Advanced Databases DATA9810: 2022-23

Assignment 2

Lecturer: Brendan Tierney

Student: Ege Bulut – D22124401 – TU059

Part A:

During our presentations on different database technologies, we discussed many aspects that effects people to choose a database technology over another. Before getting into the detailed comparison, we have established that the best database technology is the one that meets the requirements of your purpose. Therefore, the comparison of these databases is mostly describing the different needs in database technology and shows the different directions of evolution of these technologies. A unique technology out of these is the Timescale DB. This technology considers time as a data relation for the database. The categorization and partition of the data are done by their time value, which is unique only to this technology compared to the ones we have discussed. However, also because of this property, there is a lesser amount of relationship between the data considering other possible relations. This means, if the data that needs to be stored and queried is very time sensitive, then this option of technology would be very useful. Another surprising solution for fast writing speed and querying was Cassandra for me. As we were discussing the technology, it seemed very impractical. The CQL it uses cannot do joins, group by, or use foreign keys. It had an unusual way to create, read, and especially, update and delete operations. The data inserted was not organized and the system did not meet the requirements of ACID. However, it was very efficient even with a massive increase in the data because the databases are supposed to be created for the queries that are going to be run. This perspective was especially new to me, and I found it very interesting. It seems like a crude solution, but the results show that this system works well enough to adopt for the fast writing and querying of massive data like messaging systems used by Discord. Another different approach is cloud-based systems. If data extraction is not a big part of the business, Snowflake is a possible technology to choose from. It would enable the upload of data on the cloud and analysis through procedures and functions using SQL. It is a good choice if SQL’s analytical capabilities are required for data analysis or machine learning. However, if what is desired is making efficient use of queries, then Snowflake would no longer be a good choice as it does not offer many possibilities for that. Some of the other solutions that were discussed were Neo4J and Oracle DB. If the data that is going to be stored is interconnected with their relations to each other, then these solutions would be preferred as they offer property and resource description framework graphs. However, if the data is not interconnected and query speed is important, then Neo4J would not be an optimal option. Neo4J offers the ability to write Java code to run procedures, functions, and queries. Oracle DB offers pre-written algorithms for data analysis. These technologies appear to choose an aspect and make it their strength to have the uniqueness to stand out. Another technique we have discussed was in-memory storage to increase the data retrieval speed. However, this is an expensive application since the price for RAM is high and the memory is less persistent as the data is not being written to the hard drive as soon as possible. Some of the technologies allow this technique to be used and it might be a reason to choose those technologies if the advantages are suitable for the purpose. Amazon Redshift was also another product that offered aid for machine learning modules. MariaDB was yet another technology that was discussed by our peers. A comparison between MariaDB and MySQL was made and declared that they share the same standard commands. An additional property of MariaDB is that through changing between different engines, how the data is stored can be changed because each table is stored in a separate file. This is also a storage-consuming application and not efficient to access and read from the files. Moreover, using different engines may cause compatibility issues. However, if the data sets need to be exported, it might be a desirable property. Finally, Amazon Athena is a serverless query engine that was discussed. It is not a database itself; it allows the user to query through databases, which are mostly on other Amazon products. Every query’s efficiency matters for Amazon Athena as the system requires payment per query run. Athena relies on Presto, but it is 60 versions behind to be up to date. Athena does not support indexing and concurring queries are limited, as well as configuration and visibility. However, it supports all sorts of data types. To conclude, database technologies evolve based on the needs of the industry. We have seen some aspects that lead this evolution, such as fast access to the data sets, exporting data sets in certain data types, writing data fast, the information that is held together with the data set, or the analytical requirements to run on the data. The main challenge is to have all of the solutions for these aspects altogether, so the choice of the best technology depends on the need.