Database Schema

Relations and Constraints
Team Name : MangoDB
Team Number : 12
NOTE: we will submit ddl from workbench and not what is in this doc

---

1. **person**(*ID, FirstName, LastName, DOB, PhoneNumber, Gender, education, Income, Token, religion*)
2. **voter**(*ID, document, Registration_Date, constituency_id*)
3. **Voted_For**(*Voter_ID, Election_ID, Candidate_ID*) -- adding election_id
4. **Verification_Official**(id, Constituency_id, admin_id) -- D
5. **Administration_Official**(Administration_Official_ID) -- D
6. **EC_Official**(EC_Official_ID, Constituency_ID) -- D
7. **Unverified_Voter**(Unverified_id, Verification_Type, Verification_Document, Verification_Official_ID) -- D
8. **Candidate**(Candidate_ID, Criminal_Case_Certificate, Verification_Official_ID) -- adding verification_official_id
9. **BioData**(*Biodata_ID, Year, Wealth) -- D remove extra columns*
10. **Member_Of**(*Candidate_id, Election_id, Party_ID, Money_Spent) -- add foreign key*
11. **Polititcal_Party**(*Party_ID, Name, Symbol) -- D*
12. **Constituency**(*Constituency_ID, Name, Region) -- D*
13. **Election**(*id, start_time, end_time)*
14. **Candidate_Criminal_Cases**(Criminal_ID, Biodata_ID, Year, Criminal_case) --D add foregin key constraint
15. **Stands_From**(*id, election_id, constituency_id, Votes_Won, Manifesto)*

person		
ID	*INT*	*PRIMARY KEY*
FirstName	*VARCHAR(255)*	*NOT NULL*
LastName	*VARCHAR(255)*	*NOT NULL*
DOB	*DATE*	*NOT NULL*
Phone_Number	*VARCHAR(10)*	*NOT NULL*
Gender	*VARCHAR(6)*	*NOT NULL. CHECK Gender IN ('Male', 'Female' , 'Other')*
education	*VARCHAR(50)*	*NOT NULL.*
Income	*INT*	*NOT NULL. CHECK Income ≥ 0*
Token	*VARCHAR(30)*	*NOT NULL. unique.*

religion	*VARCHAR(30)*	*NOT NULL*

Functional Dependencies :
Person_ID -> Person_ID, First_Name, Last_Name, Date_Of_Birth, Phone_Number, Gender, Education_Status, Income, Django_Token, Religion
Candidate Key :
{ Person_ID }

voter		
id	*INT*	*PRIMARY KEY*
document	*BLOB*	*NOT NULL*
Registration_Date	*DATE*	*NOT NULL*
constituency_id	*INT*	*NOT NULL*

FOREIGN KEY(id) REFERENCES person(ID)
FOREIGN KEY(constituency_id) REFERENCES Constituency(Constituency_ID)

Remark : **Voter** IS A **Person**
Functional Dependencies :
Voter_ID ->Voter_ID, Voter_Card, Registration_Date, Constituency_ID
Candidate Key :
{ Voter_ID }

Voter_For		
Voter_ID	*INT*	*PRIMARY KEY*
Election_ID	*INT*	*PRIMARY KEY*
Candidate_ID	*INT*	*NOT NULL*

FOREIGN KEY(Voter_ID) REFERENCES Voter(Voter_ID)
FOREIGN KEY(Election_ID) REFERENCES Election(Election_ID)
FOREIGN KEY(Candidate_ID) REFERENCES Candidate(Candidate_ID)

Functional Dependencies :
Voter_ID, Election_ID ->Voter_ID,Election_ID, Candidate_ID
Candidate Key :
{ (Voter_ID, Election_ID) }

Verification_Official		
Verification_Official_ID	*INT*	*PRIMARY KEY*
Constituency_ID	*INT*	*NOT NULL*
admin_id	*INT*	*NOT NULL*

FOREIGN KEY(Verification_Official_ID) REFERENCES Employee_from_EC(Employee_ID)
FOREIGN KEY(Constituency_ID) REFERENCES Constituency(Constituency_ID)
FOREIGN KEY(Administration_Official_ID) REFERENCES Administration_Official(Administration_Official_ID)

Remark : **Verification_Official** IS A **Employee_from_EC**
Functional Dependencies :
Verification_Official_ID -> Verification_Official_ID, Role, Constituency_ID
Candidate Key :
{ Verification_Official_ID }

Administration_Official		
Administration_Official_ID	*INT*	*PRIMARY FOREIGN KEY*

FOREIGN KEY(Administration_Official_id) REFERENCES Employee_from_EC(Employee_ID)

Remark : **Administration_Official** IS A **Employee_from_EC**
Functional Dependencies :
Administration_Official_ID -> Administration_Official_ID , Role
Candidate Key :
{ Administration_Official_ID }

EC_Official		
EC_Official_ID	*INT*	*PRIMARY FOREIGN KEY*
Constituency_ID	*INT*	*NOT NULL FOREIGN KEY*

FOREIGN KEY(EC_Official_ID) REFERENCES Employee_from_EC(Employee_ID)
FOREIGN KEY(Constituency_ID) REFERENCES Constituency(Constituency_ID)

Remark : **EC_Official** IS A **Employee_from_EC**
Functional Dependencies :
EC_Official_ID -> EC_Official_ID, Role, Constituency_ID
Candidate Key :
{ EC_Official_ID }

Unverified_Voter		
Unverified_ID	*INT*	*PRIMARY KEY*
Verification_Type	*VARCHAR(30)*	*NOT NULL. CHECK Verification_Type IN ("Aadhar","Passport"."Driving License")*
Verification_Document	*BLOB*	*NOT NULL.*
Verification_Official_ID	*INT*	*NOT NULL.*

FOREIGN KEY(Unverified_ID) REFERENCES Person(Person_ID)
FOREIGN KEY(Verification_Official_ID) REFERENCES Verification_Official(Verification_Official_ID)

Remark : **Unverified_Voter** IS A **Person**
Functional Dependencies :
Unverified_ID -> Unverified_ID, Verification_Type, Verification_Document, Verification_Official_ID .
Candidate Key :
{ *Unverified_ID* }

Candidate		
Candidate_ID	*INT*	*PRIMARY KEY*
Criminal_Case_Certificate	*BLOB*	*NOT NULL*
Vertification_Official_ID	*INT*	*NOT NULL*
Verification_Status	*VARCHAR(30)*	*NOT NULL. CHECK Verification_Status IN ("Accepted", "Rejected")*
Date_Of_Verification	*DATE*	*NOT NULL*

FOREIGN KEY(Candidate_ID) REFERENCES Voter(Voter_ID)
FOREIGN KEY(Verification_Official_ID) REFERENCES Verification_Official(Verification_Official_ID)

Remark : **Candidate** IS A **Voter**
Functional Dependencies :
Candidate_ID -> Candidate_ID, Criminal_Case_Certificate, Vertification_Official_ID, Verification_Status
Candidate Key :
{ *Candidate_ID* }

Biodata		
Biodata_ID	*INT*	*PRIMARY KEY*
Year	*INT*	*PRIMARY KEY. CHECK Year BETWEEN 1900 AND 2021.*
Wealth	*INT*	*NOT NULL. CHECK Wealth ≥ 0*

FOREIGN KEY(Biodata_ID) REFERENCES Candidate(Candidate_ID)

Functional Dependencies :
Biodata_ID, Year -> *Biodata_ID*, *Year*, Wealth
Candidate Key :
{ (*Biodata_ID, Year)* }

Member_Of		
Candidate_id	*INT*	*PRIMARY KEY*
Election_id	*INT*	*PRIMARY KEY*
Party_ID	*INT*	*NOT NULL.*
Money_Spent	*INT*	*NOT NULL. CHECK Money_Spent >= 0. DEFAULT = 0*

FOREIGN KEY(Candidate_ID) REFERENCES Candidate(Candidate_ID)
FOREIGN KEY(Election_ID) REFERENCES Election(Election_ID)
FOREIGN KEY(Party_ID) REFERENCES Political_Party(Party_ID)

Functional Dependencies :
Candidate_ID*, Election_ID* -> Candidate_ID, *Election_ID, Party_ID, Money_Spent*
Candidate Key :
{ (Candidate_ID*, Election_ID)* }

Political_Party		
Party_ID	*INT*	*PRIMARY KEY*
Name	*VARCHAR(100)*	*NOT NULL*
Symbol	*BLOB*	*NOT NULL.*

Functional Dependencies :
Party_ID -> Party_ID, *Name, Symbol*
Candidate Key :
{ *Party_ID* }

total_money_spent_on_election is a derived attribute. how to model it. Do we need to model it ? This is like the number of clicks and number of likes in the midsem. It can directly be included in the view.

Constituency		
Constituency_ID	*INT*	*PRIMARY KEY*
Name	*VARCHAR(100)*	*NOT NULL*
Region	*VARCHAR(100)*	*NOT NULL*

Functional Dependencies :
Constituency_ID -> *Constituency_ID*, *Name, Region*
Candidate Key :
{ *Constituency_ID* }

Election		
Election_ID	*INT*	*PRIMARY KEY*
Start_Date	*DATE*	*NOT NULL. CHECK Start_Date ≥ 1901-01-01*
End_Date	*DATE*	*NOT NULL. CHECK Start_Date ≥ 1901-01-01*

CHECK End_Date ≥ Start_Date

Functional Dependencies :
Election_ID -> Election_ID, *Start_Date, End_Date*
Candidate Key :
{ *Election_ID* }

Candidate_Criminal_Cases		
Criminal_ID	*INT*	*PRIMARY KEY*
Biodata_ID	*INT*	*NOT NULL*
Year	*INT*	*NOT NULL*
Criminal_Case	*VARCHAR(50)*	*NOT NULL*

FOREIGN KEY(Biodata_ID) REFERENCES Biodata(Biodata_ID)

FOREIGN KEY(Year) REFERENCES Biodata(Year)

Remarks: Criminal_ID is a SURROGATE Key
Functional Dependencies :
Criminal_ID -> Biodata_ID, year, criminal_case
Candidate Key :
{ Criminal_ID }

Stands_From		
id	*INT*	*PRIMARY KEY*
election_id	*INT*	*PRIMARY KEY*
constituency_id	*INT*	*default NULL*
Votes_Won	*INT*	*NOT NULL. CHECK Votes_Won >= 0. DEFAULT = 0.*
Manifesto	*VARCHAR(1000)*	*NOT NULL.*

FOREIGN KEY(Candidate_ID) REFERENCES Candidate(Candidate_ID)
FOREIGN KEY(Election_ID) REFERENCES Election(Election_ID)

Functional Dependencies :
Candidate_ID, *Election_ID* -> Candidate_ID, *Election_ID*,*Votes_Won, Manifesto.*
Candidate Key :
{ (*Candidate_ID*, *Election_ID)* }

Views
Media

Candidate,Biodata,Polititcal_Party,Constituency,Constituency,Candidate_Criminal_Cases,Stands_From

Only read

Users
DISCARDED IDEA
Voter,“abhay” , able to access: his record in people,Voter.
Priviledges:Can update his

xxx
discarded
Candidate,“Kishan” , able to access: his record in people,Voter.
Priviledges:Can update his record in BioData ,Member_Of

< end of document >
Timeline :

1. The EC starts a new election. A new entry is registered into **Election** table
2. The Election_Commision appoints employees with their roles and the system admin fills their records in the following tables :
-> Person (if such a record not already exists)
-> Voter
-> Employee_from_EC
-> Verificiation
-> Administration
->

no. of consti-70
1 to 20300 : People
1 to 20140 : Voter
20131 to 20200 : EC_Offical
20201 to 20270 : Verifcatiion_Official
20271 to 20275 : Admin Official
20276 to 20

voterid:2001 to 2300 random from 1 to 5
2001 to 2150 random from 1 to 5
2001 to 2050 random from1 to 5
cofficial:76 to 145
verification:1 to 70
admin:71 to 75
unverified voters : 2345 to 2500

election5:60-candidates
election4:60
election3:0
election2:0
election1:
60

1 to 210 elec5

1050 candidate

1 to 20140

Voted_for, CREATE TABLE `Voted_for` (
  `voter_id` int NOT NULL,
  `chosen_candidate_id` int NOT NULL,
  `Election_ID` int NOT NULL,
  PRIMARY KEY (`voter_id`),
  KEY `FK_Running_Candidate` (`chosen_candidate_id`),
  CONSTRAINT `FK_Voter` FOREIGN KEY (`voter_id`) REFERENCES `voter` (`id`),
  CONSTRAINT `Voted_for_ibfk_1` FOREIGN KEY (`chosen_candidate_id`) REFERENCES `Candidate`
(`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci

Unverified_User, CREATE TABLE `Unverified_User` (
  `Unverified_id` int NOT NULL,
  `Verification_type` varchar(30) NOT NULL,
  `Verification_document` blob NOT NULL,
  `V_official_id` int NOT NULL,
  PRIMARY KEY (`Unverified_id`),
  KEY `V_official_id` (`V_official_id`),
  CONSTRAINT `Unverified_User_ibfk_1` FOREIGN KEY (`Unverified_id`) REFERENCES `person`
(`ID`),
  CONSTRAINT `Unverified_User_ibfk_2` FOREIGN KEY (`V_official_id`) REFERENCES

```
`Verification_Official` (`id`),
  CONSTRAINT `v_type` CHECK ((`Verification_type` in
(_utf8mb4'Aadhar',_utf8mb4'Passport',_utf8mb4'Driving License')))
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci

---
BioData, CREATE TABLE `BioData` (
  `id` int NOT NULL,

  `Wealth` int NOT NULL DEFAULT '0',
  `party_affiliation` int NOT NULL,
  `profession` varchar(100) NOT NULL,
  `updated_year` year DEFAULT NULL,
  PRIMARY KEY (`id`,`Election_id`),
  KEY `FK_party` (`party_affiliation`),
  CONSTRAINT `FK_id` FOREIGN KEY (`id`) REFERENCES `person` (`ID`),
  CONSTRAINT `FK_party` FOREIGN KEY (`party_affiliation`) REFERENCES `Political_Party` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```