MySQL- MySql is a very popular type of a relational database. It is also free and has a lot of features and provides an easy to use interface. It has a great focus on speed and is very reliable.

Pros:

- Free
- Can work with other databases
- Very established (large community and following meaning its easy to find anything you can do with the database because its been tried and tested)
- You are able to process enormous amounts of data

Cons:

- Does not automatically create incremental backups (while some other databases do)
- No built-in support for XML

Oracle- A constantly updating RDBMS (Relational Database Management System) that has been around since the 70's. It is constantly updating and providing users with new tools to accomplish what they need to do.

Pros:

- Practically useful for almost any single thing you can think of that you would need an RDBMS for.
- Almost always has the latest innovations

Cons:

- Can be very expensive
- Requires heavy resources to run. Updates may need to be done before the program functionally runs well.

Microsoft SQL Server- Microsoft SQL Server is a RDBMS that works on cloud-based servers. Microsoft SQL allows you use a number of different editions and in one of the newer versions allows you to track changes made over time to the data.

Pros:

- Microsoft SQL Server is very fast
- Works well with other Microsoft Products

Cons:

- Can be very expensive
- Can have issues importing files

Types of Non-Relational Databases

MongoDB- MongoDB is a type of a non-relational database. It is useful for working with both structured and unstructured data.

Pros:

- Very fast and easy
- Supports JSON
- Horizontally Scalable
- High performance for simple queries
- Can work with structured and unstructured data

Cons:
- Long to setup
- SQL not used as query language

Cassandra- Massive and an open source non-relational database. Has linear scaling and works very fast at any level (even when working with large amounts of data).
Pros:
- Can handle large amounts of information without any problems
- Manages any type of data
- Read and write scalability is supported
- Constant uptime

Cons:
- Unreliable performance (many different jobs are done that are not scheduled by the user. This makes troubleshooting problems a bit harder).
- CQL can get confused for SQL a lot causing a lot of problems when troubleshooting.

Citations

A Brief Introduction to Apache Cassandra. (2016, May 02). Retrieved from https://academy.datastax.com/resources/brief-introduction-apache-cassandra

Arsenault, C. (2017, April 20). The Pros and Cons of 8 Popular Databases. Retrieved from https://www.keycdn.com/blog/popular-databases/

Foote, K. D. (2018, April 10). A Review of Different Database Types: Relational versus Non-Relational. Retrieved from http://www.dataversity.net/review-pros-cons-different-databases-relational-versus-non-relational/

X. (2017, September 28). The SQL vs NoSQL Difference: MySQL vs MongoDB – Xplenty Blog – Medium. Retrieved from https://medium.com/xplenty-blog/the-sql-vs-nosql-difference-mysql-vs-mongodb-32c9980e67b2