Database Documentation for Student Management System

This document provides a comprehensive overview of a database system designed to manage information related to students, professors, colleges, and courses. The database is structured to store and maintain data with referential integrity across multiple entities. The goal is to create a robust system that can manage a large volume of data while ensuring efficiency, accuracy, and scalability.

The database comprises four main tables:

Student: Holds data related to students.

Professor: Contains information about professors.

College: Maintains details about different colleges.

Course: Stores information about the courses offered.

Each table is interconnected, ensuring relational integrity through primary and foreign key constraints. This documentation will cover each table in-depth, describe their relationships, and provide details on how data is structured within the database.

1. Database Schema Overview

The database consists of the following four tables, with details on their columns, data types, and relationships:

Student

Professor

College

Course

2. Table Descriptions

2.1 Student Table

- **Purpose**: The Student table stores all necessary data about students, including their personal information, the course they are enrolled in, and the college they belong to.
- **Table Structure**:

Column	Data Type	Description
stud_id	INT	Primary key, unique student identifier
stud_name	VARCHAR(100)	Name of the student
stud_add	VARCHAR(255)	Student's address
stud_phone_no	VARCHAR(15)	Student's phone number, unique
stud_age	INT	Age of the student
course_id	INT	Foreign key to Course(course_id)
prof_id	INT	Foreign key to Professor(prof_id)
college_id	INT	Foreign key to College(college_id)

- **Primary Key**: stud_id

- **Foreign Keys**:
  - course_id → Course(course_id)
  - prof_id → Professor(prof_id)
  - college_id → College(college_id)

2.2 Professor Table

- **Purpose**: The Professor table holds data about professors who are linked to students through the courses they teach.
- **Table Structure**:

Column	Data Type	Description
prof_id	INT	Primary key, unique professor identifier
prof_name	VARCHAR(100)	Name of the professor
prof_age	INT	Age of the professor
prof_gender	CHAR(1)	Gender of the professor (M/F)
prof_salary	INT	Salary of the professor
prof_email	VARCHAR(100)	Email of the professor, unique

- **Primary Key**: prof_id

2.3 College Table

- **Purpose**: The College table stores details about different colleges where students are enrolled.
- **Table Structure**:

Column	Data Type	Description
college_id	INT	Primary key, unique college identifier
college_name	VARCHAR(255)	Name of the college
city	VARCHAR(100)	City where the college is located
state	VARCHAR(100)	State where the college is located
col_established_year	YEAR	Year the college was established

- **Primary Key**: college_id

2.4 Course Table

- **Purpose**: The Course table holds information on the courses offered at various colleges, which students can enroll in.
- **Table Structure**:

Column	Data Type	Description
course_id	INT	Primary key, unique course identifier
course_name	VARCHAR(255)	Name of the course
course_duration	INT	Duration of the course in years

Column	Data Type	Description
course_level	VARCHAR(50)	Course level (Undergraduate, Graduate, etc.)

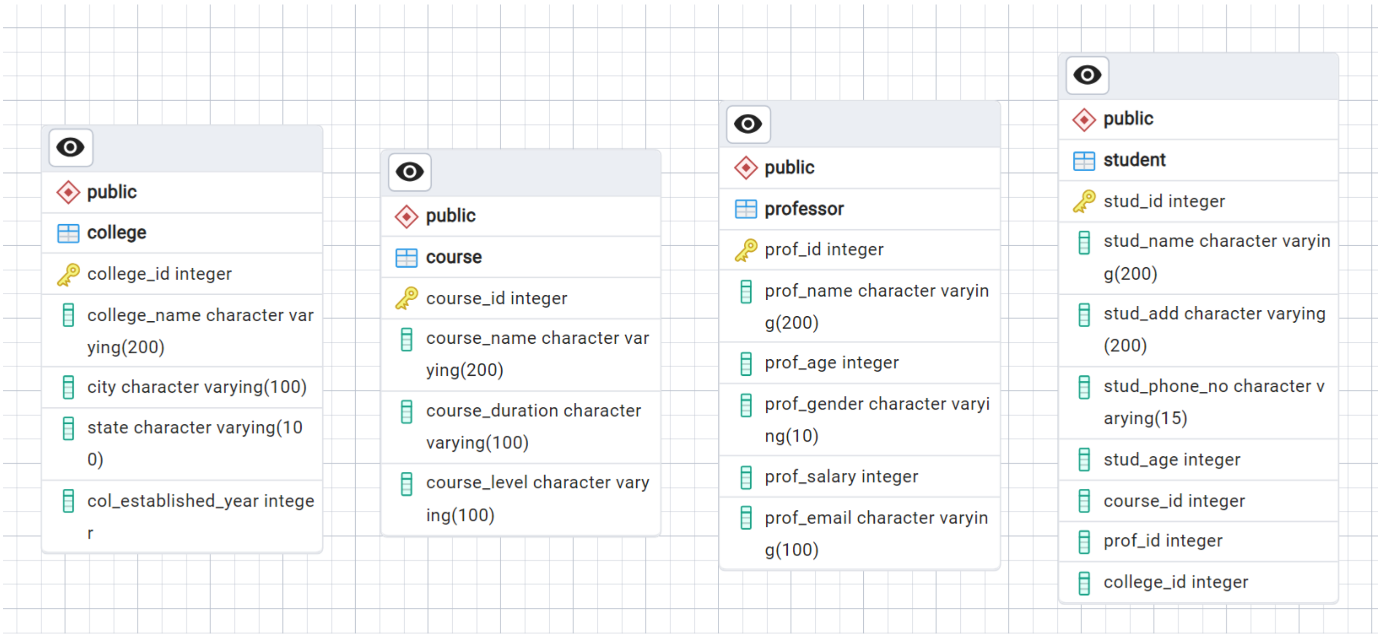
- **Primary Key**: course_id

3. Database Relationships

The following relationships exist between the tables:

- **Student-Professor Relationship**: Students are linked to their respective professors through the prof_id. A professor may teach multiple students, but a student is taught by only one professor at a time.
- **Student-Course Relationship**: Each student is enrolled in a course, which is stored in the course_id. A course can have multiple students, but a student can be associated with only one course in this structure.
- **Student-College Relationship**: Each student belongs to a specific college, identified by the college_id. A college can have many students, but each student is enrolled in only one college.

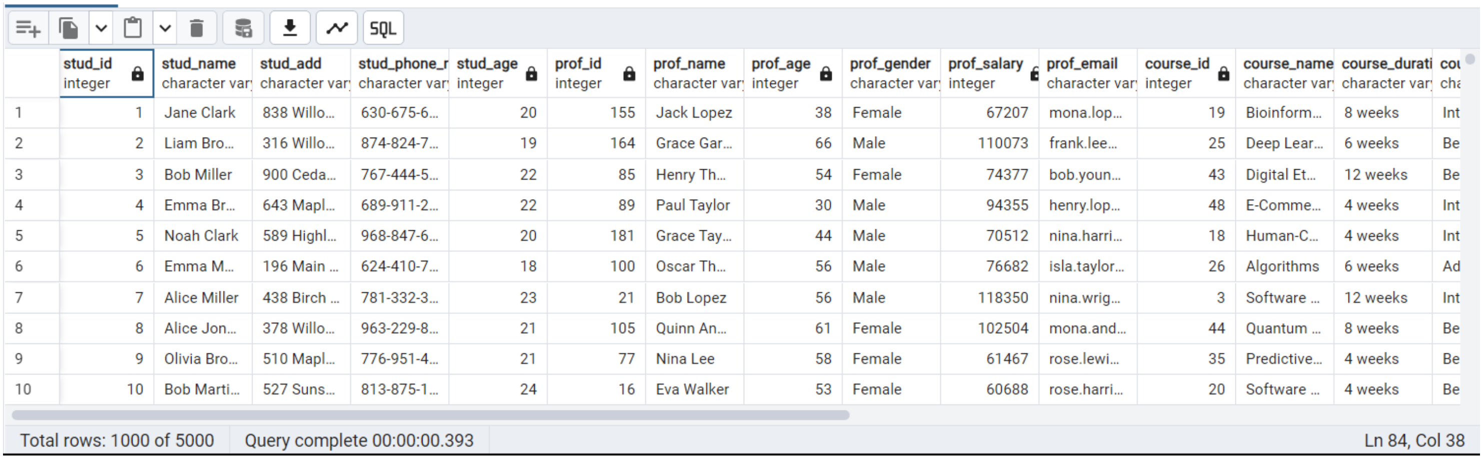
4. ER Diagram


public
college
college_id integer
college_name character varying(200)
city character varying(100)
state character varying(100)
col_established_year integer
public
course
course_id integer
course_name character varying(200)
course_duration character varying(100)
course_level character varying(100)
public
professor
prof_id integer
prof_name character varying(200)
prof_age integer
prof_gender character varying(10)
prof_salary integer
prof_email character varying(100)
public
student
stud_id integer
stud_name character varying(200)
stud_add character varying (200)
stud_phone_no character varying(15)
stud_age integer
course_id integer
prof_id integer
college_id integer

5. Database Queries

```
SELECT s.stud_id, s.stud_name, s.stud_add, s.stud_phone_no, s.stud_age,
       s.prof_id, p.prof_name, p.prof_age, p.prof_gender, p.prof_salary, p.prof_email,
       s.course_id, c.course_name, c.course_duration, c.course_level,
       s.college_id, col.college_name, col.city, col.state, col.col_established_year
FROM student AS s
FULL JOIN professor AS p
   ON s.prof_id = p.prof_id
FULL JOIN course AS c
   ON s.course_id = c.course_id
```

```
FULL JOIN college AS col
    ON s.college_id = col.college_id;
```



	stud_id integer	stud_name character var	stud_add character var	stud_phone_r character var	stud_age integer	prof_id integer	prof_name character var	prof_age integer	prof_gender character var	prof_salary integer	prof_email character var	course_id integer	course_name character var	course_durati character var	cou cha
1	1	Jane Clark	838 Willo...	630-675-6...	20	155	Jack Lopez	38	Female	67207	mona.lop...	19	Bioinform...	8 weeks	Int
2	2	Liam Bro...	316 Willo...	874-824-7...	19	164	Grace Gar...	66	Male	110073	frank.lee...	25	Deep Lear...	6 weeks	Be
3	3	Bob Miller	900 Ceda...	767-444-5...	22	85	Henry Th...	54	Female	74377	bob.youn...	43	Digital Et...	12 weeks	Be
4	4	Emma Br...	643 Mapl...	689-911-2...	22	89	Paul Taylor	30	Male	94355	henry.lop...	48	E-Comme...	4 weeks	Int
5	5	Noah Clark	589 Highl...	968-847-6...	20	181	Grace Tay...	44	Male	70512	nina.harri...	18	Human-C...	4 weeks	Int
6	6	Emma M...	196 Main ...	624-410-7...	18	100	Oscar Th...	56	Male	76682	isla.taylor...	26	Algorithms	6 weeks	Ad
7	7	Alice Miller	438 Birch ...	781-332-3...	23	21	Bob Lopez	56	Male	118350	nina.wrig...	3	Software ...	12 weeks	Int
8	8	Alice Jon...	378 Willo...	963-229-8...	21	105	Quinn An...	61	Female	102504	mona.and...	44	Quantum ...	8 weeks	Be
9	9	Olivia Bro...	510 Mapl...	776-951-4...	21	77	Nina Lee	58	Female	61467	rose.lewi...	35	Predictive...	4 weeks	Be
10	10	Bob Marti...	527 Suns...	813-875-1...	24	16	Eva Walker	53	Female	60688	rose.harri...	20	Software ...	4 weeks	Be

Total rows: 1000 of 5000 Query complete 00:00:00.393 Ln 84, Col 38

6. Indexing

Indexes have been applied on frequently queried fields such as stud_id, prof_id, course_id, and college_id to optimize query performance.

7. Conclusion

This documentation presents an in-depth look at the database design, including its schema, table descriptions, relationships, and constraints. The database is optimized for maintaining student, professor, college, and course records with data integrity and normalization at its core. This structure allows for efficient data retrieval, scalability, and support for further extensions if needed.