**Relational vs MongoDB/Neo4j**

**Report**

**MySQL**

**Advantages**:

1. ***Data Security*** - data insecurity is global trend in our society nowadays, MySQL brig the gap, as it reliable and secure for storing data. It is crucial for e-commerce to have their data save during and after each transaction.
2. ***On-Demand Scalability* -** It offers unparalleled scalability to speed-up the management of deeply embedded apps using a smaller footprint even in massive warehouses that stack terabytes of data. On-demand flexibility is the greatest feature of MySQL as it allows compete customization to e-commerce businesses with unique database server requirements
3. ***High Performance*** – It meet the most demanding applications while ensuring optimum speed, full text, indexes and unique memory caches for enhances performance
4. **Round- the clock Uptime** – It comes with the assurance of 24 x7 uptime and offers a wide range of high available solution like specialized cluster servers and master/slave replication configuration. Even be an open-source.
5. **Complete Workflow Control** – With the average download and installation time being less than 25 minutes, MySQL means usability from day one. It can also work with multi-platform e.g. Microsoft, Linux, Macintosh and UNIX. It is most popular around the world

**Disadvantages**:

1. ***Hard to scale*** – MySQL is not designed to be scalable, even though, it can be on demand, this need engineering effort to make it possible. MySQL don’t support auto sharding as the nodes are to be maintain manually.
2. **Owned by Oracle** – MySQL is an open source product but now it is acquired by Oracle who have total control of the software.
3. **Not fully compliance to SQL standard** – MySQL is not support standard features and it has some extensions that did not belongs to standard SQL.
4. **Not for large sized data** – MySQL works fine in most small or medium applications, but when data grow the performance reduce.
5. **Stored procedure and trigger is limited** – compare to PostgreSQL, MySQL has a little choice when you need to write stored procedure and trigger. The stored procedure gives you code reuse and encapsulation. But MySQL, you don’t get as many choices as in PostgreSQL

**Examples:**

PayPal, Twitter, Netflix, Spotify, GitHub, Kind Digital.

**Mongos DB**

**Advantages**:

1. **Schema-less** – MongoDB is ideal for a document to store, if have a flexible schema.
2. **Ease of scale-out** - MongoDB can be scale reads by using replica sets. Scale write using shard (auto balancing).
3. **Cost** – MongoDB is free and can run on Linux, ideal for running on cheaper commodity kit.
4. **Document-Oriented**- MongoDB can handle large volumes of structured, semi-structure and unstructured data. It can hold an array
5. **Display format** – MongoDB is clear path to horizontal scalability by using technique called shard to distribute the data across physical partitions to overcome the hardware limitations.

**Disadvantages**:

1. **No support for Transaction** – In MongoDB you need to handle this yourself
2. **Data size** – MongoDB data consumption is generally high due to de-normalization
3. **Less Flexible with query (No join)** – MongoDB don’t have joint on data.
4. **Limited query capability** –
5. **No Patching** – MongoDB is relatively new and there is need for dedicated employees to maintain it which may increase operation overhead.

**Examples:**

Expedia (travel agency) Forbes, Bosch, AstraZeneca, MetLife, Cisco, the guardian,

**Neo4j**

**Advantages**

1. **Performance** - fast queries when you are looking for relationship between nodes
2. **Visual** – it enables to get familiar with data fast and easy, with visual access And exploration tools
3. **Standard** - W3c standards compliant
4. **Domain flexibility** – support a very flexible and fine-grained data model that allows you to model and manage rich domains in an easy and intuitive way.
5. **Relationship** – join tables are transformed into relationships, columns on those table become relationship property.

**Disadvantages**

1. **Vendor -** Limited number of vendors to choose from, and smaller user base, so it is difficult to get support when run into issues.

2**. Difficult in some area** - Harder to do summing queries and max queries efficiently.

3. **No Shard** - All dataset is keep in one server

4. **Not Optimized** - **Graph** database is not optimized for large-volume analytics queries typical of data warehousing

5. Latency – query in a graph is proportional to how much of the graph you choose to explore in a query, and is not proportional to the amount t of data stored

**Examples:**

EBay, NASA, Fortune 50 Retailer, Walmart, Airbnb, Microsoft, Monsanto, HP, Tom-tom, Novartis, Network and IT Operations using graph database.

**General Learning Outcome**

I gain general knowledge about popularly used industrial database – MySQL, MongoDB and Neo4j as each of them have their own advantages and disadvantages depending on the needs and preference to the functionality that designer/user want. Each comes with distinctive functions that suit different circumstances.

I learned how to use MySQL query to create table and simultaneously insert single and or multiple data in the field - row and column of the tables and perform some functions like aggregate, sum, joint table query.

I also gain knowledge about MongoDB as a documented oriented with schema-less, MongoDB uses simple queries, easy and fast integration of data (No ERD diagram), No join nor transactions and build for scalability and designer agility.

I learned how to create database, show collection and drop/update Database Collection, insert and document, perform limited field return in queries and modify documents. And, most logical operator like aggregation, grouping – skips, sort, limits, range. I learned tool called shard that is use for scaling out processes in MongoDB.

I learned what Neo4j is - graph database and Cypher query language (CQL). And, how to create database using Neo4j and single /multiple node and create relationship, attribute among nodes. And using count in descend and ascend order by limiting number of nodes return.

**Conclusion**

I like this module because it gives me general understanding, particularly the skills that is needed in IT industry and as well as very useful idea on how these databases works.

This project enhances my interactive skills as well as in-depth knowledge of how MySQL works and different databases that is used most often in IT sector.

It made me know that there are different kinds of databases and their uses/functionality and improved upon my technical skills and planning and implementation.