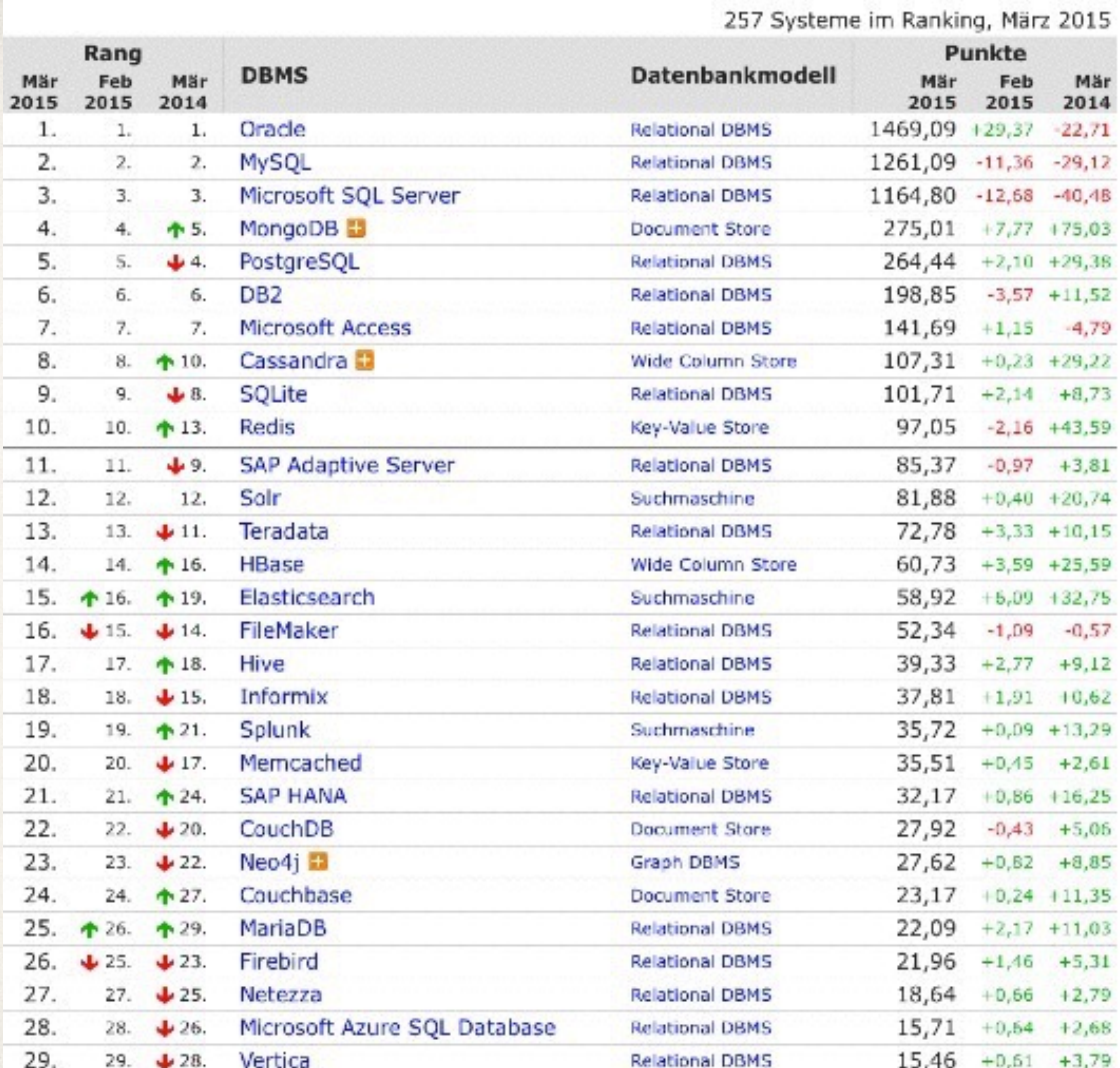
Database Implementation

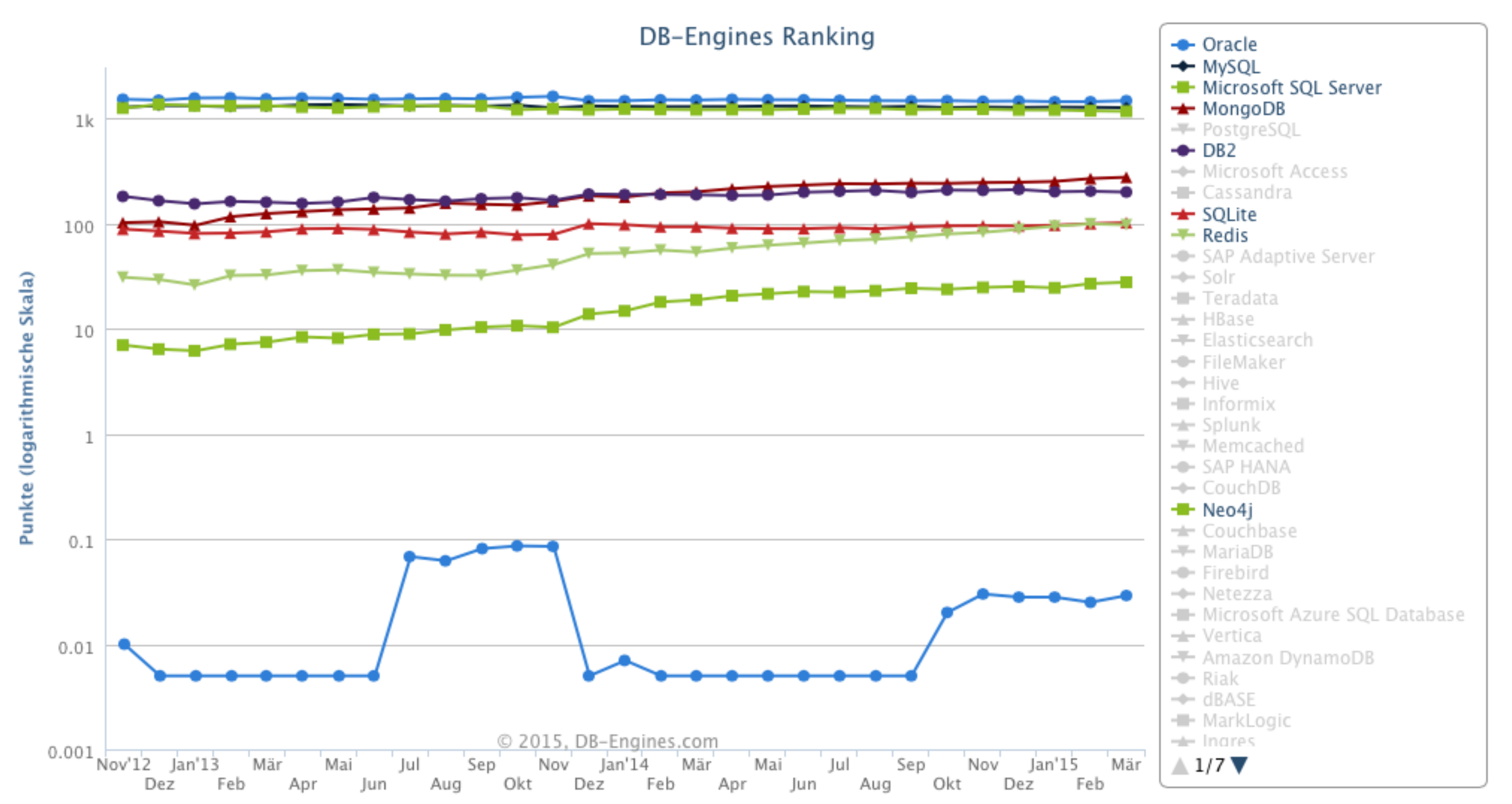
HypergraphDB

17.03.2015



257 Systeme im Ranking, März 2015

Rang			DBMS	Datenbankmodell	Punkte		
Mär 2015	Feb 2015	Mär 2014			Mär 2015	Feb 2015	Mär 2014
1.	1.	1.	Oracle	Relational DBMS	1469,09	+29,37	-22,71
2.	2.	2.	MySQL	Relational DBMS	1261,09	-11,36	-29,12
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1164,80	-12,68	-40,48
4.	4.	↑ 5.	MongoDB ⊞	Document Store	275,01	+7,77	+75,03
5.	5.	↓ 4.	PostgreSQL	Relational DBMS	264,44	+2,10	+29,38
6.	6.	6.	DB2	Relational DBMS	198,85	-3,57	+11,52
7.	7.	7.	Microsoft Access	Relational DBMS	141,69	+1,15	-4,79
8.	8.	↑ 10.	Cassandra ⊞	Wide Column Store	107,31	+0,23	+29,22
9.	9.	↓ 8.	SQLite	Relational DBMS	101,71	+2,14	+8,73
10.	10.	↑ 13.	Redis	Key-Value Store	97,05	-2,16	+43,59
11.	11.	↓ 9.	SAP Adaptive Server	Relational DBMS	85,37	-0,97	+3,81
12.	12.	12.	Solr	Suchmaschine	81,88	+0,40	+20,74
13.	13.	↓ 11.	Teradata	Relational DBMS	72,78	+3,33	+10,15
14.	14.	↑ 16.	HBase	Wide Column Store	60,73	+3,59	+25,59
15.	↑ 16.	↑ 19.	Elasticsearch	Suchmaschine	58,92	+6,09	+32,75
16.	↓ 15.	↓ 14.	FileMaker	Relational DBMS	52,34	-1,09	-0,57
17.	17.	↑ 18.	Hive	Relational DBMS	39,33	+2,77	+9,12
18.	18.	↓ 15.	Informix	Relational DBMS	37,81	+1,91	+0,62
19.	19.	↑ 21.	Splunk	Suchmaschine	35,72	+0,09	+13,29
20.	20.	↓ 17.	Memcached	Key-Value Store	35,51	+0,45	+2,61
21.	21.	↑ 24.	SAP HANA	Relational DBMS	32,17	+0,86	+16,25
22.	22.	↓ 20.	CouchDB	Document Store	27,92	-0,43	+5,06
23.	23.	↓ 22.	Neo4j ⊞	Graph DBMS	27,62	+0,82	+8,85
24.	24.	↑ 27.	Couchbase	Document Store	23,17	+0,24	+11,35
25.	↑ 26.	↑ 29.	MariaDB	Relational DBMS	22,09	+2,17	+11,03
26.	↓ 25.	↓ 23.	Firebird	Relational DBMS	21,96	+1,46	+5,31
27.	27.	↓ 25.	Netezza	Relational DBMS	18,64	+0,66	+2,79
28.	28.	↓ 26.	Microsoft Azure SQL Database	Relational DBMS	15,71	+0,64	+2,68
29.	29.	↓ 28.	Vertica	Relational DBMS	15,46	+0,61	+3,79


DB-Engines Ranking
Punkte (logarithmische Skala)
1k
100
10
1
0.1
0.01
0.001
Nov'12
Dez
Jan'13
Feb
Mär
Apr
Mai
Jun
Jul
Aug
Sep
Okt
Nov
Dez
Jan'14
Feb
Mär
Apr
Mai
Jun
Jul
Aug
Sep
Okt
Nov
Dez
Jan'15
Feb
Mär
© 2015, DB-Engines.com
Oracle
MySQL
Microsoft SQL Server
MongoDB
PostgreSQL
DB2
Microsoft Access
Cassandra
SQLite
Redis
SAP Adaptive Server
Solr
Teradata
HBase
Elasticsearch
FileMaker
Hive
Informix
Splunk
Memcached
SAP HANA
CouchDB
Neo4j
Couchbase
MariaDB
Firebird
Netezza
Microsoft Azure SQL Database
Vertica
Amazon DynamoDB
Riak
dBASE
MarkLogic
Ingres
1/7

Introduction

- General purpose & open-source
- Backed by BerkeleyDB
- Designed for knowledge management, AI and semantic web
- Can also be used as an embedded
    - Object-oriented database
    - Graph database
    - (non-SQL) relational database

Key features

- Allows edges to point to other edges and makes every node or edge carry an arbitrary value as payload. (E + N = Atom)
- Platform independent storage scheme accessible by any platform and language
- No software size limits
- Automatic mapping of POJO's
- Embedded in-process

Use cases

- Semantic web
- Bioinformatics
- Desktop application configuration storage
- Server-side Java applications
  - ➡ move to object-oriented DBs

Create DB

```
HyperGraph graph = new HyperGraph("/path/");
```

- Easy to use
- No management of other databases
- `HGEnvironment` class for more management

Storing / loading (fast)

```
graph.add(Object)

graph.get(HGHandle)
```

- No check for duplicates
- Stores any object, returns Handle for direct access
- Custom objects need to meet Java Beans convention

Querying

- Query package provides conditional expressions

```
hg.getOne(HyperGraph, HGQueryCondition)

hg.getAll(HyperGraph, HGQueryCondition)
```

- Returns list of normal Java Objects

```
hg.findAll(HyperGraph, HGQueryCondition)
```

- Returns list of handles

Querying (conditions)

- Classes for:
  - Logical expressions
  - Type matching
  - Regex string matching
  - Value matching
  - and more

Querying (conditions)

```
new And(
  new AtomTypeCondition(Book.class),
  new AtomPartCondition(
     new String[]{"author"},
     "George Bush",
     ComparisonOperator.EQ
   )
);
```

What else?

- A lot!
    - Links/relations (to make it a real hypergraph)
    - Indexing
    - Transactions
    - Caching
    - P2P framework for distributed processing

HypergraphDB Model

- atom: has value, target set, incidence set and value
  - atom with |target set| > 0: link
  - atom with |target set| = 0: node
- value: typed data
- type: atom
- Definition of hypergraph structure by atoms
- No influence on structure by values and types

HypergraphDB Model

- 2-Layer Architecture
- Primitive storage layer
  - LinkStore: ID → List < ID >
  - DataStore: ID → List < byte >
- Model layer
  - AtomID → [TypeID; ValueID; TargetID; ...; TargetID]
  - TypeID → AtomID
  - TargetID → AtomID
  - ValueID → List < ID > | List < byte >

Typing

- Types are useful:
    - constraints for DB integrity and consistency
    - define data semantics
- Types are atoms:
    - construction of new types at runtime
    - domain model part of data model
- Predefined types

Differences

- **Storage**
    - HypergraphDB:
        - Only in volatile memory (JDOs)
            - ➡ Can be serialised to disk
    - Neo4j in memory and on disk
- **Query language**
    - HypergraphDB: hgdbquery-api
    - Neo4j: API calls, REST, many more

Differences - License

- HypergraphDB:
  - LGPL (embeddable in non-GPL applications)
- Neo4j:
  - GPL, AGPL (community edition) or commercial license
    - ➡ own application has to be (A)GPL, or one needs a commercial license
    - ➡ reduced functionality compared to commercial version

Differences

- **Datatypes**
  - HypergraphDB: POJO's
  - Neo4j: (Array of) Java primitives, Strings
- **Integrity model**
  - HypergraphDB: MVCC (Multiversion concurrency control)
    - ➡ lock free, snapshots, gc
  - Neo4j: ACID, Log replication

Differences - Graph model

- HypergraphDB
    - Hypergraph with 'n-ary hyperedges'
        - Hyperedge: Connect n nodes to m nodes, n,m >=0
        - ➡ Ability to have edges from and to other edges
- Neo4j
    - Property Graph (directed, non- hypergraph)

Similarities

- Graph-oriented storage (as name suggests)
- Embeddable
- Allow transactions

Live demo

https://github.com/steilerDev/HypergraphDBProject

Conclusion

- License
- General usage very simple
- Very extensible
- Not major enough!

List of references

- HypergraphDB, Kobrix Software (2010): http://hypergraphdb.org/index

- HypergraphDB API Documentation, Kobrix Software (2012): http://www.hypergraphdb.org/docs/javadoc/index.html?org/hypergraphdb/package-summary.html

- HyperGraphDB: A Generalized Graph Database. Web-Age Information Management, pp. 25-36: http://www.hypergraphdb.org/docs/hypergraphdb.pdf

- HyperGraphDB vs. Neo4J Enterprise, Florian Heinze (2015): http://vschart.com/compare/hypergraphdb/vs/neo4j

- DB-Engines Ranking, solid IT (2015): http://db-engines.com/en/ranking

- Bretto, A. (2013). Hypergraph theory an introduction. Cham New York: Springer.