Hayden Vass
28 Aug 2018
Research Notes 1

What is a relational database? What is a non-relational database? What are the pros and cons to each?

A relational database is a collection of information stored in tables and specific columns within those tables. With similar information formatted in a uniform manner, data can be stored, optimized, compared and shared quickly and more efficiently. With relational databases, a database administrator can formulate unique information by gathering details from a particular column and use any number of functions to generate, merge, or compare data from other tables. The ability to store, track and manipulate very specific data makes relational databases a powerful and dynamic tool. Some pros to relational databases include its powerful and easy to understand query language. SQL language has been around for a while and is well known. Not to mention it's easy to learn. Relational databases are also capable of handling large amounts of data in a single query. The SQL language allows the user to tailor their queries to encompass almost all the information they need. Though the pros can be enticing, relational databases do have their down sides. They tend to predefine their data, which makes the extremely rigid. Another down side to relational databases its inability to convert and store custom data types from programming languages.

As a relational database implements a series of tables to store information, a non-relational database does not. Non-relational databases have grown more popular in recent years as the limitations of relational databases become more exposed. While relational databases are great for storing and tracking constant data, they are rigid in nature. It does not excel at capturing big data. Big data is a relatively new science that refers to data that grows quickly and is diverse in nature. Reversely, non-relational databases are typically built for specific data models and can be extremely flexible and scalable. Non-relational databases can use a variety of data models including graph, document and key-value. The pros to using a non-relational data base include its flexible schemas, which allow for faster and more hands on development. They are also have high performance. Since they are often built for a specific data model they can access and distinguish that data at an increased speed. Some disadvantages of non-relational databases usually stem from its relative age. Many non-relative databases have less support and are less mature than some of the older relational databases put out by major corporations.

Popular Relational Databases

<u>Oracle</u>
Oracle is one of, if not the most, popular SQL databases on the market today. It is considered to be one of the industry standards and has an extremely large user base. This also means that it has a large community and support system. Oracle databases are more than capable of containing exponentially large amounts of data and tend to be fast and reliable.
Pros:

- Constant updates, revisions, additions, bug fixes and improvements.

Hayden Vass
28 Aug 2018
Research Notes 1

- Versatile and robust.
- Good at handling large amounts of data with no loss in performance.

Cons:

- Expensive. Cost might not make business sense for smaller companies.
- Requires tremendous amounts of computer resources.

MySQL

MySQL is a somewhat open source database engine. I say somewhat open source because MySQL is open source while MySQL server edition is not. Since the bulk of MySQL is open source, it too provides a huge community that is more than willing to provide support. Compared to Oracle, MySQL is more lightweight and still manages to provide almost all the same utility. While MySQL does not support PLSQL, it does support a variety of programming languages. These languages include Java, PHP, C++, C#, Ruby, and many more. A major downside to MySQL is the drop in performance when storing extremely large amounts of data. While bigger databases such are Oracle are great for huge databases, MySQL is better suited for smaller projects that don't need a lot of memory or resources.

Pros:

- Free
- Extremely versatile for a free program.
- User has a variety of interfaces to choose from.
- Can be implemented with data from DB2 and Oracle.

Cons:

- Not intuitive
- You have to pay for support

Microsoft SQL Server

Microsoft SQL Server is one of the three powerhouse database systems used in corporate IT / tech environments. Like Oracle, it excels at managing large, and often daunting, amounts of data. One of the primary benefits of using MSSQL is its ease to use. When you download MSSQL you also download an entire tool kit with the sole intent of aiding the user in troubleshooting. Almost every stage of development, within MSSQL, has a ton of online support, live support, and documentation.

Pros:

- Fast and reliable.
- Options to adjust resource consumption.
- View data visualizations from mobile platforms.

Cons:

- Like Oracle, its expensive.
- Computer resource consumption can be astronomical.
- Infamous of having troubles importing SQL files.

Hayden Vass
28 Aug 2018
Research Notes 1

PostgreSQL

PostgreSQL is a smaller, open source database. While PostgreSQL doesn't provide the speed of some of the bigger databases. It does excel at joining data from tables with a large amount of information. One unique feature of PostgreSQL is that it supports programming languages and PLSQL. The languages associated with PostgreSQL are Pearl, Python, and Tcl. PostgreSQL is most commonly used for complex data manipulation, and where a lot of tables are being used simultaneously.

Pros:

- Scalable.
- Supports JSON.
- Variety of functions that aid in usability.
- Variety of interfaces.

Cons:

- Ill defined documentation can lead to a harder time trouble shooting.
- Some interfaces tend to be spotty and confusing.
- Performance drops when handling large amounts of data.

DB2

IBM's DB2 is the third of the big corporations top three SQL databases. Much like Oracle and MSSQL, DB2 was developed to be a full feature, high performance database with the express purpose of handling corporation levels of data. DB2 can be used on a variety of operating systems including UNIX, Windows and Linux. One unique feature of DB2 is its "Self-Tuning Memory". This allows the database to change its memory allocation as the work load changes.

Pros:

- Can be hosted from the cloud.
- Can run multiple jobs at once using Task Scheduler.

Cons:

- Expensive.
- Does not work well with outside data.
- Basic support expires after three years.

Popular Non-Relational Databases

MongoDB:

Mongo is probably the most well known and most popular non-relational database on the market at the moment. It utilizes documents as its datatype and stores JSON files, much as relational database stores tables. The strength of Mongo comes from its ability to convert the JSON files and use them inside of an IDE. Mongo is fast and scales well, and is consistent. One of the downsides to Mongo is its limitation on document size, which is currently 16mb. It supports a variety of different programming languages such as Java, Python and C++.

Pros:

Hayden Vass
28 Aug 2018
Research Notes 1

- Document validation
- Intergraded storage engines

Cons:

- Doesn't work well with applications that use complex transactions
- Does not work well with data from older databases.

Neo4J:

Neo4j is a graph based non-relational database written in Java. Though it represents its data in graph format, it stores its data as key value pairs. These pairs are then combined together in a graph. This way of representing and storing data gives Neo4J an edge when working with complex data models containing a lot of unique data between objects. Though Neo4J is great at navigating through data, it does clock in slower than other databases when dealing with singular or simple data request.

Pros:

- Fast queries when defining relationships between nodes.
- Easy to modify and change.

Cons:

- Not intended for task that involve simple relationships.
- Probably requires the user to learn a new query language.
- Not a lot of support available.

Cassandra:

Cassandra is a column based database developed by Apache. Compared to most other non-relational data bases, Cassandra's query language and data is most similar to traditional SQL databases. Instead of storing information in columns, Cassandra stores its data in rows. This allows Cassandra to manipulate data in various rows without having to join them together or requiring other rows to share the same column. Cassandra does exceptionally well storing large amounts of data as well as writing data to the database. It does fall short when it comes to scalability and when dealing with complex data that spans over multiple tables.

Pros:

- Highly scalable.
- Fast write output / decent read input.
- Flexible.

Cons:

- Limited or no use of aggregate functions.
- Unreliable.

Oracle NoSQL:

Much like Neo4J, Oracle NoSQL stores its data in key value pairs, but does not represent its data in graph format. Oracle NoSQL utilizes the idea of a "master" key. This master key links all the children keys together. Much like key value pairs in C# does. As one would expect from a

product put out by Oracle, the support provided to ONoSQL is provided at an enterprise level. It stores and processes large amounts of data with ease, and can process singular pieces of simple data lighting fast. The downside to Oracle NoSQL is its inability to read and write to the data base in an efficient manner. When it comes to this aspect of ONoSQL, its performance is much less of that of Oracles standard SQL database.

Pros:

- Scales well.
- Handles big data extremely efficiently.
- Flexible data models.

Cons:

- Relatively new, so community support is small.
- Along the same thread, not many experts exist making trouble shooting difficult.

Hayden Vass
28 Aug 2018
Research Notes 1

Resources

Garden, H., & Software, C. (2001). *What are relational databases?*. *HowStuffWorks*. Retrieved 28 August 2018, from https://computer.howstuffworks.com/question599.htm

Advantages of Relational Databases. (2018). *Techspirited*. Retrieved 28 August 2018, from https://techspirited.com/advantages-of-relational-databases

Top 5 SQL Databases [Infographic] - DZone Database. (2018). *dzone.com*. Retrieved 28 August 2018, from https://dzone.com/articles/top-5-sql-databases

To SQL or not to SQL. (2017). *Capgemini Engineering*. Retrieved 28 August 2018, from https://capgemini.github.io/design/sql-vs-nosql/

Farrar, G. (2016). *The Advantages and Challenges of using Microsoft SQL Server (MSSQL)*. *Turbonomic*. Retrieved 28 August 2018, from https://turbonomic.com/blog/microsoft-sql-server-advantages-challenges-virtualization-admins-view/?utm_referrer=https%3A%2F%2Fwww.google.com%2F

The Pros and Cons of 8 Popular Databases. (2017). *KeyCDN Blog*. Retrieved 28 August 2018, from https://www.keycdn.com/blog/popular-databases/

What is DB2? - Definition from Techopedia. (2018). *Techopedia.com*. Retrieved 28 August 2018, from https://www.techopedia.com/definition/24360/db2

The Advantages of DB2 | Techwalla.com. (2018). *Techwalla*. Retrieved 28 August 2018, from https://www.techwalla.com/articles/the-advantages-of-db2

What Is A Non Relational Database. (2018). *MongoDB*. Retrieved 28 August 2018, from https://www.mongodb.com/scale/what-is-a-non-relational-database

What is NoSQL? | Nonrelational Databases, Flexible Schema Data Models | AWS. (2018). *Amazon Web Services, Inc.*. Retrieved 28 August 2018, from https://aws.amazon.com/nosql/

Hayden Vass
28 Aug 2018
Research Notes 1

MongoDB vs MySQL Comparison: Which Database is Better?. (2017). *Hacker Noon*. Retrieved 28 August 2018, from https://hackernoon.com/mongodb-vs-mysql-comparison-which-database-is-better-e714b699c38b

source, O. (2018). *Consider the Apache Cassandra database*. *Ibm.com*. Retrieved 28 August 2018, from https://www.ibm.com/developerworks/library/os-apache-cassandra/index.html

10 things you should know about NoSQL databases. (2018). *TechRepublic*. Retrieved 28 August 2018, from https://www.techrepublic.com/blog/10-things/10-things-you-should-know-about-nosql-databases/