Internship database system

Description :

- The Field Training 2 system is designed to manage student internships, a mandatory requirement for Level 3 students worth 2 credit hours. The system tracks registrations, organization profiles, internship details, and evaluations to streamline the internship process.

- ***Entities:***

  Students: Register for internships, submit reports, and view grades. Each student is assigned both an internal (academic) and external mentor.

  internship Coordinator: Manages organization profiles, approves or rejects internships, assigns mentors, and finalizes student grades.

  External Mentor (Organization): Evaluates student performance during the internship and uploads evaluation reports.

  internship Evaluator: Reviews and grades student reports based on their internship performance.

  Admin: Manages all system users, including coordinators, mentors, and students, along with organization and internship data.

System Functions:

- Tracks student registrations and organization involvement.

- Generates reports on student participation, mentor assignments, and grade distributions.

- Updates weekly on student progress and evaluation status.

- Produces semester-end and annual reports to highlight performance trends and top-performing students.

This system improves efficiency in managing internships, ensuring smooth . communication between students, mentors, and coordinators.

Business Rules :

- Each student must be registered for Field Training 2 before participating in an internship.

- A student can only be enrolled in Field Training 2 if they have completed all required prerequisite courses (Field Training 1) and (UCCD Workshop).

- Each student must have two mentors: an external mentor from the organization who evaluates their performance, and an internal mentor (academic advisor) who guides the student throughout the internship.

- Each internship must be associated with one organization and have at least one external mentor from that organization.

- The system must generate weekly reports detailing active internships, student evaluations, and grade distributions. Additionally, at the end of each semester, a consolidated report of internship participation, organization involvement, and grade performance must be generated for academic review.

- Only authorized internship evaluators and internship coordinators can modify grades in the system. Changes to grades must be logged and auditable for transparency.

- Each organization must be approved by the internship coordinator before students can be assigned to it.

- Each internal mentor must be assigned a specific number of students to ensure a balanced workload.

- This ratio should be monitored, and coordinators cannot assign more students than the predefined maximum for each mentor.

Queries:

Queries:

1. What is the internship with the highest grade achieved among students?

2-Which internal mentor (doctor) that all high-grade students choose?

3-How many students are assigned to each organization during a specific semester?

4-What are the evaluations provided by external mentors and the guidance given by internal mentors?

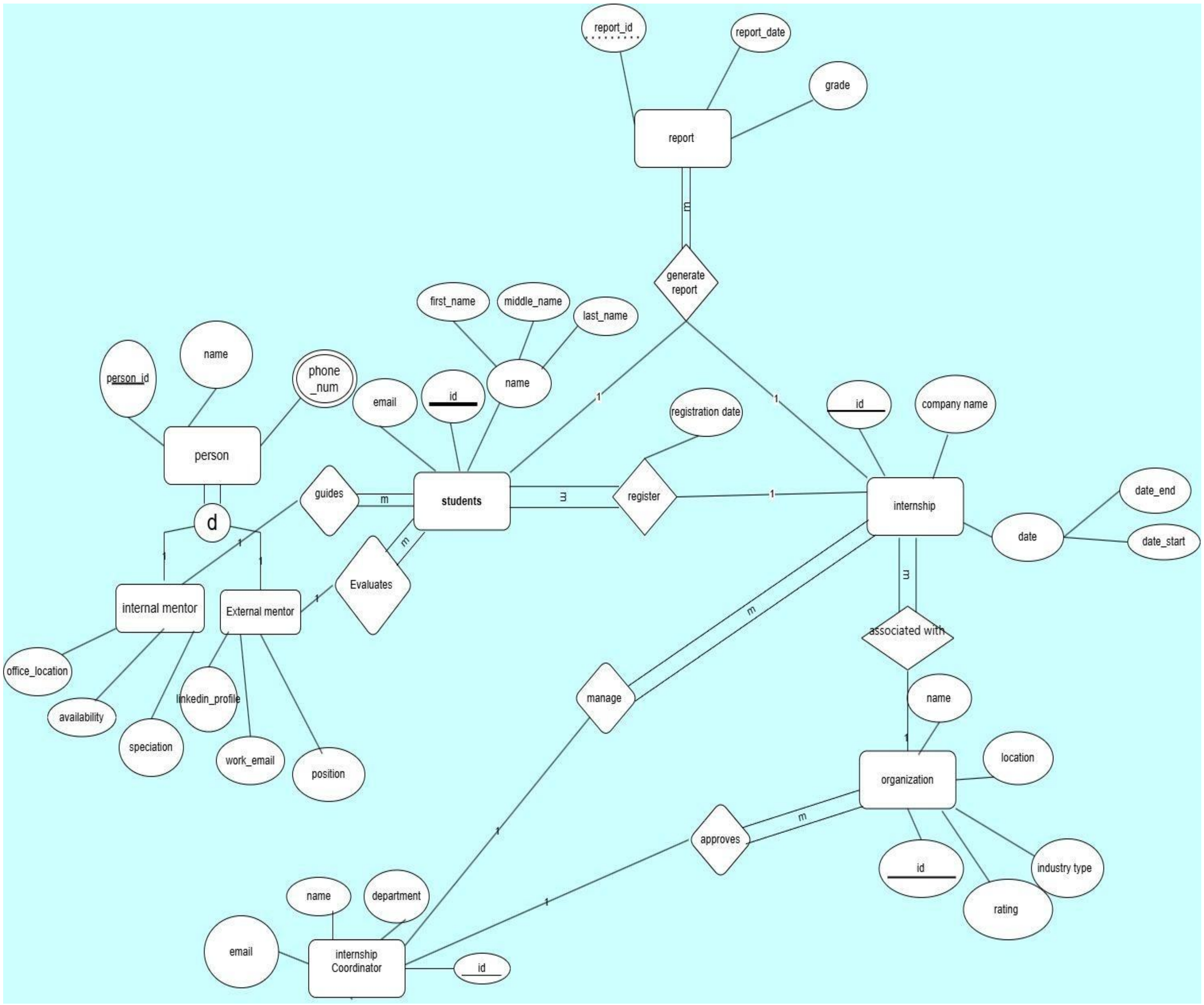
5-What is the internship duration and completion rate by field of study?

6-What is the progress of internships in terms of report submissions and grades?

7- How many students have received low grades or weak evaluations to be warned by their mentors?

8-Which students are assigned to which organizations, and who are their external mentors?

EERD:



Modification in EERD

Internal mentor - - > we added two attributes(availability & office_location)

External mentor - - > we added two attributes(work_email& linkedIn_profile)

Organization - - > we added two attributes(Rating & industry_type)

Person - - > we modified the contact_info to be phone_number

Internship - - > we added a composite attribute (date) and its subclasses start_date & end_date

We added a new weak relationship weak(generate report) ,and a new weak entity entity (Report) and its attibutes - - >(report_id & report_date & grade)

We added a relationship(register) between internship & students with a new attribute (registeration_date)

Normalization & relational schema Relational schema:

Internship coordinator(id,email,name,department)

Organization(id,rating,industry_type,Location,name,internship coordinator(Foreign key referenced to internship coordinator id)

Internship(id,company_name,date_end,date_start,internship coordinator id(Foreign key referenced to internship coordinator id),organization id(Foreign key referenced to organization id))

Student(id,email,F_name,middle_name,Last_name,internal mentor id (Foreign key referenced to internal person id),external mentor(Foreign key referenced to person id),Internal id(Foreign key referenced to internship id),registeration_date)

Person(person_id,name)

Phone_num(phone_num,person_id(Foreign key referenced to person(person id)))

Internal mentor(office_location,availability,specification,person id(Foreign key referenced to person(person id)))

External mentor(Linked_in profile,work_email,position, person id(Foreign key referenced to person(person id))

Report(report_id,report_data,grade,sid(Foreign key referenced to student_id),internship_id(Foreign key referenced to internship_id))

Generate report(report_id(Foreign key referenced to report_id),sid(Foreign key referenced to student_id),internship_id(Foreign key referenced to internship_id)

Normalization:

First normal form:

Only automatic attributes(simple,single_valued)

Schema:

Internship coordinator(id,email,name,department)

Organization(id,rating,industry_type,Location,name,internship coordinator(Foreign key referenced to internship coordinator id)

Internship(id,company_name,date_end,date_start,internship coordinator id(Foreign key referenced to internship coordinator id),organization id(Foreign key referenced to organization id))
Student(id,email,F_name,middle_name,Last_name,internal mentor id (Foreign key referenced to internal person id),external mentor(Foreign key referenced to person id),Internal id(Foreign key referenced to internship id),registeration_date)

Person(person_id,name)

Phone_num(phone_num,person_id(Foreign key referenced to person(person id)))

Internal mentor(office_location,availability,specification,person id(Foreign key referenced to person(person id)))

External mentor(Linked_in profile,work_email,position, person id(Foreign key referenced to person(person id))

Report(report_id,report_data,grade,sid(Foreign key referenced to student_id),internship_id(Foreign key referenced to internship_id))

Generate report(report_id(Foreign key referenced to report_id),sid(Foreign key referenced to student_id),internship_id(Foreign key referenced to internship_id)

Second normal form:

1NF plus every non-key attribute is fully functionally dependent on the Entire primary key

Non-partial functional dependencies

Note that out scheme already fully functional dependent and don't have any partially dependent

Schema:

Internship coordinator(id,email,name,department)
Organization(id,rating,industry_type,Location,name,internship coordinator(Foreign key referenced to internship coordinator id)

Internship(id,company_name,date_end,date_start,internship coordinator id(Foreign key referenced to internship coordinator id),organization id(Foreign key referenced to organization id))

Student(id,email,F_name,middle_name,Last_name,internal mentor id (Foreign key referenced to internal person id),external mentor(Foreign key referenced to person id),Internal id(Foreign key referenced to internship id),registeration_date)

Person(person_id,name)

Phone_num(phone_num,person_id(Foreign key referenced to person(person id)))

Internal mentor(office_location,availability,specification,person id(Foreign key referenced to person(person id)))

External mentor(Linked_in profile,work_email,position, person id(Foreign key referenced to person(person id))

Report(report_id,report_data,grade,sid(Foreign key referenced to student_id),internship_id(Foreign key referenced to internship_id))

Generate report(report_id(Foreign key referenced to report_id),sid(Foreign key referenced to student_id),internship_id(Foreign key referenced to internship_id)

Third normal form:

Eliminate certain transitive dependencies

Each attribute must describe the key, the whole key and nothing but the key

Note that out scheme already fully functional dependent and don't have any transitive dependent

Schema:

Internship coordinator(id,email,name,department)
Organization(id,rating,industry_type,Location,name,internship coordinator(Foreign key referenced to internship coordinator id)

Internship(id,company_name,date_end,date_start,internship coordinator id(Foreign key referenced to internship coordinator id),organization id(Foreign key referenced to organization id))

Student(id,email,F_name,middle_name,Last_name,internal mentor id (Foreign key referenced to internal person id),external mentor(Foreign key referenced to person id),Internal id(Foreign key referenced to internship id),registeration_date)

Person(person_id,name)

Phone_num(phone_num,person_id(Foreign key referenced to person(person id)))

Internal mentor(office_location,availability,specification,person id(Foreign key referenced to person(person id)))

External mentor(Linked_in profile,work_email,position, person id(Foreign key referenced to person(person id))

Report(report_id,report_data,grade,sid(Foreign key referenced to student_id),internship_id(Foreign key referenced to internship_id))

Generate report(report_id(Foreign key referenced to report_id),sid(Foreign key referenced to student_id),internship_id(Foreign key referenced to internship_id)

Phase 4:

New Queries:

1. What is the internship with the highest grade achieved among students?

2-Which internal mentor (doctor) that all high-grade students choose?

3-names of external mentor who is students have not succussed?

4-What is the internship duration in ITI?

5- what is the completion rate in ITI?

6- How many students have not passed?

7-names of students that have not registered in any internship?

Relation algebra:

1) $\Pi_{\text{Company name}}$ (internship $\bowtie\sigma_{\text{grade>=90}}$ (report))

2) Π_{name}(person $\bowtie$(internal mentor $\bowtie$ $\sigma_{\text{grade>=90}}$(report))

3) Π_{Name}(person $\bowtie$ (external mentor $\bowtie$(student $\sigma_{\text{grade <=50}}$ (report))))

4) Π($\sigma_{\text{Name = ITI}}$ (internship))

5) Count ($\sigma_{\text{Name = ITI}}$ (internship $\bowtie\sigma_{\text{grade >=50}}$ (report))

6) Count ($\sigma_{\text{Grade <=50}}$ (report))

7) Π_{names}(student – (student $\bowtie$ internship))

Phase 5:

Student view is to present a comprehensive overview of each student including their personal details internship information and mentor specialization **User outputs:**

Student id , full name , email , grades , internship Company , internal mentor specialization, external mentor position

Coordinator View is to present information about internship coordinators the organizations they are associated with and the students under their supervision

User outputs

Coordinator name , organization name , student first name , student last name , report grade

Admin View provides detailed information for administrative staff to manage student data including their contacts grades mentorship and internship details

User outputs

Student id , student name , student email , student grades , internal mentor specialization , external mentor details , internship company , internship start date , internship end date , organization name , report id , report grade

Stored Procedure counts the number of students who have received a failing grade (F) in their reports

Function : it returns a single value indicating the total count of students with a grade of 'F' the logic is to make it easy to obtain this information without having to rewrite the SQL query every time

Consistency for the Project

To ensure the integrity and functionality of the Field Training 2 system, all components including views stored procedures and queries must adhere to the established schema relationships and business requirements This consistency is vital for maintaining accurate data and reliable operations throughout the system

Schema

Table Structures: All views and stored procedures must correlate with the defined tables and their attributes. For instance, the StudentView should accurately reflect the columns from the Student, Internship, Internal_mentor, and External_mentor tables.
Data Types: Ensure that the data types used in views and stored procedures are consistent with those defined in the schema (e.g., VARCHAR, INT, DATE).

Relationships

Foreign Key Constraints: Views and stored procedures must respect the foreign key relationships established in the schema. For example, any references to external_mentor_id in Student must ensure that the corresponding entries exist in the External_mentor table.

Join Operations: When combining data from multiple tables, views and stored procedures should utilize appropriate join types (INNER JOIN, LEFT JOIN) based on the relationships defined in the schema.

Business Requirements

Functional Requirements: All views and stored procedures must satisfy the functional requirements of the system. For example:
StudentView must include fields necessary for students to review their progress, such as grades and mentor specifications.
CoordinatorView should provide insights into student performance to help coordinators manage internships effectively.
Performance Requirements: Ensure that queries in views and stored procedures are optimized for performance, especially when dealing with larger datasets. Use indexing and efficient join strategies to minimize execution time.

Testing and Validation

Unit Testing: Each view and stored procedure should undergo testing to validate that it returns the expected results based on the current database state. **Data Consistency Checks**: Regularly perform checks to ensure that data integrity is maintained across all related tables. For instance, ensure that all students linked to internships have corresponding entries in the Report table.

By adhering to these consistency guidelines, the Field Training 2 project will maintain a robust framework capable of supporting its intended functionalities. This will ensure that users can rely on the system for accurate information and efficient management of internships, mentors, and evaluations. Consistent implementation will also facilitate future enhancements and scalability of the system.

Written by:

Youssif Yasser said