# I am building a database for federal elections in Australia. Here are my tables:

CREATE TABLE TESTelectionMaster (

    electionSerialNo INTEGER PRIMARY KEY,

    electionDate date,

    electionType VARCHAR(50),

    totalNumDivisions INTEGER,

    totalRegVoters INTEGER,

    lastDateToVoterRegister DATE

    lastDateCandidateNominate DATE,

    lastDateToDeclareResult DATE

);

CREATE TABLE TESTelectoralDivision (

    divisionName VARCHAR(50) PRIMARY KEY,

    totalRegVoters INTEGER,

    currMember VARCHAR(50),

);

CREATE TABLE TESTelectoralDivisionHistory (

    divisionName VARCHAR(50),

    electionSerialNo Integer,

    historicRegVoters INTEGER,

    PRIMARY KEY (divisionName, electionSerialNo)

    );

CREATE TABLE TESTelectionEvent (

    electionEventID VARCHAR,

    totalVoters INTEGER,

    votesCast INTEGER,

    votesReject INTEGER,

    votesValid INTEGER,

    electionSerialNo INTEGER,

    divisionName VARCHAR(50),

    prefWinnerCandidateID VARCHAR(20),

    winnerTally Integer,

    prefLoserCandidateID VARCHAR(20),

    loserTally INTEGER,

    PRIMARY KEY (electionEventID)

);

CREATE TABLE TESTcandidateList (

    candidateID VARCHAR(20),

    candidateName VARCHAR (50),

    contactAddress VARCHAR (50),

    contactPhone INTEGER,

    contactMobile INTEGER,

    contactEmail VARCHAR(20),

    partyCode VARCHAR(60),

    PRIMARY KEY (candidateID)

);

CREATE TABLE TESTcontests (

    electionEventID VARCHAR,

    candidateID VARCHAR (20),

    PRIMARY KEY (electionEventID, candidateID)

);

CREATE TABLE TESTpoliticalParty (

    partyCode VARCHAR(60) PRIMARY KEY,

    partyName VARCHAR(100),

    partyLogo TEXT,

    postalAddress VARCHAR(50),

    partySecretary VARCHAR(50),

    contactPersonName VARCHAR(50),

    contactPersonPhone VARCHAR(20),

    contactPersonMobile VARCHAR(20),

    contactPersonEmail VARCHAR(50)

);

CREATE TABLE TESTvoterRegistry (

    title VARCHAR(5),

    voterID INTEGER,

    firstName VARCHAR(60),

    middleName VARCHAR(60),

    lastName VARCHAR(60),

    gender VARCHAR(60),

    dateOfBirth DATE,

    residentUnitNumber VARCHAR(20),

    residentStreetNumber INTEGER,

    residentStreetName VARCHAR(60),

    residentsuburb VARCHAR(30),

    residentPostcode INTEGER,

    residentState VARCHAR(30),

    postalUnitNumber INTEGER,

    postalStreetNumber INTEGER,

    postalStreetName VARCHAR(60),

    postalSuburb VARCHAR(60),

    postalPostcode INTEGER,

    postalState VARCHAR(60),

    daytimePhone INTEGER,

    mobile VARCHAR(20),

    emailAddress VARCHAR(40),

    divisionName VARCHAR(50),

    PRIMARY KEY (voterID)

);

CREATE TABLE TESTballot (

    ballotID INTEGER,

    electionEventID VARCHAR,

    PRIMARY KEY (ballotID)

);

CREATE TABLE TESTballotPreferences (

    ballotID INTEGER,

    candidateID VARCHAR (20),

    preference INTEGER,

    PRIMARY KEY (ballotID, candidateID)

);

CREATE TABLE TESTissuanceRecord (

    voterID INTEGER,

    electionEventID VARCHAR,

    issueDate DATE,

    ballotIssue Timestamp,

    pollingStation VARCHAR(50),

    PRIMARY KEY (voterID, electionEventID)

);

CREATE TABLE TESTprefCountRecord (

    electionEventID VARCHAR,

    roundNo INTEGER,

    eliminatedCandidateID VARCHAR (20),

    countStatus VARCHAR, --Done, In-progress, complete

    preferenceAggregate INTEGER,

    PRIMARY KEY (electionEventID, roundNo)

);

CREATE TABLE TESTpreferenceTallyPerRoundPerCandidate (

    electionEventID VARCHAR,

    roundNo integer,

    candidateID VARCHAR(20),

    preferenceTally INTEGER, -- Tally in a round.

    PRIMARY KEY (electionEventID, roundNo, candidateID)

);

Here are there constraints:

alter table TESTelectoralDivisionHistory add constraint TESTelectoralDivisionHistoryKeys

    FOREIGN KEY (divisionName)

        REFERENCES TESTelectoralDivision (divisionName),

    FOREIGN KEY (electionSerialNo)

        REFERENCES TESTelectionMaster (electionSerialNo);

alter table TESTcandidateList add constraint TESTcandidateListKeys

    FOREIGN KEY (partyCode)

        REFERENCES TESTpoliticalParty (partyCode);

alter table TESTvoterRegistry add constraint TESTvoterRegistryKeys

    FOREIGN KEY (divisionName)

        REFERENCES TESTelectoralDivision (divisionName);

alter table TESTcontests add constraint TESTcontestsKeys

    FOREIGN KEY (electionEventID)

        REFERENCES TESTelectionEvent (electionEventID),

    FOREIGN KEY (candidateID)

        REFERENCES TESTcandidateList (candidateID);

alter table TESTballot add constraint TESTballotKeys

    FOREIGN KEY (electionEventID)

        REFERENCES TESTelectionEvent (electionEventID);

alter table TESTballotPreferences add constraint TESTballotPreferencesKeys

    FOREIGN KEY (ballotID)

        REFERENCES TESTballot(ballotID),

    FOREIGN KEY (candidateID)

        REFERENCES TESTcandidateList (candidateID);

alter table TESTissuanceRecord add constraint TESTissuanceRecordKeys

    FOREIGN KEY (voterID)

        REFERENCES TESTvoterRegistry (voterID),

    FOREIGN KEY (electionEventID)

        REFERENCES TESTelectionEvent (electionEventID);

alter table TESTprefCountRecord add constraint TESTprefCountRecordKeys

    FOREIGN KEY (electionEventID)

        REFERENCES TESTelectionEvent (electionEventID),

    FOREIGN KEY (eliminatedCandidateID)

        REFERENCES TESTcandidateList (candidateID);

alter table TESTpreferenceTallyPerRoundPerCandidate add constraint TESTpreferenceTallyPerRoundPerCandidateKeys

    FOREIGN KEY (electionEventID)

        REFERENCES TESTelectionEvent (electionEventID),

    FOREIGN KEY (electionEventID, roundNo)

        REFERENCES TESTprefCountRecord (electionEventID, roundNo),

    FOREIGN KEY (candidateID)

        REFERENCES TESTcandidateList (candidateID);

alter table TESTelectionEvent add constraint TESTelectionEventKeys

    FOREIGN KEY (electionSerialNo)

        REFERENCES TESTelectionMaster (electionSerialNo),

    FOREIGN KEY (divisionName)

        REFERENCES TESTelectoralDivision (divisionName),

    FOREIGN KEY (prefWinnerCandidateID)

        REFERENCES TESTcandidateList (candidateID),

    FOREIGN KEY (prefLoserCandidateID)

        REFERENCES TESTcandidateList (candidateID);

Can you write a query for me?

I want you to write an SQL Query to do the following:

1. Write an SQL query to display the total number of voters registered in each of the electoral division.

* Assume that the total number of voters column (in the TESTelectoralDivisoin table) is empty. This data is to be computed by aggregating data from TESTvoterRegistry.
* Your query should produce a report consisting of the electoral division name and total number of voters only.
* The result should be displayed in the descending order of the total number of voters.
* After the Query make an annotation how the query works.

1. Build an index that would help with the query.

* Create the commands to create and delete the index(to test query performance)
* Identify the type of index and columns that are used to build these indexes (justify your design in step by step concise manner).
* Based on previous step, explain why the other types of indexes weren’t chosen (Non-clustered index, B+ Tree, Hash Index, Filtered Index)

# Task: Build a query

The names of candidates that appear on the ballot paper must be in a totally randomized order, i.e. no political party or a group gets any advantage of having their candidates on the top of the list, or candidates with names starting A always appear at the top, etc. Write an SQL query to produce candidate lists for all electoral divisions for the 2022 federal election (election event id: 20220521). The result set must be sorted by electoral division name, and then candidates within each electoral division must be randomized.

I also have created an index.

CREATE INDEX idx\_election\_event\_division ON TESTcandidateList (electionEventID, electoralDivisionName);

# I want to improve the query further as in a real-world scenario where the query could handle millions of records.

Build an index that would help with the query.

* Create the commands to create and delete the index(to test query performance)
* Identify the type of index and columns that are used to build these indexes (justify your design in step by step concise manner).
* Based on previous step, explain why the other types of indexes weren’t chosen (Clustered, Non-clustered index, B+ Tree, Hash Index, Filtered Index, etc.)

# Make the following query:

Registered voters who do not vote at an election receive a penalty (typically, a fine). Write an SQL query, using IN or NOT IN clause, to generate a report that lists the names and addresses of registered voters who did not vote in 2022 general election (election event id: 20220521) and also not voted in 2019 general election (election event id: 20190518).