SVKM's NMIMS
School of Technology Management & Engineering, Chandigarh
A.Y. 2023 - 24
Course: Database Management Systems

Project Report

Program	MBA Tech Computer Engineering	
Semester	4	
Name of the Project:	Election Database Management System	
Details of Project Members		
Batch-B1	Roll No. A161	Name:Kshiti Shilke
Batch-B1	Roll No. A167	Name:Palash Gupta
Batch-B2	Roll No. A188	Name:Sharwari Yende
Date of Submission: 04-04-2024		

Contribution of each project Members:

Roll No.	Name:	Contribution
A161	Kshiti Shilke	Created 4 Tables (parties,positions,voter_hist voters) and respective queries, Worked on ER Model and Relational Model
A167	Palash Gupta	Created 4 Tables (local_body_elections, lok_sabha_elections, rajya_sabha_elections, state_assembly_elections) and respective queries, Worked on ER Model and Relational Model
A188	Sharwari Yende	Created 4 Tables (candidates,candidates_hist election,voter_registration) and respective queries, Worked on ER Model and Relational Model

Table of Contents

I. Storyline

The database project aims to design and implement a comprehensive database system for the Election Commission of India (ECI). The ECI is responsible for overseeing elections at various levels, including national, state, and local elections. The database will facilitate efficient management of voter registration, candidate information and election schedules.

II. Components of Database Design (entities, attributes, relationships)

Entities :

- Voters
- Voter_Registration
- candidates
- elections
- lok_sabha_elections
- state_assembly_elections
- local_body_elections
- rajya_sabha_elections
- parties
- positions
- Candidate_hist
- Voter_hist

Attributes

- **voters**

1. id
2. first_name
3. last_name
4. date_of_birth
5. gender
6. address
7. city
8. state
9. zip_code
10. email
11. phone_number

- **voter_registration**

1. voter_id
2. e_id
3. registration_date
4. FOREIGN KEY (voter_id) REFERENCES voters(id),
5. FOREIGN KEY (e_id) REFERENCES elections(id),
6. UNIQUE(voter_id, election_id)

- **candidates**

1. c_id
2. first_name
3. last_name
4. date_of_birth
5. gender
6. party_name
7. address
8. city
9. state
10. zip_code
11. email
12. phone_number

- **election**

1. e_id
2. election_date
3. election_type ENUM('General', 'Primary', 'Local', 'Special', 'Referendum') NOT NULL,
4. election_status ENUM('Upcoming', 'Ongoing', 'Completed') NOT NULL,
5. registration_deadline,
6. voting_start_date,
7. voting_end_date,
8. total_registered_voters

- **lok_sabha_elections**

1. *ls_Id*
2. Election_date
3. election_status ENUM('Upcoming', 'Ongoing', 'Completed') NOT NULL,
4. registration_deadline
5. voting_start_date
6. voting_end_date
7. total_registered_voters
8. total_votes_cast

- **state_assembly_elections**

1. s_id
2. election_date
3. election_status ENUM('Upcoming', 'Ongoing', 'Completed') NOT NULL,
4. registration_deadline
5. voting_start_date
6. voting_end_date
7. total_registered_voters

- **local_body_elections**

1. lb_id
2. election_date
3. election_type ENUM('Municipal', 'Panchayat', 'Other') NOT NULL,
4. election_status ENUM('Upcoming', 'Ongoing', 'Completed') NOT NULL,
5. registration_deadline,
6. voting_start_date
7. voting_end_date,
8. total_registered_voters,
9. region_name VARCHAR(100) NOT NULL

- **rajya_sabha_elections**

1. rs_id
2. election_date
3. election_status ENUM('Upcoming', 'Ongoing', 'Completed') NOT NULL,
4. registration_deadline,

5. voting_start_date,
6. voting_end_date,
7. total_registered_voters

- **parties**

1. p_ id
2. party_name
3. party_description
4. party_leader
5. foundation_date
6. headquarters_address ,
7. ideology

- **positions**

1. po_id,
2. position_name,
3. position_description
4. term_length
5. Term_limit
6. salary
7. requirements

- **Candidates_hist**

1. Ch_id
2. first_name
3. last_name
4. date_of_birth
5. gender
6. party_name
7. address
8. city
9. state
10. zip_code
11. email
12. phone_number

13. years_served

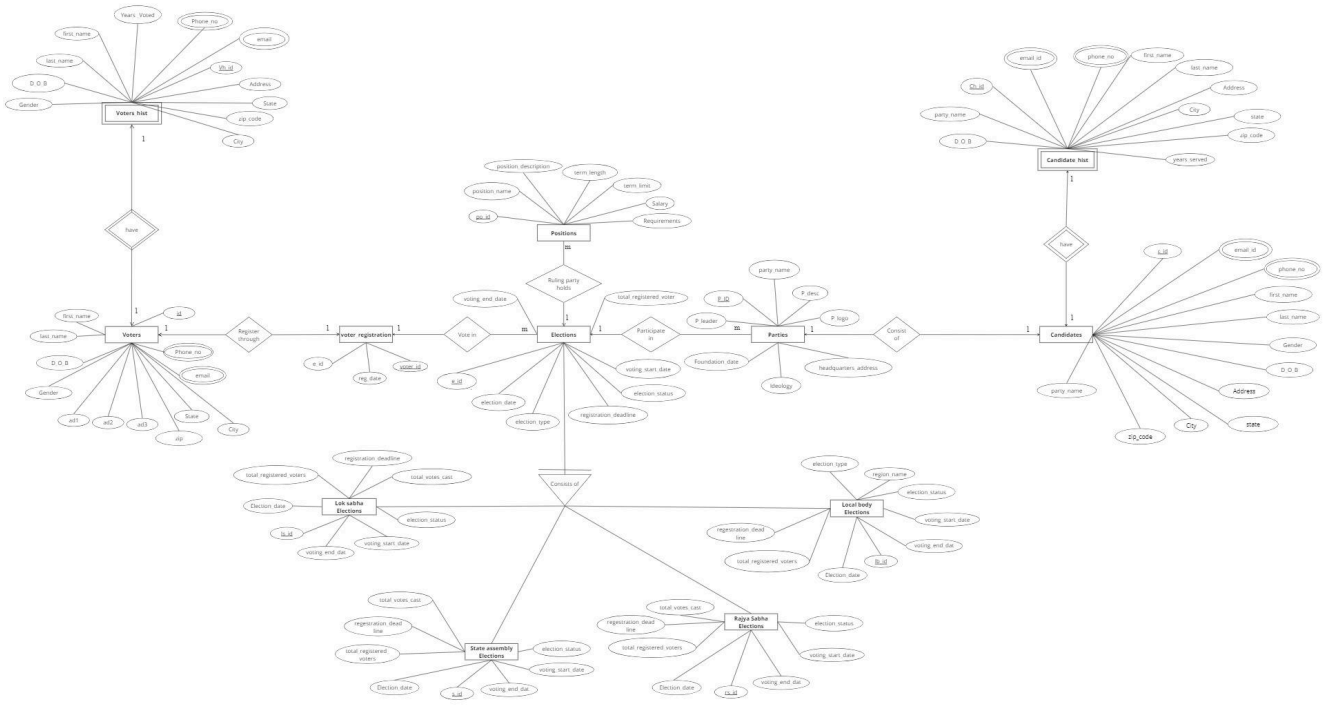
- **Voter_hist**

1. Vh_id
2. first_name
3. last_name
4. date_of_birth
5. gender
6. address
7. city
8. state
9. zip_code
10. years_voted

Relationships :

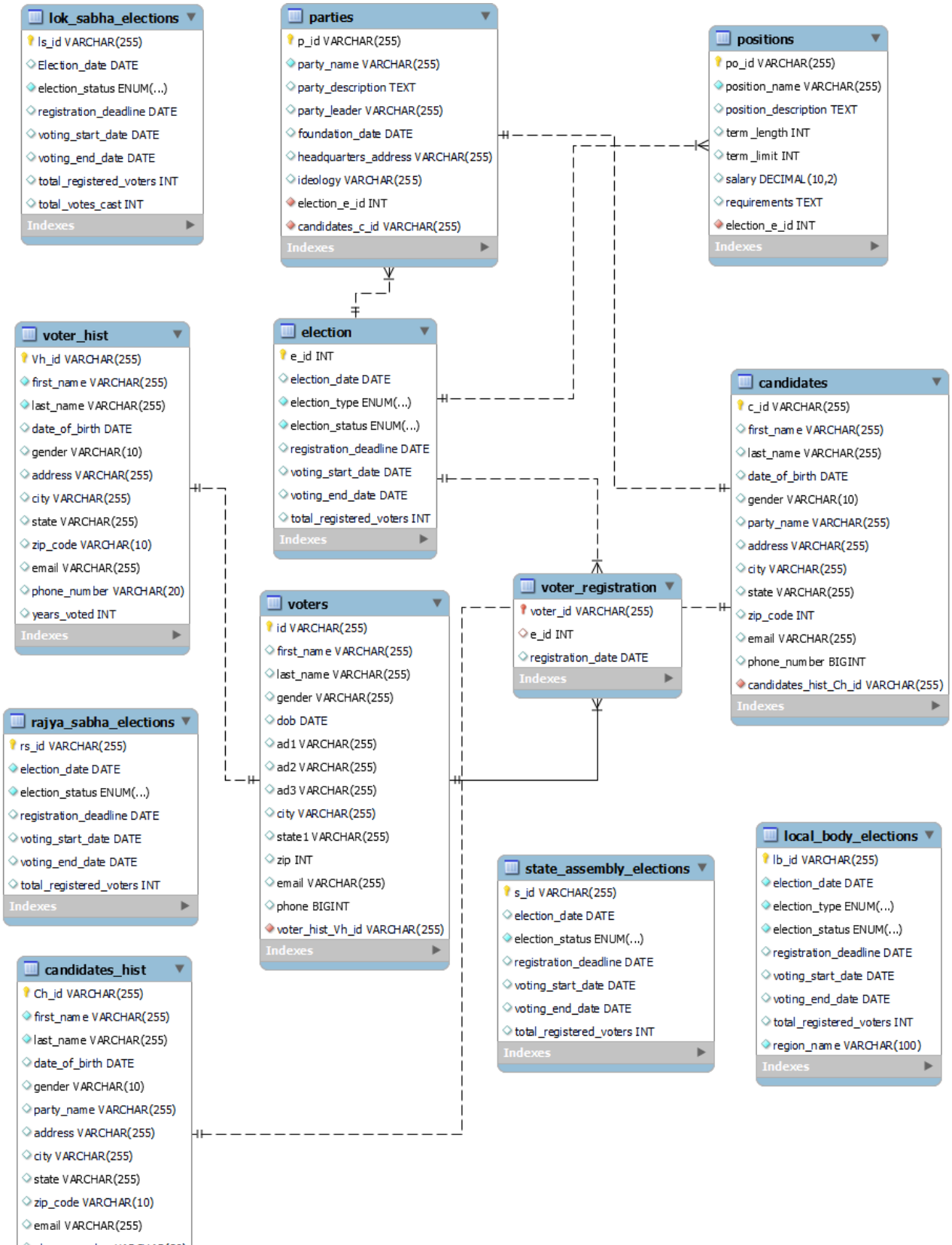
1. Voters REGISTERS THROUGH voter_registration
2. Voters REGISTERS THROUGH voter_registration VOTES IN elections.
3. Voters HAVE Voters_hist
4. elections RULING PARTY HOLDS positions
5. Parties CONSISTS OF Candidates
6. Candidates HAVE Candidate_hist
7. Elections CONSIST OF Loksabha elections
8. Elections CONSIST OF Rajyasabha elections
9. Elections CONSIST OF state assembly elections
10. Elections CONSIST OF local body elections

III. Entity Relationship Diagram



Link:https://miro.com/app/board/uXjVNgSmjDs=/

IV. Relational Model



V. Normalization

Perform normalization (1NF, 2NF, 3NF, BCNF) as applicable for the entire database.

To normalize the given schema into 1NF, 2NF, and 3NF, we need to identify the functional dependencies and then decompose the relations accordingly.

Functional Dependencies:
Let's identify some of the functional dependencies:

1. **In the voters relation:**

id → first_name, last_name, date_of_birth, gender, address, city, state, zip_code, email, phone_number

2. **In the voter_registration relation:**

{voter_id, election_id} → registration_date

3. **In the candidates relation:**

c_id → first_name, last_name, date_of_birth, gender, party_name, address, city, state, zip_code, email, phone_number

4. **In the election relation:**

e_id → election_date, election_type, election_status, registration_deadline, voting_start_date, voting_end_date, total_registered_voters

5. **In the lok_sabha_elections, state_assembly_elections, local_body_elections, and rajya_sabha_elections relations:**

ls_Id → election_date, election_status, registration_deadline, voting_start_date, voting_end_date, total_registered_voters
s_Id → election_date, election_status, registration_deadline, voting_start_date, voting_end_date, total_registered_voters
lb_Id → election_date, election_status, registration_deadline, voting_start_date, voting_end_date, total_registered_voters
rs_Id → election_date, election_status, registration_deadline, voting_start_date, voting_end_date, total_registered_voters

6. **In the parties relation:**

p_id → party_name,party_description, party_leader,foundation_date,headquarters_address, ideology

7. **In the positions relation:**

po_id → position_name, position_description, term_length, Term_limit, salary, requirements

8. **In the voter_hist relation:**

vh_id → first_name, last_name, date_of_birth, gender, address, city, state, zip_code, email, phone_number, years_voted

9. **In the candidates_hist relation:**

ch_id → first_name, last_name, date_of_birth, gender, party_name, address, city, state, zip_code, email, phone_number,years_served

Normalization:

1NF:

All relations appear to already be in 1NF, as there are no repeating groups or composite attributes.

2NF:

To achieve 2NF, we need to ensure that there are no partial dependencies. All non-key attributes should depend on the entire primary key.

- voter_registration relation is already in 2NF since the composite key {voter_id, election_id} determines the registration date, which is the only attribute in this relation.
- All other relations are already in 2NF as all attributes depend on the entire primary key.

3NF:

To achieve 3NF, we need to ensure that there are no transitive dependencies.

- voter_registration relation is already in 3NF.
- For other relations, there are no transitive dependencies.

Conclusion:

All relations appear to be in 1NF, 2NF, and 3NF based on the functional dependencies identified. Therefore, no further normalization is required.

VI. SQL Queries

SQL Queries:

1. **Select voters belonging to Maharashtra state**
   SELECT * FROM voters WHERE state1 = 'Maharashtra';

   O/P:

id	first_name	last_name	gender	dob	ad1	ad2	ad3	city	state1	zip	email	phone
2	Priya	Sharma	Female	1985-08-21	456 PQR Road	Opposite LMN School	Sector 12	Mumbai	Maharashtra	400001	priya.sharma@example.com	8765432109
7	Rakesh	Verma	Male	1998-07-12	345 BCD Road	Near HIJ Hospital	Sector 18	Pune	Maharashtra	411001	rakesh.verma@example.com	3210987654
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

2. **Select parties whose headquarters are based in New Delhi**

   SELECT party_name,party_leader,headquarters_address
   FROM parties
   WHERE headquarters_address = 'New Delhi';

   O/P:

party_name	party_leader	headquarters_address
Bharatiya Janata Party	Jagat Prakash Nadda	New Delhi
Bahujan Samaj Party	Mayawati	New Delhi
Communist Party of India (Marxist)	Sitaram Yechury	New Delhi
Indian National Congress	Sonia Gandhi	New Delhi

3. **Find the total number of registered voters for each election:**
   SELECT e_id, COUNT(*) AS total_registered_voters
   FROM voter_registration
   GROUP BY e_id;
   O/P:

e_id	total_registered_voters
1	2
2	1
3	1
4	2
5	2
6	2

4. **Find the elections that have surpassed their registration deadline but have not yet started:**

   SELECT *
   FROM election
   WHERE registration_deadline < CURDATE() AND voting_start_date > CURDATE();

   O/P:

e_id	election_date	election_type	election_status	registration_deadline	voting_start_date	voting_end_date	total_registered_voters
1	2024-05-01	General	Upcoming	2024-04-01	2024-05-15	2024-05-16	100000
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

5. **Select first name,last name and email from both voters and candidates where email id is same**

```
SELECT v.first_name AS voter_first_name,
    v.last_name AS voter_last_name,
    v.email AS voter_email,
    c.first_name AS candidate_first_name,
    c.last_name AS candidate_last_name,
    c.email AS candidate_email
FROM voters v
INNER JOIN candidates c ON v.email = c.email;
```

O/P:

voter_first_name	voter_last_name	voter_email	candidate_first_name	candidate_last_name	candidate_email
Vivek	Reddy	vivek.reddy@example.com	Vivek	Reddy	vivek.reddy@example.com
Pooja	Mehta	pooja.mehta@example.com	Pooja	Mehta	pooja.mehta@example.com

6.
Select firstname,lastname,DOB and gender of both Candidates and Voters from Candidates_hist and Voter_hist where gender is female and they reside in Mumbai.

```
SELECT first_name,
   last_name,
   date_of_birth,
   gender

FROM Candidates_hist
WHERE gender = 'Female' AND city = 'Mumbai'

UNION ALL

SELECT first_name,
   last_name,
   date_of_birth,
   gender
FROM Voter_hist
WHERE gender = 'Female' AND city = 'Mumbai';
```

O/P:

first_name	last_name	date_of_birth	gender
Aarti	Sharma	1985-08-20	Female
Priya	Sharma	1978-07-25	Female

7. Display the lok sabha election dates and local body election dates where their status is 'Onging'

```
SELECT l.election_date AS lok_sabha_election_date, lb.election_date AS
local_body_election_date
FROM lok_sabha_elections l
JOIN local_body_elections lb ON l.election_status = lb.election_status
WHERE l.election_status = 'Ongoing' AND lb.election_status = 'Ongoing';
```

O/P:

lok_sabha_election_date	local_body_election_date
2034-05-12	2026-04-20
2029-05-12	2026-04-20
2024-05-12	2026-04-20
2034-05-12	2023-08-20
2029-05-12	2023-08-20
2024-05-12	2023-08-20
2034-05-12	2024-09-30
2029-05-12	2024-09-30
2024-05-12	2024-09-30
2034-05-12	2025-10-15
2029-05-12	2025-10-15
2024-05-12	2025-10-15

8. **Select firstname,lastname,DOB and gender of both Candidates and Voters from Candidates_hist and Voter_hist where they reside in Delhi.**

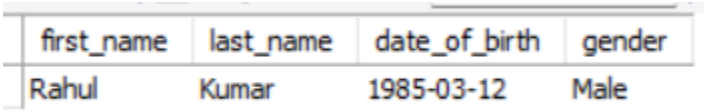
```
SELECT first_name,
    last_name,
    date_of_birth,
    gender
FROM Candidates_hist
WHERE city = 'Delhi'

UNION

SELECT first_name,
    last_name,
    date_of_birth,
    gender
FROM Voter_hist
```

WHERE city = 'Delhi';

O/P:



first_name	last_name	date_of_birth	gender
Rahul	Kumar	1985-03-12	Male

9. **Give the complete list containing firstname,lastname,DOB and gender of both Candidates and Voters from Candidates_hist and Voter_hist**

```
SELECT first_name,
    last_name,
    date_of_birth,
    gender
FROM Candidates_hist

UNION

SELECT first_name,
    last_name,
    date_of_birth,
    gender
FROM Voter_hist;
```

O/P:

first_name	last_name	date_of_birth	gender
Aarti	Sharma	1985-08-20	Female
Vivek	Patel	1978-12-10	Male
Anjali	Das	1982-03-25	Female
Sanjay	Singh	1987-07-10	Male
Priya	Gupta	1976-09-18	Female
Amit	Rao	1984-04-30	Male
Sunita	Verma	1989-11-12	Female
Rahul	Mehta	1981-06-27	Male
Deepika	Singh	1990-02-14	Female
Rahul	Kumar	1985-03-12	Male
Priya	Sharma	1978-07-25	Female
Amit	Singh	1990-11-18	Male
Anjali	Patel	1983-05-06	Female
Raj	Verma	1976-09-30	Male
Sneha	Gupta	1988-02-14	Female
Vivek	Yadav	1981-11-08	Male

10. **Insert one value in Candidates_hist table**

insert into Candidates_hist (Ch_id, first_name, last_name, date_of_birth, gender, party_name, address, city, state, zip_code, email, phone_number)
values('CH011', 'Meera', 'Kaur', '1980-12-30', 'Female','Janata Dal (United)', '852 UVW Street', 'Lucknow', 'Uttar Pradesh', '226001', 'meera.kaur@example.com', '9876543210');

O/P:

Ch_id	first_name	last_name	date_of_birth	gender	party_name	address	city	state	zip_code	email
CH001	Rajesh	Kumar	1980-05-15	Male	Bharatiya Janata Party	234 GHI Road	New Delhi	Delhi	110001	rajesh.kumar@example.com
CH002	Aarti	Sharma	1985-08-20	Female	Indian National Congress	456 JKL Street	Mumbai	Maharashtra	400001	aarti.sharma@example.com
CH003	Vivek	Patel	1978-12-10	Male	Aam Aadmi Party	789 MNO Lane	Bengaluru	Karnataka	560001	vivek.patel@example.com
CH004	Anjali	Das	1982-03-25	Female	Trinamool Congress	321 PQR Avenue	Kolkata	West Bengal	700001	anjali.das@example.com
CH005	Sanjay	Singh	1987-07-10	Male	Shiv Sena	987 STU Lane	Pune	Maharashtra	411001	sanjay.singh@example.com
CH006	Priya	Gupta	1976-09-18	Female	Bahujan Samaj Party	654 VWX Road	Chennai	Tamil Nadu	600001	priya.gupta@example.com
CH007	Amit	Rao	1984-04-30	Male	Nationalist Congress Party	147 YZT Street	Hyderabad	Telangana	500001	amit.rao@example.com
CH008	Sunita	Verma	1989-11-12	Female	Janata Dal (United)	369 ABC Road	Lucknow	Uttar Pradesh	226001	sunita.verma@example.com
CH009	Rahul	Mehta	1981-06-27	Male	All India Anna Dravida Munnetra Kazhagam	258 DEF Street	Chandigarh	Chandigarh	160001	rahul.mehta@example.com
CH010	Deepika	Singh	1990-02-14	Female	Telangana Rashtra Samithi	852 GHI Road	Jaipur	Rajasthan	302001	deepika.singh@example.com
CH011	Meera	Kaur	1980-12-30	Female	Janata Dal (United)	852 UVW Street	Lucknow	Uttar Pradesh	226001	meera.kaur@example.com
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

11.**Find the Top 5 States with the Highest Voter Registration:**

```
SELECT v.state1, COUNT(vr.voter_id) AS registered_voters_count
FROM voters v
JOIN voter_registration vr ON v.id = vr.voter_id
GROUP BY v.state1
ORDER BY registered_voters_count DESC
LIMIT 5;
```

O/P:

state1	registered_voters_count
Maharashtra	2
Delhi	1
Uttar Pradesh	1
Karnataka	1
West Bengal	1

12. Identify the Most Popular Party Affiliation among Candidates:

```
SELECT party_name, COUNT(*) AS candidate_count
FROM candidates
GROUP BY party_name
ORDER BY candidate_count DESC;
```

O/P:

party_name	candidate_count
Indian National Congress	3
Aam Aadmi Party	2
Shiv Sena	2
All India Trinamool Congress	1
Samajwadi Party	1
Bahujan Samaj Party	1

13. Find the Average Voter Age by Gender:

```
SELECT gender,
      AVG(DATEDIFF(CURRENT_DATE, dob) / 365) AS average_age
FROM voters
GROUP BY gender;
```

O/P:

gender	average_age
Male	35.13642000
Female	35.80164000

14. Find the Positions with the Highest Salary:

```
SELECT position_name, salary AS highest_salary
FROM positions
WHERE salary = (
   SELECT MAX(salary)
   FROM positions
);
```

O/P:

position_name	highest_salary
Prime Minister	100000.00

15. Find the number of registered voters in each city for upcoming elections:

```
SELECT v.city, COUNT(vr.voter_id) AS num_registered_voters
FROM voter_registration vr
INNER JOIN voters v ON vr.voter_id = v.id
INNER JOIN election e ON vr.e_id = e.e_id
WHERE e.election_status = 'Upcoming'
GROUP BY v.city;
```

O/P:

city	num_registered_voters
Delhi	1
Mumbai	1
Chennai	1
Hyderabad	1
Lucknow	1
Jaipur	1

16. List the parties and the number of candidates they have:

```
SELECT p.party_name, COUNT(c.c_id) AS num_candidates
FROM parties p
LEFT JOIN candidates c ON p.party_name = c.party_name
GROUP BY p.party_name;
```

O/P:

party_name	num_candidates
Aam Aadmi Party	2
All India Anna Dravida Munnetra Kazhagam	0
Bharatiya Janata Party	0
Bahujan Samaj Party	1
Communist Party of India (Marxist)	0
Dravida Munnetra Kazhagam	0
Indian National Congress	3
Shiv Sena	2
All India Trinamool Congress	1
Telangana Rashtra Samithi	0

17. List all elections along with the percentage of registered voters who actually voted:

```
SELECT e.e_id, e.election_type,
    (COUNT(vr.voter_id) * 100 / e.total_registered_voters) AS voting_percentage
FROM election e
INNER JOIN voter_registration vr ON e.e_id = vr.e_id
GROUP BY e.e_id, e.election_type, e.total_registered_voters;
```

O/P:

e_id	election_type	voting_percentage
1	General	0.0020
2	Local	0.0013
3	Referendum	0.0008
4	Primary	0.0022
5	General	0.0018
6	Special	0.0025

VI. Project demonstration

VII. Self -Learning beyond classroom:

- What new aspects did you learn on your own ? You have to mention learning beyond the classroom

  1- Overall, working on the Election Commission of India database in MySQL provided us with practical insights into real-world database management scenarios and enhanced our skills in database design, optimization, and administration.

  2-We've tried connecting MySQL, relational database management system, with our website. Essentially, it allowed us to store and manage information efficiently, organizing it into tables with rows and columns. This approach ensured data integrity. We've streamlined our website's functionality, enhancing its scalability. This integration represents a strategic move towards harnessing the power of databases to effectively handle and process data for our website's operations.

VIII. Learning from the Project

Include learning from the project:

- How did this project help you?

  1-Practical Application of Database Concepts: Building the database for the Election Commission of India allowed me to apply theoretical knowledge of database concepts in a real-world scenario. I gained hands-on experience in database design, normalization, and optimization, which are fundamental skills for any database administrator or developer.

  2-Problem-Solving and Decision-Making: Throughout the project, I encountered various challenges, such as resolving data inconsistencies, addressing normalization anomalies, and optimizing query execution plans. Overcoming these challenges honed my problem-solving skills and taught me how to make informed decisions to ensure the database's effectiveness and reliability.

  3-Query Optimization: Developing queries to retrieve specific information from the database required optimization techniques to enhance query performance. By experimenting with indexing, query restructuring, and other optimization strategies, I learned how to retrieve data efficiently, even from large datasets.

IX. Challenges Faced

- Working with limited sample data leads to some challenges in visualizing the output.

1-Data Complexity: Election databases often contain complex relationships between entities such as voters, candidates, elections, parties, and voting regions. Managing these interconnections while ensuring data integrity can be challenging.

2-Security Concerns: Ensuring the security and integrity of election data is paramount. Implementing robust security measures to prevent unauthorized access, tampering, or hacking attempts is crucial to maintain the credibility of the electoral process.

3-Regulatory Compliance: Election databases must comply with various regulatory frameworks and electoral laws. Ensuring that the database design meets legal requirements related to data storage, access control, and reporting is crucial for legal compliance and public trust.

X. Conclusion

This project gave us a glimpse of the working of database modeling and management. It gave us an opportunity to work with the various entities that take part in an election and the kind of requirements that database has.

It also enabled us to gain experience implementing various MySQL queries and reducing redundancies in data.