≡ Menu

- **Home**
- **Free eBook**
- **Start Here**
- **Contact**
- **About**

SQL vs NoSQL Database Differences Explained with few Example DB

by Luke P. Issac on January 14, 2014

G+   Like 3   Tweet

Most of you are already familiar with SQL database, and have a good knowledge on either MySQL, Oracle, or other SQL databases. In the last several years, NoSQL database is getting widely adopted to solve various business problems.

It is helpful to understand the difference between SQL and NoSQL database, and some of available NoSQL database that you can play around with.

SQL vs NoSQL: High-Level Differences

- SQL databases are primarily called as Relational Databases (RDBMS); whereas NoSQL database are primarily called as non-relational or distributed database.
- SQL databases are table based databases whereas NoSQL databases are document based, key-value pairs, graph databases or wide-column stores. This means that SQL databases represent data in form of tables which consists of n number of rows of data whereas NoSQL databases are the collection of key-value pair, documents, graph databases or wide-column stores which do not have standard schema definitions which it needs to adhered to.
- SQL databases have predefined schema whereas NoSQL databases have dynamic schema for unstructured data.
- SQL databases are vertically scalable whereas the NoSQL databases are horizontally scalable. SQL databases are scaled by increasing the horse-power of the hardware. NoSQL databases are scaled by increasing the databases servers in the pool of resources to reduce the load.
- SQL databases uses SQL (structured query language) for defining and manipulating the data, which is very powerful. In NoSQL database, queries are focused on collection of documents. Sometimes it is also called as UnQL (Unstructured Query Language). The syntax of using UnQL varies from database to database.
- SQL database examples: MySql, Oracle, Sqlite, Postgres and MS-SQL. NoSQL database examples: MongoDB, BigTable, Redis, RavenDb, Cassandra, Hbase, Neo4j and CouchDb
- For complex queries: SQL databases are good fit for the complex query intensive environment whereas NoSQL databases are not good fit for complex queries. On a high-level, NoSQL don't have standard interfaces to perform complex queries, and the queries themselves in NoSQL are not as powerful as SQL query language.
- For the type of data to be stored: SQL databases are not best fit for hierarchical data storage. But, NoSQL database fits better for the hierarchical data storage as it follows the key-value pair way of storing data similar to JSON data.

NoSQL database are highly preferred for large data set (i.e for big data). Hbase is an example for this purpose.

- For scalability: In most typical situations, SQL databases are vertically scalable. You can manage increasing load by increasing the CPU, RAM, SSD, etc, on a single server. On the other hand, NoSQL databases are horizontally scalable. You can just add few more servers easily in your NoSQL database infrastructure to handle the large traffic.
- For high transactional based application: SQL databases are best fit for heavy duty transactional type applications, as it is more stable and promises the atomicity as well as integrity of the data. While you can use NoSQL for transactions purpose, it is still not comparable and sable enough in high load and for complex transactional applications.
- For support: Excellent support are available for all SQL database from their vendors. There are also lot of independent consultations who can help you with SQL database for a very large scale deployments. For some NoSQL database you still have to rely on community support, and only limited outside experts are available for you to setup and deploy your large scale NoSQL deployments.
- For properties: SQL databases emphasizes on ACID properties (Atomicity, Consistency, Isolation and Durability) whereas the NoSQL database follows the Brewers CAP theorem (Consistency, Availability and Partition tolerance)
- For DB types: On a high-level, we can classify SQL databases as either open-source or close-sourced from commercial vendors. NoSQL databases can be classified on the basis of way of storing data as graph databases, key-value store databases, document store databases, column store database and XML databases.

SQL Database Examples

1. MySQL Community Edition

[MySQL database](#) is very popular open-source database. It is generally been stacked with apache and PHP, although it can be also stacked with nginx and server side javascripting using Node js. The following are some of MySQL benefits and strengths:

- Replication: By replicating MySQL database across multiple nodes the work load can be reduced heavily increasing the scalability and availability of business application
- Sharding: MySQL sharding os useful when there is large no of write operations in a high traffic website. By sharding MySQL servers, the application is partitioned into multiple servers dividing the database into small chunks. As low cost servers can be deployed for this purpose, this is cost effective.
- Memcached as a NoSQL API to MySQL: Memcached can be used to increase the performance of the data retrieval operations giving an advantage of NoSQL api to MySQL server.
- Maturity: This database has been around for a long time and tremendous community input and testing has gone into this database making it very stable.
- Wide range of Platforms and Languages: MySql is available for all major platforms like Linux, Windows, Mac, BSD and Solaris. It also has connectors to languages like Node.js, Ruby, C#, C++, C, Java, Perl, PHP and Python.
- Cost effectiveness: It is open source and free.

2. MS-SQL Server Express Edition

It is a powerful and user friendly database which has good stability, reliability and scalability with support from Microsoft. The following are some of MS-SQL benefits and strengths:

- Integrated Development Environment: Microsoft visual studio, Sql Server Management Studio and Visual Developer tools provide a very helpful way for development and increase the developers productivity.
- Disaster Recovery: It has good disaster recovery mechanism including database mirroring, fail over clustering and RAID partitioning.
- Cloud back-up: Microsoft also provides cloud storage when you perform a cloud-backup of your database

3. Oracle Express Edition

It is a limited edition of [Oracle Enterprise Edition server](#) with certain limitations. This database is free for development and deployment. The following are some of Oracle benefits and strengths:

- Easy to Upgrade: Can be easily upgraded to newer version, or to an enterprise edition.
- Wide platform support: It supports a wide range of platforms including Linux and Windows

- Scalability: Although the scalability of this database is not cost effective as MySQL server, but the solution is very reliable, secure, easily manageable and productive.

NoSQL Database Examples

1. MongoDB

Mongodb is one of the most popular document based NoSQL database as it stores data in JSON like documents. It is non-relational database with dynamic schema. It has been developed by the founders of DoubleClick, written in C++ and is currently being used by some big companies like The New York Times, Craigslist, MTV Networks. The following are some of MongoDB benefits and strengths:

- Speed: For simple queries, it gives good performance, as all the related data are in single document which eliminates the join operations.
- Scalability: It is horizontally scalable i.e. you can reduce the workload by increasing the number of servers in your resource pool instead of relying on a stand alone resource.
- Manageable: It is easy to use for both developers and administrators. This also gives the ability to shard database
- Dynamic Schema: Its gives you the flexibility to evolve your data schema without modifying the existing data

2. CouchDB

CouchDB is also a document based NoSQL database. It stores data in form of JSON documents. The following are some of CouchDB benefits and strengths:

- Schema-less: As a member of NoSQL family, it also have dynamic schema which makes it more flexible, having a form of JSON documents for storing data.
- HTTP query: You can access your database documents using your web browser.
- Conflict Resolution: It has automatic conflict detection which is useful while in a distributed database.
- Easy Replication: Implementing replication is fairly straight forward

3. Redis

Redis is another Open Source NoSQL database which is mainly used because of its lightening speed. It is written in ANSI C language. The following are some of Redis benefits and strengths:

- Data structures: Redis provides efficient data structures to an extend that it is sometimes called as data structure server. The keys stored in database can be hashes, lists, strings, sorted or unsorted sets.
- Redis as Cache: You can use Redis as a cache by implementing keys with limited time to live to improve the performance.
- Very fast: It is consider as one of the fastest NoSQL server as it works with the in-memory dataset.

If you enjoyed this article, you might also like..

1. 50 Linux Sysadmin Tutorials
2. 50 Most Frequently Used Linux Commands (With Examples)
3. Top 25 Best Linux Performance Monitoring and Debugging Tools
4. Mommy, I found it! – 15 Practical Linux Find Command Examples
5. Linux 101 Hacks 2nd Edition eBook **Free**

- Awk Introduction – 7 Awk Print Examples
- Advanced Sed Substitution Examples
- 8 Essential Vim Editor Navigation Fundamentals
- 25 Most Frequently Used Linux IPTables Rules Examples
- Turbocharge PuTTY with 12 Powerful Add-Ons

{ 31 comments… add one }

- Jalal Hajigholamali January 15, 2014, 2:06 am

  Hi,
  Very useful article..
  Thanks.

  Link
- Mohankumar January 15, 2014, 3:20 am

  Hi, thank you very much for this article…i was waiting for this…

  Link
- Dan MC January 15, 2014, 3:40 am

  It can also be added that NoSQL databases tend to represent entities in a more natural way.
  Normalization rules for relational databases often break entities in an artificial way in order to avoid redundancy (which is something NoSQL databases don't care too much about). Thus, relational databases end up, more often than not, with objects that are not good representations of the real world. NoSQL databases do a better job at representing real world objects.
  It's all fresh in my mind, I had my exam on Advanced DataBases just yesterday and thus was one of the questions!!

  Link
- Tarun Trehan January 15, 2014, 4:53 am

  Hi,
  Nice article in a very concise and apt form.
  May be, we can include some real life examples of projects that fall under which umbrella.

  Link
- Bhagyaraj January 15, 2014, 8:12 am

Added more knowledge on my bucket.
Thanks TheGeekStuff for this article.

Link
- Luis Collymoore January 15, 2014, 10:08 am

  Indeed.

  Link
- Arun January 15, 2014, 10:09 am

  Hi,
  Simple & Nice article

  Link
- Tim January 15, 2014, 1:37 pm

  I have always enjoyed reading TGS articles, even if they are sometimes out of my field. On this one, however, you do seem to have missed a vital difference between the two families.

  An SQL database has to have a design, and a designer, and when it's done right it can be _proved_ to be right. There's a load of maths theory that says there's no duplication and no internal integrity failure.

  A non-SQL database is really a pile of stuff, that may be accurate or may not, may be complete or it may not, and almost certainly has duplicated and inconsistent stuff. Not that there isn't a place for uncontrolled piles of stuff; but I don't think it's right to call them databases.

  ☺

  Tim

  Link
- Shiva January 16, 2014, 12:11 am

  Thanks for this useful article. I am going to learn about Mongo DB in detail.

  Link
- Rinshad January 16, 2014, 10:13 am

  Nice article.

  Link
- Pratik January 16, 2014, 11:09 pm

  Nice Article

  Link
- Senthil Kumar January 30, 2014, 11:21 pm

  Learnt something interesting! Thanks for your time!

  Link
- artaxerxes3 February 4, 2014, 2:47 am

  Just small addon, in the part of SQL database you don't remember main competitors of Oracle, means IBM DB2, and IBM Informix.

  Link
- Bastian VS February 19, 2014, 1:41 am

Recently I have read about Couchbase. A proudct which came from memcached and couchdb. Its a Nosql Database. Looks like Viber recently migrated from mongodb to couchbase can you share more details about couchbase as well

[Link](#)
- Jayesh March 12, 2014, 1:20 am

"Easy to read and more to learn" type of artical.

I like it (Y)

[Link](#)
- sapthagiri June 26, 2014, 5:11 am

What is the difference between vertical scalability and horizontal scalabilty please explain with example.

[Link](#)
- Kenny Fang November 18, 2014, 9:58 am

Both SQL and NoSQL databases have their own advantages and limitations, taking advantage of their strengths would be the way to go.

Generally I would cache the data from SQL database (with those complex queries) into MongoDB (whose table would be huge). Then the Webapp retrieves data from the cached tables or documents in NoSQL DB with the lightening-fast speed.

[Link](#)
- Vin Liyan December 3, 2014, 10:46 pm

Nice Article, easy to read.

[Link](#)
- Srinivas December 30, 2014, 12:46 am

Good one ☺

[Link](#)
- Shiran February 17, 2015, 12:37 am

Good job !

[Link](#)
- MOhan February 25, 2015, 12:13 am

Nice article…..I have Sql db … Just i want to dup this data into redis server for timingly. is it's possible?.. can you suggest me on this…

[Link](#)
- Sliderbahn April 8, 2015, 5:34 pm

Thanks! Great article.

[Link](#)
- seema August 10, 2015, 1:32 pm

Thanks !!!!Nicely described for beginners

[Link](#)
- Muhammad August 20, 2015, 2:03 am

Thanks a lot,
Well done…

[Link](#)

- Mahesh Koli October 24, 2015, 1:03 am

  Very Clearly Stated All points.
  Liked it ,got a concept.
  Thank you.

  [Link](#)

- Ahsan March 14, 2016, 10:53 pm

  Many Thanks for this indeed.
  🙂

  -Ahsan

  [Link](#)

- Harold Scott June 8, 2016, 10:15 am

  Thanks for this article, I understood some parts but not everything, does anybody know of a good exective overview?
  I am just starting to get into programming so the code parts confused me a bit 🙂

  [Link](#)

- Rajat July 10, 2016, 7:08 am

  Many Thanx for the description …

  [Link](#)

- Asif Billa December 13, 2016, 8:08 am

  Nice article.
  It was really helpful.

  [Link](#)

- Dileep December 19, 2016, 2:45 pm

  It is very useful.. Thanks

  [Link](#)

- Rashmi January 16, 2017, 11:56 pm

  Very good article….Thanks!!!

  [Link](#)

Leave a Comment

Name

Email

Website

Comment

Submit

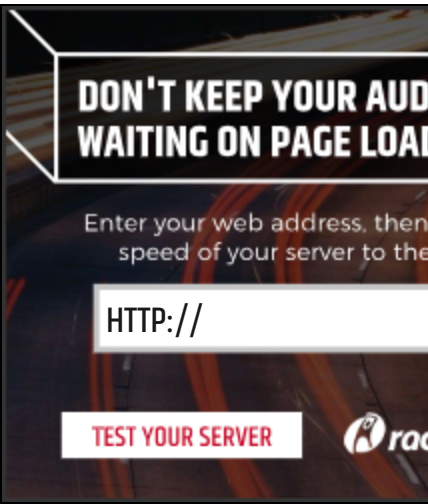☐ Notify me of followup comments via e-mail

Next post: [How to Create an Oracle OVM VM from an Linux ISO DVD Image](#)

Previous post: [How to Use GDB Backtrace to Debug Strack Frame using C Example Code](#)

[RSS](#) | [Email](#) | [Twitter](#) | [Facebook](#) | [Google+](#)

| Custom Search | Search |

EBOOKS

- **Free** [Linux 101 Hacks 2nd Edition eBook](#) - Practical Examples to Build a Strong Foundation in Linux
- [Bash 101 Hacks eBook](#) - Take Control of Your Bash Command Line and Shell Scripting
- [Sed and Awk 101 Hacks eBook](#) - Enhance Your UNIX / Linux Life with Sed and Awk
- [Vim 101 Hacks eBook](#) - Practical Examples for Becoming Fast and Productive in Vim Editor
- [Nagios Core 3 eBook](#) - Monitor Everything, Be Proactive, and Sleep Well

The Geek Stuff
17,415 likes

Like Page   Share

Be the first of your friends to like this

POPULAR POSTS

- [15 Essential Accessories for Your Nikon or Canon DSLR Camera](#)
- [12 Amazing and Essential Linux Books To Enrich Your Brain and Library](#)
- [50 UNIX / Linux Sysadmin Tutorials](#)
- [50 Most Frequently Used UNIX / Linux Commands (With Examples)](#)
- [How To Be Productive and Get Things Done Using GTD](#)
- [30 Things To Do When you are Bored and have a Computer](#)
- [Linux Directory Structure (File System Structure) Explained with Examples](#)
- [Linux Crontab: 15 Awesome Cron Job Examples](#)
- [Get a Grip on the Grep! – 15 Practical Grep Command Examples](#)
- [Unix LS Command: 15 Practical Examples](#)
- [15 Examples To Master Linux Command Line History](#)
- [Top 10 Open Source Bug Tracking System](#)
- [Vi and Vim Macro Tutorial: How To Record and Play](#)
- [Mommy, I found it! -- 15 Practical Linux Find Command Examples](#)
- [15 Awesome Gmail Tips and Tricks](#)
- [15 Awesome Google Search Tips and Tricks](#)
- [RAID 0, RAID 1, RAID 5, RAID 10 Explained with Diagrams](#)
- [Can You Top This? 15 Practical Linux Top Command Examples](#)
- [Top 5 Best System Monitoring Tools](#)
- [Top 5 Best Linux OS Distributions](#)
- [How To Monitor Remote Linux Host using Nagios 3.0](#)
- [Awk Introduction Tutorial – 7 Awk Print Examples](#)
- [How to Backup Linux? 15 rsync Command Examples](#)
- [The Ultimate Wget Download Guide With 15 Awesome Examples](#)
- [Top 5 Best Linux Text Editors](#)
- [Packet Analyzer: 15 TCPDUMP Command Examples](#)
- [The Ultimate Bash Array Tutorial with 15 Examples](#)
- [3 Steps to Perform SSH Login Without Password Using ssh-keygen & ssh-copy-id](#)
- [Unix Sed Tutorial: Advanced Sed Substitution Examples](#)
- [UNIX / Linux: 10 Netstat Command Examples](#)
- [The Ultimate Guide for Creating Strong Passwords](#)
- [6 Steps to Secure Your Home Wireless Network](#)
- [Turbocharge PuTTY with 12 Powerful Add-Ons](#)

CATEGORIES

- [Linux Tutorials](#)
- [Vim Editor](#)
- [Sed Scripting](#)
- [Awk Scripting](#)
- [Bash Shell Scripting](#)
- [Nagios Monitoring](#)
- [OpenSSH](#)
- [IPTables Firewall](#)
- [Apache Web Server](#)
- [MySQL Database](#)
- [Perl Programming](#)
- [Google Tutorials](#)
- [Ubuntu Tutorials](#)
- [PostgreSQL DB](#)
- [Hello World Examples](#)
- [C Programming](#)
- [C++ Programming](#)
- [DELL Server Tutorials](#)
- [Oracle Database](#)
- [VMware Tutorials](#)

Ramesh Natarajan

G+ Follow

About The Geek Stuff

My name is **Ramesh Natarajan**. I will be posting instruction guides, how-to, troubleshooting tips and tricks on Linux, database, hardware, security and web. My focus is to write articles that will either teach you or help you resolve a problem. Read more about Ramesh Natarajan and the blog.

Contact Us

Email Me : Use this Contact Form to get in touch me with your comments, questions or suggestions about this site. You can also simply drop me a line to say hello!.

Follow us on Google+

Follow us on Twitter

Become a fan on Facebook

Support Us

Support this blog by purchasing one of my ebooks.

Bash 101 Hacks eBook

Sed and Awk 101 Hacks eBook

Vim 101 Hacks eBook

Nagios Core 3 eBook