BIGDATA STACK

Core NoSQL Systems: [Mostly originated out of a Web 2.0 need]

## Wide Column Store / Column Families

### [Hadoop / HBase](http://hadoop.apache.org/)

API: **Java / any writer**, Protocol: **any write call**, Query Method: **MapReduce Java / any exec**, Replication: **HDFS Replication**, Written in: **Java**, Concurrency: ?, Misc: **Links**: 3 Books [[1](http://www.amazon.com/Hadoop-Action-Chuck-Lam/dp/1935182196/), [2](http://www.amazon.com/Pro-Hadoop-Jason-Venner/dp/1430219424/), [3](http://www.amazon.com/Hadoop-Definitive-Guide-Tom-White/dp/0596521979/)]

### [Cassandra](https://cassandra.apache.org/)

massively scalable, partitioned row store, masterless architecture, linear scale performance, no single points of failure, read/write support across multiple data centers & cloud availability zones. API / Query Method: **CQL and Thrift**, replication: **peer-to-peer**, written in: **Java**, Concurrency: **tunable consistency**, Misc: built-in data compression, MapReduce support, primary/secondary indexes, security features. Links: [Documentation](http://www.datastax.com/docs), [PlanetC\*](http://planetcassandra.org/), [Company](http://www.datastax.com/).

### [Hypertable](http://hypertable.org/)

API: **Thrift** (Java, PHP, Perl, Python, Ruby, etc.), Protocol: **Thrift**, Query Method: **HQL, native Thrift API**, Replication: **HDFS Replication**, Concurrency: **MVCC**, Consistency Model: **Fully consistent** Misc: High performance C++ implementation of Google's Bigtable. [» Commercial support](http://www.hypertable.com/)

### [Accumulo](http://accumulo.apache.org/)

Accumulo is based on **BigTable** and is built on top of [**Hadoop**](http://hadoop.apache.org/), [**Zookeeper**](http://zookeeper.apache.org/), and [**Thrift**](http://thrift.apache.org/). It features improvements on the BigTable design in the form of **cell-based access control**, improved**compression**, and a server-side programming mechanism that can modify key/value pairs at various points in the data management process.

### [Amazon SimpleDB](http://aws.amazon.com/simpledb/)

**Misc**: not open source / part of AWS, [Book](http://www.apress.com/book/view/1430225335) (will be outperformed by DynamoDB ?!)

### [Cloudata](http://www.cloudata.org/)

Google's Big table clone like HBase. [» Article](http://www.readwriteweb.com/hack/2011/02/open-source-bigtable-cloudata.php)

### [Cloudera](http://www.cloudera.com/)

Professional Software & Services based on Hadoop.

### [HPCC](http://www.hpccsystems.com/)

from [LexisNexis](http://www.lexisnexis.com/), [info](http://www.lexisnexis.com/government/solutions/literature/hpcc-das.pdf), [article](http://wikibon.org/blog/lexisnexis-hpcc-takes-on-hadoop-as-battle-for-big-data-supremacy-heats-up/)

### [Stratosphere](http://stratosphere.eu/)

massively parallel & flexible data analytics platform, API: **Java, Scala**, Query Method:**expressive data flows (extended M/R, rich UDFs, iteration support)**, Data Store: **independent** (e.g., HDFS, S3, MongoDB), Written in: Java, License: Apache License V2.0, Misc: **good integration with Hadoop stack**(HDFS, YARN), source code on [Github](http://github.com/stratosphere/stratosphere)[OpenNeptune, Qbase, KDI]

## Document Store

### [MongoDB](http://www.mongodb.org/)

API: **BSON**, Protocol: C, Query Method: **dynamic object-based language & MapReduce**, Replication: **Master Slave & Auto-Sharding**, Written in: **C++**,Concurrency: **Update in Place**. Misc: **Indexing, GridFS, Freeware + Commercial License** Links: [» Talk](http://www.leadit.us/hands-on-tech/MongoDB-High-Performance-SQL-Free-Database), [» Notes](http://www.paperplanes.de/2010/2/25/notes_on_mongodb.html), [» Company](http://www.10gen.com/)

### [Elasticsearch](http://www.elasticsearch.org/)

API: **REST and many languages**, Protocol: **REST**, Query Method: **via JSON**, Replication + Sharding: **automatic and configurable**, written in: **Java**, Misc: schema mapping, multi tenancy with arbitrary indexes, [» Company and Support](http://www.elasticsearch.com/), [» Article](https://www.found.no/foundation/elasticsearch-as-nosql/)

### [Couchbase Server](http://www.couchbase.com/)

**API: Memcached API+protocol** (binary and ASCII) , **most languages**, Protocol:**Memcached REST interface for cluster conf + management**, Written in: **C/C++** + **Erlang** (clustering), Replication: **Peer to Peer, fully consistent**, Misc: **Transparent topology changes during operation, provides memcached-compatible caching buckets, commercially supported version available**, Links: [» Wiki](http://wiki.membase.org/), [» Article](http://www.infoq.com/news/2010/10/membase)

### [CouchDB](http://couchdb.apache.org/)

API: **JSON**, Protocol: **REST**, Query Method: **MapReduceR of JavaScript Funcs**, Replication:**Master Master**, Written in: **Erlang**, Concurrency: **MVCC**, **Misc**:

**Links**: [» 3 CouchDB books](http://couchdb.apache.org/docs/books.html) , [» Couch Lounge](http://tilgovi.github.com/couchdb-lounge/) (partitioning / clusering), [» Dr. Dobbs](http://www.drdobbs.com/java/223100116)

### [RethinkDB](http://www.rethinkdb.com/)

API: **protobuf-based**, Query Method: **unified chainable query language (incl. JOINs, sub-queries, MapReduce, GroupedMapReduce)**; Replication: **Sync and Async Master Slave with per-table acknowledgements**, Sharding: **guided range-based**, Written in: **C++**, Concurrency: **MVCC**. Misc: log-structured storage engine with concurrent incremental garbage compactor

### [RavenDB](http://github.com/ravendb/ravendb)

.Net solution. Provides **HTTP/JSON** access. **LINQ** queries & **Sharding** supported. [» Misc](http://www.codeproject.com/KB/cs/RavenDBIntro.aspx)

### [MarkLogic Server](http://www.marklogic.com/)

(freeware+commercial) API: **JSON, XML, Java** Protocols: **HTTP, REST**Query Method:**Full Text Search, XPath, XQuery, Range, Geospatial** Written in: **C++** Concurrency: **Shared-nothing cluster, MVCC** Misc: Petabyte-scalable, cloudable, ACID transactions, auto-sharding, failover, master slave replication, secure with ACLs. Developer Community [»](http://developer.marklogic.com/)

### [Clusterpoint Server](http://www.clusterpoint.com/)

(freeware+commercial) API: **XML, PHP, Java, .NET** Protocols: **HTTP, REST, native TCP/IP** Query Method: **full text search, XML, range and Xpath queries**; Written in **C++** Concurrency:**ACID-compliant, transactional, multi-master cluster** Misc: Petabyte-scalable document store and full text search engine. Information ranking. Replication. Cloudable.

### [NeDB](https://github.com/louischatriot/nedb)

NoSQL database for Node.js in pure **javascript**. It implements the most commonly used subset of**MongoDB's API** and is quite **fast** (about 25,000 reads/s on a 10,000 documents collection with indexing).

### [Terrastore](http://code.google.com/p/terrastore/)

API: **Java & http**, Protocol: **http**, Language: **Java**, Querying: **Range queries, Predicates**, Replication: **Partitioned with consistent hashing**, Consistency: **Per-record strict consistency**, Misc: Based on Terracotta

### [JasDB](http://www.oberasoftware.com/)

Lightweight open source document database written in Java for high performance, runs in-memory, supports Android. API: **JSON, Java** Query Method: **REST OData Style Query language, Java fluent Query API** Concurrency: **Atomic document writes** Indexes: **eventually consistent indexes**

### [RaptorDB](http://www.codeproject.com/Articles/375413/RaptorDB-the-Document-Store)

JSON based, Document store database with compiled **.net map functions** and automatic hybrid bitmap indexing and **LINQ query filters**

### [djondb](http://djondb.com/)

djonDB API: **BSON**, Protocol: **C++**, Query Method: **dynamic queries and map/reduce**, Drivers:**Java, C++, PHP** Misc: ACID compliant, Full shell console over google v8 engine, djondb requirements are submited by users, not market. License: GPL and commercial

### [EJDB](http://ejdb.org/)

Embedded JSON database engine based on tokyocabinet. API: **C/C++, C# (.Net, Mono), Lua, Ruby, Python, Node.js binding**, Protocol: **Native**, Written in: **C**, Query language: **mongodb-like dynamic queries**, Concurrency: **RW locking, transactional** , Misc: **Indexing, collection level rw locking, collection level transactions, collection joins.**, License: **LGPL**

### [Amisa Server](http://www.amisalabs.com/)

Fast distributed real-time analytics and search engine. API: **REST and many languages.**Query method: **AQL (identical to SQL syntactically and functionally).** Written in **C++**. Concurrency: **MVCC**. Misc: ACID transactions, distributed file system, static and dynamic schema support, in memory processing, high availability. Commercial license.

### [densodb](http://www.densodb.net/)

DensoDB is a new NoSQL document database. Written for .Net environment in c# language. It’s simple, fast and reliable. [Source](https://github.com/teamdev/densodb)

### [SisoDB](http://www.sisodb.com/)

A Document Store on top of SQL-Server.

### [SDB](http://pagenotes.com/wordpress/2011/12/08/sdb/)

For small online databases, PHP / JSON interface, implemented in PHP.

### [NoSQL embedded db](https://github.com/petersirka/nosql)

Node.js asynchronous NoSQL embedded database for small websites or projects. Database supports: insert, update, remove, drop and supports views (create, drop, read). Written in JavaScript, no dependencies, implements small concurrency model.[CloudKit, Perservere,[Jackrabbit](http://jackrabbit.apache.org/)]

### [ThruDB](http://code.google.com/p/thrudb/)

(please help provide more facts!) Uses Apache [Thrift](http://incubator.apache.org/thrift/) to integrate multiple backend databases as BerkeleyDB, Disk, MySQL, S3.

## Key Value / Tuple Store

### [DynamoDB](http://aws.amazon.com/dynamodb/)

Automatic ultra scalable NoSQL DB based on fast SSDs. Multiple Availability Zones. Elastic MapReduce Integration. Backup to S3 and much more...

### [Azure Table Storage](http://msdn.microsoft.com/en-us/library/dd179423.aspx)

Collections of free form entities (row key, partition key, timestamp). Blob and Queue Storage available, 3 times redundant. Accessible via REST or ATOM.

### [Riak](http://riak.basho.com/)

API: **JSON**, Protocol: **REST**, Query Method: **MapReduce term matching** , Scaling: **Multiple Masters**; Written in: **Erlang**, Concurrency: **eventually consistent** (stronger then MVCC via Vector Clocks), Misc**:** ...**Links**: talk [»](http://riak.basho.com/nyc-nosql/),

### [Redis](http://redis.io/)

API: **Tons of languages**, Written in: **C**, Concurrency: **in memory** and saves asynchronous disk after a defined time. Append only mode available. Different kinds of fsync policies. Replication: **Master / Slave**, Misc: **also lists, sets, sorted sets, hashes, queues**. Cheat-Sheet: [»](http://masonoise.files.wordpress.com/2010/03/redis-cheatsheet-v1.pdf), great slides [»](http://blog.simonwillison.net/post/57956858672/redis) Admin UI [»](http://www.servicestack.net/mythz_blog/?p=381) From the Ground up [»](http://blog.mjrusso.com/2010/10/17/redis-from-the-ground-up.html)

### [Aerospike](http://www.aerospike.com/)

Fast & Web Scale DB. In-memory + Native flash. Predictable Performance - balanced 250k/50k TPS reads/writes, 99% under 1 ms. Concurrency: **ACID + Tunable Consistency.** Replication:**Zero Config, Zero Downtime, auto clustering, cross datacenter replication, rolling upgrades.** Written in:**C**. APIs: **Many**. Links: [Native Flash/ SSDs,](http://highscalability.com/blog/2012/12/10/switch-your-databases-to-flash-storage-now-or-youre-doing-it.html) [1M TPS on $5k server,](http://highscalability.com/blog/2012/9/10/russ-10-ingredient-recipe-for-making-1-million-tps-on-5k-har.html) [17x lower TCO,](http://wikibon.org/wiki/v/Data_in_DRAM_is_a_Flash_in_the_Pan) [Zero Downtime](http://www.aerospike.com/blog/impact-of-real-time-big-data-on-the-business/), [Magic Quadrant](http://www.aerospike.com/only-visionary-in-gartner-mq-2013/)

### [FoundationDB](http://www.foundationdb.com/)

An ordered key-value store with **multikey ACID transactions, replicated storage, and fault tolerance, built on a shared-nothing, distributed** architecture. API: **Python, Ruby, Node, Java, C**. Written In: **Flow, C++**. Data models: **layers for tuples, arrays, tables, graphs, documents, indexes**.

### [LevelDB](http://code.google.com/p/leveldb/)

**Fast** & Batch updates. DB from **Google**. Written in C++. Blog [»](http://google-opensource.blogspot.com/2011/07/leveldb-fast-persistent-key-value-store.html), hot Benchmark [»](http://leveldb.googlecode.com/svn/trunk/doc/benchmark.html), Article [»](http://www.golem.de/1107/85298.html) (in German). Java [access](https://github.com/fusesource/leveldbjni).

### [Berkeley DB](http://www.oracle.com/database/berkeley-db/db/index.html)

API: **Many languages**, Written in: **C**, Replication: **Master / Slave**, Concurrency: **MVCC**, License: **Sleepycat**, [Berkeley DB Java Edition](http://www.oracle.com/technetwork/database/berkeleydb/overview/index-093405.html): API: **Java**, Written in: **Java**, Replication: **Master / Slave**, Concurrency: **serializable transaction isolation**, License: **Sleepycat**

### [Oracle NOSQL Database](http://www.oracle.com/technetwork/database/database-technologies/nosqldb/overview/index.html)

Oracle NoSQL Database is a distributed key-value database. It is designed to provide highly reliable, scalable and available data storage across a configurable set of systems that function as storage nodes. NoSQL and the Enterprise Data is stored as key-value pairs, which are written to particular storage node(s), based on the hashed value of the primary key. Storage nodes are replicated to ensure high availability, rapid failover in the event of a node failure and optimal load balancing of queries. API: Java/C.

### [GenieDB](http://www.geniedb.com/)

Immediate consistency sharded KV store with an eventually consistent AP store bringing eventual consistency issues down to the theoretical minimum. It features efficient record coalescing. GenieDB speaks SQL and co-exists / do intertable joins with SQL RDBMs.

### [BangDB](http://www.iqlect.com/)

API: **Get,Put,Delete**, Protocol: **Native, HTTP,** Flavor: **Embedded, Network, Elastic Cache**, Replication: **P2P based Network Overlay**, Written in: **C++**, Concurrency: ?, Misc: **robust, crash proof, Elastic, throw machines to scale linearly, Btree/Ehash**

### [Chordless](http://sourceforge.net/projects/chordless/)

API: **Java & simple RPC to vals**, Protocol: **internal**, Query Method: **M/R inside value objects**, Scaling: **every node is master for its slice of namespace**, Written in: **Java**, Concurrency: **serializable transaction isolation**,

### [Scalaris](http://code.google.com/p/scalaris/)

(please help provide more facts!) Written in: **Erlang**, Replication: **Strong consistency over replicas**, Concurrency: **non blocking Paxos**.

### [Tokyo Cabinet / Tyrant](http://fallabs.com/tokyocabinet/)

**Links**: nice talk [»](http://www.infoq.com/presentations/grigorik-tokyo-cabinet-recipes), slides [»](http://www.scribd.com/doc/12016121/Tokyo-Cabinet-and-Tokyo-Tyrant-Presentation), Misc: **Kyoto** Cabinet [»](http://fallabs.com/kyotocabinet/)

### [Scalien](http://scalien.com/)

API / Protocol: **http** (text, html, JSON)**, C, C++, Python**, Java, Ruby, PHP,Perl. Concurrency: **Paxos**.

### [Voldemort](http://project-voldemort.com/)

Open-Source implementation of Amazons Dynamo Key-Value Store.

### [Dynomite](http://wiki.github.com/cliffmoon/dynomite/dynomite-framework)

Open-Source implementation of Amazons Dynamo Key-Value Store. written in Erlang. With "data partitioning, versioning, and read repair, and user-provided storage engines provide persistence and query processing".

### [KAI](http://sourceforge.net/projects/kai/)

Open Source Amazon Dnamo implementation, Misc: [slides](http://www.slideshare.net/takemaru/kai-an-open-source-implementation-of-amazons-dynamo-472179)

### [MemcacheDB](http://memcachedb.org/)

API: **Memcache protocol** (get, set, add, replace, etc.), Written in: **C**, Data Model: **Blob**, Misc: Is Memcached writing to BerkleyDB.

### [Faircom C-Tree](http://www.faircom.com/nosql)

API: **C, C++, C#, Java, PHP, Perl**, Written in: **C,C++**. Misc: **Transaction logging. Client/server. Embedded. SQL wrapper** (not core). Been around since 1979.

### [LSM](http://www.sqlite.org/src4/artifact/41b08c1d31c156d3916558aad89b7e7ae8a381c5)

Key-Value database that was written as part of SQLite4, They claim it is faster then LevelDB. Instead of supporting custom comparators, they have a recommended data encoding for keys that allows various data types to be sorted.

### [KitaroDB](http://www.kitarodb.com/)

: A fast, efficient on-disk data store for Windows Phone 8, Windows RT, Win32 (x86 & x64) and**.NET**. Provides for key-value and multiple segmented key access. APIs for **C#, VB, C++, C** and**HTML5/JavaScript**. Written in **pure C** for high performance and low footprint. Supports async and synchronous operations with 2GB max record size.

### [HamsterDB](http://hamsterdb.com/)

: (embedded solution) ACID Compliance, Lock Free Architecture (transactions fail on conflict rather than block), Transaction logging & fail recovery (redo logs), In Memory support – can be used as a non-persisted cache, B+ Trees – supported [Source: Tony Bain [»](http://blog.tonybain.com/)]

### [STSdb](http://stsdb.com/)

API: **C#**, Written in **C#**, embedded solution, generic XTable<TKey,TRecord> implementation, ACID transactions, snapshots, table versions, shared records, vertical data compression, custom compression, composite & custom primary keys, available backend file system layer, works over multiple volumes, petabyte scalability, LINQ.

### [Tarantool/Box](https://github.com/mailru/tarantool)

API: **C, Perl, PHP, Python, Java and Ruby**. Written in: **Objective C** ,Protocol:**asynchronous binary, memcached, text (Lua console)**. Data model: **collections of dimensionless tuples, indexed using primary + secondary keys**. Concurrency: **lock-free in memory, consistent with disk (write ahead log).** Replication: **master/slave, configurable**. Other: **call Lua stored procedures.**

### [Maxtable](http://code.google.com/p/maxtable/)

API: **C**, Query Method: **MQL, native API**, Replication: **DFS Replication**, Consistency: **strict consistency** Written in: **C**.

### [quasardb](https://www.quasardb.net/)

very high-performance associative database. Highly scalable. API: **C, C++, Java, Python and (limited) RESTful** Protocol: **binary** Query method: **key-value, iteration,** Replication: **Distributed**, Written in:**C++ 11/Assembly**, Concurrency: **ACID**, Misc: **built-in data compression, native support for FreeBSD, Linux and Windows**. License: **Commercial**.

### [Pincaster](http://github.com/jedisct1/Pincaster)

For geolocalized apps. Concurrency: **in-memory with asynchronous disk writes**. API:**HTTP/JSON**. Written in: **C**. License: **BSD**.

### [RaptorDB](http://www.codeproject.com/KB/database/RaptorDB.aspx)

A pure key value store with optimized b+tree and murmur hashing. (In the near future it will be a JSON document database much like mongodb and couchdb.)

### [TIBCO Active Spaces](https://ssl.tibcommunity.com/blogs/activespaces)

peer-to-peer distributed in-memory (with persistence) datagrid that implements and expands on the concept of the Tuple Space. Has SQL Queries and ACID (=> NewSQL).

### [allegro-C](http://www.allegro-c.de/)

Key-Value concept. Variable number of keys per record. Multiple key values, Hierarchic records. Relationships. Diff. record types in same DB. Indexing: B\*-Tree. All aspects configurable. Full scripting language. Multi-user ACID. Web interfaces (PHP, Perl, ActionScript) plus Windows client.

### [nessDB](https://github.com/shuttler/nessDB)

A fast key-value Database (using LSM-Tree storage engine), API: **Redis protocol**(SET,MSET,GET,MGET,DEL etc.), Written in: **ANSI C**

### [HyperDex](http://hyperdex.org/)

Distributed searchable key-value store. Fast (latency & throughput), scalable, consistent, fault tolerance, using hyperscpace hashing. APIs for C, C++ and Python.

### [SharedHashFile](https://github.com/simonhf/sharedhashfile)

Fast, open source, shared memory (using memory mapped files e.g. in /dev/shm or on SSD), multi process, hash table, e.g. on an 8 core i7-3720QM CPU @ 2.60GHz using /dev/shm, 8 processes combined have a 12.2 million / 2.5 to 5.9 million TPS read/write using small binary keys to a hash filecontaining 50 million keys. Uses sharding internally to mitigate lock contention. Written in **C**.

### [Symas LMDB](http://symas.com/mdb/)

Ultra-fast, ultra-compact key-value embedded data store developed by Symas for the OpenLDAP Project. It uses memory-mapped files, so it has the read performance of a pure in-memory database while still offering the persistence of standard disk-based databases, and is only limited to the size of the virtual address space, (it is not limited to the size of physical RAM)

### [Sophia](http://sphia.org/)

Sophia is a modern embeddable key-value database designed for a high load environment. It has unique architecture that was created as a result of research and rethinking of primary algorithmical constraints, associated with a getting popular Log-file based data structures, such as LSM-tree. Implemented as a small C-written, BSD-licensed library.

### [PickleDB](http://pythonhosted.org/pickleDB/)

Redis inspired K/V store for Python object serialization.

### [Mnesia](http://www.erlang.org/doc/apps/mnesia/index.html)

(ErlangDB [»](http://www.infoq.com/news/2007/08/mnesia))

### [LightCloud](http://opensource.plurk.com/LightCloud/)

(based on Tokyo Tyrant)

### [Hibari](http://hibari.sourceforge.net/)

Hibari is a highly available, strongly consistent, durable, distributed key-value data store

### [OpenLDAP](http://highlandsun.com/hyc/mdb/)

Key-value store, B+tree. Lightning fast reads+fast bulk loads. Memory-mapped files for persistent storage with all the speed of an in-memory database. No tuning conf required. Full ACID support. MVCC, readers run lockless. Tiny code, written in C, compiles to under 32KB of x86-64 object code. Modeled after the BerkeleyDB API for easy migration from Berkeley-based code. Benchmarks against LevelDB, Kyoto Cabinet, SQLite3, and BerkeleyDB are available, plus full paper and presentation slides.

### [Genomu](http://genomu.com/)

High availability, concurrency-oriented event-based K/V database with transactions and causal consistency. Protocol: **MsgPack**, API: **Erlang, Elixir, Node.js**. Written in: [Elixir](http://elixir-lang.org/), [Github-Repo](https://github.com/genomu/genomu).

### [BinaryRage](https://github.com/mchidk/BinaryRage)

BinaryRage is designed to be a lightweight ultra fast key/value store for .NET with no dependencies. Tested with more than 200,000 complex objects written to disk per second on a crappy laptop :-) No configuration, no strange driver/connector, no server, no setup - simply reference the dll and start using it in less than a minute.

### [Elliptics](http://www.ioremap.net/projects/elliptics/)

Github Page [»](https://github.com/reverbrain/elliptics)

[Scality [»](http://www.scality.com/), KaTree [»](http://www.katree.net/) TomP2P [»](http://tomp2p.net/), Kumofs [»](http://github.com/etolabo/kumofs) , TreapDB [»](http://code.google.com/p/treapdb/), Wallet [»](https://github.com/YaroslavGaponov/wallet) , NoSQLz [»](http://nosqlz.com/), NMDB, luxio, actord, keyspace, flare, schema-free, RAMCloud]

[SubRecord, Mo8onDb, Dovetaildb]

## Graph Databases [»](http://www.graph-database.org/)

### [Neo4J](http://www.neo4j.org/)

API: **lots of langs**, Protocol: **Java embedded / REST**, Query Method: **SparQL, nativeJavaAPI, JRuby**, Replication: **typical MySQL style master/slave**, Written in: **Java**, Concurrency: **non-block reads, writes locks involved nodes/relationships until commit**, Misc: **ACID** possible, Links: Video [»](http://www.infoq.com/presentations/emil-eifrem-neo4j), good Blog [»](http://blog.neo4j.org/)

### [Infinite Graph](http://www.infinitegraph.com/)

(by Objectivity) API: **Java**, Protocol: **Direct Language Binding**, Query Method: **Graph Navigation API,** **Predicate Language Qualification**, Written in: **Java (Core C++)**, Data Model: **Labeled Directed Multi Graph**, Concurrency: **Update locking on subgraphs, concurrent non-blocking ingest**, Misc:**Free for Qualified Startups**.

### [DEX](http://www.sparsity-technologies.com/dex.php)

: API: **Java, .NET, C++, Blueprints Interface** Protocol: **Embedded**, Query Method: **APIs (Java, .Net, C++) + Gremlin (via Blueprints)**, Written in: **C++**, Data Model: **Labeled Directed Attributed Multigraph**, Concurrency: **yes**, Misc: Free community edition up to 1 Mio nodes, Links: Intro [»](http://www.sparsity-technologies.com/dex_tutorials#dex_intro), Tutorial [»](http://www.sparsity-technologies.com/dex_tutorials#dex_tutorial_java)

### [TITAN](https://github.com/thinkaurelius/titan/wiki)

: API: **Java, Blueprints, Gremlin, Python, Clojure** Protocol: **Thrift, RexPro(Binary), Rexster (HTTP/REST)** Query Method: **Gremlin, SPARQL** Written In: **Java** Data Model: **labeled Property Graph, directed, multi-graph adjacency list** Concurrency: **ACID Tunable C** Replication: **Multi-Master** License:**Apache 2** Pluggable backends: **Cassandra, HBase, MapR M7 Tables, BDB, Persistit, Hazelcast** Links: [Titan User Group](https://groups.google.com/forum/#!forum/aureliusgraphs)

### [InfoGrid](http://infogrid.org/)

API: **Java, http/REST**, Protocol: **as API + XPRISO, OpenID, RSS, Atom, JSON, Java embedded**, Query Method: **Web user interface with html, RSS, Atom, JSON output, Java native**, Replication: **peer-to-peer**, Written in: **Java,** Concurrency: **concurrent reads, write lock within one MeshBase**, Misc: Presentation [»](http://www.slideshare.net/infogrid/info-grid-core-ideas)

### [HyperGraphDB](http://www.kobrix.com/hgdb.jsp)

API: **Java** (and Java Langs), Written in:**Java**, Query Method: **Java** or P2P, Replication:**P2P**, Concurrency: **STM**, Misc: Open-Source, Especially for AI and Semantic Web.

### [GraphBase](http://graphbase.net/)

Sub-graph-based API, query language, tools & transactions. Embedded Java, remote-proxy Java or REST. Distributed storage & processing. Read/write all Nodes. Permissions & Constraints frameworks. Object storage, vertex-embedded agents. Supports multiple graph models. Written in Java

### [Trinity](http://research.microsoft.com/en-us/projects/trinity/)

API: **C#**, Protocol: **C# Language Binding**, Query Method: **Graph Navigation API**, Replication: **P2P with Master Node**, Written in: **C#**, Concurrency: **Yes (Transactional update in online query mode, Non-blocking read in Batch Mode)** Misc: **distributed in-memory storage, parallel graph computation platform (Microsoft Research Project)**

### [AllegroGraph](http://www.franz.com/agraph/)

API: **Java, Python, Ruby, C#, Perl, Clojure, Lisp** Protocol: **REST**, Query Method: **SPARQL**and **Prolog**, Libraries: **Social Networking Analytics** & **GeoSpatial**, Written in: **Common** **Lisp**, Links: Learning Center [»](http://www.franz.com/agraph/support/learning/), Videos [»](http://www.franz.com/agraph/services/conferences_seminars/)

### [BrightstarDB](http://www.brightstardb.com/)

A native, **.NET**, semantic web database with code first Entity Framework, **LINQ** and**OData** support. API: **C#**, Protocol: **SPARQL HTTP, C#**, Query Method: **LINQ, SPARQL**, Written in: **C#**

### [Bigdata](http://www.systap.com/bigdata.htm)

API: **Java, Jini service discovery**, Concurrency: **very high (MVCC)**, Written in: **Java**, Misc: GPL + commercial, Data: **RDF data with inference, dynamic key-range sharding of indices**, Misc: Blog [»](http://www.bigdata.com/blog) (parallel database, high-availability architecture, immortal database with historical views)

### [Meronymy](http://www.meronymy.com/)

RDF enterprise database management system. It is cross-platform and can be used with most programming languages. Main features: high performance, guarantee database transactions with ACID, secure with ACL's, SPARQL & SPARUL, ODBC & JDBC drivers, RDF & RDFS. [»](http://en.wikipedia.org/wiki/Meronymy_SPARQL_Database_Server)

### [WhiteDB](http://whitedb.org/)

WhiteDB is a fast lightweight graph/N-tuples shared memory database library written in C with focus on speed, portability and ease of use. Both for Linux and Windows, dual licenced with GPLv3 and a free nonrestrictive royalty-free commercial licence.

### [OpenLink Virtuoso](http://www.openlinksw.com/)

**Hybrid** DBMS covering the following models: **Relational, Document, Graph**

### [VertexDB](http://www.dekorte.com/projects/opensource/vertexdb/)

### [FlockDB](http://github.com/twitter/flockdb)

by twitter [»](http://gigaom.com/2010/04/12/twitter-open-sources-the-home-of-its-social-graph/) [»](http://github.com/twitter/flockdb)

### [BrightstarDB](http://www.brightstardb.com/)

### [Execom IOG](http://iog.codeplex.com/)

### [Fallen 8](http://www.fallen-8.com/)

Github [»](https://github.com/cosh/fallen-8)

[Java Universal Network / Graph Framework, OpenRDF / Sesame, Filament, OWLim, NetworkX, iGraph, Jena]

List of SPARQL implementations can be found [here](http://www.w3.org/wiki/SparqlImplementations)

## Multimodel Databases

### [ArangoDB](http://www.arangodb.org/)

API: **REST, Graph Blueprints, C#, D, Ruby, Python, Java, PHP, Go, Python, etc.** Data Model:**K/V, JSON & graphs with shapes,** Protocol: **HTTP using JSON**, Query Method: **declarative AQL, query by example, map/reduce, key/value,** Replication: **master-slave (m-m to follow)**, Sharding: **automatic and configurable** Written in: **C/C++/Javascript (V8 integrated),** Concurrency: **MVCC, tunable** Misc: **"stored procedures" (Ruby & Javascript)**, many indices as **secondary, fulltext, geo, hash, Skip-list, bit-array, n-gram, capped collections**

### [OrientDB](http://www.orientechnologies.com/)

Languages: **Java**, Schema: **Has features of an Object-Database, DocumentDB, GraphDB or Key-Value DB**, Written in: **Java**, Query Method: **Native and SQL**, Misc: **really fast, lightweight, ACID with recovery**.

### [Datomic](http://www.datomic.com/)

API: **Many jvm languages**, Protocol: **Native + REST**, Query Method: **Datalog + custom extensions**, Scaling: **elastic via underlying DB (in-mem, DynamoDB, Riak, CouchBase, Infinispan, more to come)**, Written in: **Clojure**, Concurrency: **ACID** MISC: smart caching, unlimited read scalability, full-text search, cardinality, bi-directional refs 4 graph traversal, loves Clojure + Storm.

### [FatDB](http://fatcloud.com/net_nosql_database.html)

.NET solution with tight SQL Server integration. API: **C#** Protocol: **Protobuf or Raw Binary** Query Method: **LINQ** Replication: **All peer network, multiple consistency strategies** Written in: **C#, .NET**Concurrency: **Variable, Many Strategies** License: **Free Community Edition + Commercial Options** Misc:**Bi-Directional SQL Server sync, Integrated File Management System, Asynchronous Work Queue, Unified Routing, Fault Tolerance, Hosting agnostic (in-house, AWS, Azure etc)** Links: [1](http://fatcloud.com/overview.html), [2](http://fatcloud.com/sql.html) [Free Download](http://www.fatcloud.com/download.html)

### [AlchemyDB](http://code.google.com/p/alchemydatabase/)

GraphDB + RDBMS + KV Store + Document Store. Alchemy Database is a low-latency high-TPS NewSQL RDBMS embedded in the NOSQL datastore redis. Extensive datastore-side-scripting is provided via deeply embedded Lua. Bought and integrated with Aerospike.

### [CortexDB](http://www.cortex-ag.com/cms.php?i=206ac3c9f22020202021202a)

CortexDB is a dynamic schema-less multi-model data base providing nearly all advantages of up to now known NoSQL data base types (key-value store, document store, graph DB, multi-value DB, column DB) with dynamic re-organization during continuous operations, managing analytical and transaction data for agile software configuration,change requests on the fly, self service and low footprint.

The following section containts **Soft NoSQL Systems**

[Mostly NOT originated out of a Web 2.0 need but worth a look for great non relational solutions]

## Object Databases [»](http://odbms.org/)

### [Versant](http://www.versant.com/)

API: Languages/Protocol: **Java, C#, C++, Python**. Schema: language class model (easy changable). Modes: **always consistent and eventually consistent** Replication: **synchronous fault tolerant and peer to peer asynchronous**. Concurrency: **optimistic and object based locks**. Scaling: **can add physical nodes on fly for scale out/in** **and migrate objects between nodes without impact to application code**. Misc: **MapReduce via parallel SQL like query across logical database groupings.**

### [db4o](http://db4o.com/)

API: **Java, C#, .Net Langs**, Protocol: **language**, Query Method: **QBE (by Example), Soda, Native Queries, LINQ (.NET)**, Replication: **db4o2db4o & dRS to relationals**, Written in: **Java**, Cuncurrency: **ACID serialized**, Misc: **embedded lib,** **Links**: DZone Refcard #53 [»](http://refcardz.dzone.com/refcardz/getting-started-db4o), Book [»](http://www.amazon.com/Definitive-Guide-db4o-Stefan-Edlich/dp/1590596560/),

### [Objectivity](http://www.objectivity.com/)

API: Languages: **Java, C#, C++, Python, Smalltalk, SQL access through ODBC**. Schema:**native language class model, direct support for references, interoperable across all language bindings. 64 bit unique object ID (OID) supports multi exa-byte**. Platforms: **32 and 64 bit Windows, Linux, Mac OSX, \*Unix**. Modes: **always consistent (ACID).** Concurrency: **locks at cluster of objects (container) level.**Scaling: **unique distributed architecture, dynamic addition/removal of clients & servers, cloud environment ready.** Replication: **synchronous with quorum fault tolerant across peer to peer partitions.**

### [GemStone/S](http://gemtalksystems.com/)

API: **Java, C, C++, Smalltalk** Schema: **language class model** Platforms: **Linux, AIX, Solaris, Windows clients** Modes: **always consistent (ACID)** Replication: **shared page cache per node, hot standby failover** Concurrency: **optimistic and object based locks** Scaling: **arbitrarily large number of nodes** Misc:**SQL access via GemConnect**

### [Starcounter](http://www.starcounter.com/)

API: **C# (.NET languages)**, Schema: **Native language class model**, Query method: **SQL**, Concurrency: **Fully ACID compliant**, Storage: **In-memory with transactions secured on disk**, Reliability:**Full checkpoint recovery**, Misc: **VMDBMS - Integrating the DBMS with the virtual machine for maximal performance and ease of use**.

### [Perst](http://www.mcobject.com/perst)

API: **Java,Java ME,C#,Mono**. Query method: **OO via Perst collections, QBE, Native Queries, LINQ, native full-text search, JSQL** Replication: **Async+sync (master-slave)** Written in: **Java, C#**. Caching: **Object cache (LRU, weak, strong), page pool, in-memory database** Concurrency: **Pessimistic+optimistic (MVCC)+ async or sync (ACID)** Index types: **Many tree models + Time Series**. Misc: Embedded lib., encryption, automatic recovery, native full text search, on-line or off-line backup.

### [VelocityDB](http://www.velocitydb.com/)

Written in**100% pure C#**, Concurrency: **ACID/transactional, pessimistic/optimistic locking,**Misc: compact data, B-tree indexes, LINQ queries, 64bit object identifiers (Oid) supporting multi millions of databases and high performance. Deploy with a single DLL of around 400KB.

### [HSS Database](http://highspeed-solutions.net/db.aspx)

Written in: **100% C#**, The HSS DB v3.0 (HighSpeed-Solutions Database), is a client based, zero-configuration, auto schema evolution, acid/transactional, LINQ Query, DBMS for Microsoft .NET 4/4.5, Windows 8 (Windows Runtime), Windows Phone 7.5/8, Silverlight 5, MonoTouch for iPhone and Mono for Android

### [ZODB](http://zodb.org/)

API: **Python**, Protocol: **Internal, ZEO**, Query Method: **Direct object access, zope.catalog, gocept.objectquery,** Replication: **ZEO, ZEORAID, RelStorage** Written in: **Python, C** Concurrency: **MVCC**, License: **Zope Public License (OSI approved)** Misc: Used in production since 1998

### [Magma](http://wiki.squeak.org/squeak/2665)

Smalltalk DB, optimistic locking, Transactions, etc.

### [NEO](http://www.neoppod.org/)

API: **Python - ZODB "Storage" interface**, Protocol: **native**, Query Method: **transactional key-value**, Replication: **native**, Written in: **Python**, Concurrency: **MVCC** (internally), License: **GPL "v2 or later"**, Misc: Load balancing, fault tolerant, hot-extensible.

### [siaqodb](http://siaqodb.com/)

An object database engine that currently runs on **.NET, Mono, Silverlight,Windows Phone 7, MonoTouch, MonoAndroid, CompactFramework**; It has implemented a Sync Framework Provider and can be **synchronized** with **MS SQLServer**; Query method:**LINQ**;

### [Sterling](http://sterling.codeplex.com/)

is a lightweight object-oriented database for .NET with support for Silverlight and Windows Phone 7. It features in-memory keys and indexes, triggers, and support for compressing and encrypting the underlying data.

### [Morantex](http://www.morantex.com/)

Stores .NET classes in a datapool. Build for speed. SQL Server integration. LINQ support.

### [EyeDB](http://www.eyedb.org/)

EyeDB is an **LGPL OODBMS**, provides an advanced object model (inheritance, collections, arrays, methods, triggers, constraints, reflexivity), an object definition language based on ODMG ODL, an object query and manipulation language based on ODMG OQL. Programming interfaces for C++ and Java.

### [FramerD](http://www.framerd.org/)

Object-Oriented Database designed to support the maintenance and sharing of knowledge bases. Optimized for pointer-intensive data structures used by semantic networks, frame systems, and many intelligent agent applications. Written in: **ANSI C**.

### [Ninja Database Pro](http://www.kellermansoftware.com/p-43-ninja-net-database-pro.aspx)

Ninja Database Pro is a .NET ACID compliant relational object database that supports transactions, indexes, encryption, and compression. It currently runs on .NET Desktop Applications, Silverlight Applications, and Windows Phone 7 Applications.

### [NDatabase](http://ndatabase.wix.com/home)

API: **C#, .Net, Mono, Windows Phone 7, Silverlight,** Protocol: **language,** Query Method: Soda, LINQ (.NET), Written in: **C#**, Misc: embedded lib, indexes, triggers, handle circular ref, LinqPad support, Northwind sample, refactoring, in-memory database, Transactions Support (ACID) and more, Documentation: [»](http://ndatabase.net/)

### [PicoLisp](http://www.picolisp.com/)

Language and Object Database, can be viewed as a Database Development Framework. Schema: **native language class model with relations + various indexes**. Queries: **language build in + a small Prolog like DSL Pilog**. Concurrency: **synchronization + locks**. Replication, distribution and fault tolerance is not implemented per default but can be implemented with native functionality. Written in C (32bit) or assembly (64bit).

### [acid-state](http://acid-state.seize.it/)

API: **Haskell**, Query Method: **Functional programming**, Written in: **Haskell**, Concurrency:**ACID, GHC concurrent runtime**, Misc: In-memory with disk-based log, supports remote access Links: Wiki[»](http://acid-state.seize.it/), Docs [»](http://hackage.haskell.org/packages/archive/acid-state/0.8.3/doc/html/Data-Acid.html)

### [ObjectDB](http://www.objectdb.com/)

API: **Java (JPA / JDO)** Query method: **JPA JPQL, JDO JDOQL** Replication: **Master-slave** Written in: **100% Pure Java** Caching: **Object cache, Data cache, Page cache, Query Result cache, Query program cache** Concurrency: **Object level locking (pessimistic + optimistic)** Index types: **BTree, single, path, collection** Misc: Used in production since 2004, Embedded mode, Client Server mode, automatic recovery, on-line backup.[ BergDB [»](http://bergdb.com/), StupidDB [»](http://www.aztec-project.org/), KiokuDB [»](http://www.iinteractive.com/kiokudb/) (Perl solution), Durus [»](http://www.mems-exchange.org/software/durus/)]

## Grid & Cloud Database Solutions

### [GigaSpaces](http://www.gigaspaces.com/)

Popular SpaceBased Grid Solution.

### [GemFire](http://www.vmware.com/products/application-platform/vfabric-gemfire)

GemFire offers in-memory globally distributed data management with dynamic scalability, very high performance and granular control supporting the most demanding applications. Well integrated with the Spring Framework, developers can quickly and easily provide sophisticated data management for applications. With simple horizontal scale-out, data latency caused by network roundtrips and disk I/O can be avoided even as applications grow.

### [Infinispan](http://www.jboss.org/infinispan.html)

scalable, highly available data grid platform, **open source**, written in **Java**.

### [Queplix](http://www.queplix.com/)

NOSQL Data Integration Environment, can integrate relational, object, BigData – NOSQL easily and without any SQL.

### [Hazelcast](http://www.hazelcast.com/)

Hazelcast is a in-memory data grid that offers distributed data in Java with dynamic scalability under the Apache 2 open source license. It provides distributed data structures in Java in a single Jar file including hashmaps, queues, locks, topics and an execution service that allows you to simply program these data structures as pure java objects, while benefitting from symmetric multiprocessing and cross-cluster shared elastic memory of very high ingest data streams and very high transactional loads.[[GridGain](http://www.gridgain.com/), [ScaleOut Software](http://www.scaleoutsoftware.com/), [Galaxy](http://puniverse.github.com/galaxy/), [Joafip](http://joafip.sourceforge.net/), Coherence, eXtremeScale]

## XML Databases

### [EMC Documentum xDB](http://www.emc.com/products/detail/software/documentum-xdb.htm)

(commercial system) API: Java, XQuery, Protocol: WebDAV, web services, Query method: XQuery, XPath, XPointer, Replication: lazy primary copy replication (master/replicas), Written in: Java, Concurrency: concurrent reads, writes with lock; transaction isolation, Misc: Fully transactional persistent DOM; versioning; multiple index types; metadata and non-XML data support; unlimited horizontal scaling. Developer Network [»](http://developer.emc.com/xmltech)

### [eXist](http://exist-db.org/)

API: XQuery, XML:DB API, DOM, SAX, Protocols: HTTP/REST, WebDAV, SOAP, XML-RPC, Atom, Query Method: XQuery, Written in: Java (open source), Concurrency: Concurrent reads, lock on write; Misc: Entire web applications can be written in XQuery, using XSLT, XHTML, CSS, and Javascript (for AJAX functionality). (1.4) adds a new full text search index based on Apache Lucene, a lightweight URL rewriting and MVC framework, and support for XProc.

### [Sedna](http://modis.ispras.ru/sedna/)

Misc: ACID transactions, security, indices, hot backup. Flexible XML processing facilities include W3C XQuery implementation, tight integration of XQuery with full-text search facilities and a node-level update language.

### [BaseX](http://basex.org/)

BaseX is a fast, powerful, lightweight XML database system and XPath/XQuery processor with highly conformant support for the latest W3C Update and Full Text Recommendations. Client/Server architecture, ACID transaction support, user management, logging, Open Source, BSD-license, written in Java, runs out of the box.

### [Qizx](http://www.xmlmind.com/qizx/)

commercial and open source version, API: Java, Protocols: HTTP, REST, Query Method: XQuery, XQuery Full-Text, XQuery Update, Written in: Java, full source can be purchased, Concurrency: Concurrent reads & writes, isolation, Misc: Terabyte scalable, emphasizes query speed.

### [Berkeley DB XML](http://www.oracle.com/database/berkeley-db/xml/index.html)

API: Many languages, Written in: C++, Query Method: XQuery, Replication: Master / Slave, Concurrency: MVCC, License: Sleepycat

[ [Xindice](http://xml.apache.org/xindice/) [Tamino](http://www.softwareag.com/de/products/wm/tamino/) ]

## Multidimensional Databases

### [Globals:](http://globalsdb.org/)

by Intersystems, multidimensional array.Node.js API, array based APIs (Java / .NET), and a Java based document API.

### [Intersystems Cache](http://www.intersystems.com/)

Postrelational System. Multidimensional array APIs, Object APIs, Relational Support (Fully SQL capable JDBC, ODBC, etc.) and Document APIs are new in the upcoming 2012.2.x versions. Availible for Windows, Linux and OpenVMS.

### [GT.M](http://fis-gtm.com/)

API: **M, C, Python, Perl**, Protocol: **native, inprocess C**, Misc: Wrappers: **M/DB for SimpleDB compatible HTTP** [»](http://www.mgateway.com/mdb.html), **MDB:X** for XML [»](http://mgateway.com/), **PIP** for mapping to tables for SQL [»](http://fis-pip.com/), Features: Small footprint (17MB), Terabyte Scalability, Unicode support, Database encryption, Secure, ACID transactions (single node), eventual consistency (replication), License: AGPL v3 on x86 GNU/Linux, **Links**: Slides [»](http://www.slideshare.net/robtweed/gtm-a-tried-and-tested-schemaless-database),

### [SciDB](http://scidb.org/)

**Array** Data Model for Scientists, [» paper](http://scidb.org/Documents/SciDB-VLDB09-paper.pdf), [» poster](http://scidb.org/Documents/SciDB-VLDB09-poster.pdf), [» HiScaBlog](http://highscalability.com/blog/2010/4/29/product-scidb-a-science-oriented-dbms-at-100-petabytes.html)

### [MiniM DB](http://www.minimdb.com/)

: Multidimensional arrays, API: **M, C, Pascal, Perl, .NET, ActiveX, Java, WEB.** Available for Windows and Linux.

### [rasdaman](http://www.rasdaman.org/)

: Short description: Rasdaman is a scientific database that allows to store and retrieve multi-dimensional raster data (arrays) of unlimited size through an SQL-style query language. API: C++/Java, Written in C++, Query method: SQL-like query language rasql, as well as via OGC standards WCPS, WCS, WPS [link2](http://www.rasdaman.com/)[ [EGTM: GT.M for Erlang,](http://github.com/ztmr/egtm) ["IODB: EGTM-powered ObjectDB for Erlang](http://www.idea.cz/technology) ]

## Multivalue Databases

### [U2](http://www.rocketsoftware.com/u2)

(UniVerse, UniData): MultiValue Databases, Data Structure: MultiValued, Supports nested entities, Virtual Metadata, API: BASIC, InterCall, Socket, .NET and Java API's, IDE: Native, Record Oriented, Scalability: automatic table space allocation, Protocol: Client Server, SOA, Terminal Line, X-OFF/X-ON, Written in: C, Query Method: Native mvQuery, (Retrieve/UniQuery) and SQL, Replication: yes, Hot standby, Concurrency: Record and File Locking (Fine and Coarse Granularity)

### [OpenInsight](http://www.revelation.com/index.php/features)

API: Basic+, .Net, COM, Socket, ODBC, Protocol: TCP/IP, Named Pipes, Telnet, VT100. HTTP/S Query Method: RList, SQL & XPath Written in: Native 4GL, C, C++, Basic+, .Net, Java Replication: Hot Standby Concurrency: table &/or row locking, optionally transaction based & commit & rollback Data structure: Relational &/or MultiValue, supports nested entities Scalability: rows and tables size dynamically

### [TigerLogic PICK](http://www.tigerlogic.com/tigerlogic/pick/database/pickdatabase.jsp)

(D3, mvBase, mvEnterprise) Data Structure: **Dynamic multidimensional PICK data model, multi-valued, dictionary-driven**, API: **NET, Java, PHP, C++,** Protocol: **C/S**, Written In: **C**, Query Method: **AQL, SQL, ODBC, Pick/BASIC**, Replication: **Hot Backup, FFR, Transaction Logging + real-time replication**, Concurrency: **Row Level Locking**, Connectivity: **OSFI, ODBC, Web-Services, Web-enabled**, Security: File level AES-128 encryption

### [Reality](http://www.northgate-reality.com/)

(Northgate IS): The original MultiValue data set database, virtual machine, enquiry and rapid development environment. Delivers ultra efficiency, scalability and resilience while extended for the web and with built-in auto sizing, failsafe and more. Interoperability includes Web Services, Java Classes, XML, ActiveX, Sockets, C and, for those that have to interoperate with the SQL world, ODBC/JDBC and two-way transparent SQL data access.

### [OpenQM](http://www.openqm.com/)

Supports nested data. Fully automated table space allocation. Concurrency control via task locks, file locks & shareable/exclusive record locks. Case insensitivity option. Secondary key indices. Integrated data replication. QMBasic programming language for rapid development. OO programming integrated into QMBasic. QMClient connectivity from Visual Basic, PowerBasic, Delphi, PureBasic, ASP, PHP, C and more. Extended multivalue query language.

### [Model 204 Database](http://www.rocketsoftware.com/m204)

A high performance dbms that runs on IBM mainframes (IBM z/OS, z/VM, zVSE), +SQL interface with nested entity support API: native 4GL (SOUL + o-o support), SQL, Host Language (COBOL, Assembler, PL1) API, ODBC, JDBC, .net, Websphere MQ, Sockets Scalability: automatic table space allocation, 64-bit support Written in: IBM assembler, C Query method: SOUL, SQL, RCL ( invocation of native language from client ) Concurrency: record and file level locking Connectivity: TN3270, Telnet, Http

### [ESENT](http://msdn.microsoft.com/en-us/library/windows/desktop/gg269259%28v=EXCHG.10%29.aspx)

(by Microsoft) ISAM storage technology. Access using index or cursor navigation. Denormalized schemas, wide tables with sparse columns, multi-valued columns, and sparse and rich indexes. C# and Delphi drivers available. Backend for a number of MS Products as Exchange.

### [jBASE](http://www.jbase.com/index.html)

more info [»](http://en.wikipedia.org/wiki/InfinityDB)

## Event Sourcing

### [Event Store](http://geteventstore.com/)

## Network Model

### [Vyhodb](http://www.vyhodb.com/)

**Service oriented, schema-less, network data model DBMS**. Client application invokes methods of vyhodb services, which are written in Java and deployed inside vyhodb. Vyhodb services reads and modifies storage data. API: **Java**, Protocol: RSI - **Remote service invocation**, Written in: Java, ACID: **fully supported**, Replication: **async master slave**, Misc: **online backup**, License: proprietary

## other NoSQL related databases

### [IBM Lotus/Domino](http://www-01.ibm.com/software/lotus/)

Type: Document Store, API: Java, HTTP, IIOP, C API, REST Web Services, DXL, Languages: Java, JavaScript, LotusScript, C, @Formulas, Protocol: HTTP, NRPC, Replication: Master/Master, Written in: C, Concurrency: Eventually Consistent, Scaling: Replication Clusters

### [eXtremeDB](http://www.mcobject.com/extremedbfamily.shtml)

Type: In-Memory Database; Written in: C; API: C/C++, SQL, JNI, C#(.NET), JDBC; Replication: Async+sync (master-slave), Cluster; Scalability: 64-bit and MVCC

### [RDM Embedded](http://www.raima.com/products/rdme/)

APIs: C++, Navigational C. Embedded Solution that is ACID Compliant with Multi-Core, On-Disk & In-Memory Support. Distributed Capabilities, Hot Online Backup, supports all Main Platforms. Supported B Tree & Hash Indexing. Replication: Master/Slave, Concurrency: MVCC. Client/Server: In-process/Built-in.

### [ISIS Family](http://www.unesco.org/webworld/isis/isis.htm)

(semistructured databases) [»](http://bvsmodelo.bvsalud.org/php/level.php?lang=en&component=31)

### [Moonshadow](http://www.moonshadowmobile.com/data-visualization/big-data-visualizer/)

NoSql, in-memory, flat-file, cloud-based. API interfaces. Small data footprint and very fast data retrieval. Stores 200 million records with 200 attributes in just 10GB. Retrieves 150 million records per second per CPU core. Often used to visualize big data on maps. Written in C.

### [VaultDB](http://www.rediosoft.com/)

Next-gen NoSQL encrypted document store. Multi-recipient / group encryption. Featuers: concurrency, indices, ACID transactions, replication and PKI management. Supports PHP and many others. Written in **C++**. Commercial but has a free version. API: **JSON**

### [Prevayler](http://www.prevayler.org/)

Java RAM Data structure journalling.

### [Yserial](http://yserial.sourceforge.net/)

Python wrapper over sqlite3

## unresolved and uncategorized

### Btrieve

(by Pervasive Software) key/index/tupel DB. Using Pages. [»](http://www.wordiq.com/definition/Btrieve) (faq [»](http://www.rothgeb.net/wordpress-2.7.1/?page_id=64))

### [KirbyBase](https://rubygems.org/gems/KirbyBase)

Written in: **Ruby**. github: [»](https://github.com/gurugeek/KirbyBase)

### [Tokutek:](http://tokutek.com/)

### [Recutils:](http://www.gnu.org/software/recutils/)

GNU Tool for text files containing records and fields. Manual [»](http://www.gnu.org/software/recutils/manual/index.html)

### [FileDB:](http://www.eztools-software.com/tools/filedb/)

Mainly targeted to Silverlight/Windows Phone developers but its also great for any .NET application where a simple local database is required, extremely Lightweight - less than 50K, stores one table per file, including index, compiled versions for Windows Phone 7, Silverlight and .NET, fast, free to use in your applications

### [CodernityDB](http://labs.codernity.com/codernitydb/)

written in Python**illuminate Correlation Database** [»](http://www.datainnovationsgroup.com/), **FluidDB (Column Oriented DB)** [»](http://doc.fluidinfo.com/fluidDB/),Fleet DB [»](http://fleetdb.org/), **Btrieve**,

Twisted Storage [»](http://twistedstorage.com/), Java-Chronicle [»](https://github.com/peter-lawrey/Java-Chronicle), Ringo, Sherpa, tin, Dryad, SkyNet, Disco

Possibly the oldest NoSQL DB (together with MUMPS and IBMs IMS & IDMS [1968,1964]): [» **Adabas**](http://documentation.softwareag.com/adabas/ada814mfr/adamf/concepts/cfadais.htm)VSAM by IBM is also a good candidate.