

Sistema de Gestión Académica

Modelo Entidad-Relación.....	2
Entidades y Atributos.....	2
Relaciones	3
Consideraciones de Diseño	3
Scripts de Creación y Carga de Datos.....	4
Ejercicios	7
1. Control de Matriculación	7
2. Actualización de Calificaciones	7
3. Reporte de Rendimiento Académico	7
4. Control de Asistencia	7
5. Histórico Académico	7
6. Dashboard Académico	7
7. Gestión de Profesores.....	7
8. Auditoría de Calificaciones.....	7
9. Reporte de Asistencia	7
10. Estado Académico.....	7

Modelo Entidad-Relación

Entidades y Atributos

1. FACULTAD
 - facultad_id (PK)
 - nombre
 - director
 - fecha_creacion
2. CARRERA
 - carrera_id (PK)
 - facultad_id (FK)
 - nombre
 - duracion_años
 - estado
3. PROFESOR
 - profesor_id (PK)
 - nombre
 - apellido
 - email
 - telefono
 - fecha_contratacion
 - salario
 - estado
4. ESTUDIANTE
 - estudiante_id (PK)
 - carrera_id (FK)
 - nombre
 - apellido
 - email
 - fecha_nacimiento
 - fecha_ingreso
 - estado
5. CURSO
 - curso_id (PK)
 - carrera_id (FK)
 - nombre
 - creditos
 - nivel
 - estado
6. SECCION
 - seccion_id (PK)
 - curso_id (FK)
 - profesor_id (FK)
 - semestre
 - año
 - cupo_maximo
 - estado
7. MATRICULA
 - matricula_id (PK)
 - estudiante_id (FK)
 - seccion_id (FK)
 - fecha_matricula
 - estado

8. CALIFICACION
- calificacion_id (PK)
- matricula_id (FK)
- nota_parcial1
- nota_parcial2
- nota_final
- estado

9. ASISTENCIA
- asistencia_id (PK)
- matricula_id (FK)
- fecha
- estado
- observacion

10. HISTORICO_ACADEMICO
- historico_id (PK)
- estudiante_id (FK)
- curso_id (FK)
- nota_final
- semestre
- año
- estado

Relaciones

1. FACULTAD (1:N) CARRERA
2. CARRERA (1:N) ESTUDIANTE
3. CARRERA (1:N) CURSO
4. CURSO (1:N) SECCION
5. PROFESOR (1:N) SECCION
6. SECCION (1:N) MATRICULA
7. ESTUDIANTE (1:N) MATRICULA
8. MATRICULA (1:1) CALIFICACION
9. MATRICULA (1:N) ASISTENCIA
10. ESTUDIANTE (1:N) HISTORICO_ACADEMICO
11. CURSO (1:N) HISTORICO_ACADEMICO

Consideraciones de Diseño

1. Se implementará un sistema de auditoría para cambios en calificaciones
2. Se mantendrá un histórico de todos los cursos tomados por estudiante
3. Se implementará control de cupos en matrículas
4. Se calculará automáticamente promedios y estado académico
5. Se mantendrá registro de asistencia por clase

Scripts de Creación y Carga de Datos

-- Creación de tablas

```
CREATE TABLE Facultad (  
    facultad_id INT PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    director VARCHAR(100) NOT NULL,  
    fecha_creacion DATE NOT NULL,  
    CONSTRAINT uk_facultad_nombre UNIQUE (nombre)  
);  
  
CREATE TABLE Carrera (  
    carrera_id INT PRIMARY KEY,  
    facultad_id INT NOT NULL,  
    nombre VARCHAR(100) NOT NULL,  
    duracion_años INT NOT NULL,  
    estado CHAR(1) DEFAULT 'A',  
    CONSTRAINT fk_carrera_facultad FOREIGN KEY (facultad_id) REFERENCES  
Facultad(facultad_id),  
    CONSTRAINT uk_carrera_nombre UNIQUE (nombre),  
    CONSTRAINT chk_carrera_estado CHECK (estado IN ('A', 'I'))  
);  
  
CREATE TABLE Profesor (  
    profesor_id INT PRIMARY KEY,  
    nombre VARCHAR(50) NOT NULL,  
    apellido VARCHAR(50) NOT NULL,  
    email VARCHAR(100) NOT NULL,  
    telefono VARCHAR(20),  
    fecha_contratacion DATE NOT NULL,  
    salario DECIMAL(10,2) NOT NULL,  
    estado CHAR(1) DEFAULT 'A',  
    CONSTRAINT uk_profesor_email UNIQUE (email),  
    CONSTRAINT chk_profesor_estado CHECK (estado IN ('A', 'I'))  
);  
  
CREATE TABLE Estudiante (  
    estudiante_id INT PRIMARY KEY,  
    carrera_id INT NOT NULL,  
    nombre VARCHAR(50) NOT NULL,  
    apellido VARCHAR(50) NOT NULL,  
    email VARCHAR(100) NOT NULL,  
    fecha_nacimiento DATE NOT NULL,  
    fecha_ingreso DATE NOT NULL,  
    estado CHAR(1) DEFAULT 'A',  
    CONSTRAINT fk_estudiante_carrera FOREIGN KEY (carrera_id) REFERENCES  
Carrera(carrera_id),  
    CONSTRAINT uk_estudiante_email UNIQUE (email),  
    CONSTRAINT chk_estudiante_estado CHECK (estado IN ('A', 'I'))  
);  
  
CREATE TABLE Curso (  
    curso_id INT PRIMARY KEY,  
    carrera_id INT NOT NULL,  
    nombre VARCHAR(100) NOT NULL,  
    creditos INT NOT NULL,  
    nivel INT NOT NULL,  
    estado CHAR(1) DEFAULT 'A',  
    CONSTRAINT fk_curso_carrera FOREIGN KEY (carrera_id) REFERENCES Carrera(carrera_id),  
    CONSTRAINT uk_curso_nombre_carrera UNIQUE (carrera_id, nombre),  
    CONSTRAINT chk_curso_estado CHECK (estado IN ('A', 'I'))
```

```

);

CREATE TABLE Seccion (
    seccion_id INT PRIMARY KEY,
    curso_id INT NOT NULL,
    profesor_id INT NOT NULL,
    semestre INT NOT NULL,
    año INT NOT NULL,
    cupo_maximo INT NOT NULL,
    estado CHAR(1) DEFAULT 'A',
    CONSTRAINT fk_seccion_curso FOREIGN KEY (curso_id) REFERENCES Curso(curso_id),
    CONSTRAINT fk_seccion_profesor FOREIGN KEY (profesor_id) REFERENCES
Profesor(profesor_id),
    CONSTRAINT chk_seccion_estado CHECK (estado IN ('A', 'I'))
);

CREATE TABLE Matricula (
    matricula_id INT PRIMARY KEY,
    estudiante_id INT NOT NULL,
    seccion_id INT NOT NULL,
    fecha_matricula DATE NOT NULL,
    estado CHAR(1) DEFAULT 'A',
    CONSTRAINT fk_matricula_estudiante FOREIGN KEY (estudiante_id) REFERENCES
Estudiante(estudiante_id),
    CONSTRAINT fk_matricula_seccion FOREIGN KEY (seccion_id) REFERENCES
Seccion(seccion_id),
    CONSTRAINT uk_matricula_estudiante_seccion UNIQUE (estudiante_id, seccion_id),
    CONSTRAINT chk_matricula_estado CHECK (estado IN ('A', 'I'))
);

CREATE TABLE Calificacion (
    calificacion_id INT PRIMARY KEY,
    matricula_id INT NOT NULL,
    nota_parcial1 DECIMAL(4,2),
    nota_parcial2 DECIMAL(4,2),
    nota_final DECIMAL(4,2),
    estado CHAR(1) DEFAULT 'A',
    CONSTRAINT fk_calificacion_matricula FOREIGN KEY (matricula_id) REFERENCES
Matricula(matricula_id),
    CONSTRAINT uk_calificacion_matricula UNIQUE (matricula_id),
    CONSTRAINT chk_calificacion_estado CHECK (estado IN ('A', 'I'))
);

CREATE TABLE Asistencia (
    asistencia_id INT PRIMARY KEY,
    matricula_id INT NOT NULL,
    fecha DATE NOT NULL,
    estado CHAR(1) NOT NULL,
    observacion VARCHAR(200),
    CONSTRAINT fk_asistencia_matricula FOREIGN KEY (matricula_id) REFERENCES
Matricula(matricula_id),
    CONSTRAINT chk_asistencia_estado CHECK (estado IN ('P', 'F', 'J'))
);

CREATE TABLE Historico_Academico (
    historico_id INT PRIMARY KEY,
    estudiante_id INT NOT NULL,
    curso_id INT NOT NULL,
    nota_final DECIMAL(4,2) NOT NULL,
    semestre INT NOT NULL,
    año INT NOT NULL,

```

```

    estado CHAR(1) DEFAULT 'A',
    CONSTRAINT fk_historico_estudiante FOREIGN KEY (estudiante_id) REFERENCES
Estudiante(estudiante_id),
    CONSTRAINT fk_historico_curso FOREIGN KEY (curso_id) REFERENCES Curso(curso_id),
    CONSTRAINT chk_historico_estado CHECK (estado IN ('A', 'I'))
);

-- Inserción de datos de prueba
INSERT INTO Facultad VALUES
(1, 'Ingeniería', 'Dr. Juan Pérez', '2000-01-01'),
(2, 'Ciencias', 'Dra. María López', '2000-01-01'),
(3, 'Humanidades', 'Dr. Carlos Ruiz', '2000-01-01');

INSERT INTO Carrera VALUES
(1, 1, 'Ingeniería de Sistemas', 5, 'A'),
(2, 1, 'Ingeniería Civil', 5, 'A'),
(3, 2, 'Matemáticas', 4, 'A');

INSERT INTO Profesor VALUES
(1, 'Ana', 'García', 'ana.garcia@universidad.edu', '123456789', '2015-01-01', 5000.00,
'A'),
(2, 'Pedro', 'Martínez', 'pedro.martinez@universidad.edu', '987654321', '2016-01-01',
4500.00, 'A'),
(3, 'Laura', 'Sánchez', 'laura.sanchez@universidad.edu', '456789123', '2017-01-01',
4800.00, 'A');

INSERT INTO Estudiante VALUES
(1, 1, 'Miguel', 'Torres', 'miguel.torres@universidad.edu', '2000-05-15', '2020-01-01',
'A'),
(2, 1, 'Carmen', 'Flores', 'carmen.flores@universidad.edu', '2001-03-20', '2020-01-01',
'A'),
(3, 2, 'Roberto', 'Díaz', 'roberto.diaz@universidad.edu', '2000-08-10', '2020-01-01',
'A');

INSERT INTO Curso VALUES
(1, 1, 'Programación I', 4, 1, 'A'),
(2, 1, 'Base de Datos', 4, 2, 'A'),
(3, 1, 'Redes', 3, 3, 'A');

INSERT INTO Seccion VALUES
(1, 1, 1, 1, 2024, 30, 'A'),
(2, 2, 2, 1, 2024, 25, 'A'),
(3, 3, 3, 1, 2024, 20, 'A');

INSERT INTO Matricula VALUES
(1, 1, 1, '2024-01-15', 'A'),
(2, 1, 2, '2024-01-15', 'A'),
(3, 2, 1, '2024-01-16', 'A');

INSERT INTO Calificacion VALUES
(1, 1, 8.5, 9.0, NULL, 'A'),
(2, 2, 7.5, 8.0, NULL, 'A'),
(3, 3, 9.0, 8.5, NULL, 'A');

INSERT INTO Asistencia VALUES
(1, 1, '2024-02-01', 'P', NULL),
(2, 1, '2024-02-08', 'P', NULL),
(3, 2, '2024-02-01', 'J', 'Justificación médica');

INSERT INTO Historico_Academico VALUES
(1, 1, 1, 8.75, 1, 2024, 'A'),

```

```
(2, 1, 2, 7.75, 1, 2024, 'A'),  
(3, 2, 1, 8.75, 1, 2024, 'A');
```

Ejercicios

1. Control de Matriculación

Crear un trigger que verifique el cupo disponible antes de permitir una nueva matrícula.

2. Actualización de Calificaciones

Crear un procedimiento almacenado que calcule y actualice la nota final basada en las notas parciales.

3. Reporte de Rendimiento Académico

Crear una vista que muestre el promedio de calificaciones por estudiante y carrera.

4. Control de Asistencia

Crear un procedimiento almacenado para registrar asistencia masiva.

5. Histórico Académico

Crear un trigger que actualice automáticamente el histórico académico cuando se completa un curso.

6. Dashboard Académico

Crear una vista que muestre estadísticas generales por carrera.

7. Gestión de Profesores

Crear un procedimiento almacenado para asignar profesores a secciones.

8. Auditoría de Calificaciones

Crear un trigger que registre cambios en las calificaciones.

9. Reporte de Asistencia

Crear una vista que muestre el porcentaje de asistencia por estudiante y curso.

10. Estado Académico

Crear un procedimiento almacenado que determine el estado académico del estudiante.

Las soluciones a estos ejercicios se entregarán en un solo archivo pdf con nombre y número del grupo, indicando entre paréntesis (no participo) para el que no colaboró en el desarrollo.