

-- Crear la base de datos:

```
CREATE DATABASE UPV;
```

```
USE UPV;
```

-- Definir las tablas:

-- Tabla Estudiantes

```
CREATE TABLE Estudiantes (  
    id_estudiante INT AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    apellido1 VARCHAR(100) NOT NULL,  
    apellido2 VARCHAR(100) NOT NULL,  
    fecha_nacimiento DATE NOT NULL,  
    email VARCHAR(100) NOT NULL UNIQUE,  
    telefono VARCHAR(20) UNIQUE  
);
```

-- Tabla Profesores

```
CREATE TABLE Profesores (  
    id_profesor INT AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    apellido1 VARCHAR(100) NOT NULL,  
    apellido2 VARCHAR(100) NOT NULL,  
    email VARCHAR(100) NOT NULL UNIQUE,  
    telefono VARCHAR(20) UNIQUE  
);
```

-- Tabla Cursos

```
CREATE TABLE Cursos (  
    id_curso INT AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    descripcion TEXT NOT NULL,  
    id_profesor INT NOT NULL  
);
```

-- Tabla Matrícula

```
CREATE TABLE Matricula (  
    id_matricula INT AUTO_INCREMENT PRIMARY KEY,
```

```

    id_estudiante INT NOT NULL,

    id_curso INT NOT NULL,

    pagado BOOLEAN NOT NULL DEFAULT FALSE,

    FOREIGN KEY (id_estudiante) REFERENCES Estudiantes(id_estudiante),

    FOREIGN KEY (id_curso) REFERENCES Cursos(id_curso)

);

-- Tabla Calificaciones

CREATE TABLE Calificaciones (

    id_calificacion INT AUTO_INCREMENT PRIMARY KEY,

    id_estudiante INT NOT NULL,

    id_curso INT NOT NULL,

    id_profesor INT NOT NULL,

    calificacion DECIMAL(4,2) NOT NULL,

    FOREIGN KEY (id_estudiante) REFERENCES Estudiantes(id_estudiante),

    FOREIGN KEY (id_curso) REFERENCES Cursos(id_curso),

    FOREIGN KEY (id_profesor) REFERENCES Profesores(id_profesor)

);

-- Insertar datos en Estudiantes

INSERT INTO Estudiantes (nombre, apellido1, apellido2, fecha_nacimiento, email, telefono)

VALUES

('Juan', 'Pérez', 'García', '1998-05-15', 'juan@example.com', '123456789'),

('María', 'Gómez', 'López', '1999-07-20', 'maria@example.com', '987654321'),

('Luis', 'Martínez', 'Núñez', '1997-12-05', 'luis@example.com', '555555555'),

('Ana', 'Hernández', 'Ramos', '2000-01-25', 'ana@example.com', '444444444'),

('Pedro', 'Díaz', 'Ortiz', '1996-09-10', 'pedro@example.com', '333333333'),

('Laura', 'Vega', 'Santos', '1998-03-15', 'laura@example.com', '222222222'),

('Carlos', 'Mendoza', 'Pérez', '1995-07-20', 'carlos@example.com', '111111111'),

('Lucía', 'Ramírez', 'Flores', '2001-11-30', 'lucia@example.com', '666666666'),

('Miguel', 'Fernández', 'Ruiz', '1999-06-18', 'miguel@example.com', '777777777'),

('Sara', 'Morales', 'Jiménez', '1997-08-22', 'sara@example.com', '888888888');

-- Insertar datos en Profesores

INSERT INTO Profesores (nombre, apellido1, apellido2, email, telefono)

```

VALUES

('Carlos', 'González', 'Martínez', 'carlos@example.com', '1122334455'),

('Ana', 'Martínez', 'Fernández', 'ana@example.com', '5566778899'),

('José', 'López', 'Díaz', 'jose@example.com', '9988776655'),

('Marta', 'Ruiz', 'García', 'marta@example.com', '8877665544');

-- Insertar datos en Cursos

INSERT INTO Cursos (nombre, descripcion, id_profesor)

VALUES

('Matemáticas', 'Curso de matemáticas básicas', 1),

('Historia', 'Curso de historia mundial', 2),

('Física', 'Curso de física avanzada', 3),

('Química', 'Curso de química orgánica', 4),

('Biología', 'Curso de biología molecular', 1),

('Literatura', 'Curso de literatura española', 2),

('Filosofía', 'Curso de filosofía moderna', 3),

('Inglés', 'Curso de inglés intermedio', 4),

('Programación', 'Curso de programación en Java', 1),

('Economía', 'Curso de economía internacional', 2);

-- Insertar datos en Matrícula

INSERT INTO Matricula (id_estudiante, id_curso, pagado) VALUES (1, 1, TRUE), (1, 2, FALSE), (1, 3, TRUE), (2, 4, FALSE), (2, 5, TRUE), (2, 6, FALSE), (3, 7, TRUE), (3, 8, FALSE), (3, 9, TRUE), (4, 10, FALSE), (4, 1, TRUE), (4, 2, FALSE), (5, 3, TRUE), (5, 4, FALSE), (5, 5, TRUE), (6, 6, FALSE), (6, 7, TRUE), (6, 8, FALSE), (7, 9, TRUE), (7, 10, FALSE), (7, 1, TRUE), (8, 2, FALSE), (8, 3, TRUE), (8, 4, FALSE), (9, 5, TRUE), (9, 6, FALSE), (9, 7, TRUE), (10, 8, FALSE), (10, 9, TRUE), (10, 10, FALSE);

-- Insertar datos en Calificaciones

INSERT INTO Calificaciones (id_estudiante, id_curso, id_profesor, calificacion) VALUES (1, 1, 1, 8.5), (1, 2, 2, 7.9), (1, 3, 3, 9.2), (2, 4, 4, 8.0), (2, 5, 1, 6.5), (2, 6, 2, 7.0), (3, 7, 3, 9.5), (3, 8, 4, 8.5), (3, 9, 1, 7.5), (4, 10, 2, 9.0), (4, 1, 1, 6.8), (4, 2, 2, 8.4), (5, 3, 3, 7.2), (5, 4, 4, 8.7), (5, 5, 1, 9.1), (6, 6, 2, 6.9), (6, 7, 3, 7.8), (6, 8, 4, 8.2), (7, 9, 1, 9.4), (7, 10, 2, 7.3), (7, 1, 1, 8.0), (8, 2, 2, 7.1), (8, 3, 3, 8.9), (8, 4, 4, 7.4), (9, 5, 1, 8.6), (9, 6, 2, 9.0), (9, 7, 3, 8.1), (10, 8, 4, 7.8), (10, 9, 1, 9.2), (10, 10, 2, 7.7);

-- Consulta SQL para la nota media que da cada profesor.

SELECT p.id_profesor AS ID, CONCAT(p.nombre, ' ', p.apellido1, ' ', p.apellido2) AS Nombre_Completo,

ROUND(AVG(c.calificacion), 2) AS Nota_Media

FROM Profesores p

JOIN Calificaciones c ON p.id_profesor = c.id_profesor

GROUP BY p.id_profesor, p.nombre, p.apellido1, p.apellido2;

-- Consulta SQL para la mejor calificaciones de cada estudiante.

SELECT

e.id_estudiante AS Id, CONCAT(e.nombre, ' ', e.apellido1, ' ', e.apellido2) AS Nombre_Completo,
MAX(c.calificacion) AS Mejor_Calificacion

FROM Estudiantes e

JOIN Calificaciones c ON e.id_estudiante = c.id_estudiante

GROUP BY e.id_estudiante, e.nombre, e.apellido1, e.apellido2;

-- Consulta SQL para ordenar a los estudiantes por los cursos en los que están matriculados.

SELECT e.id_estudiante AS Id, CONCAT(e.nombre, ' ', e.apellido1, ' ', e.apellido2) AS Nombre,

c.nombre AS Curso, c.descripcion AS Descripcion FROM Estudiantes e

JOIN Matricula m ON e.id_estudiante = m.id_estudiante

JOIN Cursos c ON m.id_curso = c.id_curso

ORDER BY e.id_estudiante, c.nombre;

-- Consulta de los cursos y sus calificaciones promedio, ordenados desde el curso más desafiante (curso con la calificación promedio más baja) hasta el curso más fácil

SELECT c.id_curso AS Id, c.nombre AS Curso, c.descripcion AS Descripcion,

ROUND(AVG(ca.calificacion), 2) AS calificacion_promedio

FROM Cursos c

JOIN Calificaciones ca ON c.id_curso = ca.id_curso

GROUP BY c.id_curso, c.nombre, c.descripcion

ORDER BY calificacion_promedio ASC;

-- Consulta para encontrar qué estudiante y profesor tienen más cursos en común

SELECT e.id_estudiante, CONCAT(e.nombre, ' ', e.apellido1, ' ', e.apellido2) AS Estudiante,

p.id_profesor, CONCAT(p.nombre, ' ', p.apellido1, ' ', p.apellido2) AS Profesor,

COUNT(*) AS cursos_en_comun

FROM Matricula m

JOIN Estudiantes e ON m.id_estudiante = e.id_estudiante

JOIN Cursos c ON m.id_curso = c.id_curso

JOIN Matricula m_prof ON m.id_curso = m_prof.id_curso AND m.id_estudiante != m_prof.id_estudiante

JOIN Profesores p ON c.id_profesor = p.id_profesor

GROUP BY e.id_estudiante, p.id_profesor

ORDER BY cursos_en_comun DESC

LIMIT 1;

