Database Design

Thomas Jewers

Database Diagram

This polling system is very simple and only uses an Admin table, PollingOfficer table, Voters table, and candidates table. Candidates do not have a total vote column because the votes are stored in the voters VotedFor column to ensure the numbers are correct and have a record of who voted.

Diagram

Description automatically generated

Table Creation Query

BEGIN TRANSACTION;

CREATE TABLE candidates (

CandidateId INT PRIMARY KEY,

CandidateName VARCHAR(50) NOT NULL)

CREATE TABLE Voters (

VoterId INT PRIMARY KEY,

VoterName VARCHAR(50) NOT NULL,

VoterAddress VARCHAR(100) NOT NULL,

VoterEmail VARCHAR(50) NOT NULL,

VoterPass VARCHAR(50) NOT NULL,

VotedFor INT FOREIGN KEY REFERENCES Candidates(candidateId)

)

CREATE TABLE Admins (

AdminId INT PRIMARY KEY,

AdminName VARCHAR(50) NOT NULL,

AdminPass VARCHAR(50) NOT NULL,

)

CREATE TABLE PollingOfficer (

OfficerId INT PRIMARY KEY,

OfficerName VARCHAR(50) NOT NULL,

OfficerPass VARCHAR(50) NOT NULL,

)

COMMIT;

Insert Data into Tables

BEGIN TRANSACTION;

INSERT INTO candidates (CandidateId, CandidateName)

VALUES (1, 'Jane Doe'),

(2, 'John Smith'),

(3, 'Sarah Johnson');

INSERT INTO Voters (VoterId, VoterName, VoterAddress, VoterEmail, VoterPass, VotedFor)

VALUES (1, 'John Smith', '123 Main St, Anytown USA', 'johnsmith@email.com', 'password1', 2),

(2, 'Sarah Johnson', '456 Park Ave, Citytown USA', 'sarahjohnson@email.com', 'password2', 1),

(3, 'Michael Chen', '789 Elm St, Anytown USA', 'michaelchen@email.com', 'password3', 3),

(4, 'Karen Lee', '101 Oak St, Citytown USA', 'karenlee@email.com', 'password4', 1),

(5, 'Robert Nguyen', '222 Pine Ave, Anytown USA', 'robertnguyen@email.com', 'password5', 1),

(6, 'Jennifer Kim', '333 Maple St, Citytown USA', 'jenniferkim@email.com', 'password6', 2),

(7, 'Mark Davis', '444 Cedar Ave, Anytown USA', 'markdavis@email.com', 'password7', 2),

(8, 'Emily Lee', '555 Oak St, Citytown USA', 'emilylee@email.com', 'password8', 1);

COMMIT;

Register a new candidate.

BEGIN TRANSACTION

INSERT INTO candidates (CandidateId, CandidateName)

VALUES (4, 'John Smith');

COMMIT;

Update a voter’s login details.

BEGIN TRANSACTION

UPDATE voters

SET VoterEmail = 'newemail@example.com',

VoterPass = 'newpassword'

WHERE VoterId = 1234;

COMMIT;

Remote a specific voter from the database.

BEGIN TRANSACTION

DELETE FROM voters

WHERE VoterId = 1;

COMMIT;

Delete the data without removing the schema.

BEGIN TRANSACTION;

DELETE FROM voters;

DELETE FROM candidates;

DELETE FROM PollingOfficer;

DELETE FROM Admins;

COMMIT;

Show the top 2 candidates with the most votes.

BEGIN TRANSACTION;

SELECT TOP 2 CandidateName, COUNT(\*) AS Votes

FROM candidates

JOIN voters ON candidates.CandidateId = voters.VotedFor

GROUP BY CandidateName

ORDER BY COUNT(\*) DESC;

COMMIT;

Show candidates with at least one vote.

BEGIN TRANSACTION;

SELECT DISTINCT candidates.CandidateName

FROM candidates

JOIN voters ON candidates.CandidateId = voters.VotedFor;

COMMIT;

Check which candidates have between 3 and 15 votes (I used 3 instead of 5)

BEGIN TRANSACTION;

SELECT CandidateName, COUNT(\*) AS Votes

FROM candidates

JOIN voters ON candidates.CandidateId = voters.VotedFor

GROUP BY CandidateName

HAVING COUNT(\*) BETWEEN 3 AND 15

COMMIT;

Show the voter records (voter id, name, and who they voted for)

BEGIN TRANSACTION;

SELECT VoterId, VoterName, CandidateName

FROM voters

JOIN candidates ON voters.VotedFor = candidates.CandidateId;

COMMIT;

Check the winning candidate along with their votes.

BEGIN TRANSACTION;

SELECT TOP 1 CandidateName, COUNT(\*) AS Votes

FROM candidates

JOIN voters ON candidates.CandidateId = voters.VotedFor

GROUP BY CandidateName

ORDER BY COUNT(\*) DESC

COMMIT;