Regestenmodellierung im Graphen

Contents

1	Inh	alt	2
2	Wie 2.1 2.2 2.3	e kommen die Regesten in den Graphen? Import mit dem LOAD CSV-Befehl	2 2 4 5
3	Ers	tellen der Ausstellungsorte	12
		3.0.1 Herrscherhandeln in den Regesta Imperii	14
		3.0.2 Zitationsnetzwerke in den Regesta Imperii	15
4	Imp	oort der Registerdaten in die Graphdatenbank	16
	4.1	Vorbereitung der Registerdaten	16
	4.2	Import der Registerdaten in die Graphdatenbank	18
	4.3	Die Hierarchie des Registers der Regesten Kaiser Friedrichs III.	20
5	Aus	swertungsperspektiven	21
	5.1	Personennetzwerke in den Registern	21
		5.1.1 Graf Robert II. von Flandern in seinem Netzwerk	21
		5.1.2 Graf Robert II. von Flandern und Herzog Heinrich von	
		Niederlothringen	22
	5.2	Herrscherhandeln ausgezählt	29
	5.3	Herrscherhandeln pro Ausstellungsort ausgezählt	29
	5.4	Herrscherhandeln und Anwesenheit	30
	5.5	Regesten 200 km rund um Augsburg	36
	5.6	Welche Literatur wird am meisten zitiert	37

7 Zusammenfassung

38

1 Inhalt

 $\{:.no_toc\}$

• Will be replaced with the ToC, excluding the "Contents" header {:toc}

2 Wie kommen die Regesten in den Graphen ?

Die Regesta Imperii Online basieren momentan auf dem Content-Managment-System Typo3 welches auf eine mysql-Datenbank aufbaut. In der Datenbank werden die Regesteninformationen in verschiedenen Tabellen vorgehalten. Die Webseite bietet zum einen die Möglichkeit, die Regesten über eine REST-Schnittstelle im CEI-XML-Format oder als CSV-Dateien herunterzuladen. Für den Import in die Graphdatenbank bietet sich das CSV-Format an.

In der CSV-Datei finden sich die oben erläuterten einzelnen Elemente der Regesten in jeweils eigenen Spalten. Die Spaltenüberschrift gibt Auskunft zum Inhalt der jeweiligen Spalte.

2.1 Import mit dem LOAD CSV-Befehl

Mit dem Befehl LOAD CSV können die CSV-Dateien mit den Regesten in die Graphdatenbank importiert werden. Hierfür muss die Datenbank aber Zugriff auf die CSV-Daten haben. Dies ist einerseits über den im Datenbankverzeichnis vorhandene Ordner import oder über eine URL auf die CSV-Datei möglich. Da sich die einzelnen Zugriffswege auf den import-Ordner von Betriebssystem zu Betriebssystem unterscheiden wird hier beispielhaft der

¹Verwendet wird die Graphdatenbank neo4j. Die Community-Edition ist kostenlos erhältlich unter https://www.neo4j.com.

	~ =	7	00% -	€	% .0↓	.0	123	- A	rial	,		10	~	В	Ι	S	<u>A</u>	\rightarrow .	⊞ ∃	= -	≡	- <u>↓</u>	. *
fx	uid																						
	Α	В		С		D		Е		F		G			Н		1		J		K		
1	uid	start_date	end	date	identific	er		regnu	persi	stent_i	d title			date	string	loc	ality_stri	n sur	nmary	arc	hival_	histo	ry
2	20053	1050-11-1	1 1050)-11-11	RI III,2	3 n.	1	1	1050	-11-11	Hei	nrich	IV.	1050	Nove	m (G	oslar?)	Hei	inrich w	rc He	rim. A	ug. 10	05
3	20054	1050-12-2	25 1050)-12-25	RI III,2			2	1050	-12-25	Hei	nrich	IV.	1050	Dezei	тРö	hlde	Hei	inrichs \	/a He	rim. A	ug. 10	05
4	20055	1051-02-0			RI III,2			3	1051	-02-02	Hei	nrich	IV.	1051	Febru	a Au	gsburg		inrich fo				
5	20056	1051-03-0	04 1051	1-03-04	RI III,2	3 n.	4	4	1051	-03-04	He	nrich	IV.	1051	März	(4 Sp	eyer	Hei	inrich fo	lg He	rim. A	ug. 10	05
6	20057	1051-03-3	31 1051	1-03-31	RI III,2	3 n.	5	5	1051	-03-31	He	nrich	IV.	1051	März	3 Kö	ln	Hei	inrich er	nı He	rim. A	ug. 10	05
7	20058	1051-11-1	2 1051	I-11-12	RI III,2	3 n.	6	6	1051	-11-12	Hei	nrich	IV.	1051	Nove	m Re	gensburg	y Wä	hrend o	es Ui	ngarnf	eldzu	ıge
8	20059	1051-12-	25 1051	I-12-25	RI III,2	3 n.	7	7	1051	-12-25	He	nrich	IV.	1051	Dezei	m Go	slar	We	ihnacht	sf He	rim. A	ug. 10	05
9	20060	1052-03-	05 1052	2-03-05	RI III,2	3 n.	8	8	1052	-03-05	Hei	nrich	IV.	1052	März	5 Ka	iserswer	t Hei	inrich in	te <lir< td=""><td>k http</td><td>://opa</td><td>ac.</td></lir<>	k http	://opa	ac.
10	20061	1052-03-2	27 1052	2-03-27	RI III,2	3 n.	9	9	1052	-03-27	Hei	nrich	IV.	1052	März	2 Go	slar	Hei	inrichs v	viı <lir< td=""><td>k http</td><td>://opa</td><td>ac.</td></lir<>	k http	://opa	ac.
11	20062	1052-06-0	7 1052	2-06-07	RI III,2	3 n.	10	10	1052	-06-07	He	nrich	IV.	1052	Juni (7 Zü	rich	Hei	inrich fo	lg He	rim. A	ug. 10	05
12	20063	1052-09-	1052	2-09-30	RI III,2	3 n.	11	11	1052	-09-00	He	nrich	IV.	1052	(Sept	е –		Hei	inrichs E	Elt He	rim. A	ug. 10	05
13		1052-12-2		2-12-25	RI III,2	3 n.	12	12	1052	-12-25	He	nrich	IV.	1052	Dezei	m Wo	orms	We	ihnacht	sf He	rim. A	ug. 10	05
14		1053-11-0						13	1053	-11-03_	He	nrich	IV.		(vor d				inrich w			_	
15		1053-12-2			RI III,2					-12-25					Deze				inrich er			_	
16		1054-04-0			RI III,2					-04-03	_				April (inrich fo	_		_	
17		1054-05-2			RI III,2					-05-29							oslar?)		inrich in				
18		1054-05-		1-05-31						-05-31	_				Mai 3				inrich in				
19		1054-07-		1-07-17						-07-17 _.	_				Juli 1				inrich w				
20		1054-07-		1-07-17						-07-17 _.	_						achen?)		mutlich				
21		1054-11-		1-11-17						-11-17_	_				Nove				nrich in				
22		1054-11-		1-11-17						-11-17_					Nove				inrichs v				
23		1054-12-			RI III,2					-12-25	_								ihnacht				
24		1055-01-								_	_						edlinburg	-					
25		1055-03-			RI III,2												gensburg						
26		1055-03-															gensburg						
27	20078	1055-03-	12 1055	5-03-12	RI III,2	3 n.	26	26	1055	-03-12	He	nrich	IV.	1055	März	1 "Ut	ingen" (?	') Hei	inrichs v	viı <lir< td=""><td>ık http</td><td>://opa</td><td>ac.</td></lir<>	ık http	://opa	ac.

Figure 1: Regesten als CSV-Datei

Import mit einer URL vorgestellt. Hierfür wird ein Webserver benötigt, auf den man die CSV-Datei hochlädt und sich anschließend die Webadresse für den Download der Datei notiert.

2.2 Google-Docs für den CSV-Download

Da viele aber keinen Zugriff auf einen eigenen Webserver haben wird hier auch der Download der CSV-Dateien über Google-Docs erklärt. Zunächst benötigt man hierfür einen Google-Account. Anschließend öffnet man Google-Drive und erstellt dort eine leere Google-Tabellen-Datei in der man dann die CSV-Datei hochladen und öffnen kann.

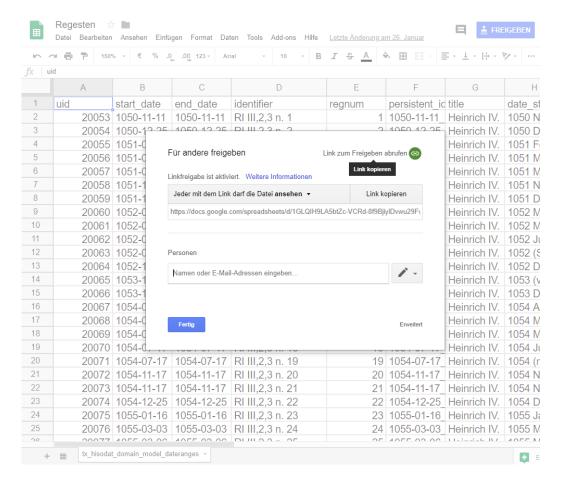


Figure 2: Freigabe der Datei zum Ansehen für Dritte!

Wichtig ist nun, die Datei zur Ansicht freizugeben (Klick auf Freigeben oben rechts im Fenster dann Link zum Freigeben abrufen und anschließend bestätigen). Jetzt ist die CSV-Datei in Google-Docs gespeichert und kann von Dritten angesehen werden. Für den Import in die Graphdatenbank benötigen wir aber einen Download im CSV-Format. Diesen findet man unter Datei/Herunterladen als/Kommagetrennte Werte.csv aktuelles Tabellenblatt.

Damit lädt man das aktuelle Tabellenblatt als CSV runter. Nach dem Download muss man nun im Browser unter Downloads den Download-Link der Datei suchen und kopieren.

2.3 Regestenmodellierung im Graphen

Nachdem mit dem LOAD CSV-Befehl ein erster Schritt des Import erfolgreich abgeschlossen ist, wird nun erst mal das Modell der Regesten im Graphen vorgestellt. In der nächsten Abbildung wird das Modell des Regests im Graphen abgebildet.

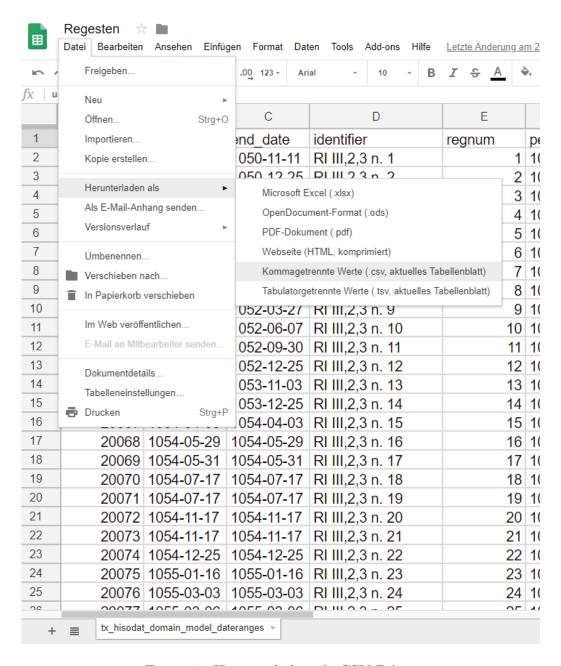


Figure 3: Herunterladen als CSV-DAtei

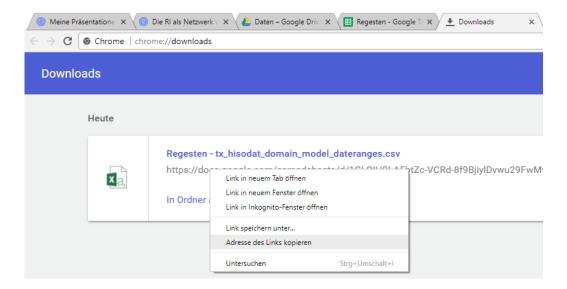


Figure 4: Download-Link der CSV-Datei



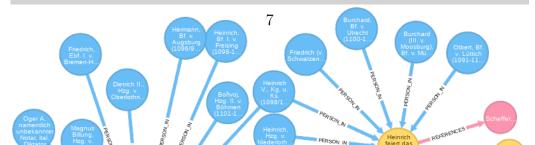
Heinrich feiert das Fest der Apostel, wobei sich Graf Robert von Flandern im Beisein mehrerer Fürsten unterwirft, namentlich der Erzbischöfe Friedrich von Köln und Bruno von Trier, der Bischöfe Otbert von Lüttich, Burchard von Münster, Burchard von Utrecht, Herzog Heinrich von Niederlothringen sowie mehrerer Grafen.

Überlieferung/Literatur

Tagesdatum bei Ann. Patherbr. 1103 (Scheffer-Boichorst 107 f.): in festo apostolorum Petri et Pauli; Gesta Galcheri Episcopi Cameracensis (SS 14, 202); Sigeb. Gembl. 1103 (SS 6, 368); Ann. Elnon. maior. 1103 (SS 5, 14); Ann. Leod., Cont. 1103 (SS 4, 29); Ann. Aquens. 1103 (SS 16, 685); Ann. necrol. Prum. 1103 (SS 13, 223).

Kommentar

Zur Lehensterminologie in den Gesta Galcheri (*Facto palam hominio*, *iurat Robertus Henrico*, *promittit*, *miles domino*, *quia fidelis amodo*) vgl. G a n s h o f, Was ist das Lehnswesen? (³1961) 72 f. – Zum Ereigniskontext vgl. Reg. 1475; Heinrich V. führte bereits 1107 wieder einen Feldzug gegen Robert von Flandern; vgl. B o s h o f, Bischofskirchen von Passau und Regensburg (Salier 2, 1991) 148. – Vgl. Kilian, Itinerar 127 mit der Vermutung einer weiteren, der Unterwerfung Roberts vorangehenden Heerfahrt Heinrichs nach Flandern; Meyer von Knonau, Jbb. 5, 179 f.; T. Reuter, Unruhestiftung (Salier 3, 1991) 324-326.



Die gelben Knoten sind die Regesten. Aus den Angaben des Regests werden mit dem o.a. Befehl noch ein Datumsknoten und ein Ortsknoten erstellt. Mit dem ersten CREATE-Befehl werden die Regesten erstellt. Mit den folgenden MERGE-Befehlen werden anschließend ergänzende Knoten für die Datumsangaben und die Ausstellungsorte erstellt. Nun ist es aber so, dass Ausstellungsort und Ausstellungsdatum mehrfach vorkommen können. Daher wird der hier nicht der CREATE-Befehl sondern der MERGE-Befehl verwendet. Dieser funktioniert wie der CREATE-Befehl, prüft aber vorher ob in der Datenbank ein solcher Knoten schon existiert. Falls es ihn noch nicht gibt wird er erzeugt, wenn es ihn schon gibt, wird er der entsprechenden Variable zugeordnet. Anschließend wird dann die Kante zwischen Regestenknoten und Ausstellungsortsknoten und Regestenknoten und Datumsknoten erstellt. In der folgenden Tabelle werden die einzelnen Befehle dargestellt und kommentiert.

Befehl	Variable	nBemerkungen
LOAD CSV	line	Import
WITH		der
HEADERS		CSV-
FROM		Dateien.
"https://doc	cs.google.d	continuing
" AS		jeweils
line		eine
		Zeile an
		die
		Variable
		line
		weitergegeben

Befehl	Variable	nBemerkungen
CREATE	line.persi	is Exercite Llichery tifier,
(r:Regest	line.sumi	ndersy
{regid:	etc.	Regesten-
line.persiste	nt_identi	flerrotens.
Text:line.su	mmary,	Für die
Über-		weiteren
lieferung:		Befehlt
line.archival	_history,	steht
ident:line.id	entifier})	der neu
	3,	erstellt
		Regesten-
		knoten
		unter
		der
		Variable
		r zur
		Verfügung.

Befehl	Variable	nBemerkungen
MERGE	line.start	<u>Estanteird</u>
(d:Datum	und	geprüft,
{start-	$line.end_{_}$	_datein
date:		Datum-
line.start_o	date,	sknoten
enddate:lin		tennit der
		Datum-
		sangabe
		schon
		ex-
		istiert,
		falls
		nicht,
		wird er
		erstellt.
		In
		jedem
		Fall
		steht an-
		schließend
		der
		Datum-
		sknoten
		unter
		der
		Variable
		d zur
		Verfügung.
		0 0

Befehl	Variable	nBemerkungen
MERGE	line.nam	eEs wird
(o:Ort	ist der	geprüft,
ort:line.nar		
lati-	sname,	
tude:toFloa		
		eslodnognitude)})
<u> </u>	Angaber	
	sind	istiert,
	die	falls
	Geo-	nicht,
	daten	wird er
	des	erstellt.
	Ortes	In
		jedem
		Fall
		steht an-
		schließend
		der Ort-
		sknoten
		unter
		der
		Variable
		o zur
		Verfügung.
MERGE (r) -	(r) ist	Zwischen
[:HAT_DA	TdeM]-	Regesten-
>(d)	Regester	n-knoten
	knoten,	und
	(d) ist	Datum-
	der	sknoten
	Datumsl	k nwite h
		eine
		HAT_DATUM-
		Kante
		erstellt.

Befehl	Variable	nBemerkungen
MERGE (r)-	(r) ist	Zwischen
[:HAT_OR	Γфer	Regesten-
>(0);	Regester	-knoten
	knoten,	und Ort-
	(o) ist	sknoten
	der	wird
	Ortsknot	z eni ne
		HAT_ORT-
		Kante
		erstellt.

3 Erstellen der Ausstellungsorte

In den Kopfzeilen der Regesten ist, soweit bekannt, der Ausstellungsort der Urkunde vermerkt. Im Rahmen der Arbeiten an den Regesta Imperii Online wurden diese Angaben zusammengestellt und soweit möglich die Orte identifiziert, so dass diese Angabe nun bei der Erstellung der Regestendatenbank im Graphen berücksichtigt werden können. Insgesamt befinden sich in den Regesta Imperii über 12.000 verschiedene Angaben für Ausstellungsorte, wobei sie sich teilweise aber auch auf den gleichen Ort beziehen können (Wie z.B. Aachen, Aquisgrani, Aquisgradi, Aquisgranum, coram Aquisgrano etc.). Allein mit den 1.000 häufigsten Ortsangaben konnten schon die Ausstellungsorte der Mehrzahl der Regesten georeferenziert werden.

Mit dem folgenden cypher-Query werden die Ausstellungsorte in die Graphdatenbank importiert:

```
// RI-Ausstellungsorte-geo erstellen
LOAD CSV WITH HEADERS FROM "https://docs.google.com/spreadsheets/d/13_f6Vja4Hf0pjv
WITH line
WHERE line.Lat IS NOT NULL
AND line.normalisiertDeutsch IS NOT NULL
MATCH (r:Regesta {origPlaceOfIssue:line.Original})
MERGE (p:Place {normalizedGerman:line.normalisiertDeutsch, longitude:line.Long, law
WITH r, p, line
```

```
MERGE (r)-[rel:PLACE_OF_ISSUE]->(p)
SET p.wikidataId = line.wikidataId
SET p.name = line.name
SET p.gettyId = line.GettyId
SET p.geonamesId = line.GeonamesId
SET rel.original = line.Original
SET rel.alternativeName = line.Alternativname
SET rel.commentary = line.Kommentar
SET rel.allocation = line.Zuordnung
SET rel.state = line.Lage
SET rel.certainty = line.Sicherheit
SET rel.institutionInCity = line.InstInDerStadt
RETURN count(p)
;
```

Da Import-Query ist etwas komplexer ist, wird er im folgenden näher erläutert. Nach dem LOAD CSV WITH HEADERS FROM-Befehl wird zunächst überprüft, ob der jeweils eingelesene Eintrag in der Spalte line.lat und in der Spalte line.normalisiertDeutsch Einträge hat. Ist dies der Fall wird überprüft, ob es einen Regestenknoten gibt, der einen Ausstellungsorteintrag hat, der der Angabe in der Spalte Original entspricht. Diese Auswahl ist notwendig, da in der Tabelle die Ausstellungsorte der gesamten Regesta Imperii enthalten sind, wir beim Import aber nur die Ortsknoten erstellen, die für die Regesten Kaiser Heinrichs IV. relevant sind. Sind die genannten Bedingungen erfüllt, wird mit dem MERGE-Befehl der Place-Knoten erstellt und anschließend mit dem Regestenknoten verknüpft. Schließlich werden noch weitere Details der Ortsangabe im Place-Knoten und in den PLACE_OF_ISSUE-Kanten ergänzt.

Mit dem folgenden Query werden die Koordinatenangaben zu Höhen- und Breitengraden der Ausstellungsorte (Place-Knoten), die in den Propertys Lat und Long abgespeichert sind in der neuen Property LatLong zusammengefasst und in point-Werte umgewandelt. Seit Version 3 kann neo4j mit diesen Werten Abstandsberechnungen durchführen (Mehr dazu siehe unten bei den Auswertungen).

```
// Regesten und Ausstellungsorte mit Koordinaten der Ausstellungsorte versehen
MATCH (r:Regesta)-[:PLACE_OF_ISSUE]->(o:Place)
SET r.latLong = point({latitude: tofloat(o.latitude), longitude: tofloat(o.longituse)
SET o.latLong = point({latitude: tofloat(o.latitude), longitude: tofloat(o.longituse)
```

```
SET r.placeOfIssue = o.normalizedGerman
SET r.latitude = o.latitude
SET r.longitude = o.longitude
;
```

In den Regesta Imperii Online sind die Datumsangaben der Regesten isokonform im Format JJJJ-MM-TT (also Jahr-Monat-Tag) abgespeichert. neo4j behandelt diese Angaben aber als String. Um Datumsberechnungen durchführen zu können, müssen die Strings in Datumswerte umgerechnet werden. Der cypher-Query hierzu sieht wie folgt aus:

```
// Date in Isodatum umwandeln
MATCH (n:Regesta)
SET n.isoStartDate = date(n.startDate);
MATCH (n:Regesta)
SET n.isoEndDate = date(n.endDate);
MATCH (d:Date)
SET d.isoStartDate = date(d.startDate);
MATCH (d:Date)
SET d.isoEndDate = date(d.endDate);
```

Zunächst werden mit dem MATCH-Befehl alle Regestenknoten aufgerufen. Anschließend wird für jeden Regestenknoten aus der String-Property startDate die Datumsproperty isoStartDate berechnet und im Regestenknoten abgespeichert. Mit Hilfe der Property können dann Datumsangaben und Zeiträume abgefragt werden (Beispiel hierzu unten in der Auswertung).

3.0.1 Herrscherhandeln in den Regesta Imperii

Regesten sind in ihrer Struktur stark formalisiert. Meist wird mit dem ersten Verb im Regest das Herrscherhandeln beschrieben. Um dies auch digital auswerten zu können haben wir in einem kleinen Testprojekt mit Hilfe des Stuttgart-München Treetaggers aus jedem Regest das erste Verb extrahiert und normalisiert. Die Ergebnisse sind in folgender Tabelle einsehbar. Diese Tabelle wird mit dem folgenden cypher-Query in die Graphdatenbank eingelesen.

```
// ReggH4-Herrscherhandeln
LOAD CSV WITH HEADERS FROM "https://docs.google.com/spreadsheets/d/1nlbZmQYcT1E3Z5
```

```
AS line FIELDTERMINATOR ','
MATCH (r:Regesta{ident:line.regid})
MERGE (l:Lemma{lemma:line.Lemma})
MERGE (r)-[:ACTION]->(1);
```

Dabei wird zunächst mit dem MATCH-Befehl das jeweilige Regest gesucht, anschließend mit dem MERGE-Befehl der Lemma-Knoten für das Herrscherhandeln angelegt (falls noch nicht vorhanden) und schließlich der Regesta-knoten mit dem Lemma-Knoten übder eine ACTION-Kante verbunden. Auswertungsperspektiven finden Sie hier ####

3.0.2 Zitationsnetzwerke in den Regesta Imperii

In vielen Online-Regesten ist die zitierte Literatur mit dem Regesta-Imperii-Opac verlinkt. Da es sich um URLs handelt, sind diese Verweise eindeutig andererseits lassen sie sich mit regulären Ausdrücken aus den Regesten extrahieren. Mit folgendem Query werden aus den Überlieferungsteilen der Regesten die mit dem Opac verlinkten Literaturangaben extrahiert und jede Literaturangabe als Refernce-Knoten angelegt.

```
// ReggH4-Literaturnetzwerk erstellen
MATCH (reg:Regesta)
WHERE reg.archivalHistory CONTAINS "link"
UNWIND apoc.text.regexGroups(reg.archivalHistory, "<link (\\S+)>(\\S+)</link>") as
MERGE (ref:Reference {url:link[1]}) ON CREATE SET ref.title=link[2]
MERGE (reg)-[:REFERENCES]->(ref);
```

Da dies mit dem MERGE-Befehl geschieht, wird in der Graphdatenbank jeder Literaturtitel nur einmal angelegt. Anschließend werden die Reference-Knoten mit den Regesten über REFERNCES-Kanten verbunden. Zu den Auswertungsmöglichkeiten vgl. unten den Abschnitt zu den Auswertungsperspektiven.

4 Import der Registerdaten in die Graphdatenbank

4.1 Vorbereitung der Registerdaten

Register spielen für die Erschließung von gedrucktem Wissen eine zentrale Rolle, da dort in alphabetischer Ordnung die im Werk vorkommenden Entitäten (z.B. Personen und Orte) hierarchisch gegliedert aufgeschlüsselt werden. Für die digitale Erschließung der Regesta Imperii sind Register von zentraler Bedeutung, da mit ihnen die in den Regesten vorkommenden Personen und Orte bereits identifiziert vorliegen. Für den Import in die Graphdatenbank wird allerdings eine digitalisierterte Fassung des Registers benötigt. Digitalisierungsprojekt Regesta Imperii Online wurden Anfang der 2000er Jahre auch die gedruckt vorliegenden Register digitalisiert. Sie dienen nun als Grundlage für die digitale Registererschließung der Regesta Imperii. Im hier gezeigten Beispiel werden die Regesten Kaiser Heinrichs IV. und das dazugehörige Register importiert. Da der letzte Regestenband der Regesten Kaiser Heinrichs IV. mit dem Gesamtregister erst vor kurzem gedruckt wurde, liegen hier aktuelle digitale Fassung von Registern und Regesten vor. Die für den Druck in Word erstellte Registerfassung wird hierfür zunächst in eine hierarchisch gegliederte XML-Fassung konvertiert, damit die Registerhierarchie auch maschinenlesbar abgelegt ist.

In der XML-Fassung sind die inhaltlichen Bereiche und die Abschnitte für die Regestennummern jeweils extra in die Tags <Inhalt und Regestennummer eingefasst. Innerhalb des Elements Regestennummer ist dann nochmal jede einzelne Regestennummer in <r>-Tags eingefasst. Die aus dem gedruckten Register übernommenen Verweise sind durch ein leeres <vw/>-Element gekennzeichnet.

Die in XML vorliegenden Registerdaten werden anschließend mit Hilfe von TuStep in einzelne CSV-Tabellen zerlegt.

In einer Tabelle werden alle Entitäten aufgelistet und jeweils mit einer ID versehen.

In der anderen Tabelle werden die Verknüpfungen zwischen Registereinträgen und den Regesten aufgelistet. Der Registereintrag Adalbero kommt also in

```
<Stufe0 id="H4P00005">
    <Inhalt>Abiram, biblische Gestalt, Sohn Eliabs, Bruder Dathans, Auflehner gegen
       Moses</Inhalt>
    <Regestennummer>
        <r>762</r>
    </Regestennummer>
</Stufe0>
<Stufe0 id="H4P00006">
    <Inhalt>AC → Gottschalk v. Aachen</Inhalt>
</Stufe0>
<Stufe0 id="H4P00007">
    <Inhalt>Achalmer, Adelsgeschlecht aus Schwaben</Inhalt>
    <Regestennummer>
       <r>363</r>
   </Regestennummer>
    <Stufe1>
        <Inhalt><vw/>→ Liutold, Gf. v. Achalm</Inhalt>
    </Stufe1>
    <Stufe1>
        <Inhalt><vw/>\rightarrow Werner (v. Achalm), Bf. II. v. Straßburg</Inhalt>
    </Stufe1>
    <Stufe1>
        <Inhalt><vw/>→ Williberga</Inhalt>
    </Stufe1>
</Stufe0>
```

Figure 5: Ausschnitt aus dem XML-Register der Regesten Heinrichs IV.

6	H4P00005	piram, biblische Gestalt, Sohn Eliabs, Bruder Dathans, Auflehner gegen Moses			
7	H4P00006	AC → Gottschalk v. Aachen			
8	H4P00007	chalmer, Adelsgeschlecht aus Schwaben			
9	H4P00008	Aczo, Sohn Rudolf Rubias, Bruder Attos			
10	H4P00009	Adalbero, Mgf. d. karantanischen Mark			
11	H4P00010	Adalbero, Gf. v. Ebersberg, Gem. Richildes			
12	H4P00011	Adalbero, Edelfreier			
13	H4P00012	Adalbero A (AA), namentlich unbekannter Bamberger Diktator			

Figure 6: Ausschnitt der Entitätentabelle des Registers der Regesten Heinrichs IV.

ID	regnum	regnum2	name1	name2				
H4P00001	805	805		<i>A.</i> ,				
H4P00002	1517	1517		<i>A.</i> ,				
H4P00003	1519	1519		<i>A.</i> , <i>centurio</i>				
H4P00005	762	762		Abiram, biblische Gestalt, Sohn Eliabs, Bruder Dathans, Auflehner gegen Moses				
H4P00007	363	363		Achalmer, Adelsgeschlecht aus Schwaben				
H4P00008	1056	1056		Aczo, Sohn Rudolf Rubias, Bruder Attos				
H4P00009	714	714		Adalbero, Mgf. d. karantanischen Mark				
H4P00010	78	78		Adalbero, Gf. v. Ebersberg, Gem. Richildes				
H4P00011	331	331		Adalbero, Edelfreier				
H4P00012	666	666		Adalbero A (AA), namentlich unbekannter Bamberger Diktator				
H4P00012	717	717		Adalbero A (AA), namentlich unbekannter Bamberger Diktator				
H4P00012	959	959		Adalbero A (AA), namentlich unbekannter Bamberger Diktator				
H4P00012	1400	1400		Adalbero A (AA), namentlich unbekannter Bamberger Diktator				
H4P00012	1403	1403		Adalbero A (AA), namentlich unbekannter Bamberger Diktator				
H4P00012	1420	1420		Adalbero A (AA), namentlich unbekannter Bamberger Diktator				
H4P00012	1481	1481		Adalbero A (AA), namentlich unbekannter Bamberger Diktator				
H4P00012	1485	1485		Adalbero A (AA), namentlich unbekannter Bamberger Diktator				

Figure 7: Ausschnitt der Verknüpfungstabelle des Registers der Regesten Heinrichs IV.

mehreren Regesten vor. Da das Register der Regesten Heinrichs IV. nur zwei Hierarchiestufen enthält, in denen beispielsweise verschiedene Amtsphasen ein und derselben Person unterschieden werden, wurden diese beim Import zusammengefasst.² Damit gibt es pro Person jeweils nur einen Knoten.

4.2 Import der Registerdaten in die Graphdatenbank

Im Gegensatz zu den Regesten Kaiser Friedrichs III., bei denen Orte und Personen in einem Register zusammengefasst sind, haben die Regesten Kaiser Heinrich IV. getrennte Orts- und Personenregister. Die digitalisierten Registerdaten können hier eingesehen werden. In dem Tabellendokument befinden sich insgesamt drei Tabellen. In der Tabelle Personen sind die Einträge des Personenregisters aufgelistet und in der Tabelle Orte befindet sich die Liste aller Einträge des Ortsregisters. Schließlich enthält die Tabelle APPEARS_IN Information dazu, welche Personen oder Orte in welchen Regesten genannt sind. Der folgende cypher-Query importiert die Einträge der Personentabelle

²Vgl. die Vorbemerkung zum Register in Böhmer, J. F., Regesta Imperii III. Salisches Haus 1024-1125. Tl. 2: 1056-1125. 3. Abt.: Die Regesten des Kaiserreichs unter Heinrich IV. 1056 (1050) - 1106. 5. Lief.: Die Regesten Rudolfs von Rheinfelden, Hermanns von Salm und Konrads (III.). Verzeichnisse, Register, Addenda und Corrigenda, bearbeitet von Lubich, Gerhard unter Mitwirkung von Junker, Cathrin; Klocke, Lisa und Keller, Markus - Köln (u.a.) (2018), S. 291.

in die Graphdatenbank und erstellt für jeden Eintrag einen Knoten vom Typ: IndexPerson:

```
// Registereinträge Personen erstellen
```

LOAD CSV WITH HEADERS FROM "https://docs.google.com/spreadsheets/d/12T-RD1Ct4aAUNN AS line

CREATE (:IndexPerson {registerId:line.ID, name1:line.name1});

Mit dem folgenden cypher-Query werden nach dem gleichen Muster aus der Tabelle Orte die Ortseinträge in die Graphdatenbank importiert.

OAD CSV WITH HEADERS FROM "https://docs.google.com/spreadsheets/d/12T-RD1Ct4aAUNNN AS line

CREATE (:IndexPlace {registerId:line.ID, name1:line.name1});

Die beiden Befehle greifen also auf verschiedene Tabellenblätter des gleichen Google-Tabellendokuments zu, laden es als CSV-Daten und übergeben die Daten zeilenweise an die weiteren Befehle (Hier an den MATCH- und den CREATE-Befehl). Im nächsten Schritt werden nun mit den Daten der APPEARS_INTabelle die Verknüpfungen zwischen den Registereinträgen und den Regesten erstellt.

```
// PLACE_IN-Kanten für Orte erstellen
```

LOAD CSV WITH HEADERS FROM "https://docs.google.com/spreadsheets/d/12T-RD1Ct4aAUNN AS line

MATCH (from:IndexPlace {registerId:line.ID})

MATCH (to:Regesta {regnum:line.regnum2})

CREATE (from)-[:PLACE_IN {regnum:line.regnum, name1:line.name1, name2:line.name2}]

Dabei werden zunächst mit den beiden MATCH-Befehlen jeweils das Regest und der Registereintrag aufgerufen und schließend mit dem CREATE-Befehl eine PLACE_IN-Kante zwischen den beiden Knoten angelegt, die als Attribute den Inhalt der Spalten name1 und name2 mitbekommt. Analog werden die Verknüpfungen zwischen Regestenknoten und Personenknoten angelegt:

```
// PERSON_IN-Kanten für Person erstellen
```

LOAD CSV WITH HEADERS FROM "https://docs.google.com/spreadsheets/d/12T-RD1Ct4aAUNN AS line

MATCH (from:IndexPerson {registerId:line.ID}), (to:Regesta {regnum:line.regnum2})
CREATE (from)-[:PERSON_IN {regnum:line.regnum, name1:line.name1, name2:line.name2}]

4.3 Die Hierarchie des Registers der Regesten Kaiser Friedrichs III.

In anderen Registern der Regesta Imperii, wie beispielsweise den Regesten Kaiser Friedrichs III. sind teilweise fünf oder mehr Hierarchiestufen vorhanden, die jeweils auch Entitäten repräsentieren. In diesen Fällen müssen die Hierarchien auch in der Graphdatenbank abgebildet werden, was durch zusätzliche Verweise auf die ggf. vorhandenen übergeordneten Registereinträge möglich wird.

nodeID	xmlID	topnodeID	name1	name3
1	A0000001		Aa, Johann von	Aa, Johann von ~
2	A00000002	1	Sophie von ∼, To	Aa, Johann von ~ // Sophie von ~, Tochter Johanns, Bürgerin zu Köln
3	A0000003		Aach	Aach (Fluß durch Aach, n. Singen, Baden-Württemberg)
4	A0000004		Aach	Aach (n. Singen, Baden-Württemberg), Stadt
5	A0000005		Aache s. Aacher	Aache s. Aachen
6	A00000006		Aachen	Aachen (Aache; Nordrhein-Westfalen), Stadt
7	A0000007	6	Einwohner und B	Aachen (Aache; Nordrhein-Westfalen), Stadt // Einwohner und Bürger; s. Coly
8	800000008	6	Fischmarkt	Aachen (Aache; Nordrhein-Westfalen), Stadt // Fischmarkt (Parwisch)
9	A00000009	6	Gerichte	Aachen (Aache; Nordrhein-Westfalen), Stadt // Gerichte
10	A0000010	6	Kurgericht	Aachen (Aache; Nordrhein-Westfalen), Stadt // Gerichte // Kurgericht
11	A0000011	6	Schöffenstuhl, k	Aachen (Aache; Nordrhein-Westfalen), Stadt // Gerichte // Schöffenstuhl, kgl.
12	A0000012	6	Richter und Schö	Aachen (Aache; Nordrhein-Westfalen), Stadt // Gerichte // Schöffenstuhl, kgl.
13	A0000013	6	Schöffen	Aachen (Aache; Nordrhein-Westfalen), Stadt // Gerichte // Schöffenstuhl, kgl.
14	A0000014	6	"Grafschaften"	Aachen (Aache; Nordrhein-Westfalen), Stadt // "Grafschaften"
15	A0000015	6	"Grasgebot"	Aachen (Aache; Nordrhein-Westfalen), Stadt // "Grasgebot"
17	A0000017	6	Meierei	Aachen (Aache; Nordrhein-Westfalen), Stadt // Meierei
19	A0000019	6	Brothaus	Aachen (Aache; Nordrhein-Westfalen), Stadt // Meierei // Brothaus
21	A00000021	6	Gewandhaus	Aachen (Aache; Nordrhein-Westfalen), Stadt // Meierei // Gewandhaus
22	A00000022	6	Grashaus	Aachen (Aache; Nordrhein-Westfalen), Stadt // Meierei // Grashaus
23	A00000023	6	Plankenhaus	Aachen (Aache; Nordrhein-Westfalen), Stadt // Meierei // Plankenhaus
24	A00000024	6	Tuchhaus	Aachen (Aache; Nordrhein-Westfalen), Stadt // Meierei // Tuchhaus
25	A00000025	6	Haus zum Haner	Aachen (Aache; Nordrhein-Westfalen), Stadt // Meierei // Haus zum Haner (zu
26	A00000026	6	zo der Geiss	Aachen (Aache; Nordrhein-Westfalen), Stadt // Meierei // zo der Geiss
27	A0000027	6	Rentmeister	Aachen (Aache; Nordrhein-Westfalen), Stadt // Rentmeister
29	A00000029	6	"Stadtbücher"	Aachen (Aache; Nordrhein-Westfalen), Stadt // "Stadtbücher"
30	A0000030	6	Vogtei	Aachen (Aache; Nordrhein-Westfalen), Stadt // Vogtei

Figure 8: Ausschnitt der Entitätentabelle des Registers der Regesten Friedrichs III.

Im Tabellenausschnitt wird jedem Registereintrag in der ersten Spalte eine nodeID als eindeutige Kennung zugewiesen. Bei Registereinträgen, die kein Hauptlemma sind, enthält die dritte Spalte topnodeID den Verweis auf die eindeutige Kennung nodeID des übergeordneten Eintrages. Beim Import in die Graphdatenbank wird diese Hierarchie über CHILD_OF-Kanten abgebildet, die vom untergeordneten Eintrag auf das übergeordnete Lemma verweisen. Damit ist die komplette Registerhierarchie im Graphen abgebildet. In der Spalte name1 ist das Lemma angegeben, in der Spalte name3 zusätzliche zum Lemma

noch der gesamte Pfad vom Hauptlemma bis zum Registereintrag, jeweils mit Doppelslahes (//) getrennt. Bei tiefer gestaffelten Registern ist teilweise ohne Kenntnis der übergeordneten Einträge eine eindeutige Identifizierung eines Eintrages nicht möglich. So wird in Zeile 17 der o.a. Abbildung allein mit der Angabe aus der Spalte name1 nicht klar ist, um welche Meierei es sich handelt. Mit dem kompletten Pfad des Registereintrages in der Spalte name3 wird dagegen deutlich, dass die Aachener Meierei gemeint ist.

5 Auswertungsperspektiven

5.1 Personennetzwerke in den Registern

5.1.1 Graf Robert II. von Flandern in seinem Netzwerk

Nach dem Import können nun die Online-Regesten und die Informationen aus dem Registern der Regesten Kaiser Heinrichs IV. in einer Graphdatenbank aus einer Vernetzungsperspektive abgefragt werden.³

Ausgangspunkt ist der Registereintrag von Graf Robert II. von Flandern. Diesen Knoten finden wir mit folgendem Query.

```
// Robert II. von Flandern
MATCH (n:IndexPerson) WHERE n.registerId = 'H4P01822'
RETURN *;
```

Mit einem Doppelklick auf den IndexPerson-Knoten öffnen sich alle Regesta-Knoten, in denen Robert genannt ist. Klickt man nun wiederum alle Regestenknoten doppelt, sieht man alle Personen und Orte, mit denen Robert gemeinsam in den Regesten genannt ist.

Dies kann auch in einem cypher-Query zusammengefasst werden.

```
// Robert II. von Flandern mit Netzwerk
MATCH (n:IndexPerson)-[:PERSON IN]->
```

³Die nun folgenden Abfragen sind zum Teil einer Präsentation entnommen, die für die Summerschool der Digitalen Akademie im Rahmen des Mainzed entwickelt wurden. Die Präsentation findet sich unter der URL https://digitale-methodik.adwmainz.net/mod5/5c/slides/graphentechnologien/RI.html.

```
(r:Regesta) <- [:PERSON_IN] -
(m:IndexPerson)
WHERE n.registerId = 'H4P01822'
RETURN *;</pre>
```

In der folgenden Abb. wird das Ergebnis dargestellt.

Hier wird der MATCH-Befehl um einen Pfad über PERSON_IN-Kanten zu Regesta-Knoten ergänzt, von denen dann wiederum eine PERSON_IN-Kante zu den anderen, in den Regesten genannten IndexPerson-Knoten führt.

Nimmt man noch eine weitere Ebene hinzu, wächst die Ergebnismenge stark an.

```
// Robert II. von Flandern mit Netzwerk und Herrscherhandeln (viel)
MATCH
(n1:IndexPerson)-[:PERSON_IN]->(r1:Regesta)<-[:PERSON_IN]-
(n2:IndexPerson)-[:PERSON_IN]->(r2:Regesta)<-[:PERSON_IN]-
(n3:IndexPerson)
WHERE n1.registerId = 'H4P01822'
RETURN *;</pre>
```

5.1.2 Graf Robert II. von Flandern und Herzog Heinrich von Niederlothringen

In der Graphdatenbank ist es über die Exploration der Beziehungen einer Person hinaus möglich explizit die Verbindungen von zwei Personen abzufragen. In unserem nächsten Beispiel suchen wir jene Regesten, in denen Graf Robert II. von Flandern und Herzog Heinrich von Niederlothringen gemeinsam genannt sind.

```
// Robert II. von Flandern und Herzog Heinrich von Niederlothringen mit Netzwerk
MATCH
(n:IndexPerson)-[:PERSON_IN]->
(r:Regesta)<-[:PERSON_IN]-(m:IndexPerson)
WHERE n.registerId = 'H4P01822'
AND m.registerId = 'H4P00926'
RETURN *;</pre>
```

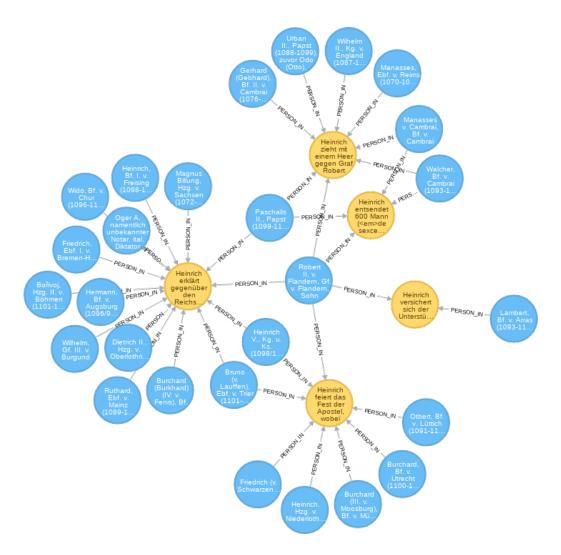


Figure 9: Robert mit den Personen, mit denen er gemeinsam in Regesten genannt wird.

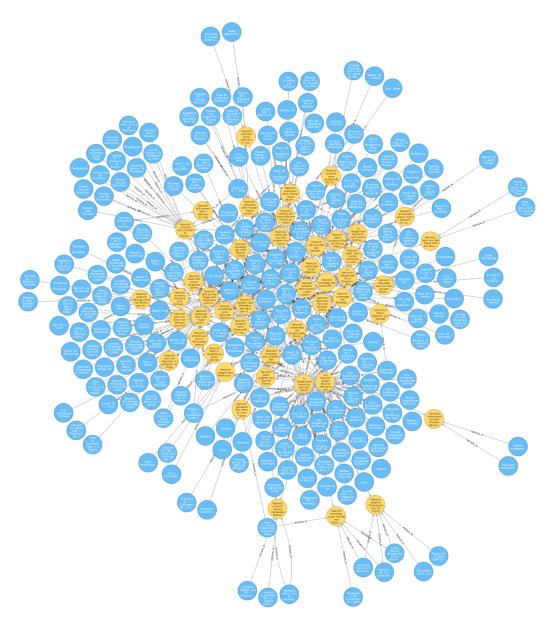


Figure 10: Robert mit Personen, die wiederum mit Personen gemeinsam in Regesten genannt sind.

Es zeigt sich, dass Robert und Heinrich in einem Regest gemeinsam genannt sind.



Figure 11: Robert und Heinrich sind in einem Regest gemeinsam genannt.

Und dieses Regest berichtet ausgerechnet über die Unterwerfung Roberts und Heinrich ${\rm IV.^4}$

Heinrich feiert das Fest der Apostel, wobei sich Graf Robert von Flandern im Beisein mehrerer Fürsten unterwirft, namentlich der Erzbischöfe Friedrich von Köln und Bruno von Trier, der Bischöfe Otbert von Lüttich, Burchard von Münster, Burchard von Utrecht, Herzog Heinrich von Niederlothringen sowie mehrerer Grafen.

Möglicherweise haben beide aber gemeinsame Bekannte, also Personen mit denen sowohl Heinrich als auch Robert in unterschiedlichen Regesten gemeinsam genannt sind. Hierfür wird der cypher-Query um eine Ebende erweitert.

```
// Robert und Heinrich mit allen gemeinsamen Personen und Regesten
MATCH (n1:IndexPerson)-[:PERSON_IN]->(r1:Regesta)<-[:PERSON_IN]-(n2:IndexPerson)-
<-[:PERSON_IN]-(n3:IndexPerson)
WHERE n1.registerId = 'H4P00926'
AND n3.registerId = 'H4P01822'</pre>
```

Ein erster Blick auf das Ergebnis zeigt, dass Heinrich allgemein besser vernetzt ist. Für die weitere Analyse ihres Verhältnisses ist nun die Lektüre der angegebenen Regesten notwendig. Hierfür lässt sich das Ergebnis noch etwas weiter aufbereiten, indem die zwischen den Personen liegenden Regesten in KNOWS-Kanten umgewandelt werden, die als zusätzliche Information die Angaben zu den Regesten enthalten.

RETURN *;

⁴Vgl. RI III,2,3 n. 1487.

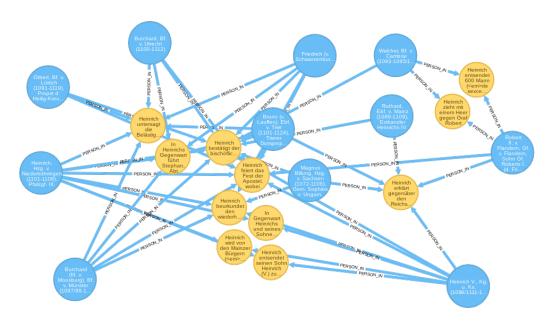


Figure 12: Robert und Heinrich mit den gemeinsamen Bekanntschaften.

```
// Rausrechnen der dazwischenliegenden Knoten
MATCH
```

(startPerson:IndexPerson)-[:PERSON IN]->

(regest:Regesta)<-[:PERSON IN]-(endPerson:IndexPerson)</pre> WHERE startPerson.registerId in ['H4P01822', 'H4P00926'] WITH startPerson, endPerson, count(regest) as anzahl,

collect(regest.ident) as idents

CALL apoc.create.vRelationship(startPerson, "KNOWS", {anzahl:anzahl, regesten:idents}, endPerson) YIELD rel RETURN startPerson, endPerson, rel

In der Abbildung sind die zwei Ego-Netzwerke von Heinrich (links) und Robert (rechts) mit den dazwischen liegenden gemeinsamen Bekanntschaften dargestellt. Es zeigt sich, dass Heinrich stärker sowohl mit Geistlichen als auch Weltlichen vernetzt war, während Robert insgesamt weniger Kontakte aber mit einem Schwerpunkt in der Geistlichkeit hatte.

Für den Historiker ist aber vor allem interessant, was in den Regesten steht, die Robert und Heinrich über die Mittelsmänner verbinden. Hierfür wird der cypher-Query angepasst und sowohl Personen und die Regestentexte

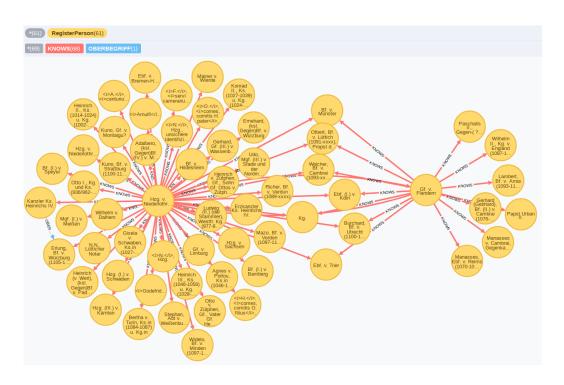


Figure 13: Robert und Heinrich mit den gemeinsamen Bekanntschaften.

ausgegeben.

```
// Liste der Regesten als Ergebnis
MATCH
(startPerson:IndexPerson)-[:PERSON_IN]->
(regest1:Regesta)<-[:PERSON_IN]-(middlePerson:IndexPerson)
-[:PERSON_IN]->(regest2:Regesta)
<-[:PERSON_IN]-(endPerson:IndexPerson)
WHERE startPerson.registerId in ['H4P00926']
AND endPerson.registerId in ['H4P01822']
RETURN DISTINCT startPerson.name1, regest1.ident, regest1.text,
middlePerson.name1, regest2.ident, regest2.text, endPerson.name1;</pre>
```

In der Abbildung wird ein Ausschnitt der Ergebnistabelle gezeigt. In der ersten Spalte der Tabelle finden sich Robert, anschließend die Angaben zum Regest, mit dem er mit der mitteleren Person (middlePerson.name1) verknüpft ist. Dem folgen schließlich die Angaben zum Regest, mit den die mittlere Person mit Robert in der letzten Spalte verbunden ist. Die Tabelle bietet einen Überblick zur Überlieferungssituation aus der Perspektive der Regesta Imperii.

startPerson.name1	regest1.ident	regest1.text	middlePerson.name1	regest2.ident	regest2.text	endPerson.name1
"Heinrich, Hzg. v.	"RI III,2,3 n.	"Heinrich bestätigt der bischöflichen	"Bruno (v. Lauffen),	"RI III,2,3 n.	"Heinrich feiert	"Robert II. v.
Niederlothringen	1489"	Kirche zu <span< td=""><td>Ebf. v. Trier (1101-</td><td>1487"</td><td>das Fest der</td><td>Flandern, Gf. v.</td></span<>	Ebf. v. Trier (1101-	1487"	das Fest der	Flandern, Gf. v.
(1101-1106),		class="spaced">Bamberg	1124), Trierer		Apostel, wobei	Flandern, Sohn Gf.
Pfalzgf. III. v.		unter Bischof Otto zum Gedenken an	Dompropst"		sich Graf Robert	Roberts I. (d.
Lothringen, Gf. v.		seine Großeltern, Kaiser Konrad (II.)			von Flandern im	Friesen) v.
Limburg"		und Kaiserin Gisela, seine Eltern,			Beisein mehrerer	Flandern, Neffe Gf.
		Kaiser Heinrich (III.) und Agnes,			Fürsten	Balduins VI. v.
		seine Gemahlin, die Kaiserin Bertha,			unterwirft,	Flandern"
		sowie besonders an seinen			namentlich der	
		Verwandten, Kaiser Heinrich (II.),			Erzbischöfe	
		den Gründer der Bamberger Kirche,			Friedrich von	
		aufgrund der Intervention seines			Köln und Bruno	
		Sohnes, König Heinrichs V., der			von Trier, der	
		Erzbischöfe Friedrich von Köln,			Bischöfe Otbert	
		Bruno von Trier und Humbert von			von Lüttich,	
		Bremen, der Bischöfe Otbert von			Burchard von	

Figure 14: Robert und Heinrich mit den gemeinsamen Bekanntschaften.

5.2 Herrscherhandeln ausgezählt

Wie bereits oben erwähnt wurde in einem ersten Test jeweils das erste Verb des Regestentextes extrahiert, lemmatisiert und in die Graphdatenbank eingespielt. Im folgenden werden nun einige cypher-Querys vorgestellt, die dies beispielhaft auswerten.

// Herrscherhandeln ausgezählt
MATCH (n:Lemma) <- [h:ACTION] - (m:Regesta)
RETURN n.lemma, count(h) as ANZAHL ORDER BY ANZAHL desc LIMIT 10;</pre>

n.lemma	ANZAHL
werden	145
schenken	133
bestätigen	109
begehen	95
verleihen	48
ernennen	36
nehmen	35
treffen	34
empfangen	29
erhalten	26

Die Ergebnisliste zeigt gleich die Einschränkungen, da das Hilfsverb werden aus dem textuellen Zusammenhang gerissen ist. Andererseits ergeben sich aber auch interessante Erkenntnisse zur Häufigkeitsverteilung von Herrscherhandeln in Regestentexten. Die Anwendung des Verfahrens auf Regestentexte ist dabei auf der einen Seite positiv, da bei der Erstellung der Regesten sehr stark auf formale Kriterien geachtet wird und so die Zusammenhänge generalisiert gut zu erfassen sind. Auf der anderen Seite ist die Auswertung aber wiederum einen weiteren Schritt von der ursprünglichen Quelle entfernt.

5.3 Herrscherhandeln pro Ausstellungsort ausgezählt

Im folgenden Query kommt eine räumliche Komponente zur Abfrage hinzu, da das Lemma hier jeweils abhängig vom Ausstellungsort der Urkunde abgefragt wird.

```
// Herrscherhandeln pro Ausstellungsort
MATCH (n:Lemma)<-[h:ACTION]-(:Regesta)-[:PLACE_OF_ISSUE]->(p:Place)
WHERE p.normalizedGerman IS NOT NULL
```

RETURN p.normalizedGerman, n.lemma, count(h) as ANZAHL ORDER BY ANZAHL desc LIMIT

p.normalizedGerman	n.lemma	ANZAHL
Mainz	begehen	15
Mainz	schenken	14
Goslar	schenken	13
Rom	werden	12
Regensburg	schenken	12
Goslar	begehen	11
Speyer	schenken	10
Worms	begehen	8
Regensburg	bestätigen	7
Regensburg	werden	7

In der ersten Spalte befindet sich der Ortsname, der aus der Property normalizedGerman des Place-Knotens stammt. In der zweiten Spalte wird das Lemma angegeben und in der dritten Spalte schließlich die Anzahl der jeweiligen Regesten. Interessant wäre hier auch noch die Ergänzung der zeitlichen Dimension, mit der dann der zeitliche Verlauf in die Auswertung miteinbezogen werden könnte.

5.4 Herrscherhandeln und Anwesenheit

Im nächsten Beispiel werden in einem Regest genannten Personen in die Auswertung des Herrscherhandelns mit einbezogen.

```
MATCH (p:IndexPerson)-[:PERSON_IN]-(r:Regesta)-[:ACTION]-(1:Lemma)
RETURN p.name1, l.lemma, count(l) AS Anzahl ORDER BY p.name1, Anzahl DESC;
```

p.namelllemmaAnzahl

Adalbeschenken 21

Met-

zer

Domkanon-

iker,

Kan-

zler

Hein-

richs

IV.,

Kan-

zler

(Gegen)Kg.

Rudolfs

v.

Rheinfelden

Adalbebæstätigen 9

Met-

zer

Domkanon-

iker,

Kan-

zler

Hein-

richs

IV.,

Kan-

zler

(Gegen)Kg.

Rudolfs

v.

Adalbe**re**rleihen 4

Met-

zer

Domkanon-

iker,

Kan-

zler

Hein-

richs

IV.,

Kan-

zler

(Gegen)Kg.

Rudolfs

v.

Rheinfelden

2 Adalbeeolassen

Met-

zer

Domkanon-

iker,

Kan-

zler

Hein-

richs

IV.,

Kan-

zler

(Gegen)Kg.

Rudolfs

Adalberichertragen2

Met-

zer

Domkanon-

iker,

Kan-

zler

Hein-

richs

IV.,

Kan-

zler

(Gegen)Kg.

Rudolfs

v.

Rheinfelden

Adalbe**eo**mäßigen 2

Met-

zer

Domkanon-

iker,

Kan-

zler

Hein-

richs

IV.,

Kan-

zler

(Gegen)Kg.

Rudolfs

v.

Adalbe**ge**statten 2

Met-

zer

Domkanon-

iker,

Kan-

zler

Hein-

richs

IV.,

Kan-

zler

(Gegen)Kg.

Rudolfs

v.

Rheinfelden

Adalbe**vo**llziehen 1

Met-

zer

Domkanon-

iker,

Kan-

zler

Hein-

richs

IV.,

Kan-

zler

(Gegen)Kg.

Rudolfs

v.

Adalbenehmen 1

Met-

zer

Domkanon-

iker,

Kan-

zler

Hein-

richs

IV.,

Kan-

zler

(Gegen)Kg.

Rudolfs

v.

Rheinfelden

Adalbemoindern 1

Met-

zer

Domkanon-

iker,

Kan-

zler

Hein-

richs

IV.,

Kan-

zler

(Gegen)Kg.

Rudolfs

v.

```
p.namelllemmaAnzahl
Adalbesetzen
                 1
Met-
zer
Domkanon-
iker,
Kan-
zler
Hein-
richs
IV.,
Kan-
zler
(Gegen)Kg.
Rudolfs
v.
Rheinfelden
```

Die Ergebnistabelle zeigt den Abschnitt zu Adalbero, einem Metzer Domkanoniker mit den mit der Häufigkeit des jeweiligen Herrscherhandeln-Lemmas.

5.5 Regesten 200 km rund um Augsburg

In den Propertys der Knoten lassen sich auch ex

```
// Entfernungen von Orten berechnen lassen
MATCH (n:Place)
WHERE n.normalizedGerman = 'Augsburg'
WITH n.latLong as point
MATCH (r:Regesta)
WHERE distance(r.latLong, point) < 200000
AND r.placeOfIssue IS NOT NULL
AND r.placeOfIssue <> 'Augsburg'
RETURN r.ident, r.placeOfIssue,
```

distance(r.latLong, point) AS Entfernung
ORDER BY Entfernung;

5.6 Welche Literatur wird am meisten zitiert

Wie bereits oben erwähnt, werden beim Import der Regesten in die Graphdatenbank die mit dem RI-Opac verlinkten Literaturtitel als eigenständige Reference-Knoten angelegt und jeweils mit dem Regesta-Knoten verknüpft. Diese Verknüpfung wird mit dem folgenden Query abgefragt, ausgezählt und aufgelistet.

MATCH (n:Reference) <- [r:REFERENCES] - (m:Regesta)
RETURN n.title, count(r) as ANZAHL ORDER BY ANZAHL desc LIMIT 10;

n.title	ANZAHL
Stumpf	215
Böhmer	201
Ldl	101
Jaffé	60
Schmale	56
Buchholz	51
Scheffer-Boichorst	50
Wauters	39
Dobenecker	33
Remling	28

6 Der Import zusammengefasst

Den cypher-Code für die Erstellung der Graphdatenbank ist zusammengefasst über ein Google-Docs-Dokument abrufbar. Es ist zu empfehlen, die aktuelle Version von neo4j-Desktop zu installieren, eine Graphdatenbank anzulegen und in der Graphdatenbank die APOC-Bibliothek zu installieren. Nach dem Start der Graphdatenbank kann dann im Reiter Terminal mit dem Befehl bin/cypher-shell die cypher-shell aufgerufen werden. In diese Shell werden dann alle Befehl gemeinsam reinkopiert und ausgeführt. Alternativ

zur Installation von neo4j kann auch auf den Internetseiten von neo4j seine Sanbox erstellt werden. Zu beachten ist, dass dort nur der neo4j-Browser für die Eingabe der cypher-Querys zur Verfügung steht und jeder Query (also alle cypher-Befehle, die mit einem Semikolon abgeschlossen sind) einzeln in die Eingabezeile kopiert werden müssen.

7 Zusammenfassung

In diesem Kapitel wurden die Schritte zum Import der Regesten Kaiser Heinrichs IV. in die Graphdatenbank neo4j erläutert und schließlich noch Auswertungsbeispiele gezeigt.