**Key Milestone 3: DBMS Lab Project SQL Implementation**



**Spring 2025**

**CSE-403L**

**Database Management System Lab**

Submitted by:

**Muhammad Musa (22PWCSE2157)**

**Muhammad Jasim (22PWCSE2021)**

Class Section: **‘C’**

“We affirm that we have completed this work with integrity”

Submitted to:

**Engr. Sumayyea Salahuddin**

May 29, 2025

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

**Project Database Details**

This document outlines the database implementation for the Online Voting System, including table schemas, metadata, sample data, and example queries.

**1. Database Schema and Metadata**

Based on the normalized relational schema (3NF), here are the tables with their attributes, primary keys (PK), foreign keys (FK), and data types.

**Table: POSITIONS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Key** | **Description** |
| id | INT | PK | Unique identifier for position |
| description | VARCHAR(255) |  | Description of the position |
| max\_vote | INT |  | Maximum votes allowed for this position |
| priority | INT |  | Priority of the position |

**Table: CANDIDATES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Key** | **Description** |
| id | INT | PK | Unique identifier for candidate |
| position\_id | INT | FK | References POSITIONS(id) |
| firstname | VARCHAR(100) |  | Candidate's first name |
| lastname | VARCHAR(100) |  | Candidate's last name |
| photo | VARCHAR(255) |  | URL or path to candidate's photo |
| platform | TEXT |  | Candidate's platform/manifesto |

**Table: ADMIN**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Key** | **Description** |
| id | INT | PK | Unique identifier for admin |
| username | VARCHAR(50) | UNIQUE | Admin's username (unique) |
| password | VARCHAR(255) |  | Hashed password for admin |

**Table: VOTERS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Key** | **Description** |
| id | INT | PK | Unique identifier for voter (surrogate key) |
| voters\_id | VARCHAR(50) | FK, UNIQUE | Unique voter ID (e.g., national ID), references VOTER\_DETAILS(voters\_id) |

**Table: VOTER\_DETAILS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Key** | **Description** |
| voters\_id | VARCHAR(50) | PK | Unique voter ID (e.g., national ID) |
| password | VARCHAR(255) |  | Hashed password for voter |
| firstname | VARCHAR(100) |  | Voter's first name |
| lastname | VARCHAR(100) |  | Voter's last name |
| photo | VARCHAR(255) |  | URL or path to voter's photo |

**Table: VOTES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Key** | **Description** |
| id | INT | PK | Unique identifier for vote record |
| voter\_id | INT | FK | References VOTERS(id) |
| candidate\_id | INT | FK | References CANDIDATES(id) |
| position\_id | INT | FK | References POSITIONS(id) |
| (voter\_id, position\_id) | UNIQUE | Composite Unique Key | Ensures a voter can only vote once per position |

**2. Sample Data**

Here is some sample data for each table to illustrate the schema.

**POSITIONS Data**

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **description** | **max\_vote** | **priority** |
| 1 | President | 1 | 1 |
| 2 | Vice President | 1 | 2 |
| 3 | Secretary | 1 | 3 |

**CANDIDATES Data**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **id** | **position\_id** | **firstname** | **lastname** | **photo** | **platform** |
| 101 | 1 | Alice | Smith | 'alice.jpg' | "Focus on education reform." |
| 102 | 1 | Bob | Johnson | 'bob.jpg' | "Economic growth and job creation." |
| 103 | 2 | Carol | Davis | 'carol.jpg' | "Community development initiatives." |
| 104 | 3 | David | Brown | 'david.jpg' | "Transparency and accountability." |

**ADMIN Data**

|  |  |  |
| --- | --- | --- |
| **id** | **username** | **password** |
| 1 | admin1 | '2a10$abcdefghijklmnopqrstuvw.xyz123456' |
| 2 | superuser | '2a10$abcdefghijklmnopqrstuvw.xyz789012' |

**VOTERS Data**

|  |  |
| --- | --- |
| **id** | **voters\_id** |
| 201 | 'VTR001' |
| 202 | 'VTR002' |
| 203 | 'VTR003' |

**VOTER\_DETAILS Data**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **voters\_id** | **password** | **firstname** | **lastname** | **photo** |
| 'VTR001' | '2a10$abcdefghijklmnopqrstuvw.xyz345678' | John | Doe | 'john.jpg' |
| 'VTR002' | '2a10$abcdefghijklmnopqrstuvw.xyz901234' | Jane | Smith | 'jane.jpg' |
| 'VTR003' | '2a10$abcdefghijklmnopqrstuvw.xyz567890' | Peter | Jones | 'peter.jpg' |

**VOTES Data**

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **voter\_id** | **candidate\_id** | **position\_id** |
| 301 | 201 | 101 | 1 |
| 302 | 202 | 103 | 2 |
| 303 | 203 | 101 | 1 |

**3. Example Queries**

Here are some example SQL queries to interact with the database.

1. **Retrieve all candidates for the 'President' position:**
2. SELECT C.firstname, C.lastname, C.platform
3. FROM CANDIDATES C
4. JOIN POSITIONS P ON C.position\_id = P.id
5. WHERE P.description = 'President';
6. **Count the total votes received by each candidate:**
7. SELECT C.firstname, C.lastname, COUNT(V.id) AS total\_votes
8. FROM CANDIDATES C
9. LEFT JOIN VOTES V ON C.id = V.candidate\_id
10. GROUP BY C.id, C.firstname, C.lastname
11. ORDER BY total\_votes DESC;
12. **Find which voter cast a vote for a specific candidate (e.g., Candidate ID 101):**
13. SELECT VD.firstname, VD.lastname, V.id AS vote\_record\_id
14. FROM VOTER\_DETAILS VD
15. JOIN VOTERS VR ON VD.voters\_id = VR.voters\_id
16. JOIN VOTES V ON VR.id = V.voter\_id
17. WHERE V.candidate\_id = 101;
18. **List all positions and the number of candidates running for each:**
19. SELECT P.description, COUNT(C.id) AS number\_of\_candidates
20. FROM POSITIONS P
21. LEFT JOIN CANDIDATES C ON P.id = C.position\_id
22. GROUP BY P.id, P.description
23. ORDER BY P.priority;
24. **Check if a specific voter (e.g., 'VTR001') has voted for the 'President' position:**
25. SELECT
26. CASE
27. WHEN EXISTS (
28. SELECT 1
29. FROM VOTERS VR
30. JOIN VOTER\_DETAILS VD ON VR.voters\_id = VD.voters\_id
31. JOIN VOTES V ON VR.id = V.voter\_id
32. JOIN POSITIONS P ON V.position\_id = P.id
33. WHERE VD.voters\_id = 'VTR001' AND P.description = 'President'
34. ) THEN 'Voted'
35. ELSE 'Not Voted'
36. END AS voting\_status;