



**TECNOLÓGICO
NACIONAL DE MÉXICO**



Tecnológico Nacional de México, Campus
Mexicali
Ingeniería en Sistemas Computacionales

Alumno:
Emanuel Padilla Valencia
23490363

Materia: Fundamentos de Bases de Datos
Docente: José Ramón Bogarin Valenzuela

***Tarea 3 Unidad 2 Unidad 2 Ejercicios,
Creación de tablas y Consultas en SQL***

Mexicali Baja California, al 23 de marzo
del 2025.

1. SISTEMA DE GESTIÓN DE INVENTARIOS.

Identificación de entidades clave

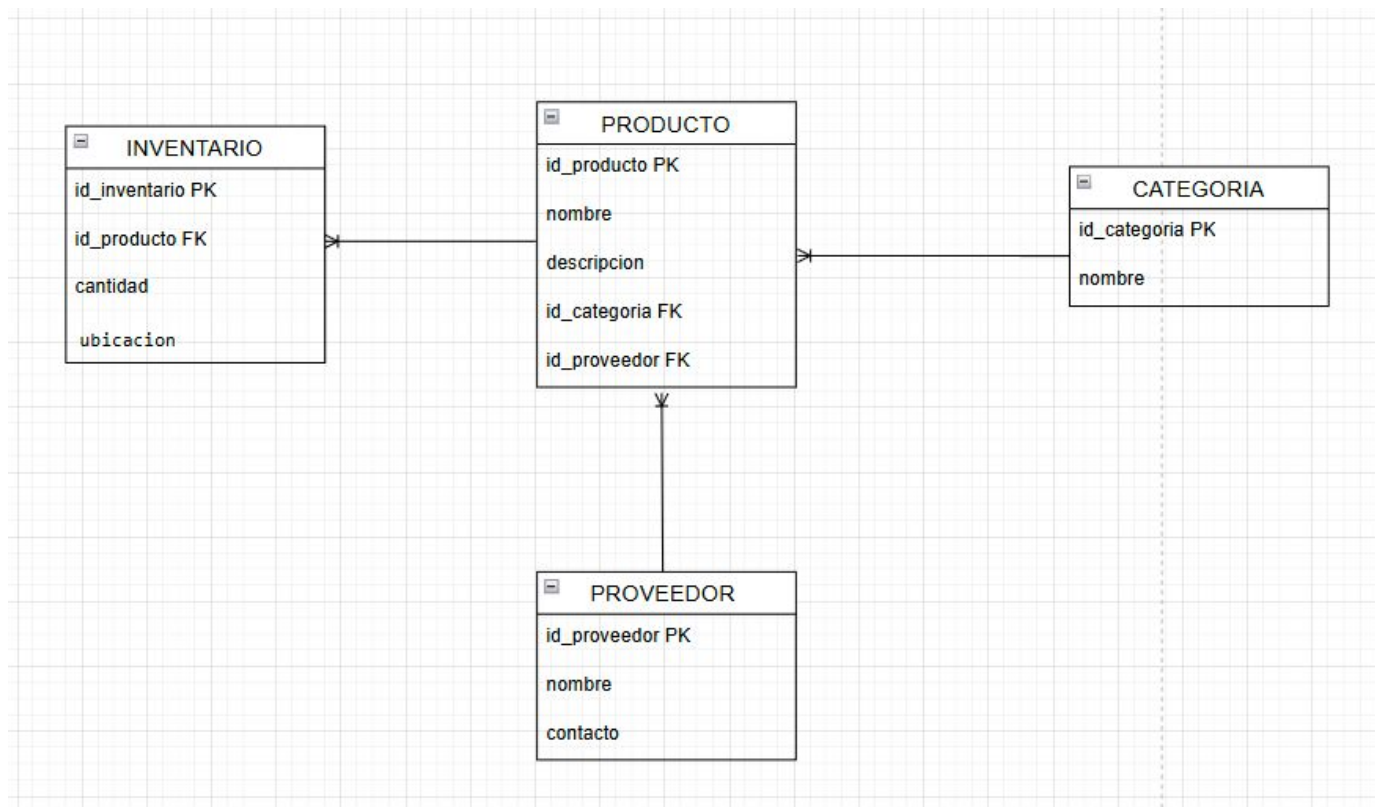
Producto: Contiene información del producto como id_producto, nombre, descripcion, id_categoria y id_proveedor.

Proveedor: Incluye id_proveedor, nombre y contacto.

Categoría: Define la categoría del producto con id_categoria y nombre.

Inventario: Representa el control de stock con id_inventario, id_producto, cantidad y ubicacion.

Diagrama Entidad-Relación.



Esquema Relacional.

Define las claves
primarias PK
y foráneas FK para
relacionar las tablas

Categoria: (id_categoria (PK), nombre)

Proveedor: (id_proveedor
(PK), nombre, contacto)

Producto: (id_producto
(PK), nombre, descripcion, id_categoria
(FK), id_proveedor (FK))

Inventario: (id_inventario (PK), id_producto
(FK), cantidad, ubicacion)

Creación de tablas con SQL.

```
CREATE TABLE Categoria (  
    id_categoria INT PRIMARY KEY,  
    nombre_categoria VARCHAR(50) NOT NULL  
);  
  
CREATE TABLE Proveedor (  
    id_proveedor INT PRIMARY KEY,  
    nombre_proveedor VARCHAR(100) NOT NULL,  
    contacto VARCHAR(100)  
);  
  
CREATE TABLE Producto (  
    id_producto INT PRIMARY KEY,  
    nombre_producto VARCHAR(100) NOT NULL,  
    id_categoria INT,  
    id_proveedor INT,  
    FOREIGN KEY (id_categoria) REFERENCES Categoria(id_categoria),  
    FOREIGN KEY (id_proveedor) REFERENCES Proveedor(id_proveedor)  
);  
  
CREATE TABLE Inventario (  
    id_inventario INT PRIMARY KEY,  
    id_producto INT,  
    cantidad INT NOT NULL,  
    fecha_actualizacion DATE,  
    FOREIGN KEY (id_producto) REFERENCES Producto(id_producto)  
);
```

Implementación en SQL con INSERT INTO.

```
INSERT INTO Categoria (id_categoria, nombre_categoria) VALUES
(1, 'Electrónica'),
(2, 'Papelería'),
(3, 'Alimentos'),
(4, 'Limpieza');

INSERT INTO Proveedor (id_proveedor, nombre_proveedor, contacto) VALUES
(1, 'Tech Solutions', 'contacto@techsolutions.com'),
(2, 'OfiExpress', 'ventas@ofiexpress.com'),
(3, 'Super Alimentos', 'info@superalimentos.com'),
(4, 'CleanPro', 'contacto@cleanpro.com');

INSERT INTO Producto (id_producto, nombre_producto, id_categoria, id_proveedor) VALUES
(1, 'Laptop HP', 1, 1),
(2, 'Mouse inalámbrico', 1, 1),
(3, 'Cuaderno profesional', 2, 2),
(4, 'Papel bond A4', 2, 2),
(5, 'Galletas integrales', 3, 3),
(6, 'Botella de agua', 3, 3),
(7, 'Desinfectante multiusos', 4, 4),
(8, 'Toallas húmedas', 4, 4);

INSERT INTO Inventario (id_inventario, id_producto, cantidad, fecha_actualizacion) VALUES
(1, 1, 10, '2025-03-01'),
(2, 2, 25, '2025-03-05'),
(3, 3, 100, '2025-03-02'),
(4, 4, 200, '2025-03-03'),
(5, 5, 50, '2025-03-04'),
(6, 6, 80, '2025-03-06'),
(7, 7, 40, '2025-03-07'),
(8, 8, 60, '2025-03-08');
```

Consulta SQL.

Se listan todos los productos junto con su categoría (obligatoria) y su proveedor (opcional). El INNER JOIN garantiza que solo se muestren productos con categoría válida, y el LEFT JOIN permite incluir productos sin proveedor.

1

2

3

4

5

6

7

8

9

10

11

SELECT

p.nombre_producto AS producto,

c.nombre_categoria AS categoria,

pr.nombre_proveedor AS proveedor

FROM

producto p

INNER JOIN categoria c ON p.id_categoria = c.id_categoria

LEFT JOIN proveedor pr ON p.id_proveedor = pr.id_proveedor

ORDER BY

p.nombre_producto ASC;

Data Output

Mensajes

Notificaciones

≡+

📄

▼

📋

▼

🗑️

📦

📶

📈

SQL

	producto character varying (100) 🔒	categoria character varying (50) 🔒	proveedor character varying (100) 🔒
1	Botella de agua	Alimentos	Super Alimentos
2	Cuaderno profesional	Papelería	OfiExpress
3	Desinfectante multiusos	Limpieza	CleanPro
4	Galletas integrales	Alimentos	Super Alimentos
5	Laptop HP	Electrónica	Tech Solutions
6	Mouse inalámbrico	Electrónica	Tech Solutions
7	Papel bond A4	Papelería	OfiExpress
8	Toallas húmedas	Limpieza	CleanPro

2. SISTEMA DE GESTIÓN DE EVENTOS

Identificación de entidades clave

Evento: Contiene información del evento como id_evento, nombre, fecha, id_ubicacion y id_organizador.

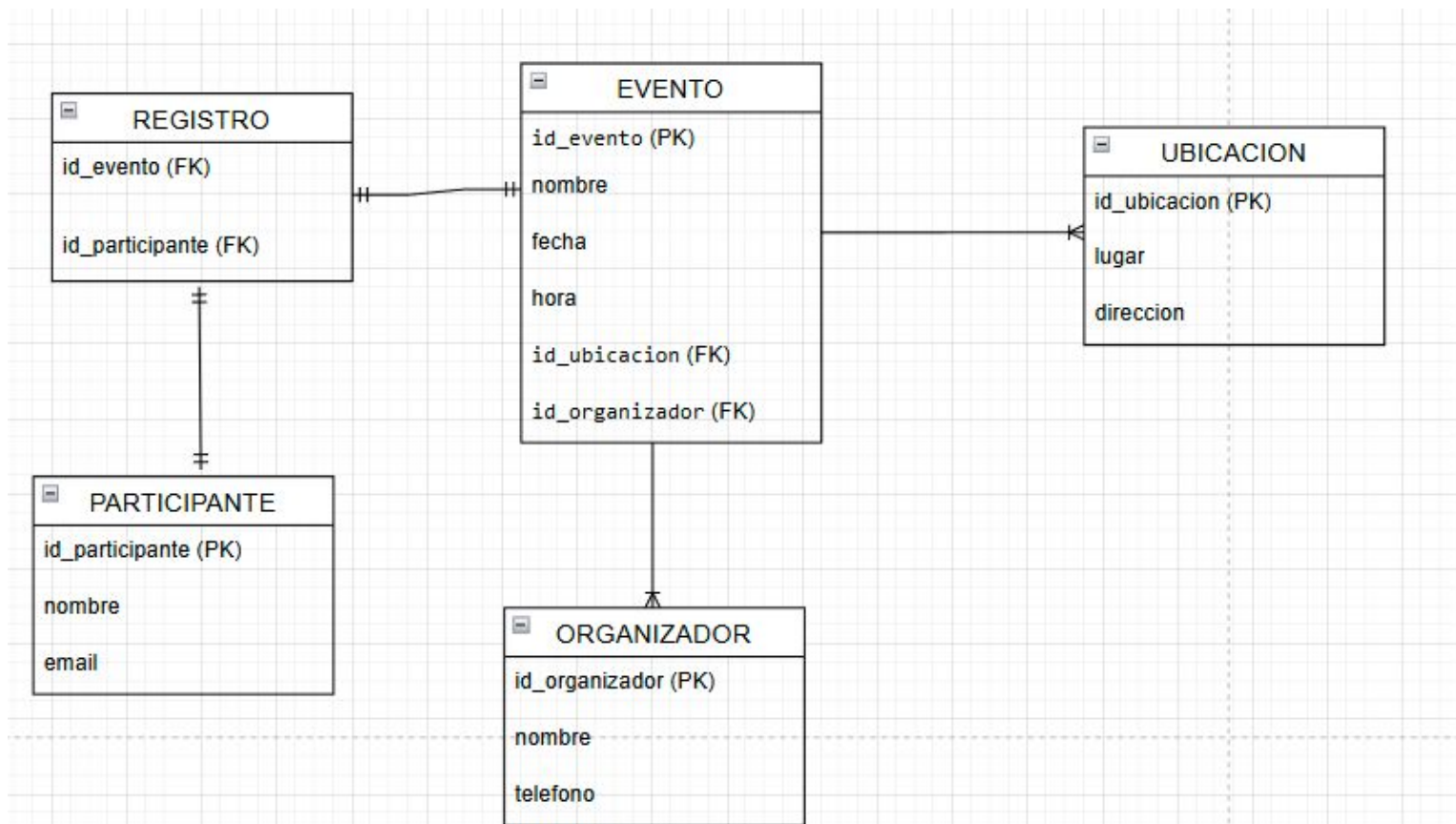
Participante: Incluye id_participante, nombre y correo.

Ubicación: Define el lugar del evento con id_ubicacion, nombre y direccion.

Organizador: Representa a quien organiza el evento con id_organizador, nombre y empresa.

Registro: Relaciona eventos y participantes mediante id_evento y id_participante.

Diagrama Entidad-Relación



Esquema relacional

Define las claves
primarias PK
y foráneas FK para
relacionar las tablas

Evento: (id_evento
(PK), nombre, fecha, id_ubicacion
(FK), id_organizador (FK))

Participante: (id_participante (PK), nombre, correo)

Ubicacion: (id_ubicacion (PK), nombre, direccion)

Organizador: (id_organizador
(PK), nombre, empresa)

Registro: (id_evento (FK), id_participante (FK))

Creación de tablas con SQL.

```
CREATE TABLE Ubicacion (  
    id_ubicacion INT PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    direccion VARCHAR(150)  
);  
  
CREATE TABLE Organizador (  
    id_organizador INT PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    empresa VARCHAR(100)  
);  
  
CREATE TABLE Evento (  
    id_evento INT PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    fecha DATE,  
    id_ubicacion INT,  
    id_organizador INT,  
    FOREIGN KEY (id_ubicacion) REFERENCES Ubicacion(id_ubicacion),  
    FOREIGN KEY (id_organizador) REFERENCES Organizador(id_organizador)  
);  
  
CREATE TABLE Participante (  
    id_participante INT PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    correo VARCHAR(100)  
);  
  
CREATE TABLE Registro (  
    id_evento INT,  
    id_participante INT,  
    PRIMARY KEY (id_evento, id_participante),  
    FOREIGN KEY (id_evento) REFERENCES Evento(id_evento),  
    FOREIGN KEY (id_participante) REFERENCES Participante(id_participante)  
);
```

Implementación en SQL con INSERT INTO.

```
-- INSERTAR DATOS EN UBICACION
```

```
INSERT INTO Ubicacion (id_ubicacion, nombre, direccion) VALUES
```

```
(1, 'Auditorio Central', 'Calle 1 #123'),  
(2, 'Salón Azul', 'Av. Reforma 456'),  
(3, 'Centro de Convenciones', 'Blvd. Eventos 101'),  
(4, 'Sala Ejecutiva', 'Calle Ejecutiva 88'),  
(5, 'Teatro Municipal', 'Calle Cultura 99'),  
(6, 'Hotel Real', 'Av. Hoteles 202'),  
(7, 'Universidad Norte', 'Campus Norte'),  
(8, 'Parque Tecnológico', 'Calle Innovación 75'),  
(9, 'Edificio Omega', 'Omega Plaza 3'),  
(10, 'Centro Cultural', 'Calle Arte 321'),  
(11, 'Plaza Eventos', 'Plaza 45'),  
(12, 'Sala VIP', 'Privada Vista 7'),  
(13, 'Hotel Gran Sol', 'Av. Mar 12'),  
(14, 'Coworking Space', 'Calle Startup 15'),  
(15, 'Salón Espejos', 'Espejo #5'),  
(16, 'Auditorio Sur', 'Zona Sur 4'),  
(17, 'Torre Ejecutiva', 'Piso 10'),  
(18, 'ExpoForum', 'Expo 88'),  
(19, 'Foro Libre', 'Av. Cultura 300'),  
(20, 'Pabellón Norte', 'Zona Industrial 99');
```

```
-- INSERTAR DATOS EN ORGANIZADOR
```

```
INSERT INTO Organizador (id_organizador, nombre, empresa) VALUES
```

```
(1, 'Ana Torres', 'Eventos Pro'),  
(2, 'Luis Gómez', 'Conferencias Globales'),  
(3, 'María Ruiz', 'Red Creativa'),  
(4, 'Jorge Díaz', 'OrgEventos'),  
(5, 'Sofía Peña', 'Elite Eventos'),  
(6, 'Carlos Méndez', 'Eventos México'),  
(7, 'Laura Castro', 'Norte Producciones'),  
(8, 'Pedro Sánchez', 'Eventos Y Más'),  
(9, 'Daniela Vega', 'Foros Latinos'),  
(10, 'Raúl Ortega', 'Visión 360'),  
(11, 'Camila López', 'Eventos XYZ'),  
(12, 'Fernando Silva', 'Grand Meetings'),  
(13, 'Patricia Mora', 'Networking Corp'),  
(14, 'Andrés Flores', 'Eventos Avanzados'),  
(15, 'Natalia Ramos', 'Innovent'),  
(16, 'Iván Castillo', 'Soluciones de Eventos'),  
(17, 'Rebeca Cruz', 'Fiesta Pro'),  
(18, 'Tomás Rivas', 'Global Talks'),  
(19, 'Elena Figueroa', 'Eventos Masivos'),  
(20, 'Roberto Núñez', 'Fusión Creativa');
```



```
-- INSERTAR DATOS EN EVENTO
INSERT INTO Evento (id_evento, nombre, fecha, id_ubicacion, id_organizador) VALUES
(1, 'Congreso Tecnología', '2025-04-10', 1, 1),
(2, 'Feria Universitaria', '2025-05-15', 2, 2),
(3, 'Conferencia Salud', '2025-06-05', 3, 3),
(4, 'Seminario Finanzas', '2025-07-20', 4, 4),
(5, 'Encuentro Educativo', '2025-08-12', 5, 5),
(6, 'Foro Ambiental', '2025-09-01', 6, 6),
(7, 'Expo Innovación', '2025-04-25', 7, 7),
(8, 'Charla Motivacional', '2025-05-30', 8, 8),
(9, 'Taller Diseño', '2025-06-15', 9, 9),
(10, 'Mesa Redonda Política', '2025-07-10', 10, 10),
(11, 'Festival Cultural', '2025-07-25', 11, 11),
(12, 'Simposio Ciencias', '2025-08-10', 12, 12),
(13, 'Convención Negocios', '2025-09-05', 13, 13),
(14, 'Expo Arte Digital', '2025-10-01', 14, 14),
(15, 'Conferencia Educación', '2025-11-15', 15, 15),
(16, 'Foro de Mujeres', '2025-12-05', 16, 16),
(17, 'Startup Meetup', '2025-12-20', 17, 17),
(18, 'Día del Emprendedor', '2026-01-15', 18, 18),
(19, 'Seminario Filosofía', '2026-02-05', 19, 19),
(20, 'Cumbre Internacional', '2026-03-01', 20, 20);
```

```
-- INSERTAR DATOS EN PARTICIPANTE
INSERT INTO Participante (id_participante, nombre, correo) VALUES
(1, 'Andrea Pérez', 'andrea@mail.com'),
(2, 'Carlos Díaz', 'carlos@mail.com'),
(3, 'Lucía Torres', 'lucia@mail.com'),
(4, 'Miguel López', 'miguel@mail.com'),
(5, 'Elena Ramírez', 'elena@mail.com'),
(6, 'Juan García', 'juan@mail.com'),
(7, 'María Hernández', 'mariah@mail.com'),
(8, 'Pedro Jiménez', 'pedro@mail.com'),
(9, 'Laura Moreno', 'laura@mail.com'),
(10, 'José Ortega', 'jose@mail.com'),
(11, 'Diana Ruiz', 'diana@mail.com'),
(12, 'Ricardo Soto', 'ricardo@mail.com'),
(13, 'Valeria Núñez', 'valeria@mail.com'),
(14, 'Francisco Castillo', 'fran@mail.com'),
(15, 'Renata Reyes', 'renata@mail.com'),
(16, 'Roberto Vázquez', 'roberto@mail.com'),
(17, 'Camila Ríos', 'camila@mail.com'),
(18, 'Martín Gómez', 'martin@mail.com'),
(19, 'Julieta Luna', 'julieta@mail.com'),
(20, 'Tomás Fuentes', 'tomas@mail.com');
```

```
-- INSERTAR DATOS EN REGISTRO (asignamos participantes a eventos)
INSERT INTO Registro (id_evento, id_participante) VALUES
(1, 1), (1, 2), (1, 3),
(2, 4), (2, 5),
(3, 6), (3, 7), (3, 8),
(4, 9), (4, 10),
(5, 11), (5, 12),
(6, 13), (6, 14), (6, 15),
(7, 16), (7, 17),
(8, 18), (8, 19), (8, 20);
```

Consulta SQL.

Se obtienen todos los eventos, aunque no tengan participantes registrados, gracias al LEFT JOIN. Se usa COUNT para contar cuántos participantes hay por evento y se agrupa por nombre y fecha.

```
1 SELECT
2     e.nombre AS evento,
3     e.fecha,
4     COUNT(r.id_participante) AS total_participantes
5 FROM
6     evento e
7 LEFT JOIN registro r ON e.id_evento = r.id_evento
8 GROUP BY
9     e.id_evento, e.nombre, e.fecha
10 ORDER BY
11     e.fecha ASC;
12
```

	evento character varying (100)	fecha date	total_participantes bigint
1	Congreso Tecnología	2025-04-10	3
2	Expo Innovación	2025-04-25	2
3	Feria Universitaria	2025-05-15	2
4	Charla Motivacional	2025-05-30	3
5	Conferencia Salud	2025-06-05	3
6	Taller Diseño	2025-06-15	0
7	Mesa Redonda Política	2025-07-10	0
8	Seminario Finanzas	2025-07-20	2
9	Festival Cultural	2025-07-25	0
10	Simposio Ciencias	2025-08-10	0
11	Encuentro Educativo	2025-08-12	2
12	Foro Ambiental	2025-09-01	3
13	Convención Negocios	2025-09-05	0
14	Expo Arte Digital	2025-10-01	0
15	Conferencia Educación	2025-11-15	0
16	Foro de Mujeres	2025-12-05	0
17	Startup Meetup	2025-12-20	0
18	Día del Emprendedor	2026-01-15	0
19	Seminario Filosofía	2026-02-05	0
20	Cumbre Internacional	2026-03-01	0

3. PLATAFORMA DE STREAMING DE MÚSICA.

Identificación de entidades clave

Usuario: Contiene la información del usuario como id_usuario, nombre y correo.

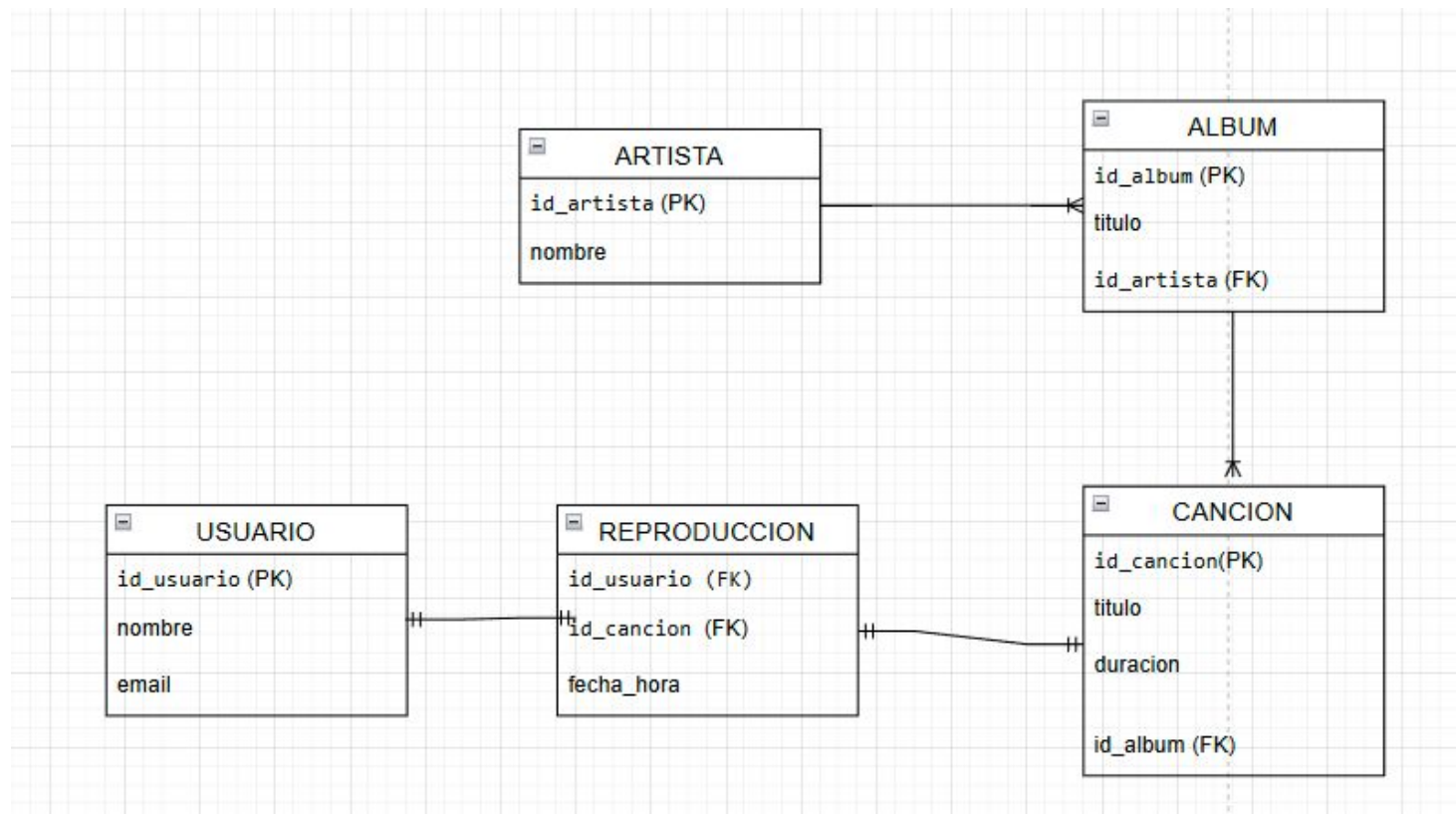
Artista: Incluye id_artista, nombre y pais.

Álbum: Relacionado con un artista, con atributos id_album, titulo, año y id_artista.

Canción: Pertenece a un álbum y contiene id_cancion, titulo, duracion, id_album.

Reproducción: Relaciona canciones y usuarios mediante id_usuario, id_cancion y fecha_reproduccion.

Diagrama Entidad-Relación.



Esquema relacional.

Define las claves primarias
PK

y foráneas FK para
relacionar las tablas

Usuario: (id_usuario (PK), nombre, correo)

Artista: (id_artista (PK), nombre, pais)

Album: (id_album (PK), titulo, año, id_artista (FK))

Cancion: (id_cancion (PK), titulo, duracion, id_album (FK))

Reproduccion: (id_usuario (FK), id_cancion (FK), fecha_reproduccion)

Creación de tablas con SQL.

```
CREATE TABLE usuario (  
    id_usuario INT PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    correo VARCHAR(100) NOT NULL  
);  
  
CREATE TABLE artista (  
    id_artista INT PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    pais VARCHAR(50)  
);  
  
CREATE TABLE album (  
    id_album INT PRIMARY KEY,  
    titulo VARCHAR(100) NOT NULL,  
    año INT,  
    id_artista INT,  
    FOREIGN KEY (id_artista) REFERENCES artista(id_artista)  
);  
  
CREATE TABLE cancion (  
    id_cancion INT PRIMARY KEY,  
    titulo VARCHAR(100) NOT NULL,  
    duracion TIME,  
    id_album INT,  
    FOREIGN KEY (id_album) REFERENCES album(id_album)  
);  
  
CREATE TABLE reproduccion (  
    id_usuario INT,  
    id_cancion INT,  
    fecha_reproduccion DATE,  
    PRIMARY KEY (id_usuario, id_cancion, fecha_reproduccion),  
    FOREIGN KEY (id_usuario) REFERENCES usuario(id_usuario),  
    FOREIGN KEY (id_cancion) REFERENCES cancion(id_cancion)  
);
```

Implementación en SQL con INSERT INTO.

```
-- USUARIOS
```

```
INSERT INTO usuario (id_usuario, nombre, correo) VALUES
```

```
(1, 'Ana Pérez', 'ana@mail.com'), (2, 'Luis Gómez', 'luis@mail.com'),  
(3, 'Laura Díaz', 'laura@mail.com'), (4, 'Carlos Ruiz', 'carlos@mail.com'),  
(5, 'Sofía Martínez', 'sofia@mail.com'), (6, 'Jorge Torres', 'jorge@mail.com'),  
(7, 'Camila Ortega', 'camila@mail.com'), (8, 'Daniela Mora', 'daniela@mail.com'),  
(9, 'Pedro Vázquez', 'pedro@mail.com'), (10, 'Lucía Ríos', 'lucia@mail.com'),  
(11, 'Tomás Suárez', 'tomas@mail.com'), (12, 'María Figueroa', 'maria@mail.com'),  
(13, 'Iván Vega', 'ivan@mail.com'), (14, 'Rebeca Castillo', 'rebeca@mail.com'),  
(15, 'Julieta Luna', 'julieta@mail.com'), (16, 'Raúl Herrera', 'raul@mail.com'),  
(17, 'Fernanda Paredes', 'fernanda@mail.com'), (18, 'Martín Aguilar', 'martin@mail.com'),  
(19, 'Patricia León', 'patricia@mail.com'), (20, 'Gabriel Campos', 'gabriel@mail.com');
```

```
-- ARTISTAS
```

```
INSERT INTO artista (id_artista, nombre, pais) VALUES
```

```
(1, 'Coldplay', 'Reino Unido'), (2, 'Shakira', 'Colombia'),  
(3, 'Adele', 'Reino Unido'), (4, 'Dua Lipa', 'Reino Unido'),  
(5, 'Bad Bunny', 'Puerto Rico'), (6, 'Karol G', 'Colombia'),  
(7, 'Ed Sheeran', 'Reino Unido'), (8, 'Taylor Swift', 'EEUU'),  
(9, 'Drake', 'Canadá'), (10, 'Rosalía', 'España'),  
(11, 'The Weeknd', 'Canadá'), (12, 'BTS', 'Corea del Sur'),  
(13, 'Billie Eilish', 'EEUU'), (14, 'Maluma', 'Colombia'),  
(15, 'Imagine Dragons', 'EEUU'), (16, 'Selena Gomez', 'EEUU'),  
(17, 'Luis Fonsi', 'Puerto Rico'), (18, 'Nathy Peluso', 'Argentina'),  
(19, 'Harry Styles', 'Reino Unido'), (20, 'J Balvin', 'Colombia');
```

```
-- ALBUMES
```

```
INSERT INTO album (id_album, titulo, año, id_artista) VALUES
```

```
(1, 'Parachutes', 2000, 1), (2, 'Laundry Service', 2001, 2),  
(3, '25', 2015, 3), (4, 'Future Nostalgia', 2020, 4),  
(5, 'YHLQMDLG', 2020, 5), (6, 'KG0516', 2021, 6),  
(7, 'Divide', 2017, 7), (8, '1989', 2014, 8),  
(9, 'Scorpion', 2018, 9), (10, 'Motomami', 2022, 10),  
(11, 'After Hours', 2020, 11), (12, 'Map of the Soul', 2019, 12),  
(13, 'Happier Than Ever', 2021, 13), (14, 'Papi Juancho', 2020, 14),  
(15, 'Night Visions', 2012, 15), (16, 'Revival', 2015, 16),  
(17, 'Vida', 2019, 17), (18, 'Calambre', 2020, 18),  
(19, 'Fine Line', 2019, 19), (20, 'Colores', 2020, 20);
```

```
-- CANCIONES
INSERT INTO cancion (id_cancion, titulo, duracion, id_album) VALUES
(1, 'Yellow', '00:04:29', 1), (2, 'Whenever, Wherever', '00:03:18', 2),
(3, 'Hello', '00:04:55', 3), (4, 'Levitating', '00:03:23', 4),
(5, 'Safaera', '00:04:55', 5), (6, 'Bichota', '00:02:58', 6),
(7, 'Shape of You', '00:03:53', 7), (8, 'Blank Space', '00:03:51', 8),
(9, 'God's Plan', '00:03:18', 9), (10, 'Saoko', '00:02:17', 10),
(11, 'Blinding Lights', '00:03:20', 11), (12, 'Boy With Luv', '00:03:50', 12),
(13, 'Therefore I Am', '00:02:54', 13), (14, 'Hawái', '00:03:18', 14),
(15, 'Radioactive', '00:03:06', 15), (16, 'Good For You', '00:03:41', 16),
(17, 'Échame la Culpa', '00:02:53', 17), (18, 'Buenos Aires', '00:03:25', 18),
(19, 'Watermelon Sugar', '00:02:54', 19), (20, 'Rojo', '00:02:50', 20);

-- REPRODUCCIONES
INSERT INTO reproduccion (id_usuario, id_cancion, fecha_reproduccion) VALUES
(1, 1, '2025-03-01'), (1, 2, '2025-03-02'), (1, 5, '2025-03-02'),
(2, 3, '2025-03-01'), (2, 4, '2025-03-03'), (2, 6, '2025-03-04'),
(3, 7, '2025-03-01'), (3, 8, '2025-03-01'), (3, 9, '2025-03-02'),
(4, 10, '2025-03-01'), (4, 11, '2025-03-03'), (4, 12, '2025-03-04'),
(5, 13, '2025-03-01'), (5, 14, '2025-03-02'), (5, 15, '2025-03-03'),
(6, 16, '2025-03-01'), (6, 17, '2025-03-03'), (6, 18, '2025-03-04'),
(7, 19, '2025-03-02'), (7, 20, '2025-03-03');
```


Consulta SQL.

Se listan las canciones reproducidas por un usuario específico, mostrando también el nombre del artista y del álbum. Se usa INNER JOIN en todas las tablas porque solo se muestran datos completos con relaciones válidas.

```
1 SELECT
2     u.nombre AS usuario,
3     c.titulo AS cancion,
4     a.titulo AS album,
5     ar.nombre AS artista,
6     r.fecha_reproduccion
7 FROM
8     reproduccion r
9 INNER JOIN usuario u ON r.id_usuario = u.id_usuario
10 INNER JOIN cancion c ON r.id_cancion = c.id_cancion
11 INNER JOIN album a ON c.id_album = a.id_album
12 INNER JOIN artista ar ON a.id_artista = ar.id_artista
13 WHERE
14     u.nombre = 'Ana Pérez'
15 ORDER BY
16     r.fecha_reproduccion ASC, c.titulo ASC;
```

Data Output Mensajes Notificaciones

SQL

	usuario character varying (100)	cancion character varying (100)	album character varying (100)	artista character varying (100)	fecha_reproduccion date
1	Ana Pérez	Yellow	Parachutes	Coldplay	2025-03-01
2	Ana Pérez	Safaera	YHLQMDLG	Bad Bunny	2025-03-02
3	Ana Pérez	Whenever, Wherever	Laundry Service	Shakira	2025-03-02

4. SISTEMA DE CONTROL DE PROYECTOS.

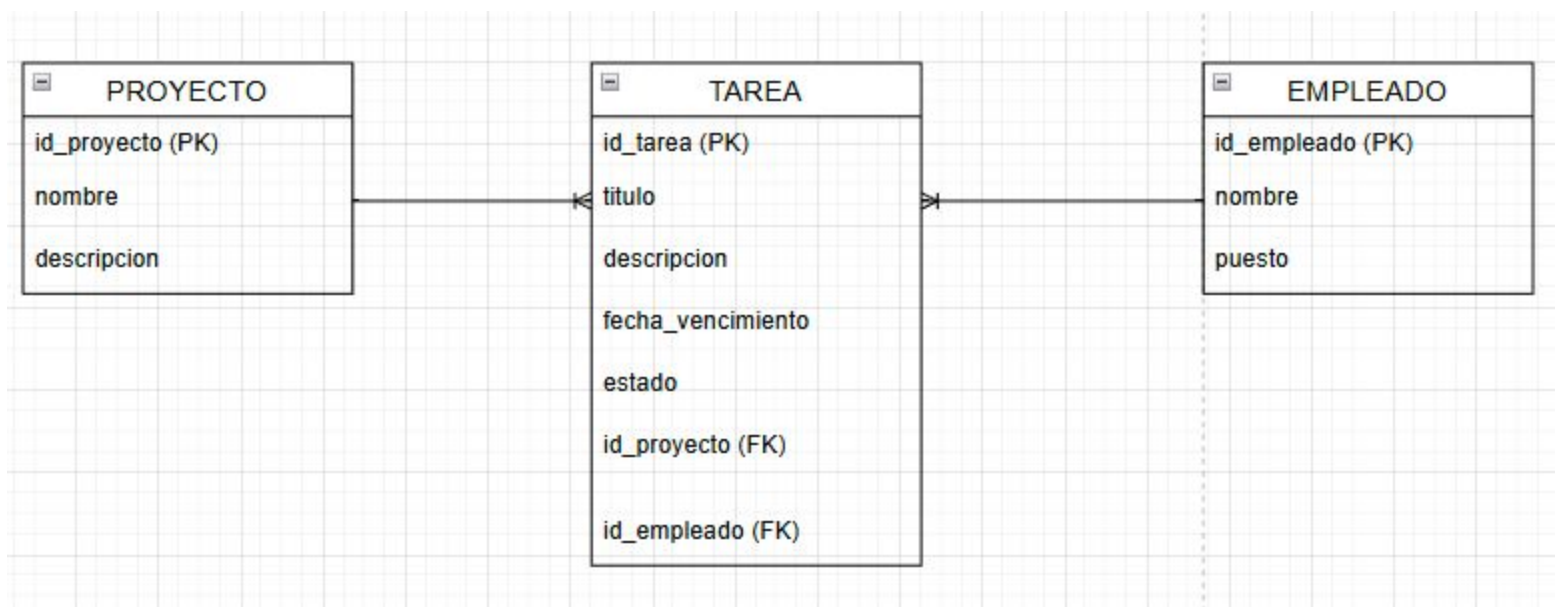
Identificación de entidades clave

Proyecto: Contiene información del proyecto como id_proyecto, nombre, descripcion y fecha_inicio.

Empleado: Incluye id_empleado, nombre, correo y puesto.

Tarea: Relacionada con un proyecto y asignada a un empleado. Tiene id_tarea, descripcion, fecha_vencimiento, estado, id_proyecto y id_empleado.

Diagrama Entidad-Relación



Esquema relacional.

Define las claves primarias
PK

y foráneas FK para
relacionar las tablas

Proyecto: (id_proyecto (PK), nombre, descripcion, fecha_inicio)

Empleado: (id_empleado (PK), nombre, correo, puesto)

Tarea: (id_tarea (PK), descripcion, fecha_vencimiento, estado, id_proyecto (FK), id_empleado (FK))

Creación de tablas con SQL.

```
CREATE TABLE proyecto (  
    id_proyecto INT PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    descripcion TEXT,  
    fecha_inicio DATE  
);  
  
CREATE TABLE empleado (  
    id_empleado INT PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    correo VARCHAR(100),  
    puesto VARCHAR(50)  
);  
  
CREATE TABLE tarea (  
    id_tarea INT PRIMARY KEY,  
    descripcion TEXT NOT NULL,  
    fecha_vencimiento DATE,  
    estado VARCHAR(20), -- 'pendiente', 'en progreso', 'completada', etc.  
    id_proyecto INT,  
    id_empleado INT,  
    FOREIGN KEY (id_proyecto) REFERENCES proyecto(id_proyecto),  
    FOREIGN KEY (id_empleado) REFERENCES empleado(id_empleado)  
);
```

Implementación en SQL con INSERT INTO.

```
-- PROYECTOS
INSERT INTO proyecto (id_proyecto, nombre, descripcion, fecha_inicio) VALUES
(1, 'Sistema de Ventas', 'Desarrollo de software para gestión de ventas', '2025-01-10'),
(2, 'App Móvil', 'Aplicación para dispositivos Android', '2025-01-15'),
(3, 'Intranet Empresarial', 'Red de comunicación interna', '2025-02-01'),
(4, 'Página Web', 'Sitio institucional de la empresa', '2025-02-10'),
(5, 'CRM', 'Sistema de gestión de clientes', '2025-02-15'),
(6, 'ERP', 'Sistema integrado de recursos', '2025-02-20'),
(7, 'ChatBot', 'Asistente virtual para clientes', '2025-02-25'),
(8, 'E-commerce', 'Plataforma de venta online', '2025-03-01'),
(9, 'Rediseño UI', 'Mejora visual de interfaces', '2025-03-05'),
(10, 'Automatización de reportes', 'Generación automática de informes', '2025-03-10'),
(11, 'Sistema de Reservas', 'Reserva de espacios y recursos', '2025-03-15'),
(12, 'Gestión de Inventario', 'Control de stock en almacenes', '2025-03-18'),
(13, 'App de Bienestar', 'Seguimiento de salud de empleados', '2025-03-20'),
(14, 'Encuestas Internas', 'Sistema de retroalimentación', '2025-03-22'),
(15, 'Control de Asistencia', 'Registro de ingreso y salida', '2025-03-24'),
(16, 'Capacitación Online', 'Portal de cursos internos', '2025-03-26'),
(17, 'Sistema de Nómina', 'Gestión de sueldos y pagos', '2025-03-28'),
(18, 'Registro de Clientes', 'Base de datos de clientes', '2025-03-30'),
(19, 'App para Reclutamiento', 'Filtrado y entrevistas', '2025-04-01'),
(20, 'Panel de Indicadores', 'KPIs y visualización de datos', '2025-04-02');
```

```
-- EMPLEADOS
INSERT INTO empleado (id_empleado, nombre, correo, puesto) VALUES
(1, 'Ana Pérez', 'ana@mail.com', 'Desarrollador'),
(2, 'Luis Gómez', 'luis@mail.com', 'Analista'),
(3, 'Laura Díaz', 'laura@mail.com', 'Diseñador'),
(4, 'Carlos Ruiz', 'carlos@mail.com', 'Desarrollador'),
(5, 'Sofía Martínez', 'sofia@mail.com', 'Tester'),
(6, 'Jorge Torres', 'jorge@mail.com', 'Scrum Master'),
(7, 'Camila Ortega', 'camila@mail.com', 'Product Owner'),
(8, 'Daniela Mora', 'daniela@mail.com', 'Desarrollador'),
(9, 'Pedro Vázquez', 'pedro@mail.com', 'Analista'),
(10, 'Lucía Ríos', 'lucia@mail.com', 'Diseñador'),
(11, 'Tomás Suárez', 'tomas@mail.com', 'Desarrollador'),
(12, 'María Figueroa', 'maria@mail.com', 'Tester'),
(13, 'Iván Vega', 'ivan@mail.com', 'Líder Técnico'),
(14, 'Rebeca Castillo', 'rebeca@mail.com', 'QA'),
(15, 'Julieta Luna', 'julieta@mail.com', 'Diseñador'),
(16, 'Raúl Herrera', 'raul@mail.com', 'Desarrollador'),
(17, 'Fernanda Paredes', 'fernanda@mail.com', 'Scrum Master'),
(18, 'Martín Aguilar', 'martin@mail.com', 'DevOps'),
(19, 'Patricia León', 'patricia@mail.com', 'Tester'),
(20, 'Gabriel Campos', 'gabriel@mail.com', 'Desarrollador');
```

-- TAREAS

```
INSERT INTO tarea (id_tarea, descripcion, fecha_vencimiento, estado, id_proyecto, id_empleado) VALUES
(1, 'Diseñar interfaz principal', '2025-04-10', 'pendiente', 1, 3),
(2, 'Programar backend', '2025-04-15', 'en progreso', 1, 1),
(3, 'Pruebas funcionales', '2025-04-20', 'pendiente', 1, 5),
(4, 'Desarrollar app móvil', '2025-04-25', 'completada', 2, 4),
(5, 'Crear manual de usuario', '2025-05-01', 'pendiente', 2, 2),
(6, 'Integración base de datos', '2025-05-05', 'pendiente', 3, 1),
(7, 'Diseño de dashboard', '2025-05-08', 'pendiente', 3, 10),
(8, 'Diseño base de datos', '2025-04-30', 'completada', 4, 9),
(9, 'Maquetado de pantallas', '2025-05-12', 'pendiente', 4, 15),
(10, 'Revisión de código', '2025-05-10', 'pendiente', 5, 12),
(11, 'Implementación de reportes', '2025-05-15', 'pendiente', 5, 13),
(12, 'Despliegue a producción', '2025-05-20', 'pendiente', 6, 18),
(13, 'Validación de requerimientos', '2025-05-22', 'pendiente', 6, 2),
(14, 'Revisión de diseño', '2025-05-25', 'pendiente', 7, 14),
(15, 'Optimización de consultas', '2025-05-30', 'pendiente', 8, 11),
(16, 'Diseño de interfaz gráfica', '2025-06-01', 'pendiente', 9, 10),
(17, 'Análisis de rendimiento', '2025-06-05', 'pendiente', 10, 13),
(18, 'Documentación del sistema', '2025-06-10', 'pendiente', 10, 12),
(19, 'Carga de datos', '2025-06-12', 'pendiente', 11, 1),
(20, 'Revisión final del sistema', '2025-06-15', 'pendiente', 12, 6);
```


Consulta SQL.











Se muestran todas las tareas pendientes del proyecto llamado "Sistema de Ventas". El RIGHT JOIN asegura que el proyecto aparezca incluso si no tiene tareas asociadas (aunque el WHERE las filtra como "pendientes").

```
1 SELECT
2     t.descripcion,
3     t.fecha_vencimiento,
4     e.nombre AS empleado,
5     p.nombre AS proyecto
6 FROM
7     tarea t
8 INNER JOIN empleado e ON t.id_empleado = e.id_empleado
9 RIGHT JOIN proyecto p ON t.id_proyecto = p.id_proyecto
10 WHERE
11     t.estado = 'pendiente'
12     AND p.nombre = 'Sistema de Ventas'
13 ORDER BY
14     t.fecha_vencimiento ASC;
```

Data Output

Mensajes

Notificaciones



	descripcion text	fecha_vencimiento date	empleado character varying (100)	proyecto character varying (100)
1	Diseñar interfaz principal	2025-04-10	Laura Díaz	Sistema de Ventas
2	Pruebas funcionales	2025-04-20	Sofía Martínez	Sistema de Ventas

5. SISTEMA DE EVALUACIÓN ACADÉMICA

Identificación de entidades clave

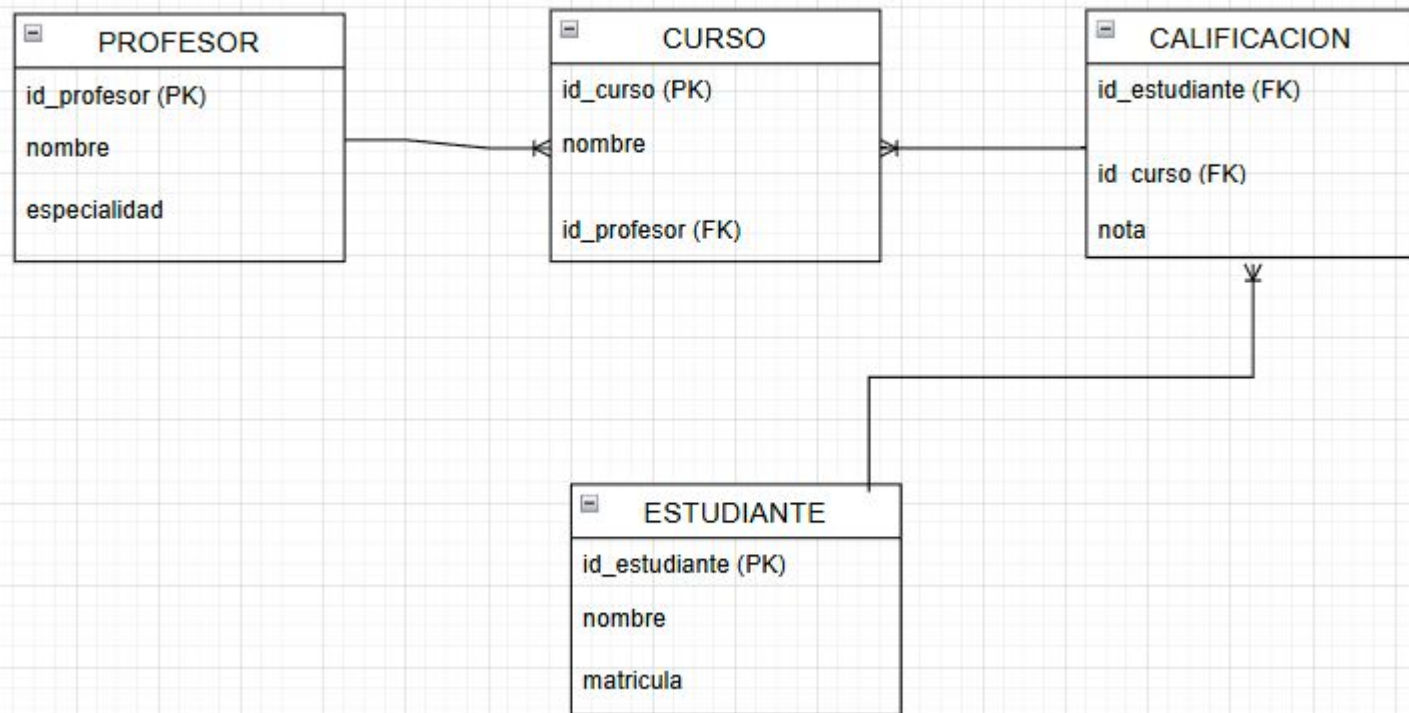
Estudiante: Contiene la información del estudiante como id_estudiante, nombre y correo.

Curso: Incluye id_curso, nombre y id_profesor.

Profesor: Representa a quien imparte el curso con id_profesor, nombre y especialidad.

Calificación: Relaciona estudiantes y cursos mediante id_estudiante, id_curso, calificacion y fecha.

Diagrama Entidad-Relación



Esquema relacional

Define las claves primarias
PK
y foráneas FK para
relacionar las tablas

Estudiante: (id_estudiante (PK),
nombre, correo)

Profesor: (id_profesor (PK),
nombre, especialidad)

Curso: (id_curso (PK), nombre,
id_profesor (FK))

Calificacion: (id_estudiante (FK),
id_curso (FK), calificacion, fecha)

Creación de tablas con SQL.

```
CREATE TABLE estudiante (  
    id_estudiante INT PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    correo VARCHAR(100)  
);  
  
CREATE TABLE profesor (  
    id_profesor INT PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    especialidad VARCHAR(100)  
);  
  
CREATE TABLE curso (  
    id_curso INT PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    id_profesor INT,  
    FOREIGN KEY (id_profesor) REFERENCES profesor(id_profesor)  
);  
  
CREATE TABLE calificacion (  
    id_estudiante INT,  
    id_curso INT,  
    calificacion DECIMAL(4,2),  
    fecha DATE,  
    PRIMARY KEY (id_estudiante, id_curso),  
    FOREIGN KEY (id_estudiante) REFERENCES estudiante(id_estudiante),  
    FOREIGN KEY (id_curso) REFERENCES curso(id_curso)  
);
```

Implementación en SQL con INSERT INTO.

```
-- ESTUDIANTES
INSERT INTO estudiante (id_estudiante, nombre, correo) VALUES
(1, 'Ana Pérez', 'ana@mail.com'), (2, 'Luis Gómez', 'luis@mail.com'),
(3, 'Laura Díaz', 'laura@mail.com'), (4, 'Carlos Ruiz', 'carlos@mail.com'),
(5, 'Sofía Martínez', 'sofia@mail.com'), (6, 'Jorge Torres', 'jorge@mail.com'),
(7, 'Camila Ortega', 'camila@mail.com'), (8, 'Daniela Mora', 'daniela@mail.com'),
(9, 'Pedro Vázquez', 'pedro@mail.com'), (10, 'Lucía Ríos', 'lucia@mail.com'),
(11, 'Tomás Suárez', 'tomas@mail.com'), (12, 'María Figueroa', 'maria@mail.com'),
(13, 'Iván Vega', 'ivan@mail.com'), (14, 'Rebeca Castillo', 'rebeca@mail.com'),
(15, 'Julieta Luna', 'julieta@mail.com'), (16, 'Raúl Herrera', 'raul@mail.com'),
(17, 'Fernanda Paredes', 'fernanda@mail.com'), (18, 'Martín Aguilar', 'martin@mail.com'),
(19, 'Patricia León', 'patricia@mail.com'), (20, 'Gabriel Campos', 'gabriel@mail.com');

-- PROFESORES
INSERT INTO profesor (id_profesor, nombre, especialidad) VALUES
(1, 'Marta Salinas', 'Matemáticas'), (2, 'Raúl Navarro', 'Física'),
(3, 'Claudia Reyes', 'Química'), (4, 'Héctor Mendoza', 'Programación'),
(5, 'Paula Gómez', 'Bases de Datos'), (6, 'Alberto Fuentes', 'Estadística'),
(7, 'Rocío Hernández', 'Literatura'), (8, 'Daniel Ríos', 'Cálculo'),
(9, 'Gabriela Sánchez', 'Álgebra'), (10, 'Pablo López', 'Redes'),
(11, 'Natalia Castro', 'Historia'), (12, 'Óscar Vargas', 'Lógica'),
(13, 'Luciana Bravo', 'Inteligencia Artificial'), (14, 'Andrés Soto', 'Ética'),
(15, 'Verónica Torres', 'Sistemas Operativos'), (16, 'Leonardo Peña', 'Diseño Gráfico'),
(17, 'Esteban Morales', 'Comunicación'), (18, 'Liliana Rivas', 'Psicología'),
(19, 'César Salgado', 'Filosofía'), (20, 'Isabel Ramírez', 'Biología');
```

```
-- CURSOS
```

```
INSERT INTO curso (id_curso, nombre, id_profesor) VALUES  
(1, 'Álgebra I', 9), (2, 'Cálculo I', 8), (3, 'Bases de Datos', 5),  
(4, 'Programación I', 4), (5, 'Matemáticas Discretas', 1),  
(6, 'Redes', 10), (7, 'Estadística', 6), (8, 'Literatura', 7),  
(9, 'Filosofía', 19), (10, 'Psicología', 18), (11, 'Biología', 20),  
(12, 'Diseño Gráfico', 16), (13, 'Historia', 11), (14, 'Lógica', 12),  
(15, 'Ética', 14), (16, 'Inteligencia Artificial', 13),  
(17, 'Comunicación Oral', 17), (18, 'Sistemas Operativos', 15),  
(19, 'Química', 3), (20, 'Física', 2);
```

```
-- CALIFICACIONES
```

```
INSERT INTO calificacion (id_estudiante, id_curso, calificacion, fecha) VALUES  
(1, 1, 85.00, '2025-04-01'), (1, 2, 90.00, '2025-04-02'),  
(1, 3, 88.50, '2025-04-03'), (1, 4, 92.00, '2025-04-04'),  
(2, 1, 77.50, '2025-04-01'), (2, 3, 80.00, '2025-04-03'),  
(2, 5, 83.00, '2025-04-05'), (3, 2, 95.00, '2025-04-02'),  
(3, 4, 89.00, '2025-04-04'), (3, 6, 91.00, '2025-04-06'),  
(4, 1, 60.00, '2025-04-01'), (4, 2, 70.00, '2025-04-02'),  
(4, 3, 65.00, '2025-04-03'), (5, 7, 88.00, '2025-04-07'),  
(5, 8, 90.00, '2025-04-08'), (5, 9, 87.00, '2025-04-09'),  
(6, 1, 92.00, '2025-04-01'), (6, 2, 85.00, '2025-04-02'),  
(6, 3, 89.00, '2025-04-03'), (6, 4, 93.00, '2025-04-04');
```

Consulta SQL.

Se calcula el promedio de calificaciones de una estudiante. El INNER JOIN se usa porque solo se consideran las calificaciones que existen y están relacionadas con ese estudiante.

```
1  SELECT
2      e.nombre AS estudiante,
3      AVG(ca.calificacion) AS promedio
4  FROM
5      calificacion ca
6  INNER JOIN estudiante e ON ca.id_estudiante = e.id_estudiante
7  WHERE
8      e.nombre = 'Laura Díaz'
9  GROUP BY
10     e.nombre;
```

Data Output Mensajes Notificaciones

	estudiante character varying (100) 🔒	promedio numeric 🔒
1	Laura Díaz	91.6666666666666667