DATABASE MANAGEMENT SYSTEM

Project Synopsis On :

Election Voting SYSTEM

SUBMITTED TO: Swati Sharma

SUBMITTED BY:

Kritika (102303415)

Arsh Garg (102303414)

Simanta (102483083)

INDEX

INTRODUCTION

In any democratic country, elections are the pillars of people's involvement and equitable administration. Nonetheless, conventional voting mechanisms are commonly associated with lengthy lines, poor management, manipulated votes, and inaccessible voters. As the pace of digitalization speeds up, it is important to adopt technology in enhancing electoral processes to make them more efficient, secure, and transparent.

This "Online Voting System" project is a database management system that aims to automate and make the process of voting more secure by converting it into a digital process. The system facilitates registered voters to cast their votes through electronic means easily, thus making the process more accessible with less operational load.

The objective of the Online Voting System initiative is to:
1.Ensure that a database of eligible electors and candidates is maintained.
2.Securely authenticate users .
3.Keep a record of the ballots, ensuring that each individual votes only once..
4.Proper data administration is essential to prevent duplication and tampering.

This DBMS project has admin management, voter registration, candidate registration, casting of votes, and compilation of results, all managed through well-organized relational databases.It is implemented using SQL and PL/SQL to maintain consistency, security, and correct relational mapping. With increasing awareness regarding online governance, this project proves that database systems can have a central role in facilitating equitable, secure, and scalable elections.

Project Resource Requirements:

A database is an application that stores the organized collection of records. It can be accessed and manage by the user very easily. Today, many databases available like MySQL, Sybase, Oracle, MongoDB, PostgreSQL, SQL Server, etc. We used MySQL server for managing data.

MySQL is a relational database management system based on the Structured Query Language,which is the popular language for accessing and managing the records in the database. MySQLis open-source and free software under the GNU license. It is supported by Oracle Company.

MySQL is a widely used relational database management system (RDBMS). It is free and open-source. MySQL is ideal for both small and large applications.
MySQL follows the working of Client-Server Architecture. This model is designed for the end-
users called clients to access the resources from a central computer known as a server using

network services. Here, the clients make requests through a graphical user interface (GUI), and the server will give the desired output as soon as the instructions are matched. The process of.MySQL environment is the same as the client-server model.

MySQL is a very powerful program that can handle a large set of functionality of the most expensive and powerful database packages. It supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).

ER Analysis:

Identifying Entity Sets and Relationship Sets:

Entity Sets:

I. Admin

a) Username

b) Password

II. Voter

a) ID

b) Name

c) Password

d) Date of Birth

e) Address

f) Constituency Number

III. Candidate

a) Candidate ID (id_c)

b) Name

c) Password

d) Constituency Number

e) Admin Username (foreign key)

IV. Party

a) Party Name

b) Founder

c) Date of Founding

d) Party Chief

V. Members

a) Member Name

b) Member ID (id_m)

c) Address

d) Date of Birth

e) Party Name

f) Candidate ID (foreign key)

VI. Area

a) Area Name

b) Constituency Number

VII. E-Ballot

a) Serial Number (s_no)

b) Date of Vote

c) Party Name

d) Constituency Number

e) Voter ID

f) Admin Username

VIII. Results

a) Number of Votes

b) Candidate ID

c) Constituency Number

d) Admin Username

Relationship Sets:

1. Keeps_Check_On

Admin monitors or keeps check on a Party

2. Manages

Admin manages a Voter

3. Elects / Votes

Voter casts vote via E-Ballot for a Party or Candidate

4. Registers / View_Profile

Voter views the profile of a Candidate

Indicates interest or engagement with a candidate

5. Stands_From

Candidate stands for election from an Area

6. Counts_Vote

Admin counts votes and stores result in Results

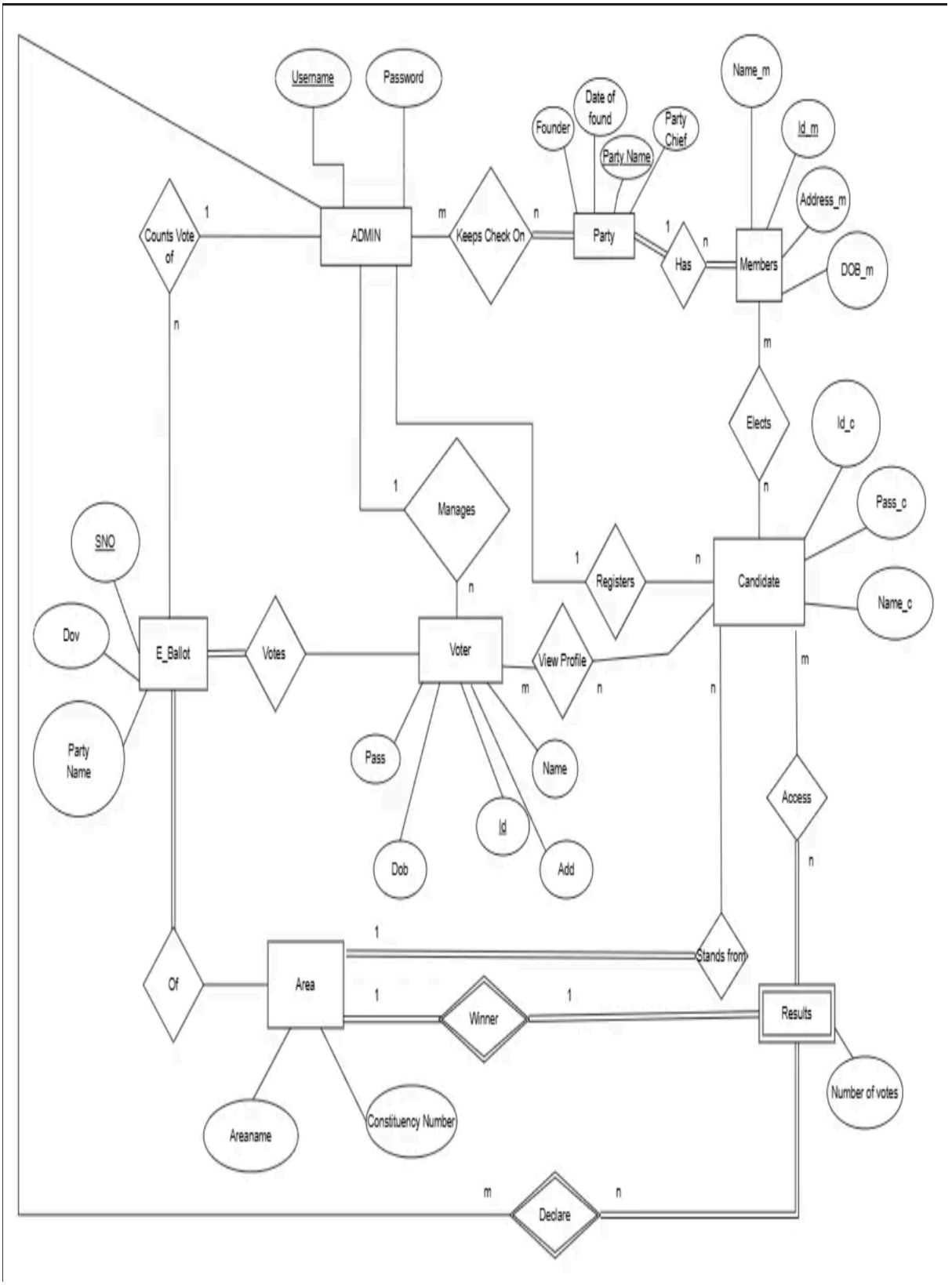
7. Declare

The system declares a winner based on Results

8. Belongs_To (Members)

Members are part of a Party and linked to a Candidate

ER-Diagram


Username
Password
Name_m
Date of found
Founder
Party Chief
Party Name
Id_m
Address_m
DOB_m
Counts Vote of
1
ADMIN
m
Keeps Check On
n
Party
1
Has
n
Members
n
m
Elects
Id_c
Pass_c
Name_c
Candidate
n
1
Manages
1
Registers
n
SNO
Dov
E_Ballot
Votes
Voter
View Profile
m
n
n
m
Party Name
Pass
Name
Id
Dob
Add
Access
n
1
Stands from
Of
Area
1
Winner
1
Results
Number of votes
Areaname
Constituency Number
m
Declare
n

Conversion of ER to Tables

1.Admin (username, password)

2.Party (party_name, founder, date_of_found, party_chief)

3.Area (area_name, constituency_number)

4.Voter (id, pass, name, DOB, address, constituency_number)

{constituency_number} is foreign key referencing constituency_number of table

5.Candidate (id_c, password_c, name_c, constituency_number, username)

{username} is foreign key referencing username of table Admin

{constituency_number} is foreign key referencing constituency_number of table

6.Members (name_m, id_m, address_m, Dob_m, party_name, id_c,phone_no.)

{party_name} is foreign key referencing party_name of table Party

{id_c} is foreign key referencing id_c of table Candidate

7.E_Ballot (s_no, date_of_vote, party_name, constituency_number, id, username)

{party_name} is foreign key referencing party_name of table Party

{constituency_number} is foreign key referencing constituency_number of table

8.Results (number_of_votes, id_c, constituency_number, username)

{id_c} is foreign key referencing id_c of table Candidate

{constituency_number} is foreign key referencing constituency_number of table

9.Keeps_Check_On (username, party_name)

{username} is foreign key referencing username of table Admin

{party_name} is foreign key referencing party_name of table Party

10.View_Profile (id, id_c)

{id} is foreign key referencing id of table Voter

{id_c} is foreign key referencing id_c of table Candidate

11.Manages (username, id)

{username} is foreign key referencing username of table Admin

{id} is foreign key referencing id of table Voter

Normalization

Expt. No. _____ Date _____

Page No. _____

Normalization

① Members table info :

(id_c, password_c, name_c, constituency_number, username), phone number)

Here, phone number is a multivalued attribute.
So, it is not in 1NF.
∴ Make two tables, user & user phn. no.

So,

User/Member table :

id_c	password_c	name_c	constituency_number	username

and

User phone_no :-

id_c	phone_no.

Now the tables are normalized.

② Other than that, rest of the tables are already normalized and don't need further normalization.

Teacher's Signature _____

Functional Dependencies

Expt. No. _____ Date _____

Page No. _____

Functional Dependencies

1) Admin:

username → password

2) Party:

party_name → founder, Date_of_found, party_chief

3) Area:

Constituency number → Area_name

4) Voter:

id → pass, name, DoB, address

5) Candidate:

id_c → passcoard_c, name_c...

6) Members:

id_m → name_m, address_m, DoB_m, party_name, id_c, phone_no

7) E_Ballot:

s_no → date_of_vote, party_name, constituency_number

8) Results:

id → number of votes, constituency_number, username

9) Keeps_Check_On:

user_name → party_name

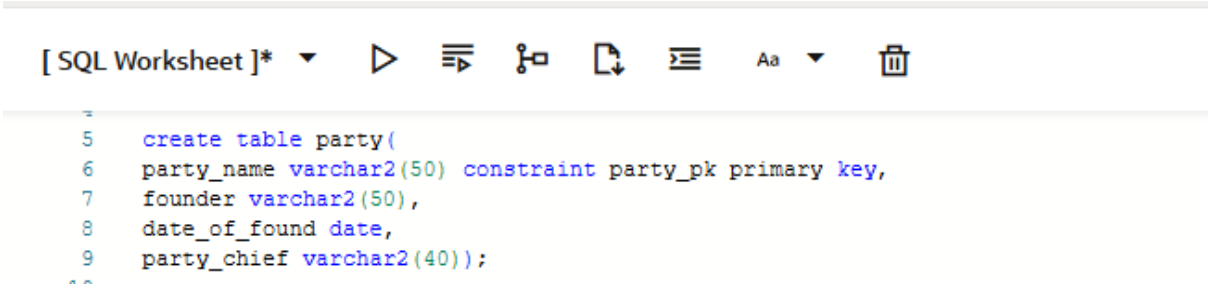
10) View Profile:

id, id_c [no dependency]

Teacher's Signature _____

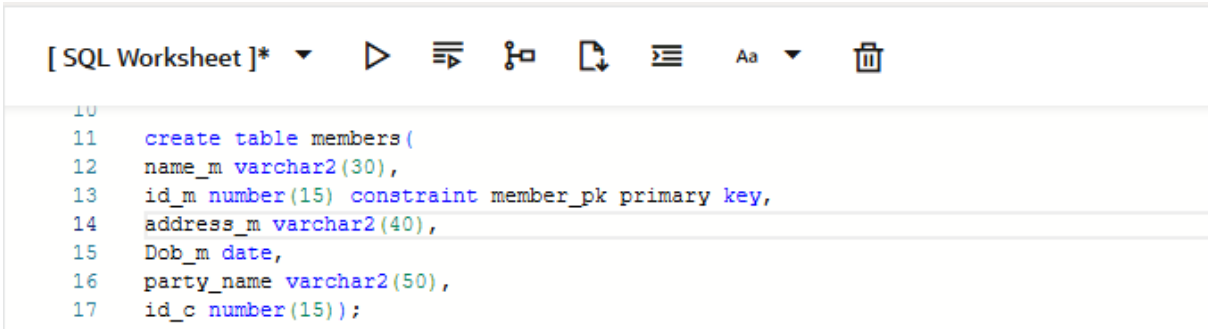
CREATING TABLES

1.create table party



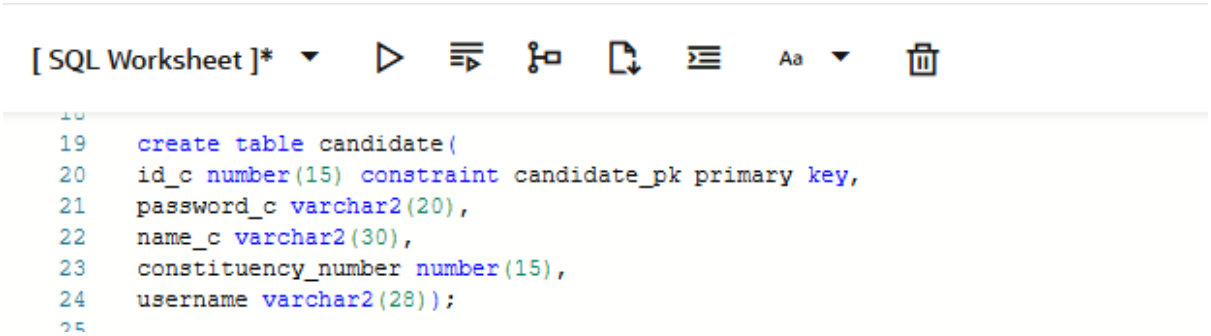

```
create table party(
party_name varchar2(50) constraint party_pk primary key,
founder varchar2(50),
date_of_found date,
party_chief varchar2(40));
```

2.create table members



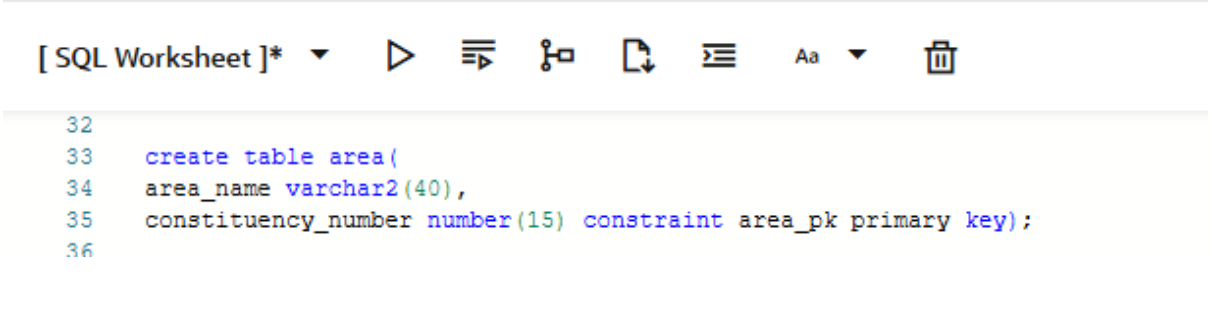

```
create table members(
name_m varchar2(30),
id_m number(15) constraint member_pk primary key,
address_m varchar2(40),
Dob_m date,
party_name varchar2(50),
id_c number(15));
```

3.create table candidate




```
create table candidate(
id_c number(15) constraint candidate_pk primary key,
password_c varchar2(20),
name_c varchar2(30),
constituency_number number(15),
username varchar2(28));
```

4.create table area




```
create table area(
area_name varchar2(40),
constituency_number number(15) constraint area_pk primary key);
```

5.create table e_ballot


```
37  create table e_ballot(
38  s_no number(15) constraint e_ballot_pk primary key,
39  date_of_vote date,
40  party_name varchar2(50),
41  constituency_number number(15),
42  id number(15),
43  username varchar2(28));
```

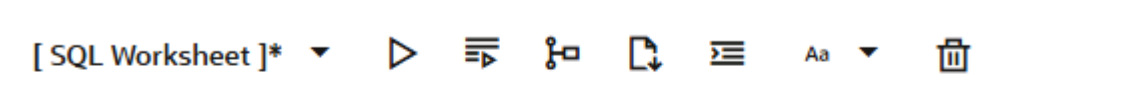
6.create table voter


```
45  create table voter(
46  id number(15) constraint voter_pk primary key,
47  pass varchar2(20),
48  name varchar2(35),
49  DOB date,
50  address varchar2(55),
51  constituency_number number(15));
```

7.create table keeps_check_on


```
53  create table keeps_check_on(
54  username varchar2(28),
55  party_name varchar2(50),
56  constraint keeps_check_on_pk primary key(username,party_name));
```

8.create table manages




```
63  create table manages(
64  username varchar2(28),
65  id number(15),
66  constraint manages_pk primary key(username,id));
67
```

9.create table member_phone

[SQL Worksheet]*

```
67
68  CREATE TABLE member_phone (
69      id_m NUMBER(15),
70      phone_no VARCHAR2(15),
71      CONSTRAINT member_phone_pk PRIMARY KEY (id_m, phone_no),
72      CONSTRAINT member_phone_fk FOREIGN KEY (id_m) REFERENCES members(id_m)
73  );
```

Output-

Query result | Script output | DBMS output | Explain Plan | SQL history

```
SQL> create table manages(
     username varchar2(28),
     id number(15),
     constraint manages_pk primary key(username,id))


Table MANAGES created.

Elapsed: 00:00:00.015
```

```
SQL> create table admin(
     username varchar2(28) constraint admin_pk primary key,
     password varchar2(20))


Table ADMIN created.

Elapsed: 00:00:00.013
```

```
SQL> create table party(
     party_name varchar2(50) constraint party_pk primary key,
     founder varchar2(50),
     date_of_found date,...
Show more...
```

Query result | Script output | DBMS output | Explain Plan | SQL history

```
SQL> create table members(
     name_m varchar2(30),
     id_m number(15) constraint member_pk primary key,
     address_m varchar2(40),...
Show more...


Table MEMBERS created.

Elapsed: 00:00:00.010
```

```
SQL> create table candidate(
     id_c number(15) constraint candidate_pk primary key,
     password_c varchar2(20),
     name_c varchar2(30),...
Show more...


Table CANDIDATE created.

Elapsed: 00:00:00.010
```

```
SQL> create table results(
     number_of_votes number(15),
     id_c number(15),
     constituency_number number(15),...
Show more...
```

Query result | Script output | DBMS output | Explain Plan | SQL history

```
SQL> create table area(
     area_name varchar2(40),
     constituency_number number(15) constraint area_pk primary key)


Table AREA created.

Elapsed: 00:00:00.011
```

```
SQL> create table e_ballot(
     s_no number(15) constraint e_ballot_pk primary key,
     date_of_vote date,
     party_name varchar2(50),...
Show more...


Table E_BALLOT created.

Elapsed: 00:00:00.009
```

```
SQL> create table voter(
     id number(15) constraint voter_pk primary key,
     pass varchar2(20),
     name varchar2(35),...
Show more...


Table VOTER created.
```

Query result | Script output | DBMS output | Explain Plan | SQL history

```
     name varchar2(35),...
Show more...


Table VOTER created.

Elapsed: 00:00:00.010
```

```
SQL> create table keeps_check_on(
     username varchar2(28),
     party_name varchar2(50),
     constraint keeps_check_on_pk primary key(username,party_name))


Table KEEPS_CHECK_ON created.

Elapsed: 00:00:00.009
```

```
SQL> create table view_profile(
     id number(15),
     id_c number(15),
     constraint view_profile_pk primary key(id,id_c))


Table VIEW_PROFILE created.

Elapsed: 00:00:00.010
```

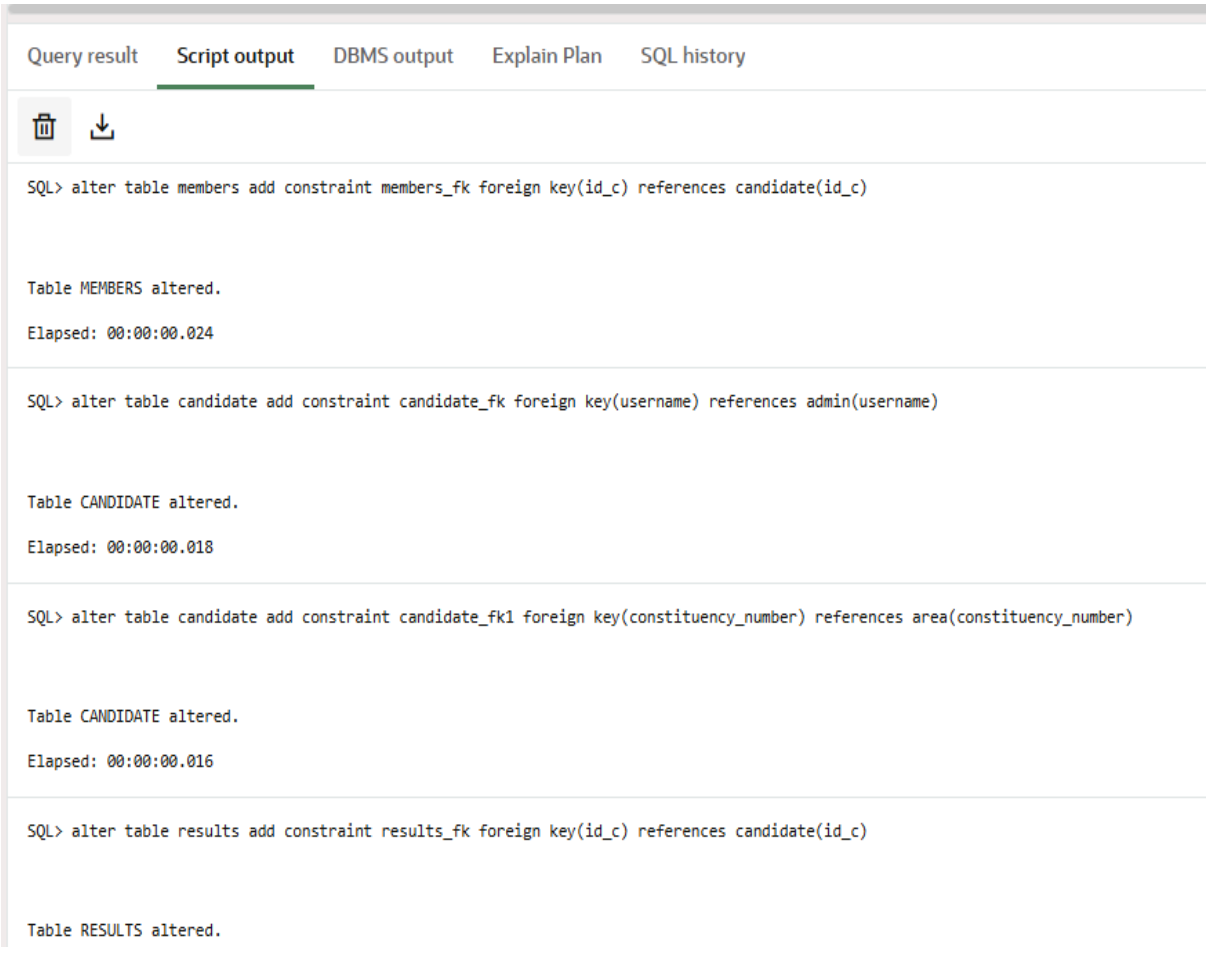
```
SQL> CREATE TABLE member_phone (
          id_m NUMBER(15),
          phone_no VARCHAR2(15),
          CONSTRAINT member_phone_pk PRIMARY KEY (id_m, phone_no),...
Show more...


Table MEMBER_PHONE created.
```

FOREIGN KEY CONSTRAINTS-

```
68
69
70  alter table members add constraint members_fk foreign key(id_c) references candidate(id_c);
71  alter table candidate add constraint candidate_fk foreign key(username) references admin(username);
72  alter table candidate add constraint candidate_fk1 foreign key(constituency_number) references area(constituency_number);
73  alter table results add constraint results_fk foreign key(id_c) references candidate(id_c);
74  alter table results add constraint results_fk1 foreign key(constituency_number) references area(constituency_number);
75  alter table results add constraint results_fk2 foreign key(username) references admin(username);
76  alter table e_ballot add constraint e_ballot_fk2 foreign key(party_name) references party(party_name);
77  alter table e_ballot add constraint e_ballot_fk1 foreign key(constituency_number) references area(constituency_number);
78  alter table e_ballot add constraint e_ballot_fk foreign key(id) references voter(id);
79  alter table e_ballot add constraint e_ballot_fk4 foreign key(username) references admin(username);
80  alter table voter add constraint voter_fk foreign key(constituency_number) references area(constituency_number);
81  alter table keeps_check_on add constraint keeps_check_on_fk foreign key(username) references admin(username);
82  alter table keeps_check_on add constraint keeps_check_on_fk1 foreign key(party_name) references party(party_name);
83  alter table view_profile add constraint view_profile_fk1 foreign key(id) references voter(id);
84  alter table manages add constraint manages_fk foreign key(username) references admin(username);
85  alter table manages add constraint manages_fk1 foreign key(id) references voter(id);
86
87
```

OUTPUT-



Query result | Script output | DBMS output | Explain Plan | SQL history

```
SQL> alter table e_ballot add constraint e_ballot_fk foreign key(id) references voter(id)

Table E_BALLOT altered.

Elapsed: 00:00:00.013
```

```
SQL> alter table e_ballot add constraint e_ballot_fk4 foreign key(username) references admin(username)

Table E_BALLOT altered.

Elapsed: 00:00:00.013
```

```
SQL> alter table voter add constraint voter_fk foreign key(constituency_number) references area(constituency_number)

Table VOTER altered.

Elapsed: 00:00:00.014
```

```
SQL> alter table keeps_check_on add constraint keeps_check_on_fk foreign key(username) references admin(username)

Table KEEPS_CHECK_ON altered.
```

Query result | Script output | DBMS output | Explain Plan | SQL history

```
SQL> alter table results add constraint results_fk1 foreign key(constituency_number) references area(constituency_number)

Table RESULTS altered.

Elapsed: 00:00:00.013
```

```
SQL> alter table results add constraint results_fk2 foreign key(username) references admin(username)

Table RESULTS altered.

Elapsed: 00:00:00.013
```

```
SQL> alter table e_ballot add constraint e_ballot_fk2 foreign key(party_name) references party(party_name)

Table E_BALLOT altered.

Elapsed: 00:00:00.015
```

```
SQL> alter table e_ballot add constraint e_ballot_fk1 foreign key(constituency_number) references area(constituency_number)

Table E_BALLOT altered.
```

Query result | Script output | DBMS output | Explain Plan | SQL history

```
SQL> alter table keeps_check_on add constraint keeps_check_on_fk1 foreign key(party_name) references party(party_name)


Table KEEPS_CHECK_ON altered.

Elapsed: 00:00:00.013
```

```
SQL> alter table view_profile add constraint view_profile_fk1 foreign key(id) references voter(id)


Table VIEW_PROFILE altered.

Elapsed: 00:00:00.014
```

```
SQL> alter table manages add constraint manages_fk foreign key(username) references admin(username)


Table MANAGES altered.

Elapsed: 00:00:00.014
```

```
SQL> alter table manages add constraint manages_fk1 foreign key(id) references voter(id)


Table MANAGES altered.
```

Inserting the data into created tables:

1.Insert into admin

```
134
135    INSERT INTO admin VALUES ('mohitverma6161','mohit123');
136    INSERT INTO admin VALUES ('aayush123','canadanoor');
137    INSERT INTO admin VALUES ('purushottam123','gangleader');
138    INSERT INTO admin VALUES ('prateek','coder');
```

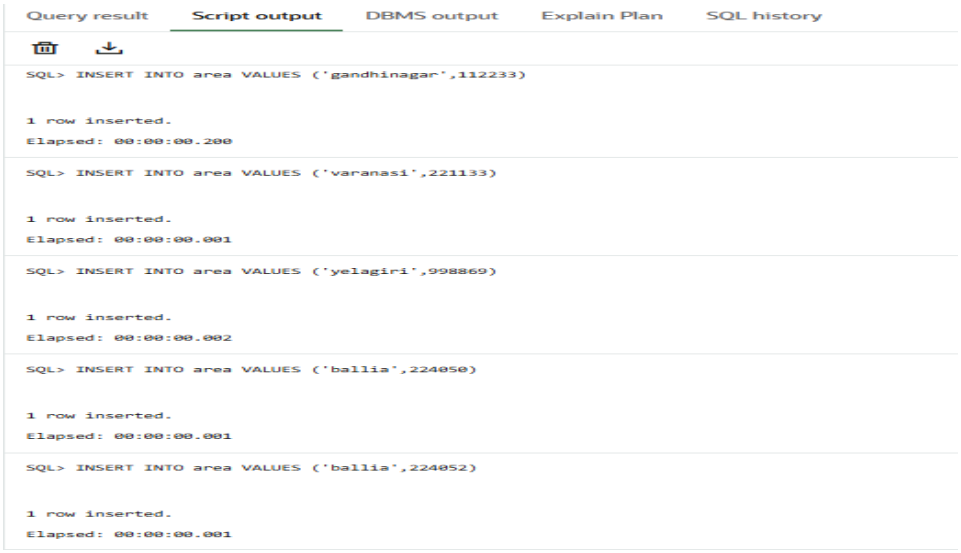
OUTPUT

-

2.Insert into area

```
139
140    INSERT INTO area VALUES ('gandhinagar',112233);
141    INSERT INTO area VALUES ('varanasi',221133);
142    INSERT INTO area VALUES ('yelagiri',998869);
143    INSERT INTO area VALUES ('ballia',224050);
144    INSERT INTO area VALUES ('ballia',224052);
```

OUTPUT-



3. Insert into voters

```
145
146    INSERT INTO voter VALUES (123, 'dbms1', 'Adarsh', TO_DATE('1998-08-15', 'YYYY-MM-DD'), 'Bhubhaneshwar', 112233);
147    INSERT INTO voter VALUES (111, 'dbms2', 'Siddharth', TO_DATE('1997-01-26', 'YYYY-MM-DD'), 'Delhi', 998869);
148    INSERT INTO voter VALUES (222, 'dbms3', 'Sankalp', TO_DATE('1996-10-02', 'YYYY-MM-DD'), 'Ghaziabad', 998869);
149    INSERT INTO voter VALUES (333, 'dbms4', 'Deepak', TO_DATE('1996-11-14', 'YYYY-MM-DD'), 'Bengaluru', 224050);
150    INSERT INTO voter VALUES (444, 'dbms5', 'Yuvi', TO_DATE('1995-10-28', 'YYYY-MM-DD'), 'Ballia', 224050);
151    INSERT INTO voter VALUES (555, 'dbms6', 'Dhoni', TO_DATE('1995-12-28', 'YYYY-MM-DD'), 'Ranchi', 224050);
152    INSERT INTO voter VALUES (666, 'dbms7', 'Shikhar', TO_DATE('1993-03-11', 'YYYY-MM-DD'), 'Lucknow', 224050);
153    INSERT INTO voter VALUES (777, 'dbms8', 'Rohit', TO_DATE('1993-01-10', 'YYYY-MM-DD'), 'Ludhiana', 998869);
154    INSERT INTO voter VALUES (888, 'dbms9', 'Suresh', TO_DATE('1999-12-28', 'YYYY-MM-DD'), 'Hisar', 112233);
155    INSERT INTO voter VALUES (999, 'dbms10', 'Ajinkya', TO_DATE('1999-11-28', 'YYYY-MM-DD'), 'Chennai', 224052);
156    INSERT INTO voter VALUES (998, 'dbms11', 'Pandya', TO_DATE('1999-06-21', 'YYYY-MM-DD'), 'Chennai', 224052);
157    INSERT INTO voter VALUES (988, 'dbms12', 'Krunal', TO_DATE('1999-06-21', 'YYYY-MM-DD'), 'Pondicherry', 998869);
158
```

OUTPUT-

Query result | Script output | DBMS output | Explain Plan | SQL history

```
SQL> INSERT INTO voter VALUES (555, 'dbms6', 'Dhoni', TO_DATE('1995-12-28', 'YYYY-MM-DD'), 'Ranchi', 224050)

1 row inserted.

Elapsed: 00:00:00.001

SQL> INSERT INTO voter VALUES (666, 'dbms7', 'Shikhar', TO_DATE('1993-03-11', 'YYYY-MM-DD'), 'Lucknow', 224050)

1 row inserted.

Elapsed: 00:00:00.001

SQL> INSERT INTO voter VALUES (777, 'dbms8', 'Rohit', TO_DATE('1993-01-10', 'YYYY-MM-DD'), 'Ludhiana', 998869)

1 row inserted.

Elapsed: 00:00:00.001

SQL> INSERT INTO voter VALUES (888, 'dbms9', 'Suresh', TO_DATE('1999-12-28', 'YYYY-MM-DD'), 'Hisar', 112233)

1 row inserted.

Elapsed: 00:00:00.001

SQL> INSERT INTO voter VALUES (999, 'dbms10', 'Ajinkya', TO_DATE('1999-11-28', 'YYYY-MM-DD'), 'Chennai', 224052)

1 row inserted.

Elapsed: 00:00:00.001

SQL> INSERT INTO voter VALUES (988, 'dbms12', 'Krunal', TO_DATE('1999-06-21', 'YYYY-MM-DD'), 'Pondicherry', 998869)

1 row inserted.

Elapsed: 00:00:00.002
```

4.Insert into candidate

```
164
165    INSERT INTO candidate VALUES (111,'pass1','Vishwas',224052,'mohitverma6161');
166    INSERT INTO candidate VALUES (666,'pass2','Mallya',998869,'aayush123');
167    INSERT INTO candidate VALUES (555,'aloo_se_sona','Pappu',112233,'purushottam123');
168    INSERT INTO candidate VALUES (212,'shaant','Manmphan',224050,'mohitverma6161');
169
```

Output-

Query result | Script output | DBMS output | Explain Plan | SQL history

```
SQL> INSERT INTO candidate VALUES (111,'pass1','Vishwas',224052,'mohitverma6161')


1 row inserted.

Elapsed: 00:00:00.189
```

```
SQL> INSERT INTO candidate VALUES (666,'pass2','Mallya',998869,'aayush123')


1 row inserted.

Elapsed: 00:00:00.001
```

```
SQL> INSERT INTO candidate VALUES (555,'aloo_se_sona','Pappu',112233,'purushottam123')


1 row inserted.

Elapsed: 00:00:00.002
```

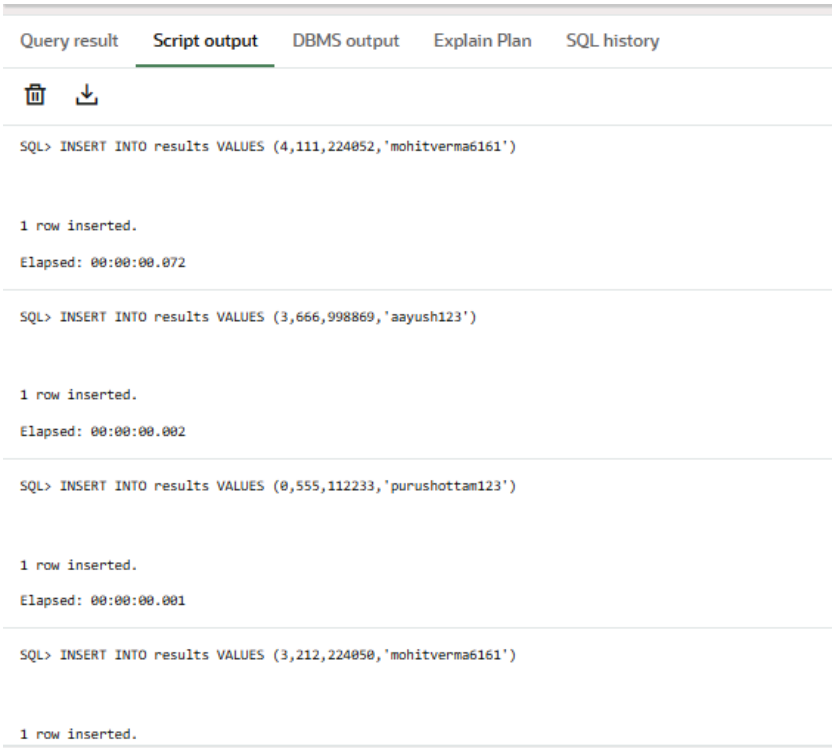
```
SQL> INSERT INTO candidate VALUES (212,'shaant','Manmphan',224050,'mohitverma6161')


1 row inserted.
```

5.Insert into results

```
208
209    INSERT INTO results VALUES (4,111,224052,'mohitverma6161');
210    INSERT INTO results VALUES (3,666,998869,'aayush123');
211    INSERT INTO results VALUES (0,555,112233,'purushottam123');
212    INSERT INTO results VALUES (3,212,224050,'mohitverma6161');
```

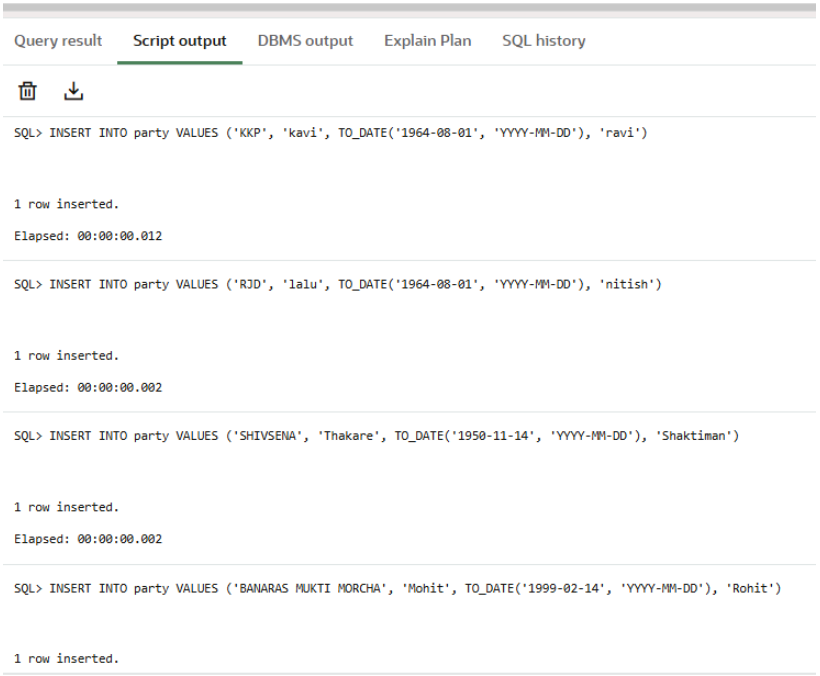
Output-



6.Insert into party

```
54
55    INSERT INTO party VALUES ('KKP', 'kavi', TO_DATE('1964-08-01', 'YYYY-MM-DD'), 'ravi');
56    INSERT INTO party VALUES ('RJD', 'lalu', TO_DATE('1964-08-01', 'YYYY-MM-DD'), 'nitish');
57    INSERT INTO party VALUES ('SHIVSENA', 'Thakare', TO_DATE('1950-11-14', 'YYYY-MM-DD'), 'Shaktiman');
58    INSERT INTO party VALUES ('BANARAS MUKTI MORCHA', 'Mohit', TO_DATE('1999-02-14', 'YYYY-MM-DD'), 'Rohit');
59
```

Output-



7.Insert into members

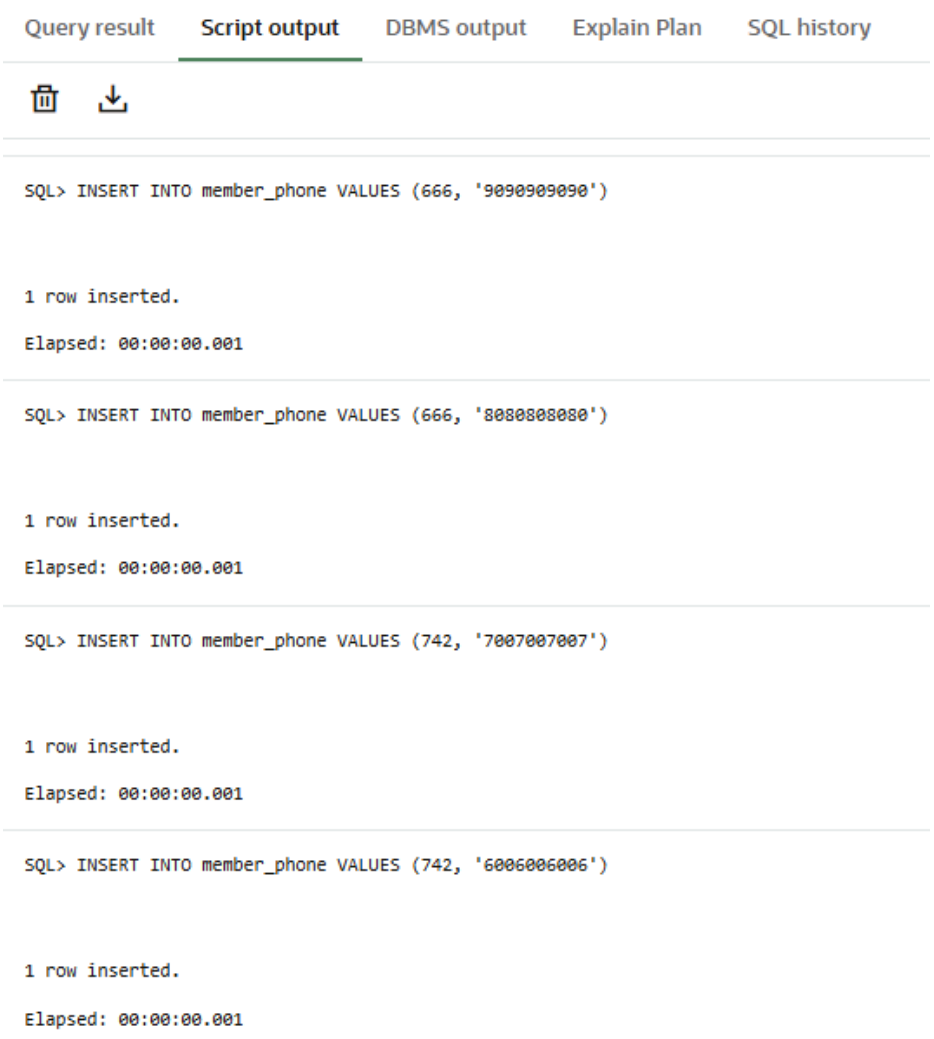
```
164   INSERT INTO members VALUES ('Pappu', 777, 'Ameethi', TO_DATE('1985-11-14', 'YYYY-MM-DD'), 'RJD', 555);
165   INSERT INTO members VALUES ('Raj', 854, 'Faridabad', TO_DATE('1994-03-30', 'YYYY-MM-DD'), 'KKP', 212);
166   INSERT INTO members VALUES ('Gautam', 666, 'Hisar', TO_DATE('1998-10-30', 'YYYY-MM-DD'), 'BANARAS MUKTI MORCHA', 111);
167   INSERT INTO members VALUES ('Vidit', 742, 'Kurushetra', TO_DATE('1998-11-30', 'YYYY-MM-DD'), 'BANARAS MUKTI MORCHA', 111);
```

Output-

8.Insert into member_phone

```
209    INSERT INTO member_phone VALUES (777, '9876543210');
210    INSERT INTO member_phone VALUES (777, '9123456780');
211    INSERT INTO member_phone VALUES (854, '9988776655');
212    INSERT INTO member_phone VALUES (854, '7766554433');
213    INSERT INTO member_phone VALUES (666, '9090909090');
214    INSERT INTO member_phone VALUES (666, '8080808080');
215    INSERT INTO member_phone VALUES (742, '7007007007');
216    INSERT INTO member_phone VALUES (742, '6006006006');
```

Output-



9.Insert into e_ballot

```
169   INSERT INTO e_ballot VALUES (12345, TO_DATE('2018-10-14', 'YYYY-MM-DD'), 'KKP', 112233, 123, 'prateek');
170   INSERT INTO e_ballot VALUES (12346, TO_DATE('2018-10-18', 'YYYY-MM-DD'), 'RJD', 224050, 333, 'mohitverma6161');
171   INSERT INTO e_ballot VALUES (12347, TO_DATE('2018-11-04', 'YYYY-MM-DD'), 'BANARAS MUKTI MORCHA', 998869, 222, 'purushottam123');
172   INSERT INTO e_ballot VALUES (12348, TO_DATE('2018-11-10', 'YYYY-MM-DD'), 'SHIVSENA', 998869, 998, 'mohitverma6161');
173   INSERT INTO e_ballot VALUES (12349, TO_DATE('2018-10-14', 'YYYY-MM-DD'), 'KKP', 112233, 777, 'prateek');
174   INSERT INTO e_ballot VALUES (12350, TO_DATE('2018-10-14', 'YYYY-MM-DD'), 'KKP', 112233, 666, 'mohitverma6161');
175   INSERT INTO e_ballot VALUES (12351, TO_DATE('2018-10-14', 'YYYY-MM-DD'), 'KKP', 112233, 888, 'mohitverma6161');
176   INSERT INTO e_ballot VALUES (12352, TO_DATE('2018-10-14', 'YYYY-MM-DD'), 'KKP', 112233, 988, 'purushottam123');
177   INSERT INTO e_ballot VALUES (12353, TO_DATE('2018-10-14', 'YYYY-MM-DD'), 'BANARAS MUKTI MORCHA', 112233, 111, 'mohitverma6161');
178   INSERT INTO e_ballot VALUES (12354, TO_DATE('2018-10-28', 'YYYY-MM-DD'), 'BANARAS MUKTI MORCHA', 112233, 444, 'mohitverma6161');
179
```

Output-

Query result | Script output | DBMS output | Explain Plan | SQL history

```
SQL> INSERT INTO e_ballot VALUES (12345, TO_DATE('2018-10-14', 'YYYY-MM-DD'), 'KKP', 112233, 123, 'prateek')

1 row inserted.

Elapsed: 00:00:00.063

SQL> INSERT INTO e_ballot VALUES (12346, TO_DATE('2018-10-18', 'YYYY-MM-DD'), 'RJD', 224050, 333, 'mohitverma6161')

1 row inserted.

Elapsed: 00:00:00.001

SQL> INSERT INTO e_ballot VALUES (12347, TO_DATE('2018-11-04', 'YYYY-MM-DD'), 'BANARAS MUKTI MORCHA', 998869, 222, 'purushottam123')

1 row inserted.

Elapsed: 00:00:00.002

SQL> INSERT INTO e_ballot VALUES (12348, TO_DATE('2018-11-10', 'YYYY-MM-DD'), 'SHIVSENA', 998869, 998, 'mohitverma6161')

1 row inserted.

Elapsed: 00:00:00.001

SQL> INSERT INTO e_ballot VALUES (12349, TO_DATE('2018-10-14', 'YYYY-MM-DD'), 'KKP', 112233, 777, 'prateek')

1 row inserted.

Elapsed: 00:00:00.002
```

Query result | Script output | DBMS output | Explain Plan | SQL history

```
SQL> INSERT INTO e_ballot VALUES (12350, TO_DATE('2018-10-14', 'YYYY-MM-DD'), 'KKP', 112233, 666, 'mohitverma6161')

1 row inserted.

Elapsed: 00:00:00.001

SQL> INSERT INTO e_ballot VALUES (12351, TO_DATE('2018-10-14', 'YYYY-MM-DD'), 'KKP', 112233, 888, 'mohitverma6161')

1 row inserted.

Elapsed: 00:00:00.002

SQL> INSERT INTO e_ballot VALUES (12352, TO_DATE('2018-10-14', 'YYYY-MM-DD'), 'KKP', 112233, 988, 'purushottam123')

1 row inserted.

Elapsed: 00:00:00.001

SQL> INSERT INTO e_ballot VALUES (12353, TO_DATE('2018-10-14', 'YYYY-MM-DD'), 'BANARAS MUKTI MORCHA', 112233, 111, 'mohitverma6161')

1 row inserted.

Elapsed: 00:00:00.002

SQL> INSERT INTO e_ballot VALUES (12354, TO_DATE('2018-10-28', 'YYYY-MM-DD'), 'BANARAS MUKTI MORCHA', 112233, 444, 'mohitverma6161')

1 row inserted.

Elapsed: 00:00:00.001
```

10.Insert into keeps_check_on

```
180
181    INSERT INTO keeps_check_on VALUES ('mohitverma6161','RJD');
182    INSERT INTO keeps_check_on VALUES ('mohitverma6161','BANARAS MUKTI MORCHA');
183    INSERT INTO keeps_check_on VALUES ('mohitverma6161','SHIVSENA');
184    INSERT INTO keeps_check_on VALUES ('prateek','KKP');
185    INSERT INTO keeps_check_on VALUES ('aayush123','RJD');
```

Output-

```
SQL> INSERT INTO keeps_check_on VALUES ('mohitverma6161','RJD')


1 row inserted.

Elapsed: 00:00:00.001
```

```
SQL> INSERT INTO keeps_check_on VALUES ('mohitverma6161','BANARAS MUKTI MORCHA')


1 row inserted.

Elapsed: 00:00:00.001
```

```
SQL> INSERT INTO keeps_check_on VALUES ('mohitverma6161','SHIVSENA')


1 row inserted.

Elapsed: 00:00:00.000
```

```
SQL> INSERT INTO keeps_check_on VALUES ('prateek','KKP')


1 row inserted.

Elapsed: 00:00:00.003
```

```
SQL> INSERT INTO keeps_check_on VALUES ('aayush123','RJD')


1 row inserted.

Elapsed: 00:00:00.001
```

11.Insert into manages

```
186
187    INSERT INTO manages VALUES ('mohitverma6161',123);
188    INSERT INTO manages VALUES ('mohitverma6161',111);
189    INSERT INTO manages VALUES ('aayush123',222);
190    INSERT INTO manages VALUES ('purushottam123',333);
191    INSERT INTO manages VALUES ('prateek',123);
192
```

Output-

Query result | Script output | DBMS output | Explain Plan | SQL history

```
SQL> INSERT INTO manages VALUES ('mohitverma6161',123)


1 row inserted.

Elapsed: 00:00:00.007
```

```
SQL> INSERT INTO manages VALUES ('mohitverma6161',111)


1 row inserted.

Elapsed: 00:00:00.001
```

```
SQL> INSERT INTO manages VALUES ('aayush123',222)


1 row inserted.

Elapsed: 00:00:00.002
```

```
SQL> INSERT INTO manages VALUES ('purushottam123',333)


1 row inserted.

Elapsed: 00:00:00.001
```

```
SQL> INSERT INTO manages VALUES ('prateek',123)


1 row inserted.

Elapsed: 00:00:00.001
```

12.Insert into view_profile

```
192
193     INSERT INTO view_profile VALUES (123,666);
194     INSERT INTO view_profile VALUES (123,111);
195     INSERT INTO view_profile VALUES (123,212);
196     INSERT INTO view_profile VALUES (111,212);
197     INSERT INTO view_profile VALUES (222,666);
198     INSERT INTO view_profile VALUES (333,555);
199     INSERT INTO view_profile VALUES (222,555);
200     INSERT INTO view_profile VALUES (111,555);
201     INSERT INTO view_profile VALUES (111,666);
202     INSERT INTO view_profile VALUES (111,111);
```

Output-

Query result | Script output | DBMS output | Explain Plan | SQL history

```
SQL> INSERT INTO view_profile VALUES (333,555)


1 row inserted.

Elapsed: 00:00:00.001
```

```
SQL> INSERT INTO view_profile VALUES (222,555)


1 row inserted.

Elapsed: 00:00:00.001
```

```
SQL> INSERT INTO view_profile VALUES (111,555)


1 row inserted.

Elapsed: 00:00:00.002
```

```
SQL> INSERT INTO view_profile VALUES (111,666)


1 row inserted.

Elapsed: 00:00:00.001
```

```
SQL> INSERT INTO view_profile VALUES (111,111)


1 row inserted.

Elapsed: 00:00:00.002
```

FUNCTIONS AND PROCEDURES

1.Function to get Area Name from given constituency number

```
219    create or replace function cons(cons number) return varchar is
220    a area.area_name%type;
221    begin
222    select area_name into a from area where constituency_number=cons;
223    return a;
224    end;
225    /
226    begin
227    dbms_output.put_line(cons(221133));
228    end;
229    /
```

Query result | Script output | DBMS output | Explain Plan | SQL history

```
SQL> create or replace function cons(cons number) return varchar is
     a area.area_name%type;
     begin
     select area_name into a from area where constituency_number=cons;...
Show more...


Function CONS compiled

Elapsed: 00:00:00.014
```

```
SQL> begin
     dbms_output.put_line(cons(221133));
     end;


varanasi


PL/SQL procedure successfully completed.
```

2. Procedure to get Area Name from candidate id

```
249  create or replace procedure area2(id in number,area out varchar) is
250  begin
251  select area_name into area from area where constituency_number in(select constituency_number from candidate where id_c=id);
252  dbms_output.put_line(area);
253  end;
254  /
255  declare
256  b varchar2(15);
257  begin
258  area2(212,b);
259  end;
260  /
```

Query result | Script output | DBMS output | Explain Plan | SQL history

```
SQL> create or replace procedure area2(id in number,area out varchar) is
     begin
     select area_name into area from area where constituency_number in(select constituency_number from candidate where id_c=id);
     dbms_output.put_line(area); ...
Show more...


Procedure AREA2 compiled

Elapsed: 00:00:00.020
```

```
SQL> declare
     b varchar2(15);
     begin
     area2(212,b); ...
Show more...


ballia


PL/SQL procedure successfully completed.

Elapsed: 00:00:00.009
```

TRIGGERS

Trigger-1

```
124   create trigger trig6
125   after update on area
126   for each row
127   begin
128   update voter set constituency_number=:new.constituency_number where constituency_number=:old.constituency_number;
129   end trig6;
```

Output-

Query result | Script output | DBMS output | Explain Plan

```
SQL> create trigger trig6
     after update on area
     for each row
     begin...
Show more...


Trigger TRIG6 compiled
```

Trigger-1 example with execution

```
131   update area set constituency_number=221133 where area_name='varanasi';
132   select * from voter;
133
```

Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.012 seconds

	ID	PASS	NAME	DOB	ADDRESS	CONSTITUENCY_NU
1	123	dbms1	Adarsh	8/15/1998, 12:00:00 AM	Bhubhaneshwar	112233
2	111	dbms2	Siddharth	1/26/1997, 12:00:00 AM	Delhi	998869
3	222	dbms3	Sankalp	10/2/1996, 12:00:00 AM	Ghaziabad	998869
4	333	dbms4	Deepak	11/14/1996, 12:00:00 AM	Bengaluru	224050
5	444	dbms5	Yuvi	10/28/1995, 12:00:00 AM	Ballia	224050
6	555	dbms6	Dhoni	12/28/1995, 12:00:00 AM	Ranchi	224050
7	666	dbms7	Shikhar	3/11/1993, 12:00:00 AM	Lucknow	224050
8	777	dbms8	Rohit	1/10/1993, 12:00:00 AM	Ludhiana	998869
9	888	dbms9	Suresh	12/28/1999, 12:00:00 AM	Hisar	112233
10	999	dbms10	Ajinkya	11/28/1999, 12:00:00 AM	Chennai	224052
11	998	dbms11	Pandya	6/21/1999, 12:00:00 AM	Chennai	224052
12	988	dbms12	Krunal	6/21/1999, 12:00:00 AM	Pondicherry	998869

Trigger-2 with Output

```
96    CREATE OR REPLACE TRIGGER trg_default_vote_date
97    BEFORE INSERT ON e_ballot
98    FOR EACH ROW
99    BEGIN
100     IF :NEW.date_of_vote IS NULL THEN
101       :NEW.date_of_vote := SYSDATE;
102     END IF;
103   END;
104   /
```

Query result | Script output | DBMS output | Explain Plan

```
SQL> CREATE OR REPLACE TRIGGER trg_default_vote_date
     BEFORE INSERT ON e_ballot
     FOR EACH ROW
     BEGIN...
Show more...


Trigger TRG_DEFAULT_VOTE_DATE compiled
```

Trigger-3 with Output

```
105
106   CREATE OR REPLACE TRIGGER trg_prevent_duplicate_vote
107   BEFORE INSERT ON e_ballot
108   FOR EACH ROW
109   DECLARE
110     v_count INTEGER;
111   BEGIN
112     SELECT COUNT(*) INTO v_count
113     FROM e_ballot
114     WHERE id = :NEW.id
115       AND constituency_number = :NEW.constituency_number
116       AND date_of_vote = :NEW.date_of_vote;
117
118     IF v_count > 0 THEN
119       RAISE_APPLICATION_ERROR(-20001, 'Duplicate vote detected:voter cannot vote twice in the election.');
120     END IF;
121   END;
122   /
123
```

Query result | Script output | DBMS output | Explain Plan | SQL history

```
Trigger TRG_PREVENT_DUPLICATE_VOTE compiled

Elapsed: 00:00:00.022
```

SQL QUERIES

Q)Find the number of votes for a candidate and his respective party details?

```
261  --FIND THE NUMBER OF VOTES FOR A CANDIDATE AND HIS RESPECTIVE PARTY DETAILS
262  select r.number_of_votes,c.id_c,m.party_name from results r,candidate c,members m where r.id_c=c.id_c and c.id_c=m.id_c;
```

OUTPUT-

Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.012 seconds

	NUMBER_OF_VOTES	ID_C	PARTY_NAME
1	0	555	RJD
2	3	212	KKP
3	4	111	BANARAS MUKTI MORCHA
4	4	111	BANARAS MUKTI MORCHA

Results

```
264
265    SELECT * FROM admin;
266
```

Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.009 seconds

	USERNAME	PASSWORD
1	mohitverma6161	mohit123
2	aayush123	canadanoor
3	purushottam123	gangleader
4	prateek	coder

```
268    SELECT * FROM party;
269
```

Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.012 seconds

	PARTY_NAME	FOUNDER	DATE_OF_FOUND	PARTY_CHIEF
1	KKP	kavi	8/1/1964, 12:00:00 AM	ravi
2	RJD	lalu	8/1/1964, 12:00:00 AM	nitish
3	SHIVSENA	Thakare	11/14/1950, 12:00:00 AM	Shaktiman
4	BANARAS MUKTI MORCHA	Mohit	2/14/1999, 12:00:00 AM	Rohit

```
271  SELECT * FROM members;
272
```

Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.009 seconds

	NAME_M	ID_M	ADDRESS_M	DOB_M	PARTY_NAME	ID_C
1	Pappu	777	Ameethi	11/14/1985, 12:00:(	RJD	555
2	Raj	854	Faridabad	3/30/1994, 12:00:0(	KKP	212
3	Gautam	666	Hisar	10/30/1998, 12:00:(	BANARAS MUKTI M(	111
4	Vidit	742	Kurushetra	11/30/1998, 12:00:(	BANARAS MUKTI M(	111

```
274  SELECT * FROM candidate;
275
```

Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.01 seconds

	ID_C	PASSWORD_C	NAME_C	CONSTITUENCY_NU	USERNAME
1	111	pass1	Vishwas	224052	mohitverma6161
2	666	pass2	Mallya	998869	aayush123
3	555	aloo_se_sona	Pappu	112233	purushottam123
4	212	shaant	Manmphan	224050	mohitverma6161

```
277  SELECT * FROM results;
```

Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.01 seconds

	ID_C	PASSWORD_C	NAME_C	CONSTITUENCY_NU	USERNAME
1	111	pass1	Vishwas	224052	mohitverma6161
2	666	pass2	Mallya	998869	aayush123
3	555	aloo_se_sona	Pappu	112233	purushottam123
4	212	shaant	Manmphan	224050	mohitverma6161

```
280  SELECT * FROM area;
281
```

Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.006 seconds

	AREA_NAME	CONSTITUENCY_NUMBER
1	gandhinagar	112233
2	varanasi	221133
3	yelagiri	998869
4	ballia	224050
5	ballia	224052

```
281
282  SELECT * FROM e_ballot;
```

Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.01 seconds

	S_NO	DATE_OF_VOTE	PARTY_NAME	CONSTITUENCY_NUMBER	ID	USERNAME
1	12345	10/14/2018, 12:00:00 AM	KKP	112233	123	prateek
2	12346	10/18/2018, 12:00:00 AM	RJD	224050	333	mohitverma6161
3	12347	11/4/2018, 12:00:00 AM	BANARAS MUKTI MO	998869	222	purushottam123
4	12348	11/10/2018, 12:00:00 AM	SHIVSENA	998869	998	mohitverma6161
5	12349	10/14/2018, 12:00:00 AM	KKP	112233	777	prateek
6	12350	10/14/2018, 12:00:00 AM	KKP	112233	666	mohitverma6161
7	12351	10/14/2018, 12:00:00 AM	KKP	112233	888	mohitverma6161
8	12352	10/14/2018, 12:00:00 AM	KKP	112233	988	purushottam123
9	12353	10/14/2018, 12:00:00 AM	BANARAS MUKTI MO	112233	111	mohitverma6161
10	12354	10/28/2018, 12:00:00 AM	BANARAS MUKTI MO	112233	444	mohitverma6161

```
284
285  SELECT * FROM voter;
286
```

Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.008 seconds

	ID	PASS	NAME	DOB	ADDRESS	CONSTITUENCY_NU
1	123	dbms1	Adarsh	8/15/1998, 12:00:0(	Bhubhaneshwar	112233
2	111	dbms2	Siddharth	1/26/1997, 12:00:0(	Delhi	998869
3	222	dbms3	Sankalp	10/2/1996, 12:00:0(	Ghaziabad	998869
4	333	dbms4	Deepak	11/14/1996, 12:00:(	Bengaluru	224050
5	444	dbms5	Yuvi	10/28/1995, 12:00:(	Ballia	224050
6	555	dbms6	Dhoni	12/28/1995, 12:00:(	Ranchi	224050
7	666	dbms7	Shikhar	3/11/1993, 12:00:0(	Lucknow	224050
8	777	dbms8	Rohit	1/10/1993, 12:00:0(	Ludhiana	998869
9	888	dbms9	Suresh	12/28/1999, 12:00:(	Hisar	112233
10	999	dbms10	Ajinkya	11/28/1999, 12:00:(	Chennai	224052
11	998	dbms11	Pandya	6/21/1999, 12:00:0(	Chennai	224052
12	988	dbms12	Krunal	6/21/1999, 12:00:0(	Pondicherry	998869

```
287
288  SELECT * FROM keeps_check_on;
289
```

Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.007 seconds

	USERNAME	PARTY_NAME
1	aayush123	RJD
2	mohitverma6161	BANARAS MUKTI M(
3	mohitverma6161	RJD
4	mohitverma6161	SHIVSENA
5	prateek	KKP

```
291  SELECT * FROM view_profile;
292
```

Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.008 seconds

	ID	ID_C
1	111	111
2	111	212
3	111	555
4	111	666
5	123	111
6	123	212
7	123	666
8	222	555
9	222	666
10	333	555

```
293
294  SELECT * FROM manages;
295
```

Query result | Script output | DBMS output | Explain Plan | SQL

Download | Execution time: 0.008 seconds

	USERNAME	ID
1	aayush123	222
2	mohitverma6161	111
3	mohitverma6161	123
4	prateek	123
5	purushottam123	333

```
296
297  SELECT * FROM member_phone;
298
```

Query result | Script output | DBMS output | Explain Plan | S

Download | Execution time: 0.009 seconds

	ID_M	PHONE_NO
1	666	8080808080
2	666	9090909090
3	742	6006006006
4	742	7007007007
5	777	9123456780
6	777	9876543210
7	854	7766554433
8	854	9988776655

Conclusion

The implementation of the Online Voting System using Database Management principles highlights how structured data storage, relational mapping, and secure access mechanisms can bring significant efficiency and reliability to a highly sensitive process like elections.

By leveraging MySQL and PL/SQL, this system ensures data integrity, scalability, and user authentication — essential factors for a secure and fair election. From ER modeling to normalization and SQL procedures, each layer of this system demonstrates how database technology can be practically applied to build real-world solutions.

Overall, this project not only reinforces DBMS concepts but also illustrates their impactful application in promoting digital governance and public trust through transparent and accessible voting mechanisms