

INSTITUTO TECNOLOGICO DE MEXICALI

Carrera:
ING. en Sistemas.

Materia:
Fundamento de base de datos.

Alumno:
Marin Salazar Juan Sebastian 22490423.

Correo Institucional:
a22490423@itmexicali.edu.mx

Profesor:
Jose Ramon Bogarin Valenzuela

Fecha de entrega:
24 de Marzo del 2025

Mexicali, Baja California a 24 de Marzo del 2025.

Sistema de gestión de inventarios

Una empresa desea controlar su inventario de productos y proveedores.

- **Identificar entidades claves: Producto, Proveedor, Categoría, Inventario.**

Tenemos las entidades:

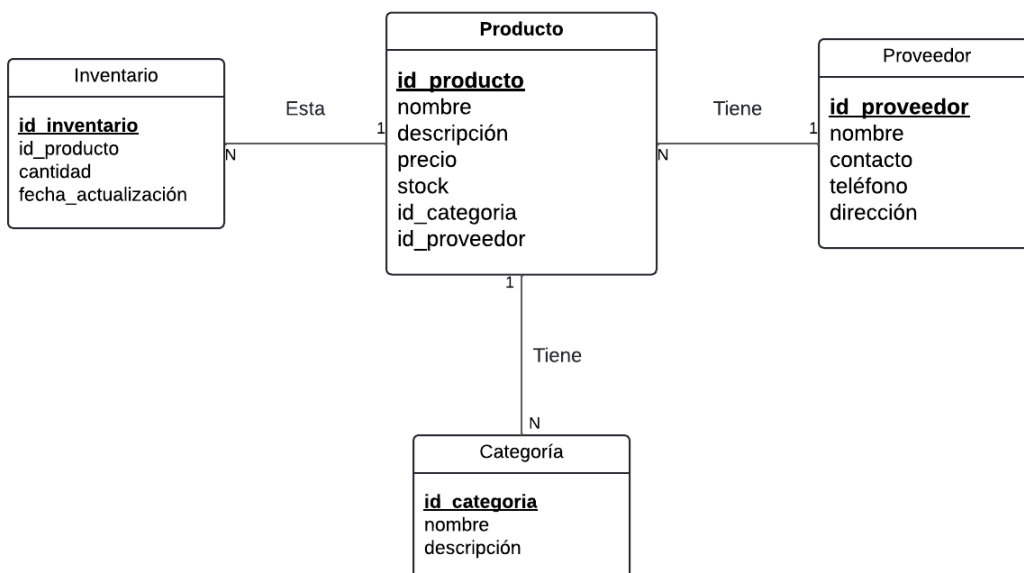
Producto: Un producto pertenece a una categoría y un producto tiene un proveedor.

Proveedor: Un proveedor puede tener múltiples productos.

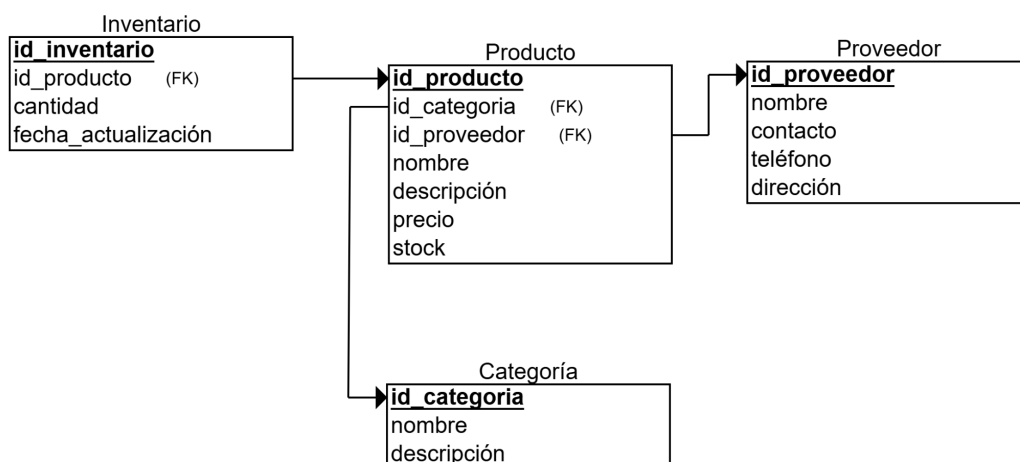
Categoría: Una categoría agrupa múltiples productos.

Inventario: Registra los movimientos de stock de un producto.

- **Diseñar el modelo E-R.**



- **Transformar el modelo en un esquema relacional.**



- **Implementar la base de datos en SQL.**

--Crear tabla Proveedor

```
CREATE TABLE Proveedor (  
    id_proveedor SERIAL PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    contacto VARCHAR(100),  
    telefono VARCHAR(20),  
    direccion TEXT  
);
```

--Crear tabla Categoría

```
CREATE TABLE Categoria (  
    id_categoria SERIAL PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    descripcion TEXT  
);
```

--Crear tabla Producto

```
CREATE TABLE Producto (  
    id_producto SERIAL PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    descripcion TEXT,  
    precio DECIMAL(10,2) NOT NULL,  
    stock INT NOT NULL DEFAULT 0,  
    id_categoria INT NOT NULL REFERENCES Categoria(id_categoria) ON DELETE  
CASCADE,  
    id_proveedor INT NOT NULL REFERENCES Proveedor(id_proveedor) ON DELETE  
SET NULL  
);
```

--Crear tabla Inventario

```
CREATE TABLE Inventario (  
    id_inventario SERIAL PRIMARY KEY,  
    id_producto INT NOT NULL REFERENCES Producto(id_producto) ON DELETE  
CASCADE,  
    cantidad INT NOT NULL,  
    fecha_actualizacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

• Consultas a hacer.

SELECT * FROM Producto;

Query Query History

```
1 SELECT * FROM Producto;
```

Data Output Messages Notifications

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

	id_producto [PK] integer	nombre character varying (100)	descripcion text	precio numeric (10,2)	stock integer	id_categoria integer	id_proveedor integer
1	1	Laptop X	Laptop con procesador i7 y 16GB RAM	1500.00	20	1	1
2	2	Silla Ergonómica	Silla de oficina cómoda	200.00	15	6	2
3	3	Teclado Mecánico	Teclado RGB con switches mecánicos	80.00	30	5	3
4	4	Software Antivirus	Licencia de antivirus por 1 año	50.00	100	4	4
5	5	Router WiFi 6	Router de alta velocidad	120.00	25	9	5
6	6	Monitor 27"	Monitor 4K de 27 pulgadas	350.00	10	1	6
7	7	SSD 1TB	Unidad de estado sólido de 1TB	110.00	50	8	7
8	8	Cámara de Seguridad	Cámara con visión nocturna	75.00	40	7	8
9	9	Multifuncional Láser	Impresora y escáner en un solo dispositivo	250.00	12	2	9
10	10	Microondas Digital	Horno microondas de 1200W	180.00	8	10	10

SELECT * FROM Producto WHERE precio > 100;

Query Query History

```
1 SELECT * FROM Producto WHERE precio > 100;
```

Data Output Messages Notifications

Showing rows: 1 to 7 Page No: 1 of 1

	id_producto [PK] integer	nombre character varying (100)	descripcion text	precio numeric (10,2)	stock integer	id_categoria integer	id_proveedor integer
1	1	Laptop X	Laptop con procesador i7 y 16GB RAM	1500.00	20	1	1
2	2	Silla Ergonómica	Silla de oficina cómoda	200.00	15	6	2
3	5	Router WiFi 6	Router de alta velocidad	120.00	25	9	5
4	6	Monitor 27"	Monitor 4K de 27 pulgadas	350.00	10	1	6
5	7	SSD 1TB	Unidad de estado sólido de 1TB	110.00	50	8	7
6	9	Multifuncional Láser	Impresora y escáner en un solo dispositivo	250.00	12	2	9
7	10	Microondas Digital	Horno microondas de 1200W	180.00	8	10	10

SELECT * FROM Producto ORDER BY precio ASC;

Query Query History

```
1 SELECT * FROM Producto ORDER BY precio ASC;
```

Data Output Messages Notifications

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

	id_producto [PK] integer	nombre character varying (100)	descripcion text	precio numeric (10,2)	stock integer	id_categoria integer	id_proveedor integer
1	4	Software Antivirus	Licencia de antivirus por 1 año	50.00	100	4	4
2	8	Cámara de Seguridad	Cámara con visión nocturna	75.00	40	7	8
3	3	Teclado Mecánico	Teclado RGB con switches mecánicos	80.00	30	5	3
4	7	SSD 1TB	Unidad de estado sólido de 1TB	110.00	50	8	7
5	5	Router WiFi 6	Router de alta velocidad	120.00	25	9	5
6	10	Microondas Digital	Horno microondas de 1200W	180.00	8	10	10
7	2	Silla Ergonómica	Silla de oficina cómoda	200.00	15	6	2
8	9	Multifuncional Láser	Impresora y escáner en un solo dispositivo	250.00	12	2	9
9	6	Monitor 27"	Monitor 4K de 27 pulgadas	350.00	10	1	6
10	1	Laptop X	Laptop con procesador i7 y 16GB RAM	1500.00	20	1	1

Query

Query History

```

1  SELECT id_categoria AS clave_categoria, COUNT(*) AS total_productos
2  FROM Producto
3  GROUP BY id_categoria;

```

Data Output

Messages

Notifications

≡+

📄

▼

📄

▼

🗑️

🗑️

📄

⬇️

📈

SQL

Showing row 1 of 9

	clave_categoria integer	total_productos bigint
1	9	1
2	5	1
3	4	1
4	10	1
5	6	1
6	2	1
7	7	1
8	1	2
9	8	1

Query

Query History

1

2

3

SELECT P.nombre AS producto, C.nombre AS categoria
FROM Producto P
INNER JOIN Categoria C ON P.id_categoria = C.id_categoria;

Data Output

Messages

Notifications

≡+

▼

▼

SQL

	producto character varying (100)	categoria character varying (100)
1	Laptop X	Electrónica
2	Silla Ergonómica	Muebles
3	Teclado Mecánico	Accesorios
4	Software Antivirus	Software
5	Router WiFi 6	Redes
6	Monitor 27"	Electrónica
7	SSD 1TB	Almacenamiento
8	Cámara de Seguridad	Seguridad
9	Multifuncional Láser	Oficina
10	Microondas Digital	Electrodomésticos

SELECT * FROM Producto WHERE nombre LIKE '%Laptop%';

Query Query History

1 SELECT * FROM Producto WHERE nombre LIKE '%Laptop%';

Data Output Messages Notifications

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

	id_producto [PK] integer	nombre character varying (100)	descripcion text	precio numeric (10,2)	stock integer	id_categoria integer	id_proveedor integer
1	1	Laptop X	Laptop con procesador i7 y 16GB RAM	1500.00	20	1	1

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

SELECT P.nombre AS producto, C.nombre AS categoria, V.nombre AS proveedor
FROM Producto P
INNER JOIN Categoria C ON P.id_categoria = C.id_categoria
INNER JOIN Proveedor V ON P.id_proveedor = V.id_proveedor
ORDER BY P.nombre ASC;

Query Query History

1 SELECT P.nombre AS producto, C.nombre AS categoria, V.nombre AS proveedor
2 FROM Producto P
3 INNER JOIN Categoria C ON P.id_categoria = C.id_categoria
4 INNER JOIN Proveedor V ON P.id_proveedor = V.id_proveedor
5 ORDER BY P.nombre ASC;

Data Output Messages Notifications

Showing rows: 1 to 1

	producto character varying (100)	categoria character varying (100)	proveedor character varying (100)
1	Cámara de Seguridad	Seguridad	Gadget World
2	Laptop X	Electrónica	Tech Supplies
3	Microondas Digital	Electrodomésticos	NextGen Supplies
4	Monitor 27"	Electrónica	Smart Devices
5	Multifuncional Láser	Oficina	Super Components
6	Router WiFi 6	Redes	Mega Distribuidora
7	Silla Ergonómica	Muebles	Office Essentials
8	Software Antivirus	Software	Digital Solutions
9	SSD 1TB	Almacenamiento	Innovatech
10	Teclado Mecánico	Accesorios	Hardware Plus

Sistema gestión de eventos

Una empresa de organización de eventos necesita administrar sus eventos y participantes.

- **Identificar entidades claves: Evento, Participante, Ubicación, Organizador.**

Tenemos las entidades:

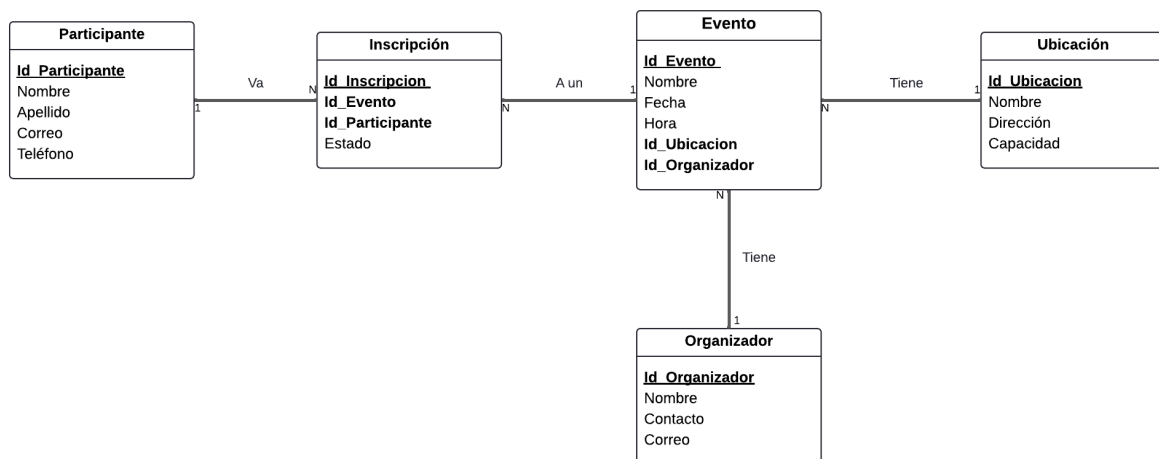
Evento: Un evento se realiza en una única ubicación.

Participante: Un participante puede inscribirse en varios eventos.

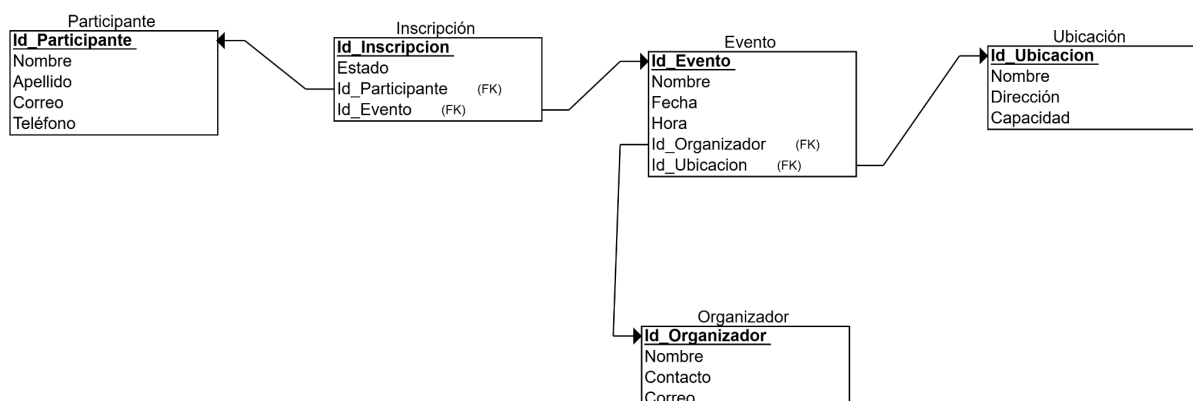
Ubicación: Una ubicación puede albergar varios eventos.

Organizador: Cada evento tiene un organizador.

- **Diseñar el modelo E-R.**



- **Transformar el modelo en un esquema relacional.**



- **Implementar la base de datos en SQL.**

--Crear tabla Ubicación

```
CREATE TABLE Ubicacion (  
    id_ubicacion SERIAL PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    direccion TEXT NOT NULL,  
    capacidad INT NOT NULL CHECK (capacidad > 0)  
);
```

--Crear tabla Organizador

```
CREATE TABLE Organizador (  
    id_organizador SERIAL PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    contacto VARCHAR(50),  
    correo VARCHAR(100) UNIQUE NOT NULL  
);
```

--Crear tabla Evento

```
CREATE TABLE Evento (  
    id_evento SERIAL PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    fecha DATE NOT NULL,  
    hora TIME NOT NULL,  
    id_ubicacion INT NOT NULL REFERENCES Ubicacion(id_ubicacion) ON DELETE  
CASCADE,  
    id_organizador INT NOT NULL REFERENCES Organizador(id_organizador) ON  
DELETE CASCADE  
);
```

--Crear tabla Participante

```
CREATE TABLE Participante (  
    id_participante SERIAL PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    apellido VARCHAR(100) NOT NULL,  
    correo VARCHAR(100) UNIQUE NOT NULL,  
    telefono VARCHAR(20)  
);
```

--Crear tabla Inscripción

```
CREATE TABLE Inscripcion (  
    id_inscripcion SERIAL PRIMARY KEY,  
    id_evento INT NOT NULL REFERENCES Evento(id_evento) ON DELETE CASCADE,  
    id_participante INT NOT NULL REFERENCES Participante(id_participante) ON  
DELETE CASCADE,  
    estado VARCHAR(20) CHECK (estado IN ('Confirmado', 'Pendiente', 'Cancelado')) NOT  
NULL DEFAULT 'Pendiente'  
);
```


• Consultas a hacer.

SELECT * FROM Evento;

Query Query History

```
1 SELECT * FROM Evento;
```

Data Output Messages Notifications

Showing rows: 1 to 10

	id_evento [PK] integer	nombre character varying (100)	fecha date	hora time without time zone	id_ubicacion integer	id_organizador integer
1	1	Congreso de Tecnología	2025-05-20	09:00:00	1	1
2	2	Expo Emprendedores	2025-06-10	10:30:00	2	2
3	3	Foro de Innovación	2025-07-15	08:00:00	3	3
4	4	Taller de Marketing Digital	2025-08-05	14:00:00	4	4
5	5	Concierto Sinfónico	2025-09-12	19:30:00	5	5
6	6	Feria de Negocios	2025-10-25	11:00:00	6	6
7	7	Seminario de Finanzas	2025-11-18	16:00:00	7	7
8	8	Conferencia de Ciencia	2025-12-05	09:30:00	8	8
9	9	Festival de Arte	2026-01-20	18:00:00	9	9
10	10	Convención de Videojue...	2026-02-14	12:00:00	10	10

SELECT * FROM Evento
WHERE fecha = '2025-06-10';

Query Query History

```
1 SELECT * FROM Evento  
2 WHERE fecha = '2025-06-10';
```

Data Output Messages Notifications

Showing rows: 1 to 1

	id_evento [PK] integer	nombre character varying (100)	fecha date	hora time without time zone	id_ubicacion integer	id_organizador integer
1	2	Expo Emprendedores	2025-06-10	10:30:00	2	2

SELECT * FROM Evento
ORDER BY fecha ASC;

Query Query History

```
1 SELECT * FROM Evento  
2 ORDER BY fecha ASC;
```

Data Output Messages Notifications

Showing rows: 1 to 10

	id_evento [PK] integer	nombre character varying (100)	fecha date	hora time without time zone	id_ubicacion integer	id_organizador integer
1	1	Congreso de Tecnología	2025-05-20	09:00:00	1	1
2	2	Expo Emprendedores	2025-06-10	10:30:00	2	2
3	3	Foro de Innovación	2025-07-15	08:00:00	3	3
4	4	Taller de Marketing Digital	2025-08-05	14:00:00	4	4
5	5	Concierto Sinfónico	2025-09-12	19:30:00	5	5
6	6	Feria de Negocios	2025-10-25	11:00:00	6	6
7	7	Seminario de Finanzas	2025-11-18	16:00:00	7	7
8	8	Conferencia de Ciencia	2025-12-05	09:30:00	8	8
9	9	Festival de Arte	2026-01-20	18:00:00	9	9
10	10	Convención de Videojue...	2026-02-14	12:00:00	10	10

```
SELECT id_evento, COUNT(id_participante) AS total_participantes
FROM Inscripcion
GROUP BY id_evento;
```

Query Query History

```
1 SELECT id_evento, COUNT(id_participante) AS total_participantes
2 FROM Inscripcion
3 GROUP BY id_evento;
```

Data Output Messages Notifications

SQL Showing

	id_evento integer	total_participantes 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	2
10	8	2

```
SELECT E.nombre AS evento, U.nombre AS ubicacion, U.direccion
FROM Evento E
INNER JOIN Ubicacion U ON E.id_ubicacion = U.id_ubicacion;
```

Query Query History

```
1 SELECT E.nombre AS evento, U.nombre AS ubicacion, U.direccion
2 FROM Evento E
3 INNER JOIN Ubicacion U ON E.id_ubicacion = U.id_ubicacion;
```

Data Output Messages Notifications

SQL Showing

	evento character varying (100)	ubicacion character varying (100)	direccion text
1	Congreso de Tecnología	Centro de Convenciones	Av. Principal #123, CDMX
2	Expo Emprendedores	Salón Imperial	Calle 45 #678, Monterrey
3	Foro de Innovación	Auditorio Nacional	Paseo de la Reforma #50, CDMX
4	Taller de Marketing Digital	Expo Guadalajara	Av. Mariano Otero #1499, Guadalajara
5	Concierto Sinfónico	Salón Ejecutivo	Av. Insurgentes #99, CDMX
6	Feria de Negocios	Hotel Fiesta	Calle 23 #567, Cancún
7	Seminario de Finanzas	Palacio de Congresos	Calle Mayor #10, Mérida
8	Conferencia de Ciencia	Salón Elite	Av. Revolución #210, Monterrey
9	Festival de Arte	Teatro Metropolitano	Centro Histórico, Querétaro
10	Convención de Videojue...	Foro Cultural	Calle de las Artes #321, Puebla

```
SELECT * FROM Participante
WHERE apellido LIKE 'García';
```

Query Query History

```
1 SELECT * FROM Participante
2 WHERE apellido LIKE 'García';
```

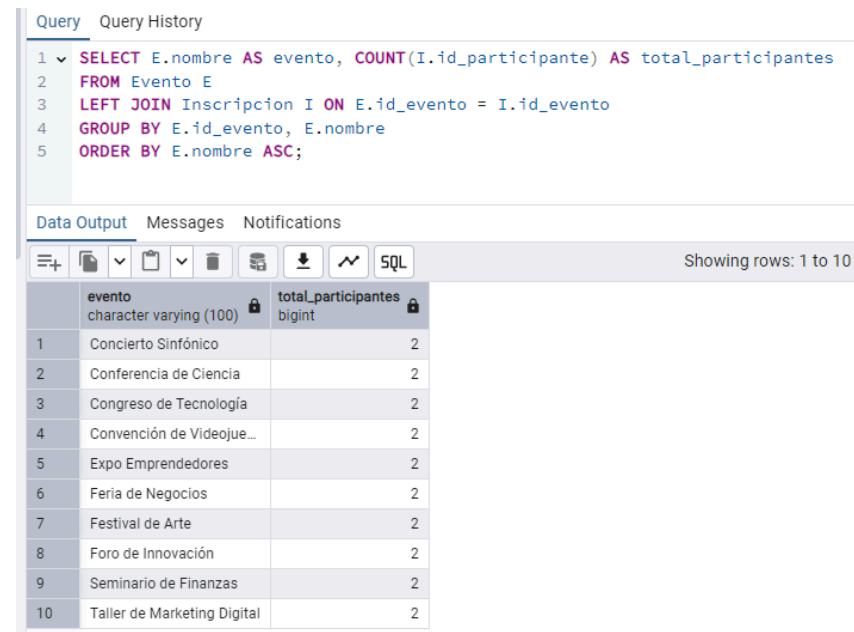
Data Output Messages Notifications

SQL Showing rows: 1 to 1 Page No.

	id_participante [PK] integer	nombre character varying (100)	apellido character varying (100)	correo character varying (100)	telefono character varying (20)
1	3	Carlos	García	carlos.garcia@mail.com	33-3333-4444

Consulta requerida: Obtener lista de eventos programados junto con la cantidad de participantes registrados por evento.

```
SELECT E.nombre AS evento, COUNT(I.id_participante) AS total_participantes
FROM Evento E
LEFT JOIN Inscripcion I ON E.id_evento = I.id_evento
GROUP BY E.id_evento, E.nombre
ORDER BY E.nombre ASC;
```



Query Query History

```
1 SELECT E.nombre AS evento, COUNT(I.id_participante) AS total_participantes
2 FROM Evento E
3 LEFT JOIN Inscripcion I ON E.id_evento = I.id_evento
4 GROUP BY E.id_evento, E.nombre
5 ORDER BY E.nombre ASC;
```

Data Output Messages Notifications

Showing rows: 1 to 10

	evento character varying (100)	total_participantes bigint
1	Concierto Sinfónico	2
2	Conferencia de Ciencia	2
3	Congreso de Tecnología	2
4	Convención de Videojue...	2
5	Expo Emprendedores	2
6	Feria de Negocios	2
7	Festival de Arte	2
8	Foro de Innovación	2
9	Seminario de Finanzas	2
10	Taller de Marketing Digital	2

Plataforma de streaming de música

Se desea un sistema para administrar usuarios, artistas y sus canciones.

- **Identificar entidades claves: Usuario, Artista, Álbum, Canción.**

Tenemos las entidades:

Usuario: Puede escuchar varias canciones y puede seguir a varios artistas.

Artista: Puede tener múltiples álbumes y puede lanzar varias canciones.

Álbum: Contiene varias canciones y pertenece a un artista.

Canción: Forma parte de un álbum y puede ser escuchada por varios usuarios.

- **Diseñar el modelo E-R.**
- **Transformar el modelo en un esquema relacional.**
- **Implementar la base de datos en SQL.**
- **Consultas a hacer.**