**Database assignments solution report**

* **Student id: 23110354**
* **Student name: Imad Eddin**
* **Section number: 1**

Table of Contents

[Technical Documentation 2](#_Toc185192513)

[1 Introduction 2](#_Toc185192514)

[2 Database Requirement 2](#_Toc185192515)

[2.1 User and System Requirement 2](#_Toc185192516)

[2.2 Data Requirements 3](#_Toc185192517)

[3 Database Design 3](#_Toc185192518)

[3.1 Conceptual Design 3](#_Toc185192519)

[3.2 Schema and Mapping 4](#_Toc185192520)

[3.3 Logical Design 6](#_Toc185192521)

[3.4 Physical Design 7](#_Toc185192522)

[3.5 Effectiveness of the design 7](#_Toc185192523)

[4 Normalization 9](#_Toc185192524)

[4.1 1st NF 9](#_Toc185192525)

[4.2 2nd NF 9](#_Toc185192526)

[4.3 3rd NF 10](#_Toc185192527)

[5 Physical Schema 11](#_Toc185192528)

[6 Database Development 12](#_Toc185192529)

[6.1 Database Overview 12](#_Toc185192530)

[6.2 Security 16](#_Toc185192531)

[6.3 User Interface 18](#_Toc185192532)

[6.3.1 Flowchart and Data Movement Diagrams 18](#_Toc185192533)

[6.3.2 Interfaces Development 20](#_Toc185192534)

[7 Maintenance 21](#_Toc185192535)

[7.1 Database recovery & backups 21](#_Toc185192536)

[7.2 Database maintenance in general 22](#_Toc185192537)

[8 Testing 23](#_Toc185192538)

[8.1 Data Validation 23](#_Toc185192539)

[8.2 Output Validation 25](#_Toc185192540)

[8.3 Security Validation 26](#_Toc185192541)

[8.4 GUI Validation 27](#_Toc185192542)

[8.5 Assess whether meaningful data has been extracted 28](#_Toc185192543)

[8.6 Assess the effectiveness of testing 29](#_Toc185192544)

[9 Evaluation of database solution 30](#_Toc185192545)

[9.1 Effectiveness of the database solution based on user and system requirement 30](#_Toc185192546)

[9.2 Suggested improvements 31](#_Toc185192547)

[9.3 Evaluation based on improvements needed 32](#_Toc185192548)

[10 User Documentation 33](#_Toc185192549)

[10.1 System Overview 33](#_Toc185192550)

[10.2 Using the system 33](#_Toc185192551)

[10.3 Frequently asked questions 35](#_Toc185192552)

[10.4 Contact information 35](#_Toc185192553)

# Technical Documentation

# Introduction

This document will describe the data base I am working on for a EDU Youth Foundation , EDU Youth Foundation has a system that handles basic Donor tracking, but unfortunately this system lacks some information’s about the doners and it’s not really helping when it comes to communicating with doners and track their behaviour , so my job is to make an integrated database that can be handy and make communication and tracking easier for the company.

The data base I am making will contain all needed data for each doner and their behaviour , starting with the doner some data that we should really have is of course the doner name, mobile number and his career details and the main communication method will be the doner e mail and to track the communication with doners having each doner with the emails sent to him will be really handy , finally we should know each doner participation In an event and know how much money did he donate

# Database Requirement

## User and System Requirement

Firstly this system will serve each employee in the company starting from the analytics department moving to the marketing department even the heads of each part of the company can use it ,employees would be able to add doners with their details and information and the analytics and head employees will be able to access the data without editing in order to track the performance and make some analysis discussions, and there is no doubt that the doners will have access to some of the information so they can track there participations and donations and this method can bee used in order to encourage the doners to donate more , and in order to achieve those users requirements we will need some systems from users and employees with different functions , theme and limits ; the doner can access al of his data using a website or a mobile application and for the employees they can do the same thing in the same platform but would have different user names and passwords with extra layer of security

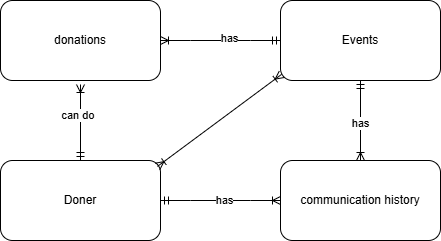
## Data Requirements

The company has many doner. Each doner has an ID, name, phone, career details, and Email Address. Each doner donate to one or multiple events, and each event gets donations from one or more doners. Each event has a name and ID, location and many donations. Each donation has an amount and a payment method a doner can make many donations for different events, and each doner will have a communication history that contains all emails sent to the doner and the time they were sent and the event that is related to this email.

# Database Design

## Conceptual Design

A conceptual design is a design that shows entities of a data base and the relations between them, describing each of them as a one to one, many to many or one to many relationships ,which can be considered as the first blueprint for the database since it helps building it and shows relations in simpler easier way.



As we can see here each doner will have multiple communication records that include the emails sent to him and each doner can do donation to one or more events, and each event will have many donations that are made for this event , and these donations are coming from doners so we can say that many doners donate to many events, when we talk about the communication history each sent email will be a about an event so we can say that each event will have many emails sent to doners to invite them, and finally doners can do donations to multiple different events so each doner can do many donations.

## Schema and Mapping

1.first form

Donor (**Doner ID**, Doner Name, Doner Phone Number, Doner Email Address, Doner career details)

Communication History (**Communication ID**, Time Stamp, Email Content)

Event (**Event ID**, Event Name, Location, Event Date)

Donation (**Donation ID**, Payment Method, Amount Donated)

2.derived attributes (No derived attributes so no change will happen.)

Donor (**Doner ID**, Doner Name, Doner Phone Number, Doner Email Address, Doner career details)

Communication History (**Communication ID**, Time Stamp, Email Content)

Event (**Event ID**, Event Name, Location, Event Date)

Donation (**Donation ID**, Payment Method, Amount Donated)

3.multivalued attributes (phone number)

(we will get the primary key from the table and the multivalued attribute and put them in one new table)

Donor (**Doner ID**, Doner Name, Doner Email Address, Doner career details)

Doner phone number (**Doner ID, Doner Phone Number**) \*

Communication History (**Communication ID**, Time Stamp, Email Content)

Event (**Event ID**, Event Name, Location, Event Date)

Donation (**Donation ID**, Payment Method, Amount Donated)

4.composite attributes (Doner Name, Location)

(any attribute that can be separated into sub information shall be mapped)

Donor (**Doner ID**, Doner First Name, Doner Second Name, Doner Email Address, Doner career details) \*

Doner phone number (**Doner ID, Doner Phone Number**)

Communication History (**Communication ID**, Time Stamp, Email Content)

Event (**Event ID**, Event Name, city, street, building, Event Date) \*

Donation (**Donation ID**, Payment Method, Amount Donated)

5.Relations (1:1) and (1:M)

Doner: Communication History (1:M)

Doner: Donations (1:M)

Event: Donations (1:M)

Event: Communication History (1:M)

(we will bring the primary key from the one side of the relation and put it in the table of the many sides of the relation)

Donor (**Doner ID**, Doner First Name, Doner Second Name, Doner Email Address, Doner career details)

Doner phone number (**Doner ID, Doner Phone Number**)

Communication History (**Communication ID**, Doner ID, Event ID, Time Stamp, Email Content) \*

Event (**Event ID**, Event Name, city, street, building, Event Date)

Donation (**Donation ID**, Event ID, Doner ID, Payment Method, Amount Donated) \*

5.Relations (M:M)

Doners: Event (M:M)

(we will have to make a new table that connects the doner id and the event id)

Donor (**Doner ID**, Doner First Name, Doner Second Name, Doner Email Address, Doner career details)

Doner phone number (**Doner ID, Doner Phone Number**)

Communication History (**Communication ID**, Doner ID, Event ID, Time Stamp, Email Content)

Event (**Event ID**, Event Name, city, street, building, Event Date)

Donation (**Donation ID**, Event ID, Doner ID, Payment Method, Amount Donated)

Participation (Doner ID, Event ID) \*

6.final schema: (weak entities written in blue)

Donor (**Doner ID**, Doner First Name, Doner Second Name, Doner Email Address, Doner career details)

Doner phone number (**Doner ID, Doner Phone Number**)

Communication History (**Communication ID**, Doner ID, Event ID, Time Stamp, Email Content)

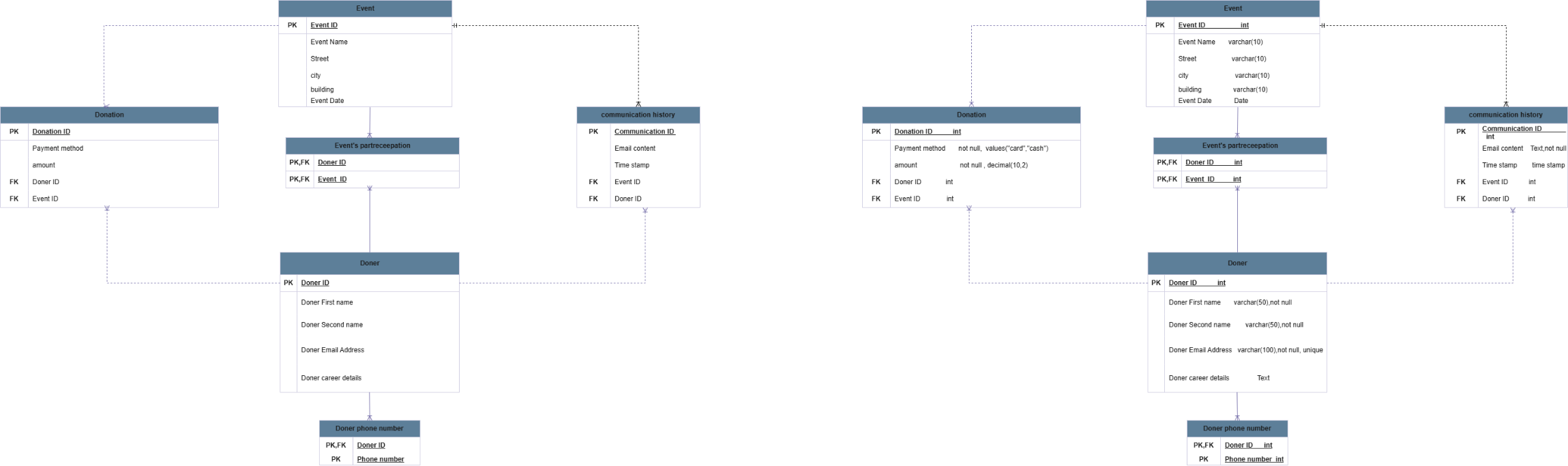
Event (**Event ID**, Event Name, city, street, building, Event Date)

Donation (**Donation ID**, Event ID, Doner ID, Payment Method, Amount Donated)

Participation (Doner ID, Event ID)

## Logical Design

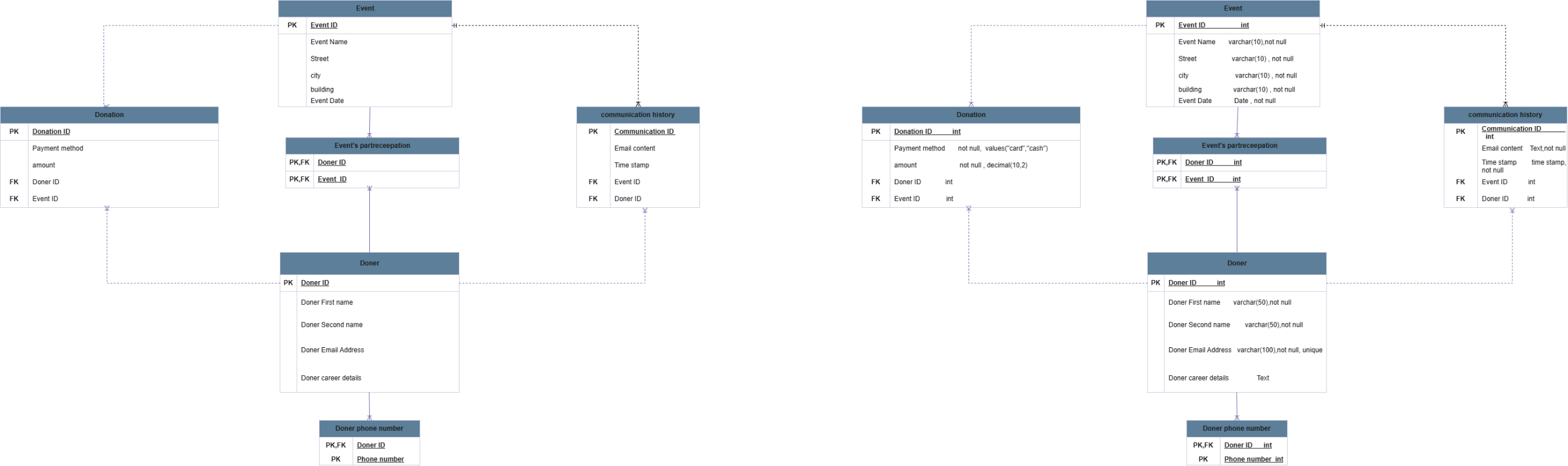
A logical model describes the data in as much detail as possible and describes the relations between the entities in a more detailed way, unlike the conceptual design it includes all the attributes of each entity , without regard to how they will be physical implemented in the database, which means that it doesn’t include the data type of each attribute , and it consists of entities and the relationships between them. And their attributes specifically and The primary and foreign keys for each entity , the logical design can be considered as the second blueprint where the data base shape is more accurate and clear and it really helps when it comes to making a physical design and the data base it self , since it can show any error between the relations and the foreign keys between entities.



Here as we can see I represented the mapping using the logical design here we can see that we have two weak entities which are the events participation and the doner phone number since they are week we represent the relation to them with a normal line which represents a strong relationship ,each relationship has come from the occurrence of a foreign key in a table , here the doner is connected to the doner phone number since it has the doner id as a primary key and it is connected to the donations table and the communication history table and to the events participation table , similarly the events table is connected to each of the events participation , communication table, and the donation table .

## Physical Design

A physical model is derived from a logical data model and it takes all technology specific details in considerations . physical model helps us to decide how the logical database design will be implemented, and it is more detailed that the logical design since it defines the suitable data types for each attribute, and it defines constraints on these tables such unique, not null and domain definitions to check if data are out of the range.



As we can see here we added the constrains of each attribute and each one fits the data usage , for example for emails we need to know the time the email was sent I and for this we need an accurate time ; so we used the time stamp and not the date , so we can include seconds and minutes not only date, same approach happens when it comes to the event data since the event date doesn’t have an exact second so using the data datatype would be more efficient .

## Effectiveness of the design

Firstly I will justify why did I select specific entities to present my system and why I chose particular attributes, starting from the Doner table , since the whole data base is about making the communication with the doners easier the Doner is something that is really important in such a data base starting with its attributes the doner id is considered as a very important attribute since it is unique for each doner it will be the key to extract any information about any doner , doner first and second name are used to make the communication in emails easier , so the name of the doner will be mentioned in the email sent to him , which is why the email address is I this table , and moving to the career details this can be used to know more about the doner and know his field so the events invitations can be more tailored , the phone number is important to contact the doner in case of any emergency or marketing for an event , now moving to the communication history table it will make tracking emails way easier using its attributes such as event and doner id , in this way we can know emails sent related to a specific event or a specific doner , the event table is considered as a storage to store all the events that were held by the company including the event locations and the name of it and the date, events participation can be used to know how many doners donated to event or how many donations s there for an event , and finally donations table ca be used to track doners behaviour over time and check the donated about from each doner or the amount donated to each event , since all of these tables are related to each other relations between them are really needed ; for example when it comer to one to many relations , its important to know that each doner can have many phoner numbers , and designing such a relation can prevent any error like adding a phone number of a non-existing doner, and such relations happen by adding the primary key of a table as a foreign key or a primary key to another table , and talking about the many to many relations I have an example for it which is the events and doners , since a doner can donate to many events and an event can have donations from multiple doners this means that there is a many to many relations and such a relation can really help with tracking doners and there participation in events and this can be done by making a table that consists of two primary keys that are foreign keys, and taking about the relations they are one of the main reason that made me create a conceptual design , because of this design mapping was easier especially when it comes to the relations steps , all the relations were revealed to me in a clear way so knowing what attributes shall go where was clear and easy , as a result creating a final schema was simpler , and as we know creating such a schema is important to create the logical and the physical design since it shows each entity with its attributes specifying them as a primary or a foreign key , and to clarify the added information to the logical design , we can say that the logical design shows the relations , entities , and attributes in a shape of tables which can be really handy when it comes to understanding the data base and its parts, and after that making the physical design involves more technical details like the data type of each attribute , for example its important to decide weather the id will be a number or numbers and letters or maybe just letters , and its important to know the type so we can decide the variable type and the data type in the data base so we can limit it to only one data type , for example the event data should be a date type to prevent any wrong data entity which means preventing any employee from entering a string for example , and this is done to make dealing with the data easier later on , normalization is important to make the database more neat and easier to deal with , starting from the first normal form , mapping the multivalve attributes by making a new table is the first step to reach a first normal form then mapping the composite values will lead to the first normal form which makes sure that there is no attribute to break down into smaller parts or no multivalued attributes can be in a table without being a primary key which avoids any misleading when it comes to reading or accessing data ; for example doners might have more than one phone number and putting them in the same doner table would make it very hard to deal with such a table , now moving to the second normal and third normal form any step in the mapping that evolved mapping the relation between tables is considered as a step to reach the third normal form , for example the event participation table is a table that came from many to many relation between the doner table and the events table .

To sum up, the end goal of every single stage in the design of the database process is to develop a system that is effective and user friendly. This is to ensure that the databases is capable of improving the desired focus of communication and interaction with the individual donor. The selection of entities and attributes, the relationships, and the application of normalization are done in such a manner that the structure of the database is going to avert errors, improve data accuracy, and ease future maintenance. Every phase of the database process starting from the conceptual model towards the logical and physical design is crucial in ensuring that the database is organized in a manner that responds to the organization’s requirements, the normalization process, ensures that redundancy is kept to a minimum level as well as ensuring that relationships between tables are defined well ,this approach makes it easier to access and manage the data within the database. All these aspects put together provide a reliable and practical system that is designed to cater to present needs as well as provide the means for easier management in the future.

# Normalization

## 1st NF

|  |  |  |  |
| --- | --- | --- | --- |
| Relations | Attributes | Violation description | Solution – Relations |
| The relations schema | The attribute name | Describe why it is not in the 1st NF (the violation) | Show the schema for each affected relation. |
| Donor (Doner ID, Doner Name, Doner Phone Number, Doner Email Address, Doner career details) | Doner Name  Phone number | The Doner name is a composite attribute, and it can be separated into multiple attributes,  And the doner phone number is a multi-valued attribute so it should have a new table | Donor (**Doner ID**, First Name, Second Name, Doner Email Address, Doner career details)  Doner Phone number (**Doner ID, Doner Phoner number**) |
| Event (Event ID, Event Name, Location, Event Date) | location | The location is a composite attribute, and it should be divided into smaller attributes | Event (**Event ID**, Event Name, city, street, building, Event Date) |

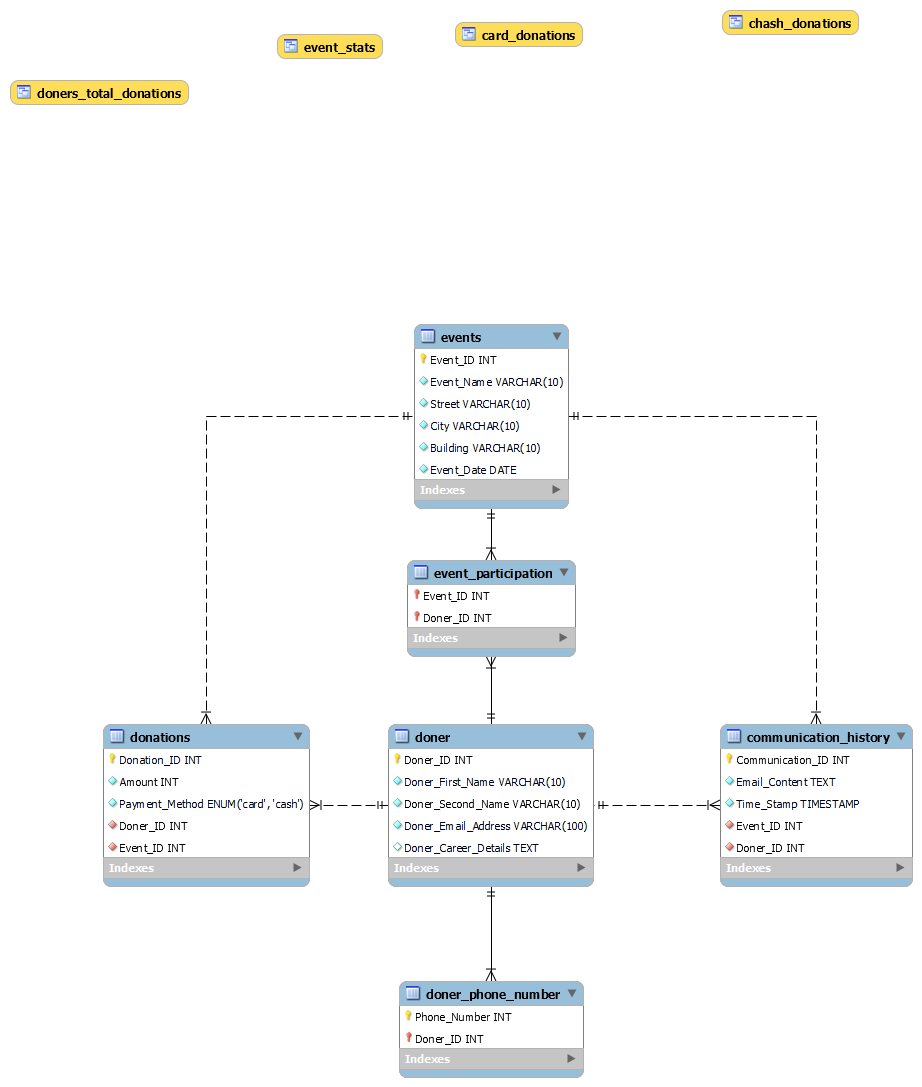
## 2nd NF

|  |  |  |  |
| --- | --- | --- | --- |
| Relations | FDs | Violation description | Solution – Relations |
| The relations schema | Show the functional dependencies causing the violation | Describe why it is not in the 2nd NF (the violation) | Show the schema for each affected relation. |
| Donor (Doner ID, Event ID, Doner First Name, Doner Second Name, Doner Email Address, Doner career details, Event Name, city, street, building, Event Date) | Doner IDà{ Doner First Name, Doner Second Name, Doner Email Address, Doner career details}  Event ID à{ Event Name, city, street, building, Event Date} | The non-prime attributes (Doner first Name, Doner second Name, Doner email address, and Doner career details) depend on a part of the key (Doner ID), and the non-prime attributes Event Name, city, street, building and Event Date depend on a part of the pk which is Event ID So we have partial dependencies | Donor (**Doner ID**, Doner First Name, Doner Second Name, Doner Email Address, Doner career details)  Event (**Event ID**, Event Name, city, street, building, Event Date)  Participation (Doner ID, Event ID) |
| Event (Event ID, Event Name, city, street, building, Event Date, Donation ID, Doner ID, Payment Method, Amount Donated) | Event ID à{ Event Name, city, street, building, Event Date}  Donation ID à{ Doner ID, Payment Method, Amount Donated} | The non-prime attributes (Event Name, city, street, building, Event Date) depend on a part of the key (Event ID), and the non-prime attributes Doner ID, Payment Method, Amount Donated depend on a part of the pk which is Donation ID So we have partial dependencies | Event (**Event ID**, Event Name, city, street, building, Event Date)  Donation (**Donation ID**, Event ID, Doner ID, Payment Method, Amount Donated) |

## 3rd NF

|  |  |  |  |
| --- | --- | --- | --- |
| Relations | FDs | Violation description | Solution – Relations |
| The relations schema | Show the functional dependencies causing the violation | Describe why it is not in the 3rd NF (the violation) | Show the schema for each affected relation. |
| Donor (Doner ID, Event ID, Doner First Name, Doner Second Name, Doner Email Address, Doner career details, Event Name, city, street, building, Event Date) | Doner IDà{ Doner First Name, Doner Second Name, Doner Email Address, Doner career details}  Doner ID àEvent ID  Event ID à{ Event Name, city, street, building, Event Date } | The PK (Doner ID) can determine any attribute in the table including Event ID. At the same time, the Event ID determine Event Name, city, street, building, Event Date As Event ID is not a candidate here, we have a transitive dependency | Donor (**Doner ID**, Doner First Name, Doner Second Name, Doner Email Address, Doner career details)  Event (**Event ID**, Event Name, city, street, building, Event Date)  Participation (Doner ID, Event ID) |
| Event (Donation ID, Event Name, city, street, building, Event Date, Donation ID, Event ID, Payment Method, Amount Donated) | Event ID à{ Event Name, city, street, building, Event Date}  Donation ID à{ Doner ID, Payment Method, Amount Donated}  Donation ID à Event ID | The PK (Donation ID) can determine any attribute in the table including Event ID. At the same time, the Event ID determine Event Name, city, street, building, Event Date and nationality. As Event ID is not a candidate here, we have a transitive dependency. | Event (**Event ID**, Event Name, city, street, building, Event Date)  Donation (**Donation ID**, Event ID, Doner ID, Payment Method, Amount Donated) |

# Physical Schema



# Database Development

## Database Overview

|  |  |  |
| --- | --- | --- |
| **Table** | **Name** | **Description** |
|  | Doner | This table is used to store all of the doners and there information such as name , email, and career details , the primary key here is the doner id and its type is an integer ; and since it’s the primary key it will be unique , and next we have the first and the last name that has a type of varchar , and the name of the doner is really important in order to contact with him so it has a constraint of not null , the email address of each user should be unique since no two doners can have the same email and each doner should put his email so the company can contact him so it has a constraint of not null, and finally we have the career details which is optional to insert and it can be help to specify invitations for specific events based on the doner field ,the doner table is related to the doner phoner number table because of a multivalued relation and its connected to the donations table since it has a one to many relation with it and its connected to the communication history due to a one to many relation too , and finally its connected to the table doners participations because of a many to many relation with the events table |
|  | Doner phone doner | The Doner Phone Number table is used to store the phone numbers of each doner, allowing for a multivalued relationship as each doner can have multiple phone numbers. This table contains the phone, which is an integer and part of the primary key, and the doner ID, which acts as a foreign key referencing the primary key of the Doner table. The relationship ensures that every phone number is linked to a valid doner while allowing the storage of multiple numbers for a single individual, and each of these values can’t be null since they are primary. |
|  | Communication history | The Communication History table is designed to record all interactions with doners, such as emails regarding events. It includes the Communication id as the primary key, which uniquely identifies each communication record. The Email Content column stores the text of the email or message, and it can’t be empty. The Timestamp records the date and time of the communication, ensuring accurate tracking of interaction history. This table also includes foreign keys: doner ID, linking to the Doner table, and event ID, linking to the Events table. |
|  | Donations | The Donations table is used to manage and track all donations made by doners to specific events. Each donation is uniquely identified by a donation ID, which is the primary key. The table includes a payment method column, defined as an ENUM type with possible values of "card" or "cash," and it can’t be empty since it is user to indicate how the payment was made. Additionally, it contains foreign keys doner ID and event ID, linking the donation to a specific doner and the event it supports, ensuring proper tracking of contributions. |
|  | Events participation | The Events Participation table captures the many-to-many relationship between doners and events, recording which doners have participated in which events. The table includes event ID and doner ID as foreign keys referencing the Events and Doner tables, respectively, and the combination of them forms the primary key. This design allows for the efficient storage of participation records, ensuring that doners' involvement in events is accurately documented. |
|  | Events | The Events table stores all the information related to events organized by the company. Each event is uniquely identified by the event ID, which serves as the primary key. The table includes details such as the event Name, which is a required, along with the Street, City, and Building where the event takes place. Additionally, the event date column records the date of the event, ensuring that all necessary event details are maintained for effective organization and tracking. |

|  |  |  |
| --- | --- | --- |
| **View** | **Name** | **Description** |
|  | **Cash donations** | This view retrieves all donations where the payment method is "cash." The aim of this view is to provide a filtered view of donations specifically made using cash as the payment method, without including any other methods of payment. The view can be used by the donation management team or the analytics team to quickly track the total donations that were made via cash. This is useful for reporting or analysis purposes where distinguishing payment methods is important, The view is implemented by selecting all records from the Donations table where the Payment Method is 'cash'. The result provides a list of all donations that were processed via cash, including the associated donation amount, donor ID, and event ID. |
|  | **card donations** | This view retrieves all donations where the payment method is "card." The objective of this view is to isolate donations made using a card for reporting or analysis purposes. This helps differentiate between the two types of payment methods (card vs. cash), allowing users to monitor trends in card donations separately from cash donations, The view is implemented by selecting all records from the Donations table where the payment method is 'card'. This enables quick retrieval of card donations, which can be used for specific analyses or reports that focus on card payments |
|  | **Event stats** | This view provides information about each event, including the event name, the number of donations associated with the event, and the total amount of money donated to each event. The aim of this view is to offer a summary of donation statistics for each event, which can help event organizers and fundraisers monitor the success and impact of their events. It brings data of both the number of donors and the financial contribution received per event, The view is implemented by joining the Events and Donations tables based on the event ID. For each event, the view calculates the total number of donations and the total amount donated by counting the donations and summing their amounts. The resulting view gives a concise summary of each event's donation performance.Bottom of Form |
|  | **Doners total donations** | This view summarizes the total donations made by each donor. It shows the donor ID, their name, and the total amount they have donated across all events. This view is useful for tracking individual donor contributions and identifying major donors. It can also be used for generating thank-you messages, reports for top donors, or analyzing the overall contribution of each donor, this view is implemented by joining the Doner and Donations tables based on the donor ID. For each donor, the total donation amount is calculated by summing the donation amounts. The resulting view displays each donor's details and their total contributions, providing a clear overview of how much each donor has contributed to various events. |

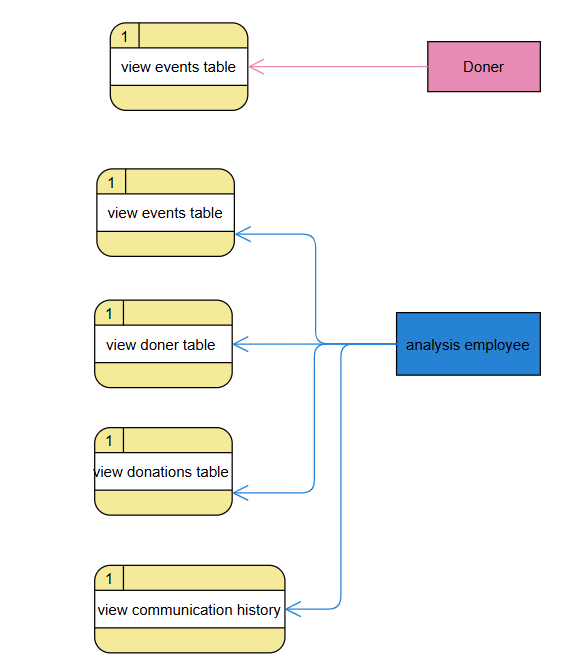
|  |  |  |
| --- | --- | --- |
| **Procedure** | **Name** | **Description** |
|  | Add Doner | This procedure adds a new donor to the Doner table. The procedure accepts the donor's first name, second name, email address, and career details as input parameters. It inserts these values into the Doner table, ensuring a smooth process for registering new donors. The procedure uses the INSERT INTO statement to add the donor details into the Doner table. |
|  | Update Donation Amount | This procedure updates the amount of a donation for a specific donation ID in the Donations table. The procedure allows for the donation amount to be modified based on a given Donation ID. The procedure uses the UPDATE statement to modify the Amount field in the Donations table where the Donation ID matches the provided input. |
|  | Doner Registration | This procedure allows for registering a new donor along with their phone number. The donor’s basic details (first name, second name, email, career details) and their phone number are passed as input parameters. The procedure first inserts the donor's information into the Doner table using an INSERT INTO statement. Then, it inserts the phone number into the Doner Phone Number table. |
|  | Donations By Event | This procedure provides a summary of donations for a specific event. It returns the total number of donors, and the total amount donated for a given event. The event ID is passed as an input parameter. The procedure uses SELECT with JOIN to connect the Donations and Events tables. It aggregates the results by counting donations and summing the donation amounts, grouping by event name. |
|  | Doner Participation Stats | This procedure retrieves the participation statistics of a specific donor across all events they have participated in. It returns the number of events the donor participated in and the total amount donated by them. The procedure joins the Doner, Event Participation, and Donations tables, grouping by the donor’s ID. It uses COUNT to find the number of events and SUM to calculate the total amount donated by the donor. |

## Security

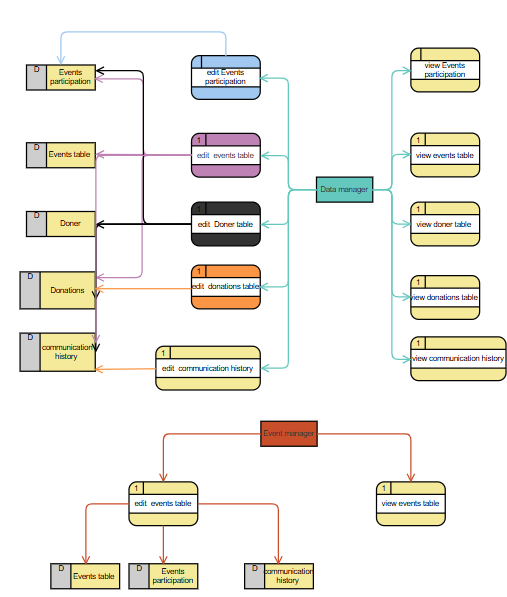
|  |  |  |  |
| --- | --- | --- | --- |
| **Username** | **Privilege Command** | **Description** | **Screenshot** |
| Doner | grant select on events to doner; | The doner can see the information about evets to see if there is any upcoming events |  |
| analysis employee | grant select on \* to analysis\_emp ; | Since the analysis department need a lot of data for each side of the company the department’s employees can see all the data of the events, donation, doners and the communication history |  |
| Events manager | grant all on events to event\_manager; | The event manager is in charge of planning events naming them and choosing their places so he can access the events table to add to it see it and edit it |  |
| Data manager | grant all on \* to Data\_manager; | The data manager is the one who is in charge of inserting data and program it in order to fill it ; for example he can make a program that inserts all the email to the communication history instantly since the data is sensitive and has a time stamp |  |

## User Interface

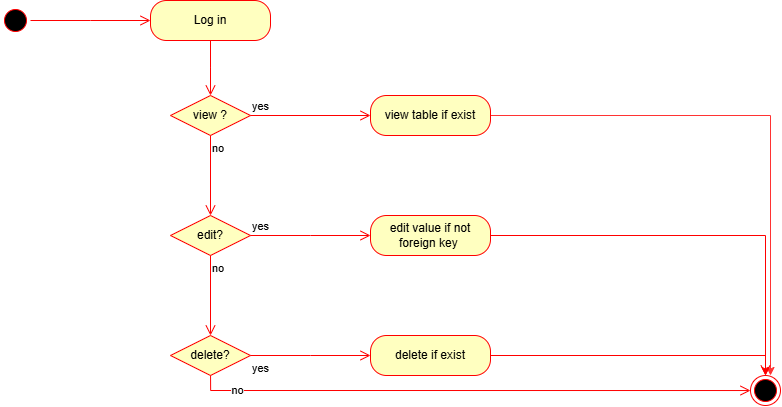
### Flowchart and Data Movement Diagrams



for the doners and analysis team they can only view data



As we can see this DFD shows each user with his privileges relating to tables that contains data each user can view or edit a table and the data in the table itself will change except for The events participation since it might change because of a change in the doner or evet id

And here is an admin flow chart to represent action the admin can make and there flow   


### Interfaces Development

|  |  |  |  |
| --- | --- | --- | --- |
| **Page ID** | **Title** | **Description** | **Screenshot** |
|  | Home page | Here we can choose the data base user we want to login as and here I have chosen to log in as an analysis employee |  |
|  | Table selection | Here we should select the tables we want to add, see or edit and only accessible tables will be shown |  |
|  | Selected tables | Here we can find the chosen tables and we can edit, add or delete rows |  |
|  | Edit table | When we edit a row, we can see the data and we can change any of the data in the row |  |
|  | Add to the table | Here I have added a new row to the donations table using the “add row” in the GUI |  |

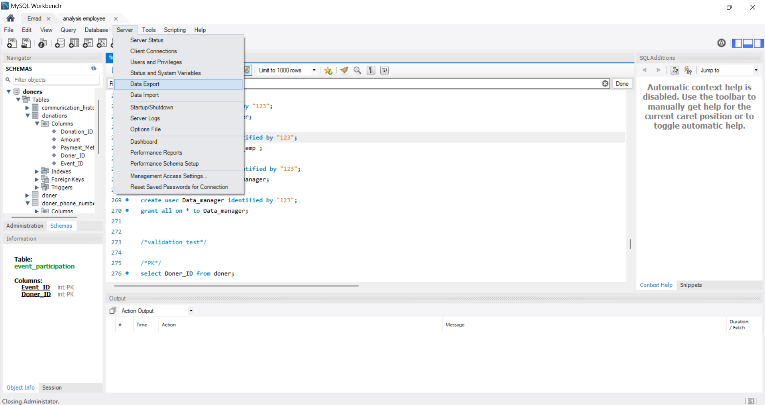
# 

# Maintenance

## Database recovery & backups

Database recovery and backup ensure data integrity by providing the same to availability and business continuity. Backups keep a safety net while restoring lost, corrupt, or deleted data by accident. Recovery thus ensures that there is minimum downtime in the event of a system failure and that operations are restarted quickly. Such regular backups also safeguard against hardware failures, software bugs, cyber-attacks, and natural disasters. Without backup and recovery measures in place, organizations risk losing their valuable data, causing financial losses, reputational damage, and operational downtimes, and here are the steps of back up using my SQL

First of all we should select export data from the server tab

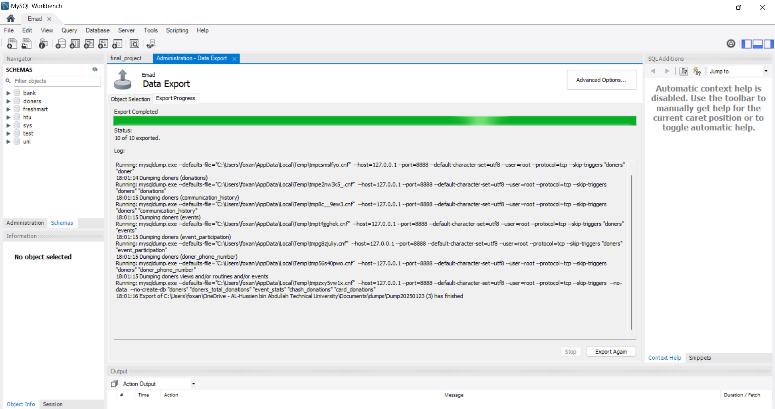


Then we should chose the data base and the tables we want to back up from it and chose the path where we want to put the back up in

A screenshot of a computer

Description automatically generated

And that it ,like this we have the data backed up



## Database maintenance in general

Most of the activities of database maintenance revolve around keeping any database reliable over time. Maintenance for the database involves the process of ensuring that the data where it has been oddly attired is now accurate, secure, and accessible for usages on demand. Otherwise, slow disorganized and full of errors become the databases, leading to serious problems in most organizations where it really drives and dictates life because it has many consequences with its dirty data values. Database maintenance is to make the system efficient, reducing 'downtime' and thus preventing any loss or corruption of data. When done religiously, the speed of performance of the database continues to be the best and can scale as and when the time demands, or the organization grows in its needs.  
  
There are several ways in which database maintenance could be performed depending on the nature of the database as well as the specific needs thereof. Backup and recovery, for example, is one of the methods that need to be implemented as all of them are critical for the protection of any data from inappropriate failures. Backup every which way ensures that an entire database is recoverable, without too much loss, when the unexpected happens, such as a hardware component failure or an attack by malware. Indexing and optimization are also among the most essential parameters of database maintenance. Indexing fastens the speed of the queries using organizing data in a way that makes it now much easier for access.  
  
Data cleaning is also among the most vital parts of maintenance. The data becomes quite often inaccurate, duplicated, or outdated. Defining cleaning means finding and fixing all such problems to keep the well-organized database free of unnecessary information. It is also important to monitor and update database applications. The truth is that databases will usually depend on a given software application that has to be updated: either to patch up any security holes found or add new features, thus making it important to stay current or risk opening up avenues for outside breaches or compromises problems.  
  
Lastly, capacity planning is another method to ensure the database can handle increased usage as more data is added over time. This involves analysing storage and performance needs and adjusting avoid system slowdowns or crashes.  
  
In conclusion, database maintenance is crucial for ensuring the database continues to meet the needs of the organization without issues. Whether it’s backing up data, cleaning and optimizing it, or monitoring performance, these tasks work together to keep the database reliable, secure, and efficient. Regular maintenance not only extends the life of the database but also makes sure it can handle any challenges in the future.

# Testing

## Data Validation

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Type** | **Description** | **screenshot** |
|  | All cases of PK | Here I will show that each primary key (Doner ID) is unique and none of them is null |  |
| Here I will show that there is no way to enter the same ID twice in the same table which ensures unique values always |  |
|  | All cases of FK | For the I will chose the event ID which is coming from the events table and its actually in the events participation ,Here I tried to add event id and doner id to event participation and it worked since both event and doner id exists in the original table ,But after that I have tried to add an event ID that doesn’t exist in the original table which resulted an error |  |
| The first photo shows the original data I had in event participation now I will change the event ID from 1 to 100, as we can see I have changed the values in events table and the values were changed on the event participation table |  |
| Now let us try to delete the event that has ID 100,  It was not deleted because of the on delete restrict |  |
|  | Unique | In the Doners table the email address is unique which means the same email can’t be repeated twice |  |
|  | ENUM | The payment method got two types of card or cash I will try to add another type |  |
|  | Not null | Similarly, I will try to add a donation without an amount |  |

## Output Validation

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Query Description** | **Screenshot (query + result)** | **Result validation** |
|  | Testing the events stat’s view |  | The view counts the numbers of donations and the sum of it and then group them by the event name, and as we can see in the picture each event name is followed by needed information |
|  | Testing the procedure add doner |  | As we can see it added the doner emad with auto increment for the maximum current ID so its working |
|  | Testing the doner participation stats procedure |  | As we can see its showing the doner name with the number of events and the total donated amount |
|  | This selects statement shows the event that got the most donations and the donated amount next to it |  | As we can see the most donated event is in need and it got 2700 JD and if we show the full table, we can make sure that this is indeed the most donated event |

## Security Validation

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Username** | **Description of privilege/no privilege** | **Screenshot (query + result)** |
|  | Doner | The doner can select from the table event |  |
|  | Doner | But can’t select from table doners |  |
|  | Doner | And cannot drop any table |  |
|  | Analysis employee | He can select from any table |  |
| **5.** | Analysis employee | But can’t change on any of them |  |

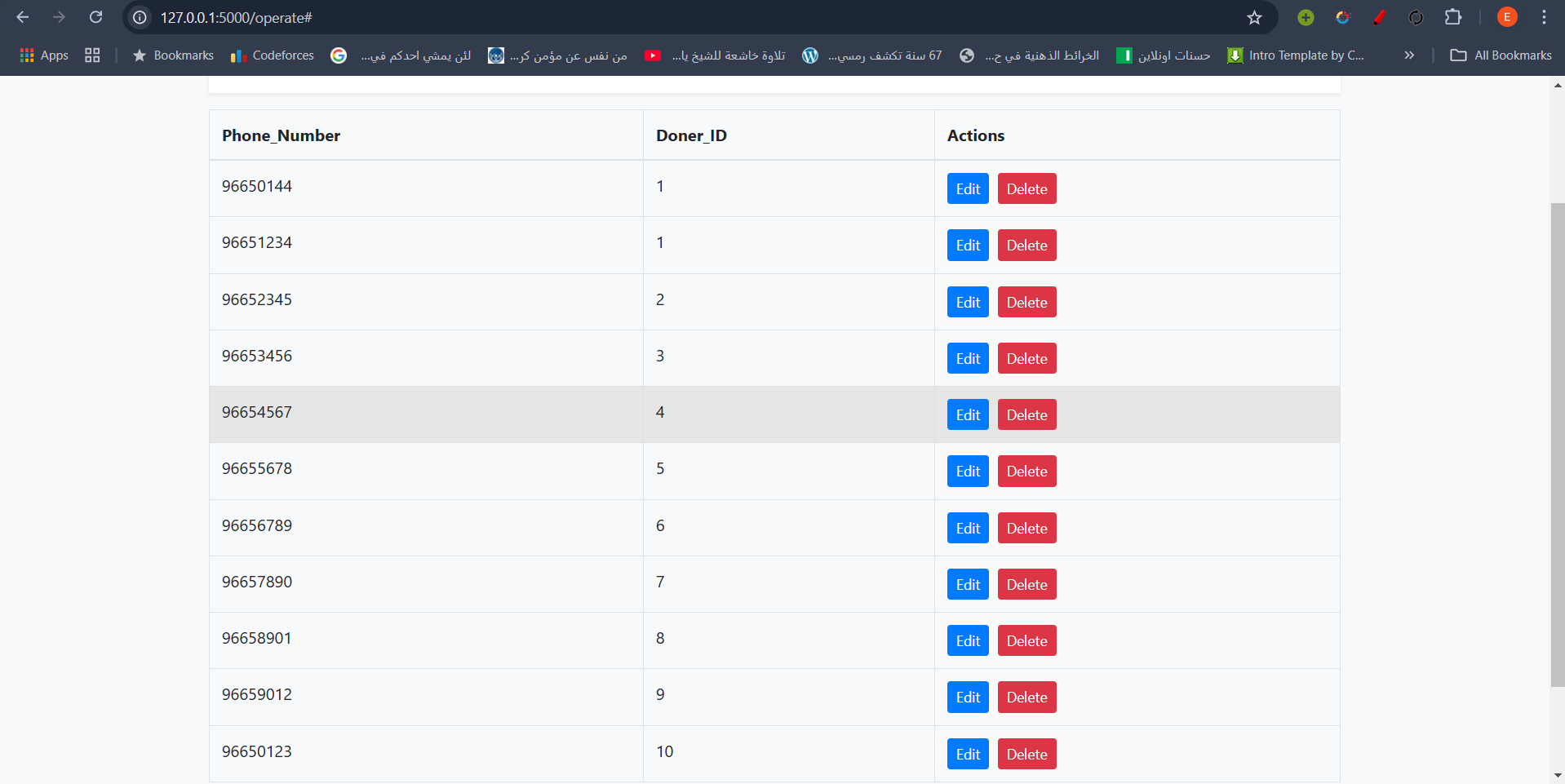
### 

## GUI Validation

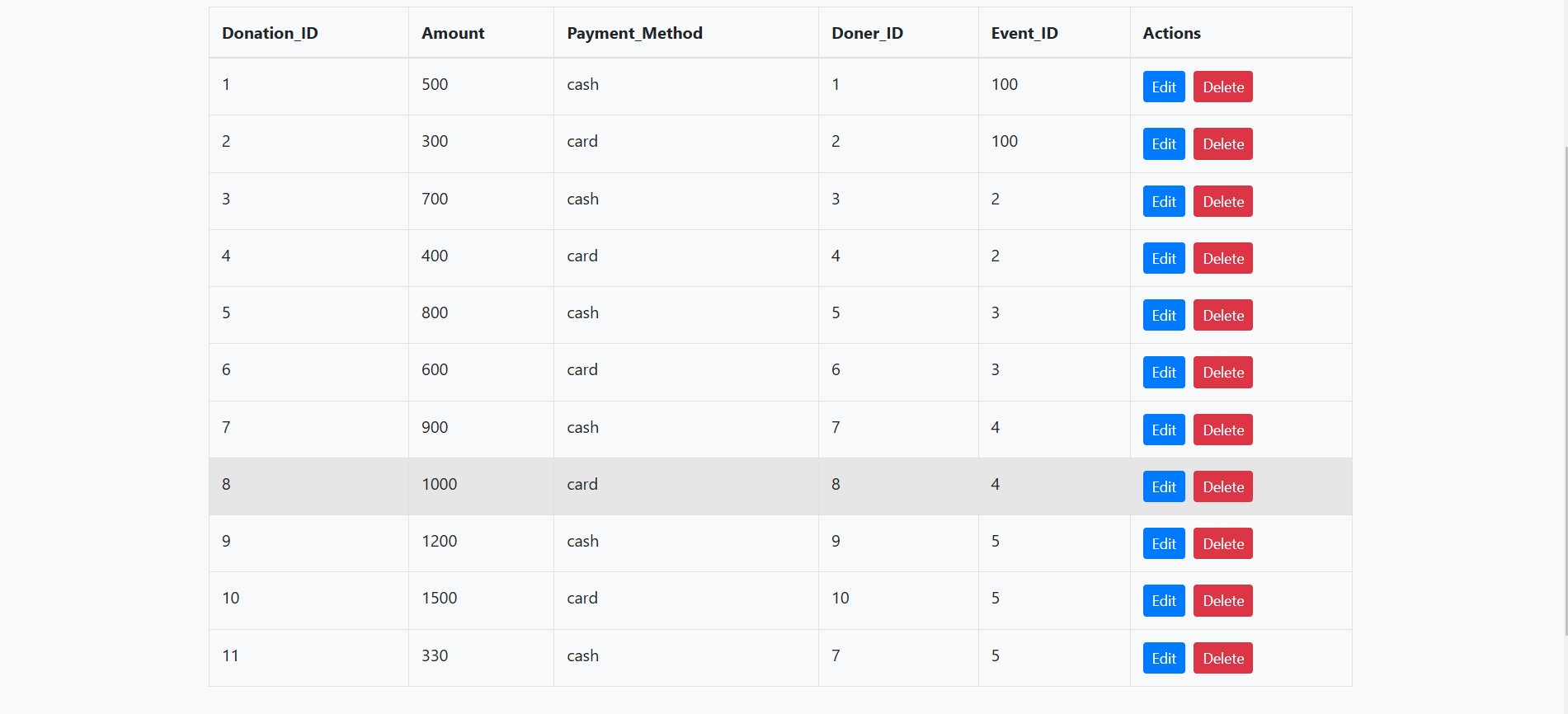
|  |  |  |
| --- | --- | --- |
| **Number** | **Description** | **screenshot** |
|  | Adding a new record for the doners table |  |
|  | editing the name of the last record in the doner table |  |
|  | Adding the doner id to the event participation table |  |
|  | Deleting a record from the doner table and his record in the events participation was automatically deleted |  |

## Assess whether meaningful data has been extracted

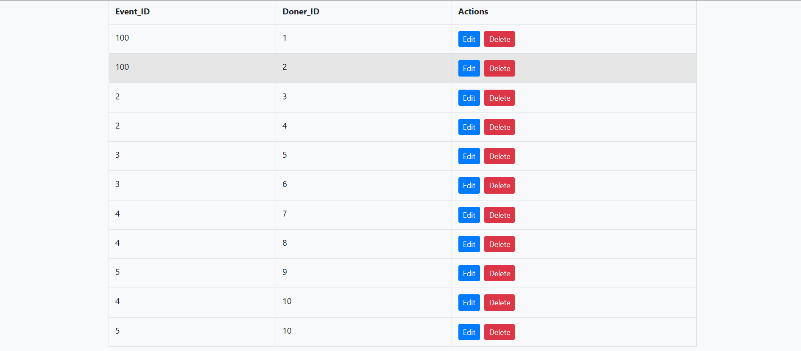
For this part I will talk about each case in my data base and an example from it to explain it in a good way, first of all lets start multivalued attributes I will talk about the doners phoner number ; each doner may have more than one phone number linked to his id and my design ensures that this case is handled well and we can see this in the doners phone number table :



As we can see here the doner that has id 1 has more than one phone number, and for the one to many each doner can make many donations and each event might get more than one donation, and we can see this in the donations table:



As we can see here doner 3 and doner 4 have donated to event two and here we can see a one to many relation between the event and the donations , and as we can see the doner number seven has donated for two events which are event five and four and here there is another one to many relation between the doner and the donations ,and finally for the many to many relationship a table were added to handle the data between the doners and events and this table is events participation , in this table a doner can participate in more than one event and each event can have more than one participant(doner):



As we can see here event number 100 has to doners who participated and doner 10 had participated in event 4 and 5

## Assess the effectiveness of testing

I have done quite a lot of selection-update-delete type under this design test. It proved vital as the reason that each added cascade method has been incorporated. These queries have established that all relationships between tables are well functioning, and updates or deletions in parent tables cascade into child tables, thus ensuring that the integrity is maintained.  
  
After this step, I did confirm data validation, meaning that there was no missing significant data or a foreign key that was no meaning in the parent table. In this step, I studied all data entries closely and searched for null values in significant fields. I also checked whether each foreign key has a valid existing corresponding primary key. Further, I executed a series of checks on data integrity to make sure that the constraints applied (e.g., NOT NULL, UNIQUE, CHECK) were functioning as intended, and I ensured that the database would be ready for use without any problems that may rise when editing or even adding data or records to tables.  
  
I've verified to set for each user the right privileges, not less or beyond what is needed for that particular user, for inappropriate access control leads to operations and security suffering serious troubles. For instance, with less privileges, work is usually not sorted completely. Like, the analysis employee without all the SELECT privileges will be unable to perform complete analysis, or draw significant assumptions based on it. On the other hand, having access to excessive privileges poses great risk, whereby an example should be if all donors would search or even view the data of other donors. It will affect the company and compromise the database's integrity and security.  
  
Therefore, I thoroughly tested each user role to confirm that their permissions were appropriately assigned, ensuring they could access only the data and functionality they required for their responsibilities.   
finally testing the GUI and make sure that its ready to use for all the employees was a very important step for the deployment of the database , while I was testing I noticed some missing cascades and wrong relationships in the database that were fixed   
  
Moreover, I tested the system's by inputting invalid data entries ;ensuring that appropriate error messages and validations were in place to guide users in correcting their input, i also verified the backup and recovery procedures, creating and restoring backups to ensure that data could be retrieved without any loss in the event of a system failure.  
  
By conducting this comprehensive testing process, I was able to confirm that the database was fully functional, reliable, and ready to support the organization’s operations.

# Evaluation of database solution

## Effectiveness of the database solution based on user and system requirement

The proposed database solution for EDU Youth Foundation will effectively solve user and system requirements by targeting major areas such as tracking donors, communication, participation in events, and management of donations. Assessment on compliance with the given requirements for the alignment of Solution to User and System Requirements, The system will be extremely successful in reaching its goal of managing donor communication and behaviour, with tables such as Donor, Communication History, Donations, Events, the database will make sure all the analytics, marketing, and management departments have all the relevant data ,Web or mobile interface for donors to view their participation and donations, ultimately enabling them to participate more with the foundation, workers have been assigned to specific roles and privileges, enabling them to securely view data related to their responsibilities in adding, analysing, or managing donor information and events and for the Achievements Data Integrity and Relationships, Constraints for primary and foreign keys ensure good relationships between such entities as donors and events for data integrity, Normalization in this case includes Event Participation and Donations tables, thus appropriately supporting one-to-many and many-to-many associations ;User Accessibility by  
Role-based access ensures that users interact only with what they are supposed to reach for security reasons , Interface design allows viewing, adding, or updating records and finally Error Handling,  
Different levels of validation have been performed in the system, which include no duplicate donor IDs or invalid foreign keys to ensure consistency in data quality ,Improved Communication ; the Communication History table keeps track of all interactions so that the foundation can keep tabs on donors and adjust marketing strategies ,for the unmet Requirements and Their Justifications:  
Dynamic Data Reports: While the "Event Stats" and "Donors Total Donations" views display insights in summarized modes, there is practically no evidence of real-time analytics that can be automated with regard to trends in donor behaviour or event performance, Increased Longer-Term Engagement with Donors: Invitations to events could contain personalized details regarding those aspects of a donor's career that were inaugurated, thus allowing for much more in the way of customized outreach, These may include features that are more time-consuming or resource-intensive to implement, such as advanced analytics integrations or more dynamic querying mechanisms.  
New Requirements Included  
An even table for maintaining donor phone numbers, and other views such as "Card Donations" and "Cash Donations," though not initially defined, were developed due to multivalued attributes for extended functionality. These illustrate the flexibility of the system toward changing needs so that it will be practical in real-life applications.

This solution will meet key requirements of the project and hence will be of considerable use for EDU Youth Foundation in significantly improving donor communications and also event management. Dynamic analytics and customized engagement features have been left out in this system's development. This would give a pretty good foundation for the systems in further development. This would cover user accessibility, data security, and integration immediate needs for the foundation and scale for growth.

## Suggested improvements

1. Database Design Enhancements  
Although the database design handles many-to-many relationships and normalization quite well, it becomes somewhat useless because of the lack of more detailed profiling of donors. For example, information on donor preferences, frequency of donation, or reasons for donation is not available.  
Since most of the requirements have already been covered, this schema has scope for only minor addition of data that can also be further enhanced in event data and advanced segmentation of donors. Introduce a new entity, Donor Preferences, that will store information such as donor interest, preferred communication, or event type likely to support for personalized outreach.  
Add attributes like Event Budget and Event Type in the Event table for tracking financial performance and categorization of events, respectively.

2. Security Enhancements  
While the system implements role-based access control, it does not encrypt sensitive data such as donor email addresses or phone numbers. It does have backup procedures but does not mention automated disaster recovery mechanisms. This will involve integrating encryption for sensitive columns-emails and phone numbers-for data protection, especially against unauthorized access. It will implement 2FA for system users in order to enhance security.  
  
3. Improved Analytics and Reporting  
  
Reporting today is via canned views such as Event Stats or Donors Total Donations. It does not support advanced analytics or real-time data visualization. There is a lack of predictive analytics to identify trends-for example, which donors will donate again. Integrate some BI tool like Tableau or Power BI that allows the employee to create dynamic real-time reports.  
It will develop an AI-based event or donation recommendation system, in the same fashion MathWorks has applied AI use in their suite of tools; it analyses behaviour regarding donor habits and targets events or donations.   
  
4. Business and User Experience Improvements  
The system does not support clear methods of encouraging donor retention or automating follow-ups from events or donations.  
GUI is functional but could be friendlier and more intuitive to use by non-technical people. Add an Automated Follow-up System that will send personalized thank-you emails to donors for each contribution and event participation, thus improving donor retention.  
Enhance the GUI with drag-and-drop for event organizers to plan events and add graphical charts that summarize donation statistics. MathWorks does a great job with their tooling, and adopting their iterative GUI testing and user feedback would go a long way toward refinement of the design.

## Evaluation based on improvements needed

The improvements listed in Section 9.2 will make the database solution highly effective to realize its shortcomings while at the same time putting it into best practices as done by industry leaders like MathWorks. By adding new entities like Donor Preferences, and new attributes such as Event Budget, the database will provide detailed information for personal outreach and strategic planning of events. Data accuracy will improve further with normalization and better schema design; redundancy will decrease, which again will help avoid errors. Security: Encryption of sensitive data and 2FA application will improve data protection and help gain trust among donors. Automation of backups and disaster recovery processes to minimize downtime and ensure business continuity in case of failures will be similar to the robust disaster recovery mechanisms put in place by MathWorks while designing their cloud systems.  
  
There will also be great improvement in scalability when proper indexing, replication of databases, and cloud infrastructure are applied to maintain the responsiveness and efficiency of the system during the growth of the organization. Advanced analytics integrated with business intelligence tools, such as Tableau or Power BI, would facilitate real-time reporting and predictive insight, thus enabling management to make informed decisions. For instance, predictive analytics may identify donor behaviour trends that can be used to enhance fundraising efforts. Furthermore, it will introduce improvements in the user experience by implementing automatic follow-up to strengthen relationships with donors; a more refined GUI has drag-and-drop capabilities and graphical summaries of donations, which will make the system more user-friendly for employees.  
  
Of course, all these enhancements will bring in some challenges in their implementation. Development time and costs will increase due to new features and advanced integration of security protocols. The more significant the change-especially in tools and interfaces-sometimes requires additional training for the employees, even infrastructure upgrades for example, the migration to cloud services-can carry high costs of their own, despite these potential challenges, the long-term benefits considerably outweigh the short-term pains: aside from being highly secure and scalable, the upgraded database solution will also be more user-friendly, better positioned to meet organizational needs, and easier to support going forward.

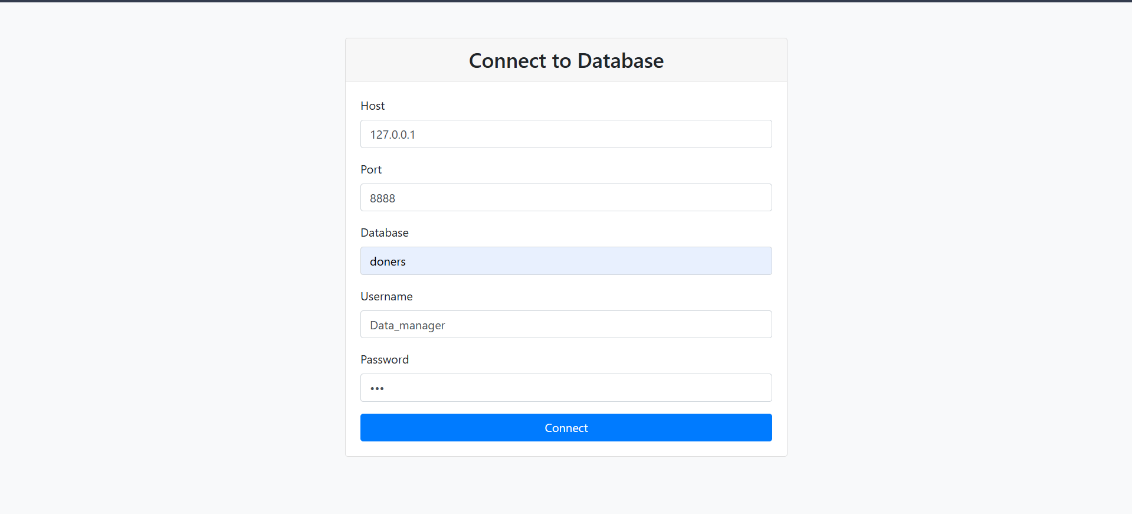
# User Documentation

## System Overview

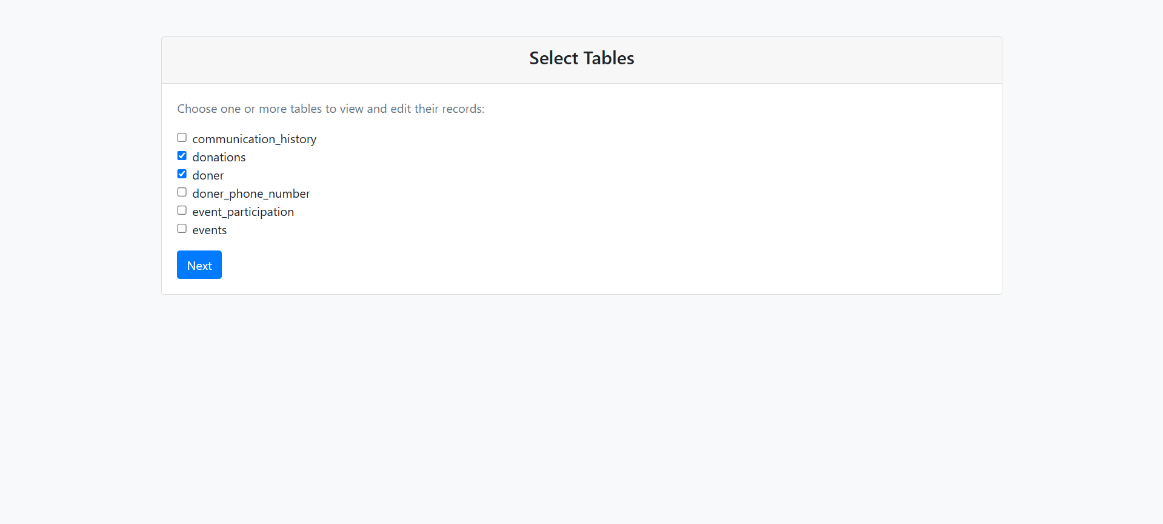
This database system stands poised as a multifarious tool, guiding the user smoothly with the requisite information to manage data securely and efficiently. The aim is to allow users to enter and edit certain pieces of data according to their role. The system makes use of tables, which are well-connected, of course, for different purposes like storing donor details or record of participation in any event. For instance, you cannot add a record in the events participation table if it has a doner that is not in the database. And to log in you should type in the username and password, which, on being correctly transmitted, allows access to the tables to be provided. The selected tables are displayed in a tabbed interface using an Excel-like format where users can add or delete records according to their permissions.

## Using the system

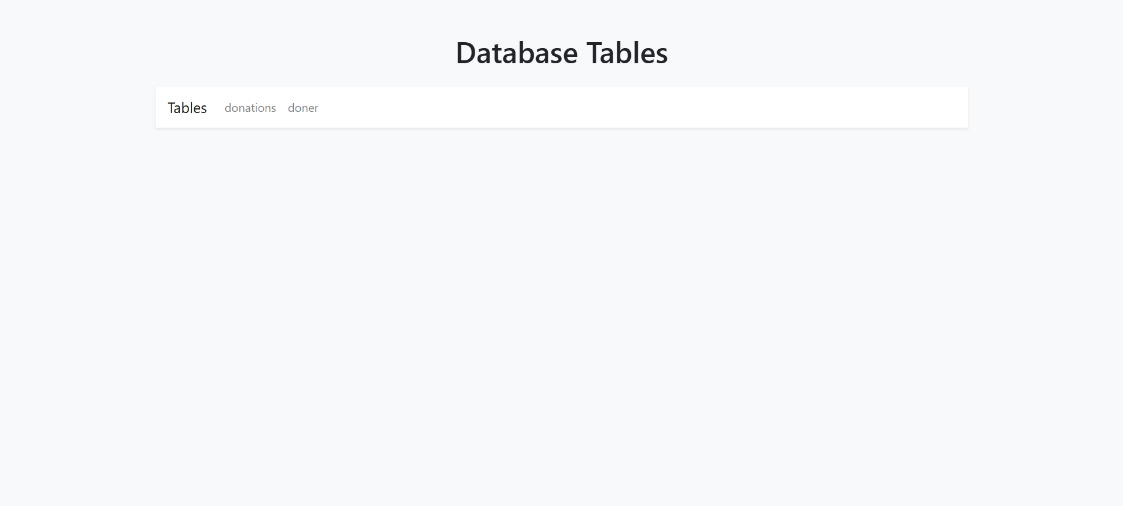
This part will be a guide of how to use the system and what can you do in it, however I will log in as a data manager you should login based on your job or role

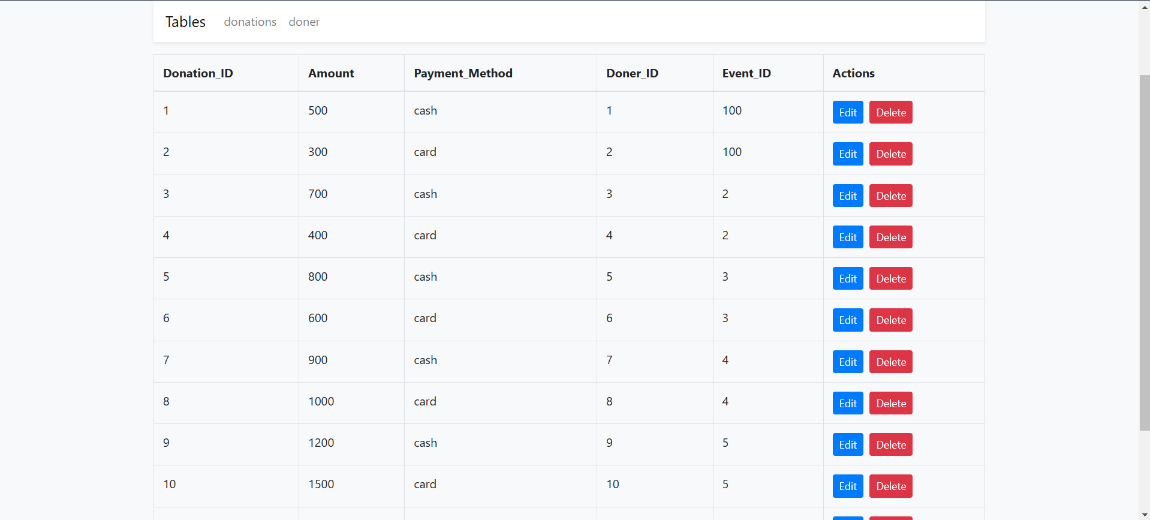


Firstly, you should fill in the username and password

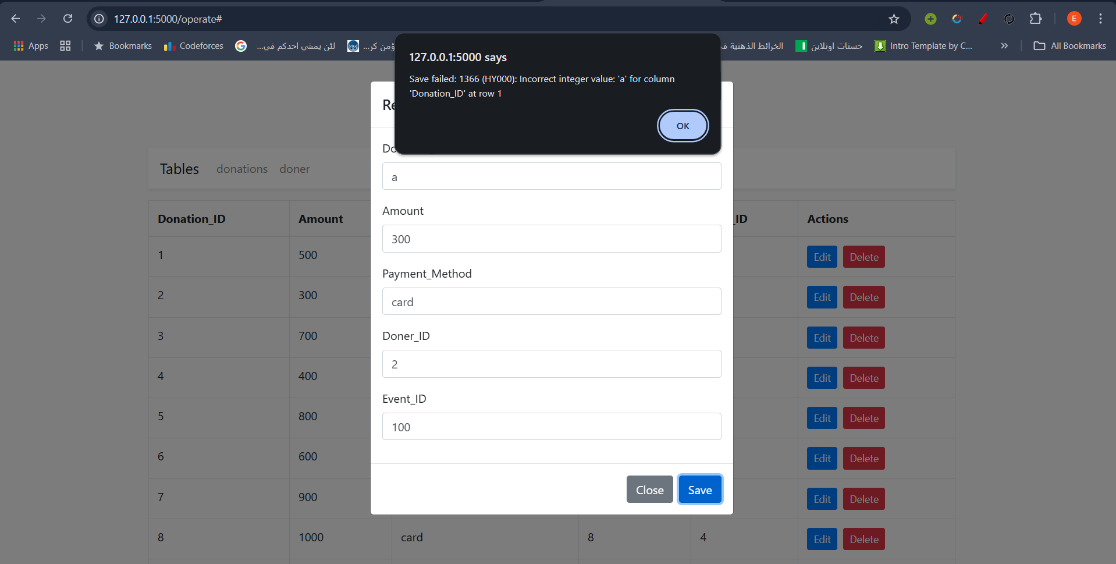


After that all the tables that you have access on will appear and you can choose any of them here I have chosen the donations table and the doner table

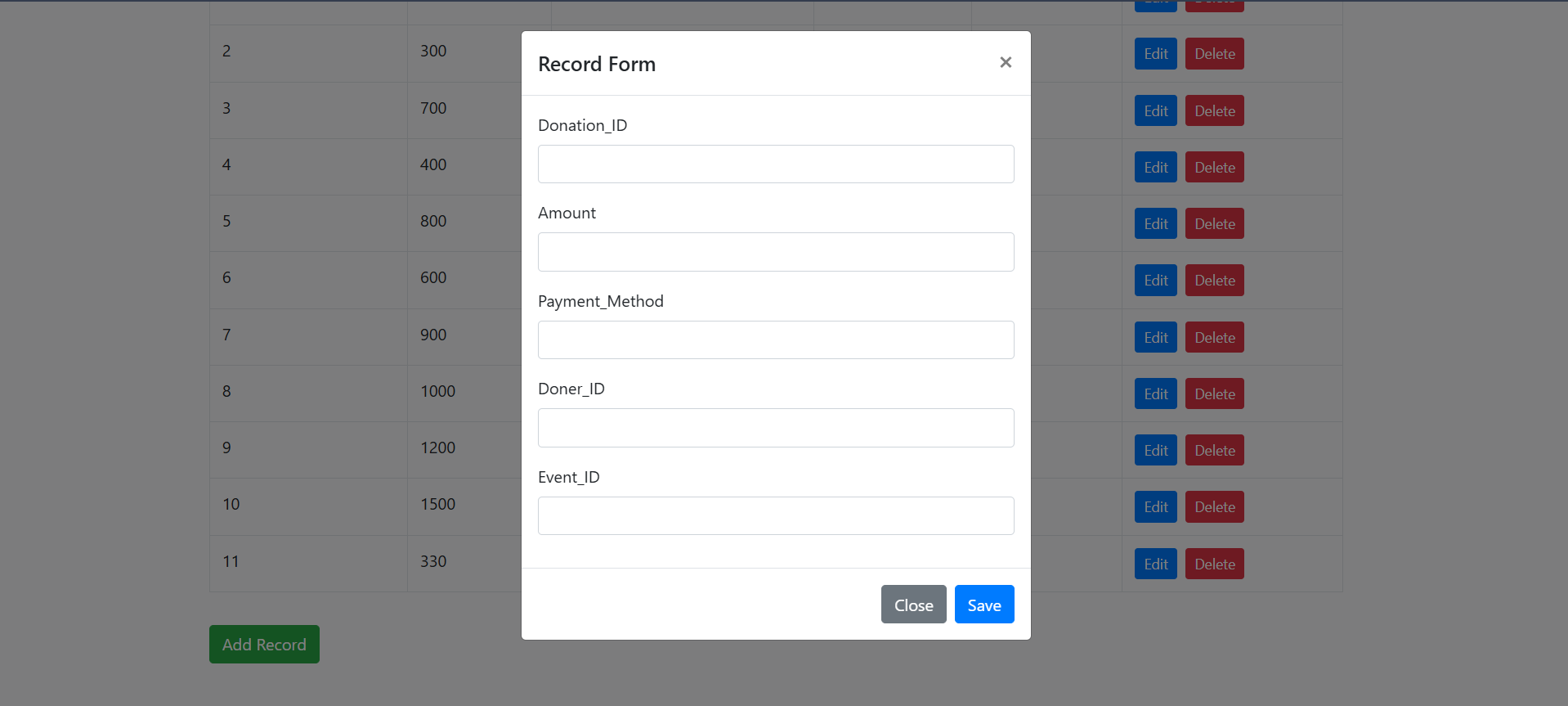


This tab will appear and you will have to choose the table you want to edit 

After that you will see this page that has all the options of editing deleting and adding records



And don’t worry about the data entry every possible error will be well handled with a pop massage like this one



And for adding any record just press the green button in the down left corner and a data entry window will pop, just fill in the data and press the save button and the record will be added

## Frequently asked questions

### -Why can’t I edit or see all the tables of the data base?

Each user was given access based on his role in the company to avoid any mistakes that can lead to loss of data or loss of the integrity

### -What is the password and username?

Such information should be supplied by the company or the head of the department

### -how long would it take to make such a system for other uses?

Based on the traffic it can take from 5 days to 3 weeks

### -How can I edit in the system or customize it?

Feel free to contact me for customizing any detail in the system

## Contact information

If any help needed feel free to contact me in [23110354@htu.edu.jo](mailto:23110354@htu.edu.jo)

You can email me any day from 2 pm until 5pm