Course Reflection

1. What this class is really about

When I started the course, I thought it would be an overall exploration of the challenges involved in managing large, complex databases, with a particular importance on applications based on SQL. This expectation was met, but the course offered much more than what I expected, by providing a great deal of additional knowledge into the various topics of data management. It provided a detailed look at how data management is essential to manage today's information systems, going beyond the basics. The course's complete scope (especially the preparation and the discussion assignments) provided a solid foundation, setting the base for practical learning.

Instead of being a just theoretical study of database strategies, the course went above and beyond that. It felt like a journey that covered the whole data management world. The milestones were structured, allowing me to learn about NoSQL databases. The milestones were very helpful to me, as they have shown me the different ways data can be stored, retrieved, and processed.

It is appreciative that this course combined theoretical knowledge with practical implementation. As every milestone was thoughtfully planned to build upon the one before it, the learning process was very structured. From setting up server instances on AWS to handling complex ETL operations, the course carefully covered every aspect of data management, aligning theoretical concepts with real-world applications. This practical approach helped me improve my educational learning and gave me different skills that I can apply in various practical situations.

A major part of my learning came from NoSQL databases, since I had no knowledge about them prior to this course. Learning about MongoDB and JSON was a new experience, it showed me how adaptable and powerful NoSQL is for dealing with unstructured data. The hands-on experience with these technologies connected the gap between abstract learning concepts and tangible skills, preparing me for the challenges of modern data environments.

To conclude, this course offered much more than what I expected. It helped me understand the broader scope of data management instead of focusing the whole course on advanced SQL queries. The course effectively transitioned from traditional database management to emerging trends in NoSQL databases, leaving me with the knowledge and skills to solve the problems of data management in my future career. This course not only helped me expand my technical skillset but also honed my logical and strategic thinking ability.

2. The Best Part about Participating in Class

The Advanced Data Management course was a mix of various interesting learning approaches. Looking back at the class, there are few things that I can recollect which made the class very interesting and gave me an overall unique academic experience.

Firstly, for me the best part about participating in the class were the milestones. A major part of the course was completing the project milestones and submitting them by Friday. These milestones were not just assignments they were well designed and made us think outside the box to successfully complete them. From setting up the server infrastructure in the first milestone to dealing with complex SQL techniques, each milestone has taught me new lessons and gave me new experiences. Dr. Gomillion made sure that the milestones were interesting by always making sure that the project would align with the real-world scenarios. This practical approach helped me understand the concepts in a much easier way.

Secondly, the perusall videos provided by Dr. Gomillion for each milestone served as an invaluable learning aid. With clear and simple explanation these videos had helped me learn many complex concepts with ease. The websites which were included as a part of the perusall videos also played a crucial role in solving the milestones. Dr. Gomillion had made the videos even more interactive by providing us with a comment section, which has enabled us to ask doubts and learn many new things from our peers as well.

Another highlight of the course were the group presentations that we had to do on various NoSQL databases. Working on these presentations was quite interesting as it took us beyond the scope of the textbook. All the teams had to work on unique and different NoSQL database management systems which were not described in the textbook. This helped everyone to learn about different NoSQL databases in a much easier way which eventually made me understand the range and versatility of different NoSQL systems across different domains.

What made these presentations stand out was the research aspect and the range of databases we looked at. We looked at systems that were designed for various business cases and how each one performed well in its specific field/domain. This exploration gave us valuable insight into how various NoSQL databases can be used in different business and technical contexts.

In conclusion, what really stood out about this course was the combination of milestones, interactive learning materials and group presentations. This diverse approach not only deepened our grasp of data management but also readied us for the complexities and demands of real-world data scenarios.

3. How to select the best persistence layer

Selecting the best persistence layer is very important to maintain the efficiency and functionality of any application or system. There are various factors that one should consider before choosing a persistence layer, such as, scalability needs, access patterns and consistency requirements. By understanding the strengths and limitations of different types of databases, one can choose a database that closely aligns with their projects objectives and constraints.

Files

File based databases store data in CSV, JSON, or XML files. It is recommended to use file-based databases to store any form of simple structured data. Data is stored in either plain text or in structured formats. File based databases can be used for simple non-relational data storage needs especially when data volume is low. File based storage systems are highly efficient for small scale applications and businesses needing to store information such as, configurations or user preferences.

File-based databases I would consider using would be XML Files and JSON Files. XML files would be suitable for scenarios where hierarchical data structures are needed and JSON files would be suitable for storing data in key-value format. Microsoft office uses XML files for data storage.

Relational Databases

Relational databases use SQL to define and manipulate data in a database. They are based on a relational model, where data is stored in tables with rows and columns. These tables are connected through using various keys such as primary and foreign key to enable complex queries and transactions. Relational databases excel in situations where complex transactions, data integrity and consistency are required. They are used across various sectors such as banking and e-commerce where maintaining transactional integrity is crucial.

MySQL and Oracle Database would be two of my choices in choosing a relational database. MySQL is a best choice for web applications. Oracle database is used by various banks such as J.P. Morgan to handle complex transactions.

Key-Value Datastores

Key-Value databases store data in the form of Key-value pairs. They are used for fast data retrieval when the key is known. Key-Value databases are best suited for situations where high performance and scalability is required. Data is stored in simple pairs, with a unique key linked to a value. It is used for session management in web applications, where data is stored in key-value pairs for faster retrieval.

Redis and Amazon DynamoDB are two popular key-value databases. Redis is known for its speed and used for session management in web applications like twitter. Snapchat uses Amazon DynamoDB.

Document Databases

Document databases store data in documents like JSON or XML providing more flexibility in data modelling. These databases are ideal for applications where documents can vary in structure. Document databases are schema less and hence offer flexibility which is very useful for today's rapidly evolving data structures. Data is stored in document format and each document can have a different structure.

MongoDB and Couchbase are two popular document databases which are widely used. MongoDB is popular for its ability to store JSON-like documents and it is also used by eBay for storing catalogs and user data. Couchbase is another great option. Couchbase follows a four-layer hierarchy, Buckets, Scope, Class, and documents. Documents can be compared to actual rows in a relational database.

Column-Family Databases

Column-Family databases store data in columns rather than rows. Column-Family databases are used in cases where data volume is high and efficient read/write is a necessity. Data is stored in columns and each column can have any number of columns associated with a row key. These are commonly used in systems managing time series data.

Two databases I would consider using would be HBase and Apache Cassandra. Cassandra is known for its scalability and availability. Netflix uses Cassandra for managing large volumes of data. HBase is suitable for real-time data processing in large applications.

Graph Databases

Graph databases use graph structure for processing queries. They consist of nodes, edges, and properties. Graph databases are perfect for data with complex relationships. They are widely used in social networks and fraud detection systems. Graph databases are typically used in scenarios where understanding the interconnections between data points is essential.

Amazon Neptune and Neo4j are two commonly used graph databases. NASA uses Neo4j for organizing space mission data. Amazon Neptune is used by Siemens for IoT and network graph solutions.

4. Elevator Pitch

The Advanced Data Management course provided me with a clear understanding of both concepts and practical applications. Throughout the course I learned about various databases including NoSQL options like MongoDB, Cassandra and Neo4j. This gave me the ability to decide which database technology is best suited for a particular application or for a business need.
I became more skilled, in SQL as I learned how to design database structures, create views, indexes, handle queries, and complex transactions. Working on the milestones helped me in acquiring knowledge about the complexities of data security and recovery protocols. Through this experience my understanding of data management has become even stronger, allowing me to provide efficient solutions to the employers.