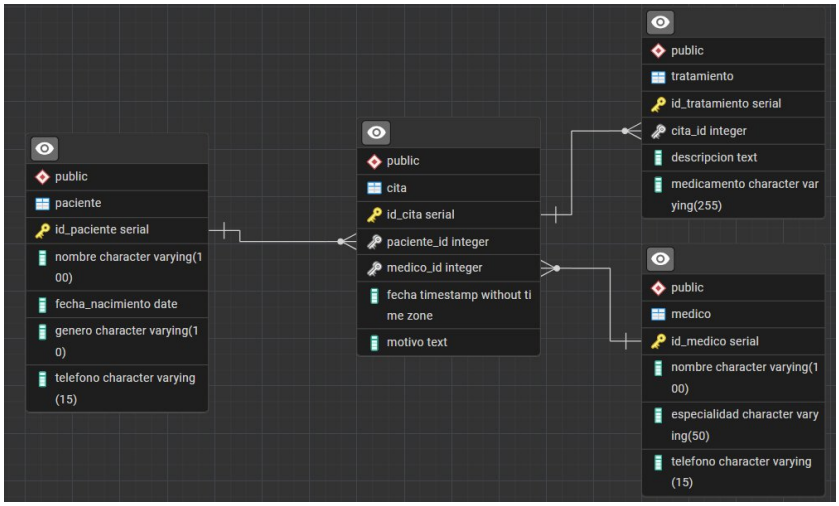
```sql
CREATE TABLE Paciente (
    id_paciente SERIAL PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    fecha_nacimiento DATE NOT NULL,
    genero VARCHAR(10),
    telefono VARCHAR(15)
);

CREATE TABLE Medico (
    id_medico SERIAL PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    especialidad VARCHAR(50) NOT NULL,
    telefono VARCHAR(15)
);

CREATE TABLE Cita (
    id_cita SERIAL PRIMARY KEY,
    paciente_id INT NOT NULL,
    medico_id INT NOT NULL,
    fecha TIMESTAMP NOT NULL,
    motivo TEXT,
    FOREIGN KEY (paciente_id) REFERENCES Paciente(id_paciente) ON DELETE CASCADE,
    FOREIGN KEY (medico_id) REFERENCES Medico(id_medico) ON DELETE CASCADE
);
```

```sql
CREATE TABLE Tratamiento (
    id_tratamiento SERIAL PRIMARY KEY,
    cita_id INT NOT NULL,
    descripcion TEXT NOT NULL,
    medicamento VARCHAR(255),
    FOREIGN KEY (cita_id) REFERENCES Cita(id_cita) ON DELETE CASCADE
);
```

Una vez creadas podemos comprobar que nuestro diagrama inicial es parecido al hecho por postgres con este query



Ahora haremos algunas instruccion, primero llenaremos nuestras tablas con un insert into

```sql
INSERT INTO Paciente (nombre, fecha_nacimiento, genero, telefono) VALUES
('Juan Pérez', '1990-05-14', 'Masculino', '555-1234'),
('María López', '1985-07-22', 'Femenino', '555-2345'),
('Carlos Méndez', '1992-11-10', 'Masculino', '555-3456'),
('Ana Torres', '2000-03-15', 'Femenino', '555-4567'),
('Pedro Ramírez', '1988-09-25', 'Masculino', '555-5678'),
('Luisa Fernández', '1995-12-05', 'Femenino', '555-6789'),
('Jorge Castillo', '1980-06-17', 'Masculino', '555-7890'),
('Beatriz Gómez', '1975-08-30', 'Femenino', '555-8901'),
('Raúl Sánchez', '1999-01-12', 'Masculino', '555-9012'),
('Sofía Herrera', '2003-05-07', 'Femenino', '555-0123');
```

```sql
INSERT INTO Medico (nombre, especialidad, telefono) VALUES
('Dra. Ana Gómez', 'Cardiología', '555-5678'),
('Dr. Luis Martínez', 'Pediatría', '555-1111'),
('Dra. Camila Rojas', 'Dermatología', '555-2222'),
('Dr. Eduardo Pérez', 'Neurología', '555-3333'),
('Dra. Valeria Sánchez', 'Ginecología', '555-4444'),
('Dr. Martín Torres', 'Oftalmología', '555-5555'),
('Dra. Gabriela Castro', 'Oncología', '555-6666'),
('Dr. Felipe Gómez', 'Ortopedia', '555-7777'),
('Dra. Natalia Jiménez', 'Psiquiatría', '555-8888'),
('Dr. Andrés Suárez', 'Cardiología', '555-9999');

INSERT INTO Cita (paciente_id, medico_id, fecha, motivo) VALUES
(1, 1, '2025-03-20 10:00:00', 'Chequeo anual'),
(2, 2, '2025-03-21 11:00:00', 'Revisión de piel'),
(3, 3, '2025-03-22 14:30:00', 'Dolores de cabeza frecuentes'),
(4, 4, '2025-03-23 09:45:00', 'Chequeo ginecológico'),
(5, 5, '2025-03-24 08:30:00', 'Problemas de visión'),
(6, 6, '2025-03-25 16:00:00', 'Evaluación oncológica'),
(7, 7, '2025-03-26 13:15:00', 'Dolor de rodilla'),
(8, 8, '2025-03-27 12:00:00', 'Consulta psiquiátrica'),
(9, 9, '2025-03-28 10:45:00', 'Chequeo de corazón'),
(10, 1, '2025-03-29 15:30:00', 'Chequeo pediátrico');
```

```sql
INSERT INTO Tratamiento (cita_id, descripcion, medicamento) VALUES
(1, 'Tratamiento para hipertensión', 'Losartán 50mg'),
(2, 'Tratamiento para dermatitis', 'Cremas y antihistamínicos'),
(3, 'Tratamiento para migrañas', 'Sumatriptán'),
(4, 'Tratamiento para salud femenina', 'Vitaminas y control hormonal'),
(5, 'Corrección visual', 'Uso de lentes'),
(6, 'Tratamiento oncológico', 'Quimioterapia'),
(7, 'Fisioterapia para rodilla', 'Ejercicios y analgésicos'),
(8, 'Tratamiento para ansiedad', 'Sertralina 50mg'),
(9, 'Control de presión arterial', 'Atenolol 25mg'),
(10, 'Suplementos pediátricos', 'Hierro y vitamina D');
```

Teniendo esto finalmente podremos hacer una prueba de nuestras tablas, para ello consultaremos las citas de un paciente en especifico, en este caso escogeremos al paciente “Carlos Mendez” el query quedaria asi:

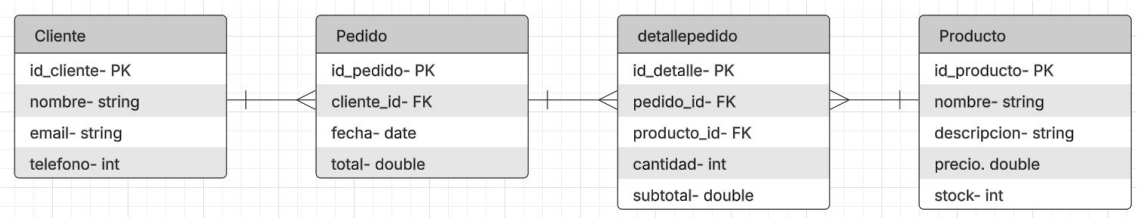
```sql
SELECT c.id_cita AS id_cita, c.fecha, c.motivo, m.nombre AS nombre_medico, m.especialidad
FROM cita c INNER JOIN medico m
ON c.medico_id = m.id_medico
INNER JOIN paciente p
ON c.paciente_id = p.id_paciente
WHERE p.nombre = 'Carlos Méndez'
```

Lo que nos da como resultado:

	id_cita integer	fecha timestamp without time zone	motivo text	nombre_medico character varying (100)	especialidad character varying (50)
1	3	2025-03-22 14:30:00	Dolores de cabeza frecuentes	Dra. Camila Rojas	Dermatología

Tienda en linea

Una empresa quiere mejorar la administracion de sus pedidos en linea, para ello tenemos las siguientes entidades: Cliente, producto, pedido, detallepedido. Con esto podemos hacer nuestro diagrama de entidad-relacion



Con ello podemos crear nuestras tablas:

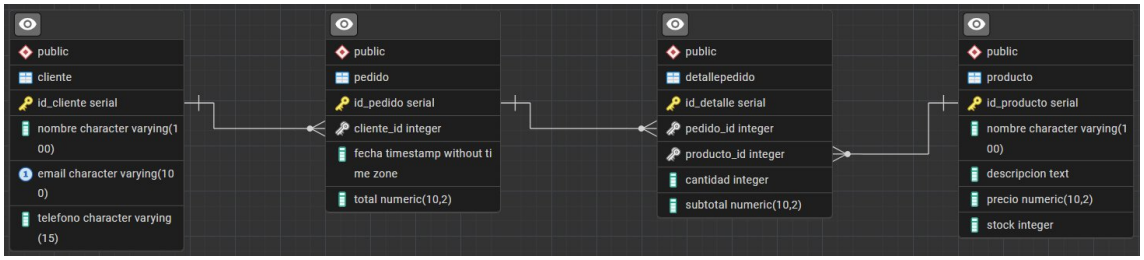
```sql
CREATE TABLE Cliente (
    id_cliente SERIAL PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    telefono VARCHAR(15)
);

CREATE TABLE Producto (
    id_producto SERIAL PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    descripcion TEXT,
    precio DECIMAL(10,2) NOT NULL,
    stock INT NOT NULL CHECK (stock >= 0)
);

CREATE TABLE Pedido (
    id_pedido SERIAL PRIMARY KEY,
    cliente_id INT NOT NULL,
    fecha TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    total DECIMAL(10,2) NOT NULL CHECK (total >= 0),
    FOREIGN KEY (cliente_id) REFERENCES Cliente(id_cliente) ON DELETE CASCADE
);
```

```sql
CREATE TABLE DetallePedido (
    id_detalle SERIAL PRIMARY KEY,
    pedido_id INT NOT NULL,
    producto_id INT NOT NULL,
    cantidad INT NOT NULL CHECK (cantidad > 0),
    subtotal DECIMAL(10,2) NOT NULL CHECK (subtotal >= 0),
    FOREIGN KEY (pedido_id) REFERENCES Pedido(id_pedido) ON DELETE CASCADE,
    FOREIGN KEY (producto_id) REFERENCES Producto(id_producto) ON DELETE CASCADE
);
```

Con ello podemos ver que nos queda igual que en nuestro diagrama de entidad-relacion



Ahora haremos un query para insertar datos:

```sql
INSERT INTO Cliente (nombre, email, telefono) VALUES
('Juan Pérez', 'juan.perez@email.com', '555-1234'),
('María López', 'maria.lopez@email.com', '555-2345'),
('Carlos Méndez', 'carlos.mendez@email.com', '555-3456'),
('Ana Torres', 'ana.torres@email.com', '555-4567'),
('Pedro Ramírez', 'pedro.ramirez@email.com', '555-5678'),
('Luis Fernández', 'luis.fernandez@email.com', '555-6789'),
('Gabriela Castro', 'gabriela.castro@email.com', '555-7890'),
('Jorge Castillo', 'jorge.castillo@email.com', '555-8901'),
('Beatriz Gómez', 'beatriz.gomez@email.com', '555-9012'),
('Raúl Sánchez', 'raul.sanchez@email.com', '555-0123');

INSERT INTO Producto (nombre, descripcion, precio, stock) VALUES
('Laptop', 'Laptop de 15 pulgadas con 16GB RAM', 1200.50, 10),
('Teléfono', 'Smartphone 5G con 128GB de almacenamiento', 850.75, 15),
('Auriculares', 'Auriculares inalámbricos con cancelación de ruido', 200.99, 25),
('Teclado', 'Teclado mecánico retroiluminado', 99.90, 30),
('Monitor', 'Monitor 4K de 27 pulgadas', 450.00, 20),
('Mouse', 'Mouse ergonómico inalámbrico', 49.99, 50),
('Impresora', 'Impresora multifuncional láser', 299.99, 10),
('Tablet', 'Tablet 10 pulgadas con 64GB almacenamiento', 399.99, 20),
('Silla Gamer', 'Silla ergonómica para gaming y oficina', 199.99, 15),
('Disco Duro', 'Disco Duro externo 1TB USB 3.0', 89.99, 30);
```

```sql
INSERT INTO Pedido (cliente_id, total) VALUES
(1, 2100.50),
(2, 350.75),
(3, 99.90),
(4, 450.00),
(5, 1200.50),
(6, 349.98),
(7, 199.99),
(8, 389.98),
(9, 99.99),
(10, 450.00);

INSERT INTO DetallePedido (pedido_id, producto_id, cantidad, subtotal) VALUES
(1, 1, 1, 1200.50),
(1, 2, 1, 850.00),
(2, 3, 1, 200.99),
(2, 4, 1, 99.90),
(3, 4, 1, 99.90),
(4, 5, 1, 450.00),
(5, 1, 1, 1200.50),
(6, 6, 2, 99.98),
(6, 7, 1, 299.99),
(7, 9, 1, 199.99),
(8, 10, 2, 389.98),
(9, 6, 2, 99.99);
```

Con esto podemos pedir consultar especificas como en el anterior ejercicio, en este caso pediremos los productos comprados de un cliente, el cual sera “María López”

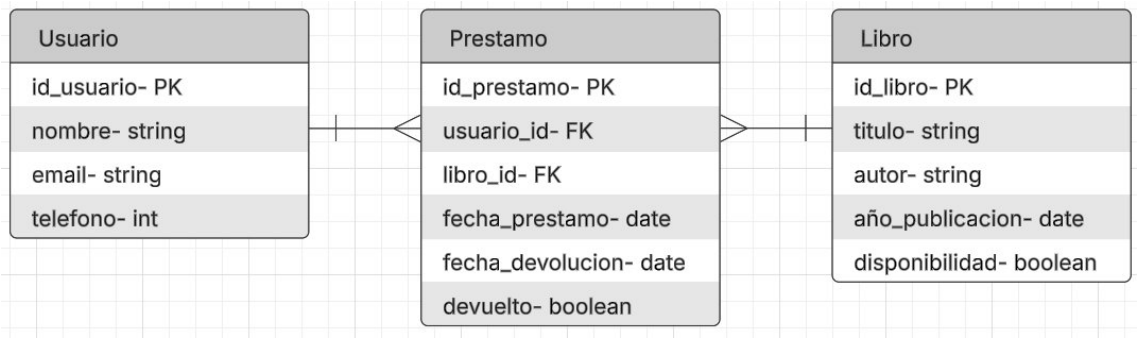
```sql
SELECT p.id_pedido AS idpedido_PK, prod.nombre AS producto, det.cantidad, det.subtotal
FROM pedido p
INNER JOIN detallepedido det
ON p.id_pedido = det.pedido_id
INNER JOIN producto prod
ON det.producto_id = prod.id_producto
INNER JOIN cliente c
ON p.cliente_id = c.id_cliente
WHERE c.nombre = 'María López';
```

Lo que nos da como resultado

	idpedido_pk integer	producto character varying (100)	cantidad integer	subtotal numeric (10,2)
1	2	Auriculares	1	200.99
2	2	Teclado	1	99.90

Bibliotecta digital

Para un sistema de administracion de prestamos de libros en una biblioteca digital tenemos las siguientes entidades: Usuario, libro, prestamo. Teniendo esto podemos crear nuestro diagrama de entidad-relacion



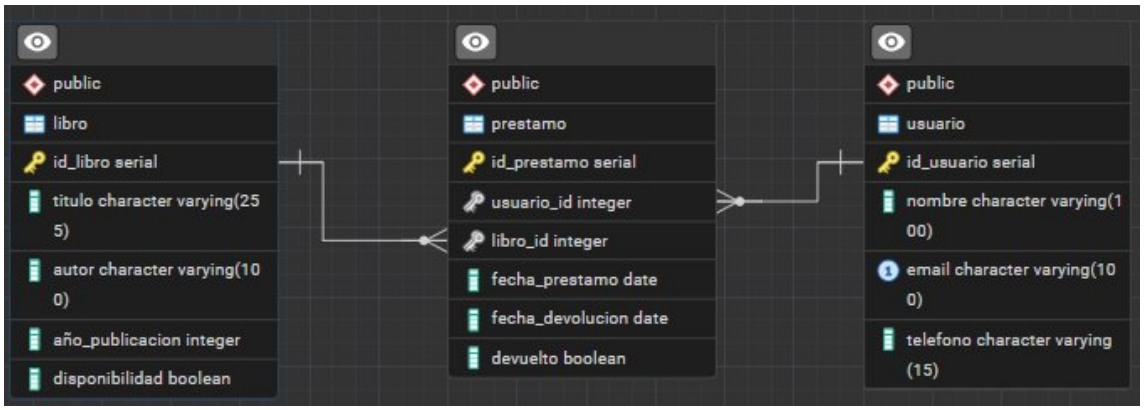
Ahora con esto crearemos nuestro query

```sql
CREATE TABLE Usuario (
    id_usuario SERIAL PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    telefono VARCHAR(15)
);

CREATE TABLE Libro (
    id_libro SERIAL PRIMARY KEY,
    titulo VARCHAR(255) NOT NULL,
    autor VARCHAR(100) NOT NULL,
    año_publicacion INT,
    disponibilidad BOOLEAN DEFAULT TRUE
);

CREATE TABLE Prestamo (
    id_prestamo SERIAL PRIMARY KEY,
    usuario_id INT NOT NULL,
    libro_id INT NOT NULL,
    fecha_prestamo DATE DEFAULT CURRENT_DATE,
    fecha_devolucion DATE,
    devuelto BOOLEAN DEFAULT FALSE,
    FOREIGN KEY (usuario_id) REFERENCES Usuario(id_usuario) ON DELETE CASCADE,
    FOREIGN KEY (libro_id) REFERENCES Libro(id_libro) ON DELETE CASCADE
);
```

Y ahora podemos ver que nuestro diagramas E-R y el query coinciden



Ahora insertamos nuestros datos en las tablas

```sql
INSERT INTO Usuario (nombre, email, telefono) VALUES
('Juan Pérez', 'juan.perez@email.com', '555-1234'),
('María López', 'maria.lopez@email.com', '555-2345'),
('Carlos Méndez', 'carlos.mendez@email.com', '555-3456'),
('Ana Torres', 'ana.torres@email.com', '555-4567'),
('Pedro Ramírez', 'pedro.ramirez@email.com', '555-5678'),
('Sofía Herrera', 'sofia.herrera@email.com', '555-6789'),
('David Guzmán', 'david.guzman@email.com', '555-7890'),
('Fernanda Rojas', 'fernanda.rojas@email.com', '555-8901'),
('Ricardo Vargas', 'ricardo.vargas@email.com', '555-9012'),
('Elena Morales', 'elena.morales@email.com', '555-0123');

INSERT INTO Libro (titulo, autor, año_publicacion, disponibilidad) VALUES
('Cien años de soledad', 'Gabriel García Márquez', 1967, TRUE),
('1984', 'George Orwell', 1949, TRUE),
('El principito', 'Antoine de Saint-Exupéry', 1943, TRUE),
('Don Quijote de la Mancha', 'Miguel de Cervantes', 1605, TRUE),
('Crónica de una muerte anunciada', 'Gabriel García Márquez', 1981, TRUE),
('La sombra del viento', 'Carlos Ruiz Zafón', 2001, TRUE),
('El código Da Vinci', 'Dan Brown', 2003, TRUE),
('Los juegos del hambre', 'Suzanne Collins', 2008, TRUE),
('Rayuela', 'Julio Cortázar', 1963, TRUE),
('El Hobbit', 'J.R.R. Tolkien', 1937, TRUE);
```

```sql
INSERT INTO Prestamo (usuario_id, libro_id, fecha_prestamo, fecha_devolucion, devuelto) VALUES
(1, 1, '2025-03-01', '2025-03-15', FALSE),
(2, 3, '2025-03-05', '2025-03-20', FALSE),
(3, 2, '2025-02-28', '2025-03-14', TRUE),
(4, 5, '2025-03-10', '2025-03-24', FALSE),
(5, 4, '2025-03-12', '2025-03-26', FALSE),
(6, 6, '2025-03-15', '2025-03-30', FALSE),
(7, 8, '2025-03-18', '2025-04-02', FALSE),
(8, 7, '2025-03-20', '2025-04-05', FALSE),
(9, 9, '2025-03-22', '2025-04-07', FALSE),
(10, 10, '2025-03-25', '2025-04-10', FALSE);
```

Una vez teniendo nuestras tablas llenas podemos cumplir con la ultima instruccion de consultar los prestamos activos de un usuario

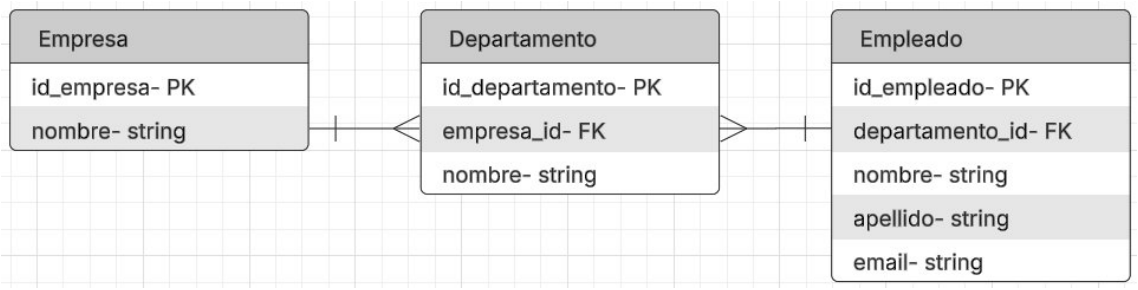
```sql
SELECT p.id_prestamo AS id_de_prestamo, l.titulo AS libro, p.fecha_prestamo, p.fecha_devolucion
FROM prestamo p
INNER JOIN libro l
ON p.libro_id = l.id_libro
INNER JOIN usuario u
ON p.usuario_id = u.id_usuario
WHERE u.nombre = 'Juan Pérez' AND p.devuelto= FALSE;
```

Resultando en la siguiente tabla

	id_de_prestamo integer	libro character varying (255)	fecha_prestamo date	fecha_devolucion date
1	1	Cien años de soledad	2025-03-01	2025-03-15

Sistema de recursos humanos

Necesitamos gestionar empleados y departamentos, primero haremos el diagrama de entidad-relacion



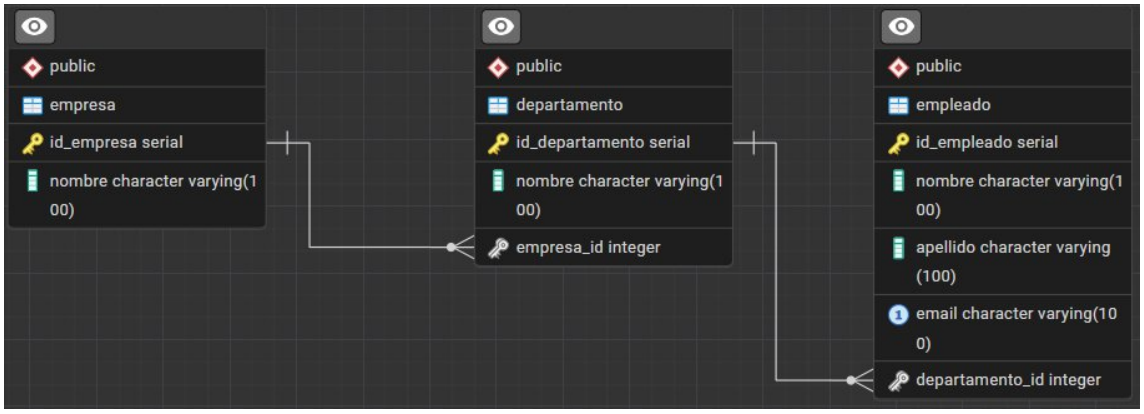
Ahora con esto haremos nuestras tablas

```sql
CREATE TABLE Empresa (
    id_empresa SERIAL PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL
);

CREATE TABLE Departamento (
    id_departamento SERIAL PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    empresa_id INT NOT NULL,
    FOREIGN KEY (empresa_id) REFERENCES Empresa(id_empresa) ON DELETE CASCADE
);

CREATE TABLE Empleado (
    id_empleado SERIAL PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    apellido VARCHAR(100) NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    departamento_id INT NOT NULL,
    FOREIGN KEY (departamento_id) REFERENCES Departamento(id_departamento) ON DELETE SET NULL
);
```

Y con esto podemos ver que nuestro diagrama de E-R coincide con el query}



Teniendo esto podemos llenar nuestras tablas

```sql
INSERT INTO Empresa (nombre) VALUES
('TechCorp'),
('Innovate Inc.');

INSERT INTO Departamento (nombre, id_empresa) VALUES
('Recursos Humanos', 1),
('Desarrollo', 1),
('Ventas', 2),
('Finanzas', 1),
('Soporte Técnico', 1),
('Marketing', 2),
('Logística', 2),
('Legal', 1),
('Atención al Cliente', 2),
('Investigación y Desarrollo', 1);

INSERT INTO Empleado (nombre, apellido, email, id_departamento) VALUES
('Juan', 'Pérez', 'juan.perez@email.com', 1),
('María', 'Gómez', 'maria.gomez@email.com', 2),
('Carlos', 'Díaz', 'carlos.diaz@email.com', 2),
('Ana', 'López', 'ana.lopez@email.com', 3),
('Pedro', 'Ramírez', 'pedro.ramirez@email.com', 4),
('Lucía', 'Fernández', 'lucia.fernandez@email.com', 5),
('Andrés', 'Torres', 'andres.torres@email.com', 6),
('Elena', 'Rojas', 'elena.rojas@email.com', 7),
('Sergio', 'Mendoza', 'sergio.mendoza@email.com', 8),
('Valeria', 'Suárez', 'valeria.suarez@email.com', 9),
('Gabriel', 'Ortega', 'gabriel.ortega@email.com', 10);
```

Ahora podemos consultar empleados por departamento

```sql
SELECT e.id_empleado, e.nombre, e.apellido, e.email, d.nombre AS departamento
FROM empleado e
INNER JOIN departamento d ON e.departamento_id = d.id_departamento
ORDER BY d.nombre;
```

	id_empleado integer	nombre character varying (100)	apellido character varying (100)	email character varying (100)	departamento character varying (100)
1	21	Valeria	Suárez	valeria.suarez@email.com	Atención al Cliente
2	14	Carlos	Díaz	carlos.diaz@email.com	Desarrollo
3	13	María	Gómez	maria.gomez@email.com	Desarrollo
4	16	Pedro	Ramírez	pedro.ramirez@email.com	Finanzas
5	22	Gabriel	Ortega	gabriel.ortega@email.com	Investigación y Desarrollo
6	20	Sergio	Mendoza	sergio.mendoza@email.c...	Legal
7	19	Elena	Rojas	elena.rojas@email.com	Logística
8	18	Andrés	Torres	andres.torres@email.com	Marketing
9	12	Juan	Pérez	juan.perez@email.com	Recursos Humanos
10	17	Lucía	Fernández	lucia.fernandez@email.com	Soporte Técnico
11	15	Ana	López	ana.lopez@email.com	Ventas

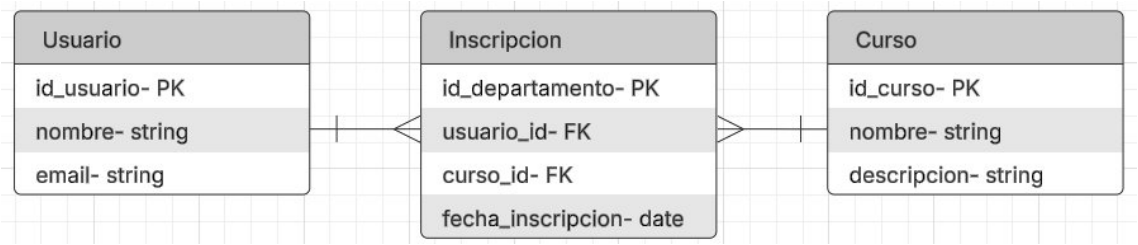
tambien podemos hacer lo siguiente:

```sql
SELECT e.id_empleado, e.nombre, e.apellido, e.email, d.nombre AS departamento
FROM empleado e
INNER JOIN departamento d ON e.departamento_id = d.id_departamento
WHERE d.nombre = 'Desarrollo';
```

	id_empleado integer	nombre character varying (100)	apellido character varying (100)	email character varying (100)	departamento character varying (100)
1	13	María	Gómez	maria.gomez@email.com	Desarrollo
2	14	Carlos	Díaz	carlos.diaz@email.com	Desarrollo

Plataforma de cursos en linea

Para un sistema par gestionar inscripciones de cursos en linea tenemos las siguientes entidades: Usuario, curso e inscripcion, con esto podemos crear nuestro diagrama entidad-relacion.



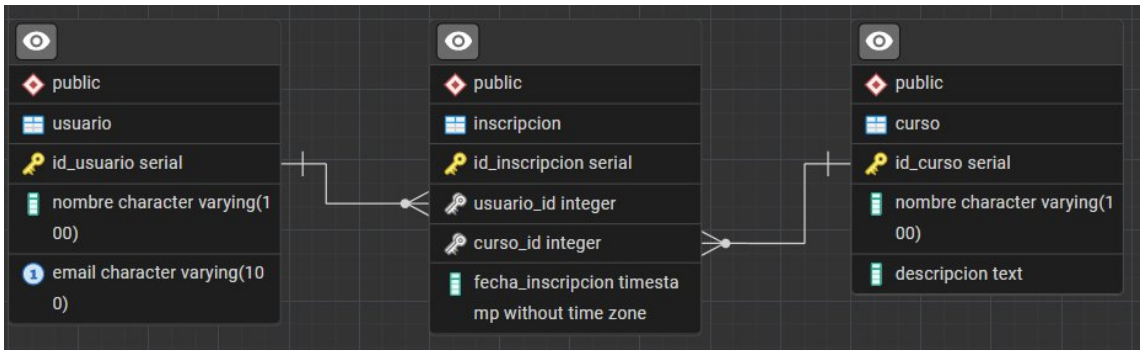
Con esto pasamos a la creacion de nuestras tablas

```
CREATE TABLE Usuario (
    id_usuario SERIAL PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL
);

CREATE TABLE Curso (
    id_curso SERIAL PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    descripcion TEXT
);

CREATE TABLE Inscripcion (
    id_inscripcion SERIAL PRIMARY KEY,
    usuario_id INT NOT NULL,
    curso_id INT NOT NULL,
    fecha_inscripcion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (usuario_id) REFERENCES Usuario(id_usuario) ON DELETE CASCADE,
    FOREIGN KEY (curso_id) REFERENCES Curso(id_curso) ON DELETE CASCADE,
    UNIQUE (usuario_id, curso_id)--evita que alguien se inscribe a un curso dos veces al mismo tiempo
);
```

Y ahora con esto podemos ver que ambos diagramas coinciden



Ahora pasaremos a insertar nuestros datos en las tablas

```sql
INSERT INTO Usuario (nombre, email) VALUES
('Juan Pérez', 'juan.perez@email.com'),
('María Gómez', 'maria.gomez@email.com'),
('Carlos Díaz', 'carlos.diaz@email.com'),
('Lucía Fernández', 'lucia.fernandez@email.com'),
('Andrés Torres', 'andres.torres@email.com'),
('Elena Rojas', 'elena.rojas@email.com'),
('Sergio Mendoza', 'sergio.mendoza@email.com'),
('Valeria Suárez', 'valeria.suarez@email.com'),
('Gabriel Ortega', 'gabriel.ortega@email.com'),
('Diana Castro', 'diana.castro@email.com');

INSERT INTO Curso (nombre, descripcion) VALUES
('PostgreSQL Básico', 'Introducción a PostgreSQL para principiantes.'),
('SQL Avanzado', 'Consultas avanzadas y optimización en SQL.'),
('Desarrollo Web', 'Fundamentos del desarrollo web moderno.'),
('Python para Principiantes', 'Curso básico de programación en Python.')
('Machine Learning', 'Introducción al aprendizaje automático.'),
('Ciberseguridad', 'Principios de seguridad informática.'),
('Redes de Computadoras', 'Conceptos clave sobre redes y comunicación.')
('Gestión de Proyectos', 'Fundamentos para la gestión de proyectos.'),
('Big Data y Analítica', 'Análisis de datos a gran escala.'),
('Blockchain y Criptomonedas', 'Fundamentos de tecnología blockchain.');
```

```sql
INSERT INTO Inscripcion (usuario_id, curso_id) VALUES
(21, 21),
(21, 22),
(22, 23),
(23, 21),
(23, 23),
(24, 24),
(25, 25),
(26, 26),
(27, 27),
(28, 28);
```

Finalmente podemos consultar los cursos inscritos por un usuario, nuestro usuario en este caso sera “María Gómez”

```sql
SELECT u.nombre AS usuario, c.nombre AS curso, i.fecha_inscripcion
FROM inscripcion i
INNER JOIN usuario u
ON i.usuario_id = u.id_usuario
INNER JOIN curso c
ON i.usuario_id = c.id_curso
where u.nombre = 'María Gómez';
```

El resultado seria el siguiente

	usuario character varying (100)	curso character varying (100)	fecha_inscripcion timestamp without time zone
1	María Gómez	SQL Avanzado	2025-03-18 01:40:14.739092