Introduction

Hello, and welcome to Ultimate SQL for Data Analytics! Nowadays, data are EVERYWHERE yet few of us understand exactly how to effectively establish the systems to answer complex questions through thorough analyses. After being exposed to the history of data as well as best practices of modern data collection and retrieval, we will come to appreciate organizational and structured processes that allow for extreme learning for any individual or many companies — both large and small—across the planet. This book aims to expose you to the full breadth of relational theory, database design, and the Structured Query Language (SQL) to prepare you for independence as an elite analyst or data scientist. That is because data processing is an overlooked, but intrinsic, facet of all successful people, organizations, and societies. I love this topic, and I firmly believe that your journey through this material will prove to be a very rewarding experience. I will be with you every step of the way!

By the end of this chapter, you will (a) understand the progress advanced societies have made with organized learning from the days of oral storytelling thousands of years ago, to manual paper-based systems all the way up to modern times with computerized databases; (b) realize the limitations of data by itself and the need for context to create information as well as the systems that enable extensive knowledge and wisdom; (c) define the purpose of the systems development life cycle (SDLC) and the difference between the framework and various methodologies; (d) recognize the similarity between the human need to learn how to make nearly everything more efficient (through innovation and optimization) and the development of a detailed framework like the SDLC in addition to hundreds of specific methodologies to implement it; and (e) think in terms of efficient database design! See how to build systems to process billions (yep…with a 'B') of transactions each day quickly to recognize patterns and opportunities.

Structure

In this chapter, the following topics will be covered:

- Organize or Die: Human Need for being prepared and 'winning'
- Characteristics of learning during the rise of ancient civilizations
- Connecting this theme with modern processes including the Systems Development Lifecycle (SDLC)
- Brief overview of several popular methodologies under the SDLC
- Brief overview of the process for innovation inside repetitive systems
- Historical database systems (paper-based systems and hierarchical databases)
- Flaws and limitations of modern-day spreadsheets

Organize or Die

Organize or Die

Let us agree on several common characteristics across all societies and many generations of people: we are competitive and lazy at the same time. We want to ‘win’ both as individuals for personal prizes and recognition as well as with others aligned in the organizations, clans, or groups of people we identify with. Simultaneously, we are often seeking a better way to get similar or greater output with the same or less effort. This has been proven many times over in the development of ancient civilizations, inventions across industry, as well as in more mundane activities like team sports. While ‘lazy’ is perhaps a dramatic, loaded, and provocative adjective, I use it to make my point memorable. Please feel free to substitute ‘efficient’, ‘productive’, or ‘profitable’ as you see fit.

People have often sought a competitive advantage in most of our repetitive daily activities through innovation of tools and optimization of process. This is perhaps the single greatest quality of being human! The phrase I have been pushing on my students to reflect the urgency of leveraging data to innovate is ‘Organize or Die’. This encapsulates the human characteristic of increasing efficiency, productivity, and profit all the while maintaining or improving quality. Although this may sound daunting, it is fundamentally a process of LEARNING. While this phrase has a potential morbid connotation, it is intended to be inspirational; a roadmap for being assertive in making improvements to any process.

Organize or Die can also find similarity to writings from Charles Darwin and his concept of ‘Natural Selection’ (‘On the Origin of Species’ – 1859). To simplify, Darwinism postulates that each environment has a set of consistent conditions that define it. Compare how a desert in South Africa is different from the frozen tundra of Siberia and each from a dense jungle in Brazil or a coastal island in Western Canada; not only is the temperature, humidity, availability of water, and amount of nitrogen in the soil different, there are perhaps hundreds of other conditions that are also materially different. Therefore, it only makes sense that plants and animals (or flora and fauna if you prefer) are different in each climate. This is due to the traits required to survive demand specialized adaptation over many years.

Furthermore, as conditions change in any environment, there are a handful of characteristics that will determine a higher probability for survival moving forward. Over time, all species must adapt via mutation or willful adoption of competitive practices to continue to propagate and pass along their progeny.

This book is not about human evolution or really anything about plants and animals; we use these examples to get readers to recognize the following:

- Competition for resources (like customers, raw materials, real estate) is fierce
- Not everyone “wins” or gains control of a limited resource; we must learn HOW to position ourselves and our organizations to be better suited and prepared for the conditions that determine success in obtaining control of the resources we desire.
- Business and market conditions will change over time; that means we must re-learn how to win
- New competitive characteristics will prevail as market conditions evolve
- Data has been leveraged to determine competitive actions forever
- The modern economy relies heavily on data to get ahead on new opportunities
- ‘Organize or Die’ is knowing when and how to adapt as market conditions change

It can be argued our common interests are to control the factors in our lives that allow us to have success. Stripped down to its most basic elements, individuals and our collective societies have two simple goals: 1) *produce* the goods or services that enable us to provide for ourselves and our closest associates/family (or corporate stakeholders), and 2) increase the probability we all *survive* and continue to prosper. Recognize please that these also align with the motivations for organizations seeking knowledge from data scientists.

This is a strong parallel between ancient human development and survival through exceptionally harsh environments with current corporations of today; each had to compete for limited resources, adapt to uncaring conditions, and reaction to adversity by INVENTING tools on the fly as problems arose. We can learn about human behavior and how we might be better able to compete in the current economy if we understand the success metrics of our ancestors and incorporate their collective resilience, and attitudes of seeking efficiencies for nearly every process. Moreover, they did not have modern computer systems to collect, clean, and present data for easy analyses; most of the discoveries for survival tens of thousands of years ago were mostly the result of sheer necessity and experimentation trying to make crude processes better.

Produce (short-term/immediate needs)

The first motivation of most ventures, from an ancient clan of folks on their way to building a society or a modern-day corporation, are defining and meeting short-term or immediate obligations. We cannot plan beyond tomorrow afternoon if we are struggling to feed ourselves (or in modern times, find customers) and our fundamental needs are unmet!

Studies indicate that upwards of perhaps 90% of start-ups FAIL in their initial year due to not being able to meet short-term production or revenue demands. Likewise, out of the hundreds of millions of early humans who ever existed, consider how many burgeoning societies failed to take hold due to not being able to establish production of food sources that were reliable and sustainable. Perhaps these earlier people failed to design adequate permanent shelter or develop processes that allowed for extended storage of goods through non-growing seasons, rough weather, infestations or other disruptions to agriculture. The reasons for failure are many (and all unfortunate) but the result was the same; those that were not able to build systems or processes for immediate basic needs perished. We can learn from this even today.

Survive and Prosper (long-term/continuous success)

Assuming a burgeoning ancient community was part of the miracle 5% who were able to accommodate their basic needs beyond a generation or two, their long-term survival was far from guaranteed. What did the 7 civilizations who survived for thousands of years do (independently of each other by the way) to find long-term success?

Again, this text is not about anthropology; however, we do need to understand however the characteristics of success that allowed only a few collections of people to have long-term success in the development of human civilizations.

Some of the more prominent characteristics from early civilizations include the following:

- Communication: language both spoken and written
- Organized acceptable codes of behavior (laws)
- Diversified labor skills
- Healthcare specifically as well as general accommodation for the infirmed or elderly, those not directly involved in production of food or shelter
- Formal Education: general knowledge but also specific training on production methodologies
- Communal defense of the entire clan
- Recognition of the need to adapt by LEARNING and incorporating tasks that were effective or more efficient in time/resources

Essentially everyone and each living thing has had this dream of these basic needs; even animals and plants (including plants might be a bit of a stretch). Now, connect the dots here please. Internalize the concept that competition is real and has always been present in each society and helped them innovate and succeed. Those with the specialized skills demanded by the ever-changing conditions of a particular environment will have a distinct advantage over others.

For thousands of years, societies had to compete with primarily manual processes of handwritten notes, oral traditions, and first-hand observations. The scope of their analyses was limited when compared to the systems and processes of modern times that can assess billions of times the quantity of data than before.

It is important to recognize that today's economy rewards organizations (and by extension the individuals) that can learn and act before others. This means being able to become predictive; being able to quickly learn from massive amounts of data to find the critical patterns, trends, outliers, and anomalies that have been determined to be indicative of higher productivity, profit, or prosperity. Essentially, being in control of data is now akin to being in control of our environment and success.

The next few sections are going to explore a bit more into the process of innovation as well as cite examples from history to cement the point that previous generations have always used data effectively even though their processes were rudimentary. While this may not be on your list of interesting topics, in my experience, interviewers tend to respond better to those of us who can relate the tasks we do today with those of previous eras in human history, being able to articulate similarities as well as how the breakthroughs of technology make things easier. If you are in a hurry to get into the meat of database design and coding with SQL, feel free to skip ahead.

Information Ladder: DIKW Pyramid

For decades, there have been numerous academic models defining the differences between data, information, knowledge, and wisdom. I find these conversations interesting as they can provide clarity around finding imperfections within a process of communication as well as helping systems engineers and information architects develop effective reporting dashboards. It benefits anyone who desires a career as a professional working with data as we need to see how it fits as a component within a larger system.

The common components in this information hierarchy represent the acronym DIKW.

- DATA
- INFORMATION
- KNOWLEDGE
- WISDOM

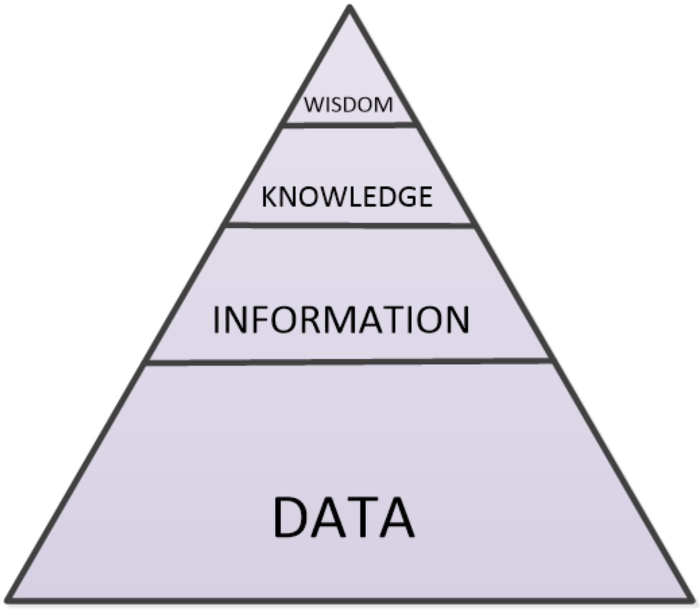


Figure 1.1: DIKW Pyramid

These components are often presented in the form of a hierarchy because there is a sequence of building blocks where subsequent components rely on the previous one. Simply put, wisdom is the result of having broad prior knowledge, and knowledge does not exist without a plethora of appropriate information. Obviously, the whole hierarchy begins with data as the raw bits of fact. Also be aware that proportionally, the pyramid is predominantly covered with the section of data and the others are increasingly smaller by scale. Even the print font gets smaller trying to emphasize this relationship!

The key point here is that most systems in use today have a ton of data and many people using the system do not have an equal proportion of wisdom. The process of learning begins with data but requires adequate documentation and commitment by users to engage the data honestly, paying attention to nuanced indicators, patterns, and trends to gain full awareness. This is not easy!

Understand please also that many people more qualified than me have already defined each of these components (do a search on 'DIKW Pyramid' for more background and perspective). I will share what this means to me in the context of promoting a well-intentioned technology career where people have depth of understanding.

DATA: raw and meaningless

Data are all around us; there must be billions of tiny bits of data we encounter each day (certainly more than we can consume). Everything we hear, see, or feel can be considered tiny pieces of data.

We equate data minute pixels from a high-definition television screen. By themself, we can't effectively envision the larger picture since the individual dots lack perspective and depth found in relation to others. Raw data lacks context of purpose, definition, or even problem space. This missing context means a person cannot take any action because of being presented with raw data.

Think about it; if we happen to be on a city bus (perhaps a light rail train?) and a stranger next to us starts spouting random bits of data (such as 'red', 'A24', 'LAX', 'Martha', 'Tuesday', and 'Argentina'), we may decide to avert our eyes or move away from them. If the nonsensical gibberish continues, we may even choose to exit the ride. Data lacks context or relation to broader concepts, we can't connect the disparate words into a cohesive idea. This may cause frustration and possible anxiety as we struggle to comprehend the intentions of the person in front of us.

Consider how the vast amount of data present in the world might frustrate an organization that lacks professional expertise in data management. How might executives and prominent stakeholders react to a similar presence of 'gibberish' as they seek to solve very real issues? Recognize that data by itself does not provide direction on how to react or which action has a priority.

INFORMATION: data in context; action can be taken

Data that has definition or context allows people to at least consider acting on it. Without context or meaning, we are stuck. Using the raw data elements from above ('red', 'A24', 'LAX', 'Martha', 'Tuesday', and 'Argentina'), let's add context (if only to reduce the anxiety you may have felt trying to figure out what these words might mean!).

Example context could be as simple as, 'My neighbor is Martha; she is arriving at Los Angeles International airport ('LAX') on Tuesday from Argentina at gate A24. She will be wearing a red backpack.' Please note that there is more context around the previously unrelated elements of data, which allows more people to be able to visualize the situation of facts. While you might be less anxious with this information, there may still be a few questions on your mind, such as '*so…did you want me to pick her up? If so, what time is she* *expected to arrive*?' Just because I provided more context in the data does not mean gaps still exist or you know inherently how to react!

KNOWLEDGE: processed information; definite targeted action

To me, knowledge is the accumulation of many pieces of information from all forms, such as from reading, listening during thousands of conversations, and observations from living our lives and engaging others. I will argue that knowledge (and eventually *wisdom*) is earned through diligent work of processing the available information around us. Much of the most valuable knowledge people obtain in our lives is created through purposeful investigation and discovery; it is not guaranteed by being awake a set number of hours.

Knowledge includes understanding the context of how the information is produced, processed, and the purpose that it is presented (alliteration anyone?). This can be phrased as knowing the 'meaning' of the information. From the earlier example, being told that 'Martha is coming into LAX on Tuesday at gate A24' might be for a specific meeting for which I am coordinating. Previous information from conversations, meetings, or news is relevant to my understanding of any expected reactions to the new information. My knowledge of the context of this information aids my decision-making about the situation. As a result, I may check the itinerary for exact details of Martha's flight and create a spot in my calendar for the time it should take for me to be out of the office.

As people gain more knowledge in a specific area or topic, we theoretically become more effective. Having knowledge does not guarantee any particular outcome. It is important to realize that good people, with accurate information, considerable experience and noble intentions can still make honest mistakes!

WISDOM: processed knowledge; experience with consequences of action and expected outcomes

Finally, we get to the last letter of the DIKW acronym: wisdom. As defined by others, the DIKW hierarchy is a pyramid, which means it is narrow at the top of the triangle. I interpret this as meaning that wisdom is difficult to obtain, relatively rare, and therefore valuable. In my 20 plus years of industry experience, I have been engaged in solving problems that were a direct result of gaps in wisdom.

Perhaps the reaction to my knowledge and wisdom of previous trips to LAX is to contact Martha directly and arrange a meeting spot near baggage claim. I will make sure she has my phone number in case things change at the last moment or the flight is delayed. I may check the traffic reports while I am heading to LAX and select the most appropriate temporary parking space based on real-time information. Probably the wisest thing based on previous experience is to set a reminder in my calendar to get an Uber ride for her!

Data in Ancient Societies

Summarizing 40,000 years in a page or two is a fool's errand, so I will apologize at the very beginning. Again, I am not an anthropologist, historian, or sociologist; I did, however, study these topics as a college student many years ago and have always found them interesting. Therefore, as I progressed through my career in technology, I always sought to understand similarities to previous challenges and discoveries in human history if only to feel some sort of camaraderie and affinity to those folks.

Diving a bit deeper into the dominant human trait of being competitive and needing to control our environment and 'win', let's go all the way back to the original civilizations and take a brief look at what each had in common. Each of these civilizations relied on various forms of data to learn and optimize the thousands of processes each faced in their unique setting to ease hardships and burden of the fundamental tasks for survival. This comparison will hopefully not only put in context our basic competitive nature as humans but to also help you appreciate the opportunity to materially contribute to a remarkable legacy of progress established by our ancestors.

Most academics will refer to the original civilizations as being the following (I am aware the labels of 'old world' and 'new world' have more than a slight odor of ethnocentrism):

Old World	**New World**
Mesopotamia (Iraq)	Olmec (Mexico)
Ancient Egypt	Caral-Supe (Peru)
Ancient China	
Indus Valley (India)	

Table 1.1: List of Ancient Civilizations

How these civilizations each succeeded for centuries effectively independently from each other is astonishing. This means they each had to discover ‘better methods’ of doing repetitive tasks as well as controlling their environment with various infrastructure to reduce risks of weather variations and protect from hostile outsiders. These are essentially very much like projects we still embark on in modern times yet on a much simpler scale.

Try to imagine the world and harsh conditions early humans had to cope with; no internet, zero organized manufacturing, or engineering schools. Heck, tools were hand-crafted from raw materials found around camp. Their achievements are remarkable and speak to the brilliance and ingenuity of our species!

Many of the innovations early civilizations developed are considered so basic and fundamental to us nowadays many of us will take them for granted. Do not overlook these innovations as they were the difference between life and death, not only for individuals but for the entire society and emerging civilization. Some of the more prominent developments each original civilization developed independently include the following:

- Language (both written as well as spoken)
- Sedentary settlements
- Prominence near large river floodplains
- Separation of labor
- Administrative structure
- Organizational learning
- Common defense

Language must be the single greatest invention ever. The power of being able to communicate effectively is not only practical for getting tasks completed but also allows for better emotional connection and establishment of clans or common groups. Consider the power of collaboration; being able to coordinate effort significantly reduces duplication of tasks and needless competition between family and affiliated clan members.

Language enabled the quick transfer of knowledge. This means people could begin to understand the wider world beyond their limited lived experience. They were able to gain insight into best practices for basic diet, healthcare, and trade. Additionally, language fostered generational learning of specific skills like farming, woodworking, animal husbandry, and hunting. Younger clan members were able to become artisans much easier by understanding the mistakes and valuable experience of predecessors.

Specialized skills allowed for the transformation of nomadic peoples to become sedentary and establish permanent settlements, usually on or near fertile river floodplains. These flood plains were rich in minerals and produced a range of edible vegetation, reducing the need to forage. Having control of a stable food source by domesticating animals and establishing farming practices thousands of years ago allowed civilizations to further concentrate on the division of labor. It is this division of labor, where individuals were able to become experts in a skill as opposed to a generalist. Experts are the ones who can deconstruct technical processes and determine which steps are inefficient. Understanding flaws and inefficiencies leads to experimentation and innovation and continuous optimization. This is true to this day, however with the aid of computers and artificial intelligence, we can complete a cycle of development, analyses, and optimization much more quickly.

Once these early civilizations became sedentary, developed a surplus of food, and specialized skills, they exploded in population thereby expanding their collective capabilities even more. They each developed other aspects that strengthened their core, including formal administration (creating laws and rules), organized education, and common defense.

Societies that had the characteristics noted above were able to reduce the risks associated with solitary existence and ultimately prospered. This is important to consider in a book focused on data science because it recognizes how each succeeded under harsh conditions based on their ability to learn and innovate every single day. Each civilization captured data, processed it into relevant information, and strived for the creation of a functional body of knowledge of best practices that provided a blueprint for efficiency and success. All of this expanded the collective body of knowledge of each civilization across generations, making each society exceptionally well-suited for their unique challenges (such as geographic location, regional fauna, and weather patterns). Evidence is the world population has exploded 10-fold in just the past 200 years since the beginning of the industrial revolution. These are the same aspirations many organizations seek today!

Systems

So now that we have established that humans love to learn, make things better, and compete to obtain more resources, are we finally ready to dive into database design, SQL, and analytics? Almost! No database or SQL discussion is complete without introducing the idea of how 'systems' tie everything together. Although I have more than 20 years' experience working in industry in addition to having taught a systems analysis class at the University of Washington for 5 years, I am far from being an expert in this area. I will present a simplified interpretation of how systems and associated processes are involved in doing analyses but will also encourage readers to explore online resources for further clarity if so desired.

In brief, a system is an established structure that has a desired outcome, a set of processes that are organized sequentially to achieve that goal. These sequences of events, tasks, steps, are the smaller parts that collectively act as a larger cohesive unit. The purpose of a system is to provide efficiency in a repetitive process whereby the output is consistently higher quality and with less input resources required than if the system did not exist. More sophisticated systems even have built-in steps that are intended to measure progress of the previous steps, validate status, effectiveness of work completed, or otherwise verify that the desired outcome is still viable.

In short, a system organizes a repetitive process to save either time or money (hopefully both!), by breaking down the process into a collection of many smaller tasks or steps. Perhaps the biggest benefit of implementing a system is learning (this is straight from the mantra ‘Organize or Die’).

To illustrate these points, let’s explore some examples of systems, one complex, and the other potentially less so.

Example System/Repetitive Process #1: Farming

First off, farming is not in my area of expertise, although I recall planting carrots and string beans for a class project in perhaps 3rd grade. However, since every society in the world engages in some form of agriculture, it has broad appeal for conveying an example in addition to being representative of a complex system that benefits from having a sequential structure of organized steps.

Desired outcome: produce a crop of some edible vegetables like potatoes, corn, beets, or carrots to feed family and/or sell at a market if there is any surplus

Sequential Steps: 1 - 4

1. Locate available arable land
2. Decide on appropriate crop(s)
3. Throw seeds in the ground
4. Come back in 3 months with a shovel, leather gloves, and a medium-sized bucket

I present farming as facetiously simple (for the time being) to illustrate several points. First, farming is a complex process that has been refined over thousands of years with many processes, sequential steps, dependent tasks, and analytical measurements that are collectively critical to successfully producing viable nutritious items year after year. Many of these, however, are not documented and must be learned through a combination of oral conversations, observation, and lived experience. Second, essentially any sophisticated process like farming can be reduced to an unrealistic and simple form from a distance if the analyst is uninformed, ignorant, or just lazy.

Let’s see another process that may be more familiar to people:

Example System/Repetitive Process #2: Commuting to work

A common repetitive task for many people is traveling to-and-from a place of employment (I am thinking before/after pandemic). Again, as a species, we are programmed to experiment when defining processes, find inefficient steps, innovate, and measure outcomes to obtain ‘better’ results. As such, driving to and from work every day is ripe for analysis and optimization.

Desired outcome: Getting to place of employment in a predictable timeframe efficiently.

Sequential Steps: 1 - 6

1. Find quickest route to commute destination according to online mobile phone app
2. Drive as aggressively as traffic permits without being reckless (speeding, changing lanes mostly)
3. Locate the closest parking lot to destination
4. Try different routes, departure times, and parking lots for maybe a week or two
5. Review/evaluate for time, cost, and overall convenience for maybe 2 weeks
6. Settle into optimal routine with modification only as conditions change (such as traffic volume, charges for tolls, or parking)

Many of us spend an inordinate amount of time and effort during our commute navigating thousands of split-second moments of optimization. Assuming a normal 30-minute commute in each direction, many of us will accumulate hundreds of hours each year (and months of our lives!) engaged in this high-stress activity with insufficient planning.

Questions when evaluating the two example systems of farming and a daily commute:

- Is example #1 (farming) any less complex than example #2 (driving to work) simply because it has fewer steps?
- How does writing down the sequence of steps (and resulting output) for any system affect the probability of successfully hitting our desired output objectives?
- What might happen as these processes mature, and we complete the processes dozens of times?

Addressing these questions may be the first time we have considered systems design, so here are my thoughts:

- *Is example #1 (farming) any less complex than example #2 (driving to work) simply because it has fewer steps?*

While the number of steps might be an indicator to the complexity of any process, farming is incredibly complicated with many factors and conditions that span a growing cycle that is several months long. The number of steps, tasks, and considerations for producing a viable crop probably exceeds 1000 even for the planting of a resilient vegetable like carrots or potatoes. While the inputs for driving during a commute are dynamic and occur multiple times per second, I will argue the complexity of farming far exceeds those of driving. For the sake of argument, anything that the average 17-year-old can do with a drink in one hand and a half-eaten burger in the other *WHILE TEXTING* cannot be that difficult! Side note: While I was a math tutor at Rainier Beach High School in Seattle in the late 1990's, I taught more than 20 of my students how to drive and lived to write about it.

The system of farming often represents dozens of generations of learning over thousands of years and might be one of the most complex, sophisticated, and critical humans have ever developed. Yet, many of us might be tempted to minimize the difficulty of the system followed for producing successful agricultural production. This may be the ultimate instance of 'Organize or Die'.

Let's compare the system for farming to the system many of us follow to get to work each day. I will argue that driving to work has little 'deep' analysis and relies on emotion and impulsive reactions to perception every few seconds.

According to a simple search on the internet, the average duration for Americans is 27 minutes each direction of their daily commute. Since I have had several jobs where the commute often exceeded 60 minutes each way on a good day, for illustrative purposes I propose rounding this number to 30 minutes. We often adopt aggressive driving during a typical commute to save mere minutes; unspoken or realized is that one missed elevator, canceled meeting, or being placed on hold during a phone call negates everything we have saved in our earlier commute.

To fully compare these examples, let's establish several broad characteristics of a typical aggressive driver (not just those in the United States as I thought all the bad drivers in the world had moved to Seattle for some reason, but then I visited Kazakhstan).

The characteristics of many aggressive drivers, which include me occasionally, are all forms of innovation and the innate desire to 'win':

- We will often choose alternate routes to avoid paying tolls
- We often increase risks by switching lanes numerous times to 'win' even just a few car-lengths (number 1 cause of most car accidents is during or right after a driver has made a lane switch)
- We often cut in line 'unfairly' during a merging scenario
- Drive through parking lots or residential neighborhoods to get past slower traffic

The above characteristics do not include other less legal behavior, such as exceeding the speed limit, driving in High-Occupancy Vehicle (HOV) lanes with no passengers, or perhaps running red-lights, tailgating, or passing on the shoulder.

As our commute is not complete until we park and exit our vehicle, the 'winning' behavior includes aggressive parking tactics. I have been guilty of dozens of occurrences of violating the following parking rules (especially after driving around for 20 minutes to play fair):

- double-parking in a turn lane (once…as a famous coffee brand's parking lot was full)
- truck loading zone (many times)
- someone else's reserved parking (not often)
- disabled parking (rarely)

I have even put a cardboard coffee sleeve on my rear-view mirror crafted in the shape of a monthly parking pass to get more free-if-not-noticed time of unattended parking.

If most commuters conducted a deep analysis of how much time, money, and aggravation we invest into getting to work, most of us would soon realize that perhaps the most-efficient behavior is taking the bus!!

- *How does writing down the sequence of steps (and resulting output) for any system affect the probability of successfully hitting our desired output objectives?*

Writing down the sequence of steps is a great way to improve the probability of success. Some of the important goals of most systems are to increase efficiency, improve predictability, and reliability of the outcomes from engaging the process. Many factors will change over the lifetime of a system, such as our knowledge, the capabilities of technology, legal requirements, competitive forces, and perhaps even the weather. This means as we learn more about each process over multiple cycles of the system, we must make UPDATES. These updates are not just limited to what the written or documented steps are; it may also include facilities, equipment, personnel, and training to name just a few. A system is quite often a work in progress. To be most effective, it should represent the best-known steps and reflect current operational knowledge based on evidence.

This is both logical and practical; it not only helps us visualize the process in a cohesive arc, so we reduce the chance of missing a step, but it also allows the transfer of knowledge to someone else unfamiliar with the process. Duh.

In the case of the steps for farming that I included in example #1 above, it is woefully inadequate in the approach to a complex process of farming. At a minimum, additional steps should be added, such as scientific evaluation of the mineral content of the soil, a reasonable amount of fertilization, and validation steps to indicate progress such as weekly measurements and surveys from customers.

- *What might happen as these processes mature and we complete the processes dozens of times?*

We learn!! Again, a system is about improving efficiency, reliability, quality, and predictability. As there is no such thing as a PERFECT system, each cycle of a systems cycle is an opportunity to reflect on the collective steps to assess whether it has any obvious flaws and is still providing value. We know that repetition allows for familiarity and emergence of people with potential expert understanding. The ability for experienced personnel to suggest innovations can be evaluated. Some of society's greatest inventions were born as workarounds, band-aid experiments, and dumb luck guesses in the middle of normalized or well-established processes.

Again, a system can be complex or simple. While recognizing that whatever works for a particular situation is often the rule, please also realize that no system is perfect, and every system has a lifespan of effectiveness. The window might only last several months (such as cleaning up a one-time oil spill) or perhaps centuries (rail transportation systems for example). This means that every system can be improved or optimized somewhere in one of its more mundane components. The only thing is, while there may be a slight inefficiency somewhere, it may not be a sound investment of time, money or other resources to embark on the journey to make any changes. We usually avoid taking on system improvement projects until the benefit of spending resources on the overhaul is significantly greater than the cost of maintaining the way things are.

My point in showing these examples is twofold:

1) people are always innovating or seeking new ways to gain efficiency (often based on impulse or flawed perception)

2) we can almost always find inefficiency within or improve any process by analyzing data.

Like data, systems are all around us in various levels of complexity. The average person probably participates in more than a dozen systems each day before noon, including electrical, plumbing/sewer, transportation, economic, communication, and education systems, just to name a few. Data science is not just one more system, rather it should be viewed as a smaller yet critical component in thousands and thousands of larger more complex systems.

Systems Analysis

No system is perfect! Each system has inefficient steps; the only question is whether the cost of inefficiency with the existing system exceeds the cost of fixing it. Stated another way, after spending time and money to make the steps of a process more efficient (through training, adding more people, adopting existing technology into the system, or by innovating something completely new) do we get our money back through greater competitive advantages in the future? Every generation, civilization, and organization that has been successful has benefited from analyzing their systems of production and making improvements. It is again tied back to human nature and our need to be efficient or maintain a competitive advantage. Let us take a quick look into this process of innovation.

Process of Innovation

For every innovation, tool, or process that was ultimately adopted, there may have been several dozen or more that were attempted and tried, that failed to provide value. Consider the beautiful drink known as coffee; someone thousands of years ago in Ethiopia figured out that the red berries of a particular plant held light green beans. This person peeled the beans, burned and crushed them into a fine powder before soaking them in hot water and drinking the concoction. What a treasure! While this discovery was a success, consider how many other hundreds of people became sick (or worse) when experimenting with other plants that happened to be poisonous?

One quote from Greek philosopher Plato states that ‘necessity is the mother of invention’; meaning when faced with an otherwise insurmountable obstacle or deadline, we become desperately creative in search of a solution. Any new tool to reduce the effort required to accomplish a common task represents somebody trying to gain efficiency. Think about it: throughout human history, there must have been millions of tools developed to make some process quicker, easier, or more effective. Just looking around my work desk at 2:00 in the morning on this random Tuesday, I see dozens of tools: speakers, coffee cups, stapler, batteries, cell phone, cat tree, pens and paper. Each of these items were ‘invented’ at some point to create a benefit that did not previously exist.

In our modern lives, examples of innovation are determining the quickest way to get to the freeway through side streets on the way to work. We may try 4 or 5 different combinations trying to avoid red lights, tolls, or heavy traffic before falling into a regular path. Our victory may only result in saving just a few minutes or dollars, but we work hard to determine it, nonetheless.

In the competitive economy of today, finding a definitive “better way” often requires a plethora of documentation to support analyses. Gone are the days of winging new processes or investing millions of dollars on a ‘new approach’ based on a hunch. We must be able to point to a precise pattern or correlation based on actual findings from analyses. This begins with identifying the exact step of a process that either takes the most time, resources, or perhaps has significant variance in outcomes that affect our organization’s ability to predict or control costs. Knowing that we can identify the biggest problem step in a complicated process is not easy, but let’s assume with enough experience of engaging any process we can break down each separate component by sequence and dependency.

The simplest amount of data required includes a record of what was attempted for a particularly flawed process as well as the ensuing results that have been deemed inefficient or otherwise undesirable. Prior to recorded history, these observations had to be shared orally as libraries did not exist.

We might categorize the documentation into smaller chunks including the establishment of end-result goals, articulation of step-by-step processes, analyses of each step, determination of inefficiencies or flaws, experimentation of alternative steps, followed by observation and implementation.

We are building on your understanding of data and the mechanism we use to contain and categorize data. The following diagram is a rough estimation of the steps people go through when learning about a process with a goal to make it better. Please note that the entire collection of steps outlined is iterative and leads to continual process improvement.

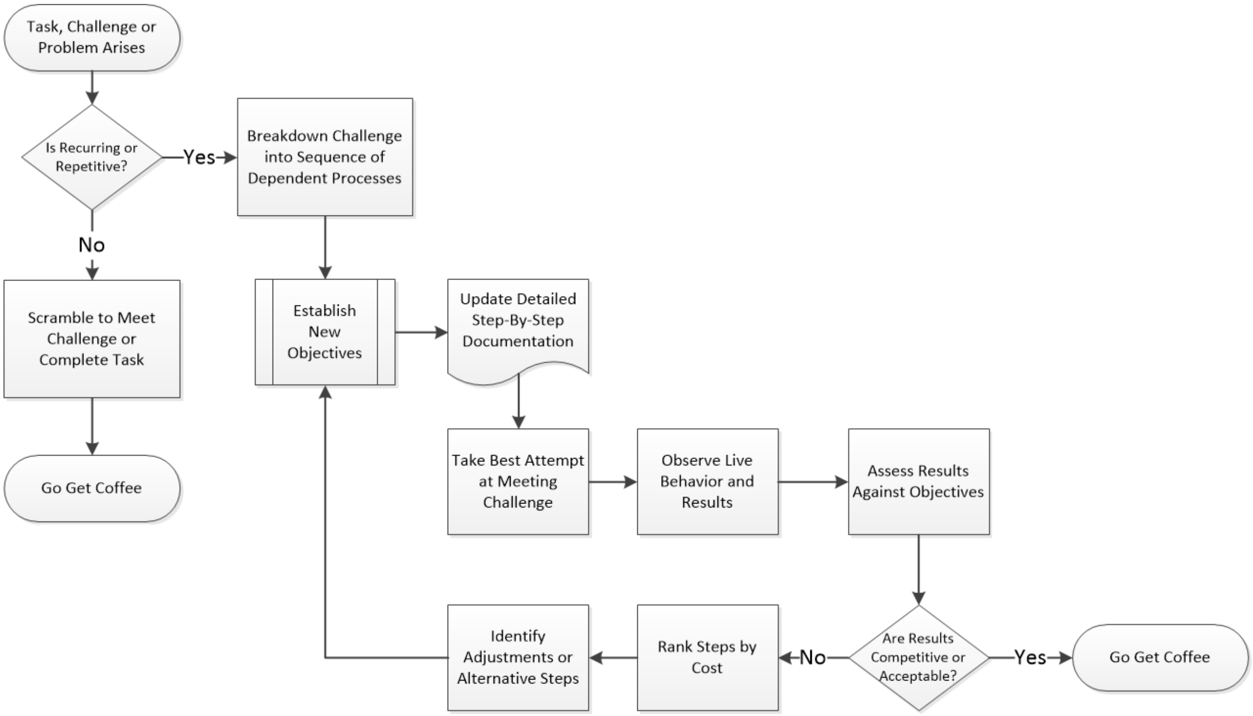

Task, Challenge or Problem Arises
Is Recurring or Repetitive?
Yes
Breakdown Challenge into Sequence of Dependent Processes
No
Scramble to Meet Challenge or Complete Task
Go Get Coffee
Establish New Objectives
Update Detailed Step-By-Step Documentation
Take Best Attempt at Meeting Challenge
Observe Live Behavior and Results
Assess Results Against Objectives
Are Results Competitive or Acceptable?
Yes
Go Get Coffee
No
Rank Steps by Cost
Identify Adjustments or Alternative Steps

Figure 1.2: Example Process of Innovation in Workflow Diagram

Documentation of Steps in a Process

To simplify, a system is made up of components or steps that are frequently sequential. If we are to analyze an existing process, the first challenge is to document the big steps of how things are being done. The next few tasks are to determine the sequence of these steps. Next is to document how and when it is known that a step is completed or ready to transition to the next one. If possible, write these details of requirements, dependencies, and expected outcomes down!

Key Performance Indicators (KPIs)

Each step most likely has desired outcomes; establish these as precisely as possible. It is important to define the metrics as well as key performance indicators (KPIs) to recognize that a step is successful. More important is knowing quickly when a particular step is failing or even just lagging ‘normal’ or expected behavior.

Observation of Real-Life Behavior

After completing the first pass of documenting a process, we will want to observe an actual execution of the process running in real-life, paying attention to how the steps are being completed. Measure the outcomes of each step according to the predetermined key-performance indicators (KPI); note which steps hit their desired marks and which did not. During this observation, I expect that there will be variance from what is documented and the behavior of people and machines. When there is a discrepancy from the expected behavior, make a note and either modify the documentation or process of execution so they align. Additionally, document the amount of time and resources such as people, money, and equipment each step takes.

Identify Least Efficient and Costly Steps

After a few cycles of going through a particular process completely, we can begin measuring which step is taking up most resources. Again, resource consumption can be measured in time, people, expenses, or equipment. We will want to rank the steps in order of resource consumption from highest to lowest. Also, our goal should be to determine the steps that had the highest variance in terms of resources required. Try and identify the root cause of the variance and determine if additional controls, training, or sub-tasks are required to level out the cost of executing this step. The steps at the top of this list are candidates for redesigning to gain better consistency of outcomes.

Experiment with Alternatives

After a few cycles of going through a particular process completely, we can begin measuring the effectiveness of replacement or alternative processes within the problematic step. This experimentation may be as simple as adding additional steps, changing the order of execution, adopting new tools, or being more extensive with new materials or technology. If the results provide better consistent outcomes, the innovation may be formally adopted as a standard. If the results do not provide consistent improvement, then they may be discarded as the next experiment is tried. Rinse and repeat until desired results are achieved.

Formally Adopt innovation as New Step in Process

Believe it or not, throughout human history, our predecessors have followed standard systems and process improvements very similar to this for centuries without computers!

While continuous improvement does not guarantee long-term success or profitability, it certainly provides a competitive advantage in most business scenarios. Storing data allows for immediate as well as long-term analyses and introduces the opportunity for investigation. Organizations that have structured the documentation and learning for their processes will continually improve the quality of goods and services they provide.

So, to repeat the crux of this book is not about human evolution or anything about plants and animals; for this book to be most effective, each reader must recognize data is the core of everything we do in the modern economy. There is simply no learning of any kind without data. There will be ‘winners’ and ‘losers’ in this upcoming era. Those who are skilled at consuming data, and recognizing patterns, trends, outliers, and effectiveness of innovations will be in a better position to win than those who do not.

To simplify into a single word the complexity of human behavior and our common need for winning control of scarce resources, it would have to be “*learning*”. It could be argued that nearly every single discovery, invention, or optimization in history was the result of the steps of innovation from above. While I am sure more than a few significant innovations have happened by pure accident, most are from someone being frustrated and saying, “there must be a better way to get this done”.

Recognizing that we are always searching for a competitive advantage through innovation or optimization, we must realize the most likely discoveries will be found in the vast amounts of data we are collecting. Anyone who believes we are ‘done’ with innovation is misguided; there is much more coming in all industries. Consider the famous quote from the head of the US patent office saying in 1899 that "everything that can be invented has been invented." That was before radio, cars, air travel, television, or even the widespread use of electricity in homes. Now, connect the dots please. Consider again that competition is real and only those with specialized skills will have a distinct competitive advantage.

Let’s wrap up this section with some final thoughts on recognizing how critical systems have been throughout human history for tens of thousands of years. First, systems continue to be just as critical as ever before in our data-driven economy. While the advent of electronic systems has been only a single lifetime (essentially 1950s but wide-spread adoption has only been since the late 1980s), the amount of data captured and used in analyses has been growing exponentially ever since. The human need for learning and ‘winning’ through innovative problem-solving has only become fiercer and more reliant on valuable insight from data. As a result, data is at the crux of most industries with the difference between dominating an industry and perhaps failing completely is very minor.

Essentially every company is investing in learning from data analyses. This includes understanding our collective behaviors and impulses better than we know ourselves. Examples include customer preferences & satisfaction with brands, supply chain & distribution efficiencies, as well as product manufacturing and materials procurement. No industry will be untouched by strategists learning from historical data!

System Development Life Cycle (SDLC)

We would like to introduce you to a very important framework, the **systems development life cycle (SDLC)**. The SDLC gives us a high-level overview of the various phases of a system's development. These are the phases each development team will gradually progress through as they construct a system.

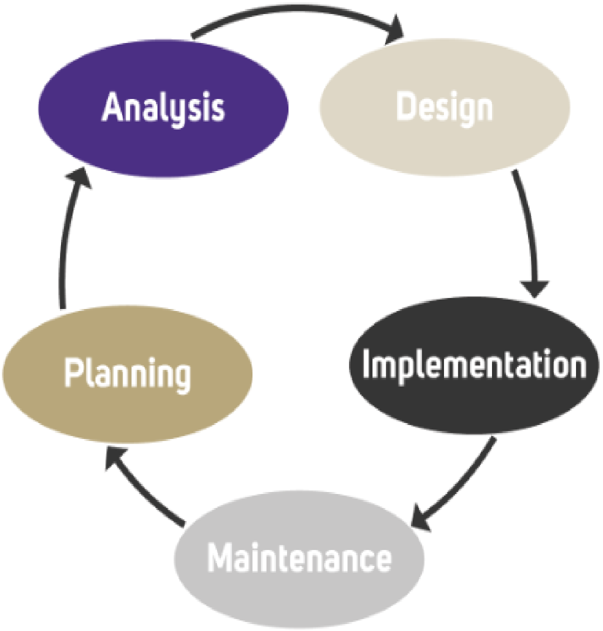


Figure 1.3: Example of Iterative Phases in SDLC

Taking a closer look at this framework, it is important to understand why we use it, how it is followed, and how it is different from a methodology. First off, we use SDLC (like any framework) as a template to increase the probability that we will have success by the end of the project. Success means that the features addressing user requirements are delivered in the timeline agreed upon and within the budget assigned. This is known as the 'scope' of a project.

Having a set of well-defined phases allows a distributed team to coordinate efforts, track progress, communicate, and be better prepared when the project hits snags or delays. Basically, the template of how to articulate the steps when building a system allows for better information; when there is better information, there is a better opportunity for learning. This means that as a team iterates through multiple projects---and therefore has multiple cycles of traversing the SDLC framework---they will have greater experience and anticipation of challenges. They become seasoned professionals who are more flexible and confident when the invariable project glitches arise.

The SDLC is a template of phases, beginning with PLANNING, followed by ANALYSIS, DESIGN, IMPLEMENTATION, and finishing with MAINTENANCE. In many common methodologies, the SDLC is iterative, and a single project may include several complete cycles of the framework, depending on the project needs. Many projects initiate in the planning phase as this is where an organization sets priorities and assigns budgets. Once a potential project has been identified, it must be investigated for viability and feasibility from a cost/benefit perspective. It is important to remember that only a handful of projects are funded (aka 'green lighted'). Each organization must make sure that a project receiving money is going to eventually pay for itself by generating additional revenue or reducing expenses.

Once a project gets the go-ahead, the analysis phase begins. This phase is where the current system is explored and the details around the problem space are defined. Users are interviewed and research is conducted to determine a set of requirements. Potential solutions are considered here as well. During the design phase, individual components like computer hardware, data models, database structures, cloud architecture, and other software aspects are explored as an integrated system. The implementation phase is where the code for each component is generated by developers. This includes testing system performance against benchmarks and training users. If the solution is a replacement of an existing system, the older system is phased out.

The final phase in the SDLC is maintenance. From a high level, this is the phase where a staff of operations engineers monitor the overall system performance and conduct the daily tasks that keep the system working. Tweaks and relatively small changes are made as necessary during the immediate post-implementation period. This phase may be several years (or even decades) in length and exceed the duration of all other phases combined, depending on the life of the system.

Again, the SDLC is a framework that can be implemented in thousands of different methods. The specific routine of implementation is called a methodology. The selection of an appropriate methodology almost always aligns with industry best practices and norms established by market leaders. For example, the building construction industry helped define what is called the waterfall methodology because it has been proven over many decades that successful projects in this industry benefit greatly from the controls completing entire phases sequentially. The factors that are prominent in the construction industry are the reluctance at issuing changes due to exceptionally high risk and subsequent cost associated with doing so.

Methodology

A methodology is simply the methods by which the SDLC is implemented. There can be thousands of distinct methodologies across dozens of industries. Several common characteristics usually define each methodology, including an emphasis on thorough documentation, measurements for success at key points (called 'Key Performance Indicators' or KPIs), as well as iteration sequential phases.

Perhaps the biggest indicator of which methodology is preferred for implementing the SDLC is the industry in which the project is being conducted. The most common methodologies and their associated industry are outlined below:

Methodology #1: Waterfall (construction)

This methodology gets its name from the visual of water cascading down and filling up a series of buckets completely before the excess water falls from one full bucket into a lower, further downstream bucket, followed by a third and fourth. This methodology is known for being exhaustive, thorough, and somewhat slow. This is a great methodology for implementing the SDLC framework for industries that have many inflexible, deliberate, and dependent requirements that also need approval or validation. This methodology also is good for industries that do not need nimbleness around rapid customer interaction; meaning once a set of project requirements are agreed upon, they rarely ever change.

Waterfall is perhaps the most popular and common methodology for construction. The reasons for this are changes are exceptionally expensive or even prohibited. Imagine having built up to the third floor of a new apartment building designed to be 4 stories in height. A customer then decides they instead want to build an office park with a little league football stadium on the site. Any work conducted in the planning, analysis, or implementation phases are rendered moot as the construction team must reverse pivot and essentially start-over. Much of the permitting, foundational work and infrastructure must be scratched as well. Changes are exceptionally expensive in the land use and construction industries, so the methodology implemented to improve success needs to reflect that.

Methodology #2: Lean/Kanban (repetitive manufacturing)

This methodology was introduced by Toyota way back in the late 1940s and revolutionized the repetitive manufacturing of heavy equipment that traversed assembly lines. It essentially breaks down each critical job (such as putting doors on a half-built car) and makes that a specialized task. This means that a person was trained and became an expert over thousands and thousands of repetitions of doing the same task. This acquired expertise enabled the experts to redefine the process, develop specialized tools, schedule a delivery of parts that was more efficient, as well as shared validation of quality. Moreover, quality became a shared experience, with each specialist being enabled to ‘shut down the line’ if they see any defect or suspect workmanship in any aspect of the manufacturing process.

The effectiveness of Kanban has not only dramatically improved the quality and lifespan of personal vehicles, but also nearly every appliance and electronic system that hits an assembly line. Cars in the 1970s, for example, had a life that would end after 50,000 miles of use. Vehicles engineered in the 2020s often exceed 200,000 miles with fewer surprises and failures. Kanban is a very interesting study into continuous improvement and is worth additional consideration for anyone even slightly intrigued about creating a culture of efficiency and quality.

Methodology #3: Scrum/Agile (software development)

The Agile methodology took hold in the late 1990s and early 2000s after several colossal failures of large companies in the technology sector. Before agile, software development teams often managed projects by waterfall methodology, which again is deliberate, precise, and SLOW. Changes to agreed-upon features are difficult to incorporate and customer interaction is diminished. The process often took 3 years to complete a single product release. Unfortunately, the problems being addressed when the project began are no longer relevant either or the customers no longer needed the solution.

Agile emphasizes rapid ‘sprints’ that seek to deliver only a handful of features in a span of only a few weeks. Customers often attend daily ‘scrum’ meetings, with eager engagement. Changes and updates are more easily included frequently with little concern or fanfare.

This methodology understands the dynamic nature of software (as well as the short attention span of most software engineers) and builds a process that makes these beneficial to getting a high-quality product in the hands of customers very quickly. Customers win because they feel listened to, and they have functional products in a matter of months if not weeks. Development companies win because they have more happy customers and get revenue quicker, with less bloat and risk. The coders win because they can specialize, find interesting and challenging work that keeps them busy and valued.

Again, agile is the dominant methodology in the technology industry even if the tech workers are employed at a large manufacturing company. Take the time to become familiar with this methodology if you intend on becoming a professional with data, software engineering, or analytics.

Key Takeaways

Companies and their customers benefit when projects get completed on time. Having flexible and customizable methodologies that align with the unique culture of each different industry allows an organization to adopt a set of processes to improve their learning about HOW to create their product or service. The companies learn over time the specific way of delivering their goods (be it construction, manufacturing, or coding) with the identification of mistakes as well as successful corrections. This repetitive reflection over time allows each interested company better insight and accuracy in the following:

- Planning (budgeting)
- Better efficiencies in spending
- Task management (Recognition of dependencies, Prioritization
- Avoidance of problematic situations
- Troubleshooting (ever-expanding knowledge base and embedded expertise)
- Reaction to issues quicker and more efficient

Again, each industry (and to a lesser extent, each organization) can develop a unique methodology that fits their culture and approach to delivering their goods. The goal is to find a process that works! This is accomplished by capturing data throughout the life of many projects and building a body of knowledge from which analysts can learn from. This repetitive experience allows for continual improvement, establishment of ‘best practices’, the nurturing of specialists for each process as well as being better able to on-board or train new employees quickly with little disruption. Customers benefit because the quality of the product or service purchased is predictable with less variance.

Perhaps the greatest benefit of developing a methodology is when a project schedule begins to slip, or something occurs that was unplanned. This introduces risk to the overall project scope. With a proper methodology, the risk is often mitigated as mistakes and corrections are anticipated into the process. We say, “bad news early is better than bad news late” and the sooner a risk is identified, the sooner (and cheaper) it can be corrected. Future projects learn from the work and projects done previously, allowing the company to be more accurate on making estimates of budgeting and timelines.

This is an opportunity to iron out any confusion you may have about the relevance and application of the SDLC in the realm of database management. Before we can begin working with real data, we need a methodology and framework to contain our ideas and ground our innovation. For now, I want you to hone your understanding of the SDLC.

Comparing SDLC with Organize or Die

Now that we have been exposed to “Organize or Die” (an adaptation of “Survival of the Fittest” by Charles Darwin), as well as the system development life cycle (SDLC), let us compare the two. Organize or Die can be seen as an overriding philosophy that affects an individual society’s evolution or an individual person or organization’s approach to a position, problem, or career.

The SDLC is a framework that is effective in response to Organize of Die; it is a method used to become more effective and efficient in a competitive environment through repetition, evaluation/analyses, and continuous innovation.

Data is at the core of all learning; being able to quickly, effectively, and strategically build and implement well-designed databases will support any organization seeking to learn. Again, in the modern economy, every organization understands they must learn at a rapid rate if they want to compete for customers, manufacturing efficiencies, or other insight into their operations.

In this section, we have established a foundational need for data and information to support the never-ending quest humans must improve. The next section covers the methods of data management people employed throughout history to learn and improve efficiencies to gain a competitive advantage.

Database History: Paper-Based, Hierarchical, and Network

We have used paper-based systems for centuries! We have kept track of things using paper, and we still have paper-based systems to this day. Post—World War II, the goal was to automate a lot of paper processes. Come the 1970s, we introduced **relational theory**, and we have not looked back. If you want to see a grown man cry, just ask me about the significance of this evolution!

You need to be able to put data collection (and more importantly, "learning") into context. We have a human need to learn for understanding as well as for gaining efficiency (we are a little lazy at our core). Ultimately, we are competitive and continuously seek to reinvent processes to gain a competitive advantage. Data is at the core of all these motivations (without data in all its forms, none of these would be possible).

Throughout history, humans have sought to learn. As long as there have been people and societies, there has been focused learning to understand the complexities of science, the natural world, and us. Data has always been at the core of learning as we document all things starting with our goals and objectives, step-by-step processes of each objective task, results of these processes as well as our observations and analyses of outcomes in relation to our original goals and objectives.

As we saw previously, for thousands of years of humankind (minus the last 75 years or so), data has been managed manually with pen and paper (sometimes chalk or even crayon, depending on how old we were at the time). While this is easy to implement, it has obvious limitations in security, durability, and the ability to digest at any valuable scale.

These limitations combined with the competitive nature of post-World War II enterprises saw opportunities to gain significant advantages with computerizing their processes. Their primary objectives were to gain efficiency in production and distribution and increase the output of goods and services to reach the burgeoning middle class.

The hierarchical model was the *de facto* standard with paper because there is essentially no other way to manually organize physical items. We group physical items in similar clusters all the time, like a closet full of clothes being arranged by season, outing, size, or even material. Tools in a shed might be organized by their functional task, such as landscaping, woodworking, painting, and minor household repair.

Computers were effectively a new tool suddenly dropped in the middle of long-standing and standardized business processes that had existed for centuries, such as order processing, supply chain management, and inventory control. Obviously, these processes were optimized around manual data collection and reading, which often was structured in a top-down or hierarchical manner intended specifically to answer questions! As many companies had whole departments and processes organized around paper-based systems, there was no apparent need to change "how" data was being processed. This led to the first computers mimicking the hierarchical model by default as it simplified the time and expense of adopting computers.

Early computerized systems had flaws, as hierarchical design has limitations that were previously less noticeable in manual paper-based systems. The hierarchical model has a rigid top-down structure where each level has only one "parent" that controls access to the lower level. Well, hopefully we can realize that this rigid structure of one parent is difficult when applied to real-world scenarios of business operations. Not every relationship is hierarchical (strictly defined as one-parent only) especially when tracked over a 50-year period. Most businesses engage in complex markets, where flexibility and reactions to change or fluctuations require speed and agility to remain competitive.

Also, to save time and resources, hierarchies are essentially designed as if people are reading the data 'in place on disk' directly from the hierarchy. This line of thinking causes designers to organize the hierarchy in the shape of an instantly readable report with all data required for a report included in the structure of the hierarchy. Taking this report-design problem one step further, each hierarchy must have its own copy of common data.

Multiple hierarchies in a database require multiple copies of similar data, and artificially inflate the overall size, time for processing simple inserts in multiple locations, as well as the costs for maintaining the extra storage. You may have heard analysts complain of 'duplicate data' or 'data redundancy' previously; there are reasons why!

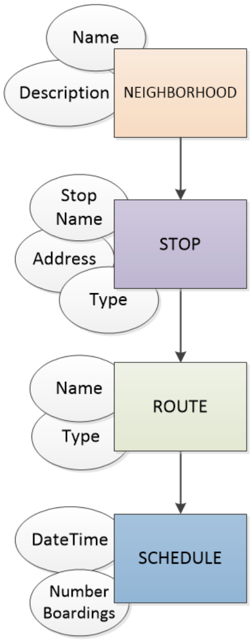
Insertion/Deletion Anomalies

The hierarchical model has what are called Insertion and Deletion Anomalies. These are caused by the rigid design structure that requires a record to have only one single parent. The insertion anomaly becomes visible when a company receives a record of data, yet a parent record does not exist to attach it under. Similarly, a deletion anomaly occurs when a parent record is set to be deleted but there exists one or more child/dependent records underneath it that are still considered valuable and not desired to be deleted.

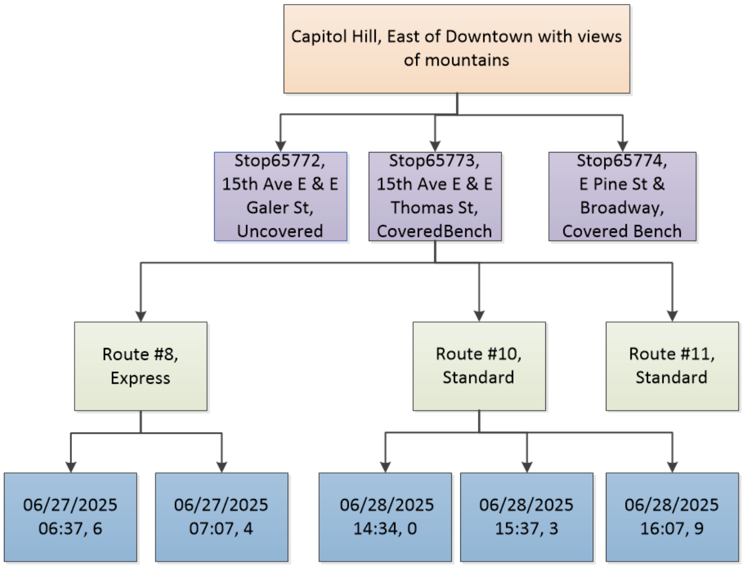
These challenges were difficult and often affected how companies conducted business operations. Many times, companies tried to force their practices into a rigid hierarchical model. An example that I encountered in one of my first jobs at age 17 at a pizza chain restaurant was not being able to work a shift at a different store under the same company. In the pizza chain's hierarchical database system for payroll, employees were hired at a store (not just in general). That meant when I was asked to cover a single shift at a different location than my home store, the company would have been required to hire me again under that other store. This meant filing official paperwork with the State of Washington and Internal Revenue Service (IRS). This was considered too much of a hassle to do things the right way, so I was 'paid' by the manager letting me borrow his car to go see a ZZ Top concert in Tacoma with friends. Work-arounds like this were very common during the 1980's, as many companies struggled with inflexible database systems that did not align easily with the demands of real-life business operations.

One more detailed example of a hierarchy is a bus transit system where a database is organized in a collection of several hierarchies. When looking at these hierarchies, consider that they are trying to follow a cabinet filing system of order forms, invoices, and paper receipts. In paper-based systems, each original record is frequently run through a copy-machine for each folder where a duplicate is needed. The data stored in these transit hierarchies may be centered around a few common areas of management:

- NEIGHBORHOODS
- DRIVERS
- VEHICLES
- ROUTES
- TRIPS/SCHEDULES



Hierarchy A: Neighborhood Focus


Capitol Hill, East of Downtown with views of mountains
Stop65772, 15th Ave E & E Galer St, Uncovered
Stop65773, 15th Ave E & E Thomas St, CoveredBench
Stop65774, E Pine St & Broadway, Covered Bench
Route #8, Express
Route #10, Standard
Route #11, Standard
06/27/2025 06:37, 6
06/27/2025 07:07, 4
06/28/2025 14:34, 0
06/28/2025 15:37, 3
06/28/2025 16:07, 9

Hierarchy A: sample data

Figure 1.4: Example hierarchy of neighborhood trips for METRO_TRANSIT

Let us see another example from METRO_TRANSIT:

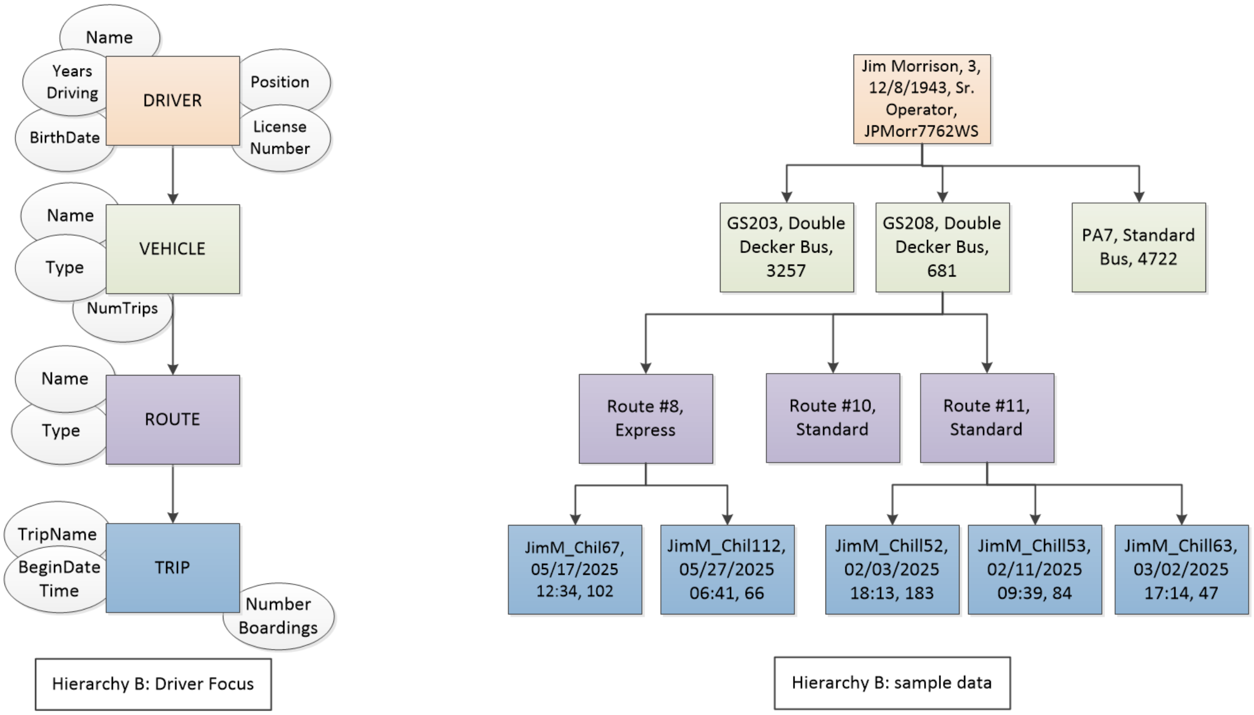

Name
Years Driving
BirthDate
DRIVER
Position
License Number
Name
Type
VEHICLE
NumTrips
Name
Type
ROUTE
TripName
BeginDate Time
TRIP
Number Boardings
Hierarchy B: Driver Focus
Jim Morrison, 3, 12/8/1943, Sr. Operator, JPMorr7762WS
GS203, Double Decker Bus, 3257
GS208, Double Decker Bus, 681
PA7, Standard Bus, 4722
Route #8, Express
Route #10, Standard
Route #11, Standard
JimM_Chil67, 05/17/2025 12:34, 102
JimM_Chil112, 05/27/2025 06:41, 66
JimM_Chill52, 02/03/2025 18:13, 183
JimM_Chill53, 02/11/2025 09:39, 84
JimM_Chill63, 03/02/2025 17:14, 47
Hierarchy B: sample data

Figure 1.5: Example hierarchy of driver-centric data for METRO_TRANSIT
Inefficiencies with Hierarchical Model

There are several inefficiencies associated with the data stored in the hierarchies above. First, words for the types of bus stop and route are used to provide clarity. These words are necessary in this design because users will want to know if they should be sure to bring an umbrella if the bus stop is uncovered. Also, the type of route (express or standard) may be important for the passenger in a hurry. The use of words for these values is of concern not only because they take up more storage space than reference, but also because they introduce the opportunity for typos or misspellings.

A second inefficiency with the NEIGHBORHOOD hierarchy is having a summary of Number of Boardings at the base of the hierarchy. While easy and convenient for a user to read if they are seeking data regarding a single route, these values are difficult to include in aggregations across multiple routes as a full traverse through the hierarchy is required to obtain additional values. Also, an update is required each time a bus completes a scheduled route to make sure the boarding at each stop is properly recorded.

Please note that the driver-centric hierarchy is an entirely different structure from the first one focused on neighborhoods, yet it is filled with similar data. There are similar inefficiencies as before with using words to describe ‘type’ values for vehicle and route, but since these entire hierarchies have duplicate values, any INSERT, UPDATE, or DELETE will need to be written in both structures. This means basic transactions will take longer to process even for simple statements. Any slight mistake will introduce discrepancies where different values are returned for the same question depending only upon which hierarchy is engaged.

One final inefficiency is the math of ‘Number of Boardings’ recorded at the base of the hierarchy under TRIP. This value just might be the result of a human calculation adding up the values either as they happen or by tallying values from the neighborhood hierarchy. Either way, integrity and consistency of these values across hierarchies are exceptionally vulnerable and at risk of being accidentally misrepresented. Our ability to make decisions based on accurate data is in jeopardy.

Again, imagine this hierarchy as a collection of paper that is placed in folders and stored in a steel file cabinet. When a person wants to learn from the data, their search begins by first recognizing which perspective or focus they need to take, such as DRIVER or NEIGHBORHOOD, as this will determine the hierarchy (or file cabinet) they will search. The same data is stored in each hierarchy but the path to the desired data will be shorter depending on which hierarchy is chosen.

Do we need one more example? In the 15 years I have been teaching college, students have often asked for more specific examples in different data subjects. If the above example from METRO_TRANSIT was clear for you, please feel free to skip ahead a few paragraphs.

One more example of a hierarchical structure could be how we track music from an online streaming service (Amazon Music, Spotify, Pandora come to mind). We may want to be able to retrieve songs and play them from a certain artist, genre, or perhaps from another person’s playlist. How might this be organized in a hierarchical structure? The short answer is a different hierarchy for each! Please review the following:

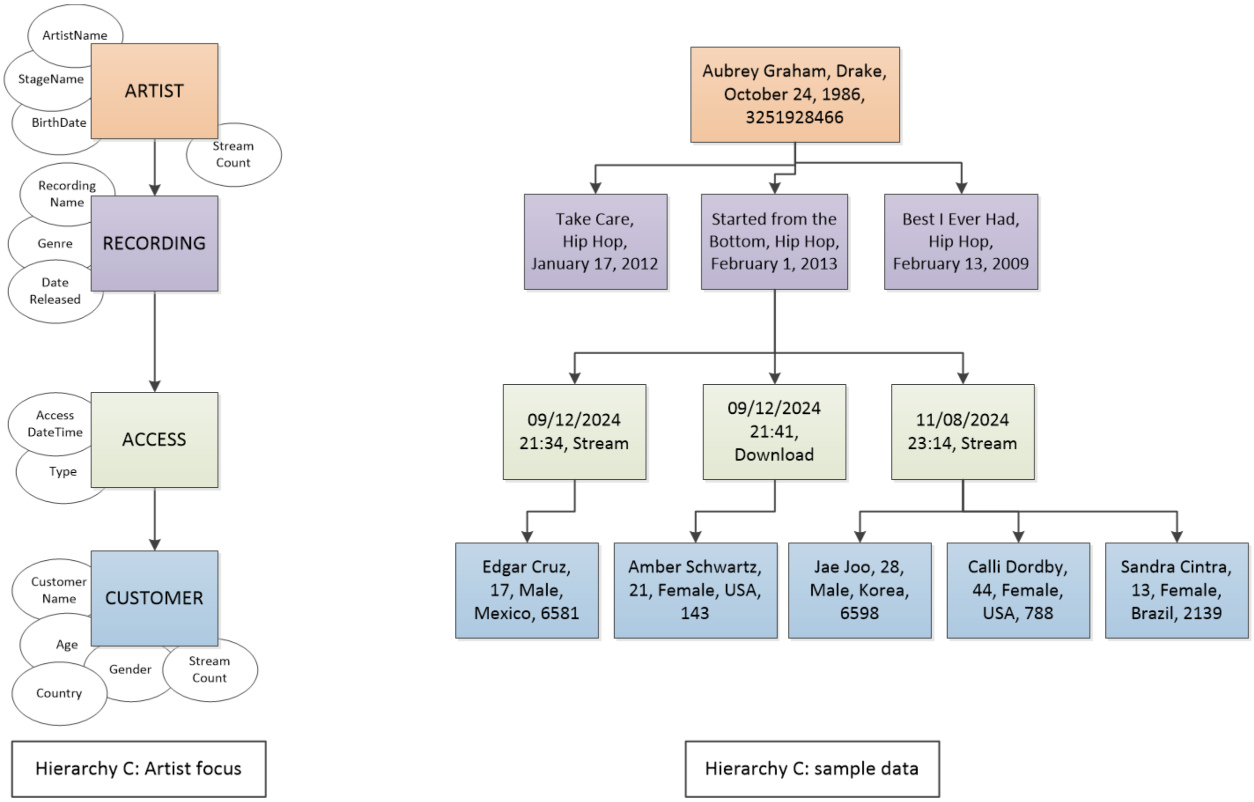

ArtistName
StageName
BirthDate
ARTIST
Stream Count
Recording Name
Genre
Date Released
RECORDING
Access DateTime
Type
ACCESS
Customer Name
Age
Country
Gender
Stream Count
CUSTOMER
Hierarchy C: Artist focus
Aubrey Graham, Drake, October 24, 1986, 3251928466
Take Care, Hip Hop, January 17, 2012
Started from the Bottom, Hip Hop, February 1, 2013
Best I Ever Had, Hip Hop, February 13, 2009
09/12/2024 21:34, Stream
09/12/2024 21:41, Download
11/08/2024 23:14, Stream
Edgar Cruz, 17, Male, Mexico, 6581
Amber Schwartz, 21, Female, USA, 143
Jae Joo, 28, Male, Korea, 6598
Calli Dordby, 44, Female, USA, 788
Sandra Cintra, 13, Female, Brazil, 2139
Hierarchy C: sample data

Figure 1.6: Example hierarchy of artist-centric data for MUSIC_STREAM

Once again there are challenges for efficiency and maintaining data integrity with the example hierarchy above for a music streaming service. Please note the duplicate words for genre, access type, gender, and country. These will be problematic as the subscription base increases and the number of users and overall streaming activity builds and any typos or abbreviations become more impactful.

The most notable flaws of trying to query data from a hierarchical model were when the exact hierarchy to start the query was unknown. Imagine trying to locate the details about a specific TRIP when none of the higher-level facts are known (such as ROUTE, BUS, or DRIVER). Locating the TRIP data would require a significant time-consuming task of traversing each possible combination of the higher-level values, perhaps with hundreds or even thousands of iterations. Please notice the inefficiencies associated with a hierarchical data storage structure:

- Redundant copies of data are written to several hierarchies on each transaction, which increases the time required to process a single transaction (writing to three or four hierarchies equals three- or four-times duration)
- Redundant data requires detailed understanding of ALL hierarchies as well as a manual process to validate integrity throughout the group

- One slight mistake of either missing an update or deletion creates inaccurate data; this means that different answers are returned from the system depending on WHERE the question is searched

These alternate structures created redundant values that effectively doubled or tripled data storage demands. Beyond being bloated, slow, and expensive, these duplicate structures required extra care to maintain accuracy. Again, each insert, update, and/or delete had to be performed in each hierarchy (again, significantly slowing each operation). Unfortunately, over a period of months and years, many operations were not completed in each hierarchy as required and inaccurate data ensued. The inaccurate data led to people not trusting any data that was retrievable, resulting in many people resorting to managing data offline and even retreating to paper-based systems!

Additional flaws of the hierarchical database model were effectively having computer systems dictate limitations to the business, where tasks or actions that did not fit neatly into the established set of hierarchies were not being allowed. An example of this frustrating restriction is when a driver “covers a shift” of a second driver in either another route or with a different bus. These rigid hierarchical database designs did not allow drivers to be assigned a bus, route, or trip dynamically. The result is that a business had to rehire the driver for a single trip, thereby duplicating their personal data and creating downstream confusion or misinformation when not cleaned up properly in the computer system. Pain!

These flaws led to the creation of the network database model that was still hierarchical in structure but allowed a workaround or band-aid for records to be “owned” by more than one parent (basically an object higher in the hierarchy such as DRIVER to TRIP). While the network database model better aligned with the navigation processes, it did not address the major issues of duplicate data and resulting error-prone inconsistencies over time.

Breakthrough to a new data model

There was a breakthrough in database management around 1970. The relational model was articulated in late 1970 by Dr. Edgar Codd, a data scientist working for IBM. As we will see, the relational model effectively solved ALL previous flaws for collection and retrieval of large data sets and remains unchanged for over 50 years (tears should be welling up in your eyes!). We have seen so far that database design history follows human history as each civilization had their own methods for communicating, record-keeping, and learning. Each civilization sought to lessen the burden of survival by the innovation of essential processes such as farming, carpentry, healthcare, and basic education. As a global society, we continue to search for efficiency via innovation every single day. As data is at the center of all learning, the major development in the relational model cannot be overstated as it has allowed for massive increases in learning, changing every aspect of our lives from how goods and services are manufactured, delivered, and consumed throughout the world.

Flaws and Limitations of Spreadsheets

One last topic to be covered in this first chapter is to understand when to think of data management in terms of a 'spreadsheet' and when to abandon it. Simply put, early in the data management process, such as when we are capturing the creation of original transactional data for long-term storage, we want to avoid thinking like we would if we are working with a spreadsheet. A spreadsheet is best when we are conducting 'last mile' analysis of data and perhaps making visualizations of potential scenarios. When designing robust data storage systems with millions of rows of data, it is critical we avoid treating data as something people will be reading; instead, we need to switch our thinking to treating data as something machines will be reading.

As mentioned in the preface to this book, the first computer-industry job I ever had was being a teacher's assistant and computer lab aid at Shoreline Community College in March 1986. The second computer-industry job I ever had was being a contractor at Microsoft, beginning in May 1987 testing the Excel spreadsheet product on the new Windows platform under development at the time called Windows 386. I share this brief story to perhaps add credibility to my love for Excel and sincere belief that Excel is perhaps the greatest technical piece of software yet created.

While Excel might be the most popular application ever, we need to curb our excitement around using it to solve every data-related problem we encounter (believe me…I have tried). Excel has its place, but it is time to cut the umbilical cord and venture into the beautiful realm of relational theory with a purpose and awareness of these limitations.

Spreadsheets are perhaps the single most common computer application in use across the world after Internet browsing and email. This is not an accident! Spreadsheets are tremendously valuable when organizing simple tasks, estimating costs, or calculating future values. There are several significant limitations that curtail the effectiveness of using spreadsheets in most high-volume, data-intensive mission-critical business scenarios. These include security, consistency, scalability/automation, and querying data at volume.

Security

One of the benefits of working with spreadsheets is the ease of access and manipulation of data. They are often completely available to modify either by over-writing with new values or changing the underlying formulas that do the math. This great flexibility is part of the power and attraction to spreadsheets being one of the world's most popular applications. With this flexibility, unfortunately there are often insufficient protections and security for data. Many times, there are no restrictions for users to access a spreadsheet at the file-level or the cell level. This often means we must be diligent about how and when we share sensitive data. Additionally, not all data in a spreadsheet can be trusted to be transactionally sound.

Consistency

Again, the brilliance of spreadsheets is its power, flexibility, and ease of use. Unfortunately, the integrity associated with transactional data is difficult if not impossible to enforce in a spreadsheet. This ability to overwrite data on a whim may be great when we are exploring data with 'what if' projections, but it introduces data anomalies and inconsistencies.

Scalability/Automation

As stated previously, spreadsheets are fantastic analytical tools yet are not well-suited for managing high volume transactions. This is a significant limitation of spreadsheets. People frequently try to store transactional data in a spreadsheet because it is a tool they are comfortable with. They end up 'copy and pasting' chunks of data into their spreadsheets in a manual process because they are unfamiliar with more robust methods of capturing transactions, such as relational systems.

Querying Data

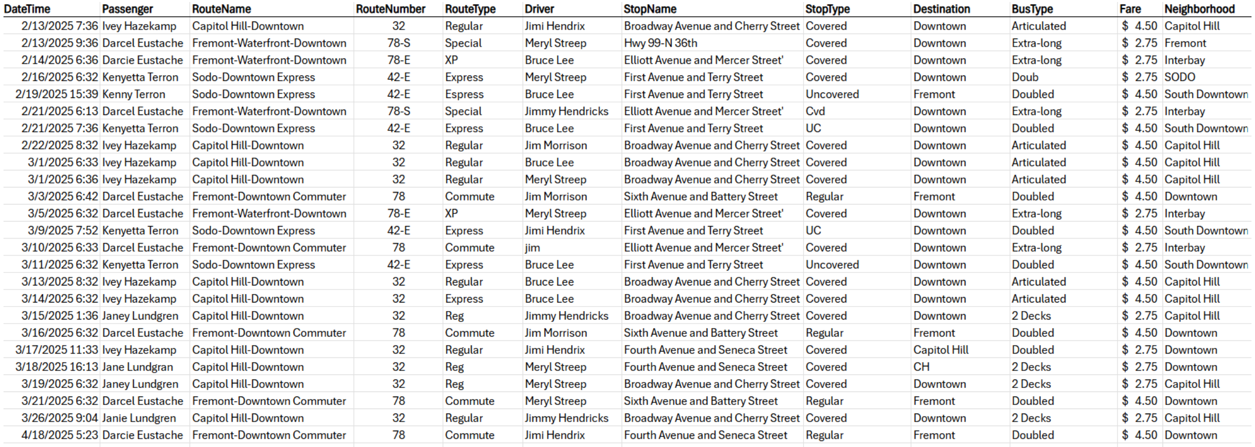
As we shall see shortly, transactional details are difficult to query in a spreadsheet beyond basic aggregates, like average, maximum, and minimum. This is because a spreadsheet works best with summarized or aggregated data, not the details of thousands or millions of individual transactions. Better methods exist to engage transactional data without losing the ability to conduct complex queries across millions of rows of transactional data.

Understanding the purpose and limitations of a spreadsheet will help align your expectations, simplify your search for tools, reduce frustrations, and increase productivity and effectiveness. In the next section, you will unpack the relevance of a book club in the database management sphere.

METRO_TRANSIT as a Spreadsheet Example

You may have previous experience with a rail or bus system as most major cities have some level of public transportation. Please review the following table that contains example data of a bus and rail system, specifically tracking passengers who have boarded various vehicles and paid fares.

While reviewing the data in the spreadsheet, consider how easy (or not!) reading and ultimately learning from this table will be.



DateTime	Passenger	RouteName	RouteNumber	RouteType	Driver	StopName	StopType	Destination	BusType	Fare	Neighborhood
2/13/2025 7:36	Ivey Hazekamp	Capitol Hill-Downtown	32	Regular	Jimi Hendrix	Broadway Avenue and Cherry Street	Covered	Downtown	Articulated	$ 4.50	Capitol Hill
2/13/2025 9:36	Darcel Eustache	Fremont-Waterfront-Downtown	78-S	Special	Meryl Streep	Hwy 99-N 36th	Covered	Downtown	Extra-long	$ 2.75	Fremont
2/14/2025 6:36	Darcie Eustache	Fremont-Waterfront-Downtown	78-E	XP	Bruce Lee	Elliott Avenue and Mercer Street'	Covered	Downtown	Extra-long	$ 2.75	Interbay
2/16/2025 6:32	Kenyetta Terron	Sodo-Downtown Express	42-E	Express	Meryl Streep	First Avenue and Terry Street	Covered	Downtown	Doub	$ 2.75	SODO
2/19/2025 15:39	Kenny Terron	Sodo-Downtown Express	42-E	Espress	Bruce Lee	First Avenue and Terry Street	Uncovered	Fremont	Doubled	$ 4.50	South Downtown
2/21/2025 6:13	Darcel Eustache	Fremont-Waterfront-Downtown	78-S	Special	Jimmy Hendricks	Elliott Avenue and Mercer Street'	Cvd	Downtown	Extra-long	$ 2.75	Interbay
2/21/2025 7:36	Kenyetta Terron	Sodo-Downtown Express	42-E	Express	Bruce Lee	First Avenue and Terry Street	UC	Downtown	Doubled	$ 4.50	South Downtown
2/22/2025 8:32	Ivey Hazekamp	Capitol Hill-Downtown	32	Regular	Jim Morrison	Broadway Avenue and Cherry Street	Covered	Downtown	Articulated	$ 4.50	Capitol Hill
3/1/2025 6:33	Ivey Hazekamp	Capitol Hill-Downtown	32	Regular	Bruce Lee	Broadway Avenue and Cherry Street	Covered	Downtown	Articulated	$ 4.50	Capitol Hill
3/1/2025 6:36	Ivey Hazekamp	Capitol Hill-Downtown	32	Regular	Meryl Streep	Broadway Avenue and Cherry Street	Covered	Downtown	Articulated	$ 4.50	Capitol Hill
3/3/2025 6:42	Darcel Eustache	Fremont-Downtown Commuter	78	Commute	Jim Morrison	Sixth Avenue and Battery Street	Regular	Fremont	Doubled	$ 4.50	Downtown
3/5/2025 6:32	Darcel Eustache	Fremont-Waterfront-Downtown	78-E	XP	Meryl Streep	Elliott Avenue and Mercer Street'	Covered	Downtown	Extra-long	$ 2.75	Interbay
3/9/2025 7:52	Kenyetta Terron	Sodo-Downtown Express	42-E	Express	Jimi Hendrix	First Avenue and Terry Street	UC	Downtown	Doubled	$ 4.50	South Downtown
3/10/2025 6:33	Darcel Eustache	Fremont-Downtown Commuter	78	Commute	jim	Elliott Avenue and Mercer Street'	Covered	Downtown	Extra-long	$ 2.75	Interbay
3/11/2025 6:32	Kenyetta Terron	Sodo-Downtown Express	42-E	Express	Bruce Lee	First Avenue and Terry Street	Uncovered	Downtown	Doubled	$ 4.50	South Downtown
3/13/2025 8:32	Ivey Hazekamp	Capitol Hill-Downtown	32	Regular	Bruce Lee	Broadway Avenue and Cherry Street	Covered	Downtown	Articulated	$ 4.50	Capitol Hill
3/14/2025 6:32	Ivey Hazekamp	Capitol Hill-Downtown	32	Express	Bruce Lee	Broadway Avenue and Cherry Street	Covered	Downtown	Articulated	$ 4.50	Capitol Hill
3/15/2025 1:36	Janey Lundgren	Capitol Hill-Downtown	32	Reg	Jimmy Hendricks	Broadway Avenue and Cherry Street	Covered	Downtown	2 Decks	$ 2.75	Capitol Hill
3/16/2025 6:32	Darcel Eustache	Fremont-Downtown Commuter	78	Commute	Jim Morrison	Sixth Avenue and Battery Street	Regular	Fremont	Doubled	$ 4.50	Downtown
3/17/2025 11:33	Ivey Hazekamp	Capitol Hill-Downtown	32	Regular	Jimi Hendrix	Fourth Avenue and Seneca Street	Covered	Capitol Hill	Doubled	$ 2.75	Downtown
3/18/2025 16:13	Jane Lundgran	Capitol Hill-Downtown	32	Reg	Meryl Streep	Fourth Avenue and Seneca Street	Covered	CH	2 Decks	$ 2.75	Downtown
3/19/2025 6:32	Janey Lundgren	Capitol Hill-Downtown	32	Reg	Meryl Streep	Broadway Avenue and Cherry Street	Covered	Downtown	2 Decks	$ 2.75	Capitol Hill
3/21/2025 6:32	Darcel Eustache	Fremont-Downtown Commuter	78	Commute	Meryl Streep	Sixth Avenue and Battery Street	Regular	Fremont	Doubled	$ 4.50	Downtown
3/26/2025 9:04	Janie Lundgren	Capitol Hill-Downtown	32	Regular	Jimmy Hendricks	Broadway Avenue and Cherry Street	Covered	Downtown	2 Decks	$ 2.75	Capitol Hill
4/18/2025 5:23	Darcie Eustache	Fremont-Downtown Commuter	78	Commute	Jimi Hendrix	Fourth Avenue and Seneca Street	Regular	Fremont	Doubled	$ 4.50	Downtown

Table 1.2: Example METRO_TRANSIT data in a spreadsheet

Please note that this spreadsheet is flawed in the sense that it is not normalized and is written for people as opposed to a database application. While using a spreadsheet to track this data might be relatively easy to get started with, it proves to be cumbersome and difficult as we add more and more data.

After 'eyeballing' this data for a few minutes, consider some of the following questions:

1) Which route generated the most revenue from fares?
2) Which driver had the most trips?
3) Which passengers took the most trips to Capitol Hill?

If these questions cause frustration, you are beginning to see why 'eyeballing' data is a losing proposition! Perhaps the single-greatest reason companies transitioned from paper-based systems when computers became affordable is that their organizational learning was severely restricted under a manual process. The very few questions that can be answered by manually looking at transactional data are so simple that the effort is almost not even worth it.

Each of the questions above requires reading each row (perhaps multiple times) and handwriting notes! Obviously, manual processing is slow, cumbersome, dependent on secondary note-taking, and prone to human error. Probably not enough value or timely insight to discover patterns or trends in behavior or feel confident making bold or competitive decisions.

The important part of the spreadsheet example is to recognize the limitations of managing high-volume, transactional data that includes millions of rows per day or more. We must break free of our reliance on 'eyeballing' data and embrace the power of relational theory and the associated processes that support data-driven decision making.

Post-Chapter Challenges

Based on your objectives and intentions on learnings from this book, please approach the following challenges as appropriate.

Track 1 (THINK): Data Tourist Seeking Ancillary Awareness
Please spend a total of 10 minutes reviewing the following questions, exploring your thoughts. These questions and your responses cut to the essence of this chapter:

- How important are systems for societies, organizations, and individuals? Why?
- What are some example breakthroughs of innovation throughout human history that can be attributed to a sense of Organize or Die? What is the significance of these innovations in people's everyday lives?
- Which contemporary companies or organizations are executing Organize or Die well? Which companies/organizations did you choose? Why?
- How come something as simple as scheduling a bus trip explodes in complexity?
- Why/How is the use of a spreadsheet to manage data problematic or flawed?

Track 2 (WRITE): Dedicated Student or Recent Graduate:
WRITE several paragraphs in response to each of the following questions, exploring each as if you are being asked a similar question during a job interview!

- Which data are required to track a passenger’s travel habits and preferences?
- What are the questions a typical passenger may want to have answered when making the selection of which route is most appropriate?
- What are the metrics/measurements to determine whether a route, driver, or stop is ‘successful’?
- How frequently will these metrics be reviewed? Why?
- Reflect on the presence of misspellings in the spreadsheet; how might it be best to manage the typos and/or abbreviations out of the spreadsheet?

Track 3 (BUILD): Full Speed Learner Seeking Job
This track targets readers of this book who want to develop professional skills working with data to launch a career or obtain a more satisfying job. Let’s begin with a challenge to structure data collection to keep track of a large-scale metropolitan transit system:

- Copy the column headers (as well as some of the data from figure 1.3) into an Excel spreadsheet that allows for greater tracking of a metropolitan transit system.
- Assume the users of your data set want to learn about route safety; begin tracking data on incidents of injuries, assaults, or aggressive panhandling. What data needs to be captured? Why? Don’t worry about proper data modeling---yet. Just get your ideas down in a manner that makes sense to you.

Conclusion

Why do people need to have a competitive advantage in nearly every aspect of our lives? Why are we continuously seeking efficiency and ‘better ways’ of doing ordinary things? We cannot help ourselves; we are ingrained to improve processes, produce more, save time, and prosper. The core to all innovation, discovery, and gains in product quality have always been based on data. For centuries, innovations were slowly adopted because the process of innovation was manual. Any discovery that produced a better way was probably only able to be shared with people immediately within earshot. Nowadays, not only can problem-solving issues include anyone with access to the internet, but learnings are shared across the globe in mere minutes. Collaboration, experimentation, and learning occur at lightning speed based on immense amounts of digital databases. Anyone with the right technical skills can participate and help shape the future!

Next up is becoming more familiar with the structure of relational databases to better prepare everyone for later concepts in this book. Relational databases are often the birthplace of much of the data we have at our disposal for analytical processing. Virtually every organization creates original data from many aspects of doing daily operations of their business, such as manufacturing products or providing services. Additionally, secondary data, from customer purchasing preferences can be combined with data from other businesses in unrelated industries to determine very complex relationships and patterns of behavior. When done correctly (and ethically), the learning across industries can be truly groundbreaking. Keep on rolling!