

Sistema de Gestión de Inventario

1. Definición de Entidades y Relaciones

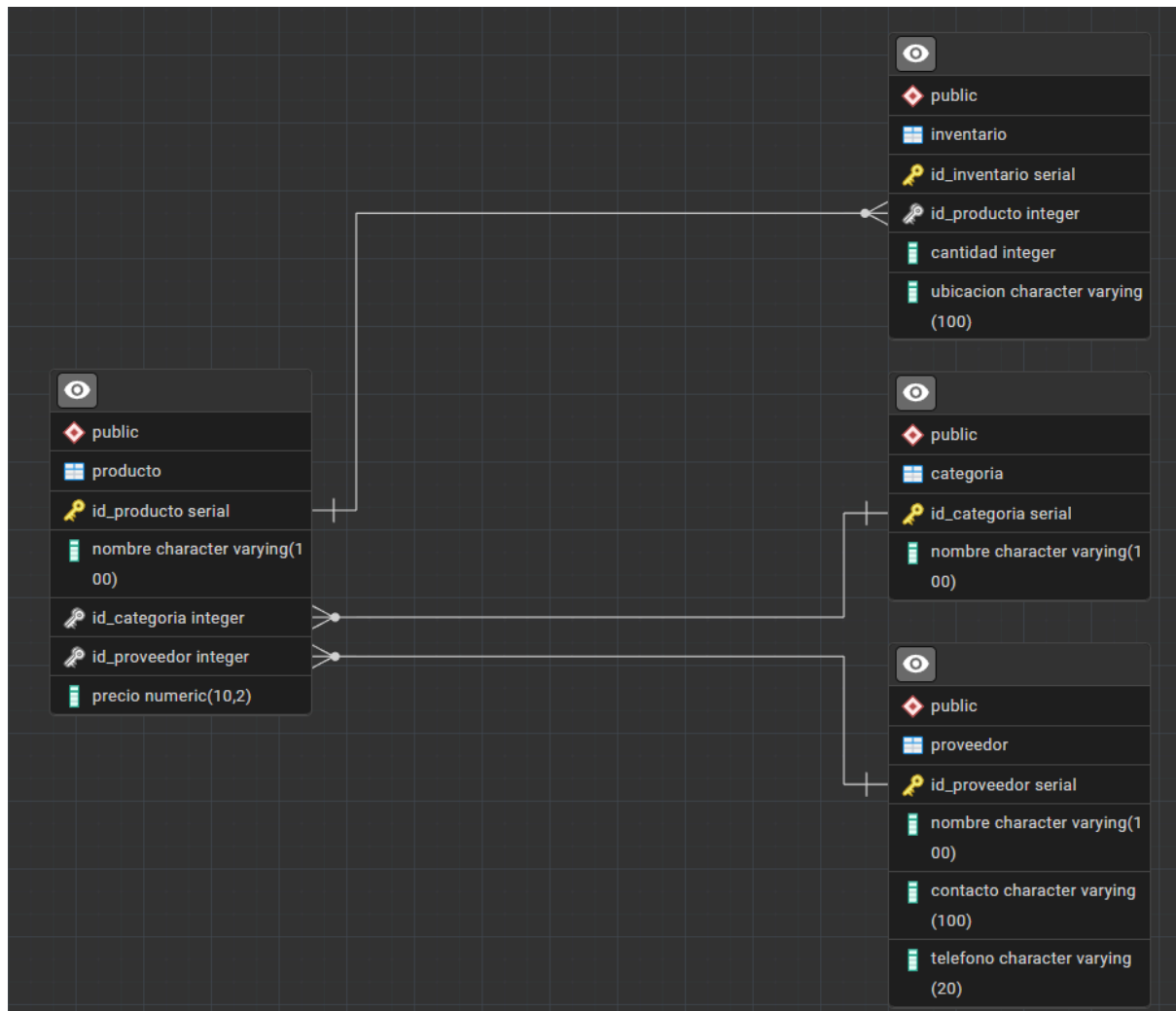
Las entidades clave para la gestión de productos, proveedores e inventario son las siguientes:

- **Categoría:** Representa las categorías de productos, como "Electrónica", "Ropa", etc. Atributos: ID, nombre.
- **Proveedor:** Representa los proveedores de productos. Atributos: ID, nombre, contacto, teléfono.
- **Producto:** Representa los productos disponibles en el inventario. Atributos: ID, nombre, precio, categoría y proveedor.
- **Inventario:** Registra la cantidad disponible de cada producto y su ubicación en el almacén. Atributos: ID, producto, cantidad, ubicación.

Las relaciones entre estas entidades son las siguientes:

- Un **proveedor** puede suministrar varios **productos**, pero cada **producto** es suministrado por un solo **proveedor**. (Relación uno a muchos)
- Un **producto** pertenece a una sola **categoría**, pero una **categoría** puede tener varios **productos**. (Relación uno a muchos)
- Un **producto** tiene registros en el **inventario** que indican la cantidad disponible y su ubicación. (Relación uno a uno)

2. Diseño del Modelo Conceptual (Diagrama E-R)



3. Conversión a Esquema Relacional

A partir del modelo E-R, las tablas quedan definidas de la siguiente manera:

- **Categoria (id_categoria, nombre)**
Clave primaria: id_categoria
- **Proveedor (id_proveedor, nombre, contacto, telefono)**
Clave primaria: id_proveedor
- **Producto (id_producto, nombre, id_categoria, id_proveedor, precio)**
Clave primaria: id_producto
Claves foráneas:

- id_categoria referencia a Categoria(id_categoria)
- id_proveedor referencia a Proveedor(id_proveedor)
- **Inventario (id_inventario, id_producto, cantidad, ubicacion)**
 Clave primaria: id_inventario
 Clave foránea:
 - id_producto referencia a Producto(id_producto)

4. Implementación en SQL

Creación de Tablas

| Query | Query History |
|-------------|--|
| 4 | |
| 5 | CREATE TABLE Categoria (|
| 6 | id_categoria SERIAL PRIMARY KEY, |
| 7 | nombre VARCHAR(100) NOT NULL |
| 8 |); |
| 9 | |
| 10 | CREATE TABLE Proveedor (|
| 11 | id_proveedor SERIAL PRIMARY KEY, |
| 12 | nombre VARCHAR(100) NOT NULL, |
| 13 | contacto VARCHAR(100), |
| 14 | telefono VARCHAR(20) |
| 15 |); |
| 16 | |
| 17 | CREATE TABLE Producto (|
| 18 | id_producto SERIAL PRIMARY KEY, |
| 19 | nombre VARCHAR(100) NOT NULL, |
| 20 | id_categoria INT NOT NULL, |
| 21 | id_proveedor INT NOT NULL, |
| 22 | precio DECIMAL(10,2) NOT NULL, |
| 23 | FOREIGN KEY (id_categoria) REFERENCES Categoria(id_categoria), |
| 24 | FOREIGN KEY (id_proveedor) REFERENCES Proveedor(id_proveedor) |
| 25 |); |
| 26 | |
| 27 | CREATE TABLE Inventario (|
| 28 | id_inventario SERIAL PRIMARY KEY, |
| 29 | id_producto INT NOT NULL, |
| 30 | cantidad INT NOT NULL, |
| 31 | ubicacion VARCHAR(100), |
| 32 | FOREIGN KEY (id_producto) REFERENCES Producto(id_producto) |
| 33 |); |
| 34 | |
| Total rows: | |

Inserción de Datos

| Query | Query History |
|-------------|--|
| 39 | -- Categorías |
| 40 | INSERT INTO Categoria (nombre) VALUES |
| 41 | ('Electrónica'), |
| 42 | ('Ropa'), |
| 43 | ('Alimentos'), |
| 44 | ('Juguetes'), |
| 45 | ('Hogar'), |
| 46 | ('Deportes'), |
| 47 | ('Libros'), |
| 48 | ('Cuidado Personal'), |
| 49 | ('Papelería'), |
| 50 | ('Mascotas'); |
| 51 | |
| 52 | -- Proveedores |
| 53 | INSERT INTO Proveedor (nombre, contacto, telefono) VALUES |
| 54 | ('Tech Supplier Inc.', 'Juan Pérez', '555-1234'), |
| 55 | ('Moda Express', 'Ana López', '555-5678'), |
| 56 | ('Distribuidora Gourmet', 'Carlos Gómez', '555-9012'), |
| 57 | ('Juguetes Kids', 'Laura Martínez', '555-3456'), |
| 58 | ('Hogar Plus', 'Pedro Sánchez', '555-7890'), |
| 59 | ('Sport Life', 'Marta Ramírez', '555-2345'), |
| 60 | ('Book House', 'Luis Díaz', '555-6789'), |
| 61 | ('Cuidado Total', 'Sofía Torres', '555-4321'), |
| 62 | ('Papel y Más', 'Ricardo Flores', '555-8765'), |
| 63 | ('Pet World', 'María Ruiz', '555-6543'); |
| 64 | |
| 65 | -- Productos |
| 66 | INSERT INTO Producto (nombre, id_categoria, id_proveedor, precio) VALUES |
| 67 | ('Laptop Gamer', 1, 1, 1200.00), |
| 68 | ('Smartphone X', 1, 1, 800.00), |
| 69 | ('Pantalón Jeans', 2, 2, 40.00), |
| 70 | ('Camiseta Deportiva', 2, 2, 25.00), |
| 71 | ('Caja de Chocolates', 3, 3, 15.00), |
| 72 | ('Cereal Integral', 3, 3, 8.00), |
| 73 | ('Set de Juguetes', 4, 4, 30.00), |
| 74 | ('Lámpara LED', 5, 5, 45.00), |
| 75 | ('Balón de Fútbol', 6, 6, 22.00), |
| 76 | ('Libro "Aprende SQL"', 7, 7, 18.00); |
| 77 | |
| 78 | -- Inventario |
| 79 | INSERT INTO Inventario (id_producto, cantidad, ubicacion) VALUES |
| 80 | (1, 10, 'Almacén A1'), |
| 81 | (2, 15, 'Almacén A2'), |
| 82 | (3, 50, 'Almacén B1'), |
| 83 | (4, 40, 'Almacén B2'), |
| 84 | (5, 30, 'Almacén C1'), |
| 85 | (6, 25, 'Almacén C2'), |
| 86 | (7, 20, 'Almacén D1'), |
| 87 | (8, 35, 'Almacén E1'), |
| 88 | (9, 18, 'Almacén F1'), |
| 89 | (10, 60, 'Almacén G1'); |
| 90 | |
| 91 | |
| Total rows: | |

5. Consultas SQL

- **Consulta requerida:** Obtener la lista de productos con sus respectivas categorías y proveedores, ordenados alfabéticamente por nombre de producto.

```
95 SELECT
96     p.nombre AS producto,
97     c.nombre AS categoria,
98     pr.nombre AS proveedor
99 FROM Producto p
100 JOIN Categoria c ON p.id_categoria = c.id_categoria
101 JOIN Proveedor pr ON p.id_proveedor = pr.id_proveedor
102 ORDER BY p.nombre ASC;
```

Data Output Messages Notifications

| | producto character varying (100) | categoria character varying (100) | proveedor character varying (100) |
|----|-------------------------------------|--------------------------------------|--------------------------------------|
| 1 | Balón de Fútbol | Deportes | Sport Life |
| 2 | Caja de Chocolates | Alimentos | Distribuidora Gourmet |
| 3 | Camiseta Deportiva | Ropa | Moda Express |
| 4 | Cereal Integral | Alimentos | Distribuidora Gourmet |
| 5 | Lámpara LED | Hogar | Hogar Plus |
| 6 | Laptop Gamer | Electrónica | Tech Supplier Inc. |
| 7 | Libro "Aprende SQL" | Libros | Book House |
| 8 | Pantalón Jeans | Ropa | Moda Express |
| 9 | Set de Juguetes | Juguetes | Juguetes Kids |
| 10 | Smartphone X | Electrónica | Tech Supplier Inc. |

- Consulta SELECT

```
109 -- Esta consulta selecciona los nombres de productos y las categorías a las que pertenecen.
110 SELECT
111     p.nombre AS producto,
112     c.nombre AS categoria
113 FROM Producto p
114 JOIN Categoria c ON p.id_categoria = c.id_categoria;
```

Data Output Messages Notifications

| | producto character varying (100) 🔒 | categoria character varying (100) 🔒 |
|----|---------------------------------------|--|
| 1 | Laptop Gamer | Electrónica |
| 2 | Smartphone X | Electrónica |
| 3 | Pantalón Jeans | Ropa |
| 4 | Camiseta Deportiva | Ropa |
| 5 | Caja de Chocolates | Alimentos |
| 6 | Cereal Integral | Alimentos |
| 7 | Set de Juguetes | Juguetes |
| 8 | Lámpara LED | Hogar |
| 9 | Balón de Fútbol | Deportes |
| 10 | Libro "Aprende SQL" | Libros |

- Consulta WHERE

```
117 -- Esta consulta selecciona los productos que pertenecen a la categoría 'Electrónica'.
118 SELECT
119     p.nombre AS producto,
120     c.nombre AS categoria
121 FROM Producto p
122 JOIN Categoria c ON p.id_categoria = c.id_categoria
123 WHERE c.nombre = 'Electrónica';
```

Data Output Messages Notifications

| | producto character varying (100) 🔒 | categoria character varying (100) 🔒 |
|---|---------------------------------------|--|
| 1 | Laptop Gamer | Electrónica |
| 2 | Smartphone X | Electrónica |

- Consulta ORDER BY

```
126 -- Esta consulta selecciona los productos y categorías, ordenados alfabéticamente por el nombre del producto.
127 SELECT
128     p.nombre AS producto,
129     c.nombre AS categoria
130 FROM Producto p
131 JOIN Categoria c ON p.id_categoria = c.id_categoria
132 ORDER BY p.nombre ASC;
```

| | producto character varying (100) | categoria character varying (100) |
|----|-------------------------------------|--------------------------------------|
| 1 | Balón de Fútbol | Deportes |
| 2 | Caja de Chocolates | Alimentos |
| 3 | Camiseta Deportiva | Ropa |
| 4 | Cereal Integral | Alimentos |
| 5 | Lámpara LED | Hogar |
| 6 | Laptop Gamer | Electrónica |
| 7 | Libro "Aprende SQL" | Libros |
| 8 | Pantalón Jeans | Ropa |
| 9 | Set de Juguetes | Juguetes |
| 10 | Smartphone X | Electrónica |

- Consulta GROUP BY

```
135 -- Esta consulta muestra la cantidad total de productos por cada categoría.
136 SELECT
137     c.nombre AS categoria,
138     COUNT(p.id_producto) AS total_productos
139 FROM Producto p
140 JOIN Categoria c ON p.id_categoria = c.id_categoria
141 GROUP BY c.nombre;
```

| | categoria character varying (100) | total_productos bigint |
|---|--------------------------------------|---------------------------|
| 1 | Electrónica | 2 |
| 2 | Hogar | 1 |
| 3 | Juguetes | 1 |
| 4 | Deportes | 1 |
| 5 | Alimentos | 2 |
| 6 | Libros | 1 |
| 7 | Ropa | 2 |

- Consulta JOIN

```

144 -- Esta consulta muestra los productos junto con sus proveedores.
145 SELECT
146     p.nombre AS producto,
147     pr.nombre AS proveedor
148 FROM Producto p
149 JOIN Proveedor pr ON p.id_proveedor = pr.id_proveedor;

```

Data Output Messages Notifications

| | producto character varying (100) | proveedor character varying (100) |
|----|-------------------------------------|--------------------------------------|
| 1 | Laptop Gamer | Tech Supplier Inc. |
| 2 | Smartphone X | Tech Supplier Inc. |
| 3 | Pantalón Jeans | Moda Express |
| 4 | Camiseta Deportiva | Moda Express |
| 5 | Caja de Chocolates | Distribuidora Gourmet |
| 6 | Cereal Integral | Distribuidora Gourmet |
| 7 | Set de Juguetes | Juguetes Kids |
| 8 | Lámpara LED | Hogar Plus |
| 9 | Balón de Fútbol | Sport Life |
| 10 | Libro "Aprende SQL" | Book House |

- Consulta LIKE

```

152 -- Esta consulta selecciona los productos cuyo nombre contiene la palabra 'Laptop'.
153 SELECT
154     p.nombre AS producto,
155     c.nombre AS categoria
156 FROM Producto p
157 JOIN Categoria c ON p.id_categoria = c.id_categoria
158 WHERE p.nombre LIKE '%Laptop%';
159

```

Data Output Messages Notifications

| | producto character varying (100) | categoria character varying (100) |
|---|-------------------------------------|--------------------------------------|
| 1 | Laptop Gamer | Electrónica |

Sistema de Gestión de Eventos

1. Definición de Entidades y Relaciones

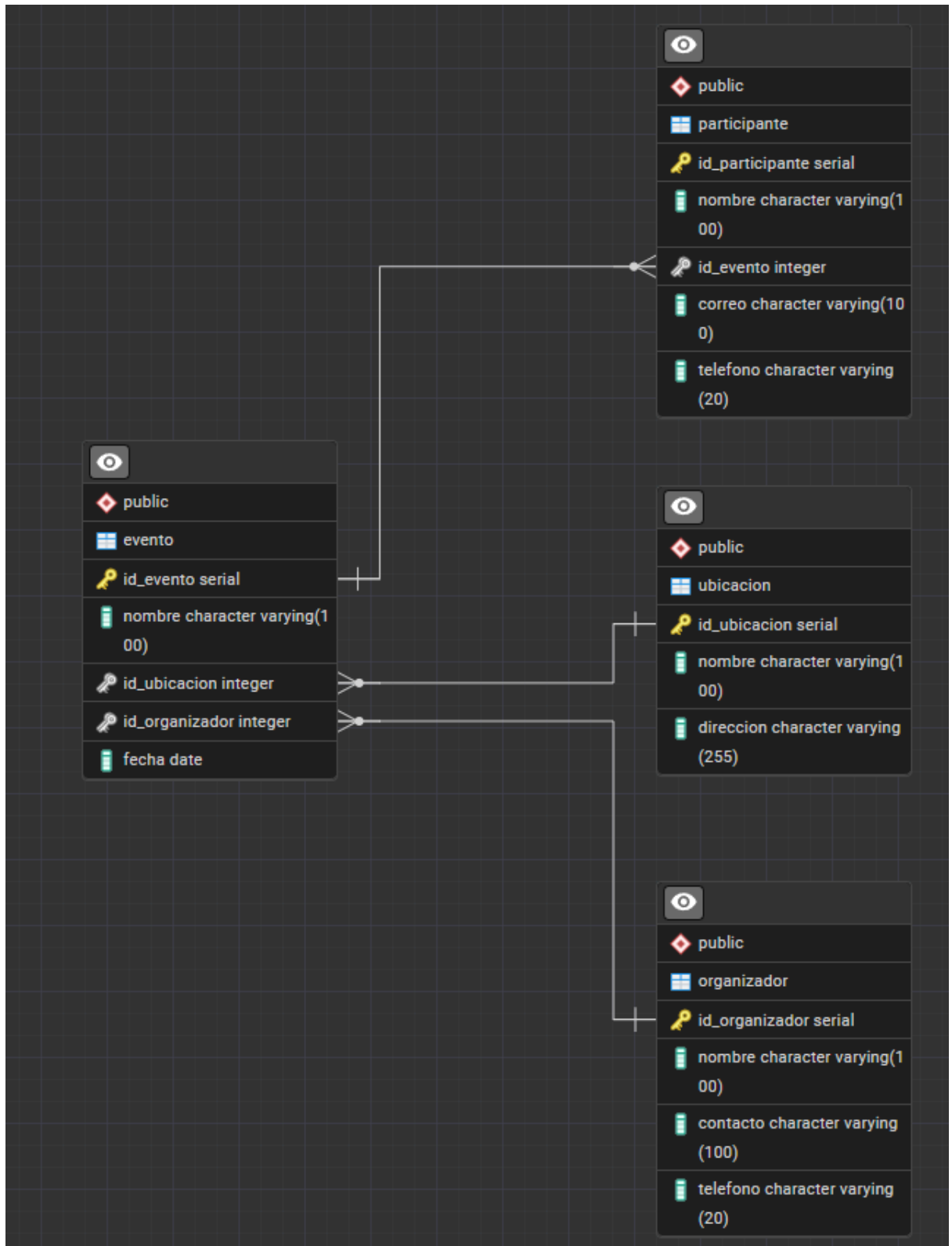
Para la gestión de eventos y participantes, se identifican las siguientes entidades clave:

- **Ubicación:** Representa los lugares donde se llevan a cabo los eventos. Contiene atributos como ID, nombre y dirección.
- **Organizador:** Representa a las empresas o personas encargadas de coordinar los eventos. Contiene atributos como ID, nombre, contacto y teléfono.
- **Evento:** Representa los eventos programados en distintas ubicaciones y organizados por diferentes entidades. Contiene atributos como ID, nombre, ID de ubicación, ID de organizador y fecha del evento.
- **Participante:** Representa a las personas inscritas en un evento específico. Contiene atributos como ID, nombre, ID de evento, correo y teléfono.

Las relaciones entre estas entidades son las siguientes:

- Un organizador puede gestionar múltiples eventos, pero cada evento tiene un solo organizador.
- Un evento se lleva a cabo en una sola ubicación, pero una ubicación puede albergar múltiples eventos.
- Un evento puede tener múltiples participantes, y cada participante solo puede estar registrado en un evento a la vez.

2. Diseño del Modelo Conceptual (Diagrama E-R)



3. Conversión a Esquema Relacional

A partir del modelo E-R, se definen las siguientes tablas:

- **Ubicación** (id_ubicacion, nombre, dirección)
- **Organizador** (id_organizador, nombre, contacto, teléfono)
- **Evento** (id_evento, nombre, id_ubicacion, id_organizador, fecha)
 - id_ubicacion es clave foránea que referencia a Ubicación(id_ubicacion).
 - id_organizador es clave foránea que referencia a Organizador(id_organizador).
- **Participante** (id_participante, nombre, id_evento, correo, teléfono)
 - id_evento es clave foránea que referencia a Evento(id_evento).

4. Implementación en SQL

- Creación de Tablas

```
5  -- Crear tabla de Ubicaciones
6  ✓ CREATE TABLE Ubicacion (
7      id_ubicacion SERIAL PRIMARY KEY,
8      nombre VARCHAR(100) NOT NULL,
9      direccion VARCHAR(255) NOT NULL
10 );
11
12 -- Crear tabla de Organizadores
13 ✓ CREATE TABLE Organizador (
14     id_organizador SERIAL PRIMARY KEY,
15     nombre VARCHAR(100) NOT NULL,
16     contacto VARCHAR(100),
17     telefono VARCHAR(20)
18 );
19
20 -- Crear tabla de Eventos
21 ✓ CREATE TABLE Evento (
22     id_evento SERIAL PRIMARY KEY,
23     nombre VARCHAR(100) NOT NULL,
24     id_ubicacion INT NOT NULL,
25     id_organizador INT NOT NULL,
26     fecha DATE NOT NULL,
27     FOREIGN KEY (id_ubicacion) REFERENCES Ubicacion(id_ubicacion),
28     FOREIGN KEY (id_organizador) REFERENCES Organizador(id_organizador)
29 );
30
31 -- Crear tabla de Participantes
32 ✓ CREATE TABLE Participante (
33     id_participante SERIAL PRIMARY KEY,
34     nombre VARCHAR(100) NOT NULL,
35     id_evento INT NOT NULL,
36     correo VARCHAR(100),
37     telefono VARCHAR(20),
38     FOREIGN KEY (id_evento) REFERENCES Evento(id_evento)
39 );
```

- Inserción de Datos

```
45 -- Insertar datos en Ubicaciones
46 ▾ INSERT INTO Ubicacion (nombre, direccion) VALUES
47 ('Auditorio Nacional', 'Av. Paseo de la Reforma 50, CDMX'),
48 ('Centro de Convenciones', 'Av. Juárez 90, Guadalajara'),
49 ('Teatro Metropolitano', 'Av. Independencia 55, Monterrey'),
50 ('Palacio de los Deportes', 'Periférico Sur 3300, CDMX'),
51 ('Arena Ciudad de México', 'Av. De la Viga 120, CDMX'),
52 ('Teatro Telmex', 'Av. Río Mixcoac 1000, CDMX'),
53 ('Auditorio Banamex', 'Av. Conscripto 311, Monterrey'),
54 ('Centro Expo Guadalajara', 'Calle 5, Guadalajara'),
55 ('Zócalo de la Ciudad de México', 'Plaza Mayor, CDMX'),
56 ('Explanada de los Hércules', 'Av. Tecnológico 1200, Monterrey');
57
58 -- Insertar datos en Organizadores
59 ▾ INSERT INTO Organizador (nombre, contacto, telefono) VALUES
60 ('Eventi', 'Carlos Pérez', '555-1234'),
61 ('Conexión Eventos', 'Ana Gómez', '555-5678'),
62 ('ProEventos', 'Luis Martínez', '555-6789'),
63 ('Grand Show', 'María López', '555-2345'),
64 ('Max Eventos', 'Jorge Ruiz', '555-3456'),
65 ('Live Entertainment', 'Claudia Vega', '555-4567'),
66 ('Producciones del Norte', 'Eduardo Ramírez', '555-5670'),
67 ('Eventos Creativos', 'Carlos Mendoza', '555-6780'),
68 ('Mundo Eventos', 'Gabriela Sánchez', '555-7891'),
69 ('Mega Producciones', 'Oscar Ruiz', '555-8901');
```

```

71 -- Insertar datos en Eventos
72 INSERT INTO Evento (nombre, id_ubicacion, id_organizador, fecha) VALUES
73 ('Concierto de Rock', 1, 1, '2025-05-01'),
74 ('Congreso de Tecnología', 2, 2, '2025-06-10'),
75 ('Festival de Cine', 3, 3, '2025-07-15'),
76 ('Concierto de Salsa', 4, 4, '2025-08-20'),
77 ('Feria Internacional', 5, 5, '2025-09-25'),
78 ('Evento Deportivo', 6, 6, '2025-10-10'),
79 ('Conferencia de Marketing', 7, 7, '2025-11-05'),
80 ('Festival Gastronómico', 8, 8, '2025-12-10'),
81 ('Concierto de Jazz', 9, 9, '2025-01-12'),
82 ('Carnaval Cultural', 10, 10, '2025-02-15');
83
84 -- Insertar datos en Participantes
85 INSERT INTO Participante (nombre, id_evento, correo, telefono) VALUES
86 ('Juan Pérez', 1, 'juan@mail.com', '555-9876'),
87 ('Ana López', 1, 'ana@mail.com', '555-2345'),
88 ('Luis Gómez', 2, 'luis@mail.com', '555-3456'),
89 ('Marta Rodríguez', 2, 'marta@mail.com', '555-4567'),
90 ('Carlos Fernández', 3, 'carlos@mail.com', '555-5678'),
91 ('Gabriela Sánchez', 3, 'gabriela@mail.com', '555-6789'),
92 ('Oscar Ramírez', 4, 'oscar@mail.com', '555-7890'),
93 ('Pedro Martínez', 4, 'pedro@mail.com', '555-8901'),
94 ('Claudia Ruiz', 5, 'claudia@mail.com', '555-1234'),
95 ('Sofía Gómez', 5, 'sofia@mail.com', '555-2345');

```

5. Consultas SQL

- Consulta requerida:

```

101 -- Obtener la lista de eventos programados junto con la cantidad de participantes registrados por evento
102 SELECT
103     e.nombre AS evento,
104     COUNT(p.id_participante) AS participantes_registrados
105 FROM Evento e
106 LEFT JOIN Participante p ON e.id_evento = p.id_evento
107 GROUP BY e.id_evento
108 ORDER BY e.fecha ASC;
109

```

Data Output Messages Notifications

| | evento character varying (100) | participantes_registrados bigint |
|----|-----------------------------------|-------------------------------------|
| 1 | Concierto de Jazz | 0 |
| 2 | Carnaval Cultural | 0 |
| 3 | Concierto de Rock | 2 |
| 4 | Congreso de Tecnología | 2 |
| 5 | Festival de Cine | 2 |
| 6 | Concierto de Salsa | 2 |
| 7 | Feria Internacional | 2 |
| 8 | Evento Deportivo | 0 |
| 9 | Conferencia de Marketing | 0 |
| 10 | Festival Gastronómico | 0 |

- **Consulta SELECT**

```

115 -- Esta consulta obtiene los nombres de los eventos y la cantidad de participantes registrados por evento.
116 SELECT
117     e.nombre AS evento,
118     COUNT(p.id_participante) AS participantes_registrados
119 FROM Evento e
120 LEFT JOIN Participante p ON e.id_evento = p.id_evento
121 GROUP BY e.id_evento;

```

| | evento character varying (100) | participantes_registrados bigint |
|----|-----------------------------------|-------------------------------------|
| 1 | Feria Internacional | 2 |
| 2 | Concierto de Salsa | 2 |
| 3 | Carnaval Cultural | 0 |
| 4 | Evento Deportivo | 0 |
| 5 | Congreso de Tecnología | 2 |
| 6 | Conferencia de Marketing | 0 |
| 7 | Concierto de Rock | 2 |
| 8 | Festival Gastronómico | 0 |
| 9 | Concierto de Jazz | 0 |
| 10 | Festival de Cine | 2 |

- **Consulta WHERE**

```

124 -- Esta consulta selecciona los eventos cuyo nombre contiene la palabra 'Concierto'.
125 SELECT
126     e.nombre AS evento,
127     e.fecha AS fecha,
128     u.nombre AS ubicacion
129 FROM Evento e
130 JOIN Ubicacion u ON e.id_ubicacion = u.id_ubicacion
131 WHERE e.nombre LIKE '%Concierto%';

```

| | evento character varying (100) | fecha date | ubicacion character varying (100) |
|---|-----------------------------------|---------------|--------------------------------------|
| 1 | Concierto de Rock | 2025-05-01 | Auditorio Nacional |
| 2 | Concierto de Salsa | 2025-08-20 | Palacio de los Deportes |
| 3 | Concierto de Jazz | 2025-01-12 | Zócalo de la Ciudad de México |

- **Consulta ORDER BY**

```

134  -- Esta consulta obtiene los eventos y los organiza por fecha de manera ascendente.
135  SELECT
136      e.nombre AS evento,
137      e.fecha AS fecha,
138      u.nombre AS ubicacion
139  FROM Evento e
140  JOIN Ubicacion u ON e.id_ubicacion = u.id_ubicacion
141  ORDER BY e.fecha ASC;

```

Data Output Messages Notifications

| | evento character varying (100) | fecha date | ubicacion character varying (100) |
|----|-----------------------------------|---------------|--------------------------------------|
| 1 | Concierto de Jazz | 2025-01-12 | Zócalo de la Ciudad de México |
| 2 | Carnaval Cultural | 2025-02-15 | Explanada de los Hércules |
| 3 | Concierto de Rock | 2025-05-01 | Auditorio Nacional |
| 4 | Congreso de Tecnología | 2025-06-10 | Centro de Convenciones |
| 5 | Festival de Cine | 2025-07-15 | Teatro Metropolitano |
| 6 | Concierto de Salsa | 2025-08-20 | Palacio de los Deportes |
| 7 | Feria Internacional | 2025-09-25 | Arena Ciudad de México |
| 8 | Evento Deportivo | 2025-10-10 | Teatro Telmex |
| 9 | Conferencia de Marketing | 2025-11-05 | Auditorio Banamex |
| 10 | Festival Gastronómico | 2025-12-10 | Centro Expo Guadalajara |

- **Consulta GROUP BY**

```

144  -- Esta consulta obtiene el número de eventos organizados por cada organizador.
145  SELECT
146      o.nombre AS organizador,
147      COUNT(e.id_evento) AS total_eventos
148  FROM Evento e
149  JOIN Organizador o ON e.id_organizador = o.id_organizador
150  GROUP BY o.nombre;

```

Data Output Messages Notifications

| | organizador character varying (100) | totalEventos bigint |
|----|--|------------------------|
| 1 | Producciones del Norte | 1 |
| 2 | Conexión Eventos | 1 |
| 3 | Max Eventos | 1 |
| 4 | ProEventos | 1 |
| 5 | Eventos Creativos | 1 |
| 6 | Mundo Eventos | 1 |
| 7 | Grand Show | 1 |
| 8 | Mega Producciones | 1 |
| 9 | Live Entertainment | 1 |
| 10 | Eventi | 1 |

- **Consulta JOIN**

```
153 -- Esta consulta muestra los participantes de un evento específico ('Concierto de Rock').
154 SELECT
155     p.nombre AS participante,
156     p.telefono AS telefono
157 FROM Participante p
158 JOIN Evento e ON p.id_evento = e.id_evento
159 WHERE e.nombre = 'Concierto de Rock';
```

Data Output Messages Notifications

| | participante character varying (100) | telefono character varying (20) |
|---|---|------------------------------------|
| 1 | Juan Pérez | 555-9876 |
| 2 | Ana López | 555-2345 |

- **Consulta LIKE**

```
162 -- Esta consulta selecciona los eventos que ocurrirán en la fecha '2025-05-01'.
163 SELECT
164     e.nombre AS evento,
165     e.fecha AS fecha,
166     o.nombre AS organizador
167 FROM Evento e
168 JOIN Organizador o ON e.id_organizador = o.id_organizador
169 WHERE e.fecha = '2025-05-01';
170
```

Data Output Messages Notifications

| | evento character varying (100) | fecha date | organizador character varying (100) |
|---|-----------------------------------|---------------|--|
| 1 | Concierto de Rock | 2025-05-01 | Eventi |

Plataforma de Streaming de Música

1. Definición de Entidades y Relaciones

Para la administración de reproducciones de música en una plataforma, se identifican las siguientes entidades clave:

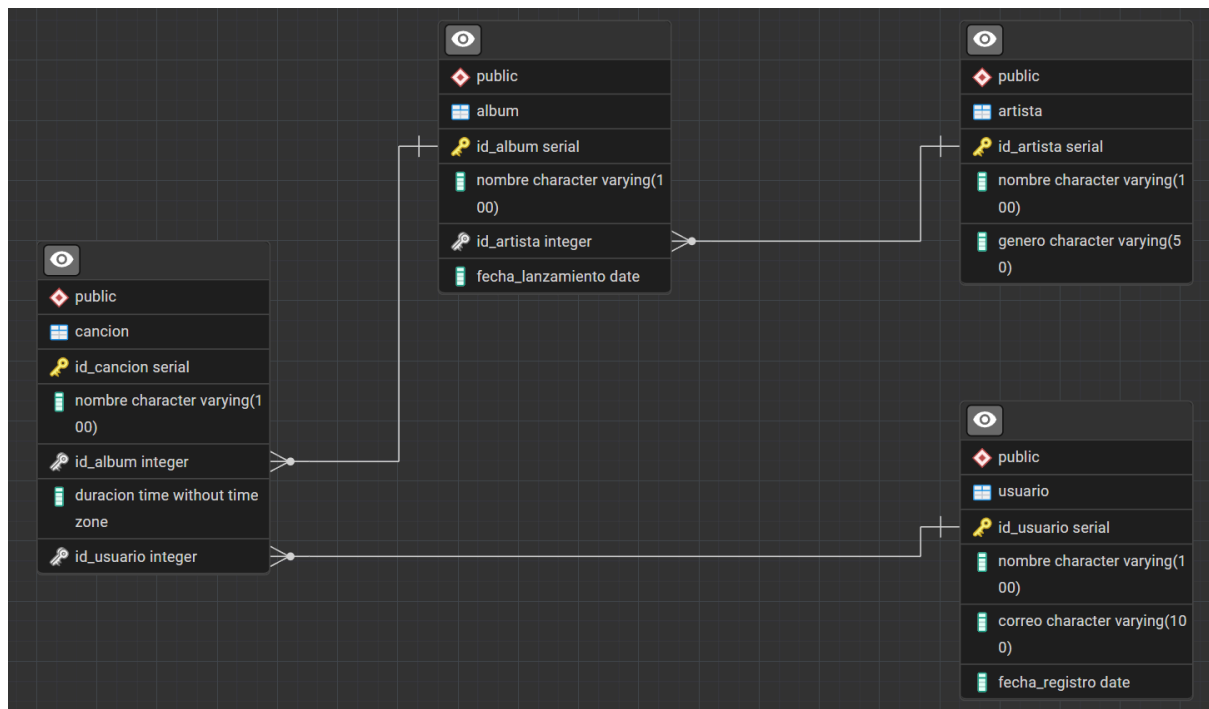
- **Usuario:** Representa a los usuarios registrados en la plataforma. Tiene atributos como ID, nombre, correo electrónico y fecha de registro.
- **Artista:** Representa a los músicos o bandas. Contiene atributos como ID, nombre y género musical.
- **Álbum:** Representa los discos publicados por los artistas. Contiene atributos como ID, nombre, ID del artista y fecha de lanzamiento.

- **Canción:** Representa las canciones incluidas en los álbumes. Contiene atributos como ID, nombre, ID del álbum y duración.
- **Reproducción:** Registra las veces que un usuario ha reproducido una canción. Incluye ID del usuario, ID de la canción y la fecha de reproducción.

Las relaciones entre estas entidades son las siguientes:

- Un **artista** puede tener varios **álbumes**, pero un álbum pertenece a un solo artista.
- Un **álbum** contiene varias **canciones**, pero una canción pertenece a un solo álbum.
- Un **usuario** puede reproducir varias **canciones**, y una canción puede ser reproducida por varios usuarios (relación muchos a muchos).

2. Diseño del Modelo Conceptual (Diagrama E-R)



3. Conversión a Esquema Relacional

A partir del modelo E-R, se definen las siguientes tablas:

- **Usuario** (ID_Usuario, Nombre, Correo, Fecha_Registro)
- **Artista** (ID_Artista, Nombre, Género)
- **Álbum** (ID_Álbum, Nombre, ID_Artista, Fecha_Lanzamiento)
 - ID_Artista es clave foránea que referencia a Artista(ID_Artista).

- **Canción** (ID_Canción, Nombre, ID_Álbum, Duración)
 - ID_Álbum es clave foránea que referencia a Álbum(ID_Álbum).
- **Reproducción** (ID_Reproducción, ID_Usuario, ID_Canción, Fecha_Reproducción)
 - ID_Usuario es clave foránea que referencia a Usuario(ID_Usuario).
 - ID_Canción es clave foránea que referencia a Canción(ID_Canción).

4. Implementación en SQL

Creación de Tablas

| Query | Query History |
|-------|---|
| 1 | -- Crear tabla de Usuarios |
| 2 | CREATE TABLE Usuario (|
| 3 | id_usuario SERIAL PRIMARY KEY, |
| 4 | nombre VARCHAR(100) NOT NULL, |
| 5 | correo VARCHAR(100) NOT NULL, |
| 6 | fecha_registro DATE NOT NULL |
| 7 |); |
| 8 | |
| 9 | -- Crear tabla de Artistas |
| 10 | CREATE TABLE Artista (|
| 11 | id_artista SERIAL PRIMARY KEY, |
| 12 | nombre VARCHAR(100) NOT NULL, |
| 13 | genero VARCHAR(50) NOT NULL |
| 14 |); |
| 15 | |
| 16 | -- Crear tabla de Álbumes |
| 17 | CREATE TABLE Album (|
| 18 | id_album SERIAL PRIMARY KEY, |
| 19 | nombre VARCHAR(100) NOT NULL, |
| 20 | id_artista INT NOT NULL, |
| 21 | fecha_lanzamiento DATE NOT NULL, |
| 22 | FOREIGN KEY (id_artista) REFERENCES Artista(id_artista) |
| 23 |); |
| 24 | |
| 25 | -- Crear tabla de Canciones |
| 26 | CREATE TABLE Cancion (|
| 27 | id_cancion SERIAL PRIMARY KEY, |
| 28 | nombre VARCHAR(100) NOT NULL, |
| 29 | id_album INT NOT NULL, |
| 30 | duracion TIME NOT NULL, |
| 31 | id_usuario INT, -- Relacionamos al usuario con la canción |
| 32 | FOREIGN KEY (id_album) REFERENCES Album(id_album), |
| 33 | FOREIGN KEY (id_usuario) REFERENCES Usuario(id_usuario) -- Establecemos la relación con Usuario |
| 34 |); |

Inserción de Datos

| Query | Query History |
|-------|---|
| 42 | -- Insertar datos en Usuarios |
| 43 | ✓ INSERT INTO Usuario (nombre, correo, fecha_registro) VALUES |
| 44 | ('Juan Pérez', 'juan@mail.com', '2024-01-01'), |
| 45 | ('Ana López', 'ana@mail.com', '2024-02-15'), |
| 46 | ('Luis Gómez', 'luis@mail.com', '2024-03-20'), |
| 47 | ('Marta Rodríguez', 'marta@mail.com', '2024-04-10'), |
| 48 | ('Carlos Fernández', 'carlos@mail.com', '2024-05-05'), |
| 49 | ('Gabriela Sánchez', 'gabriela@mail.com', '2024-06-30'), |
| 50 | ('Oscar Ramírez', 'oscar@mail.com', '2024-07-25'), |
| 51 | ('Pedro Martínez', 'pedro@mail.com', '2024-08-10'), |
| 52 | ('Claudia Ruiz', 'claudia@mail.com', '2024-09-12'), |
| 53 | ('Sofía Gómez', 'sofia@mail.com', '2024-10-01'); |
| 54 | |
| 55 | -- Insertar datos en Artistas |
| 56 | ✓ INSERT INTO Artista (nombre, genero) VALUES |
| 57 | ('Shakira', 'Pop'), |
| 58 | ('Juanes', 'Rock'), |
| 59 | ('Carlos Vives', 'Vallenato'), |
| 60 | ('Bad Bunny', 'Reggaeton'), |
| 61 | ('Luis Fonsi', 'Pop'), |
| 62 | ('Maluma', 'Reggaeton'), |
| 63 | ('Pablo Alborán', 'Balada'), |
| 64 | ('Kany García', 'Pop'), |
| 65 | ('Ricardo Arjona', 'Balada'), |
| 66 | ('J Balvin', 'Reggaeton'); |
| 67 | |

```
68 -- Insertar datos en Álbumes
69 INSERT INTO Album (nombre, id_artista, fecha_lanzamiento) VALUES
70 ('El Dorado', 1, '2017-05-26'),
71 ('La Vida es un Ratico', 2, '2007-10-14'),
72 ('Cumbiana', 3, '2020-05-22'),
73 ('YHLQMDLG', 4, '2020-02-29'),
74 ('Despacito', 5, '2017-01-13'),
75 ('11:11', 6, '2019-11-09'),
76 ('Terra1', 7, '2015-10-23'),
77 ('Baila que Baila', 8, '2020-08-28'),
78 ('Viaje al Centro del Corazón', 9, '2019-11-29'),
79 ('Colores', 10, '2020-03-20');
80
81 -- Insertar datos en Canciones
82 INSERT INTO Cancion (nombre, id_album, duracion) VALUES
83 ('Chantaje', 1, '03:11'),
84 ('A Dios le Pido', 2, '03:39'),
85 ('La Bicicleta', 3, '03:47'),
86 ('Vete', 4, '03:21'),
87 ('Despacito', 5, '04:41'),
88 ('Felices los 4', 6, '03:56'),
89 ('Por Fin', 7, '03:42'),
90 ('Hasta Hoy', 8, '04:10'),
91 ('Fuiste Tú', 9, '04:11'),
92 ('Mi Gente', 10, '03:09');
```

5. Consultas SQL

- **Consulta requerida**

```
98 -- Listar las canciones de un artista específico
99 SELECT
100     c.nombre AS cancion,
101     a.nombre AS album,
102     ar.nombre AS artista
103 FROM Cancion c
104 JOIN Album a ON c.id_album = a.id_album
105 JOIN Artista ar ON a.id_artista = ar.id_artista
106 WHERE ar.nombre = 'Shakira';
107
```

| | cancion character varying (100) | album character varying (100) | artista character varying (100) |
|---|------------------------------------|----------------------------------|------------------------------------|
| 1 | Chantaj | El Dorado | Shakira |

- Consulta SELECT

Query

Query History

```

113 -- Listar las canciones, junto con su álbum y artista
114 SELECT
115     c.nombre AS cancion,
116     a.nombre AS album,
117     ar.nombre AS artista
118 FROM Cancion c
119 JOIN Album a ON c.id_album = a.id_album
120 JOIN Artista ar ON a.id_artista = ar.id_artista;
121

```

Data Output

Messages

Notifications

≡+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

| | cancion character varying (100) 🔒 | album character varying (100) 🔒 | artista character varying (100) 🔒 |
|----|--------------------------------------|------------------------------------|--------------------------------------|
| 1 | Chantaje | El Dorado | Shakira |
| 2 | A Dios le Pido | La Vida es un Ratico | Juanes |
| 3 | La Bicicleta | Cumbiana | Carlos Vives |
| 4 | Vete | YHLQMDLG | Bad Bunny |
| 5 | Despacito | Despacito | Luis Fonsi |
| 6 | Felices los 4 | 11:11 | Maluma |
| 7 | Por Fin | Terral | Pablo Alborán |
| 8 | Hasta Hoy | Baila que Baila | Kany García |
| 9 | Fuiste Tú | Viaje al Centro del Corazón | Ricardo Arjona |
| 10 | Mi Gente | Colores | J Balvin |

- Consulta WHERE

Query

Query History

123

-- Filtrar canciones de un álbum específico

124

SELECT

125

c.nombre AS cancion,

126

a.nombre AS album,

127

ar.nombre AS artista

128

FROM Cancion c

129

JOIN Album a ON c.id_album = a.id_album

130

JOIN Artista ar ON a.id_artista = ar.id_artista

131

WHERE a.nombre = 'El Dorado';

Data Output

Messages

Notifications

≡+

▼

▼

SQL

| | <div>cancion</div> <div>character varying (100)</div> <div></div> | <div>album</div> <div>character varying (100)</div> <div></div> | <div>artista</div> <div>character varying (100)</div> <div></div> |
|---|---|---|---|
| 1 | Chantaje | El Dorado | Shakira |

- Consulta ORDER BY

Query

Query History

```

134 -- Ordenar las canciones por nombre
135 SELECT
136     c.nombre AS cancion,
137     a.nombre AS album,
138     ar.nombre AS artista
139 FROM Cancion c
140 JOIN Album a ON c.id_album = a.id_album
141 JOIN Artista ar ON a.id_artista = ar.id_artista
142 ORDER BY c.nombre ASC;

```

Data Output

Messages

Notifications

+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

| | cancion character varying (100) 🔒 | album character varying (100) 🔒 | artista character varying (100) 🔒 |
|----|--------------------------------------|------------------------------------|--------------------------------------|
| 1 | A Dios le Pido | La Vida es un Ratico | Juanes |
| 2 | Chantaje | El Dorado | Shakira |
| 3 | Despacito | Despacito | Luis Fonsi |
| 4 | Felices los 4 | 11:11 | Maluma |
| 5 | Fuiste Tú | Viaje al Centro del Corazón | Ricardo Arjona |
| 6 | Hasta Hoy | Baila que Baila | Kany García |
| 7 | La Bicicleta | Cumbiana | Carlos Vives |
| 8 | Mi Gente | Colores | J Balvin |
| 9 | Por Fin | Terral | Pablo Alborán |
| 10 | Vete | YHLQMDLG | Bad Bunny |

- Consulta GROUP BY

```
145 -- Contar el número de canciones por cada álbum
146 SELECT
147     a.nombre AS album,
148     COUNT(c.id_cancion) AS total_canciones
149 FROM Album a
150 JOIN Cancion c ON a.id_album = c.id_album
151 GROUP BY a.id_album;
152
```

Data Output Messages Notifications

| | album character varying (100) 🔒 | total_canciones bigint 🔒 |
|----|------------------------------------|-----------------------------|
| 1 | Despacito | 1 |
| 2 | YHLQMDLG | 1 |
| 3 | Colores | 1 |
| 4 | 11:11 | 1 |
| 5 | La Vida es un Ratico | 1 |
| 6 | Terral | 1 |
| 7 | El Dorado | 1 |
| 8 | Baila que Baila | 1 |
| 9 | Viaje al Centro del Corazón | 1 |
| 10 | Cumbiana | 1 |

- Consulta JOIN

```

154 -- Listar las canciones con su respectivo artista y álbum
155 SELECT
156     c.nombre AS cancion,
157     a.nombre AS album,
158     ar.nombre AS artista
159 FROM Cancion c
160 JOIN Album a ON c.id_album = a.id_album
161 JOIN Artista ar ON a.id_artista = ar.id_artista;
162

```

Data Output Messages Notifications

SQL

| | cancion character varying (100) | album character varying (100) | artista character varying (100) |
|----|------------------------------------|----------------------------------|------------------------------------|
| 1 | Chantaje | El Dorado | Shakira |
| 2 | A Dios le Pido | La Vida es un Ratico | Juanes |
| 3 | La Bicicleta | Cumbiana | Carlos Vives |
| 4 | Vete | YHLQMDLG | Bad Bunny |
| 5 | Despacito | Despacito | Luis Fonsi |
| 6 | Felices los 4 | 11:11 | Maluma |
| 7 | Por Fin | Terral | Pablo Alborán |
| 8 | Hasta Hoy | Baila que Baila | Kany García |
| 9 | Fuiste Tú | Viaje al Centro del Corazón | Ricardo Arjona |
| 10 | Mi Gente | Colores | J Balvin |

- **Consulta LIKE**

```

Query  Query History
164  -- Buscar canciones que contengan 'Despacito' en el nombre
165  SELECT
166      c.nombre AS cancion,
167      a.nombre AS album,
168      ar.nombre AS artista
169  FROM Cancion c
170  JOIN Album a ON c.id_album = a.id_album
171  JOIN Artista ar ON a.id_artista = ar.id_artista
172  WHERE c.nombre LIKE '%Despacito%';

```

Data Output Messages Notifications

| | cancion character varying (100) | album character varying (100) | artista character varying (100) |
|---|------------------------------------|----------------------------------|------------------------------------|
| 1 | Despacito | Despacito | Luis Fonsi |

Sistema de Control de Proyectos

1. Definición de Entidades y Relaciones

Para la gestión de proyectos y tareas dentro de una empresa, se identifican las siguientes entidades clave:

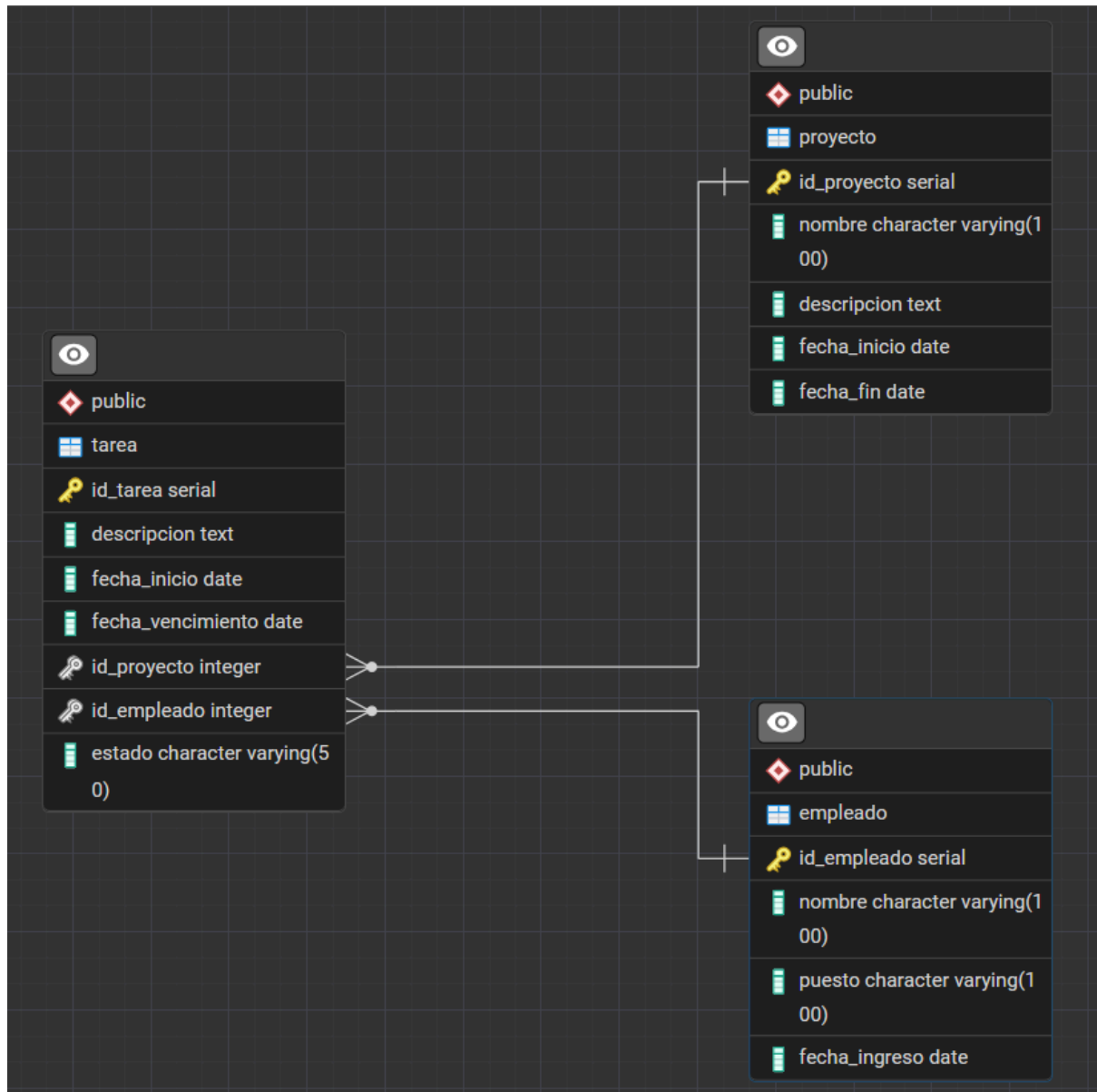
- **Empleado:** Representa a los trabajadores de la empresa. Tiene atributos como ID, nombre, puesto y fecha de ingreso.
- **Proyecto:** Representa los proyectos en desarrollo. Contiene atributos como ID, nombre, descripción, fecha de inicio y fecha de finalización.
- **Tarea:** Representa las actividades asignadas a los empleados dentro de un proyecto. Tiene atributos como ID, descripción, fecha de inicio, fecha de vencimiento, estado, ID del proyecto y ID del empleado.

Las relaciones entre estas entidades son las siguientes:

- Un empleado puede estar asignado a múltiples tareas, pero una tarea solo puede estar asignada a un empleado.

- Un proyecto puede tener varias tareas, pero cada tarea pertenece a un solo proyecto.
- Un proyecto puede estar relacionado con múltiples empleados a través de las tareas.

2. Diseño del Modelo Conceptual (Diagrama E-R)



3. Conversión a Esquema Relacional

A partir del modelo E-R, se definen las siguientes tablas:

- **Empleado** (id_empleado, nombre, puesto, fecha_ingreso)
- **Proyecto** (id_proyecto, nombre, descripcion, fecha_inicio, fecha_fin)

- Tarea (id_tarea, descripcion, fecha_inicio, fecha_vencimiento, id_proyecto, id_empleado, estado)

Claves foráneas:

- **id_proyecto** en la tabla Tarea referencia a **Proyecto(id_proyecto)**.
- **id_empleado** en la tabla Tarea referencia a **Empleado(id_empleado)**.

4. Implementación en SQL

Creación de Tablas

| Query | Query History |
|-------|--|
| 5 | -- Crear tabla de Empleados |
| 6 | CREATE TABLE Empleado (|
| 7 | id_empleado SERIAL PRIMARY KEY, |
| 8 | nombre VARCHAR(100) NOT NULL, |
| 9 | puesto VARCHAR(100) NOT NULL, |
| 10 | fecha_ingreso DATE NOT NULL |
| 11 |); |
| 12 | |
| 13 | -- Crear tabla de Proyectos |
| 14 | CREATE TABLE Proyecto (|
| 15 | id_proyecto SERIAL PRIMARY KEY, |
| 16 | nombre VARCHAR(100) NOT NULL, |
| 17 | descripcion TEXT, |
| 18 | fecha_inicio DATE NOT NULL, |
| 19 | fecha_fin DATE |
| 20 |); |
| 21 | |
| 22 | -- Crear tabla de Tareas |
| 23 | CREATE TABLE Tarea (|
| 24 | id_tarea SERIAL PRIMARY KEY, |
| 25 | descripcion TEXT NOT NULL, |
| 26 | fecha_inicio DATE NOT NULL, |
| 27 | fecha_vencimiento DATE NOT NULL, |
| 28 | id_proyecto INT NOT NULL, |
| 29 | id_empleado INT NOT NULL, |
| 30 | estado VARCHAR(50) NOT NULL, -- Pendiente, En progreso, Completada |
| 31 | FOREIGN KEY (id_proyecto) REFERENCES Proyecto(id_proyecto), |
| 32 | FOREIGN KEY (id_empleado) REFERENCES Empleado(id_empleado) |
| 33 |); |
| 34 | |

Inserción de Datos

Query

Query History

```
39 -- Insertar datos en Empleados
40 INSERT INTO Empleado (nombre, puesto, fecha_ingreso) VALUES
41 ('Juan Pérez', 'Desarrollador', '2022-01-15'),
42 ('Ana López', 'Analista', '2023-03-22'),
43 ('Luis Gómez', 'Project Manager', '2021-11-10'),
44 ('Marta Rodríguez', 'Diseñadora', '2020-07-09'),
45 ('Carlos Fernández', 'Desarrollador', '2022-05-01'),
46 ('Gabriela Sánchez', 'Tester', '2023-02-10'),
47 ('Oscar Ramírez', 'Analista', '2020-10-17'),
48 ('Pedro Martínez', 'Desarrollador', '2022-06-12'),
49 ('Claudia Ruiz', 'Project Manager', '2021-08-25'),
50 ('Sofía Gómez', 'Tester', '2020-03-20');
51
52 -- Insertar datos en Proyectos
53 INSERT INTO Proyecto (nombre, descripcion, fecha_inicio, fecha_fin) VALUES
54 ('Proyecto A', 'Desarrollo de un sistema de gestión de inventarios', '2023-01-10', '2023-06-30'),
55 ('Proyecto B', 'Rediseño de la página web corporativa', '2023-02-01', '2023-05-15'),
56 ('Proyecto C', 'Aplicación móvil para ventas en línea', '2023-03-05', '2023-08-20'),
57 ('Proyecto D', 'Sistema de control de proyectos', '2023-04-15', '2023-09-30'),
58 ('Proyecto E', 'Desarrollo de plataforma de streaming', '2023-01-01', '2023-12-31'),
59 ('Proyecto F', 'Implementación de inteligencia artificial', '2023-05-25', '2024-05-25'),
60 ('Proyecto G', 'Automatización de procesos de negocio', '2023-06-10', '2023-11-10'),
61 ('Proyecto H', 'Desarrollo de una app de realidad aumentada', '2023-07-01', '2024-01-15'),
62 ('Proyecto I', 'Desarrollo de plataforma de e-commerce', '2023-02-20', '2023-07-30'),
63 ('Proyecto J', 'Migración a la nube', '2023-04-01', '2023-10-15');
64
65 -- Insertar datos en Tareas
66 INSERT INTO Tarea (descripcion, fecha_inicio, fecha_vencimiento, id_proyecto, id_empleado, estado) VALUES
67 ('Definir los requisitos del sistema', '2023-01-15', '2023-01-31', 1, 1, 'Pendiente'),
68 ('Desarrollar la interfaz de usuario', '2023-02-01', '2023-02-28', 2, 4, 'Pendiente'),
69 ('Establecer base de datos', '2023-03-01', '2023-03-15', 3, 2, 'En progreso'),
70 ('Rediseñar el sistema de pagos', '2023-04-10', '2023-04-25', 4, 3, 'Pendiente'),
71 ('Realizar pruebas de usabilidad', '2023-05-01', '2023-05-10', 5, 6, 'Completada'),
72 ('Implementar API de pagos', '2023-06-01', '2023-06-20', 6, 5, 'Pendiente'),
73 ('Automatizar pruebas de integración', '2023-07-01', '2023-07-15', 7, 9, 'Pendiente'),
74 ('Desarrollar el módulo de compras', '2023-08-01', '2023-08-20', 8, 7, 'Pendiente'),
75 ('Actualizar el sistema de inventarios', '2023-09-01', '2023-09-15', 9, 8, 'Pendiente'),
76 ('Realizar la migración de servidores', '2023-10-01', '2023-10-10', 10, 10, 'En progreso'),
77 ('Revisión de seguridad', '2025-03-24', '2025-03-31', 1, 1, 'Pendiente');
```

5. Consultas SQL

- Consulta requerida

```

84 -- Mostrar todas las tareas pendientes de un proyecto específico, ordenadas por fecha de vencimiento
85 SELECT
86     t.descripcion AS tarea,
87     t.fecha_vencimiento,
88     p.nombre AS proyecto,
89     e.nombre AS empleado
90 FROM Tarea t
91 JOIN Proyecto p ON t.id_proyecto = p.id_proyecto
92 JOIN Empleado e ON t.id_empleado = e.id_empleado
93 WHERE p.id_proyecto = 1 -- Cambiar el ID de proyecto según sea necesario
94 AND t.estado = 'Pendiente'
95 ORDER BY t.fecha_vencimiento ASC;

```

Data Output Messages Notifications

Showing rows: 1 to 4

| | tarea text | fecha_vencimiento date | proyecto character varying (100) | empleado character varying (100) |
|---|------------------------------------|---------------------------|-------------------------------------|-------------------------------------|
| 1 | Definir los requisitos del sistema | 2023-01-31 | Proyecto A | Juan Pérez |
| 2 | Definir los requisitos del sistema | 2023-01-31 | Proyecto A | Juan Pérez |
| 3 | Revisión de seguridad | 2025-03-31 | Proyecto A | Juan Pérez |
| 4 | Revisión de seguridad | 2025-03-31 | Proyecto A | Juan Pérez |

- Consulta SELECT

```

102 -- Selecciona todos los empleados en la empresa.
103 SELECT * FROM Empleado;

```

Data Output Messages Notifications

| | id_empleado [PK] integer | nombre character varying (100) | puesto character varying (100) | fecha_ingreso date |
|----|-----------------------------|-----------------------------------|-----------------------------------|-----------------------|
| 1 | 1 | Juan Pérez | Desarrollador | 2022-01-15 |
| 2 | 2 | Ana López | Analista | 2023-03-22 |
| 3 | 3 | Luis Gómez | Project Manager | 2021-11-10 |
| 4 | 4 | Marta Rodríguez | Diseñadora | 2020-07-09 |
| 5 | 5 | Carlos Fernández | Desarrollador | 2022-05-01 |
| 6 | 6 | Gabriela Sánchez | Tester | 2023-02-10 |
| 7 | 7 | Oscar Ramírez | Analista | 2020-10-17 |
| 8 | 8 | Pedro Martínez | Desarrollador | 2022-06-12 |
| 9 | 9 | Claudia Ruiz | Project Manager | 2021-08-25 |
| 10 | 10 | Sofía Gómez | Tester | 2020-03-20 |
| 11 | 11 | Juan Pérez | Desarrollador | 2022-01-15 |
| 12 | 12 | Ana López | Analista | 2023-03-22 |
| 13 | 13 | Luis Gómez | Project Manager | 2021-11-10 |

- Consulta WHERE

```
106 -- Muestra las tareas que están en estado "Pendiente".
107 SELECT * FROM Tarea WHERE estado = 'Pendiente';
108
```

| Data Output Messages Notifications | | | | | | | |
|------------------------------------|--------------------------|--------------------------------------|----------------------|---------------------------|------------------------|------------------------|----------------------------------|
| Showing rows: 1 to 16 | | | | | | | |
| | id_tarea [PK] integer | descripcion text | fecha_inicio date | fecha_vencimiento date | id_proyecto integer | id_empleado integer | estado character varying (50) |
| 1 | 1 | Definir los requisitos del sistema | 2023-01-15 | 2023-01-31 | 1 | 1 | Pendiente |
| 2 | 2 | Desarrollar la interfaz de usuario | 2023-02-01 | 2023-02-28 | 2 | 4 | Pendiente |
| 3 | 4 | Rediseñar el sistema de pagos | 2023-04-10 | 2023-04-25 | 4 | 3 | Pendiente |
| 4 | 6 | Implementar API de pagos | 2023-06-01 | 2023-06-20 | 6 | 5 | Pendiente |
| 5 | 7 | Automatizar pruebas de integraci... | 2023-07-01 | 2023-07-15 | 7 | 9 | Pendiente |
| 6 | 8 | Desarrollar el módulo de compras | 2023-08-01 | 2023-08-20 | 8 | 7 | Pendiente |
| 7 | 9 | Actualizar el sistema de inventarios | 2023-09-01 | 2023-09-15 | 9 | 8 | Pendiente |
| 8 | 11 | Revisión de seguridad | 2025-03-24 | 2025-03-31 | 1 | 1 | Pendiente |
| 9 | 12 | Definir los requisitos del sistema | 2023-01-15 | 2023-01-31 | 1 | 1 | Pendiente |
| 10 | 13 | Desarrollar la interfaz de usuario | 2023-02-01 | 2023-02-28 | 2 | 4 | Pendiente |
| 11 | 15 | Rediseñar el sistema de pagos | 2023-04-10 | 2023-04-25 | 4 | 3 | Pendiente |
| 12 | 17 | Implementar API de pagos | 2023-06-01 | 2023-06-20 | 6 | 5 | Pendiente |
| 13 | 18 | Automatizar pruebas de integraci... | 2023-07-01 | 2023-07-15 | 7 | 9 | Pendiente |

- Consulta ORDER BY

```
110 -- Lista los proyectos ordenados por fecha de inicio, del más reciente al más antiguo.
111 SELECT * FROM Proyecto ORDER BY fecha_inicio DESC;
112
```

| Data Output Messages Notifications | | | | | |
|------------------------------------|-----------------------------|-----------------------------------|---|----------------------|-------------------|
| Showing rows: 1 to 20 | | | | | |
| | id_proyecto [PK] integer | nombre character varying (100) | descripcion text | fecha_inicio date | fecha_fin date |
| 1 | 18 | Proyecto H | Desarrollo de una app de realidad aumentada | 2023-07-01 | 2024-01-15 |
| 2 | 8 | Proyecto H | Desarrollo de una app de realidad aumentada | 2023-07-01 | 2024-01-15 |
| 3 | 7 | Proyecto G | Automatización de procesos de negocio | 2023-06-10 | 2023-11-10 |
| 4 | 17 | Proyecto G | Automatización de procesos de negocio | 2023-06-10 | 2023-11-10 |
| 5 | 16 | Proyecto F | Implementación de inteligencia artificial | 2023-05-25 | 2024-05-25 |
| 6 | 6 | Proyecto F | Implementación de inteligencia artificial | 2023-05-25 | 2024-05-25 |
| 7 | 14 | Proyecto D | Sistema de control de proyectos | 2023-04-15 | 2023-09-30 |
| 8 | 4 | Proyecto D | Sistema de control de proyectos | 2023-04-15 | 2023-09-30 |
| 9 | 20 | Proyecto J | Migración a la nube | 2023-04-01 | 2023-10-15 |
| 10 | 10 | Proyecto J | Migración a la nube | 2023-04-01 | 2023-10-15 |
| 11 | 3 | Proyecto C | Aplicación móvil para ventas en línea | 2023-03-05 | 2023-08-20 |
| 12 | 13 | Proyecto C | Aplicación móvil para ventas en línea | 2023-03-05 | 2023-08-20 |
| 13 | 9 | Proyecto I | Desarrollo de plataforma de e-commerce | 2023-02-20 | 2023-07-30 |

- Consulta GROUP BY

```

114 -- Cuenta cuántas tareas tiene cada proyecto.
115 SELECT id_proyecto, COUNT(*) AS total_tareas FROM Tarea GROUP BY id_proyecto;
116

```

| Data Output | | | Messages | Notifications |
|-------------|------------------------|------------------------|----------|---------------|
| | id_proyecto integer | total_tareas bigint | | |
| 1 | 9 | 2 | | |
| 2 | 3 | 2 | | |
| 3 | 5 | 2 | | |
| 4 | 4 | 2 | | |
| 5 | 10 | 2 | | |
| 6 | 6 | 2 | | |
| 7 | 2 | 2 | | |
| 8 | 7 | 2 | | |
| 9 | 1 | 4 | | |
| 10 | 8 | 2 | | |

- Consulta JOIN

```

118 -- Muestra las tareas junto con el nombre del empleado y el proyecto correspondiente.
119 SELECT
120     t.descripcion AS tarea,
121     t.fecha_vencimiento,
122     p.nombre AS proyecto,
123     e.nombre AS empleado
124 FROM Tarea t
125 JOIN Proyecto p ON t.id_proyecto = p.id_proyecto
126 JOIN Empleado e ON t.id_empleado = e.id_empleado;

```

| Data Output | | | | | Messages | Notifications |
|-------------|--------------------------------------|---------------------------|-------------------------------------|-------------------------------------|----------|---------------|
| | tarea text | fecha_vencimiento date | proyecto character varying (100) | empleado character varying (100) | | |
| 1 | Definir los requisitos del sistema | 2023-01-31 | Proyecto A | Juan Pérez | | |
| 2 | Desarrollar la interfaz de usuario | 2023-02-28 | Proyecto B | Marta Rodríguez | | |
| 3 | Establecer base de datos | 2023-03-15 | Proyecto C | Ana López | | |
| 4 | Rediseñar el sistema de pagos | 2023-04-25 | Proyecto D | Luis Gómez | | |
| 5 | Realizar pruebas de usabilidad | 2023-05-10 | Proyecto E | Gabriela Sánchez | | |
| 6 | Implementar API de pagos | 2023-06-20 | Proyecto F | Carlos Fernández | | |
| 7 | Automatizar pruebas de integraci... | 2023-07-15 | Proyecto G | Claudia Ruiz | | |
| 8 | Desarrollar el módulo de compras | 2023-08-20 | Proyecto H | Oscar Ramírez | | |
| 9 | Actualizar el sistema de inventarios | 2023-09-15 | Proyecto I | Pedro Martínez | | |
| 10 | Realizar la migración de servidores | 2023-10-10 | Proyecto J | Sofía Gómez | | |
| 11 | Revisión de seguridad | 2025-03-31 | Proyecto A | Juan Pérez | | |
| 12 | Definir los requisitos del sistema | 2023-01-31 | Proyecto A | Juan Pérez | | |

- **Consulta LIKE**

129

-- Busca empleados cuyo nombre comience con "C".

130

SELECT * FROM Empleado WHERE nombre LIKE 'C%';|

Data Output

Messages

Notifications

≡+

▼

▼

SQL

| | id_empleado [PK] integer | nombre character varying (100) | puesto character varying (100) | fecha_ingreso date |
|---|-----------------------------|-----------------------------------|-----------------------------------|-----------------------|
| 1 | 5 | Carlos Fernández | Desarrollador | 2022-05-01 |
| 2 | 9 | Claudia Ruiz | Project Manager | 2021-08-25 |
| 3 | 15 | Carlos Fernández | Desarrollador | 2022-05-01 |
| 4 | 19 | Claudia Ruiz | Project Manager | 2021-08-25 |

Sistema de Evaluación Académica

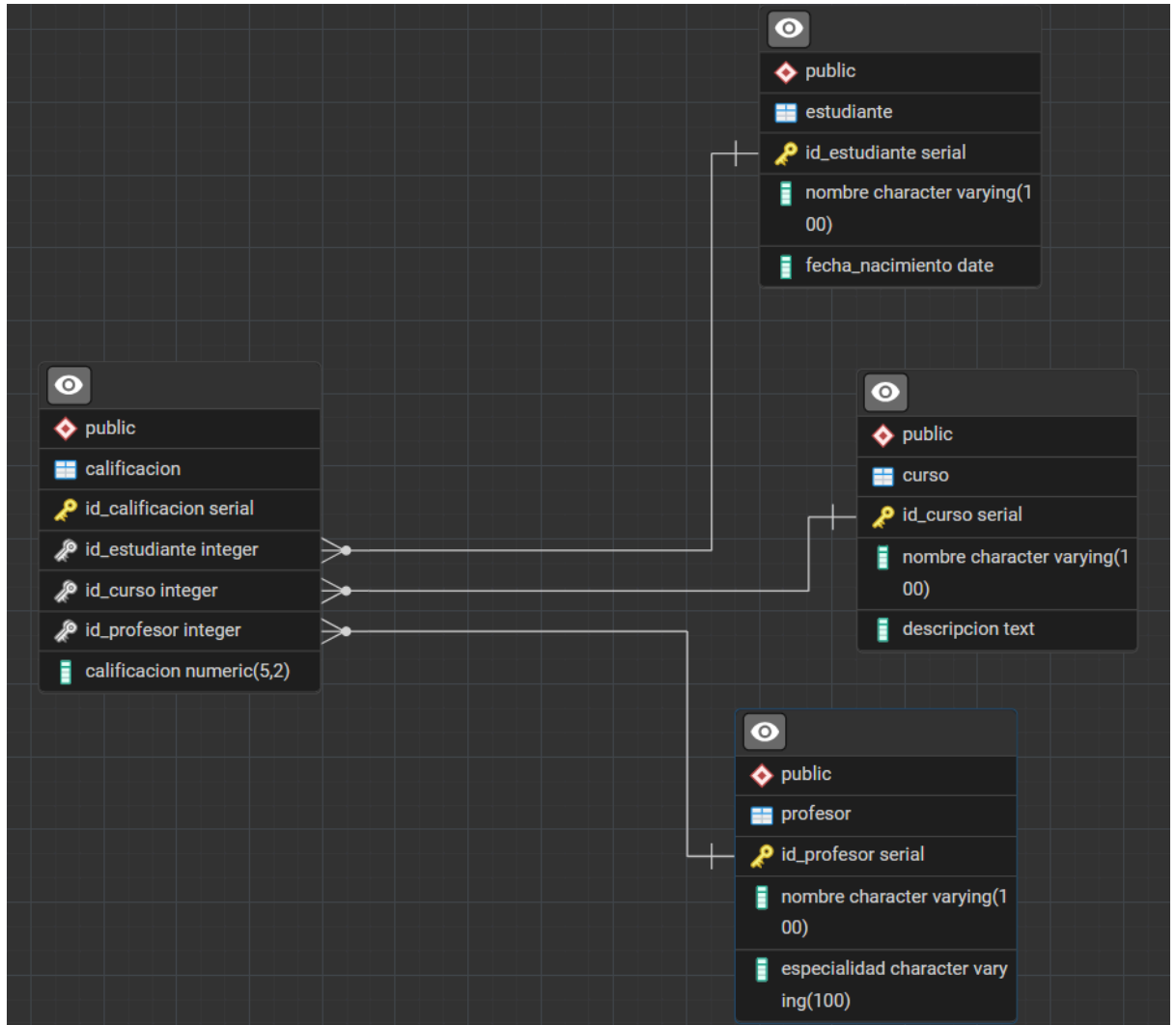
1. **Definición de Entidades y Relaciones** Para la gestión de música en una plataforma, se identifican las siguientes entidades clave:
 - **Usuario:** Representa a los usuarios registrados en la plataforma. Tiene atributos como ID de usuario, nombre, correo electrónico y fecha de registro.
 - **Artista:** Representa a los artistas que crean la música. Incluye atributos como ID de artista, nombre y género musical.
 - **Álbum:** Representa los álbumes musicales. Tiene atributos como ID de álbum, nombre, fecha de lanzamiento, y la relación con el artista (ID de artista).
 - **Canción:** Representa las canciones dentro de los álbumes. Incluye atributos como ID de canción, nombre, duración, y la relación con el álbum (ID de álbum).
 - **Reproducción:** Representa las canciones reproducidas por los usuarios. Incluye atributos como ID de reproducción, ID de usuario, ID de canción y fecha de reproducción.

Las relaciones entre estas entidades son las siguientes:

- Un **artista** puede tener varios **álbumes**, pero un **álbum** está relacionado con un solo **artista**.
- Un **álbum** puede tener varias **canciones**, pero cada **canción** está asociada con un solo **álbum**.

- Un **usuario** puede reproducir varias **canciones**, y una **canción** puede ser reproducida por varios **usuarios** (relación muchos a muchos entre **usuario** y **canción**).

2. Diseño del Modelo Conceptual (Diagrama E-R)



3. Conversión a Esquema Relacional A partir del modelo E-R, se definen las siguientes tablas:

- **Usuario** (ID_usuario, Nombre, Correo Electrónico, Fecha_registro)
- **Artista** (ID_artista, Nombre, Género)
- **Álbum** (ID_album, Nombre, Fecha_lanzamiento, ID_artista)
 - **ID_artista** es clave foránea que referencia a **Artista** (ID_artista).
- **Canción** (ID_cancion, Nombre, Duración, ID_album)

- **ID_album** es clave foránea que referencia a **Álbum** (ID_album).
- **Reproducción** (ID_reproduccion, ID_usuario, ID_cancion, Fecha_reproduccion)
 - **ID_usuario** es clave foránea que referencia a **Usuario** (ID_usuario).
 - **ID_cancion** es clave foránea que referencia a **Canción** (ID_cancion).

4. Implementación en SQL

Creación de Tablas

| Query | Query History |
|-------------|---|
| 6 | CREATE TABLE Estudiante (|
| 7 | id_estudiante SERIAL PRIMARY KEY, |
| 8 | nombre VARCHAR(100) NOT NULL, |
| 9 | fecha_nacimiento DATE NOT NULL |
| 10 |); |
| 11 | |
| 12 | -- Crear tabla de Cursos |
| 13 | CREATE TABLE Curso (|
| 14 | id_curso SERIAL PRIMARY KEY, |
| 15 | nombre VARCHAR(100) NOT NULL, |
| 16 | descripcion TEXT |
| 17 |); |
| 18 | |
| 19 | -- Crear tabla de Profesores |
| 20 | CREATE TABLE Profesor (|
| 21 | id_profesor SERIAL PRIMARY KEY, |
| 22 | nombre VARCHAR(100) NOT NULL, |
| 23 | especialidad VARCHAR(100) |
| 24 |); |
| 25 | |
| 26 | -- Crear tabla de Calificaciones |
| 27 | CREATE TABLE Calificacion (|
| 28 | id_calificacion SERIAL PRIMARY KEY, |
| 29 | id_estudiante INT NOT NULL, |
| 30 | id_curso INT NOT NULL, |
| 31 | id_profesor INT NOT NULL, |
| 32 | calificacion DECIMAL(5,2) NOT NULL, |
| 33 | FOREIGN KEY (id_estudiante) REFERENCES Estudiante(id_estudiante), |
| 34 | FOREIGN KEY (id_curso) REFERENCES Curso(id_curso), |
| 35 | FOREIGN KEY (id_profesor) REFERENCES Profesor(id_profesor) |
| 36 |); |
| Total rows: | Query complete 00:00:00.081 |

Inserción de Datos

```
42 -- Insertar datos en Estudiantes
43 v INSERT INTO Estudiante (nombre, fecha_nacimiento) VALUES
44 ('Carlos López', '2001-02-15'),
45 ('Ana Sánchez', '2000-11-21'),
46 ('Juan Pérez', '2001-05-10'),
47 ('María García', '2000-08-03'),
48 ('Pedro Martínez', '2001-07-12'),
49 ('Sofía Rodríguez', '2001-03-20'),
50 ('Luis González', '2000-12-30'),
51 ('Claudia Ruiz', '2000-09-25'),
52 ('Carlos Ramírez', '2001-01-17'),
53 ('Laura Torres', '2001-06-14');
54
55 -- Insertar datos en Cursos
56 v INSERT INTO Curso (nombre, descripcion) VALUES
57 ('Matemáticas I', 'Curso introductorio de álgebra y geometría'),
58 ('Física I', 'Introducción a la física clásica y sus leyes'),
59 ('Química I', 'Estudio de la química básica y sus aplicaciones'),
60 ('Historia Universal', 'Estudio de la historia desde la prehistoria hasta la Edad Media'),
61 ('Lengua Española', 'Estudio del idioma español y su gramática'),
62 ('Programación Básica', 'Curso de introducción a la programación en Java'),
63 ('Economía', 'Conceptos básicos de economía y mercado'),
64 ('Biología', 'Estudio de la biología celular y molecular'),
65 ('Literatura', 'Estudio de los grandes autores literarios'),
66 ('Geografía', 'Estudio de la geografía mundial y local');
67
```

Query Query History

```
68 -- Insertar datos en Profesores
69 v INSERT INTO Profesor (nombre, especialidad) VALUES
70 ('Dr. José Pérez', 'Matemáticas'),
71 ('Lic. Marta Gómez', 'Física'),
72 ('Dr. Luis Hernández', 'Química'),
73 ('Prof. Andrés Díaz', 'Historia'),
74 ('Dr. Roberto Martínez', 'Lengua Española'),
75 ('Ing. Ana García', 'Programación'),
76 ('Lic. Felipe Castro', 'Economía'),
77 ('Dr. Clara Jiménez', 'Biología'),
78 ('Prof. Luis Martín', 'Literatura'),
79 ('Lic. Elena López', 'Geografía');
80
81 -- Insertar datos en Calificaciones
82 v INSERT INTO Calificacion (id_estudiante, id_curso, id_profesor, calificacion) VALUES
83 (1, 1, 1, 8.5),
84 (1, 2, 2, 9.0),
85 (1, 3, 3, 7.5),
86 (1, 4, 4, 8.0),
87 (1, 5, 5, 9.5),
88 (2, 1, 1, 7.0),
89 (2, 2, 2, 6.5),
90 (2, 3, 3, 8.0),
91 (2, 4, 4, 7.5),
92 (2, 5, 5, 8.0),
93 (3, 1, 1, 8.0),
94 (3, 2, 2, 7.0),
95 (3, 3, 3, 9.0),
96 (3, 4, 4, 8.5),
97 (3, 5, 5, 9.0),
98 (4, 1, 1, 6.5).
```

Total rows: Query complete 00:00:00.081

5. Consultas SQL

- Consulta requerida

```
139 -- Obtener el promedio de calificaciones de un estudiante en todos sus cursos
140 SELECT
141     e.nombre AS estudiante,
142     ROUND(AVG(c.calificacion), 2) AS promedio_calificacion
143 FROM Calificacion c
144 JOIN Estudiante e ON c.id_estudiante = e.id_estudiante
145 WHERE e.id_estudiante = 1 -- Cambiar el ID de estudiante según sea necesario
146 GROUP BY e.id_estudiante;
```

Data Output Messages Notifications

| | estudiante character varying (100) | promedio_calificacion numeric |
|---|---------------------------------------|----------------------------------|
| 1 | Carlos López | 8.62 |

- Consulta SELECT

```
153 -- Muestra todos los estudiantes registrados en la base de datos.
154 SELECT * FROM Estudiante;
```

Data Output Messages Notifications

| | id_estudiante [PK] integer | nombre character varying (100) | fecha_nacimiento date |
|----|-------------------------------|-----------------------------------|--------------------------|
| 1 | 1 | Carlos López | 2001-02-15 |
| 2 | 2 | Ana Sánchez | 2000-11-21 |
| 3 | 3 | Juan Pérez | 2001-05-10 |
| 4 | 4 | María García | 2000-08-03 |
| 5 | 5 | Pedro Martínez | 2001-07-12 |
| 6 | 6 | Sofía Rodríguez | 2001-03-20 |
| 7 | 7 | Luis González | 2000-12-30 |
| 8 | 8 | Claudia Ruiz | 2000-09-25 |
| 9 | 9 | Carlos Ramírez | 2001-01-17 |
| 10 | 10 | Laura Torres | 2001-06-14 |

- Consulta WHERE

```

157 -- Muestra los cursos de la base de datos que tienen "Matemáticas" en su nombre.
158 SELECT * FROM Curso WHERE nombre LIKE '%Matemáticas%';

```

Data Output Messages Notifications

SQL

| | id_curso [PK] integer | nombre character varying (100) | descripcion text |
|---|--------------------------|-----------------------------------|--|
| 1 | 1 | Matemáticas I | Curso introductorio de álgebra y geometría |

- Consulta ORDER BY

```

161 -- Lista las calificaciones ordenadas de mayor a menor.
162 SELECT * FROM Calificacion ORDER BY calificacion DESC;

```

Data Output Messages Notifications

SQL

| | id_calificacion [PK] integer | id_estudiante integer | id_curso integer | id_profesor integer | calificacion numeric (5,2) |
|----|---------------------------------|--------------------------|---------------------|------------------------|-------------------------------|
| 1 | 30 | 6 | 5 | 5 | 9.50 |
| 2 | 48 | 10 | 3 | 3 | 9.50 |
| 3 | 20 | 4 | 5 | 5 | 9.50 |
| 4 | 36 | 8 | 1 | 1 | 9.50 |
| 5 | 5 | 1 | 5 | 5 | 9.50 |
| 6 | 51 | 1 | 6 | 6 | 9.20 |
| 7 | 13 | 3 | 3 | 3 | 9.00 |
| 8 | 2 | 1 | 2 | 2 | 9.00 |
| 9 | 15 | 3 | 5 | 5 | 9.00 |
| 10 | 21 | 5 | 1 | 1 | 9.00 |
| 11 | 25 | 5 | 5 | 5 | 9.00 |
| 12 | 49 | 10 | 4 | 4 | 9.00 |

Total rows: 51 | Query complete 00:00:00.167

- **Consulta GROUP BY**

```

165 -- Muestra el promedio de calificaciones por curso.
166 SELECT id_curso, ROUND(AVG(calificacion), 2) AS promedio FROM Calificacion GROUP BY id_curso;

```

Data Output Messages Notifications

Showing rows: 1 to 6

| | id_curso integer | promedio numeric |
|---|---------------------|---------------------|
| 1 | 6 | 9.20 |
| 2 | 1 | 8.05 |
| 3 | 3 | 8.05 |
| 4 | 5 | 8.80 |
| 5 | 4 | 7.95 |
| 6 | 2 | 8.05 |

- **Consulta JOIN**

```

169 -- Muestra las calificaciones con el nombre del estudiante, el curso y el profesor.
170 SELECT
171     e.nombre AS estudiante,
172     cu.nombre AS curso,
173     p.nombre AS profesor,
174     c.calificacion
175 FROM Calificacion c
176 JOIN Estudiante e ON c.id_estudiante = e.id_estudiante
177 JOIN Curso cu ON c.id_curso = cu.id_curso
178 JOIN Profesor p ON c.id_profesor = p.id_profesor;
179

```














Data Output Messages Notifications

| | estudiante character varying (100) | curso character varying (100) | profesor character varying (100) | calificacion numeric (5,2) |
|----|---------------------------------------|----------------------------------|-------------------------------------|-------------------------------|
| 1 | Carlos López | Matemáticas I | Dr. José Pérez | 8.50 |
| 2 | Carlos López | Física I | Lic. Marta Gómez | 9.00 |
| 3 | Carlos López | Química I | Dr. Luis Hernández | 7.50 |
| 4 | Carlos López | Historia Universal | Prof. Andrés Díaz | 8.00 |
| 5 | Carlos López | Lengua Española | Dr. Roberto Martínez | 9.50 |
| 6 | Ana Sánchez | Matemáticas I | Dr. José Pérez | 7.00 |
| 7 | Ana Sánchez | Física I | Lic. Marta Gómez | 6.50 |
| 8 | Ana Sánchez | Química I | Dr. Luis Hernández | 8.00 |
| 9 | Ana Sánchez | Historia Universal | Prof. Andrés Díaz | 7.50 |
| 10 | Ana Sánchez | Lengua Española | Dr. Roberto Martínez | 8.00 |
| 11 | Juan Pérez | Matemáticas I | Dr. José Pérez | 8.00 |
| 12 | Juan Pérez | Física I | Lic. Marta Gómez | 7.00 |
| 13 | Juan Pérez | Química I | Dr. Luis Hernández | 9.00 |
| 14 | Juan Pérez | Historia Universal | Prof. Andrés Díaz | 8.50 |

- Consulta LIKE

```
181 -- Busca profesores cuya especialidad contenga la palabra "Historia".
182 SELECT * FROM Profesor WHERE especialidad LIKE '%Historia%';
183
```

Data Output Messages Notifications

| |           | | |
|---|---|---|---|
| | id_profesor [PK] integer  | nombre character varying (100)  | especialidad character varying (100)  |
| 1 | 4 | Prof. Andrés Díaz | Historia |