

| Political Campaign  **SUBJECT AREAS** |
| --- |
| **Logo / Image** |

**Contents**

[1 Political campaign Description 3](#_heading=h.1fob9te)

[1.1 Political campaign background 3](#_heading=h.3znysh7)

[1.2 Problems. Current Situation 3](#_heading=h.2et92p0)

[1.3 The Benefits of implementing a database. Project Vision 3](#_heading=h.tyjcwt)

[2 Model description 3](#_heading=h.3dy6vkm)

[2.1 Definitions & Acronyms 3](#_heading=h.1t3h5sf)

[2.2 Logical Scheme 3](#_heading=h.4d34og8)

[2.3 Objects 3](#_heading=h.2s8eyo1)

# 

# Political Campaign Description

## Political campaign background

The aim of this political campaign is to methodically plan and coordinate the complex elements of election activities. These activities include event planning, voter contact, volunteer management, research, and fundraising. The infrastructure of the campaign is carefully crafted to effectively manage social media interactions, surveys, and financial transactions, thus optimizing voter engagement and support.

## Problems. Current Situation

Political campaigns today frequently encounter problems of data fragmentation, poor communication, and lack of coordination among volunteers, voters, and campaign staff. Using hand tracking or outdated systems to manage campaign events, voter engagement, and money transfers most frequently leads to errors, mismanagement, and even loss of critical information. Without a centralized database, accessing critical information is a slow and laborious process.

## The Benefits of implementing a database. Project Vision

The application of a comprehensive and well-tabulated database in the political campaign offers numerous advantages:

*Efficient Volunteer Managemen*t: Application of dedicated volunteer tables enables easy assignment tracking, contact information, and history of involvement.

*Organization of Campaign Events*: An organized process of campaign event planning and budgeting enhances execution and monitoring of attendance.

*Data-Driven Decision Making*: Systematic storage of research outcomes and survey data of voters helps in strategic campaign planning.

*Successful Voter Contact*: Maintaining thorough records of voter demography and political orientation makes it possible to tailor appropriate communication.

*Social Media Impact Analysis*: Following social media engagement metrics in order to continually refine online outreach strategy.

*Financial Transparency*: Accurate accounting for all campaign fiscal transactions ensures accountability and proper use of funds.

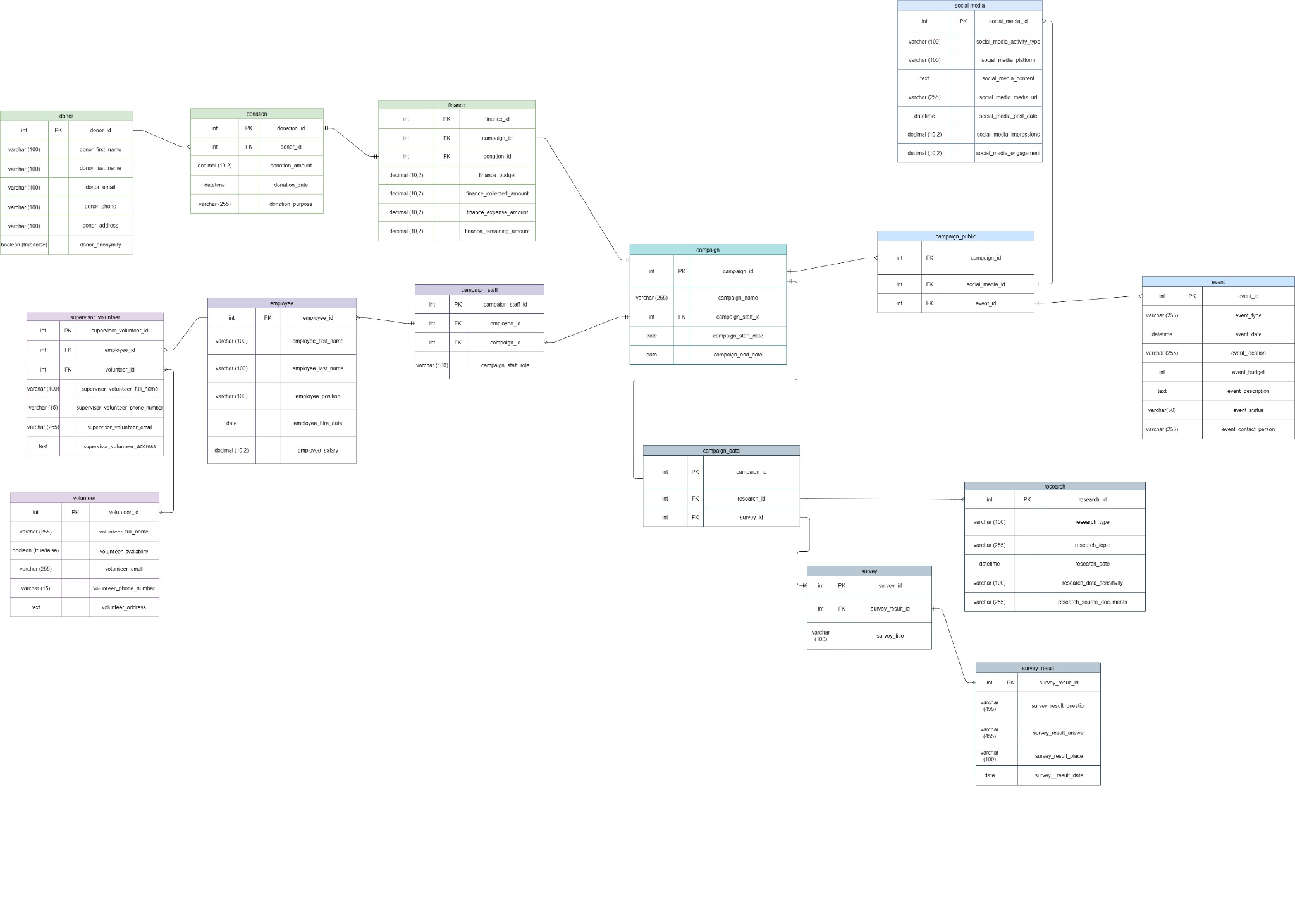
The project proposes designing an efficient, data-driven political campaign to enhance operational efficiency, voter engagement, and overall campaign success. By implementing a comprehensive database, the campaign aims to streamline processes and improve communication. This system is intended to provide a strategic advantage in managing campaign activities, optimizing resource utilization, and ultimately increasing voter support and campaign effectiveness.

# Model description

## Definitions & Acronyms

none

## Logical Scheme



## Objects

The database schema manages political campaigns with tables for: campaigns (core campaign data), donation (financial contributions), volunteers (supporter information), social\_media (platform presence), campaign\_staff (employee assignments), campaign\_events (organized activities), finance (budget tracking), research\_resources (campaign research), campaign\_survey (feedback collection), tasks (action items), voter (demographic data), voter\_political\_view (political preferences), and employee (staff details). This comprehensive system effectively tracks everything from finances and personnel to voter information and campaign activities, allowing campaign managers to coordinate resources, monitor progress, and make data-driven decisions.

| Table Name | Field name | Field Description | Data Type |
| --- | --- | --- | --- |
| campaign | campaign\_id | Unique identifier for the campaign, can not be null, PK | int |
| campaign\_name | The name of the campaign, can not be NULL | varchar (255) |
|  | campaign\_staff\_id | FK, shows relation to the campaign staff table | int |
|  | campaign\_start\_date | Date when campaign started | date |
|  | campaign\_end\_date | date when campaign ended | date |

Comments on table relationships

**One-to-many** relationship between:

campaign (campaign\_id) and campaign\_staff (campaign\_id) - many employees or staff (groups for different purpose) can work for the specific campaign

campaign (campaign\_id) and campaign\_public\_appearance (campaign\_id) - usually politics make multiple appearances in the public during events, so it is important to track events for one campaign. In order to make this tracking simpler, a campaign public appearance table was added to combine social media and events.

campaign (campaign\_id) and campaign\_data (campaign\_id) - it is possible to do one research for campaign needs, but in order to keep track of changing political environment, there is need to do multiple research and surveys for one campaign in order to make changes in campaign decision-making. In order to make it easier to find collected data, a bridge table was created.

**One-to-one** relationship between:

finance (campaign\_id) and (campaign\_id) - campaign has a one budget (finance), maybe a bank account which is used for this one campaign needs.

Example with data

| campaign\_id | campaign\_name | campaign\_staff\_id | campaign\_start\_date | campaign\_end\_date |
| --- | --- | --- | --- | --- |
| 568 | Go, Greens, Go! | 004512 | 2021-04-01 | 2021-07-01 |
| 16578 | Let’s win | 007852 | 2020-03-08 | 2020-06-01 |

| Table Name | Field name | Field Description | Data Type |
| --- | --- | --- | --- |
| campaign\_public | campaign\_id | Unique identifier for the campaign, can not be null, FK | int |
|  | social\_media\_id | Unique identifier for the social media table, can not be null, FK | int |
|  | event\_id | Unique identifier for the event table, can not be null, FK | int |

Comments on table relationships

**One-to-many** relationship between:

campaign (campaign\_id) and campaign\_public (campaign\_id) - the table helps separate data for social networks and events from the overall campaign table. Also that means that a campaign can be associated with multiple entries in the campaign\_public table. This means that a single campaign can have multiple events and multiple social media activities, which are linked through the campaign\_public table.

campaign\_public (social\_media\_id) and social media (social\_media\_id) - a social media post can be linked to multiple entries in the campaign\_public table

campaign\_public (event\_id) and event ( event\_id) - reference to events, that means that **event** can also be associated with multiple entries in the campaign\_public table.

Example with data

| campaign\_id | social\_media\_id | event\_id |
| --- | --- | --- |
| Go, Greens, Go! | 004512 | 2021-04-01 |
| Let’s win | 007852 | 2020-03-08 |

| Table Name | Field name | Field Description | Data Type |
| --- | --- | --- | --- |
| event | event\_id | Unique identifier for each event, PK | int |
|  | event\_type | Type or category of the event, NOT NULL | varchar (255) |
|  | event\_date | Scheduled date and time of the event, Not NULL | datetime |
|  | event\_location | Venue or location where the event will take place | varchar (255) |
|  | event\_budget | Allocated budget for the event, Check (>= 0) | int |
|  | event\_description | Detailed information about the event | text |
|  | event\_status | Current status of the event (e.g., scheduled, canceled) | varchar(50) |
|  | event\_contact\_person | Contact person responsible for the event | varchar (255) |

Comments on table relationships

**One-to-many** relationship between:

campaign\_public (event\_id) and event ( event\_id) - reference to events, that means that **event** can also be associated with multiple entries in the campaign\_public table.

Example with data

| campaign\_event\_id | campaign\_id | campaign\_event\_type | campaign\_event\_date |
| --- | --- | --- | --- |
| 526 | 005 | Rally | 2025-04-26 10:30:00 |
| 8294 | 005 | Debate | 2025-08-02 14:00:00 |

| campaign\_event\_location | campaign\_event\_budget | campaign\_event\_description | campaign\_event\_status |
| --- | --- | --- | --- |
| Sport Hall | 5000 | Major campaign rally featuring the candidate and several celebrity endorsers | Scheduled |
| VMU Auditorium | 300 | Public debate between candidates on key issues | Scheduled |

| campaign\_event\_contact\_person |
| --- |
| Jonas Jonaitis |
| Petras Petraitis |

| Table Name | Field name | Field Description | Data Type |
| --- | --- | --- | --- |
| social media | social\_media\_id | Unique identifier for social media activity | int |
| campaign\_id | Links post to campaign, Foreign Key referencing campaign | int |
|  | social\_media\_activity\_type | Type of post,Not Null | varchar (100) |
|  | social\_media\_platform | Social media platform, Not Null | varchar (100) |
|  | social\_media\_content | Post content, Not Null | text |
|  | social\_media\_media\_url | Link to media | varchar (255) |
|  | social\_media\_post\_date | Date of post, Not Null | datetime |
|  | social\_media\_impressions | Number of people reached, Check (>= 0) | decimal (10,2) |
|  | social\_media\_engagement | User engagement count, Check (>= 0) | decimal (10,2) |

Comments on table relationships

**one-to-many**

campaign\_public (social\_media\_id) and social media (social\_media\_id) - a social media post can be linked to multiple entries in the campaign\_public table,

Example with data

| social\_media\_id | campaign\_id | social\_media\_activity\_type | social\_media\_platform |
| --- | --- | --- | --- |
| 52 | 895 | reels | instagram |
| 98 | 923 | post | facebook |

| social\_media\_content | social\_media\_media\_url | social\_media\_post\_date | social\_media\_impressions |
| --- | --- | --- | --- |
| about new deal with opponent | www.facebook/reels/politics | 2023-05-01 06:00 | 13025 |
| about meeting with voters | www.instagram/reels/meeting/hall/456789 | 2023-05-05 08:40 | 18632 |

| social\_media\_engagement |
| --- |
| 89 |
| 45 |

| Table Name | Field name | Field Description | Data Type |
| --- | --- | --- | --- |
| campaign\_staff | campaign\_staff\_id | Unique identifier for the campaign staff, can not be null, PK | int |
| employee\_id | Unique identifier for the campaign staff, can not be null, PK | int |
|  | campaign\_id | Unique identifier for the campaign staff, can not be null, PK | int |
|  | campaign\_staff\_role | shows what role the team performs | varchar (100) |

Comments on table relationships

**One-to-many** relationship between:

campaign\_staff (employee\_id) and employee (employee\_id) - many employees work for the one many employees work on one campaign team.

campaign (campaign\_id) and campaign\_staff (campaign\_id) - many employees or staff (groups for different purpose) can work for the specific campaign

Example with data

| campaign\_staff\_id | employee\_id | campaign\_id | campaign\_staff\_role |
| --- | --- | --- | --- |
| 98 | 19684 | 251 | Media |
| 41 | 2861 | 93 | Supervisor |

| Table Name | Field name | Field Description | Data Type |
| --- | --- | --- | --- |
| employee | employee\_id | Unique identifier for each employee, PK | int |
|  | employee\_first\_name | First name of the employee, Not Null | varchar (100) |
|  | employee\_last\_name | Last name of the employee, Not Null | varchar (100) |
|  | employee\_position | Job position of the employee, Not Null | varchar (100) |
|  | employee\_hire\_date | Date of hiring, Not Null | date |
|  | employee\_salary | Employee’s salary, Check (>= 0) | decimal (10,2) |

Comments on table relationships

**One-to-many** relationship

campaign\_staff (employee\_id) and employee (employee\_id) - many employees work for the one many employees work on one campaign team.

Example with data

| employee\_id | employee\_first\_name | employee\_last\_name |
| --- | --- | --- |
| 19684 | Pranas | Pranaitis |
| 2861 | Lukas | Lukaitis |

| employee\_position | employee\_hire\_date | employee\_salary |
| --- | --- | --- |
| team lead | 2020-05-04 | 3000 |
| ceo | 2019-01-20 | 5200 |

| Table Name | Field name | Field Description | Data Type |
| --- | --- | --- | --- |
| supervisor\_volunteer | supervisor\_volunteer\_id | Unique identifier for volunteer supervisor, PK | int |
| employee\_id | Links supervisor to a employee table, FK, not null | int |
|  | volunteer\_id | Links supervisor to a volunteer, Foreign Key referencing volunteers | int |
|  | supervisor\_volunteer\_full\_name | Supervisor's full name, Not Null | varchar (100) |
|  | supervisor\_volunteer\_phone\_number | Contact number, Unique, Not Null | varchar (15) |
|  | supervisor\_volunteer\_email | Email address, Unique, Not Null | varchar (255) |
|  | supervisor\_volunteer\_address | Residential address | text |

Comments on table relationships

**one-to-many**

supervisor\_volunteer (employee\_id) and employee\_id (employee\_id) - one employee can supervise many volunteers, meaning an employee (likely in a supervisor role) can have multiple entries in the supervisor\_volunteer table

**many-to-many**

supervisor\_volunteer (volunteer\_id) and volunteer (volunteer\_id) - many supervisors can supervise many volunteers at the time, it depends on the number of supervisors.

Example with data

| supervisor\_volunteer\_id | campaign\_staff\_id | volunteer\_id | supervisor\_volunteer\_full\_name |
| --- | --- | --- | --- |
| 89 | 4895641 | 1651 | Linas Linaitis |
| 5 | 61 | 52 | Petras Petraitis |

| supervisor\_volunteer\_phone\_number | supervisor\_volunteer\_email | supervisor\_volunteer\_address |
| --- | --- | --- |
| 3701111111 | supervisor@gmail.com | Putvinskis street 56, Kaunas, Lithuania |
| 4508211684 | campaignvolunter@yahoo.com | Post street 89, Vilnius, Lithuania |

| Table Name | Field name | Field Description | Data Type |
| --- | --- | --- | --- |
| volunteer | volunteer\_id | Unique identifier for each volunteer, PK | Int |
| volunteer\_full\_name | Volunteer’s full name, Not Null | varchar (255) |
|  | volunteer\_availability | Indicates if the volunteer is available, Default: False | Boolean (true/false) |
|  | volunteer\_email | Contact email, unique, Not Null | varchar (255) |
|  | volunteer\_phone\_number | Contact phone number, unique, Not Null | varchar (15) |
|  | volunteer\_address | Volunteer’s address | text |

Comments on table relationships

**many-to-many** relationship

volunteers (volunteer\_id) and supervisor\_volunteer (volunteer\_id) - one volunteer can work in different teams with different supervisors and one supervisor can supervise many volunteers.

Example with data

| volunteer\_id | volunteer\_full\_name | volunteer\_availability | volunteer\_email |
| --- | --- | --- | --- |
| 125 | Robert Johnson | yes | robert.johnson@email.com |
| 63 | John Smith | no | john.smith@email.com |

| volunteer\_phone\_number | volunteer\_address |
| --- | --- |
| 451111208 | 456 Oak Ave, Apt 2B, Springfield, IL 62702 |
| 458928582 | 101 Pine Ln, Springfield, IL 62704 |

| Table Name | Field name | Field Description | Data Type |
| --- | --- | --- | --- |
| donation | donation\_id | Unique identifier for each donation, PK | int |
|  | donor\_id | Unique identifier for each donor, FK,not Null, links to donor table | int |
|  | donation\_amount | Amount donated, Check (>= 0) | decimal (10,2) |
|  | donation\_date | Date of donation, Not Null | datetime |
|  | donation\_purpose | Purpose of the donation | varchar (255) |

Comments on table relationships

One-to-one relationship:

donation (donotion\_id) and finance (donotion\_id) - ensures every donation showed in the finance table is uniquely identifiable for financial reporting

One-to-many

donation (donor\_id) and donor (donor\_id) - the same donor can make multiple donations

Example with data

| donation\_id | donor\_id | donation\_amount |
| --- | --- | --- |
| 128520 | 684712 | 350 |
| 1820645 | 984651 | 405 |

| donation\_date | donation\_purpose |
| --- | --- |
| 2025-02-02 | for the Mr. Smith |
| 2025-03-10 | to support this campaign |

| Table Name | Field name | Field Description | Data Type |
| --- | --- | --- | --- |
| donor | donor\_id | Unique identifier for each donor, PK | int |
| donor\_first\_name | Donor’s first name, not null | varchar (100) |
|  | donor\_last\_name | Donor’s last name, unique,not null | varchar (100) |
|  | donor\_email | Email address, Unique, Not Null | varchar (255) |
|  | donor\_phone | Donor’s phone number, unique, not null | varchar (255) |
|  | donor\_address | Donor’s address,unique, not null | varchar (255) |
|  | donor\_anonymity | To check if donor wants to be anonymous in the public lists | boolean (true/false) |

Comments on the table relationships

One-to-many relationship:

donation (donor\_id) and donor (donor\_id) - the same donor can make mane donations to the campaign

Example with data

| donor\_id | donor\_first\_name | donor\_last\_name | donor\_email |
| --- | --- | --- | --- |
| 191621 | Lukas | Lukaitis | lukaslukas@gmail.com |
| 9854149 | Tina | Marinaite | tinm@gmail.com |

| donor\_phone | donor\_address | donor\_anonymity |
| --- | --- | --- |
| +3706519231 | 707 Apple Ave, Apt 96D, Springfield, UT 8716 | Yes |
| +42089955 | 502 Mango St., Apt 3C, Washington D.C., 9824 | No |

| Table Name | Field name | Field Description | Data Type |
| --- | --- | --- | --- |
| finance | finance\_id | Unique identifier for finance records, PK | Int |
| campaign\_id | Links finance records to a campaign, FK, not null, unique | int |
| donation\_id | Links finance records to a donation table, FK, not null, unique | int |
|  | finance\_collected\_amount | Funds collected, Check (>= 0) | decimal (10,2) |
|  | finance\_expense\_amount | Total expenses, Check (>= 0) | decimal (10,2) |
|  | finance\_remaining\_amount | Remaining funds | decimal (10,2) |

Comments on table relationships

One-to-one relationship:

finance (campaign\_id) and campaign (campaign\_id) - one campaign has one budget (finance).

donation (donotion\_id) and finance (donotion\_id) - ensures every donation showed in the finance table is uniquely identifiable for financial reporting

Example with data

| finance\_id | campaign\_id | donation\_id | finance\_budget | finance\_collected\_amount |
| --- | --- | --- | --- | --- |
| 1743 | 2964 | 952491 | 860 000 | 50 000 |
| 164 | 129681 | 2988982 | 30 000 | 67 800 |

| finance\_expense\_amount | finance\_remaining\_amount |
| --- | --- |
| 30 800 | 879 200 |
| 41 000 | 56800 |

| Table Name | Field name | Field Description | Data Type |
| --- | --- | --- | --- |
| campaign\_data | campaign\_id | Links survey to campaign table, FK, Unique identifier | int |
|  | research\_id | Links to the campaign survey result table, FK, Unique identifier | int |
|  | survey\_id | Survey title, Not Null | int |

**One-to-one**

campaign\_data (campaign\_id) and campaign (campaign\_id) - one campaign has one database for a research purpose. This table links to a scientific research table and conducted survey table.

**One-to-many**

campaign\_data (research\_id) and research (research\_id) - many entries can be in the research table and link to one campaign data table. The campaign staff can conduct a lot of research for one campaign.

campaign\_data (survey\_id) and survey (survey\_id) - many entries or surveys can be conducted and linked to one data table.

Example with data

| campaign\_id | research\_id | survey\_id |
| --- | --- | --- |
| 18 | 852 | survey |
| 5 | 1520 | survey |

| Table Name | Field name | Field Description | Data Type |
| --- | --- | --- | --- |
| research | research\_id | Unique identifier for research, PK | int |
|  | research\_type | Type of research, Not Null | varchar (100) |
|  | research\_topic | Research focus area, Not Null | varchar (255) |
|  | research\_collection\_date | Date data was collected, Not Null | datetime |
|  | research\_data\_sensitivity | Sensitivity level of data, Not Null | Boolean (true/false) |
|  | research\_source\_documents | Sources used | varchar (255) |

Comments on table relationships

one-to-many

campaign\_data (research\_id) and research (research\_id) - many entries can be in the research table and link to one campaign data table. The campaign staff can conduct a lot of research for one campaign.

Example with data

| research\_campaign\_id | research\_type | research\_topic |
| --- | --- | --- |
| 18 | survey | opponents |
| 5 | survey | opinion about green deal |

| research\_collection\_date | research\_data\_sensitivity | research\_source\_documents |
| --- | --- | --- |
| 2024-05-18 10:12 | not sensitive | wikipedia |
| 2021-06-03 17:03 | not sensitive | surveys collected from the voters directly |

| Table Name | Field name | Field Description | Data Type |
| --- | --- | --- | --- |
| survey | survey\_id | Unique survey identifier, PK | int |
|  | survey\_result\_id | Links to the campaign survey result table, FK, Unique identifier | int |
|  | survey\_title | Survey title, Not Null | varchar (100) |

Comments on table relationships

One-to-many

survey (survey\_id) and campaign data (survey\_id) - many surveys can be done in order to research opinion or other matters for the campaign.

survey (survey\_result\_id) and survey\_result (survey\_result\_id) - it is possible to get many results for multiple surveys, so all results can be under one survey\_id.

Example with data

| survey\_id | survey\_result\_id | survey\_title |
| --- | --- | --- |
| 5962 | 178416 | The most liked politicians |
| 98 | 63125 | Opinion about city plans to build a new mall |

| Table Name | Field name | Field Description | Data Type |
| --- | --- | --- | --- |
| survey\_result | survey\_result\_id | Unique survey identifier, PK | int |
| survey\_result\_question | Links survey to campaign table, FK, Unique identifier | varchar (455) |
|  | survey\_result\_answer | Links to the campaign survey result table, FK, Unique identifier | varchar (455) |
|  | survey\_result\_place | This is the place where the survey was conducted, not null | varchar (100) |
|  | survey\_result\_date | Date when data was collected, not null | date |

Comments on table relationships

**One-to-many**

survey (survey\_result\_id) and survey\_result (survey\_result\_id) - it is possible to get many results for multiple surveys, so all results can be under one survey\_id.

Example with data

| survey\_result\_id | survey\_result\_question | survey\_result\_answer | survey\_result\_place |
| --- | --- | --- | --- |
| 178416 | Which politician do you like the most? | I really love Mr. Wallace, he is great! | Kaunas, Akropolis |
| 63125 | What do you think about the city's plans to build a new store? | I think we do not need another mall, it is really necessary. | Vilnius, Gediminas St. |

| survey\_result\_date |
| --- |
| 2025-03-03 |
| 2025-03-08 |