Some details of the entities and relationships in the reference database model.

1. Election Master: Summarises each year electin
This entity stores the most basic, but important information about elections. Some such information include, total number of registered voters in the election, number of electorates, the date the election is declared, the date of the election, the last date voters can register to vote in a given election, last date the candidate can lodge their candidature applications, the last date to officially declare results. Once implemented, there will be one row per each election (e.g. one row for 2019 election, one row for 2022 election etc) in the corresponding table.

2. Election Event:
On an Election Day, there will be a number of election events, held for each of the electoral divisions. In 2022, there were 151 electoral divisions in Australia. As such, on Saturday 21 May 2022, there were 151 election events happened concurrently, one per each electoral division. In this entity, we store information relevant to each of the election events, for example, number of candidates contesting the election event, number of registered voters for the particular election event, number of votes cast, number of valid votes cast, number of invalid votes cast, current status of the result (say, not counted yet, primary votes counted, pref count on-going, result unofficial, result official, etc), swing based on current and last election, etc.

There is a 1:M relationship between Election Master and Election Event entities. Many elections get stored in 1 Table.

3. Ballot:
This is the entity used to store raw (confidential) data. For each of the election event, there will be hundreds of thousands of ballot papers cast.

Each ballot paper is uniquely identified by a BallotID, but this ID should not have any association with the voter who cast that ballot. Make sure there are no identifications kept once votes are cast, as the confidentiality of ballots is of utmost importance.

In this design Ballots are linked to an election event, and further on, election event is linked to VoterRegistry. However, the second relationship (IssuanceRecord) is a M:N relationship. As such, a Ballot cannot be traced back to a voter.

<- Create Issuance between voters and electionEvent to avoid many to many.

4. Ballot Preferences:
Each ballot is consisted of a number of candidates and the preferences cast by the voter. If you attempt to store this information within the Ballot entity itself, it can be considered as a multi-valued attribute. However, it is important to relate each preference to a candidate. If you use a multi-valued attribute, it is not possible to establish such association. So, in my design, I

A ballot can have many candidates. Candidates can be part of many ballot. We need to insert an assocation relationship. To limit to a single starting value for the entity.

store the preferences as a relationship attribute on the BallotPreferences (M:N) relationship between Ballot and Candidate.

Another way to represent this part is to have two entities (Ballot and BallotPreferences) connect with a 1:N relationship. In such a design, for each ballot row there will be a number of Ballot Preferences rows (as many as the number of candidates contesting that particular election event).

PrefCountRound

5. Preference Count Rounds:
Depending on the number of contesting (say, n), there will be n-2 redistribution rounds to be completed in the counting process. This entity helps us to store information for each of the distribution rounds. They may include: Round number, count status, eliminated candidate, and preference aggregate (total number preferences re-distributed in the round. That aggregate must be the same for all rounds, but, due to counting errors there could be slight differences between rounds).

6. Preferences Tally:
At the end of each round, we have to store preference tally for each candidate. (Of course eliminated candidates' tally is 0). So, the preference tally is stored as a relationship attribute on the PreferenceTallyPerCandidate (M:N) relationship.

Another way to store the tallies is to add a separate entity (say: CandidateTally) and link it with a 1:N relationship with PrefCountRound entity.

<- This refers to Tally

e.g.
Election Event: Chisholm election event on 21 May 2022.

Vote Count Round:
Round 0: count completed, Tally 96518
Round 1: count completed, Tally 96518

<-- Shouldn't there be 11 rows for vote count

...
...
Round 9: count completed, Tally 96518

Note: In this election event, there were 12 candidates. So, 10 counting rounds were required. As a result, there are 10 rows in the Vote Count Round table. Then, for each row in the Vote Count Round, there will be 10 rows in Preference Distribution table, one row per candidate. So, all in all, for 2022 Chisholm election event, there must be 10 rows in Vote Count Rounds table and 120 (10 X 12)rows in Preferences Distribution table.

We count per devission, each round of candidates.
A vote consists of multiple preferences.

Preference Distribution:
Round Number Preference Tally at end of round
Round 0: (first round)
ANTONIE 567

```
DARE            384
GARLAND         38692
KEMPSON         2295
KING            1620
LIU             35038
MURPHY          1590
NEWMAN          12130
STANFIELD       946
TSENG           757
TYRRELL         1377
WHITFIELD       1122

Eliminated in this round: DARE

Round 1:
ANTONIE         586
DARE            0
GARLAND         38751
...
...


Round 9: (final round)
ANTONIE         0
DARE            0
GARLAND         54448
KEMPSON         0
KING            0
LIU             42070
MURPHY          0
NEWMAN          0
STANFIELD       0
TSENG           0
TYRRELL         0
WHITFIELD       0
----------------------
```

7. Two-Candidate-Preferred (Final) Result:
Even if we can retrieve the final result based on data on Preference tally, it is messy and complex to extract the final results from it. So, After all the count rounds are finished, the tallies for the two remaining candidates are stored separately in two relationship attributes.

Instead of