Mapping AEC Entity-Relationship Model into Relational Schema

Step 1: Strong Entities

ElectionMaster (ElectionSerialNo, ElectionDate, Type, TotalNumDivisions, TotalRegVoters, LastDateToVoterRegister, LastDateCandidateNominate, LastDateToDeclareResult)
ElectoralDivision (DivisionName, TotalRegVoters, CurrMember)
ElectionEvent (ElectionEventID, TotalVoters, VotesCast, VotesReject, VotesValid)
Candidate (CandidateID, Name, ContactAddress, ContactPhone, ContactMobile, ContactEmail)
PoliticalParty (PartyCode, PartyName, PartyLogo, PostalAddress, PartySecretary, ContactPersonName, ContactPersonPhone, ContactPersonMobile, ContactPersonEmail)
VoterRegistry (VoterID, FirstName, MiddleNames, LastName, Address, DoB, Gender, ResidentialAddress, PostalAddress, ContactPhone, ContactMobile, ContactEmail)
Ballot (BallotID)

Step 2: Weak Entities
Build a composite PK, by combining partial key with the primary key of its identifying entity.
PrefCountRecord (ElectionEventID*, RoundNo, CountStatus, PreferenceAggregate)
This relies on other table infomation.
Step 3: 1:1 Relationships
None

Step 4: 1:N Relationships.
Export the PK of the one-side as a foreign key into many-side.
ElectionEvent (ElectionEventID, TotalVoters, VotesCast, VotesReject, VotesValid, ElectionSerialNo*, DivisionName*, Two-Candidate-PrefWinnerCandidateID*, WinnerTally, Two-Candidate-PrefLoserCandidateID*, LoserTally)
Candidate (CandidateID, Name, ContactAddress, ContactPhone, ContactMobile, ContactEmail, PartyCode*)
VoterRegistry (VoterID, FirstName, MiddleNames, LastName, Address, DoB, Gender, ResidentialAddress, PostalAddress, ContactPhone, ContactMobile, ContactEmail, DivisionName*)
Ballot (BallotID, ElectionEventID*)
PrefCountRecord (ElectionEventID*, RoundNo, EliminatedCandidateID*, CountStatus, PreferenceAggregate)

Other relations will remain unchanged
ElectionMaster (ElectionSerialNo, ElectionDate, Type, TotalNumDivisions, TotalRegVoters, LastDateToVoterRegister, LastDateCandidateNominate, LastDateToDeclareResult)
PoliticalParty (PartyCode, PartyName, PartyLogo, PostalAddress, PartySecretary, ContactPersonName, ContactPersonPhone, ContactPersonMobile, ContactPersonEmail)
ElectoralDivision (DivisionName, TotalRegVoters, CurrMember)

Step 5: M:N Relationships
Create a new relation with a composite PK, by importing PK from each of the participating relations.
ElectoralDivisionHistory (DivisionName*, ElectionSerialNo*, HistoricRegVoters)
BallotPreferences (BallotID*, CandidateID*, Preference)
Contests (ElectionEventID*, CandidateID*)
IssuanceRecord (VoterID*, ElectionEventID*, IssueDate, Timestamp, PollingStation)
PreferenceTallyPerRoundPerCandidate (ElectionEventID*, RoundNo*, CandidateID*, PreferenceTally)

Step 6: Multi-valued Attributes
None

Step 7: Higher-degree Relationships
None

I feel DivisionName* should create it's own

Final Schema:

ElectionMaster (ElectionSerialNo, ElectionDate, Type, TotalNumDivisions, TotalRegVoters, LastDateToVoterRegister, LastDateCandidateNominate, LastDateToDeclareResult)
ElectoralDivision (DivisionName, TotalRegVoters, CurrMember) <-- Missing current member of parliament
ElectoralDivisionHistory (DivisionName*, ElectionSerialNo*, HistoricRegVoters) <-- Missing date
Removed "TWO"
ElectionEvent (ElectionEventID, TotalVoters, VotesCast, VotesReject, VotesValid, ElectionSerialNo*, DivisionName*, Two-Candidate-PrefWinnerCandidateID*, WinnerTally, Two-Candidate-PrefLoserCandidateID*, LoserTally)
Candidate (CandidateID, Name, ContactAddress, ContactPhone, ContactMobile, ContactEmail, PartyCode*)
Contests (ElectionEventID*, CandidateID*)

PoliticalParty (PartyCode, PartyName, PartyLogo, PostalAddress, PartySecretary, ContactPersonName, ContactPersonPhone, ContactPersonMobile, ContactPersonEmail)
VoterRegistry (VoterID, FirstName, MiddleNames, LastName, Address, DoB, Gender, ResidentialAddress, PostalAddress, ContactPhone, ContactMobile, ContactEmail, DivisionName*)
Ballot (BallotID, ElectionEventID*) 20190518
BallotPreferences (BallotID*, CandidateID*, Preference)
IssuanceRecord (VoterID*, ElectionEventID*, IssueDate, Timestamp, PollingStation)

PrefCountRecord (ElectionEventID*, RoundNo, EliminatedCandidateID*, CountStatus, PreferenceAggregate) This is the status per round
PreferenceTallyPerRoundPerCandidate (ElectionEventID*, RoundNo*, CandidateID*, PreferenceTally) This counts per round.

Removal of results table: Instead of having a results table, we pass the values to the electionEvent table. These values are ElectionSerialNo, DivisionName,

Difference between PrefCountRecord and PreferenceTallyPerRoundPerCandidate: Preference Tally counts the votes a candidate has at each round while preference tally is the progress of counting (in progress, done). Once a round is finished, PrefCountRound is updated, and then the process continues, with new data being recorded in PreferenceTally for the next round. This cycle continues until all counting is complete, and a winner is declared.

Q: Relationship between PreCountRecord and Canddidate? Can't the preference count have 1 or many canididates eliminated? A: We talk in the context of rounds. At each round a candidate is eliminated in preferential voting. They are not elminated all at once.