Data Warehouse Final Paper

In the world of data warehouse, things are ever changing. What was once the standard is no longer relevant. Now when speaking about data warehouses, terms, and systems such as ETL process, IT telemetry, IT technical debt, MongoDB, and Microsoft PowerBI are known across the field. As things are continuing to change, we as developers, engineers, or even some scientists must adjust and change everything.

Over the course of this semester, we have discussed so many aspects of data warehouses and how to create and manage them. During this paper, we will discuss several aspects of data warehouses and how a company can utilize the tools out there in order to help ease the flow of data to power their companies’ data.

To know and understand big data, we need to know what big data is. Oracle defines big data as data that contains greater variety, arriving in increasing volumes and with more velocity. This is also known as the three V’s. Let’s talk volume. The amount of data that is moving matters to a data warehouse. This is talking about high volumes of low density and unstructured data. This could mean using terabytes or petabytes, depending on the size of company that you are working for.

Next, we will discuss velocity. Velocity is measured by the rate of speed at which data is received or acted upon. Depending on what you choose to rate velocity. If we are discussing some of the highest rate of velocity, we think of data streaming straight into memory. Slower velocity data will be written to disk.

The last “V” is variety. This refers to the several types of data that are out there and available at our finger tips. Usually we are used to seeing more of the traditional style of data types that are structured and fit easily into a relational database. But as times are changing. We are seeing data coming in new unstructured and semi structured forms. These two forms include things such as text, audio, and video.

So, how does this help us to determine what is best when it comes to managing our companies data with keeping high telemetry and low technical debt? We now understand what big data is. What it entails. How the three Vs are at the bottom of big data and how they move data from one place to another.

Now we will discuss some of the methods, tools, and techniques that will can use in order to keep our data warehouse working with high telemetry and a low technical debt. This will include when we should use Inmon or Kimball methods. But more than that, we will discuss seven other aspects of working with our data warehouses. Those include data visualization, analytics, online analytical processing, document management, decision services, integrations, and big data integration. By the time we reach the end, we will have a better understanding of what we can do within our positions to help our company’s data warehouses work more efficiently, and more effectively.

With all databases or systems that work with databases, we need to look at an ETL and see what it is. There are three main stages within an ETL process. These stages consist of the following, the extract stage, the transformation stage, and the load stage. In the extraction stage, data is extracted from the source system into the staging area. Within the transformation stage, the data extracted from the source is cleansed and transformed. Lastly, in the loading stage, the data is loaded into the target data warehouse.

Next, let’s look at what IT telemetry is. Telemetry is the automatic measurement and wireless transmission of data from remote sources. It works by having something such as a sensor that measures something like electrical data. Then electronic devices send that data to remote locations for monitoring and analysis. Those who are IT analysts or IT administrators then can use that telemetry to remotely monitor the health, security and performance of different applications and components in real time.

Technical debt is something else that we must understand before we can figure out a way to keep it at bay while we work on a data warehouse. You will find technical debt called a couple of other names such as tech debt or code debt. This happens when a team or an individual delivers a project to their customer that will later need to be refactored. What this means is that they chose speed over perfect code. When this happens, we now have tech debt that we must go and fix. Which then costs the company time and money. Depending on the kind of company you work for, it could cost jobs or even customers. That is why it’s so important to keep tech debt at bay so we don’t end up with nasty results.

Now, let’s go over the seven aspects of data analysis that I have mentioned above. First, we will discuss data visualization. Data visualization is the graphical representation of information and data. Commonly used ways are charts, graphs, and maps. The advantages of using this within our systems are the fact that those who we report to or when we need to give presentations. Everyone can see what is going on in a clearer and more precise way. This also helps while using a source like MongoDB where you can retrieve such information in a fashion that allows ease of understanding but not enough to cause a lot of tech debt while using the system.

Analytics is the next aspect we need to look at. There are so many companies that use analytics for almost all parts of their divisions. For instance, if we look at a company that primarily manufactures cars. That company is going to be using analytics for several different areas. Areas might include research and development, actual manufacturing, sales, and growth. With this, the company might want to use JSON files in order to keep the data simple and use less granularity. They also might want to use NoSQL in order to not have to write a ton of code. Or they might want to use Microsoft SQL Server in order to maintain their data warehouses.

Next is the online analytical processing also known as the OLAP. OLAP also known as an online analytical processing is a method to allow users to selectively extract and query data in or to analyze it form different points of view. They are used in business intelligence. This helps with trends, financial reporting, sales forecasting, budgeting, and planning. With this a key advantage of using an OLAP is that they respond faster to end-user queries. This is due to the OLAP system pre-aggregate data. This means that there aren’t time consuming calculations when an end-user query is processed.

Document management is always important to any company who has a wide variety of data that they need stored and secured so that when someone needs to look at a specific data or document. They know exactly where to go to retrieve it. This is one reason why using something like MongoDB is good because you can go there and keep the data or the code within one location and when you need to find it. It’s right there and easy to access.

Another key factor to any data warehouse and the systems, method, or practices that a company wants to use is their decision services. Why is this important? Because everything out there is based on a decision. This is where we can use things like vertical scalability or horizontal scalability. Both are important to a data warehouse, and both have implications for decision services. Things like an API or XML or Cassandra, or MongoDB are all good choices when looking at decision services. Each of these tools allows the developer to take data and allow the system to analyze it so the company can see different outcomes.

Integration is another important aspect of data warehouses. So, what is data integration? Data integration is a way to consolidate data from different sources into a single dataset so that it can be used in order to help with either business intelligence or analytics. Either way, data integration is important for businesses to keep up with what is going on with their companies.

Just as data integration is important, big data integration is important as well. Now we need to look at what big data integration is. Big data integration is known as the practice of using either people, processes, suppliers, and or technologies collaboratively to either retrieve, reconcile, or make better use of data from disparate sources for decision making. Big data is known by the main three Vs (volume, velocity, and variety). Let’s look at a couple of another Vs to go with them. We have veracity (the reliability of data), variability (data is inconsistent and has to be managed by other sources), value (data must have value in order to be processed. Not all data has value), and visualization (data must be meaningful and a customer must be able to understand it). This is why big data integration is so important to companies. If the companies aren’t keeping up to date with the new technology that is going on around them, then they will either be forced to keep up or they will fall by the wayside.

There are two main types of data warehouses methods that we have learned about this semester. The first is the Inmon and the second is the Kimball. Both have their advantages and disadvantages. Yet both are super effective depending on what your needs are for your company. Let’s take a closer look at these methods and find a conclusion on which one we would use for our company to give us the most output of data but will also leave our tech debt to a minimum.

The Inmon method is a great way to cluster your main entities into one data warehouse and have all of the dependencies as outside sources. For instance, your data warehouse would store your meta data, summary data and your raw data. On the outside you would have your data sources. This would include your operation system, and flat files. On the other side you would have your users. This would include analysis, reporting, and mining. Both the users and data sources would then get retrieved by the data within the warehouse and then be applied where needed.

Some of the advantages of the Inmon method is that data warehouse acts as a unified source of truth. It also allows for low data redundancy. Which will help with ETL process being less susceptible to failure. Something else that is nice is that it allows for greater flexibility as it is easier to update as business needs are required. Lastly, it can handle diverse business reporting requirements.

The Kimball method is a bottom-up approach. This is used in which we would create data marts first to fulfill the business requirements. This is where we would use a star schema in which would be a table with several dimensions. One nice thing about the Kimball method is that it allows the developers to create several star schemas to hold and maintain all the data needed for the company.

There are a few advantages of the Kimball method. One of these includes that it’s a dimensional model in which it is fast to construct as there is no normalization to occur. Which leads to a swift execution of the initial stage. Another advantage is utilizing a star schema. The reason for this is because the star schema is a denormalized structure which simplifies querying and analysis. This allows several users to use the star schema with ease. Something else that is nice about the Kimball method is that it focuses on individual areas and processes rather than focusing on the entire enterprise. Which leads to it taking up less space in the database. Lastly, the Kimball method doesn’t need a huge team to operate and manage it. This is due to the data source systems are stable and the data warehouse is process-oriented.

So, which one is better? It depends on what you need for your company. I would recommend the Kimball method over the Inmon method. Its easier to work with and more straight forward. It allows the developers to create data marts at a fast-moving pace. Allowing more time for the development team to take advantage of looking at new ways to work with the database and data marts.

With all of this in mind. Now we need to figure out what we would want to use as a company that will give us the best flow of data with the least amount of tech debt. For this reason, I would like to start out by stating that the Kimball method is my choice of data warehouses to use. With this, I would use MongoDB as my main database. I also like Docker to build containers. The cloud is a great tool to use as well for databases. Cassandra is a great tool to run documents. JSON is a great way in order to make files that are easy to write and for another user to read.

Cassandra is defined as an open-source NoSQL data storage system. This allows high availability, scalability, and reliability. This allows the transfer of a vast amount of data to spread out across multiple servers without a single point of failure. The difference between Cassandra and a relational database is that one cannot perform joins in Cassandra. Also, Cassandra doesn’t require a primary key or a foreign key within the tables.

The biggest thing that you would need to do if you wanted to implement a snowflake with Cassandra is use a third-party system to do so. This is simply since both of these are different types of databases and there isn’t an easy way to implement it. Utilizing a third-party system would allow you to recreate the Cassandra database so that you could make a snowflake model. With this, the scalability of the database and the speed with be compromised.

Vertical scaling is about adding more resources to the server. It is commonly used in applications and products. You would find it in more of the small or medium-sized companies. With this, companies will find themselves upgrading their server hardware. This includes increasing IOPS, amplifying CPU/RAM capacity, and disk capacity. Some examples of horizontal scaling are Cassandra and MongoDB.

Horizontal scaling on the other hand is found around high availability of services are required. With this, companies add more physical machines to their server or databases. This adds more nodes in the cluster, which reduces the responsibility of each member node by spreading the key space wider and providing more end-points for the client connections. Companies that have high levels of computing or applications or services will use horizontal scaling. The load for the nodes is decreased due to the distribution between separate server nodes. Some examples of vertical scaling are MySQL.

A big question that companies have is, what scaling should we use? Vertical or horizontal? That is a valid question and one that must be determined carefully regarding the costs and the risk to each. With my research and study of each. I would recommend companies to use horizontal scaling. This is due to the fact that horizontal scaling has less down time, has less limits, and has more ability to get the job done without the possibility of the system crashing. Sure, the startup cost of horizontal scaling will be higher, the outcome will be better and well worth it. Where vertical scaling will cost less up front, the limit of space and possible downtime will cost more in the end.

MongoDB is a database but with a twist. It uses a document database with scalability and flexibility for indexing and querying. It utilized JSON-like documents as its stored data types. With this, it’s easy to update and change over time.

So, what does this mean for the database management group? This means that the group will have easy access to it. They will be able to update and change the documents as time moves on. They will be able to update the data structure as well. They can scale it both horizontally and vertically. This is one of the main reasons companies choose MongoDB is for that very purpose. They have a lot more flexibility with it. Not too many challenges should take place once the team understands how it works and how to get the data to go where it needs to go.

That leaves the developer group left to discuss. With the developers, MongoDB is very easy for them to work with. They simply create JSON files with the information that they need and go from there. They will need to understand how it works, which could be one issue that arises. But they will be able to access the information as they would like any other way. They can change the data when they need to or add to it. It’s a nice system to use when you have a lot of data to work with.

The first thing we will cover is how MongoDB uses a schemeless collection. So what is a schemeless collection in MongoDB? This means that MongoDB doesn’t require a rigid, pre-defined schema like a relational database does. The DBMS or database management system, enforces a partial schema as data is written. With this it is explicitly listing collections and indexes.

The next item to cover is how to update data in a MongoDB. With this we can use a single line of code to do so. This code is as follows, db.collection.updateOne(). With this, we can update a single document. To update multiple documents we can use db.collection.updateMany(). To replace a document we can use db.collection.replaceOne(). These are the commands we can use to update a database within MongoDB.

There are a few different things that will happen if we change a field or field value in a document. Since the field or field value is most likely connected to other fields or values. Then those would be affected by this change. Luckily, MongoDB has a plan for that. You can either manually change the connected fields or values or you can create an automated change so that you can get those fields or values to change when any field or value changes.

Lastly, we will cover how all of this can have an impact on the technical debt of the company. Having a MongoDB source flowing into the data warehouse, there is a potential for technical debt. What do I mean? Technical debt is a place within the system where there is a potential for any bugs or fixes which in turn could cause performance issues, more downtime, and slow delivery. So, could having a MongoDB system within another system cause this? Yes, it could. With this, as a company we need to be mindful of what we plan to change within the system of the MongoDB. How often are we planning on changing documents, fields, and/or values within the database. The more often you plan on making changes, the potential for technical debt is higher.

Let’s look at how we would use XSLT in the ETL process. In the ETL process, we would use XSLT to transform XML documents into other XML documents such as HTML. With this, we can see this in web pages and on data dashboards. There are three main advantages to using XSLT. First, XSLT has been designed to work with enterprise XML technologies. Second, XSLT makes most data conversion tasks very easy to understand and implement. Lastly, solutions written using XSLT can be proven to be correct.

Now let’s look to understand the difference between an XML data source and an XML datatype in a relational data source. XML data source is a control in order to understand data-bound controls. An XML datatype is a document that is assigned to an element on the instance using the attribute or through an XML schema. This has the potential to impact an ETL by knowing the hierarchy of the data source and the datatypes.

The last thing we wanted to take a look at was what is XPath and how does it relate to XSLT. XPath is a major element in the XSLT standard. It can be used to navigate through elements and attributes in an XML document. It contains over 200 built-in functions.

Let’s look at XQuery interface with a MarkLogic database. The main reason this is possible is due to XQuery is a type of code that utilizes files. MarkLogic is a server that can take that code and transform it into what the developer designed it to become. That is why XQuery can interface with MarkLogic database.

Second, we are looking at why XQuery is a better option to work with a MarkLogic database. One reason why this is a better option is because the two sides work closely hand in hand. With documents being stored in forests, XML documents are created in tree structures as well. With this, we understand that databases exist as a logical abstraction. So having a MarkLogic database to work with such as XQuery. We see that these two systems work hand in hand to make the data work and allow both the user and the creator to see what they need to see without having too much tech debt.

Lastly, MarkLogic optimizes indexing to support XQuery. One might ask how this is possible. But as we look at different ways to go about it. We see that the XQuery code that is written allows MarkLogic to look at the code and optimize it as it indexes the code. Part of this is how XQuery is written and how MarkLogic reads the code. The other part of this is how less tech debt comes along with these two systems working together. As they work side by side, they can run fast and effectively.

All in all, utilizing all of these methods, tools, database warehouse methods, and systems will allow our company to have the ability to maximize our IT telemetry while minimizing our IT technical debt. With this, we will be able to save money, time, and resources by relying on third party companies to create, write, work, and manage our database warehouses.

References

* <https://www.oracle.com/big-data/what-is-big-data/>
* <https://www.techtarget.com/whatis/definition/telemetry>
* <https://www.productplan.com/glossary/technical-debt/>
* <https://www.tableau.com/learn/articles/data-visualization>
* <https://streamsets.com/learn/data-integration/?gclid=Cj0KCQjwla-hBhD7ARIsAM9tQKtxDPlBJNogPa8sdPgeyJr9aOfi4qn9ToicgirENiGFRg-wK9i2djgaAsYiEALw_wcB>
* <https://www.actian.com/blog/data-integration/big-data-integration/#:~:text=Big%20data%20integration%20is%20the,variability%2C%20value%2C%20and%20visualization>.
* <https://www.astera.com/type/blog/data-warehouse-concepts/>
* <https://www.thoughtspot.com/data-trends/data-modeling/star-schema-vs-snowflake-schema>
* <https://www.educba.com/kimball-methodology/>
* <https://www.astera.com/type/blog/data-warehouse-concepts/>
* <https://ecapitaladvisors.com/blog/why-olap/#:~:text=OLAP%20systems%20respond%20much%20faster,end%2Duser%20query%20is%20processed>.
* <https://byui.instructure.com/courses/221906/pages/vertical-and-horizontal-scaling-and-cassandra>
* <https://www.spiceworks.com/tech/big-data/articles/what-is-cassandra/>
* <https://cassandra.apache.org/doc/latest/cassandra/data_modeling/data_modeling_rdbms.html>
* <https://www.integrate.io/integrations/snowflake/cassandra/>
* <https://community.snowflake.com/s/question/0D50Z00009U8LYcSAN/cassandra-to-snowflake>
* <https://www.trustradius.com/compare-products/apache-cassandra-vs-snowflake#pricing>
* <https://www.esds.co.in/blog/vertical-scaling-horizontal-scaling/#:~:text=Vertical%20scaling%20refers%20to%20adding,small%20and%20middle%2Dsized%20companies>.
* <https://www.mongodb.com/what-is-mongodb>
* <https://www.mongodb.com/unstructured-data/schemaless#:~:text=As%20a%20NoSQL%20database%2C%20MongoDB,explicitly%20listing%20collections%20and%20indexes>.
* <https://www.mongodb.com/docs/mongodb-shell/crud/update/>
* <https://www.etl-tools.com/wiki/knowledgebase/etl_and_xslt>
* <https://avato.co/developers/advantages-xslt/>
* <https://www.ibm.com/docs/en/db2/11.5?topic=overview-comparison-xml-relational-models>
* <https://www.w3schools.com/xml/xpath_intro.asp#:~:text=XPath%20is%20a%20major%20element,over%20200%20built%2Din%20functions>
* <https://developer.marklogic.com/learn/technical-overview/>
* <https://docs.marklogic.com/10.0/guide/concepts/indexing>
* <https://datafortune.com/what-are-the-benefits-of-etl-for-your-company/>
* chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://iopscience.iop.org/article/10.1088/1757-899X/128/1/012020/pdf
* <https://www.guru99.com/etl-extract-load-process.html#:~:text=In%20the%20first%20step%20extraction,step%20of%20the%20ETL%20process>.
* <https://streamsets.com/learn/data-integration/?gclid=Cj0KCQiA8aOeBhCWARIsANRFrQGhhWb9KPiIDbLGsA2VQ4UczXf6cZ-Xg2z4kDRQWrPlvKMFeEnwyaMaAl46EALw_wcB>
* <https://www.altexsoft.com/blog/datascience/who-is-etl-developer-role-description-process-breakdown-responsibilities-and-skills/>
* <https://www.snowflake.com/guides/what-etl>