

# Sistema de Gestión de Hospitales

## Introducción

### Descripción del problema:

El sistema de gestión de hospitales busca organizar y administrar información sobre pacientes, médicos, citas y tratamientos. Este sistema facilitará el acceso rápido a los datos, mejorando la eficiencia en la administración hospitalaria.

### Objetivo:

El objetivo es crear una base de datos que permita gestionar la información de pacientes, médicos, citas y tratamientos, con la capacidad de consultar y modificar los datos de manera eficiente.

## Análisis de Requerimientos

### Entidades clave:

- **Paciente:** Representa a los pacientes que visitan el hospital.
- **Médico:** Representa a los médicos que atienden a los pacientes.
- **Cita:** Representa las citas médicas entre pacientes y médicos.
- **Tratamiento:** Representa los tratamientos asociados a las citas médicas.

### Atributos de las entidades:

- **Paciente:** id\_paciente, nombre, edad, direccion.
- **Médico:** id\_medico, nombre, especialidad, tel\_contacto.
- **Cita:** id\_cita, id\_paciente, id\_medico, fecha, motivo.
- **Tratamiento:** id\_tratamiento, id\_cita, descripcion, duracion\_dias.

### Relaciones entre las entidades:

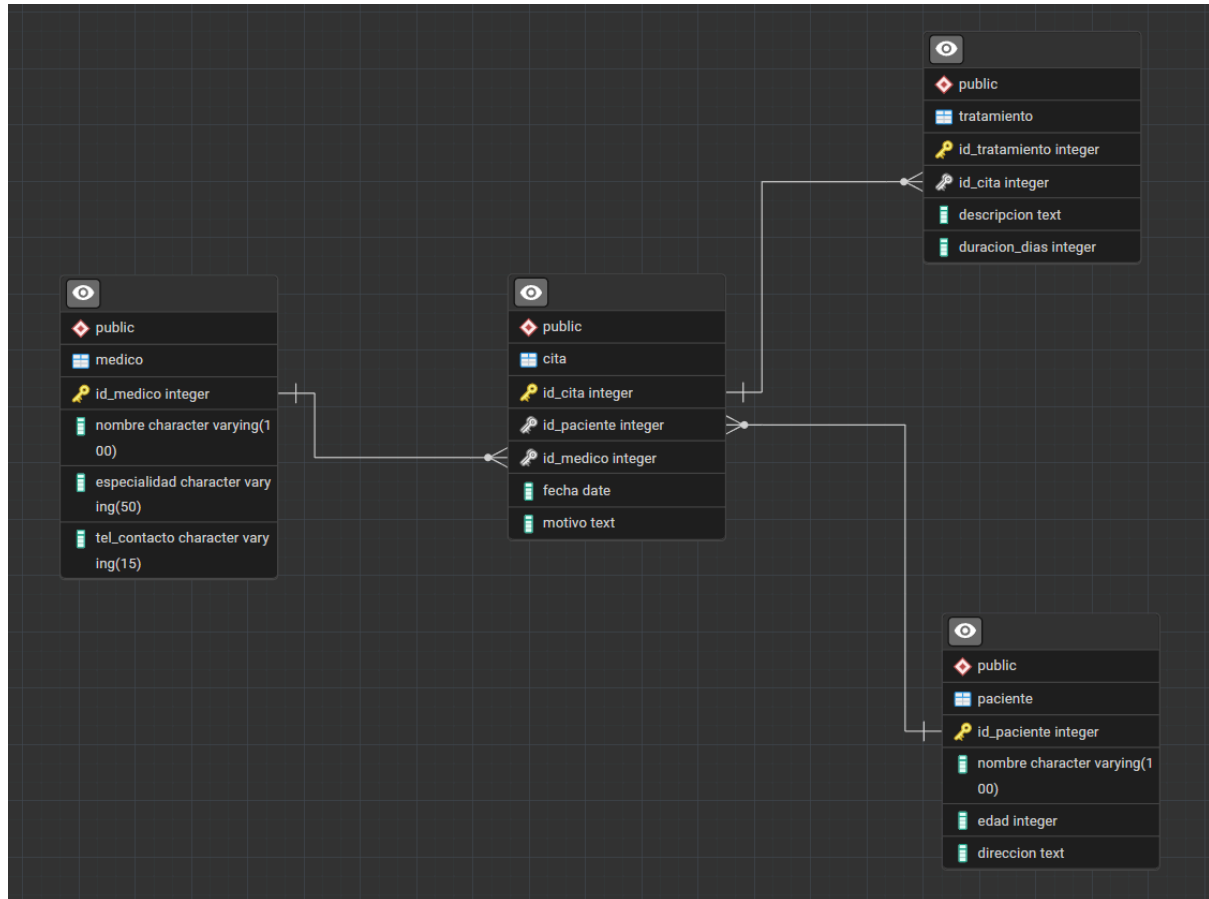
- Un **Paciente** puede tener muchas **Citas**.
- Una **Cita** está asociada a un único **Médico** y a un único **Paciente**.
- Una **Cita** puede tener muchos **Tratamientos**, pero cada tratamiento está asociado a una sola cita.

---

## Modelo Entidad-Relación (E-R)

A continuación, se presenta el diagrama E-R que muestra las relaciones entre las entidades clave del sistema.

## Diagrama E-R:



## Conversión a Esquema Relacional

### Descripción del esquema relacional:

El modelo E-R se convierte en un conjunto de tablas que representan las entidades y sus relaciones. Cada tabla tiene claves primarias y foráneas para asegurar la integridad referencial.

- **Tabla Paciente:**
  - Clave primaria: **id\_paciente**
- **Tabla Médico:**
  - Clave primaria: **id\_medico**
- **Tabla Cita:**
  - Claves primarias: **id\_cita**

- Claves foráneas: id\_paciente (referencia a Paciente), id\_medico (referencia a Medico)
  - **Tabla Tratamiento:**
    - Clave primaria: id\_tratamiento
    - Clave foránea: id\_cita (referencia a Cita)
- 

## Implementación en DBMS (LDD)

```
CREATE TABLE Paciente (  
    id_paciente INT PRIMARY KEY,  
    nombre VARCHAR(100),  
    edad INT,  
    direccion TEXT  
);  
  
CREATE TABLE Medico (  
    id_medico INT PRIMARY KEY,  
    nombre VARCHAR(100),  
    especialidad VARCHAR(50),  
    tel_contacto VARCHAR(15)  
);  
  
CREATE TABLE Cita (  
    id_cita INT PRIMARY KEY,  
    id_paciente INT,  
    id_medico INT,  
    fecha DATE,  
    motivo TEXT,  
    FOREIGN KEY (id_paciente) REFERENCES Paciente(id_paciente),  
    FOREIGN KEY (id_medico) REFERENCES Medico(id_medico)  
);  
  
CREATE TABLE Tratamiento (  
    id_tratamiento INT PRIMARY KEY,  
    id_cita INT,  
    descripcion TEXT,  
    duracion_dias INT,  
    FOREIGN KEY (id_cita) REFERENCES Cita(id_cita)  
);
```

---

## Manipulación de Datos (LMD)

```
45 INSERT INTO Cita (id_cita, id_paciente, id_medico, fecha, motivo) VALUES
46 (1, 1, 1, '2025-03-20', 'Chequeo general'),
47 (2, 2, 2, '2025-03-21', 'Revisión anual'),
48 (3, 3, 3, '2025-03-22', 'Control de piel'),
49 (4, 4, 4, '2025-03-23', 'Dolor de cabeza'),
50 (5, 5, 5, '2025-03-24', 'Revisión dental'),
51 (6, 6, 6, '2025-03-25', 'Control ginecológico'),
52 (7, 7, 7, '2025-03-26', 'Consulta general'),
53 (8, 8, 8, '2025-03-27', 'Fractura brazo'),
54 (9, 9, 9, '2025-03-28', 'Control ocular'),
55 (10, 10, 10, '2025-03-29', 'Revisión de cáncer'),
56 (11, 11, 11, '2025-03-30', 'Chequeo infantil'),
57 (12, 12, 12, '2025-03-31', 'Cirugía de rodilla'),
58 (13, 13, 13, '2025-04-01', 'Consulta urológica'),
59 (14, 14, 14, '2025-04-02', 'Chequeo estomacal'),
60 (15, 15, 15, '2025-04-03', 'Revisión cardiológica'),
61 (16, 16, 16, '2025-04-04', 'Consulta psicológica'),
62 (17, 17, 17, '2025-04-05', 'Consulta de cirugía'),
63 (18, 18, 18, '2025-04-06', 'Dolor de articulaciones'),
64 (19, 19, 19, '2025-04-07', 'Tratamiento de la piel'),
65 (20, 20, 20, '2025-04-08', 'Consulta pediátrica');
66
67 INSERT INTO Tratamiento (id_tratamiento, id_cita, descripcion, duracion_dias) VALUES
68 (1, 1, 'Recomendación de ejercicio y dieta', 30),
69 (2, 2, 'Vacunas necesarias', 7),
70 (3, 3, 'Crema para dermatitis', 14),
71 (4, 4, 'Medicación para migraña', 21),
72 (5, 5, 'Limpieza dental', 3),
73 (6, 6, 'Revisión de anticonceptivos', 30),
74 (7, 7, 'Medicación para fiebre', 7),
75 (8, 8, 'Yeso para fractura', 21),
76 (9, 9, 'Gotas para los ojos', 7),
77 (10, 10, 'Tratamiento de quimioterapia', 30),
78 (11, 11, 'Vacunas para niños', 5),
79 (12, 12, 'Rehabilitación post-operatoria', 60),
80 (13, 13, 'Medicamentos para infección urinaria', 14),
81 (14, 14, 'Medicamentos para gastritis', 30),
82 (15, 15, 'Medicamentos para hipertensión', 30),
83 (16, 16, 'Sesiones de terapia', 20),
```

```

1  ✓ INSERT INTO Paciente (id_paciente, nombre, edad, direccion) VALUES
2  (1, 'Juan Pérez', 45, 'Calle Ficticia 123'),
3  (2, 'María Gómez', 32, 'Avenida Libertad 456'),
4  (3, 'Carlos Ruiz', 29, 'Calle Sol 789'),
5  (4, 'Ana Martínez', 38, 'Calle del Lago 101'),
6  (5, 'Pedro González', 50, 'Calle de la Paz 202'),
7  (6, 'Luis Fernández', 60, 'Calle Real 303'),
8  (7, 'Clara Sánchez', 25, 'Avenida Central 404'),
9  (8, 'Ricardo López', 33, 'Calle del Sol 505'),
10 (9, 'Elena Díaz', 40, 'Calle del Río 606'),
11 (10, 'Javier Morales', 52, 'Calle Nueva 707'),
12 (11, 'Patricia Romero', 36, 'Calle de la Luna 808'),
13 (12, 'Miguel Álvarez', 41, 'Calle Mayor 909'),
14 (13, 'Laura Martínez', 27, 'Avenida 15 1010'),
15 (14, 'Santiago Pérez', 30, 'Calle del Mar 1111'),
16 (15, 'Raúl Sánchez', 22, 'Calle del Viento 1212'),
17 (16, 'Carla López', 55, 'Avenida del Este 1313'),
18 (17, 'David Rodríguez', 43, 'Calle Santa Fe 1414'),
19 (18, 'Marta Herrera', 28, 'Calle del Sol 1515'),
20 (19, 'Juan José Ruiz', 31, 'Calle del Centro 1616'),
21 (20, 'Adriana González', 39, 'Calle Alta 1717');
22
23 ✓ INSERT INTO Medico (id_medico, nombre, especialidad, tel_contacto) VALUES
24 (1, 'Dr. Roberto Sánchez', 'Cardiología', '555-1234'),
25 (2, 'Dra. Laura Fernández', 'Pediatría', '555-5678'),
26 (3, 'Dr. Javier López', 'Dermatología', '555-8765'),
27 (4, 'Dr. Pedro Gutiérrez', 'Neurología', '555-2345'),
28 (5, 'Dra. Marta Rodríguez', 'Odontología', '555-3456'),
29 (6, 'Dr. Hugo Pérez', 'Ginecología', '555-4567'),
30 (7, 'Dra. Isabel Gómez', 'Medicina General', '555-5679'),
31 (8, 'Dr. Luis Fernández', 'Traumatología', '555-6780'),
32 (9, 'Dra. Sonia Martínez', 'Oftalmología', '555-7891'),
33 (10, 'Dr. Felipe Álvarez', 'Oncología', '555-8902'),
34 (11, 'Dra. Carmen Hernández', 'Pediatría', '555-9013'),
35 (12, 'Dr. Andrés Rodríguez', 'Cirugía', '555-1123'),
36 (13, 'Dr. Mario López', 'Urología', '555-2234'),
37 (14, 'Dra. Silvia Díaz', 'Gastroenterología', '555-3345'),
38 (15, 'Dr. Víctor Sánchez', 'Cardiología', '555-4456'),
39 (16, 'Dra. Paula Ramírez', 'Psicología', '555-5567'),
40 (17, 'Dr. Gabriel Martínez', 'Cirugía General', '555-6678'),
41 (18, 'Dra. Teresa Jiménez', 'Reumatología', '555-7789'),
42 (19, 'Dr. Eduardo García', 'Dermatología', '555-8890'),

```

## Consultas SQL (LMD)

### 1. Consulta para insertar un nuevo paciente:

Esta consulta se utiliza para insertar un nuevo paciente en la tabla Paciente.

```

INSERT INTO Paciente (id_paciente, nombre, edad, direccion)
VALUES (21, 'Carlos Martínez', 35, 'Calle Ficticia 123');

```

---

## 2. Consulta para consultar las citas de un paciente específico:

Esta consulta selecciona las citas de un paciente específico usando el WHERE para filtrar por

The screenshot shows a SQL IDE interface with a query editor and a data output table. The query editor contains the following SQL code:

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

The data output table displays the results of the query:

	id_cita integer	fecha date	motivo text	medico character varying (100)
1	1	2025-03-20	Chequeo general	Dr. Roberto Sánchez

---

## 3. Consulta para ordenar las citas de un paciente por fecha (ORDER BY):

Esta consulta muestra las citas de un paciente específico, ordenadas por fecha (más recientes primero).

The screenshot shows a SQL IDE interface with a query editor and a data output table. The query editor contains the following SQL code:

```
1 SELECT Cita.id_cita, Cita.fecha, Cita.motivo, Medico.nombre AS medico
2 FROM Cita
3 JOIN Medico ON Cita.id_medico = Medico.id_medico
4 WHERE Cita.id_paciente = 1
5 ORDER BY Cita.fecha DESC;
6
```

The data output table displays the results of the query:

	id_cita integer	fecha date	motivo text	medico character varying (100)
1	1	2025-03-20	Chequeo general	Dr. Roberto Sánchez

#### 4. Consulta para obtener el número de citas por médico (GROUP BY):

Esta consulta muestra el número de citas que tiene cada médico, agrupadas por el nombre del médico.

Query

Query History

1

2

3

4

5

```
SELECT Medico.nombre, COUNT(Cita.id_cita) AS total_citas
FROM Cita
JOIN Medico ON Cita.id_medico = Medico.id_medico
GROUP BY Medico.nombre;
```

Data Output

Messages

Notifications

≡+

▼

▼

SQL

	nombre character varying (100)	total_citas bigint
1	Dra. Paula Ramírez	1
2	Dr. Eduardo García	1
3	Dra. Carmen Hernández	1
4	Dr. Pedro Gutiérrez	1
5	Dra. Sonia Martínez	1
6	Dr. Víctor Sánchez	1
7	Dr. Andrés Rodríguez	1
8	Dra. Isabel Gómez	1
9	Dra. Marta Rodríguez	1
10	Dr. Hugo Pérez	1
11	Dra. Laura Fernández	1
12	Dra. Julia Pérez	1
13	Dr. Luis Fernández	1
14	Dra. Silvia Díaz	1
15	Dr. Mario López	1
16	Dr. Javier López	1
17	Dr. Roberto Sánchez	1
18	Dr. Felipe Álvarez	1
19	Dr. Gabriel Martínez	1
20	Dra. Teresa Jiménez	1

---

## 5. Consulta para obtener los tratamientos de una cita específica:

Esta consulta obtiene los tratamientos de una cita específica utilizando el WHERE para filtrar por id\_cita.

Query		Query History	
1	SELECT	Tratamiento.descripcion,	Tratamiento.duracion_dias
2	FROM	Tratamiento	
3	WHERE	Tratamiento.id_cita =	1;
4			
Data Output		Messages	
		Notifications	
	descripcion		duracion_dias
	text		integer
1	Recomendación de ejercicio y dieta		30

---

## 6. Consulta para obtener todos los pacientes atendidos por un médico:

Esta consulta usa JOIN para obtener los pacientes atendidos por un médico específico.

Query

Query History

1

2

3

4

5

SELECT

Paciente.nombre,

Cita.fecha,

Cita.motivo

FROM

Cita

JOIN

Paciente

ON

Cita.id\_paciente =

Paciente.id\_paciente

WHERE

Cita.id\_medico =

1;

Data Output

Messages

Notifications

SQL

	<div>nombre</div> <div>character varying (100)</div> <div></div>	<div>fecha</div> <div>date</div> <div></div>	<div>motivo</div> <div>text</div> <div></div>
1	Juan Pérez	2025-03-20	Chequeo general



## 7. Consulta para obtener las citas y tratamientos de un paciente específico:

Esta consulta utiliza JOIN para combinar información de citas y tratamientos de un paciente específico.

Query		Query History	
1	SELECT	Cita.id_cita, Cita.fecha, Cita.motivo, Tratamiento.descripcion, Tratamiento.duracion_dias	
2	FROM	Cita	
3	JOIN	Tratamiento ON Cita.id_cita = Tratamiento.id_cita	
4	WHERE	Cita.id_paciente = 2;	
5			

Data Output		Messages		Notifications	
≡	+	📄	▼	🗑️	📄
📄	▼	🗑️	📄	📄	SQL
	id_cita	fecha	motivo	descripcion	duracion_dias
	integer	date	text	text	integer
1	2	2025-03-21	Revisión anual	Vacunas necesarias	7

## 8. Consulta para buscar un paciente por nombre (LIKE):

Esta consulta utiliza LIKE para buscar pacientes cuyo nombre coincida parcialmente con "Juan".

Query		Query History	
1	SELECT	id_paciente, nombre, edad, direccion	
2	FROM	Paciente	
3	WHERE	nombre LIKE '%Juan%';	
4			

Data Output		Messages		Notifications	
≡	+	📄	▼	🗑️	📄
📄	▼	🗑️	📄	📄	SQL
	id_paciente	nombre	edad	direccion	
	[PK] integer	character varying (100)	integer	text	
1	1	Juan Pérez	45	Calle Ficticia 123	
2	19	Juan José Ruiz	31	Calle del Centro 1616	

# Tienda en Línea

## Introducción

### Descripción del problema:

El sistema de gestión de pedidos de una tienda en línea busca organizar y administrar la información relacionada con los clientes, productos, pedidos y detalles de cada pedido. Este sistema tiene como objetivo optimizar la gestión de los productos adquiridos por los clientes, facilitando el acceso a la información relevante para los procesos de ventas y atención al cliente.

### Objetivo:

El objetivo principal es crear una base de datos que permita gestionar la información de clientes, productos, pedidos y detalles de los pedidos. Además, se pretende realizar consultas eficientes para obtener datos clave como los productos comprados por un cliente específico, entre otros.

---

## Análisis de Requerimientos

### Entidades clave:

- **Cliente:** Representa a los clientes que realizan pedidos en línea.
- **Producto:** Representa los productos que están disponibles para la venta.
- **Pedido:** Representa los pedidos realizados por los clientes.
- **DetallePedido:** Especifica los productos que forman parte de un pedido, junto con la cantidad y el precio.

### Atributos de las entidades:

- **Cliente:** id\_cliente, nombre, direccion, telefono.
- **Producto:** id\_producto, nombre, precio.
- **Pedido:** id\_pedido, id\_cliente, fecha, total.
- **DetallePedido:** id\_detalle, id\_pedido, id\_producto, cantidad, precio.

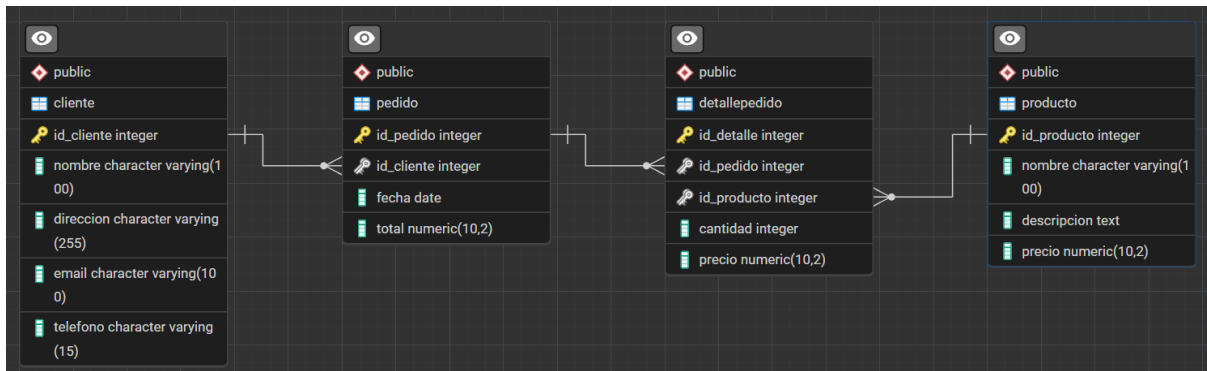
### Relaciones entre las entidades:

- Un cliente puede realizar varios pedidos, pero un pedido pertenece a un solo cliente.
  - Un pedido puede incluir varios productos, y un producto puede estar en varios pedidos.
  - Un pedido tiene uno o más detalles de pedido, y cada detalle está asociado a un solo producto.
- 

## Modelo Entidad-Relación (E-R)

A continuación, se presenta el diagrama E-R que refleja las relaciones entre las entidades clave del sistema.

### Diagrama E-R:



### Explicación:

El diagrama E-R muestra cómo se relacionan las entidades Cliente, Producto, Pedido y DetallePedido. Las relaciones están indicadas mediante líneas entre las entidades, y las claves foráneas garantizan la integridad referencial entre las tablas.

## Conversión a Esquema Relacional

### Descripción del esquema relacional:

El modelo E-R se convierte en un conjunto de tablas que representan las entidades y sus relaciones. Cada tabla tiene claves primarias y foráneas para garantizar la integridad referencial.

### Tablas:

- **Tabla Cliente:**  
Clave primaria: `id_cliente`
- **Tabla Producto:**  
Clave primaria: `id_producto`
- **Tabla Pedido:**  
Claves primarias: `id_pedido`  
Claves foráneas: `id_cliente` (referencia a Cliente)
- **Tabla DetallePedido:**  
Clave primaria: `id_detalle`  
Claves foráneas: `id_pedido` (referencia a Pedido), `id_producto` (referencia a Producto)

## Implementación en DBMS (LDD)

A continuación, se presentan las sentencias SQL para crear las tablas necesarias para el sistema:

Query	Query History
1	-- Crear la tabla Cliente
2	CREATE TABLE Cliente (
3	id_cliente INT PRIMARY KEY,
4	nombre VARCHAR(100),
5	direccion VARCHAR(255),
6	email VARCHAR(100),
7	telefono VARCHAR(15)
8	);
9	
10	-- Crear la tabla Producto
11	CREATE TABLE Producto (
12	id_producto INT PRIMARY KEY,
13	nombre VARCHAR(100),
14	descripcion TEXT,
15	precio DECIMAL(10, 2)
16	);
17	
18	-- Crear la tabla Pedido
19	CREATE TABLE Pedido (
20	id_pedido INT PRIMARY KEY,
21	id_cliente INT,
22	fecha DATE,
23	total DECIMAL(10, 2),
24	FOREIGN KEY (id_cliente) REFERENCES Cliente(id_cliente)
25	);
26	
27	-- Crear la tabla DetallePedido
28	CREATE TABLE DetallePedido (
29	id_detalle INT PRIMARY KEY,
30	id_pedido INT,
31	id_producto INT,
32	cantidad INT,
33	precio DECIMAL(10, 2),
34	FOREIGN KEY (id_pedido) REFERENCES Pedido(id_pedido),
35	FOREIGN KEY (id_producto) REFERENCES Producto(id_producto)
36	);
37	

---

## Manipulación de Datos (LMD)

A continuación, se presentan las consultas SQL utilizadas para manipular los datos y realizar las consultas pertinentes.

```
Query  Query History
1  -- Insertar datos en la tabla Cliente
2  ✓ INSERT INTO Cliente (id_cliente, nombre, direccion, email, telefono)
3  VALUES
4  (1, 'Juan Pérez', 'Av. Ficticia 123', 'juan.perez@example.com', '5551234567'),
5  (2, 'María López', 'Calle Real 456', 'maria.lopez@example.com', '5552345678'),
6  (3, 'Carlos García', 'Av. Libertad 789', 'carlos.garcia@example.com', '5553456789'),
7  (4, 'Ana Rodríguez', 'Calle Larga 101', 'ana.rodriguez@example.com', '5554567890'),
8  (5, 'Luis Martínez', 'Calle Centro 202', 'luis.martinez@example.com', '5555678901'),
9  (6, 'Pedro Sánchez', 'Calle Norte 303', 'pedro.sanchez@example.com', '5556789012'),
10 (7, 'Laura Díaz', 'Calle Oeste 404', 'laura.diaz@example.com', '5557890123'),
11 (8, 'Javier Fernández', 'Calle Este 505', 'javier.fernandez@example.com', '5558901234'),
12 (9, 'Marta Ruiz', 'Calle Sol 606', 'marta.ruiz@example.com', '5559012345'),
13 (10, 'Ricardo Gómez', 'Calle Luna 707', 'ricardo.gomez@example.com', '5550123456'),
14 (11, 'Sofía Pérez', 'Calle Estrella 808', 'sofia.perez@example.com', '5551234589'),
15 (12, 'Daniel González', 'Calle Fuego 909', 'daniel.gonzalez@example.com', '5552345690'),
16 (13, 'Patricia Jiménez', 'Calle Mar 1010', 'patricia.jimenez@example.com', '5553456791'),
17 (14, 'Alberto Martínez', 'Calle Flor 1111', 'alberto.martinez@example.com', '5554567892'),
18 (15, 'Victoria Moreno', 'Calle Agua 1212', 'victoria.moreno@example.com', '5555678903'),
19 (16, 'Francisco Ruiz', 'Calle Viento 1313', 'francisco.ruiz@example.com', '5556789014'),
20 (17, 'Elena Fernández', 'Calle Tierra 1414', 'elena.fernandez@example.com', '5557890125'),
21 (18, 'Óscar Hernández', 'Calle Nieve 1515', 'oscar.hernandez@example.com', '5558901236'),
22 (19, 'Isabel Castro', 'Calle Sol 1616', 'isabel.castro@example.com', '5559012347'),
23 (20, 'Gustavo Sánchez', 'Calle Luna 1717', 'gustavo.sanchez@example.com', '5550123458');
24
25 -- Insertar datos en la tabla Producto
26 ✓ INSERT INTO Producto (id_producto, nombre, descripcion, precio)
27 VALUES
28 (1, 'Laptop HP', 'Laptop HP de 15 pulgadas, 8GB RAM', 1200.00),
29 (2, 'Smartphone Samsung', 'Smartphone Samsung Galaxy S21', 800.00),
30 (3, 'Tablet Apple', 'Tablet Apple iPad Pro', 900.00),
31 (4, 'Audífonos Sony', 'Audífonos inalámbricos Sony WH-1000XM4', 350.00),
32 (5, 'Reloj Garmin', 'Reloj deportivo Garmin Forerunner 245', 250.00),
33 (6, 'Teclado Mecánico', 'Teclado mecánico RGB con switches Cherry', 100.00),
34 (7, 'Mouse Logitech', 'Mouse inalámbrico Logitech MX Master 3', 80.00),
35 (8, 'Cámara Canon', 'Cámara reflex Canon EOS 80D', 900.00),
36 (9, 'Monitor LG', 'Monitor LG UltraWide 34 pulgadas', 450.00),
37 (10, 'Altavoces Bose', 'Altavoces inalámbricos Bose SoundLink Revolve', 200.00),
38 (11, 'Cargador Anker', 'Cargador rápido Anker PowerPort 3', 30.00),
```

```
Query  Query History
49  -- Insertar datos en la tabla Pedido
50  INSERT INTO Pedido (id_pedido, id_cliente, fecha, total)
51  VALUES
52  (1, 1, '2023-01-10', 1200.00),
53  (2, 2, '2023-01-12', 800.00),
54  (3, 3, '2023-01-14', 900.00),
55  (4, 4, '2023-01-15', 350.00),
56  (5, 5, '2023-01-16', 250.00),
57  (6, 6, '2023-01-17', 100.00),
58  (7, 7, '2023-01-18', 80.00),
59  (8, 8, '2023-01-19', 900.00),
60  (9, 9, '2023-01-20', 450.00),
61  (10, 10, '2023-01-21', 200.00),
62  (11, 11, '2023-01-22', 30.00),
63  (12, 12, '2023-01-23', 120.00),
64  (13, 13, '2023-01-24', 70.00),
65  (14, 14, '2023-01-25', 25.00),
66  (15, 15, '2023-01-26', 50.00),
67  (16, 16, '2023-01-27', 250.00),
68  (17, 17, '2023-01-28', 40.00),
69  (18, 18, '2023-01-29', 15.00),
70  (19, 19, '2023-01-30', 35.00),
71  (20, 20, '2023-01-31', 60.00);
72
73  -- Insertar datos en la tabla DetallePedido
74  INSERT INTO DetallePedido (id_detalle, id_pedido, id_producto, cantidad, precio)
75  VALUES
76  (1, 1, 1, 1, 1200.00),
77  (2, 2, 2, 1, 800.00),
78  (3, 3, 3, 1, 900.00),
79  (4, 4, 4, 1, 350.00),
80  (5, 5, 5, 1, 250.00),
81  (6, 6, 6, 1, 100.00),
82  (7, 7, 7, 1, 80.00),
83  (8, 8, 8, 1, 900.00),
84  (9, 9, 9, 1, 450.00),
85  (10, 10, 10, 1, 200.00),
86  (11, 11, 11, 1, 30.00),
```

## Consultas SQL (LMD)

### 1. INSERT INTO: Insertar un nuevo pedido

Esta consulta inserta un nuevo pedido en la tabla Pedido.

```
Query  Query History
1  INSERT INTO Pedido (id_pedido, id_cliente, fecha, total)
2  VALUES (21, 1, '2023-02-01', 1500.00);
3
```

**Explicación:** Esta consulta agrega un nuevo pedido a la tabla Pedido. En este caso, se está agregando un pedido con ID 21 para el cliente con ID 1 realizado el 2023-02-01 con un total de 1500.00.

---

## 2. SELECT con WHERE: Consultar pedidos de un cliente específico

Query		Query History		
1	SELECT *			
2	FROM Pedido			
3	WHERE id_cliente = 1;			
4				
Data Output		Messages		Notifications
≡+	📄	▼	📋	▼
🗑	📦	📥	📤	SQL
	id_pedido [PK] integer	id_cliente integer	fecha date	total numeric (10,2)
1	1	1	2023-01-10	1200.00

**Explicación:** Esta consulta selecciona todos los pedidos de un cliente específico, en este caso, el cliente con ID 1. El filtro se hace utilizando la cláusula WHERE.

---

### 3. ORDER BY: Consultar productos ordenados por precio

Query		Query History	
1	SELECT	nombre,	precio
2	FROM	Producto	
3	ORDER BY	precio	DESC;
4			
Data Output		Messages	
Notifications			
	nombre character varying (100)	precio numeric (10,2)	
1	Laptop HP	1200.00	
2	Cámara Canon	900.00	
3	Tablet Apple	900.00	
4	Smartphone Samsung	800.00	
5	Monitor LG	450.00	
6	Audífonos Sony	350.00	
7	Reloj Garmin	250.00	
8	Silla Gaming	250.00	
9	Altavoces Bose	200.00	
10	Bocina JBL	120.00	
11	Teclado Mecánico	100.00	
12	Mouse Logitech	80.00	
13	Impresora HP	70.00	
14	Kit de Herramientas	60.00	
15	Disco Duro Externo	50.00	
16	Papelera Inteligente	40.00	
17	Cargador Inalámbrico	35.00	
18	Cargador Anker	30.00	
19	Cargador Samsung	25.00	
20	Funda de Teléfono	15.00	

**Explicación:** Esta consulta muestra una lista de productos ordenados de mayor a menor según su precio, utilizando la cláusula ORDER BY. En este caso, DESC indica que el orden es descendente (de mayor a menor).

---



#### 4. GROUP BY: Consultar la cantidad de productos vendidos por cada pedido

Query		Query History	
1	SELECT	id_pedido,	SUM(cantidad) AS total_productos
2	FROM	DetallePedido	
3	GROUP BY	id_pedido;	
4			
Data Output		Messages	
Data Output		Messages	
	id_pedido integer		total_productos bigint
1	11		1
2	9		1
3	15		1
4	19		1
5	3		1
6	17		1
7	5		1
8	4		1
9	10		1
10	6		1
11	14		1
12	13		1
13	2		1
14	16		1
15	7		1
16	12		1
17	20		1
18	1		1
19	18		1
20	8		1

**Explicación:** Esta consulta calcula la cantidad total de productos vendidos por cada pedido. La función SUM(cantidad) agrupa los productos por el ID del pedido (id\_pedido) y muestra el total de productos vendidos por cada pedido.

---

## 5. JOIN: Consultar los productos y sus detalles por pedido

Query

Query History

```

1 SELECT Pedido.id_pedido, Producto.nombre, DetallePedido.cantidad, DetallePedido.precio
2 FROM DetallePedido
3 JOIN Producto ON DetallePedido.id_producto = Producto.id_producto
4 JOIN Pedido ON DetallePedido.id_pedido = Pedido.id_pedido;
5

```

Data Output

Messages

Notifications

≡+

▼

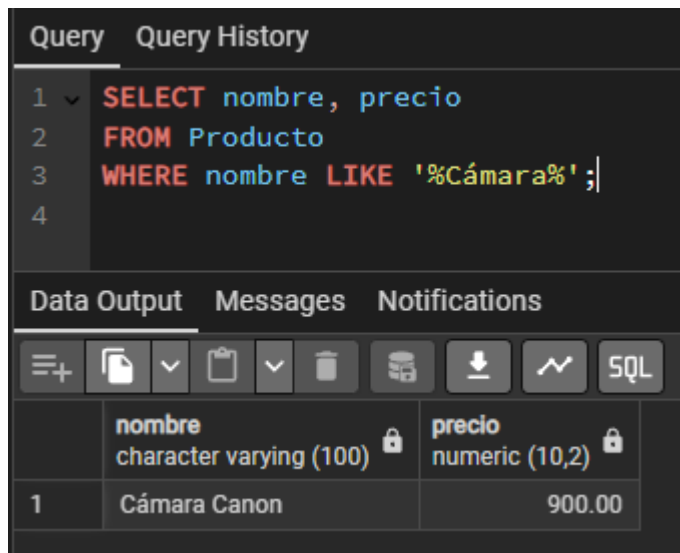
▼

SQL

	id_pedido integer	nombre character varying (100)	cantidad integer	precio numeric (10,2)
1	1	Laptop HP	1	1200.00
2	2	Smartphone Samsung	1	800.00
3	3	Tablet Apple	1	900.00
4	4	Audífonos Sony	1	350.00
5	5	Reloj Garmin	1	250.00
6	6	Teclado Mecánico	1	100.00
7	7	Mouse Logitech	1	80.00
8	8	Cámara Canon	1	900.00
9	9	Monitor LG	1	450.00
10	10	Altavoces Bose	1	200.00
11	11	Cargador Anker	1	30.00
12	12	Bocina JBL	1	120.00
13	13	Impresora HP	1	70.00
14	14	Cargador Samsung	1	25.00
15	15	Disco Duro Externo	1	50.00
16	16	Silla Gaming	1	250.00
17	17	Papelera Inteligente	1	40.00
18	18	Funda de Teléfono	1	15.00
19	19	Cargador Inalámbrico	1	35.00
20	20	Kit de Herramientas	1	60.00

**Explicación:** Esta consulta utiliza JOIN para combinar las tablas DetallePedido, Producto y Pedido. Se obtiene la información del pedido, el nombre del producto, la cantidad comprada y el precio del producto. Se enlazan las tablas mediante las claves foráneas id\_producto y id\_pedido.

## 6. LIKE: Buscar productos que contengan una palabra clave



**Explicación:** Esta consulta busca productos cuyo nombre contiene la palabra "Cámara". El operador LIKE se utiliza con el símbolo %, que representa cualquier secuencia de caracteres antes o después de la palabra "Cámara".

---

## Biblioteca digital

### Introducción

**Descripción del problema:** El sistema de gestión de una biblioteca digital busca organizar y administrar la información sobre los usuarios, libros y préstamos. Este sistema facilitará el acceso rápido a los datos y mejorará la eficiencia en la administración de los préstamos.

**Objetivo:** El objetivo principal es crear una base de datos que permita gestionar la información de usuarios, libros y préstamos. Además, se pretende realizar consultas eficientes para obtener datos clave como los libros prestados a un usuario específico, entre otros.

---

### Análisis de Requerimientos

#### Entidades clave:

- **Usuario:** Representa a los usuarios registrados en la biblioteca digital.
- **Libro:** Representa los libros disponibles para préstamo.
- **Préstamo:** Representa los registros de préstamos realizados por los usuarios.

#### Atributos de las entidades:

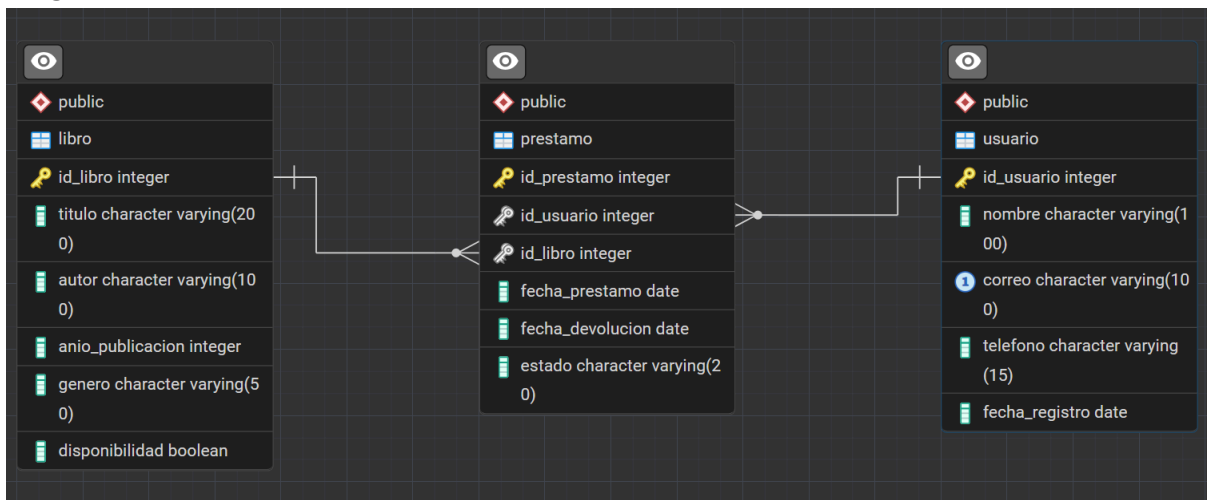
- **Usuario:** id\_usuario, nombre, correo, telefono, fecha\_registro.
- **Libro:** id\_libro, titulo, autor, anio\_publicacion, genero.
- **Préstamo:** id\_prestamo, id\_usuario, id\_libro, fecha\_prestamo, fecha\_devolucion.

#### Relaciones entre las entidades:

- Un usuario puede realizar varios préstamos, pero un préstamo pertenece a un solo usuario.
- Un libro puede ser prestado varias veces a diferentes usuarios.

**Modelo Entidad-Relación (E-R)** A continuación, se presenta el diagrama E-R que refleja las relaciones entre las entidades clave del sistema.

#### Diagrama E-R:



**Explicación:** El diagrama E-R muestra cómo se relacionan las entidades **Usuario**, **Libro** y **Préstamo**. Las relaciones están indicadas mediante líneas entre las entidades, y las claves foráneas garantizan la integridad referencial entre las tablas.

#### Conversión a Esquema Relacional

**Descripción del esquema relacional:** El modelo E-R se convierte en un conjunto de tablas que representan las entidades y sus relaciones. Cada tabla tiene claves primarias y foráneas para garantizar la integridad referencial.

#### Tablas:

- **Tabla Usuario:** Clave primaria: id\_usuario.
- **Tabla Libro:** Clave primaria: id\_libro.
- **Tabla Préstamo:**
  - Clave primaria: id\_prestamo.

- Claves foráneas: id\_usuario (referencia a Usuario), id\_libro (referencia a Libro).

---

**Implementación en DBMS (LDD)** A continuación, se presentan las sentencias SQL para crear las tablas necesarias para el sistema:

```
-- Tabla Usuario
CREATE TABLE Usuario (
    id_usuario INT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    correo VARCHAR(100) UNIQUE NOT NULL,
    telefono VARCHAR(15),
    fecha_registro DATE NOT NULL
);
```

```
-- Tabla Libro
CREATE TABLE Libro (
    id_libro INT PRIMARY KEY,
    titulo VARCHAR(200) NOT NULL,
    autor VARCHAR(100) NOT NULL,
    anio_publicacion INT,
    genero VARCHAR(50),
    disponibilidad BOOLEAN DEFAULT TRUE
);
```

```
-- Tabla Prestamo
CREATE TABLE Prestamo (
    id_prestamo INT PRIMARY KEY,
    id_usuario INT,
    id_libro INT,
    fecha_prestamo DATE NOT NULL,
    fecha_devolucion DATE,
    estado VARCHAR(20) CHECK (estado IN ('Prestado', 'Devuelto', 'Retrasado')),
    FOREIGN KEY (id_usuario) REFERENCES Usuario(id_usuario) ON DELETE CASCADE,
    FOREIGN KEY (id_libro) REFERENCES Libro(id_libro) ON DELETE CASCADE
);
```

---

## Manipulación de Datos (LMD)

A continuación, se presentan las consultas SQL utilizadas para manipular los datos y realizar las consultas pertinentes.

```
-- Insertar datos en la tabla Libro
INSERT INTO Libro (id_libro, titulo, autor, anio_publicacion, genero, disponibilidad) VALUES
(1, 'Cien años de soledad', 'Gabriel García Márquez', 1967, 'Ficción', TRUE),
(2, 'Don Quijote de la Mancha', 'Miguel de Cervantes', 1605, 'Ficción', TRUE),
(3, 'El amor en los tiempos del cólera', 'Gabriel García Márquez', 1985, 'Romántica', TRUE),
(4, '1984', 'George Orwell', 1949, 'Distopía', TRUE),
(5, 'El principito', 'Antoine de Saint-Exupéry', 1943, 'Infantil', TRUE),
(6, 'Orgullo y prejuicio', 'Jane Austen', 1813, 'Romántica', TRUE),
(7, 'Crimen y castigo', 'Fiódor Dostoyevski', 1866, 'Ficción', TRUE),
(8, 'La casa de los espíritus', 'Isabel Allende', 1982, 'Ficción', TRUE),
(9, 'Los pilares de la tierra', 'Ken Follett', 1989, 'Histórica', TRUE),
(10, 'La sombra del viento', 'Carlos Ruiz Zafón', 2001, 'Misterio', TRUE),
(11, 'Matar a un ruiseñor', 'Harper Lee', 1960, 'Ficción', TRUE),
(12, 'La Odisea', 'Homero', -800, 'Clásico', TRUE),
(13, 'La guerra y la paz', 'León Tolstói', 1869, 'Histórica', TRUE),
(14, 'Cumbres borrascosas', 'Emily Brontë', 1847, 'Romántica', TRUE),
(15, 'La chica del tren', 'Paula Hawkins', 2015, 'Misterio', TRUE),
(16, 'Harry Potter y la piedra filosofal', 'J.K. Rowling', 1997, 'Fantasía', TRUE),
(17, 'El código Da Vinci', 'Dan Brown', 2003, 'Misterio', TRUE),
(18, 'Fahrenheit 451', 'Ray Bradbury', 1953, 'Ciencia ficción', TRUE),
(19, 'El alquimista', 'Paulo Coelho', 1988, 'Ficción', TRUE),
(20, 'La catedral del mar', 'Ildefonso Falcones', 2006, 'Histórica', TRUE);
```

```
-- Insertar datos en la tabla Prestamo
INSERT INTO Prestamo (id_prestamo, id_usuario, id_libro, fecha_prestamo, fecha_devolucion, estado) VALUES
(1, 1, 1, '2024-02-01', '2024-02-15', 'Devuelto'),
(2, 2, 2, '2024-02-02', '2024-02-16', 'Devuelto'),
(3, 3, 3, '2024-02-03', '2024-02-17', 'Devuelto'),
(4, 4, 4, '2024-02-04', '2024-02-18', 'Devuelto'),
(5, 5, 5, '2024-02-05', '2024-02-19', 'Devuelto'),
(6, 6, 6, '2024-02-06', '2024-02-20', 'Devuelto'),
(7, 7, 7, '2024-02-07', '2024-02-21', 'Devuelto'),
(8, 8, 8, '2024-02-08', '2024-02-22', 'Devuelto'),
(9, 9, 9, '2024-02-09', '2024-02-23', 'Devuelto'),
(10, 10, 10, '2024-02-10', '2024-02-24', 'Devuelto'),
(11, 11, 11, '2024-02-11', '2024-02-25', 'Devuelto'),
(12, 12, 12, '2024-02-12', '2024-02-26', 'Devuelto'),
(13, 13, 13, '2024-02-13', '2024-02-27', 'Devuelto'),
(14, 14, 14, '2024-02-14', '2024-02-28', 'Devuelto'),
(15, 15, 15, '2024-02-15', '2024-03-01', 'Devuelto'),
(16, 16, 16, '2024-02-16', '2024-03-02', 'Devuelto'),
(17, 17, 17, '2024-02-17', '2024-03-03', 'Devuelto'),
(18, 18, 18, '2024-02-18', '2024-03-04', 'Devuelto'),
(19, 19, 19, '2024-02-19', '2024-03-05', 'Devuelto'),
(20, 20, 20, '2024-02-20', '2024-03-06', 'Devuelto');
```

---

## Consultas SQL (LMD)

### 1. INSERT INTO: Insertar un nuevo usuario

Esta consulta agrega un nuevo usuario a la base de datos.

QueryQuery History

```
1  INSERT INTO Usuario (id_usuario, nombre, correo, telefono, fecha_registro)
2  VALUES (21, 'Andrea Castillo', 'andreacastillo@email.com', '1112223334', '2024-03-10');
3  |
```

## 2. SELECT con WHERE: Consultar préstamos de un usuario específico

Se obtiene la lista de préstamos realizados por un usuario con `id_usuario = 5`.

QueryQuery History

```
1  SELECT * FROM Prestamo
2  WHERE id_usuario = 5;
```

Data OutputMessagesNotifications

≡+

Showing rows: 1 to 1

Pa

	id_prestamo [PK] integer	id_usuario integer	id_libro integer	fecha_prestamo date	fecha_devolucion date	estado character varying (20)
1	5	5	5	2024-02-05	2024-02-19	Devuelto

## 3. ORDER BY: Ordenar libros por año de publicación

Muestra todos los libros de la biblioteca ordenados por año de publicación en orden descendente.

Query

Query History

Scratch Pad

1

2

3

SELECT \* FROM Libro

ORDER BY anio\_publicacion DESC;

Data Output

Messages

Notifications

Showing rows: 1 to 20

Page No: 1











of

	id_libro [PK] integer	titulo character varying (200)	autor character varying (100)	anio_publicacion integer	genero character varying (50)	disponibilidad boolean
1	15	La chica del tren	Paula Hawkins	2015	Misterio	true
2	20	La catedral del mar	Ildefonso Falcones	2006	Histórica	true
3	17	El código Da Vinci	Dan Brown	2003	Misterio	true
4	10	La sombra del viento	Carlos Ruiz Zafón	2001	Misterio	true
5	16	Harry Potter y la piedra filosofal	J.K. Rowling	1997	Fantasía	true
6	9	Los pilares de la tierra	Ken Follett	1989	Histórica	true
7	19	El alquimista	Paulo Coelho	1988	Ficción	true
8	3	El amor en los tiempos del cól...	Gabriel García Márquez	1985	Romántica	true
9	8	La casa de los espíritus	Isabel Allende	1982	Ficción	true
10	1	Cien años de soledad	Gabriel García Márquez	1967	Ficción	true
11	11	Matar a un ruiseñor	Harper Lee	1960	Ficción	true
12	18	Fahrenheit 451	Ray Bradbury	1953	Ciencia ficción	true

#### 4. GROUP BY: Contar la cantidad de préstamos por usuario

Se agrupan los préstamos por usuario y se cuenta cuántos préstamos ha realizado cada uno.



Query		Query History	
1	SELECT	id_usuario, COUNT(*) AS total_prestamos	
2	FROM	Prestamo	
3	GROUP BY	id_usuario;	
4			
Data Output		Messages	Notifications
			
			
			
	id_usuario integer	total_prestamos bigint	
1	11	1	
2	9	1	
3	15	1	
4	19	1	
5	3	1	
6	17	1	
7	5	1	
8	4	1	
9	10	1	
10	6	1	
11	14	1	

## 5. JOIN: Obtener los datos del usuario, el libro prestado y la fecha del préstamo

Se combinan las tablas Prestamo, Usuario y Libro para mostrar quién prestó qué libro y cuándo.

QueryQuery History

1

2

3

4

5

```
SELECT Usuario.nombre AS Nombre_Usuario, Libro.titulo AS Libro_Prestado, Prestamo.  
FROM Prestamo  
JOIN Usuario ON Prestamo.id_usuario = Usuario.id_usuario  
JOIN Libro ON Prestamo.id_libro = Libro.id_libro;
```

Data OutputMessagesNotifications

SQL

Showing rows: 1 to 20

Page No

	nombre_usuario character varying (100)	libro_prestado character varying (200)	fecha_prestamo date
1	Juan Pérez	Cien años de soledad	2024-02-01
2	María García	Don Quijote de la Mancha	2024-02-02
3	Carlos López	El amor en los tiempos del cól...	2024-02-03
4	Ana Torres	1984	2024-02-04
5	Luis Rodríguez	El principito	2024-02-05
6	Patricia Sánchez	Orgullo y prejuicio	2024-02-06
7	José Martínez	Crimen y castigo	2024-02-07
8	Laura Pérez	La casa de los espíritus	2024-02-08
9	Miguel Fernández	Los pilares de la tierra	2024-02-09
10	Sofía Díaz	La sombra del viento	2024-02-10

## 6. LIKE: Buscar libros con la palabra "amor" en el título

Busca libros cuyo título contenga la palabra "amor", sin importar si está al inicio, en medio o al final.

QueryQuery HistoryScratch Pad X

1

2

3

```
SELECT * FROM Libro  
WHERE titulo LIKE '%amor%';
```

Data OutputMessagesNotifications

SQL

Showing rows: 1 to 1

Page No: 1 of 1

	id_libro [PK] integer	titulo character varying (200)	autor character varying (100)	anio_publicacion integer	genero character varying (50)	disponibilidad boolean
1	3	El amor en los tiempos del cólera	Gabriel García Márquez	1985	Romántica	true

# Sistema de recursos humanos

## Introducción

**Descripción del problema:**

El sistema de gestión de recursos humanos busca organizar y administrar la información de empleados, departamentos y la empresa. Este sistema facilitará la gestión de personal, la asignación de departamentos y la administración de los registros de la empresa.

**Objetivo:**

El objetivo principal es crear una base de datos que permita gestionar la información de los empleados, los departamentos y la empresa. Además, se pretende realizar consultas eficientes para obtener datos clave como el número de empleados por departamento, el salario promedio de los empleados, entre otros.

---

**Análisis de Requerimientos****Entidades clave:**

1. **Empleado:** Representa a los empleados de la empresa.
2. **Departamento:** Representa los departamentos dentro de la empresa.
3. **Empresa:** Representa la empresa que tiene varios departamentos y empleados.

**Atributos de las entidades:**

1. **Empleado:**
  - id\_empleado
  - nombre
  - correo
  - telefono
  - salario
  - fecha\_ingreso
  - id\_departamento (clave foránea que referencia a Departamento)
2. **Departamento:**
  - id\_departamento
  - nombre
  - presupuesto
3. **Empresa:**
  - id\_empresa
  - nombre
  - ubicacion

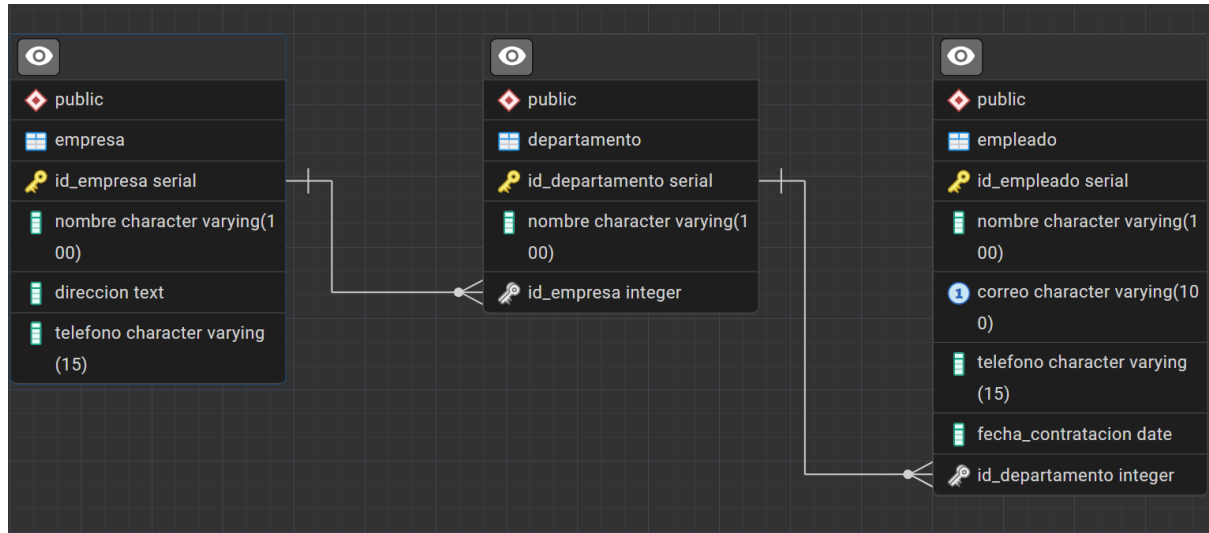
**Relaciones entre las entidades:**

- Un empleado pertenece a un solo departamento, pero un departamento puede tener muchos empleados.
- Una empresa tiene varios departamentos.

---

## Modelo Entidad-Relación (E-R)

El diagrama E-R muestra cómo se relacionan las entidades Empleado, Departamento y Empresa. Las relaciones están indicadas mediante líneas entre las entidades, y las claves foráneas garantizan la integridad referencial entre las tablas.



---

## Conversión a Esquema Relacional

El modelo E-R se convierte en el siguiente conjunto de tablas que representan las entidades y sus relaciones.

### Tablas:

1. **Empleado:**
  - **Clave primaria:** id\_empleado
  - **Clave foránea:** id\_departamento (referencia a Departamento)
2. **Departamento:**
  - **Clave primaria:** id\_departamento
3. **Empresa:**
  - **Clave primaria:** id\_empresa

---

## Implementación en DBMS (LDD)

A continuación, se presentan las sentencias SQL para crear las tablas necesarias para el sistema:

```
Query  Query History
1  CREATE TABLE Empresa (
2      id_empresa SERIAL PRIMARY KEY,
3      nombre VARCHAR(100) NOT NULL,
4      direccion TEXT NOT NULL,
5      telefono VARCHAR(15) NOT NULL
6  );
7
8  CREATE TABLE Departamento (
9      id_departamento SERIAL PRIMARY KEY,
10     nombre VARCHAR(100) NOT NULL,
11     id_empresa INT NOT NULL,
12     FOREIGN KEY (id_empresa) REFERENCES Empresa(id_empresa) ON DELETE CASCADE
13 );
14
15 CREATE TABLE Empleado (
16     id_empleado SERIAL PRIMARY KEY,
17     nombre VARCHAR(100) NOT NULL,
18     correo VARCHAR(100) UNIQUE NOT NULL,
19     telefono VARCHAR(15) NOT NULL,
20     fecha_contratacion DATE NOT NULL,
21     id_departamento INT NOT NULL,
22     FOREIGN KEY (id_departamento) REFERENCES Departamento(id_departamento) ON DELETE CASCADE
23 );
```

## Manipulación de Datos (LMD)

A continuación, se presentan las consultas SQL utilizadas para manipular los datos y realizar las consultas pertinentes.

```
Query  Query History
1  -- Insertar en la tabla Empresa
2  INSERT INTO Empresa (nombre, direccion, telefono) VALUES
3  ('TechCorp', 'Calle 123, Ciudad A', '1234567890'),
4  ('SoftSolutions', 'Avenida 456, Ciudad B', '2345678901'),
5  ('InnovaTech', 'Calle 789, Ciudad C', '3456789012'),
6  ('NextGen Systems', 'Boulevard 101, Ciudad D', '4567890123'),
7  ('Alpha Software', 'Plaza 202, Ciudad E', '5678901234');
8
9  -- Insertar en la tabla Departamento
10 INSERT INTO Departamento (nombre, id_empresa) VALUES
11 ('Recursos Humanos', 1), ('Desarrollo', 1), ('Soporte Técnico', 1), ('Ventas', 1),
12 ('Marketing', 2), ('Contabilidad', 2), ('Desarrollo', 2), ('Soporte Técnico', 2),
13 ('RRHH', 3), ('Investigación', 3), ('Desarrollo', 3), ('Ventas', 3),
14 ('Administración', 4), ('Finanzas', 4), ('Atención al Cliente', 4), ('Marketing', 4),
15 ('Producción', 5), ('Legal', 5), ('Desarrollo', 5), ('Innovación', 5);
16
```

```
Query  Query History
17 -- Insertar en la tabla Empleado
18 INSERT INTO Empleado (nombre, correo, telefono, fecha_contratacion, id_departamento) VALUES
19 ('Juan Pérez', 'juan.perez@email.com', '5551112222', '2020-01-15', 1),
20 ('María López', 'maria.lopez@email.com', '5551113333', '2019-03-20', 2),
21 ('Carlos García', 'carlos.garcia@email.com', '5551114444', '2021-07-10', 3),
22 ('Ana Torres', 'ana.torres@email.com', '5551115555', '2022-09-05', 4),
23 ('Luis Méndez', 'luis.mendez@email.com', '5551116666', '2018-05-14', 5),
24 ('Elena Sánchez', 'elena.sanchez@email.com', '5551117777', '2020-11-30', 6),
25 ('Pedro Rojas', 'pedro.rojas@email.com', '5551118888', '2017-08-25', 7),
26 ('Laura Gómez', 'laura.gomez@email.com', '5551119999', '2023-02-12', 8),
27 ('Ricardo Díaz', 'ricardo.diaz@email.com', '5552221111', '2016-06-18', 9),
28 ('Sofía Castro', 'sofia.castro@email.com', '5552222222', '2021-10-22', 10),
29 ('Manuel Herrera', 'manuel.herrera@email.com', '5552223333', '2019-12-01', 11),
30 ('Gabriela Vargas', 'gabriela.vargas@email.com', '5552224444', '2022-04-07', 12),
31 ('Jorge Núñez', 'jorge.nunez@email.com', '5552225555', '2020-09-14', 13),
32 ('Carmen Silva', 'carmen.silva@email.com', '5552226666', '2018-01-28', 14),
33 ('Diego Ramírez', 'diego.ramirez@email.com', '5552227777', '2023-06-15', 15),
34 ('Rosa Ortega', 'rosa.ortega@email.com', '5552228888', '2017-11-09', 16),
35 ('Hugo Moreno', 'hugo.moreno@email.com', '5552229999', '2021-05-23', 17),
36 ('Valeria Paredes', 'valeria.paredes@email.com', '5553331111', '2019-07-17', 18),
37 ('Esteban Cruz', 'esteban.cruz@email.com', '5553332222', '2022-08-31', 19),
38 ('Paula Fuentes', 'paula.fuentes@email.com', '5553333333', '2016-03-05', 20);
39
```

## Consultas SQL (LMD)

### 1. INSERT INTO

Agregar un nuevo usuario:

```
Query  Query History
1  INSERT INTO Empleado (id_empleado, nombre, correo, telefono, fecha_ingreso, id_departamento)
2  VALUES (1, 'Juan Pérez', 'juanperez@email.com', '1234567890', '2024-03-01', 1);
3
```

### 2. SELECT con WHERE

Seleccionar todos los libros publicados después de 1950:

QueryQuery History

1

2

3

4

```
SELECT nombre, correo, telefono, fecha_contratacion
FROM Empleado
WHERE id_departamento = 2;
```

Data OutputMessagesNotifications

≡+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

Showing rows: 1 to 1

	nombre character varying (100) 🔒	correo character varying (100) 🔒	telefono character varying (15) 🔒	fecha_contratacion date 🔒
1	María López	maria.lopez@email.com	5551113333	2019-03-20

---

### 3. ORDER BY

Seleccionar todos los préstamos ordenados por fecha de préstamo de más reciente a más antiguo:

Query

Query History

1

2

3

4

SELECT nombre, fecha\_contratacion

FROM Empleado

ORDER BY fecha\_contratacion DESC;

Data Output

Messages

Notifications

≡+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

	nombre character varying (100) 🔒	fecha_contratacion date 🔒
1	Diego Ramírez	2023-06-15
2	Laura Gómez	2023-02-12
3	Ana Torres	2022-09-05
4	Esteban Cruz	2022-08-31
5	Gabriela Vargas	2022-04-07
6	Sofía Castro	2021-10-22
7	Carlos García	2021-07-10
8	Hugo Moreno	2021-05-23
9	Elena Sánchez	2020-11-30
10	Jorge Núñez	2020-09-14
11	Juan Pérez	2020-01-15

Total rows: 20

Query complete 00:00:00.108

## 4. GROUP BY

Contar la cantidad de préstamos por cada usuario:



```
Query  Query History
1  SELECT d.nombre AS departamento, COUNT(e.id_empleado) AS cantidad_empleados
2  FROM Empleado e
3  JOIN Departamento d ON e.id_departamento = d.id_departamento
4  GROUP BY d.nombre;
5

Data Output  Messages  Notifications
Showing rows: 1 to 14


|    | departamento<br>character varying (100) | cantidad_empleados<br>bigint |
|----|-----------------------------------------|------------------------------|
| 1  | Marketing                               | 2                            |
| 2  | RRHH                                    | 1                            |
| 3  | Legal                                   | 1                            |
| 4  | Investigación                           | 1                            |
| 5  | Desarrollo                              | 4                            |
| 6  | Contabilidad                            | 1                            |
| 7  | Producción                              | 1                            |
| 8  | Soporte Técnico                         | 2                            |
| 9  | Atención al Cliente                     | 1                            |
| 10 | Innovación                              | 1                            |
| 11 | Finanzas                                | 1                            |


Total rows: 14  Query complete 00:00:00.120
```

## 5. JOIN

Seleccionar los detalles del préstamo con el nombre del usuario y el título del libro:

Query

Query History

1

▼

SELECT e.nombre AS empleado, d.nombre AS departamento, e.fecha\_contratacion

2

FROM Empleado e

3

JOIN Departamento d ON e.id\_departamento = d.id\_departamento;

4

5

Data Output

Messages

Notifications

≡+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

Showing rows: 1 to 20

✎

Page N

	empleado character varying (100) 🔒	departamento character varying (100) 🔒	fecha_contratacion date 🔒
1	Juan Pérez	Recursos Humanos	2020-01-15
2	María López	Desarrollo	2019-03-20
3	Carlos García	Soporte Técnico	2021-07-10
4	Ana Torres	Ventas	2022-09-05
5	Luis Méndez	Marketing	2018-05-14
6	Elena Sánchez	Contabilidad	2020-11-30
7	Pedro Rojas	Desarrollo	2017-08-25
8	Laura Gómez	Soporte Técnico	2023-02-12
9	Ricardo Díaz	RRHH	2016-06-18
10	Sofía Castro	Investigación	2021-10-22
11	Manuel Herrera	Desarrollo	2019-12-01

Total rows: 20

Query complete 00:00:00.115

## 6. LIKE

Query

Query History

1

▼

SELECT nombre, correo

2

FROM Empleado

3

WHERE nombre LIKE '%María%';

4

Data Output

Messages

Notifications

≡+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

	<div>nombre</div> <div>character varying (100)</div> <div>🔒</div>	<div>correo</div> <div>character varying (100)</div> <div>🔒</div>
1	María López	maria.lopez@email.com

# Plataforma de cursos en línea

## Introducción

### Descripción del problema:

El sistema de gestión de cursos en línea permite la inscripción de usuarios a los cursos, la administración de los cursos disponibles y el seguimiento de las inscripciones de los usuarios. Este sistema facilitará la gestión de cursos, usuarios y las inscripciones, proporcionando información clave sobre los usuarios que están inscritos en los cursos.

### Objetivo:

El objetivo principal es crear una base de datos para gestionar la información de usuarios, cursos e inscripciones. Se pretende realizar consultas eficientes para obtener datos como los cursos a los que un usuario está inscrito, el listado de usuarios inscritos en un curso, entre otros.

## Análisis de Requerimientos

### Entidades clave:

- **Usuario:** Representa a los usuarios que se registran en la plataforma de cursos en línea.
- **Curso:** Representa los cursos que están disponibles en la plataforma.
- **Inscripción:** Representa los registros de inscripción de los usuarios en los cursos.

### Atributos de las entidades:

- **Usuario:**
  - id\_usuario: Clave primaria
  - nombre: Nombre completo del usuario
  - correo: Correo electrónico del usuario
  - telefono: Número de teléfono del usuario
  - fecha\_registro: Fecha en la que el usuario se registró
- **Curso:**
  - id\_curso: Clave primaria
  - titulo: Título del curso
  - descripcion: Descripción del curso
  - fecha\_inicio: Fecha de inicio del curso
  - fecha\_fin: Fecha de finalización del curso
  - precio: Precio del curso
- **Inscripción:**
  - id\_inscripcion: Clave primaria
  - id\_usuario: Clave foránea que hace referencia a la tabla Usuario

- id\_curso: Clave foránea que hace referencia a la tabla Curso
- fecha\_inscripcion: Fecha en la que el usuario se inscribe en el curso
- estado: Estado de la inscripción (por ejemplo: 'Inscrito', 'Completado', 'Abandonado')

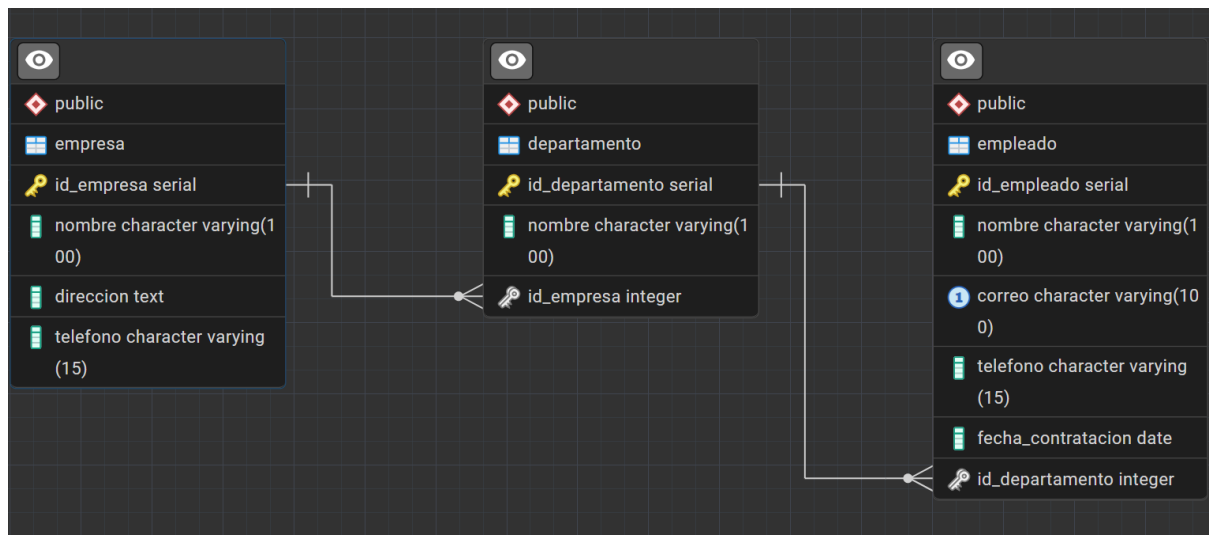
### Relaciones entre las entidades:

- Un **usuario** puede estar inscrito en varios **cursos**, y un **curso** puede tener varios **usuarios** inscritos.
- La relación entre **usuario** y **curso** es de muchos a muchos, lo que justifica la creación de la entidad **Inscripción**.

### Modelo Entidad-Relación (E-R)

El modelo E-R refleja cómo las entidades se relacionan entre sí:

- **Usuario** tiene una relación de muchos a muchos con **Curso**, a través de la entidad **Inscripción**.



### Conversión a Esquema Relacional

#### Tablas:

- **Usuario:**  
Clave primaria: id\_usuario
- **Curso:**  
Clave primaria: id\_curso
- **Inscripción:**  
Clave primaria: id\_inscripcion  
Claves foráneas:
  - id\_usuario (referencia a Usuario)

- id\_curso (referencia a Curso)

## Implementación en DBMS (LDD)

```
Query  Query History
1  CREATE TABLE Empresa (
2      id_empresa SERIAL PRIMARY KEY,
3      nombre VARCHAR(100) NOT NULL,
4      direccion TEXT NOT NULL,
5      telefono VARCHAR(15) NOT NULL
6  );
7
8  CREATE TABLE Departamento (
9      id_departamento SERIAL PRIMARY KEY,
10     nombre VARCHAR(100) NOT NULL,
11     id_empresa INT NOT NULL,
12     FOREIGN KEY (id_empresa) REFERENCES Empresa(id_empresa) ON DELETE CASCADE
13 );
14
15 CREATE TABLE Empleado (
16     id_empleado SERIAL PRIMARY KEY,
17     nombre VARCHAR(100) NOT NULL,
18     correo VARCHAR(100) UNIQUE NOT NULL,
19     telefono VARCHAR(15) NOT NULL,
20     fecha_contratacion DATE NOT NULL,
21     id_departamento INT NOT NULL,
22     FOREIGN KEY (id_departamento) REFERENCES Departamento(id_departamento) ON DELETE CASCADE
23 );
```

## Manipulación de Datos (LMD)

A continuación, se presentan las consultas SQL utilizadas para manipular los datos y realizar las consultas pertinentes.

```
Query  Query History
1  -- Insertar en la tabla Empresa
2  INSERT INTO Empresa (nombre, direccion, telefono) VALUES
3  ('TechCorp', 'Calle 123, Ciudad A', '1234567890'),
4  ('SoftSolutions', 'Avenida 456, Ciudad B', '2345678901'),
5  ('InnovaTech', 'Calle 789, Ciudad C', '3456789012'),
6  ('NextGen Systems', 'Boulevard 101, Ciudad D', '4567890123'),
7  ('Alpha Software', 'Plaza 202, Ciudad E', '5678901234');
8
9  -- Insertar en la tabla Departamento
10 INSERT INTO Departamento (nombre, id_empresa) VALUES
11 ('Recursos Humanos', 1), ('Desarrollo', 1), ('Soporte Técnico', 1), ('Ventas', 1),
12 ('Marketing', 2), ('Contabilidad', 2), ('Desarrollo', 2), ('Soporte Técnico', 2),
13 ('RRHH', 3), ('Investigación', 3), ('Desarrollo', 3), ('Ventas', 3),
14 ('Administración', 4), ('Finanzas', 4), ('Atención al Cliente', 4), ('Marketing', 4),
15 ('Producción', 5), ('Legal', 5), ('Desarrollo', 5), ('Innovación', 5);
16
```

```
Query  Query History
17 -- Insertar en la tabla Empleado
18 INSERT INTO Empleado (nombre, correo, telefono, fecha_contratacion, id_departamento) VALUES
19 ('Juan Pérez', 'juan.perez@email.com', '5551112222', '2020-01-15', 1),
20 ('María López', 'maria.lopez@email.com', '5551113333', '2019-03-20', 2),
21 ('Carlos García', 'carlos.garcia@email.com', '5551114444', '2021-07-10', 3),
22 ('Ana Torres', 'ana.torres@email.com', '5551115555', '2022-09-05', 4),
23 ('Luis Méndez', 'luis.mendez@email.com', '5551116666', '2018-05-14', 5),
24 ('Elena Sánchez', 'elena.sanchez@email.com', '5551117777', '2020-11-30', 6),
25 ('Pedro Rojas', 'pedro.rojas@email.com', '5551118888', '2017-08-25', 7),
26 ('Laura Gómez', 'laura.gomez@email.com', '5551119999', '2023-02-12', 8),
27 ('Ricardo Díaz', 'ricardo.diaz@email.com', '5552221111', '2016-06-18', 9),
28 ('Sofía Castro', 'sofia.castro@email.com', '5552222222', '2021-10-22', 10),
29 ('Manuel Herrera', 'manuel.herrera@email.com', '5552223333', '2019-12-01', 11),
30 ('Gabriela Vargas', 'gabriela.vargas@email.com', '5552224444', '2022-04-07', 12),
31 ('Jorge Núñez', 'jorge.nunez@email.com', '5552225555', '2020-09-14', 13),
32 ('Carmen Silva', 'carmen.silva@email.com', '5552226666', '2018-01-28', 14),
33 ('Diego Ramírez', 'diego.ramirez@email.com', '5552227777', '2023-06-15', 15),
34 ('Rosa Ortega', 'rosa.ortega@email.com', '5552228888', '2017-11-09', 16),
35 ('Hugo Moreno', 'hugo.moreno@email.com', '5552229999', '2021-05-23', 17),
36 ('Valeria Paredes', 'valeria.paredes@email.com', '5553331111', '2019-07-17', 18),
37 ('Esteban Cruz', 'esteban.cruz@email.com', '5553332222', '2022-08-31', 19),
38 ('Paula Fuentes', 'paula.fuentes@email.com', '5553333333', '2016-03-05', 20);
39
```

## Manipulación de Datos (LMD)

A continuación, se presentan las consultas SQL utilizadas para manipular los datos y realizar las consultas pertinentes.

### 1. INSERT INTO

Este query se usa para agregar nuevos registros a la base de datos. Aquí te doy un ejemplo de inserción de un nuevo usuario.

```
Query  Query History
1  INSERT INTO Empleado (id_empleado, nombre, correo, telefono, fecha_ingreso, id_departamento)
2  VALUES (1, 'Juan Pérez', 'juanperez@email.com', '1234567890', '2024-03-01', 1);
3
```

## 2. SELECT con WHERE

Este query se usa para seleccionar datos basados en una condición específica. Aquí seleccionamos todos los usuarios cuya fecha de registro es posterior al 1 de enero de 2024.

Query

Query History

1

2

3

4

SELECT nombre, correo, telefono, fecha\_contratacion

FROM Empleado

WHERE id\_departamento = 2;

Data Output

Messages

Notifications

≡+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

Showing rows: 1 to 1

	nombre character varying (100) 🔒	correo character varying (100) 🔒	telefono character varying (15) 🔒	fecha_contratacion date 🔒
1	María López	maria.lopez@email.com	5551113333	2019-03-20

### 3. ORDER BY

Este query se usa para ordenar los resultados de una consulta. Aquí ordenamos los cursos por su duración de forma ascendente.

Query

Query History

1

2

3

4

SELECT

nombre,

fecha\_contratacion

FROM

Empleado

ORDER BY

fecha\_contratacion

DESC;

Data Output

Messages

Notifications

≡+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

	nombre character varying (100) 🔒	fecha_contratacion date 🔒
1	Diego Ramírez	2023-06-15
2	Laura Gómez	2023-02-12
3	Ana Torres	2022-09-05
4	Esteban Cruz	2022-08-31
5	Gabriela Vargas	2022-04-07
6	Sofía Castro	2021-10-22
7	Carlos García	2021-07-10
8	Hugo Moreno	2021-05-23
9	Elena Sánchez	2020-11-30
10	Jorge Núñez	2020-09-14
11	Juan Pérez	2020-01-15

Total rows: 20

Query complete 00:00:00.108



## 4. GROUP BY

Este query se usa para agrupar los resultados de acuerdo a una columna específica. Aquí mostramos el número de usuarios registrados por cada mes de registro.

```
Query Query History
1 SELECT d.nombre AS departamento, COUNT(e.id_empleado) AS cantidad_empleados
2 FROM Empleado e
3 JOIN Departamento d ON e.id_departamento = d.id_departamento
4 GROUP BY d.nombre;
5
```

Data Output Messages Notifications

Showing rows: 1 to 14

	departamento character varying (100)	cantidad_empleados bigint
1	Marketing	2
2	RRHH	1
3	Legal	1
4	Investigación	1
5	Desarrollo	4
6	Contabilidad	1
7	Producción	1
8	Soporte Técnico	2
9	Atención al Cliente	1
10	Innovación	1
11	Finanzas	1
Total rows: 14		Query complete 00:00:00.120

## 5. JOIN

Este query se usa para combinar datos de dos o más tablas relacionadas. Aquí unimos las tablas **Usuario** y **Inscripcion** para obtener los usuarios que están inscritos en algún curso.

QueryQuery History

1

2

3

4

5

```
SELECT e.nombre AS empleado, d.nombre AS departamento, e.fecha_contratacion
FROM Empleado e
JOIN Departamento d ON e.id_departamento = d.id_departamento;
```

Data OutputMessagesNotifications

≡+

SQL

Showing rows: 1 to 20Page N

	empleado character varying (100)	departamento character varying (100)	fecha_contratacion date
1	Juan Pérez	Recursos Humanos	2020-01-15
2	María López	Desarrollo	2019-03-20
3	Carlos García	Soporte Técnico	2021-07-10
4	Ana Torres	Ventas	2022-09-05
5	Luis Méndez	Marketing	2018-05-14
6	Elena Sánchez	Contabilidad	2020-11-30
7	Pedro Rojas	Desarrollo	2017-08-25
8	Laura Gómez	Soporte Técnico	2023-02-12
9	Ricardo Díaz	RRHH	2016-06-18
10	Sofía Castro	Investigación	2021-10-22
11	Manuel Herrera	Desarrollo	2019-12-01

Total rows: 20

Query complete 00:00:00.115

## 6. LIKE

Este query se usa para buscar una cadena de texto dentro de una columna. Aquí buscamos los usuarios cuyo nombre contenga la palabra "Luis".

Query

Query History

1

2

3

4

SELECT

nombre,

correo

FROM

Empleado

WHERE

nombre

LIKE

'%María%';

Data Output

Messages

Notifications

≡+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

	<div>nombre</div> <div>character varying (100)</div> <div>🔒</div>	<div>correo</div> <div>character varying (100)</div> <div>🔒</div>
1	María López	maria.lopez@email.com