Database Administration & Database Management Concepts

Patrol Services Database Project

Olga Perera

12/15/2018

[**Part 1: Design**](https://docs.google.com/document/d/1eYOD0k4RUN7ok-umpmdUqCal6Bc4sKuECRO9tG1gm4c/edit#heading=h.elmz979f1uqm)

[Project Summary     3](https://docs.google.com/document/d/1eYOD0k4RUN7ok-umpmdUqCal6Bc4sKuECRO9tG1gm4c/edit#heading=h.2f4ietx1oos8)

Project Goals 3

Business Impact 3

Project Scope 4

Conceptual Model 4-6

Conceptual Model Diagram 6

Normalized Logical Model Summary 7

Normalized Model Diagram 8

[**Part 2: Implementation**](https://docs.google.com/document/d/1eYOD0k4RUN7ok-umpmdUqCal6Bc4sKuECRO9tG1gm4c/edit#heading=h.spfj7uy2zkwz)

[Database Table: SQL CREATE / DROP TABLE   9-11](https://docs.google.com/document/d/1eYOD0k4RUN7ok-umpmdUqCal6Bc4sKuECRO9tG1gm4c/edit#heading=h.2kprgnbxmo9w)

[Data Creation: Procedures, SQL INSERT](https://docs.google.com/document/d/1eYOD0k4RUN7ok-umpmdUqCal6Bc4sKuECRO9tG1gm4c/edit#heading=h.wsdi3utu5iag)  12-16

Data Manipulation: UPDATE statements 16-17

Answering Data Questions: VIEWS 17-18

Implementation: Web Interface 18-20

**Part 1. Design**

**Project Summary**

Patrol services provide additional neighborhood security coverage by off-duty police officers. It has been observed that with continuous increase in criminal activity and during certain hours, response time to incidents from county police departments may be delayed, in some cases reaching to one hour until a police officer is able to arrive to the scene. By increasing off duty police coverage, patrol services can provide immediate police presence and quick response and investigation to any incident in the area, and to prevent/deter criminal activity in the patrol area. Patrol services would like to digitally manage their services and requested to create a database with messaging and web interface.

**Goals:**

The goal of this project is to improve information management within the organizational structure by providing improved solution to following organizational tasks:

* Instant membership vacation requests capability via web interface;
* Instant vacation properties list generation for each officer’s shift;
* Instant vacation patrol reporting to membership;
* Improved communication capabilities between officers and members;
* Daily shift reporting data accessible via web interface;
* Easy membership management and member’s data access.

**Business Impact:**

Increase in patrol time will provide following benefits to the membership and community:

1. Membership satisfaction with patrol services.
2. Increase in membership retention.
3. Decrease in police officer response time to incidents in the covered area.
4. Deter criminal activity.

**Project Scope:**

After initial interviews with organizational stakeholders, including officers, officer in charge (manager), and individual members, following project description was established:

Patrol services provide security services to several localized neighborhoods. It hires active police officers to perform street patrols, investigate criminal incidents, and perform vacation home checks. Each police officer serves several neighborhoods. Patrol services are provided to active members of specified neighborhoods and limited to predetermined geographical area. Members can request vacation patrol when away from home and communicate with police officers on duty regarding any incidents. Officers perform following duties: street patrol, vacation home patrol, member communications, general reports for each shift describing events during his/her shift. During vacation patrol officers check on each individual property and provide a report with property status, photo, and any additional notes that require attention. Officer’s geo location is captured for better accountability. Messaging communication system is available for on duty officer – member communications.

**Conceptual Model:**

|  |  |
| --- | --- |
| **Entity** | **Attribute** |
| Neighborhood | Name [ru] |
| Member | Username [ru]  Password [r]  Primary Member Name [rc]  Secondary Member Name [rc]  Property Address [rc]  Cell Phone #1 [r]  Cell Phone #2  Home Phone  Email Address #1 [r]  Email Address #2  Property Information  Emergency Information  Member Status [r] |
| Officer | Officer Name [rc]  Password [r]  Officer Status [r] |
| Vacation Request | Start Date [r]  End Date [r]  Instructions  Email Tag [r]  Text Tag [r] |
| Vacation Patrol | Patrol Date [r]  Patrol Time [r]  Patrol Status [r]  Patrol Notes  Photo  Latitude  Longitude |
| Officer Report | Report Date [r]  Report Notes [r] |
| Text Message | Message Date [r]  Message Time [r]  Message [r] |
| **Relationship**  ----------------------------------------------------------------------------------------------------------------------------  Each neighborhood has one or more members, and each member belongs to one neighborhood.  ----------------------------------------------------------------------------------------------------------------------------  Each member can make zero or more vacation requests. Each vacation request can be made only by one member.  ----------------------------------------------------------------------------------------------------------------------------  Each member can send zero or more txt messages. Each message is sent by one member.  ----------------------------------------------------------------------------------------------------------------------------  Each officer can make zero or more officer reports. Each officer report can be done by one officer.  ----------------------------------------------------------------------------------------------------------------------------  Each officer can send zero or more txt messages. Each message is sent by one officer.  ----------------------------------------------------------------------------------------------------------------------------  Each officer can perform zero or more vacation patrols. Each vacation patrol can be done by one officer.  ----------------------------------------------------------------------------------------------------------------------------  Each vacation request can have zero or more vacation patrols. Each vacation patrol belongs to one vacation request. | |

**Conceptual Model Diagram:**

**A screenshot of a social media post

Description generated with very high confidence**

**Logical model summary:**

1. Mapped all existing entities into tables and created surrogate primary keys for each table.
2. Data types were assigned according to relevant need and several composite attributes, identifying name and address, were split into simple attributes, for example name was split into first and last name attributes. There are several attributes with “bit” value, indicating “yes/no” choice, active/inactive” status. “Status” attribute was selected to preserve data integrity for former members and former employees (officers). Email and Text tags indicate member’s choice to opt in or opt out of receiving email and text notifications. Text Message tag indicate to/from connection.
3. There are no multi value attributes in the original conceptual model.
4. Foreign Keys were created to identify relationship connections between various entities or tables in one to many relationships.
5. Logical model creation process resulted in 3N normalized form.

**Normalized Model Diagram:**

**A screenshot of a cell phone

Description generated with very high confidence**

**Part 2. Physical Database Implementation**

**Database Tables Creation:**  
SQL code to create database tables from normalized logical model. First step is to check if tables exist in our physical database, drop them if they exist, then create new tables.

-- Drop tables if already exist

-- Drop vacation\_patrol table if already exists

IF EXISTS (SELECT \* FROM INFORMATION\_SCHEMA.TABLES WHERE TABLE\_NAME = 'vacation\_patrol')

BEGIN

DROP TABLE vacation\_patrol

END

-- Drop vacation\_request table if already exists

IF EXISTS (SELECT \* FROM INFORMATION\_SCHEMA.TABLES WHERE TABLE\_NAME = 'vacation\_request')

BEGIN

DROP TABLE vacation\_request

END

-- Drop text\_message table if already exists

IF EXISTS (SELECT \* FROM INFORMATION\_SCHEMA.TABLES WHERE TABLE\_NAME = 'text\_message')

BEGIN

DROP TABLE text\_message

END

-- Drop officer\_report table if already exists

IF EXISTS (SELECT \* FROM INFORMATION\_SCHEMA.TABLES WHERE TABLE\_NAME = 'officer\_report')

BEGIN

DROP TABLE officer\_report

END

-- Drop member table if already exists

IF EXISTS (SELECT \* FROM INFORMATION\_SCHEMA.TABLES WHERE TABLE\_NAME = 'member')

BEGIN

DROP TABLE member

END

-- Drop neighborhood table if already exists

IF EXISTS (SELECT \* FROM INFORMATION\_SCHEMA.TABLES WHERE TABLE\_NAME = 'neighborhood')

BEGIN

DROP TABLE neighborhood

END

-- Drop officer table if already exists

IF EXISTS (SELECT \* FROM INFORMATION\_SCHEMA.TABLES WHERE TABLE\_NAME = 'officer')

BEGIN

DROP TABLE officer

END

-- Create officer table

CREATE TABLE officer (

-- Columns for the officer table

officerID int identity,

officer\_last\_name varchar(30) not null,

officer\_first\_name varchar(30) not null,

officer\_password varchar(20) not null,

officer\_status bit not null,

-- Constraints to the officer table

CONSTRAINT PK\_officerID PRIMARY KEY (officerID),

)

-- End creating the officer table

-- Create neighborhood table

CREATE TABLE neighborhood (

-- Columns for the neighborhood table

nhoodID int identity,

nhood\_name varchar(30) not null,

-- Constraints to the neighborhood table

CONSTRAINT PK\_nhoodID PRIMARY KEY (nhoodID),

CONSTRAINT U1\_nhood\_name Unique (nhood\_name)

)

-- End creating the neighborhood table

-- Create officer\_report table

CREATE TABLE officer\_report (

-- Columns for the officer\_report table

officer\_reportID int identity,

report\_date datetime not null,

report\_notes varchar(500) not null,

officerID int not null,

-- Constraints to the officer\_report table

CONSTRAINT PK\_officer\_reportID PRIMARY KEY (officer\_reportID),

CONSTRAINT FK3\_officerID FOREIGN KEY (officerID) REFERENCES officer(officerID),

)

-- End creating the officer\_report table

-- Create member table

CREATE TABLE member (

-- Columns for the member table

memberID int identity,

member\_username varchar(30) not null,

member\_password varchar(30) not null,

nhoodID int not null,

primary\_first\_name varchar(30) not null,

primary\_last\_name varchar(30) not null,

second\_first\_name varchar(30),

second\_last\_name varchar(30),

street\_number varchar(10) not null,

street varchar(30) not null,

city varchar(30) not null,

us\_state varchar(20) not null,

zipcode varchar(10) not null,

cell\_one varchar(20) not null,

cell\_two varchar(20),

home\_phone varchar(20),

email\_one varchar(30) not null,

email\_two varchar(30),

property\_notes varchar(500),

emergency\_notes varchar(500),

member\_status bit not null,

-- Constraints to the member table

CONSTRAINT PK\_memberID PRIMARY KEY (memberID),

CONSTRAINT FK3\_nhoodID FOREIGN KEY (nhoodID) REFERENCES neighborhood(nhoodID),

CONSTRAINT U1\_member\_Username UNIQUE (member\_Username)

)

-- End creating the member table

-- Create vacation\_request table

CREATE TABLE vacation\_request (

-- Columns for the vacation\_request table

vacation\_requestID int identity,

vr\_start\_date datetime not null,

vr\_end\_date datetime not null,

instructions varchar(500),

email\_tag bit not null,

text\_tag bit not null,

memberID int not null,

-- Constraints to the vacation\_requests table

CONSTRAINT PK\_vacation\_requestID PRIMARY KEY (vacation\_requestID),

CONSTRAINT FK4\_memberID FOREIGN KEY (memberID) REFERENCES member(memberID),

)

-- End creating the vacation\_request table

-- create vacation\_patrol table

-- Create vacation\_patrol table

CREATE TABLE vacation\_patrol (

-- Columns for the vacation\_patrol table

vacation\_patrolID int identity,

vp\_datetime datetime not null,

vp\_status varchar(50) not null,

notes varchar(500),

photo\_url varchar(100),

latitude decimal(10,2),

longitude decimal(10,2),

officerID int not null,

vacation\_requestID int not null,

-- Constraints to the vacation\_patrol table

CONSTRAINT PK\_vacation\_patrolID PRIMARY KEY (vacation\_patrolID),

CONSTRAINT FK5\_officerID FOREIGN KEY (officerID) REFERENCES officer(officerID),

CONSTRAINT FK6\_vacation\_requestID FOREIGN KEY (vacation\_requestID) REFERENCES vacation\_request(vacation\_requestID),

)

-- End creating the vacation\_patrol table

-- create text\_mesaage table

-- Create text\_message table

CREATE TABLE text\_message (

-- Columns for the text\_message table

text\_messageID int identity,

text\_date\_time datetime not null,

text\_message varchar(500) not null,

text\_message\_tag bit not null,

officerID int not null,

memberID int not null,

-- Constraints to the text\_message table

CONSTRAINT PK\_text\_messagelID PRIMARY KEY (text\_messageID),

CONSTRAINT FK7\_officerID FOREIGN KEY (officerID) REFERENCES officer(officerID),

CONSTRAINT FK6\_memberID FOREIGN KEY (memberID) REFERENCES member(memberID),

)

GO  
-- End creating the text\_message table  
-- End of table creation

**Data Creation:**

During data creation step, several PROCEDURES were created to insert individual neighborhood, membership and officer data. INSERT statements were also used to input vacation requests, vacation patrol, officer report, and text messaging data.

-- Data creation

-- Drop procedure if exists

IF OBJECT\_ID('dbo.addNeighborhood', 'P') IS NOT NULL

DROP PROCEDURE dbo.addNeighborhood

GO

IF OBJECT\_ID('dbo.addMember', 'P') IS NOT NULL

DROP PROCEDURE dbo.addMember

GO

IF OBJECT\_ID('dbo.addOfficer', 'P') IS NOT NULL

DROP PROCEDURE dbo.addOfficer

GO

-- Procedure to Insert Neighborhood Data

CREATE PROCEDURE addNeighborhood(@nhoodname varchar(50))

AS

BEGIN

INSERT INTO neighborhood(nhood\_name)

VALUES (@nhoodname)

return @@identity

END

GO

-- Procedure to Insert Member Data

CREATE PROCEDURE dbo.addMember(@m\_username varchar(30),

@m\_password varchar(20),

@m\_pfn varchar(30),

@m\_pln varchar(30),

@m\_sfn varchar(30),

@m\_sln varchar(30),

@m\_street\_num varchar(10),

@m\_street varchar(30),

@m\_city varchar(30),

@m\_state varchar(20),

@m\_zipcode varchar(10),

@m\_cell\_one varchar(20),

@m\_cell\_two varchar(20),

@m\_home\_phone varchar(20),

@m\_email\_one varchar(30),

@m\_email\_two varchar(30),

@m\_property varchar(500),

@m\_emergency varchar(500),

@m\_status bit,

@nhoodID int)

AS

BEGIN

INSERT INTO member(

member\_username,

member\_password,

nhoodID,

primary\_first\_name,

primary\_last\_name,

second\_first\_name,

second\_last\_name,

street\_number,

street,

city,

us\_state,

zipcode,

cell\_one,

cell\_two,

home\_phone,

email\_one,

email\_two,

property\_notes,

emergency\_notes,

member\_status)

VALUES (@m\_username,

@m\_password,

@nhoodID,

@m\_pfn,

@m\_pln,

@m\_sfn,

@m\_sln,

@m\_street\_num,

@m\_street,

@m\_city,

@m\_state,

@m\_zipcode,

@m\_cell\_one,

@m\_cell\_two,

@m\_home\_phone,

@m\_email\_one,

@m\_email\_two,

@m\_property,

@m\_emergency,

@m\_status

)

return @@identity

END

GO

-- Procedure to Insert Officer data

CREATE PROCEDURE dbo.addOfficer(@first\_name varchar(30), @last\_name varchar(30), @of\_password varchar(30), @of\_status bit)

AS

BEGIN

INSERT INTO officer(officer\_last\_name, officer\_first\_name, officer\_password, officer\_status)

VALUES (@first\_name, @last\_name, @of\_password, @of\_status)

return @@identity

END

GO

-- Declare variables to hold Neighborhood ID value

DECLARE @m1 AS int

DECLARE @m2 AS int

DECLARE @m3 AS int

-- Execute procedure to add Druid Hills neighborhood and assign neighborhoodID to @m1

EXEC @m1 = dbo.addNeighborhood 'Druid Hills'

-- Add five members to Druid Hills neighborhood

EXEC dbo.addMember 'funnybunny', 'frek123', 'Olga', 'Perera', NULL, NULL, '34', 'Oakdale RD', 'Atlanta',

'GA', '30303', '404-111-2367', NULL, NULL, 'olga@mail.com', NULL, 'Two large dogs on property', 'mom: 650-222-6789', '1', @m1

EXEC dbo.addMember 'sillythings', 'dt3771', 'Jane', 'Smith', 'Paul', 'Smith', '322', 'Briardale RD', 'Atlanta',

'GA', '30303', '404-121-2567', NULL, NULL, 'janesmith@mail.com', 'paulsmith@mail.com', 'Renters on property', 'no contact', '1', @m1

EXEC dbo.addMember 'happyholiday', 'fresdasd', 'Patti', 'Holly', NULL, NULL, '134', 'Oakdale RD', 'Atlanta',

'GA', '30303', '770-541-2555', NULL, NULL, 'pattih@mail.com', NULL, 'Cats on property', 'dad: 543-222-2311', '0', @m1

EXEC dbo.addMember 'wenthome', 'fas999', 'Alex', 'Bikler', NULL, NULL, '811', 'Springdale RD', 'Atlanta',

'GA', '30303', '404-491-5689', NULL, NULL, 'alexb@mail.com', NULL, 'no comment', 'no', '1', @m1

EXEC dbo.addMember 'wannaice', 'shgduuy3', 'John', 'Singh', NULL, NULL, '32', 'Oakdale RD', 'Atlanta',

'GA', '30303', '404-299-2678', NULL, NULL, 'johns@mail.com', NULL, 'Dog on property', 'no comments', '0', @m1

-- Execute procedure to add Briar Woods neighborhood and assign neighborhoodID to @m2

EXEC @m2 = dbo.addNeighborhood 'Briar Woods'

-- Add five members to Briar Woods neighborhood

EXEC dbo.addMember 'nickelpip', 'hdghaue3', 'Andrea', 'Carter', NULL, NULL, '105', 'Cornell RD', 'Atlanta',

'GA', '30303', '404-111-2367', NULL, NULL, 'andrea@mail.com', NULL, NULL, 'other contact: 344-255-6709', '1', @m2

EXEC dbo.addMember 'friendsofus', '3438748b', 'Pavel', 'Smirnov', 'Lana', 'Smernov', '677', 'Harvard RD', 'Atlanta',

'GA', '30303', '404-670-2660', NULL, NULL, 'pavels@mail.com', 'lanas@mail.com', 'Renters on property', NULL, '1', @m2

EXEC dbo.addMember 'fabu45', 'fdhgr44', 'Jennifer', 'Foodman', 'Ricky', 'Nofood', '33', 'Harvard RD', 'Atlanta',

'GA', '30303', '678-455-5725', '404-971-6677', NULL, 'jen@mail.com', NULL, 'Gates', 'dad: 678-908-4312', '1', @m2

EXEC dbo.addMember 'grownman', 'fazzy77', 'Alex', 'Blair', NULL, NULL, '91', 'Cornell RD', 'Atlanta',

'GA', '30303', '404-461-5601', NULL, NULL, 'alexb@mail.com', NULL, 'no comment', 'no', '1', @m2

EXEC dbo.addMember 'princeand', 'dude45', 'John', 'Singh', NULL, NULL, '32', 'Oakdale RD', 'Atlanta',

'GA', '30303', '770-299-2678', NULL, NULL, 'johns@mail.com', NULL, 'Dog on property', 'no comments', '0', @m2

-- Execute procedure to add Briar Woods neighborhood and assign neighborhoodID to @m3

EXEC @m3 = dbo.addNeighborhood 'Emory'

-- Add five members to Briar Woods neighborhood

EXEC dbo.addMember 'frownfm', 'league3', 'Karen', 'Fletcher', NULL, NULL, '222', 'Moreland AVE', 'Atlanta',

'GA', '30307', '404-267-2367', NULL, NULL, 'karenf@mail.com', NULL, NULL, 'other contact: 255-266-6709', '1', @m3

EXEC dbo.addMember 'octopus', '3438748b', 'Harvey', 'Smirnov', 'Lana', 'Smernov', '677', 'Sesame ST', 'Atlanta',

'GA', '30303', '404-670-2660', NULL, NULL, 'pavels@mail.com', 'lanas@mail.com', 'Renters on property', NULL, '1', @m3

EXEC dbo.addMember 'fabulili', 'fdhgr44', 'Lana', 'Poole', 'Ricky', 'Poole', '136', 'Moreland AVE', 'Atlanta',

'GA', '30303', '678-455-5725', '404-971-6677', NULL, 'jen@mail.com', NULL, 'Gates', 'dad: 678-908-4312', '1', @m3

EXEC dbo.addMember 'santa', 'fazzy7777', 'Alexander', 'Miller', NULL, NULL, '91', 'Smith RD', 'Atlanta',

'GA', '30303', '404-461-5601', NULL, NULL, 'alexb@mail.com', NULL, 'no comment', 'no', '1', @m3

EXEC dbo.addMember 'thor', 'dude45', 'John', 'Sloth', NULL, NULL, '332', 'Yale RD', 'Atlanta',

'GA', '30303', '770-779-2098', NULL, NULL, 'johnsl@mail.com', NULL, 'Dog on property', 'no comments', '0', @m3

-- Excute procedure to add officers

EXEC dbo.addOfficer 'Ferguson', 'David', 'happy6788', '1'

EXEC dbo.addOfficer 'Singh', 'Tony', 'cars743', '1'

EXEC dbo.addOfficer 'Baxter', 'Jeff', 'hhjja44', '1'

EXEC dbo.addOfficer 'John', 'Smith', 'shja44', '1'

GO

--Insert Vacation Request records

INSERT INTO vacation\_request(vr\_start\_date, vr\_end\_date, instructions, email\_tag, text\_tag, memberID)

VALUES ('2018-12-01','2018-12-08','pet sitter will be on property occasionally',1,1,1)

INSERT INTO vacation\_request(vr\_start\_date, vr\_end\_date, instructions, email\_tag, text\_tag, memberID)

VALUES ('2018-12-10','2018-12-31','no one should be on property',1,0,3)

INSERT INTO vacation\_request(vr\_start\_date, vr\_end\_date, instructions, email\_tag, text\_tag, memberID)

VALUES ('2018-12-10','2018-12-15','out of contry',1,0,7)

-- Select vacation request records

SELECT \* FROM vacation\_request

-- Insert Vacation Patrol records

INSERT INTO vacation\_patrol(vp\_datetime, vp\_status, notes, photo\_url, latitude, longitude, vacation\_requestID, officerID)

VALUES ('2018-12-15','OK','Property was secured by Officer Baxter',NULL,87.345,31.432,1,3)

INSERT INTO vacation\_patrol(vp\_datetime, vp\_status, notes, photo\_url, latitude, longitude, vacation\_requestID, officerID)

VALUES ('2018-12-15','OK','Mail was moved to back porch. Officer Baxter',NULL,87.565,32.412,2,3)

INSERT INTO vacation\_patrol(vp\_datetime, vp\_status, notes, photo\_url, latitude, longitude, vacation\_requestID, officerID)

VALUES ('2018-12-15','OK','Property was secured by Officer Baxter',NULL,87.555,33.432,3,3)

INSERT INTO vacation\_patrol(vp\_datetime, vp\_status, notes, photo\_url, latitude, longitude, vacation\_requestID, officerID)

VALUES ('2018-12-16','Attention needed','Front gate opened. Contacted homeowner. Officer Singh',NULL,87.35,32.412,1,2)

INSERT INTO vacation\_patrol(vp\_datetime, vp\_status, notes, photo\_url, latitude, longitude, vacation\_requestID, officerID)

VALUES ('2018-12-16','OK','Property was secured by Officer Singh',NULL,87.565,33.412,2,2)

INSERT INTO vacation\_patrol(vp\_datetime, vp\_status, notes, photo\_url, latitude, longitude, vacation\_requestID, officerID)

VALUES ('2018-12-16','OK','Property checked by Officer Singh',NULL,87.555,33.412,3,2)

-- Select vacation request records

SELECT \* FROM vacation\_patrol

--Insert Officer Report records

INSERT INTO officer\_report(report\_date, report\_notes, officerID)

VALUES ('2018-12-15','December 15th report by Officer Baxter: I patrolled three vacation properties, responded to several membership concerns.',3)

INSERT INTO officer\_report(report\_date, report\_notes, officerID)

VALUES ('2018-12-16','December 16th report by Officer Singh: I spoke to a member regarding an incident. Nothing else to report.',2)

-- Select Officer report records

SELECT \* FROM officer\_report

--Insert Text Message records

INSERT INTO text\_message(text\_date\_time, text\_message, text\_message\_tag, officerID, memberID)

VALUES ('2018-12-16','Theft on our property',0,2,5)

INSERT INTO text\_message(text\_date\_time, text\_message, text\_message\_tag, officerID, memberID)

VALUES ('2018-12-16','Will investigate asap',1,2,5)

-- Select Text Message records

SELECT \* FROM text\_message

GO

**Data Manipulation:**

Data manipulation is the process of changing data and consists of updating and deleting database records. In general I advise against deleting database records, because this action is permanent and might cause data integrity issues if specific data was related to another data through relationship connection.

-- Updating vacation request records: set start date to today for vacation request #1

DECLARE @newstartdate datetime

SET @newstartdate = getdate()

UPDATE vacation\_request SET vr\_start\_date = @newstartdate WHERE vacation\_requestID=1

-- Procedure to Update vacation request start date

IF OBJECT\_ID('dbo.setVRStartDate', 'P') IS NOT NULL

DROP FUNCTION dbo.setVRStartDate

GO

CREATE PROCEDURE dbo.setVRStartDate (@startDate datetime, @vacation\_requestID int)

AS BEGIN

UPDATE vacation\_request

SET vr\_start\_date = @startDate WHERE vacation\_requestID = @vacation\_requestID

END

GO

EXEC dbo.setVRStartDate '12-02-2018', 1

SELECT \* FROM vacation\_request

**Answering Data Questions:**

Views is a great way to present your data. A View to generate a list of today’s vacation properties and View to review vacation patrol data sorted by request and member are coded below.

-- Creating Views

-- VIEW to generate current (today's) vacation requests list for individual officer shift.

IF OBJECT\_ID('dbo.patrol\_VacationList', 'V') IS NOT NULL

DROP VIEW dbo.patrol\_VacationList

GO

IF OBJECT\_ID('dbo.getTodayDate') IS NOT NULL

DROP FUNCTION dbo.getTodayDate

GO

-- Step 1 -> create function to get current date

CREATE FUNCTION dbo.getTodayDate ()

RETURNS datetime AS

BEGIN

DECLARE @today AS datetime

SET @today = getdate()

RETURN @today

END

GO

-- create view to today's list of vacation requests

CREATE VIEW dbo.patrol\_VacationList AS

SELECT

member.primary\_first\_name +' '+ member.primary\_last\_name AS MemberName,

member.street\_number +' '+ member.street AS MemberAddress,

member.street AS Street,

vacation\_request.instructions AS PatrolInstructions,

member.cell\_one AS PrimaryCell,

member.email\_one AS Email

FROM vacation\_request

JOIN member ON vacation\_request.memberID = member.memberID

WHERE dbo.getTodayDate() BETWEEN vr\_start\_date AND vr\_end\_date

GO

-- Select records from View ordered by Street

SELECT MemberName, MemberAddress, PatrolInstructions, PrimaryCell, Email FROM patrol\_VacationList ORDER BY Street

GO

-- View to create a list of vacation patrol for a specific vacation request

IF OBJECT\_ID('dbo.member\_VacationPatrol', 'V') IS NOT NULL

DROP VIEW dbo.member\_VacationPatrol

GO

CREATE VIEW member\_VacationPatrol AS

SELECT

member.memberID AS MemberID,

member.primary\_first\_name +' '+ member.primary\_last\_name AS MemberName,

member.street\_number +' '+ member.street AS MemberAddress,

member.street AS Street,

vacation\_request.vr\_start\_date AS StartDate,

vacation\_patrol.vp\_datetime AS PatrolDate,

vacation\_patrol.vp\_status AS PatrolStatus

FROM member

JOIN vacation\_request ON member.memberID = vacation\_request.memberID

JOIN vacation\_patrol ON vacation\_patrol.vacation\_requestID = vacation\_request.vacation\_requestID

GO

SELECT MemberName, MemberAddress, StartDate, PatrolDate, PatrolStatus FROM member\_VacationPatrol

ORDER BY MemberID

GO

**Implementation: Web Interface**

It was determined by organizational stakeholders that several web interfaces will be the best solution for various data users, because it provides access to data from any device and location. Management web interface provides an instant access to the entire organizational data with “easy to use” interface. Management access allows unrestricted creation and manipulation of organizational data and is available only to an employee responsible for membership and officer management.

Screenshot 1: Main Menu

A screenshot of a cell phone

Description generated with very high confidence

Screenshot 2: Membership records by Neighborhood (nhoodID relationship). Each darker aqua colored button provides access and shows number of existing records (members) for each neighborhood.

**A screenshot of a stereo

Description generated with very high confidence**

Screenshot 3: Officer records with dependent records for Officer Report, Text Message, Vacation Patrol for individual officers.

A screenshot of a video game

Description generated with high confidence

Screenshot 4: Inline version of Insert new officer information into the database.

A screenshot of a stereo

Description generated with very high confidence

Screenshot 5: Vacation patrol reports sorted by individual vacation requests.

A screenshot of a video game

Description generated with high confidence