

```
CREATE DATABASE GuildElectionDBExam2
    DEFAULT CHARACTER SET = 'utf8mb4';
```

```
USE GuildElectionDBExam2;
```

```
-- Step 1: Create the User Table
```

```
CREATE TABLE Users (
    UserID INT AUTO_INCREMENT PRIMARY KEY,
    Username VARCHAR(50) NOT NULL UNIQUE,
    PasswordHash VARCHAR(255) NOT NULL,
    Role ENUM('Admin', 'Voter') NOT NULL,
    Email VARCHAR(255) NOT NULL UNIQUE,
    CreatedAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

```
-- Step 2: Create the Faculty Table
```

```
CREATE TABLE Faculty (
    FacultyID INT AUTO_INCREMENT PRIMARY KEY,
    FacultyName VARCHAR(255) NOT NULL UNIQUE
);
```

```
-- Step 3: Create the Student Table
```

```
CREATE TABLE Students (
    StudentID INT AUTO_INCREMENT PRIMARY KEY,
    Name VARCHAR(255) NOT NULL,
    DateOfBirth DATE,
    EnrollmentYear INT,
    Eligibility_Status BOOLEAN NOT NULL,
    FacultyID INT NOT NULL,
    IsInternational BOOLEAN NOT NULL,
    FOREIGN KEY (FacultyID) REFERENCES Faculty(FacultyID) ON DELETE CASCADE
);
```

```
-- Step 4: Create the Election Table
```

```
CREATE TABLE Elections (
    ElectionID INT AUTO_INCREMENT PRIMARY KEY,
    ElectionName VARCHAR(255) NOT NULL,
    StartDate DATE NOT NULL,
    EndDate DATE NOT NULL,
    Status ENUM('Upcoming', 'Active', 'Completed') NOT NULL DEFAULT 'Upcoming'
);
```

```
-- Step 5: Create the Position Table
```

```
CREATE TABLE Positions (
```

```

PositionID INT AUTO_INCREMENT PRIMARY KEY,
PositionName VARCHAR(255) NOT NULL,
ElectionID INT NOT NULL,
RestrictedTo ENUM('Faculty', 'Residence', 'Open', 'International') NOT NULL,
Description TEXT,
FOREIGN KEY (ElectionID) REFERENCES Elections(ElectionID) ON DELETE CASCADE,
UNIQUE (PositionName, ElectionID) -- Ensure no two candidates can run for the same
position in the same election
);

```

-- Step 6: Create the Candidates Table

```

CREATE TABLE Candidates (
CandidateID INT AUTO_INCREMENT PRIMARY KEY,
StudentID INT NOT NULL,
PositionID INT NOT NULL,
CampaignMessage TEXT,
FOREIGN KEY (StudentID) REFERENCES Students(StudentID) ON DELETE CASCADE,
FOREIGN KEY (PositionID) REFERENCES Positions(PositionID) ON DELETE CASCADE
);

```

-- Step 7: Create the Votes Table

```

CREATE TABLE Votes (
VoteID INT AUTO_INCREMENT PRIMARY KEY,
VoterID INT NOT NULL,
CandidateID INT NOT NULL,
ElectionID INT NOT NULL,
VoteMethod ENUM('Online') NOT NULL,
VoteTime TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
FOREIGN KEY (VoterID) REFERENCES Students(StudentID) ON DELETE CASCADE,
FOREIGN KEY (CandidateID) REFERENCES Candidates(CandidateID) ON DELETE
CASCADE,
FOREIGN KEY (ElectionID) REFERENCES Elections(ElectionID) ON DELETE CASCADE
);

```

-- Step 8: Create a View for Vote Counting

```

CREATE VIEW VoteCount AS
SELECT
c.CandidateID,
c.PositionID,
p.PositionName,
COUNT(v.VoteID) AS TotalVotes
FROM
Candidates c
LEFT JOIN Votes v ON c.CandidateID = v.CandidateID

```

```
LEFT JOIN Positions p ON c.PositionID = p.PositionID
GROUP BY c.CandidateID;
```

-- Step 9: Create a Trigger for Voting Eligibility

```
DELIMITER $$
```

```
CREATE TRIGGER Check_Student_Eligibility
```

```
BEFORE INSERT ON Votes
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    DECLARE studentEligible INT;
```

```
    DECLARE positionRestrictedTo ENUM('Faculty', 'Residence', 'Open', 'International');
```

```
    DECLARE isInternational BOOLEAN;
```

-- Check if the student is eligible

```
    SELECT COUNT(*) INTO studentEligible
```

```
    FROM Students
```

```
    WHERE StudentID = NEW.VoterID AND Eligibility_Status = TRUE;
```

-- Check the restriction of the position

```
    SELECT RestrictedTo INTO positionRestrictedTo
```

```
    FROM Positions
```

```
    WHERE PositionID = NEW.CandidateID;
```

-- Check if the student is international

```
    SELECT IsInternational INTO isInternational
```

```
    FROM Students
```

```
    WHERE StudentID = NEW.VoterID;
```

-- If not eligible, raise an error

```
    IF studentEligible = 0 THEN
```

```
        SIGNAL SQLSTATE '45000'
```

```
        SET MESSAGE_TEXT = 'Student is not eligible to vote.';
```

```
    END IF;
```

-- If national student voting for international position, raise an error

```
    IF isInternational = FALSE AND positionRestrictedTo = 'International' THEN
```

```
        SIGNAL SQLSTATE '45000'
```

```
        SET MESSAGE_TEXT = 'National students cannot vote for international representatives.';
```

```
    END IF;
```

```
END$$
```

```
DELIMITER ;
```

-- Step 10: Create an Audit Log Table

```
CREATE TABLE VoteLogs (
```

```

LogID INT AUTO_INCREMENT PRIMARY KEY,
VoterID INT NOT NULL,
CandidateID INT NOT NULL,
ElectionID INT NOT NULL,
ActionTime TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
Action ENUM('Vote Cast', 'Vote Revoked') NOT NULL,
FOREIGN KEY (VoterID) REFERENCES Students(StudentID),
FOREIGN KEY (CandidateID) REFERENCES Candidates(CandidateID),
FOREIGN KEY (ElectionID) REFERENCES Elections(ElectionID)
);

```

-- Step 11: Insert Test Data

-- Faculty Data

```

INSERT INTO Faculty (FacultyName) VALUES
('Law'),
('Medicine'),
('Engineering'),
('Social Sciences'),
('Business');

```

-- Students Data

```

INSERT INTO Students (Name, DateOfBirth, EnrollmentYear, Eligibility_Status, FacultyID,
IsInternational) VALUES
('Alice Johnson', '2001-01-01', 2022, TRUE, 1, FALSE),
('Bob Smith', '2001-02-01', 2022, FALSE, 2, FALSE),
('Charlie Brown', '2001-03-01', 2023, TRUE, 3, FALSE),
('Diana Prince', '2001-04-01', 2022, TRUE, 4, FALSE),
('Eve Adams', '2001-05-01', 2022, TRUE, 5, FALSE),
('Frank White', '2001-06-01', 2023, TRUE, 1, TRUE),
('Grace Black', '2001-07-01', 2022, TRUE, 2, FALSE),
('Henry Green', '2001-08-01', 2023, TRUE, 3, FALSE),
('Ivy Blue', '2001-09-01', 2022, TRUE, 4, TRUE),
('Jack Gold', '2001-10-01', 2022, TRUE, 5, FALSE),
('Kate Silver', '2001-01-01', TRUE, 1, FALSE),
('Leo Copper', '2000-02-01', TRUE, 2, FALSE),
('Mia Amber', '2000-03-01', TRUE, 3, TRUE),
('Noah Jade', '2000-04-01', TRUE, 4, FALSE),
('Olivia Pearl', '2000-05-01', TRUE, 5, FALSE),
('Paul Onyx', '2000-06-01', TRUE, 1, TRUE),
('Quinn Ruby', '2000-07-01', TRUE, 2, FALSE),
('Ryan Sapphire', '2000-08-01', TRUE, 3, FALSE),
('Sophia Emerald', '2000-09-01', TRUE, 4, FALSE),
('Tom Diamond', '2000-11-01', TRUE, 5, FALSE),
('Uma Quartz', '2000-10-01', TRUE, 1, FALSE),

```

```
(
'Victor Opal', '2000-12-01', TRUE, 2, TRUE),
'Wendy Topaz', '2002-01-01', TRUE, 3, FALSE),
'Xander Garnet', '2002-02-01', TRUE, 4, FALSE),
'Yara Amethyst', '2002-03-01', TRUE, 5, FALSE),
'Zane Crystal', '2002-04-01', TRUE, 1, FALSE),
'Abby Flint', '2002-05-01', TRUE, 2, FALSE),
'Ben Steel', '2002-06-01', TRUE, 3, FALSE),
'Cara Bronze', '2002-07-01', TRUE, 4, FALSE),
'Dean Iron', '2002-08-01', TRUE, 5, FALSE);
```

-- Elections Data

```
INSERT INTO Elections (ElectionName, StartDate, EndDate, Status) VALUES
('Guild Elections 2024', '2024-12-01', '2024-12-15', 'Upcoming');
```

-- Positions Data

```
INSERT INTO Positions (PositionName, ElectionID, RestrictedTo, Description) VALUES
('Guild President', 1, 'Open', 'Leads the student guild.'),
('Law Faculty Representative', 1, 'Faculty', 'Represents the Law Faculty.'),
('Medicine Faculty Representative', 1, 'Faculty', 'Represents the Medicine Faculty.'),
('Engineering Faculty Representative', 1, 'Faculty', 'Represents the Engineering Faculty.'),
('Social Sciences Faculty Representative', 1, 'Faculty', 'Represents the Social Sciences Faculty.'),
('Business Faculty Representative', 1, 'Faculty', 'Represents the Business Faculty.'),
('International Student Representative', 1, 'International', 'Represents the needs of international students.');
```

-- Candidates Data

```
INSERT INTO Candidates (StudentID, PositionID, CampaignMessage) VALUES
(1, 1, 'Vote for me for a better guild!'),    -- Alice Johnson for Guild President
(4, 2, 'Your voice in the Law Faculty!'),    -- Diana Prince for Law Faculty Representative
(6, 3, 'Healthcare for all!'),               -- Frank White for Medicine Faculty Representative
(8, 4, 'Engineering solutions for the future!'), -- Henry Green for Engineering Faculty Representative
(10, 5, 'Social issues matter!'),            -- Jack Gold for Social Sciences Faculty Representative
(7, 6, 'Business for the students!'),        -- Grace Black for Business Faculty Representative
(9, 7, 'International student rights!');     -- Ivy Blue for International Student Representative
```

-- Votes Data

```
INSERT INTO Votes (VoterID, CandidateID, ElectionID, VoteMethod) VALUES
(1, 1, 1, 'Online'),
(3, 1, 1, 'Online'),
(4, 2, 1, 'Online'),
```

(5, 3, 1, 'Online'),
(6, 4, 1, 'Online'),
(7, 5, 1, 'Online'),
(8, 6, 1, 'Online'),
(9, 7, 1, 'Online'),
(10, 2, 1, 'Online'),
(11, 3, 1, 'Online'),
(12, 4, 1, 'Online'),
(13, 5, 1, 'Online'),
(14, 6, 1, 'Online'),
(15, 1, 1, 'Online'),
(16, 2, 1, 'Online'),
(17, 3, 1, 'Online'),
(18, 4, 1, 'Online'),
(19, 5, 1, 'Online'),
(20, 6, 1, 'Online'),
(21, 1, 1, 'Online'),
(22, 2, 1, 'Online'),
(23, 3, 1, 'Online'),
(24, 4, 1, 'Online'),
(25, 5, 1, 'Online'),
(26, 6, 1, 'Online'),
(27, 1, 1, 'Online'),
(28, 2, 1, 'Online'),
(29, 3, 1, 'Online'),
(30, 4, 1, 'Online');
,
(10, 2, 1),
(11, 3, 1),
(12, 4, 1),
(13, 5, 1),
(14, 6, 1),
(15, 1, 1),
(16, 2, 1),
(17, 3, 1),
(18, 4, 1),
(19, 5, 1),
(20, 6, 1),
(21, 1, 1),
(22, 2, 1),
(23, 3, 1),
(24, 4, 1),
(25, 5, 1),
(26, 6, 1),

(27, 1, 1),
(28, 2, 1),
(29, 3, 1),
(30, 4, 1);

-- Vote Logs Data

INSERT INTO VoteLogs (VoterID, CandidateID, ElectionID, Action) VALUES

(1, 1, 1, 'Vote Cast'),
(2, 1, 1, 'Vote Cast'),
(3, 2, 1, 'Vote Cast'),
(4, 2, 1, 'Vote Cast'),
(5, 3, 1, 'Vote Cast'),
(6, 1, 1, 'Vote Revoked'),
(7, 3, 1, 'Vote Cast'),
(8, 4, 1, 'Vote Cast'),
(9, 5, 1, 'Vote Cast'),
(10, 6, 1, 'Vote Cast'),
(11, 1, 1, 'Vote Revoked'),
(12, 2, 1, 'Vote Cast'),
(13, 3, 1, 'Vote Cast'),
(14, 4, 1, 'Vote Cast'),
(15, 5, 1, 'Vote Cast'),
(16, 6, 1, 'Vote Cast'),
(17, 1, 1, 'Vote Cast'),
(18, 2, 1, 'Vote Revoked'),
(19, 3, 1, 'Vote Cast'),
(20, 4, 1, 'Vote Cast'),
(21, 5, 1, 'Vote Revoked'),
(22, 6, 1, 'Vote Cast'),
(23, 1, 1, 'Vote Cast'),
(24, 2, 1, 'Vote Cast'),
(25, 3, 1, 'Vote Revoked'),
(26, 4, 1, 'Vote Cast'),
(27, 5, 1, 'Vote Cast'),
(28, 6, 1, 'Vote Revoked'),
(29, 1, 1, 'Vote Cast'),
(30, 2, 1, 'Vote Cast');