





# Diccionario

## Estudiante

Campo	Tipo	Descripción
id-estudiante	INT	identificador Estudiante
nombre	VARCHAR(100)	nombre estudiante
correo	VARCHAR(100)	correo estudiante
Carrera	VARCHAR(100)	Carrera que estudia

## Profesor

Campo	Tipo	Descripción
id-Profesor	INT	identificador Profesor
nombre	VARCHAR(100)	nombre Profesor
Especialidad	VARCHAR(100)	Area de Conocimiento Principal

## Curso

Campo	Tipo	Descripción
id-curso	INT	identificador Curso
nombre	VARCHAR(100)	nombre del curso
Creditos	INT	Creditos académicos

## Inscripción

Campo	Tipo	Descripción
id-estudiante	INT	Fk al estudiante
id-curso	INT	Fk al curso
Semestre	VARCHAR(10)	Periodo en que realiza inscripciones

## Curso-Profesor

Campo	Tipo	Descripción
id-curso	INT	Fk al curso
id-Profesor	INT	Fk al profesor

## Nota

Campo	Tipo	Descripción
id-estudiante	INT	Fk al estudiante
id-curso	INT	Fk al curso
nota	DECIMAL(3,1)	Nota Final por estudiante

## Facultad

Campo	Tipo	Descripción
id-Facultad	INT	identificador Facultad
nombre	VARCHAR	nombre Facultad



```
CREATE DATABASE Colegio;  
USE Colegio;
```

```
CREATE TABLE Estudiante (  
  id_Estudiante INT PRIMARY KEY;  
  nombre VARCHAR(100),  
  Correo VARCHAR(100),  
  Carrera VARCHAR(100),  
);
```

```
CREATE TABLE Profesor (  
  id_Profesor INT PRIMARY KEY;  
  nombre VARCHAR(100),  
  Especialidad VARCHAR(100),  
);
```

```
CREATE TABLE Curso (  
  id_Curso INT PRIMARY KEY,  
  nombre VARCHAR(100),  
  Creditos INT,  
);
```

```
CREATE TABLE Inscripcion (  
  id_Estudiante,  
  id_Curso INT,  
  Semestre VARCHAR(10),  
  PRIMARY KEY (id_Estudiante, id_Curso),  
  FOREIGN KEY (id_Estudiante) REFERENCES Estudiante (id_Estudiante),  
  FOREIGN KEY (id_Curso) REFERENCES Curso (id_Curso),  
);
```

```
CREATE TABLE Curso_Profesor (  
  id_Curso INT,  
  id_Profesor INT,  
  PRIMARY KEY (id_Curso, id_Profesor),  
  FOREIGN KEY (id_Curso) REFERENCES Curso (id_Curso),  
  FOREIGN KEY (id_Profesor) REFERENCES Profesor (id_Profesor),  
);
```

```
CREATE TABLE Nota (  
  id_Estudiante INT,  
  id_Profesor INT,  
  PRIMARY KEY (id_Estudiante, id_Profesor),  
  FOREIGN KEY (id_Estudiante) REFERENCES Estudiante (id_Estudiante),  
  FOREIGN KEY (id_Profesor) REFERENCES Profesor (id_Profesor),  
);
```



```

CREATE TABLE Facultad
(id_Facultad INT PRIMARY KEY,
nombre VARCHAR (100),
);

```

```

INSERT INTO Facultad VALUES (1, 'Ingeniería');
INSERT INTO Facultad VALUES (2, 'Medicina');
INSERT INTO Facultad VALUES (3, 'Derecho');
INSERT INTO Facultad VALUES (4, 'Ciencias');
INSERT INTO Facultad VALUES (5, 'Humanidades');

```

```

INSERT INTO Estudiantes VALUES (1, 'Laura Ramos', 'laura@email.com', 'Inge');
INSERT INTO Estudiantes VALUES (2, 'María Pías', 'maria@email.com', 'Medicina');
INSERT INTO Estudiantes VALUES (3, 'Ana Pérez', 'ana@email.com', 'Derecho');
INSERT INTO Estudiantes VALUES (4, 'Luis Torres', 'luis@email.com', 'Ciencias');
INSERT INTO Estudiantes VALUES (5, 'Sofía Gómez', 'sofia@email.com', 'Humanidades');

```

```

INSERT INTO Profesor VALUES (1, 'Dr. Andrade', 'Física');
INSERT INTO Profesor VALUES (2, 'Dr. Rivera', 'Biología');
INSERT INTO Profesor VALUES (3, 'Dr. Morales', 'Programación');
INSERT INTO Profesor VALUES (4, 'Dr. Mejía', 'Psicología');
INSERT INTO Profesor VALUES (5, 'Dr. Castro', 'Derecho Penal');

```

```

INSERT INTO Curso VALUES (1, 'Física 1', 3);
INSERT INTO Curso VALUES (2, 'Biología General', 4);
INSERT INTO Curso VALUES (3, 'Estructura de datos', 4);
INSERT INTO Curso VALUES (4, 'Psicología Social', 3);
INSERT INTO Curso VALUES (5, 'Derecho Civil', 4);

```

```

INSERT INTO Curso_Profesor VALUES (1, 1);
INSERT INTO Curso_Profesor VALUES (2, 2);
INSERT INTO Curso_Profesor VALUES (3, 3);
INSERT INTO Curso_Profesor VALUES (4, 4);
INSERT INTO Curso_Profesor VALUES (5, 5);

```

```

INSERT INTO Inscripción VALUES (1, 1, '2024-09-10');
INSERT INTO Inscripción VALUES (2, 2, '2024-09-11');
INSERT INTO Inscripción VALUES (3, 3, '2024-09-12');
INSERT INTO Inscripción VALUES (4, 4, '2024-09-13');
INSERT INTO Inscripción VALUES (5, 5, '2024-09-14');

```

```

INSERT INTO Nota VALUES (1, 1, 4.5);
INSERT INTO Nota VALUES (2, 2, 3.8);
INSERT INTO Nota VALUES (3, 3, 4.2);
INSERT INTO Nota VALUES (4, 4, 4.9);
INSERT INTO Nota VALUES (5, 5, 3.5);

```



```

SELECT P.nombre, C.nombre
FROM Inscricion i
JOIN Estudiante e ON i.id_estudiante = e.id_estudiante
JOIN Curso c ON i.id_curso = c.id_curso;

```

```

SELECT P.nombre, C.nombre
FROM Curso_Profesor CP
JOIN Profesor P ON CP.id_Profesor = P.id_Profesor
JOIN Curso c ON CP.id_curso = c.id_curso;

```

```

SELECT P.nombre, n.nota
FROM Nota n
JOIN Estudiante e ON n.id_estudiante = e.id_estudiante;

```

```

SELECT nombre
FROM Curso
WHERE Creditos > 3;
SELECT COUNT(*) AS total_estudiante
FROM Estudiante;

```

```

DELETE FROM Nota WHERE id_estudiante = 1 AND id_curso = 1;
DELETE FROM Inscricion WHERE id_estudiante = 2 AND id_curso = 2;
DELETE FROM Curso_Profesor WHERE id_curso = 3 AND id_Profesor = 3;
DELETE FROM Curso WHERE id_curso = 4;
DELETE FROM Profesor WHERE id_Profesor = 5;

```

```

CREATE FUNCTION PromedioEstudiantes (eid INT) RETURNS DECIMAL
(3,1) BEGIN DECLARE p DECIMAL (3,1);
SELECT AVG(nota) INTO p FROM Nota WHERE id_estudiante = eid;
RETURN p;

```

```

CREATE FUNCTION CursorInscripciones (eid INT) RETURNS INT
BEGIN DECLARE c INT;
SELECT COUNT(*) INTO c FROM Inscricion WHERE id_estudiante = eid;
RETURN c;
END;

```

```

CREATE FUNCTION totalCurso () RETURNS INT BEGIN DECLARE
c INT;
SELECT COUNT(*) INTO c FROM Inscricion WHERE id_estudiante =
eid;
RETURN c;
END;

```

```

CREATE FUNCTION totalProfesores () RETURNS INT BEGIN DECLARE
c INT;
SELECT COUNT(*) INTO c FROM Profesor;

```



```
RETURN C;  
END;
```

```
CREATE FUNCTION EstudiantePorCurso (cid INT) RETURNS INT  
BEGIN DECLARE C INT;  
SELECT COUNT(*) INTO C FROM Inscripcion WHERE id-curso = cid;  
RETURN C;  
END;
```

```
CREATE FUNCTION PromedioCurso (cid INT) RETURNS INT BEGIN  
DECLARE P DECIMAL (3,1);  
SELECT AVG(nota) INTO P FROM Nota WHERE id-curso = cid;  
RETURN C;  
END;
```

```
CREATE FUNCTION CantidadFacultades () RETURNS INT BEGIN  
DECLARE C INT;  
SELECT COUNT(*) INTO C FROM Facultad;  
RETURN P;  
END;
```

```
CREATE FUNCTION EstudiantesPorFacultad (FAC. VARCHAR (40)) RETURNS  
INT BEGIN DECLARE C INT  
SELECT COUNT(*) INTO C FROM Estudiante WHERE Carrera = FAC;  
RETURN C;  
END;
```

```
CREATE FUNCTION CursoPorProfesores (pid INT) RETURNS INT  
BEGIN DECLARE C INT;  
SELECT COUNT(*) INTO C FROM Curso-Profesores WHERE id-Pro-  
fesor = pid;  
RETURN C;  
END;
```

```
CREATE FUNCTION CursosPorCursos (cid INT) RETURNS INT BE-  
GIN DECLARE C INT;  
SELECT COUNT(*) INTO C FROM Nota WHERE id-curso = cid  
AND nota >= 30;  
RETURN C;  
END;
```

```
CREATE PROCEDURE RegistrarNota (IN pid INT, IN cid INT, IN  
nota DECIMAL (3,1)) BEGIN INSERT INTO Nota (id-estudiante, id-curso, nota)  
VALUES (cid, cid, nota); END;
```

```
SELECT nombre FROM Curso WHERE id-curso IN (  
SELECT id-curso FROM Nota GROUP BY id-curso HAVING AVG(nota)  
> 4.0 );
```