

# **DATABASE SYSTEMS**

## **PROJECT REPORT**



**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE,  
PILANI (Rajasthan)**

(April 11th 2023)

**By:**

**Samraddh Saxena : 2021A7PS1455P**

**Aditya Patel : 2021A7PS2433P**

**Project Number: 13**

**Project Title: Voting Management System**

# Contribution Table

Name	Contribution
Samraddh Saxena	Frontend, Queries, backend and Normalisation
Aditya Patel	Documentation, Backend, Queries and ER Diagram

## ONLINE LINKS TO ALL VIDEOS:

[https://drive.google.com/drive/folders/10bSdeprBSJS4QRHFvvJs6e7Q9yRlh6GC  
?usp=sharing](https://drive.google.com/drive/folders/10bSdeprBSJS4QRHFvvJs6e7Q9yRlh6GC?usp=sharing)

This folder contains the following videos:

- a) Showing the functioning of the website and project
- b) Troubleshooting Video for rare errors
- c) One video per member explaining ER Diagram, Normalisation etc.

# Interpretation of Problem Statement:

The problem statement is as follows:

“The Voting Management System project aims to develop an online platform for managing elections. The system will be managed by an administrator who will be responsible for setting up the election, adding candidates, and ensuring overall system security. The system will allow users to register and cast their votes for their preferred candidates.

Users will have to register with their personal information such as name, address, and email in order to be able to cast their vote. Once registered, users will be able to log in to the system and view the candidates they can vote for. They will be able to cast their vote for their preferred candidate.

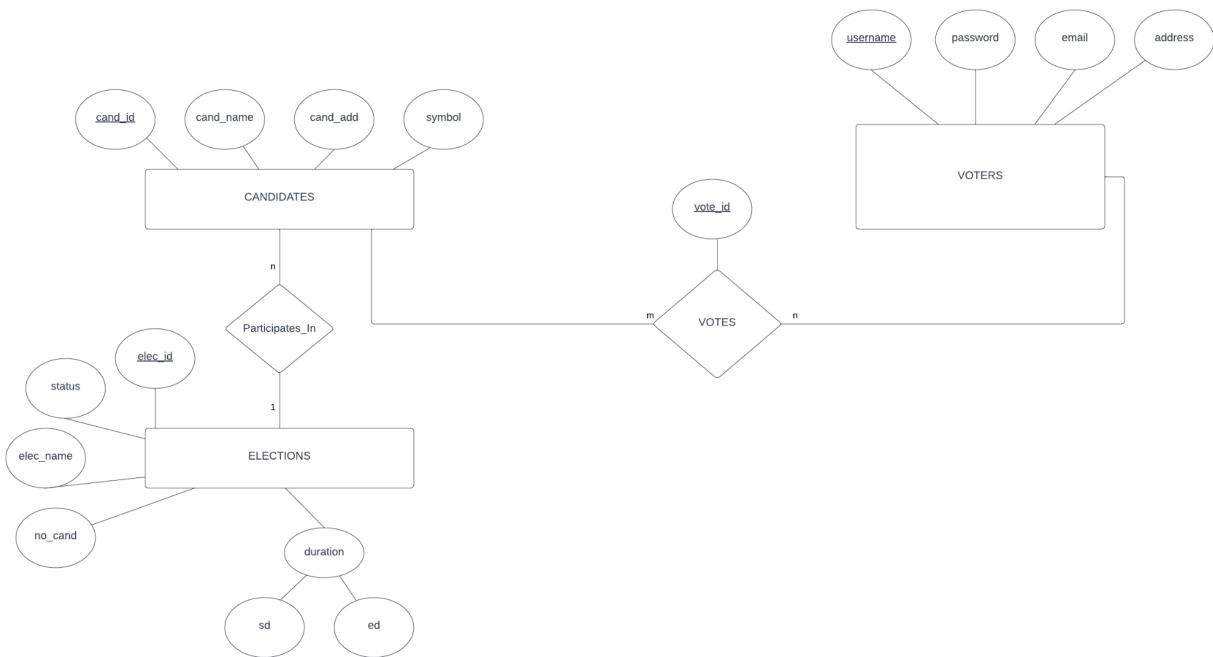
The administrator will be able to generate reports showing the total number of votes cast, the number of votes received by each candidate, the percentage of votes received by each candidate, the winner of the election, and the rank of each candidate based on the number of votes received.”

In approaching this problem statement we first undertook the following assumptions while designing our program:

- a) Each candidate can participate in only one election at a time.
- b) Each voter is eligible to vote for any candidate participating in any election.

We created the ER diagram to visualise the problem. Further we normalised our model to 3NF. We converted this ER diagram into a relational schema. Further in implementation of our project we made our backend by using php, and designed the website by using HTML, CSS and JavaScript.

# ER DIAGRAM:



## ALTERNATE DRIVE LINK:

[https://drive.google.com/drive/folders/1HUHnW\\_tcO6SCUCb8MmuGwkOf\\_NYRdFCO?usp=sharing](https://drive.google.com/drive/folders/1HUHnW_tcO6SCUCb8MmuGwkOf_NYRdFCO?usp=sharing)

ER Diagram consists of three entities, namely CANDIDATES, VOTERS AND ELECTIONS. There exist two relations one between CANDIDATES and ELECTIONS called Participates\_in, the other being between candidates and voters called votes. The relations present have cardinality of N:1.

## **RELATIONAL SCHEMA AND CONVERSION**

### **TO RELATIONAL SCHEMA :**

#### **INITIAL RELATIONAL SCHEMA:**

CANDIDATES					
Cand_id	cand_name	cand_add	symbol		
FOREIGN KEYS: NONE    CANDIDATE KEYS: cand_id					
ELECTIONS					
elec_id	status	elec_name	no_cand	sd	ed
FOREIGN KEYS: NONE    CANDIDATE KEYS: elec_id					
VOTERS					
username	password	email	address		
FOREIGN KEYS: NONE    CANDIDATE KEYS: username					
VOTES					
vote_id	cand_id	username			
FOREIGN KEYS: cand_id, username    CANDIDATE KEYS: vote_id					
participates_in					
cand_id	elec_id				
FOREIGN KEYS: cand_id, elec_id    CANDIDATE KEYS: cand_id					

Following constraints are presents:

- Cand\_id,elec\_id,username,vote\_id are all PRIMARY KEYS
- Cand\_id and username act as Foreign Keys

## **MOVING REDUNDANT ENTRIES:**

To simplify the database structure we remove the participates\_in relation by adding elec\_id (the primary key from elections) to candidates entity as an attribute allowing us to remove the relation and simplify our database.

## **UPDATED RELATIONAL SCHEMA:**

CANDIDATES					
cand_id	cand_name	cand_add	symbol	elec_id	
FOREIGN KEYS: elec_id    CANDIDATE KEYS: cand_id					
ELECTIONS					
elec_id	status	elec_name	no_cand	sd	ed
FOREIGN KEYS: NONE    CANDIDATE KEYS: elec_id					
VOTERS					
username	password	email	address		
FOREIGN KEYS: NONE    CANDIDATE KEYS: username					
VOTES					
vote_id	cand_id	username			
FOREIGN KEYS: cand_id,username    CANDIDATE KEYS: vote_id					

Updated Constraints: elec\_id acts as a FOREIGN KEY.

# **FUNCTIONAL DEPENDENCIES AND NORMALISATION:**

## **FUNCTIONAL DEPENDENCIES:**

### **Candidates:**

cand\_id  $\rightarrow$  cand\_name  
cand\_id  $\rightarrow$  cand\_add  
cand\_id  $\rightarrow$  symbol  
cand\_id  $\rightarrow$  elec\_id

### **Elections:**

elec\_id  $\rightarrow$  status  
elec\_id  $\rightarrow$  elec\_name  
elec\_id  $\rightarrow$  no\_cand  
elec\_id  $\rightarrow$  sd  
elec\_id  $\rightarrow$  ed

### **Voters:**

username  $\rightarrow$  password  
username  $\rightarrow$  email  
username  $\rightarrow$  address

### **Votes:**

vote\_id  $\rightarrow$  cand\_id  
vote\_id  $\rightarrow$  username

## **NORMALISATION TO 3NF:**

### **1NF:**

Our initial ER diagram shows that the duration attribute of the elections entity is a composite attribute with two sub-attributes sd (start date) and ed (end date). This makes our ER not in 1NF. To normalise our database to 1NF we simply add both sub-attributes as part of the elections entity.

### **2NF and 3NF:**

Upon analysis of all the functional dependencies listed above we come to the following conclusions:

While checking for 2NF, we observe:

- a) It is present in 1NF
- b) There exist no partial dependencies, i.e. no proper subset of candidate keys can determine non prime attributes.

**Hence we see that our schema is already in 2NF.**

Further while checking for 3NF form we observe:

- a) It is present in 2NF
- b) There exist no transitive dependencies, i.e no non-prime attribute determines another non-prime attribute.

**Hence we conclude that our schema is in 3NF.**

## **QUERIES TO INSERT DATA IN RELEVANT TABLES:**

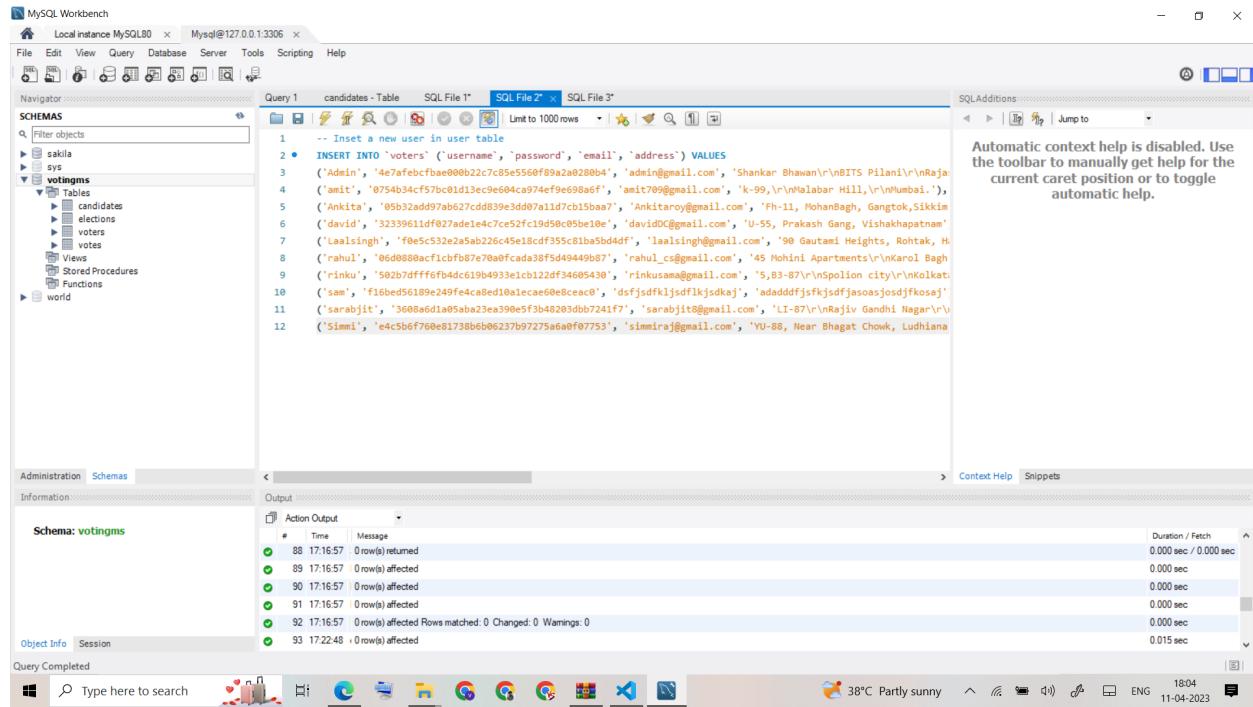
### **INSERT USER IN NEW USER TABLE:**

-- Insert a new user in user table

```
INSERT INTO `voters` (`username`, `password`, `email`, `address`) VALUES  
('Admin', '4e7afebcfbæ000b22c7c85e5560f89a2a0280b4', 'admin@gmail.com', 'Shankar  
Bhawan\r\nBITS Pilani\r\nRajasthan'),  
('amit', '0754b34cf57bc01d13ec9e604ca974ef9e698a6f', 'amit709@gmail.com',  
'k-99,\r\nMalabar Hill,\r\nMumbai.'),  
('Ankita', '05b32add97ab627cdd839e3dd07a11d7cb15baa7', 'Ankitaroy@gmail.com', 'Fh-11,  
MohanBagh, Gangtok,Sikkim.'),
```

('david', '32339611df027ade1e4c7ce52fc19d50c05be10e', 'davidDC@gmail.com', 'U-55,  
 Prakash Gang, Vishakhapatnam'),  
 ('Laalsingh', 'f0e5c532e2a5ab226c45e18cdf355c81ba5bd4df', 'laalsingh@gmail.com', '90  
 Gautami Heights, Rohtak, Haryana'),  
 ('rahul', '06d0880acf1cbfb87e70a0fcada38f5d49449b87', 'rahul\_cs@gmail.com', '45 Mohini  
 Apartments\r\nKarol Bagh \r\nDelhi'),  
 ('rinku', '502b7dff6fb4dc619b4933e1cb122df34605430', 'rinkusama@gmail.com',  
 '5,B3-87\r\nSpolian city\r\nKolkata'),  
 ('sam', 'f16bed56189e249fe4ca8ed10a1ecae60e8ceac0', 'dsfjsdkljsdflkjsdkaj',  
 'adadddfjsfkjsdfjasoasjosdfjkosaj'),  
 ('sarabjit', '3608a6d1a05aba23ea390e5f3b48203dbb7241f7', 'sarabjit8@gmail.com',  
 'LI-87\r\nRajiv Gandhi Nagar\r\nAmritsar'),  
 ('Simmi', 'e4c5b6f760e81738b6b06237b97275a6a0f07753', 'simmiraj@gmail.com', 'YU-88,  
 Near Bhagat Chowk, Ludhiana');

## OUTPUT:

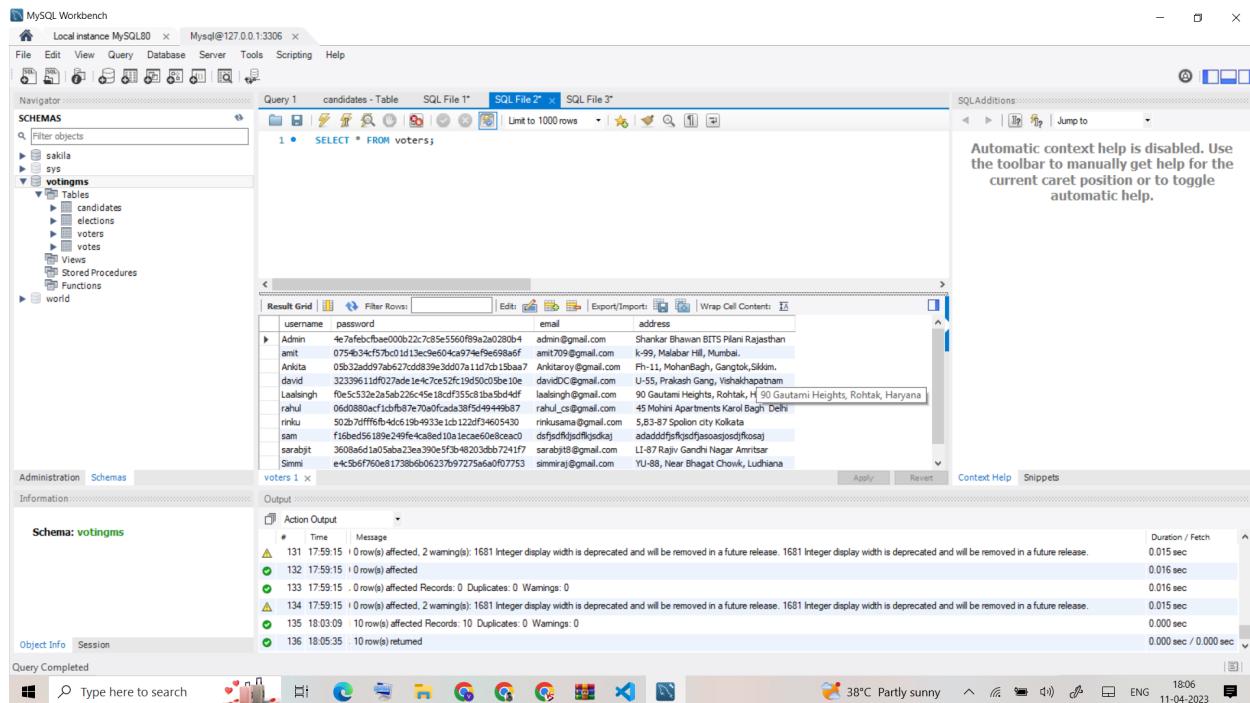


The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbar:** Standard database management icons.
- Navigator:** Shows the schema 'votingms' containing tables like 'candidates', 'elections', 'voters', and 'votes'.
- Query Editor:** Contains the following SQL code:
 

```

1 -- Insert a new user in user table
2 • INSERT INTO `voters` (`username`, `password`, `email`, `address`) VALUES
3 ('Admin', '4e7afebcfae00b22c7c85e5560ff99a2e0280b1', 'admin@gmail.com', 'Shankar Bhawan\r\nBITS Pilani\r\nRajasthan')
4 ('amit', '0754b34cf57bc01d13ec9e04ca074ef9e698a6f', 'amit709@gmail.com', 'k-99,\r\nMalabar Hill,\r\nMumbai'), 
5 ('Ankit', '05b52add97ab627cd5393d3d97a11d7c15baa7', 'Ankitaroy@gmail.com', 'H-11, MohanBagh, Gangtok,Sikkim')
6 ('david', '32339611df027ade1e4c7ce52fc19d50c05be10e', 'david@gmail.com', 'U-55,\r
        Prakash Gang, Vishakhapatnam')
7 ('Laalsingh', 'f0e5c532e2a5ab226c45e18cdf355c81ba5bd4df', 'laalsingh@gmail.com', 'Gautami Heights, Rohtak, Haryana')
8 ('rahul', '06d0880acf1cbfb87e70a0fcada38f5d49449b87', 'rahul_cs@gmail.com', '45 Mohini Apartments\r
        Karol Bagh New Delhi')
9 ('rinku', '502b7dff6fb4dc619b4933e1cb122df34605430', 'rinkusama@gmail.com', '5,B3-87\r
        Spolian city\r
        Kolkata')
10 ('sam', 'f16bed56189e249fe4ca8ed10a1ecae60e8ceac0', 'dsfjsdkljsdflkjsdkaj', 'adadddfjsfkjsdfjasoasjosdfjkosaj')
11 ('sarabjit', '3608a6d1a05aba23ea390e5f3b48203dbb7241f7', 'sarabjit8@gmail.com', 'LI-87\r
        Rajiv Gandhi Nagar\r
        Amritsar')
12 ('Simmi', 'e4c5b6f760e81738b6b06237b97275a6a0f07753', 'simmiraj@gmail.com', 'YU-88, Near Bhagat Chowk, Ludhiana')
            
```
- Output Window:** Shows the execution results with 0 rows affected for each insert statement, and a final row with 0 rows affected and 0.015 sec duration.
- System Bar:** Shows the system tray with icons for battery, signal, volume, and date/time (18:04, 11-04-2023).

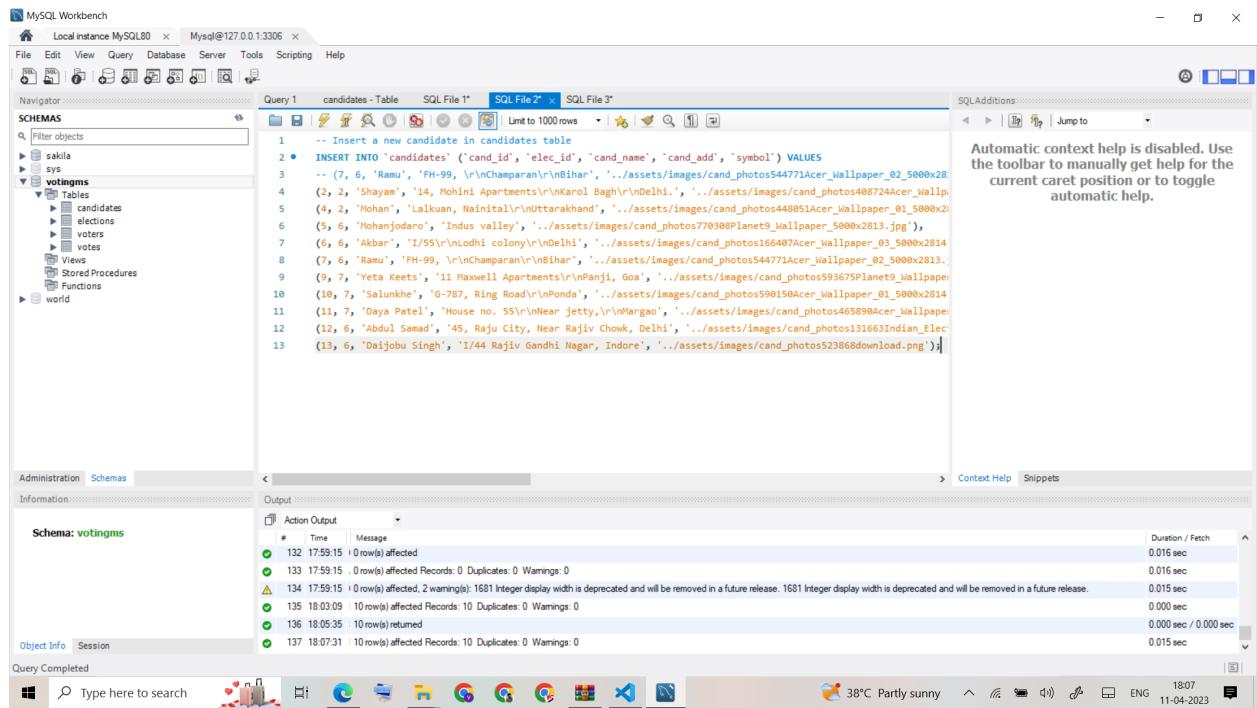


## INPUT OF 10 CANDIDATES IN TABLE:

```
INSERT INTO `candidates` (`cand_id`, `elec_id`, `cand_name`, `cand_add`, `symbol`)
VALUES
-- (7, 6, 'Ramu', 'FH-99, \r\nChamparan\r\nBihar',
'./assets/images/cand_photos544771Acer_Wallpaper_02_5000x2813.jpg');
(2, 2, 'Shayam', '14, Mohini Apartments\r\nKarol Bagh\r\nDelhi.',
'./assets/images/cand_photos408724Acer_Wallpaper_02_5000x2813.jpg'),
(4, 2, 'Mohan', 'Lalkuan, Nainital\r\nUttarakhand',
'./assets/images/cand_photos448051Acer_Wallpaper_01_5000x2814.jpg'),
(5, 6, 'Mohanjodaro', 'Indus valley',
'./assets/images/cand_photos770308Planet9_Wallpaper_5000x2813.jpg'),
(6, 6, 'Akbar', 'I/55\r\nLodhi colony\r\nDelhi',
'./assets/images/cand_photos166407Acer_Wallpaper_03_5000x2814.jpg'),
(7, 6, 'Ramu', 'FH-99, \r\nChamparan\r\nBihar',
'./assets/images/cand_photos544771Acer_Wallpaper_02_5000x2813.jpg'),
(9, 7, 'Yeta Keets', '11 Maxwell Apartments\r\nPanji, Goa',
'./assets/images/cand_photos593675Planet9_Wallpaper_5000x2813.jpg'),
```

```
(10, 7, 'Salunkhe', 'G-787, Ring Road\nPonda',
'./assets/images/cand_photos590150Acer_Wallpaper_01_5000x2814.jpg'),
(11, 7, 'Daya Patel', 'House no. 55\nNear jetty,\nMargao',
'./assets/images/cand_photos465890Acer_Wallpaper_02_5000x2813.jpg'),
(12, 6, 'Abdul Samad', '45, Raju City, Near Rajiv Chowk, Delhi',
'./assets/images/cand_photos131663Indian_Election_Symbol_Telephone.png'),
(13, 6, 'Daijobu Singh', 'I/44 Rajiv Gandhi Nagar, Indore',
'./assets/images/cand_photos523868download.png');
```

## OUTPUT:



The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Local instance MySQL80, Mysql@127.0.0.1:3306.
- Navigator:** Shows the schema structure under the votingms database, including tables like candidates, elections, voters, and votes.
- Query Editor:** Contains the SQL script used to insert candidates. The script includes comments and values for each candidate's name, election ID, candidate name, address, and symbol.
- Output Window:** Shows the execution results, including 132 rows affected (records inserted), 0 warnings, and 0 errors. It also lists 137 rows returned from the last query.
- System Bar:** Includes a search bar, system icons, and a status bar indicating the date and time (11-04-2023, 18:07).

```

-- Insert a new candidate in candidates table
INSERT INTO `candidates` (`cand_id`, `elec_id`, `cand_name`, `cand_addr`, `symbol`) VALUES
-- (7, 6, 'Ramu', 'FH-99, \nChampanar\nBihar', './assets/images/cand_photos544771Acer_Wallpaper_02_5000x2814.jpg'),
-- (2, 2, 'Shyam', '14, Mohini Apartments\nKarol Bagh\nDelhi', './assets/images/cand_photos448051Acer_Wallpaper_01_5000x2813.jpg'),
-- (4, 6, 'MohanJodaro', 'Indus valley', './assets/images/cand_photos770388Planet9_Wallpaper_5000x2814.jpg'),
-- (5, 6, 'Akbar', 'I/55\nLodhi colony\nDelhi', './assets/images/cand_photos166487Acer_Wallpaper_03_5000x2814.jpg'),
-- (7, 6, 'Ramu', 'FH-99, \nChampanar\nBihar', './assets/images/cand_photos544771Acer_Wallpaper_02_5000x2813.jpg'),
-- (9, 7, 'Yeta Keets', '11 Maxwell Apartments\nPanji, Goa', './assets/images/cand_photos936759Planet9_Wallpaper_01_5000x2814.jpg'),
-- (10, 7, 'Salunkhe', 'G-787, Ring Road\nPonda', './assets/images/cand_photos590150Acer_Wallpaper_01_5000x2814.jpg'),
-- (11, 7, 'Daya Patel', 'House no. 55\nNear jetty,\nMargao', './assets/images/cand_photos465890Acer_Wallpaper_02_5000x2813.jpg'),
-- (12, 6, 'Abdul Samad', '45, Raju City, Near Rajiv Chowk, Delhi', './assets/images/cand_photos131663Indian_Election_Symbol_Telephone.png'),
-- (13, 6, 'Daijobu Singh', 'I/44 Rajiv Gandhi Nagar, Indore', './assets/images/cand_photos523868download.png')
```

The screenshot shows the MySQL Workbench interface. In the top navigation bar, there are tabs for 'File', 'Edit', 'View', 'Query', 'Database', 'Server', 'Tools', and 'Help'. The 'Query' tab is active. Below the navigation bar, the 'Schemas' section shows the 'votingms' schema selected. Under 'Tables', there are two tables: 'candidates' and 'votes'. The 'candidates' table has 13 rows of data, which are displayed in a result grid. The columns are 'cand\_id', 'elec\_id', 'cand\_name', 'cand\_addr', and 'symbol'. The 'votes' table also has 13 rows of data, which are not visible in the current view. On the right side of the interface, there is a 'SQLAdditions' panel with a message about automatic context help being disabled. The bottom of the screen shows the Windows taskbar with various pinned icons and the system tray.

## INPUT OF 10 VOTES IN TABLE:

```
INSERT INTO `votes` (`vote_id`, `cand_id`, `username`) VALUES
(3, 2, 'rahul'),
(4, 2, 'sarabjit'),
(6, 2, 'amit'),
(9, 5, 'rahul'),
(11, 7, 'sarabjit'),
(12, 7, 'amit'),
(13, 2, 'sam'),
(15, 7, 'rinku'),
(16, 7, 'sam'),
(17, 9, 'rahul'),
(19, 11, 'sam');
```

## OUTPUT:

**MySQL Workbench**

Local instance MySQL80 Mysql@127.0.0.1:3306

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas

SCHEMAS
 

- sakila
- sys
- votingsms
  - Tables
    - candidates
    - elections
    - voters
    - votes
  - Views
  - Stored Procedures
  - Functions
- world

Query 1 candidates - Table SQL File 1\* SQL File 2\* SQL File 3\*

```

1 INSERT INTO `votes` (`vote_id`, `cand_id`, `username`) VALUES
2 (3, 2, 'rahul'),
3 (4, 2, 'sarabjit'),
4 (6, 2, 'amit'),
5 (9, 5, 'rahul'),
6 (11, 7, 'sarabjit'),
7 (12, 7, 'amit'),
8 (13, 2, 'sam'),
9 (15, 7, 'rinku'),
10 (16, 7, 'sam'),
11 (17, 9, 'rahul'),
12 (19, 11, 'sam')
13

```

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Administration Schemas

Information Output

Action Output

#	Time	Message	Duration / Fetch
134	17:59:15	0 row(s) affected, 2 warning(s): 1681 Integer display width is deprecated and will be removed in a future release. 1681 Integer display width is deprecated and will be removed in a future release.	0.015 sec
135	18:03:09	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
136	18:05:35	10 row(s) returned	0.000 sec / 0.000 sec
137	18:07:31	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.015 sec
138	18:10:31	10 row(s) returned	0.000 sec / 0.000 sec
139	18:11:55	11 row(s) affected Records: 11 Duplicates: 0 Warnings: 0	0.000 sec

Object Info Session

Query Completed

Type here to search 38°C Partly sunny 18:11 ENG 11-04-2023

**MySQL Workbench**

Local instance MySQL80 Mysql@127.0.0.1:3306

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas

SCHEMAS
 

- sakila
- sys
- votingsms
  - Tables
    - candidates
    - elections
    - voters
    - votes
  - Views
  - Stored Procedures
  - Functions
- world

Query 1 candidates - Table SQL File 1\* SQL File 2\* SQL File 3\*

```

1 * select * from votes;
2

```

Result Grid

vote_id	cand_id	username
3	2	rahul
4	2	sarabjit
6	2	amit
9	5	rahul
11	7	sarabjit
12	7	amit
13	2	sam
15	7	rinku
16	7	sam
17	9	rahul
19	11	sam

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Administration Schemas

Information Output

Action Output

#	Time	Message	Duration / Fetch
135	18:03:09	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
136	18:05:35	10 row(s) returned	0.000 sec / 0.000 sec
137	18:07:31	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.015 sec
138	18:10:31	10 row(s) returned	0.000 sec / 0.000 sec
139	18:11:55	11 row(s) affected Records: 11 Duplicates: 0 Warnings: 0	0.000 sec
140	18:12:27	: 11 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

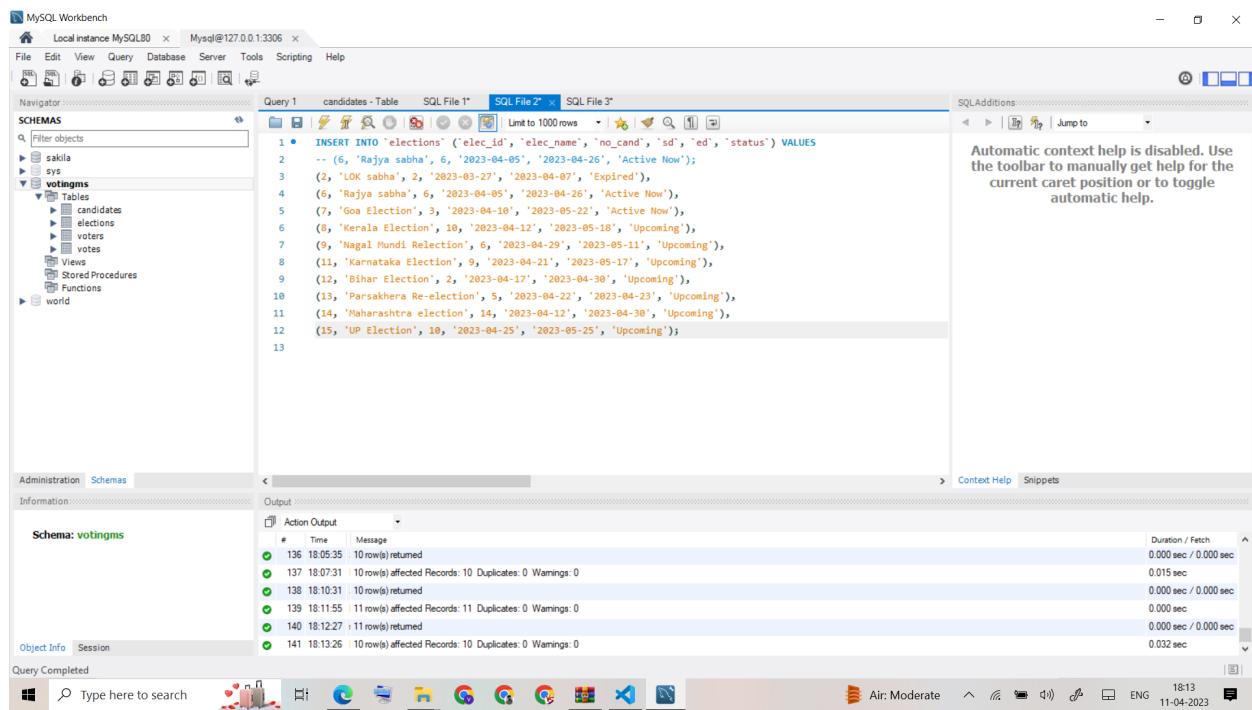
Query Completed

Type here to search 38°C Partly sunny 18:12 ENG 11-04-2023

## INPUT OF 10 ELECTIONS IN TABLE:

```
INSERT INTO `elections` (`elec_id`, `elec_name`, `no_cand`, `sd`, `ed`, `status`) VALUES
-- (6, 'Rajya sabha', 6, '2023-04-05', '2023-04-26', 'Active Now'),
(2, 'LOK sabha', 2, '2023-03-27', '2023-04-07', 'Expired'),
(6, 'Rajya sabha', 6, '2023-04-05', '2023-04-26', 'Active Now'),
(7, 'Goa Election', 3, '2023-04-10', '2023-05-22', 'Active Now'),
(8, 'Kerala Election', 10, '2023-04-12', '2023-05-18', 'Upcoming'),
(9, 'Nagal Mundi Relection', 6, '2023-04-29', '2023-05-11', 'Upcoming'),
(11, 'Karnataka Election', 9, '2023-04-21', '2023-05-17', 'Upcoming'),
(12, 'Bihar Election', 2, '2023-04-17', '2023-04-30', 'Upcoming'),
(13, 'Parsakhera Re-election', 5, '2023-04-22', '2023-04-23', 'Upcoming'),
(14, 'Maharashtra election', 14, '2023-04-12', '2023-04-30', 'Upcoming'),
(15, 'UP Election', 10, '2023-04-25', '2023-05-25', 'Upcoming');
```

## OUTPUT:

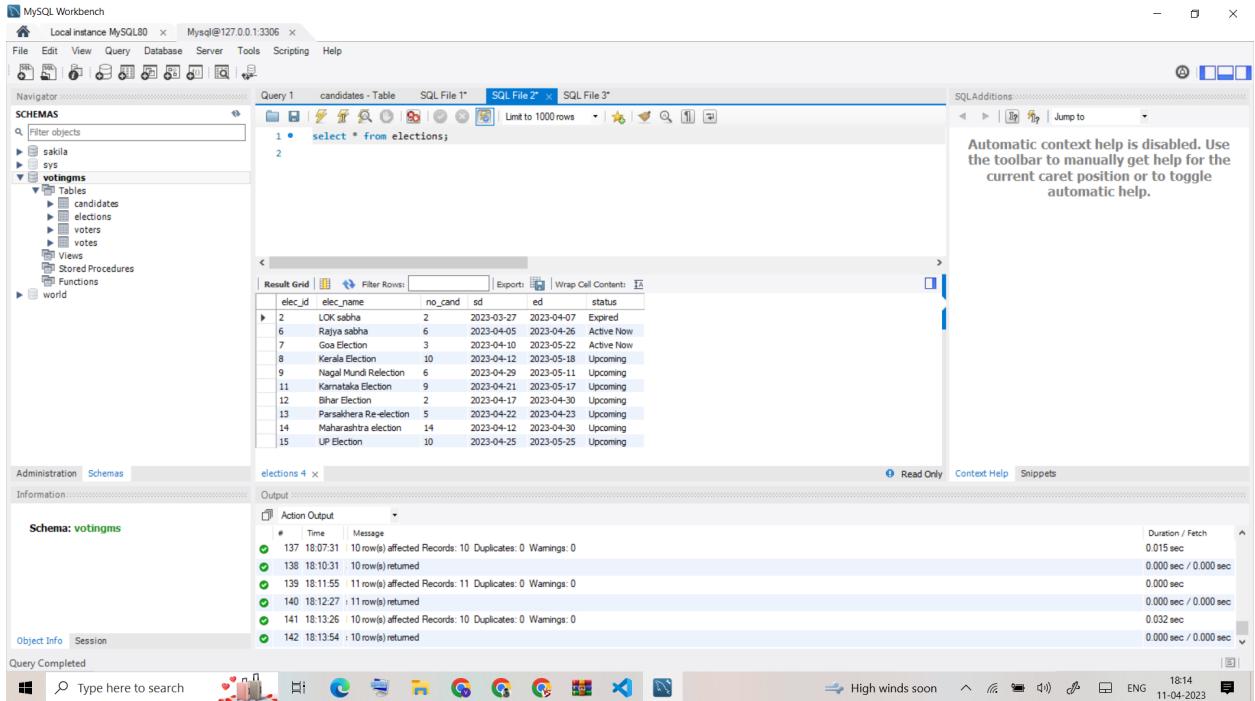


The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Shows the schema `votingms` containing tables `candidates`, `elections`, `voters`, and `votes`.
- Query Editor:** Contains the 15-line SQL insert statement provided in the text above.
- Output Window:** Shows the "Action Output" section with the following log entries:

#	Time	Message	Duration / Fetch
136	18:05:35	10 row(s) returned	0.000 sec / 0.000 sec
137	18:07:31	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.015 sec
138	18:10:31	10 row(s) returned	0.000 sec / 0.000 sec
139	18:11:55	11 row(s) affected Records: 11 Duplicates: 0 Warnings: 0	0.000 sec
140	18:12:27	11 row(s) returned	0.000 sec / 0.000 sec
141	18:13:26	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.032 sec

- System Bar:** Shows the taskbar with various application icons and the system clock indicating 18:13 on 11-04-2023.



# USER DEFINED QUERIES IN SQL:

## CREATE TABLE QUERIES:

-- create table queries

```
CREATE TABLE `candidates` (
  `cand_id` int(11) NOT NULL,
  `elec_id` int(11) NOT NULL,
  `cand_name` varchar(220) NOT NULL,
  `cand_add` varchar(400) DEFAULT NULL,
  `symbol` text DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
CREATE TABLE `elections` (
  `elec_id` int(15) NOT NULL,
  `elec_name` varchar(220) NOT NULL,
  `no_cand` int(30) NOT NULL,
  `sd` date NOT NULL,
  `ed` date NOT NULL,
  `status` text DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
CREATE TABLE `voters` (
  `username` varchar(225) NOT NULL,
  `password` text NOT NULL,
```

```

`email` varchar(200) NOT NULL,
`address` varchar(300) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
ALTER TABLE voters
ADD PRIMARY KEY (username);

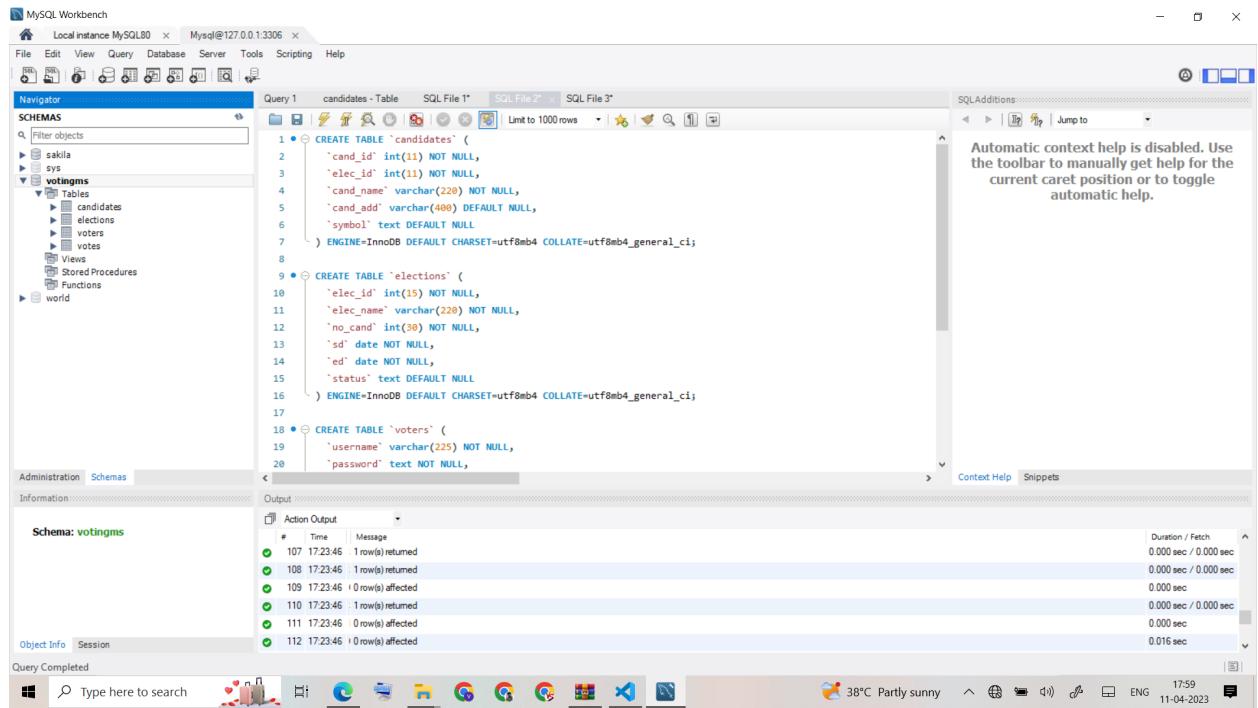
```

```

CREATE TABLE `votes` (
`vote_id` int(20) NOT NULL,
`cand_id` int(20) NOT NULL,
`username` varchar(225) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

## OUTPUT:



## **RETRIEVE NUMBER OF VOTES FOR EACH CANDIDATE:**

-- Retrieve number of votes for each candidate given that cand\_id is given

SELECT COUNT(\*) FROM votes WHERE cand\_id=2;

MySQL Workbench

Local instance MySQL800 x Mysql@127.0.0.1:3306 x

File Edit View Query Database Server Tools Scripting Help

Navigator

Schemas

- sakila
- sys
- votingsms
- Tables
- Views
- Stored Procedures
- Functions
- world

Query 1 candidates - Table SQL File 1\* SQL File 2\* SQL File 3\*

```

1 -- Retrieve number of votes for each candidate given that cand_id is given
2 • SELECT COUNT(*) FROM votes WHERE cand_id=2;
3

```

Result Grid Filter Rows: Export Wrap Cell Content: COUNT(\*)

COUNT(*)
4

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Administration Schemas Result 5 x

Information

Action Output

Schema: votingsms

#	Time	Message	Duration / Fetch
138	18:10:31	10 row(s) returned	0.000 sec / 0.000 sec
139	18:11:55	11 row(s) affected Records: 11 Duplicates: 0 Warnings: 0	0.000 sec
140	18:12:27	11 row(s) returned	0.000 sec / 0.000 sec
141	18:13:26	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.032 sec
142	18:13:54	10 row(s) returned	0.000 sec / 0.000 sec
143	18:14:37	1 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Query Completed

Type here to search

High winds soon

18:14 ENG 11-04-2023

## TOTAL NUMBER OF VOTES IN AN ELECTION:

MySQL Workbench

Local instance MySQL800 x Mysql@127.0.0.1:3306 x

File Edit View Query Database Server Tools Scripting Help

Navigator

Schemas

- sakila
- sys
- votingsms
- Tables
- Views
- Stored Procedures
- Functions
- world

Query 1 candidates - Table SQL File 1\* SQL File 2\* SQL File 3\*

```

1
2     dates.elec_id FROM votes INNER JOIN candidates ON votes.cand_id = candidates.cand_id) AS q1 WHERE q1.elec_id=6;

```

Result Grid Filter Rows: Export Wrap Cell Content: count

count
5

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Administration Schemas Result 6 x

Information

Action Output

Schema: votingsms

#	Time	Message	Duration / Fetch
139	18:11:55	11 row(s) affected Records: 11 Duplicates: 0 Warnings: 0	0.000 sec
140	18:12:27	11 row(s) returned	0.000 sec / 0.000 sec
141	18:13:26	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.032 sec
142	18:13:54	10 row(s) returned	0.000 sec / 0.000 sec
143	18:14:37	1 row(s) returned	0.000 sec / 0.000 sec
144	18:15:58	1 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Query Completed

Type here to search

38°C Partly sunny

18:16 ENG 11-04-2023

-- Total number of vote casted in an election if election id is known  
SELECT COUNT(\*) as count FROM (SELECT votes.cand\_id, votes.username, candidates.elec\_id FROM votes INNER JOIN candidates ON votes.cand\_id = candidates.cand\_id) AS q1 WHERE q1.elec\_id=6;

# PERCENTAGE OF VOTES IN AN ELECTION:

The screenshot shows the MySQL Workbench interface. In the top-left corner, it says "Local instance MySQL80" and "Mysql@127.0.0.1:3306". The main area has three tabs: "Query 1" (candidates - Table), "SQL File 1\*", and "SQL File 2\*". The "Query 1" tab contains the following SQL code:

```
-- percentage of votes received by a candidate when elec_id and cand_id is known here cand_id=7 and elec_id=6
SELECT (100* (SELECT CAST(COUNT(*) AS decimal(9,2)) FROM votes WHERE cand_id=7)/(SELECT CAST(COUNT(*) AS decimal(9,2)) FROM votes WHERE elec_id=6)) AS percentage;
```

The result grid shows one row with the value 80.000000. The bottom part of the interface shows the "Information" panel with the "Schema: votingms" selected, displaying a log of actions from 142 to 147.

-- percentage of votes received by a candidate when elec\_id and cand\_id is known here  
cand\_id=7 and elec\_id=6

```
SELECT (100* (SELECT CAST(COUNT(*) AS decimal(9,2)) FROM votes WHERE cand_id=7)/(SELECT CAST(COUNT(*) AS decimal(9,2)) AS count FROM (SELECT votes.cand_id, votes.username, candidates.elec_id FROM votes INNER JOIN candidates ON votes.cand_id = candidates.cand_id) AS q1 WHERE q1.elec_id=6)) AS percentage;
```

# RETRIEVE THE WINNER OF AN ELECTION:

The screenshot shows the MySQL Workbench interface. In the top-left pane, the 'Schemas' tree is visible, showing databases like sakila, sys, and votingms. The votingms database is selected, revealing tables such as candidates, elections, users, and votes. In the central pane, three tabs are open: 'Query 1' (candidates - Table), 'SQL File 1\*' (containing the SQL code), and 'SQL File 2\*' (empty). The 'Result Grid' shows a single row for 'Ramu'. The bottom pane displays the 'Information' tab for the votingms schema, specifically the 'Object Info' section, which lists various objects and their details. The status bar at the bottom right shows system information like temperature (38°C), battery level (Partly sunny), and date/time (11-04-2023).

```
-- retrieve the the winner of an election when elec_id is known
CREATE VIEW candv AS (SELECT cand_id,COUNT(*) as count FROM votes WHERE cand_id IN (SELECT cand_id FROM candidates WHERE elec_id=6) GROUP BY cand_id);
SELECT cand_name FROM candidates WHERE cand_id IN (SELECT cand_id FROM candv WHERE count=(SELECT MAX(count) FROM candv));
DROP VIEW candv;
```

# RETRIEVE RANK OF EACH CANDIDATE:

The screenshot shows the MySQL Workbench interface. In the Navigator pane, under the 'votingms' schema, there are tables for candidates, elections, voters, and votes. The 'Query 1' tab contains the following SQL code:

```
-- retrieve rank of each candidate
CREATE VIEW candv AS (SELECT cand_id,COUNT(*) as count FROM votes WHERE cand_id IN (SELECT cand_id FROM candidates WHERE elec_id=6) GROUP BY cand_id);
CREATE VIEW result AS (SELECT cand_id,RANK() OVER (ORDER BY count desc) election_rank FROM candv);
SELECT candidates.cand_name, result.election_rank FROM candidates INNER JOIN result ON candidates.cand_id = result.cand_id;
DROP VIEW candv;
DROP VIEW result;
```

The 'Result Grid' pane shows the output of the query:

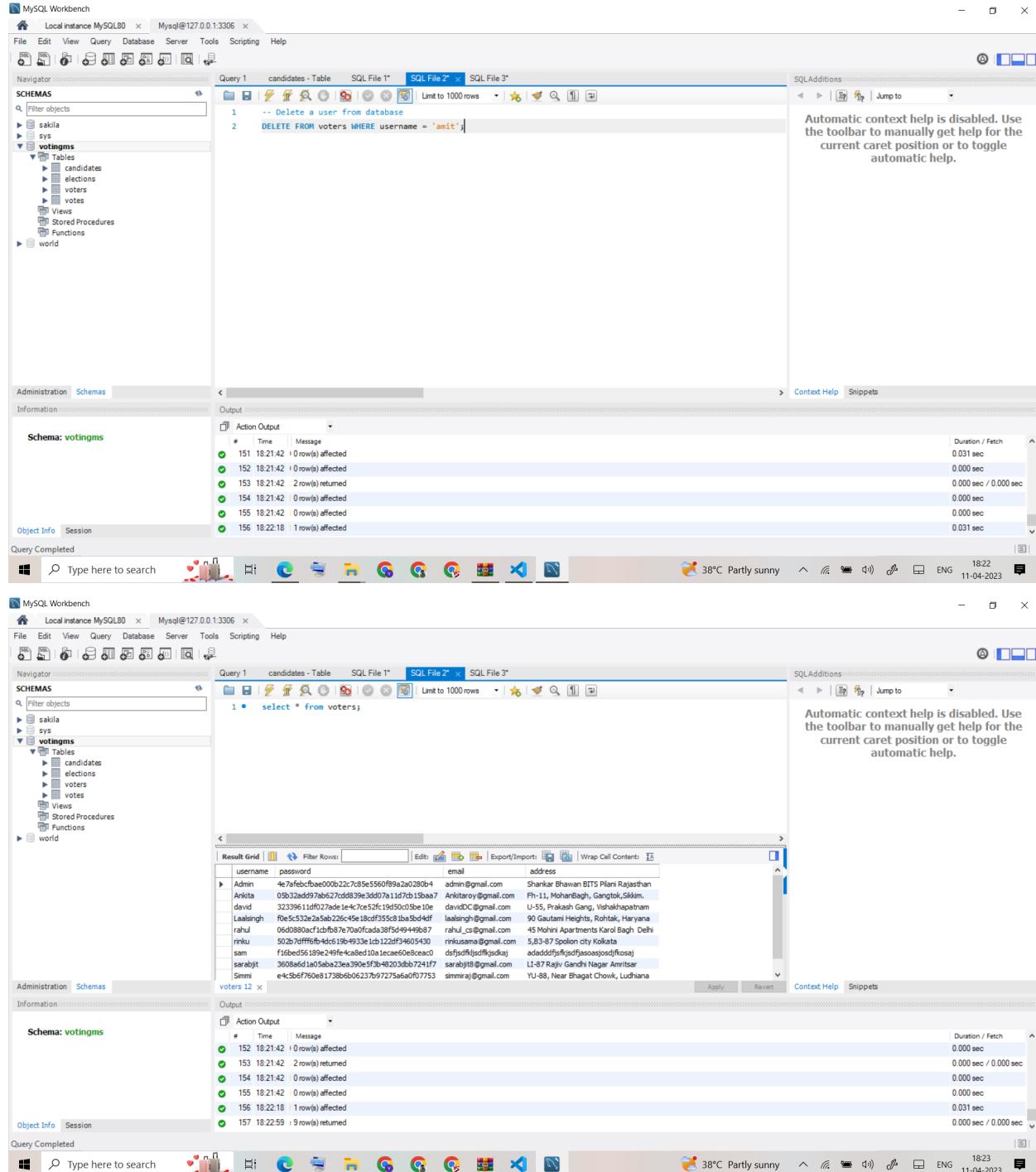
cand_name	election_rank
Mohanjodaro	2
Ramu	1

The 'Output' pane shows the action log:

#	Time	Message	Duration / Fetch
150	18:20:55	0 rows affected	0.000 sec
151	18:21:42	0 rows affected	0.031 sec
152	18:21:42	0 rows affected	0.000 sec
153	18:21:42	2 rows returned	0.000 sec / 0.000 sec
154	18:21:42	0 rows affected	0.000 sec
155	18:21:42	0 rows affected	0.000 sec

```
-- retrieve rank of each candidate
CREATE VIEW candv AS (SELECT cand_id,COUNT(*) as count FROM votes WHERE cand_id IN (SELECT cand_id FROM candidates WHERE elec_id=6) GROUP BY cand_id);
CREATE VIEW result AS (SELECT cand_id,RANK() OVER (ORDER BY count desc)
election_rank FROM candv);
SELECT candidates.cand_name, result.election_rank FROM candidates INNER JOIN result ON
candidates.cand_id = result.cand_id;
DROP VIEW candv;
DROP VIEW result;
```

# DELETE A USER FROM DATABASE



The screenshot shows two instances of MySQL Workbench. Both instances are connected to a Local instance MySQL80 at port 127.0.0.1:3306.

**Query 1 (Top Window):**

```
-- Delete a user from database
DELETE FROM voters WHERE username = 'amit';
```

**Output (Top Window):**

#	Time	Message	Duration / Fetch
151	18:21:42	0 row(s) affected	0.031 sec
152	18:21:42	0 row(s) affected	0.000 sec
153	18:21:42	2 row(s) returned	0.000 sec / 0.000 sec
154	18:21:42	0 row(s) affected	0.000 sec
155	18:21:42	0 row(s) affected	0.000 sec
156	18:22:18	1 row(s) affected	0.031 sec

**Query 2 (Bottom Window):**

```
select * from voters;
```

**Result Grid (Bottom Window):**

username	password	email	address
admin	e47afebcfae000227d5e556ff9ea20a3bd-4	admin@gmail.com	Shankar Bhawan BTS Pipli Rajastan
Anikita	05b32add7a7b627cd839c36d7a1d7cb13ba47	Anikitroy@gmail.com	Fh-11, MohanBagh, Gangtok,Sikkim.
david	323396116f027ade1e47e5e7fc19d500c9e10a	davidDC@gmail.com	U-55, Prakash Gang, Vishakhapatnam
Laalisingh	f0e5c532e2a5ab226c45e18cf3f5583b1ba5b4df	laalisingh@gmail.com	90 Gautami Heights, Rohitk, Haryana
rahul	06d0880a0ac1dfb87e70a0cada38f5d4944b67	rahul_cs@gmail.com	45 Mohini Apartments Karol Bagh, Delhi
rinku	502b7dfffb46d19b4933e1cb1220f34605430	rinkusama@gmail.com	5,83-87 Spolion city Kolkata
sam	f16bed56189e249fe4a8ed10a1ecae60e8ceac0	adaddff4fc9pdfasasjssj9rossaj	dfsfdsfkjlkdfjkjksdkj
simmi	3608ae51a05aba3e3a390e5f3b48203dbb72417	simmiraj@gmail.com	L1-87 Rajiv Gandhi Nagar Amritsar
surajit	e4c5b5f760e81738bd06237b97275a6a0f07753	surajit8@gmail.com	YU-88, Near Bhagat Chowk, Ludhiana

**Output (Bottom Window):**

#	Time	Message	Duration / Fetch
152	18:21:42	0 row(s) affected	0.000 sec
153	18:21:42	2 row(s) returned	0.000 sec
154	18:21:42	0 row(s) affected	0.000 sec
155	18:21:42	0 row(s) affected	0.000 sec
156	18:22:18	1 row(s) affected	0.031 sec
157	18:22:59	9 row(s) returned	0.000 sec / 0.000 sec

-- Delete a user from database

DELETE FROM voters WHERE username = 'amit';

# UPDATE USER INFO:

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Local instance MySQL80, Mysql@127.0.0.1:3306.
- Query Editor:** SQL File 2\* contains the following SQL code:

```
1 -- Update user info(here email)
2 UPDATE voters SET email = 'rahul745@gmail.com' WHERE username = 'rahul';
3 select * from voters;
```
- Result Grid:** Shows the results of the SELECT query on the 'voters' table. The columns are username, password, email, and address. The data includes rows for Admin, Ankita, david, Laal Singh, rinku, sam, sarabjit, and Simmi, each with their respective details.
- Output Tab:** Action Output section shows the log of events:

#	Time	Message	Duration / Fetch
154	18:21:42	0 row(s) affected	0.000 sec
155	18:21:42	0 row(s) affected	0.000 sec
156	18:22:18	1 row(s) affected	0.031 sec
157	18:22:59	9 row(s) returned	0.000 sec / 0.000 sec
158	18:24:03	1 rows affected Rows matched: 1 Changed: 1 Warnings: 0	0.031 sec
159	18:24:03	9 row(s) returned	0.000 sec / 0.000 sec
- System Bar:** Includes a search bar, system icons, and a status bar showing 38°C Partly sunny, ENG, 18:24, and 11-04-2023.

-- Update user info(here email)

UPDATE voters SET email = 'rahul745@gmail.com' WHERE username = 'rahul';

**THE END**