Graphdatenbanken

-

Neo4J

Yavuz Arslan
30.05.2017

Hochschule für Angewandte Wissenschaften
Hamburg

Inhalt

I. NoSQL
II. Graphdatenbanken
III. Neo4J
IV. Panama Papers

Relationale Datenbanken

- SQL (Structured Query Language)
- „Eine für alles“-Anspruch
- Basieren auf **ACID**:
  - Atomicity - Atomarität
  - Consistency - Konsistenzerhaltung
  - Isolation - Isolation
  - Durability - Dauerhaftigkeit
- Vertikale Skalierung

Big Data

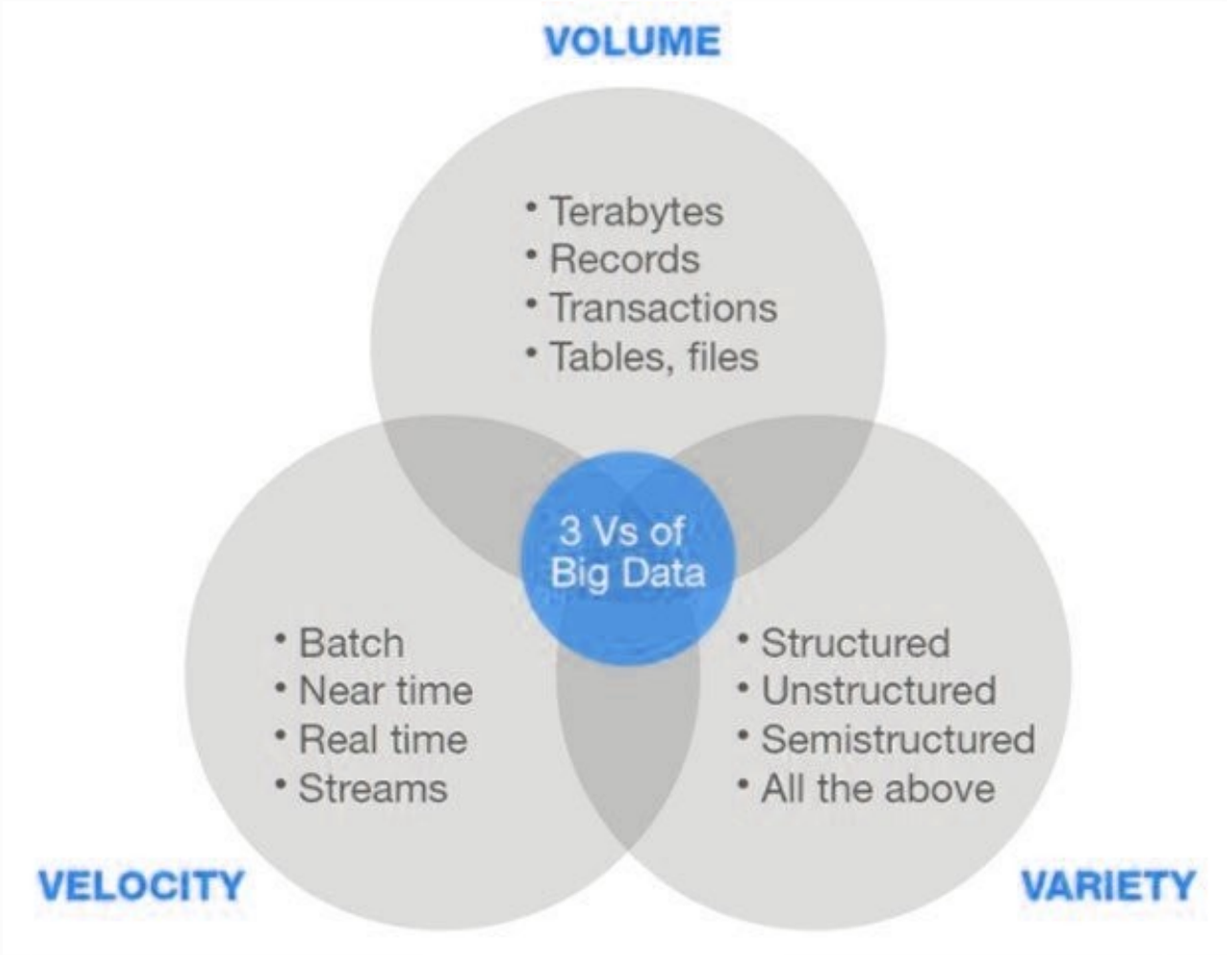


Abb. 0.1

I. NoSQL

Charakteristiken von NoSQL Datenbanken

- Nicht relational
- “Cluster friendly” - horizontale Skalierung
  - Cloud computing
- Kein festes Schema
- Basieren auf **BASE:**
  - **B**asically **A**vailable
  - **S**oft state
  - **E**ventual consistency

Key-Value Stores

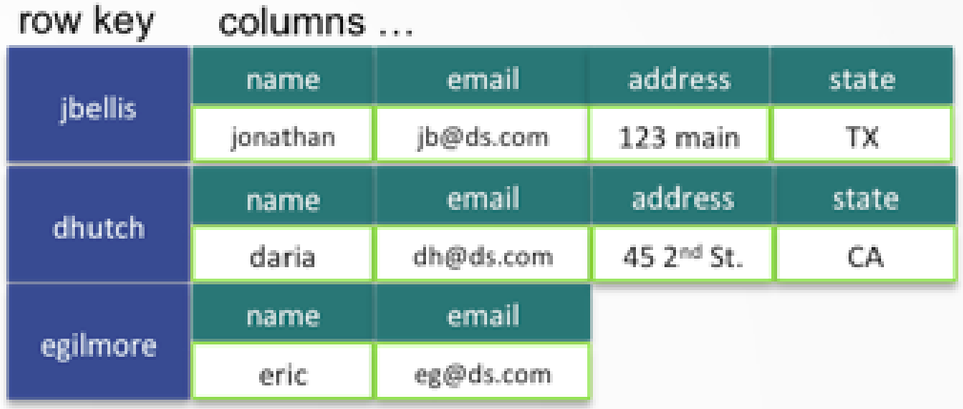
Key	Value
K1	AAA,BBB,CCC
K2	AAA,BBB
K3	AAA,DDD
K4	AAA,2,01/01/2015
K5	3,ZZZ,5623

Abb. 1.1

- Agieren als große, verteilte Hashmap
  - Zugriff auf Daten (Values) über Schlüssel (Keys)
- ‚Values' haben kein vorgeschriebenes Format
  - Interpretation auf Anwendungsebene
- Content Caching / Session Daten / Image Stores

→ Redis, Amazon DynamoDB

Column Family



row key	columns …			
jbellis	name	email	address	state
	jonathan	jb@ds.com	123 main	TX
dhutch	name	email	address	state
	daria	dh@ds.com	45 2nd St.	CA
egilmore	name	email		
	eric	eg@ds.com		

Abb. 1.2

- Basieren auf Googles **Big Table**
- Verwenden Key-Value Paare:
  - Values sind ein Set aus Spalten
- ‚Row keys' dienen als eingebauter Index
- Daten Verarbeitung (z.B. Netzwerk-Traffic oder Log Daten)

→ Apache Hbase, Cassandra

Document Stores

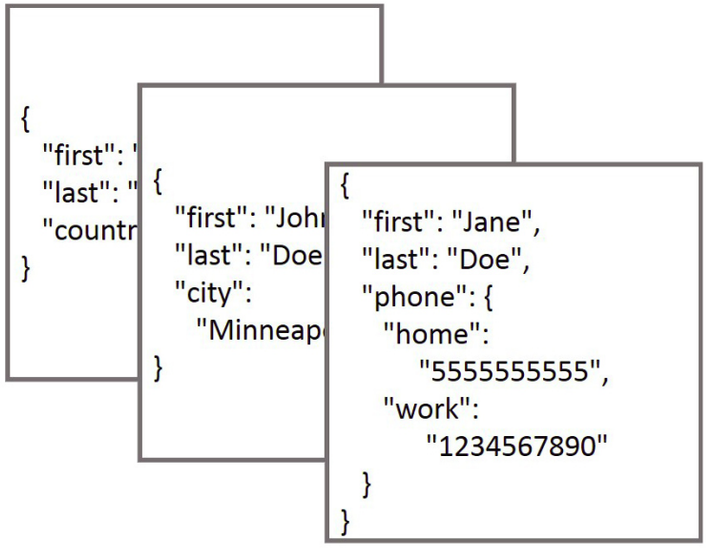


Abb. 1.3

- Sammlung von Dokumenten
- Dokument ist Key-Value Sammlung
  - Unterklasse von Key-Value DB
- CMS, Datentypen mit variablen Attributen (z.B. Produkte)

→ MongoDB, CouchDB

II. Graphdatenbanken

Graphdatenbanken - Einführung

- Datenbank mit expliziter Graph-Struktur
- Beziehungen sind anderen Informationen gleichgestellt
  - First-Class Citizens
- Bieten elementare CRUD-Operationen

Labeled Property Graph Model

- Besteht aus **Nodes** (Knoten), **Relationships** (Beziehungen/Kanten), **Properties** (Eigenschaften) und **Labels**
- **Knoten** und **Beziehungen** können **Eigenschaften** in Form von Key-Value Paaren zugeordnet werden
- **Knoten** können beliebige **Labels** erhalten. **Labels** gruppieren **Knoten**
- **Beziehungen** verbinden **Knoten**

Labeled Property Graph Model

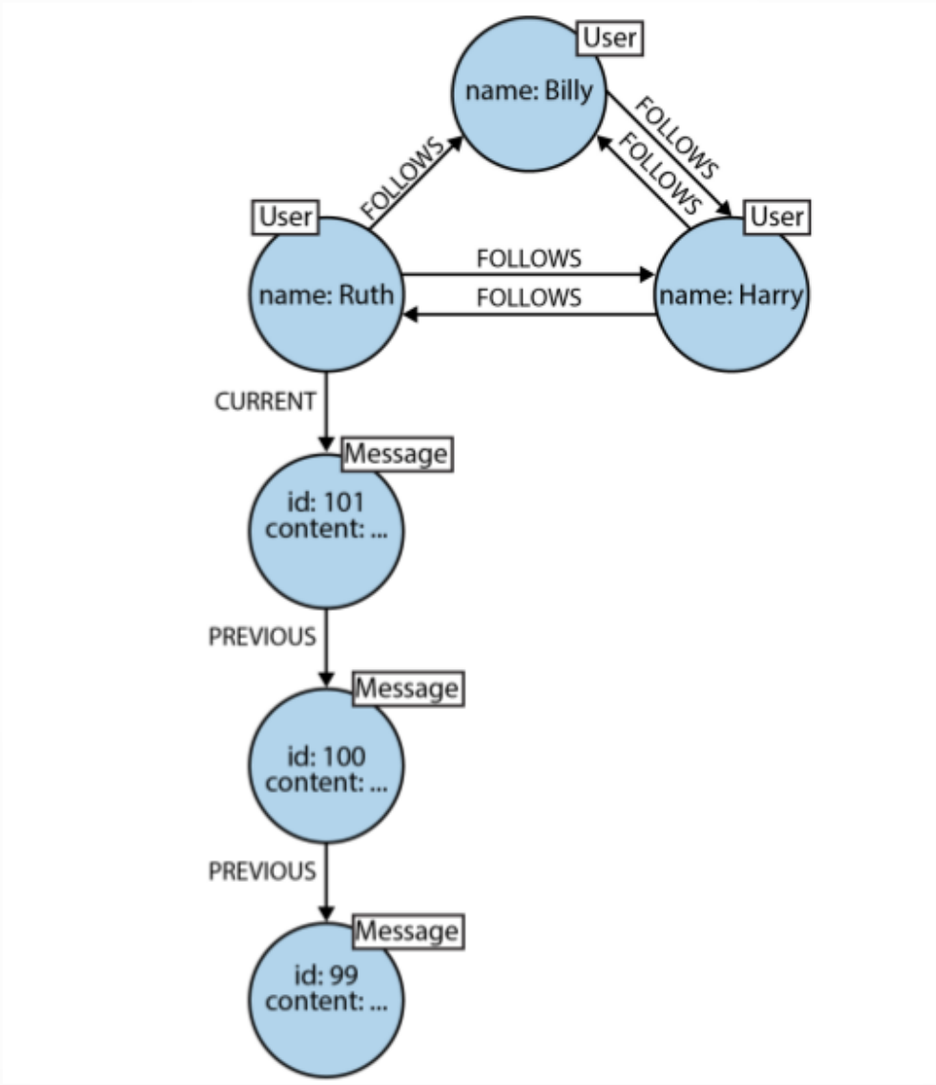


Abb. 2.1

Hypergraph

- Beziehungen können beliebig viele Knoten verbinden
- Nützlich bei vielen viele-zu-viele-Beziehungen

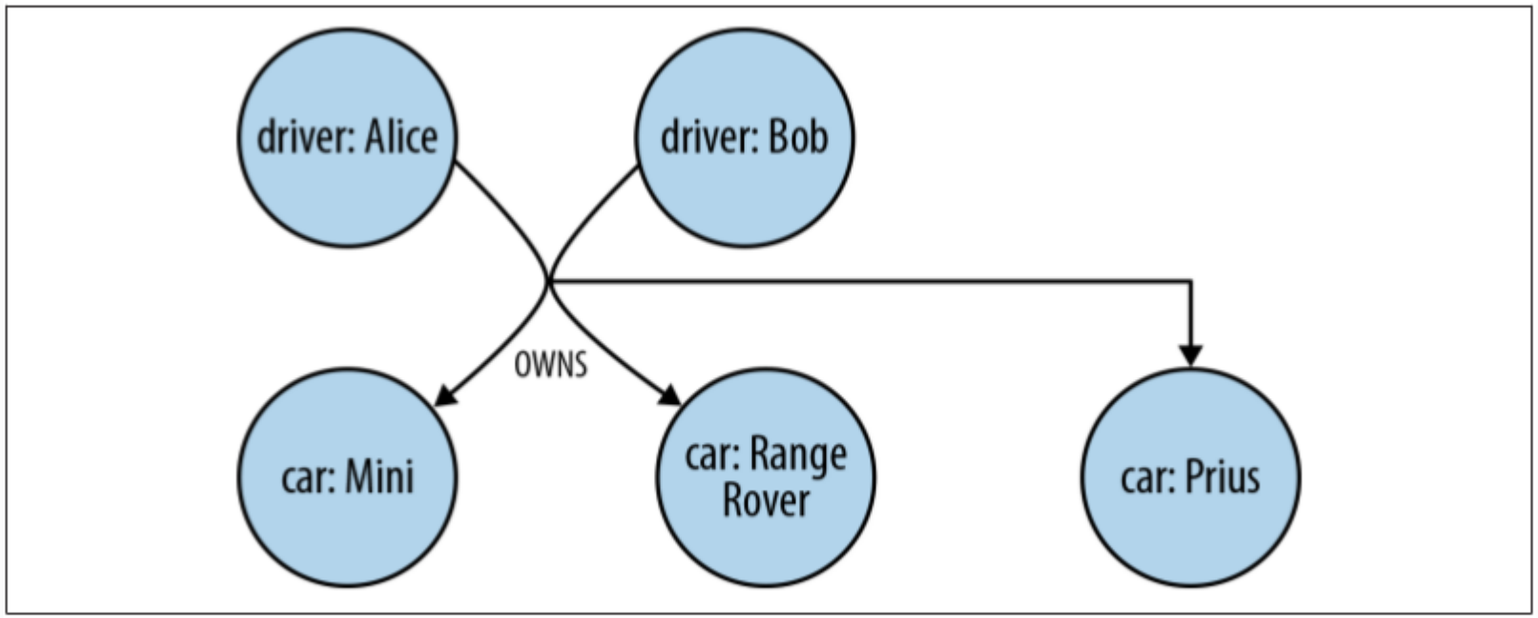


Abb. 2.2

Graphdatenbanken – Use Cases

- Social Networks
- Recommendation Engine
- Geospatial / Logistik
- Netzwerk und Datencenter Management

III. Neo4J

Neo4J - Einführung

- Open Source NoSQL Graphdatenbank
- Entwicklung seit 2003, Veröffentlichung 2007
- Nutzt das Property Graph Model
- Cypher als Abfragesprache
- Geschrieben in Java

Neo4J Modi

- Embedded Mode:
  - Läuft im Anwendungsprozess
  - Kein Netzwerk-Overhead
  - Zugriff auf Core API
- Server Mode:
  - REST API
  - Plattformunabhängig

Neo4J in Aktion

Freunde von Freunden:

- Max. Tiefe von 5
- Soziales Netz mit 1.000.000 Personen
- Jede Person hat ca. 50 Freunde

Neo4J in Aktion

Depth	RDBMS execution time(s)	Neo4j execution time(s)	Records returned
2	0.016	0.01	~2500
3	30.267	0.168	~110,000
4	1543.505	1.359	~600,000
5	Unfinished	2.132	~800,000

Abb. 3.1

Cypher

- Deklarative Abfragesprache
- Ursprünglich entwickelt für Neo4J
  - Seit 2015 durch das openCypher Projekt für andere DB verfügbar
- Einfache Lesbarkeit durch ASCII-Art

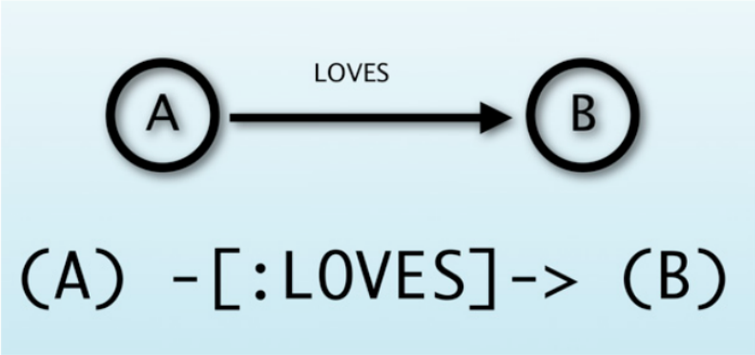


Abb. 3.2

Cypher

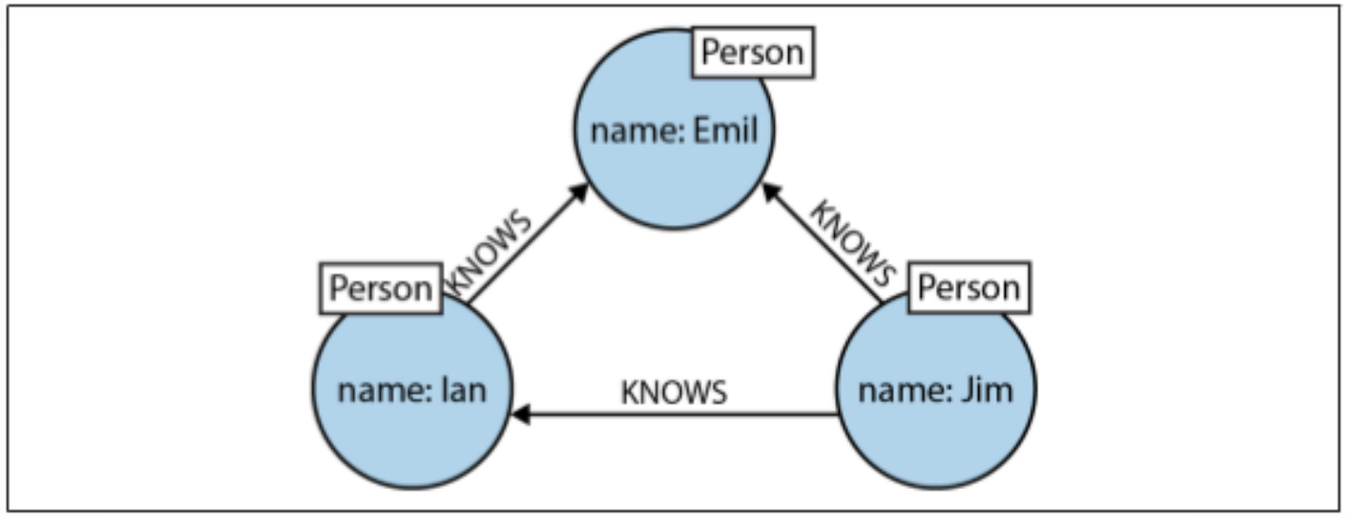


Abb. 3.3

```
CREATE (emil:Person {name:'Emil'})
       <-[:KNOWS]-(jim:Person {name:'Jim'})
       -[:KNOWS]->(ian:Person {name:'Ian'})
       -[:KNOWS]->(emil)
```

Cypher

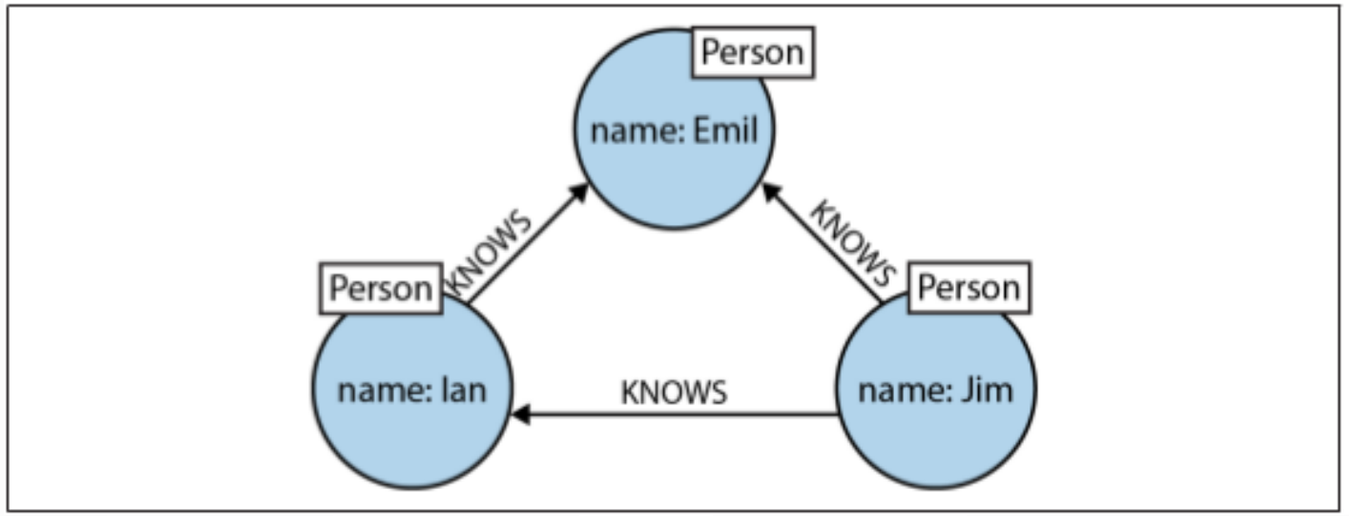


Abb. 3.3

```
MATCH (a:Person)-[:KNOWS]->(b)-[:KNOWS]->(c),
      (a)-[:KNOWS]->(c)
WHERE a.name = 'Jim'
RETURN b, c
```

Cypher

- **`WHERE`**: Kriterien zum Filtern von Ergebnissen
- **`CREATE`**: Erstellt Knoten und Beziehungen
- **`[DETACH]DELETE`**: Löscht Knoten und Beziehungen
- **`SET`**: Setzt Eigenschaften
- **`REMOVE`**: Löscht Eigenschaften
- **`WITH`**: Verkettet aufeinander folgende Anfrage-Abschnitte und leitet diese weiter (ähnlich wie Unix-Pipe)

Cypher HTTP API

Abfrage: **POST** /db/data/transaction/commit

```
{"statements": [{"statement":"MATCH (u:Person)
RETURN u"}]}
```

Ergebnis: ==> 200 OK

```
{"results":
 [{"columns":["u"],
 "data":[{"row":[{"login":"Peter"}]}]}],
 "errors":[ ]}
```

IV. Panama Papers

Panama Papers

- Daten von ‚Mossack Fonseca', Anwaltskanzlei aus Panama
- 140 Politiker auf über 50 Staaten
  - Darunter 12 ehemalige Staatsoberhäute
- Mehr als 500 Banken involviert
- 16.000 Unternehmen für Steuerhinterziehung gegründet

Panama Papers

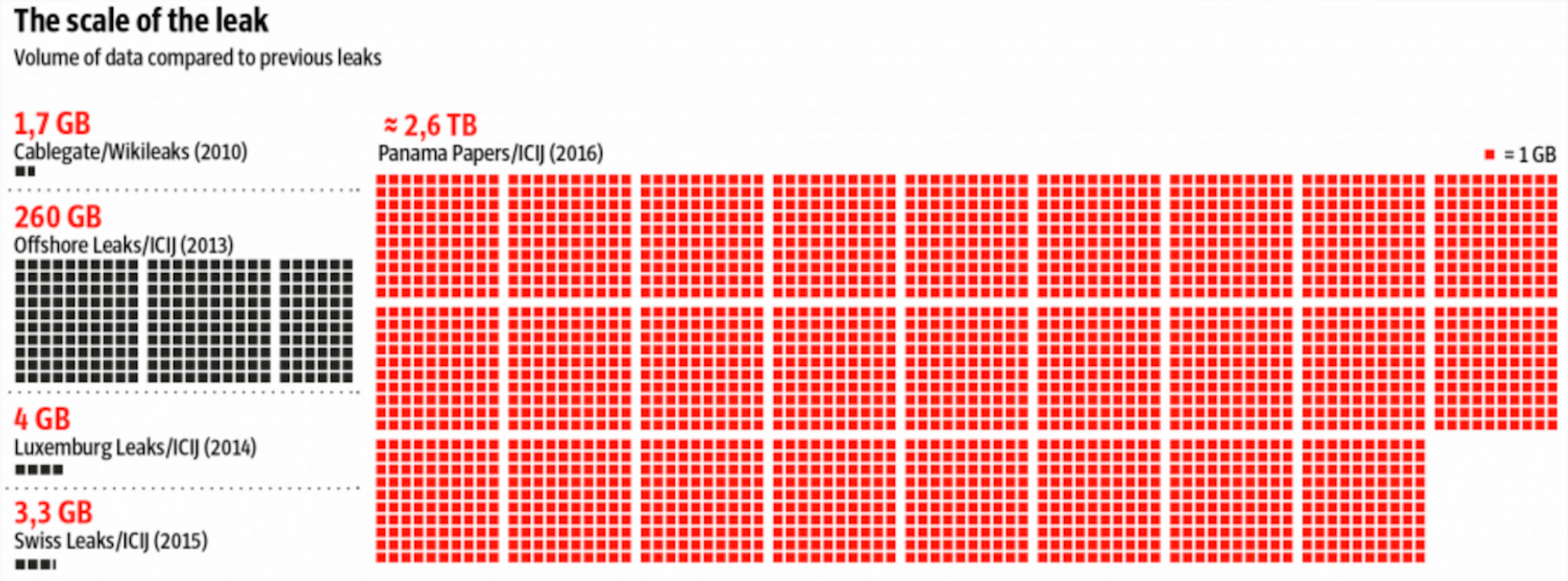


Abb. 4.1

Panama Papers

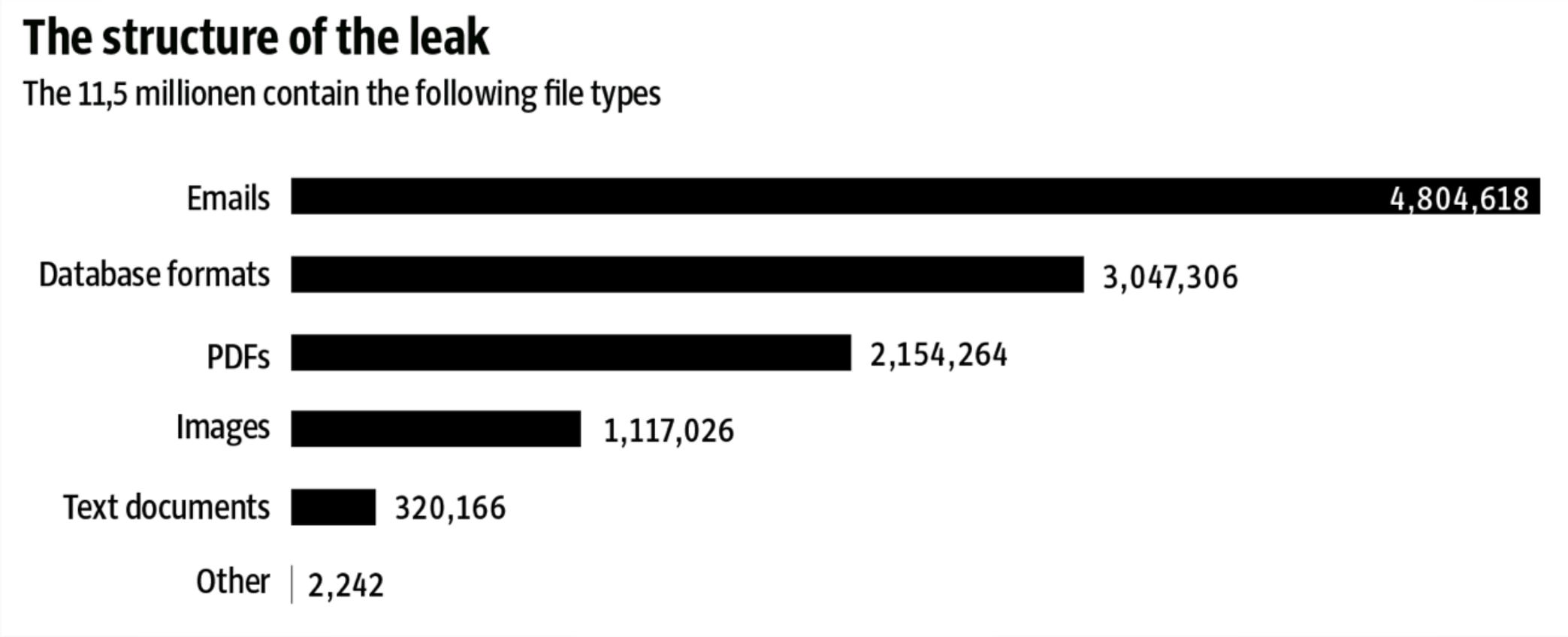
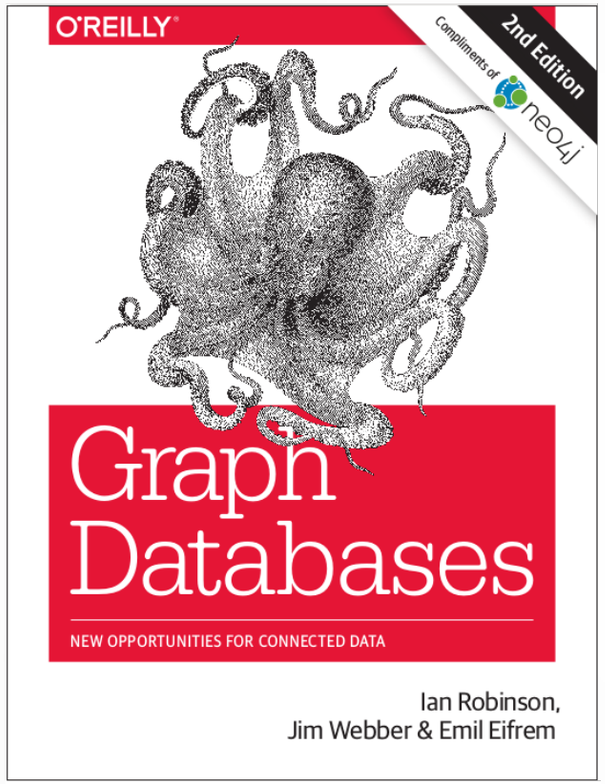


Abb. 4.2

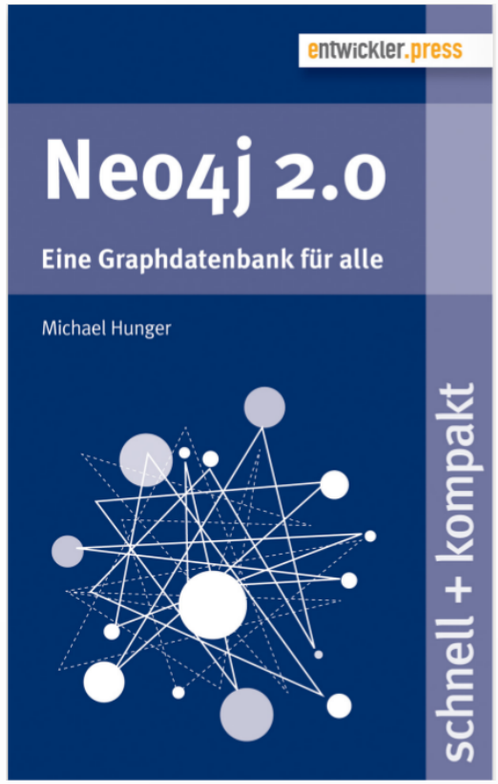
Panama Papers Analyse Ablauf

1. Dokumente akquirieren
2. Dokumente klassifizieren
   a) Scan / OCR
   b) Metadaten extrahieren
3. Whiteboard
   a) Entitäten und deren Beziehungen bestimmen
   b) Potenzielle Entitäts- und Beziehungs-Eigenschaften ermitteln
4. Analyseprogramme/Regeln/Parser für Dokumente entwickeln
5. Dokumente, Metadaten und Entitäts-Graphen abspeichern
6. Daten mit Hilfe von Graph Abfragen und Visualisierungstools analysieren

Literaturempfehlung



Robinson, Webber & Eifrem:
Graph Databases



Michael Hunger:
Neo4j 2.0 - Eine Graphdatenbank für alle

Quellen - Grafiken

[Abb. 0.1]	http://www.dataintensity.com/characteristics-of-big-data-part-one/
[Abb. 1.1]	https://en.wikipedia.org/wiki/Key-value_database
[Abb. 1.2]	http://alexminnaar.com/building-a-shoutbox-app-with-cassandra-and-nodejs.html
[Abb. 1.3]	http://greachconf.com/speakers/jennifer-strater-no-nonsense-nosql/
[Abb. 2.1]	*Robinson, Webber & Eifrem:* Graph Databases. O'Reilly Media, Inc., 2015, 2. Auflage CA, USA. Seite 3
[Abb. 2.2]	*Robinson, Webber & Eifrem:* Graph Databases. O'Reilly Media, Inc., 2015, 2. Auflage CA, USA. Seite 207
[Abb. 3.1]	*Robinson, Webber & Eifrem:* Graph Databases. O'Reilly Media, Inc., 2015, 2. Auflage CA, USA. Seite 21
[Abb. 3.2]	*Michael Hunger*: Neo4j 2.0. entwickler.press, 2014., 1. Auflage Paderborn, DE. Seite 13
[Abb. 3.3]	*Robinson, Webber & Eifrem:* Graph Databases. O'Reilly Media, Inc., 2015, 2. Auflage CA, USA. Seite 28
[Abb. 4.1]	https://neo4j.com/blog/icij-neo4j-unravel-panama-papers/
[Abb. 4.2]	https://neo4j.com/blog/analyzing-panama-papers-neo4j/

Vielen Dank für die Aufmerksamkeit.