



**Prifysgol Cymru**  
Y Drindod Dewi Sant  
**University of Wales**  
Trinity Saint David

University of Wales Trinity Saint David

DVLA: MSc Software Engineering (Web Development)

---

19/01/2022

Assignment 2

ACCA702 – Database Systems

By Rania Abdalla - 2222148

## Table of Contents

|                                      |           |
|--------------------------------------|-----------|
| <b>Introduction .....</b>            | <b>2</b>  |
| <b>1 Types of Data.....</b>          | <b>4</b>  |
| 1.1 Structured Data.....             | 4         |
| 1.2 Unstructured Data.....           | 4         |
| 1.3 Semi-Structured Data .....       | 4         |
| <b>2. What is a Database?.....</b>   | <b>5</b>  |
| 2.1 Relational Databases.....        | 5         |
| 2.2 NoSQL Databases .....            | 5         |
| 2.2.1 Graph .....                    | 5         |
| 2.2.2 Document .....                 | 6         |
| 2.2.3 Key-Value .....                | 7         |
| 2.2.4 Wide Column .....              | 7         |
| <b>3 Use Cases SQL vs NoSQL.....</b> | <b>8</b>  |
| <b>4 Conclusion .....</b>            | <b>9</b>  |
| <b>Bibliography.....</b>             | <b>10</b> |

## Introduction

Data is the foundation of businesses and in technology. It is used in various aspects and allows for better decision making, provides insights and improves customer experiences.

Traditionally a relational database is used to store, manage, and retrieve data. It organises data in a structured way in tables and creates relationships between them and maintains high data integrity. The Structured Query Language (SQL) is used to manage data in relational databases. In today's world data and its usage has increased dramatically. The types of data that needs to be stored also vary, meaning its storage needs to be flexible and accessible. It would become extremely complicated to store varied data in a relational database as it is not structured and would need many tables to determine relationships. Therefore, NoSQL databases were introduced. This report focuses on the four main types of NoSQL databases Graph, Document, Key-Value and Wide column, their use case, and their comparison to the relational database.

# 1 Types of Data

Data is everywhere, all around us from employee details at a company to text images and videos on a webpage where the former is an example of structured data and the latter unstructured data. With the continuous expansion in technology and the use of the internet, more and more data is generated. It is vital that applications have efficient data storage solutions to collect and query this data. [1]

## 1.1 Structured Data

Structured data is stored in a highly organised way, usually presented in rows and columns. It is typically stored in relational databases and can easily be queried using the Structured Query Language (SQL). It is also machine-readable which means it can be easily analysed using software and programming languages. [2]

## 1.2 Unstructured Data

Unstructured data on the other hand does not have a specific format and is not easily analysable. It is typically categorized as qualitative and is best managed by non-relational (NoSQL) database. While structured data is the traditional way of storing data the importance of unstructured data is rapidly increasing. [2]

## 1.3 Semi-Structured Data

Semi-structured data is a type of data that is in between structured and unstructured data. It does not have set ways of storing data but does need to be organised. It is also more difficult to work with than structured data, but it's easier to store than unstructured data. It uses special labels called "metadata" to identify certain parts of the data making it easier to find, organise and study. Examples are JSON, CSV, and XML. [2]

Now that we know about the different types of data let's look at the Databases that are used to store and manage them.

## 2. What is a Database?

A database is a collection of information (or data) stored and read electronically from a computing system. This information is often retrieved, managed, controlled, and arranged to perform data processing operations. [3] There are two types of databases to store data, Relational Databases and NoSQL Databases.

### 2.1 Relational Databases

Relational databases are the traditional databases storing structured data, in tables with rows and columns. Rows contain data that usually have unique identifiers referred to as primary key. They are vertically scalable and can perform complex queries. The Structured Query Language (SQL) is the preferred language used to communicate with Relational databases.

Relational databases which use SQL must conform 4 properties Atomicity, Consistency, Isolation and Durability (ACID).

Though it is the traditional database is not always the best choice for handling many relational data tables as this affects its performance.

### 2.2 NoSQL Databases

Previously data that was stored in databases was very structural and related, like patient records at a hospital or employee details in a company. Nowadays various data is being transmitted and used in applications and via the internet, for example social media and blogs, that use text, images, and videos as well as person details and much more. As this type of data is very large, it is not handled well by SQL databases. Storing such large amount of data is an issue and so is trying to access it quickly and receive accurate results. [4] Hence why NoSQL databases are becoming more popular. NoSQL databases are used for storing unstructured, semi-structured and even structured data. They are highly scalable, handle large amounts of transactions well and have low latency for data retrieval.

There are four main types of NoSQL databases, Document, Graph, Key-Value and Wide Column.

#### 2.2.1 Graph

Graph databases are often used for applications that involve complex relationships and connections, such as in social media. In graph databases data is stored in the form of nodes and edges. [5] Where the nodes represent entities such as a person or a product and the edges represent the relationship between the nodes such as friendships or purchases. These nodes and edges are organised into a graph data structure, which is a collection of interconnected nodes and allows for efficient querying and processing of the data. [5]

Nodes in a graph are assigned a unique identifier and can have properties which are key-value pairs that describe the node. Edges also have unique identifiers and can also have properties.

Graph databases use a specific query language, like Cypher, that allows for querying the data in a more intuitive way by traversing the graph. [5] When a query is executed, the database traverses the graph, starting from one or more specified nodes, and follows the edges to the other nodes. The query can include conditions on the properties of the nodes and edges and can specify the relationships to follow. The result of the query is a set of nodes and edges that match the conditions specified.

Graph databases can also be used to perform complex traversals across the graph, such as finding the shortest path between two nodes, finding all nodes connected to a specific node or finding the most central node in a graph.

They are well suited for use cases that involve modelling relationships between data. For example, it can be used for fraud detection to model the relationship between financial transactions and the entities involved in them. [6]

In summary, a graph database stores data in the form of nodes and edges, it is organised into a graph structure, it allows for traversing the graph and querying the data in an intuitive way and it can perform complex traversals and is well suited for modelling relationships between data.

## 2.2.2 Document

The traditional relational database is not always the best choice as it has limitations when it comes to many relationship tables. It can become quite complex and difficult to manage as the number of tables and relationships between them increase, the number of joins required to retrieve data can also increase, which leads to slow query performance. [7]

It is also difficult to adapt relational databases as they are typically based on a fixed schema, we would not be able to delete or add columns or tables easily. For this reason, NoSQL databases are becoming more in demand. [8]

A document database is another example of a NoSQL database that stores semi-structured collections of data as documents, such as JSON or XML. [7] These documents have fields and nested elements, which allows for a flexible schema that can evolve over time.

Each document in a collection represents a single instance of data, such as a user or product. The documents in a collection can have a different structure, and fields can be added and removed if needed. Document databases use a unique identifier (like a primary key) to identify each document. This key is used to retrieve and amend documents in the database.

Document databases are designed to handle high volume, high performance, and high availability of read and write operations. It is flexible and suited for the world of the internet, and applications with many users that can supply different types of information to store. [7]

A document database allows for efficient storage of each user's specific attributed by keeping only the details that are unique to each user. [9]

In summary, document databases store semi-structured data as JSON or XML in collections. They are adaptable and are well suited for storing large data of different types.

### 2.2.3 Key-Value

Key-value databases store data in, as the name says key and value pairs. A key is used to identify a specific piece of data and the value is the data itself. They are the most flexible type of NoSQL databases and considered the most basic and simple to implement. [10] It is very suited to caches and quick access information due to its lack of structure [10]. The Key-value structure allows for fast lookups as the database only needs to search for the specific key rather than going through all the data.

It efficiently stores data in memory, which allows for faster access to data than disk-based storage. [10] They are at an advantage of speedy operations, as they do not require complex data validation or referential integrity checks which can slow down other databases.

In summary, key-value databases have high performance because of their simple data model, efficient data storage and are optimised for read heavy workloads and minimal overhead.

### 2.2.4 Wide Column

Wide column databases are a type of NoSQL database that store data in tables with rows and columns. The columns in a wide column database can be added or removed dynamically, also the names and format of the columns can vary across rows even within the same table. This allows for more flexibility in terms of data modelling compared to relational databases.

There are 4 features of a wide column database: [11]

A **Row key (RK)**, which has a unique value use to identify a specific record, a **timestamp (ts)**: an integer used to identify a specific version of a data value. A wide column database has at least one **Column Family (CF)**, that has a format of "Family: Qualifier = Value" where "Family" is the name of a CF and "Qualifier" is the name of a **Column Qualifier (CQ)** and value is the real value of the CQ stored in text.

Wide column databases are often used for big data and real-time analytics applications and are often optimized for high write and read throughput. Wide column databases are designed to handle large amounts of structured and semi-structured data and are often used in use cases such as Ecommerce.

In contrast to wide column databases, which allow for dynamic column addition and removal, relational databases use a fixed schema and require the entire table to be altered when adding or removing a column, filling other rows with null or default values. Generally wide column databases are more flexible and better suited to handling large amounts of unstructured data, and high read and write throughput, while relational databases are more suited to handling complex queries and transactions.

### 3 Use Cases SQL vs NoSQL

It is best to use relational databases when working with related data. Relational databases are good for maintaining data integrity because all information is stored in one place and if a user updates a specific record, it is automatically updated in every instance it is used in the database. Examples of SQL databases include but are not limited to MySQL, PostgreSQL, Oracle Database and Microsoft SQL Server [12]

NoSQL is particularly good when wanting to access large data fast. Scalability is a massive advantage of NoSQL as opposed to SQL allowing companies flexibility when it comes to changing requirements. Examples of NoSQL databases include MongoDB, Cassandra and Redis. [12]

## 4 Conclusion

To conclude relational databases also known as SQL databases, store structured data in tables with rows and columns and use a fixed schema. Each row represents an entity, and each column represents an attribute of that entity. SQL databases are better suited to handle complex queries and transactions.

NoSQL databases in contrast do not use a fixed schema and can handle a wide variety of data, including structured, unstructured, and semi-structured data. NoSQL databases are in high demand due to their scalability and ability to manage large volumes of data and process queries quickly.

Both databases have their use cases and advantages and the choice between them depends on the specific requirements of the application or system.

## Bibliography

- [1] P. a. W. P. a. S, "A Comparative Study of Relational Database and Key-Value Database for Big Data Applications," IEEE, Pattaya, Thailand, Mar 2017.
- [2] I. C. Education, "IBM," 29 06 2021. [Online]. Available: <https://www.ibm.com/cloud/blog/structured-vs-unstructured-data>. [Accessed 17 01 2022].
- [3] Oracle, "What is a Database," Oracle, [Online]. Available: <https://www.oracle.com/uk/database/what-is-database/>. [Accessed Jan 2023].
- [4] W. A. B. L. E. A. W Khan, "SQL Database with physical database tuning technique and NoSQL graph database comparisons," *3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC 2019)*, pp. 110-116, 2019.
- [5] Amazon, "What Is a Graph," Amazon, [Online]. Available: <https://aws.amazon.com/nosql/graph/>.
- [6] Oracle, "17 Use Cases for Graph Databases and Graph Analytics," [Online]. Available: <https://www.oracle.com/a/ocom/docs/graph-database-use-cases-ebook.pdf>.
- [7] A. A. a. M. Namoune, "A Method for Transforming Object-relational to Document-oriented Databases," *2020 2nd International Conference on Mathematics and Information Technology*, pp. 154-158, 2020.
- [8] A. S. a. P. Kaur, "Implementing a Multitenant System using a Document-based NoSQL Database," *5th International Conference on Information Systems and Computer Networks (ISCON)*, pp. 1-5, 2021.
- [9] Amazon, "Amazon DocumentDB Developer Guide," [Online]. Available: <https://docs.aws.amazon.com/pdfs/documentdb/latest/developerguide/developerguide.pdf#document-database-use-cases>.
- [10] M. B. a. Y. K. A. E. Alami, "Supply of a key value database redis in-memory by data from a relational database," *19th IEEE Mediterranean Electrotechnical Conference (MELECON)*, pp. 46-51, 2018.
- [11] J. -K. Chen and W. -Z. Lee, "The Transformation of Relational Database to Wide Column Store Database," *2020 International Symposium on Computer, Consumer and Control (IS3C)*, pp. 384-386, 2020.
- [12] B. A. a. B. Nicholsan, "SQL vs NoSQL Databases: Whats the Differnce?," IBM Cloud Databases, 12 06 2022. [Online]. Available: <https://www.ibm.com/cloud/blog/sql-vs-nosql>.