|  |
| --- |
| Advanced Databases |
| CA1 Group Project |
| Section A (Data Mart) and Section B (Report) |

|  |
| --- |
| Cree Gunning, Lee Harold  3/6/2021 |

**Table of Contents**

[1 Section A 2](#_Toc68088830)

[1.1 Industry Topic and Business Area 2](#_Toc68088831)

[2 Section B 3](#_Toc68088832)

[2.1 Analysis and comparison of NoSQL databases with relational databases 3](#_Toc68088833)

[2.1.1 Strengths and Weaknesses 3](#_Toc68088834)

[2.1.2 Future Prospects 4](#_Toc68088835)

[2.2 Motivation for the use of NoSQL databases 4](#_Toc68088836)

[2.2.1 Main characteristics of key-value store databases, column-oriented databases, graph databases and document store databases. 5](#_Toc68088837)

[2.3 How Redis Works (Odd) 6](#_Toc68088838)

[2.3.1 Key Features 6](#_Toc68088839)

[2.4 How Neo4j Works 7](#_Toc68088840)

[2.4.1 Key Features 8](#_Toc68088841)

[2.5 How Cassandra Works (Even) 9](#_Toc68088842)

[2.5.1 Key Features 9](#_Toc68088843)

[2.6 How HBase Works (Even) 11](#_Toc68088844)

[2.6.1 Key Features 11](#_Toc68088845)

[3 Bibliography 11](#_Toc68088846)

# Section A

## Industry Topic and Business Area

**Industry Topic: Sports**

**Business Area: Purchase Orders (PO – Items)**

**1. Business Requirements**

*Shipping Delay:* Helps business as it can see delay time between when a customer places an order with the company and when that order has been shipped. This enables the business to see how long processing time is for each order.

*Highest Orders by County:* This allows the business to see the county where the most orders are coming from. This in turn can be used to decide if there should be more stock allocated to specific branches that are busy around a particular county.

*Average Sales Over Christmas:* Gets average sales over Christmas period. Using a time table, this can be easily checked (October-December).

*Least Popular Brand Sold:* Discovers least popular brand (search for all orders that by brand, count least amount).

**2. Data Mart (Snowflake Schema)**

**3. SQL Script to create Fact table and Dimension tables**

**4. SQL script to load data into the Data Mart**

**5. Queries and Outputs**

# Section B

## Analysis and comparison of NoSQL databases with relational databases

NoSQL is a non-relational DBMS, that does not require a fixed schema, avoids joins, and is easy to scale. NoSQL databases are used for distributed data stores with large data storage. NoSQL is used for big data and real-time web apps such as Facebook, Twitter and Google. (guru99, n.d.)

Relational databases define relationships through tables. SQL is the standard language for relational databases. SQL programming is used for CRUD, optimization and maintenance of databases. Relational databases use SQL to create multiple databases with different tables and attributes. (guru99, n.d.)

### Strengths and Weaknesses

* Relational databases are vertically scalable. NoSQL databases are horizontally scalable.
* Relational databases have predefined schemas. NoSQL databases use dynamic schemas for unstructured data.
* Relational prioritises data validity. NoSQL prioritises speed.
* SQL requires specialized DB hardware for better performance. NoSQL uses commodity hardware.
* Relational is used for dynamic queries. NoSQL is used for scaling.
* Relational databases offer strong consistency. Few NoSQL databases offer strong consistency with others only offering eventual consistency.
* NoSQL offers document-based, graph and key-value databases. Relational is table based only. Each have advantages and disadvantages.

### Future Prospects

NoSQL databases may grow in popularity over relational. Many companies are thinking about or already expanding into NoSQL databases over relational databases.

“According to a data and analytics survey Forrester conducted in 2016, 29% of tech decision makers said they have already implemented NoSQL technology, or are currently implementing it. Another 12% said they were expanding their implementations.” (little\_fish (admin), 2020)

The move towards NoSQL databases over relational is due to scalability, flexible data models, impressive read and write capabilities for demanding customer apps, simplifying data management for any application, and lowering data management costs. (little\_fish (admin), 2020)

It is not certain that NoSQL will become the new norm but it is very promising.

## Motivation for the use of NoSQL databases

* **“The pace of development with NoSQL databases can be much faster than with a SQL database.”**

Development speed is increased as structure of data is controlled by developers increasing efficiency among development with code pushes being more basic alongside modern database development.

NoSQL is better suited across storing all types of databases from completely structured databases down to fully unstructured databases.

* **“The amount of data in many applications cannot be served affordably by a SQL database.”**

NoSQL databases are built for big data, unlike SQL databases. Scalability is cheaper and easier than SQL in large volumes of data.

* **Traffic, downtime and paradigms.**

SQL databases cannot match traffic and zero downtime and scalability associated with NoSQL databases.

NoSQL can be transactional and analytical systems. SQL databases are restricted. NoSQL adapted well to cloud and real-time services.

### Main characteristics of key-value store databases, column-oriented databases, graph databases and document store databases.

**Key-Value Store Databases:**

* Simple key-value method.
* Data stored in key-value pairs.
* Allows high-level horizontal scaling.

**Column-Oriented Databases:**

* Functions similarly to relational databases.
* More flexible than relational databases.
* Data stored in columns instead of rows.

**Graph Databases**

* Relationships equal importance to information within table.
* Nodes, edges and properties to store and represent data.
* Very flexible.

**Document Store Databases**

* Stores data/information using documented-oriented model.
* All data stored within single document.
* Semi-Structured.

## How Redis Works (Odd)

Redis is an open-source, in-memory data store for database, cache, message broker, and queue usage. Redis offers sub-millisecond response times, allowing huge numbers of requests/s for real-time applications within Healthcare, Financial Services, Gaming, and more. Redis is one of the most popular choices for caching, real-time analytics, session management, gaming, leaderboards, etc. (AWS, n.d.)

All Redis data resides in-memory, other databases store data on disk or SSDs. By eliminating the need to access disks/hard drives, this gives in-memory data stores like Redis an advantage over others as it can avoid search time delays and can access data in almost instantly. Redis features versatile data structures, high availability, geospatial and more, making it easier to build real-time internet scale apps. (AWS, n.d.)

### Key Features

**Transactions**

Transactions consist of blocks of commands between MULTI and EXEC (or DISCARD for rollback). Once a MULTI has been found, commands on that connection are queued. When EXEC is found, they are all applied in a single unit If a DISCARD is seen instead of an EXEC, everything is removed. As commands inside the transaction are queued, you can’t make decisions inside the transaction. (StackExchange.Redis, n.d.)

A WATCH command within Redis tells the program that the user is interested in a specific key in the database, used to check different types of data and make the appropriate changes after if needed. UNWATCH forgets the keys when done.

**Recoverability**

Redis supports database recovery if corruption or cluster fails occur. Recovery steps are:

1. If the cluster failed, then you must recover the cluster.

2. Identify your recoverable databases.

3. Restore the database with its previous configuration and data.

4. And finally verify that your databases are active.

(RedisLabs, n.d.)

More detail on their ‘Recovering a failed database’ webpage, and simple to follow. Redis also offers database backup.

**ACID/BASE Properties**

Redis is single threaded, allowing itself to be ACID compliant. NoSQL databases usually don’t provide ACID compliance (using BASE properties instead), or only provide it partially. Having ACID properties is an advantage as the data within remains accurate and reliable.

**Concurrency**

Redis provides I/O level concurrency, using I/O multiplexing mechanisms and event loops. Redis operations are atomic of single-threaded event loops. Atomicity is included at no extra cost. This can be exploited by the user to implement assured locking and other patterns without paying extra synchronization expenses.

**Scalability**

Redis provides options to scale up and/or scale in or out cluster sizes, allowing clusters to easily grow with demands. This advantage provides consistent performance and reliability alongside scalability.

## How Neo4j Works

Neo4j is the most common graph database, implemented in Java. First released in February 2010, Neo4j is now under two types of license: commercial or general public. It is developed by a company called Neo Technology. Companies eBay, Walmart, and more have used Neo4j to improve their services.

Neo4j uses graphs to represent data and the relationships. Several different types of graphs can be used to display this data, such as:

– *Undirected graphs:* nodes and relationships are interchangeable, relationships can be interpreted in any way, e.g., Facebook friends

– *Directed graphs:* nodes and relationships are not bidirectional by default. Twitter is a type of directed graph. A user can follow a user without being followed back.

– *Property graphs:* weighted graph with labels allowing assignment of attributes to both nodes and relationships (e.g., name, age, nationality, etc.). This is the most complex yet most useful graph.

Neo4j uses property graphs to extract added value of data of any company with great performance and in an agile, flexible, and scalable way. (BBVA API MARKET, 2015)

### Key Features

**Transactions**

Neo4j uses DBMS transactions. Transactions while connected to a DBMS will start a DBMS-level transaction. This is a container for transactions. A database transaction has begun when the first query to a specific database is issued. Database transactions opened within a DBMS-level transaction are committed/rolled back when transaction is committed/rolled back.

**Recoverability**

If a data center somehow becomes ‘lost’, for example, you can’t add instances to the current data center, and can only use the current read-only cluster, and then Neo4j has you covered. There are 2 recovery options available depending on your situation; both easy to follow and enable easy database restoration. Neo4j also supports data safety and reliability through backups.

**ACID/BASE Properties**

Neo4j supports full ACID transactions. This ultimately allows the authentication of data to be done to prevent corruption. You can also plan your database around BASE (Basic Availability, Soft-State, Eventual Consistency) properties; however, the ease, reliability and consistency of ACID makes it a better option.

**Concurrency**

Neo4j has concurrency control. It uses locks for transactions which may lead to deadlock. If a deadlock is detected, the transaction is marked for rollback. The transaction may be reinitialized if needed by the user. This decision to retry is left up to the user.

**Scalability**

The database can be scaled without affecting query processing speed and data integrity. This is a huge advantage as adding more nodes and relationships to an existing graph will not hinder or compromise the performance and structure of the database. Neo4j also provides support for replication for data safety and reliability.

## How Cassandra Works (Even)

Cassandra is a NoSQL database. It is a free, open-source database management system (DMS). The DMS is designed and used in the management of large amounts of data across multiple servers. Cassandra is widely used among some of the biggest companies and brands in the world such as Apple, Spotify, Uber, McDonald’s, Microsoft and many more, “Cassandra is used by 40% of the Fortune 100”. (Apache Software Foundation, 2016)

### Key Features

**Types of Data –**

Cassandra supports multiple data types, such as:

* **Built-in Data Types –** Also known as “primitive data types”. Can be directly referred to as they are pre-defined, including but not limited to, Boolean, decimal, double and int.
* **Collection Data Types –** These are for collection of data through storing multiple values in one unit. Cassandra supports three types of collection:
  + **Maps** - storing data with key names for convenience
  + **Sets -** multiple unique values stored, but not in order
  + **Lists** - can store duplicate values and multiple values in specific order.
* **User-Defined Data Types –** Creating own data type around what’s needed. Multiple data fields of any type in one column. Data fields can be changed or removed as needed.

(TutorialsPoint, 2021)

**Query Language – CQL.** Similar to SQL. Data stored in rows and columns within tables it is used within the Cassandra database in creating, inserting and managing data within tables.

**DBaaS (Database as a Service)**. DataStax Astra provides “cloud-native” application development for Cassandra to simplify and encourage application development processing alongside providing automatic backups and cloud storage to make development easier, accessible and scalable for companies.

(DataStax, 2021)

**Cost –** Cassandra is free and open-source.

**Security –** Cassandra provides three types of security:

* **TLS/SSL Encryption** – Secure communication between client and a database cluster and secure communication between nodes within the cluster.
* **Client Authentication** – Accessing a server securely using a password/login.
* **Authorisation** –S etting permission and restrictions for users, similar to an admin or teacher in a school setting.

(Apache Software Foundation, 2016)

## How HBase Works (Even)

HBase is an open-source non-relational database. Written in Java and modelled after Google’s Bigtable.. HBase is column-oriented; attributes are grouped together into column families. This is different from relational databases that are row-oriented, as the columns of a row are all stored together.

### Key Features

**Types of Data –** HBase’s data types are saved in bytes and bits, with data types ranging from a byte of a variable to RawDoubles and RawIntegers.

**Query Language –**HBase is a non-relational data store, HBase “does not support a structured query language. HBase is written in Java.

**DBaaS –** HBase has no adequate DBaaS.

**Cost –** HBase is freeand open-source.

**Security –** Good standard of security, HBase has similar security levels to Cassandra with User Authentication, secure communication and user authorisation through granting and revoke permissions.

# Bibliography

Neo4j

<https://www.tutorialspoint.com/neo4j/neo4j_overview.htm#:~:text=ACID%20properties%20%E2%88%92%20Neo4j%20supports%20full,processing%20speed%20and%20data%20integrity>.

<https://www.bbvaapimarket.com/en/api-world/neo4j-what-graph-database-and-what-it-used/>

<https://neo4j.com/docs/cypher-manual/current/introduction/transactions/>

<https://neo4j.com/docs/operations-manual/current/clustering-advanced/multi-data-center/disaster-recovery/>

Redis

<https://aws.amazon.com/redis/>

<https://dalibornasevic.com/posts/60-redis-in-the-nosql-ecosystem>

<https://stackexchange.github.io/StackExchange.Redis/Transactions.html>

<https://stackoverflow.com/questions/10489298/redis-is-single-threaded-then-how-does-it-do-concurrent-i-o>

Part 1.

<https://www.guru99.com/sql-vs-nosql.html>

https://forum.huawei.com/enterprise/en/sql-vs-nosql-databases-is-nosql-the-future-of-databases/thread/660677-899#:~:text=Relational%20databases%20are%20often%20referred%20to%20as%20SQL%20systems.&text=For%20general%20research%2C%20NoSQL%20databases,to%20their%20speed%20and%20flexibility.