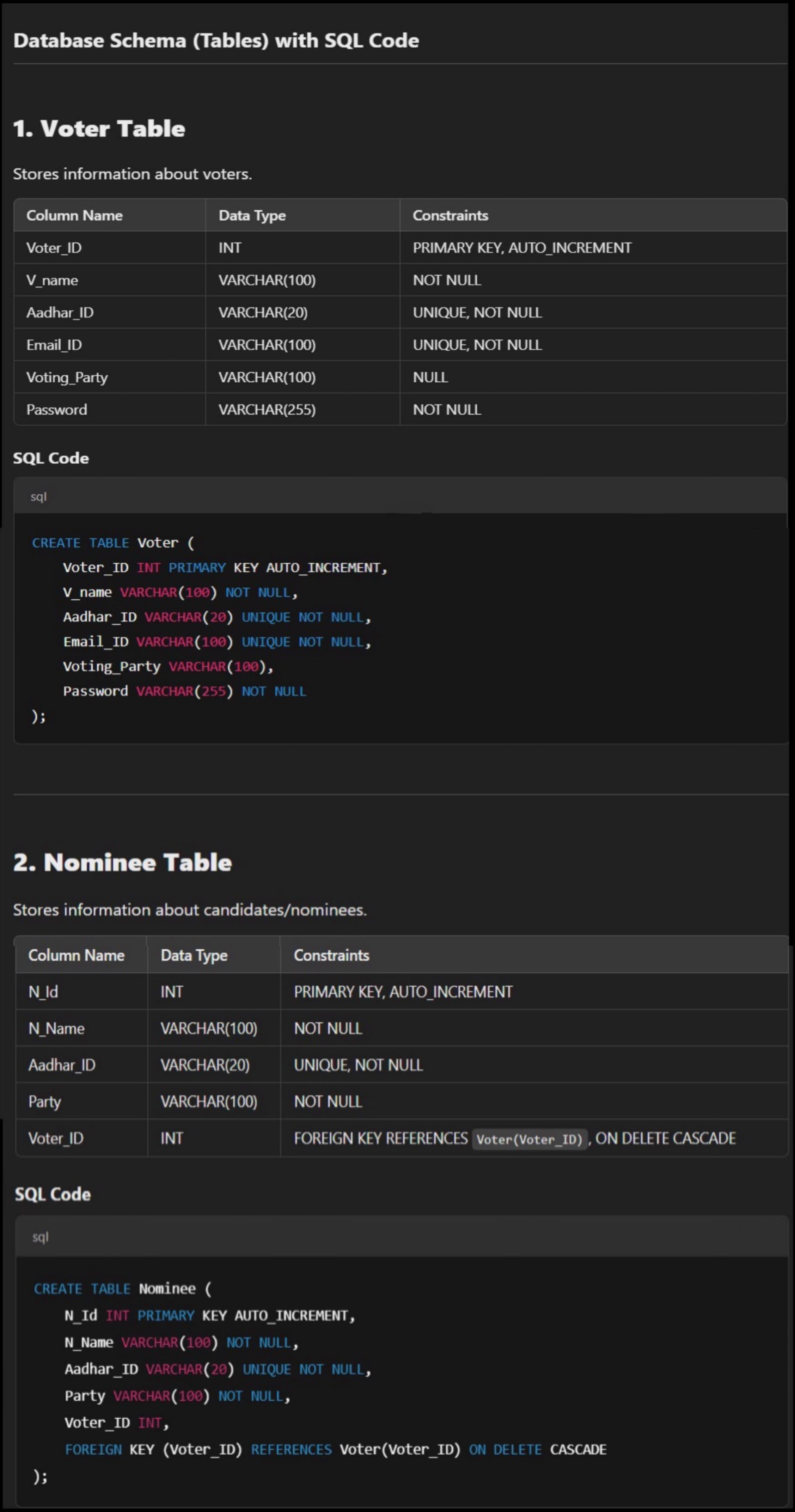


Database Schema (Tables) with SQL Code

1. Voter Table

Stores information about voters.

Column Name	Data Type	Constraints
Voter_ID	INT	PRIMARY KEY, AUTO_INCREMENT
V_name	VARCHAR(100)	NOT NULL
Aadhar_ID	VARCHAR(20)	UNIQUE, NOT NULL
Email_ID	VARCHAR(100)	UNIQUE, NOT NULL
Voting_Party	VARCHAR(100)	NULL
Password	VARCHAR(255)	NOT NULL

SQL Code

sql

```sql
CREATE TABLE Voter (
    Voter_ID INT PRIMARY KEY AUTO_INCREMENT,
    V_name VARCHAR(100) NOT NULL,
    Aadhar_ID VARCHAR(20) UNIQUE NOT NULL,
    Email_ID VARCHAR(100) UNIQUE NOT NULL,
    Voting_Party VARCHAR(100),
    Password VARCHAR(255) NOT NULL
);
```

2. Nominee Table

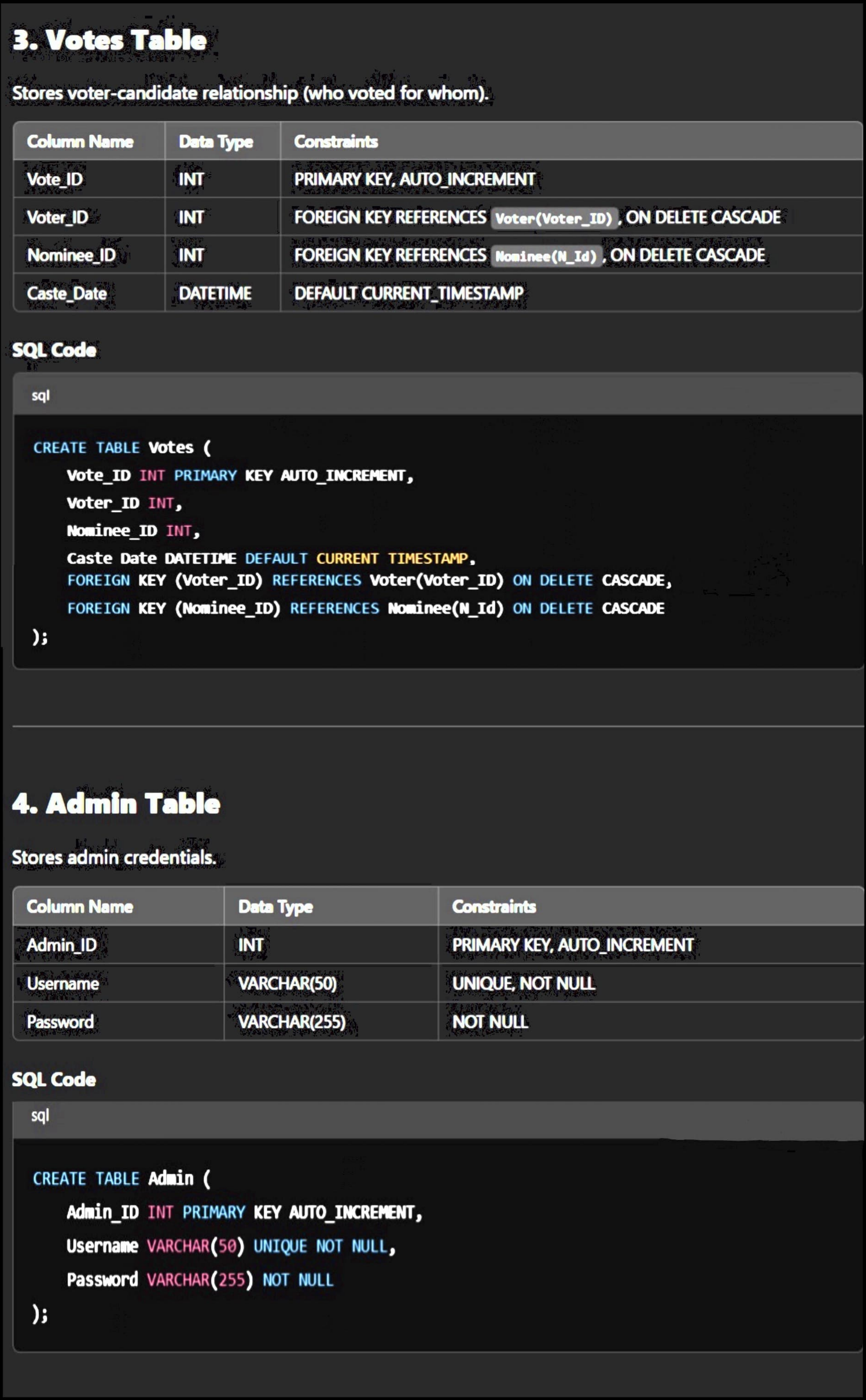
Stores information about candidates/nominees.

Column Name	Data Type	Constraints
N_Id	INT	PRIMARY KEY, AUTO_INCREMENT
N_Name	VARCHAR(100)	NOT NULL
Aadhar_ID	VARCHAR(20)	UNIQUE, NOT NULL
Party	VARCHAR(100)	NOT NULL
Voter_ID	INT	FOREIGN KEY REFERENCES `Voter(Voter_ID)` , ON DELETE CASCADE

SQL Code

sql

```sql
CREATE TABLE Nominee (
    N_Id INT PRIMARY KEY AUTO_INCREMENT,
    N_Name VARCHAR(100) NOT NULL,
    Aadhar_ID VARCHAR(20) UNIQUE NOT NULL,
    Party VARCHAR(100) NOT NULL,
    Voter_ID INT,
    FOREIGN KEY (Voter_ID) REFERENCES Voter(Voter_ID) ON DELETE CASCADE
);
```



3. Votes Table

Stores voter-candidate relationship (who voted for whom).

Column Name	Data Type	Constraints
Vote_ID	INT	PRIMARY KEY, AUTO_INCREMENT
Voter_ID	INT	FOREIGN KEY REFERENCES `Voter(Voter_ID)` , ON DELETE CASCADE
Nominee_ID	INT	FOREIGN KEY REFERENCES `Nominee(N_Id)` , ON DELETE CASCADE
Caste_Date	DATETIME	DEFAULT CURRENT_TIMESTAMP

SQL Code

sql

```sql
CREATE TABLE Votes (
    Vote_ID INT PRIMARY KEY AUTO_INCREMENT,
    Voter_ID INT,
    Nominee_ID INT,
    Caste Date DATETIME DEFAULT CURRENT TIMESTAMP,
    FOREIGN KEY (Voter_ID) REFERENCES Voter(Voter_ID) ON DELETE CASCADE,
    FOREIGN KEY (Nominee_ID) REFERENCES Nominee(N_Id) ON DELETE CASCADE
);
```

4. Admin Table

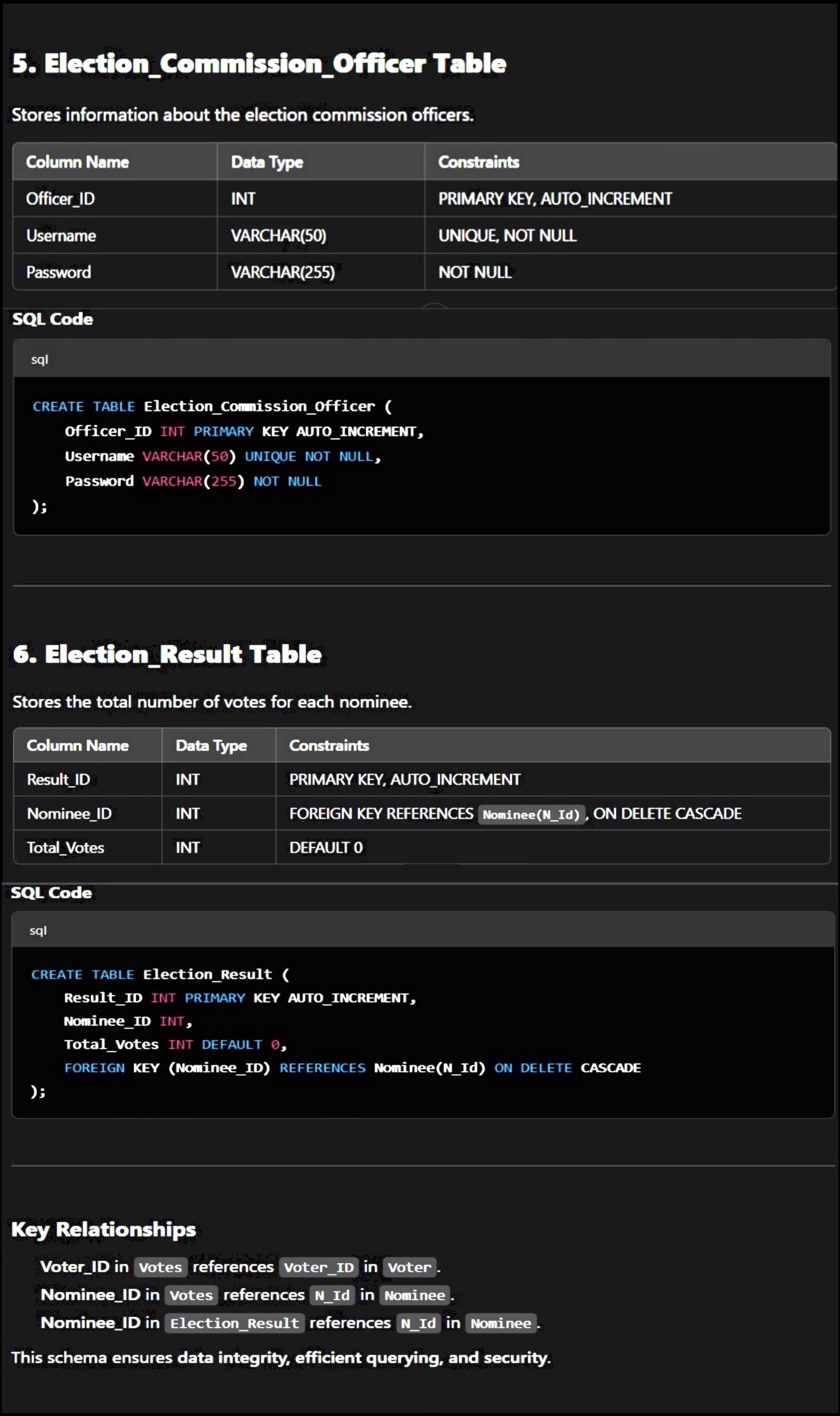
Stores admin credentials.

Column Name	Data Type	Constraints
Admin_ID	INT	PRIMARY KEY, AUTO_INCREMENT
Username	VARCHAR(50)	UNIQUE, NOT NULL
Password	VARCHAR(255)	NOT NULL

SQL Code

sql

```sql
CREATE TABLE Admin (
    Admin_ID INT PRIMARY KEY AUTO_INCREMENT,
    Username VARCHAR(50) UNIQUE NOT NULL,
    Password VARCHAR(255) NOT NULL
);
```



5. Election_Commission_Officer Table

Stores information about the election commission officers.

Column Name	Data Type	Constraints
Officer_ID	INT	PRIMARY KEY, AUTO_INCREMENT
Username	VARCHAR(50)	UNIQUE, NOT NULL
Password	VARCHAR(255)	NOT NULL

SQL Code

sql

```sql
CREATE TABLE Election_Commission_Officer (
    Officer_ID INT PRIMARY KEY AUTO_INCREMENT,
    Username VARCHAR(50) UNIQUE NOT NULL,
    Password VARCHAR(255) NOT NULL
);
```

6. Election_Result Table

Stores the total number of votes for each nominee.

Column Name	Data Type	Constraints
Result_ID	INT	PRIMARY KEY, AUTO_INCREMENT
Nominee_ID	INT	FOREIGN KEY REFERENCES `Nominee(N_Id)`, ON DELETE CASCADE
Total_Votes	INT	DEFAULT 0

SQL Code

sql

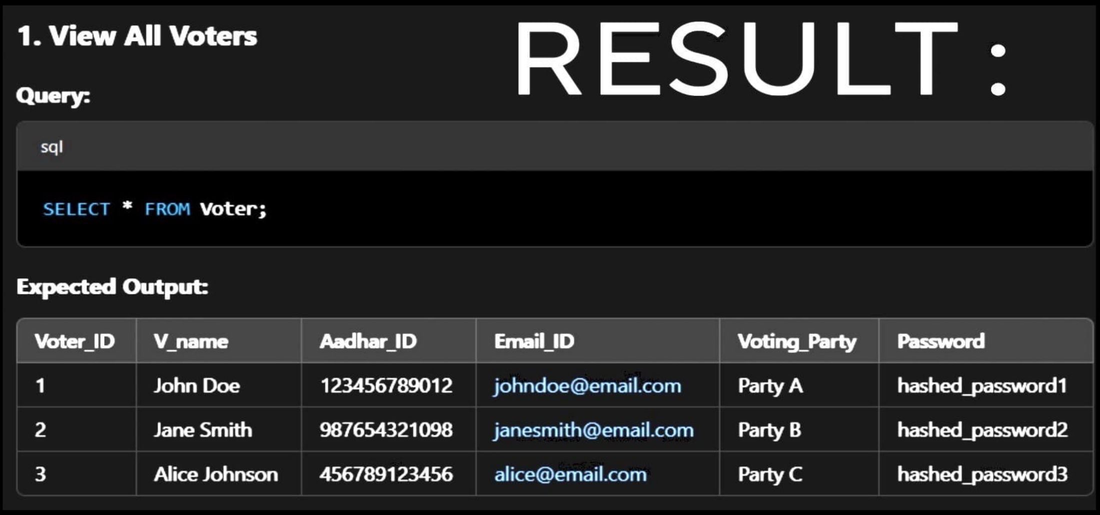
```sql
CREATE TABLE Election_Result (
    Result_ID INT PRIMARY KEY AUTO_INCREMENT,
    Nominee_ID INT,
    Total_Votes INT DEFAULT 0,
    FOREIGN KEY (Nominee_ID) REFERENCES Nominee(N_Id) ON DELETE CASCADE
);
```

Key Relationships

- Voter_ID in `Votes` references `Voter_ID` in `Voter`.
- Nominee_ID in `Votes` references `N_Id` in `Nominee`.
- Nominee_ID in `Election_Result` references `N_Id` in `Nominee`.

This schema ensures data integrity, efficient querying, and security.

RESULT :



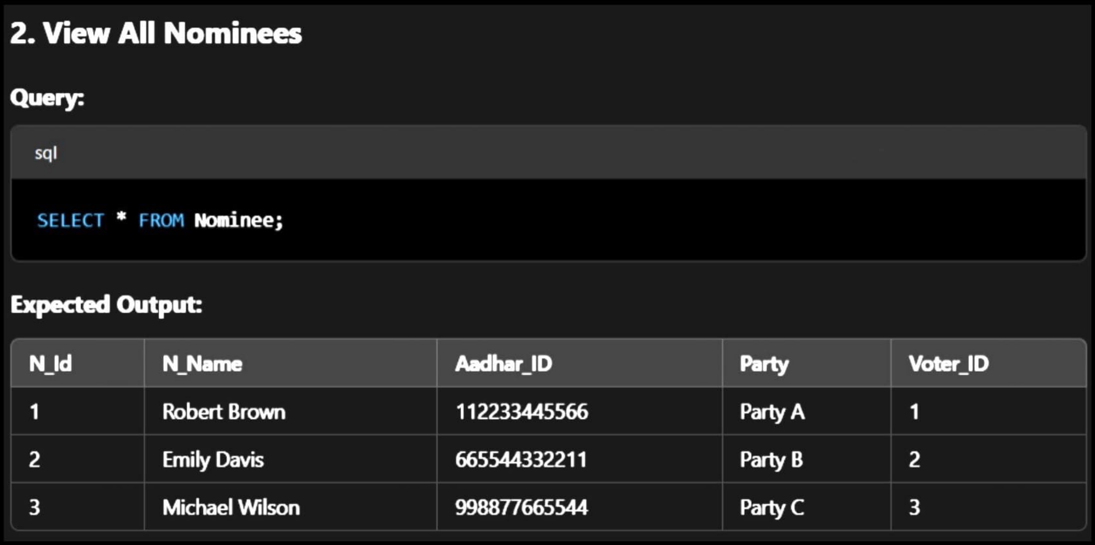
1. View All Voters

Query:

```sql
SELECT * FROM Voter;
```

Expected Output:

Voter_ID	V_name	Aadhar_ID	Email_ID	Voting_Party	Password
1	John Doe	123456789012	johndoe@email.com	Party A	hashed_password1
2	Jane Smith	987654321098	janesmith@email.com	Party B	hashed_password2
3	Alice Johnson	456789123456	alice@email.com	Party C	hashed_password3



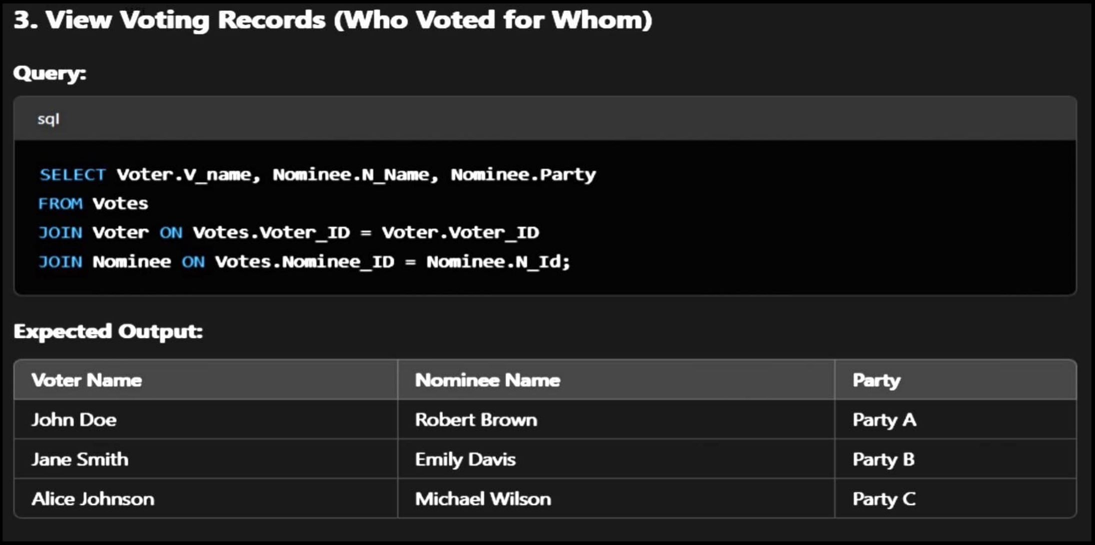
2. View All Nominees

Query:

```sql
SELECT * FROM Nominee;
```

Expected Output:

N_Id	N_Name	Aadhar_ID	Party	Voter_ID
1	Robert Brown	112233445566	Party A	1
2	Emily Davis	665544332211	Party B	2
3	Michael Wilson	998877665544	Party C	3



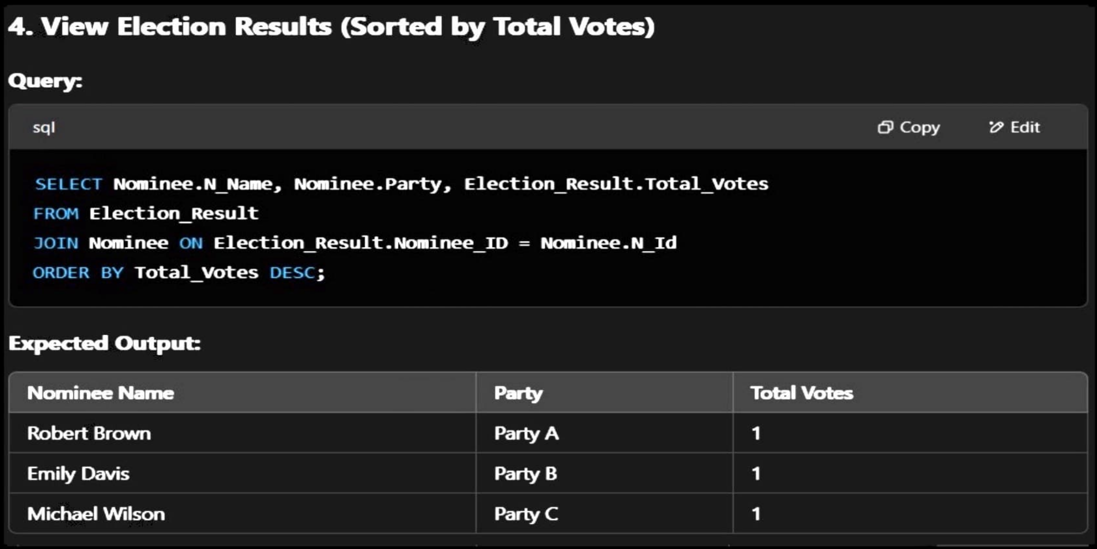
3. View Voting Records (Who Voted for Whom)

Query:

```sql
SELECT Voter.V_name, Nominee.N_Name, Nominee.Party
FROM Votes
JOIN Voter ON Votes.Voter_ID = Voter.Voter_ID
JOIN Nominee ON Votes.Nominee_ID = Nominee.N_Id;
```

Expected Output:

Voter Name	Nominee Name	Party
John Doe	Robert Brown	Party A
Jane Smith	Emily Davis	Party B
Alice Johnson	Michael Wilson	Party C



4. View Election Results (Sorted by Total Votes)

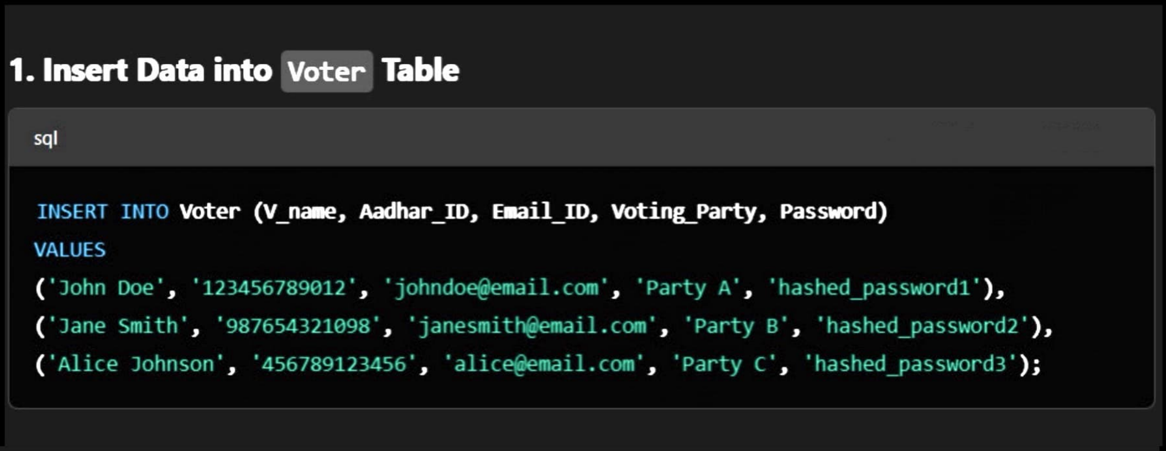
Query:

```sql
SELECT Nominee.N_Name, Nominee.Party, Election_Result.Total_Votes
FROM Election_Result
JOIN Nominee ON Election_Result.Nominee_ID = Nominee.N_Id
ORDER BY Total_Votes DESC;
```

Expected Output:

Nominee Name	Party	Total Votes
Robert Brown	Party A	1
Emily Davis	Party B	1
Michael Wilson	Party C	1

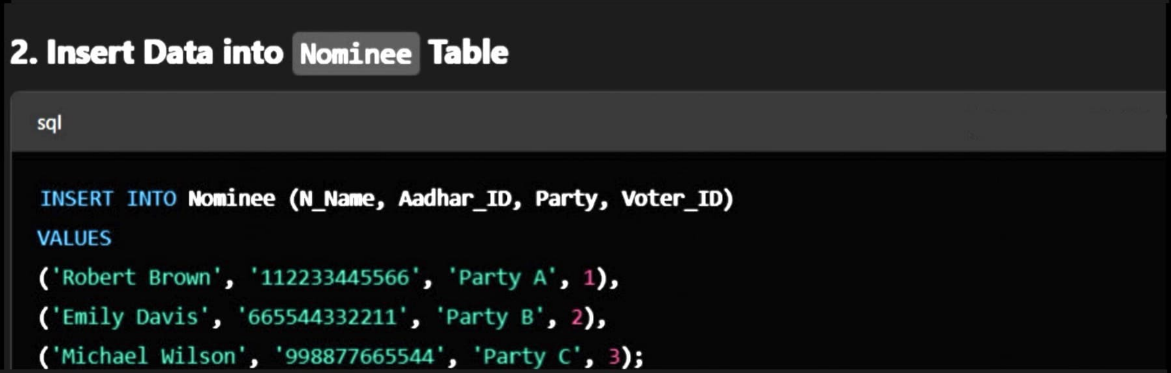
SQL queries to insert sample data into tables:



1. Insert Data into `Voter` Table

sql

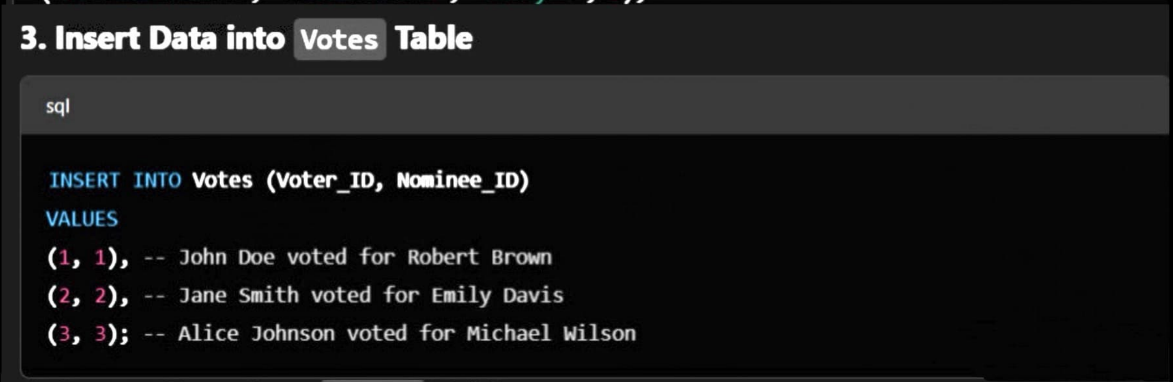
```sql
INSERT INTO Voter (V_name, Aadhar_ID, Email_ID, Voting_Party, Password)
VALUES
('John Doe', '123456789012', 'johndoe@email.com', 'Party A', 'hashed_password1'),
('Jane Smith', '987654321098', 'janesmith@email.com', 'Party B', 'hashed_password2'),
('Alice Johnson', '456789123456', 'alice@email.com', 'Party C', 'hashed_password3');
```



2. Insert Data into `Nominee` Table

sql

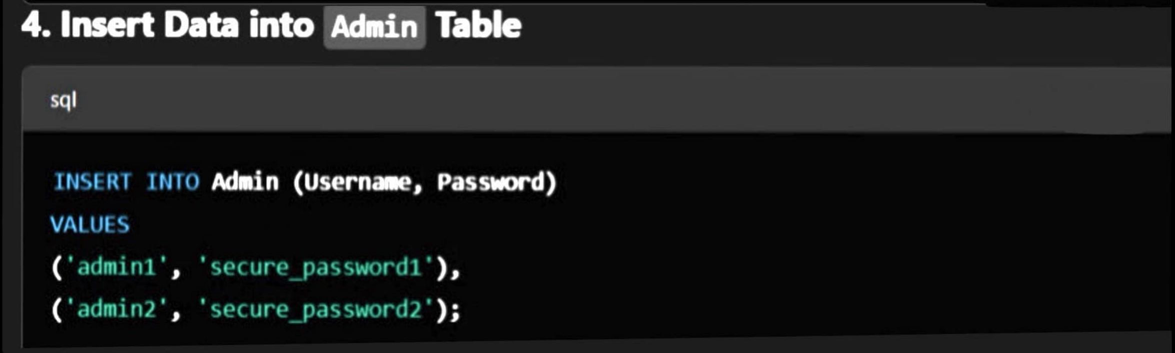
```sql
INSERT INTO Nominee (N_Name, Aadhar_ID, Party, Voter_ID)
VALUES
('Robert Brown', '112233445566', 'Party A', 1),
('Emily Davis', '665544332211', 'Party B', 2),
('Michael Wilson', '998877665544', 'Party C', 3);
```



3. Insert Data into `Votes` Table

sql

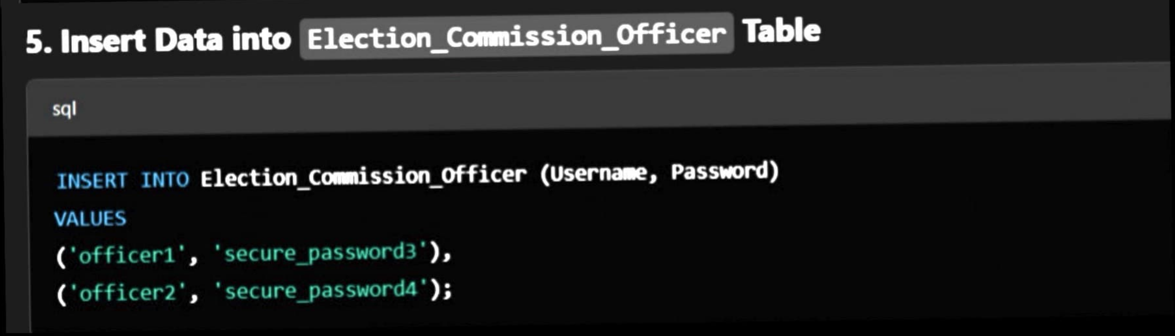
```sql
INSERT INTO Votes (Voter_ID, Nominee_ID)
VALUES
(1, 1), -- John Doe voted for Robert Brown
(2, 2), -- Jane Smith voted for Emily Davis
(3, 3); -- Alice Johnson voted for Michael Wilson
```



4. Insert Data into `Admin` Table

sql

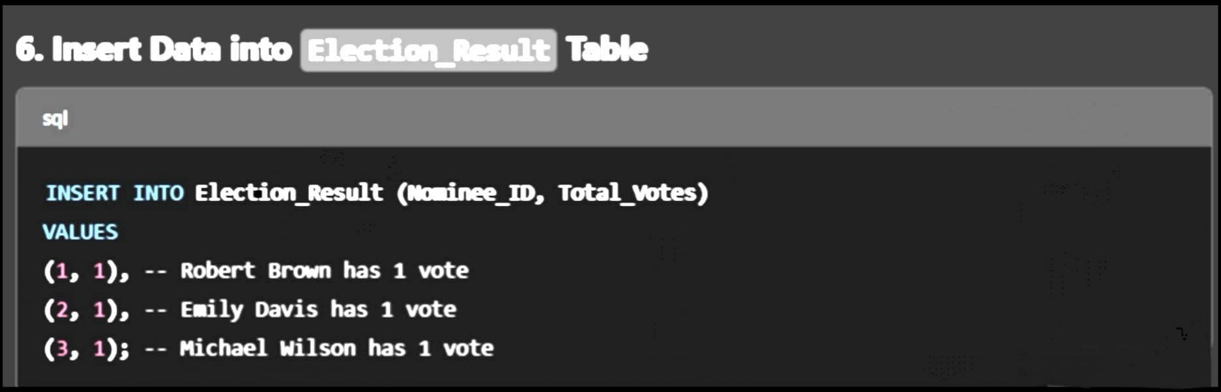
```sql
INSERT INTO Admin (Username, Password)
VALUES
('admin1', 'secure_password1'),
('admin2', 'secure_password2');
```



5. Insert Data into `Election_Commission_Officer` Table

sql

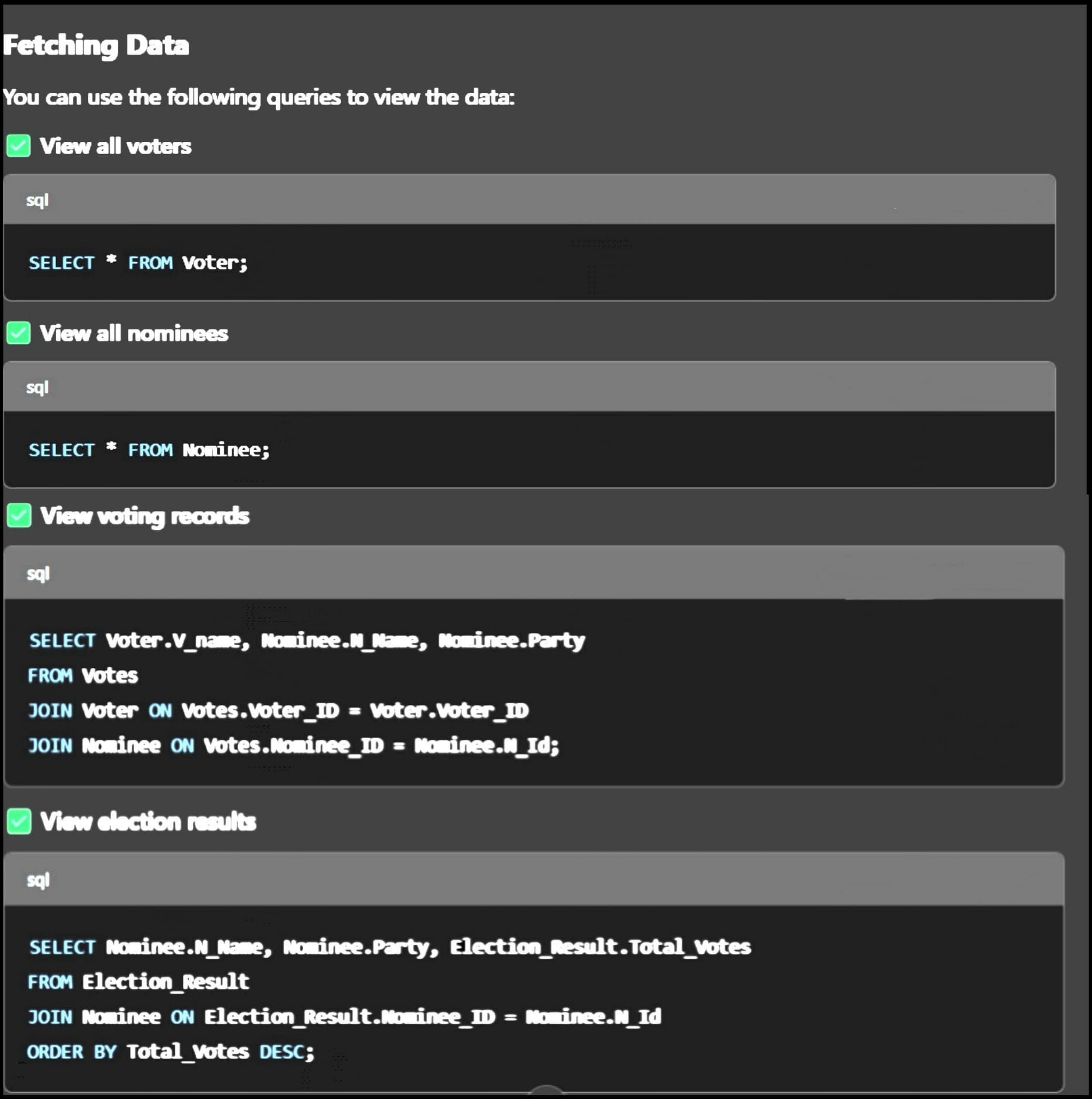
```sql
INSERT INTO Election_Commission_Officer (Username, Password)
VALUES
('officer1', 'secure_password3'),
('officer2', 'secure_password4');
```



6. Insert Data into `Election_Result` Table

sql

```sql
INSERT INTO Election_Result (Nominee_ID, Total_Votes)
VALUES
(1, 1), -- Robert Brown has 1 vote
(2, 1), -- Emily Davis has 1 vote
(3, 1); -- Michael Wilson has 1 vote
```



Fetching Data

You can use the following queries to view the data:

✅ **View all voters**

sql

```sql
SELECT * FROM Voter;
```

✅ **View all nominees**

sql

```sql
SELECT * FROM Nominee;
```

✅ **View voting records**

sql

```sql
SELECT Voter.V_name, Nominee.N_Name, Nominee.Party
FROM Votes
JOIN Voter ON Votes.Voter_ID = Voter.Voter_ID
JOIN Nominee ON Votes.Nominee_ID = Nominee.N_Id;
```

✅ **View election results**

sql

```sql
SELECT Nominee.N_Name, Nominee.Party, Election_Result.Total_Votes
FROM Election_Result
JOIN Nominee ON Election_Result.Nominee_ID = Nominee.N_Id
ORDER BY Total_Votes DESC;
```