“Schema” in Neo4J

In Neo4j, we do not really have a schema, as we follow a schema-less model, where the database structure is flexible and can evolve over time. Instead of enforcing a rigid schema, Neo4j allows nodes and relationships to have variable sets of properties and labels. This flexibility means that data modeling is driven by application needs and use cases rather than predefined schema constraints. This approach provides agility in handling diverse and changing data, making Neo4j well-suited for scenarios where the data structure is dynamic or not fully known in advance.

Anyway, we load the data in a standardized way and this is how the data and its relationships look like right now:

ARTICLE

For this node type, we imported articles from the German Grundgesetz (GG), which we got from https://www.gesetze-im-internet.de/gg/ in XML format. We extracted when articles cite other articles.

- <elementId>: Automatically generated by Neo4j.
- <id>: Automatically generated by Neo4j.
- **citing_cases:** How many unique cases cited this article.
- **number:** Article number (alphanumeric).
- **text:** The article’s text content.
- resource: The underlying law book
- **total_case_citations:** How many citations this article received from cases in total (one case can cite the same article multiple times).

Example:

<elementId>: 4
<id>: 4
citing_cases: 272
number: 5
text: (1) Jeder hat das Recht, seine Meinung in Wort, Schrift und Bild frei zu äußern und zu verbreiten und sich aus allgemein zugänglichen Quellen ungehindert zu unterrichten. Die Pressefreiheit und die Freiheit der Berichterstattung durch Rundfunk und Film werden gewährleistet. Eine Zensur findet nicht statt. (2) Diese Rechte finden ihre Schranken in den Vorschriften der allgemeinen Gesetze, den gesetzlichen Bestimmungen zum Schutze der Jugend und in dem Recht der persönlichen Ehre. (3) Kunst und Wissenschaft, Forschung und Lehre sind frei. Die Freiheit der Lehre entbindet nicht von der Treue zur Verfassung.
resource: GG
total_case_citations: 2602

CASE

The case nodes consist of cases from the Bundesverfassungsgericht (BVerfGE) which we obtained from here: https://zenodo.org/records/10369205. These data were enriched with metadata from here: https://zenodo.org/records/10378324. Cases **refer to** other cases and articles.

- <elementId>: Automatically generated by Neo4j.
- <id>: Automatically generated by Neo4j.
- **bverfge_references:** References extracted from this case to other cases of the Bundesverfassungsgericht (BVerfGE).
- **citing_cases:** How many unique cases cited this case.
- **decision_type:** How the decision has been classified [Beschluss, Gutachten, Urteil]
- **facts:** The facts of this case (German: Tatbestand / Sachverhalt).
- **gg_references: :** References extracted from this case to articles of the Grundgesetz (GG).
- **headnotes:** The headnotes of a case are concise summaries of the legal principles and key points addressed in the decision, typically providing a quick reference to the case's core issues and rulings.
- **id:** “Aktenzeichen” from case metadata.
- **judgment:** The decision made.
- **number:** How this case is referenced by literature, extracted from the “fundstelle” metadata and standardized. Normally, we have a span inside "BVerfGE 151, 101-151", we removed the last digit, whitespace and the hyphen, resulting in this format "BVerfGE151,101".
- **panel_of_judges:** Which judicial panel ruled this decision.
- **reasoning:** The reasoning behind the decision.
- **total_case_citations:** How many citations this case received from other cases in total (one case can cite the same case multiple times).
- **year:** Year of the judgment (for composing a date).

Example (facts and reasoning were shortened):

<elementId>: 201
<id>: 201
bverfge_references: BVerfGE18,85;BVerfGE5,9;BVerfGE10,234;BVerfGE8,274;BVerfGE15,275;BVerfGE24,33;BVerfGE10,264
citing_cases: 3
decision_type: Beschluss
facts: Gründe: A.-I. 1. Der Beschwerdeführer ist durch Urteil des Amtsgerichts Ingolstadt vom 14. Mai 1962 wegen Betruges in zwei Fällen zu einer Gesamtstrafe von 3 Monaten Gefängnis verurteilt worden. […] Dagegen vermöge die Rechtsordnung keine Regeln dafür aufzustellen, in welchen Fällen es ermessensmißbräuchlich, also rechtswidrig wäre, die rechtmäßige Ordnung durch einen ablehnenden Gnadenbescheid aufrechtzuerhalten.
gg_references: 19;19;19;101;19;19;101;103;19;19;19;60;60;60;19;19;60;19;1;19;19;19;19;19;19
headnotes: Zur Frage der Justitiabilität von Gnadenentscheidungen (Art. 19 Abs. 4 GG).
id: 2 BvR 552/63
judgment: Die Verfassungsbeschwerde wird zurückgewiesen.
number: BVerfGE25,352
panel_of_judges: II

reasoning: II. Gegen die Zulässigkeit der Verfassungsbeschwerde bestehen keine Bedenken. Zwar macht der Beschwerdeführer auch die Verletzung der Art. 20 und 28 GG, die in Art. 93 Abs. 1 Nr. 4 a GG, § 90 Abs. 1 BVerfGG nicht aufgeführt sind, geltend. […] Die angefochtene Entscheidung hätte deshalb nach der hier vertretenen Auffassung aufgehoben werden müssen, weil sie Art. 19 Abs. 4 GG verletzt. (gez.) Seuffert Der Richter Dr. Leibholz ist an der Unterschrift verhindert. Seuffert Geller Dr. v. Schlabrendorff Dr. Rupp Dr. Geiger Dr. Kutscher Dr. Rinck
total_case_citations: 6
year: 1969

TOC

We used two textbooks “Grundrechte Klausur- und Examenswissen” and “Grundrechte Klausur- und Examensfälle”, which contain references to both the cases and articles. We parsed the table of contents and have extracted it as a hierarchy tree, forming the TOC nodes. A TOC node of a lower level is connected to a TOC node of a higher level, as it is a part of it. The TOC nodes have Reference nodes (see below) attached to them, whenever in the respective section a case or article has been referenced.

- <elementId>: Automatically generated by Neo4j.
- <id>: Automatically generated by Neo4j.
- **id:** A representation of the hierarchy tree (= identifying information) of the current node.
- **next_toc:** A representation of the hierarchy tree (= identifying information) of the parent node (higher in the hierarchy).
- **text:** The section heading.
- **weblink:** url to wikibook section

Example:

<elementId>: 5459
<id>: 5459
id: Grundrechte-Klausur-und-Examenswissen > Abschnitt 2 Aufbau der Prüfung eines Freiheitsgrundrechts > § 5 Schutzbereich & Eingriff
next_toc: Grundrechte-Klausur-und-Examenswissen > Abschnitt 2 Aufbau der Prüfung eines Freiheitsgrundrechts
text: § 5 Schutzbereich & Eingriff
weblink: https://some-link-to-online-version.com

REFERENCE

These are the references extracted from the textbooks. They are connected to the TOC node that they are part of and to the respective article or case node that they mention.

<elementId>: Automatically generated by Neo4j.
<id>: Automatically generated by Neo4j.
context: One or more sentences around the reference to better understand the context in which the legal text was mentioned.
id: A representation of the hierarchy tree (= identifying information) of the current node inside the TOC hierarchy.
resource: Which type of law or court the reference mentions.

text: Extracted reference text.

Example:

<elementId>: 5533
<id>: 5533
context: Eigentum, Erbrecht und Sozialisierung in den Art. 14, 15 GG finden sich in einem Kapitel.
id: Grundrechte-Klausur-und-Examenswissen > Vorwort der Herausgeber:innen
next_toc: Grundrechte-Klausur-und-Examenswissen
resource: GG
text: Art. 14, 15 GG

NAME

A well-known and important case is named in a certain way, which may be used for search later.

- <elementId>: Automatically generated by Neo4j.
- <id>: Automatically generated by Neo4j.
- **id:** The number of the case.
- **short:** The name given for the case or article.
- type: Article or Case

Example:

<elementId>: 3683
<id>: 3683
id: BVerfGE100,226
short: Denkmalschutz

RELATIONSHIPS

Diving deeper into relationships, we have the following ones (with relationship properties in brackets):

- CITES
  - Article → Article
- IS_NAMED
  - Case → Name
  - Article → Name
- MENTIONS
  - Reference → Article
  - Reference → Case
- PART_OF
  - Reference → TOC
  - TOC → TOC

- REFERS_TO (number of references = how often the case referenced the target node)
    - Case → Case
    - Case → Article