Datenbanksystem für die Deutsche Bundestagswahl

WS 10/11

Gruppe 3

Felix Kaser, Eva Nießner

Inhalt

1 Informationsstrukturanforderungen 3

1.1 Objektbeschreibungen 3

1.2 Beziehungsbeschreibungen 8

1.3 Prozessbeschreibungen 10

1.4 ER-Diagramm 12

2 Anforderungen 13

2.1 Datenschutzanforderungen 13

2.2 Integritätsbedingungen 13

2.3 Anwendungsfälle 13

3 GUI Mockup 14

4 Implementierung 16

4.1 SQL-Schema 16

4.2 Datenimport 16

4.3 Wahlzettelgenerierung 16

4.4 Sitzplatzberechnung (Q1) 16

4.5 Weitere Queries (Q2 – Q7) 17

4.6 Online Stimmabgabe 18

5 Benchmark 19

5.1 Implementierung 19

5.2 Clients-Laufzeit-Diagramm 20

1 Informationsstrukturanforderungen

1.1 Objektbeschreibungen

Objekt: **Partei**

- Anzahl: 30
- Attribute:
  - Nummer
    - Typ: Integer
    - Anzahl Wiederholungen: 0
    - Definiertheit: 100%
    - Identifizierend: ja
  - Kurzbezeichnung
    - Typ: Char
    - Länge: 50
    - Anzahl Wiederholungen: 0
    - Definiertheit: 80%
    - Identifizierend: nein
  - Name
    - Typ: Char
    - Länge: 100
    - Anzahl Wiederholungen: 0
    - Definiertheit: 100%
    - Identifizierend: nein

Objekt: **Kandidat**

- Anzahl: 1.000
- Attribute:
  - Ausweisnummer
    - Typ: Integer
    - Anzahl Wiederholungen: 0
    - Definiertheit: 100%
    - Identifizierend: ja
  - Vorname
    - Typ: Char
    - Länge: 50
    - Anzahl Wiederholungen: 0
    - Definiertheit: 100%
    - Identifizierend: nein
  - Nachname
    - Typ: Char
    - Länge: 50
    - Anzahl Wiederholungen: 0

        - Definiertheit: 100%
        - Identifizierend: nein

Objekt: **Bundesland**
- Anzahl: 16
- Attribute:
    - Nummer
        - Typ: Integer
        - Anzahl Wiederholungen: 0
        - Definiertheit: 100%
        - Identifizierend: ja
    - Kürzel
        - Typ: Char
        - Länge: 10
        - Anzahl Wiederholungen: 0
        - Definiertheit: 100%
        - Identifizierend: ja
    - Name
        - Typ: Char
        - Länge: 50
        - Anzahl Wiederholungen: 0
        - Definiertheit: 100%
        - Identifizierend: nein

Objekt: **Wahlkreis**
- Anzahl: 300
- Attribute:
    - Nummer
        - Typ: Integer
        - Anzahl Wiederholungen: 0
        - Definiertheit: 100%
        - Identifizierend: ja
    - Name
        - Typ: Char
        - Länge: 100
        - Anzahl Wiederholungen: 0
        - Definiertheit: 100%
        - Identifizierend: nein

Objekt: **Wahlbezirk**
- Anzahl: 100000
- Attribute:
    - Nummer
        - Typ: Integer

        - Anzahl Wiederholungen: 0
        - Definiertheit: 100%
        - Identifizierend: nein

    (Primary Key: Nummer, Wahlkreis)

Objekt: **Landesliste**

- Anzahl: 1.000
- Attribute:
    - Id
        - Typ: Integer
        - Anzahl Wiederholungen: 0
        - Definiertheit: 100%
        - Identifizierend: ja

Objekt: **Wahlberechtigte**

- Anzahl: 70.000.000
- Attribute:
    - Ausweisnummer
        - Typ: Integer
        - Anzahl Wiederholungen: 0
        - Definiertheit: 100%
        - Identifizierend: ja
    - Vorname
        - Typ: Char
        - Länge: 50
        - Anzahl Wiederholungen: 0
        - Definiertheit: 100%
        - Identifizierend: nein
    - Nachname
        - Typ: Char
        - Länge: 50
        - Anzahl Wiederholungen: 0
        - Definiertheit: 100%
        - Identifizierend: nein
    - Geburtsdatum
        - Typ: Datum
        - Anzahl Wiederholungen: 0
        - Definiertheit: 100%
        - Identifizierend: nein
    - Straße
        - Typ: Char
        - Länge: 50
        - Anzahl Wiederholungen: 0
        - Definiertheit: 100%

        - Identifizierend: nein
    - Hausnummer
        - Typ: Integer
        - Anzahl Wiederholungen: 0
        - Definiertheit: 90%
        - Identifizierend: nein
    - Postleitzahl
        - Typ: Integer
        - Anzahl Wiederholungen: 0
        - Definiertheit: 100%
        - Identifizierend: nein
    - Stadt
        - Typ: Char
        - Länge: 50
        - Anzahl Wiederholungen: 0
        - Definiertheit: 100%
        - Identifizierend: nein

(Anmerkung: Vorname, Nachname, Geburtstdatum, Straße, Hausnummer, Postleitzahl und Stadt sind nicht in der Datenbank realisiert, da uns die Informationen nicht vorliegen.)

Objekt: **Wahlergebnis**

- Anzahl: 2060 (2 Vorgängerwahlen)
- Attribute:
    - Id
        - Type: Integer
        - Anzahl Wiederholungen: 0
        - Definiertheit: 100%
        - Identifizierend: ja
    - Wahljahr
        - Typ: Integer
        - Anzahl Wiederholungen: 0
        - Definiertheit: 100%
        - Identifizierend: nein
- besteht aus: Direktergebnis, Listenergebnis (part-of)

Objekt: **Direktergebnis**

- Anzahl: 2000
- Attribute:
    - Id
        - Type: Integer
        - Anzahl Wiederholungen: 0
        - Definiertheit: 100%
        - Identifizierend: ja
    - Stimmenanzahl

        - Typ: Integer
        - Anzahl Wiederholungen: 0
        - Definiertheit: 100%
        - Identifizierend: nein
    - Kandidat
        - Typ: Integer
        - Anzahl Wiederholungen: 0
        - Definiertheit: 100%
        - Identifizierend: nein
    - Partei
        - Typ: Integer
        - Anzahl Wiederholungen: 0
        - Definiertheit: 50%
        - Identifizierend: nein

    (Anmerkung: Kandidat und Partei ist aus Convenience als Beziehung implementiert)

Objekt: **Listenergebnis**

- Anzahl: 60
- Attribute:
    - Nummer
        - Type: Integer
        - Anzahl Wiederholungen: 0
        - Definiertheit: 100%
        - Identifizierend: ja
    - Stimmenanzahl
        - Typ: Integer
        - Anzahl Wiederholungen: 0
        - Definiertheit: 100%
        - Identifizierend: nein
    - Partei
        - Typ: Integer
        - Anzahl Wiederholungen: 0
        - Definiertheit: 100%
        - Identifizierend: nein

    (Primary Key: Partei, Wahlergebnis)
    (Anmerkung: Partei ist aus Convenience als Referenz auf Partei implementiert)

Objekt: **Wahlzettel**

- Anzahl: 70.000.000
- Attribute:
    - Id
        - Type: Integer
        - Anzahl Wiederholungen: 0
        - Definiertheit: 100%

        - Identifizierend: ja
    - Wahlbezirk
        - Typ: Integer
        - Anzahl Wiederholungen: 0
        - Definiertheit: 100%
        - Identifizierend: nein
    - Wahlkreis
        - Typ: Integer
        - Anzahl Wiederholungen: 0
        - Definiertheit: 100%
        - Identifizierend: nein

1.2 Beziehungsbeschreibungen

Beziehung: „ist Mitglied in"
- Anzahl: 1.000
- Beteiligte Objekte:
    - Kandidat
    - Partei

Beziehung: „WK liegt in"
- Anzahl: 300
- Beteiligte Objekte:
    - Wahlkreis
    - Bundesland

Beziehung: „WB liegt in"
- Anzahl: 90.000
- Beteiligte Objekte:
    - Wahlkreis
    - Wahlbezirk

Beziehung: „gehört zu"
- Anzahl: 1.000
- Beteiligte Objekte:
    - Partei
    - Landesliste

Beziehung: „aufgestellt für"
- Anzahl: 1.000
- Beteiligte Objekte:
    - Bundesland
    - Landesliste

Beziehung: „wählt in“

- Anzahl: 70.000.000
- Beteiligte Objekte:
  - Wahlberechtigter
  - Wahlbezirk
- Attribute:
  - gewählt
    - Typ: Boolean
    - Anzahl Wiederholungen: 0
    - Definiertheit: 100%
    - Identifizierend: nein

Beziehung: „hat“

- Anzahl: 600
- Beteiligte Objekte:
  - Wahlkreis
  - Wahlergebnis

Beziehung „Direkterg. part of“

- Anzahl: 2000
- Beteiligte Objekte:
  - Wahlergebnis
  - Direktergebnis

Beziehung „Listenerg. part of“

- Anzahl: 60
- Beteiligte Objekte:
  - Wahlergebnis
  - Direktergebnis

Beziehung: „enthält Erststimme“

- Anzahl: 70.000.000
- Beteiligte Objekte:
  - Kandidat
  - Wahlzettel

Beziehung: „enthält Zweitstimme“

- Anzahl: 70.000.000
- Beteiligte Objekte:
  - Partei
  - Wahlzettel

Beziehung: „abgegeben in“

- Anzahl: 70.000.000
- Beteiligte Objekte:
  - Wahlzettel
  - Wahlbezirk

Beziehung „Str. gehört zu“
- Anzahl: 300
- Beteiligte Objekte:
  - Struktur
  - Wahlkreis

Beziehung: „vertreten auf“
- Anzahl: 1.000
- Beteiligte Objekte:
  - Kandidat
  - Landesliste
- Attribute:
  - Listenplatz
    - Typ: Integer
    - Anzahl Wiederholungen: 0
    - Definiertheit: 100%
    - Identifizierend: nein

Beziehung: „kandidiert in“
- Anzahl: 1.000
- Beteiligte Objekte:
  - Kandidat
  - Wahlkreis

Beziehung: „gewählt in“
- Anzahl: 300
- Beteiligte Objekte:
  - Kandidat
  - Wahlkreis

1.3 Prozessbeschreibungen

Prozess: Wahlauswertung
- Häufigkeit: stündlich
- benötigte Daten:
  - Wahlzettel
  - Kandidat
  - Partei

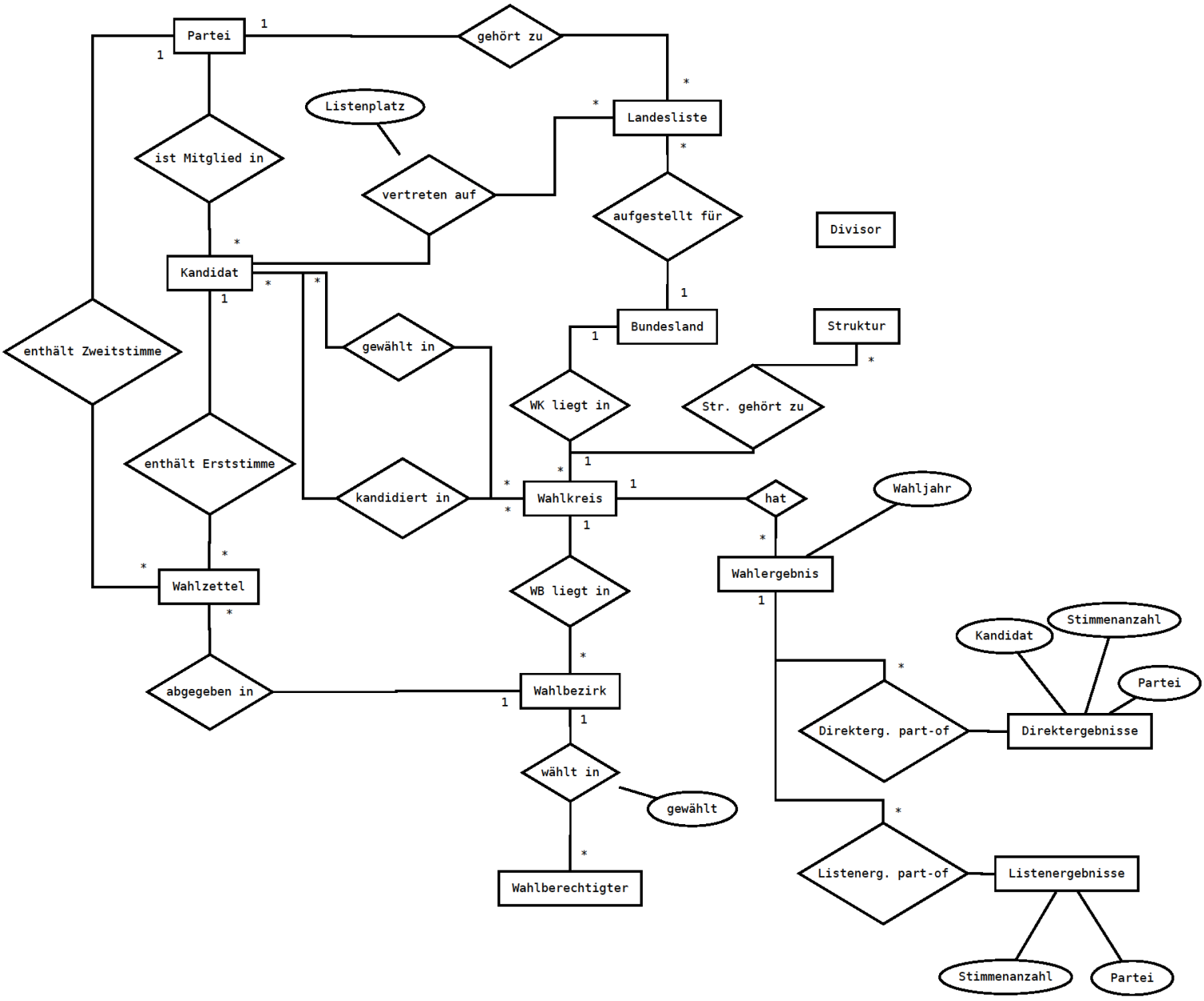
    - Wahlkreis
    - Bundesland
    - Divisor
- Priorität: hoch
- Datenmenge:
    - 70.000.000 Wahlzettel
    - 1.000 Kandidaten
    - 30 Parteien
    - 300 Wahlkreise
    - 16 Bundesländer

Prozess: Wahlbeteiligung

- Häufigkeit: stündlich
- benötigte Daten:
    - Wahlberechtigte
    - Wahlkreis
    - Wahlbezirk
- Priorität: mittel
- Datenmenge:
    - 70.000.000 Wahlberechtigte
    - 300 Wahlkreise
    - 90.000 Wahlbezirke

1.4 ER-Diagramm

(siehe auch bundestagswahl_er_diagramm.pdf)



2 Anforderungen

2.1 Datenschutzanforderungen

Die Stimmenabgabe ist zu 100% von den Wahlberechtigten unabhängig. Pro Wahlberechtigten wird lediglich abgespeichert, ob derjenige gewählt hat. Dabei wird vorausgesetzt, dass die Wahl anonym ist und nicht abgespeichert werden darf, wer für welchen Kandidat bzw. Partei abgestimmt hat.

2.2 Integritätsbedingungen

- Falls ein Kandidat nicht parteilos in der Direktkandidatur-Beziehung auftritt, darf er sich nicht für eine Landesliste einer anderen Partei aufstellen lassen (in der Landeslisten-Beziehung auftreten).
- Gegeben den Fall, dass auch ungültige Wahlzettel in die Datenbank aufgenommen werden, muss die Anzahl der Einträge in der Wahlzettel-Beziehung gleich der Anzahl der Einträge in der Wahlzuteilung-Beziehung mit Attribut gewählt=1 sein.
- Jeder Stimmzettel muss sowohl eine Erst- als auch Zweitstimme enthalten, da entweder Erst- und Zweitstimme gültig oder beide ungültig sind.
- Jeder Kandidat darf nur auf einer Landesliste vertreten sein. Jeder Listenplatz einer Landesliste darf nur einmal besetzt sein.
- Jede Partei darf maximal eine Liste pro Bundesland haben.
- Ein Kandidat darf nur für einen Wahlkreis direkt kandidieren.

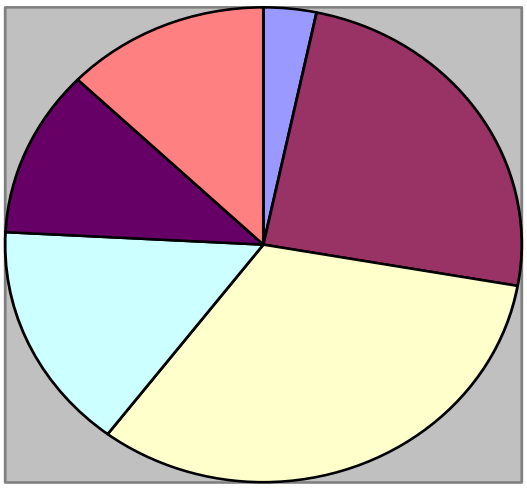
2.3 Anwendungsfälle

1. Sitzverteilung im Bundestag als Tortendiagramm und Tabelle nach Partei mit Anzahl der Sitze
2. Mitglieder des Bundestages als Tabelle inkl. Partei, Wahlkreis für Direktkandidatur (falls vorhanden), Platz und Bundesland für Landeslistenkandidatur (falls vorhanden)
3. Wahlkreisübersicht: Auswahl des Bundeslands und des Wahlkreises; Anzeige des gewählten Direktkandidaten, der prozentualen Stimmverteilung als Tortendiagramm, der absoluten Stimmverteilung als Balkendiagramm und der prozentualen Stimmentwicklung im Vergleich zur letzten Wahl
   a) Live-Berechnung: Übersicht wird aus Einzelstimmen berechnet
   b) keine Live-Berechnung: Übersicht wird aus aggregierten Ergebnissen berechnet
4. Wahlkreissieger jeweils für Erststimmen (inkl. Direktkandidat, Partei, Stimmen) und Zweitstimmen (inkl. Partei, Stimmen) als Tabelle
5. Überhangmandate: Auswahl des Bundeslands auf einer Deutschlandkarte; Anzeige der Anzahl der Überhangmandate mit Einfärbung; Anzeige der Partei und Anzahl der Überhangmandate als Tabelle
6. Top 10 der knappsten Sieger: Auswahl der Partei, Anzeige der 10 knappsten Wahlkreissieger bzw. der knappsten Niederlage
7. Online Stimmabgabe: Der Wahlberechtigte darf über Angabe seiner Personalausweisnummer aus einer Liste von Kandidaten und Parteien seine Erst- und Zweitstimme abgeben.

3 GUI Mockup

Die GUI wird mit JSPs realisiert. Auf der rechten Seite befindet sich die Navigation. Bei Auswahl eines Navigationspunktes wird der entsprechende Inhalt dynamisch in die JSP geladen. Zwei exemplarische Navigationspunkte sind hier aufgeführt:

Beispiel Sitzverteilung:



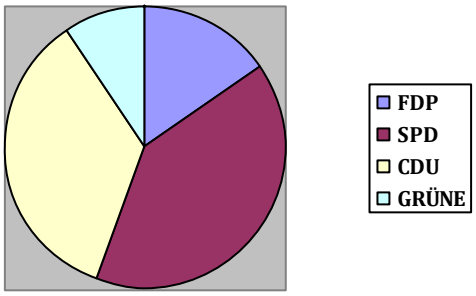
- Sitzverteilung
- Mitglieder des Bundestages
- Wahlkreisübersicht
- Wahlkreissieger
- Überhangmandate
- Top 10 der knappsten Sieger
- Online Stimmabgabe

Partei	Sitze
CDU	194
SPD	146
FDP	93
DIE LINKE	76
GRÜNE	68

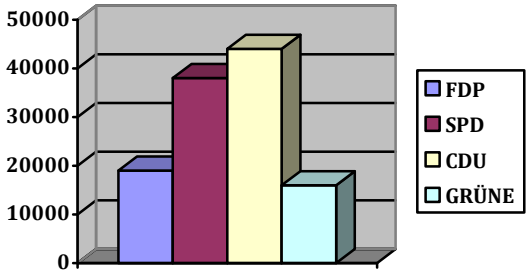
Beispiel Wahlkreisübersicht:

Bundesland: Hessen, Wahlkreisnummer: 170

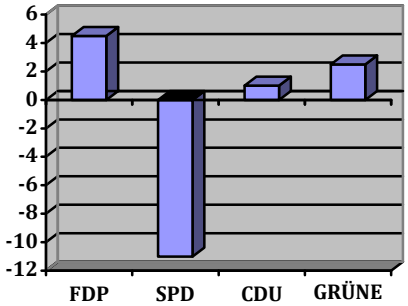
Direktkandidat: Michael Roth (SPD)
Wahlbeteiligung: 0,74%



Stimmverteilung (absolut)



Stimmentwicklung (prozentual)



Sitzverteilung
Mitglieder des Bundestages
Wahlkreisübersicht
Wahlkreissieger
Überhangmandate
Top 10 der knappsten Sieger
Online Stimmabgabe

4 Implementierung

4.1 SQL-Schema

Das aktuelle Datenbankschema befindet sich in *schema.sql*. Durch Ausführung der in dieser Datei programmierten Statements wird die Datenbank zurück- und neu aufgesetzt. Hierbei werden die Tabellen erstellt.

4.2 Datenimport

Der Import der Stammdaten erfolgt in zwei Schritten: Zunächst werden die flachen CSV-Dateien eingelesen. Die darin enthaltenen Daten werden zum Import in die Datenbank vorbereitet, indem für die Datensätze SQL-Inserts generiert werden. Die SQL-Inserts werden in gesonderten Dateien gespeichert. Die Datentransformation findet im Python-Skript *datenbank.py* statt. Um einen optimalen Entwicklungsprozess zu ermöglichen, wurden die beiden Shellskripte *setupdb.sh* sowie *setupdb.bat* implementiert, die die Datenbank inklusive Stammdaten zurücksetzen und die aktuellste Version des Schemas sowie der Stammdaten einspielen.

4.3 Wahlzettelgenerierung

Ebenfalls in Python ist ein Datengenerator für die Einzelstimmen auf Wahlzetteln geschrieben. In *wahlzettel_generierung.py* kann unter Angabe eines Bundeslandes für dieses Wahlzettel generiert werden, sodass die Sitzplatzberechnung basierend auf den generierten Wahlzetteln die aggregierten Ergebnisse berechnen. Dabei werden Erst- und Zweitstimmen in Queues gesammelt und anschließend in Paaren Wahlzetteln zugeordnet.

4.4 Sitzplatzberechnung (Q1)

<u>Pseudocode:</u>

```
anzahl_stimmen = SELECT COUNT(*) FROM Wahlzettel
anzahl_sitze = 598 - (anzahl direktmandate ohne partei *)
hare_quote = anzahl_stimmen/anzahl_sitze

// * = Direktmandate ohne Partei bzw. Partei unter Sperrklausel (unter 5% und keine
3 Direktmandate und keine Minderheitenpartei)
// pro Partei bundesweite Stimmenanzahl herausfinden und Sainte-Lague berechnen
/*  Sainte-Lague mit Höchstzahl:
      Neue Tabelle erstellen mit Tmp = (partei, durchgang, stimmenanzahl) und für
      jede Partei anz.stimmen/(0.5, 1.5, 2.5, ...) berechnen (Metrik: bis
      Stimmenanzahl < Hare Quote); dann nach Stimmenanzahl sortieren (desc) und die
      ersten 598 auswählen (Limit?). Danach kann man die Parteien zählen und die
      Anzahl der Sitze für jede Partei herausfinden
*/
```

```
foreach partei
      bundes_stimmen = SELECT COUNT(*) FROM Wahlzettel w WHERE w.partei = partei

      // Parteien, die die Sperrklausel nicht erfüllen, überspringen
      if (sperrklausel(bundesstimmen))
            continue;

      durchgang = 0.5
      durchgangs_stimmen = bundes_stimmen / durchgang
      while bundes_stimmen > hare_quote
            INSERT INTO Tmp VALUES (partei, durchgang, bundes_stimmen)
            durchgang = durchgang + 1
            durchgangs_stimmen = bundesstimmen / durchgang

// Sitzverteilung berechnen
SELECT Partei, COUNT(durchgang) as Sitze
FROM (SELECT * FROM Tmp ORDER BY stimmenanzahl DESC LIMIT 0, <anz zu verteilende
sitze>)
GROUP BY Partei
```

Beschreibung:

Zunächst werden die Wahlkreisergebnisse auf Bundesland-Ebene berechnet, danach deutschlandweit. Daraufhin wird das Divisorverfahren angewandt um die Sitzplätze zu berechnen. Danach wir das Ergebnis unter Berücksichtigung der Überhangmandate auf Bundesland-Ebene herunter gebrochen, um die gewählten Landeslistenkandidaten zu berechnen.

Das zugehörige SQL-Skript befindet sich in *sitzplatzberechnung.sql*.

4.5 Weitere Queries (Q2 – Q7)

In diesem Dokument werden nur die Beschreibungen der restlichen Queries aufgelistet. Die Queries selbst sind in SQL Form in der Datei *statements.sql* zu finden.

4.5.1 Q2: Mitglieder des Bundestages

Diese Abfrage ist analog zu Q1, der Berechnung der Sitzverteilung im Bundestag.

4.5.2 Q3: Wahlkreisübersicht

In Q3 werden in mehreren SQL Abfragen die Wahlbeteiligung, gewählten Direktkandidaten, Anzahl an Stimmen pro Partei und die Entwicklung der Stimmen im Vergleich zum Vorjahr berechnet. Alle Berechnungen erfolgen auf Wahlkreisebene und verwenden aggregierte Wahlkreisergebnisse. „Anzahl der Stimmen pro Partei“ bedeutet hier „Anzahl Zweitstimmen pro Partei“.

4.5.3 Q4: Wahlkreissieger

Q4 berechnet auf Wahlkreisebene die Siegerparteien (bezüglich Erst- und Zweitstimmen). Auch hier werden nur aggregierte Ergebnisse verwendet.

4.5.4 Q5: Überhangmandate

Die Überhangmandatberechnung ist ein Teil der Sitzverteilungsberechung. Im Grunde verläuft die Berechnung analog zu Q1: Es wird berechnet wie viele Sitze einer Partei auf Bundeslandebene zustehen und vergleicht das Ergebnis mit der Anzahl der Direktmandate, die eine Partei in dem Bundesland bereits hat.

4.5.5 Q6: Knappste Sieger

Bei Q6 wurde aus Komplexitätsgründen ein Teil der Berechnung in Java ausgelagert. Die SQL Query berechnet eine Liste aller Wahlkreissieger mit dem dazugehörigen Abstand zum zweiten Sieger im Wahlkreis (inklusive Zusatzinformationen wie Name und Partei). Eine separate Anfrage berechnet für jene Parteien, die in keinem Wahlkreis gewonnen haben, eine Liste der Kandidaten, die in einem Wahlkreis die knappsten Verlierer waren (also bester der Partei im Wahlkreis, aber doch nicht gewonnen) zusammen mit dem Abstand zum Wahlkreissieger (wieder mit Informationen zum Sieger). In Java werden nun pro Partei die ersten 10 Sieger ausgewählt. Falls es nicht 10 knappste Sieger sind, wird die Liste noch mit den knappsten Verlierern gefüllt.

4.5.6 Q7: Wahlkreisübersicht – live Berechnung

Diese Berechnung verläuft analog zu Q3, mit dem Unterschied dass auf die „echten“ Wahlzettel zugegriffen wird und nicht auf Aggregate. Daher ergibt sich bei dieser Abfrage auch ein größerer Zeitaufwand.

4.6 Online Stimmabgabe

Die Stimmabgabe läuft in folgenden Schritten ab:

Zur Identifizierung muss der Wahlberechtigte seine Personalausweisnummer angeben. Diese wird daraufhin auf Gültigkeit geprüft, d.h. ob die Personalausweisnummer in der Datenbank vorhanden ist und ob der Wahlberechtigte nicht schon gewählt hat. Durch diesen Check wird der Schutz vor Wahlbetrug realisiert. Denkbar als zusätzlicher Check wäre die Angabe des vollständigen Namens des Wählers zusammen mit der Personalausweisnummer, was die Identität des Wahlberechtigten zusätzlich überprüfen würde. (Dies konnte aus Mangel an Personendaten nicht realisiert werden.)

Nach Authentifizierung wird dem Wähler der Stimmzettel angezeigt. Der Wähler gibt eine Erststimme und Zweitstimme an und klickt „Stimme abgeben“. Hierbei ist ein Vorteil der online Stimmabgabe die Möglichkeit, ungültige Stimmzettel zu verbieten: Ein weiterer Check stellt nämlich sicher, dass genau eine Erststimme und genau eine Zweitstimme abgegeben wird.

Der Datenschutz wird gesichert, indem in zwei unabhängigen Methoden der Wahlberechtigte als „gewählt“ markiert wird und die Stimmen eingetragen werden.

5 Benchmark

5.1 Implementierung

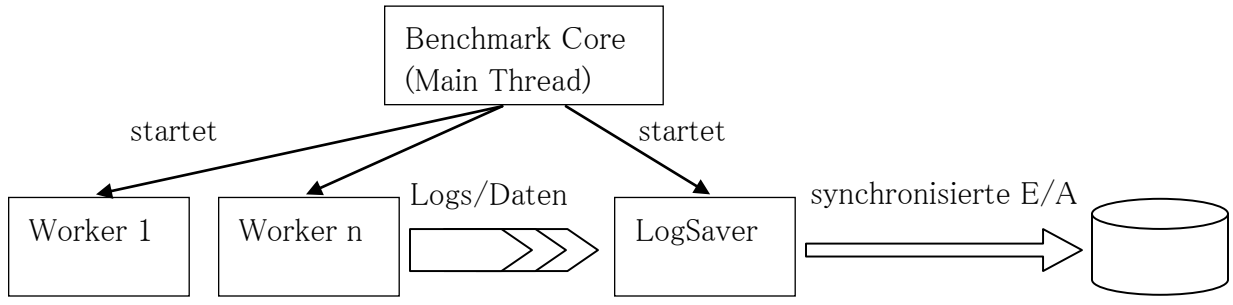
Der Benchmarkclient ist in Java implementiert. Er lässt sich nach Laufzeit, Anzahl an emulierten Clients und Workload-Mix konfigurieren. Dabei wird ein Kartendeck für jeden Client simuliert, das entsprechend des Workload-Mix Karten mit den aufzurufenden URLs enthält. Jedes Ziehen einer Karte entspricht einem Aufruf der Query. Ist das Kartendeck leer, wird solange ein neues erzeugt, bis die konfigurierte Laufzeit abgelaufen ist.

Der Workloadmix kann unter Angabe von URL und Aufrufhäufigkeit in Prozent als Kommandozeilenargument konfiguriert werden. Wird keine URL angegeben, werden die Queries 1 bis 6 ausgeführt.

Die Benchmarkergebnisse werden von einem parallel laufenden Thread (*LogSaver.java*) gesammelt und formatiert in eine CSV-Datei gespeichert. Die Wartezeiten der einzelnen Clients werden emuliert durch eine Normalverteilung mit vorgegebenem Mittelwert.

Über die Kommandozeilenargumente lassen sich genannte Parameter setzen:

Name	**Beschreibung**
--workers	Anzahl der emulierten Clients
--file_path	Pfad zu der CSV-Datei, in der die Ergebnisse ausgegeben werden
--runtime	Laufzeit des Benchmarks
--wait	Mittelwert der Wartezeit zwischen Aufrufen
--url	Eine aufzurufende URL (<url>,<percent>)



5.2 Clients-Laufzeit-Diagramm

Die folgenden Diagramme wurden mit einer mittleren Wartezeit von 0 Millisekunden erstellt.

Die komplette Auswertung befindet sich in *Auswertung.xlsx*.