MIGUEL ANGEL MENDEZ

Ejercicios de Base de Datos

Ejercicio 1 — Gestión de Biblioteca

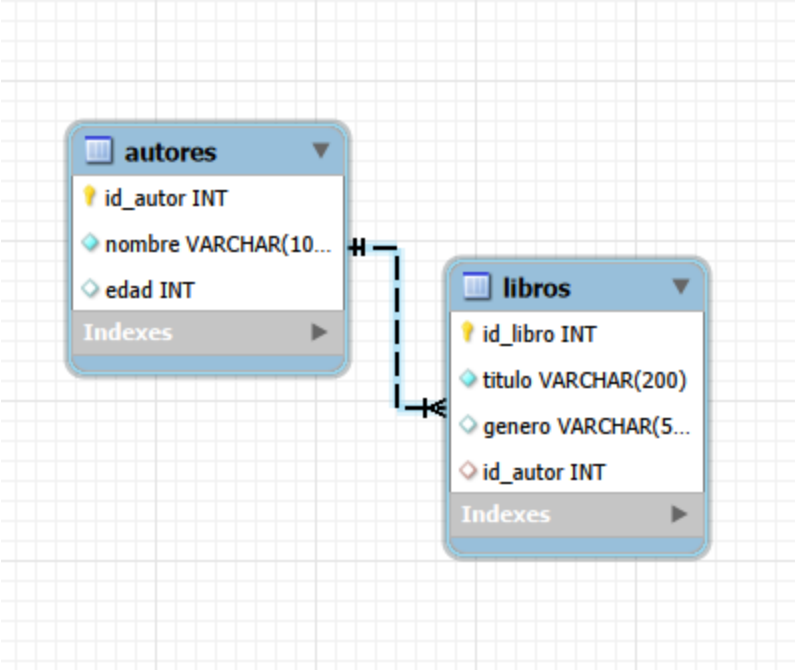
Crea una base de datos llamada 'biblioteca' con las tablas 'autores' y 'libros', incluyendo claves primarias y foráneas, inserciones y consultas.

```
CREATE DATABASE biblioteca;
USE biblioteca;
CREATE TABLE autores (
   id_autor INT AUTO_INCREMENT PRIMARY KEY,
   nombre VARCHAR(100) NOT NULL,
   edad INT
);

CREATE TABLE libros (
   id_libro INT AUTO_INCREMENT PRIMARY KEY,
   titulo VARCHAR(200) NOT NULL,
   genero VARCHAR(50),
   id_autor INT,

   FOREIGN KEY (id_autor) REFERENCES autores(id_autor)
   ON DELETE CASCADE
);

INSERT INTO autores (nombre, edad) VALUES ('Xay', 8); --
INSERT INTO autores (nombre, edad) VALUES ('Zab', 4); --

INSERT INTO libros (titulo, genero, id_autor) VALUES
('Las aventuras de Lila la Furia Nocturna', 'Fantasía', 1),
('El vuelo del Nader Rojo', 'Aventura', 2),
```

('Guía del Casco Multicolor', 'Manual', 1);

SELECT libros.titulo, autores.nombre AS escritor
FROM libros
INNER JOIN autores ON libros.id_autor = autores.id_autor;

CAPTURA:



Ejercicio 2 — Sistema de Ventas

Crea una base de datos llamada 'ventas' con las tablas 'clientes', 'productos' y 'facturas'. Define las relaciones, inserta datos y realiza consultas.

CREATE DATABASE ventas;
USE ventas;

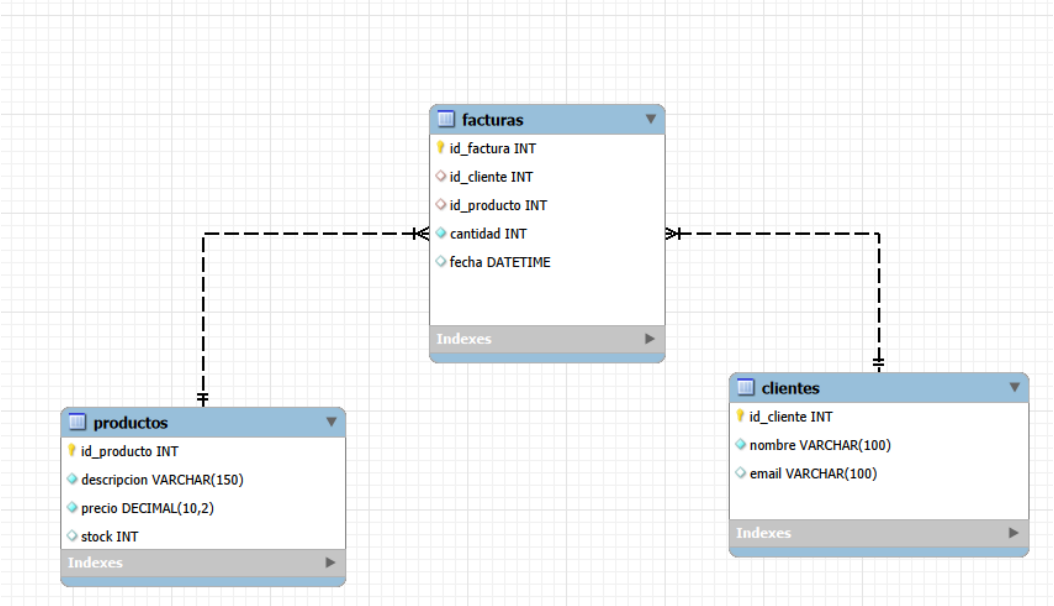
CREATE TABLE clientes (
  id_cliente INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(100) NOT NULL,
  email VARCHAR(100) UNIQUE
);

CREATE TABLE productos (
  id_producto INT AUTO_INCREMENT PRIMARY KEY,

```
    descripcion VARCHAR(150) NOT NULL,
    precio DECIMAL(10, 2) NOT NULL,
    stock INT DEFAULT 0
);

CREATE TABLE facturas (
    id_factura INT AUTO_INCREMENT PRIMARY KEY,
    id_cliente INT,
    id_producto INT,
    cantidad INT NOT NULL,
    fecha DATETIME DEFAULT CURRENT_TIMESTAMP,
    -- Definición de relaciones (Claves Foráneas)
    FOREIGN KEY (id_cliente) REFERENCES clientes(id_cliente),
    FOREIGN KEY (id_producto) REFERENCES productos(id_producto)
);

-- INSERTAR  DATOS --
INSERT INTO clientes (nombre, email) VALUES
('Juan Pérez', 'juan.perez@email.com'),
('María García', 'm.garcia@email.com');

-- Insertar Productos
INSERT INTO productos (descripcion, precio, stock) VALUES
('Laptop Pro 15', 1200.00, 10),
('Mouse Inalámbrico', 25.50, 50),
('Monitor 24 Pulgadas', 180.00, 15);

INSERT INTO facturas (id_cliente, id_producto, cantidad) VALUES
(1, 1, 1),
(2, 3, 2);

-- CONSULTA --
SELECT
    f.id_factura,
    c.nombre AS cliente,
    p.descripcion AS producto,
    f.cantidad,
    (f.cantidad * p.precio) AS total_pagar
FROM facturas f
INNER JOIN clientes c ON f.id_cliente = c.id_cliente
```

INNER JOIN productos p ON f.id_producto = p.id_producto;

CAPTURA:



Ejercicio 3 — Control de Estudiantes

Crea una base de datos llamada 'colegio' con las tablas 'estudiantes', 'cursos' y 'matriculas'. Incluye claves primarias, foráneas, inserciones y consultas.

```
CREATE DATABASE colegio;
USE colegio;

CREATE TABLE estudiantes (
  id_estudiante INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(100) NOT NULL,
  edad INT
);

CREATE TABLE cursos (
  id_curso INT AUTO_INCREMENT PRIMARY KEY,
  nombre_curso VARCHAR(100) NOT NULL,
  descripcion TEXT
);

CREATE TABLE matriculas (
  id_matricula INT AUTO_INCREMENT PRIMARY KEY,
  id_estudiante INT,
```

```
  id_curso INT,
  fecha_inscripcion DATE DEFAULT (CURRENT_DATE),

  FOREIGN KEY (id_estudiante) REFERENCES estudiantes(id_estudiante) ON
DELETE CASCADE,
  FOREIGN KEY (id_curso) REFERENCES cursos(id_curso) ON DELETE CASCADE
);

-- INSERCION --


INSERT INTO estudiantes (nombre, edad) VALUES
('Xay', 8), --
('Zab', 4); --


INSERT INTO cursos (nombre_curso, descripcion) VALUES
('Vuelo de Dragones Nivel 1', 'Introducción al vuelo con Furia Nocturna y Nader'),
('Mantenimiento de Cascos Multicolor', 'Cuidado y pintura de equipo de
protección');


INSERT INTO matriculas (id_estudiante, id_curso) VALUES (1, 1), (1, 2);
INSERT INTO matriculas (id_estudiante, id_curso) VALUES (2, 1);


 -- CONSULTA --

SELECT
  e.nombre AS Estudiante,
  c.nombre_curso AS Curso,
  m.fecha_inscripcion AS "Fecha de Inicio"
FROM matriculas m
INNER JOIN estudiantes e ON m.id_estudiante = e.id_estudiante
INNER JOIN cursos c ON m.id_curso = c.id_curso;

SELECT
  c.nombre_curso,
  COUNT(m.id_estudiante) AS total_alumnos
FROM cursos c
```

LEFT JOIN matriculas m ON c.id_curso = m.id_curso
GROUP BY c.nombre_curso;

CAPTURA: