



Trabajo Final

Diseñar y desarrollar una base de datos en MySQL que refleje la relación 1:M entre tablas, aplicando claves primarias, foráneas, consultas básicas, joins, estadísticas y manipulación de datos (insertar, actualizar y eliminar).

Estructura del proyecto

Tendremos un sistema de gestión académica simple con estas entidades:

- **Departamento**
- **Estudiante**
- **Profesor**
- **Curso**
- **Clase**
- **Inscripcion**
- **Calificacion**

Crear una carpeta llamada trabajo final BD en su repositorio de base de datos.

```
-- =====
-- 1. CREACIÓN DE BASE DE DATOS Y TABLAS
-- =====

CREATE DATABASE IF NOT EXISTS GestionAcademica;
USE GestionAcademica;

-- Tabla: Departamento
CREATE TABLE Departamento (
    id_departamento INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    ubicacion VARCHAR(100)
);

-- Tabla: Profesor (Relación N:1 con Departamento)
CREATE TABLE Profesor (
    id_profesor INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(50) NOT NULL,
    apellido VARCHAR(50) NOT NULL,
    email VARCHAR(100) UNIQUE,
    id_departamento INT,
    FOREIGN KEY (id_departamento) REFERENCES Departamento(id_departamento)
);

-- Tabla: Curso (Relación N:1 con Departamento)
CREATE TABLE Curso (
    id_curso INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    creditos INT NOT NULL,
    id_departamento INT,
    FOREIGN KEY (id_departamento) REFERENCES Departamento(id_departamento)
);

-- Tabla: Estudiante
CREATE TABLE Estudiante (
    id_estudiante INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(50) NOT NULL,
    apellido VARCHAR(50) NOT NULL,
    fecha_nacimiento DATE,
    email VARCHAR(100) UNIQUE
);
```

-- Tabla: Clase (Instancia de un Curso, Relación N:1 con Curso y Profesor)

```
CREATE TABLE Clase (
    id_clase INT AUTO_INCREMENT PRIMARY KEY,
    id_curso INT,
    id_profesor INT,
    semestre VARCHAR(20),
    anio INT,
    FOREIGN KEY (id_curso) REFERENCES Curso(id_curso),
    FOREIGN KEY (id_profesor) REFERENCES Profesor(id_profesor)
);
```

-- Tabla: Inscripcion (Tabla intermedia, Relación N:M entre Estudiante y Clase)

```
CREATE TABLE Inscripcion (
    id_inscripcion INT AUTO_INCREMENT PRIMARY KEY,
    id_estudiante INT,
    id_clase INT,
    fecha_inscripcion DATE DEFAULT (CURRENT_DATE),
    FOREIGN KEY (id_estudiante) REFERENCES Estudiante(id_estudiante),
    FOREIGN KEY (id_clase) REFERENCES Clase(id_clase)
);
```

-- Tabla: Calificacion (Relación 1:1 o N:1 con Inscripción)

```
CREATE TABLE Calificacion (
    id_calificacion INT AUTO_INCREMENT PRIMARY KEY,
    id_inscripcion INT,
    nota DECIMAL(4, 2),
    fecha_evaluacion DATE,
    FOREIGN KEY (id_inscripcion) REFERENCES Inscripcion(id_inscripcion)
);
```

-- =====

-- 2. INserción DE DATOS

-- =====

-- Insertar Departamentos

```
INSERT INTO Departamento (nombre, ubicacion) VALUES
('Ingeniería de Sistemas', 'Edificio A'),
('Matemáticas', 'Edificio B'),
('Humanidades', 'Edificio C');
```

-- Insertar Profesores

```
INSERT INTO Profesor (nombre, apellido, email, id_departamento) VALUES
('Carlos', 'Pérez', 'carlos.perez@uni.edu', 1),
('Ana', 'Gómez', 'ana.gomez@uni.edu', 1),
('Luis', 'Martínez', 'luis.martinez@uni.edu', 2);
```

-- Insertar Cursos

```
INSERT INTO Curso (nombre, creditos, id_departamento) VALUES
('Base de Datos I', 4, 1),
('Programación Avanzada', 5, 1),
('Cálculo Integral', 4, 2),
('Ética Profesional', 2, 3);
```

-- Insertar Estudiantes

```
INSERT INTO Estudiante (nombre, apellido, fecha_nacimiento, email) VALUES
('Juan', 'Díaz', '2000-05-15', 'juan.diaz@student.edu'),
('Maria', 'Rodríguez', '2001-08-20', 'maria.rod@student.edu'),
('Pedro', 'Sánchez', '1999-12-10', 'pedro.san@student.edu');
```

-- Insertar Clases

```
INSERT INTO Clase (id_curso, id_profesor, semestre, anio) VALUES
(1, 1, '2023-1', 2023), -- Base de Datos con Carlos Pérez
(2, 2, '2023-1', 2023), -- Programación con Ana Gómez
(3, 3, '2023-1', 2023); -- Cálculo con Luis Martínez
```

-- Insertar Inscripciones

```
INSERT INTO Inscripcion (id_estudiante, id_clase, fecha_inscripcion) VALUES
(1, 1, '2023-01-15'), -- Juan en BD
(1, 2, '2023-01-16'), -- Juan en Prog
(2, 1, '2023-01-15'), -- Maria en BD
(3, 3, '2023-01-20'); -- Pedro en Cálculo
```

-- Insertar Calificaciones

```
INSERT INTO Calificacion (id_inscripcion, nota, fecha_evaluacion) VALUES
(1, 85.5, '2023-05-20'),
(2, 90.0, '2023-05-22'),
(3, 78.0, '2023-05-20'),
(4, 65.5, '2023-05-25');
```

```
-- =====  
-- 3. MANIPULACIÓN DE DATOS (UPDATE / DELETE)  
-- =====
```

-- Actualizar el email de un estudiante

UPDATE Estudiante

SET email = 'juan.nuevo@student.edu'

WHERE id_estudiante = 1;

-- Actualizar los créditos de un curso

UPDATE Curso

SET creditos = 3

WHERE id_curso = 4;

-- Eliminar una calificación específica

DELETE FROM Calificacion

WHERE id_calificacion = 4;

```
-- =====  
-- 4. CONSULTAS BÁSICAS, JOINS Y ESTADÍSTICAS  
-- =====
```

-- Consulta Básica: Listar todos los estudiantes

SELECT * FROM Estudiante;

-- JOIN: Listar estudiantes y las clases que están tomando (con nombre del curso)

SELECT

e.nombre AS Estudiante,

e.apellido,

c.nombre AS Curso,

cl.semestre

FROM Estudiante e

JOIN Inscripcion i ON e.id_estudiante = i.id_estudiante

JOIN Clase cl ON i.id_clase = cl.id_clase

JOIN Curso c ON cl.id_curso = c.id_curso;

-- JOIN: Profesores y sus Departamentos

SELECT

p.nombre AS Profesor,

p.apellido,

d.nombre AS Departamento

```
FROM Profesor p
JOIN Departamento d ON p.id_departamento = d.id_departamento;
```

-- ESTADÍSTICAS: Promedio de notas por Curso

```
SELECT
    c.nombre AS Curso,
    AVG(cal.nota) AS Promedio_Nota
FROM Curso c
JOIN Clase cl ON c.id_curso = cl.id_curso
JOIN Inscripcion i ON cl.id_clase = i.id_clase
JOIN Calificacion cal ON i.id_inscripcion = cal.id_inscripcion
GROUP BY c.nombre;
```

-- ESTADÍSTICAS: Cantidad de estudiantes por Departamento

```
SELECT
    d.nombre AS Departamento,
    COUNT(DISTINCT i.id_estudiante) AS Total_Estudiantes
FROM Departamento d
JOIN Curso c ON d.id_departamento = c.id_departamento
JOIN Clase cl ON c.id_curso = cl.id_curso
JOIN Inscripcion i ON cl.id_clase = i.id_clase
GROUP BY d.nombre;
```