

Homework 3

Database System for Industrial Internships

Introduction.

In this homework assignment, the primary goal is to develop a user-friendly application that interacts with my database system (DBS). To begin, I will define the mission and objectives of my DBS, determining its purpose and how it aligns with the requirements of my stakeholders.

By leveraging the results and findings from my previous homework assignments, I will adjust and refine them to meet the current requirements of the project. This ensures that my application effectively addresses the needs of the stakeholders involved.

One crucial aspect of this assignment is the normalization of my data. Drawing from the techniques and knowledge gained from seminars I will apply normalization principles to organize my data and eliminate redundancies. This step is vital for maintaining data integrity and optimizing the efficiency of your DBS.

Ultimately, the objective is to develop an application that provides a seamless and comfortable user experience when working with my database. This application will enable stakeholders to interact with the DBS, perform various operations, retrieve relevant information, and manipulate data efficiently. By creating a user-friendly interface and incorporating functionalities tailored to the requirements of my stakeholders, I aim to enhance productivity and facilitate a smooth workflow within my industrial internships domain.

1.1 Specify the Mission & Objectives for your DBS

1. Why required?

A DBS is essential for the engineering internships domain to efficiently manage and store large amounts of data related to internships, such as student information, internship availability, company details, and project information. The DBS can help to streamline the internship application and matching process, ensuring that students are matched with appropriate internships based on their skills and interests.

2. Which part of the business is covered?

The DBS covers the internship management and matching process, which is a critical part of the business for engineering schools, students, and companies offering internships. It can also help in tracking student progress and evaluating the effectiveness of the internship program.

3. Required for whom? What stakeholders (user groups) are involved?

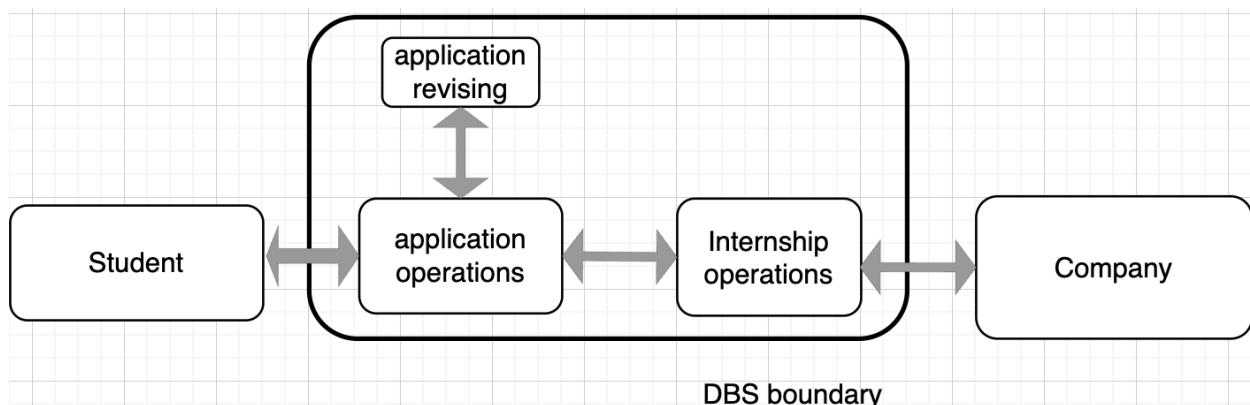
The stakeholders involved in the engineering internship domain are students, and companies offering internships. The DBS is required by these stakeholders to efficiently manage and track internships, streamline the application process, and match students with appropriate internships.

4. Is this a development of a fully new or refinement of an existing DBS?

DBS would be a development of a fully new system.

5. How can efficiency and effectiveness be measured?

Efficiency and effectiveness can be measured by tracking metrics such as the time taken to match students with internships, the number of successful matches, the number of internships offered, the satisfaction levels of students and companies with the process, and the number of repeat participants in the program. Additionally, feedback from stakeholders can be collected to identify areas for improvement and measure the overall success of the DBS in meeting the needs of the engineering internship domain.



1. Application operations

- Stakeholders: Students, Companies offering internships
- Tasks:
 - Maintaining application
 - Track the status of applications

2. Application revising

- Stakeholders: companies offering internships
- Tasks:
 - give feedback to applications

3. Internship operations

- Stakeholders: companies offering internships, students
- Tasks:
 - Post internship opportunities
 - Make contacts with successful candidates

Mission objectives

Unit: application operations

Task: Maintaining application

Stakeholders:

- Students
- Companies offering internships

Operations:

- Create application
- Delete applications

Task: Track the status of applications

Stakeholders:

- Students

Operations:

- Check for updates for application status
- Participate in interview if required

Unit: Application revising

Task: give feedback to applications

Stakeholders:

- Companies offering internships

Operations:

- Check for applications
- Review info about candidate
- Give feedback

Unit: Internship operations

Task: Post internship opportunities

Stakeholders:

- Students
- Companies offering internships

Operations:

- Create internships
- Post internship

Task: Make contacts with successful candidates

Stakeholders:

- Students
- Companies offering internships

Operations:

- Check for successful candidates
- Contact them for future internships

1.2 Elicit Requirements for Your DBS.

Collect

– What information to be collected?

1. Application operations Unit:

- Received Data:
 - Internship applications submitted by students
- Own Data:
 - Info about student
- Produced Data:
 - Application status updates for students and companies
- Within a Task:
 - Reviewing and managing internship applications
- Projected on Stakeholders:
 - Students, companies offering internships
- Data:
 - Document: Internship application, application status updates
 - Attribute: Student name, internship, application status, student's information

2. Application revising Unit:

- Received Data:
 - Internship applications submitted by students
- Own Data:
 - student's motivation, education information
- Produced Data:
 - feedback to applications
- Within a Task:
 - revising application
- Projected on Stakeholders:
 - companies offering internships, students
- Data:
 - Document: feedback
 - Attribute: further information if candidate was accepted, reasons why he was not accepted

3. Internship operations Unit:

- Received Data:
 - Internship opportunities posted by companies
 - Internship applications submitted by students
- Own Data:
 - Internship opportunity information (position, location, duration, etc.)
- Produced Data:
 - Internship opportunity listings for students
- Within a Task:
 - Posting internship opportunities
 - Sending applications to companies
- Projected on Stakeholders:
 - companies offering internships, students
- Data:
 - Document: Internship opportunity listing, internship application
 - Attribute: Company name, position, location, duration, application status

– How would I collect it?

- Collect all relevant documents within the scope of the project, such as application forms, internship descriptions, and feedback forms.
- Analyze, document, and validate the information contained in these documents. This could involve identifying key data points, such as internship duration, location, and requirements, and ensuring that they are accurately represented in my database.
- If I find that there is not enough information available in these documents, conduct additional research to fill any gaps in my knowledge. This could involve searching online resources, such as job boards and career websites, or reaching out to individuals involved in the internship process, such as company representatives or students who have completed internships in the past.
- Check how different companies and engineering schools handle internships, either by talking to individuals involved in the process or by researching publicly available information. This could help me identify best practices and common challenges in the industry.
- Observe actual internship operations by participating in the process as a viewer. This can help me gain a firsthand understanding of how the process works and identify any potential issues or areas for improvement.
- Analyze, document, and validate the information gathered through my examination, research, and observation. This could involve comparing my findings to existing data points in my database and identifying any discrepancies or gaps in my knowledge.
- If necessary, iterate on the process by repeating any of the previous steps to ensure that my database contains accurate, reliable, and up-to-date information that meets the needs of all stakeholders involved in the internship process.

Analyze

– What are the foci of analysis?

Table of operations:

Operation	Stakeholder	Actor	Type	Frequency	Input data				Output data			
					Document/Q-ry	Data object /Q-ty	Field/ Q-ty	Access	Document/Q-ry	Data object /Q-t y	Field/ Q-ty	Access
Create application	student	appliers	interactive	As needed	application/1	application/1	Motivation_Ietter, projects	W	Application_id/1	-	-	R
Delete application	student	appliers	interactive	As needed	application/1	application/1	id	W	-	-	-	-
Check for updates for application status	student	appliers	batch	As needed	-	-	-	-	feedback/1	feedback/1	-	R
Check for applications	company	revisor	batch	As needed	-	-	-	-	Application_id/1	id/1	-	R
Review application	company	revisor	interactive	As needed	application/1	application/1	Motivation_Ietter, projects	R	feedback/1	feedback/1	-	W
Create internships	company	hr	interactive	As needed	internship/1	internship/1	End, start	R	internship/1	internship/1	-	W

							, title, description					
Check for successful candidates	company	hr	batch	As needed	feedback/1	feedback/1	Motivation letter, projects	R	Student id/1	login /1		R

Table of data:

Stakeholder	Data object					Data Property						
	Name(ID)	Involv ed in operat ion	Input Docume nt	Outpu t docu ment	Est. No Instances	Name(id)	Datatype, Format	Operat ion	Rule	Est. No Instances		
student	application	Create applic ation	Application record		10-1000 per internship	Student id	Varchar(20)					1 per doc
						Internship id	INT					1 per doc
						Motivation letter	Varchar(1000)					1 per doc
						projects	Varchat(150)					1 per doc
company	internshi[Create intern ship	Internsh ip record		As needed	title	Varchar(30)					1 per doc
						description	Varchar(1000)					1 per doc
						start	Date					1 per doc

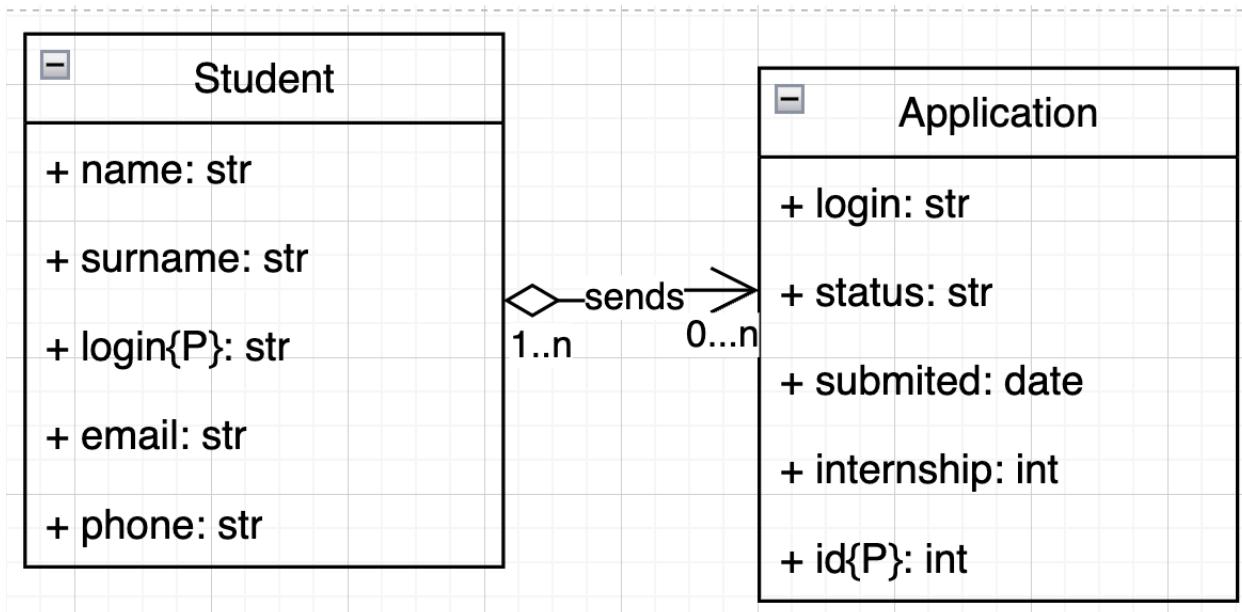
1.3 Develop External Views as UML Class Diagrams.

Stakeholders:

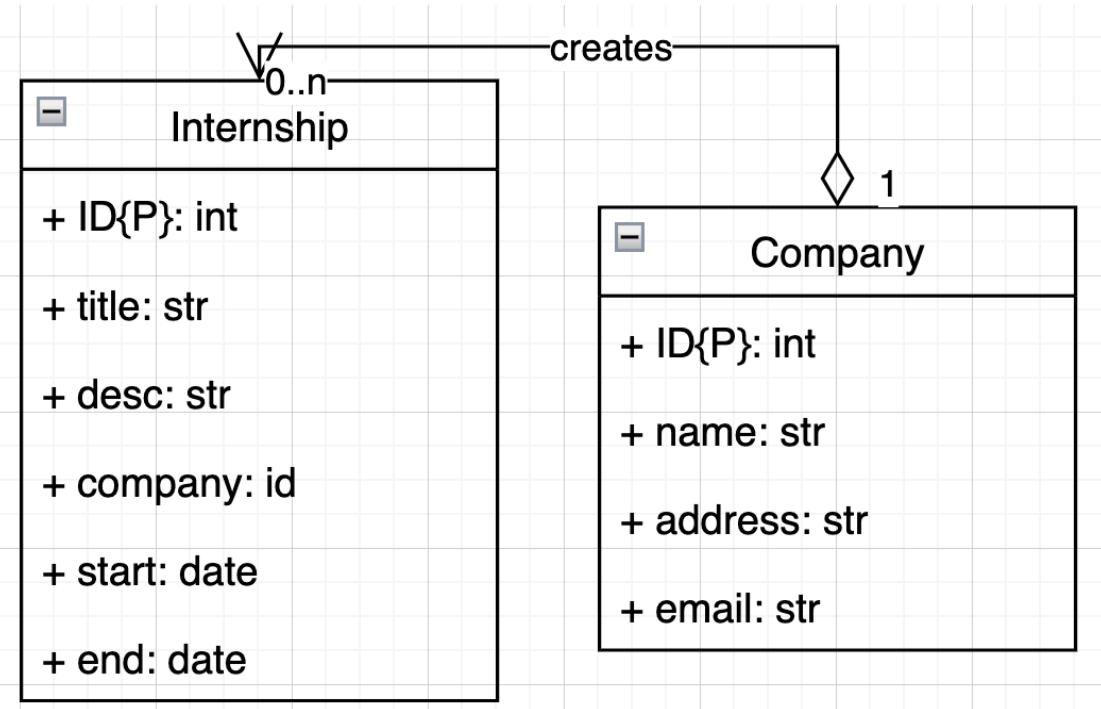
Student

Company

Industrial internships: External Views: Student

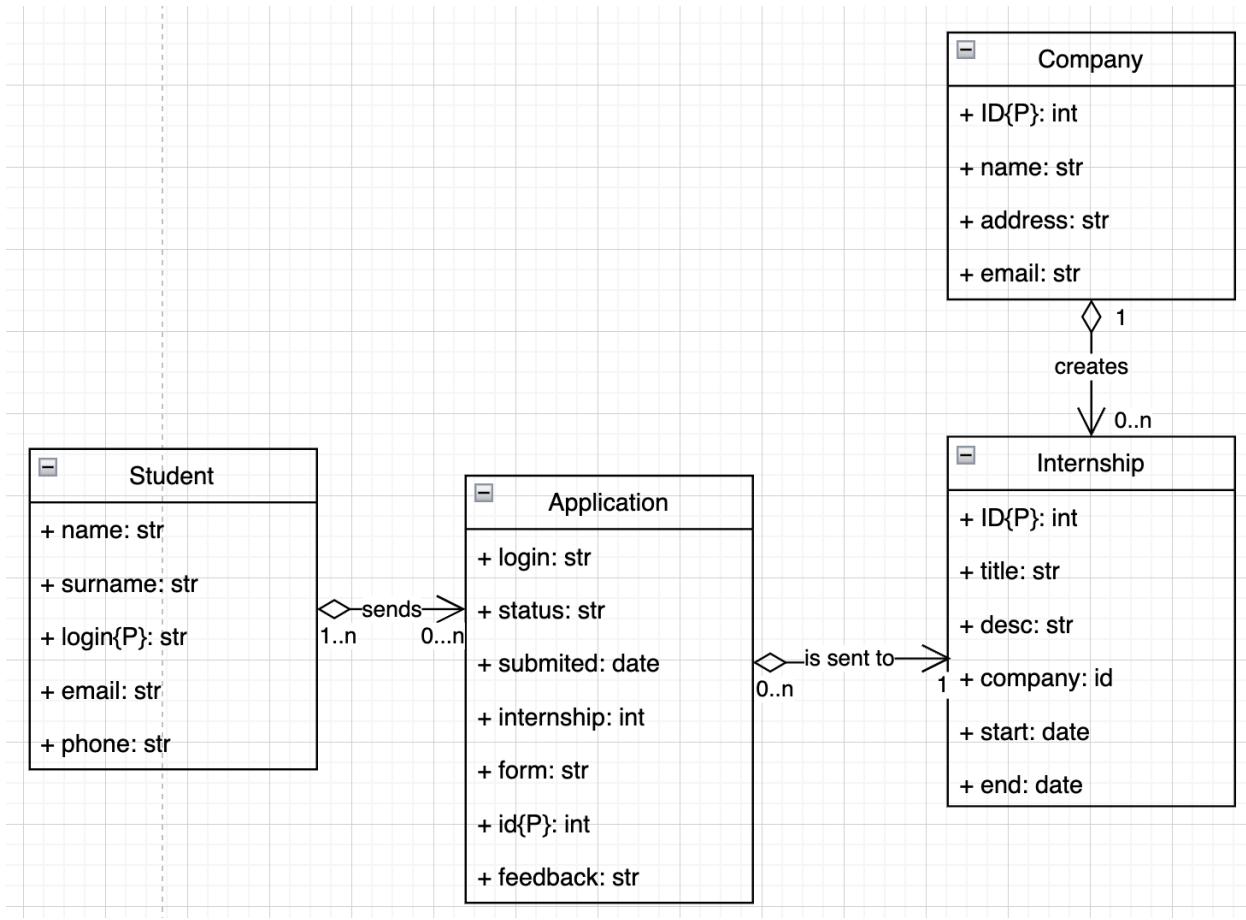


Industrial internships: External Views: Company



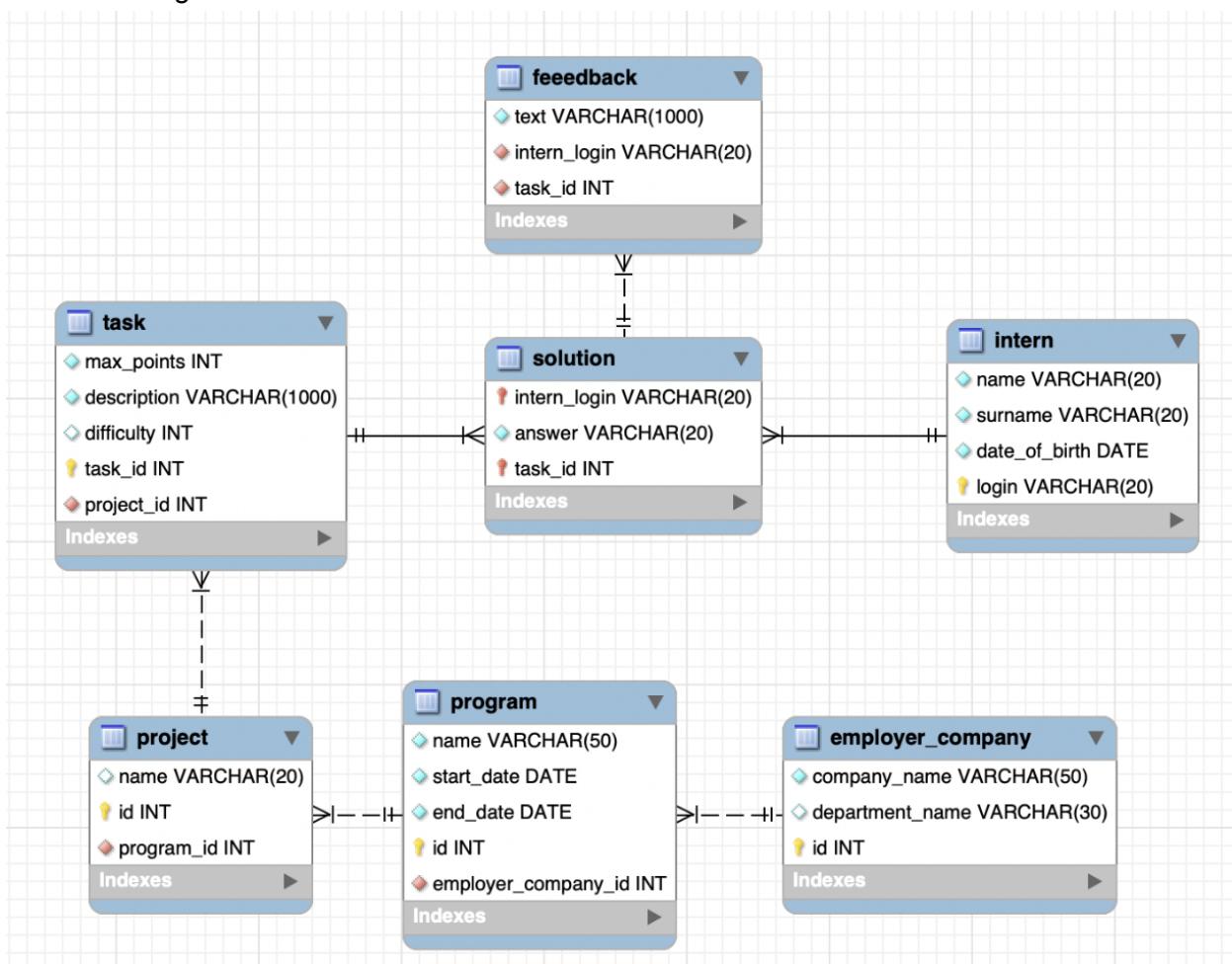
1.4 Refine Your Conceptual Model (HW1) Based on the External Views

I decided to remove the evaluation table, because I moved the responsibility of evaluation to the company (they will receive applications and send feedbacks). Also I removed the project and task table (student will need only send an application statement). As a result, I got that diagram:

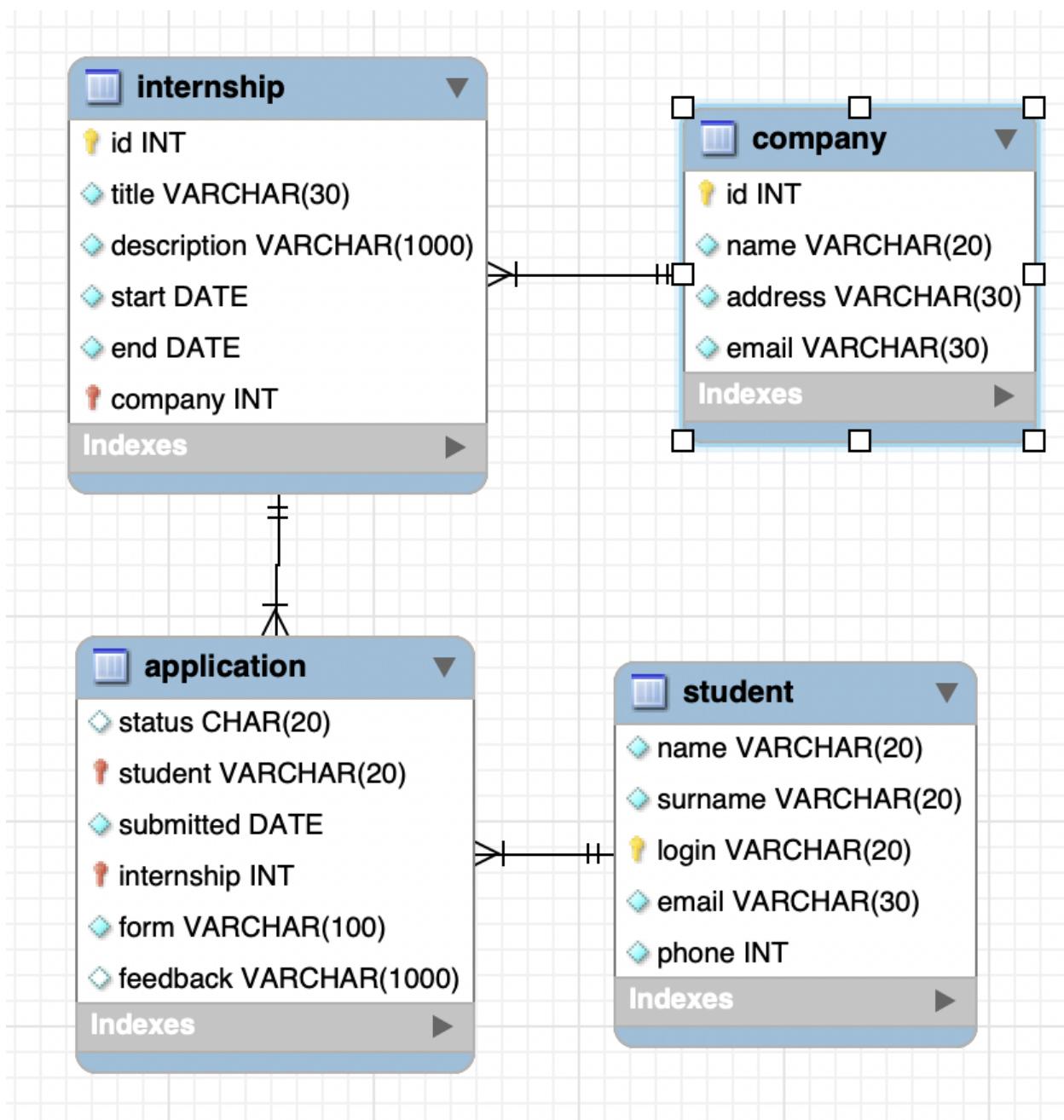


1.5 Refine your Logical Model and Database (HW2)

Before refining:



After refining:



In refinement I decided to delete project and task tables(their responsibility was moved to the application table). I slightly changed the logic of my dbs(now it will be a mediator between company and student, it will not be responsible for grading student's applications. The duty of grading was moved to company). Also I changed columns of almost all tables. Now each table will have only necessary information and will make it easier to create communication channels between students and companies. All my improvements greatly simplified the tasks of the database and gave it clear objectives.

DB Schema modification queries:

Application:

```
CREATE TABLE `application` (
  `status` char(20) DEFAULT NULL,
  `student` varchar(20) NOT NULL,
  `submitted` date NOT NULL,
  `form` varchar(1000) NOT NULL,
  `internship` int NOT NULL,
  PRIMARY KEY (`student`, `internship`),
  KEY `fk_application_internship1_idx` (`internship`),
  CONSTRAINT `fk_application_internship1` FOREIGN KEY (`internship`) REFERENCES `internship` (`id`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `fk_application_student` FOREIGN KEY (`student`) REFERENCES `student` (`login`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

Company:

```
CREATE TABLE `company` (
  `id` int NOT NULL,
  `name` varchar(20) NOT NULL,
  `address` varchar(30) NOT NULL,
  `email` varchar(30) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `name_UNIQUE` (`name`),
  UNIQUE KEY `email_UNIQUE` (`email`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

Internships:

```
CREATE TABLE `internship` (
  `id` int NOT NULL AUTO_INCREMENT,
  `title` varchar(30) NOT NULL,
  `description` varchar(1000) NOT NULL,
  `start` date NOT NULL,
  `end` date NOT NULL,
  `company` int NOT NULL,
  PRIMARY KEY (`id`, `company`),
  UNIQUE KEY `id_UNIQUE` (`id`),
  KEY `fk_internship_company1_idx` (`company`),
  CONSTRAINT `fk_internship_company1` FOREIGN KEY (`company`) REFERENCES `company` (`id`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

Student:

```
CREATE TABLE `student` (
    `name` varchar(20) NOT NULL,
    `surname` varchar(20) NOT NULL,
    `login` varchar(20) NOT NULL,
    `email` varchar(30) NOT NULL,
    `phone` int NOT NULL,
    PRIMARY KEY (`login`),
    UNIQUE KEY `login_UNIQUE` (`login`),
    UNIQUE KEY `email_UNIQUE` (`email`),
    UNIQUE KEY `phone_UNIQUE` (`phone`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

I decided to populate my tables after 1.6 step, because normalization can change some of the table's columns or even add new helping tables.

1.6 Do Normalization to 3NF.

Normalization to 1NF:

Student:

Name - the "name" column represents the student's first_name, it is atomic.

Surname - the "surname" column represents the student's last name, it is atomic.

Login - "login" column represents a unique identifier for each student, it is atomic.

Email - the "email" column represents the student's email address, it is atomic.

Phone - the "phone" column represents the student's phone number. Phone numbers often consist of multiple components such as the country code, area code, and local number. But this information is not necessary for our dbs, so we may assume that the phone number is atomic.

Application:

Status - the "status" column represents the status of the application (e.g., pending, accepted, rejected), it is atomic.

Student - the "student" column represents a unique identifier or login for each student, it is atomic.

Submitted - the "submitted" column represents the date the application was submitted, it is atomic.

Form - the "form" column represents the application form submitted by the student for the internship, it is not atomic. The form may contain multiple fields such as personal information, education details,, motivation letter, and done projects. To achieve atomicity, the "form" column should be decomposed into separate columns representing each field.

Internship - the "internship" column represents a unique identifier for each internship, it is atomic.

Feedback:

Student - the "student" column represents a unique identifier or login for each student, it is atomic.

Internship - the "internship" column represents a unique identifier for each internship, it is atomic.

Feedback - the "feedback" column represents the feedback given to a student for a specific internship, it is atomic.

Internship:

ID - the "ID" column represents a unique identifier for each internship, it is atomic.

Title - the "title" column represents the title or name of the internship, it is atomic.

Description - the "description" column represents a description of the internship, it is atomic.

Start - the "start" column represents the start date of the internship, it is atomic.

End - the "end" column represents the end date of the internship, it is atomic.

Company - the "company" column represents a unique identifier for the company offering the internship, it is atomic.

Company:

ID - the "ID" column represents a unique identifier for each company, it is atomic.

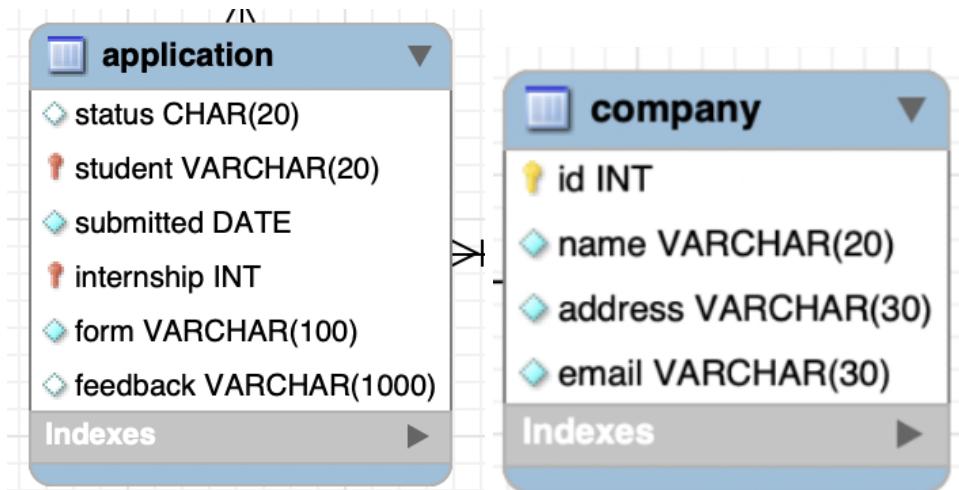
Name - the "name" column represents the name of the company, it is atomic.

Address - the "address" column represents the address of the company, it consists of town and address in that town. To achieve atomicity, the "address" column should be decomposed into separate columns representing each field.

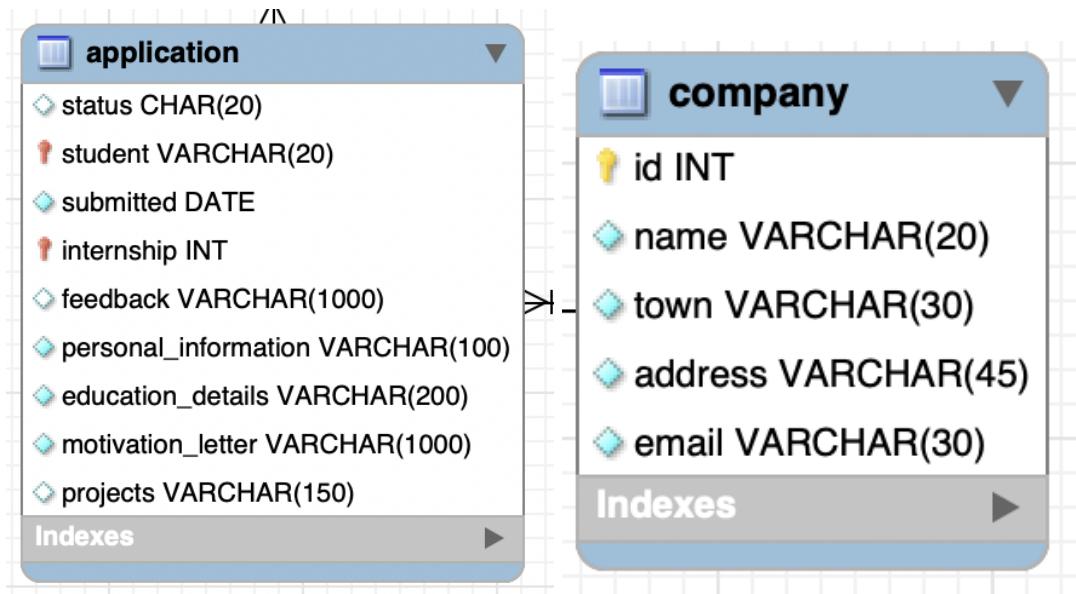
Email - the "email" column represents the email address of the company, it is atomic.

My tables violated some 1NF principles(form is not atomic), so I changed this table:

Before:



After:



Normalization to 2NF:

Student:

- Prime attributes: login, Non-prime attributes: name, surname, email, phone
- Functional dependencies: {name, surname, email, phone} \rightarrow {login}
- Table is already in 2NF

Application:

- Prime attributes: student, internship Non-prime attributes: status, submitted, personal_information, education_details, motivation_letter, projects, feedback
- Functional dependencies: {personal_information, education_details} \rightarrow {student}
- Refinement: move personal_information, education_details to student table
- Refined tables are in 2NF
- Information was not lost because – All the combinations of student and internship persist – New combinations do not appear

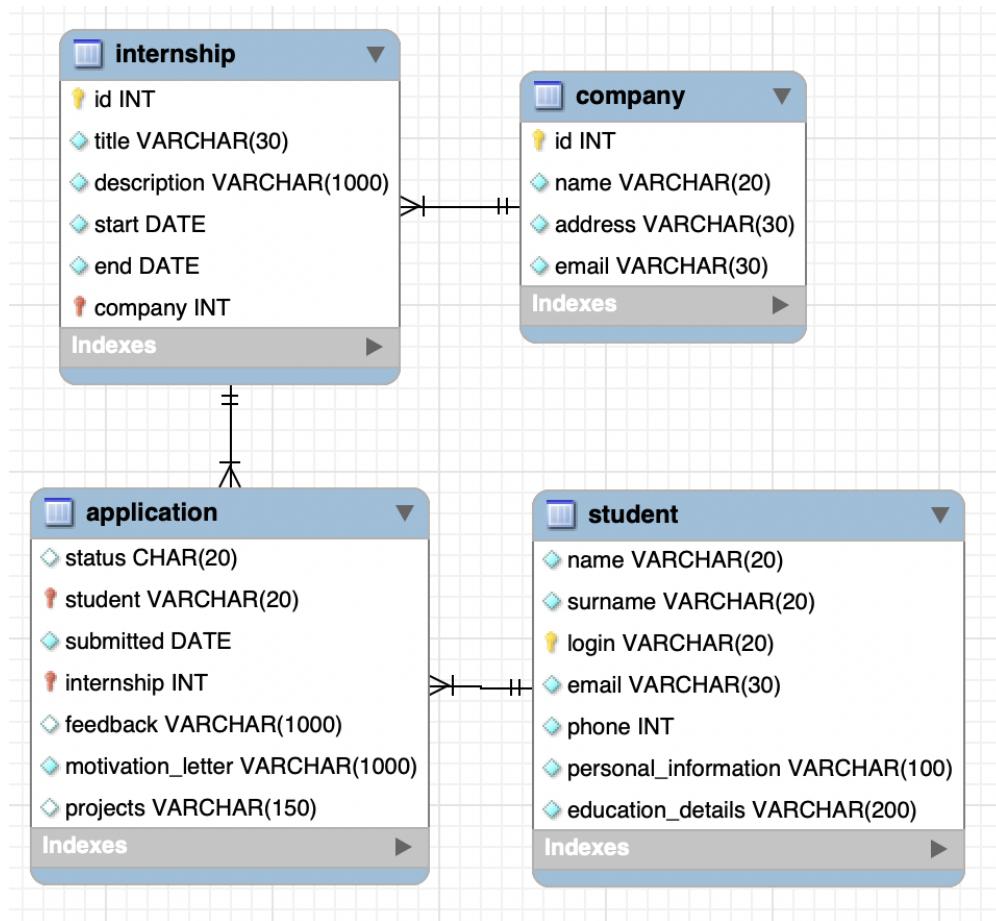
Internship:

- Prime attributes: id Non-prime attributes: title, description, start, end, company
- Functional dependencies: { title, description, start, end, company }
- } \rightarrow {id}
- Table is already in 2NF

Company:

- Prime attributes: id Non-prime attributes: name, address, email
- Functional dependencies: {name, address, email} \rightarrow {id}
- Table is already in 2NF

After refinement tables look like this:



Normalization to 3NF:

Student:

- Prime attributes: login, Non-prime attributes: name, surname, email, phone
- Functional dependencies: {name, surname, email, phone, personal_information, education_details} \rightarrow {login}
- Table is already in 3NF

Application:

- Prime attributes: student, internship Non-prime attributes: status, submitted, motivation_letter, projects, feedback
- Functional dependencies: $\{status\} \rightarrow \{feedback\}$, $\{submitted, motivation_letter, projects\} \rightarrow \{student, internship\}$
- Refinement: remove transitive dependency by:
Make feedback_id new foreign key
Move feedback and status to new table feedback
- Refined tables are in 3NF
- Information was not lost because – All the combinations of student and internship persist – New combinations do not appear

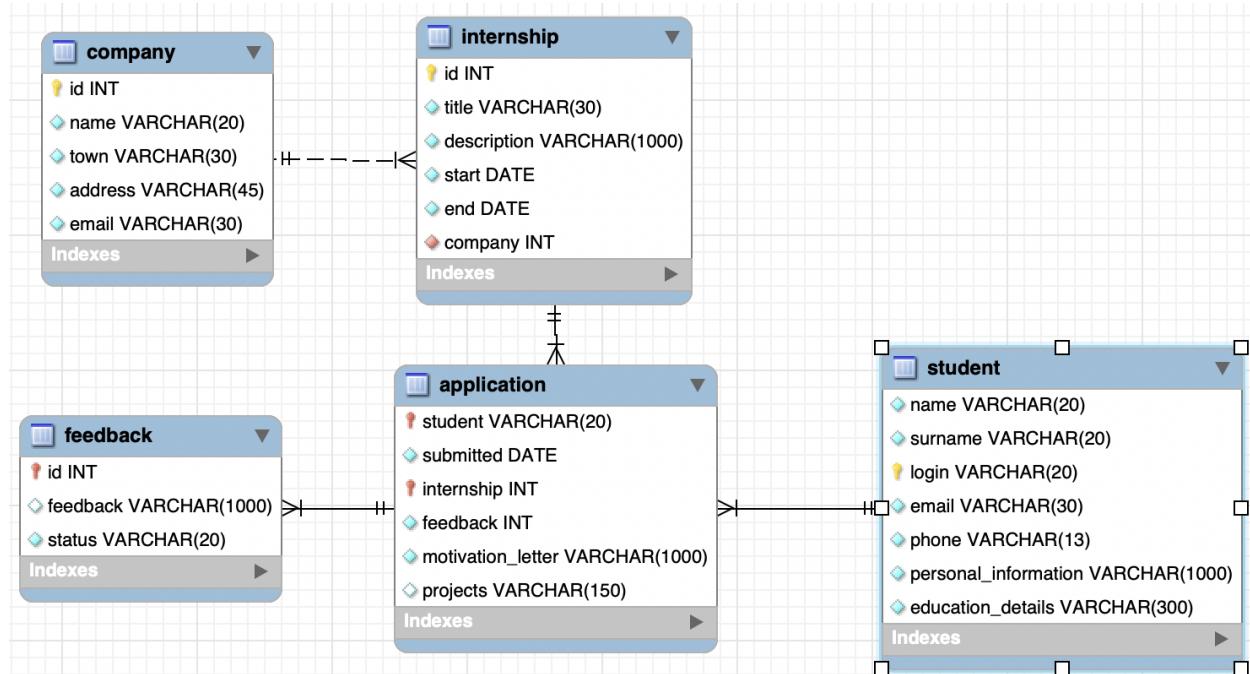
Internship:

- Prime attributes: id Non-prime attributes: title, description, start, end, company
- Functional dependencies: $\{title, description, start, end, company\} \rightarrow \{id\}$
- Table is already in 3NF

Company:

- Prime attributes: id Non-prime attributes: name, address, email
- Functional dependencies: $\{name, address, email\} \rightarrow \{id\}$
- Table is already in 3NF

Now my tables look like this:



After that I started populating my database.

Company:

id	name	town	address	email
1	ABC Corporation	New York City	123 Main St	info@abccorp....
2	XYZ Industries	Los Angeles	456 Elm St	info@xyzindus...
3	Global Enterpri...	London	789 Oxford Rd	info@globale.c...
4	Tech Solutions	San Francisco	987 Market St	info@techsolut...
NULL	NULL	NULL	NULL	NULL

Internship:

id	title	description	start	end	company
1	Software Devel...	Assist in devel...	2023-06-01	2023-08-31	1
2	Marketing Intern	Contribute to...	2023-07-01	2023-09-30	2
3	Data Science I...	Work on data a...	2023-06-15	2023-09-15	3
4	Business Devel...	Assist in identi...	2023-07-15	2023-10-15	4

Student:

name	surname	login	email	phone	personal_inform...	education_details
Alex	Johnson	ajohnson	ajohnson@example.com	5551234567	I love photographing...	Bachelor of Arts in...
Emily	Brown	ebrown	ebrown@example.com	4445556666	I enjoy playing...	Bachelor of English
John	Doe	jdoe	jdoe@example.com	1234567890	I enjoy playing...	Bachelor of Science
Jane	Smith	jsmith	jsmith@example.com	9876543210	I am passionate...	Master of Business

Application:

student	submitted	internship	feedback	motivation_letter	projects
ajohnson	2023-05-10	3	9	I am passionate...	Analyzed customer...
ebrown	2023-05-10	4	10	As an aspiring...	Assisted in design...
jdoe	2023-05-10	1	7	I am highly mot...	Developed a web...
jdoe	2023-05-10	2	11	Having gained...	Developed a mobile...
jsmith	2023-05-10	2	8	As a marketing...	Managed social media...
jsmith	2023-05-10	3	12	I have a strong...	Implemented product...

Feedback:

id	feedback	status
7	The candidate...	accepted
8	The candidate...	rejected
9	The candidate...	accepted
10	The candidate...	pending
11	The candidate...	accepted
12	The candidate...	rejected

1.7 Develop a Software App (3-5 Forms) for Manipulating.

I developed a software app with following functional:

Student:

- Viewing all student
- Creating new student
- Modifying existing ones
- Deleting student

All forms looks like this:

View students	add student	Delete student	modify student
	<input name="name" type="text"/> <input name="surname" type="text"/> <input name="login" type="text"/> <input name="email" type="text"/> Phone Number: <input name="phone" type="text"/> personal information <input name="info" type="text"/> education <input name="education" type="text"/>	<input name="login" type="text"/> delete student	<input name="name" type="text"/> <input name="surname" type="text"/> <input name="login" type="text"/> <input name="email" type="text"/> Phone Number: <input name="phone" type="text"/> personal information <input name="info" type="text"/> education modify student

Same functional for company:

[View companies](#)

add company

name

town

address

email

[create company](#)

Delete company

id

[delete company](#)

modify company

id

name

town

address

email

[modify company](#)

Also I can add and delete internships:

[View internships](#)

add internship

title

description

start
 dd.mm.yyyy [calendar](#)

end
 dd.mm.yyyy [calendar](#)

company

[create internship](#)

Delete internship

id

[delete internship](#)

And apply and delete applications for internships:

[View applications](#)

add application

student

submitted
 dd.mm.yyyy [calendar](#)

internship

motivation_letter

projects

[apply](#)

Delete application

student

internship

[delete application](#)

1.8 Develop 3 Reports and Respective Queries. Include in the Software App.

Add query

Creating a student query:

```
@app.route('/add-student', methods=['POST'])
def add_student():
    student_data = request.form

    # Extract the student details from the request data
    name = student_data.get('name')
    surname = student_data.get('surname')
    login = student_data.get('login')
    email = student_data.get('email')
    phone = student_data.get('phoneNumber')
    personal_information = student_data.get('personal_information')
    education_details = student_data.get('education_details')

    response = execute_query(conn, f"INSERT INTO student (name, surname, login, email, phone, \n
    personal_information, education_details) VALUES ('{name}', '{surname}', '{login}', '{email}', \n
    '{phone}', '{personal_information}', '{education_details}')")

    if response is not True:
        print(response)
        return render_template("with_output.html", error=response)

    return redirect("/")
```

Example of using:

add student

name

surname

login

email

Phone Number:

personal information

education

Table before:

	name	surname	login	email	phone	personal_information	education_details
0	Alex	Johnson	ajohnson	ajohnson@example.com	5551234567	I love photography and capturing special moments.	Bachelor of Arts in English Literature, QWE College
1	Emily	Brown	ebrown	ebrown@example.com	4445556666	I enjoy playing musical instruments and composing music	Bachelor of Engineering in Civil Engineering, ZXC University
2	John	Doe	jdoe	jdoe@example.com	1234567890	I enjoy playing soccer and reading books in my free time.	Bachelor of Science in Computer Science, XYZ University
3	Jane	Smith	jsmith	jsmith@example.com	9876543210	I am passionate about traveling and exploring new cultures.	Master of Business Administration, ABC University

Table after:

	name	surname	login	email	phone	personal_information	education_details
0	Alex	Johnson	ajohnson	ajohnson@example.com	5551234567	I love photography and capturing special moments.	Bachelor of Arts in English Literature, QWE College
1	Emily	Brown	ebrown	ebrown@example.com	4445556666	I enjoy playing musical instruments and composing music.	Bachelor of Engineering in Civil Engineering, ZXC University
2	John	Doe	jdoe	jdoe@example.com	1234567890	I enjoy playing soccer and reading books in my free time.	Bachelor of Science in Computer Science, XYZ University
3	Jane	Smith	jsmith	jsmith@example.com	9876543210	I am passionate about traveling and exploring new cultures.	Master of Business Administration, ABC University
4	Newname	newsurname	new	new@gmail	1322424232	New added student	UCU

Modify query

```
@app.route('/modify-company', methods=['POST'])
def modify_company():
    company_data = request.form
    id = company_data.get("id")
    name = company_data.get('name')
    town = company_data.get('town')
    address = company_data.get('address')
    email = company_data.get('email')
    response = execute_query(conn, f"UPDATE company SET name = '{name}', town = '{town}', address = '{address}', email = '{email}' WHERE id = '{id}';")
    if response is not True:
        print(response)
        return render_template("with_output.html", error=response)

    return redirect("/")
```

Lets modify company's entry:

For that I have to know it's id

Table before:

id	name	town	address	email
0	ABC Corporation	New York City	123 Main St	info@abccorp.com
1	XYZ Industries	Los Angeles	456 Elm St	info@xyzindustries.com
2	Global Enterprises	London	789 Oxford Rd	info@globale.com
3	Tech Solutions	San Francisco	987 Market St	info@techsolutions.com

Filled form:

modify company

id

name

town

address

email

modify company

Result:

	id	name	town	address	email
0	1	ABC Corporation	New York City	123 Main St	info@abccorp.com
1	2	XYZ Industries	Los Angeles	456 Elm St	info@xyzindustries.com
2	3	Global Enterprises	London	789 Oxford Rd	info@globale.com
3	4	modifiedname	modifiedtown	modifiedaddress	modifiedemail

Delete query

```
@app.route('/delete-internship', methods=['POST'])
def del_internship():
    internship_data = request.form

    id = internship_data.get('id')
    response = execute_query(conn, f"DELETE FROM internship WHERE id = '{id}'")
    if response is not True:
        print(response)
        return render_template("with_output.html", error=response)

    return redirect("/")
```

Let's delete some internship

Internship's table

	id	title	description	start	end	company
0	1	Software Development Intern	Assist in developing software applications	2023-06-01	2023-08-31	1
1	2	Marketing Intern	Contribute to marketing campaigns and strategies	2023-07-01	2023-09-30	2
2	3	Data Science Intern	Work on data analysis and modeling projects	2023-06-15	2023-09-15	3
3	4	Business Development Intern	Assist in identifying new business opportunities	2023-07-15	2023-10-15	4

Application's table:

	student	submitted	internship	feedback		motivation_letter	projects
0	ajohnson	2023-05-10	3	9	I am passionate about data science and its potential to drive informed decision-making. With my strong statistical background and proficiency in programming languages like Python, I am confident in my ability to extract meaningful insights from complex datasets during this internship.	Analyzed customer behavior data and identified key factors influencing purchase decisions.	
1	jdoe	2023-05-10	1	7	I am highly motivated to gain experience in software development and contribute to innovative projects. My previous coding experience and problem-solving skills make me a strong candidate for this internship.	Developed a web application using HTML, CSS, and JavaScript.	
2	jdoe	2023-05-10	2	11	Having gained experience in both software development and marketing, I am excited about the prospect of combining these skills to create innovative marketing solutions. With my technical expertise and creative mindset, I believe I can make a valuable contribution to marketing campaigns during this internship.	Developed a mobile application for tracking and analyzing marketing data.	
3	jsmith	2023-05-10	2	8	As a marketing enthusiast, I am eager to learn and apply new strategies to drive business growth. With my creative thinking and strong analytical skills, I believe I can make a significant impact on marketing campaigns during this internship.	Managed social media accounts and increased follower engagement by 30%.	
4	jsmith	2023-05-10	3	12	I have a strong passion for data science and its potential to drive business growth. With my solid foundation in statistical analysis and machine learning, I am eager to apply my skills to solve complex business problems during this internship.	Implemented predictive models to optimize inventory management and reduce costs.	

Our form:

Delete internship

id

2

delete internship

After executing:

Internship's table:

	id	title	description	start	end	company
0	1	Software Development Intern	Assist in developing software applications	2023-06-01	2023-08-31	1
1	3	Data Science Intern	Work on data analysis and modeling projects	2023-06-15	2023-09-15	3
2	4	Business Development Intern	Assist in identifying new business opportunities	2023-07-15	2023-10-15	4

Application's table:

	student	submitted	internship	feedback		motivation_letter	projects
0	ajohnson	2023-05-10	3	9	I am passionate about data science and its potential to drive informed decision-making. With my strong statistical background and proficiency in programming languages like Python, I am confident in my ability to extract meaningful insights from complex datasets during this internship.	Analyzed customer behavior data and identified key factors influencing purchase decisions.	
1	jdoe	2023-05-10	1	7	I am highly motivated to gain experience in software development and contribute to innovative projects. My previous coding experience and problem-solving skills make me a strong candidate for this internship.	Developed a web application using HTML, CSS, and JavaScript.	
2	jsmith	2023-05-10	3	12	I have a strong passion for data science and its potential to drive business growth. With my solid foundation in statistical analysis and machine learning, I am eager to apply my skills to solve complex business problems during this internship.	Implemented predictive models to optimize inventory management and reduce costs.	

As we can see our form successfully deleted entries in both tables, which it has to do, because those tables have on delete cascade property.

1.9 Test Your DBS.

Let's test our previous queries:

Add query:

Trying to add new student with duplicated login:

add student

name

surname

login

email

Phone Number:

personal information

education

create student

Result:

1062 (23000): Duplicate entry 'jdoe' for key 'student.PRIMARY'

If we try to manipulate some entry with not existing id, we will not get an exception as DBS just will not find that entry and will not execute that command

If we try to apply from not existing student then we get error like that:

add application

student

submitted

internship

motivation_letter

projects

apply

1452 (23000): Cannot add or update a child row: a foreign key constraint fails ('industrial_internships3`.`application`, CONSTRAINT `fk_application_student` FOREIGN KEY (`student`) REFERENCES `student` (`login`) ON DELETE CASCADE ON UPDATE CASCADE)

Summary.

I started by specifying the mission and objectives for my database system (DBS), outlining the goals and purpose of your app in facilitating industrial internships. After that I conducted a thorough requirements elicitation process, gathering the necessary information and understanding the needs of both students and industries involved in the internship process. Created external Views as UML Class Diagrams representing the relationships and structures of the entities involved in the internship domain.

Also I refined the conceptual model based on the external views, made adjustments and improvements to ensure it accurately represents the relationships and entities in my industrial internship domain. Using the refined conceptual model, I further refined my logical model and database design. This step involved organizing the data and establishing the necessary tables, fields, and relationships for efficient data management.

After those steps I've done normalization to 3NF to ensure data integrity and eliminate redundancies. This process involved structuring the data in a way that minimizes duplication and improves efficiency.

As a result I developed a software application with many forms, creating a user-friendly interface for manipulating data within my DBS. These forms likely include functionalities such as student registration, internship creating, application submission, and industry partner interactions.

By following these steps, I have created an industrial internships app with a robust database system that effectively connects students and industries, streamlines the internship process, and provides valuable insights for stakeholders involved.