

Database Design Document

1.0 Overview

The database for the Internship Placement Manager is designed using a relational model to efficiently store and manage information about students, mentors, employers, placements, and evaluations. The schema is implemented using **SQLModel**, which provides an object-oriented approach to database table creation and relationships.

2.0 Entity-Relationship Diagram

3.0 Table Descriptions and Relationships

Student Table

- **Purpose:** Stores information about the students participating in the program.
- **Key Fields:**
 - `id` (Primary Key)
 - `full_name` (String)
 - `email` (Unique String)
 - `major` (String)
 - `hashed_password` (String)
- **Relationships:**
 - **Many-to-Many with Placement:** A student can be on multiple placements, and a placement can have multiple students. This is handled by the `StudentPlacementLink` association table.

- **Many-to-Many with Mentor:** A student can have multiple mentors, and a mentor can have multiple students. This is handled by the MentorStudentLink association table.
- **One-to-Many with Evaluation:** A student can give multiple evaluations (evaluations_given) and receive multiple evaluations (evaluations_received).

Mentor Table

- **Purpose:** Stores information about the mentors.
- **Key Fields:**
 - id (Primary Key)
 - full_name (String)
 - email (Unique String)
 - field (String)
 - hashed_password (String)
- **Relationships:**
 - **One-to-Many with Placement:** A mentor can be assigned to multiple placements.
 - **One-to-Many with Evaluation:** A mentor can give multiple evaluations (evaluations_given).
 - **Many-to-Many with Student:** (See Student table description).

Employer Table

- **Purpose:** Stores information about the companies offering placements.
- **Key Fields:**
 - id (Primary Key)
 - company_name (String)
 - email (Unique String)

- contact_person (String)
- industry (String)
- hashed_password (String)
- **Relationships:**
 - **One-to-Many with Placement:** An employer can offer multiple placements.
 - **One-to-Many with Evaluation:** An employer can give multiple evaluations (evaluations_given).

Placement Table

- **Purpose:** Stores details about each internship placement.
- **Key Fields:**
 - id (Primary Key)
 - title (String)
 - description (String)
 - start_date (Date)
 - end_date (Date)
 - status (String)
 - employer_id (Foreign Key to Employer table)
 - mentor_id (Foreign Key to Mentor table)
- **Relationships:**
 - **Many-to-One with Employer:** Each placement belongs to one employer.
 - **Many-to-One with Mentor:** Each placement is managed by one mentor.
 - **Many-to-Many with Student:** (See Student table description).
 - **One-to-Many with Evaluation:** A placement can have multiple evaluations.

Evaluation Table

- **Purpose:** Stores the feedback and ratings for a placement.
- **Key Fields:**
 - id (Primary Key)
 - feedback (String)
 - rating (Integer)
 - created_at (Datetime)
 - placement_id (Foreign Key to Placement table)
 - subject_id (Foreign Key to Student table - the student being evaluated)
 - mentor_evaluator_id (Foreign Key to Mentor table)
 - employer_evaluator_id (Foreign Key to Employer table)
 - student_evaluator_id (Foreign Key to Student table - the student giving the evaluation)
- **Relationships:**
 - **Many-to-One with Placement:** Each evaluation is for one placement.
 - **Many-to-One with Student:** Each evaluation is about a single student (subject).
 - **Many-to-One with Mentor:** Each evaluation can be given by a mentor (mentor_evaluator).
 - **Many-to-One with Employer:** Each evaluation can be given by an employer (employer_evaluator).
 - **Many-to-One with Student:** Each evaluation can be given by a student (student_evaluator).

4.0 One-to-One vs. One-to-Many vs. Many-to-Many

- **One-to-Many:**

- Employer to Placement: An employer can have many placements.
- Mentor to Placement: A mentor can oversee many placements.
- Placement to Evaluation: A placement can have many evaluations.

- **Many-to-Many:**

- Student to Placement: A student can be in many placements, and a placement can have many students.
- Student to Mentor: A student can have many mentors, and a mentor can have many students.

- **One-to-One:** There are no direct one-to-one relationships in this schema.