Database Design Document

## Create Table Statements

Refer to the database design at the end of this document, in figure 1, to better see the connections between tables as well as my explanation for my design choices.

CREATE TABLE User\_Type (

Type varchar(2),

Name varchar(15) NOT NULL,

PRIMARY KEY (Type)

);

CREATE TABLE Person (

ID int,

Type varchar(2),

DateAdded date NOT NULL,

DateLastActive date,

PRIMARY KEY (ID),

FOREIGN KEY (Type) REFERENCES User\_Type(Type)

);

CREATE TABLE Login (

ID int,

Username varchar(50) NOT NULL,

Password varchar(16) NOT NULL,

FOREIGN KEY (ID) REFERENCES Person(ID)

);

CREATE TABLE Candidate (

ID int,

FName varchar(50) NOT NULL,

LName varchar(50) NOT NULL,

Address varchar(50) NOT NULL,

PhoneNum varchar(50) NOT NULL,

Votes int,

FOREIGN KEY (ID) REFERENCES Person(ID)

);

CREATE TABLE Voter (

ID int,

FName varchar(50) NOT NULL,

LName varchar(50) NOT NULL,

Address varchar(50) NOT NULL,

PhoneNum varchar(50) NOT NULL,

Voted bit NOT NULL,

Valid\_User bit DEFAULT 0,

FOREIGN KEY (ID) REFERENCES Person(ID)

);

CREATE TABLE Admin (

ID int,

FName varchar(50) NOT NULL,

LName varchar(50) NOT NULL,

Address varchar(50) NOT NULL,

PhoneNum varchar(50) NOT NULL,

FOREIGN KEY (ID) REFERENCES Person(ID)

);

CREATE TABLE Polling\_Officer (

ID int,

FName varchar(50) NOT NULL,

LName varchar(50) NOT NULL,

Address varchar(50) NOT NULL,

PhoneNum varchar(50) NOT NULL,

FOREIGN KEY (ID) REFERENCES Person(ID)

);

## Inserting Data

INSERT INTO User\_Type

VALUES ('AD', 'Admin'),

('CA', 'Candidate'),

('PO', 'Polling Officer'),

('VO', 'Voter');

INSERT INTO Person (Type, DateAdded)

VALUES ('PO', '2023-01-01'),

('PO', '2023-01-01'),

('PO', '2023-01-01'),

('AD', '2023-01-01'),

('CA', '2023-01-01'),

('AD', '2023-01-01'),

('VO', '2023-01-01'),

('VO', '2023-01-01'),

('VO', '2023-01-01'),

('VO', '2023-01-01'),

('VO', '2023-01-01'),

('VO', '2023-01-01'),

('VO', '2023-01-01'),

('CA', '2023-01-01'),

('VO', '2023-01-01'),

('VO', '2023-01-01'),

('VO', '2023-01-01'),

('VO', '2023-01-01'),

('VO', '2023-01-01'),

('VO', '2023-01-01'),

('VO', '2023-01-01'),

('VO', '2023-01-01'),

('VO', '2023-01-01'),

('VO', '2023-01-01'),

('VO', '2023-01-01'),

('VO', '2023-01-01'),

('CA', '2023-01-01'),

('AD', '2023-01-01'),

('VO', '2023-01-01');

ALTER TABLE Login

ADD UNIQUE (Username);

--Added the

INSERT INTO Login

VALUES (1, 'PO1', 'Pass'),

(2, 'PO2', 'PASS'),

(3, 'PO3', 'PaSs'),

(4,'AD1', 'PassWord'),

(5, 'CA1', 'test'),

(6, 'AD2', 'TEST'),

(7, 'VO1', 'TeSt'),

(8, 'VO2', 'passw0rd'),

(9, 'VO3', 'p@ssword'),

(10, 'VO4', 'p@ssw0rd'),

(11, 'VO5', 'p@55w0rd'),

(12, 'VO6', 'passwo4d'),

(13, 'VO7', 'voterman'),

(14, 'CA2', 'vote2win'),

(15, 'VO8', 'popvote'),

(16, 'VO9', 'majora'),

(17, 'VO10', 'skyrim'),

(18, 'VO11', 'hornoplenty'),

(19, 'VO12', 'adventuretime'),

(20, 'VO13', 'marceline'),

(21, 'VO14', 'jakethedog'),

(22, 'VO15', 'finthehuman'),

(23, 'VO16', 'legomyeggo'),

(24, 'VO17', 'pixelking'),

(25, 'VO18', 'jester'),

(26, 'VO19', 'frenchpress'),

(27, 'CA3', 'frenchhorn'),

(28, 'AD3', 'chainmaille'),

(29, 'VO20', 'blueberry');

INSERT INTO Candidate

VALUES(5, 'Paul', 'Sharp', '123 Fake Street', '9025558576', 2),

(14, 'Jimi', 'Hendrix', '69 Woodstock Lane', '9025556969', 15),

(27, 'Wood', 'Board', '5 Juniper Street', '9025551234', 3);

INSERT INTO Voter

VALUES(7, 'Caitlyn', 'Stark', '140 Fake Street', '9025558576', 0),

(8, 'Janis', 'Joplin', '70 Woodstock Lane', '9025556969', 0),

(9, 'Kurt', 'Cobain', '10 Heart Lane', '9025551234', 0),

(10, 'Chad', 'Kroeger', '88 Nickel Way', '9025552004', 0),

(11, 'Benjamin', 'Kowalewicz', '2 Sparrow Crescent', '9025558574', 0),

(12, 'Ian', 'DSa', '2 Sparrow Crescent', '9025555390', 1),

(13, 'Jonathan', 'Gallant', '2 Sparrow Crescent', '9025557000', 1),

(15, 'Aaron', 'Solowoniuk', '2 Sparrow Crescent', '9025559453', 0),

(16, 'Krist', 'Novoselic', '10 Heart Lane', '9025558675', 1),

(17, 'Dave', 'Grohl', '10 Heart Lane', '9025558473', 1),

(18, 'Mitch', 'Mitchell', '69 Woodstock Lane', '9025550012', 0),

(19, 'Noel', 'Redding', '69 Woodstock Lane', '9025559173', 0),

(20, 'Billy', 'Cox', '69 Woodstock Lane', '9025559862', 0),

(21, 'Nathan', 'Feuerstein', '19 Breathe Boulevard', '9025557284', 1),

(22, 'Mel', 'Bryant', '5 Sunrise Way', '9025553434', 1),

(23, 'Colm', 'McGuinness', '150 Hoist Street', '9025557685', 0),

(24, 'Tyler', 'Joseph', '21 Mistle Street', '9025557859', 0),

(25, 'Josh', 'Dun', '21 Mistle Street', '9025552134', 0),

(26, 'Ash', 'Ketchum', '6 Pokemon Way', '9025559005', 0),

(29, 'Finn', 'Mertens', '1 Treehouse', '9025558702', 0);

INSERT INTO Admin

VALUES(4, 'Rob', 'Stark', '200 Winterfell Hold', '9025556000'),

(6, 'Billy', 'Joel', '88 Piano Court', '9025557794'),

(28, 'Jeff', 'Buckley', '654 Song Plaza', '9025557777');

INSERT INTO Polling\_Officer

VALUES(1, 'Steve', 'Irwin', '2001 Sycamore Tree Tunnel', '9025558275'),

(2, 'Marceline', 'The-Vampire', '8 ooo Cul-de-Sac', '9025550000'),

(3, 'Bonnibel', 'Bubblegum', '101 Candy Kingdom', '9025558071');

## Specific Tasks

--Register Voter

BEGIN TRANSACTION Register1

INSERT INTO Person (Type, DateAdded)

VALUES ('VO', '2023-04-09');

SAVE TRANSACTION Register1

DECLARE @ID AS int

SELECT @ID = MAX(ID)

FROM Person

PRINT @ID;

SAVE TRANSACTION Register1

INSERT INTO Voter (ID, FName, LName, Address, PhoneNum, Voted)

VALUES (@ID, 'Harry', 'Potter', '4 Privet Drive', '9025558764', 0);

SAVE TRANSACTION Register1

INSERT INTO Login (ID, Username, Password)

VALUES (@ID, 'Pen', 'Managed');

COMMIT TRANSACTION Register1;

--Register Candidate

BEGIN TRANSACTION RegisterCandidate1

INSERT INTO Person (Type, DateAdded)

VALUES ('VO', '2023-04-09');

SAVE TRANSACTION RegisterCandidate1

DECLARE @ID AS int

SELECT @ID = MAX(ID)

FROM Person

PRINT @ID;

SAVE TRANSACTION RegisterCandidate1

INSERT INTO Candidate (ID, FName, LName, Address, PhoneNum, Votes)

VALUES (@ID, 'Albus', 'Dumbledore', 'Hogwarts', '9025558743', 0);

SAVE TRANSACTION RegisterCandidate1

INSERT INTO Login (ID, Username, Password)

VALUES (@ID, 'Lemon', 'Lemon');

COMMIT TRANSACTION RegisterCandidate1;

--Updating candidate and voter info

UPDATE Candidate

SET FName = @var2, LName = @var3, Address = @var4, PhoneNum = @var5, Votes = @var6

WHERE ID = @var1;

UPDATE Voter

SET FName = @var2, LName = @var3, Address = @var4, PhoneNum = @var5, Voted = @var6, Valid\_User = @var7

WHERE ID = @var1;

--Changing login info for valid user

DECLARE @Valid AS bit

SELECT @Valid = Valid\_User

FROM Voter

WHERE ID = 38

PRINT @Valid;

IF @Valid = 1

BEGIN

UPDATE Login

SET Username = 'Tester', Password = 'PASSWORD'

WHERE ID = 38;

END

ELSE

PRINT 'Non-Valid User';

--Deleting a specific user

--First find out what type of user that account is

SELECT Name, Person.ID

FROM Person

INNER JOIN User\_Type

ON Person.Type = User\_Type.Type

WHERE Person.ID = 20;

--In this example the account was of a voter

--So that table is deleted from, where the

--id matches the account to be deleted

DELETE FROM Voter

WHERE ID = 20;

--The account is deleted from the Login table, where the

--id matches the account to be deleted

DELETE FROM Login

WHERE ID = 20;

--The account is deleted from the Person table, where the

--id matches the account to be deleted

DELETE FROM Person

WHERE ID = 20;

--Use this to clear all data from a table without deleting the schema itself.

--Simply repeat this for each table to have data deleted from.

DELETE FROM Candidate;

--The following views do what the views say in regards to candidates

--These were all situations I had to write a script for.

CREATE VIEW [Total Voting Record] AS

SELECT FName, LName, Votes

FROM Candidate;

CREATE VIEW [Top 2] AS

SELECT TOP 2 FName, LName, Votes

FROM Candidate

ORDER BY Votes DESC;

CREATE VIEW [Winner] AS

SELECT TOP 1 FName, LName, Votes

FROM Candidate

ORDER BY Votes DESC;

CREATE VIEW [Least Votes] AS

SELECT TOP 1 FName, LName, Votes

FROM Candidate

ORDER BY Votes;

CREATE VIEW [Votes between 5-15] AS

SELECT FName, LName, Votes

FROM Candidate

WHERE Votes > 4 AND Votes < 16;

I decided to have a separate table of the account types and codes for each. It makes navigating the database a bit harder but ensures that all IDs are unique across all users without having to use characters. Also has date added and date last active in a table along with the ID and type of account. When the account is accessed the date last active field will update with that date stamp. Possibly in the future could implement something where unused accounts are safely deleted after a certain amount of time of inactivity. Theres a table that’s organized by ID that contains the usernames and passwords. The passwords shouldn’t be stored in plain text, this is for demonstration purposes only. Then there’s 4 tables for each type of person that there is data stored for. The voter, admin, polling officer and candidates. All the tables depend on the person table. But this allows them all to be connected and only depend on the one table for that connection.

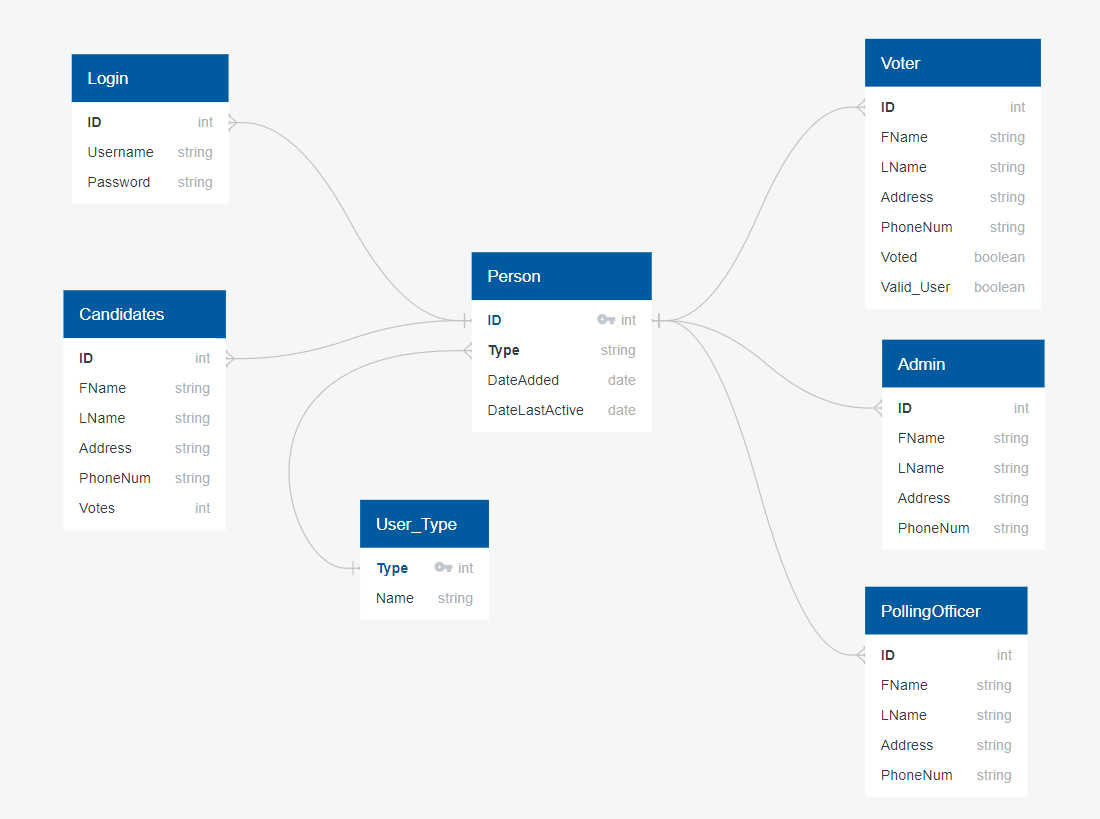


Figure : Database Diagram