

KÖNIGSWEG



Alexander Hendorf

# Databases for Data-Science



## Alexander C. S. Hendorf

- Senior Consultant Information Technology
- Program Chair EuroPython, PyConDE & PyData Karlsruhe, EuroSciPy, PSF Managing Member
- PyData Frankfurt and PyData Südwest Organiser
- Program Committee Percona Live
- MongoDB Masters / MongoDB Certified DBA
- Speaker Europe & USA MongoDB World New York / San José, PyCon Italy, CEBIT, BI Forum, IT-Tage FFM, PyData, PyParis, PyCon UK, Budapest BI,....



ah@koenigsweg.com  
 @hendorf

## Services



### Data

We guide our clients through development and implementation processes of technologies and applications to analyse, evaluate and visualize business data.

### Strategy & Operations

We advise SME, start-ups and public institutions on building efficient sales structures, on process optimization and on business development.

### Communication

We provide sound communication strategies and creative campaigns to communicate your content and messages authentically throughout all channels.

### Financial Service Technologies

Our industry experts support clients in the financial sector in developing powerful and compliant FinTech applications.



Let's get to know each other!

## » What's your background? «

- Data scientist
- Database admin
- Curious Pythonista
- Consultant, decision maker (IT Executive, Consultant, Innovation Manager, YouTube influencer)



Let's get to know each other!

## » What's your Experience in...«

- **RDBSs**
  - none
  - some
  - a lot
- **Hadoop et al.**
  - none
  - some
  - a lot
- **NoSQL Systems**  
without Hadoop et al.
  - none
  - some
  - a lot



**Let's get to know each other!**

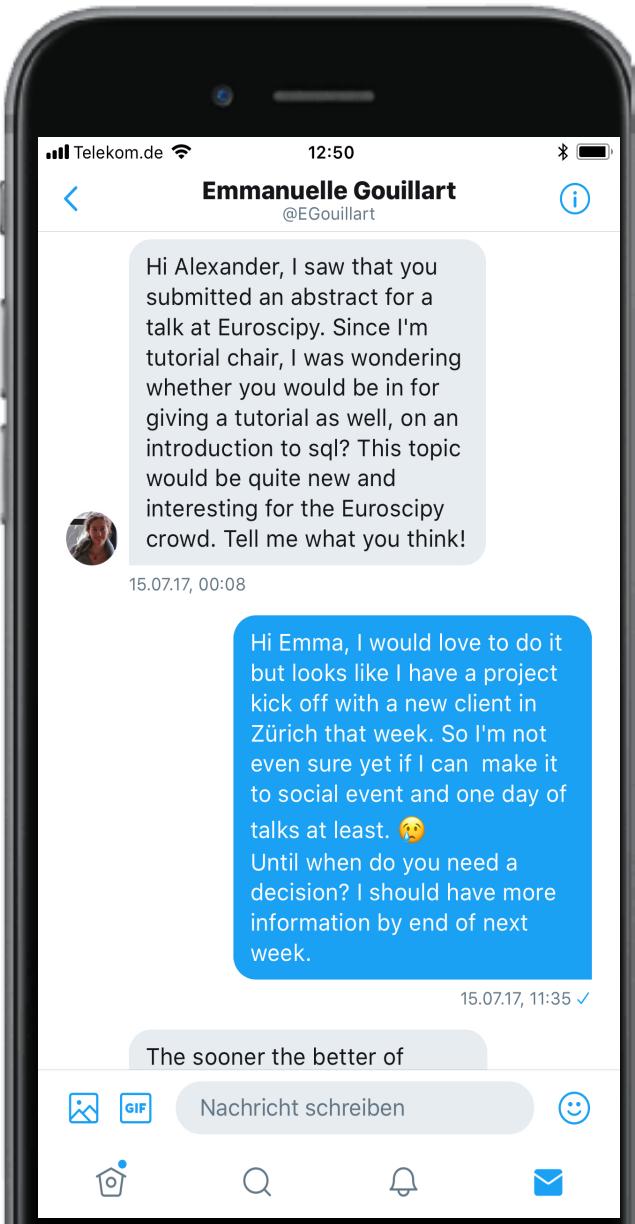
## **» What's are you looking for? «**

- Integration for data science into existing ecosystems
- Learning about databases for data science projects
- General interest / curiosity
- Just killing time until PyFiorentina
- ...?

## Three Angles for Databases for Data Science

---

- Choosing a database for data science projects
- Evaluating an existing database for data science requirements
- How to integrate into an existing ecosystem





-

*„How deeply do data science & data base understand each other?.“*



Ask Google:

## Data Scientist? Database Admin?





data scientist



Gespeicherte Inhalte ansehen SafeSearch ▾

infographic

m



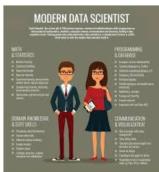
data scientist icon  
data scientist gehalt  
data scientist sexist job  
data scientist  
data scientist resume  
data scientist venn diagram  
data scientist infographic  
data scientist resume sample  
data scientist meme  
data scientist salary

m



m

m



m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

m

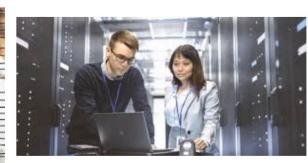
m

m

database administrator

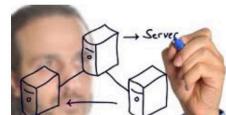
database administrator  
 database administrator salary  
 database administrator humor  
 database administrator resume entry level  
 database administrator clipboard  
 database administrator meme  
 database administrator in dbms  
 database administrator cover letter  
 database administrator cv  
 database administrator responsibilities

oracle database



DBA

Salary



Database Administrator

PRIMARY ROLES OF DBA

- Database design
- Database accessibility
- Performance issues
- Capacity issues

THIS IS YOUR MACHINE LEARNING SYSTEM?

YUP! YOU POUR THE DATA INTO THIS BIG  
PILE OF LINEAR ALGEBRA, THEN COLLECT  
THE ANSWERS ON THE OTHER SIDE.

WHAT IF THE ANSWERS ARE WRONG?

JUST STIR THE PILE UNTIL  
THEY START LOOKING RIGHT.



## What are the Benefits of a Database for Data Science Anyway?

---

- Common source of data
- Avoiding redundancy (e.g. files)
- Persistence
- Optimized for handling and accessing data for decades
- Scalability
- Staying very close to the data

## A Quick Recap on Database History

---

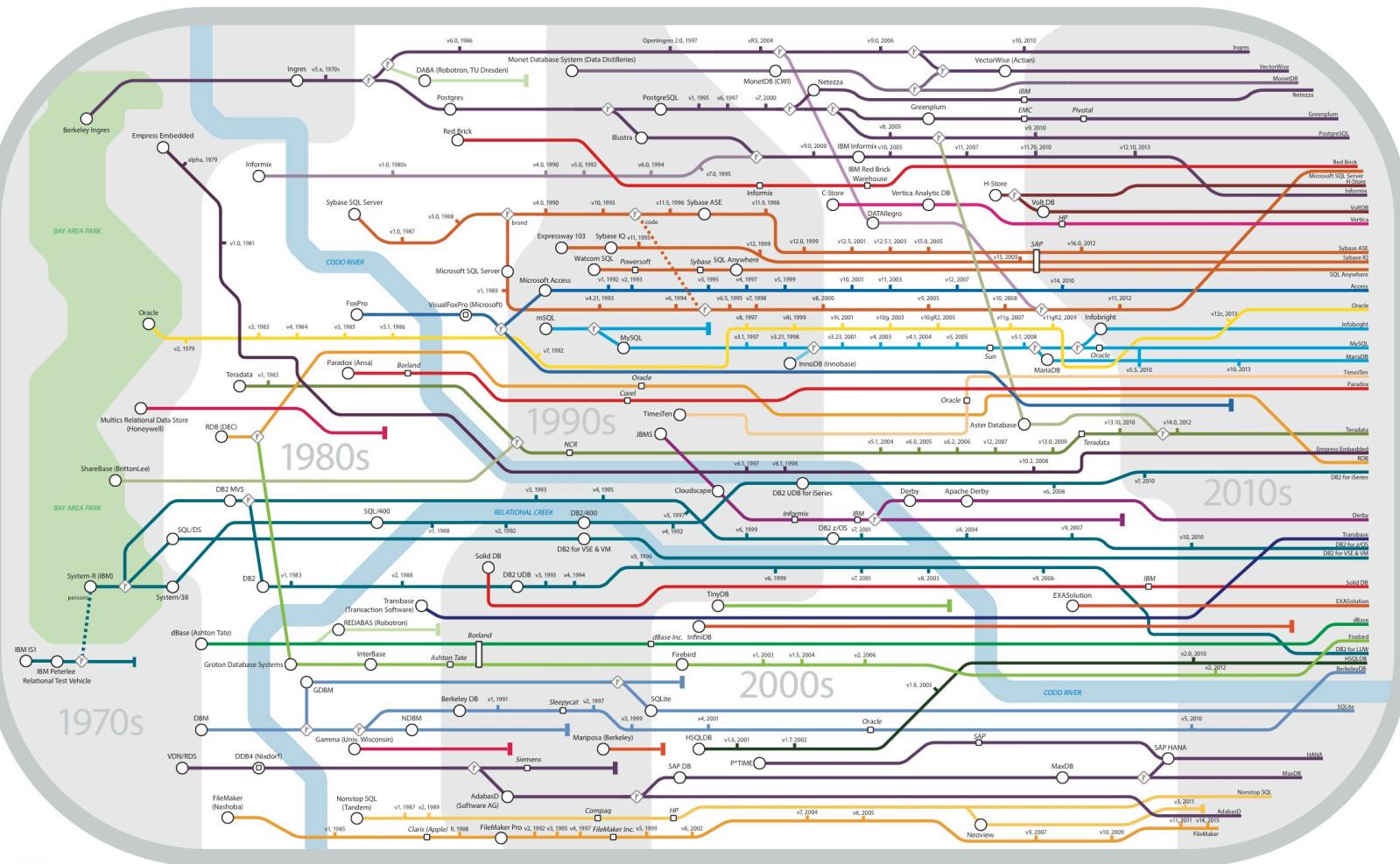
- **1960s, navigational DBMS (disks & drums)**
- **1970s, relational DBMS**
- **1980s, on the desktop**
- **1990s, object-oriented**
- **2000s, NoSQL**

## Relational Databases

---

- **Records are organised into tables**
- **Rows of these table are identified by unique keys**
- **Data spans multiple tables, linked via ids**
- **Data is ideally normalised**
- **Data can be denormalized for performance**
- **Transactions are ACID [Atomic, Consistent , Isolated, Durable]**

# Genealogy of Relational Database Management Systems



## HOW STANDARDS PROLIFERATE:

(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC)

SITUATION:  
THERE ARE  
14 COMPETING  
STANDARDS.

14?! RIDICULOUS!  
WE NEED TO DEVELOP  
ONE UNIVERSAL STANDARD  
THAT COVERS EVERYONE'S  
USE CASES.



SOON:

SITUATION:  
THERE ARE  
15 COMPETING  
STANDARDS.

## Relational Databases Benefits

---

- Widely used and supported
- Normalized data
- Comprehensive querying via SQL language
  - though some differences between databases
- Well researched and optimized over decades

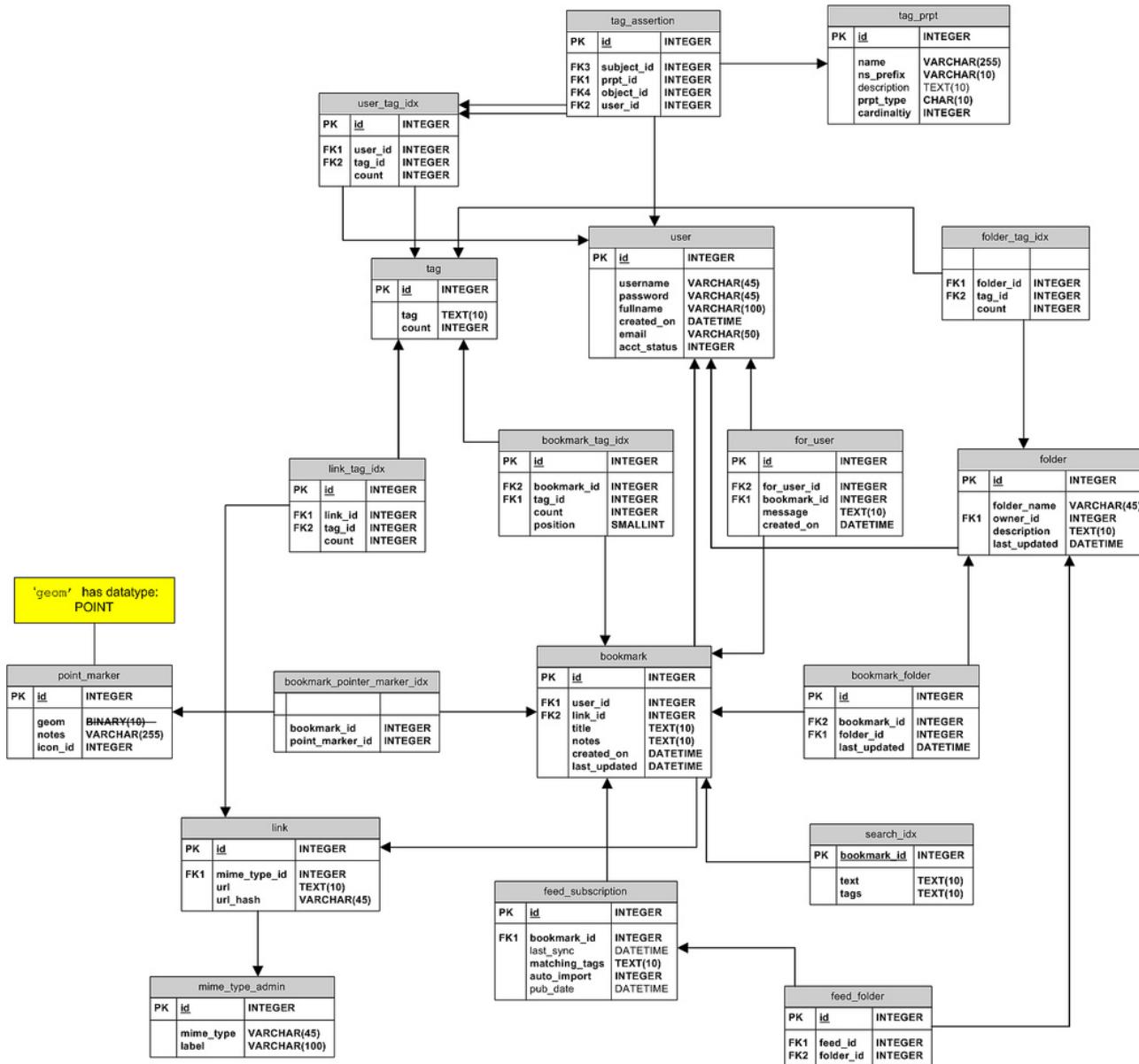
## Relational Model

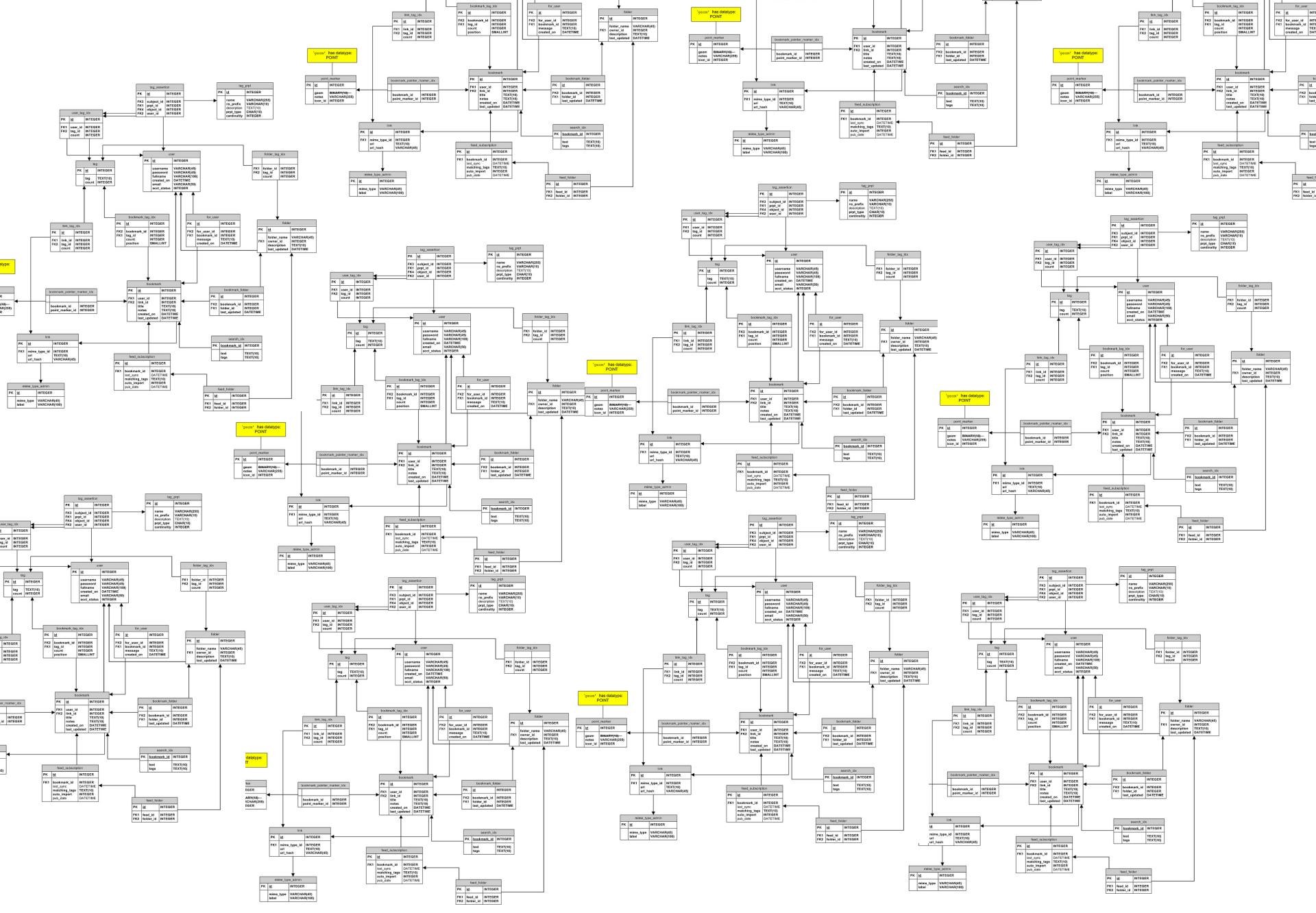
Activity Code	Activity Name
23	Patching
24	Overlay
25	Crack Sealing

Key = 24

Activity Code	Date	Route No.
24	01/12/01	I-95
24	02/08/01	I-66

Date	Activity Code	Route No.
01/12/01	24	I-95
01/15/01	23	I-495
02/08/01	24	I-66





## Relational Databases Downsides

---

- **Schemas are fixed and have to be pre-defined upfront (*schema-first*)**
- **Altering schema is not trivial**
- **Joining tables, depending on complexity, data volume may be costly, also consider overhead understanding a schema with many tables**
- **Difficult to scale out**
- **Few data structures (tables, rows)**

## NoSQL Types

---

- **Key-Value Store**  
simplest form of a NoSQL database (no big value for data science)
- **Document databases (JSON style)**  
open schema  
can handle complex data structures as arrays and list
- **Wide column databases, most like relational DBs:**  
columns are not fixed, data is de-normalised,  
can handle complex data structures as arrays and lists
- **Graph**  
network of connected entities linked by edges with properties, query on properties and links

## NoSQL Databases Benefits

---

- No need to normalise data (*schema-later*)
- Maintain complex data structures
- Supports data sharding
- New ways to query
- Collections can be copied

## NoSQL Databases Downsides

---

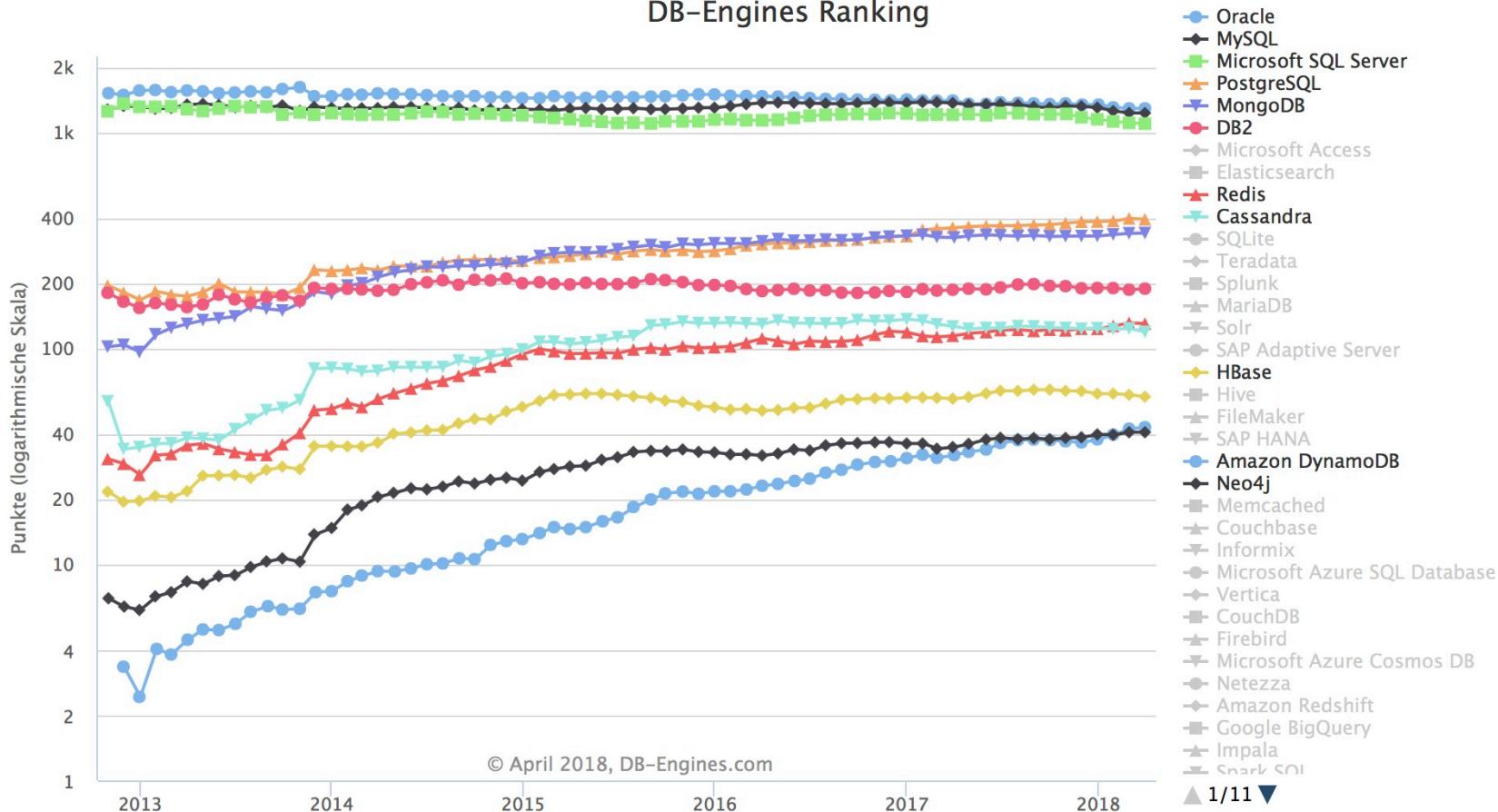
- Eventual Consistency (is this a real problem for data science at all?)
- Flexibility requires more responsibility (schema, attribute typos)
- Complexity

## A Quick Bird's Eye View

—

*There are hundreds of databases around nowadays.  
Let's focus on the top database systems.*

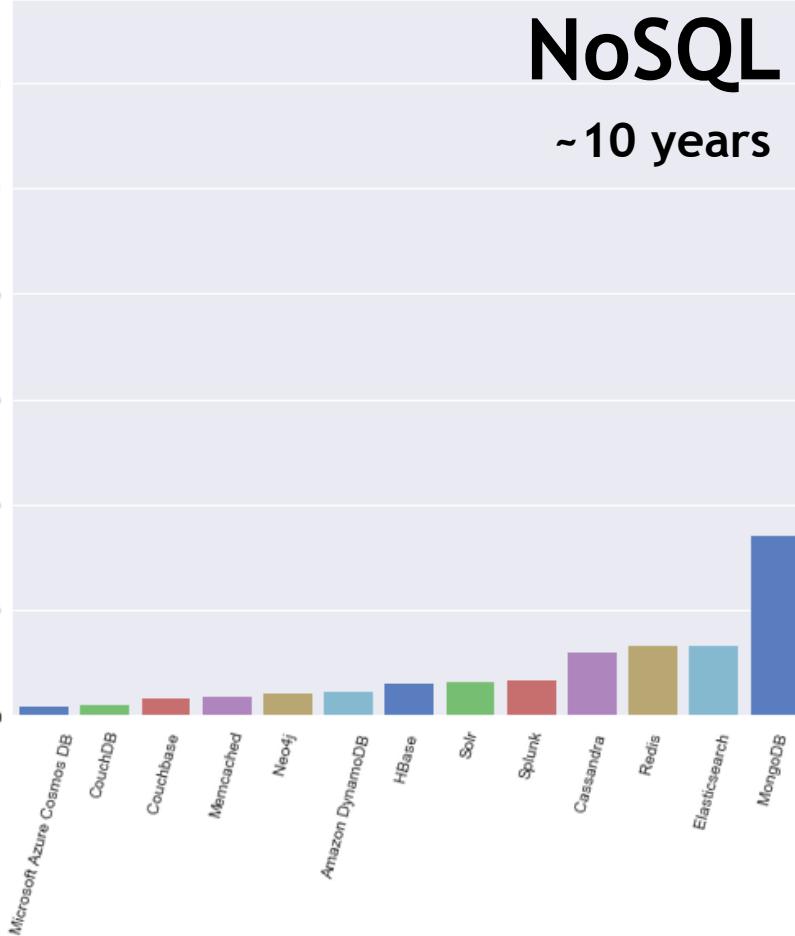
## DB-Engines Ranking



<https://db-engines.com>

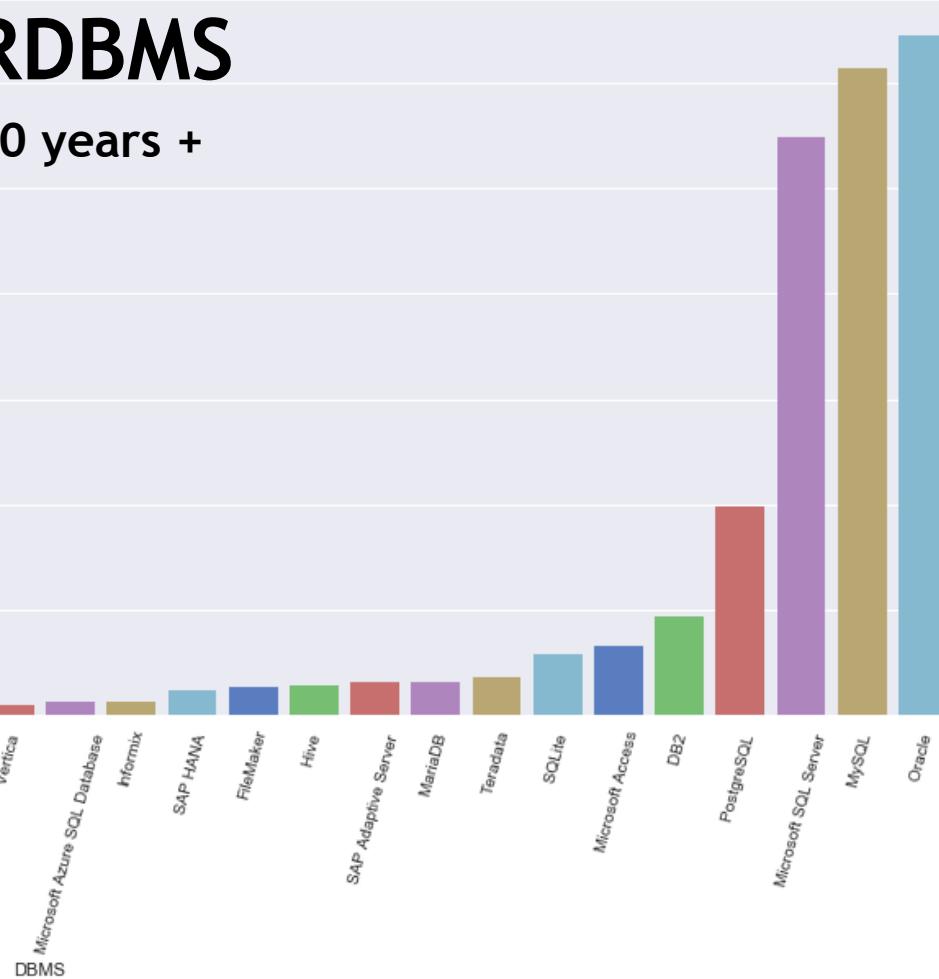
# NoSQL

~10 years



# RDBMS

40 years +



Points according to <https://db-engines.com/de/ranking>

## Consistency Models of Databases

---

- **A** tomicity
- **C** onsistency
- **I** solation
- **D** urability
- **B**
- **A**asic Availability
- **S**oft-State
- **E**ventual consistency

## Open Source Check

---

– **Security**

– **Transparency**

– **Engaging Collaboration**

– **Quality**

– **Auditability**

– **Try Before You Buy (EE)**

– **Rule of Thumb:**

**Open Software is way more  
affordable than closed**

\* via vendors

	Open Source	Enterprise Editions
Oracle	x	+
MySQL	+?!?	+
MicroSoft SQL Server		+
PostgreSQL	+	**
DB2		+
MongoDB	+	+
Redis	+	**
Cassandra	+	**
HBase	+	**
Amazon DynamoDB		DAAS
Neo4J	+	+

## The Contenders

	Type	Chosen
PostgreSQL	RDBMS	Top OS RDBMS
MongoDB	Document-store	Top NoSQL (DS)
Cassandra	Wide-column store	Top NoSQL (WCS)
Neo4J	Graph	Top NoSQL (Graph)

# Relational Database Management System

Relational Model

Activity Code	Activity Name
23	Patching
24	Overlay
25	Crack Sealing

Key = 24

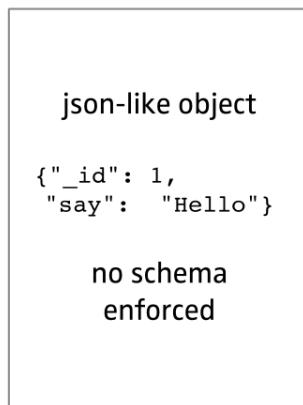
Activity Code	Date	Route No.
24	01/12/01	I-95
24	02/08/01	I-66

Date	Activity Code	Route No.
01/12/01	24	I-95
01/15/01	23	I-495
02/08/01	24	I-66

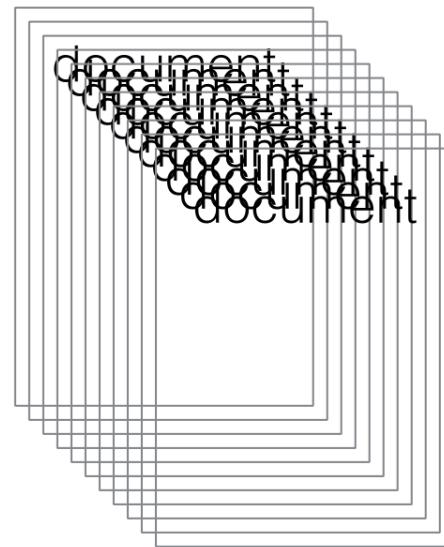
Dat  
aba  
se

# Document oriented databases in 15 seconds

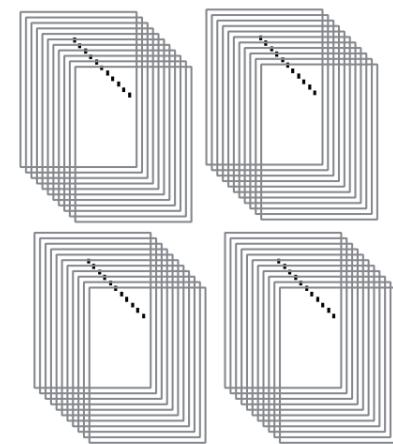
document



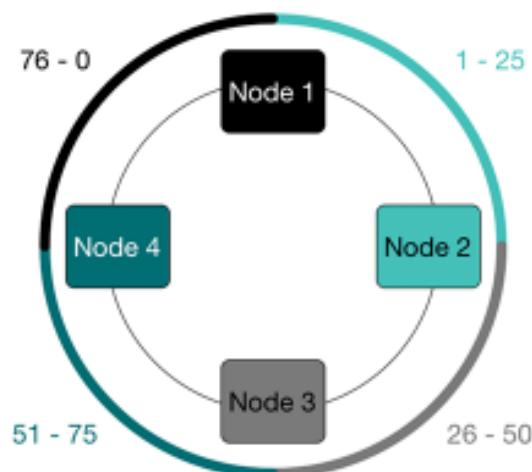
collection



database



# Cassandra



## KeySpace

### Column Family

Key	Column Name	Column Name	Column Name	
	Value	Value	Value	
Key	Column Name	Column Name	Column Name	
	Value	Value	Value	
Key	Column Name	Column Name	Column Name	Column Name
	Value	Value	Value	Value

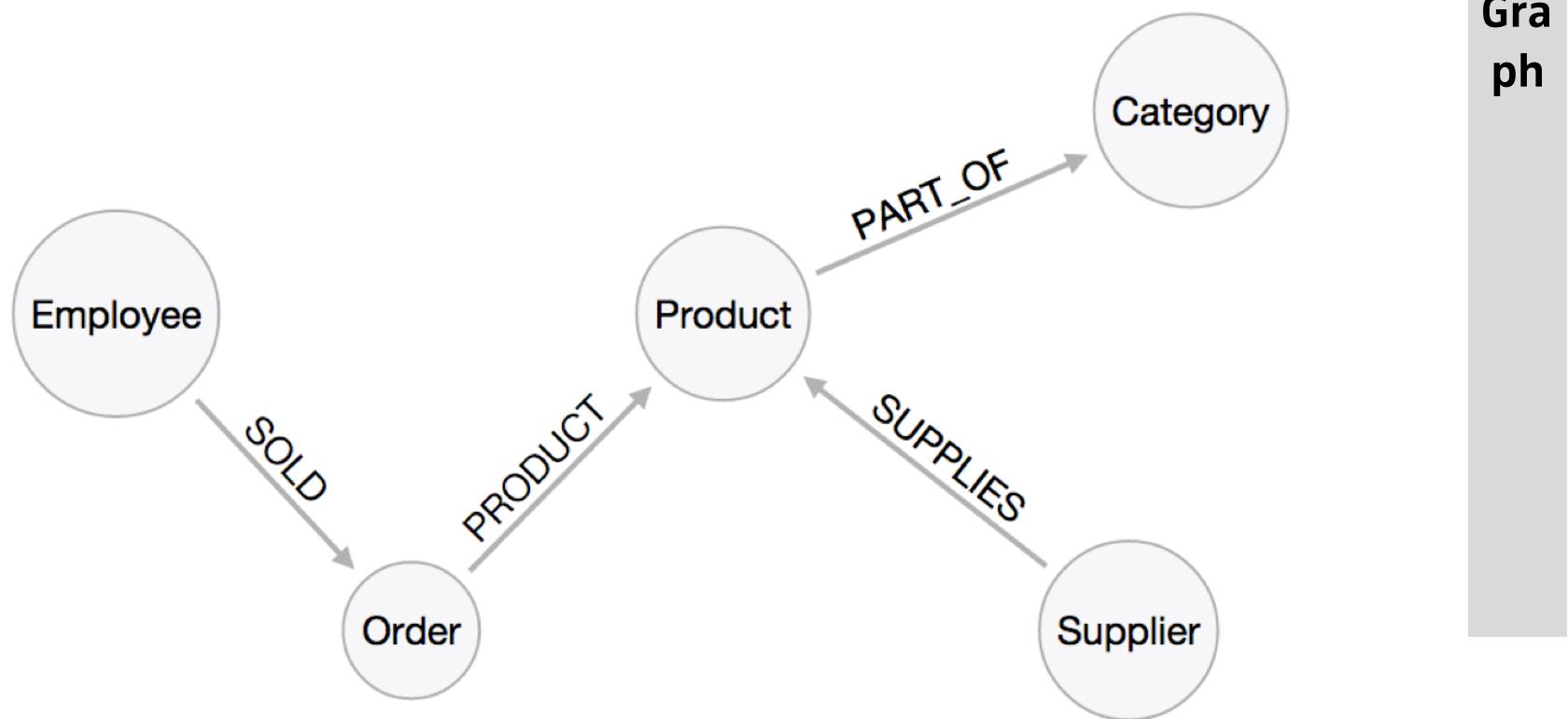
**Column**

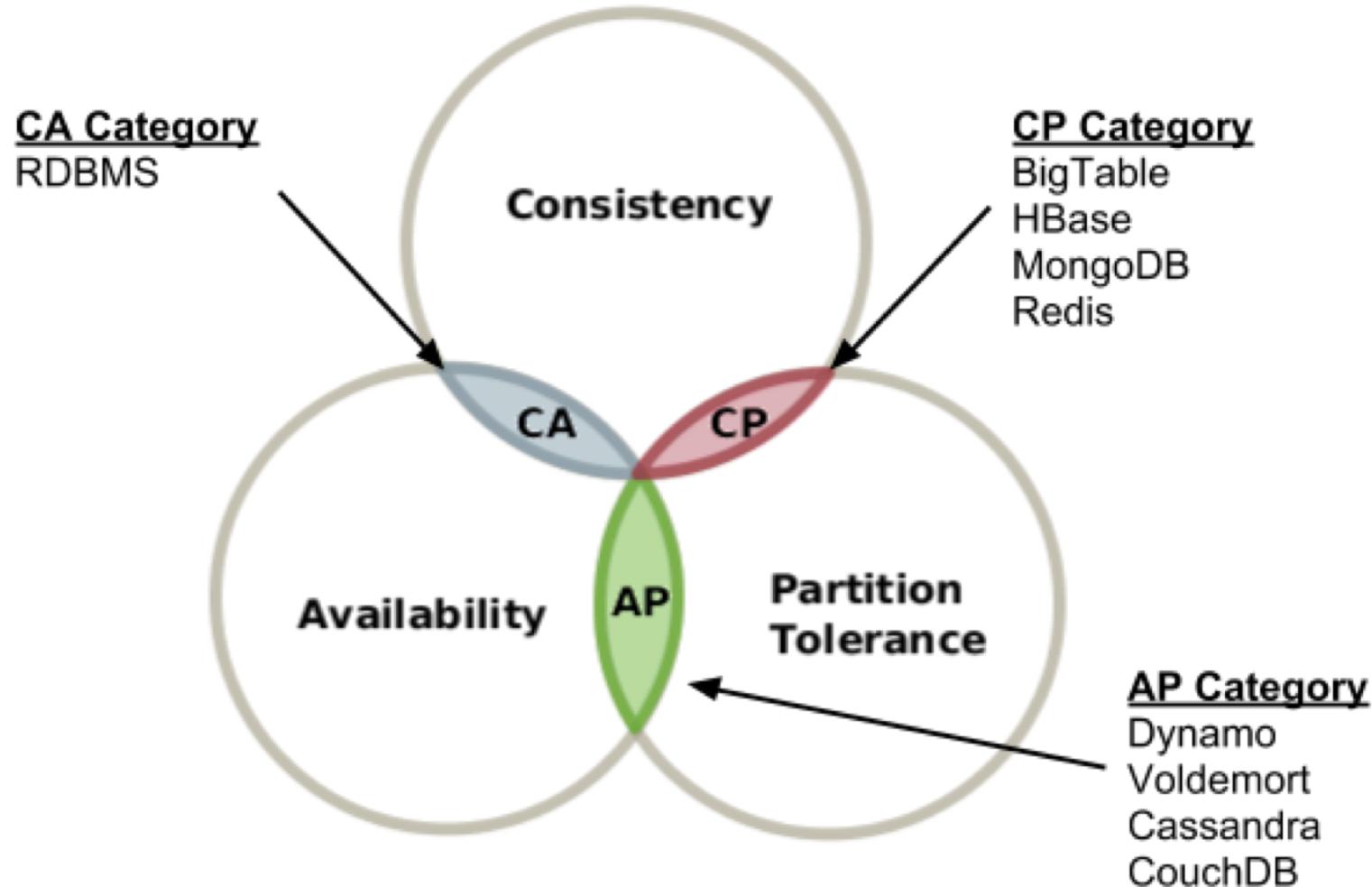
### Column Family

Key	Column Name	Column Name	Column Name
	Value	Value	Value
Key	Column Name	Column Name	Column Name
	Value	Value	Value

## KeySpace

## Graph Database





## How Hard is it to Collect Data?

---

	Data Collection, cleaning and restructuring	Multiple data sources	Data retention
PostgreSQL	Depends on schema complexity	Depends on schema complexity	easy
MongoDB	easy	easy	medium
Cassandra	easy	easy	hard
Neo4J	(easy)	N/A	easy

## What about Data Types

---

	enforced	flexible	enforceable
PostgreSQL	yes	(NoSQL feature)	predefined
MongoDB	possible	yes	yes
Cassandra	yes	untyped collection columns	predefined
Neo4J	(yes)	N/A	N/A

## How Hard is it to Consolidate Data?

---

	Linking	Missing data	Dirty data	Persisting cleaned dataset
PostgreSQL	built schema	pre-processing recommended	pre-processing recommended	easy
MongoDB	easy (within db)	flexible post-processing	flexible post-processing	easy
Cassandra	partitioning	hard	hard	easy
Neo4J	yes*	hard	hard	easy

## How Hard is it to Write Queries Against These Databases?

---

	Language	Basic Queries	Advanced Queries
PostgreSQL	SQL	easy	hard
MongoDB	MQL	query: easy aggregation: medium	query: medium aggregation: medium
Cassandra	CSQL	easy	hard
Neo4J	Cypher	easy	hard

## How Hard is Querying to Learn?

---

	Language	Basic Queries	Advanced Queries
PostgreSQL	SQL	easy	hard
MongoDB	MQL	query: easy aggregation: easy	query: medium aggregation: medium
Cassandra	CSQL	easy-medium	hard
Neo4J	Cypher	medium	hard

## SQL Benefits and Downsides

---

- Common standard
- Long-established
- Mother of many others e.g. CSQL, ABAP, Pig, SPARQL,...
- Set based logic
- Complexity increases fast
- Badly designed JOINs vs. performance
- Overhead understanding a large schema
- Set based logic

# SQL

```
SELECT EmployeeID, FirstName, LastName, HireDate, City  
FROM Employees  
WHERE HireDate BETWEEN '1-june-1992' AND '15-december-1993'
```

```
SELECT A.SD1, B.ED1 FROM  
  
(SELECT SD1, ROW_NUMBER() OVER (ORDER BY SD1) AS RN1 FROM (SELECT T1.Start_Date AS  
SD1, T2.Start_Date AS SD2 FROM (SELECT * FROM Projects ORDER BY Start_Date) T1  
  
LEFT JOIN (SELECT * FROM Projects ORDER BY Start_Date) T2  
  
ON T1.Start_Date=(T2.Start_Date+1)  
  
ORDER BY T1.Start_Date) WHERE SD2 IS NULL) A  
  
INNER JOIN  
  
(SELECT ED1, ROW_NUMBER() OVER (ORDER BY ED1) AS RN2 FROM (SELECT T1.End_Date AS  
ED1, T2.Start_Date AS SD2 FROM (SELECT * FROM Projects ORDER BY Start_Date) T1  
  
LEFT JOIN (SELECT * FROM Projects ORDER BY Start_Date) T2  
  
ON T1.End_Date=(T2.Start_Date) ORDER BY T1.Start_Date) WHERE SD2 IS NULL) B  
  
ON A.RN1=B.RN2  
  
ORDER BY (B.ED1-A.SD1), A.SD1;
```

# Cassandra

```
SELECT * FROM numberOfRequests  
WHERE cluster = 'cluster1'  
AND date = '2015-06-05'  
AND datacenter = 'US_WEST_COAST'  
AND (hour, minute) IN ((14, 0), (15, 0));
```

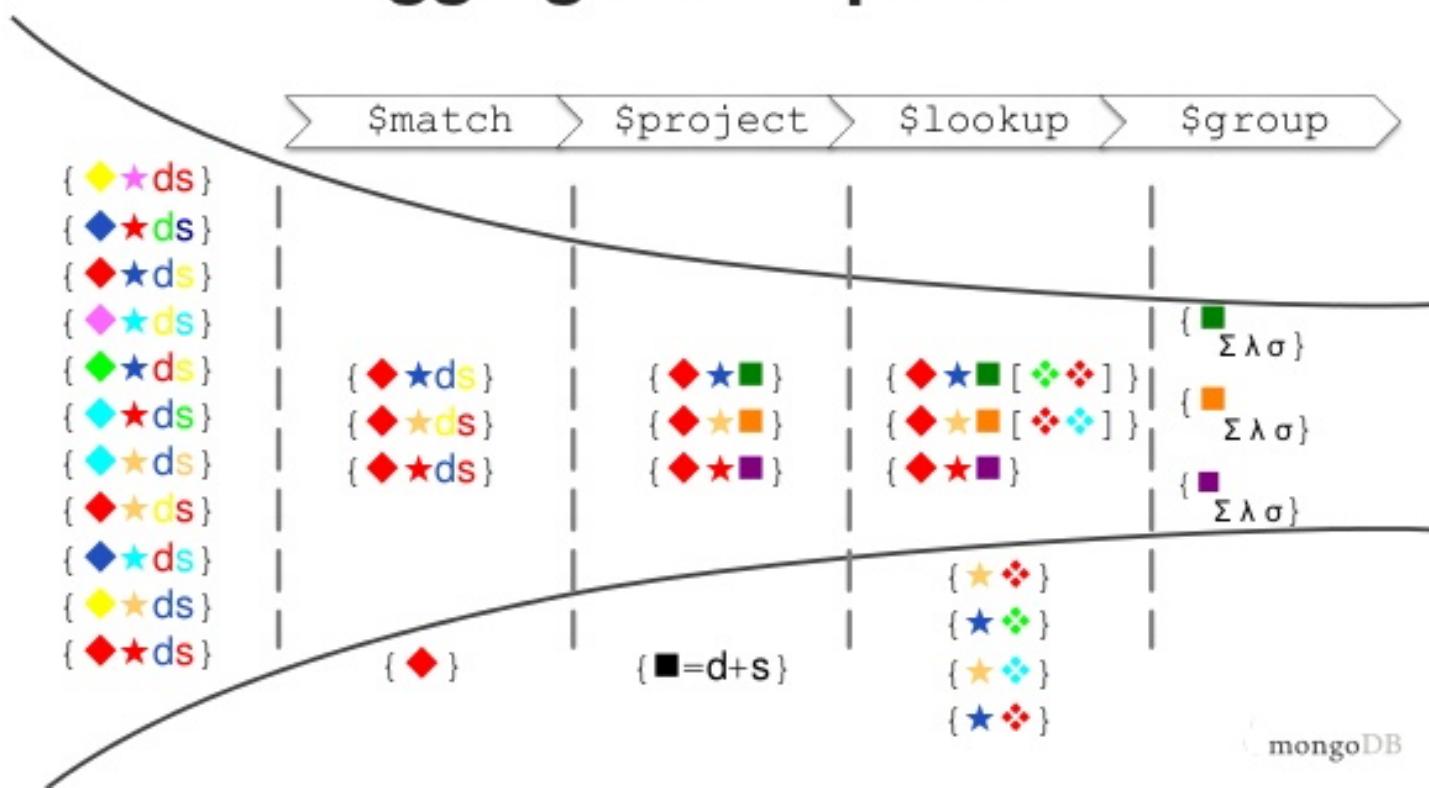
# Neo4J

```
MATCH (c:Customer {companyName:"Drachenblut Delikatessen"})  
OPTIONAL MATCH (p:Product)-[pu:PRODUCT]-(:Order)-[:PURCHASED]-(c)  
RETURN p.productName, tolnt(sum(pu.unitPrice * pu.quantity)) AS volume  
ORDER BY volume DESC;
```

# MongoDB Aggregation Pipeline

```
pipeline = [
    {"$match": {"artistName": "Suppenstar"}},
    {"$sort": {"info.releaseDate": 1})}),
    {"$group": {
        "_id": {"$year": "$info.releaseDateEpoch"},
        "count": {"$sum": "1}}},
    {"$project": {"year": "$_id.year", "count": 1}}},
]
```

# Aggregation Pipeline



## Aggregation Pipeline / SQL

- |             |                   |
|-------------|-------------------|
| — \$match   | — WHERE   HAVING  |
| — \$sort    | — ORDER BY        |
| — \$limit   | — LIMIT           |
| — \$project | — SELECT          |
| — \$group   | — GROUP BY        |
| — \$unwind  | — (JOIN)          |
| — \$lookup  | — LEFT OUTER JOIN |

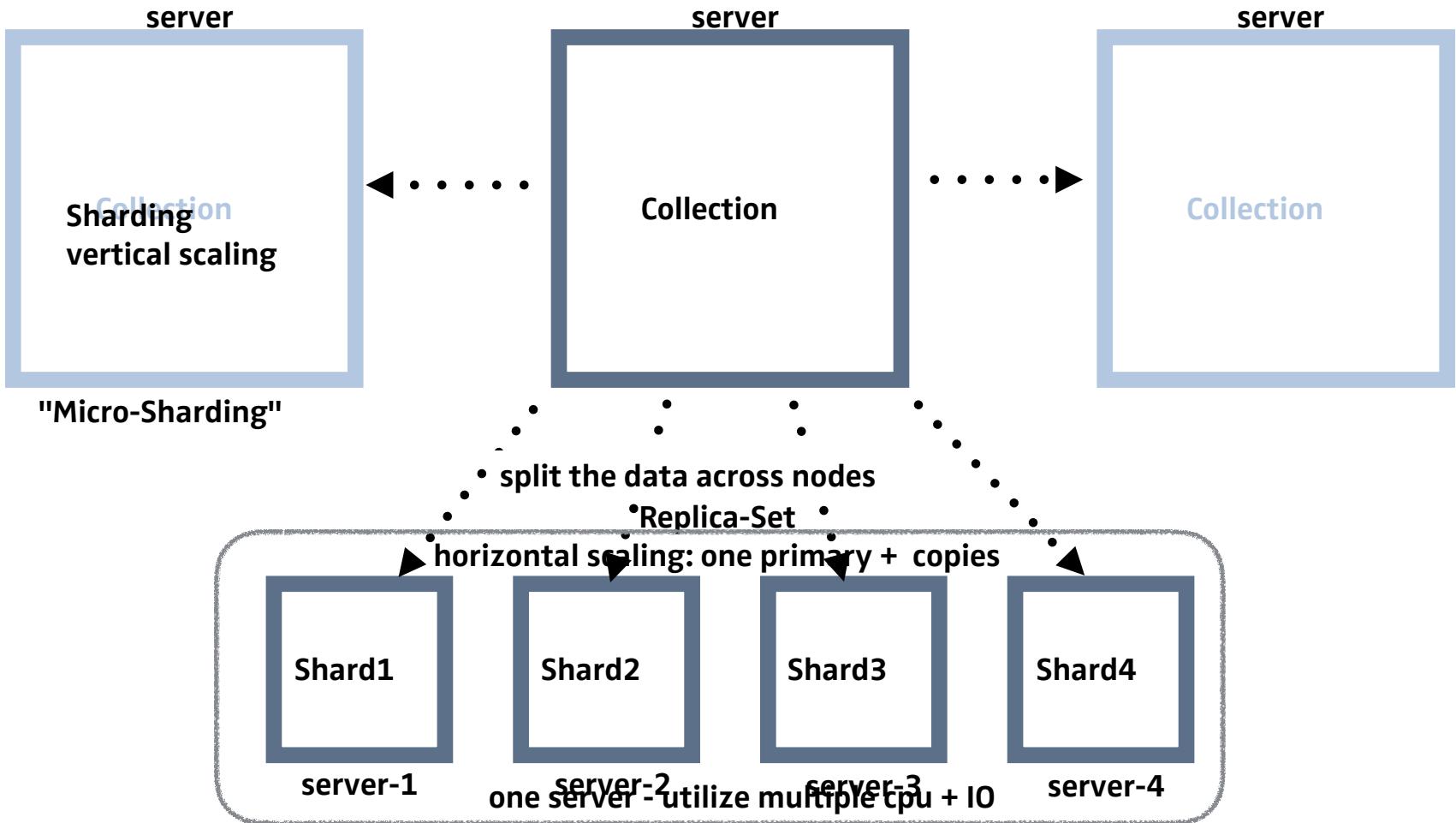
## How Hard is it to Run?

- Installation
- Maintenance
  - Cleaning up
  - Compacting
- Backup
  - Replica (or continuous)
  - File System backup
  - Dump

	Set-Up	Maintenance	Backup
PostgreSQL	easy	medium	medium
MongoDB	easy	low	<b>easy</b> (Replica / CS)
Cassandra	easy	intense	<b>easy</b> (Replica)
Neo4J	easy	medium	<b>easy</b>

## How Hard is Run Analytics without Affecting Production Performance?

	replica	shard	
PostgreSQL	medium	medium (if run on overnight backup)	
MongoDB	easy: hidden replica node	medium: hidden replica node with shard-key	
Cassandra	depends on number of nodes & partitioning	depends on number of nodes & partitioning	
Neo4J	medium	medium	



## How Hard is it to Integrate into Existing Systems?

---

	task	type	
PostgreSQL	easy	<b><i>just an additional SQL database</i></b>	
MongoDB	easy	<b>replica suggests multiple servers</b>	
Cassandra	medium	<b>requires multiple servers</b>	
Neo4J	easy	<b><i>just an additional database</i></b>	

## How Hard is it to Access / Change These Systems (Authorization)?

	User Auth	Granularity
PostgreSQL	Role-Model	Field
MongoDB	Role-Model	Collection level
Cassandra	Role-Model	Table
Neo4J	Fixed Roles	Graph

## How Hard is it to Add New Data?

---

	Known attributes	Unknown (ext.) data	
PostgreSQL	easy - medium	medium - hard	PostgreSQL also has a NoSQL feature
MongoDB	easy	easy	
Cassandra	easy	easy	
Neo4J	easy	N/A	data needs to be graph

## How Hard is it to Understand the Data Structure?

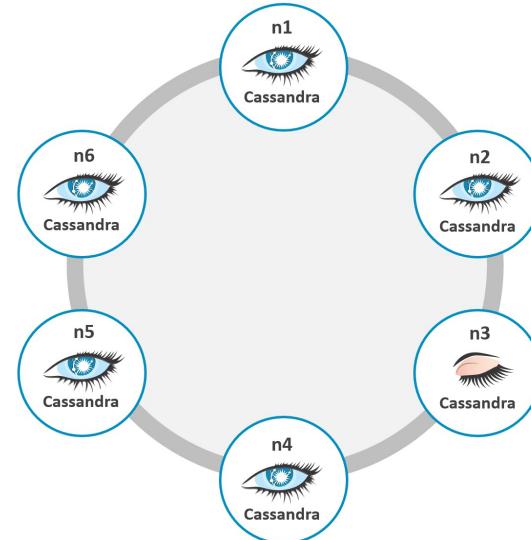
---

	small system	medium system	extensive system
PostgreSQL	easy	medium (partitioned)	hard
MongoDB	easy	easy	easy - medium
Cassandra	easy	hard (highly partitioned)	hard (highly partitioned)
Neo4J	easy	easy	easy - medium

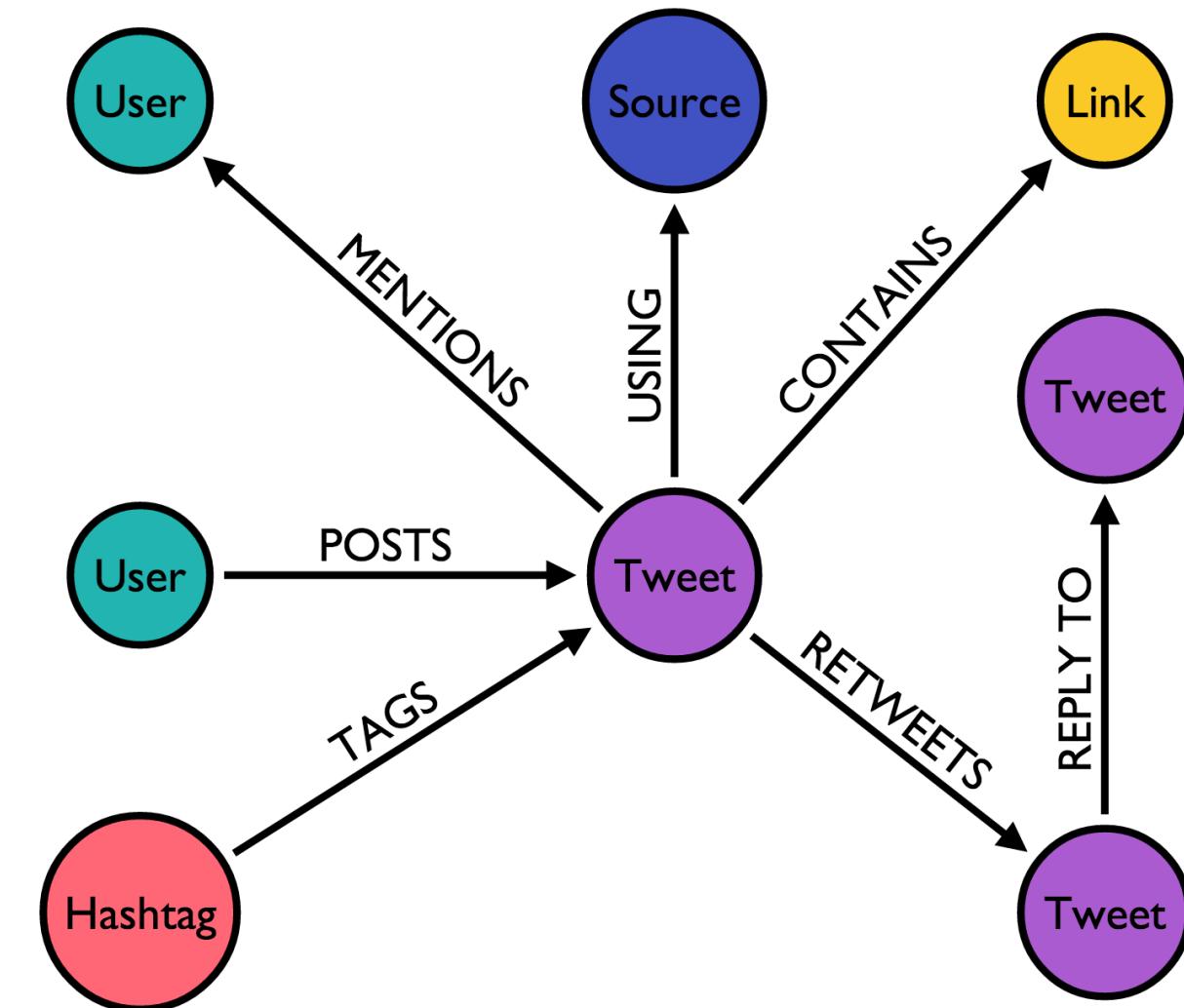
# Cassandra

RowKey: john

```
=> (column=, value=, timestamp=1374683971220000)
=> (column=map1:doug, value='555-1579', timestamp=1374683971220000)
=> (column=map1:patricia, value='555-4326', timestamp=1374683971220000)
=> (column=list1:26017c10f48711e2801fdf9895e5d0f8, value='doug', timestamp=1374683971220000)
=> (column=list1:26017c12f48711e2801fdf9895e5d0f8, value='scott', timestamp=1374683971220000)
=> (column=set1:'patricia', value=, timestamp=1374683971220000) => (column=set1:'scott', value=,
timestamp=1374683971220000)
```



# Neo4J



## Relational Model

forename	name	id
Alexander	Hendorf	1

email	type	id
ah@koenigsweg.com	1	1
alexander@private.me	2	1

type	id
work	1
private	2

phone	type	id
+49 621 12281814	1	1
+49 179 9876541	2	1

type	id
office	1
mobile	2

## Document Model

```
{  
  "_id":  
    ObjectId("507c7f79bcf86cd7994f6c0e"),  
  "forename": "Alexander",  
  "name": "Hendorf",  
  "email": [  
    {"work": "ah@koenigsweg.com"},  
    {"private": "alexander@private.me"}  
],  
  "phone": [  
    {"office": "+49 621 12281814"},  
    {"mobile": "+49 179 9876541"}  
]  

```

SCHEMA DOCUMENTS EXPLAIN PLAN INDEXES

{}

Query returned **2,708** documents. This report is based on a sample of **1,000** documents (36.93%). ⓘ

**lat**

number

A histogram showing the distribution of 'lat' values. The x-axis ranges from 36.8112 to 46.952, and the y-axis shows percentages from 0% to 16%. The distribution is skewed right, with most values between 37 and 45.

Min	Max
36.8112	46.952

**lng**

number

A histogram showing the distribution of 'lng' values. The x-axis ranges from 6.67292 to 17.07874, and the y-axis shows percentages from 0% to 10%. The distribution is relatively uniform across the range.

Min	Max
6.67292	17.07874

**loc**

coordinates

A map of Europe focusing on Italy and surrounding countries. Numerous blue dots represent the locations of tollbooths, forming a dense cluster along the Italian Peninsula and extending into parts of France, Austria, and Greece.

## Saved scripts



Get some data

Count nodes

What is related, and how

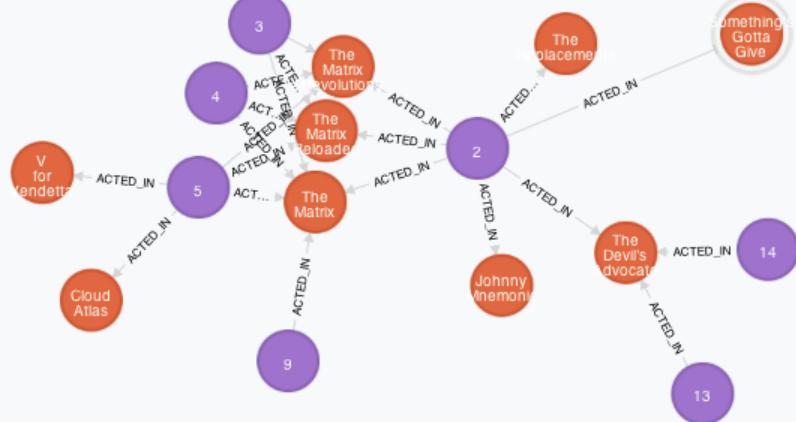
## System

Drop Cypher script file to import

```
$ MATCH (a)-[:ACTED_IN]->(b) RETURN a,b LIMIT 25
```

```
CYPHER MATCH (a)-[:ACTED_IN]->(b) RETURN a,b LIMIT 25
```

-  Default
-  Movie
-  Person



✓ Displaying 21 nodes, 25 relationships



## How Hard is it to Handle Growth?

---

	read	capacity	
PostgreSQL	medium	advanced	
MongoDB	easy	medium	
Cassandra	medium	medium	
Neo4J	medium	medium	

## Some More Use Cases for Databases in Data Science

---

- **Storing model parameters (even models)**
- **Documenting experiments**
- **Collecting performance metrics of models**
- ...

## Conclusion

- Analyse all your real needs and focus on them
- Choose an accessible, simple solution
- Do not only focus on performance
- Try, play and test before making a decision
- If you have a very specific use case, consider using Neo4j
- A good choice for general purposes
- If you work only on graphs use Neo4j
- If you have simple tables and know what you want, use MySQL



## My Advise If You Are New in the Database Space.

---

- Document store is easy to understand and maintain
- Less querying overhead for multi-dimensional data
- Aggregation pipeline
  - Grouping
  - \$relational (LO JOIN)
  - \$graphLookup
  - Many built-in operators
- Easy install and replicate
- Compressed storage by default, in-memory avail.
  
- Learning at least Basic SQL and Set Theory is a MUST
- SQLAlchemy if you work with RDBMS

## Thanks for Contributing



—Jens Dittrich, Professor [bigdata.uni-saarland.de](mailto:bigdata.uni-saarland.de) @jensdittrich

*Databases for Data Science is still an actively discussed topic in the experts' community.*

*This presentation will be constantly updated.*

*Newer findings and updates will be added.*

***Stay informed:***

Follow me on [Twitter @hendorf](#) or [LinkedIn](#)

Or drop me an email [ah@koenigsweg.com](mailto:ah@koenigsweg.com)

KÖNIGSWEG

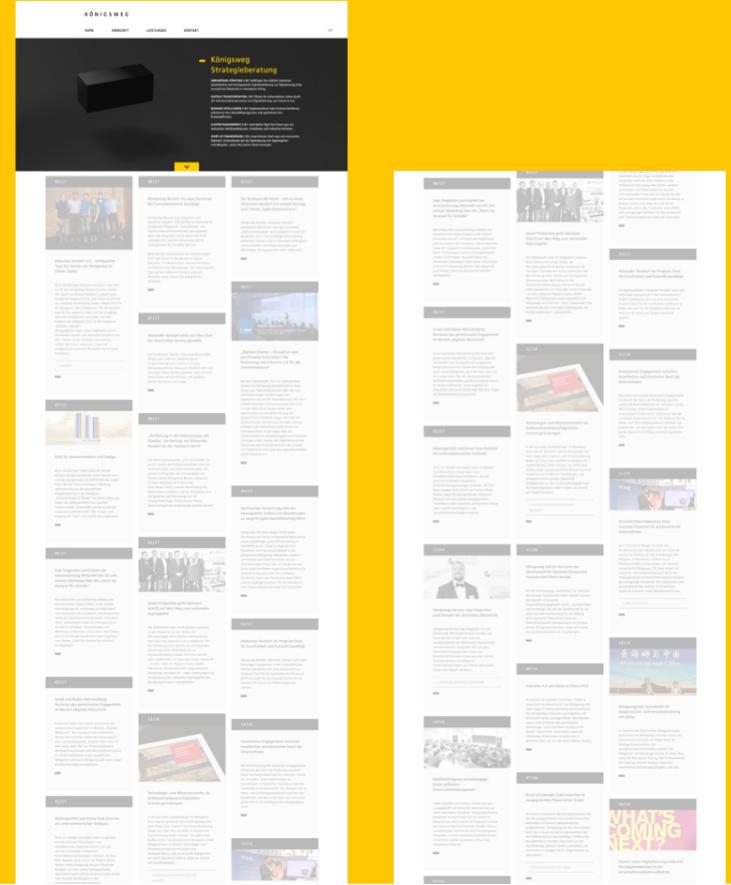
Thank you!

Q & A

[koenigsweg.com](http://koenigsweg.com)

ah@koenigsweg.com

 @hendorf



KÖNIGSWEG

KÖNIGSWEG

## Kontakt

**Königsweg GmbH**

■ Musikpark Mannheim  
Hafenstraße 49  
68159 Mannheim

■ Mafinex Technologiezentrum  
Julius-Hatry-Straße 1  
68163 Mannheim

**Telefon: +49 621 43 74 10 22**

**E-Mail: [info@koenigsweg.com](mailto:info@koenigsweg.com)**

**Telefax: +49 621 43 74 10 25**

**Web: [www.koenigsweg.com](http://www.koenigsweg.com)**

