UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Foundations of Databases A.Y. 2021-2022

Master Degree in
ICT for Internet and Multimedia

Homework 2 – Conceptual and Logical Design

Group Name: RD
Project: non profit organizations

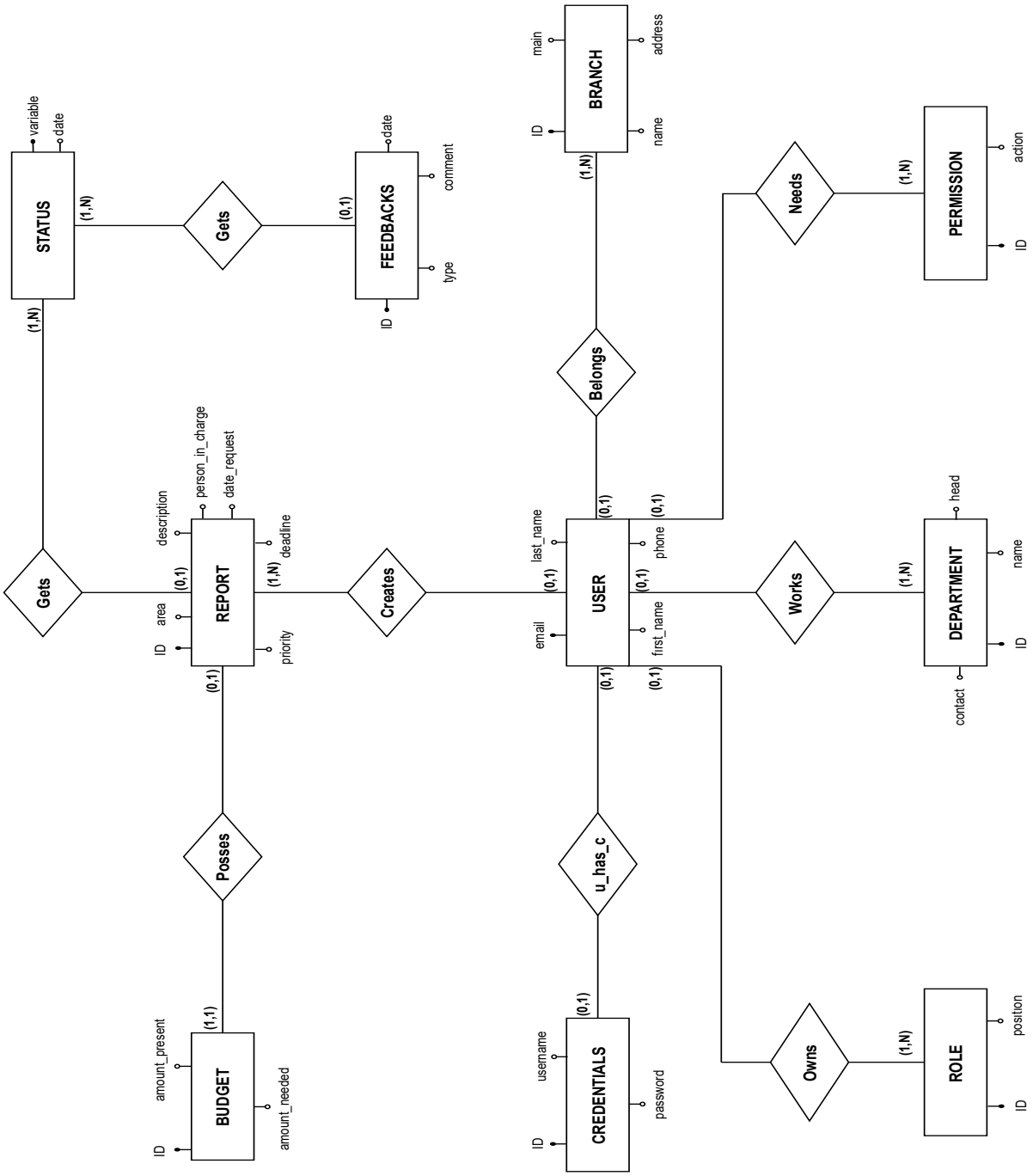
Last Name	*First Name*	*Student Num*
ATES	HAKAN	2049550
CICEK	KADER	2049526
LOKO	ISIDORUS MAU	2040498
MURDAYAN	YEKATERINA	2049529
OZTURK	MEHMET	2049527
PAJAJ	XHEINA	2055575
PRAVISHI	GAURI	2041448
SHOFI	ALFI BAQIATUS	2041377
ZARE	MOHAMMAD MEHDI	2041379
ZIU	VANESA	2050797

Conceptual Design

Variations to the Requirement Analysis

There are no relevant variations to Requirement Analysis

Entity-Relationship Schema



Data Dictionary

Entities Table

Entity	Description	Attributes	Identifier
Users	Data of the employees that access the system.	• First Name(User's first name) • Last name (User's last name) • Email-ID (User's Email Address) • Phone (User's phone number) • number_of_reports (calculating the reports users have created)	Email
Branch	The employees of the branch offices who will insert the request in the system.	• ID(branch identifier) • main_or_branch (Boolean value which indicates whether the employees belong to the main office or branch office) • Name (If the employees belong to the branch office, this field will hold the location of it else it will hold the value, "Main Office") • Address: (this field will hold The branch's exact location)	ID
Credentials	User's data to enter the system.	• ID(Credential identifier) • username(user's name shown on the system) • password (user's pass key to enter the system securely)	ID
Role	Position that have a certain set of responsibilities.	• position_ID(Position Identifier) • Position (User's Position inside a department)	position_ID

Department	An administrative division of the organisation.	• department_ID (Department Identifier) • Name(the specific department e.g. Finance, Logistics, HR, etc.) • head_of_department(The person leading the work in the specific department) • Contact_Number(The phone number to reach the department)	department_ID
Permissions	The type of rights that define the access the users will have on the system	• permissions_ID (Permissions identifier) • Actions (all the types of interactions allowed with the system for a user role.	permissions_ID
Report	The physical document that describes the details of the project.	• report_ID(Report identifier) • Area (used to show the region from which the request comes from) • Person_in_charge (used to store information about the person to be contacted for further information.) • Date of Request (used to specify the date the request was received on) • Priority (the interger value representing the priority of the project, maximum 10, decreasing with time) • Deadline (shows the date by which the project must be implemented) • Description (Specific explanation about the report)	report_ID
Status	The current state of the project.	• Status_variable (current status of the report showing if the project is approved, rejected or pending) • Date_of_Status (Used to used to show the last time, the status was updated)	Date_of_Status

Feedback	The description showing the reason why the actual status is chosen or not.	• feedback_ID(Feedback Identifier) • Type (based on the status updated by the responsible office this is a drop-down menu with 2 options positive or negative) • Comment (optional field representing the reason why a specific status was updated) • Date_of_Feedback (used to show the time the comment was added)	feedback_ID
Budget	The amount of money needed to develop the project.	• budget_ID(Budget Identifier) • Amount_present (the amount of money the branch office can contribute for the project) • Amount_needed (The amount of money the branch office needs from the main office to develop the project).	budget_ID

Relationship Table

Relationship	**Description**	**Component Entities**
u_has_c	Associates each user to it's credentials.	• User(0,1) • Credentials(0,1)
owns	Associates each user to the role they own in the organization	• User(0,1) • Role(1,N)
belongs	Relates the employees to the branch they belong to.	• User(0,1) • Branch(1,N)
works	Relates each user to the department they work to.	• User(0,1) • Department(1,N)
needs	Defines the permission the user needs to access the system.	• User(0,1) • Permission(1,N)
creates	Shows the user that has created the report	• User(0,1) • Report(1,N)

gets_commented	Shows the type of feedback that has been returned for each status.	• Status(1,N) • Feedback(0,1)
possess	Shows the amount of budget a report needs to be developed	• Report(0,1) • Budget(1,1)
gets	Relates each report to the status it gets.	• Report(0,1) • Status(1,N)

External Constraints

- The feedback written by the responsible employee from the main office, generalate team should be addressed to the writer. So, in any case the corresponding employee can be contacted to give further details.
- In addition to the status features, there should be included a status history so that in some unexpected situations the responsible employees can better understand the issue.
- If an employee is working at two different positions at the same time, he can not be defined on both of them and having the corresponding permissions.
- There should be some more information about how the â€œhaveâ€ part of the budget is collected.

Functional Requirements Satisfaction Check

The main function of the system is to hold all the information for the particular project.

- **Authentication of users and their roles**.
  Data is stored in User, Credentials and Roles. The list of the information contains feature such as email, username, password and role.
- **The data of the request**.
  The exact time in the date-time format that the request was submitted. The information of the user who made the request.
  Data of the request is stored in Report entity.
- **The project ID, a unique ID associated with each project.**
  The area where the request comes from. The detailed information regarding the country, the city and if possible, the zip code.
  The project ID it's a serial number identifying each project and is stored in the report entity.
- **The branch information.**
  The branch location,The branch manager,The person in-charge for the specific project and his contact details.
  The branch information is stored in the Branch Entity.
- **Project Information.**
  The deadline of the project, specifying the latest day of starting the project implementation. The type of the project, specifying what the project is dealing with.The description of the project which gives a brief explanation about the background of the problem that needs to be solved.
  The project information is stored in the Report entity.

- **The budget needed by a particular project to be implemented. Here they will inform about the part of the funding that they can afford (if they can) and the part of the funding that they need from the organisation.**
  The data related to the budget is stored in Budget entity.
- **The target community that will benefit from the project.**
- **The duration of the project. This tells about the total time a project needs to be fully developed.**

The next function of this system is to manage the login operations and to provide different interfaces as per their specific roles in the organisation.

- Information about user's roles are stored in the **Role** entity defining the position each user has in the organisation. Depending on the role and also the type of the permission the users can access the system performing different and various tasks.
- The permissions are stored in the **Permission** entity

Logical Design

Transformation of the Entity-Relationship Schema

Redundancy Analysis

In the user entity, it is defined an attribute called number_of_reports, which will represent the number of reports added by a particular user. This will be the only cyclic feature since the schema does not contain any cycle of entities.

Analysis of Data Load

The load analysis is calculated in two ways, one of them keeping the number_of_reports attribute and the other one removing it from the User entity. The idea of this is to define the most efficient way.

Data Volume(example)

Concept	Construct	Volume
Creates	Relationship	4000
Report	Entity	4000
User	Entity	20

We conclude that for 4000 reports created the access number of ‘creates’ relationship is: 4000 : 20 = 200

Example with redundancy

O1

Concept	Conctruct	Access	Type	Average Access
Creates	Relationship	1	W	1x20x2=40
Report	Entity	1	W	1x20x2=40
User	Entity	1	R	1x20x1=20
User	Entity	1	W	1x20x2=40

Total Access: 140

O2

Concept	Conctruct	Access	Type	Average Access
User	Entity	1	R	1x2=2

Total Access: 2

Example without redundancy

O1

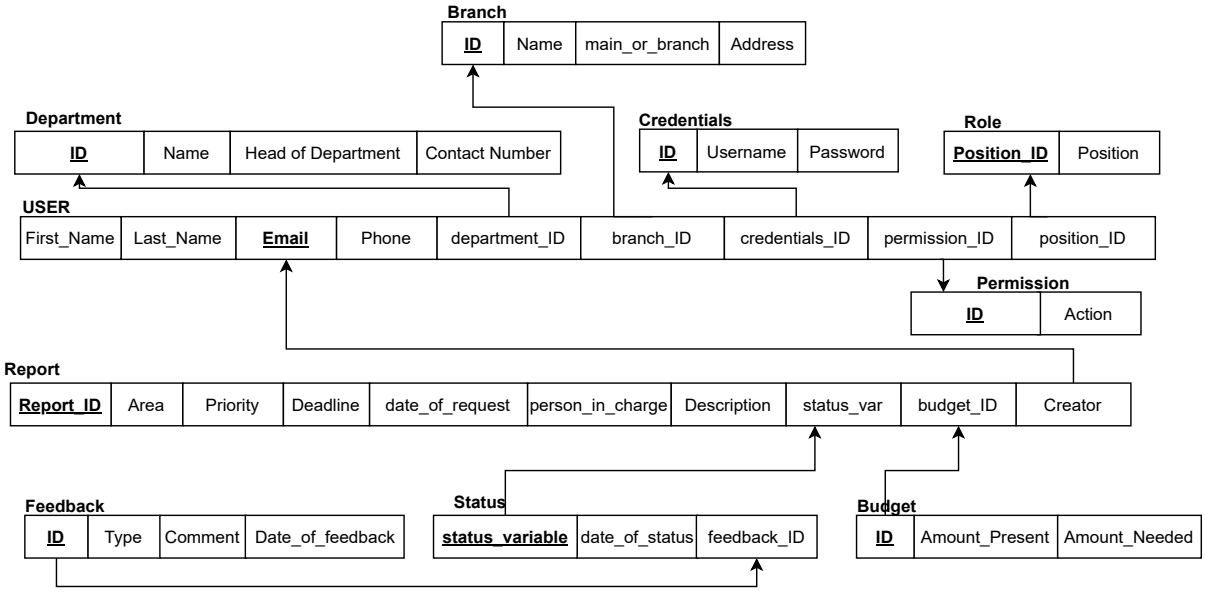
Concept	Conctruct	Access	Type	Average Access
Creates	Relationship	1	W	1x20x2=40
Report	Entity	1	W	1x20x2=40

Total Access: 80

O2

Concept	Conctruct	Access	Type	Average Access
User	Entity	1	R	1x2=2

Relational Schema



Our schema contaınts the logıcal connectıon between entıtıes. We have transformed the relationsips straightforwardly because we have chosen to directly connect each entıty wıth dırect relationships. We have added ID serial numbers as Primary Keys to identify them from each other. Also other attributes are defined as Not NULL because the entities are well separated and we don't have optional requirements, that require at least an attribute to be NULL.

Data Dictionary

Relation	Attribute	Description	Domain	Constraints
User	Email	User's Email-ID	Text	Primary key
	First_name	User's Name	Text	Not Null
	Last_name	User's Surname	Text	Not Null
	Phone	User's Phone Number	Text	Not Null
	Department_ID	ID's department	Serial	Not Null
	Branch_ID	Branch Identifier	Serial	Not Null
	Credentials_ID	Credential identifier	Serial	Not Null
	Permission_ID	Permissions identifier	Serial	Not Null
	Position_ID	Position Identifier	Serial	Not Null
Branch	ID	Branch Identifier	Serial	Primary key
	main_or_branch	Boolean Value to represent if the employee belongs to main office or branch office	Text	Not Null
	Name	If the Employee belongs to main office This would indicate "main Office" else it will hold the name of the branch they belong to.	Text	Not Null
	Address	It holds the exact location of the branch	Text	Not Null
Credentials	ID	Credential identifier	Serial	Primary key
	Username	User's name shown on the system	Text	Not Null

	Password	User's Passkey to enter the system	Text	Not Null
Role	Position_ID	Position Identifier	Serial	Primary Key
	Position	User's Position for the role	Text	Not Null
Department	ID	Department Identifier	Serial	Primary key
	Name	The specific department e.g.: HR, Finance, Logistics, etc.	Text	Not Null
	Head of Department	The person leading the work for the specific department	Text	Not Null
	Contact Number	Phone Number to reach the Department	Text	Not Null
Permissions	ID	Permissions identifier	Serial	Primary key
	Action	All types of interactions allowed for a user role	Text	Not Null
Report	Report_ID	Report identifier	Serial	Primary key
	Area	Used to show the region where the concerned request comes from.	Text	Not Null
	Person in Charge	Used to store the information about the person who prepared the report and the one to be contacted for further information	Text	Not Null
	Date of Request	Used to specify the date the request was received	Integer	Not Null
	Priority	Integer value representing the priority of the project ranging from 1-10 and decreasing with time	Integer	Not Null
	Deadline	Shows the deadline by which the project should be implemented	Integer	Not Null
	Description	Specific explanations about the report	Text	Not Null
	Creator	Creator of the report	Text	Not Null
	Status_var	Current Status of the report showing if the project is approved or rejected or pending	Text	Not Null
	Budget_ID	Budget Identifier	Serial	Not Null
Status	status_variable	Current Status of the report showing if the project is approved or rejected or pending	Text	Primary key
	date_of_status	The date of latest status update	Text	Not Null
	Feedback_ID	Feedback Identifier	Serial	Not Null
Feedback	ID	Feedback Identifier	Serial	Primary Key
	Type	Based on the status updated by the responsible office this is a drop-down menu with 2 options i.e. positive or negative	Text	Not Null
	Comment	Optional field representing the reason for the specific status update	Text	Not Null
	Date_of_feedback	(used to show the time the comment was added)	Integer	Not Null
Budget	ID	Budget Identifier	Serial	Primary key

	Amount_present	The amount of money the branch can contribute for the project	Integer	Not Null
	Amount_needed	The amount of money the branch office needs from the main office to develop the project	Integer	Not Null

External Constraints

- The feedback written by the responsible employee from the main office, generelate team should be addressed to the correct User. So, in any case the corresponding employee can be contacted to give further details and the feedback_type cannot be NULL.
- In addition to the status features which should not be NULL, there should be included a status history so that in some unexpected situations the responsible USERS can better understand the issue.
- If a USER is working at two different ROLES at the same time(to define them we will have an attribute called position_ID, which is gonna be the PRIMARY KEY), he can not be defined on both of them and having the corresponding permissions.
- There should be some more information about how the "HAVE" (integer value) attribute of the BUDGET is collected.