

|  |
| --- |
| Business Template  Political Campaign |
| **Logo / Image** |

Contents

[1 Business Description 3](#_Toc182195081)

[1.1 Business background 3](#_Toc182195082)

[1.2 Problems. Current Situation 3](#_Toc182195083)

[1.3 the Benefits of implementing a database. Project Vision 3](#_Toc182195084)

[2 Model description 3](#_Toc182195085)

[2.1 Definitions & Acronyms 3](#_Toc182195086)

[2.2 Logical Scheme 3](#_Toc182195087)

[2.3 Objects 4](#_Toc182195088)

[2.4 Database Structure Changes 14](#_Toc182195089)

# 

# Business Description

## Business background

The following business model describes a database for managing political campaign data.

The key functionalities include managing voter demographics, tracking contributions from donors, organizing campaign events, assigning volunteers to tasks, and storing candidate information. By linking entities such as campaign activities, various forms of contributions with related parties, the campaign team will be able to monitor the process.

## Problems. Current Situation

Potential issues/limitations that may arise in the future as the database grows are related to the scalability of the data. As the campaign grows, the volume of data will increase in entities like Voters, Contributions, Finances, and Events which can affect the performance of the database.

## the Benefits of implementing a database. Project Vision

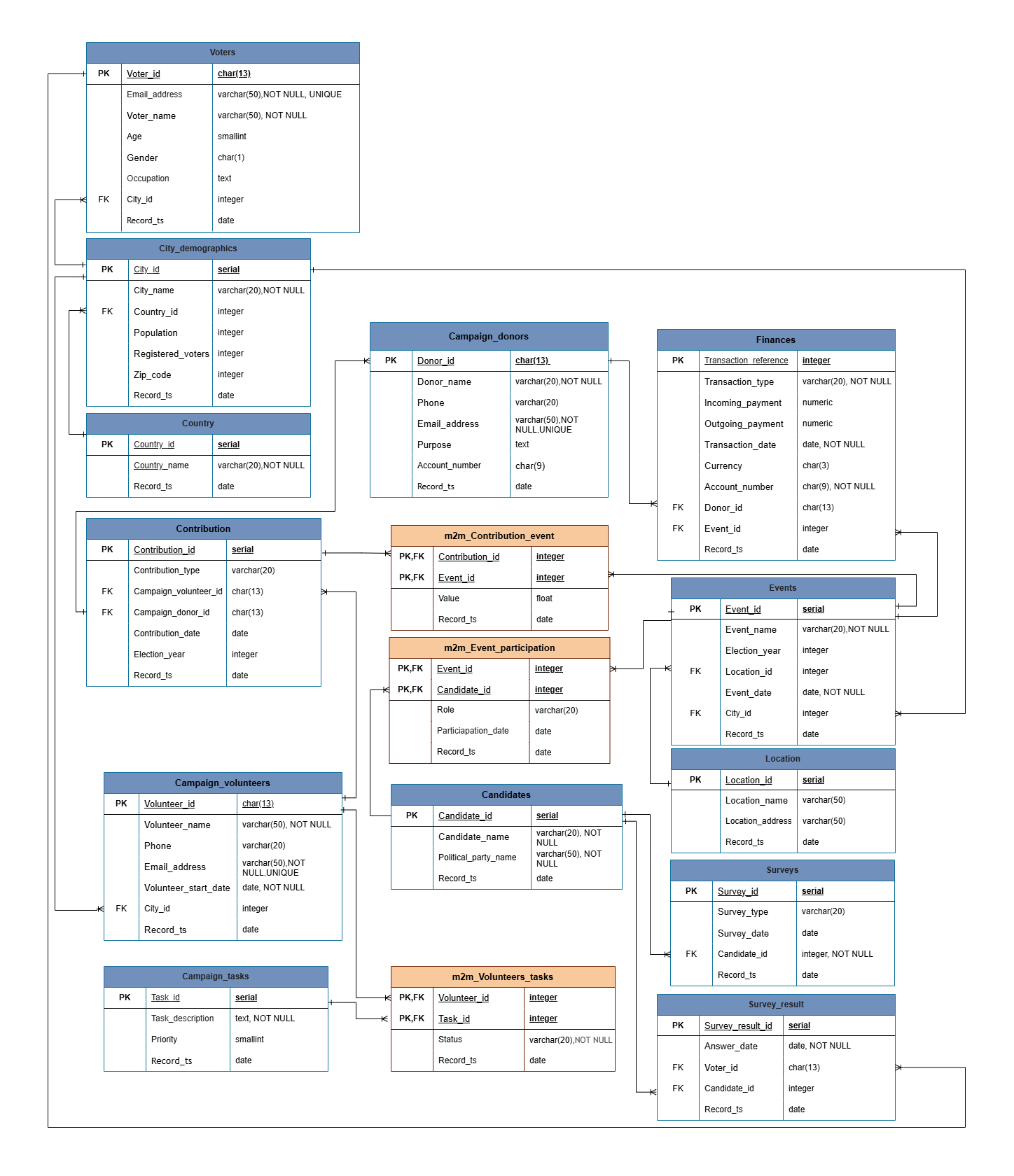
The vision of the political campaign database is to evolve throughout multiple election years and to adapt to growing campaign needs such as the need for historical data or the need for additional functionalities and activities.

# Model description

## Definitions & Acronyms

This database is designed to manage and organize the key operations of a political campaign, providing a repository for tracking voters, donations, events, volunteers, candidates, and financial transactions.  
  
*Note: Detailed entity descriptions are included in the Objects section of the documentation.*

## Logical Scheme



## Objects

1. **Voters table** – stores information about voters, including their contact details and associated city.

***One-to-Many relationship***: One voter can participate in multiple surveys via Voter\_id however each Survey\_result refers to only one voter.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Voters | Voter\_id | **PK**, unique identifier for each voter, natural primary key | char(13) |
| Email\_address | voter's email address, unique, NOT NULL,UNIQUE, CHECK: '%\_@\_\_%.\_\_%' | varchar(50) |
| Voter\_Name | name of the voter, NOT NULL | varchar(50) |
| Age | voter’s age | smallint |
| Gender | voter’s gender, (CHECK: ‘F’,’M’,’O’) | char(1) |
| Occupation | voter's occupation | text |
| City\_id | **FK** connecting to the City\_Demographics table | integer |
| Record\_ts | Default current\_date | date |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Voter\_id | Email\_address | Voter\_Name | Age | Gender | Occupation | City\_id |
| 0510992823100 | James\_Smith@gmail.com | James Smith | 40 | M | Professor | 2 |

1. **Campaign\_volunteers table** - tracks details of volunteers and their assigned tasks.

***Many-to-Many relationship:*** the m2m\_Volunteers\_Tasks table connects multiple volunteers to multiple tasks meaning that a volunteer can work on multiple tasks, and one task can have multiple volunteers.  
***One-to-Many relationship:*** one volunteer can have many contributions, but each contribution is linked to one volunteer via the Contributor\_id.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Campaign\_volunteers | Volunteer\_id | PK,unique identifier for each volunteer, natural primary key | char(13) |
| Volunteer\_name | name of the volunteer, NOT NULL | varchar(50) |
| Phone | volunteer’s phone number | VARCHAR(20) |
| Email\_address | volunteer’s email address. NOT NULL, UNIQUE, CHECK: '%\_@\_\_%.\_\_%' | varchar(50) |
| Volunteer\_start\_date | date when the volunteer joined the political campaign, NOT NULL, CHECK: date has to be greater than January 1, 2000 | date |
| City\_id | FK referring to the City\_Demographics table | integer |
| Record\_ts | Default current\_date | date |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Volunteer\_id | Volunteer\_name | Phone | Email\_address | Volunteer\_start\_date | City\_id |
| 0210996826300 | John Williams | 812546 | JW@gmail.com | 2024-01-01 | 4 |

1. **Campaign\_donors table** - stores donor information and their designated contribution purpose.

***One-to-Many relationship:*** A donor via Donor\_id can be linked to many financial transactions, but each transaction is linked to one donor.  
***One-to-Many relationship:*** One donor can have many contributions, but each contribution is linked to one donor through the Contributor\_id.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Campaign\_donors | Donor\_id | PK,unique identifier for each donor, natural primary key | char(13) |
| Donor\_name | name of the donor, NOT NULL | varchar(20) |
| Phone | donor’s phone number | varchar(20) |
| Email\_address | donor’s email address, NOT NULL, UNIQUE, CHECK: '%\_@\_\_%.\_\_%' | varchar(50) |
| Purpose | purpose of the donation | text |
| Account\_number | donor’s account number | char(9) |
| Record\_ts | Default current\_date | date |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Donor\_id | Donor\_name | Phone | Email\_address | Purpose | Account\_number |
| 0811999825200 | Sophie Evans | 812613 | Evansllc@gmail.com | Sponsorship | 511234567 |

1. **Campaign\_tasks** - stores information about tasks that volunteers need to complete.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Campaign\_tasks | Task\_id | PK,unique identifier for each task, surrogate primary key | SERIAL |
| Task\_description | description of the task, NOT NULL | text |
| Priority | priority of the task on a 1-5 scale where 1 has the highest priority and 5 has the lowest priority (CHECK: 1,2,3,4,5) | smallint |
| Record\_ts | Default current\_date | date |

|  |  |  |
| --- | --- | --- |
| Task\_id | Task\_description | Priority |
| 1 | Data collection | 2 |

1. **Finances table** - tracks all financial transactions (incoming and outgoing payments) related to the campaign.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Finances | Transaction\_reference | PK,unique identifier for each financial transaction, primary key | Varchar(10) |
| Transaction\_type | type of the transaction (wire transfer, check, cash), NOT NULL | varchar(20) |
| Incoming\_payment | amount of incoming payment, CHECK value that cannot be negative | numeric |
| Outgoing\_payment | amount of outgoing payment, CHECK: value that cannot be negative | numeric |
| Transaction\_date | date of the transaction, NOT NULL, CHECK: date must be greater than January 1, 2000 | date |
| Currency | currency used for the transaction, DEFAULT ‘USD’ | char(3) |
| Account\_number | payer's account number, NOT NULL | CHAR(9) |
| Donor\_id | FK referring to the Campaign\_donors table | char(13) |
| Event\_ID | FK referring to the Events table | integer |
| Record\_ts | Default current\_date | date |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Transaction\_reference | Transaction\_type | Incoming\_payment | Outgoing\_payment | Transaction\_date | Currency | Account\_number | Donor\_id | Event\_ID |
| 123 | Wire transfer | 1000 | 0 | 2024-01-01 | USD | 5112345678 | 0811999825200 | 1 |

1. **Events table** - stores information about campaign events, such as rallies or fundraising dinners.

***Many-to-Many relationship:*** Through the table m2m\_Contribution\_event, multiple contributions can be linked to multiple events, and vice versa.  
***Many-to-Many relationship:*** The m2m\_Event\_participation table allows multiple events to be linked with multiple candidates meaning that a candidate can participate in multiple events, and one event can have multiple candidates participating.  
***One-to-Many relationship:*** multiple financial transactions can be associated with a single event, but a financial transaction can only belong to one event.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Events | Event\_ID | PK, unique identifier for each event, surrogate primary key | SERIAL |
| Event\_name | name of the event, NOT NULL | varchar(20) |
| Election\_year | year in which the political campaign took place and generated from the Event\_date | integer |
| Location\_id | FK referring to the location table | integer |
| Event\_date | date of the event, NOT NULL, CHECK: date must be greater than January 1, 2000 | date |
| City\_id | FK referring to the City\_Demographics table | integer |
| Record\_ts | Default current\_date | date |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Event\_ID | Event\_name | Election\_year | Location\_id | Event\_date | City\_id |
| 1 | Town Hall Meeting | 2024 | 1 | 2024-01-01 | 2 |

1. **Contribution table** - Tracks contributions made by any contributor (donors, volunteers) whether they were in the form of financial contributions or goods and services.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Contribution | Contribution\_id | PK,unique identifier for each contribution, surrogate primary key | SERIAL |
| Contribution\_type | type of the contribution (e.g.financial, goods, services) | varchar(20) |
| Campaign\_volunteer\_id | FK referring to the Campaign\_volunteers table | char(13) |
| Campaign\_donor\_id | FK referring to the Campaign\_donors table | char(13) |
| Contribution\_date | date of the contribution which must be greater than January 1, 2000 | date |
| Election\_year | year in which the contribution was made, generated from the Contribution\_date | integer |
| Record\_ts | Default current\_date | date |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Contribution\_id | Contribution\_type | Campaign\_volunteer\_id | Campaign\_donor\_id | Contribution\_date | Election\_year |
| 1 | Food | 0210996826300 | 0 | 2024-01-01 | 2024 |

1. **Surveys table** - stores information about surveys conducted for candidates.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Surveys | Survey\_id | PK, unique identifier for each survey, surrogate primary key | SERIAL |
| Survey\_type | type of the survey (CHECK: online, face to face, via phone, paper survey) | varchar(20) |
| Survey\_date | date when the survey was conducted which must be greater than January 1, 2000 | date |
| Candidate\_id | FK referring to the Candidates table, NOT NULL | integer |
| Record\_ts | Default current\_date | date |

|  |  |  |  |
| --- | --- | --- | --- |
| Survey\_id | Survey\_type | Survey\_date | Candidate\_id |
| 1 | online | 2024-01-01 | 1 |

1. **Survey\_result table** - stores responses from voters related to campaign surveys.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Survey\_result | Survey\_result\_id | PK,unique identifier for each survey result, surrogate primary key | SERIAL |
| Answer\_date | date when the survey answer was given, NOT NULL which must be greater than January 1, 2000 | date |
| Voter\_id | FK referring to the Voters table | char(13) |
| Candidate\_id | FK referring to the Candidates table | integer |
| Record\_ts | Default current\_date | date |

|  |  |  |  |
| --- | --- | --- | --- |
| Survey\_result\_id | Answer\_date | Voter\_id | Candidate\_id |
| 1 | 2024-01-01 | 0510992823100 | 1 |

1. **Candidates table** - stores information about political candidates participating in a campaign.

***One-to-Many relationship:*** A candidate can have many surveys related to them, but each survey is linked to one candidate.   
***One-to-Many relationship:*** One survey result belongs to only one candidate however multiple survey results can be linked to one candidate.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Candidates | Candidate\_id | PK, unique identifier for each candidate, surrogate primary key | SERIAL |
| Candidate\_name | name of the candidate, NOT NULL | varchar(20) |
| Political\_party\_name | name of the candidate’s political party, NOT NULL | varchar(50) |
| Record\_ts | Default current\_date | date |

|  |  |  |
| --- | --- | --- |
| Candidate\_id | Candidate\_name | Political\_party\_name |
| 1 | Emily Brown | Independent Party |

1. **City\_Demographics table** - stores information about cities where voters reside and the associated demographic data.  
     
   ***One-to-Many relationship:*** Each voter belongs to one city via City\_id and each city may have multiple voters.  
   ***One-to-Many relationship:*** Similarly, each volunteer can be assigned to only one city via City\_id but each city may have multiple volunteers.  
   ***One-to-Many relationship:*** An event can only take place in one city, but a city can host multiple events.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| City\_Demographics | City\_id | PK, unique identifier for each city, surrogate primary key | SERIAL |
| City\_name | name of the city, NOT NULL | varchar(20) |
| Country\_id | FK referring to the Country table | integer |
| Population | total population of the city | integer |
| Registered\_voters | number of registered voters in the city | integer |
| Zip\_code | Postal code of the city | integer |
| Record\_ts | Default current\_date | date |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| City\_id | City\_name | Country\_id | Population | Registered\_voters | Zip\_code |
| 2 | Madison | 1 | 273372 | 176000 | 53701 |

1. **m2m\_Event\_participation table** - tracks contributions made during specific events.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| m2m\_Event\_participation | Event\_id | PK, FK referring to the Events table, composite key | integer |
| Candidate\_id | PK, FK referring to the Candidates table, composite key | integer |
| Role | The role of the candidate at the event | varchar(20) |
| Particiapation\_date | Date of the candidate's participation, which must be greater than January 1, 2000 | date |
| Record\_ts | Default current\_date | date |

|  |  |  |  |
| --- | --- | --- | --- |
| Event\_id | Candidate\_id | Role | Particiapation\_date |
| 1 | 1 | Speaker | 2024-01-01 |

1. **m2m\_Contribution\_event table** - records which candidates participated in which campaign events and captures the role they played and when the participation occurred.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| m2m\_Contribution\_event | Contribution\_id | PK, FK referring to the Contribution table, surrogate key | integer |
| Event\_id | PK, FK referring to the Events table, surrogate key | integer |
| Value | Approximate value of the contribution for the event. Value that cannot be negative | numeric |
| Record\_ts | Default current\_date | date |

|  |  |  |
| --- | --- | --- |
| Contribution\_id | Event\_id | Value |
| 1 | 1 | 200 |

1. **m2m\_Volunteers\_tasks table** - tracks which volunteers are assigned to which tasks and their status in completing those tasks.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| m2m\_Volunteers\_tasks | Volunteer\_id | PK, FK referring to the Campaign\_volunteers table, natural key | char(13) |
| Task\_id | PK, FK referring to the Campaign\_tasks table, surrogate key | integer |
| Status | Status of the volunteer’s task (CHECK:Not yet started, in progress, Finished), NOT NULL | varchar(20) |
| Record\_ts | Default current\_date | date |

|  |  |  |
| --- | --- | --- |
| Volunteer\_id | Task\_id | Status |
| 1 | 1 | In progress |

1. **Country table** - stores information about the country's names

***One-to-Many relationship:*** Each city belongs to exactly one country however one country can have many cities

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Country | Country\_id | unique identifier for each country, surrogate primary key | SERIAL |
| Country\_name | the name of the country | varchar(20) |
| Record\_ts | Default current\_date | date |

|  |  |
| --- | --- |
| Country\_id | Country\_name |
| 1 | USA |

1. **Location table** - stores distinct locations where events can be held

***One-to-Many relationship:*** One location can host many events however each event occurs at exactly one location

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Location | Location\_id | unique identifier for each loaction, surrogate primary key | SERIAL |
| Location\_name | stores the location name | varchar(50) |
| Location\_address | stores the location address | varchar(50) |
| Record\_ts | Default current\_date | date |

|  |  |  |
| --- | --- | --- |
| Location\_id | Location\_name | Location\_address |
| 1 | Town Hall | 8 Campus Drive |

## Database Structure Changes – HW DDL

**Datatype modifications:**

The Voter\_id, Donor\_id and Volunteer\_id fields, initially were defined as an integer and it was changed to CHAR(13). This allows for more flexibility with identification as IDs may contain leading zeros.

The Phone and Account Number fields were changed from integer to VARCHAR. This change allows the inclusion of symbols such as dashes or country codes and preserves leading zeros, which can be lost when numbers are stored as integers.

**New 'record\_ts' column:**

Each table now includes a record\_ts column, defined as NOT NULL with the default value set to the current date. This column was added using ALTER TABLE statements, and the default value was applied to existing rows.

**New check constraints:**

Date Constraint: A check constraint was added that require that any date inserted must be greater than January 1, 2000.

Non-negative values: A constraint was added to prevent negative values to be inserted into the database.

Election year generation: In the events and contribution tables, the value in the election\_year column is calculated using the GENERATED ALWAYS AS syntax to calculate the election year based on existing date columns.  
In the events table, election\_year is generated based on the event\_date column.  
In the contribution table, election\_year is generated based on the contribution\_date column.