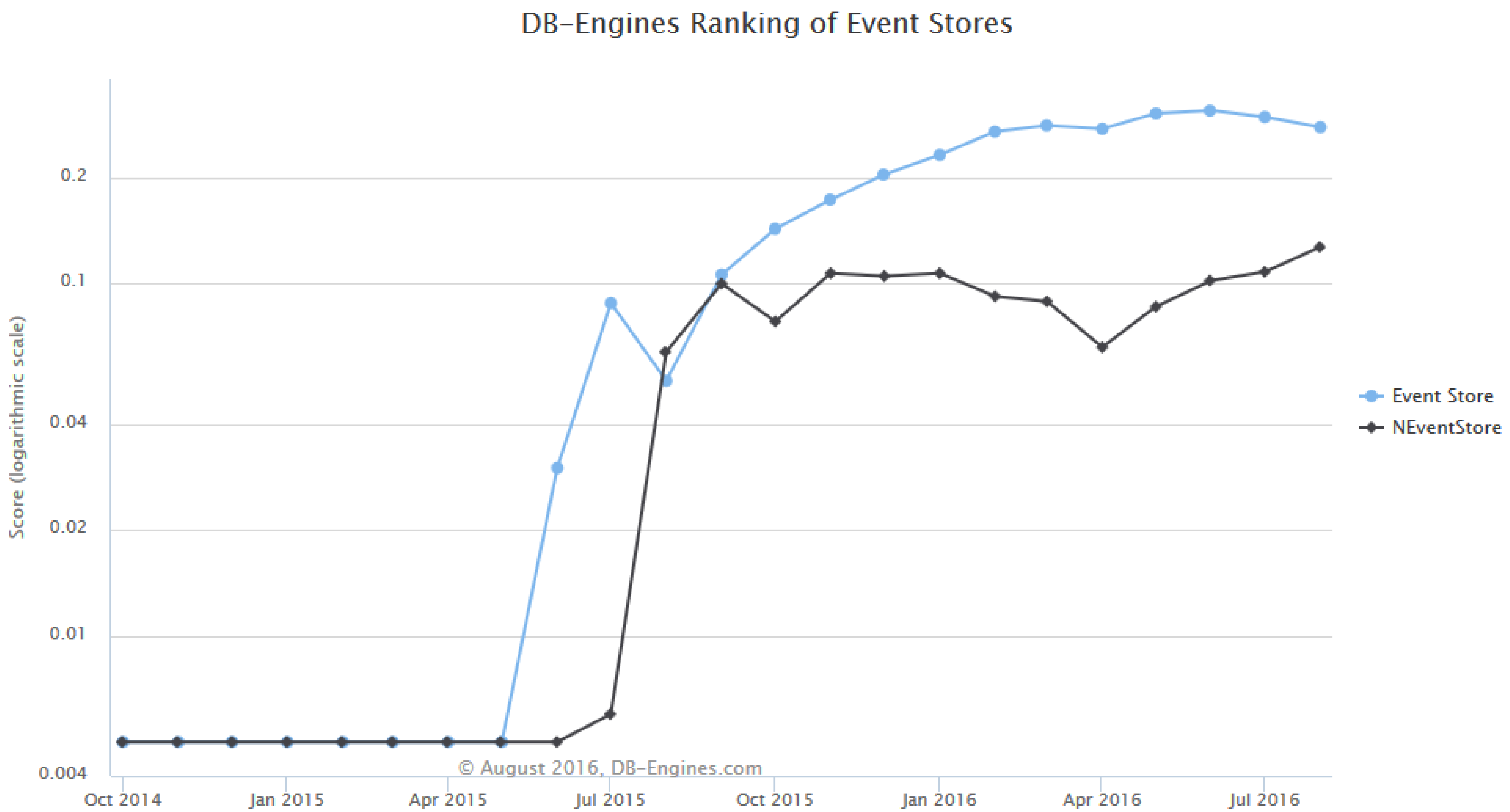
Event Stores (I)

- Event stores are database management systems implementing the concept of event sourcing. They keep all state changing events for an object together with a timestamp, thereby creating a time series for individual objects. The current state of an object can be inferred by replaying all events for that object from time 0 till the current time.
- In contrast, the other types of DBMS store the current state of an object (and loose the history, if not explicitly modeled).
- Example: For a shopping cart object each insertion of a product (product name, quantity, price) would be kept as a new event.
- Supported operations: the addition of new events, querying event time series for objects
- Not supported operations: Modifications or deletions of already kept events. Reason: simpler to maintaining consistency in distributed systems
- Performance issue: Many Event Stores realize concepts for holding snapshots (object states at a specific point in time).

[Source: DB-Engines.com, accessed on August 28, 2016]

Event Stores (II)

- 2 Systems in ranking



[Source: DB-Engines.com, accessed on August 28, 2016]

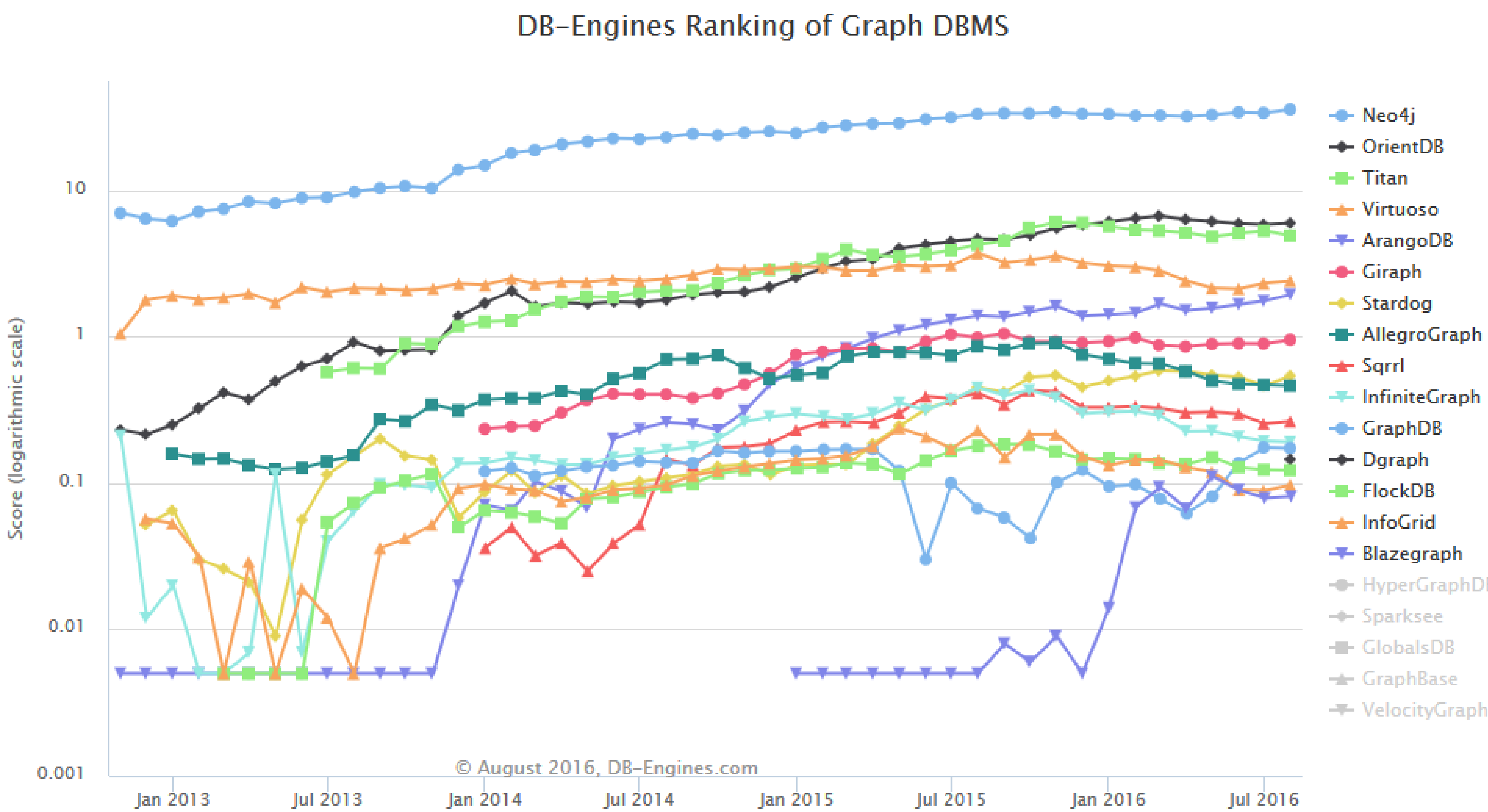
Graph DBMS (I)

- Also called graph-oriented DBMS or graph database.
- Goals
  - Represent data in graph structures as nodes and edges
  - Edges describe relationships between nodes
  - Allow easy processing of data in that form
  - Allow simple calculation of specific properties of the graph such as the number of steps needed to get from one node to another node.
- Graph DBMSs usually don't provide indexes on all nodes
- Direct access to nodes based on attribute values is not possible in these cases.
- Applications: fraud detection, trading platforms, large social networks

[Source: DB-Engines.com, accessed on August 28, 2016]

Graph DBMS (II)

❑ 21 systems in ranking, only the first 15 systems are shown graphically



[Source: DB-Engines.com, accessed on August 28, 2016]

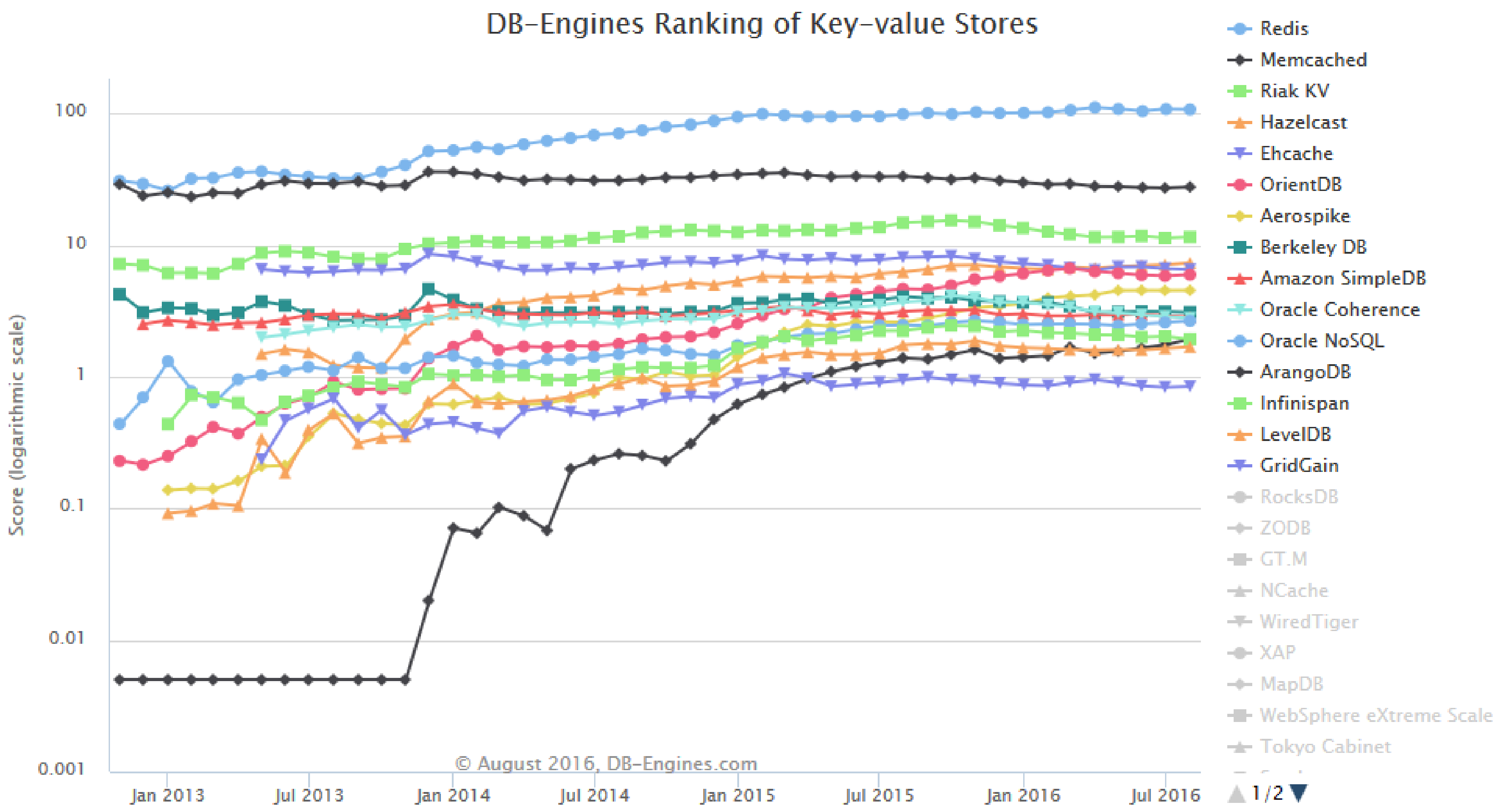
Key-value Stores (I)

- Simplest form of database management systems
- They can only store pairs of keys and values, as well as retrieve values when a key is known.
- Not adequate for complex applications
- However, their simplicity makes such systems attractive in certain situations
- Example: Resource-efficient key-value stores are often applied in embedded systems or as high performance in-process databases.
- Advanced Forms
  - An extended form of key-value stores is able to sort the keys, and thus enables range queries as well as an ordered processing of keys.
  - Many systems provide further extensions so that we see a fairly seamless transition to document stores and wide column stores.

[Source: DB-Engines.com, accessed on August 28, 2016]

Key-value Stores (II)

- 52 systems in ranking, only the first 15 systems are shown graphically



[Source: DB-Engines.com, accessed on August 28, 2016]

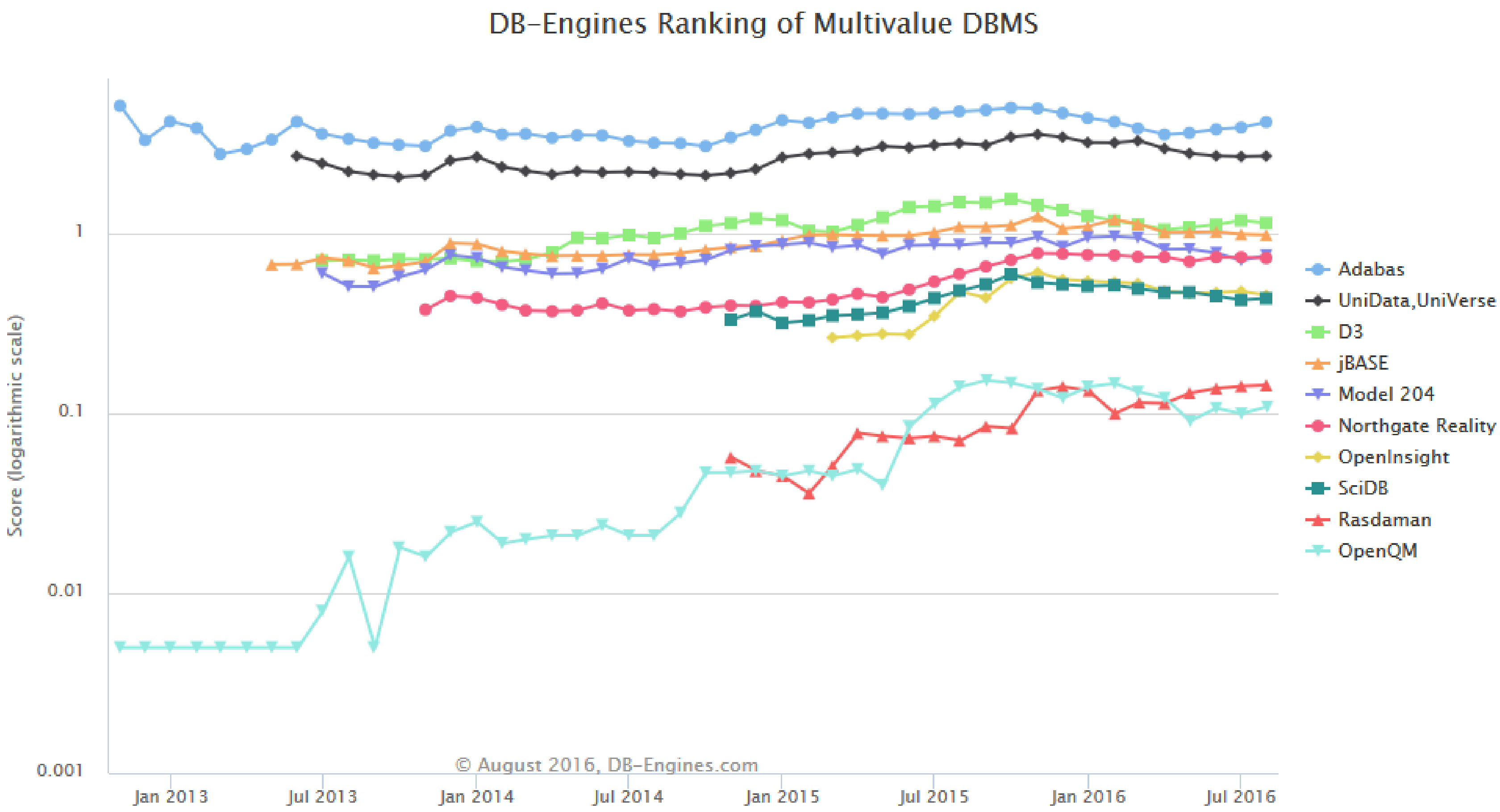
Multivalue DBMS (I)

- Store data in tables (similar to relational systems)
- However, they can assign more than one value to a record's attribute
  - This contradicts the first normal form
  - These systems are sometimes called NF2 (non-first normal form) systems
- Some relational systems have been extended to include features that allow the modeling of multivalent attributes (for example, arrays of values). In RDBMS these features should be used only in special cases.
- Multivalent attributes form the basis for data modeling in multivalue DBMSs, not least because these systems are not optimized for joins.

[Source: DB-Engines.com, accessed on August 28, 2016]

Multivalue DBMS (II)

- 10 systems in ranking



[Source: DB-Engines.com, accessed on August 28, 2016]

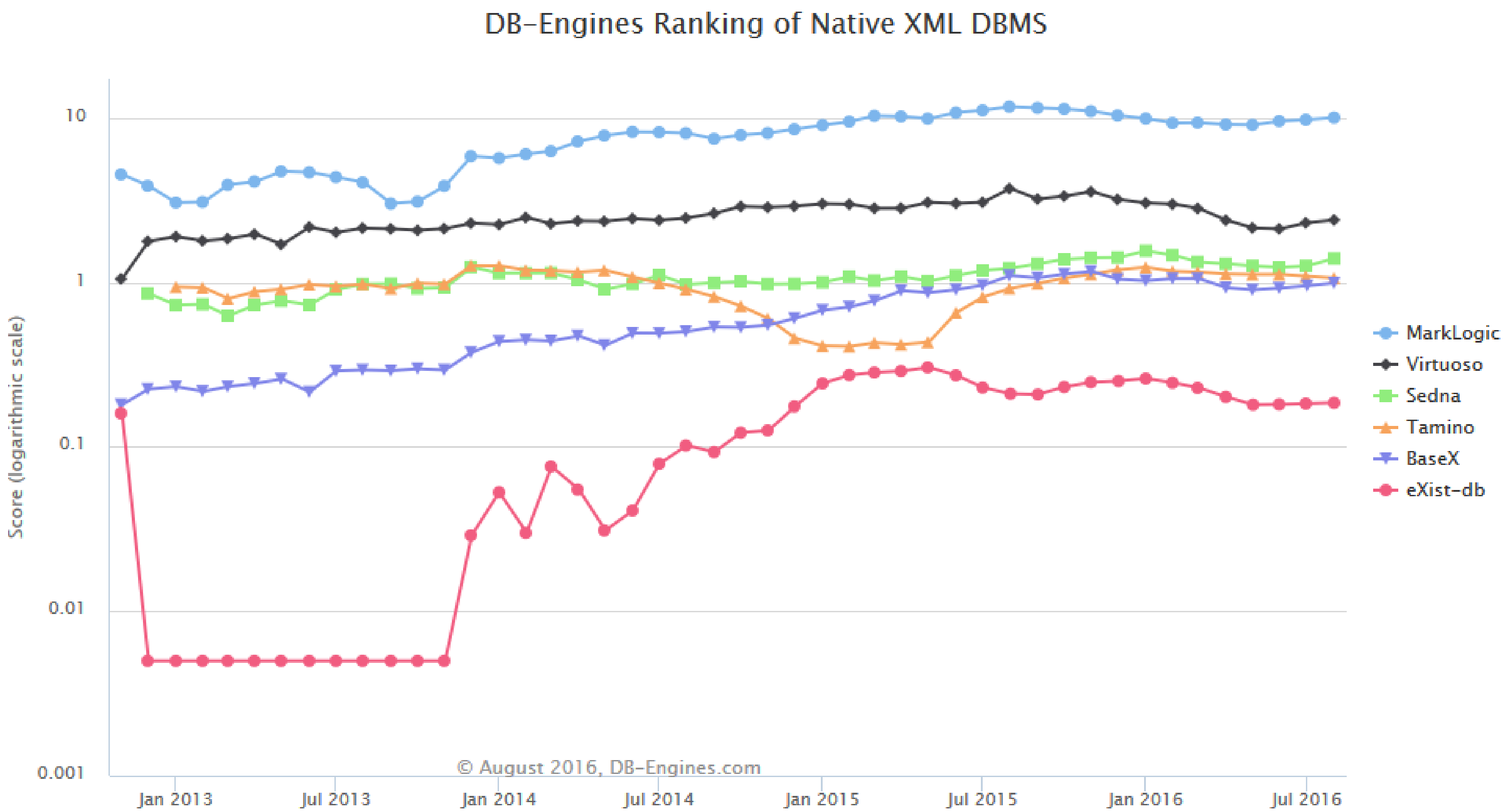
Native XML DBMS (I)

- Database systems whose internal data model corresponds to XML documents.
- In contrast to them, XML-enabled DBMS make use of a relational data model and are only capable of storing data as XML documents
- Native XML DBMS's use the full power of XML
  - They can represent hierarchical data
  - They support XML-specific query languages such as XPath, XQuery or XSLT
- Native XML DBMS do not necessarily store data as XML documents, they can use other formats for better efficiency.

[Source: DB-Engines.com, accessed on August 28, 2016]

Native XML DBMS (II)

- 7 systems in ranking



[Source: DB-Engines.com, accessed on August 28, 2016]

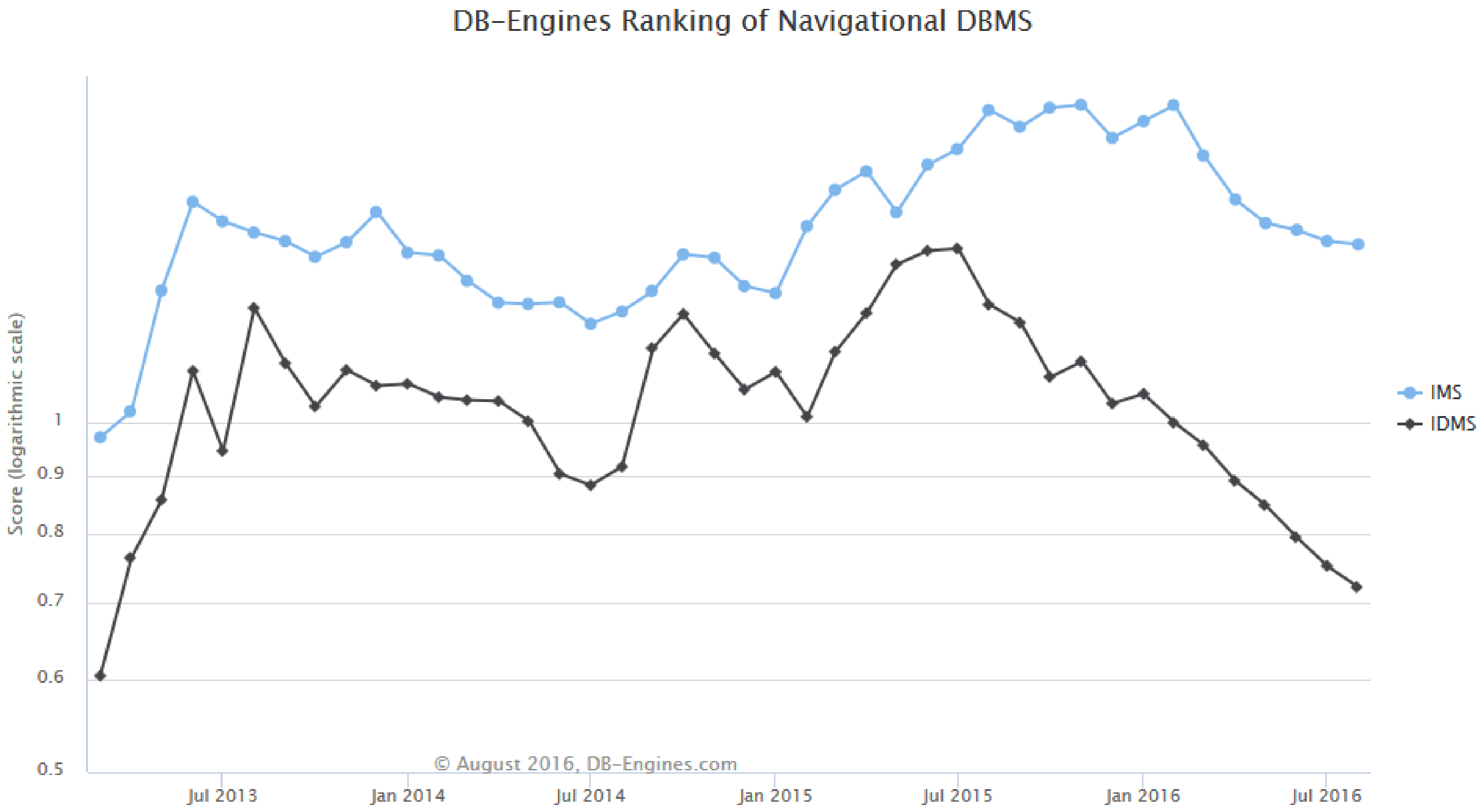
Navigational DBMS (I)

- Database systems that allow access to data sets only via linked records
- These systems were developed in the 1960s
- They were the first established systems able to manage large amounts of data
- Depending on the flexibility of linking, they are grouped into hierarchical DBMS and network DBMS
- The most popular systems are IMS by IBM, and IDMS by Computer Associates, which even today are occasionally in use

[Source: DB-Engines.com, accessed on August 28, 2016]

Navigational DBMS (II)

- ❑ 2 systems in ranking



[Source: DB-Engines.com, accessed on August 28, 2016]

Object-oriented DBMS (I)

- Also called object databases, developed in the 1980s
- Motivated by the common use of object-oriented programming languages
- Follows an object oriented data model with classes (the schema of objects), properties and methods
- Objects are managed as a unit, no joins
- Goals
  - Store objects in a database in a simple way that corresponds to their representation in a programming language
  - No conversion or decomposition should be required
  - Relationships between the objects, e.g., inheritance, should also be maintained in the database.

[Source: DB-Engines.com, accessed on August 28, 2016]

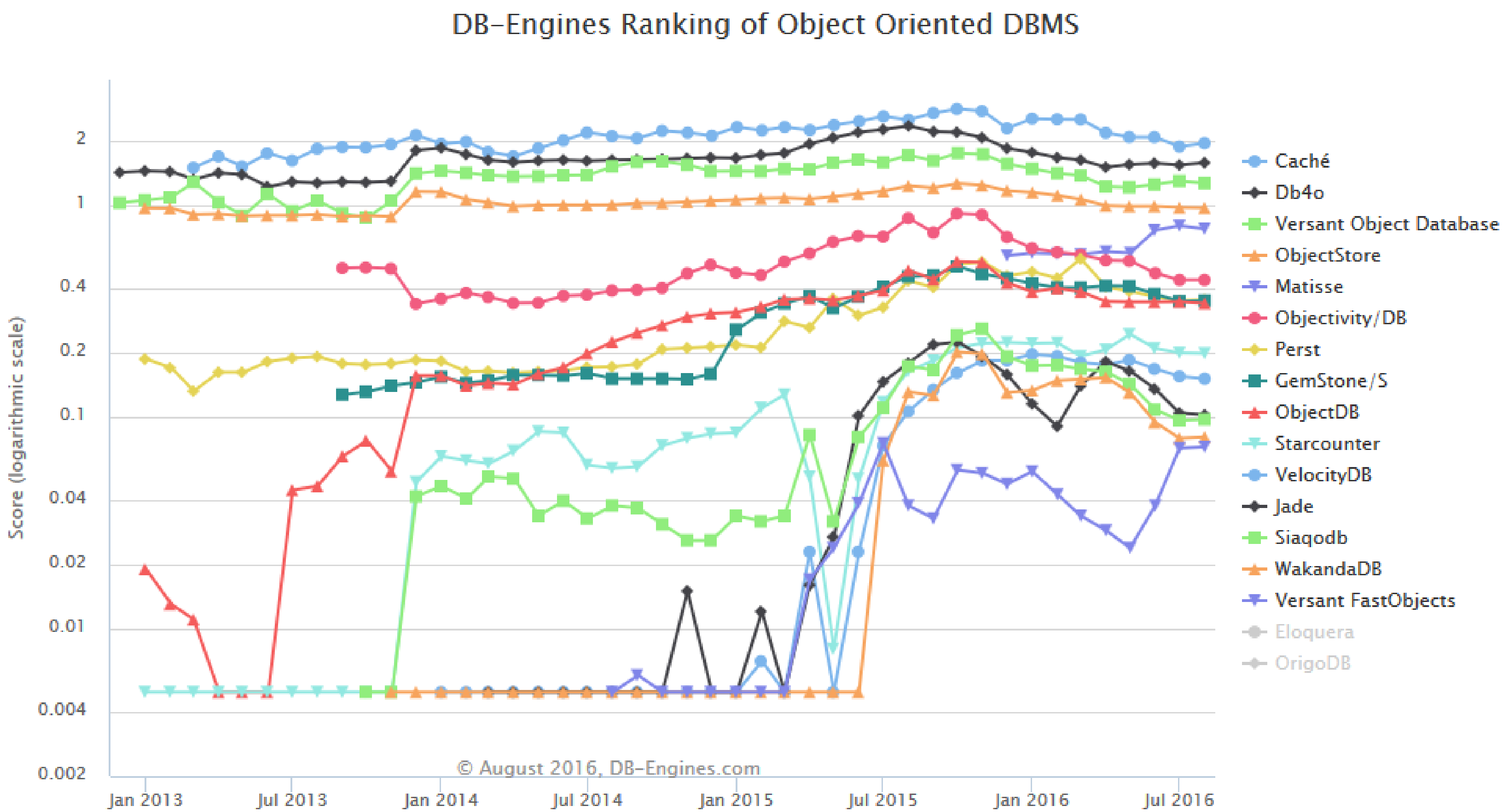
Object-oriented DBMS (II)

- Object databases often use their own SQL-like query languages for the manipulation of objects. Other non-SQL like query languages exist too.
- Object-relational DBMS: Extension of the classic relational DBMS with object oriented features, such as user-defined data types and structured attributes
- This fact and the comfortable features, tools and architectures that are now provided for the storage of objects into relational databases (such as Hibernate or JPA), hinder the widespread use of object oriented systems.

[Source: DB-Engines.com, accessed on August 28, 2016]

Object-oriented DBMS (III)

- ❑ 17 systems in ranking, only the first 15 systems are shown graphically



[Source: DB-Engines.com, accessed on August 28, 2016]

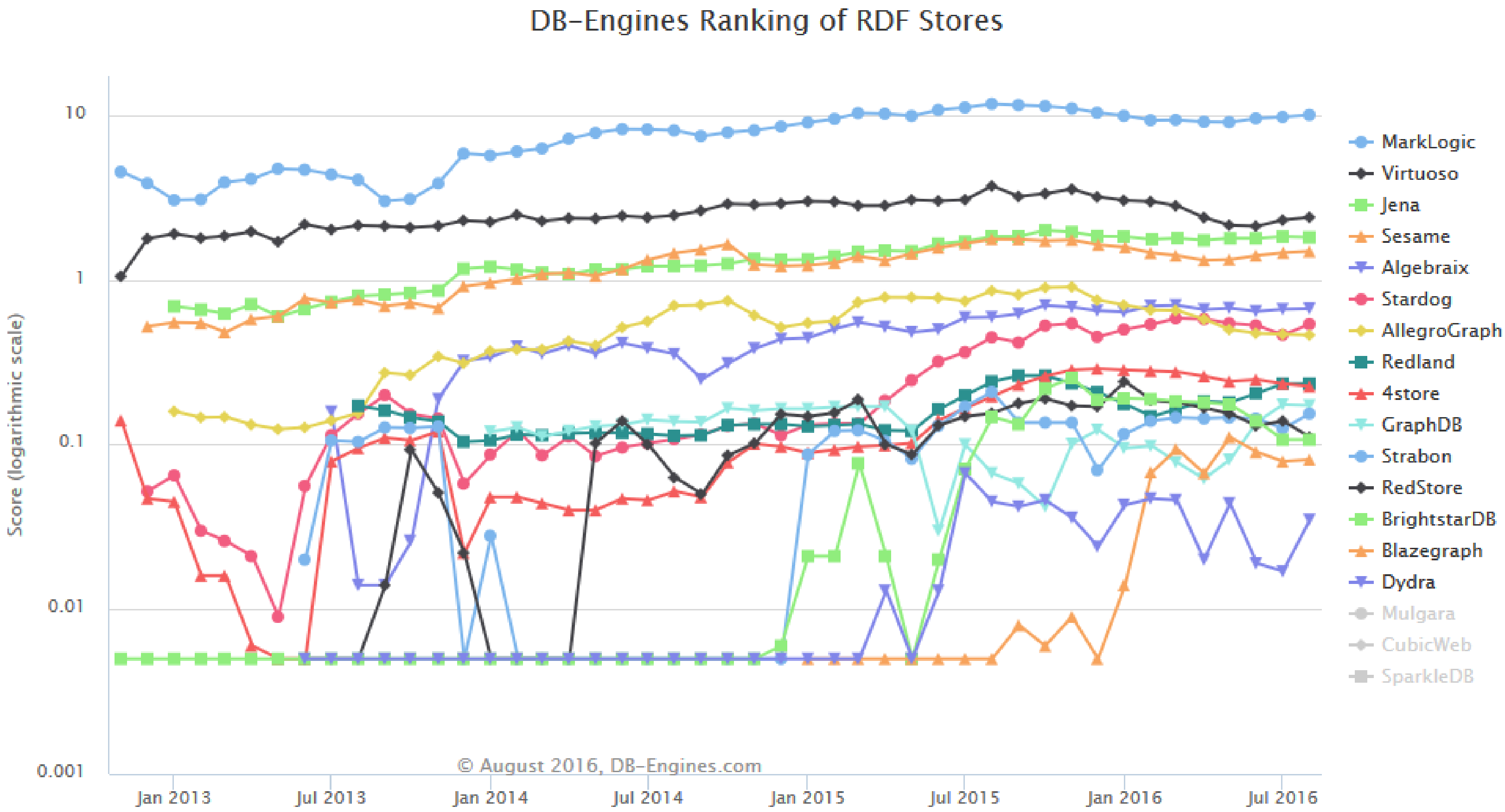
RDF Stores (I)

- The Resource Description Framework (RDF) is a methodology for the description of information
- Originally developed for describing metadata of IT resources
- Today used much more generally, often in connection with the semantic web, but also in other applications
- The RDF model represents information as triples in the form of subject-predicate-object
- Database management systems, which are able to store and process such triples, are called RDF stores or triple stores
- RDF stores can be seen as a subclass of graph DBMS by interpreting the predicate as a connection between subject and object
- However, RDF stores offer specific methods that go beyond those of the general graph DBMS. For example, SPARQL, an SQL-like query language for RDF data, is supported by most RDF stores.

[Source: DB-Engines.com, accessed on August 28, 2016]

RDF Stores (II)

- 18 systems in ranking, only the first 15 systems are shown graphically



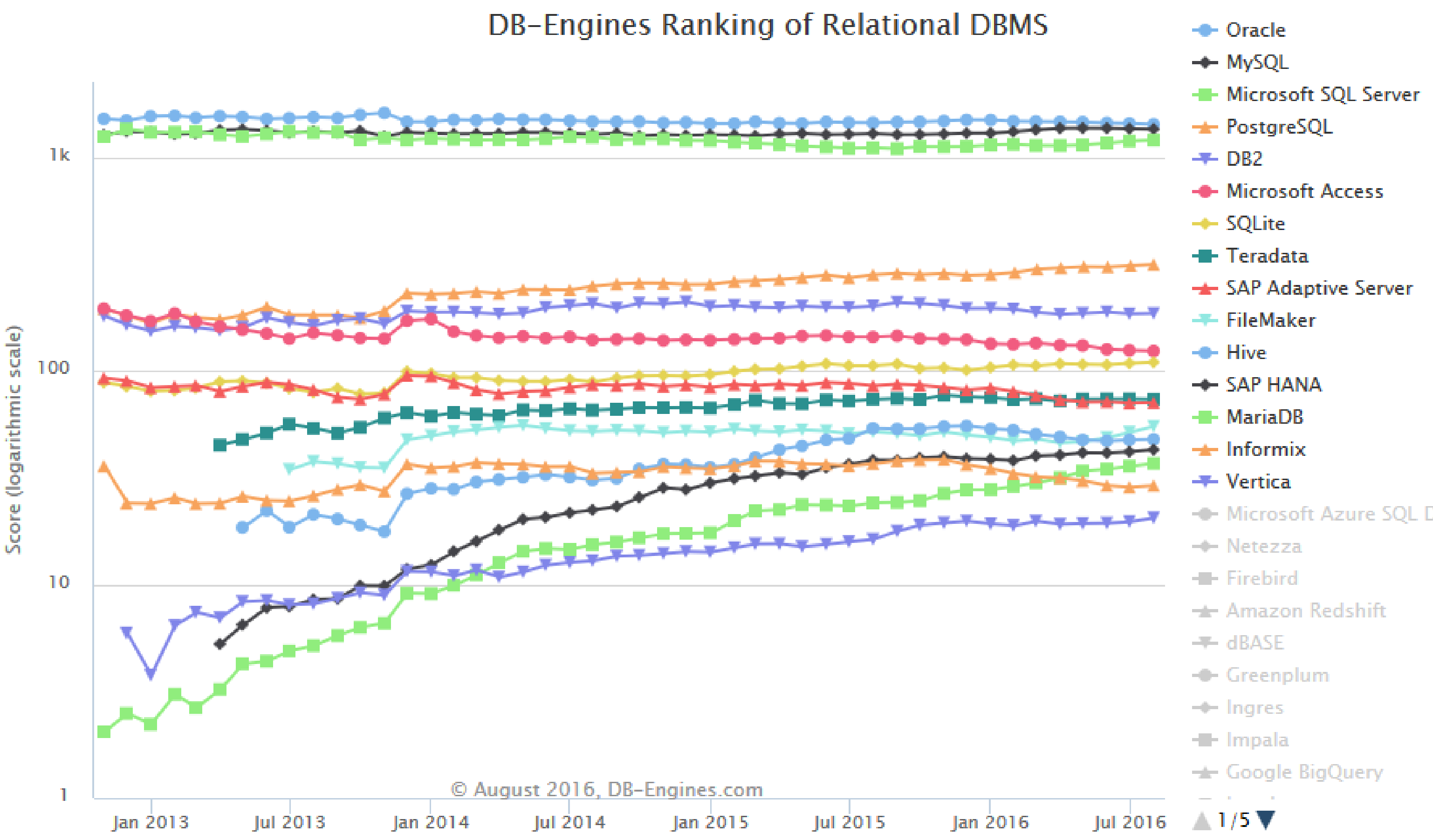
[Source: DB-Engines.com, accessed on August 28, 2016]

Relational DBMS (I)

- Known

Relational DBMS (II)

- 121 systems in ranking, only the first 15 systems are shown graphically



[Source: DB-Engines.com, accessed on August 28, 2016]

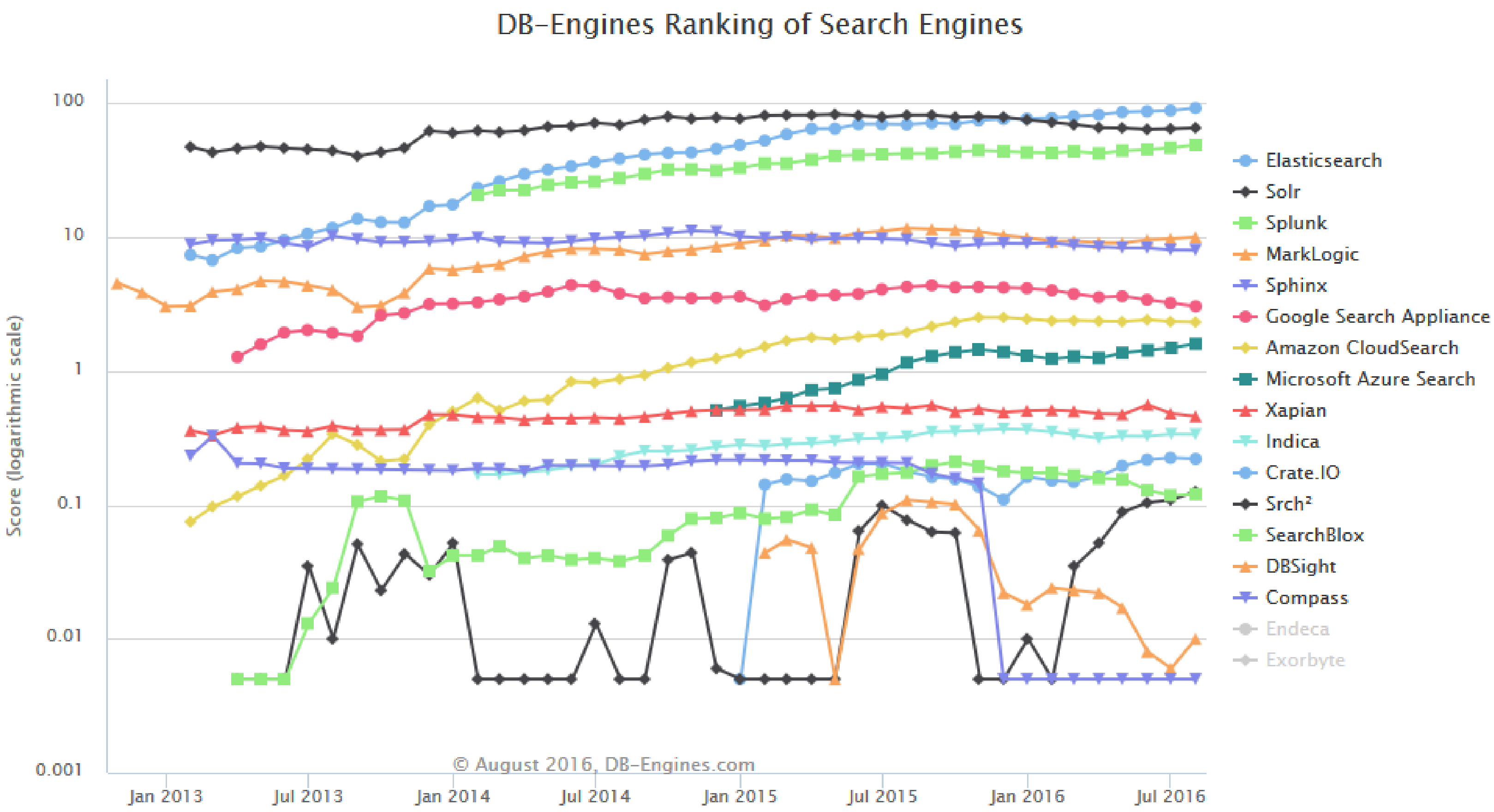
Search Engines (I)

- Search engines are NoSQL database management systems dedicated to the search for data content in an optimized manner
- Further features
  - Support for complex search expressions
  - Full text search
  - Ranking and grouping of search results
  - Geospatial search
  - Distributed search for high scalability

[Source: DB-Engines.com, accessed on August 28, 2016]

Search Engines (II)

- ❑ 17 systems in ranking, only the first 15 systems are shown graphically



[Source: DB-Engines.com, accessed on August 28, 2016]

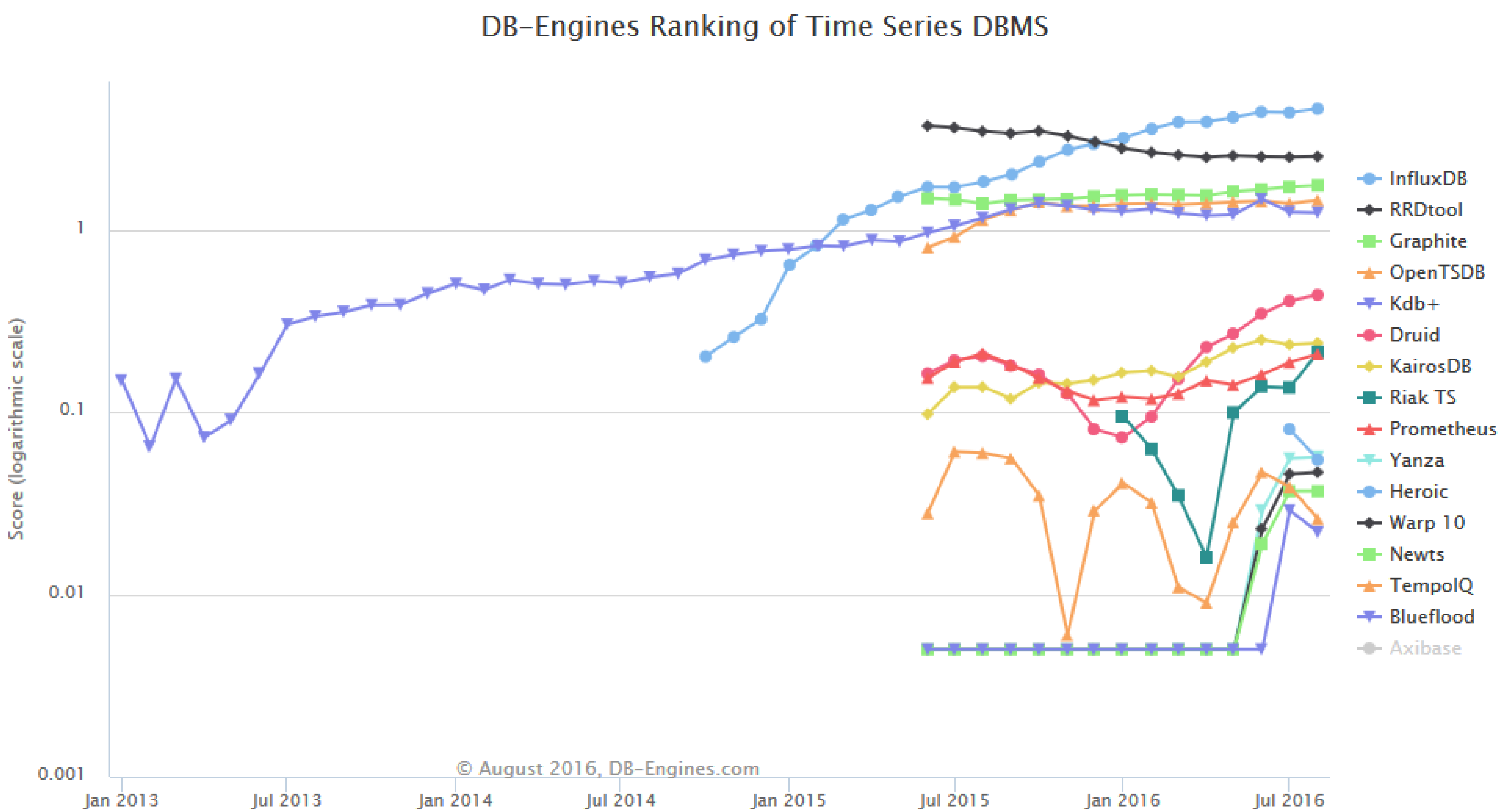
Time Series DBMS (I)

- Optimized for handling time series data: each entry is associated with a timestamp.
- Examples: Time series data may be produced by sensors, smart meters, or RFIDs
- Designed to efficiently collect, store, and query various time series with high transaction volumes
- Time series data can also be managed with other categories of DBMS (from key-value stores to relational systems) but Time Series DBMS are specialized systems that can cope with specific challenges.

[Source: DB-Engines.com, accessed on August 28, 2016]

Time Series DBMS (II)

- ❑ 16 systems in ranking, only the first 15 systems are shown graphically



[Source: DB-Engines.com, accessed on August 28, 2016]

Wide Column Stores (I)

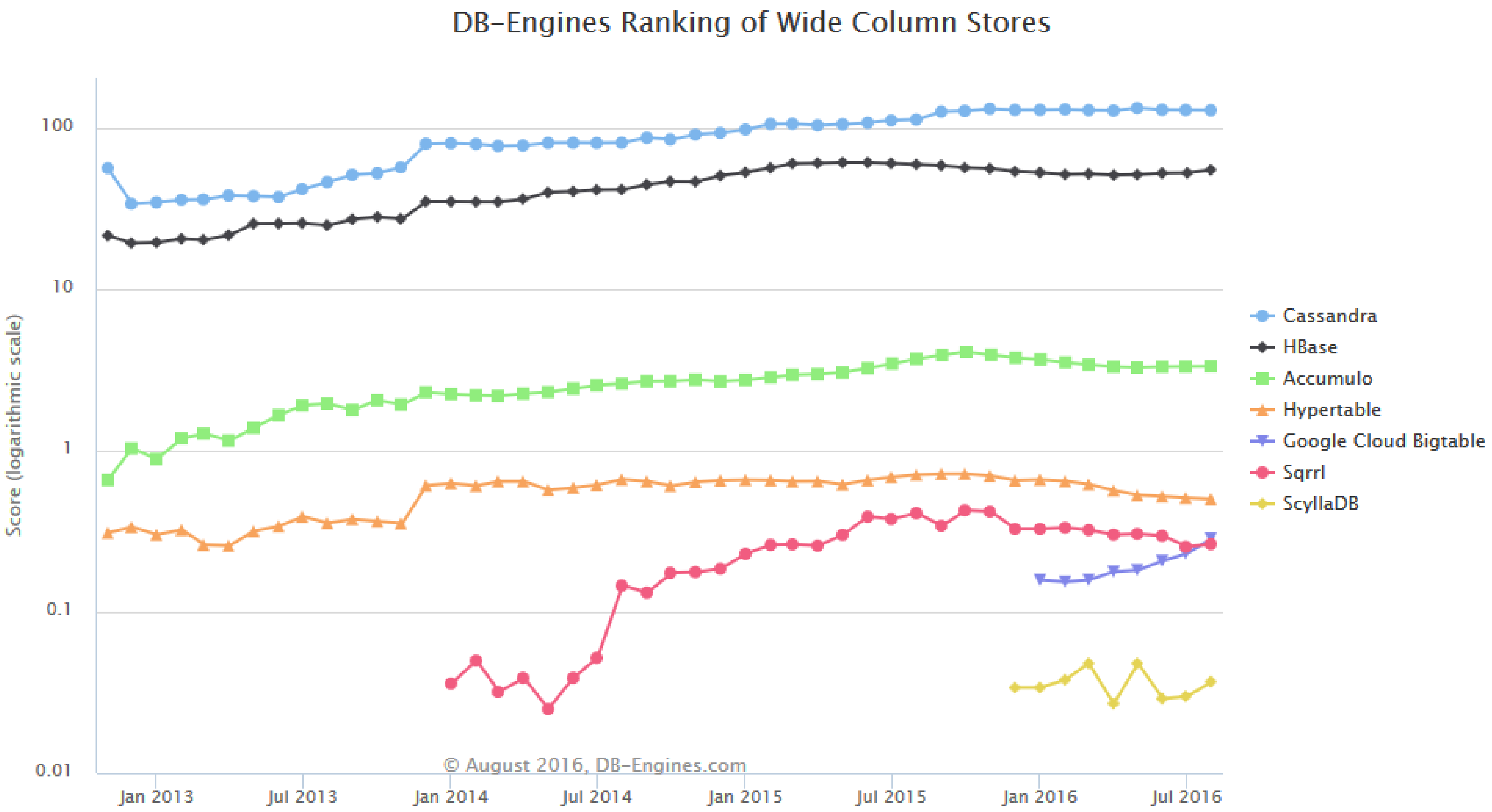
- Also called extensible record stores
- They store data in records with an ability to hold very large numbers of dynamic columns
  - Column names as well as the record keys are not fixed
  - A record can have billions of columns

  → Wide column stores can be seen as two-dimensional key-value stores
- Wide column stores share the characteristic of being schema-free with document stores but the implementation is very different.

[Source: DB-Engines.com, accessed on August 28, 2016]

Wide Column Stores (II)

- 7 systems in ranking



[Source: DB-Engines.com, accessed on August 28, 2016]

Complete Ranking (I)

- 309 systems in ranking, only the first 15 systems are shown

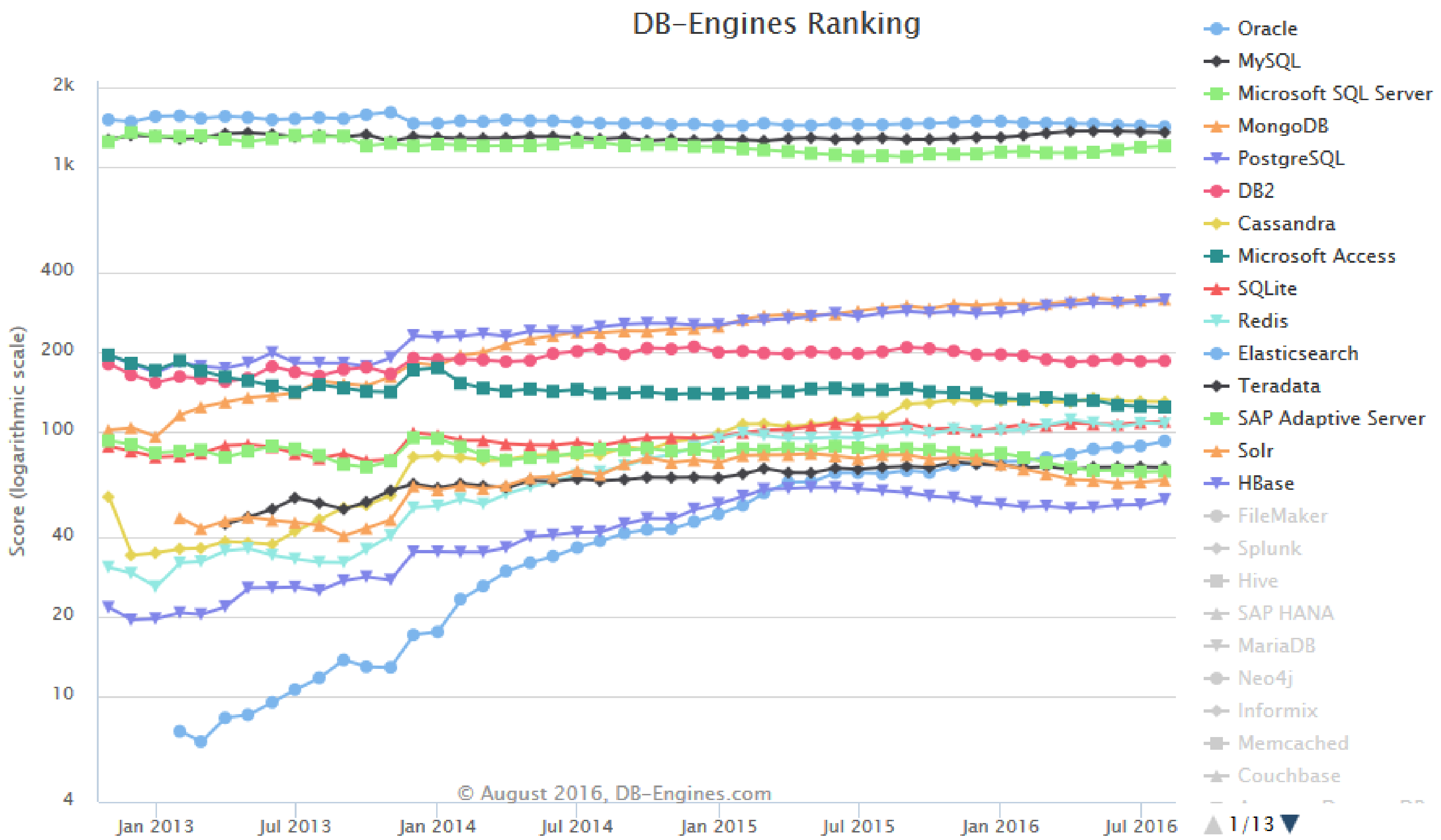
309 systems in ranking, August 2016

Rank Aug 2016	Rank Jul 2016	Rank Aug 2015	DBMS	Database Model	Score Aug 2016	Score Jul 2016	Score Aug 2015
1.	1.	1.	Oracle	Relational DBMS	1427.72	-13.81	-25.30
2.	2.	2.	MySQL	Relational DBMS	1357.03	-6.25	+65.00
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1205.04	+12.16	+96.39
4.	4.	4.	MongoDB	Document store	318.49	+3.49	+23.84
5.	5.	5.	PostgreSQL	Relational DBMS	315.25	+4.10	+33.39
6.	6.	6.	DB2	Relational DBMS	185.89	+0.81	-15.35
7.	7.	↑ 8.	Cassandra	Wide column store	130.24	-0.47	+16.24
8.	8.	↓ 7.	Microsoft Access	Relational DBMS	124.05	-0.85	-20.15
9.	9.	9.	SQLite	Relational DBMS	109.86	+1.32	+4.04
10.	10.	10.	Redis	Key-value store	107.32	-0.71	+8.51
11.	11.	↑ 14.	Elasticsearch	Search engine	92.49	+3.87	+22.85
12.	12.	↑ 13.	Teradata	Relational DBMS	73.64	-0.29	+0.05
13.	13.	↓ 11.	SAP Adaptive Server	Relational DBMS	71.04	+0.31	-14.07
14.	14.	↓ 12.	Solr	Search engine	65.77	+1.08	-16.13
15.	15.	15.	HBase	Wide column store	55.51	+2.37	-4.43

[Source: DB-Engines.com, accessed on August 28, 2016]

Complete Ranking (II)

- ❑ 309 systems in ranking, only the first 15 systems are shown graphically



[Source: DB-Engines.com, accessed on August 28, 2016]