



Project 3: Big Data and Data Management  
Sebastian Huynh  
Professor Ahmed Azam  
December 7, 2023

## Table of Contents

Cover Page.....	1
Table of Contents.....	2
Statement of Academic Honesty.....	3
Introduction.....	4
Article Response.....	5-6
Management Response.....	7-13
Analysis and Findings.....	14
Lessons Learned.....	14
Conclusion.....	14
Cited References.....	15

### Statement of Academic Honesty

My name is: Sebastian Huynh, I declare that, except where fully referenced, no aspect of this project has been copied from any other source. I understand that any act of Academic Dishonesty such as plagiarism or collusion may result in serious offense and punishments. I promise not to lie about my academic work, to cheat, or to steal the words or ideas of others, nor will I help fellow students to violate the Code of Academic Honesty.

Name: Sebastian Huynh

Date: December 7, 2023

## Introduction

Big data and data management are increasingly interconnected concepts that are utilized by top firms around the world. With are around 328.77 million terabytes of data being generated everyday (Duarte, 2023), businesses are trying to find better ways of gathering, extracting, transforming, storing, and loading large amounts of data while keeping things cost-effective, efficient, and effective. Data itself is changing as well. A large portion of all generated data, around 53.72% (Duarte, 203), is video-based. This means the content and types of data has moved far beyond conventional text-based data and so has data technology. In the past projects we had a deeper look at relational dbms and the power of sql commands. However, we can now take a look at the limitations of rdbms due to it being limited to certain data types, slowing down when dealing with large amounts of data, and how costly it is to upgrade to meet massive demands. Project 3 will explore the emergence of nosql database systems, “not only sql”, and how it compares with relational data technology. We will also see the unique capabilities and objectives nosql databases have in handling big data as well as key theorems associated with maintaining data usability. Some other concepts that are related to data management will be discussed as well including data warehousing and how modern data management provides businesses with many valuable tools such as cloud-based services.

## Article Questions

### *Question #1 - Summarize the paper:*

In “Big Data Management and NoSQL”, the authors give a detailed look into non-relational database technologies responsible for keeping track of the vast amounts of unstructured and semi-structured data being collected. With the rise of IOT, a major source of big data, they mentioned how relational database capabilities are not well suited for this level of data collection speed and volume. Many different nosql database software were listed as examples including HBase, Oracle NoSql, Redis, Cassandra, and MongoDB to name a few. They described how there are more than one ways to organize data depending on the nosql database of choice: with some organizing data by identifying values using a key, by document where it is also identified using a key, by non-indexed graph where values are located with its adjacent position, or even by columns. Nosql being absent of “predefined schemas” is a concept frequently emphasized in the paper’s closing.

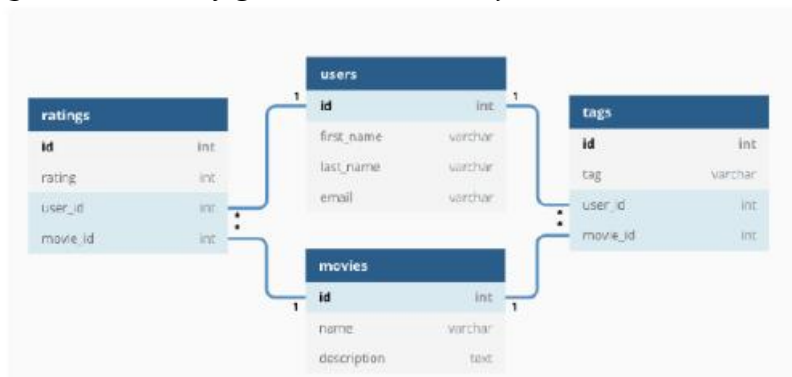
### *Question #2 - What are the main characteristics of NoSQL?*

One of the most important characteristics of nosql is flexibility. It is less restrictive than mysql or mssql, allowing for more “dynamic” changes to existing models of data present in a database. Another Nosql characteristic is its ability to scale outward with additional nodes. Nosql is also based on achieving “consistency, availability, and partition tolerance” of data, this concept is known as the CAP theory. Consistency measures how well data can be accessed simultaneously by multiple users, availability measures how likely a request to the database will be fulfilled, and partition tolerance refers to how well a database system will still operate if one or more nodes in the system fail.

### *Question #3 - Compare SQL and NoSQL*

Both sql and nosql databases seek to manage, organize, and store data but the approach they take in doing so differentiate. Relational databases sought to eliminate the redundancy of data through normalization for the purpose of saving costs related to data storage. Nosql, on the other hand, allows data duplication in exchange for faster processing time and storage of unstructured data like videos, images, and even audio. They both have the ability to scale with nosql being better at this, due to it being more cost efficient to expand by adding additional servers rather than upgrading the power of an individual server needed for sql databases.

### *Question #4 - In figure 1, list all the keys in all tables*



There are 4 tables connected in figure 1. The users table and the movies table contains one primary key each, functioning as the surrogate identifier. Although the ratings table functions

like an associative entity due to having foreign keys from the users and movies table (user\_id, movie\_id) it is considered a normal entity due to having its own surrogate primary key. The tags table has the same structure as the ratings table (tags id, user\_id, movie\_id).

*Question #5 - What are the advantages and disadvantages of relational DB*

One advantage of relational databases includes how it is standardized using sql language. This means that data can be shared between different databases because of such standardization, as long as they follow the same guidelines according to the most up-to-date sql standard then they should be compatible. Another advantage is that rdbms tends to be very secure as it is designed to meet the ACID properties and triggers can be utilized to prevent human errors. Keeping transactions consistent and secure by making sure only transactions where all steps have been completed cause change to the database and that successful transactions still take effect even if there are system problems (Kaur, 2023) are all addressed with ACID. As seen, there are several dimensions relational databases excel in, however they have some drawbacks as well. With the rise of big data, rdbms has trouble storing novice data types like documents or other “conventional” forms of data. It also uses a lot of processing power and can slow down when running complex queries on large amounts of data.

*Question #6 - List and define the Characteristics of big data*

The major characteristics defining big data are the 10 v's (Firican, 2017). Nosql database technology attempts to handle:

1. Volume - large amounts, quantities of data at a given time.
2. Velocity - the rapid rate of speed that data is being created and collected.
3. Variety - wide diversity in the types of data to be stored.
4. Variability - problems associated with any data “inconsistencies” affecting the speed of collecting data successfully.
5. Veracity - source of data determines how much trust we have of a particular piece of data
6. Validity - even if the data is trusted, is the data in a form that can be used accurately for its purpose.
7. Vulnerability - sensitive data needs more security measures to be in place in case of data breach.
8. Volatility - lifespan of collected data, how long can data be stored until it becomes no longer useful.
9. Visualization - complex data can be difficult to represent graphically in a way that can be understood by just looking at it
10. Value - data must provide some sort of value to a business if not then the data serves no purpose.

*Question #7 - What is MongoDB?*

MongoDB atlas is a nosql database software that is oriented around storing documents like “json, bson, and xml” files. There are other document store types similar to MongoDB such as CouchDB and Couchbase. According to the article, MongoDB excels in partitioning which means that it can still function very smoothly even if some nodes happen to fail. Although MongoDB does not use sql, it has a find function that works similarly.

## Upper Management Questions

*Question #8 - Prepare a report that evaluates possible client/server solutions to handle new customer application system for 6 branch offices. What technological characteristic will you evaluate?*

Preface and technological characteristic of interest:

Servers are the machines or software that provide the service requested by the clients which are other machines or programs. For multiple offices this would entail many computers communicating with one or more servers. However, there are many possible ways to configure client-server relationships. Every client-server system consists of creating a process based on three main components. These components range from presentation logic, processing logic, and storage logic and are distributed depending on the number of tiers or divisions present in the structure of a client-server system (Hoffer, 2018).

Objective/Goal for Possible Scenario A:

Allow customers to submit an application through kiosks located at physical office locations connecting the six office branches. Submissions should be treated like transactions and return errors if required fields are left empty or contain incorrect data. Incomplete or incorrect submissions should not be processed or effect the database according to ACID principles.

I. Solution based on two-tier client server system:

- a) It is more effective to use a thin client setup, so that the servers can be the one doing the heavy lifting (processing and data logic) instead of the kiosk machine. We want the kiosk machine to have simple gui and be as fast responding as possible for a better interactive customer experience. If we had placed the processing role onto the kiosks it would require them to have more powerful hardware in order to run smoothly, expensive hardware that would not be safe around the constant use of customers. Then the kiosks can be connected to cloud-based servers provided by the likes of aws or azure for example. This is a less costly setup than thick clients but has a drawback being heavy dependency on network connectivity (Fernandez, 2023).

Objective/Goal for Possible Scenario B:

Allow customers to submit an application remotely online by a main website connecting the six office branches. Submissions should be treated like transactions and return errors if required fields are left empty or contain incorrect data OR if there is poor internet connection and submission stops midway through processing. Incomplete or incorrect submissions should not be processed or effect the database according to ACID principles.

II. Solution based on three-tier client server system:

- b) Similar to the thin client setup, this system will be remote based and over the web. The “browser or smartphone app” (Hoffer, 2018) will be the thin client while the web+application (middleware, middle tier) servers facilitate the exchange between the user accessing the website page and the database servers. For example, a JSP application program will send back a visible “form” for the user to fill out once the application page is clicked upon. This is also another way of saying that there is a need for dynamic pages on the offices’ website for this to work.

*Question 9 - What is an Index? Create an alphabetical index on the customer name in the Customer Table. (ref: project 2)*

An index is a useful tool that can be implemented in a database to help speed up queries. Instead of having database software scan through entire tables before reaching its destination, it simply jumps to the correct index position where the wanted data is stored in the table. It basically functions like the order that you would see on a file cabinet, where it is much easier to find someone's record if you look for the first letter in that person's name and search for the matching alphabet section indexed by alphabetical order for example. Using data from project 2, we can run a sql query to create an index in a-z order.

```
SQLQuery2.sql - (I...CK6AGK\crota (63))*
CREATE INDEX az_name ON dbo.customers (customer_first_name, customer_last_name);
```

Running this code in SSMS, creates a multi-column alphabetical index for the customers table for customer first and last name's in ascending order. The name of this created index is "az\_name". The program should now be able to search automatically using the index but we can explicitly make the query run with the index using the "with" clause (Eder, 2011).

```
SQLQuery2.sql - (I...CK6AGK\crota (63))*
SELECT customer_first_name, customer_last_name FROM customers WITH(INDEX(az_name));
```

100 %

Results Messages Client Statistics

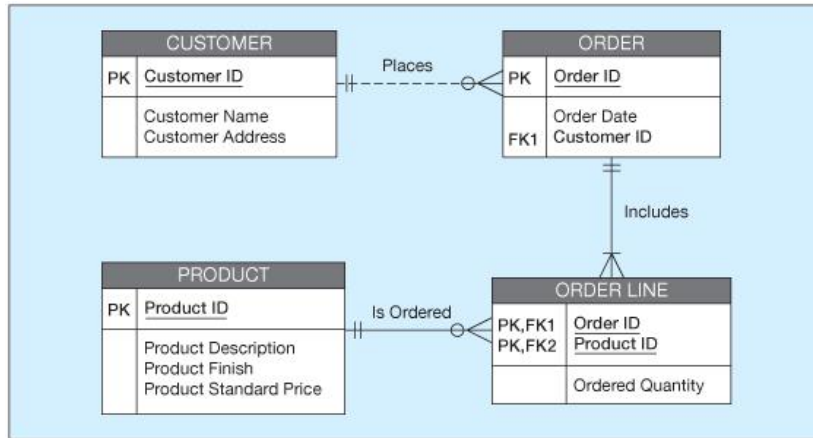
	customer_first_name	customer_last_name
1	Ahmed	Azam
2	Anders	Rohansen
3	Ania	Irvin
4	Dakota	Baylee
5	Deborah	Davis
6	Derek	Chaddick
7	Erick	Kaleigh
8	Gonzalo	Keeton
9	Johnathon	Millerton
10	Julian	Carson
11	Justin	Javen
12	Kaitlin	Hostlery
13	Karina	Lacy
14	Kelsey	Eulalia
15	Kendall	Mayte
16	Kirsten	Story
17	Korah	Blanca
18	Kurt	Nickalus
19	Kyle	Marissa
20	Marvin	Quintin
21	Mikayla	Davis
22	Rashad	Holbrooke
23	Samuel	Jacobsen
24	Thalia	Neftaly
25	Trisha	Anum
26	Yash	Randall

Using a select statement shows that the results have been found in alphabetical order.



*Question 10 - What are the data elements necessary to create an invoice for a customer? Create and save this query as a view named Invoice View (MSSQL)*

Page 180 of our textbook demonstrates a sample customer invoice for Pine Valley Furniture. The main components of this invoice were the relationships between three entities: customer, order, and product. An invoice shows a customer details about their order in case they need to contact customer support with order issues or concerns, and it should show what products were part of the order. Mimicking the attributes given from the sample invoice as depicted below with a few additions, I chose the fields: customer first and last name, customer address, order date, shipped date, order quantity, item title, item price, and all the surrogate keys.



P.185, Hoffer

Creating the invoice using mssql:

```

SQLQuery5.sql - (\\...CK6AGK\crota (53))* customer_invoice_v...CK6AGK\crota (51))
CREATE VIEW customer_invoice
AS SELECT customers.customer_id, customers.customer_first_name, customers.customer_last_name,
customers.customer_address, orders.order_id, orders.order_date, orders.shipped_date,
order_details.order_qty, items.item_id, items.title, items.unit_price
FROM customers
INNER JOIN orders ON customers.customer_id = orders.customer_id
INNER JOIN order_details ON orders.order_id = order_details.order_id
INNER JOIN items ON order_details.item_id = items.item_id;
    
```

100 %

Messages

Commands completed successfully.

Completion time: 2023-12-07T00:01:57.4335794-08:00

The query above joins four tables based on matching records and is saved as a view being named “customer\_invoice”, they are being joined on the common id fields shared between them.

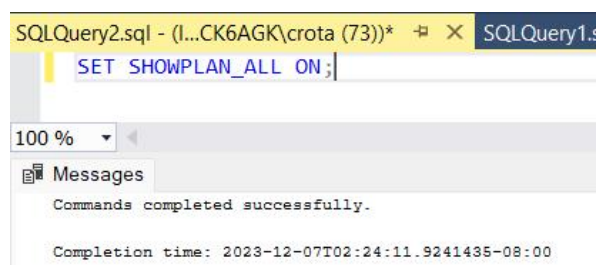
```

SQLQuery1.sql - (\\...CK6AGK\crota (53))*
SELECT * FROM dbo.customer_invoice
WHERE shipped_date IS NOT NULL;
    
```

This query shows the invoices for all completed orders (ones where orders are already shipped).

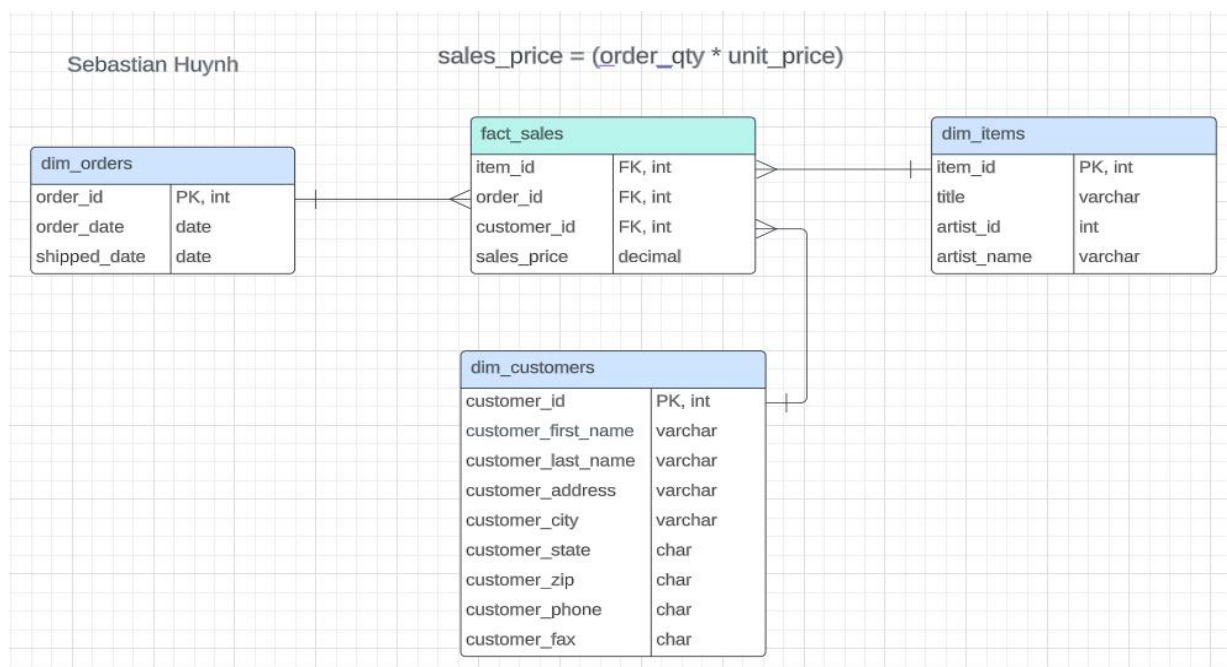
*Question 11 - What is a query optimizer in SQL? How is it different than SQL Query? Give an example.*

A sql query is the actual requests that we command the database to fulfill, like filtering out data that fits the criteria and conditions in a given sql statement. On the other hand, a query optimizer is the “module” or part of a database management system that analyzes the structure of data present and “physical database” (Hoffer, 2018) specs and makes decision on how to use resources to process the steps of a query more efficiently. More efficiency means the downtime of sql query processing is drastically reduced. There is a possible drawback, where an experienced DBMS designer may be able to find quicker processing paths than the optimizer because they understand how to creatively implement business rules into the design of a system. Parallel processing of a sql query can also be managed by the optimizer where multiple server resources are distributed as needed according to the optimizer’s decisions.



For example like the image shared above, the `showplan_all` statement acts as an explain function allowing us to see information about each step in a sql statement (Microsoft, 2023). It shows us how we should execute the statement and what kind of processing or storage “resources” are needed to complete each statement step.

*Question 12 - Create a star schema for your Project 2.*



This star schema I have developed in lucidcharts is good for describing the sales process. The facts table contains one quantitative field (sales\_price) which is derived and calculated from order quantity multiplied by item price per unit. This is the aggregated price we are actually charging the customer. Another key thing to note is that the items table has been denormalized to include descriptive attributes of the artist associated with an item. Overall, the three dimension tables contain only qualitative/descriptive attributes that are important in the sales process.

*Question 13 - How can you benefit from the concepts of data warehousing and DataMart? Is it visible to use it for your project, and why?*

Data warehousing and data mart concepts are very important in the field of business intelligence and for business operations. They are important for analytics because warehousing is concerned with storing and organizing historical data that a business can use to make predictive models and use statistical methods upon, like finding correlation and linear regressions, in order to make better decisions. Using software like a DSS, short for a decisions support system, managers can make better decisions in a relatively short amount of time. However, for a DSS to be useful it needs to have access to data from both the present and past. Some business tasks that rely on DSS assistance includes management of inventory, sales projection, and meeting industry specific needs (Fairlie, 2023). From the previous project the data recorded is historical because it was recorded at an undefined period of time in the past, so it should be loaded into a data warehouse where it can be fed into “end-user presentation tools” (Hoffer, 2018). It is important to note that a data mart is similar to a data warehouse but much more narrowly focused to quickly serve a specific purpose as it carries data from an overreaching, much larger, and centralized data warehouse like an EDW. Also, data marts and data warehouses are focused upon providing historical data, a different type of storage is needed called an operational data store for providing real-time data.

*Question 14 - What do you think about the quality of the data in your projects 1 and 2?*

I believe the quality of data in projects 1 and 2 are fairly satisfactory as the process of normalization up to third normal form prevents potential data anomalies from incurring when using sql queries that update, insert into, or delete records from the tables. Establishing relational integrity through the use of primary and foreign keys also keep data consistent across the dbms.

*How do you measure the data quality?*





The quality of data is measured by different characteristics and according to the textbook, there are about eight characteristics that are very important. Quality data can be described as:

1. Timely - the data is ready-available on time when a business needs to use, if data is not timely then businesses will not be able to react accordingly to changes in events.
2. Unique - there should not be duplicate records in the same table which means redundancy and possibility of inconsistency if one of the duplicates has something altered than the other and we cannot tell which is the correct one.
3. Complete - it should not be null if specified that it needs to be an actual value and includes having all the necessary parts if the value is being computed.
4. Current - the data is as recent as other data is in the database to prevent variability caused by some data being much older or recent then the rest of the database.

5. Conforming - follows important data definitions like specified data type and other planned-out metadata.
6. Accurate - it actually reflects what it is supposed to in reality. Data that is not exaggerate or downplay how things actually are.
7. Consistent - the data is the same no matter how many different locations it happens to be in.
8. Referentially Strong - should comply with any relationship ties with other data

### *How do you calculate the cost of data?*

Data-related spending breaks down into four areas.

				
	1. Data sourcing	2. Data architecture	3. Data governance	4. Data consumption
Description	Cost associated with procuring data from customers, <sup>1</sup> 3rd-party vendors, etc	Cost associated with data infrastructure (procuring software, hardware) and data engineering (building and maintaining infrastructure)	Cost of data-quality monitoring, remediation, and maintaining data-governance artifacts (eg, data dictionary, data lineage)	Cost associated with data analysis and report generation (including spending on data access and cleanup)
Components	3rd-party data	Labor, infrastructure, and software	Labor, software	Labor, software
Typical owner of spend	Head of business unit	CIO	Chief data officer	Head of function or business unit
Typical spend, % of IT spend	5–25 <sup>2</sup>	8–15	2.5–7.5	5–10
Example for a midsize financial institution, <sup>3</sup> \$ million	70–100	90–120	20–50	60–90

<sup>1</sup>Excludes internal data-capture processes.

<sup>2</sup>Industries that don't directly touch consumers (eg, consumer packaged goods) spend a higher share (>20%) on data sourcing.

<sup>3</sup>For midsize organizations with revenues of \$5 billion to \$10 billion and operating expenses of \$4 billion to \$6 billion. Absolute values vary by industry and size of the organization; eg, absolute spend is, on average, higher for the telecommunications industry.

McKinsey  
& Company

(Grande, 2020)

The cost of data is not from the data itself but due to building the capacity to manage it properly. This includes the costs it takes to collect and source data from cloud-based vendors like Google through the use of APIs. According to Mckinsey Company, up to a quarter of IT spending goes to data sourcing. The other costs include planning and implementing a “data architecture” that will be capable of making sure data is clean, compliant, and secure (Grande, 2020).

### *What are your recommendations to improve the quality of data?*

I believe that having a well-thought out plan and building a solid foundation when it comes to the initial design and structure of a data system will lead to higher quality of data. Also, the sources of data must be trusted and there needs to be protocols and staff in place to ensure data integrity.

### *Question 15 - Since there are many recovery techniques available, explain which of the recovery techniques is most appropriate for your project, and why*

Being able to recover lost or damaged data is important even for personal use as in the case of previous projects. The most likely causes of for potential failure on a personal computer would be either system failure or database destruction due to hard-drive failure. One of the recovery methods is loading a backup/mirrored copy of the database which is the simplest way for no need of extra reprocessing. Dbms software usually has checkpoints where it will take “images” of whatever was changed and save that to log records so if you lose your work it will be up caught up to the most recent checkpoint. This works when a backup database copy is not

up-to-date, thus these more recent “images” rolled forward and added to that outdated backup up until it has reached the latest checkpoint.

*Question 16 - What is cloud computing, and is it viable to use for your project? Why?*

Cloud computing is the term used for using anything that has been made available remotely through the use of online-based means, cloud is a term for the “virtual space” (Frankenfield, 2023) these activities end up creating. These are called cloud computing services and can range from infrastructure as a service, platform as a service, or software as a service (Frankenfield, 2023). For example, a company does not need to deal with having unnecessary costs related to adding servers for periods of heavy usage during the year only to have these servers be waste of space the rest of the year or even the cost of building a data-center from scratch. Instead, companies can use “infrastructure as a service” like aws, where they can pay only for cloud-based servers (“lent out” by amazon) actually used at the demand at a given point in time. I used cloud computing “software as a service” like gmail and word online to create my project report and submission of my project over the internet. For my project, the way I could incorporate the use of iaas is by connecting my database to a cloud server instead of having it stored in my personal computer under a local server as it is now. This would add the feature-ability that I could access my specific project database from any internet-connected device.

## Analysis and Findings

There is no single, best way when it comes to designing and implementing a database management system. In order to do so successfully, the database designer must first understand what a firm is trying to achieve with its data while also making sure to design in such a way so that business rules clearly define the framework. Quality data results only from quality data management through careful consideration and therefore keeping database systems up to standards like acid and cap make sure data is being secure, available, and useful for any type of client or business. There are many ways to streamline a dbms, like implementing a sql optimizer or using indexes, that can shorten processing times without causing adverse effects. All of these findings are important for a firm's competitive advantage in predicting outcomes and making better decisions.

## Lessons Learned

1. Star schemas are dimensional modelling constructs that differ from ERD data models in various ways. It is more than just creating relationships between different entities, a star schema allows you to build a map for a particular business process or subject (fact table) you are trying to describe using the dimensional tables. Star schemas are also limited that the dimensional tables cannot be further branched out like in ERD or a snowflake schema (Smallcombe, 2019).
2. Vertical scaling is more costly than horizontal scaling. I used to think there was just one process of scaling but after some research it makes sense that upgrading a single server to be more powerful will be more expensive than adding less powerful but additional servers. This is because when a server has reached its limit and is old then many components need to be changed out to higher-grade components which are much more expensive then buying the alternative equivalent of many lower-grade components with the same specs spread across many servers.
3. There are more dimensions to big data then just the big 3 (volume, velocity, and variety). For example, veracity is one that sticks out to me because the source of data must be trusted in order for us to trust any kind of analytical processing we do to the data. Also, internal data having higher veracity due to it being maintained within giving more assurance.

## Conclusion

Data is not going away anytime soon and is going to be more prevalent as time goes by. By learning the history and future of data technology, business intelligence majors can be sure to provide high value to any company or client they service. Data and information is now being shared quicker and quicker on a global scale and data-solution companies such as Alteryx and Databricks are developing software that even small businesses can use to improve and keep track of their daily operations. The future of data is bright and there are many opportunities ahead as new technologies such as artificial intelligence and automation continue to emerge.

## Cited References

1. Duarte, F. (2023, April 3). Amount of data created daily (2023). Exploding Topics.  
<https://explodingtopics.com/blog/data-generated-per-day>
2. Eder, L. (2011, July 6). *How to use index in select statement?*. Stack Overflow.  
<https://stackoverflow.com/questions/6593765/how-to-use-index-in-select-statement>
3. Fairlie, M. (2023, February 21). *What are decision support systems?*. business.com.  
<https://www.business.com/articles/decision-support-systems-dss-applications-and-uses/>
4. Fernandez, R. (2023, August 8). *What is a thin client? complete guide*. ServerWatch.  
<https://www.serverwatch.com/virtualization/thin-client/>
5. Firican, G. (2017, February 8). *The 10 vs of Big Data*. Transforming Data with Intelligence.  
<https://tdwi.org/articles/2017/02/08/10-vs-of-big-data.aspx>
6. Frankenfield, J. (2023, April 5). *What is cloud computing? pros and cons of different types of services*. Investopedia. <https://www.investopedia.com/terms/c/cloud-computing.asp>
7. Grande, D., Machado, J., Petzold, B., & Roth, M. (2020, July 31). *Reducing data costs without jeopardizing growth*. McKinsey & Company.  
<https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/reducing-data-costs-without-jeopardizing-growth>
8. Hoffer, J., Venkataraman, R., & Topi, H. (2018). Modern Database Management (13th ed.). Pearson Education (US). <https://online.vitalsource.com/books/9780134792293>
9. Kaur, A. (2023, April 21). *Acid properties in DBMS*. GeeksforGeeks.  
<https://www.geeksforgeeks.org/acid-properties-in-dbms/#>
10. Microsoft. (2023, March 10). *Set showplan\_all (transact-SQL) - SQL server*. SQL Server | Microsoft Learn. <https://learn.microsoft.com/en-us/sql/t-sql/statements/set-showplan-all-transact-sql?view=sql-server-ver16>
11. Smallcombe, M. (2019, August 29). *Star schema vs snowflake schema: 5 key differences*. Integrate.io. <https://www.integrate.io/blog/snowflake-schemas-vs-star-schemas-what-are-they-and-how-are-they-different/>