ITEC617-MODERN DATABASE MANAGEMENT

Assessment 3: Database

Semester 2, 2024

Submitted To:

Submitted By:

Bishal Bidari

S00391799

Date of Submission:

Word Count:

# Introduction

This study describes how Greenpeace NGO developed a safe and effective database system to manage vital information about volunteers, donors, and campaigns. Our strategy places a strong emphasis on structured data management, safe data storage, and actionable insights to aid in the organization's decision-making. Every component of the database was constructed with concern for data protection, ethical issues, and simplicity of data retrieval, from schema design to SQL query development. The report will examine the relational structure, basic architecture, and applied business rules that govern data entry, storage, and security in the database.

The database provides a structure for organising and protecting data with high integrity, allowing Greenpeace to better track contributions, campaign progress, and volunteer participation. The research also assesses ethical and security risks, including recommendations for continuing security, Azure-based monitoring, and ethical data management. This foundation helps Greenpeace achieve its mission by strengthening data reliability and accessibility, ensuring that information is secure, accurate, and relevant for operational purposes.

## Assessment Background

The main goal of this study is to create a safe, organised database that will let Greenpeace NGO effectively handle its vital information about volunteers, sponsors, and campaigns. The project's foundation is the requirement for a well-structured system that facilitates the NGO's day-to-day activities while guaranteeing data correctness and convenient information access. This project intends to simplify data management and give Greenpeace trustworthy insights by building a relational database with precisely defined elements and relationships. This foundation supports the organization's mission and improves operational effectiveness by enabling the safe storage, retrieval, and analysis of data.

## The scope of the Report

The design, implementation, and security of a relational database created especially for the Greenpeace NGO are included in the scope of this paper. This database offers an organised method of data handling by storing and managing information on funders, volunteers, and campaigns. The paper does not address complex analytics or integration with external platforms, but it does cover fundamental topics including database structure, security features, and basic SQL queries. Therefore, the scope is restricted to creating a basic database system that satisfies the operational requirements of the company.

## The purpose of the Report

The construction of a database system that allows Greenpeace NGO to store, safeguard, and retrieve important data pertaining to its operations is the focus of this paper. The database structure, data flow, and security procedures are explained in this paper for the benefit of users and stakeholders. It describes the system's features, restrictions, and data-handling procedures, assisting Greenpeace in effectively managing its data while upholding ethical and privacy standards.

## The overall objective of the report

The overall goal of this report is to securely organize and manage the donor, volunteer, and campaign data of Greenpeace NGO in order to build a strong database that supports the organization's mission. The database offers crucial tools for data retrieval, reporting, and analysis through the implementation of a systematic and safe storage system. This goal is in line with Greenpeace's objectives to guarantee data accuracy, improve data accessibility, and maximize data usage for well-informed decision-making.

## Important of the report

The reason this report is significant is that it creates a safe and organized data base that helps Greenpeace track and evaluate its campaigns, donations, and volunteer work. The report guarantees responsible data handling by offering thorough documentation of the database's architecture and security protocols. This enables Greenpeace to safeguard individual data and make data-driven decisions that improve operational efficacy.

## Any limitation of the report

There are certain restrictions on the report. It focuses on fundamental database structure and security rather than complex data analytics, interactions with external tools, or real-time data streaming. The project also makes the assumption that the data is accurate and consistent, which may need to be updated frequently to stay relevant as the company changes. Future modifications might be required to conform to shifts in Greenpeace's organizational structure or data requirements.

## Business rule

A number of business rules are incorporated into the database to ensure the integrity and accuracy of the data. To avoid duplication and provide precise tracking of contributions and involvement, each contributor, volunteer, and campaign is given a unique identification number. Because donor contributions are tracked for each campaign, a thorough examination of campaign funding is possible. In order to support a many-to-many relationship model that fits the operational demands of the organization, volunteers can take part in several campaigns. These guidelines offer a solid basis for managing data.

## Assumption

To ensure successful implementation, this study is predicated on a number of assumptions. It makes the assumptions that the information supplied is correct and that database users will have distinct roles with restricted access rights. Furthermore, it makes the assumption that the stakeholder and campaign structures won't change frequently, enabling the database design to continue to meet organizational requirements. These presumptions aid in directing the database system's architecture and security.

# Conceptual Design

## 2.1. About Dataset

The Greenpeace project dataset consists of numerous tables that capture critical information about volunteers, campaigns, donors, environmental resources, and climate impacts. The Volunteers table keeps track of volunteer names, dates of birth, regions, and contact information, allowing Greenpeace to manage its global network of activists. The Campaigns table lists each environmental initiative's name, focus, and fundraising targets. Donors are tracked using the Donor table, which includes donation amounts and dates. Furthermore, the Resources and changes tables contain critical information about the environmental resources that are affected by climate changes, as well as the degree of those impacts.The dataset captures many-to-many relationships using associative tables such as Volunteer\_Campaign and Campaign\_Resource. This organized dataset enables Greenpeace to efficiently manage its global operations, track campaign progress, and optimize resource allocation.

## 2.2. Entity: Definition with proper Reference

An entity in a database is a unique, identifiable thing or concept that is relevant to the organization's operations. It is represented as a table in a relational database, with each instance of the entity corresponding to a row and the entity's attributes represented as columns. Entities can be both real items (such as products or resources) and abstract concepts.According to Elmasri and Navathe (2016) define an entity as "anything about which data is to be collected and stored" (p. 122). This concept stresses that entities serve as the core of a database schema, providing a systematic method for capturing and organizing information essential to an organization's business activities.

## 2.2.1. Identified entities from dataset

From the dataset I have designed, the following entities were identified based on the requirements:

* Volunteers
* Campaigns
* Donors
* Resources
* Impacts
* Volunteer\_Campaign
* Campaign\_Donor
* Campaign\_Resource
* Impact\_Resource

## 2.3. Attributes: Definition with proper Reference

Attributes are generally descriptors that provide additional insight about an entity, defining its characteristics within any dataset. For example, in the case of the Greenpeace NGO database, the selection of attributes carefully encompasses data necessary to support business processes, such as enacting commands on campaigns, volunteers, donors, resources, and their impact. The attributes enumerate the following critical information: dates, amounts, statuses, relationships with other entities. Gathering volunteers' backgrounds, campaigns, donations, and environmental changes will thus allow Greenpeace to cross-reference demographic background versus campaigns, donations against anything, the impact of environmental changes, etc., hence ensuring its database is stocked with information efficaciously needed for analysis or operations. In addition, the attributes include restrictions for the effectiveness and integrity of data.

## 2.3.1. Identified attributes from dataset

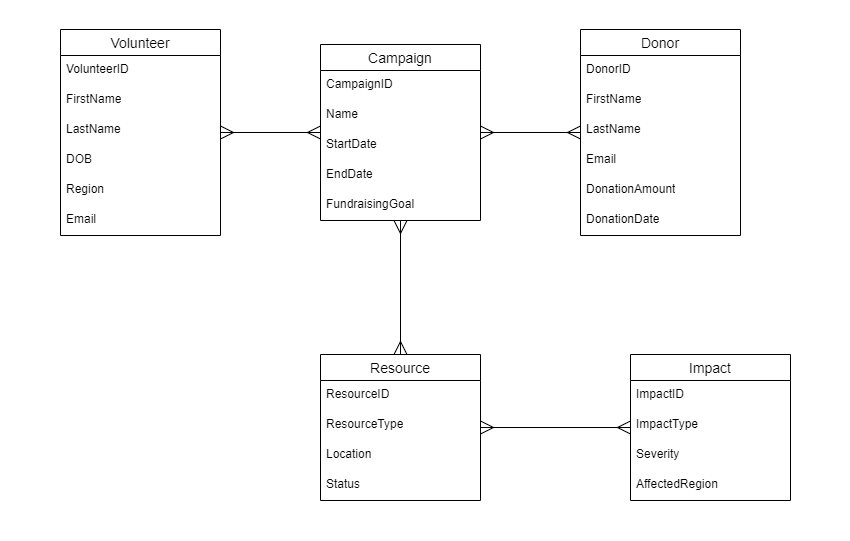
Greenpeace NGO's dataset includes multiple important entities, each with accompanying properties. The following image identifies properties for each entity in the dataset: A screenshot of a computer

Description automatically generated

## 2.4. Relationship: Definition with proper Reference Diagram

The NGO Greenpeace database relationships can be described as constituted by main entities constituting this type of organization, such as volunteers, donors, campaigns, impacts, and resources. The diagram below shows how these elements interact with a view to realize organizational objectives. There are four main many-to-many relationships:

* **Volunteers to Campaigns:** A volunteer can work for many campaigns, and one campaign can consist of many volunteers. This relationship is stored in the *Volunteer\_Campaign* table.
* **Donors to Campaigns:** One donor can donate to several campaigns, and one campaign can get its donations from numerous donors; this is reflected in the puzzlingly named *Donor\_Campaign* table.
* **Campaigns to Resources:** One campaign needs a resource, and one resource may be used for many campaigns. This relationship is maintained in the *Campaign\_Resource* table.
* **Impacts to Resource:** Impacts are driven by resources, and one resource can drive many impacts, based on what is recorded in the *Impact\_Resource* table.

Below is an Entity-Relationship Diagram (ERD) illustrating these relationships: 

## 2.4.1. Identified relationship from dataset

The following dataset for the Greenpeace NGO project describes the general ways key relations among entities can help in understanding how volunteers, donors, resources, campaigns, and environmental impacts interact within the organization's operations. For each relationship described below, consider relevant entities, the relationship type, the supporting table, and associated attributes.

* **Volunteer to Campaign**

***Relationship Type***: Many-to-Many  
***Entities******Involved***: Volunteer and Campaign  
***Description***: One volunteer can participate in several campaigns, and each campaign can be supported by any number of volunteers.  
***Supporting Table:*** Volunteer\_Campaign  
***Attributes in Relationship Table***: VolunteerID, CampaignID

* **Donor to Campaign**

***Relationship Type:*** Many-to-Many  
***Entities Involved:*** Donor and Campaign  
***Description:*** One donor can donate to many campaigns, and one campaign can get contributions from many donors.  
***Supporting Table:*** Donor\_Campaign  
***Valid attributes in the relationship table:*** DonorID, CampaignID

* **Campaign to Resource**

***Relationship Type:*** Many-to-Many  
***Entities Involved:*** Campaign and Resource  
***Description:*** A campaign is associated with many resources, and a resource may take part in multiple campaigns.  
***Supporting Table:*** Campaign\_Resource  
***Attributes of the Relationship Table:*** CampaignID, ResourceID

* **Impact on Resource**

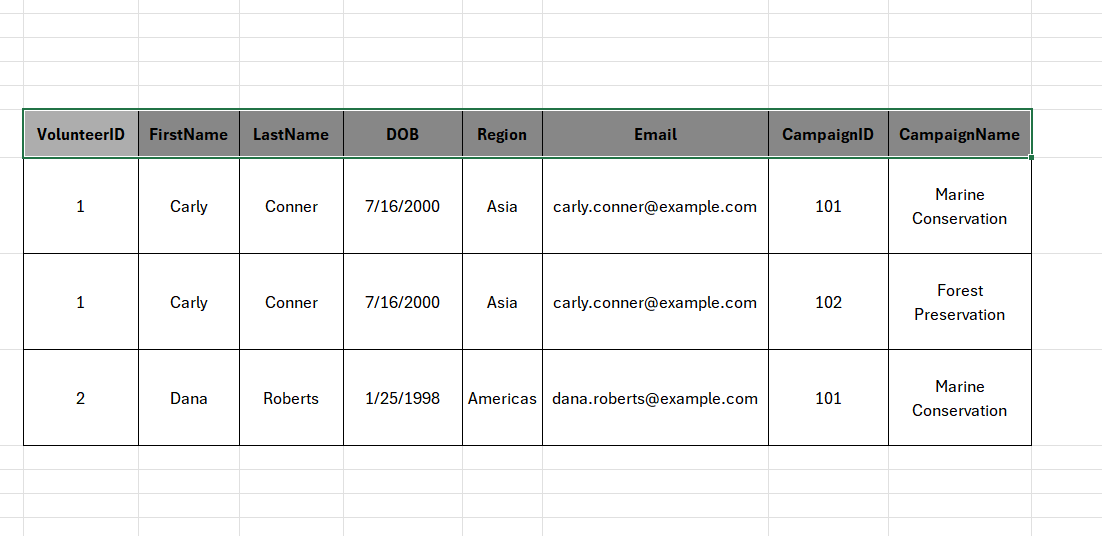
***Relationship Type:*** Many-to-Many  
***Entities Involved:*** Impact and Resource  
***Description:*** Each resource can contribute to several environmental impacts, and every impact may involve many resources.  
***Supporting Table:*** Impact\_Resource  
***Attributes in Relationship Table:*** ImpactID, ResourceID

## 2.5. Data Schema

The schema of data represents the structure of the database, showing normalization steps from the unnormalized form to 3NF. Each stage refines the dataset into lesser redundancy for improved data integrity and in a more structured format.

### 2.5.1. Unnormalized dataset:

The dataset provided for the Greenpeace NGO database is already in normalized form. In this post, we will not go into any details on normalization but take one example of what the Volunteers table looks like in an unnormalized format and how its data is organized as normalized for demonstration.

This might result in duplication rather than efficiency, because in an unnormalized structure, data for a volunteer may be duplicated numerous times if the same person volunteers or gives to multiple campaigns. For example, if campaign information is stored in the same table as volunteer information, it could look like this: 

In this unnormalized formredundancy is created with the CampaignID and CampaignName fields in that if a volunteer responds to more than one campaign, their information will then appear across rows many times. Redundancy can create inconsistencies and complexities when it comes to updating, for example, if a volunteer record changes. This example just shows why normalization is a necessary process and not an option to take; now, we will apply the same to our data structure in consecutive sections with 1NF, 2NF, and 3NF where finally data consistency can be guaranteed with adequate efficiency.

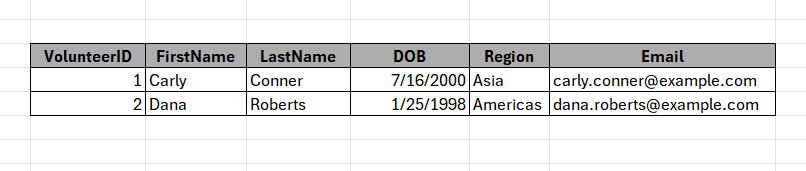
### 2.5.2. 1NF

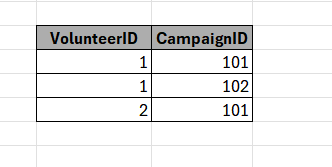
To show how normalization enables us to structure data more efficiently, let's try and normalize the Volunteers table. The goal of First Normal Form is to get rid of repeated or duplicate data and have every piece stored in its simplest, most atomic form in short, each cell having a single value only.

The Volunteers table, as it currently exists in unnormalized form, creates a lot of duplicate information, especially for volunteers who engage across many campaigns. For instance, VolunteerID 1 shows twice: once for each campaign that she worked on. This redundancy forces her data to repeat, which could result in inconsistencies and additional time spent on manual entry so, now let transform the table to 1NF:

* Transformation to 1NF

To achieve 1NF in this scenario, we need to separate volunteer information from campaign information, making sure that each row in the volunteers table is unique and hold atomic values. For doing that the first step is by creating new Volunteer\_Campaign table to record each volunteer campaign involvement separately.

* Volunteers Table in 1NF
* Volunteer\_Campaign Table in 1NF



Now the only distinct volunteer data is now present in the Volunteers table in 1NF, with a single value per cell in each row. A more effective and consistent data structure is ensured by the new Volunteer\_Campaign table, which records the relationship between volunteers and campaigns without duplicating volunteer data.

### 2.5.3. 2NF

Second Normal Form (2NF) aims to remove the partial dependencies such that all non-key attributes are fully dependent on whole primary key. This step creates all associated data that be joining against our main table, however, this is formal to similar sizes of generated datasets being used for join benchmarking purposes and assumes not familiar with normalization approaches. 2NF is mainly applied to composite keys, i.e., the primary key has more than one column.

* Transformation to 2NF

Once we convert the Volunteers table to 1NF, we have two tables instead: Volunteers and Volunteer\_Campaign. We examine each to determine if they meet the requirements of 2NF.

* + **Volunteers Table:**

The table has only a single primary key, VolunteerID and each attribute (FirstName LastName DOB Region Email) of the Volunteering data records are fully functional dependent on this Primary key. There are no partial dependencies in this table thus, it is already fulfilling 2NF normal forms.

* + **Volunteer\_Campaign Table**:

This table also has a composite primary key, VolunteerID and CampaignID, which represent the many-to-many relationship between volunteers and campaigns. Since this table includes only the composite key attributes and no other non-key attribute, it satisfied 2NF naturally.

### 2.5.4. 3NF

The Volunteers and Volunteer\_Campaign tables have already been simplified to meet the requirements of 2NF, but we will now check that each table adheres to 3NF.

* **Volunteers Table:** This table depends solely on the primary key, VolunteerID, for all its non-key attributes, which include FirstName, LastName, DOB, Region, and Email. The Volunteer’s table is in 3NF because the non-key attributes are not dependent on each other.
* **Volunteer\_Campaign** **Table**: This table connects VolunteerID and CampaignID. It naturally satisfies 3NF as it contains only primary key attributes and no additional non-key data.

## 2.6. Database type and constraint

As for this project, I chose MySQL as the database type. MySQL is a relational database management system type with its data and schema strictly defined. As MySQL incorporates the architecture of a complex data system into its own framework, most applications are now built inside it. MySQL data can be stored in structured tables, rules can be applied to keep it accurate, and tables can be joined smoothly. While column constraints specify certain attributes inside a record depending on the type of data and its relevance to the record's role, table constraints set restrictions based on the type of data and relationships between columns. Table constraints apply rules that guarantee data integrity across several columns, whereas column constraints limit specific items within a record.

A screenshot of a computer

Description automatically generatedTo maintain data integrity and make sure it obeys the expected structure, I defined constraints for each table. Here’s an overview of what I chose to do which keeps the database clean and reliable:

# SQL Queries:

## Create a blank SQL database in MySQL. Use your surname as the database name.

In this section, I set up a new MySQL database named after my surname and created tables based on the entities and relationships defined in the conceptual design.

### 3.1.1. Database Creation:

To begin, I created a blank SQL database named **Bidari.**

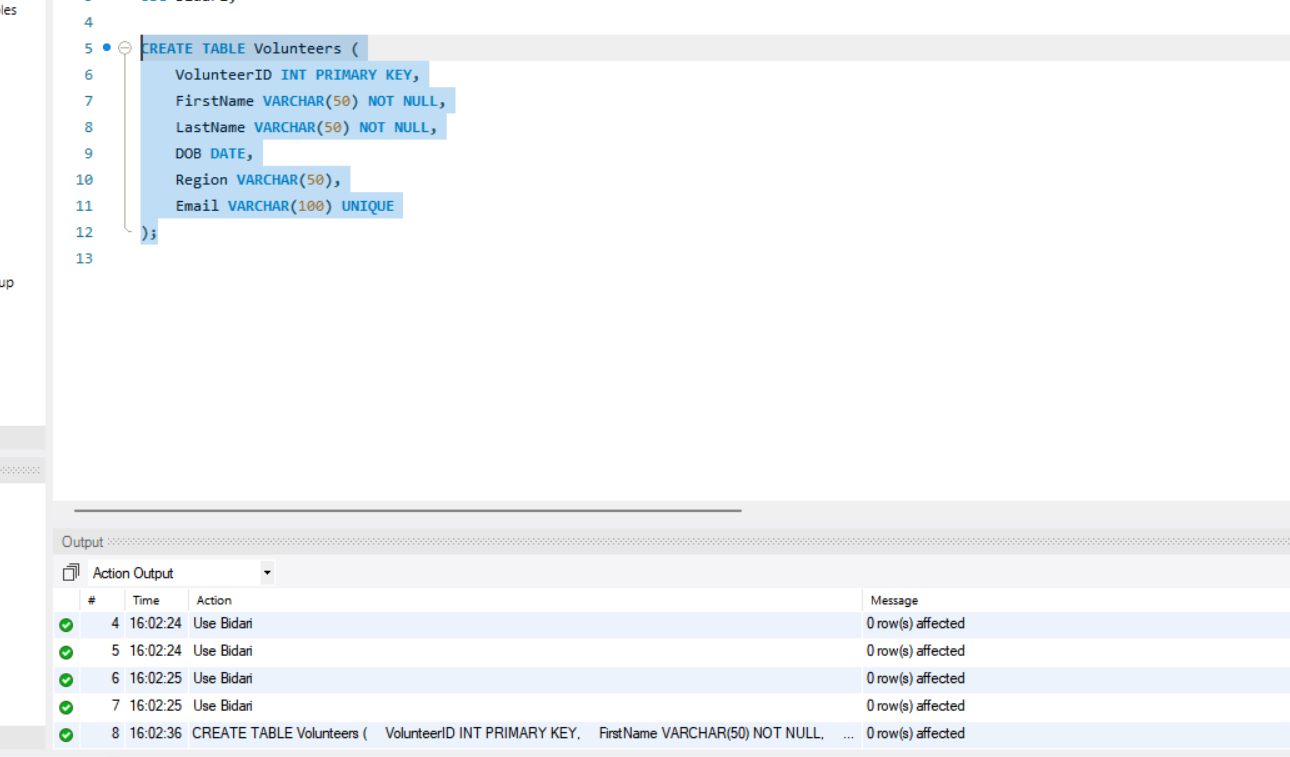
A screenshot of a computer

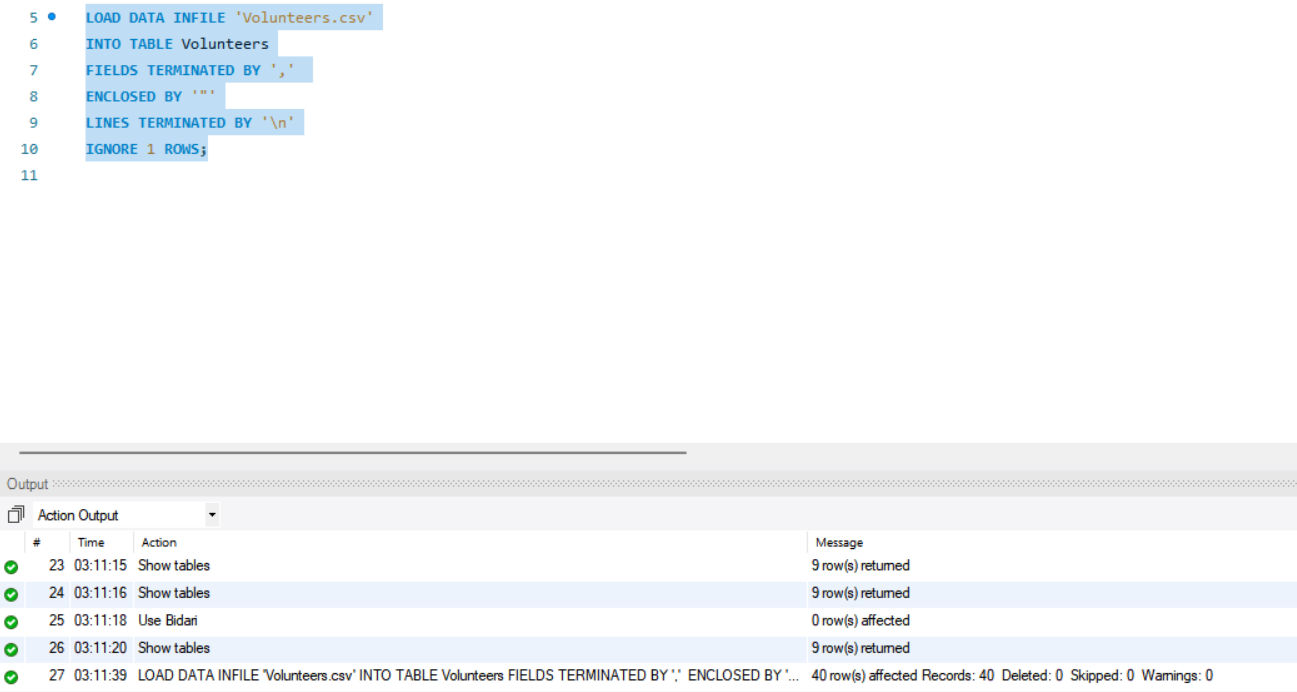
Description automatically generated

### 3.1.2. Table Setup and Relationships:

After Creating Database I must create Table in it according to the Dataset and Load Data in it so, as per the Dataset and conceptual design I created 9 table in the database. Below are the image file of table creation and Data loading query:

#### 3.1.2.1. Volunteers Table

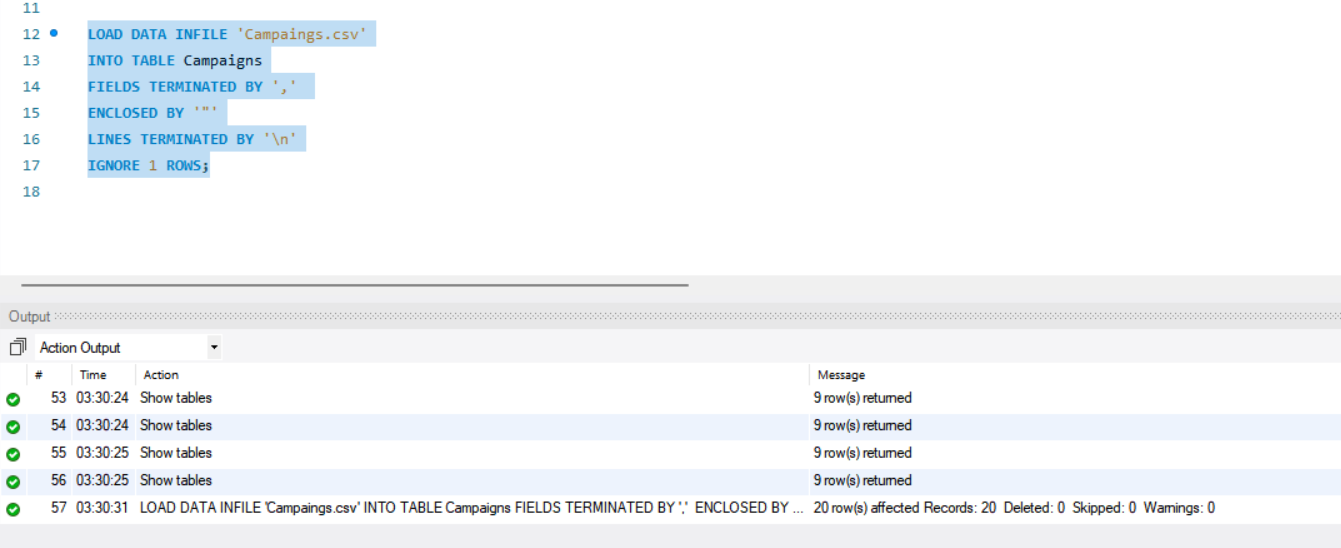


Load Data in Table:

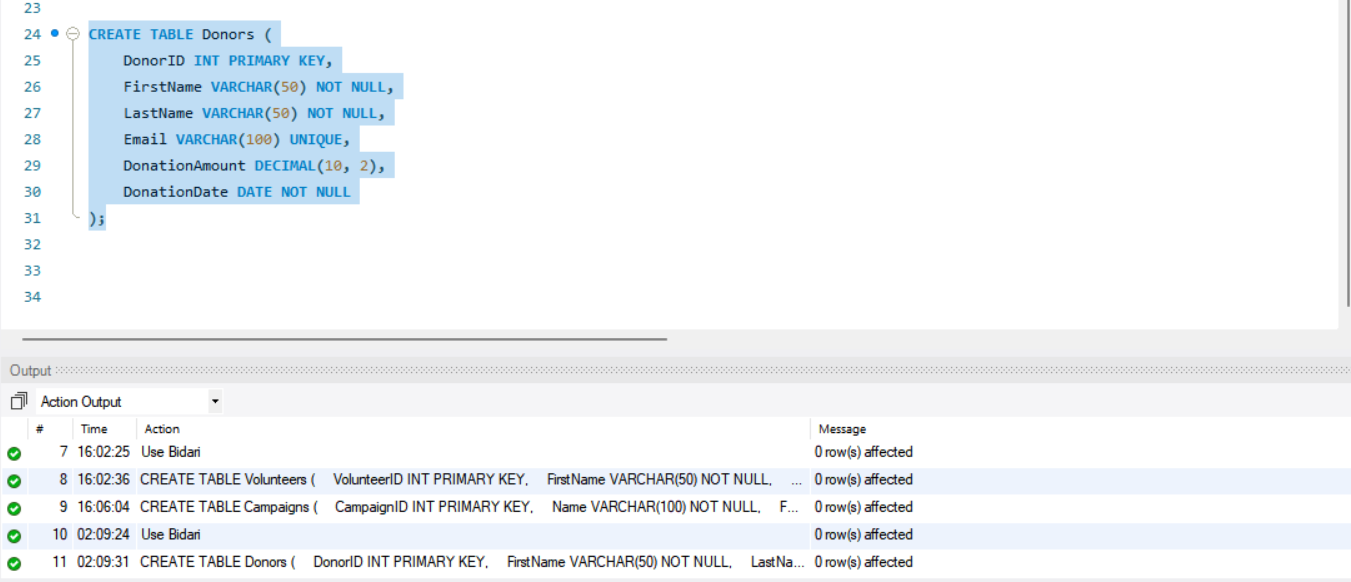
#### 3.1.2.2. Campaigns Table

A screenshot of a computer

Description automatically generated

Load Data in Table:

#### 3.1.2.3. Donors Table

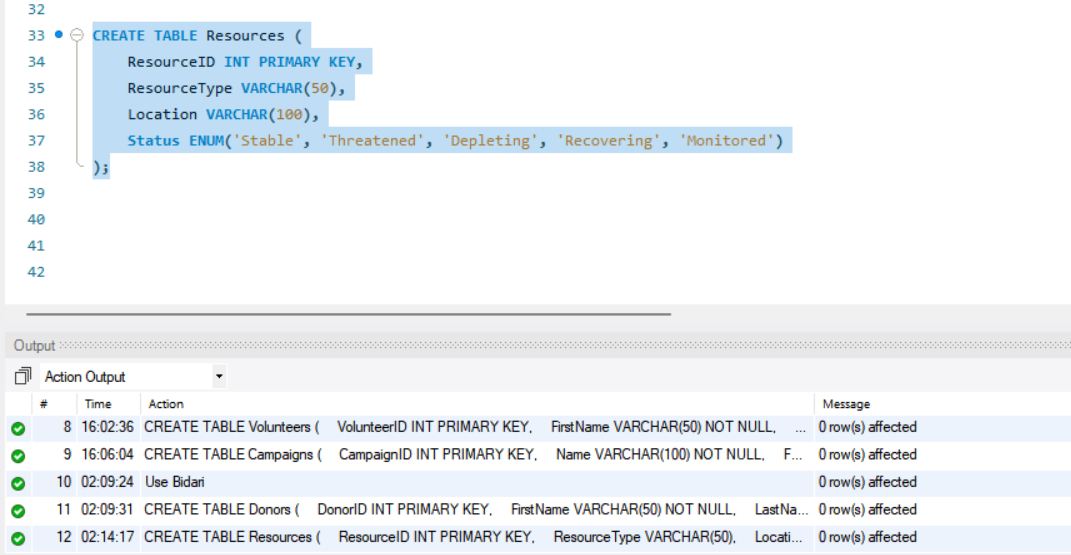


Load Data in Table:

A screenshot of a computer

Description automatically generated

#### 3.1.2.4. Resources Table



Load Data in Table: A screenshot of a computer

Description automatically generated

#### 3.1.2.5. Impacts Table



Load Data in Table:

A screenshot of a computer

Description automatically generated

#### 3.1.2.6. Volunteer\_Campaign Table

A screenshot of a computer

Description automatically generated

Load Data in Table:



#### 3.1.2.7. Donor\_Campaign Table

A screenshot of a computer

Description automatically generated

Load Data in Table:

A screenshot of a computer

Description automatically generated

#### 3.1.2.8. Campaign\_Resource Table

A screenshot of a computer

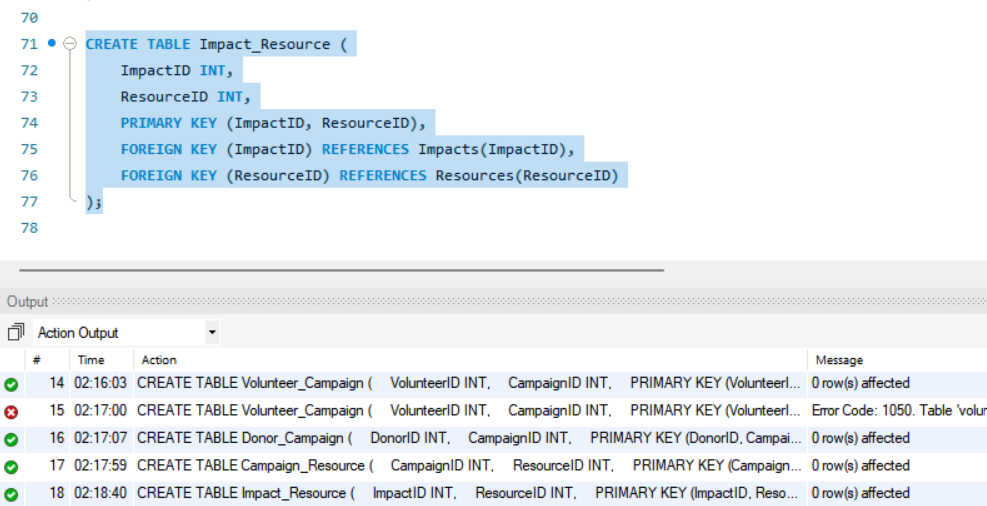
Description automatically generated

Load Data in Table:

A screenshot of a computer

Description automatically generated

#### 3.1.2.9. Impact\_Resource Table

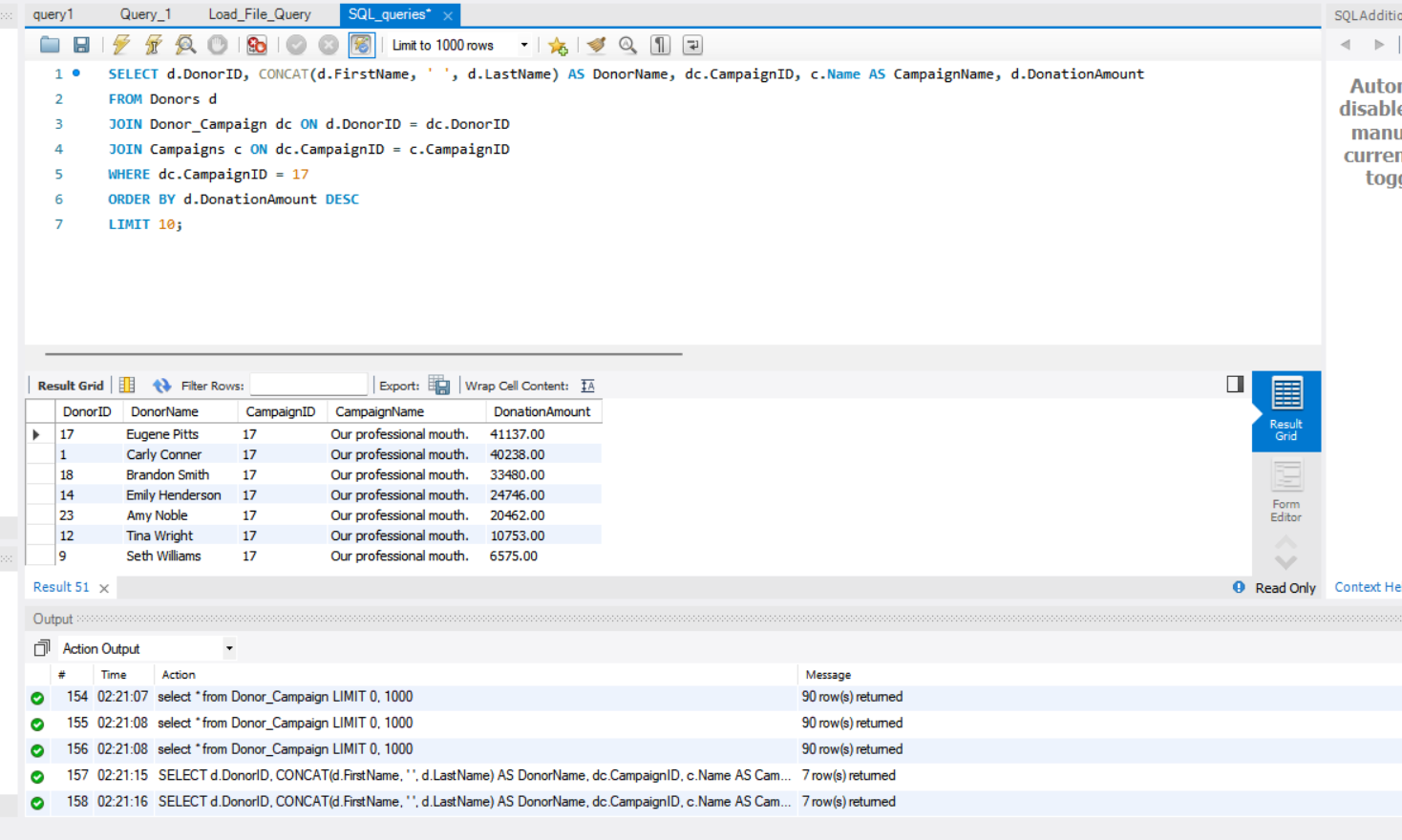


Load Data in Table:

A screenshot of a computer

Description automatically generated

## Retrieve the top 10 donors who have made the largest donations in a specific campaign.

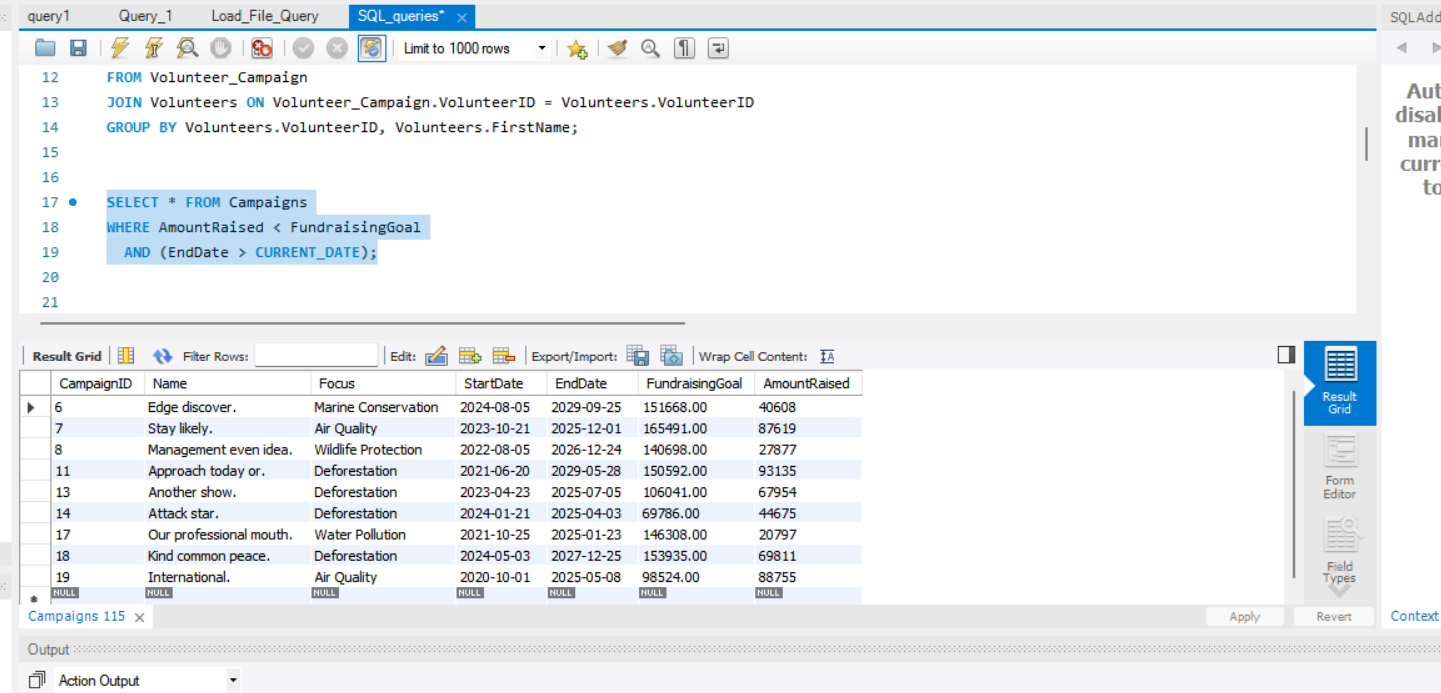


## Retrieve the number of campaigns each volunteer has participated in.

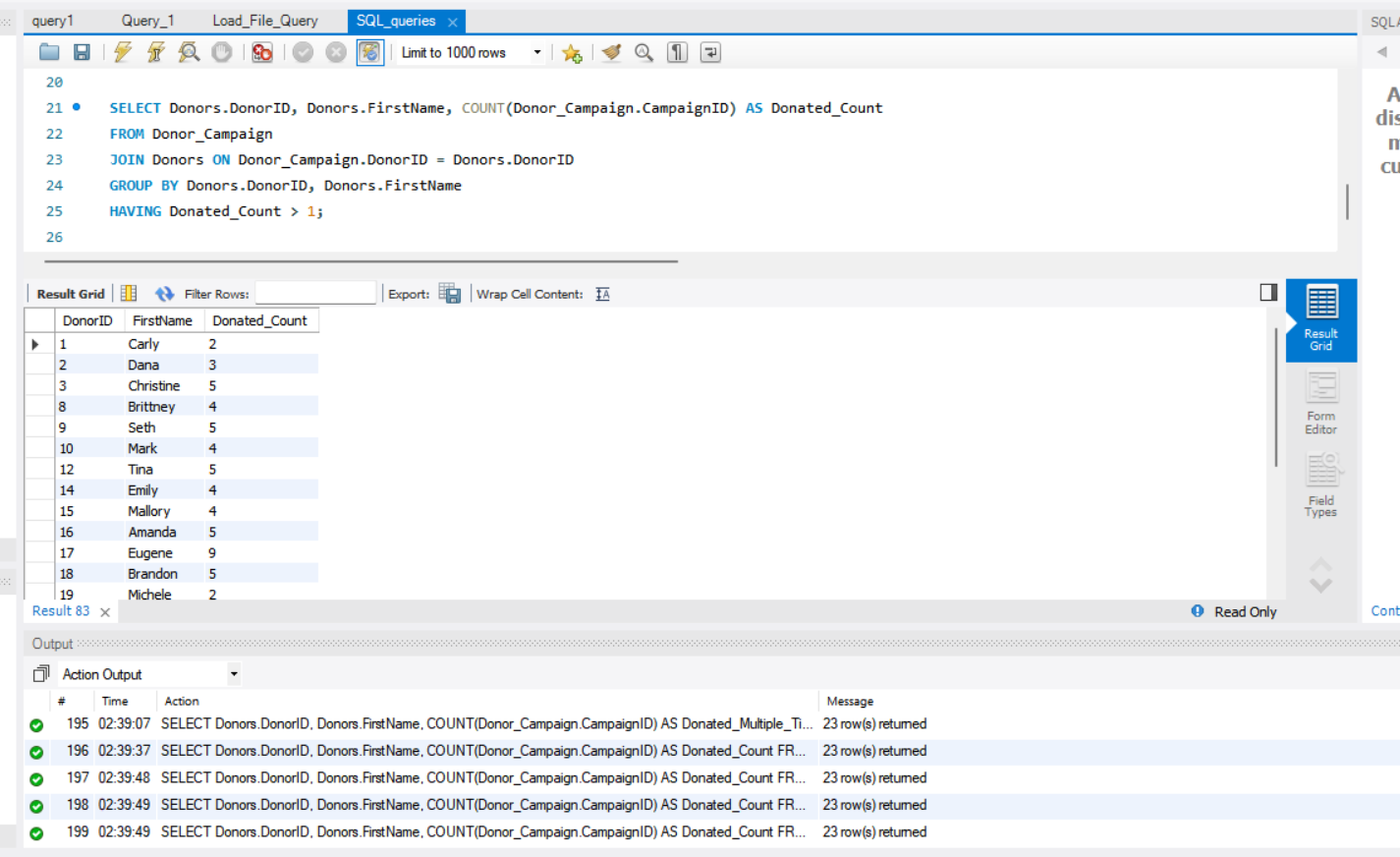
A screenshot of a computer

Description automatically generated

## Retrieve the campaigns that are still accepting donations.



## Retrieve the donors who have donated to multiple campaigns.



## Retrieve the average donation amount per donor across all campaigns they have donated to.

A screenshot of a computer

Description automatically generated

## Retrieve the number of campaigns that have been affected by each climate impact.

A screenshot of a computer

Description automatically generated

## Retrieve the volunteers who have participated in campaigns that have raised more than a certain amount

A screenshot of a computer

Description automatically generated

## Retrieve the campaigns that have not yet met their fundraising goals and have less than a certain number of days remaining until the end date.



## Retrieve the top 10 resources that have the greatest impact on climate change, based on the number of campaigns they have been associated with.

A screenshot of a computer

Description automatically generated

## “Find the top 3 campaigns with the highest amount raised, along with the number of volunteers and donors who contributed to each campaign. Include the campaign name, amount raised, total number of volunteers, and total number of donors in the result. Only include campaigns that have both volunteers and donors. Sort the result by the total number of volunteers in descending order.”

A screenshot of a computer

Description automatically generated

# Database Security and ethical issues:

The Greenpeace NGO database must address security and ethical concerns in order to protect personal data and to ensure responsible data practices. We will be dealing with sensitive information about donors, volunteers, and campaign activities, so privacy, security, and ethical data handling must be prioritized and in a robust manner.

## 4.1. Ethical

One of the important ethical challenges is data privacy. Details such as the names of donors and volunteers, their contact information and their giving history need protection from improper disclosure. One of the practices of data privacy is that only authorized users can access personal data. Another important thing about accuracy; fake information might hurt the NGO way of working and lose trust. The only way to avoid this is by routinely reviewing your data to ensure that all the correct procedures are in place to fix the errors that arise in the system. Lastly, data subjects should be informed transparently. Donors must know how their data is being captured and stored, and in what manner it is used to further campaign objectives.

## 4.2. Security

It is required to take certain security measures(in order to protect your data) Access control must limit data access, depending on the roles of the users, allowing the access to data to just one who needs it to perform their task. Encryption provides additional security, even if data is captured, it is not intelligible to third parties. It is another crucial practice to create backups regularly to protect data loss. Backups allow us to restore data in the event it is lost in security incidents or deletion that were committed, providing continuity and availability of data.

## 4.3. Storing and maintaining data

Data minimization is key to efficient storage. In essence, retain as little data as possible to avoid unnecessary risk. Third, a data retention policy also serves to indicate how long data is stored (especially personal information). By regularly reviewing and purging data that is out-dated they mitigate the risk of having to process useless or too much data.

## 4.4. Recommendations

DB security is set up by implementing role-based access control (RBAC) that allows each user to access data according to his role. We strongly recommend audit logs to keep records of data access and modifications to trace and detect unauthorized operations. Ensure regular employee training on best practices for data security (eg, recognizing phishing attacks), as the majority of breaches occur due to human error.

## 4.5. Measure (incl. specific Azure security tools/features) to protect data against security threats and handle data ethically.

Azure Active Directory is great, for ensuring user authentication security and controlling access based on roles; Azure Key Vault is perfect for storing data like encryption keys and passwords; and Azure Security Center offers a thorough security summary, with alerts to detect risks and ensure a safe database environment.

By implementing security protocols the NGO can safeguard data integrity preserve confidentiality and establish credibility, with stakeholders.

# Presentation: Using Zoom

# Conclusion

This project was effective in developing a comprehensive database for Greenpeace NGO, with a focus on securely storing and managing donors, volunteer, and campaign data. By creating an effective schema and implementing critical SQL queries, the database facilitates data retrieval, analysis, and reporting, all of which are required for informed decision-making. Ethical and security procedures were also prioritized, ensuring that data is accurate, private, and secure. Future enhancements may include increased data visualization and deeper interaction with Azure security technologies to improve data integrity and accessibility.

# References: