Short Description of the Seating Algorithm

Raul Persa, Lukas Vogel

January 16, 2016

To guarantee the best possible performance we decided against an iterative method for seat assignment and used the method illustrated in `http://www.wahlrecht.de/bundestag/index.htm`.

The algorithm is fully implemented in SQL and does not depend on any application logic. Parts that cannot be expressed by pure SQL-Queries are implemented as plpgsql-functions. This guarantees a clean interface between the database and the application layer.

The algorithm generates views for a variation of sub-tasks:

- The winners of a direct mandate as `directmandate_winners`
- The parties eligible for seats in the Bundestag as `parties_in_bundestag`
- A Bundesdivisor, specifying the amount of votes needed to gain a seat in the Bundestag as `bundesdivisor`
- The amount of seats for each party in the Bundestag as `total_num_seats` calculated with the Sainte-Lague method and the Bundesdivisor.

A plpgsql-function `find_partydivisor` algorithmically determines a party divisor for each party. The divisor is used to distribute the seats by party back to each bundesland and the correspondent landesliste. A binary search is used for performance reasons.

The view `members_of_bundestag` finally specifies all members of the Bundestag, consisting of:

- All `directmandate_winners`
- All candidates (that have not already won via direct mandate) specified by `remaining_cand_on_ll`. They are a member if they are on a landesliste with a listenplatz lower than the number of remaining seats for a party in a bundesland.

The algorithm needs about 200ms to calculate the whole Bundstag-composition on commodity hardware (Laptop with Intel i5-4210U, Postgres 9.5, default settings).