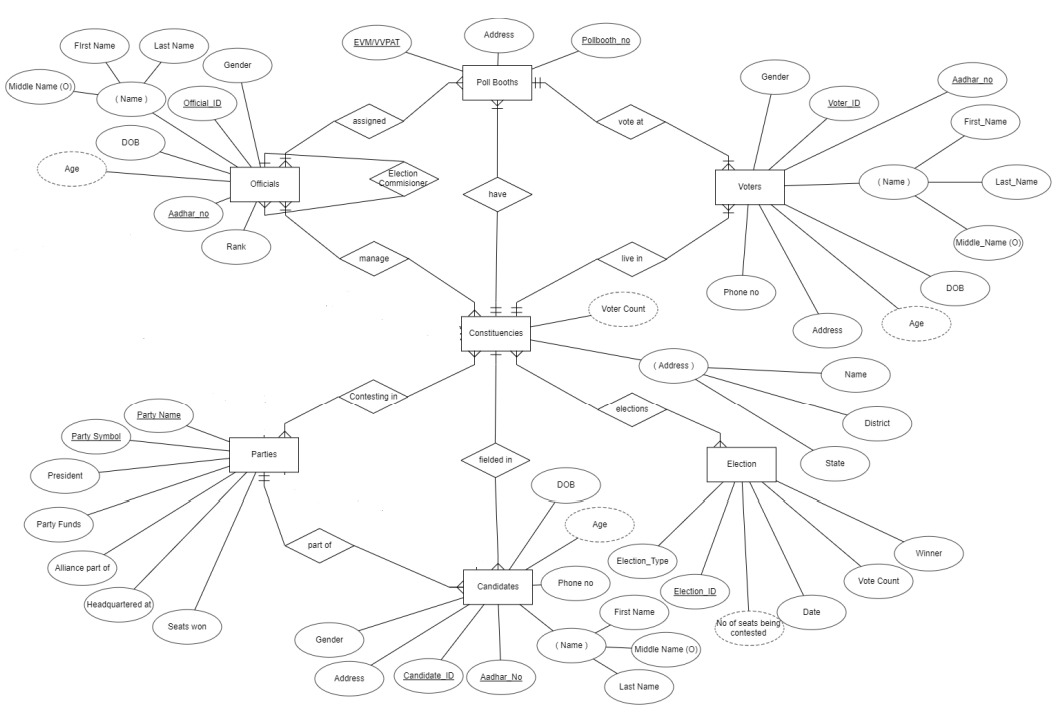
**DBMS Project Report**

**Election Management System**

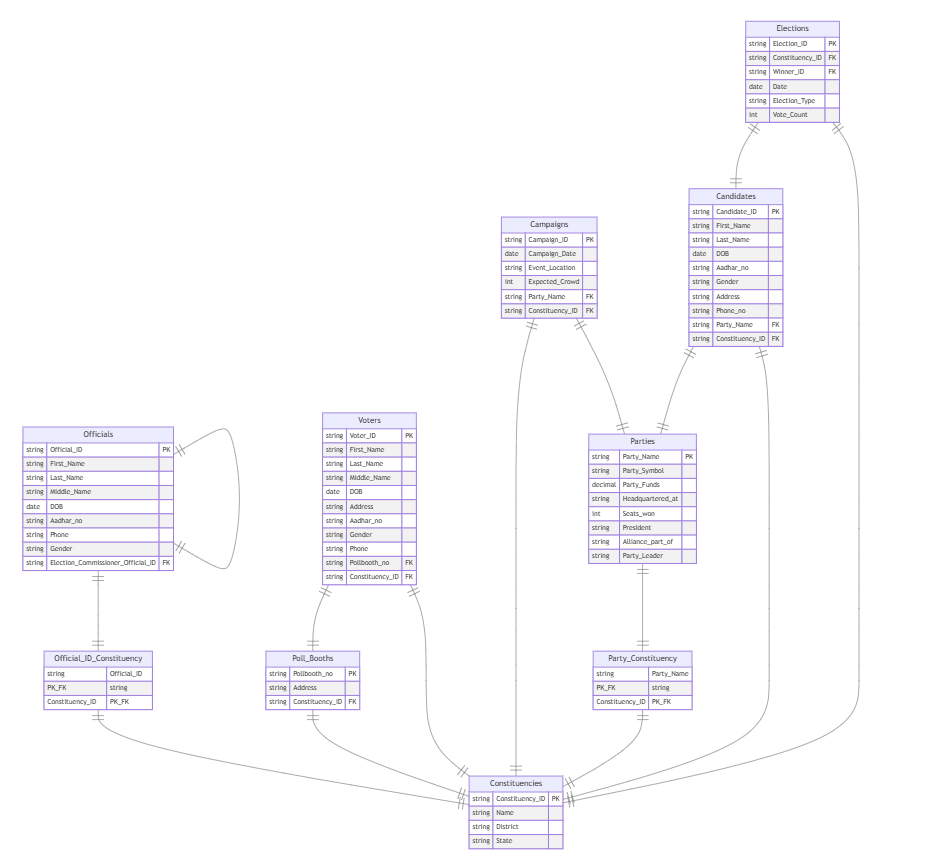
**Team:  
Manish P: PES1UG22CS332  
Priyam R: PES1UG22CS453**

**The Election Management System (EMS) is designed to centralize and streamline the management of elections, covering all election-related tasks such as voter registration, candidate management, election setup, polling booth allocation, voting, and result generation. This SRS describes the full system's requirements, functionality, and constraints, ensuring transparency, security, and efficiency in the electoral process.**

**ER Diagram:**  


**Relational Schema:**





**List of Triggers Used**

1. **voter\_age**  
   Trigger to calculate the age of a voter before inserting into the voter table.
2. **voter\_age\_on\_update**  
   Trigger to update the voter's age if their date of birth changes during an update.
3. **prevent\_delete\_constituency**  
   Trigger to prevent deletion of a constituency if voters are still associated with it.
4. **prevent\_delete\_pollbooth**  
   Trigger to prevent deletion of a poll booth if voters are still associated with it.
5. **update\_seats**  
   Trigger to update the number of seats in an election when a new entry is added to the election\_cons table.
6. **set\_official\_id**  
   Trigger to set the official ID before inserting into the official table.
7. **off\_age**  
   Trigger to calculate the age of an official before inserting into the official table.
8. **official\_age\_on\_update**  
   Trigger to update an official's age if their date of birth changes during an update.
9. **set\_candidate\_id**  
   Trigger to set the candidate ID before inserting into the candidate table.
10. **cand\_age**  
    Trigger to calculate the age of a candidate before inserting into the candidate table.
11. **increment\_party\_member\_count**

Trigger to increase the party member count when a candidate registers

12.**add\_gender**

Trigger to auto increment the gender count

**Functions Used:**

* 1. **malecount:**

Used to calculate number of male voters in a constituency

* 1. **femalecount:**

Used to calculate number of female voters in a constituency

**List of Procedures Used:**

1. **getconsdets:**

This procedure retrieves detailed information about each constituency, including the number of male and female voters and the count of polling booths.

1. **getvoterdets:**

This procedure retrieves the details of all voters.

1. **getcanddets:**

This procedure retrieves details about candidates running for election.

1. **getpartydets:**

This procedure retrieves details about political parties.

1. **getofficialdets:**

This procedure retrieves details about officials working in the election process.

**List of Nested Queries Used:**

1. Query to fetch constituency name

**List of joint operations:**

1. create procedure getconsinfo(in input\_voter\_id varchar(20)) begin select c.constituency\_name, c.state, c.voter\_count, concat(coalesce(cand.first\_name,''),' ',coalesce(cand.middle\_name,''),' ',coalesce(cand.last\_name,'')) as candidate\_name, cand.age,p.party\_name,p.party\_symbol from constituency c left join candidate cand on cand.cons\_fight = c.constituency\_name left join party p on cand.party\_rep = p.party\_name where cand.cons\_fight in (select constituency\_name from voter where voter\_id = input\_voter\_id);

This is a left joint used inside a procedure. The left joint clause is used to combine rows from the constituency table (c) and the candidate table (cand):

1. create procedure getconsdets() begin select c.constituency\_name, malecount(c.constituency\_name) as male\_count, femalecount(c.constituency\_name) AS female\_count, count(distinct pb.poll\_booth\_id) as poll\_booth\_count from constituency c left join poll\_booth pb on c.constituency\_id = pb.constituency\_id group by c.constituency\_name;