

Nombre:

Fecha:

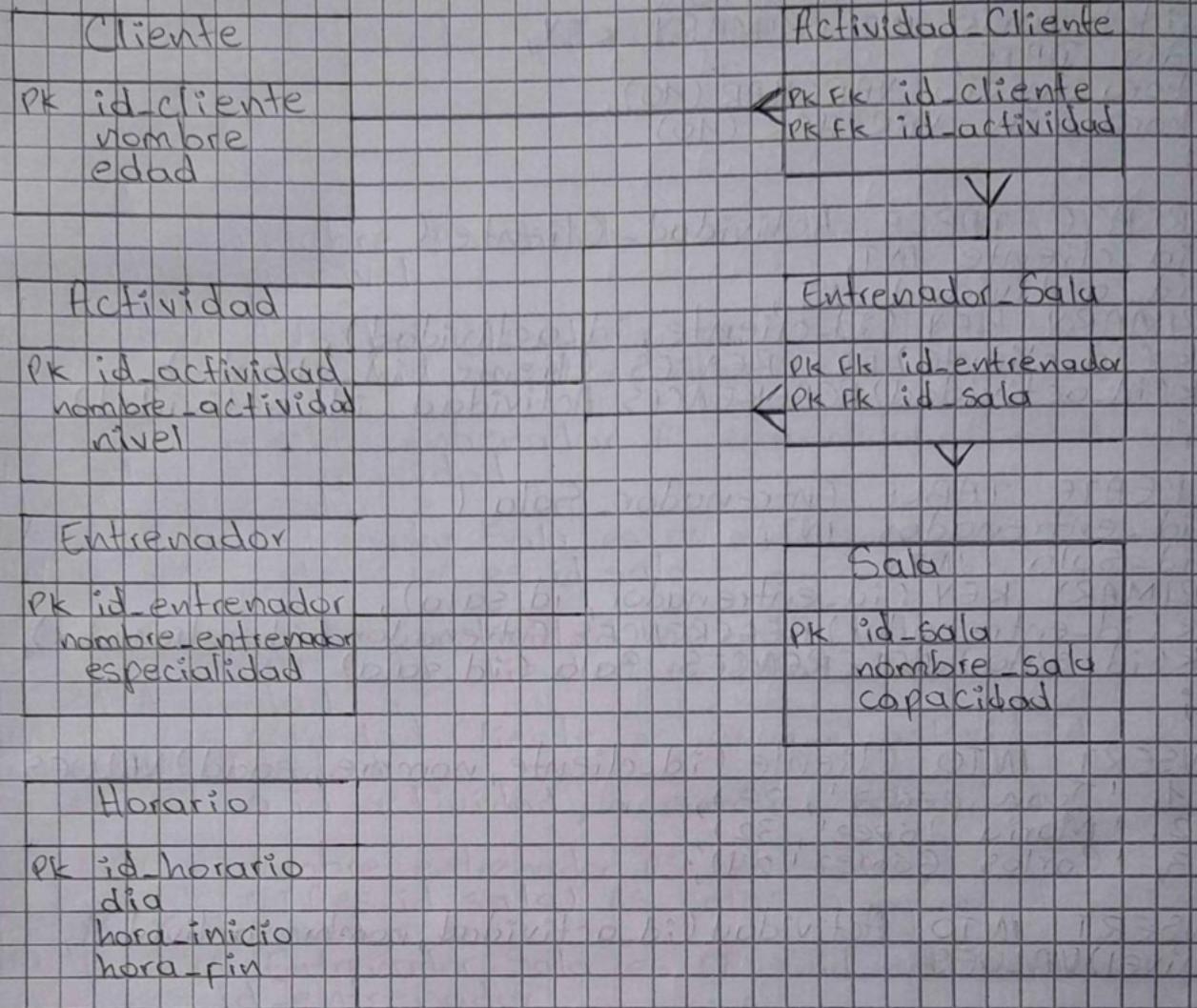
Profesor:

Materia:

Institución:

Curso:

Nota:



CREATE DATABASE Gimnasio;
USE Gimnasio;

CREATE TABLE Cliente (
 id_cliente INT PRIMARY KEY,
 nombre VARCHAR(100),
 edad INT);

CREATE TABLE Actividad (
 id_actividad INT PRIMARY KEY,
 nombre_actividad VARCHAR(100),
 nivel VARCHAR(20)
);

CREATE TABLE Entrenador (
 id_entrenador INT PRIMARY KEY,
 nombre_entrenador VARCHAR(100),
 especialidad VARCHAR(50)
);

```
CREATE TABLE Sala (
    id_sala INT PRIMARY KEY,
    nombre_sala VARCHAR(100),
    capacidad INT
);
```

```
CREATE TABLE Horario (
    id_horario INT PRIMARY KEY,
    dia DATE,
    hora_inicio VARCHAR(10),
    hora_fin VARCHAR(10)
);
```

```
CREATE TABLE Actividad_Cliente (
    id_cliente INT,
    id_actividad INT,
    PRIMARY KEY (id_cliente, id_actividad),
    FK (id_cliente) REFERENCES Cliente (id_cliente),
    FK (id_actividad) REFERENCES Actividad (id_actividad)
);
```

```
CREATE TABLE Entrenador_Sala (
    id_entrenador INT,
    id_sala INT,
    PRIMARY KEY (id_entrenador, id_sala),
    FK (id_entrenador) REFERENCES Entrenador (id_entrenador),
    FK (id_sala) REFERENCES Sala (id_sala)
);
```

```
INSERT INTO Cliente (id_cliente, nombre, edad) VALUES
(1, 'Juan Pérez', 28),
(2, 'María López', 32),
(3, 'Carlos Gómez', 24);
```

```
INSERT INTO Actividad (id_actividad, nombre_actividad, nivel) VALUES
```

```
(101, 'Yoga', 'básico'),
(102, 'Spinning', 'intermedio'),
(103, 'Pilates', 'avanzado');
```

```
INSERT INTO Entrenador (id_entrenador, nombre_entrenador, especialidad) VALUES
```

```
(201, 'Ana Suárez', 'Fuerza'),
(202, 'Roberto Daza', 'Cardio'),
(203, 'Luis Torres', 'Funcional');
```

```
INSERT INTO Sala (id_sala, nombre_sala, capacidad) VALUES
(301, 'Sala 1', 20),
(302, 'Sala 2', 15),
(303, 'Sala 3', 25);
```

```
INSERT INTO Horario (id_horario, dia, hora_inicio, hora_fin) VALUES
(401, '2025-07-01', '08:00', '09:00'),
(402, '2025-07-02', '09:00', '10:00'),
(403, '2025-07-03', '18:00', '19:00');
```

```
INSERT INTO Actividad-Cliente (id-cliente, id-actividad)
VALUES
(1, 101),
(2, 102),
(3, 103);
```

```
INSERT INTO Entrenador-Sala (id-entrenador, id-sala) VALUES
(201, 301),
(202, 302),
(203, 303);
```

```
SELECT c.nombre AS cliente, a.nombre_actividad AS actividad, a.nivel
FROM Cliente c
INNER JOIN Actividad-Cliente ac ON c.id-cliente = ac.id-cliente
INNER JOIN Actividad a ON ac.id-actividad = a.id-actividad;
```

```
SELECT e.nombre_entrenador AS entrenador, s.nombre_sala AS sala, s.capacidad
FROM Entrenador e
INNER JOIN Entrenador-Sala es ON e.id-entrenador = es.id-entrenador
INNER JOIN Sala s ON es.id-sala = s.id-sala;
```

```
SELECT a.nombre_actividad AS actividad,
COUNT(ac.id-cliente) AS total_clientes
FROM Actividad a
INNER JOIN Actividad-Cliente ac ON a.id-actividad = ac.id-actividad
GROUP BY a.id-actividad, a.nombre-actividad;
```

```
SELECT e.nombre_entrenador AS entrenador,
COUNT(es.id-sala) AS salas_asignadas
FROM Entrenador e
INNER JOIN Entrenador-Sala es ON e.id-entrenador = es.id-entrenador
GROUP BY e.id-entrenador, e.nombre_entrenador;
```

```
SELECT s.nombre_sala AS sala, e.nombre_entrenador
FROM Sala s
INNER JOIN Entrenador-Sala es ON s.id-sala = es.id-sala
INNER JOIN Entrenador e ON es.id-entrenador = e.id-entrenador;
```

```
SELECT nombre_entrenador
FROM Entrenador
WHERE id_entrenador IN (
    SELECT id_entrenador
    FROM Entrenador-Sala
    WHERE id_sala IN (
        SELECT id_sala
        FROM Sala
        WHERE capacidad > 20 )
);
```

```
SELECT nombre
FROM Cliente
WHERE id_cliente IN (
    SELECT id_cliente
    FROM Actividad_Cliente
    WHERE id_actividad IN (
        SELECT id_actividad
        FROM Actividad
        WHERE nivel = 'avanzado'
    )
);
```

```
SELECT nombre_actividad
FROM Actividad
WHERE id_actividad IN (
    SELECT id_actividad
    FROM Actividad_Cliente
    WHERE id_cliente IN (
        SELECT id_cliente
        FROM Cliente
        WHERE edad < 26
    )
);
```

```
SELECT nombre_sala
FROM Sala
WHERE id_sala IN (
    SELECT id_sala
    FROM Entrenador_Sala
    WHERE id_entrenador IN (
        SELECT id_entrenador
        FROM Entrenador
        WHERE especialidad = 'Cardio'
    )
);
```

```
SELECT nombre
FROM Cliente
WHERE id_cliente IN (
    SELECT id_cliente
    FROM Actividad_Cliente
    WHERE id_actividad
    SELECT id_actividad
    FROM Actividad
    WHERE nivel = 'básico'
);
```

```
SELECT MAX(capacidad) AS capacidad_maxima
FROM Sala;
```

```
SELECT a.nivel,
       AVG(c.edad) AS edad_promedio
FROM Cliente c
INNER JOIN Actividad_Cliente ac ON c.id_cliente = ac.id_cliente
INNER JOIN Actividad a ON ac.id_actividad = a.id_actividad
GROUP BY a.nivel;
```

```
SELECT MAX(nombre_actividad) AS actividades_maxima
FROM Actividad;
```

Nombre:

Fecha:

dom

lun

mié

Profesor:

Materia:

Institución:

Curso:

Nota:

```
SELECT CONCAT(nombre_entrenador, ' - ', especialidad)
AS entrenador_info
FROM Entrenador;
```

```
SELECT id_actividad,
UPPER(nombre_actividad) AS actividad_mayuscula
FROM Actividad;
```

```
SELECT MIN(edad) AS edad_minima
FROM Cliente;
```

```
SELECT MAX(hora_inicio) AS hora_maxima
FROM Horario;
```

```
SELECT COUNT(*) AS total_actividades
FROM Actividades;
```

```
SELECT c.nombre,
(SELECT COUNT(*)
FROM Actividad_Cliente ac
WHERE ac.id_cliente = c.id_cliente) AS total_actividades
FROM Cliente c;
```

```
SELECT CONCAT(nombre, ' - edad: ', edad) AS cliente_info
FROM Cliente;
```

Procedimiento

```
CREATE PROCEDURE RegistrarCliente (
```

```
    IN p_id_cliente INT,
    IN p_nombre VARCHAR(100),
    IN p_edad INT
)
```

```
BEGIN
```

```
    INSERT INTO Cliente(id_cliente, nombre, edad)
    VALUES (p_id_cliente, p_nombre, p_edad);
```

```
END
```

```
EXEC RegistrarCliente(4, 'johan acero', 18);
```

```
UPDATE Cliente SET edad = 30  
WHERE id_cliente = 1;  
UPDATE Actividad SET nivel = 'intermedio'  
WHERE id_actividad = 101;  
UPDATE Entrenador SET especialidad = 'yoga'  
WHERE id_entrenador = 202;  
UPDATE Sala SET capacidad = 30  
WHERE id_sala = 302;  
UPDATE Horario SET hora-fin = '10:30'  
WHERE id_horario = 402;
```

```
DELETE FROM Actividad_Cliente WHERE id_cliente = 3  
AND id_actividad = 103;  
DELETE FROM Entrenador_Sala WHERE id_entrenador = 202  
AND id_sala = 302;  
DELETE FROM Horario WHERE id_horario = 403;  
DELETE FROM Sala WHERE id_sala = 303;  
DELETE FROM Cliente WHERE id_cliente = 3;
```

```
ALTER TABLE Cliente ADD telefono VARCHAR(20);  
ALTER TABLE Actividad DROP COLUMN nivel;  
ALTER TABLE Sala ADD piso INT;  
ALTER TABLE Entrenador ADD edad INT;  
ALTER TABLE Horario DROP COLUMN hora-fin;
```

```
TRUNCATE TABLE Actividad_Cliente;  
TRUNCATE TABLE Entrenador_Sala;  
TRUNCATE TABLE Horario;  
TRUNCATE TABLE Sala;  
TRUNCATE TABLE Cliente;
```

```
DROP TABLE Actividad_Cliente;  
DROP TABLE Entrenador_Sala;  
DROP TABLE Actividad;  
DROP TABLE Entrenador;  
DROP TABLE Cliente;
```

Usuario

PK id_usuario
nombre
correo

Libro -

PK id-libro
titulo
año_publicacion

Categoría

PK id_categoria
nombre_categoria
descripcion

Prestamo

PK id-prestamo
FK id_usuario
FK id-libro
fecha_prestamo
fecha_devolucion

Categoría Libro

PK FK id-libro
PK FK id_categoria

Autor

PK id-autor
nombre_autor
nacionalidad

Libro-Autor

PK FK id-libro
PK FK id-autor

```
CREATE DATABASE Biblioteca;
USE Biblioteca;
```

```
CREATE TABLE Usuario (
    id_usuario INT PRIMARY KEY,
    nombre VARCHAR(100),
    correo VARCHAR(100)
```

```
CREATE TABLE Libro (
    id-libro INT PRIMARY KEY,
    titulo VARCHAR(150),
    año-publicación INT
```

```
CREATE TABLE Prestamo (
    id-prestamo INT PRIMARY KEY,
    id_usuario INT,
    id-libro INT
    Fecha_prestamo DATE,
    fecha_devolucion DATE,
    FOREIGN KEY (id_usuario) REFERENCES Usuario (id_usuario),
    FOREIGN KEY (id-libro) REFERENCES Libro (id-libro)
```

```
CREATE TABLE Categoría (
    id-categoría INT PRIMARY KEY,
    nombre-categoría VARCHAR(100),
    descripción VARCHAR(100)
```

CREATE TABLE Autor (

id-autor INT PRIMARY KEY,

nombre-autor VARCHAR(100),

nacionalidad VARCHAR(50)

CREATE TABLE Categoria-Libro (

id-libro INT

id-Categoría INT

PRIMARY KEY (id-libro, id-Categoría),

FOREIGN KEY (id-libro) REFERENCES Libro(id-libro),

FOREIGN KEY (id-Categoría) REFERENCES Categoría(id-Categoría)

CREATE TABLE Libro-autor (

id-libro INT

id-autor INT,

PRIMARY KEY (id-libro, id-autor)

FOREIGN KEY (id-libro) REFERENCES Libro(id-libro),

FOREIGN KEY (id-autor) REFERENCES Autor(id-autor)

INSERT INTO Usuario (id-usuario, nombre, correo) VALUES

(1, 'ana Pérez', 'ana@correo.com'),

(2, 'Luis gómez', 'luis@correo.com'),

(3, 'maría díaz', 'maria@correo.com');

INSERT INTO Libro (id-libro, título, año-publicación) VALUES

(101, 'Cien años de Soledad', 1967),

(102, 'el Principito', 1943),

(103, 'La sombra del Viento', 2001)

INSERT INTO Autor (id-autor, nombre-autor, nacionalidad) VALUES

(201, 'Gabriel García Márquez', 'colombiana'),

(202, 'Antoine de Saint-Exupéry', 'francesa'),

(203, 'Carlos Ruiz Zafón', 'española');

INSERT INTO Categoría (id-Categoría, nombre-Categoría, descripción) VALUES

(301, 'novela', 'ficción narrativa'),

(302, 'infantil', 'lecturas para niños'),

(303, 'misterio', 'novela de suspense');

INSERT INTO Prestamo (id-prestamo, id-usuario, id-libro, fecha-prestamo, fecha-devolución) VALUES

(401, 1, 101, '2025-07-01', '2025-07-15'),

(402, 2, 102, '2025-07-02', '2025-07-12'),

(403, 3, 103, '2025-07-03', NULL);

INSERT INTO Categoría-libro (id-libro, id-Categoría) VALUES

(101, 301),

(102, 302),

(103, 303);

INSERT INTO Libro-autor (id-libro, id-autor) VALUES

(101, 201),

(102, 202),

(103, 203);

Nombre:

Fecha:

Profesor:

Materia:

Institución:

Curso:

Nota:

SELECT u.nombre AS usuario, l.titulo AS libro
 FROM usuario u

INNER JOIN Prestamo p ON u.id_usuario = p.id_usuario,
 INNER JOIN Libro l ON p.id_libro = l.id_libro;

SELECT l.titulo AS libro, a.nombre_autor AS autor.
 FROM Libro l

INNER JOIN Libro_Autor la ON l.id_libro = la.id_libro
 INNER JOIN Autor a ON la.id_autor = a.id_autor;

SELECT l.titulo AS libro, c.nombre_categoria AS categoria
 FROM Libro l
 INNER JOIN Categoria_Libro cl ON l.id_libro = cl.id_libro
 INNER JOIN Categoria c ON cl.id_categoria = c.id_categoria;

SELECT p.id_prestamo, u.nombre AS usuario, l.titulo AS libro,
 p.fecha_prestamo
 FROM prestamo p
 INNER JOIN usuario u ON p.id_usuario = u.id_usuario
 INNER JOIN Libro l ON p.id_libro = l.id_libro;

SELECT titulo
 FROM Libro
 WHERE id_libro IN
 (SELECT id_libro
 FROM prestamo
 WHERE id_usuario = (SELECT id_usuario
 FROM usuario
 WHERE nombre = 'Ana Pérez'))

SELECT nombre_autor.
 FROM Autor
 WHERE id_autor IN
 (SELECT id_autor
 FROM Libro_Autor
 WHERE id_libro IN
 (SELECT id_libro
 FROM Libro
 WHERE año_publicación > 2000))

SELECT nombre_categoria
 FROM Categoria
 WHERE id_categoria IN
 (SELECT id_categoria
 FROM Categoria_Libro
 WHERE id_libro IN
 (SELECT id_libro
 FROM prestamo)))

```
SELECT titulo  
FROM Libro  
WHERE id-libro IN C  
SELECT id-libro  
FROM Prestamo  
GROUP BY id-libro  
HAVING COUNT(*) > 1
```

```
SELECT titulo  
FROM Libro  
WHERE id-libro IN C  
SELECT id-libro  
FROM Libro-Autor  
WHERE id-autor IN (SELECT id-autor  
FROM Autor  
WHERE nacionalidad = 'colombiana')
```

```
SELECT u.nombre AS usuario  
COUNT(p.id-prestamo) AS total_prestamo  
FROM usuario u  
LEFT JOIN Prestamo p ON u.id-usuario = p.id-usuario  
GROUP BY u.id-usuario, u.nombre;
```

```
SELECT a.nombre-autor AS autor,  
AVG(l.anio-publicación) AS promedio_año  
FROM Autor a  
JOIN Libro-autor la ON a.id-autor = la.id-autor  
JOIN libro l ON la.id-libro = l.id-libro  
GROUP BY a.id-autor, a.nombre-autor;
```

```
SELECT MAX(anio-publicación) AS año_maximo  
FROM libro;
```

```
SELECT l.titulo  
COUNT(cl.id-categoría) AS total_categoría  
FROM libro l  
LEFT JOIN Categoría-libro cl ON l.id-libro = cl.id-libro  
GROUP BY l.id-libro, l.titulo;
```

```
SELECT CONCAT(a.nombre-categoría, ' - ', c.descripcion) AS  
Categoría-información  
FROM Categoría c;
```

```
SELECT CONCAT(a.nombre-autor, ' - ', a.nacionalidad) AS  
autor_info  
FROM Autor a;
```

```
SELECT a.nacionalidad  
COUNT(la.id-libro) AS total-libros  
FROM Autor a  
JOIN Libro-autor la ON a.id-autor = la.id-autor  
GROUP BY a.nacionalidad;
```

```
SELECT MIN(Fecha-prestamo) AS fecha-minima-prestamo  
FROM prestamo;
```

```
SELECT MAX(fecha_devolucion) AS fecha_maxima_prestamo  
FROM Prestamo;
```

```
SELECT UPPER(nombre_autor) AS autor_mayuscula  
FROM Autor;
```

PROCEDIMIENTO

```
CREATE PROCEDURE RegistrarLibro
```

```
@id-libro INT,  
@titulo VARCHAR(150),  
@ano_publicacion INT  
AS  
BEGIN  
    INSERT INTO Libro (id_libro, titulo, ano_publicacion)  
    VALUES (@id-libro, @titulo, @ano_publicacion);  
END;  
GO
```

```
EXEC RegistrarLibro @id-libro = 104, @titulo = 'Señor de los  
anillos', @ano_publicacion = 1954;
```

```
UPDATE Usuari0 SET nombre = 'ana p. pérez'  
WHERE id_usuari0 = 1;  
UPDATE Libro SET año_publicación = 2005  
WHERE id_libro = 103;  
UPDATE Prestamo SET fecha_devolucion = '2025-07-20'  
WHERE id_prestamo = 404;  
UPDATE Categoría SET nombre_categoria = 'clasicos'  
WHERE id_categoria = 301;  
UPDATE Autor SET nacionalidad = 'mexicana'  
WHERE id_autor = 201;
```

```
DELETE FROM Prestamo WHERE id_prestamo = 403;  
DELETE FROM Categoría_Libro WHERE id_libr0 = 102  
AND id_categoria = 302;  
DELETE FROM Libro_Autor WHERE id_libr0 = 101 AND  
id_autor = 201;  
DELETE FROM Libro WHERE id_libr0 = 103;  
DELETE FROM Usuari0 WHERE id_usuari0 = 3;
```

```
ALTER TABLE Usuari0 ADD telefono VARCHAR(20);  
ALTER TABLE Categoría DROP COLUMN descripción;  
ALTER TABLE Libro ADD isbn VARCHAR(20);  
ALTER TABLE Prestamo DROP COLUMN fecha_devolución;  
ALTER TABLE Usuari0 ADD fecha_registro DATE, PROP  
COLUMN telefono;
```

```
TRUNCATE TABLE Prestamo;  
TRUNCATE TABLE Categoría_Libro;  
TRUNCATE TABLE Libro_Autor;  
TRUNCATE TABLE Categoría;  
TRUNCATE TABLE Autor;
```

```
DROP TABLE Libro;
DROP TABLE Categoria_Libro;
DROP TABLE Libro_Autor;
DROP TABLE Categoria;
DROP TABLE Prestamo;
```

Nombre:

Fecha:

Profesor:

Materia:

Institución:

Curso:

Nota:

Funcion

PK id_funcion
 FK id_pelicula
 FK id_sala
 fecha
 hora
 precio_ticket



Sala

PK id_sala
 nombre_sala
 capacidad

Cliente_Funcion

PK FK id_cliente
 PK FK id_funcion
 boletos



Cliente

PK id_cliente
 nombre_cliente
 correo

Pelicula

PK id_pelicula
 titulo
 duracion_min
 clasificacion



Pelicula_Empleado

PK FK id_pelicula
 PK FK id_empleado



Empleado

PK id_empleado
 nombre_empleado
 cargo

CREATE DATABASE Cine;
 USE Cine;

CREATE TABLE Pelicula (
 id_pelicula INT PRIMARY KEY,
 titulo VARCHAR(100),
 duracion_min INT,
 clasificacion VARCHAR(10)
>);

CREATE TABLE Cliente (
 id_cliente INT PRIMARY KEY,
 nombre_cliente VARCHAR(100),
 correo VARCHAR(100)
>);

```
CREATE TABLE Sala (
    id_sala INT PRIMARY KEY,
    nombre_sala VARCHAR (50),
    capacidad INT
);
```

```
CREATE TABLE Funcion (
    id_funcion INT PRIMARY KEY,
    id_pelicula INT,
    id_sala INT,
    fecha DATE,
    hora TIME,
    precio_ticket DECIMAL (6, 2),
    FOREIGN KEY (id_pelicula) REFERENCES Pelicula (id_pelicula),
    FOREIGN KEY (id_sala) REFERENCES Sala (id_sala)
);
```

```
CREATE TABLE Empleado (
    id_empleado INT PRIMARY KEY,
    nombre_empleado VARCHAR (100),
    cargo VARCHAR (50)
);
```

```
CREATE TABLE Pelicula_Emppleado (
    id_pelicula INT,
    id_empleado INT,
    rol VARCHAR (50),
    PRIMARY KEY (id_pelicula, id_empleado),
    FK (id_pelicula) REFERENCES Pelicula (id_pelicula),
    FK (id_empleado) REFERENCES Empleado (id_empleado)
);
```

```
CREATE TABLE Cliente_Funcion (
    id_cliente INT,
    id_funcion INT,
    boletos INT,
    PRIMARY KEY (id_cliente, id_funcion),
    FK (id_cliente) REFERENCES Cliente (id_cliente),
    FK (id_funcion) REFERENCES Funcion (id_funcion)
);
```

```
INSERT INTO Pelicula (id_pelicula, titulo, duracion_min, clasificacion) VALUES
(1, 'inception', 148, 'PG-13'),
(2, 'avatar', 162, 'PG-13'),
(3, 'coco', 105, 'PG');
```

```
INSERT INTO Cliente (id_cliente, nombre_cliente, correo) VALUES
(10, 'ana rodriguez', 'ana@acine.com'),
(11, 'luis martinez', 'luis@acine.com'),
(12, 'maria perez', 'maria@acine.com');
```

```
INSERT INTO Sala (id_sala, nombre_sala, capacidad) VALUES
(20, 'Sala 1', 120),
(21, 'Sala 2', 100),
(22, 'Sala 3', 80);
```

```
INSERT INTO Empleado(id_empleado, nombre_empleado, cargo) VALUES  
(40, 'pedro gonzález', 'Proyecciónista'),  
(41, 'carla suárez', 'Taquilla'),  
(42, 'javier lopez', 'Limpieza');
```

```
INSERT INTO Funcion(id_funcion, id_pelicula, id_sala, fecha, hora, precio_ticket) VALUES
```

```
(30, 1, 20, '2025-08-01', '18:00', 9.500),  
(31, 2, 21, '2025-08-01', '20:00', 12.000),  
(32, 3, 22, '2025-08-02', '16:00', 8.000);
```

```
INSERT INTO Pelicula_Emppleado(id_pelicula, id_empleado, rol) VALUES
```

```
(1, 40, 'proyección'),  
(2, 41, 'venta boletos'),  
(3, 42, 'mantenimiento');
```

```
INSERT INTO Cliente_Funcion(id_cliente, id_funcion_boletos) VALUES
```

```
(10, 30, 2),  
(11, 31, 1),  
(12, 32, 3);
```

```
SELECT f.id_funcion, f.fecha, f.hora, p.titulo, s.nombre_sala  
FROM Funcion f  
INNER JOIN Pelicula p ON f.id_pelicula = p.id_pelicula  
INNER JOIN Sala s ON f.id_sala = s.id_sala;
```

```
SELECT c.nombre_cliente, p.titulo, f.fecha, f.hora  
FROM Cliente c  
INNER JOIN Cliente_Funcion cf ON c.id_cliente = cf.id_cliente  
INNER JOIN Funcion f ON cf.id_funcion = f.id_funcion  
INNER JOIN Pelicula p ON f.id_pelicula = p.id_pelicula;
```

```
SELECT e.nombre_empleado, p.titulo, pe.rol  
FROM Empleado e  
INNER JOIN Pelicula_Emppleado pe ON e.id_empleado = pe.id_empleado  
INNER JOIN Pelicula p ON pe.id_pelicula = p.id_pelicula;
```

```
SELECT s.nombre_sala, f.fecha, f.hora, f.precio_ticket  
FROM Sala s  
INNER JOIN Funcion f ON s.id_sala = f.id_sala;
```

```
SELECT c.nombre_cliente, f.id_funcion, f.fecha, cf.boletos  
FROM Cliente c  
INNER JOIN Cliente_Funcion cf ON c.id_cliente = cf.id_cliente  
INNER JOIN Funcion f ON cf.id_funcion = f.id_funcion;
```

```
SELECT nombre_cliente  
FROM Cliente  
WHERE id_cliente IN  
    (SELECT id_cliente  
     FROM Cliente_Funcion  
     WHERE boletos > 1 );
```

```
SELECT titulo  
FROM Pelicula  
WHERE id_pelicula IN  
    (SELECT id_pelicula  
     FROM Funcion  
     WHERE id_funcion = 30 );
```

```
SELECT nombre_empleado  
FROM Empleado  
WHERE id_empleado IN  
    (SELECT id_empleado  
     FROM Pelicula_Emppleado );
```

```
SELECT precio_ticket  
FROM Funcion  
WHERE precio_ticket =  
    (SELECT MAX(precio_ticket)  
     FROM Funcion );
```

```
SELECT nombre_cliente  
FROM Cliente  
WHERE id_cliente IN  
    (SELECT id_cliente  
     FROM Cliente_Funcion  
     WHERE id_funcion = 32 );
```

```
SELECT p.titulo, COUNT(f.id_funcion) AS funciones  
FROM Pelicula p  
INNER JOIN Funcion f ON p.id_pelicula = f.id_pelicula  
GROUP BY p.id_pelicula, p.titulo;
```

```
SELECT f.id_funcion, SUM(cf.boletos) AS boletos_vendidos  
FROM Funcion f  
INNER JOIN Cliente_Funcion cf ON f.id_funcion = cf.id_funcion  
GROUP BY f.id_funcion;
```

```
SELECT f.id_funcion, SUM(cf.boletos * f.precio_ticket) AS ingreso  
FROM Funcion f  
INNER JOIN Cliente_Funcion cf ON f.id_funcion = cf.id_funcion  
GROUP BY f.id_funcion;
```

```
SELECT AVG(precio_ticket) AS promedio_precio  
FROM Funcion;
```

```
SELECT MAX(precio_ticket) AS max_precio, MIN(precio_ticket)  
AS min_precio  
FROM Funcion;
```

```
SELECT CONCAT(titulo, ' - ', clasificacion) AS titulo_full  
FROM Pelicula;
```

```
SELECT UPPER(titulo) AS titulo_mayuscula  
FROM Pelicula;
```

```
SELECT COUNT(*) AS total_peliculas  
FROM Pelicula;
```

Nombre:

Fecha:

Profesor:

Materia:

Institución:

Curso:

Nota:

```
SELECT UPPER(nombre_cliente) AS cliente-mayuscula
FROM Cliente;
```

```
SELECT p.Titulo, COUNT(pe.id_empleado) AS total_empleados
FROM Pelicula p
LEFT JOIN Pelicula_Emppleado pe ON p.id_pelicula =
pe.id_pelicula
GROUP BY p.id_pelicula, p.Titulo;
```

Procedimiento

```
CREATE PROCEDURE RegistrarSala
@id_Sala INT,
@nombre_Sala VARCHAR(50),
@capacidad INT
```

AS

BEGIN

```
    INSERT INTO (id_Sala, nombre_Sala, capacidad)
    VALUES (@id_Sala, @nombre_Sala, @capacidad);
END;
GO
```

```
EXEC RegistrarSala @id_Sala= 28, @nombre_Sala = 'Sala VIP',
@capacidad= 80;
```

```
UPDATE Pelicula SET titulo = 'interesstellur'
WHERE id_pelicula = 1;
```

```
UPDATE Cliente SET correo = 'ana@meveo.com'
WHERE id_cliente = 10;
```

```
UPDATE Sala SET capacidad = 90
WHERE id_Sala = 22;
```

```
UPDATE Funcion SET precio_ticket = 10.500
WHERE id_funcion = 32;
```

```
UPDATE Empleado SET cargo = 'Supervisor'
WHERE id_empleado = 42;
```

```
DELETE FROM Pelicula_Emppleado WHERE id_pelicula = 2
AND id_empleado = 41;
```

```
DELETE FROM Cliente_Funcion WHERE id_cliente = 11
AND id_funcion = 31;
```

```
DELETE FROM Funcion WHERE id_funcion = 30;
```

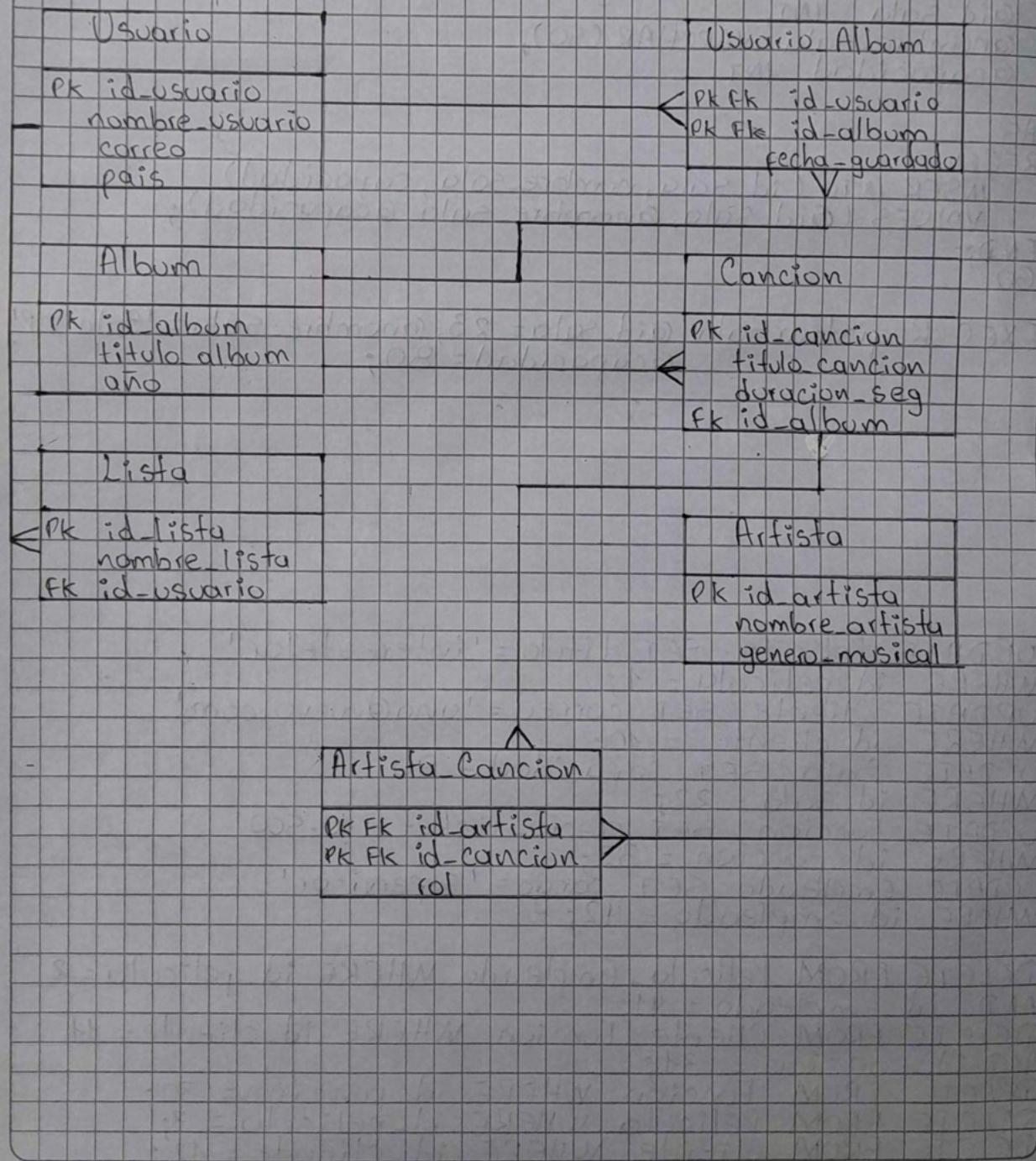
```
DELETE FROM Pelicula WHERE id_pelicula = 3;
```

```
DELETE FROM Cliente WHERE id_cliente = 12;
```

ALTER TABLE Pelicula ADD idioma VARCHAR(30);
ALTER TABLE Cliente DROP COLUMN correo;
ALTER TABLE Sala ADD tipo VARCHAR(20);
ALTER TABLE Empleado ADD telefono VARCHAR(10);
ALTER TABLE Funcion DROP COLUMN hora;

TRUNCATE TABLE Pelicula_Emppleado;
TRUNCATE TABLE Cliente_Funcion;
TRUNCATE TABLE Funcion;
TRUNCATE TABLE Sala;
TRUNCATE TABLE Empleado;

DROP TABLE Pelicula;
DROP TABLE Cliente;
DROP TABLE Funcion;
DROP TABLE Sala;
DROP TABLE Empleado;



```
CREATE DATABASE StreamingMusical;
USE StreamingMusical;
```

```
CREATE TABLE Usuario (
    id_usuario INT PRIMARY KEY,
    nombre_usuario VARCHAR(100),
    correo VARCHAR(100),
    pais VARCHAR(50)
);
```

```
CREATE TABLE Artista (
    id_artista INT PRIMARY KEY,
    nombre_artista VARCHAR(100),
    genero_musical VARCHAR(50)
);
```

```
CREATE TABLE Album (
    id_album INT PRIMARY KEY,
    titulo_album VARCHAR(150),
    año INT
);
```

```
CREATE TABLE Cancion (
    id_cancion INT PRIMARY KEY,
    titulo_cancion VARCHAR(150),
    duracion_seg INT,
    id_album INT,
    FK(id_album) REFERENCES Album(id_album)
);
```

```
CREATE TABLE Lista (
    id_lista INT PRIMARY KEY,
    nombre_lista VARCHAR(100),
    id_usuario,
    FK(id_usuario) REFERENCES Usuario(id_usuario)
);
```

```
CREATE TABLE Usuario_Album (
    id_usuario INT,
    id_libro INT,
    fecha_guardado DATE,
    PRIMARY KEY (id_usuario, id_libro),
    FOREIGN KEY(id_usuario) REFERENCES Usuario(id_usuario),
    FOREIGN KEY(id_libro) REFERENCES Album(id_album)
);
```

```
CREATE TABLE Artista_Cancion (
    id_artista INT,
    id_cancion INT,
    rot VARCHAR(50),
    PRIMARY KEY (id_artista, id_cancion),
    FOREIGN KEY(id_artista) REFERENCES Artista(id_artista),
    FOREIGN KEY(id_cancion) REFERENCES Cancion(id_cancion)
);
```

```
INSERT INTO Usuario(id_usuario, nombre_usuario, correo, pais) VALUES
(1, 'ana rodriguez', 'ana@correo.com', 'colombia'),
(2, 'luis martinez', 'luis@correo.com', 'mexico'),
(3, 'maria perez', 'maria@correo.com', 'argentina');
```

```

INSERT INTO Artista (id_artista, nombre_artista, genero_musical) VALUES
(10, 'coldplay', 'rock'),
(11, 'shakira', 'pop'),
(12, 'eminem', 'rap');

INSERT INTO Album (id_album, titulo_album, anno) VALUES
(100, 'viva la vida', 2008),
(101, 'laundry service', 2001),
(102, 'the marshall mathers lp', 2000);

INSERT INTO Cancion (id_cancion, titulo_cancion, duracion_seg, id_album) VALUES
(200, 'viva la vida', 242, 100),
(201, 'whenever wherever', 215, 101),
(202, 'stan', 404, 102);

INSERT INTO Lista (id_lista, nombre_lista, id_usuario) VALUES
(300, 'favoritos', 1),
(301, 'entretenimiento', 2),
(302, 'relax', 3);

INSERT INTO Usuario_Album (id_usuario, id_album, fecha_guardado) VALUES
(1, 100, '2025-08-01'),
(2, 101, '2025-08-02'),
(3, 102, '2025-08-03');

INSERT INTO Artista_Cancion (id_artista, id_cancion, rdi) VALUES
(10, 200, 'principal'),
(11, 201, 'principal'),
(12, 202, 'principal');

SELECT u.nombre_usuario, a.titulo_album
FROM Usuario u
INNER JOIN Usuario_Album ua ON u.id_usuario = ua.id_usuario
INNER JOIN Album a ON ua.id_album = a.id_album;

SELECT c.titulo_cancion, a.titulo_album
FROM Cancion c
INNER JOIN Album a ON c.id_album = a.id_album;

SELECT ar.nombre_artista, c.titulo_cancion
FROM Artista ar
INNER JOIN Artista_Cancion ac ON ar.id_artista = ac.id_artista
INNER JOIN Cancion c ON ac.id_cancion = c.id_cancion;

SELECT al.titulo_album, ar.nombre_artista
FROM Album al
INNER JOIN Cancion c ON al.id_album = c.id_album
INNER JOIN Artista_Cancion ac ON c.id_cancion = ac.id_cancion
INNER JOIN Artista ar ON ac.id_artista = ar.id_artista;

```

Nombre:

Fecha:

Profesor:

Materia:

Institución:

Curso:

Nota:

```
SELECT l.nombre-lista, u.nombre-usuario  
FROM Lista l  
INNER JOIN Usuario u ON l.id-usuario = u.id-usuario;
```

```
SELECT titulo-album  
FROM Album  
WHERE id-album IN  
    SELECT id-album  
    FROM Usuario-Album );
```

```
SELECT nombre-usuario  
FROM Usuario  
WHERE id-usuario IN  
    SELECT id-usuario  
    FROM Usuario-Album  
    WHERE id-album IN  
SELECT id-album  
FROM Album  
WHERE año > 2000 )  
;
```

```
SELECT nombre-artista  
FROM Artista  
WHERE id-artista IN  
    SELECT id-artista  
    FROM Artista-Cancion  
    WHERE id-cancion = 202  
;
```

```
SELECT titulo-cancion  
FROM Cancion  
WHERE id-album IN  
    SELECT id-album  
    FROM Album  
    WHERE titulo-album = 'viva la vida'  
;
```

```
SELECT titulo-cancion  
FROM Cancion  
WHERE id-cancion IN  
    SELECT id-cancion  
    FROM Artista-Cancion  
    WHERE rol = 'principal'  
;
```

```
SELECT u.nombre-usuario  
    COUNT(va.id-album) AS total-albums  
FROM Usuario u  
LEFT JOIN Usuario-Album va ON u.id-usuario = va.id-usuario  
GROUP BY u.id-usuario, u.nombre-usuario;
```

```
SELECT a.id_album,
       SUM(c.duracion_seg) duracion_total_seg
  FROM Album a
 INNER JOIN Cancion c ON a.id_album = c.id_album
 GROUP BY a.id_album, a.titulo_album;
```

```
SELECT a.id_album
       MAX(c.duracion_seg) AS duracion_max_seg
  FROM Album a
 INNER JOIN Cancion c ON a.id_album = c.id_album
 GROUP BY a.id_album, a.titulo_album;
```

```
SELECT AVG(duracion_seg) AS duracion_promedio_seg
  FROM Cancion;
```

```
SELECT MAX(año) AS año_mas_reciente
  FROM Album;
```

```
SELECT CONCAT(u.nombre_usuario, ' - ', u.pais) AS usuario_pais
  FROM Usuario;
```

```
SELECT MIN(duracion_seg) AS duracion_min_seg
  FROM Cancion;
```

```
SELECT UPPER(titulo_cancion) AS titulo_mayuscula
  FROM Cancion;
```

```
SELECT MIN(fecha_guardado) AS fecha_mas_antigua
  FROM Usuario_Album;
```

```
SELECT l.nombre_lista,
       COUNT(l.id_lista) AS total_lista
  FROM Lista l
 LEFT JOIN Usuario u ON u.id_usuario = l.id_usuario
 GROUP BY u.id_usuario, l.nombre_lista
```

PROCEDIMIENTO

```
CREATE PROCEDURE RegistrarAlbum
```

```
@id_album INT,  
@titulo_album VARCHAR(150),  
@año INT
```

```
AS
```

```
BEGIN
```

```
    INSERT INTO Album(id_album, titulo_album, año)  
    VALUES (@id_album, @titulo_album, @año);
```

```
END;
```

```
GO
```

```
EXEC RegistrarAlbum @id_album=103, @titulo_album='Thriller',  
@año=2009;
```

```
UPDATE Usuario SET pais = 'chile'  
WHERE id_usuario = 3;  
UPDATE Artista SET genero_musical = 'indie'  
WHERE id_artista = 10;  
UPDATE Album SET año = 2012  
WHERE id_album = 101;  
UPDATE Cancion SET duracion_seg = 250  
WHERE id_cancion = 201;  
UPDATE Lista SET nombre_lista = 'hits 2025'  
WHERE id_lista = 300;
```

```
DELETE FROM Usuario_Album WHERE id_usuario = 2 AND  
id_album = 101;  
DELETE FROM Artista_Cancion WHERE id_artista = 11 AND  
id_cancion = 201;  
DELETE FROM Lista WHERE id_lista = 302;  
DELETE FROM Cancion WHERE id_cancion = 202;  
DELETE FROM Artista WHERE id_artista = 12;
```

```
ALTER TABLE Usuario ADD fecha_registro DATE;  
ALTER TABLE Album ADD sello VARCHAR(80);  
ALTER TABLE Cancion DROP COLUMN duracion_seg;  
ALTER TABLE Lista ADD descripcion VARCHAR(50);  
ALTER TABLE Artista DROP COLUMN genero_musical;
```

```
TRUNCATE TABLE Usuario_Album;  
TRUNCATE TABLE Artista_Cancion;  
TRUNCATE TABLE Lista;  
TRUNCATE TABLE Cancion;  
TRUNCATE TABLE Album;
```

```
DROP TABLE Cancion;  
DROP TABLE Artista;  
DROP TABLE Lista;  
DROP TABLE Album;  
DROP TABLE Usuario;
```

Huesped

- PK id_huesped
nombre
correo

Empleado

PK id_empleado
nombre_empleado
puesto

Habitacion

PK id_habitacion
numero
tipo
capacidad
precio_noche

Huesped_Servicio

<PK FK id_huesped
<PK FK id_servicio
fecha
cantidad

Reserva

PK id_reserva
FK id_huesped
FK id_empleado
fecha_entrada
fecha_salida

Reserva_Habitacion

PK FK id_reserva
PK FK id_habitacion
noches

Servicio

PK id_servicio
nombre_servicio
costo

```
CREATE DATABASE Hotel;  
USE Hotel;
```

```
CREATE TABLE Huesped (  
id_huesped INT PRIMARY KEY,  
nombre VARCHAR(100),  
correo VARCHAR(100)  
);
```

```
CREATE TABLE Habitacion (  
id_habitacion INT PRIMARY KEY,  
numero INT,  
tipo VARCHAR(50),  
capacidad INT,  
precio_noche DECIMAL(8,2)  
);
```

```
CREATE TABLE Empleado (  
id_empleado INT PRIMARY KEY,  
nombre_empleado VARCHAR(100),  
puesto VARCHAR(50)  
);
```

Nombre:

Fecha:

Profesor:

Materia:

Institución:

Curso:

Nota:

```

CREATE TABLE Servicio (
    id_servicio INT PRIMARY KEY,
    nombre_servicio VARCHAR(100),
    costo DECIMAL (8,2)
);

CREATE TABLE Reserva (
    id_reserva INT PRIMARY KEY,
    id_huesped INT,
    id_empleado INT,
    fecha_ingreso DATE,
    fecha_salida DATE,
    FK(id_huesped) REFERENCES Huesped (id_huesped),
    FK(id_empleado) REFERENCES Empleado (id_empleado)
);

CREATE TABLE Reserva_Habitacion (
    id_reserva INT,
    id_habitacion INT,
    noches INT,
    PRIMARY KEY (id_reserva, id_habitacion),
    FK(id_reserva) REFERENCES Reserva (id_reserva),
    FK(id_habitacion) REFERENCES Habitacion (id_habitacion)
);

CREATE TABLE Huesped_Servicio (
    id_huesped INT,
    id_servicio INT,
    fecha DATE,
    cantidad INT,
    PRIMARY KEY (id_huesped, id_servicio),
    FK(id_huesped) REFERENCES Huesped (id_huesped),
    FK(id_servicio) REFERENCES Servicio (id_servicio)
);

INSERT INTO Huesped (id_huesped, nombre, correo) VALUES
(1, 'ana rodríguez', 'ana@correo.com'),
(2, 'luis martínez', 'luis@correo.com'),
(3, 'maría pérez', 'maria@correo.com');

INSERT INTO Habitacion (id_habitacion, numero, tipo, capacidad, precio_noche) VALUES
(101, 101, 'individual', 1, 150.00),
(102, 102, 'doble', 2, 220.00),
(103, 103, 'suite', 4, 450.00);

INSERT INTO Empleado (id_empleado, nombre_empleado, puesto) VALUES
(201, 'pedro gonzález', 'recepcionista'),
(202, 'carla suárez', 'supervisor'),
(203, 'javier lopez', 'recepcionista');

```

INSERT INTO Servicio (id_servicio, nombre_servicio, costo) VALUES
(301, 'room service', 25.000),
(302, 'lavanderia', 15.000),
(303, 'spa', 60.000);

INSERT INTO Reserva (id_reserva, id_huesped, id_empleado, fecha_ingreso, fecha_salida) VALUES

(401, 1, 201, '2025-07-10', '2025-07-12'),
(402, 2, 202, '2025-07-11', '2025-07-15'),
(403, 3, 203, '2025-07-12', '2025-07-13');

INSERT INTO Reserva_Habitacion (id_reserva, id_habitacion, noches) VALUES

(401, 101, 2),
(402, 102, 4),
(403, 103, 1);

INSERT INTO Huesped_Servicio (id_huesped, id_servicio, fecha, cantidad) VALUES

(1, 301, '2025-07-10', 1),
(2, 302, '2025-07-12', 2),
(3, 303, '2025-07-13', 1);

SELECT r.id_reserva, h.nombre, e.nombre_empleado
FROM Reserva r
INNER JOIN Huesped h ON r.id_huesped = h.id_huesped
INNER JOIN Empleado e ON r.id_empleado = e.id_empleado;

SELECT h.nombre, hab.numero, hab.tipo
FROM Reserva_Habitacion rh
INNER JOIN Reserva r ON rh.id_reserva = r.id_reserva
INNER JOIN Huesped h ON r.id_huesped = h.id_huesped
INNER JOIN Habitacion hab ON rh.id_habitacion = hab.id_habitacion;

SELECT hu.nombre, s.nombre_servicio, hs.fecha, hs.cantidad
FROM Huesped_Servicio hs
INNER JOIN Huesped hu ON hs.id_huesped = hu.id_huesped
INNER JOIN Servicio s ON hs.id_servicio = s.id_servicio;

SELECT r.id_reserva, hab.numero, hab.precio_noche, rh.noches,
hab.precio_noche * rh.noches AS costo
FROM Reserva r
INNER JOIN Reserva_Habitacion rh ON r.id_reserva = rh.id_reserva
INNER JOIN Habitacion hab ON rh.id_habitacion = hab.id_habitacion;

SELECT h.nombre, AS nombre_huesped, e.nombre_empleado,
r.fecha_ingreso
FROM Reserva r
INNER JOIN Huesped h ON r.id_huesped = h.id_huesped
INNER JOIN Empleado e ON r.id_empleado = e.id_empleado;

```
SELECT id_reserva
FROM Reserva_Habitacion
WHERE id_habitacion IN(
    SELECT id_habitacion
    FROM Habitacion
    WHERE capacidad > 2 );
```

```
SELECT nombre_servicio
FROM Servicio
WHERE costo > (
    SELECT AVG(costo)
    FROM Servicio );
```

```
SELECT id_reserva
FROM Reserva
WHERE id_empleado IN(
    SELECT id_empleado
    FROM Empleado
    WHERE puesto = 'recepcionista' );
```

```
SELECT nombre_empleado
FROM Empleado
WHERE id_empleado IN(
    SELECT id_empleado
    FROM Reserva
    WHERE id_reserva IN(
        SELECT id_reserva
        FROM Reserva_Habitacion
        WHERE noches > 2 ) );
);
```

```
SELECT nombre_servicio
FROM Servicio
WHERE id_servicio IN(
    SELECT id_servicio
    FROM Huesped_Servicio
    WHERE id_huesped IN(
        SELECT id_huesped
        FROM Huesped
        WHERE correo = 'ana@correo.com' ) );
);
```

```
SELECT AVG(rh.noches) AS noches_promedio
FROM Reserva_Habitacion rh;
```

```
SELECT MAX(precio_noche) AS precio_maximo
FROM Habitacion;
```

```
SELECT MIN(precio_noche) AS precio_minimo
FROM Habitacion;
```

```
SELECT CONCAT(nombre_servicio, '-$', costo) AS servicio_precio
FROM Servicio;
```

```
SELECT AVG(costo) AS costo_promedio
FROM Servicio;
```

```
SELECT UPPER(nombre) AS nombre_huesped_mayuscula  
FROM Huesped;
```

```
SELECT tipo, COUNT(*) AS habitaciones_tipo  
FROM Habitacion  
GROUP BY tipo;
```

```
SELECT h.numero, SUM(rh.noches) AS noches_reservadas  
FROM Habitacion h  
LEFT JOIN Reserva_Habitacion rh ON h.id_habitacion =  
rh.id_habitacion  
GROUP BY h.id_habitacion, h.numero;
```

```
SELECT e.nombre_empleado, COUNT(r.id_reserva) AS total_reservas  
FROM Empleado e  
LEFT JOIN Reserva r ON e.id_empleado = r.id_empleado  
GROUP BY e.id_empleado, e.nombre_empleado;
```

```
SELECT CONCAT(UPPER(nombre_empleado), '-' ,  
UPPER(puesto)) AS empleado_puesto  
FROM Empleado;
```

PROCEDIMIENTO

```
CREATE PROCEDURE RegistrarEmpleado  
@id_empleado INT,  
@nombre_empleado VARCHAR(100),  
@puesto VARCHAR(50)
```

AS

BEGIN

```
    INSERT INTO Empleado (id_empleado, nombre_empleado, puesto)  
    VALUES (@id_empleado, @nombre_empleado, @puesto);
```

END;

GO

```
EXEC RegistrarEmpleado @id_empleado=204,  
@nombre_empleado='Lucía García',  
@puesto='gerente';
```

```
UPDATE Huesped SET correo='ana@nuevo.com'  
WHERE id_huesped = 1;
```

```
UPDATE Habitacion SET precio_noche= 180.000  
WHERE id_habitacion = 101;
```

```
UPDATE Reserva SET fecha_salida='2025-07-13'  
WHERE id_reserva = 401;
```

```
UPDATE Servicio SET costo = 30.000  
WHERE id_servicio = 301;
```

```
UPDATE Empleado SET puesto = 'gerente'  
WHERE id_empleado= 202;
```

Nombre:

Fecha:

Profesor:

Materia:

Institución:

Curso:

Nota:

DELETE FROM Huesped-Servicio WHERE id-huesped = 2
AND id-servicio = 302;

DELETE FROM Reserva-Habitacion WHERE id-reserva = 402
AND id-habitacion = 102;

DELETE FROM Reserva WHERE id-reserva = 403;

DELETE FROM Habitacion WHERE id-habitacion = 103;

DELETE FROM Servicio WHERE id-servicio = 303;

ALTER TABLE Huesped ADD telefono VARCHAR(20);

ALTER TABLE Habitacion DROP COLUMN capacidad;

ALTER TABLE Reserva-Habitacion ADD importe DECIMAL(8,2);

ALTER TABLE Empleado DROP COLUMN puesto;

ALTER TABLE Servicio ADD descripcion VARCHAR(100);

TRUNCATE TABLE Huesped-Servicio;

TRUNCATE TABLE Reserva-Habitacion;

TRUNCATE TABLE Reserva;

TRUNCATE TABLE Habitacion;

TRUNCATE TABLE Servicio;

DROP TABLE Habitacion;

DROP TABLE Huesped;

DROP TABLE Reserva-Habitacion;

DROP TABLE Reserva;

DROP TABLE Empleado;

Tabla: Cliente

1-MER

Clave	Campo	Tipo de Dato	Tamaño	Descripción
PK	id-cliente	INT		Identificador único del cliente
	nombre	VARCHAR	100	Nombre completo del cliente
	edad	INT		Edad del cliente en años

Tabla: Actividad

Clave	Campo	Tipo de Dato	Tamaño	Descripción
PK	id-actividad	INT		Identificador único de la actividad
	nombre_actividad	VARCHAR	100	Nombre de la actividad
	nivel	VARCHAR	20	Dificultad o nivel

Tabla: Entrenador

Clave	Campo	Tipo de Dato	Tamaño	Descripción
PK	id_entrenador	INT		Identificador único del entrenador
	nombre_entrenador	VARCHAR	100	Nombre del entrenador
	especialidad	VARCHAR	50	Área de especialidad

Tabla: Sala

Clave	Campo	Tipo de Dato	Tamaño	Descripción
PK	id-sala	INT		Identificador único de la sala
	nombre_sala	VARCHAR	100	Nombre de la sala
	capacidad	INT		Máximo permitido en la sala

Tabla: Horario

Clave	Campo	Tipo de Dato	Tamaño	Descripción
PK	id-horario	INT		Identificador único del horario
	dia	DATE		Fecha del día programado
	hora-inicio	VARCHAR	10	Hora de inicio
	hora-fin	VARCHAR	10	Hora fin

Tabla: Actividad-Cliente

Clave	Campo	Tipo de Dato	Tamaño	Descripción
PK FK	id-cliente	INT		Referencia a cliente
PK FK	id-actividad	INT		Referencia a actividad

Tabla: Entrenador-Sala

Clave	Campo	Tipo de Dato	Tamaño	Descripción
PK FK	id-entrenador	INT		Referencia a entrenador
PK FK	id-Sala	INT		Referencia a sala

2- MER Tabla: Usuario

Clave	Campo	Tipo de Dato	Tamaño	Descripción
PK	id-usuario	INT		Identificador único de usuario
	nombre	VARCHAR	100	Nombre del usuario
	correo	VARCHAR	100	Correo electrónico del usuario

Tabla: Libro

Clave	Campo	Tipo de Dato	Tamaño	Descripción
PK	id-libro	INT		Identificador único del libro
	título	VARCHAR	150	Título del libro
	año-publicación	INT		Año en que se publicó el libro

Tabla: Prestamo

Clave	Campo	Tipo de Dato	Tamaño	Descripción
PK	id-prestamo	INT		Identificador único del préstamo
Fk	id-usuario	INT		Referencia al usuario
Fk	id-libro	INT		Referencia al libro
	fecha-prestamo	DATE		Fecha préstamo del libro
	fecha-devolución	DATE		Fecha devolución del libro

Tabla: Categoria

Clave	Campo	Tipo de Dato	Tamaño	Descripción
Pk	id_categoria	INT		Identificador único de categoría
	nombre_categoria	VARCHAR	100	Nombre de la categoría
	descripcion	VARCHAR	100	Descripción corta de categoría

Tabla: Autor

Clave	Campo	Tipo de Dato	Tamaño	Descripción
Pk	id_autor	INT		Identificador único del autor
	nombre_autor	VARCHAR	100	Nombre completo del autor
	nacionalidad	VARCHAR	50	País de origen del autor

Tabla: Categoria_Libro

Clave	Campo	Tipo de Dato	Tamaño	Descripción
Pk Fk	id_libro	INT		Referencia al libro
Pk Fk	id_categoria	INT		Referencia a la categoría

Tabla: Libro-Autor

Clave	Campo	Tipo de Dato	Tamaño	Descripción
Pk Fk	id_libro	INT		Referencia al libro
Pk Fk	id_autor	INT		Referencia al autor

3 - MER Tabla: Pelicula

Clave	Campo	Tipo de Dato	Tamaño	Descripción
Pk	id_pelicula	INT		Identificador de película
	titulo	VARCHAR	100	Título de la película
	duracion_min	INT		Duración en minutos
	clasificacion	VARCHAR	10	Clasificación de edad

Nombre:

Fecha:

Profesor:

Materia:

Institución:

Curso:

Nota:

Tabla: Cliente

Clave	Campo	Tipo de Dato	Tamaño	Descripción
PK	id_cliente	INT		Identificador único del cliente
	nombre_cliente	VARCHAR	100	Nombre del cliente
	correo	VARCHAR	100	Correo del cliente

Tabla: Sala

Clave	Campo	Tipo de Dato	Tamaño	Descripción
PK	id_sala	INT		Identificador de la sala
	nombre_sala	VARCHAR	50	Nombre de la sala
	capacidad	INT		Aforo máximo de la sala

Tabla: Funcion

Clave	Campo	Tipo de Dato	Tamaño	Descripción
PK	id_funcion	INT		Identificador único de función
FK	id_pelicula	INT		Referencia a la película
FK	id_sala	INT		Referencia a la sala
	fecha	DATE		Fecha de la función
	hora	TIME		Hora de la función
	precioticket	DECIMAL	6,2	Precio de la función

Tabla: Empleado

Clave	Campo	Tipo de Dato	Tamaño	Descripción
PK	id_empleado	INT		Identificador único del empleado
	nombre_empleado	VARCHAR	100	Nombre del empleado
	cargo	VARCHAR	50	Cargo o puesto

Tabla: Pelicula_Empleado

Clave	Campo	Tipo de Dato	Tamaño	Descripción
PK FK	id_pelicula	INT		Referencia a pelicula
PK FK	id_empleado	INT		Referencia a empleado
	rol	VARCHAR	50	Rol del empleado

Tabla: Cliente_Funcion

Clave	Campo	Tipo de Dato	Tamaño	Descripción
PK FK	id_cliente	INT		Referencia al cliente
PK FK	id_funcion	INT		Referencia a funcion
	boletos	INT		Cantidad de boletos

4-MER Tabla: Usuario

Clave	Campo	Tipo de Dato	Tamaño	Descripción
PK	id_usuario	INT		Identificador unico de usuario
	nombre_usuario	VARCHAR	100	Nombre del usuario
	correo	VARCHAR	100	Correo del usuario
	pais	VARCHAR	50	Pais de origen del usuario

Tabla: Artista

Clave	Campo	Tipo de Dato	Tamaño	Descripción
PK	id_artista	INT		Identificador unico del artista
	nombre_artista	VARCHAR	100	Nombre del artista
	genero_musical	VARCHAR	50	Genero de la musica

Tabla: Album

Clave	Campo	Tipo de Dato	Tamaño	Descripción
PK	id_album	INT		Identificador unico del album
	titulo_album	VARCHAR	150	Titulo del album
	año	INT		Año de publicacion

Tabla: Cancion

Clave	Campo	Tipo de Dato	Tamaño	Descripción
PK	id_cancion	INT		Identificador único de la canción
FK	id_album	INT		Referencia al álbum
	titulo_cancion	VARCHAR	150	Título de la canción
	duracion_seg	INT		Duración de la canción en seg.

Tabla: Lista

Clave	Campo	Tipo de Dato	Tamaño	Descripción
PK	id_lista	INT		Identificador único de lista
FK	id_usuario	INT		Referencia al usuario
	nombre_lista	VARCHAR	100	Nombre de la lista

Tabla: Usuario-Album

Clave	Campo	Tipo de Dato	Tamaño	Descripción
PK FK	id_usuario	INT		Referencia al usuario
PK FK	id_album	INT		Referencia al álbum
	fecha_guardado	DATE		Fecha de guardado del álbum

Tabla: Artista-Cancion

Clave	Campo	Tipo de Dato	Tamaño	Descripción
PK FK	id_artista	INT		Referencia del artista
PK FK	id_cancion	INT		Referencia de Canción
	rol	VARCHAR	50	Rol

S-MER Tabla: Huesped

Clave	Campo	Tipo de Dato	Tamaño	Descripción
PK	id_huesped	INT		Identificador único de huesped
	nombre	VARCHAR	100	Nombre del Huesped
	correo	VARCHAR	100	Correo del huesped

Tabla: Habitacion

Clave	Campo	Tipo de Tamaño	Dato	Descripción
PK	id_habitacion	INT		Identificador unico del huésped
	numero	INT		Número de la habitación
	tipo	VARCHAR	50	Tipo de habitación
	capacidad	INT		Capacidad de la habitación
	precio_noche	DECIMAL	8,2	Precio de la habitación por noche

Tabla: Empleado

Clave	Campo	Tipo de Tamaño	Dato	Descripción
PK	id_empleado	INT		Identificador unico del empleado
	nombre_empleado	VARCHAR	100	Nombre del empleado
	puesto	VARCHAR	50	El puesto que tiene el empleado

Tabla: Servicio

Clave	Campo	Tipo de Tamaño	Dato	Descripción
PK	id_servicio	INT		Identificador unico del servicio
	nombre_servicio	VARCHAR	100	Nombre del servicio
	costo	DECIMAL	8,2	Costo del servicio

Tabla: Reserva

Clave	Campo	Tipo de Tamaño	Dato	Descripción
PK	id_reserva	INT		Identificador unico de la reserva
FK	id_huesped	INT		Referencia del huésped
FK	id_empleado	INT		Referencia del empleado
	fecha_ingreso	DATE		Fecha ingreso a la reserva
	fecha_salida	DATE		Fecha de salida de la reserva

Nombre:

Fecha:

Profesor:

Materia:

Institución:

Curso:

Nota:

Tabla: Reserva_Habitacion

Clave Campo	Tipo de Tamaño	Dato	Descripción
PK FK id_reserva	INT		Referencia de la reserva
id_habitacion	INT		Referencia de la habitación
Noches	INT		Cantidad de noches hospedada

Tabla: Huesped_Servicio

Clave Campo	Tipo de Tamaño	Dato	Descripción
PK FK id-huesped	INT		Referencia de huésped
PK FK id-servicio	INT		Referencia de servicio
fecha	DATE		Fecha del día de hospedaje
Cantidad	INT		Cantidad de servicio