

**National Military Family Association**

**Database Management System**

**Project Implementation Report**

IST 659 M005 Group Ralph Parlin and Jieer Chen

**Table of Contents**

[**Problem and Scope** 3](#_Toc26282888)

[**Project Summary** 3](#_Toc26282889)

[**Problem** 3](#_Toc26282890)

[**Solution** 4](#_Toc26282891)

[**Database Schema** 5](#_Toc26282892)

[**Physical Model- Supertype/Subtype Relationship** 5](#_Toc26282893)

[**Physical Model – Entity Relationship Diagram** 8](#_Toc26282894)

[**Data Dictionary** 9](#_Toc26282895)

[*Participant* 9](#_Toc26282896)

[*Donor* 10](#_Toc26282897)

[*Subscriber* 10](#_Toc26282898)

[*Member* 10](#_Toc26282899)

[*Volunteer* 10](#_Toc26282900)

[*ScholarshipApplication* 11](#_Toc26282901)

[*ScholarshipRecipient* 11](#_Toc26282902)

[*ScholarshipRecieved* 11](#_Toc26282903)

[*Scholarship* 12](#_Toc26282904)

[*ProgramApplication* 12](#_Toc26282905)

[*ProgramRecipient* 12](#_Toc26282906)

[*ProgramRecieved* 13](#_Toc26282907)

[*Program* 13](#_Toc26282908)

[*Donate* 13](#_Toc26282909)

[*DonationCampaign* 14](#_Toc26282910)

[*EMarkingInvolvement* 14](#_Toc26282911)

[*EMarketingCampaign* 14](#_Toc26282912)

[**Structure and Query Implementation** 15](#_Toc26282913)

[**Database System Infrastructure** 15](#_Toc26282914)

[**SQL Scripts (Creating and Inserting Data)** 15](#_Toc26282915)

[**SQL Statements for Answering Major Data Questions** 27](#_Toc26282916)

[**Interface Implementation** 33](#_Toc26282917)

[**Additional Forms Not Part of the Business Questions** 33](#_Toc26282918)

[**Additional Reports Not Part of the Business Questions** 34](#_Toc26282919)

**Problem and Scope**

**Project Summary**

This project assists the Marketing department at the National Military Family Association (NMFA) establish a single Database Management System. The National Military Family Association (NMFA) is the leading 501(c)(3) non-profit association serving the families of the currently serving, veteran, retired, wounded or fallen members of the Army, Navy, Marine Corps, Air Force, and Coast Guard.

Building the public’s awareness of this non-profit is the primary job of the Marketing department. The Marketing department advances the NMFA brand through a collection of techniques and tactics, but our project is designed to help them with their primary tool: email. The Marketing department is solely responsible for sending all electronic communications for NMFA. As part of this responsibility, the Marketing department must help the Development Department connect with donors so they can be made aware of and contribute to donation campaigns. The Marketing department also assists the Programs department with touting the scholarship and family programs NMFA offers by emailing potential military family candidates.

**Problem**

Performing the support needed to the Development and Programs department requires the Marketing manager to have access to the audiences each respective department works with. However, under the current structure, each of the departments maintains their own sperate system and process for capturing, storing and managing data about the people with whom they interact.

The Programs department uses an excel spreadsheet file that includes information about people that are applying for, or have participated in, programs and scholarships offered by NMFA. The Programs department inputs information into their spreadsheet manually after downloading new records from their website’s online application intake form. Updates to their spreadsheet are made manually as required. The Programs department also maintains an excel spreadsheet of all NMFA’s volunteers across the U.S. These volunteers help host functions and administer programs and the excel spreadsheet is used to keep record of where the volunteers live and their contact information. When an event is happening near their location, the spreadsheet is used to develop email lists to that will inquire about a volunteer’s interest in supporting a local event. This spreadsheet was created manually and remains manually updated and managed.

The Development department uses a robust online cloud-based system called Raiser’s Edge to manage the data of their donors and members. It’s automated and provides sufficient querying capability for the Development department. However, this system is isolated from all other data in the organization resulting in a silo of information. The Development department extracts the data from Raiser’s Edge to an excel spreadsheet in order to provide the information with the organization. Each time the Development department needs an email sent to donors and members, they download a new spreadsheet and provide it to the marketing department.

The Marketing department uses an excel spreadsheet to manage data about the people they engage with through email marketing. They collect this data through an online form on their website and then manually add new entries into their “master-file” which includes a running list of everyone that has ever signed up to receive information from NMFA. Updates to their spreadsheet are made manually from feedback they get from email recipients. The Marketing department uses Constant Contact (a cloud-based email service provider) to deliver their email collateral, but they must manage recipient lists manually before uploading into Constant Contact. The Marketing department lacks a system that helps them segment their list and properly target recipients according to the objectives of an email campaign. They would be very challenged to segment their list to include only recipients that, for example, live in a certain state, but did not receive the latest newsletter. This would be even further challenged if they wanted to send an email to recipients that live in a certain state, and have participated in a certain program offered by NMFA as this would require the merging of the Programs and Marketing spreadsheets.

These disparate systems coupled with no consistency in data integrity presents significant challenges to the Marketing manager when conducting the necessary data merging tasks required to support the Development and Programs departments. Simply trying to consolidate or segment email lists across the departments takes hours of reconciliation when this should be near instantaneous.

**Solution**

The data integration required across the Development, Programs, and Marketing departments requires an extensive Data Lake/Data Warehouse architecture with an extract, transform, load (ETL) capability and it exceeds the scope of this project. Instead, our proposed system will focus on addressing the challenges faced by the Marketing department and will provide a proof of concept for NMFA to consider as a model for a broader database solution. Our system will result in the Marketing manager being able to manage targeted messaging for specific people in the database. It will allow them to build records for each marketing campaign they develop, as well as glean insights from marketing and donation campaign trends to assist the Development and Programs departments.

A critical first step in this process will be to ensure each department has a clear, and shared understanding of the different audiences each department works with. Furthermore, each department must subscribe to the idea that there needs to be select data consistency across each department system. These understanding will be codified in unifying business rules and process. These business rules are the perquisite needed to provide the Marketing manager a system that allows her to integrate the Development and Program department data into the more comprehensive database management system. Once complete, questions that cripple the NMFA to answer now (see “Major Data Questions” below) data, will become near effortless.

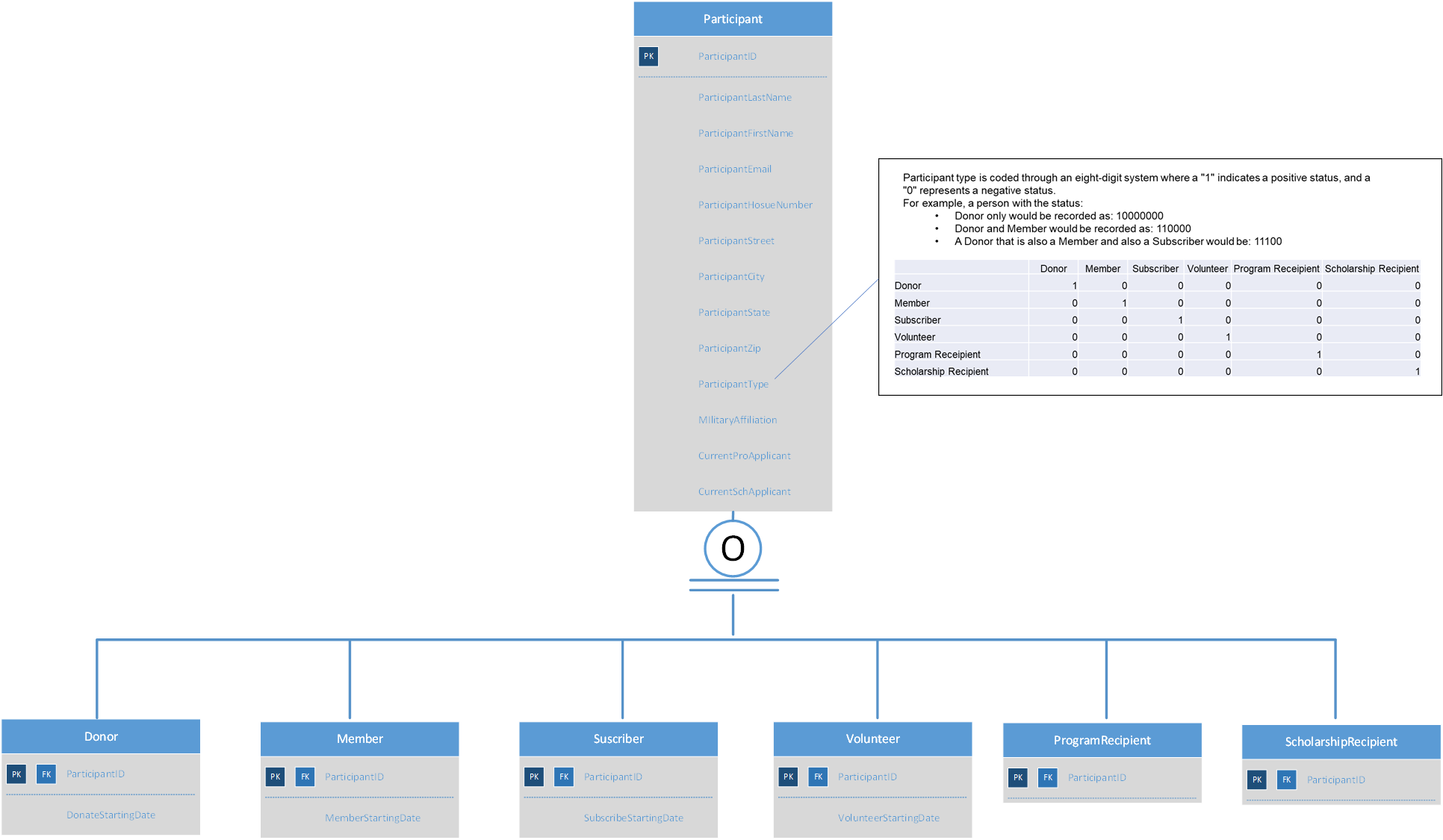
**Database Schema**

Our sponsor provided us with specific information requirements for the DBMS and we used these to develop and refine out major data questions which in turn drove the database design and entity requirements.

**Physical Model- Supertype/Subtype Relationship**

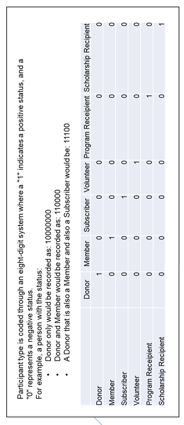
The first thing we learned was that NMFA had no clear understanding of the many ways a single person interacts with their organization and the need to have these “roles” clearly defined. For example, a person can simultaneously be a “donor” and a “subscriber”, or a person can be a “member” and a “scholarship participant.” There are many different “role” permutations in which a person can be categorized by NMFA, yet they lacked a system to capture this. This leads to significant data management problems as some people existed in their data base several times under different roles. To solve this problem, we titled all people that interact with NMFA as “participants” and gave them a coding structure to help define the roles of each participant. The super-type/sub-type model on how we did this can be seen at below on next page.

*(Physical Model- Supertype subtype relationship)*

****

**Physical Model – Entity Relationship Diagram**

Below is the physical model.



**Data Dictionary**

*Participant*

This table represents the participant entity. It records information about the people that engage with the NMFA. This is the core table form which the system is built from.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entity Name: Participant** | | | | | |
| **Attribute Name** | **Field Type** | **Nullable** | **Primary / Foreign Key Constraints** | **Domain Constraints** | **Description** |
| ParticipantID | VARCHAR(30) | NOT NULL | PK |  | Primary Key |
| ParticipantLastName | VARCHAR(40) | NOT NULL |  |  | Participant’s last name |
| ParticipantFirstName | VARCHAR(40) | NOT NULL |  |  | Participant’s first name |
| ParticipantEmail | VARCHAR(100) | NOT NULL |  |  |  |
| ParticipantHouseNumber | VARCHAR(10) | NOT NULL |  |  | Participant’s house number |
| ParticipantStreet | VARCHAR(60) | NOT NULL |  |  | Participant’s street name |
| ParticipantCity | VARCHAR(30) | NOT NULL |  |  | Participant’s city name |
| ParticipantState | VARCHAR(30) | NOT NULL |  |  | Participant’s state name |
| ParticipantZip | VARCHAR(10) | NOT NULL |  |  | Participant’s zip code |
| ParticipantType | VARCHAR(30) | NOT NULL |  |  | Coded field that represents the different status a given person can have with the organization. |
| MilitaryAffiliation | VARCHAR(50) | NOT NULL |  |  | Participant’s affiliation with the military, e.g. Army, Navy, Civilian, etc. |
| CurrentProApplicant | VARCHAR(10) | NOT NULL |  | CHECK “Yes” or “No” | Indicates if the participant is a current program applicant |
| CurrentSchApplicant | VARCHAR(10) | NOT NULl |  | CHECK “Yes” or “No” | Indicates if the participant is a current scholarship applicant |

*Donor*

This table represents the donor entity. It records select attributes of donors. It relates back to the participant table, and forward to the donate associative table, and the donation entity table.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entity Name: Donor** | | | | | |
| **Attribute Name** | **Field Type** | **Nullable** | **Primary / Foreign Key Constraints** | **Domain Constraints** | **Description** |
| ParticipantID | VARCHAR(30) | NOT NULL | PK,FK |  | Primary Key and Foreign Key |
| DonateStartingDate | DATE | NOT NULL |  |  | Date the participant started donating |

*Subscriber*

This table represents the subscriber entity. It records select attributes of subscribers. It relates back to the participant table.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entity Name: Subscriber** | | | | | |
| **Attribute Name** | **Field Type** | **Nullable** | **Primary / Foreign Key Constraints** | **Domain Constraints** | **Description** |
| ParticipantID | VARCHAR(30) | NOT NULL | PK,FK |  | Primary Key and Foreign Key |
| SubStartingDate | DATE | NOT NULL |  |  | Date the participant subscribed to NMFA |

*Member*

This table represents the member entity. It records select attributes of members. It relates back to the participant table.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entity Name: Member** | | | | | |
| **Attribute Name** | **Field Type** | **Nullable** | **Primary / Foreign Key Constraints** | **Domain Constraints** | **Description** |
| ParticipantID | VARCHAR(30) | NOT NULL | PK,FK |  | Primary Key and Foreign Key |
| MemberStartingDate | DATE | NOT NULL |  |  | Date the participant became a member |

*Volunteer*

This table represents the volunteer entity. It records select attributes of volunteers. It relates back to the participant table.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entity Name: Volunteer** | | | | | |
| **Attribute Name** | **Field Type** | **Nullable** | **Primary / Foreign Key Constraints** | **Domain Constraints** | **Description** |
| ParticipantID | VARCHAR(30) | NOT NULL | PK,FK |  | Primary Key and Foreign Key |
| VolunteerStartingDate | DATE | NOT NULL |  |  | Date the volunteer became a volunteer |

*ScholarshipApplication*

This table is an associative table that relates a scholarship applicant to a scholarship. It records select attributes of the application process, to include a decision. It relates back to the participant table and forward to the scholarship table.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entity Name: ScholarshipApplication** | | | | | |
| **Attribute Name** | **Field Type** | **Nullable** | **Primary / Foreign Key Constraints** | **Domain Constraints** | **Description** |
| ParticipantID | VARCHAR(30) | NOT NULL | PK,FK |  | Primary Key and Foreign Key |
| ScholarshipID | VARCHAR(30) | NOT NULL | PK,FK |  | Primary Key and Foreign Key |
| SchApplyDecision | VARCHAR(20) | NOT NULL |  | “Approved” or “Disapproved” | Decision (Y/N) of application |

*ScholarshipRecipient*

This table represents the scholarship recipient entity. It relates back to the participant table.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entity Name: ScholarshipRecipient** | | | | | |
| **Attribute Name** | **Field Type** | **Nullable** | **Primary / Foreign Key Constraints** | **Domain Constraints** | **Description** |
| ParticipantID | VARCHAR(30) | NOT NULL | PK,FK |  | Primary Key and Foreign Key |

*ScholarshipRecieved*

This table represents the scholarship received entity. It records select attributes of the participant IDs of those that have received scholarships, as well as records the scholarship ID for the scholarship they received and the data they received it. It relates back to the ScholarshipRecipient table and forward to the Scholarship entity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entity Name: ScholarshipReceived** | | | | | |
| **Attribute Name** | **Field Type** | **Nullable** | **Primary / Foreign Key Constraints** | **Domain Constraints** | **Description** |
| ParticipantID | VARCHAR(30) | NOT NULL | PK,FK |  | Primary Key and Foreign Key |
| ScholarshipID | VARCHAR(30) | NOT NULL | PK,FK |  | Primary Key and Foreign Key |
| SchRecieveDate | DATE | NOT NULL |  |  | Date scholarship is received |

*Scholarship*

This table represents the scholarship entity. It records select attributes of each scholarship. It relates back to the ScholarshipRecieved entity and the ScholarshipApplication entity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entity Name: Scholarship** | | | | | |
| **Attribute Name** | **Field Type** | **Nullable** | **Primary / Foreign Key Constraints** | **Domain Constraints** | **Description** |
| ScholarshipID | VARCHAR(30) | NOT NULL | PK |  | Primary Key |
| SchName | VARCHAR(40) | NOT NULL |  |  | Name of school |
| SchAmountPerPeriod | INTEGER | NOT NULL |  | >0 | Amount ($) of scholarship per period |
| SchLength | INTEGER | NOT NULL |  | >0 | Number of periods of scholarship benefit |

*ProgramApplication*

This table represents the program application entity. It records select attributes of each program application. It relates back to the ProgramApplicant entity and forward to the Program entity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entity Name: ProgramApplication** | | | | | |
| **Attribute Name** | **Field Type** | **Nullable** | **Primary / Foreign Key Constraints** | **Domain Constraints** | **Description** |
| ParticipantID | VARCHAR(30) | NOT NULL | PK,FK |  | Primary Key and Foreign Key |
| ProgramID | VARCHAR(30) | NOT NULL | PK,FK |  | Primary Key and Foreign Key |
| ProApplyDecision | VARCHAR(20) | NOT NULL |  | “Approved” or “Disapproved” | Decision of application |

*ProgramRecipient*

This table represents the program recipient entity. It relates back to the participant table.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entity Name: ProgramRecipient** | | | | | |
| **Attribute Name** | **Field Type** | **Nullable** | **Primary / Foreign Key Constraints** | **Domain Constraints** | **Description** |
| ParticipantID | VARCHAR(30) | NOT NULL | PK,FK |  | Primary Key and Foreign Key |

*ProgramRecieved*

This table represents the program received entity. It records the participant IDs of those that have received programs, as well as records the program ID for the program they received. It relates back to the ProgramRecipient entity and forward to the Program entity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entity Name: ProgramReceived** | | | | | |
| **Attribute Name** | **Field Type** | **Nullable** | **Primary / Foreign Key Constraints** | **Domain Constraints** | **Description** |
| ParticipantID | VARCHAR(30) | NOT NULL | PK,FK |  | Primary Key and Foreign Key |
| ProgramID | VARCHAR(30) | NOT NULL | PK,FK |  | Primary Key and Foreign Key |

*Program*

This table represents the program entity. It records select attributes of each program. It relates back to the ProgramRecievedentity and the ProgramApplication entity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entity Name: Program** | | | | | |
| **Attribute Name** | **Field Type** | **Nullable** | **Primary / Foreign Key Constraints** | **Domain Constraints** | **Description** |
| ProgramID | VARCHAR(30) | NOT NULL | PK |  | Primary Key |
| ProName | VARCHAR(50) | NOT NULL |  |  | Name of program |
| ProCity | VARCHAR(30) | NOT NULL |  |  | City location of program |
| ProState | VARCHAR(30) | NOT NULL |  |  | State location of program |
| ProStartingDate | DATE | NOT NULL |  |  | Starting date of the program |
| ProEndingDate | DATE | NOT NULL |  |  | Ending date of the program |

*Donate*

This associative table links the Donor and Donation entities. It records select attributes of each donation. It relates back to the Donorentity and forward to the Donation entity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entity Name: Donate** | | | | | |
| **Attribute Name** | **Field Type** | **Nullable** | **Primary / Foreign Key Constraints** | **Domain Constraints** | **Description** |
| DonationID | VARCHAR(30) | NOT NULL | PK,FK |  | Primary Key and Foreign Key |
| ParticipantID | VARCHAR(30) | NOT NULL |  |  | Primary Key and Foreign Key |
| DonateAmount | INTEGER | NOT NULL |  | >0 | Amount of each donation for each donor |

*DonationCampaign*

This table represents the donation campaign entity. It records select attributes of each donation campaign. It relates back to the donateentity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entity Name: DonationCampaign** | | | | | |
| **Attribute Name** | **Field Type** | **Nullable** | **Primary / Foreign Key Constraints** | **Domain Constraints** | **Description** |
| DonCamID | VARCHAR(30) | NOT NULL | PK |  | Primary Key |
| DonCamName | VARCHAR(50) | NOT NULL |  |  | Amount of each donation for each donor |
| DonStartingDate | DATE | NOT NULL |  |  | Beginning date of campaign |
| DonEndingDate | DATE | NOT NULL |  |  | End date of campaign |

*EMarkingInvolvement*

This associative table links a participant and an e-marketing campaign entities. It records select attributes of each e-marketing campaign involvement. It relates back to the Participantentity and forward to the EMarketingCampaign entity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entity Name: EMarkigInvolvement** | | | | | |
| **Attribute Name** | **Field Type** | **Nullable** | **Primary / Foreign Key Constraints** | **Domain Constraints** | **Description** |
| ParticipantID | VARCHAR(30) | NOT NULL | PK,FK |  | Primary Key and Foreign Key |
| ECampID | VARCHAR(40) | NOT NULL | PK,FK |  | Primary Key and Foreign Key (Name of email campaign) |
| EmailSentDate | DATE | NOT NULL |  |  | Date email was sent |

*EMarketingCampaign*

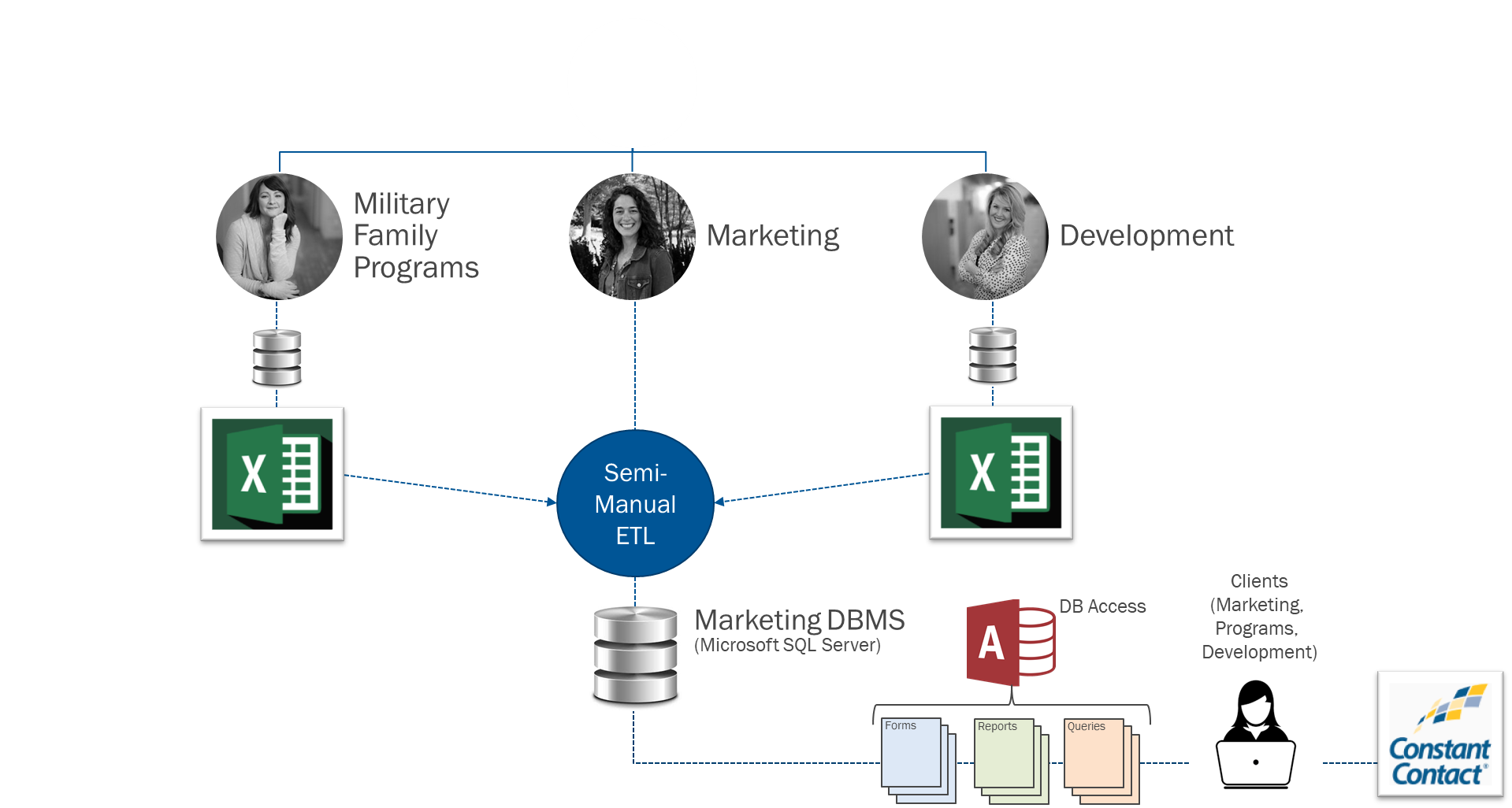
This table represents the e-marketing campaign entity. It records select attributes of each e-marketing campaign. It relates back to the EMarkingInvolvement entity.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entity Name: EMarketingCampaign** | | | | | |
| **Attribute Name** | **Field Type** | **Nullable** | **Primary / Foreign Key Constraints** | **Domain Constraints** | **Description** |
| ECampID | VARCHAR(30) | NOT NULL | PK |  | Primary Key |
| ECampName | VARCHAR(50) | NOT NULL |  |  | Name of email campaign |
| ECampStartingDate | DATE | NOT NULL |  |  | Start date of email campaign |
| ECampEndingDate | DATE | NOT NULL |  |  | End date of email campaign |

**Structure and Query Implementation**

**Database System Infrastructure**

The following illustration highlights the database system infrastructure. Given the constraints for this project imposed on us by the sponsor, the Development and Programs departments were required to maintain their current systems. As such, and with our sponsor’s permission, we modeled the extraction, transformation and loading of the data from the Development and Program’s data systems as a function performed by the Marketing department. In future development, this will become an enterprise function. Additionally, the only users of this DBMS will be the marketing team until further development expands the capability to the Programs and Development departments.

**

**SQL Scripts (Creating and Inserting Data)**

CREATE TABLE Participant

(

ParticipantID VARCHAR(30) NOT NULL PRIMARY KEY,

ParticipantFirstaName VARCHAR (40) NOT NULL,

ParticipantLastName VARCHAR (40) NOT NULL,

ParticipantEmail VARCHAR(100) NOT NULL,

ParticipantHouseNumber VARCHAR (10) NOT NULL,

ParticipantStreet VARCHAR (60) NOT NULL,

ParticipantCity VARCHAR (30) NOT NULL,

ParticipantState VARCHAR (30) NOT NULL,

ParticipantZip VARCHAR (10) NOT NULL,

ParticipantType VARCHAR(30) NOT NULL,

MilitaryAffliation VARCHAR(50) NOT NULL,

CurrentProApplicant VARCHAR (10) NOT NULL,

CurrentSchApplicant VARCHAR (10) NOT NULL,

CONSTRAINT chk\_CurrentProApplicant CHECK (CurrentProApplicant ='YES' OR CurrentProApplicant ='NO') ,

CONSTRAINT chk\_CurrentSchApplicant CHECK (CurrentSchApplicant ='YES' OR CurrentSchApplicant ='NO') ,

);

CREATE TABLE Donor

(

ParticipantID VARCHAR(30) NOT NULL ,

DonateStartingDate DATETIME NOT NULL DEFAULT GETDATE (),

CONSTRAINT Donor\_PK PRIMARY KEY (ParticipantID),

CONSTRAINT Donor\_FK FOREIGN KEY (ParticipantID) REFERENCES Participant (ParticipantID)

);

CREATE TABLE Subscriber

(

ParticipantID VARCHAR(30) NOT NULL ,

SubStartingDate DATETIME NOT NULL DEFAULT GETDATE (),

CONSTRAINT Subscriber\_PK PRIMARY KEY (ParticipantID),

CONSTRAINT Subscriber\_FK FOREIGN KEY (ParticipantID) REFERENCES Participant (ParticipantID)

);

CREATE TABLE Member

(

ParticipantID VARCHAR(30) NOT NULL ,

MemberStartingDate DATETIME NOT NULL DEFAULT GETDATE (),

CONSTRAINT Member\_PK PRIMARY KEY (ParticipantID),

CONSTRAINT Member\_FK FOREIGN KEY (ParticipantID) REFERENCES Participant (ParticipantID)

);

CREATE TABLE Volunteer

(

ParticipantID VARCHAR(30) NOT NULL ,

VolunteerStartingDate DATETIME NOT NULL DEFAULT GETDATE (),

CONSTRAINT Volunteer\_PK PRIMARY KEY (ParticipantID),

CONSTRAINT Volunteer\_FK FOREIGN KEY (ParticipantID) REFERENCES Participant (ParticipantID)

);

CREATE TABLE ScholarshipRecipient

(

ParticipantID VARCHAR(30) NOT NULL PRIMARY KEY,

CONSTRAINT  ScholarshipRecipient\_FK FOREIGN KEY (ParticipantID) REFERENCES Participant (ParticipantID)

);

CREATE TABLE ProgramRecipient

(

ParticipantID VARCHAR(30) NOT NULL PRIMARY KEY,

CONSTRAINT  ProgramRecipient\_FK FOREIGN KEY (ParticipantID) REFERENCES Participant (ParticipantID)

);

CREATE TABLE Program

(

ProgramID VARCHAR(30) NOT NULL PRIMARY KEY,

ProName VARCHAR(50) NOT NULL,

ProCity VARCHAR (30) NOT NULL,

ProState VARCHAR (30) NOT NULL,

ProStartingDate DATETIME NOT NULL DEFAULT GETDATE (),

ProEndingDate DATETIME NOT NULL DEFAULT GETDATE (),

) ;

CREATE TABLE Scholarship

(

ScholarshipID VARCHAR(30) NOT NULL PRIMARY KEY,

SchName VARCHAR(40) NOT NULL,

SchAmountPerPeriod INTEGER NOT NULL,

SchLength INTEGER NOT NULL,

CONSTRAINT chk\_SchAmountPerPeriod CHECK (SchAmountPerPeriod >0),

CONSTRAINT chk\_SchLength CHECK (SchLength >0)

) ;

CREATE TABLE DonationCampaign

(

DonCamID VARCHAR(30) NOT NULL PRIMARY KEY,

DonCamName VARCHAR(50) NOT NULL,

DonStartingDate DATETIME NOT NULL DEFAULT GETDATE (),

DonEndingDate DATETIME NOT NULL DEFAULT GETDATE (),

) ;

CREATE TABLE EMarketingCampaign

(

ECampID VARCHAR(30) NOT NULL PRIMARY KEY,

ECampName VARCHAR(50) NOT NULL,

ECampStartingDate DATETIME NOT NULL DEFAULT GETDATE (),

ECampEndingDate DATETIME NOT NULL DEFAULT GETDATE (),

) ;

CREATE TABLE ScholarshipApplication

(

ParticipantID VARCHAR (30) NOT NULL,

ScholarshipID VARCHAR (30) NOT NULL,

SchApplyDecision VARCHAR(20),

CONSTRAINT ScholarshipApplication\_PK1 PRIMARY KEY (ParticipantID,ScholarshipID),

CONSTRAINT ScholarshipApplication\_FK1 FOREIGN KEY (ParticipantID) REFERENCES Participant (ParticipantID),

CONSTRAINT ScholarshipApplication\_FK2 FOREIGN KEY (ScholarshipID) REFERENCES Scholarship (ScholarshipID),

CONSTRAINT chk\_ScholarshipApplication CHECK (SchApplyDecision ='Approved' OR SchApplyDecision ='Disapproved') ,

) ;

CREATE TABLE ScholarshipReceived

(

ParticipantID VARCHAR (30) NOT NULL,

ScholarshipID VARCHAR (30) NOT NULL,

SchRecieveDate DATETIME NOT NULL DEFAULT GETDATE (),

CONSTRAINT ScholarshipReceived\_PK1 PRIMARY KEY (ParticipantID,ScholarshipID),

CONSTRAINT ScholarshipReceived\_FK1 FOREIGN KEY (ParticipantID) REFERENCES Participant (ParticipantID),

CONSTRAINT ScholarshipReceived\_FK2 FOREIGN KEY (ScholarshipID) REFERENCES Scholarship (ScholarshipID)

) ;

CREATE TABLE ProgramApplication

(

ParticipantID VARCHAR (30) NOT NULL,

ProgramID VARCHAR (30) NOT NULL,

ProApplyDecision VARCHAR(20),

CONSTRAINT ProApplication\_PK1 PRIMARY KEY (ParticipantID,ProgramID),

CONSTRAINT ProApplication\_FK1 FOREIGN KEY (ParticipantID) REFERENCES Participant (ParticipantID),

CONSTRAINT ProApplication\_FK2 FOREIGN KEY (ProgramID) REFERENCES Program (ProgramID),

CONSTRAINT chk\_ProApplyDecision CHECK (ProApplyDecision ='Approved' OR ProApplyDecision ='Disapproved'),

) ;

CREATE TABLE ProgramReceived

(

ParticipantID VARCHAR (30) NOT NULL,

ProgramID VARCHAR (30) NOT NULL,

CONSTRAINT ProgramReceived\_PK1 PRIMARY KEY (ParticipantID,ProgramID),

CONSTRAINT ProgramReceived\_FK1 FOREIGN KEY (ParticipantID) REFERENCES Participant (ParticipantID),

CONSTRAINT ProgramReceived\_FK2 FOREIGN KEY (ProgramID) REFERENCES Program (ProgramID),

) ;

CREATE TABLE Donate

(

ParticipantID VARCHAR(30) NOT NULL,

DonCamID VARCHAR(30) NOT NULL,

DonateAmount INTEGER NOT NULL,

CONSTRAINT Donate\_PK1 PRIMARY KEY (ParticipantID,DonCamID),

CONSTRAINT Donate\_FK1 FOREIGN KEY (ParticipantID) REFERENCES Participant (ParticipantID),

CONSTRAINT Donate\_FK2 FOREIGN KEY (DonCamID) REFERENCES DonationCampaign (DonCamID),

CONSTRAINT chk\_DonateAmount CHECK (DonateAmount>0),

) ;

CREATE TABLE EMarketingInvolvement

(

ParticipantID VARCHAR(30) NOT NULL,

ECampID VARCHAR(30) NOT NULL,

EmailSentDate DATETIME NOT NULL DEFAULT GETDATE (),

CONSTRAINT EMarketingInvolvement\_PK1 PRIMARY KEY (ParticipantID,ECampID),

CONSTRAINT EMarketingInvolvement\_FK1 FOREIGN KEY (ParticipantID) REFERENCES Participant (ParticipantID),

CONSTRAINT EMarketingInvolvement\_FK2 FOREIGN KEY (ECampID) REFERENCES EMarketingCampaign (ECampID),

) ;

--Participant Table

INSERT INTO Participant VALUES

('1','Abner','Amaro Rivera','privateamaro2016@gmail.com','410',' Garland Dr',' Ausitn','TX','77566','1,1,1,0,0,0','Active Military','YES','NO'),

('2','Aimee','Amato','aimeebamato@gmail.com','28','E 12th St','New York','NY','10003','0,0,1,0,0,0','NationalGuard','YES','NO'),

('3','Adrienne','Almestica','adriennealmestica@yahoo.com','78','State','Bath','ME','14530','0,0,0,1,0,0','Civilia','NO','NO'),

('4','Edwards','Ryan','firefighter8143@yahoo.com','636','Crestview Dr', 'Diamond Springs','CA','95619','0,0,0,1,0,1','Civilian','NO','YES'),

('5','Gaffney','Sean','lsgaffney@gmail.com','478','9th St','Brooklyn','NY','11215','1,0,0,0,0,0','Retired','YES','NO'),

('6','Candelario','Sindy','candelariosindy@yahoo.com','245','Licoln Rd','Juana Diaz','PR','11795','0,0,1,0,1,0','Civilian','YES','NO'),

('7','Cancel Martinez','Miriam','mili.59@hotmail.com','345','Lexington Dr','Aguadilla', 'PR','78603','0,0,1,0,1,0','Civilian','YES','NO'),

('8','Campolo','Thomas','tommyc6182@yahoo.com','1014','Branch Rd','Vineland','NJ','98360', '0,0,0,0,0,1','Civilian','YES','YES'),

('9','Cannon','Greggory','gregcannon314@gmail.com','2509','Steep Hill Dr','Rockville','MD','20853', '1,0,1,0,0,0,','Active Military','YES','YES'),

('10','Kelly','Barb','kellybarb7@gmail.com','110','Cardinal Court', 'Ashland','OH','44805','1,1,1,0,0,0','Spouse of Service Member','NO','YES'),

('11','Rodriguez','Erving','empyaucano@coqui.net','4','Wellington Ave','Carolina','PR','12987','1,1,1,1,1,0','Civilian','YES','NO'),

('12','Melinda','Albright','melindysue@sbcglobal.net','2321','Houston Rd','San Diego','CA','92107','0,0,0,0,0,1','Spouse of Service Member','NO','YES'),

('13','Stern','Jonathan','jstern13@earthlink.net','1561','Fort Hunt Rd','San Diego','CA','92109', '0,0,1,0,1,0','Civilian','YES','YES'),

('14','Vaz','Godvin','godvin.vaz@gmail.com','3769','Boulder Canyon Dr','Castro Valley','CA','94552','1,1,1,0,0,0','Civilian','NO','NO'),

('15','Otto','Stevial','stevial\_otto@yahoo.com','105','189th Street','Saint Albans','NY','11412','0,0,0,0,0,1','Civilian','NO','YES');

SELECT \* FROM Participant



 --Donor Table

INSERT INTO Donor VALUES

('1','2013-10-29 00:00:00.000'),

('5','2012-9-1 00:00:00.000'),

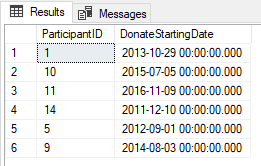
('9','2014-8-3 00:00:00.000'),

('10','2015-7-5 00:00:00.000'),

('11','2016-11-9 00:00:00.000'),

('14','2011-12-10 00:00:00.000');

SELECT \* FROM DONOR



--Member Table

INSERT INTO Member VALUES

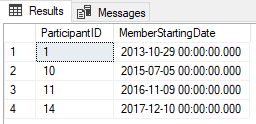
('1','2013-10-29 00:00:00.000'),

('10','2015-7-5 00:00:00.000'),

('11','2016-11-9 00:00:00.000'),

('14','2017-12-10 00:00:00.000');

SELECT \* FROM Member



--Subscriber Table

INSERT INTO Subscriber VALUES

('1','2012-9-15 00:00:00.000'),

('2','2013-4-18 00:00:00.000'),

('6','2015-12-11 00:00:00.000'),

('7','2016-8-12 00:00:00.000'),

('9','2014-12-24 00:00:00.000'),

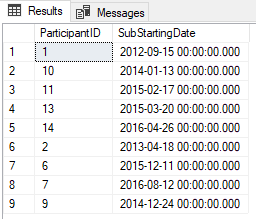
('10','2014-1-13 00:00:00.000'),

('11','2015-2-17 00:00:00.000'),

('13','2015-3-20 00:00:00.000'),

('14','2016-4-26 00:00:00.000');

SELECT \* FROM Subscriber



--Volunteer Table

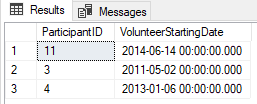
INSERT INTO Volunteer VALUES

('3','2011-5-2 00:00:00.000'),

('4','2013-1-6 00:00:00.000'),

('11','2014-6-14 00:00:00.000');

SELECT \* FROM Volunteer



--Scholarship Table

INSERT INTO Scholarship VALUES

('1','Bright Star',1000,2),

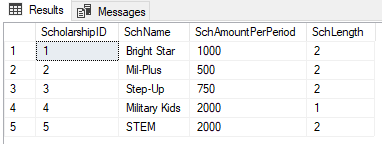
('2','Mil-Plus',500,2),

('3','Step-Up',750,2),

('4','Military Kids',2000,1),

('5','STEM',2000,2);

SELECT \* FROM Scholarship



--Program Table

INSERT INTO Program VALUES

('1','Purple Camp','Mayo','MD','2019-11-20 00:00:00.000','2019-11-29 00:00:00.000'),

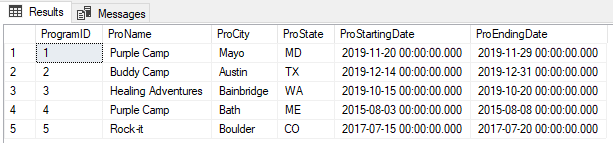
('2','Buddy Camp','Austin','TX','2019-12-14 00:00:00.000','2019-12-31 00:00:00.000'),

('3','Healing Adventures', 'Bainbridge','WA', '2019-10-15 00:00:00.000','2019-10-20 00:00:00.000'),

('4','Purple Camp','Bath','ME','2015-08-03 00:00:00.000','2015-08-08 00:00:00.000'),

('5','Rock-it','Boulder','CO','2017-07-15 00:00:00.000','2017-07-20 00:00:00.000');

SELECT \* FROM Program



--EMarketingCampaign Table

INSERT INTO EMarketingCampaign VALUES

('1','Giving Tuesday1','2018-04-01 00:00:00.000','2018-04-02 00:00:00.000'),

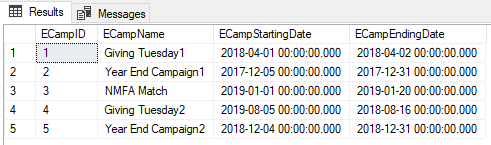
('2','Year End Campaign1','2017-12-05 00:00:00.000','2017-12-31 00:00:00.000'),

('3','NMFA Match','2019-1-1 00:00:00.000','2019-1-20 00:00:00.000'),

('4','Giving Tuesday2','2019-08-05 00:00:00.000','2018-08-16 00:00:00.000'),

('5','Year End Campaign2','2018-12-04 00:00:00.000','2018-12-31 00:00:00.000');

SELECT \* FROM EMarketingCampaign



--DonationCampaign Table

INSERT INTO DonationCampaign VALUES

('1','Giving Tuesday1','2018-04-02 00:00:00.000','2018-04-03 00:00:00.000'),

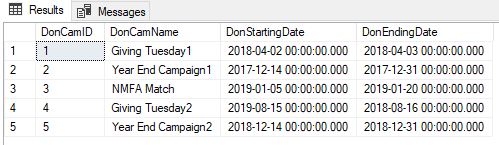
('2','Year End Campaign1','2017-12-14 00:00:00.000','2017-12-31 00:00:00.000'),

('3','NMFA Match','2019-01-5 00:00:00.000','2019-01-20 00:00:00.000'),

('4','Giving Tuesday2','2019-08-15 00:00:00.000','2018-08-16 00:00:00.000'),

('5','Year End Campaign2','2018-12-14 00:00:00.000','2018-12-31 00:00:00.000');

SELECT \* FROM DonationCampaign



--Program Application Table

INSERT INTO ProgramApplication VALUES

('1','1','Disapproved'),

('2','2','Disapproved'),

('5','2','Disapproved'),

('6','3','Approved'),

('7','3','Approved'),

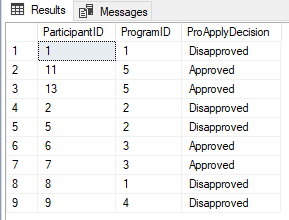
('8','1','Disapproved'),

('11','5','Approved'),

('9','4','Disapproved'),

('13','5','Approved');

SELECT \* FROM ProgramApplication



--Program Recepient Table

INSERT INTO ProgramRecipient VALUES

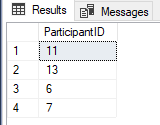
('6'),

('7'),

('11'),

('13');

SELECT \* FROM ProgramRecipient



--Scholarship Application Table

INSERT INTO ScholarshipApplication VALUES

('4','1','Approved'),

('8','2','Approved'),

('9','2','Disapproved'),

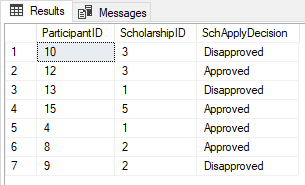
('10','3','Disapproved'),

('12','3','Approved'),

('13','1','Disapproved'),

('15','5','Approved');

SELECT \* FROM ScholarshipApplication



--Scholalrship Recepient Table

INSERT INTO ScholarshipRecipient VALUES

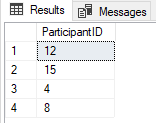
('4'),

('8'),

('12'),

('15');

SELECT \* FROM ScholarshipRecipient



--Donate Table

INSERT INTO Donate VALUES

('5','1',30),

('9','2',80),

('10','2',100),

('11','3',200),

('14','3',350),

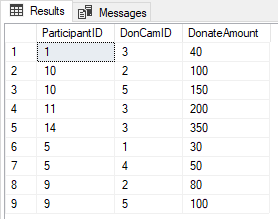
('1','3',40),

('5','4',50),

('9','5',100),

('10','5',150);

SELECT \* FROM Donate



--EMarketingInvolvement

INSERT INTO EMarketingInvolvement VALUES

('1','1','2018-04-01 00:00:00.000'),

('2','1','2018-04-01 00:00:00.000'),

('3','2','2017-12-14 00:00:00.000'),

('4','2','2017-12-15 00:00:00.000'),

('5','3','2019-1-1 00:00:00.000'),

('6','3','2019-1-4 00:00:00.000'),

('7','3','2019-1-10 00:00:00.000'),

('8','4','2018-08-5 00:00:00.000'),

('9','4','2018-08-7 00:00:00.000'),

('10','4','2018-08-9 00:00:00.000'),

('11','4','2018-08-11 00:00:00.000'),

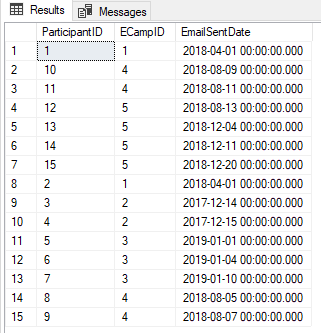
('12','5','2018-08-13 00:00:00.000'),

('13','5','2018-12-04 00:00:00.000'),

('14','5','2018-12-11 00:00:00.000'),

('15','5','2018-12-20 00:00:00.000');

SELECT \* FROM EMarketingInvolvement



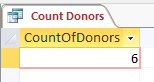
**SQL Statements for Answering Major Data Questions**

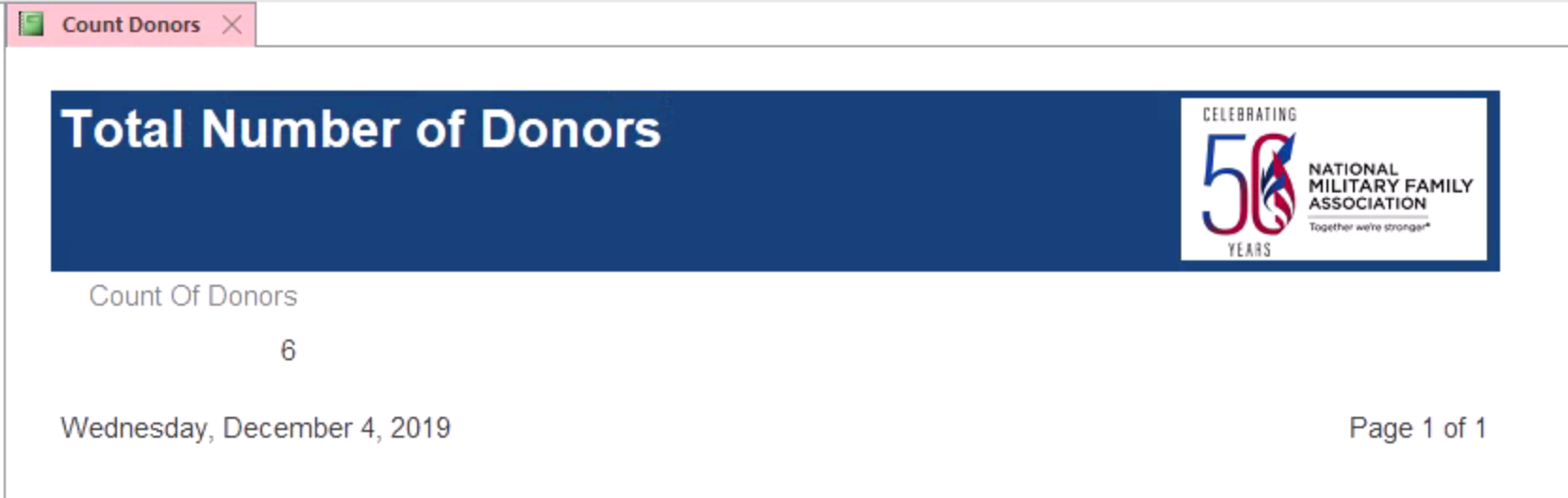
Our sponsor provided us with specific information requirements for the DBMS and we used these to develop and refine out major data questions. and we generated the required schema and SQL statements to provide answers. The Marketing manager requested more Queries developed than forms and reports to help her team answer specific questions and to export the query results to MS Excel.

1. **How many donors do we have?**

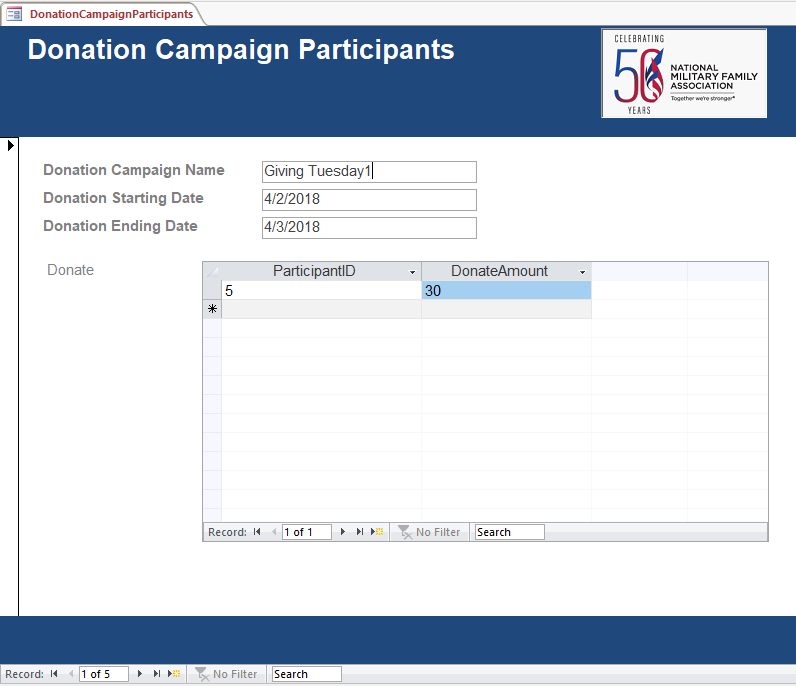
SELECT Count(dbo\_Donor.ParticipantID) AS CountOfParticipantID

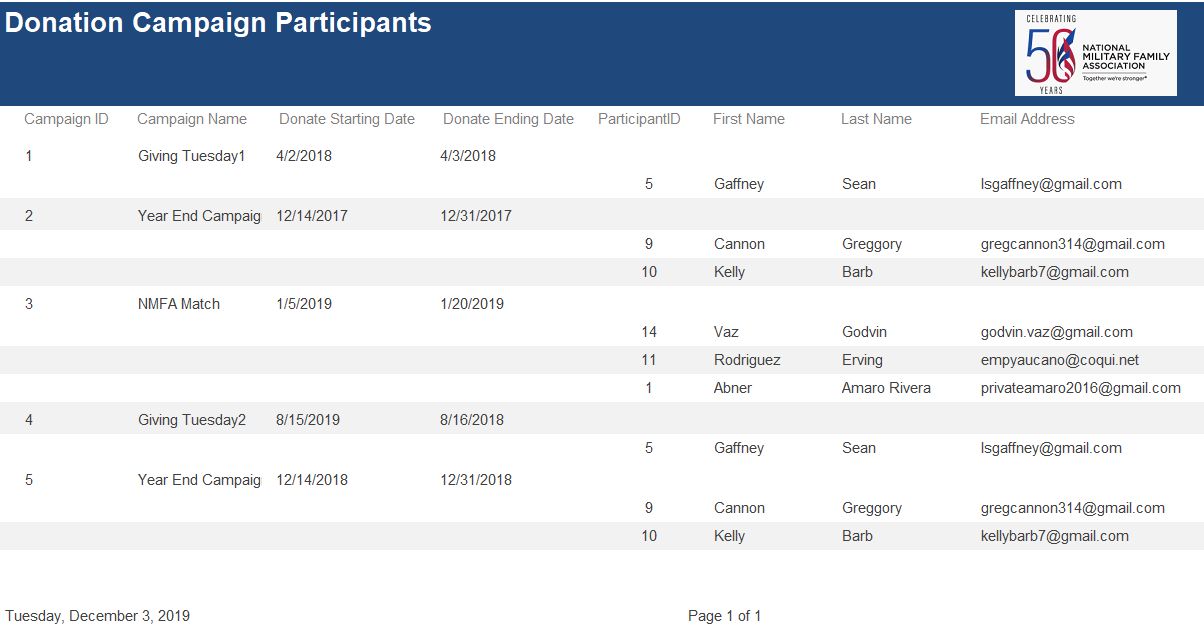
FROM dbo\_Donor;





1. **Which donors participated in a given donation activity?**



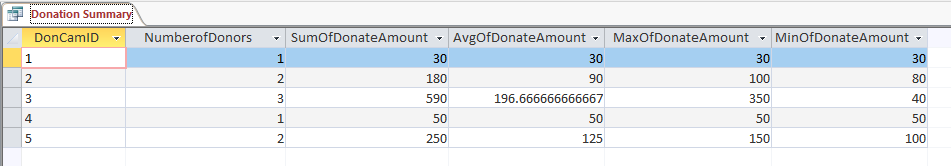


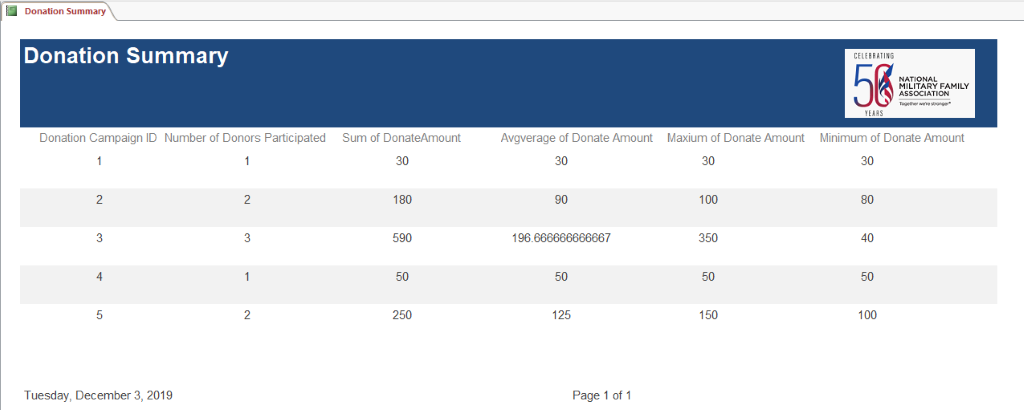
1. **What is the performance of each donation campaign?**

SELECT dbo\_Donate.DonCamID, Count(dbo\_Donor.ParticipantID) AS NumberofDonors, Sum(dbo\_Donate.DonateAmount) AS SumOfDonateAmount, Avg(dbo\_Donate.DonateAmount) AS AvgOfDonateAmount, Max(dbo\_Donate.DonateAmount) AS MaxOfDonateAmount, Min(dbo\_Donate.DonateAmount) AS MinOfDonateAmount

FROM dbo\_Donor INNER JOIN (dbo\_DonationCampaign INNER JOIN dbo\_Donate ON dbo\_DonationCampaign.[DonCamID] = dbo\_Donate.[DonCamID]) ON dbo\_Donor.[ParticipantID] = dbo\_Donate.[ParticipantID]

GROUP BY dbo\_Donate.DonCamID;



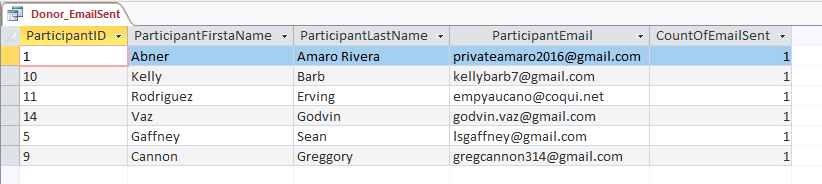


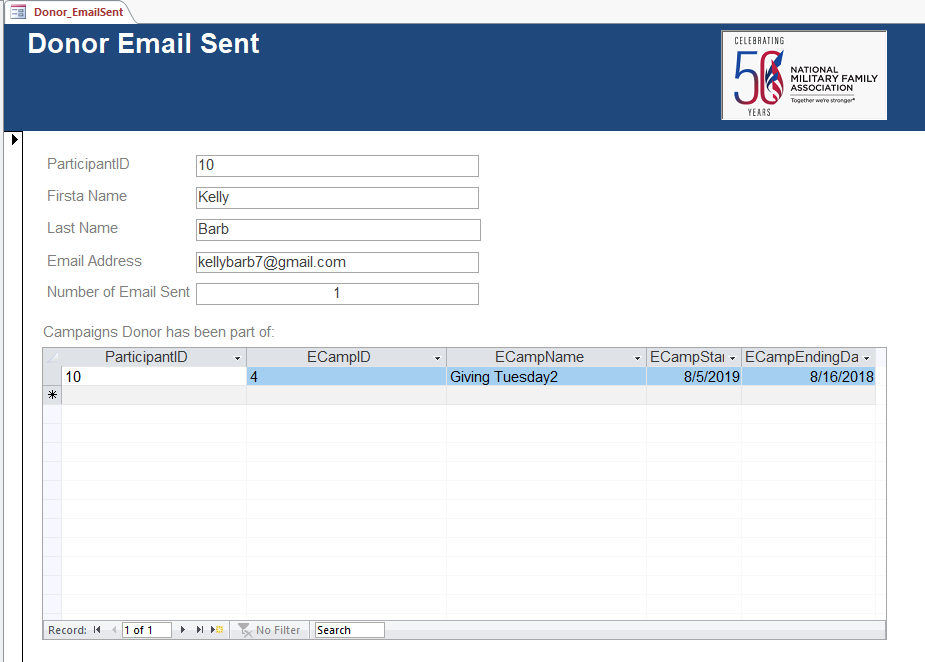
1. **How many emails have we sent to donor X?**

SELECT dbo\_Participant.ParticipantID, dbo\_Participant.ParticipantFirstaName, dbo\_Participant.ParticipantLastName, dbo\_Participant.ParticipantEmail, Count(dbo\_EMarketingInvolvement.EmailSentDate) AS CountOfEmailSent

FROM (dbo\_Participant INNER JOIN dbo\_Donor ON dbo\_Participant.[ParticipantID] = dbo\_Donor.[ParticipantID]) INNER JOIN dbo\_EMarketingInvolvement ON dbo\_Participant.[ParticipantID] = dbo\_EMarketingInvolvement.[ParticipantID]

GROUP BY dbo\_Participant.ParticipantID, dbo\_Participant.ParticipantFirstaName, dbo\_Participant.ParticipantLastName, dbo\_Participant.ParticipantEmail;





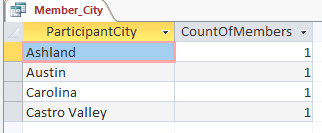
1. **How many members live in city X?**

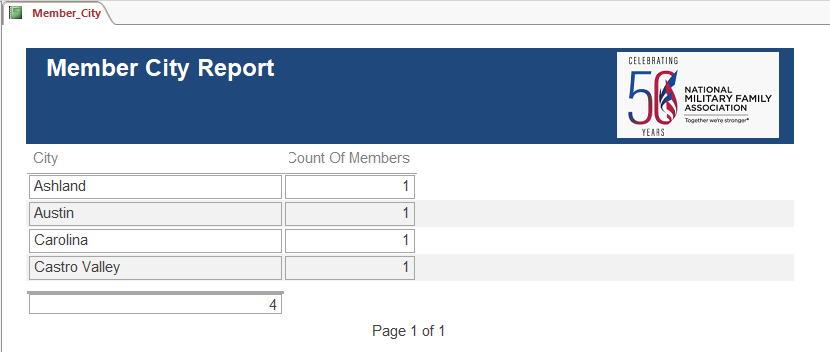
SELECT dbo\_Participant.ParticipantCity, Count(dbo\_Member.ParticipantID) AS CountOfParticipantID

FROM dbo\_Participant INNER JOIN dbo\_Member ON dbo\_Participant.[ParticipantID] = dbo\_Member.[ParticipantID]

GROUP BY dbo\_Participant.ParticipantCity

ORDER BY Count(dbo\_Member.ParticipantID) DESC;





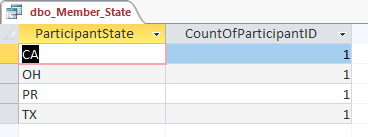
1. **In what state do most of our members live?**

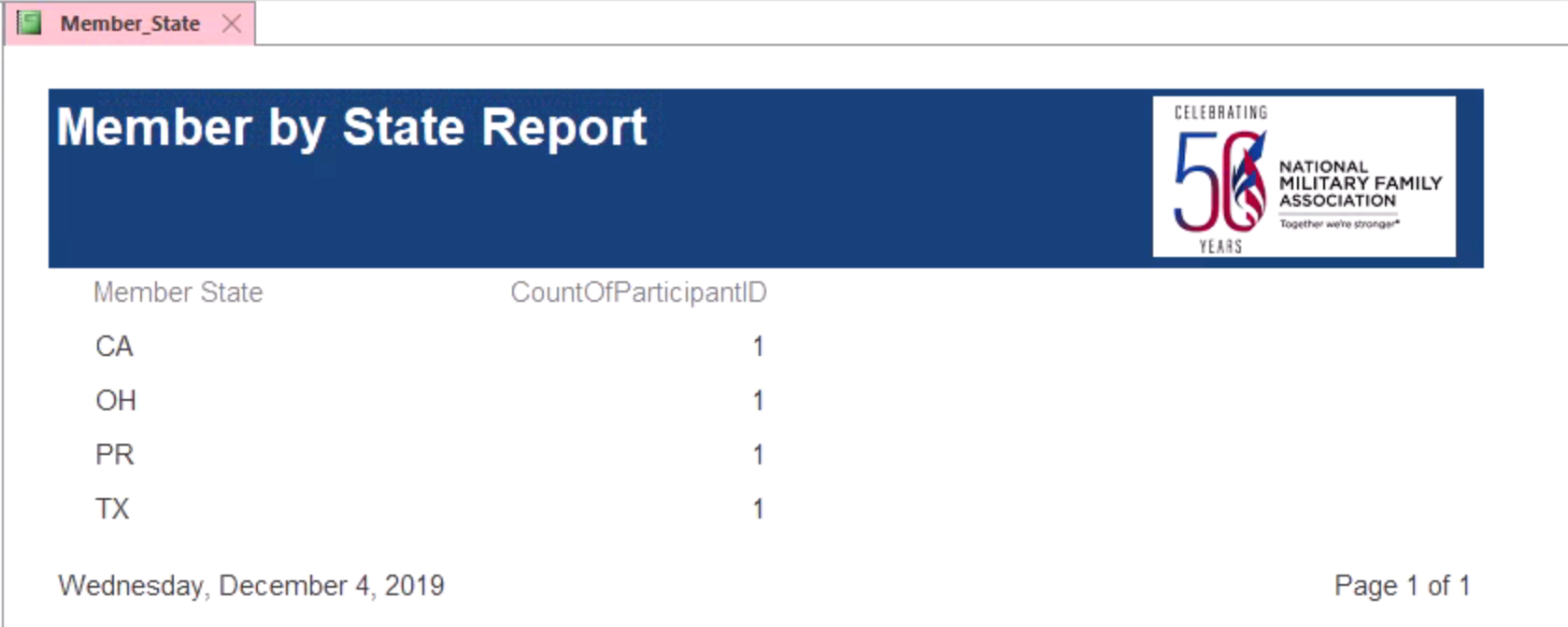
SELECT dbo\_Participant.ParticipantState, Count(dbo\_Member.ParticipantID) AS CountOfParticipantID

FROM dbo\_Participant INNER JOIN dbo\_Member ON dbo\_Participant.[ParticipantID] = dbo\_Member.[ParticipantID]

GROUP BY dbo\_Participant.ParticipantState

ORDER BY Count(dbo\_Member.ParticipantID) DESC;

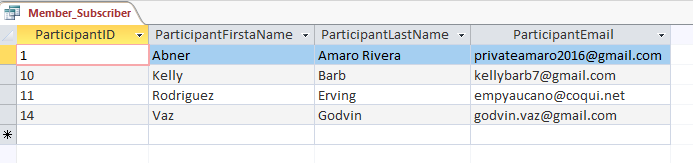


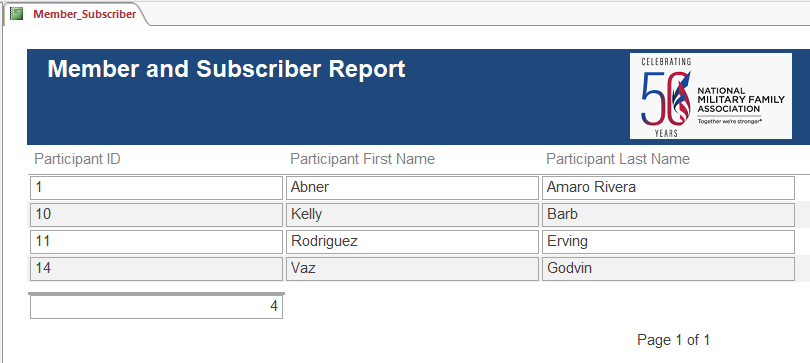


1. **How many subscribers are also members?**

SELECT dbo\_Member.ParticipantID, dbo\_Participant.ParticipantFirstaName, dbo\_Participant.ParticipantLastName, dbo\_Participant.ParticipantEmail

FROM (dbo\_Participant INNER JOIN dbo\_Subscriber ON dbo\_Participant.[ParticipantID] = dbo\_Subscriber.[ParticipantID]) INNER JOIN dbo\_Member ON dbo\_Participant.[ParticipantID] = dbo\_Member.[ParticipantID];

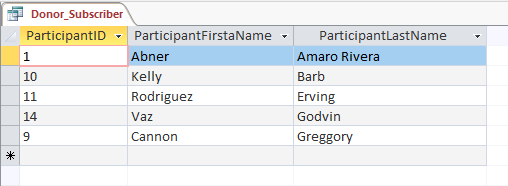


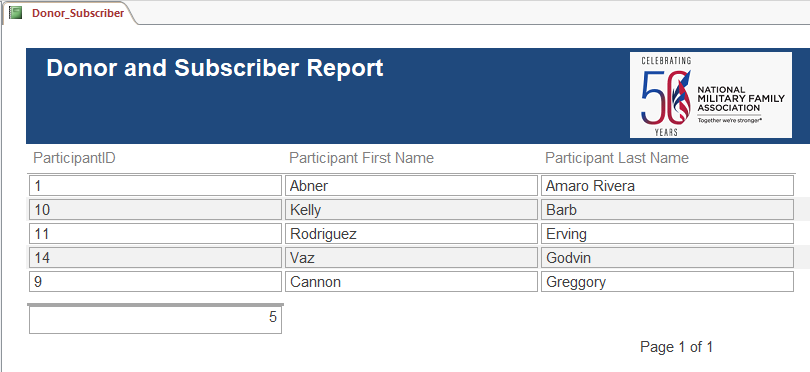


1. **Which subscribers are donors?**

SELECT dbo\_Subscriber.ParticipantID, dbo\_Participant.ParticipantFirstaName, dbo\_Participant.ParticipantLastName

FROM (dbo\_Participant INNER JOIN dbo\_Donor ON dbo\_Participant.[ParticipantID] = dbo\_Donor.[ParticipantID]) INNER JOIN dbo\_Subscriber ON dbo\_Participant.[ParticipantID] = dbo\_Subscriber.[ParticipantID];



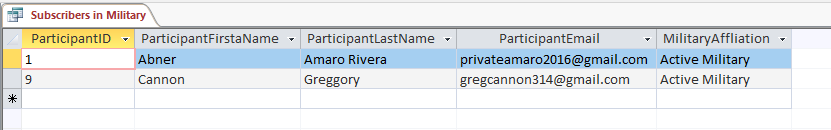


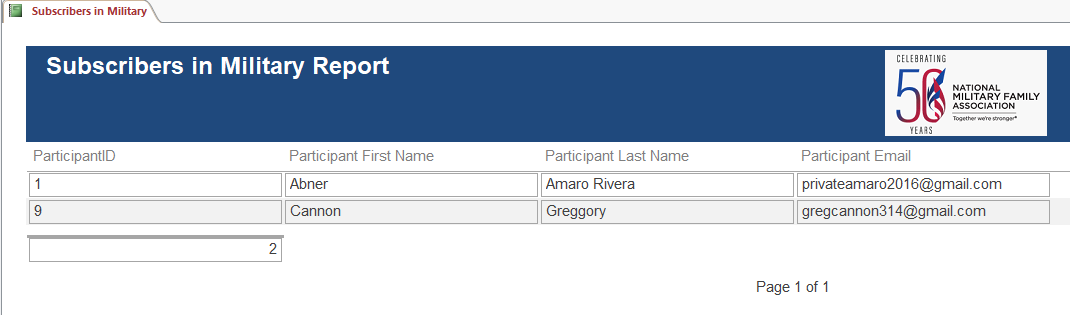
1. **Which subscribers are in the military?**

SELECT dbo\_Subscriber.ParticipantID, dbo\_Participant.ParticipantFirstaName, dbo\_Participant.ParticipantLastName, dbo\_Participant.ParticipantEmail, dbo\_Participant.MilitaryAfflication AS MilitaryAffliation

FROM dbo\_Participant INNER JOIN dbo\_Subscriber ON dbo\_Participant.ParticipantID = dbo\_Subscriber.ParticipantID

WHERE (MilitaryAfflication="Active Military")

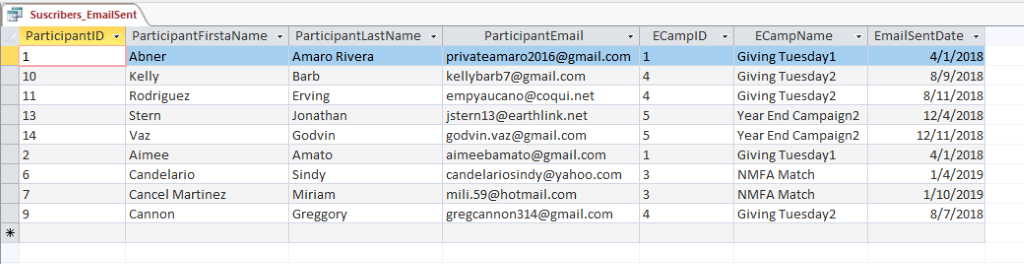


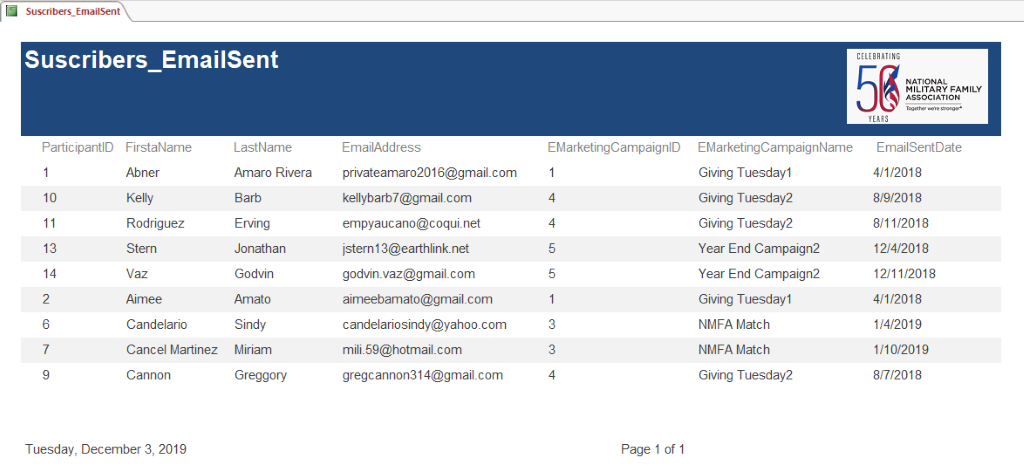


1. **When was the last time we sent an email to subscriber X**?

SELECT dbo\_Subscriber.ParticipantID, dbo\_Participant.ParticipantFirstaName, dbo\_Participant.ParticipantLastName, dbo\_Participant.ParticipantEmail, dbo\_EMarketingCampaign.ECampID, dbo\_EMarketingCampaign.ECampName, dbo\_EMarketingInvolvement.EmailSentDate

FROM (dbo\_Participant INNER JOIN dbo\_Subscriber ON dbo\_Participant.[ParticipantID] = dbo\_Subscriber.[ParticipantID]) INNER JOIN (dbo\_EMarketingCampaign INNER JOIN dbo\_EMarketingInvolvement ON dbo\_EMarketingCampaign.[ECampID] = dbo\_EMarketingInvolvement.[ECampID]) ON dbo\_Participant.[ParticipantID] = dbo\_EMarketingInvolvement.[ParticipantID];





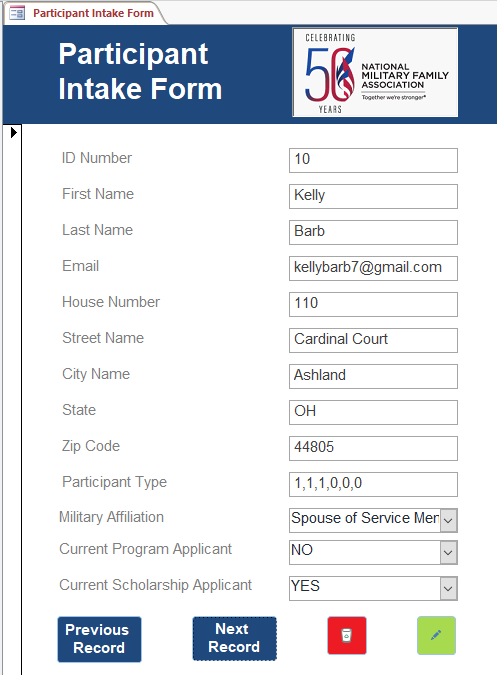
**Interface Implementation**

The interface for the Marketing DBMS will be MS Access. At this phase of development, only three marketing users will have access to the DBMS. Additional access will be granted to select users in the Development and Programs department at a future development milestone.

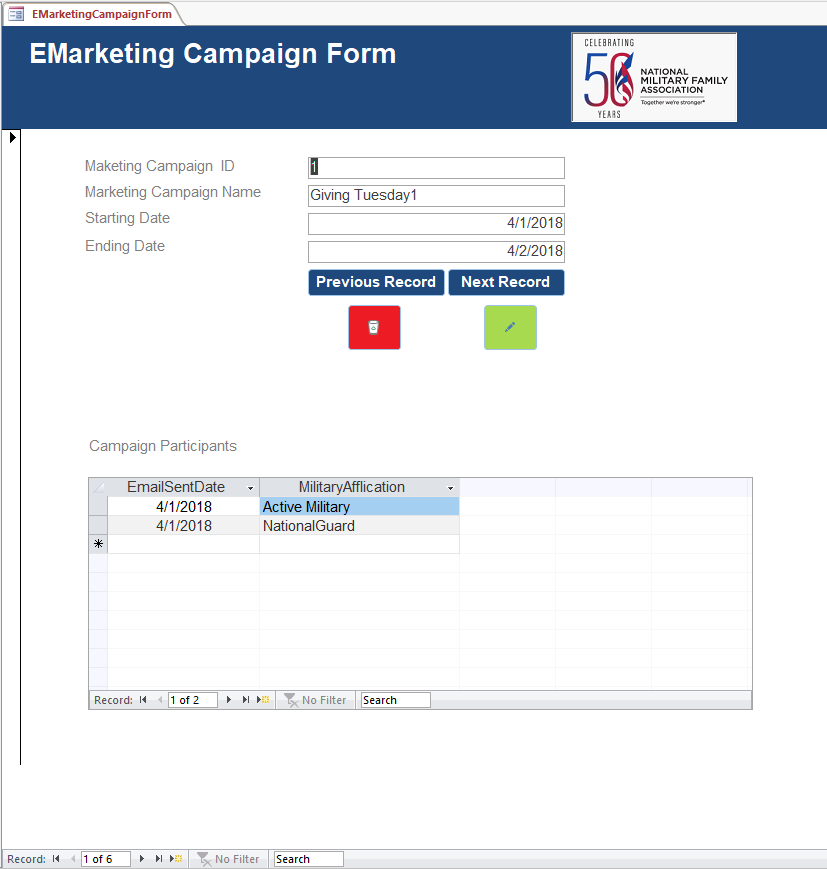
The marketing users will use forms and reports to input data and report on their operations. The users will use queries to develop exportable spreadsheets to upload into Constant Contact in order to deliver the correct emails to the correct people at the correct time. The database will reside on the enterprise server and only authorized users will be allo

**Additional Forms Not Part of the Business Questions**

Participant Intake Form

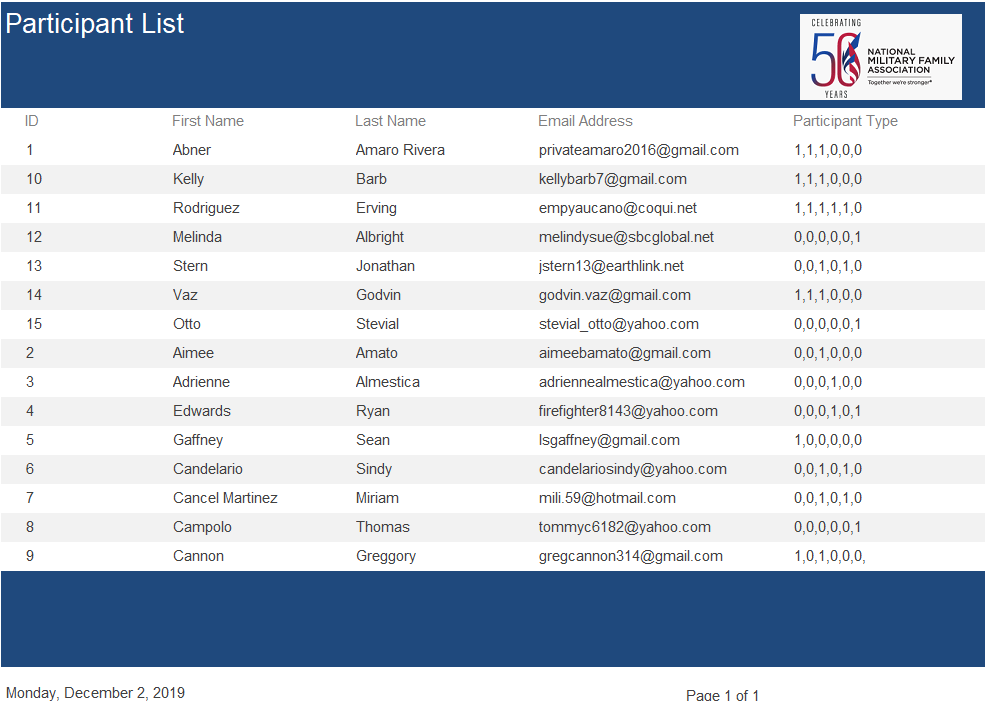


Emarketing Campaign Form



**Additional Reports Not Part of the Business Questions**

Participant Report



Donor Marketing Email report (Grouped by Military Affiliation)

