

1702330 - Big Data 2

by Daniel Newman

FILE	1623_DANIEL_NEWMAN_1702330_- _BIG_DATA_2_245230_721406091.DOCX (916.98K)		
TIME SUBMITTED	07-MAY-2020 05:35PM (UTC+0100)	WORD COUNT	3146
SUBMISSION ID	126657887	CHARACTER COUNT	15983

Student Name: Daniel Newman

Student Number: 1702330

Module: Big Data

Module Code: ACCA6012

Lecturer: Mike Dacey

Contents

Table of Figures	2
Introduction	3
Key-Value Database	3
What are Key-Value Databases?	3
Benefits	3
Drawbacks	4
Suitable Application	4
Document-Oriented Database	6
What are document-oriented databases	6
Benefits	6
Drawbacks	6
Suitable Application	7
Graph Database	8
Social Media Website application	8
What is a graph database?	8
Why should this website use a graph database?	9
Reflection	9
References	10

Table of Figures

Figure 1 - Key Value Database Diagram	4
Figure 2 - Key Value Database Diagram 2	5
Figure 3 - Key Value Database Diagram 3	5
Figure 4 - Document Orientated Database Diagram	7
Figure 5 - Document Orientated Database Diagram 2	8
Figure 6 - Graph Database Diagram	8

Introduction

NoSQL refers to databases that are not relational. In Big Data it is very difficult to keep a relational database as they are not as easily scalable and tolerant to large traffic or data loads and so these NoSQL systems are useful for ease of development and ease of use. [1] In this report the author will discuss Key Value Databases, Document Orientated Databases and Graph Databases. The benefits and drawbacks of each and when discussing graph databases a case study with regards to social media will be discussed.

Key-Value Database

In this section the author will discuss key value databases bringing attention to the benefits and drawbacks and the idea of hash functions. The author will also provide diagrams of key value databases.

What are Key-Value Databases?

Key-Value databases are in some respects similar to arrays. Redis would be an example of a NoSQL system that uses key value databases. In key value databases the developers would likely create keys that would hold information in namespaces, for example if a university had a key value database then they would want different information given to the lecturers themselves and to a head master, in this scenario they would use different namespaces to access that information. The point of key value databases is to allow for horizontal scaling. Horizontal scaling in this case refers to having a partition key which is the unique identifier of each object and also a sort key and then that keys attributes will come later. For example, a person at a university could be given the partition key of 1702330 but then they will also be given a sort key to say what they are, so using StudentID or if they were a lecturer then LecturerID. These make the information searchable as the sort key will allow the user to sort the information. [2] Key value databases can hold simple data like text and Booleans but also has the capacity to store videos, images and audio if required. [3] Key value databases work best when the information that is stored within it do not need to undergo strenuous queries such as searching for all students who have achieved at least an B grade in each subject. [4]

Another key feature of Key Value Databases are Hash Functions, these allow developers to turn the value of any key into a hash code which can be used to identify it and be used as a key. So the strings are condensed into smaller numbers that can be accessed. Key value databases are usually also using SQL databases alongside them and when using these hash functions the intent is to search in the key-value cache first and if the desired content is not found the hash function can then look in the SQL table. [5]

Benefits

The main benefit of using key value databases is the simplicity of using them, there is no relational aspect to these databases and so no particular format is necessary. When new data is added to these databases they can just be added as a new field entirely (Game1 and Game2 for example) as well as being able to use various datatypes such as audio and video. As a result of the simplicity of these systems developers can use these systems quickly.

In terms of the speed of key value databases, limited information is held within them as older values will be removed over time when new data is entered into the system, similar to a caching system. As a result of this it is very quick to write and read to this database type, once data has been around too

long and removed from this database it can be offloaded to an SQL database. These systems are also scalable as more servers can be added so that they can hold more information over time.

Drawbacks

In order to find information in a key value database the user must know that items specific key otherwise this information could very well be difficult to find after this. Though this is not the case with some key value databases systems such as Redis make it difficult to find these values without the key.

In terms of the values that are stored in these keys, they should be small for the best performance in the future however in the case of key value databases allow video this is not always possible and so performance can be damaged by these larger values. If a user decides to change a small part of a larger value then the system needs to read the entire value before it can edit the specific section and so this may not always be the swiftest method of editing when compared to SQL databases or Document Orientated databases.

Queries are flawed in key value databases as users cannot perform intricate queries such as the example given earlier with students with specific grades, developers would be better off using a different database type if this would be a negative impact on what they hope to achieve. [6]

Suitable Application

A suitable application for key value databases would be a caching system. So when information is added to the database it is first held in a key value database to be accessed quicker as the assumption will be that the newest information is the most relevant at the moment. For example applications such as Facebook and Instagram use a caching system so that when the user accesses the application without internet the cache will display the items held within it for the user and so they will see the most recent items from the last time the user accessed the app. Key value databases are perfect for this type of application as some key value databases actually run off of RAM and in the case of mobile application caches they will store the limited key value data in the phone's storage.

KeyID	Sort Key	Contents		
0001	Book	To Kill a Mockingbird	Harper Lee	
0002	Movie	Frozen 2	Walt Disney	Idna Menzel
0003	Book	The Raven	Edgar Allan Poe	
0004	Movie	The Room	Tommy Wiseu	

Figure 1 - Key Value Database Diagram

The above image is a representation of how a Key Value database looks. There is a Key ID which is the unique identifier, the sort key is not unique but helps for simple searches for this data then finally the content is saved after the sort key in separate attributes.

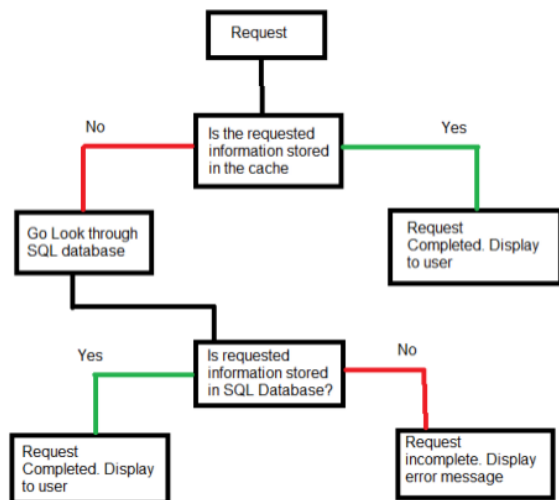


Figure 2 - Key Value Database Diagram 2

SQL databases can still be used in a caching system as when the information no longer needs to be cached they can be offloaded to an SQL database as the first item entered will be the first item out of the cache.

Movies

ID	Name	Direction	Actor(s)
0001	Frozen 2	Walt Disney	
0004	The Room	Tommy Weisau	
0002	Die Hard	John McTiernan	
0008	Spider-Man	Sam Raimi	Tobey Maguire

Books

ID	Name	Author
0003	The Raven	Edgar Allan Poe
0005	Dear Evan Hansen	Steven Levensen
0010	Harry Potter	JK Rowling
0017	Hobbit	J R Tolkien

Figure 3 - Key Value Database Diagram 3

The above SQL database is how these information can be placed using the attributes and ID inside each table and placing them in tables that match their sort key.

Document-Oriented Database

In this section the author will discuss document orientated databases with regards to their benefits and drawbacks whilst also suggesting a suitable application for them with diagrams.

What are document-oriented databases

A document orientated databases are a form of NoSQL database but instead they use an index to connect keys and documents together. What this means is that the unique identifier (keys) are connected to the contents/documents so that data is still searchable. Document databases are unstructured and do not follow the standard database schema. Instead when information is entered to a key it is added to that document. [7]

In document orientated database management document refers to a unit of stored data, in other words they are the object held within a table. So a student's information in a school database would be considered that student's document.

A collection then is another word for what would traditionally be a database table, however a collection has a few distinctions from tables. Since document oriented databases do not have a schema it means that collections can include documents that do not share the same attributes. So for example though it would be perfectly acceptable to have a collection of Student information as the attributes would line up like a normal SQL database, however it would also be perfectly acceptable to create a collection of students and teachers even though traditionally these tables would not be able to interact without the use of a secondary key. [8]

Normalisation in traditional databases refers to avoid using the same data more than once as it could lead to double entry of data where one version of the data is redundant and so these two instances of data could instead be joined through tables using a secondary key so that changing one mention of data will also change the other section. In this database however the opposite is wanted, as a document orientated database is often used to store Big Data it would take a long time for these joins to work properly and so the denormalisation process is required. To do this programmers will instead use the same information in each document they need to include it in. [9]

Benefits

The first benefit to Document Orientated Databases is the simplicity of them. As a result of it being schemaless and so programming these databases becomes easier as each added item is added as its own document and so it does not matter what items are entered into the database and they will allow it to be entered.

When collections of documents are made these collections can be queried easily and quickly and as a result of denormalisation these queries run even faster.

More and more collections and documents can be created easily and so this system would need to be able to accept high amounts of data and so these systems are very scalable so long as enough space is allocated to them.

Document orientated databases are very flexible as new data can be added easily and many varieties of these data can exist concurrently. [10]

Drawbacks

As a result of these databases being very flexible it is very easy to become lazy when developing them and so collections could be created with bad design if too many non-similar things are included in them like mixing books and CDs into the same collection for no reason. These bad design

collections can very easily result in further issues later on like performance worsening by having to sort through useless information in searches.

Though denormalisation has a positive effect on the speed of the database, this also comes with the drawback of data being repeated and in some cases being different from what it should be if it was changed in one place but not another making them inconsistent. [11]

Suitable Application

A good use of document orientated databases would be user profiles and user information. The below example is for student details as they could potentially be kept in a system. If a business used this for user details they could use denormalisation when storing user details and also storing user preferences. If a user entered their information (username and password) to a website then it could look through a user details collection, this way it gets a quicker search for the log in as it looks though only the relevant information to test the details given.

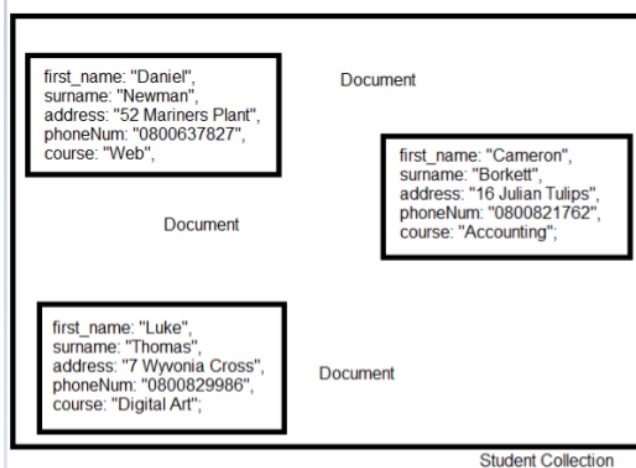


Figure 4 - Document Orientated Database Diagram

In the above example 3 example documents are set up inside a collection. The collection is designed for students and the documents are student data. Additionally if required special dietary information could be added easily due to the lack of schema.

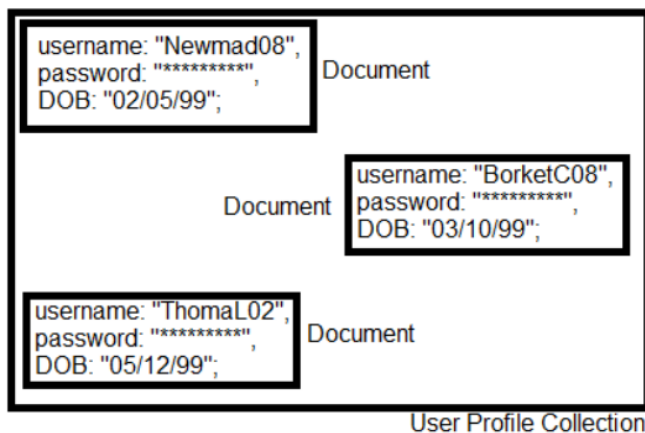


Figure 5 - Document Orientated Database Diagram 2

This example then is for the User Profiles so the users have a username, password and date of birth on the system saved as documents inside a user profile collection.

Graph Database

Social Media Website application

A social media website is in need of a database, the author has been tasked with suggesting a graph database for them to use and how this would benefit them. The website should store user details as well as relationships between them such as a friends list.

What is a graph database?

Graph databases are predicted to become the go-to database within a few years. The main prospect of using graph databases is that the relationship between each piece of information is integral to the system. This is perfect for social media platforms as the social aspect of them really relies on these relationships. So when a data set is entered it is first very void of information until the user starts following pages or adding friends, that original data set then becomes much more valuable as these relationships can be used to suggest new friends or suggest new pages to follow. For example a celebrity on twitter would be one node of information but their followers are all related to them in this system, thousands of nodes being linked to one node, then as a result the links between these nodes have shared interests and so these nodes may get suggested to be linked to other nodes. [12] Nodes in this case refer to the documents of the graph so any user details and such would be stored in a node as attributes these nodes would also contain metadata which refers to the constraints of that node, so information like user preferences such as nodes not to link them with. Relationships refer to the links these nodes have between each other, each connection has a start node and an end node and also an indication of which way that connection is going, in the case of a follow button then that connection only goes one way but in a friends list those connections go both ways. [13] [14]

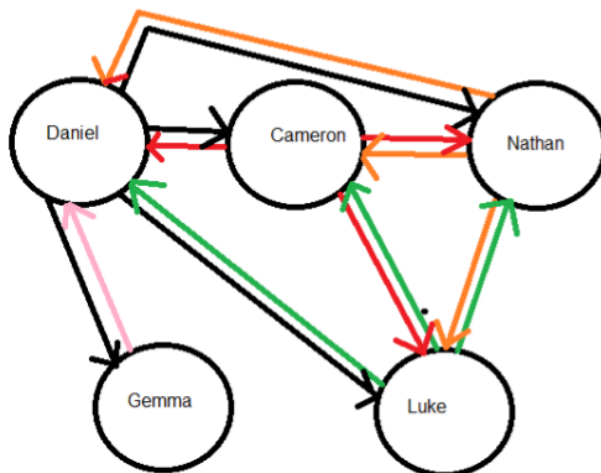


Figure 6 - Graph Database Diagram

The above diagram demonstrates an example friends list with what nodes are connected between friends having a start node, end node and also a clear direction of relation as this group of individuals might not all be friends with each other but some of them are. In one case one user is friends with another who has a friend that they don't have, in social media platforms these accounts would get suggested as "People you might know" so that they could link as friends.

Why should this website use a graph database?

A couple benefits that graph databases offer that will certainly help the social media website are the following.

First is performance, the performance of graph databases are very good. This is because no unrelated data will be searched through in the same way they would in a relational database. Instead if the social media platform wanted to search for a user's friends who are over the age of 20 for whatever reason they could start from that users' node and search each of their related nodes until it compiles a list of each of the desired results. A relational database would search the entire database all at once which if the website had millions of users this would damage performance a lot.

Another advantage graph databases have over relational databases and even over most other NoSQL databases is that a graph database can update at the same time as the database is being updated. This is especially important to a social media platform where people are posting and adding friends all the time. This is possible as queries are only searching relevant nodes and so at the same time any other nodes can be updated since it is not currently being queried. In relational databases since each query must search through the entire table this isn't possible as the update must wait before being able to commit.

These graphs similar to the previous two NoSQL databases are scalable which is very useful to a social media platform in which many users can join at any minute, Facebook for example has 2.6 billion monthly users [15] and if their system was not scalable this would be very difficult to manage and store the data required to keep up with the large data loads. Since graph databases are scalable then it ensures that the social media website can hold the data and know that it is fault tolerant to large traffic and data loads. [16]

Reflection

In this exam replacement assignment the author believes the quality of research is well done and the analysis of the research and of the different database types has been done to a good standard. The author is satisfied with the work they have completed as a whole and believe that the work has been conducted to a good standard.

References

- [1] AWS, "What is NoSQL?," Amazon, [Online]. Available: <https://aws.amazon.com/nosql/>. [Accessed 6 May 2020].
- [2] AWS, "What Is a Key-Value Database?," Amazon, [Online]. Available: <https://aws.amazon.com/nosql/key-value/>. [Accessed 5 May 2020].
- [3] M. Knight, "What is a Key-Value Database?," Dataveristy, 30 September 2017. [Online]. Available: <https://www.dataversity.net/key-value-database/#>. [Accessed 5 May 2020].
- [4] MongoDB, "NoSQL Databases Explained," MongoDB, [Online]. Available: <https://www.mongodb.com/nosql-explained>. [Accessed 5 May 2020].
- [5] GeeksForGeeks, "What are Hash Functions and How to choose a good Hash Function?," [Online]. Available: <https://www.geeksforgeeks.org/what-are-hash-functions-and-how-to-choose-a-good-hash-function/>. [Accessed 5 May 2020].
- [6] M. Dacey, *Key-Value Databases*, Swansea: UWTSD, 2020.
- [7] K. D. Foote, "Fundamentals of Document Databases," Dataversity, 7 November 2019. [Online]. Available: <https://www.dataversity.net/fundamentals-of-document-databases/>. [Accessed 6 May 2020].
- [8] w3, "Databases, documents and collections in MongoDB," w3, 26 February 2020. [Online]. Available: <https://www.w3resource.com/mongodb/databases-documents-collections.php>. [Accessed 6 May 2020].
- [9] "Denormalization in Databases," GeeksForGeeks, [Online]. Available: <https://www.geeksforgeeks.org/denormalization-in-databases/>. [Accessed 6 May 2020].
- [10] "What is a Document Database?," MongoDB, [Online]. Available: <https://www.mongodb.com/document-databases>. [Accessed 6 May 2020].
- [11] M. Dacey, *Intro to Document Databases*, Swansea: UWTSD, 2020.
- [12] K. Cagle, "Graph Databases Go Mainstream," Forbes, 18 July 2019. [Online]. Available: <https://www.forbes.com/sites/cognitiveworld/2019/07/18/graph-databases-go-mainstream/#5f1d8245179d>. [Accessed 7 May 2020].
- [13] "What is a Graph Database?," Neo4J, [Online]. Available: <https://neo4j.com/developer/graph-database/>. [Accessed 7 May 2020].
- [14] AWS, "What Is a Graph Database?," Amazon, [Online]. Available: <https://aws.amazon.com/nosql/graph/>. [Accessed 7 May 2020].
- [15] J. Clement, "Facebook: number of monthly active users worldwide 2008-2020," Statista, 30 April 2020. [Online]. Available: <https://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/>. [Accessed 7 May 2020].

- [16] M. Wu, "What Are the Major Advantages of Using a Graph Database?," DZone, 19 October 2019. [Online]. Available: <https://dzone.com/articles/what-are-the-pros-and-cons-of-using-a-graph-databa>. [Accessed 7 May 2020].

FINAL GRADE

61 / 100

GENERAL COMMENTS

Instructor

KV DB (10): 10
KV Des (20): 12

Doc DB (20): 13
Doc Des (20): 13

Graph DB (20): 13

Feedback

=====

I would have liked a little more detail in the Key-Value database design and application.

Discussion of Doc DB was a lacked a little detail. In the design I would have liked to have seen more analysis of how these documents could be searched and of denormalisation.

For the graph DB you have covered a number of interesting points but the debate of the applicability of this database was a limited.

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11
