**Sri Lankan Presidential Election Results 2024**

**By: Maheshika Wijesooriya**

**Introduction**

This report presents an in-depth analysis of the 2024 Sri Lankan Presidential Election, demonstrating the use of modern data analytics and visualization tools to gain valuable insights into electoral outcomes. The project begins with structured data modeling in SQL, integrating information on candidates, districts, and polling stations. SQL queries were used to extract and analyze key metrics, such as vote distributions, leading and trailing candidates, and voter turnout across districts.

The analysis is further enhanced with Power BI dashboards, which provide a visual representation of the data through interactive maps, charts, and KPI indicators. These dashboards illustrate district-wise leading candidates, voter engagement, and overall election trends, allowing stakeholders to explore and interpret the data effectively.

By combining SQL and Power BI, this project showcases the potential of data-driven decision-making, providing a foundation for future applications in elections and beyond. It aims to serve as a guide for leveraging analytics to simplify complex datasets and deliver meaningful insights.

**Content**

**Page**

Tables -------------------------------------------------------- 4

Meaning of Each Column ------------------------------------------------------- 5

Primary Keys and Foreign Keys ------------------------------------------------------ 7

Fact Table & Dimension Tables ------------------------------------------------------ 8

Structure of the Table ------------------------------------------------------ 12

SQL Data Analysis ------------------------------------------------------ 13

KPIs ------------------------------------------------------ 22

Power BI Visualization ------------------------------------------------------ 23

References for Data ------------------------------------------------------ 31

1. **Tables**
2. **Candidatewise\_details**
   * **Columns:** CandidateID, CandidateName, PartyName, PartySymbol, Experience
   * **Purpose:** Stores essential information about each candidate, including their political affiliation and background details.
3. **CandidateWise Results**
   * **Columns:** CandidateID, PartyName, CandidatewiseTotalVotes, CandidatewisePreferential, CandidatewiseTotal (calculated as the sum of CandidatewiseTotalVotes and CandidatewisePreferential)
   * **Purpose:** Captures the total votes for each candidate based on different voting methods. We can calculate the percentage of total votes in queries.
4. **District**
   * **Columns:** DistrictID, DistrictName , Province, Country
   * **Purpose:** Define each district for referencing in other tables.
5. **Districtwise\_results**
   * **Columns:** LeadingCandidateID, TrailingCandidateID, LeadingCandidateEVM, LeadingCandidatePostal, LeadingCandidatePreferential, LeadingCandidateTotal, DistrictID, DistrictName
   * **Purpose:** Summarizes the results for each district, including vote counts by method for the leading candidate. We can also calculate the percentage of votes for the leading candidate during queries.
6. **PollingStationwise\_Details**
   * **Columns:** PollingStation, PollingStationID, DistrictID, WinningCandidateID, NoRegisteredElectors, TotalNoValidVotes, NoRejectedVotes, NoVotesPolled , DistrictName
   * **Purpose:** Stores details about each polling station, including the winning candidate and voter turnout statistics.
7. **Meaning of Each Column**

**2.1 Candidatewise\_details**

**Primary Key**: CandidateID

**Columns**:

* CandidateID (VARCHAR): Unique identifier for each candidate (Primary Key).
* CandidateName (STRING): Full name of the candidate.
* PartyName (STRING): Name of the candidate’s political party.
* PartySymbol (STRING): Symbol representing the candidate’s party.
* Experience (INTEGER or STRING): Candidate’s experience, such as years in politics or related roles.

**2.2 CandidateWise Results**

**Primary Key**: CandidateID

**Columns**:

* CandidateID (VARCHAR): Unique identifier linking to each candidate (Primary Key, Foreign Key to Candidatewise\_Details).
* PartyName (STRING): Name of the candidate’s political party.
* CandidatewiseTotalVotes (INTEGER): Total general votes received by each candidate.
* CandidatewisePreferential (INTEGER): Total preferential votes received by each candidate.
* CandidatewiseTotal (INTEGER): Combined total of general and preferential votes.

**2.3 District**

**Primary Key**: DistrictID

**Columns**:

* DistrictID (VARCHAR): Unique identifier for each district (Primary Key).
* DistrictName (STRING): Name of each district.
* Country (STRING)
* Province (STRING)

**2.4 Districtwise\_results**

**Primary Key**: DistrictID

**Columns**:

* DistrictID (VARCHAR): Unique identifier linking to each district (Primary Key, Foreign Key to District).
* LeadingCandidateID (VARCHAR): Unique identifier of the leading candidate in each district (Foreign Key to Candidatewise\_Details).
* TrailingCandidateID (VARCHAR): Unique identifier of the trailing candidate in each district (Foreign Key to Candidatewise\_Details).
* LeadingCandidateEVM (INTEGER): Total votes received by the leading candidate through EVM.
* LeadingCandidatePostal (INTEGER): Total postal votes received by the leading candidate.
* LeadingCandidatePreferential (INTEGER): Preferential votes received by the leading candidate.
* LeadingCandidateTotal (INTEGER): Total votes received by the leading candidate, combining all voting types.
* DistrictName (STRING): Name of each district.

**2.5 PollingStationwise\_Details**

**Primary Key**: PollingStationID

**Columns**:

* PollingStationID (VARCHAR): Unique identifier for each polling station (Primary Key).
* PollingStation (STRING): Name of the polling station.
* DistrictID (VARCHAR): Identifier linking to the district of the polling station (Foreign Key to District).
* WinningCandidateID (VARCHAR): Identifier of the winning candidate at this polling station (Foreign Key to Candidatewise\_Details).
* NoRegisteredElectors (INTEGER): Number of registered electors at the polling station.
* TotalNoValidVotes (INTEGER): Total number of valid votes cast at the polling station.
* NoRejectedVotes (INTEGER): Number of votes rejected.
* NoVotesPolled (INTEGER): Total number of votes polled at the station, including both valid and rejected votes.
* DistrictName (STRING): Name of each district.

1. **Primary Keys and Foreign Keys**

* **Primary Keys**:
* CandidateID in Candidatewise\_Details and Candidatewise\_Results.
* DistrictID in District.
* PollingStationID in PollingStationwise\_Details.
* **Foreign Keys:**
* CandidateID in Candidatewise\_Results, and LeadingCandidateID and TrailingCandidateID in Districtwise\_Results linking to Candidatewise\_Details.
* DistrictID in PollingStationwise\_Details and Districtwise\_Results linking to the District table.
* WinningCandidateID in PollingStationwise\_Details linking to Candidatewise\_Details.

1. **Fact Table & Dimension Tables**

**4.1 Fact Table**

**CandidateWise Results**

* **Reason for Fact Table**:
* This table contains the core quantitative data to be analyzed, specifically the vote counts (CandidatewiseTotalVotes, CandidatewisePreferential, CandidatewiseTotal) for each candidate.
* Fact tables in data models typically store measurable data with foreign keys referencing dimensions. Here, CandidateWise Results provides the essential measurable elements for analysis (vote counts per candidate).
* As the central fact table, it enables aggregation and filtering by various dimensions, such as party, district, and polling station.
* **Columns**:
  + CandidateID (Foreign Key to link to candidate details)
  + Party Name
  + CandidatewiseTotalVotes
  + CandidatewisePreferential
  + CandidatewiseTotal

**4.2 Dimension Tables**

Each of the following tables serves to add descriptive or contextual information to the model. Dimension tables store qualitative data and provide context for the quantitative data in the fact table.

**4.2.1 Candidatewise\_details**

* **Reason for Dimension Table**:
  + This table contains descriptive attributes of each candidate, such as CandidateID, CandidateName, PartyName, PartySymbol, and Experience.
  + Dimension tables typically hold descriptive details, useful for adding context without increasing the complexity of fact tables.
  + This dimension allows slicing and dicing of fact table data by candidate attributes (e.g., analyzing votes by candidate’s party or experience).**Primary Key**: CandidateID, which links to CandidateWise Results.
* **Primary Key :** CandidateID, which links to CandidateWise Results.

**4.2.2 District**

* **Reason for Dimension Table**:
  + The District table provides geographic context, aiding in the analysis of votes by location.
  + Dimension tables often represent entities like locations, time, or categories. District is a location category dimension, offering valuable filtering options without storing redundant location information directly in the fact table.
* **Primary Key**: DistrictID, which is used to link with district-level or constituency-level tables.

**4.2.3 Districtwise\_Results**

* **Reason for Dimension Table**:
  + This table summarizes the results at the district level, with attributes like LeadingCandidateID, TrailingCandidateID, LeadingCandidateEVM, LeadingCandidatePostal, LeadingCandidatePreferential, and LeadingCandidateTotal, DistrictName
  + It provides valuable context on candidate performance in each district, enabling analysis by leading and trailing results without adding complexity to the main fact table.
  + This table serves as a useful summary of district performance, enhancing the detail available in regional analysis.
* **Primary Key**: DistrictID (Foreign Key to District), with LeadingCandidateID and TrailingCandidateID linking to candidates in Candidatewise\_Details.

**4.2.4 PollingStationwise\_Details**

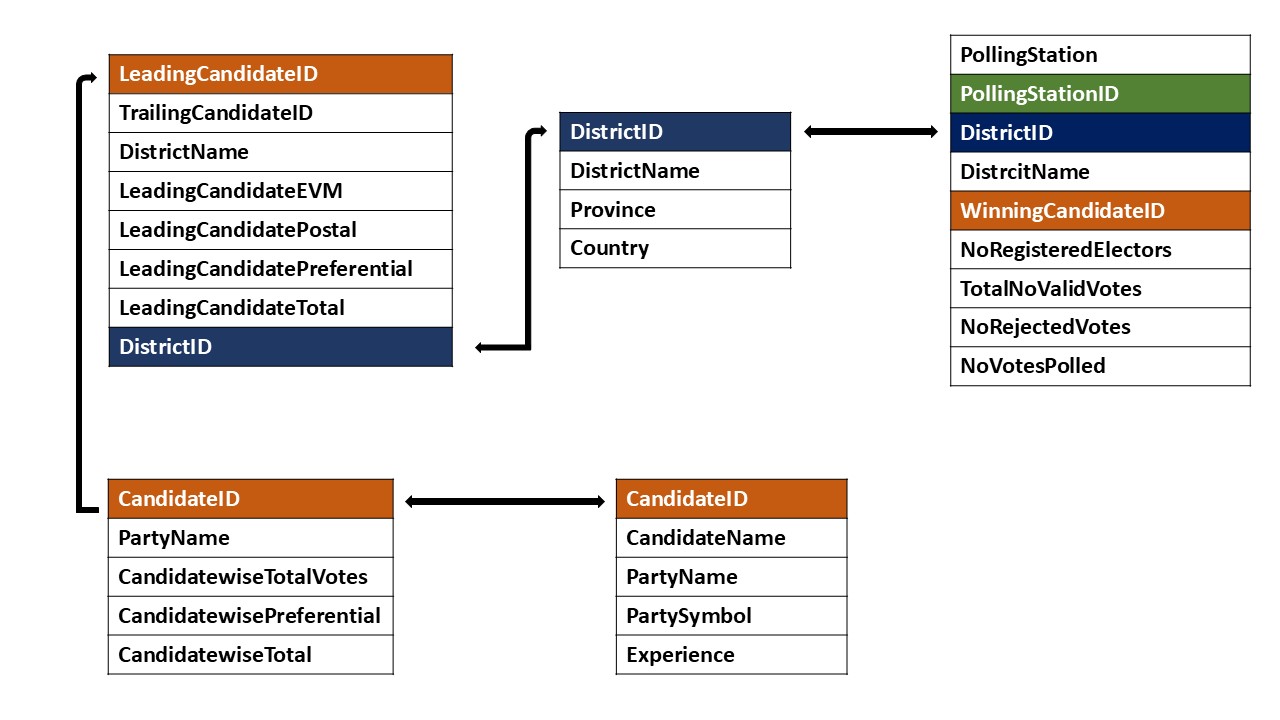
* **Reason for Dimension Table**:
  + This table provides details on each constituency, including PollingStation, PollingStationID, DistrictID, WinningCandidateID, NoRegisteredElectors, TotalNoValidVotes, NoRejectedVotes, NoVotesPolled, DistrictName
  + By adding context on voter registration and turnout at the constituency level, this table allows for a granular analysis of voting behavior by region.
  + This data enhances the analysis of voter engagement and candidate success at the local level.
* **Primary Key**: PollingStationID, which links to constituency-specific data and provides regional context for the fact data.

**Summary of Structure**

* **Fact Table**:
  + **CandidateWise Results**: This table captures the primary quantitative metrics (vote counts) that you want to analyze.
* **Dimension Tables**:
  + **Candidatewise\_details**: Attributes related to candidates.
  + **District**: Attributes related to districts.
  + **Districtwise\_results**: Summary data that provides context for district-level performance.
  + **PollingStation\_Details**: Attributes related to constituencies and voter statistics.

**Why This Structure?**

* **Ease of Analysis**:  
  This structure allows you to perform complex queries and analyses easily. For example, you can analyze how different candidates performed in various districts and constituencies.
* **Flexibility**:  
  It provides flexibility in reporting and visualization, enabling you to slice and dice the data based on candidates, regions, and voter behavior.
* **Performance**:  
  In a star schema, querying is generally faster because the fact table contains the metrics of interest, while dimension tables hold the attributes that can be joined to filter or categorize those metrics.

1. **Structure of the Tables**
2. **SQL Data Analysis**

**1. Which candidate received the highest total votes overall?**

* This query combines the Candidatewise\_details and CandidateWise Results tables to find the candidate with the highest total votes.

SELECT TOP 1

cd.CandidateName,

cd.PartyName,

cr.CandidatewiseTotalVotes

FROM

Candidatewise\_Details AS cd

JOIN

Candidatewise\_Results AS cr

ON

cd.CandidateID=cr.CandidateID

ORDER BY

cr.CandidatewiseTotalVotes DESC;



**2. What is the leading and trailing candidate in each district?**

* This query combines Districtwise\_results, Candidatewise\_details, and District tables to show the leading and trailing candidates for each district.

SELECT

d.DistrictName,

lc.CandidateName AS Leading\_Candidate,

tc.CandidateName AS Trailing\_Candidate,

dr.LeadingCandidateTotal

FROM

Districtwise\_Results AS dr

JOIN

Candidatewise\_Details AS lc

ON

dr.LeadingCandidateID=lc.CandidateID

JOIN

Candidatewise\_Details AS tc

ON

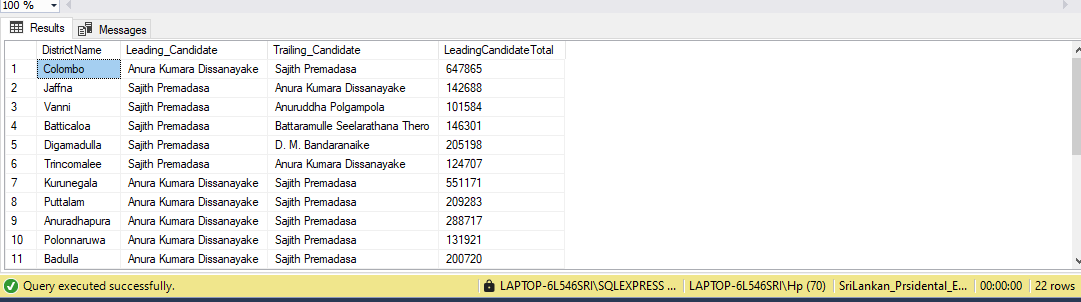
dr.TrailingCandidateID=tc.CandidateID

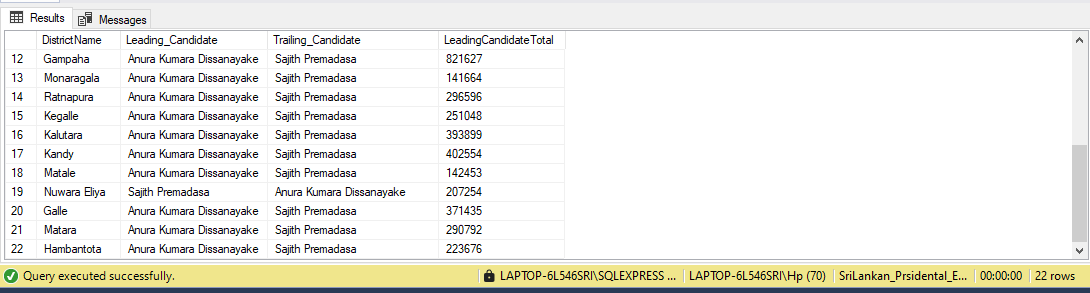
JOIN

District AS d

ON

dr.DistrictID=d.DistrictID;





**3. What is the total number of registered electors and votes polled per district?**

* This query combines the PollingStation\_Results and District tables to get the total number of electors and votes polled by district.

SELECT

d.DistrictName,

SUM(pr.NoRegisteredElectors) AS TotalRegisteredElectors,

SUM(pr.NoVotesPolled) AS TotalVotesPolled

FROM

PollingStationwise\_Results AS pr

JOIN

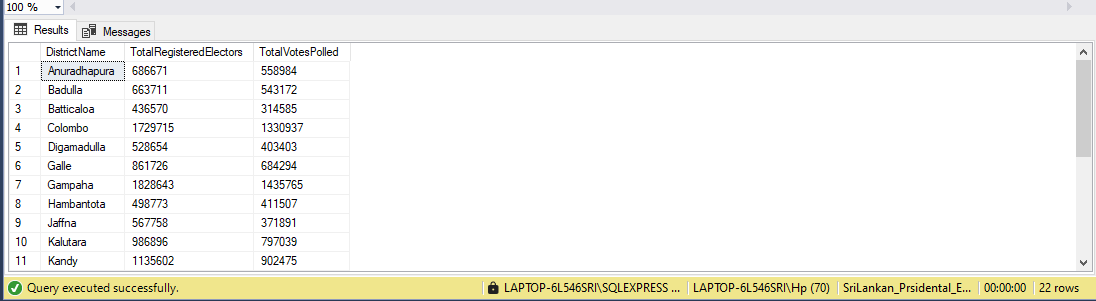
District AS d

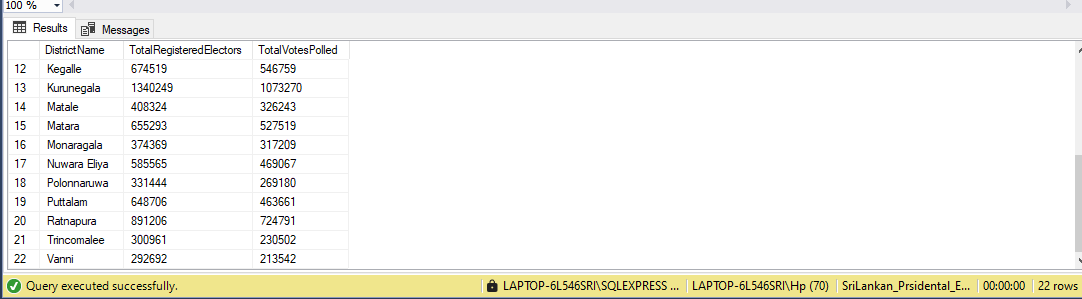
ON

pr.DistrictID=d.DistrictID

GROUP BY

d.DistrictName;





**4. What is the vote distribution by type (EVM, Postal, Preferential) for each candidate?**

* This query will provide details on each vote type for every candidate, combining Candidatewise\_details and CandidateWise Results.

SELECT

cd.CandidateName,

cr.PartyName,

cr.CandidatewiseTotalVotes,

cr.CandidatewisePreferential,

cr.CandidatewiseTotal

FROM

Candidatewise\_details cd

JOIN

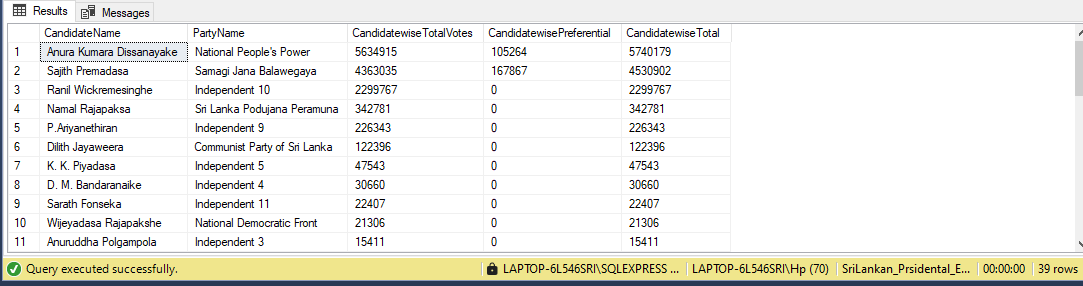
CandidateWise\_Results cr

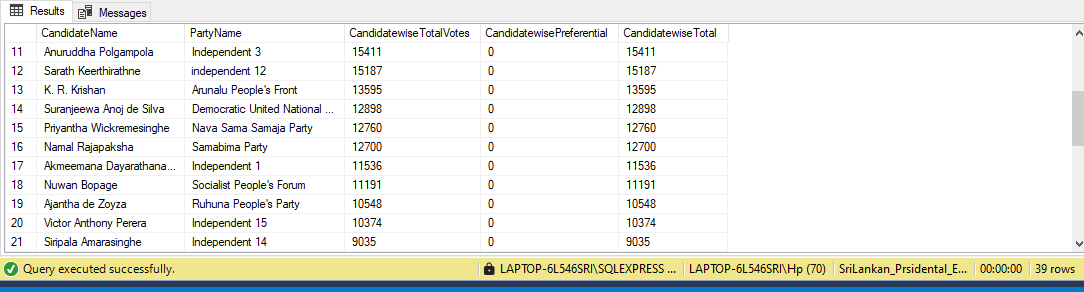
ON

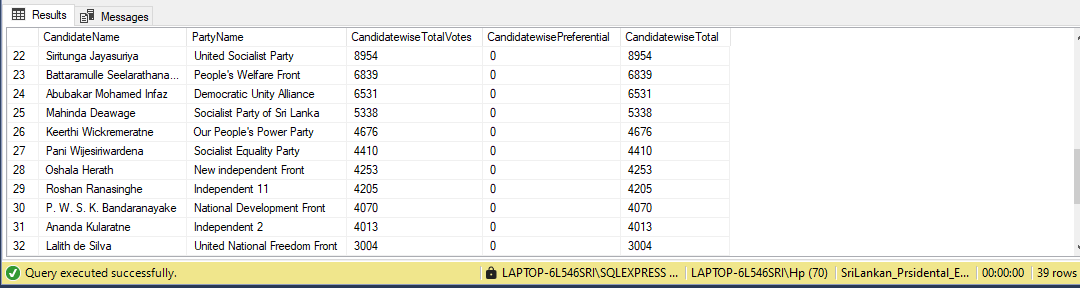
cd.CandidateID = cr.CandidateID

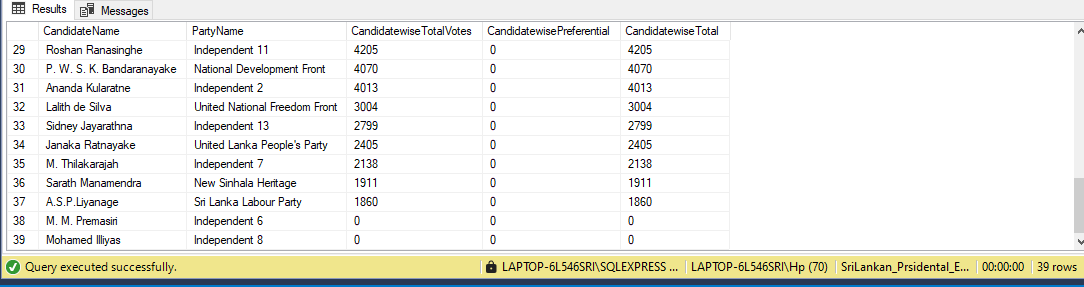
ORDER BY

cr.CandidatewiseTotal DESC;









**5.Retrieve each candidate's name, party, and total votes only for candidates whose party received more than 500,000 votes overall.**

SELECT

cd.CandidateName,

cd.PartyName,

cr.CandidatewiseTotal

FROM

Candidatewise\_Results AS cr

JOIN

Candidatewise\_Details AS cd

ON

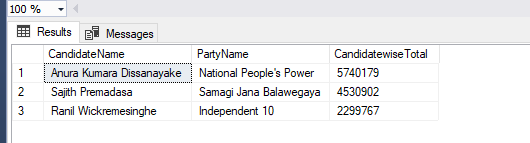
cr.CandidateID=cd.CandidateID

WHERE

cr.CandidatewiseTotalVotes > 500000

ORDER BY

cr.CandidatewiseTotalVotes DESC;



1. **Retrieve each district’s name and the total votes of the Leading candidate, only for districts where the Leading candidate’s name starts with "A" and their total votes exceed 100,000.**

SELECT

d.DistrictName,

cd.CandidateName AS 'Anura as Winning candidate',

dr.LeadingCandidateTotal AS Results

FROM

District AS d

JOIN

Districtwise\_Results AS dr

ON

d.DistrictID=dr.DistrictID

JOIN

Candidatewise\_Details AS cd

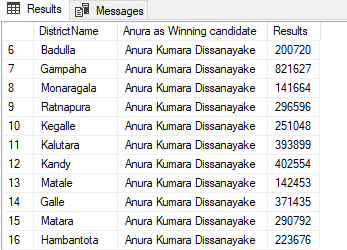
ON

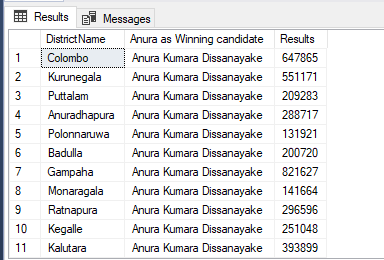
dr.LeadingCandidateID=cd.CandidateID

WHERE

cd.CandidateName LIKE 'A%'

AND dr.LeadingCandidateTotal > 100000;

****

****

1. **Find the top three candidates in terms of total votes across all polling stations, along with their polling station details**

SELECT

cd. CandidateName,

cd.PartyName,

pr.PollingStation,

pr.TotalNoValidVotes

FROM

PollingStationwise\_Results AS pr

JOIN

Candidatewise\_Details AS cd

ON

pr.WinningCandidateID=cd.CandidateID

WHERE

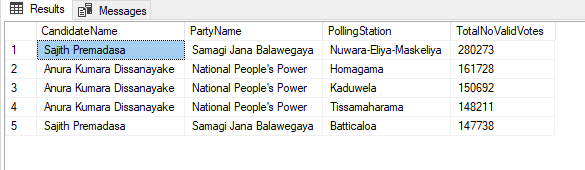
pr.TotalNoValidVotes

IN

(SELECT TOP 5 TotalNoValidVotes FROM PollingStationwise\_Results ORDER BY TotalNoValidVotes DESC)

ORDER BY

TotalNoValidVotes DESC;



**8. Calculate the total number of registered electors, valid votes, and rejected votes for each district, only for districts where the total registered electors exceed 100,000.**

SELECT

d.DistrictName,

SUM(pr.NoRegisteredElectors) AS 'Total No. Registered Electors',

SUM(pr.TotalNoValidVotes) AS 'Total No. Valid Votes',

SUM(pr.NoRejectedVotes) AS 'Total No. Rejected Votes',

SUM(pr.NoVotesPolled ) AS 'Total No. Polled Votes'

FROM

PollingStationwise\_Results AS pr

JOIN

District AS d

ON

pr.DistrictID=d.DistrictID

GROUP BY

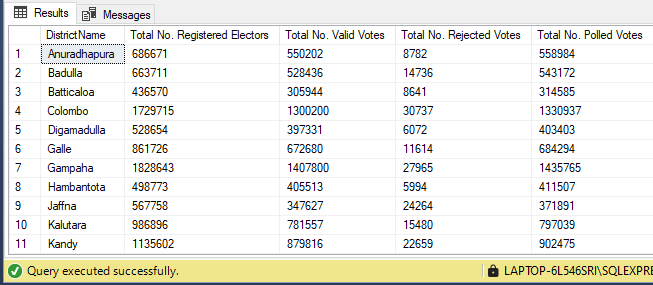
d.DistrictName

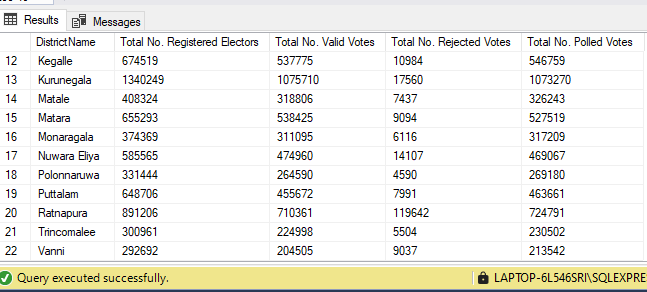
HAVING

SUM(pr.NoRegisteredElectors) > 100000

ORDER BY

d.DistrictName;





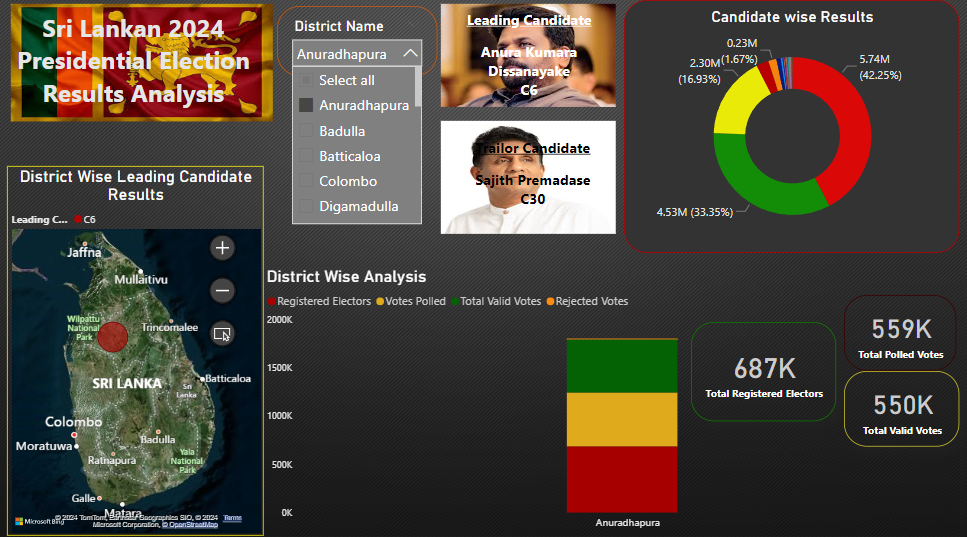
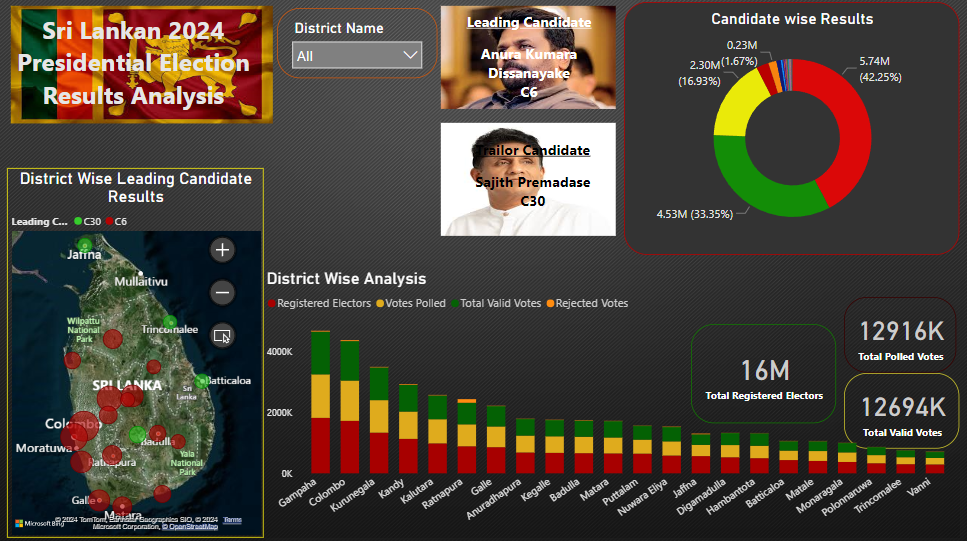
1. **KPI**
2. **District-Wise Leading Candidate Analysis -** Shows which candidate leads in each district, offering a visual overview of candidate dominance geographically.
3. **Average District-Wise Voting Statistics:**

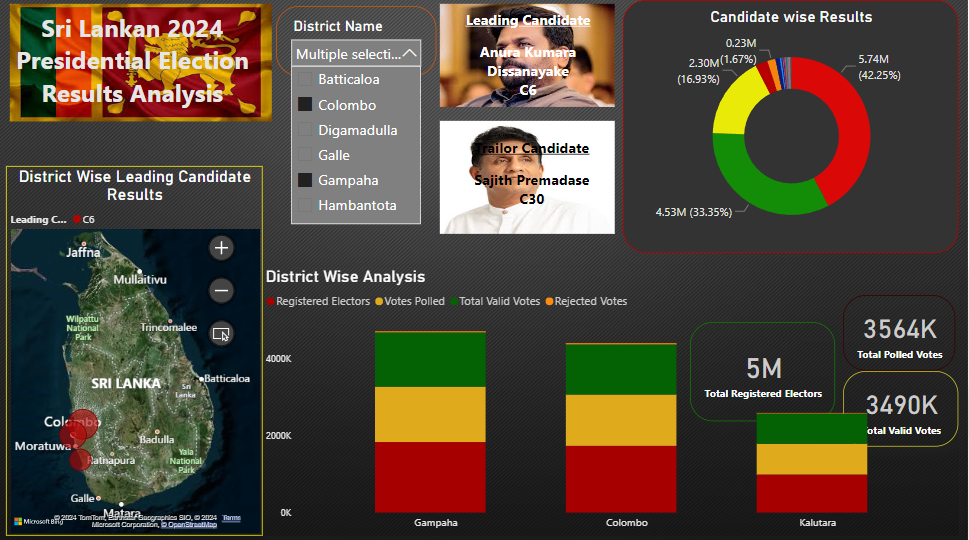
* Average Registered Electors per District - Average count of voters registered per district.
* Average Votes Polled per District - Provides insight into voter turnout averages.
* Average Valid Votes per District - Indicates how many votes counted as valid on average.
* Average Rejected Votes per District - Shows the average rate of rejected votes, useful for assessing any issues in the voting process.

1. **Total Voter Metrics -** KPI cards showing overall turnout and participation:

* Total Registered Electors - Total number of people eligible to vote across all polling stations.
* Total Polled Voters - Total number of votes cast, indicating overall voter turnout.
* Total Valid Votes - Total valid votes cast across all polling stations, a measure of effective participation.

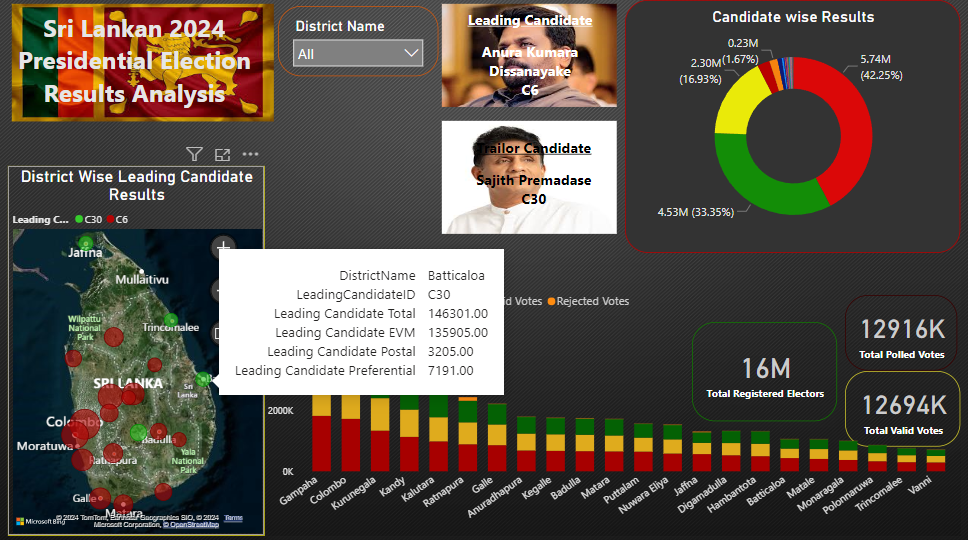
1. **Candidate Performance Breakdown (Pie Chart) -** Displays vote distribution among candidates, providing insight into each candidate's share of the total vote.
2. **Power BI Visualization**

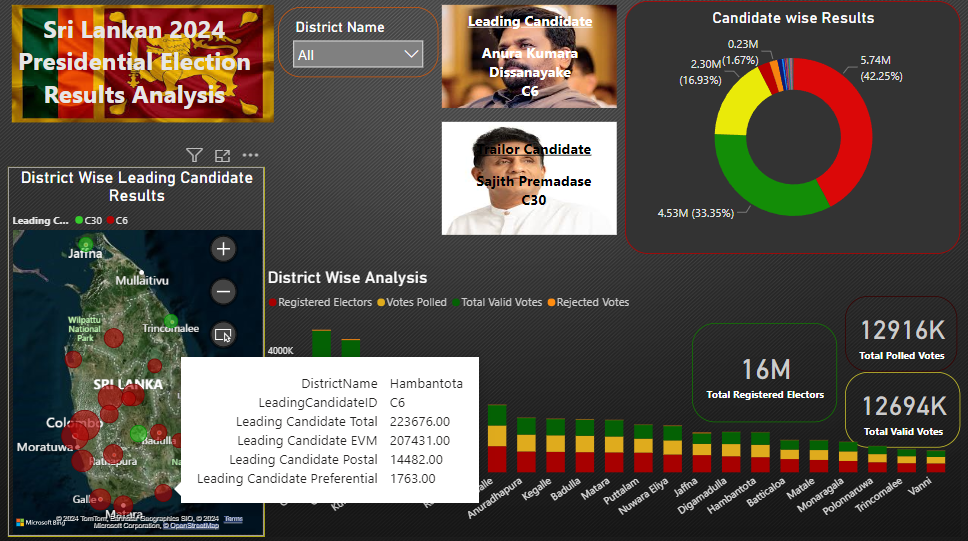
****



1. **District-Wise Leading Candidate Overview**

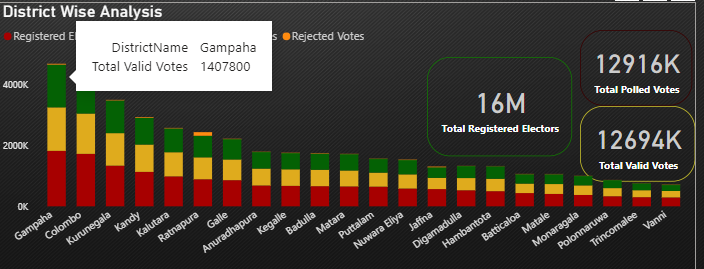
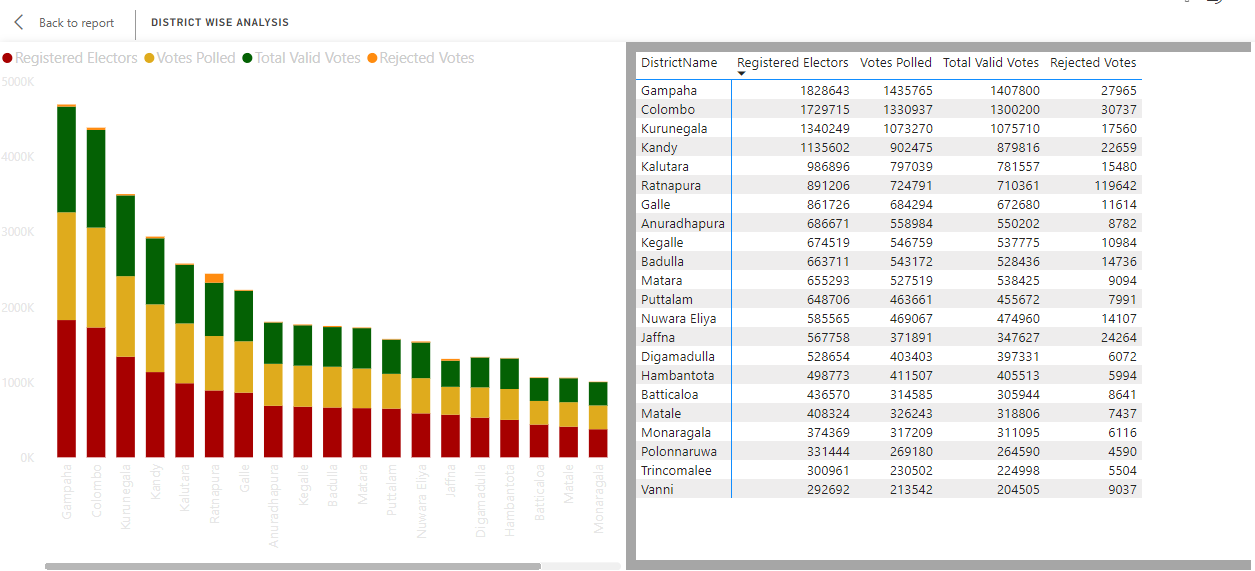
* Visualization: Filled Map
* Details: Displays the leading candidate per district using LeadingCandidateID, with color-coding for each candidate. Tooltips provide additional insights into each district's leading candidate.
* Purpose: Offers a geographic perspective of candidate performance, making it easy to see where each candidate holds the most support.





1. **District-Wise Voting Statistics Analysis**

* Visualization: Stacked Column Chart
* Details: Displays the average values of NoRegisteredElectors, NoVotesPolled, TotalValidVotes, and NoRejectedVotes for each district, giving a breakdown of voter engagement across districts.
* Purpose: Helps to visualize district-level engagement, including overall participation and rejected votes, highlighting areas with high or low voter turnout.

****

1. **Total Voter Metrics**

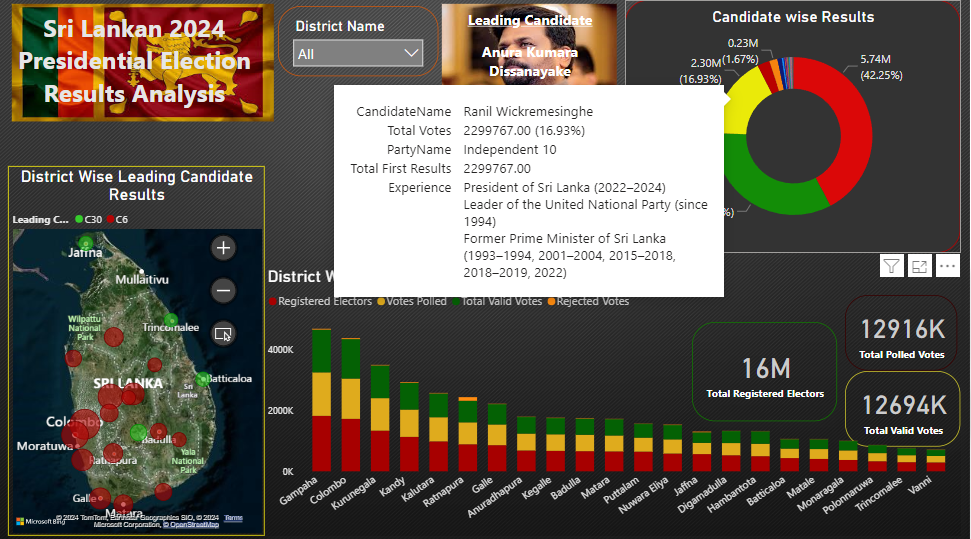
* Visualization: KPI Cards
* Details: The KPI cards show Total Registered Electors, Total Polled Voters, and Total Valid Votes from PollingStationwise\_Results, providing an overall summary of key voter statistics.
* Purpose: Offers a quick overview of core metrics, summarizing voter turnout and participation across the election.

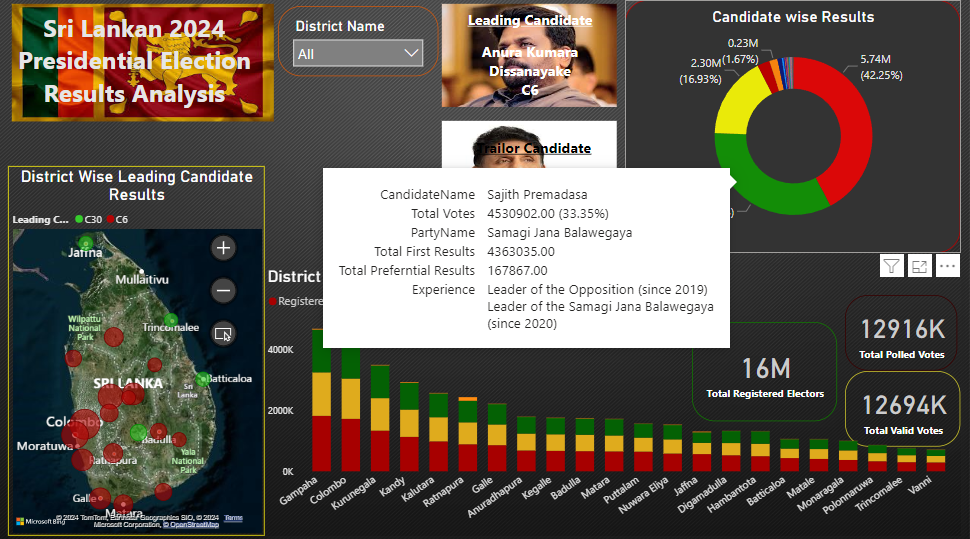
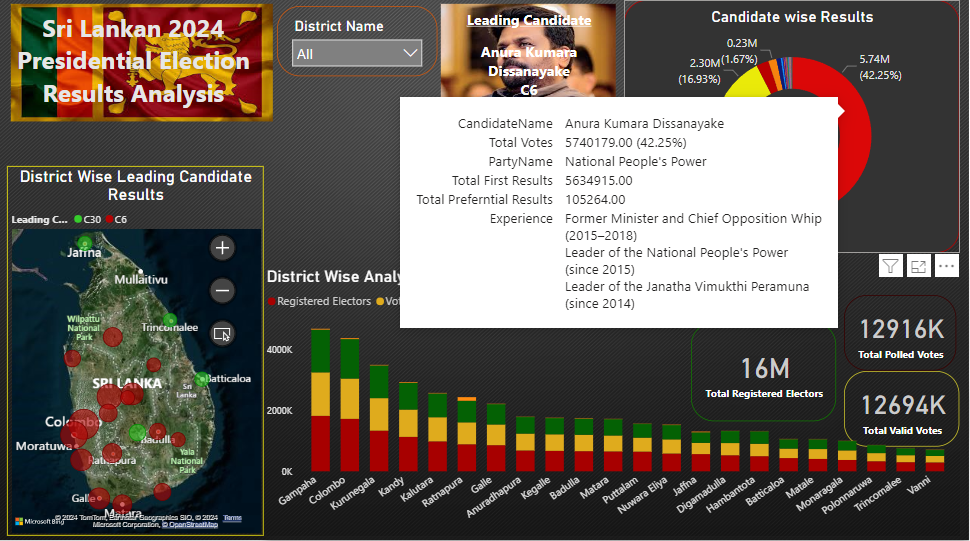
1. **District Slicer**

* Visualization: Slicer for District Name
* Details: Allows filtering of all visuals by District Name, enabling focused analysis on specific regions.
* Purpose: Enhances flexibility in data exploration by allowing users to drill down into district-specific details across all visualizations.

1. **Candidate Vote Share**

* Visualization: Pie Chart
* Details: Displays the percentage share of votes each candidate received, pulling data from CandidateWise\_Results for a comprehensive overview of candidate popularity.
* Purpose: Provides a quick visual of each candidate's share of the total votes, highlighting the leading candidates in terms of vote percentage.

****

****

**References for Data**

<https://elections.gov.lk/si/elections/PRE_2024_Results.html>

<https://results.elections.gov.lk/pre2024/division_results.php?district=Colombo&pd_division=Postal%20Votes>

<https://en.wikipedia.org/wiki/2024_Sri_Lankan_presidential_election>

<https://en.wikipedia.org/wiki/Category:Candidates_in_the_2024_Sri_Lankan_presidential_election>