

Virtuelle Vorlesungsreihe Wirtschaftsinformatik

#vivowi

Sommersemester 2021

Organisation:

Prof. Dr. Christian Drumm, drumm@fh-aachen.de

Prof. Dr. Michael Gröschel, m.groeschel@hs-mannheim.de



SCAN ME

Veranstaltungen und Terminplan

12.4.2021	Portalverbund, Bürgerkonto und Onlinezugangsgesetz - der weite Weg zur digitalen Verwaltung	Prof. Dr. Gunnar Auth
19.4.2021	Was Sie schon immer über Digitalisierung wissen wollten aber bisher nicht zu fragen wagten.	Prof. Dr. Stephan Jacobs
3.5.2021	Design Thinking und Innovation	Prof. Dr. Mahsa Fischer
17.5.2021	In welcher Sprache kommunizieren Webseiten mit Maschinen? - From Scraping to Parsing.	Prof. Dr. Vera G. Meister
31.5.2021	To SQL or not to SQL, that is the query - Wann lohnt sich der Einsatz von NoSQL?	Prof. Dr. Oliver Hummel
14.6.2021	Wenn Sie denken, IT-Sicherheit ist kostspielig, dann versuchen Sie es mal ohne.	Prof. Dr. Achim Schmidtman
28.6.2021	Anwendungsfälle der Künstlichen Intelligenz	Prof. Dr. Raphael Volz

Organisatorische Hinweise

- Zeitrahmen: Vortrag 45min, Diskussion 10min
- Teilnehmer sind stumm geschaltet, können aber Fragen per Chat stellen
- Fragen werden am Ende moderiert besprochen.
- Vortrag wird aufgezeichnet und später auf YouTube bereitgestellt.
 - Playlist: https://www.youtube.com/playlist?list=PLoHu_WG_4r3VPd6fwMizSf7XQ4MnTjLUb

Prof. Dr. Oliver Hummel

- Oliver Hummel... seit März 2017 Professor für Big Data
 - *davor als Entwicklungsleiter der IQSER GmbH für die Entwicklung einer Big-Data-Middleware verantwortlich*
- Leiter des neuen StartupLabs der Hochschule
 - <https://www.space.hs-mannheim.de>
- Weitere Stationen
 - Vertretungsprofessor am KIT
 - Juniorprofessur an der Uni Mannheim
 - Consulting für Perot Systems
 - Promotion an der Uni Mannheim
 - Informatikstudium an der TU Kaiserslautern

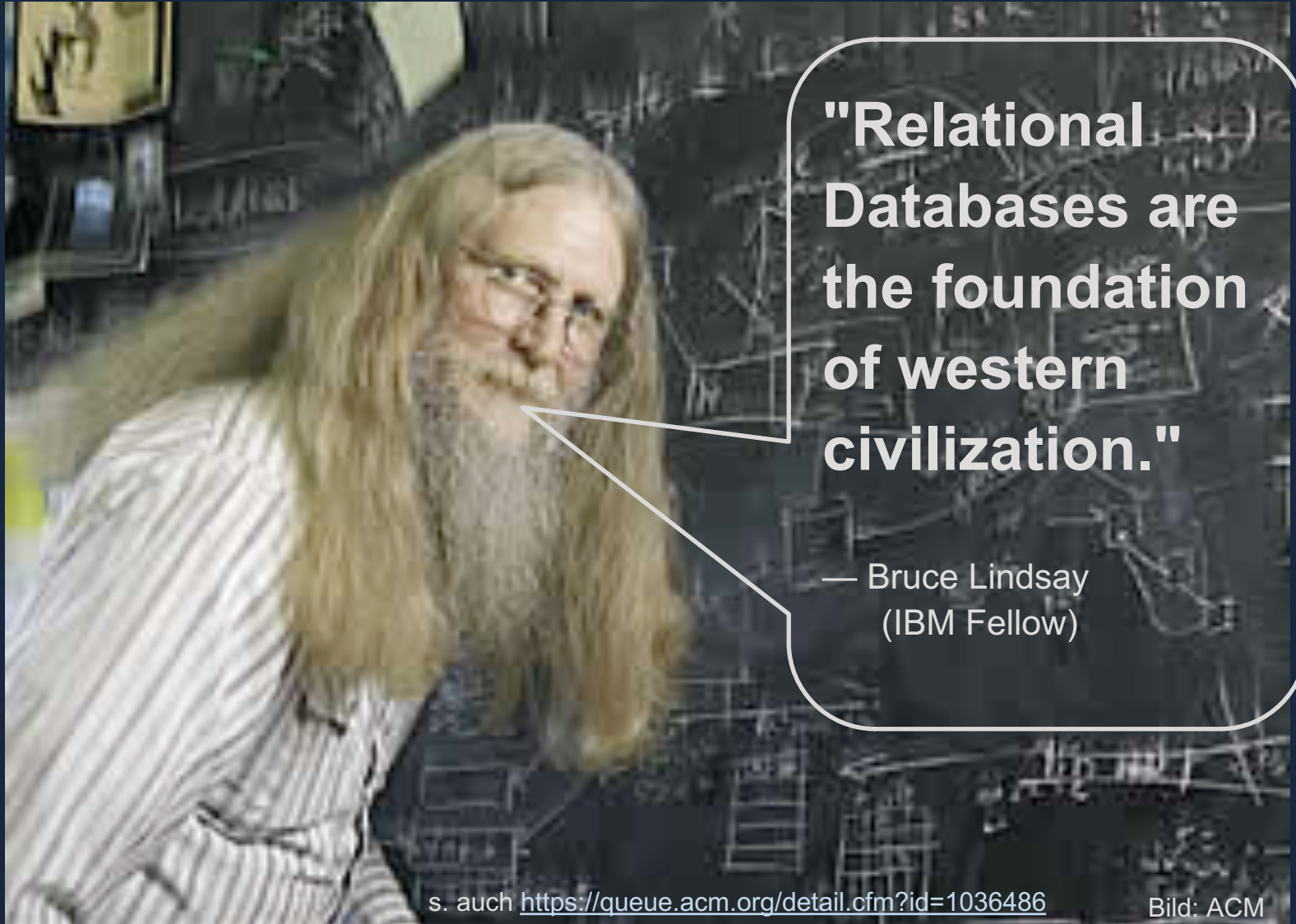
To SQL or not to SQL, that is the query – Wann lohnt sich der Einsatz von NoSQL?

#vivowi

Aufzeichnung startet jetzt!

31.5.2021, 17-18 Uhr

Referent: Prof. Dr. Oliver Hummel, Hochschule Mannheim





Und dann (ca. 2005) kam Big-Data...

Plötzlich nicht mehr nur SQL, sondern
NotOnly SQL



„Welches Datenbänkerl
hätten's denn gern?“

„Und wenn ja, wie viele?“

- evtl. polyglot persistence notwendig





Die weltweit verfügbare Datenmenge wächst kontinuierlich

- Verdopplung etwa alle zwei Jahre
- bis 2025 auf ca. 175 Zettabyte
 - = 175 000 000 000 000 000 000 000 Byte
 - = 35 Mrd. 5 TB Festplatten → gestapelt ca. 1,3 mal die Entfernung Erde-Mond

→ Big Data ist zum geflügelten Wort geworden, bekannt ist vor allem die folgende Definition mit 3 bis 4 Vs:

- 1. Volume → TB
- 2. Velocity
- 3. Variety
- 4. Veracity

→ kurz zusammengefasst:

Big Data befasst sich mit
der Verarbeitung von Datenmengen

an der Grenze des technologisch Machbaren

vgl.: <https://www.statista.com/statistics/871513/worldwide-data-created>



... Big Data wird es, wenn es nicht mehr auf eine Kiste passt





Bitte um Vorschläge und Abstimmung unter:

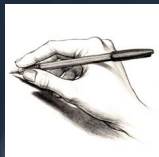
<https://pollev.com/oliverhummel921>

ACID, Konsistenz, Integrität

Schwierigkeiten dadurch?

→ vor allem diese Beiden:

- Schnelligkeit
- Skalierbarkeit



- WHERE-Klauseln können sich immer nur auf eine Zeile einer Tabelle beziehen
 - wir benötigen sog. Fremdschlüssel

Professoren			
PersNr	Name	Rang	Raum
2125	Sokrates	W3	226
2126	Russel	W3	232
2136	Curie	W3	36
2137	Kant	W3	7
2127	Kopernikus	W2	310
2133	Popper	W2	52
2134	Augustinus	W2	309

Vorlesungen			
VorlNr	Titel	SWS	Gelesen Von
5001	Grundzüge	4	2137
5041	Ethik	4	2125
5043	Erkenntnistheorie	3	2126
5049	Mäeutik	2	2125
4052	Logik	4	2125
5052	Wissenschaftstheorie	3	2126
5216	Bioethik	2	2126
5259	Der Wiener Kreis	2	2133
5022	Glaube und Wissen	2	2134
4630	Die 3 Kritiken	4	2137

Beispiel: [Kemper & Eickler: Datenbanksysteme]

Für Abfragen über mehrere Tabellen müssen diese zusammengeführt („gejoint“) werden

PersNr	Name	Rang	Raum	VorlNr	Titel	SWS	gelesenVon
2125	Sokrates	C4	226	5001	Grundzüge	4	2137
2125	Sokrates	C4	226	5041	Ethik	4	2125
...	...	...	...	...	...	...	...
2125	Sokrates	C4	226	5049	Mäeutik	2	2125
...	...	...	...	...	...	...	...
2126	Russel	C4	232	5001	Grundzüge	4	2137
...	...	...	...	...	...	...	...

→ Was bedeutet das in der Praxis bei großen Datenmengen?

Fun with databases: how to scale?

- with Dr. Sheldon Cooper

S1

Vorlesungen			
VorlNr	Titel	SWS	gelesen von
5001	Grundzüge	4	2137
5041	Ethik	4	2125
5043	Erkenntnistheorie	3	2126
5049	Mäeutik	2	2125
4052	Logik	4	2125
5052	Wissenschaftstheorie	3	2126
5216	Bioethik	2	2126
5259	Der Wiener Kreis	2	2133
5022	Glaube und Wissen	2	2134
4630	Die 3 Kritiken	4	2137

- Synchronisierung
- Erreichbarkeit
- Suche

S2

Professoren			
PersNr	Name	Rang	Raum
2125	Sokrates	W3	226
2126	Russel	W3	232
2136	Curie	W3	36
2137	Kant	W3	7
2127	Kopernikus	W2	310
2133	Popper	W2	52
2134	Augustinus	W2	309

Joins?





>> Everything gets faster in memory <<

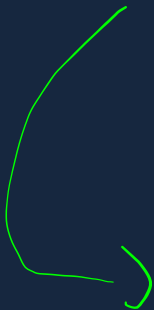
Soll heißen:

- ein entsprechendes **Caching** oder eine tatsächlich in-memory Nutzung versprechen deutliche **Geschwindigkeitsgewinne**
- Sie bringen aber auch neue Probleme mit sich
 - sehr hohe Kosten bei großen Datenmengen
 - Was passiert bei einem Hardware-Fehler (Absturz)?



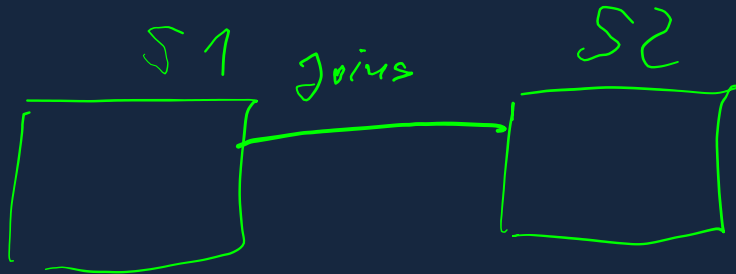
... blinding performance depends on **removing overhead**.

Such overhead has nothing to do with SQL, but instead revolves around traditional implementations of ACID transactions, multi-threading, and disk management.



<https://cacm.acm.org/blogs/blog-cacm/50678-the-no-sql-discussion-has-nothing-to-do-with-sql/fulltext>

Was ist die zentrale Idee von NoSQL?



- De-Normalisierung
- Query-Driven





1. Relational SQL vs. ...

- inkl. ggf. auch column-oriented DBs, **nicht zu verwechseln mit 4**

2. Key Value Stores

NoSQL

- meist nur einfache CRUD-Operationen für Key-Value-Paare
 - auch in memory möglich → *verteilter HashTable*
- Value ist meist mit client-seitigem „Schema on Read“

3. Document Databases

- im Prinzip Key Value Stores mit Schema on Write

4. Wide-Column Stores

- zweistufige Key-Struktur mit Row Keys und ...
- ... Key-Value-Paaren in den Columns



5. Sonstige: Distributed File Systems, Graph Databases, Time Series DBs

High Performance

- de-normalisierte Dokumente ersparen Joins & I/O
- Gute Indexierungsmöglichkeiten
- In-Memory Storage Engine möglich

Reiche Abfragesprache

- Daten-Aggregationen
- auch für Text- und Geodaten-Suche

Gute Verfügbarkeit

- durch Failover und Redundanz
- Horizontale Skalierbarkeit durch Sharding
 - *(anfangs) keine Joins!*

Vergleich zu MySQL: <https://www.simform.com/mongodb-vs-mysql-databases>



Document

Key-Value-Pair

```
{  
  name: "sue",  
  age: 26,  
  status: "A",  
  groups: [ "news", "sports" ]  
}
```

← field: value
← field: value
← field: value
← field: value

Liste 🤖

```
{
  "_id" : ObjectId("54c955492b7c8eb21818bd09"),
  "address" : {
    "street" : "2 Avenue",
    "zipcode" : "10075",
    "building" : "1480",
    "coord" : [ -73.9557413, 40.7720266 ]
  },
  "borough" : "Manhattan",
  "cuisine" : "Italian",
  "grades" : [
    {
      "date" : ISODate("2014-10-01T00:00:00Z"),
      "grade" : "A",
      "score" : 11
    },
    {
      "date" : ISODate("2014-01-16T00:00:00Z"),
      "grade" : "B",
      "score" : 17
    }
  ],
  "name" : "Vella",
  "restaurant_id" : "41704620"
}
```

Primärschlüssel

eingebettetes Objekt

Liste von Bewertungen



Sharding ist eine bestimmte Art,
Daten auf mehrere Knoten zu verteilen

- nämlich **horizontal partitioniert**
- jeder **Shard** (dt. **Splitter**) enthält einen Teil der Dokumente
- die Zuteilung erfolgt auf Basis eines festzulegenden **Shard Key**

Shards heißen in Mongo **Chunks** (Bereiche; Shards sind die Rechnerknoten)

- diese werden **gleichmäßig** auf die Knoten verteilt
- sog. mongos (Routing Server) kümmern sich um das Query Routing
- Shards sollten in einem Cluster **repliziert** werden (Datensicherheit)

Patient ID	Last Name	First Name	Personal Email	Work Email	Home Phone	Work Phone	Cell Phone	Emergency Contact Phone	Home Street	Home City	Home State	Home Zip	Work Street	Work City	Work State	Work Zip
1	Dan	Jane	jane@acme.com		555-1212		555-9090		800 Tenth St	Shoreville	VA	99999	700 Columa Row	Shoreville	VA	99999
2	Deer	James	james@sharding.net		555-2121		555-1111		800 Reaction Drive	Jordanstown	NC	99999				
3	April	Larsen	larsen@sharding.net		202-555-0115		202-555-0193		676 George Street	Sheltonville	WI	53081				
4	Denny	Tierney	tierney@shard.com		202-555-0107		202-555-0195		69 Coffee St	Harvardville	NY	11779				
5	John	Smith	john.smith@shard.com		202-555-0108		202-555-0196		672 St. James	Harvardville	NY	11779				
6	Cambridge	Yoshi	yoshi@acme.com		202-555-0109		202-555-0124		483 Brookshire Bl	Camelot	NY	10512				
7	Mary	Vernoy	vernoy@shard.com		202-555-0110		202-555-0188		412 E. Highland Lane	Hartsville	AL	36649				
8	Mia	Pluff	pluff@shard.com		202-555-0188		202-555-0188		492 Pearl Street	Saint Charles	IL	60174				
9	Raymon	Nyer	nyer@shard.com		202-555-0190		202-555-0190		21 St Louis Street	Clarksville	TN	37040	3 St. Logan St	Anaville	CO	80010
10	Kenna	Orwell	orwell@shard.com		202-555-0142				7120 Homestead St	Denton	AR	48423				
11	Luciano	Culshaw	culshaw@shard.com		202-555-0154		202-555-0170		21 Nichols Court	Newport	NY	10954				
12	Berni	Elroy	schwarz@shard.com		202-555-0162				8204 Eagle Drive	Rome	NY	13440				
13	Dagmar	Morano	morano@shard.com		202-555-0168				3 Green St	Saint Cloud	MN	56301				
14	Tina	Knudsen	knudsen@shard.com		614-555-0184		614-555-0188		370 Bay View Drive	Olney Branch	MS	39054				
15	Larry	Walton	walton@shard.com		614-555-0119				2 Phibson Avenue	Lynch	MS	48448	900 W Hudson St	Olympia	WA	98512
16	Julia	Marion	marion@shard.com		614-555-0125		614-555-0119		980 Sycamore St	Enfield	CT	8002				
17	Ernie	Essewanger	esewanger@shard.com		614-555-0118		614-555-0119		77 Holmes Court	Eden Prairie	MN	55347				
18	Johnnie	Duffee	duffee@shard.com		614-555-0116		614-555-0128		94 Charles Rd	Millersville	PA	8004				
19	Bertie	Brandis	brandis@shard.com		614-555-0187				11 Olive Drive	Stout Falls	SD	57103				
20	April	Larsen	larsen@shard.com		202-555-0115				676 George Street	Sheltonville	WI	53081				

Verwendung eines Clustering Key

- alles, was im Primary Key nach dem Partition Key kommt
- sorgt für die Sortierung der Spalten in der Zeile

traditionelle Darstellung

```
create table gyms_loc (
  state text, city text,
  name text, address text,
  primary key (state, city, name)
);
```

state	city	name	address
MO	Ozark	CrossFit 417	5108 N. 22nd St
MO	Springfield	CrossFit Provision	425 W. Plainview
MO	Springfield	CrossFit RTB	4234 S. Reed
MO	Springfield	CrossFit Springfield	1900 W. Sunset
MO	Springfield	Royal Barbell CrossFit	506 S. Jefferson

Wide-Row-Speicherung einer Partition

state: MO	city: Ozark name: Springfield 417 address: 5108 N. 22nd St	city: Springfield name: CrossFit Provision address: 425 W. Plainview	city: Springfield name: CrossFit RTB address: 4234 S. Reed	city: Springfield name: CrossFit Springfield address: 1900 W. Sunset	city: Springfield name: Royal Barbell CrossFit address: 506 S. Jefferson
--------------	--	--	--	--	--



Nicht wie gewohnt mit der Datenmodellierung beginnen

- E/R-Diagramm und Übertragung in ein SQL-Datenmodell

stattdessen: Query-Driven Data Modelling

- zuerst die Queries überlegen und sammeln
 - read vs. write vs. update vs. scale
- dann schauen, **welche DB** (-Familie) sich dafür eignet
 - Indexierung bei NoSQL meist sehr beschränkt (außer Document Stores)
 - ggf. keine Angst vor polyglot persistences
- und schließlich die **Daten spezifisch** dafür modellieren
 - dabei keine Angst vor De-Normalisierung



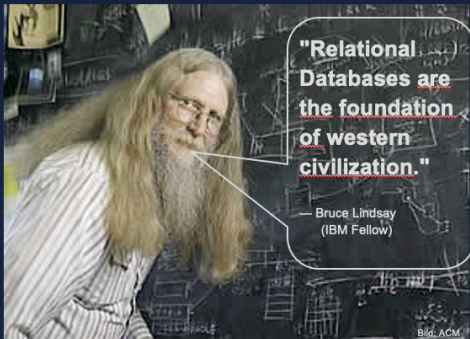
Welches Schweinderl hätten's denn gern?



Versuch einer Einordnung der verschiedenen Ansätze

- bitte ggf. mit gesundem Menschenverstand kritisch hinterfragen

Datenmodell	Skalierbarkeit	Latenz	Flexibilität	Funktionalität	(Betriebs-) Komplexität
Relational	Begrenzt	Hoch	Sehr hoch (mit Schema)	Sehr hoch	Sehr hoch
Key-Value	Sehr hoch	Sehr gering	Mittel	Sehr gering	Gering
Document Store	Hoch bis sehr hoch	Gering	Hoch	Mittel	Gering
Wide-Column Store	Sehr hoch	Gering bis mittel	Hoch	Gering	Gering-Mittel
Graph DB	Eher begrenzt	Gering bis mittel	Gering	Hoch	Mittel



... Big Data wird es, wenn es nicht mehr auf eine Kiste passt



Fun with databases: how to scale?

• with Dr. Sheldon Cooper

Notiz	Text	Werte	Ergebnis
5001	Grundzüge	4	2137
5001	Erk	4	2135
5003	Erkennungsrate	3	2129
5009	Mitwirk	2	2125
4002	Logik	4	2125
5002	Wissenschaftstheorie	3	2126
5209	Statistik	2	2129
5209	Der Welterbe	2123	
5002	Der Welterbe	154	
4009	Der Welterbe	2	



1. Relational SQL vs. ...
 - inkl. ggf. auch column-oriented DBs, nicht zu verwechseln
2. Key Value Stores
 - meist nur einfache CRUD-Operationen für Key-Value-Paare
 - auch in memory → verteilte
 - Value ist ...
3. Document Databases
 - im Prinzip Key-Value Stores
4. Wide-Column Stores
 - zweistufige Key-Struktur mit Row Keys und ...
 - ... Key-Value-Paaren in den Columns
5. Sonstige: Distributed File Systems, Graph Databases, Time Series DBs



Beispiel: JSON-Dokument für Museum

```

{
  "id": "Object1",
  "name": "Museum",
  "address": {
    "street": "123 Main St",
    "city": "New York",
    "state": "NY",
    "zip": "10001",
    "country": "USA"
  },
  "location": {
    "lat": 40.7128,
    "lon": -74.0060
  },
  "description": "A museum dedicated to the history of the city of New York.",
  "grades": [
    {
      "date": "2014-10-01T00:00:00Z",
      "grade": "A",
      "score": 11
    },
    {
      "date": "2014-11-01T00:00:00Z",
      "grade": "B",
      "score": 17
    }
  ],
  "restaurant_id": "41784620"
}

```

Primärschlüssel

eingebettetes Objekt

Liste von Bewertungen



Versuch einer Einordnung der verschiedenen Ansätze
• bitte ggf. mit gesundem Menschenverstand kritisch hinterfragen

Datenmodell	Skalierbarkeit	Latenz	Flexibilität	Funktionalität	(Betriebs-) Komplexität
Relational	Begrenzt	Hoch	Sehr hoch (mit Schema)	Sehr hoch	Sehr hoch
Key-Value	Sehr hoch	Sehr gering	Mittel	Sehr gering	Gering
Document Store	Hoch bis sehr hoch	Gering	Hoch	Mittel	Gering
Wide-Column Store	Sehr hoch	Gering bis mittel	Hoch	Gering	Gering-Mittel
Graph DB	Eher begrenzt	Gering bis mittel	Gering	Hoch	Mittel

Diskussion und Fragen

#vivowi

Wenn Sie denken, IT-Sicherheit ist kostspielig, dann versuchen Sie es mal ohne.

#vivowi

Nächster Vortrag!

14.6.2021, 17-18 Uhr

Referent: Prof. Dr. Achim Schmidtman, Fachhochschule Bielefeld

Vielen Dank für
Ihr Interesse!

#vivowi