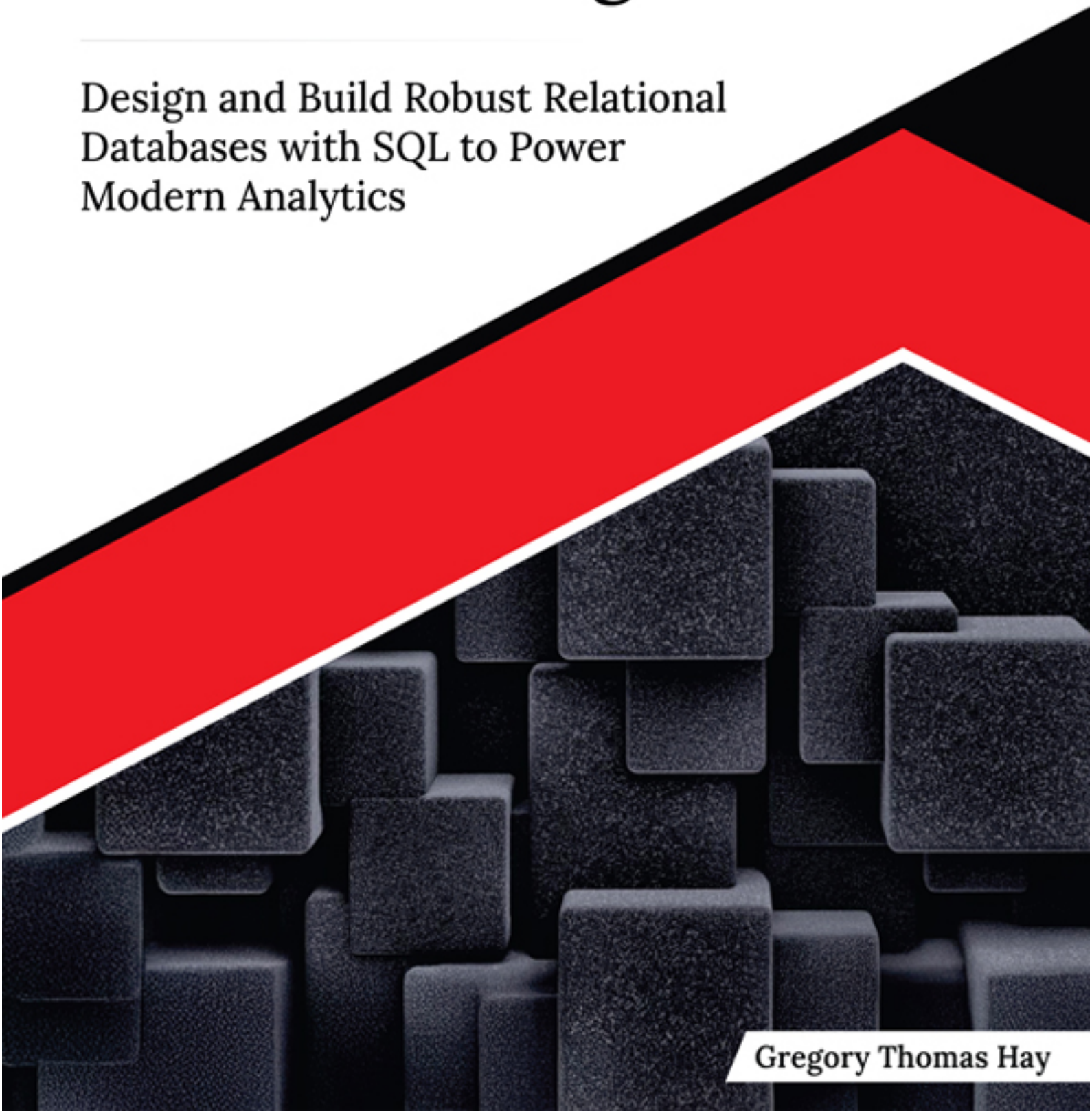
ULTIMATE

SQL for Relational Database Design

Design and Build Robust Relational Databases with SQL to Power Modern Analytics



Gregory Thomas Hay

First Published: February 2026

Published by: Orange Education Pvt Ltd, AVA®

Address: 9, Daryaganj, Delhi, 110002, India

275 New North Road Islington Suite 1314 London,
N1 7AA, United Kingdom

ISBN (PBK): 978-93-49887-47-3

ISBN (E-BOOK): 978-93-49887-48-0

Scan the QR code to explore our entire catalogue

www.orangeava.com

Table of Contents

1. Organize or Die: Learning from Data

- Introduction
- Structure
- Organize or Die
  - *Short-Term or Immediate Needs*
  - *Long-Term or Continuous Success*
- Information Ladder: DIKW Pyramid
  - *Data*
  - *Information*
  - *Knowledge*
  - *Wisdom*
- Data in Ancient Societies
  - *Systems*
  - *Systems Analysis*
  - *Process of Innovation*
- System Development Life Cycle (SDLC)
  - *Methodology*
  - *Methodology #1: Waterfall (Construction)*
  - *Methodology #2: Lean/Kanban (Repetitive Manufacturing)*
  - *Methodology #3: Scrum/Agile (Software Development)*
  - *Key Takeaways*
  - *Comparing SDLC with Organize or Die*
- Database History: Paper-Based, Hierarchical, and Network
  - *Insertion/Deletion Anomalies*
  - *Inefficiencies with Hierarchical Model*
  - *Breakthrough to a New Data Model*
- Flaws and Limitations of Spreadsheets
  - *Security*
  - *Consistency*
  - *Scalability/Automation*
  - *Querying Data*
- Conclusion
- Post-Chapter Challenges

2. Brief Overview of Relational Model
- Introduction
- Structure
- Hierarchical Database: A Quick Review
  - *Relational Models: `METRO_TRANSIT` and `MUSIC_STREAM`*
  - *Initial Explanation*
  - *Primary Key*
  - *Foreign Key*
  - *Relationships*
  - *Cardinality*
  - *Crow's Foot Notation*
- Data Types
  - *Categories of Data Types*
  - *Numeric Data Types*
  - *Whole Numbers and Integers*
  - *Numbers with Decimals*
  - *Exact Data Types*
  - *Approximate Data Types*
  - *Money Data Types*
  - *Date and Time Data Types*
  - *String Data Types*
  - *Fixed-Length and Variable-Length*
  - *Other Data Types*
  - *Advantages of Relational Model*
  - *Disadvantages of Relational Model*
- Conclusion
- Post-Chapter Challenges

3. Data Modeling and Normalization
- Introduction
- Structure
- Systems Development Lifecycle
  - *Overview of `METRO_TRANSIT`*
  - *Method 1: Focus on Each Object with a Vertical Design Approach*
  - *Method 2: Focus on the Puzzle with a Horizontal Design Approach*
  - *BRAINSTORMING*
  - *Common Parent*

Method 1: Focus on Each Object with a Vertical Design Approach
Defining Relationships
Multiplicity
Awkward Primary Keys
Yes/No Questions as Column Names
Conclusion
Post-Chapter Challenges

4. Learning About Normalization
Introduction
Structure
A Closer Look at Normalization
Two Methods of Conducting Normalization Process
Method 1: Relational Calculus
Basic Rules for Normalization
Unnormalized Form (UNF)
Example Normalization: Going from UNF to 1NF
First Normal Form (1NF)
Second Normal Form (2NF)
Third Normal Form (3NF)
Fourth Normal Form (4NF)
Method 2: 'Ironing' a Database from UNF to 4NF
Database Design Patterns
Relationships Will Change
Resolving M:M
PK/FK Going in the Wrong Direction
Referential Loops
Conclusion
Post-Chapter Challenges

5. Beginning with SQL
Introduction
Structure
Download and Install SQL Server 2025 Express
DDL: CREATE, ALTER, and DROP
DML: INSERT, UPDATE, DELETE
Look-Up Tables Must be Filled First

- *SELECT Statement*
- *Wildcards*
- *SELECT Statement without FROM*
- *NULL*
- *Practice*

Conclusion
- *Questions*

Answers
Post-Chapter Challenges

6. Intermediate SQL Part 1

Introduction
Structure
Shortcuts
- *Shortcut #1: Asterisk (*)*
- *Shortcut #2: SELECT..INTO*
- *Shortcut #3: INSERT INTO…SELECT*
- *Shortcut #4: ALIASES*
- *DISTINCT*

JOIN Statement
- *INNER JOIN*
- *Multiple JOIN Statements in a Single Query*
- *OUTER JOIN*
- *LEFT JOIN*
- *RIGHT JOIN*
- *SELF JOIN*
- *CROSS JOIN*
- *Multiple Conditions in JOIN*
- *Functions*
- *String Functions*
- *CONCAT()*
- *LEFT() and RIGHT()*
- *LEN()*
- *LOWER() and UPPER()*
- *TRIM(), LTRIM(), and RTRIM()*
- *REPLACE()*
- *STR()*

SUBSTRING()
CAST()
Numeric Functions
ABS()
ISNUMERIC()
MOD() and %
SQUARE(), SQRT() and POWER()
RAND()
NULL Functions
Aggregate Functions
AVG()
MAX()
MIN()
COUNT()
COUNT() versus SUM()
SUM()
Common Mistake with SUM()
Other Aggregate Functions
Conclusion
Post-Chapter Challenges

7. Intermediate SQL Part 2
Introduction
Structure
GROUP BY Command
HAVING
ROLLUP
CUBE
GROUPING SETS
CAUTION: Incorrect Interpretation of Query Results
DATE Functions
Absolute versus Relative Dates
DAY(), MONTH(), YEAR()
More on Relative Date functions
DATEADD()
DATEDIFF()
DATEPART()

DATENAME()
Subqueries
Correlated Subqueries
Subquery in SELECT line
Subquery in WHERE Clause
Basic Temporary Objects
Derived Table
View
#Table (read 'temp table')
Conclusion
Post-Chapter Challenges

8. Putting it All Together
Introduction
Structure
Industry Challenge: Retail Coffee Shop
Steps for Establishing a Fully Normalized Database
Read Case Study
Conduct Additional Research
Select a Perspective
Establish User Requirements
Create Data Flow Diagram
Create Conceptual Database Diagram
Create a Logical Database Design
Create Physical Database Design
Manually Populate Database
Run the Entire Script as Single Executable
Write Initial REPORTING Queries
Conclusion
Post-Chapter Challenges

ADDENDUM Case Study: Quick-Serve Retail Coffee Shop
Introduction
Structure
Industry Overview
Business Challenges
Employee Burnout in the Retail Industry

Data Needs
Major Companies
Market Size
Popular Products and Merchandise
Profile of a Typical Customer
Typical Jobs
Significant Innovations
Future of the Coffee Industry
Overview of Starbucks Coffee
Mission Statement of Starbucks
Reasons behind Starbucks' Success
Data Needs of Starbucks
Challenges and Goals of Starbucks
Pleasing Starbucks Investors
Employee Relationship Challenges
Starbucks Global Supply Chain Challenges
Additional Resources
Conclusion

Index

CHAPTER 1

Organize or Die: Learning from Data

Introduction

Hello, and welcome to Ultimate Introduction to SQL for Data Analytics! While data is practically omnipresent in today's vast technological landscape, few of us understand how to effectively establish the systems that answer complex questions through thorough analyses. This lesson will be our introduction to the history of data as well as the best practices of modern data collection and retrieval. We will also review the organizational and structured processes that allow for extended learning for any individual or many enterprises from around the world. This book aims to introduce relational theory, database design, and the Structured Query Language (SQL) to prepare readers for their respective professional journeys as an elite analyst or data scientist.

By the end of this chapter, readers will be equipped with several key thoughts that will frame how and when they seek additional data. These include an understanding of the progress advanced societies have made with organized learning and computerized databases as well as a better grasp of the limitations of data by itself and how most organizations need the context of data to color the information they rely on.

As this approach is presented, consider artifacts we use today that are evidence of learning by previous generations. For example, standard processes like the Systems Development Life Cycle (SDLC) are widely used today for a very good reason: they work!

Please also recognize the human passion to learn about and understand almost everything around us combined with making every day activities more efficient. We try to innovate and optimize repetitive tasks across all industries and cultures. Finally, apply this attitude of leveraging data to optimize performance as you develop database skills. Harnessing learning at tremendous scale, speed, and breadth will be the greatest asset of your future.

Structure

In this chapter, we will explore the following topics:

- Organize or Die
- Information Ladder: The DIKW Pyramid
- Data in Ancient Societies
- System Development Life Cycle (SDLC)
- Historical Database Systems
- Flaws and Limitations of Modern-Day Spreadsheets

Organize or Die

It is a human streak observed through history that we aim for personal recognition as well as when aligned as a team in professional organizations. However, we also seek better ways to get similar or greater output with the same or less effort. This has been proven many times during the evolution of ancient civilizations, inventions across industry, as well as in mundane activities like team sports.

People have often sought a competitive advantage in repetitive daily activities through the innovation of tools and optimization of processes. This is the reason that we use the adage of 'organize or die' to reflect the urgency of leveraging for innovation, while dwelling on the human zeal of increasing efficiency, productivity, and profit in professional undertakings. Although this might be daunting, the terms illustrate the roadmap for being assertive in making improvements to data organization and management processes.

To explain further, the 'organize or die adage' also finds a place in Charles Darwin's concept of 'natural selection' (*On the Origin of Species*, 1859). To simplify it, Darwinism postulates that each environment has a set of consistent conditions that define it. For instance, a desert in South Africa is different from the frozen tundra of Siberia or a dense rainforest in Brazil. Since the material and geographical conditions of each area are different, it only makes sense that the endemic species of flora and fauna will be different as well. These species are able to survive through specialized adaptation occurring over many years.

Furthermore, as conditions change in any environment, some characteristics determine a higher probability for survival moving forward. Over time, all the species must adapt via mutation or willful adoption of competitive practices to propagate to their progeny.

However, this book is not about human evolution as these examples are intended to get readers to recognize the following aspects, which will be instrumental in understanding why we manage data in the ways we do:

- Competition for resources 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 are always changing; therefore, we must re-learn how to win.
- New competitive characteristics will prevail as market conditions evolve.
- Data has been leveraged to determine competitive actions for a long time.
- The modern economy relies heavily on data to get ahead on new opportunities.
- 'Organize or die' is knowing when and how to adapt to changing market conditions.

It can be argued that our common interests control the factors in our lives and allow us to succeed. When stripped down to its most basic elements, individuals and our collective societies have two simple goals: producing the goods or services that enable us to provide for ourselves and our closest associates, increasing the probability of survival.

This is a strong parallel between ancient human development and survival in exceptionally harsh environments with current corporations of today; each has to compete for limited resources, adapt to uncaring conditions, and react to adversity by inventing tools on the fly. We can learn more about dealing with competition in the current economy if we can understand how to leverage collective resilience to make processes efficient.

There are several objectives or motivations that entice organizations to follow the principles of the philosophy of Organize or Die. These include gaining control of resources to solve immediate or short-term needs, as well as building a set of repeatable processes that allow for improved outcomes and ongoing optimization over time. This has a focus on longer-term opportunities and future achievements. Let's take a closer look at each.

Short-Term or Immediate Needs

The first motivation of most ventures is to define and meet short-term or immediate obligations. The concepts of Organize or Die is paramount as

According to Founders Forum Group (*The Ultimate Startup Guide With Statistics (2024–2025*) | Founders Forum Group), almost 70% of start-ups fail before reaching year five as they are not able to meet short-term production or revenue demands. While the reasons for professional failure can be many, it comes with a stark realization that not being able to build systems or processes to tackle immediate goals does not guarantee survival in a competitive data ecosystem.

Long-Term or Continuous Success

While majority of enterprises fail due to not being able to meet immediate goals, the remaining examples exist as they also cater to long-term goals, and plan for continuous success besides immediate goals.

The pattern has been observed through history and highlights the presence of intrinsic features that allow only a few enterprises to achieve long-term success. Some of these features include the following:

- Efficient spoken and written communication between organizational units
- Presence of organized codes of behavior
- Diversified skills spanning multiple domains
- Generic training as well as special expertise in production methodologies
- A vision prioritizing the goals of the organization
- Adapting through learning and incorporating effective and efficient processes

While competition has always been present in the society, it has prompted organizations and individuals to innovate and succeed. Those with specialized skills demanded by the market will have a distinct advantage over others as they can stay relevant.

Moreover, for thousands of years, societies had to run on limited resources and tedious organizational processes of handwritten notes, oral traditions, and first-hand observations.

As a result, the scope of analyses was limited when compared to modern systems and processes that can assess copious amounts of data. Consider for a moment how brilliant the innovations from just a few generations ago given the limitations they had with communication and technology!

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 and quickly learn from massive amounts of data to find the critical patterns, trends, outliers, and anomalies. Successfully adapting to trends can result in higher productivity, profit, and prosperity. Essentially, being in control of data is now akin to being in control of our environment and success.

The next few sections will explain the process of innovation and the DIKW pyramid. We will also consider parallel examples from history to reinforce how using data to achieve goals has been a constant feature of humanity, despite rudimentary processes.

Information Ladder: DIKW Pyramid

Numerous academic models have defined the differences between data, information, knowledge, and wisdom. These conversations have provided clarity about finding and tackling imperfections within communication as well as helping systems engineers and information architects develop effective reporting dashboards. It benefits prospective data science professionals immensely as it illustrates how analysis fits into a larger system of processes.

The common components in this information hierarchy represent the acronym DIKW, which stands for:

- Data
- Information

- Knowledge
- Wisdom

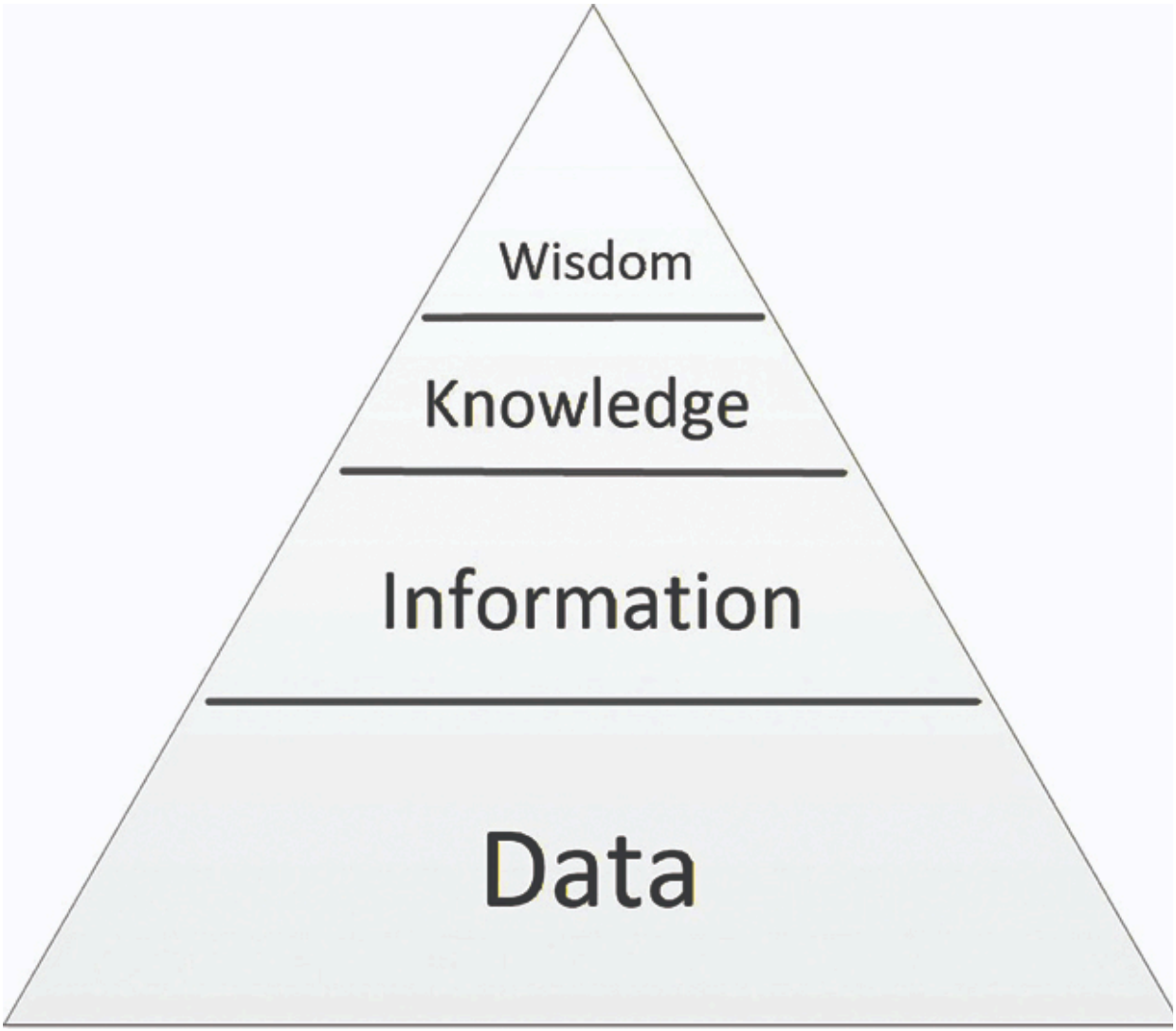


Figure 1.1: *The DIKW pyramid*

These components are ordered as a hierarchy because the subsequent components rely on the previous block. Simply put, wisdom is the result of having broad prior knowledge, and knowledge does not exist without appropriate information. You may also observe that the pyramid is predominantly occupied by data and the other components get increasingly smaller in scale as we move up.

Most systems operate on enormous amounts of data; however, many users might not be aware of how to translate it into actionable insights. The process of learning begins with data but requires adequate documentation

and commitment by users to with it, while paying attention to nuanced indicators, patterns, and trends to gain full awareness of a given scenario. You can look up 'DIKW Pyramid' on the web for more background and perspective.

Data

Data conveys everything we hear, see, feel, and use to document experiences and trends. However, data by itself cannot help us effectively envision the larger picture as these individual packets of information lack perspective and depth without analysis and application. Raw data lacks purpose, definition, or even problem space. This missing context discourages a person from taking any action on raw data.

Think about our reaction in a situation where we are on a city bus and the stranger next to us starts spouting random words all of a sudden (like 'red', 'A24', 'LAX', 'Martha', 'Tuesday', and 'Argentina'). We might decide to avert our eyes or move away from them. If the gibberish continues, we might even exit the ride. This example illustrates how without a relation to broader concepts, we cannot connect disparate data bits into a cohesive idea. This is often frustrating as we struggle to comprehend the intentions of the person we are communicating with.

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

Information

Data that has definition or context allows people to act on it. Let us use the raw data elements from the preceding example ('red', 'A24', 'LAX', 'Martha', 'Tuesday', and 'Argentina') and add context to make them coherent.

It can be, 'My neighbor is Martha; she is arriving at Los Angeles International airport ('LAX') on Tuesday from Argentina at gate A24. She will be carrying a red backpack.' Note the increased context about the

previously unrelated elements of data, which us to visualize the scenario. While you might be less anxious with this information, there may still be a few questions, such as '*So…do you want me to pick her up from the airport? If so, when is she expected to arrive?*' It indicates how despite having more context, one may not know how to react on it. This is where knowledge comes in.

Knowledge

To many people, knowledge is the accumulation of several pieces of information from a variety of mechanisms and acts such as reading, listening to conversations, and observations from living our lives while engaging with others. It can be argued that knowledge is earned through diligently processing the available information. Much of the valuable knowledge people obtain is created through purposeful investigation and discovery; it is not guaranteed by mere listening or knowing.

Knowledge includes understanding the context of how information is produced, processed, and the purpose for which it is presented. 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 refer to a specific meeting for which you are coordinating with someone. Previous information from conversations, meetings, or news might be relevant to their understanding of any expected reactions to the new information. Their knowledge of the context of this information aids any decision-making about the situation. As a result, a person may check the itinerary for exact details of Martha's flight and create a spot in their calendar to pick her up.

As people gain more knowledge in a specific area or topic, they become more effective at processing it. As we gain expertise in a process we are able to break down the sequential steps and even be able to train or instruct others on the nuances of how each component fits in with the whole. Additionally, being knowledgeable about a topic or a process means that we can easily detect inefficiencies in a given process. This is perhaps the most critical for advancing the collective knowledge of a team about a task. By identifying inefficiencies, we can conduct controlled experiments and adapt to challenges and demands. This aspect has propelled markets, industries, organizations and societies towards success.

However, having knowledge does not guarantee any particular outcome. It is important to realize that it is possible to make mistakes even with accurate information and considerable experience.

Wisdom

Finally, we get to the last letter of the DIKW acronym: wisdom. As we know, wisdom lies at the top of the DIKW pyramid. Therefore, it is not only rare, but is the most valuable part of organizing data.

Drawing from the previous example, a person's knowledge of previous trips to LAX might be to contact Martha directly and arrange a meeting spot near the baggage claim. A person scheduled to pick her up will most likely make sure that she has their phone number in case there are any issues. Moreover, you would like to check the traffic reports while heading to LAX and select the most appropriate temporary parking space based on real-time information. Ultimately, the wisest decision to make in this situation might be to get an Uber ride-share for her.

Data in Ancient Societies

The story of data does not begin with the advent of modern technologies, but some 40,000 years ago. While this text does not deal with cultural anthropology, history, or sociology, it elaborates that recording data and pandering to the 'organize or die' adage has been a constant feature of humanity and its tryst for data collection.

As we know, the urge to be competitive and control our environment to 'win' has been a known trait observed in every civilization. Each of such civilizations relied on various forms of data to learn and optimize processes for easing hardships and ensuring survival.

The astonishing success of each civilization mentioned here depended on how they discovered better methods of doing repetitive tasks as well as using infrastructure to reduce risks, enhance protection, and provide for basic short-term and long-term goals. These efforts are akin to the concerns faced by many organizations and practitioners in data science today.

Try to imagine the world and harsh conditions early humans had to cope with; no internet, no organized manufacturing, zero engineering schools or elaborate global supply chains. While it was admirable to survive back then,

many innovations of the era are considered extremely basic in the face of the modern developments we have now. However, it will not be wise to ignore these fundamentals that still inform our efforts at organizing data today:

- Written and Spoken Language
- Division of Labor
- Detailed Administrative Structures
- Organizational Learning
- Common Defense Architectures

Out of these, language has been the single greatest invention ever. The power 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.

Language enabled the quick transfer of information as well as organized and prescribed knowledge. This means people could begin to understand the world beyond their limited lived experience. They were able to gain insight into the best practices for 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 earlier by understanding the mistakes and the valuable experience of their predecessors. Their ability to subsequently innovate and build on the knowledge of their mentors propelled each civilization further in being successful.

Specialized skills allowed for nomadic groups to establish permanent settlements near fertile river floodplains. These flood plains were rich in minerals and produced a range of edible vegetation, which reduced 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.

When it comes to the division of labor, individuals were able to gain specialization in a skill as opposed to being trained for generic tasks. In a way, developing expertise helped people deconstruct technical processes and determine which steps are inefficient. Understanding flaws and inefficiencies led to experimentation and continuous optimization. These principles are

still relevant in the world of data and technology. With the help of computers and Artificial Intelligence (AI), we can complete several cycles of development, analyses, and optimization more quickly.

Once these early civilizations became sedentary, developed a surplus of food, and specialized skills, they expanded 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. Each of these structures was intended to improve the outcomes of the collective civilization and increase their strength and resilience.

Societies with these characteristics were able to reduce the risks associated with survival and ultimately prospered. This is important to consider as database design draws from the same principles of constant learning and innovation. Each civilization captured data, processed it into relevant information, and created several best practices that provided a blueprint for efficiency and success. All of this expanded the collective body of knowledge and made each society exceptionally well-equipped to embrace their unique challenges.

Systems

Now that we have established the timeline of human innovation and the drive to excel, we are ready to dive into database design, SQL, and analytics. No database or SQL discussion is complete without knowing how ‘systems’ tie everything together. Briefly put, a system is an established structure that has a desired outcome; it is a set of processes that are organized sequentially to achieve a goal. These sequences of events, tasks, and steps collectively act as a larger cohesive unit. The purpose of a system is to provide efficiency in a repetitive process to generate better outputs with limited input resources. More sophisticated systems have built-in steps to measure progress, validate status, gauge the effectiveness of the work completed, and verify if the desired outcome is still viable.

Ultimately, the value of any system is how well it can simplify or organize a repetitive process. Again, this is achieved by breaking down each process into many smaller tasks or steps that are connected to provide a solution that saves time and money. Perhaps the biggest benefit of implementing a system

is learning. To illustrate these points, let us explore some examples of simple and complex systems.

Example System/Repetitive Process #1: Farming

Agriculture perhaps is a fine example of a complex system that benefits from having a sequential structure of organized steps.

Desired Outcome: Produce a crop of edible vegetables like to feed family and/or sell at a market if there is any surplus.

Sequential Steps:

1. Locate available arable land.
2. Decide on appropriate crop(s).
3. Plant seeds in the ground and tend to them.
4. Harvest the produce after it has grown.

In this example, 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 critical to produce viable crops. The steps however are very simplified and many innovations must be learned through conversations, observation, and lived experience. Essentially, any sophisticated process like farming can be reduced to an unrealistic and simple form by an uninformed analyst.

Let us 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. We are programmed to experiment while defining routine or repetitive processes, finding inefficient steps, innovating, and measuring outcomes to obtain ‘better’ results. As such, driving to and from work every day is a good example for analysis and optimization.

Desired outcome: Getting to work in a predictable timeframe efficiently.

Sequential Steps:

1. Find quickest route to commute destination according to the GPS.
2. Drive as aggressively as traffic permits without being reckless.
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 some time.
6. Settle into optimal routine with modification only as conditions change.

Many of us spend an inordinate amount of time and effort during our commute to reach at our workplace. When we assume a normal 30-minute commute each way, we might end up spending hundreds of hours each year on a high-stress activity with insufficient planning.

We might consider the following questions when evaluating the two example systems here:

- 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 for any system affect the probability of successfully achieving our desired objectives?
- What might happen as these processes mature and we complete the processes several times?

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 of the complexity of any process, farming is incredibly complicated with many factors and conditions that span a cycle that is several months long. The number of steps, tasks, and considerations for producing a viable crop can be even more, despite planting resilient crops. On the other hand, while the inputs for driving during a commute are dynamic and can occur multiple times, it can be argued the complexity of farming far exceeds that of driving.

The system of farming represents generational knowledge spread over thousands of years and is complex and sophisticated. Yet, many of us might be tempted to minimize it. This may be the ultimate instance of ‘Organize or Die’.

Let us now compare the system for farming to the system many of us follow to get to work each day. Maybe we can assume that driving to work has little ‘deep’ analysis and relies mostly on emotion and impulsive reactions to perception every few seconds.

According to a simple search on the internet, the average duration for an American citizen to commute to work is 27 minutes. For illustrative purposes, let us round off this number to 30 minutes. We often drive aggressively during a typical day to save minutes; however, the 'saved' minutes from our commute are lost during a typical work day. Consider that even one missed elevator, canceled meeting, or being placed on hold during a phone call negates everything we have saved during the commute.

Aggressive driving is an example of the innate desire to 'win' and might also include:

- Choosing alternate routes to avoid paying tolls
- Switching lanes numerous times to 'win' even just a few car-lengths
- Cutting in the line 'unfairly' during a merging scenario
- Driving through parking lots or residential neighborhoods to get past slower traffic

The above characteristics do not include 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 such as:

- Double-parking in a turn lane
- Truck loading zone
- Someone else's reserved parking
- Disabled parking

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

How does writing down the sequence of steps for any system affect the probability of successfully achieving our desired 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 ensuring the reliability of outcomes. 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 that 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; they may also include facilities, equipment, personnel, and training. A system is quite often a work in progress. To be most effective, it should represent the best-known steps and reflect the current operational knowledge based on evidence.

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

In the case of the steps for farming that are included in example #1 above, it is woefully inadequate to understand a complex process. Extra steps 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 can be added to enrich the process.

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

A system is about improving efficiency, reliability, quality, and predictability. As there is no such thing as a perfect system, each 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 expertise. 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 lucky guesses in the middle of well-established processes.

Again, a system can be complex or simple. While we can recognize that whatever works for a particular situation is often the rule, we can acknowledge that no system is perfect and every system has a lifespan of effectiveness. The window might only last several months or perhaps centuries. This means that every system can be improved or optimized. While there may be a slight inefficiency at any stage, it may not be a sound investment of time, money or other resources. We usually avoid taking on system improvement projects until the benefit of spending resources on the overhaul is significantly greater than the cost of maintenance.

The point of these examples is twofold:

- People are always innovating or seeking new ways to gain efficiency (often based on impulse or flawed perception).
- Inefficiencies can be discovered in nearly any process by analyzing data.

Like data, systems exist at various levels of complexity. Data science is a system too, however, it should be viewed as a critical component in several larger and more complex systems.

Systems Analysis

No system is perfect! Each system might have inefficient steps; the only question is whether the cost of inefficiency exceeds the cost of fixing it. In other words, after spending time and money to make a process more efficient, do we get our investment back through greater competitive advantages achieved in the future?

Every generation, civilization, and organization has benefited from analyzing their systems of production and making improvements. It is related to our need to be efficient and 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 have been several that were attempted and tried, that failed to provide value. Consider 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, roasted and crushed them into a fine powder before soaking them in hot water and drinking the concoction. 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.

When Greek philosopher Plato stated that ‘necessity is the mother of invention’; it meant that 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 might have been millions of tools developed to make a

process quicker, easier, or more effective. Just looking around a work desk at 2:00 in the morning on this random Tuesday, dozens of tools are visible: speakers, coffee cups, stapler, batteries, cell phone, cat tree, pens and paper. Each of these items was ‘invented’ at some point to create a benefit that did not exist previously.

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 many different combinations trying to avoid red lights, tolls, or heavy traffic before figuring out 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 today’s competitive economy, finding a definitive “better way” often requires a plethora of documentation to support analyses. Gone are the days of 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 an organization’s ability to predict or control costs. Identifying the biggest problem step in a complicated process is not easy, but let us assume that with enough experience of engaging any process we can break down each separate component by sequence and dependency.

The simplest processes involving data include a record of what was attempted for a particularly flawed process as well as the ensuing results that were deemed inefficient or undesirable. Prior to recorded history, these observations had to be shared orally. Today, we might categorize the documentation into smaller chunks by establishing end-result goals, articulating step-by-step processes, analyzing each step, determining inefficiencies or flaws, experimenting with alternative steps, and finally implementing the best step after observing the impact.

We will build further concepts on your existing understanding of data and the mechanism that is used to contain and characterize it. 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. Note that the entire collection of steps outlined is iterative and leads to continual process improvement.

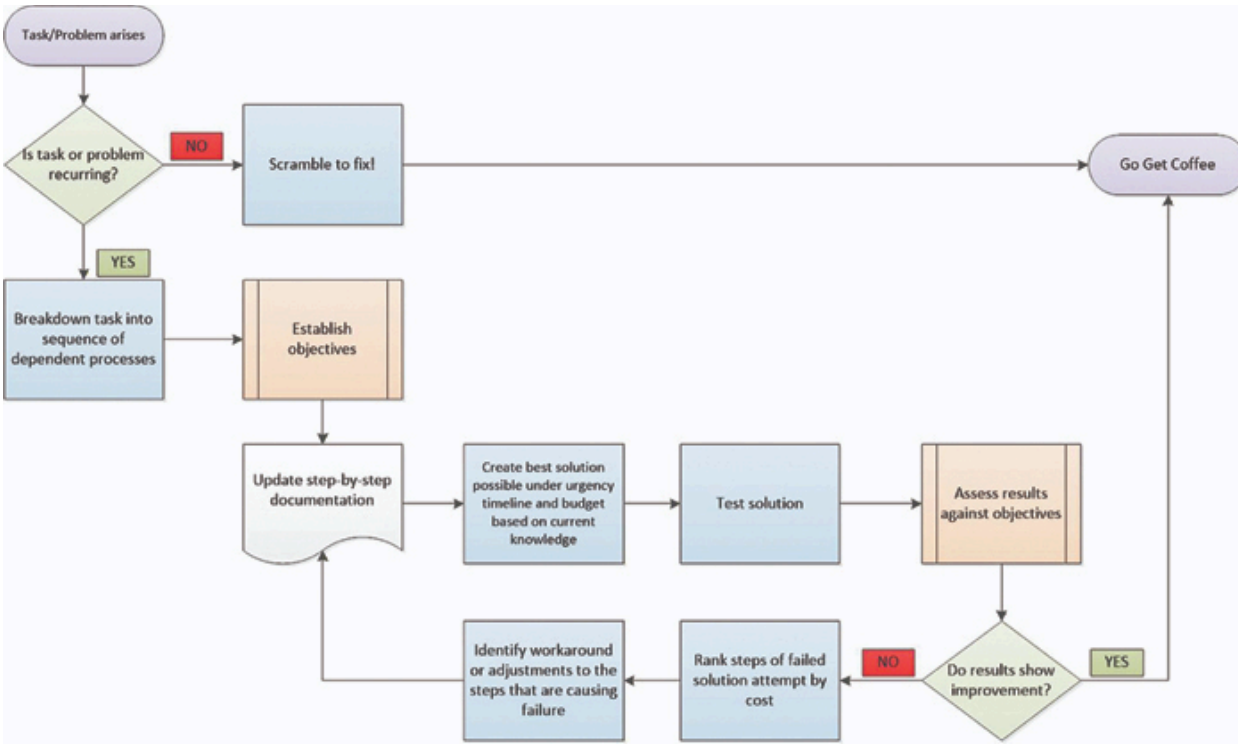


Figure 1.2: *Innovation as a workflow diagram*

Documentation of Steps in a Process: To simplify the diagram, a system is made up of components or steps that are sequential. If we are to analyze an existing process, the first challenge is to document the big steps of how things are done. The next few tasks are to determine the sequence of these steps. We will then document how and when it is known that a step is completed or is 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 might want to observe an actual execution of the process running in real-life, while 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 achieved the desired objective and which did not. During this observation, 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 that they align. Additionally, document the amount of time and resources such as people, money, and the equipment each step might require.

Identify Least Efficient and Costly Steps: After a few cycles of going through a particular process completely, we can measure 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 experimenting 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 for the next experiment.

Formally Adopt Innovation as a New Step: Throughout human history, our predecessors have followed standard systems and process improvements very similar to modern ones for centuries. 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 of their processes can continually improve the quality of goods and services they provide.

Hence, this practical learning adequately captures the complexity of human behavior and our common need for winning control of scarce resources. It could be argued that nearly every single discovery, invention, or optimization in history was the result of innovation. While I am sure more

than a few significant innovations have happened by pure accident, most are from someone thinking about a better way to do things.

As we search for a competitive advantage through innovation or optimization, we must realize that we might end up with many discoveries from the data we are collecting. Innovation does not end; there is much more coming up in several industries. In other words, in an intensely competitive world, experts and professionals with specialized skills will have a formidable and distinct competitive advantage over those who lack them.

We will conclude this section with some thoughts on recognizing how critical systems have existed throughout human history. Firstly, systems continue to be as critical as before in our data-driven economy. While the advent of electronic systems has been pretty recent, 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 more reliant on valuable insights from data. As a result, data lies at the heart of most industries.

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

System Development Life Cycle (SDLC)

The **systems development life cycle (SDLC) is an important framework that provides a** high-level overview of the various phases encountered by development teams as they construct a system from scratch.

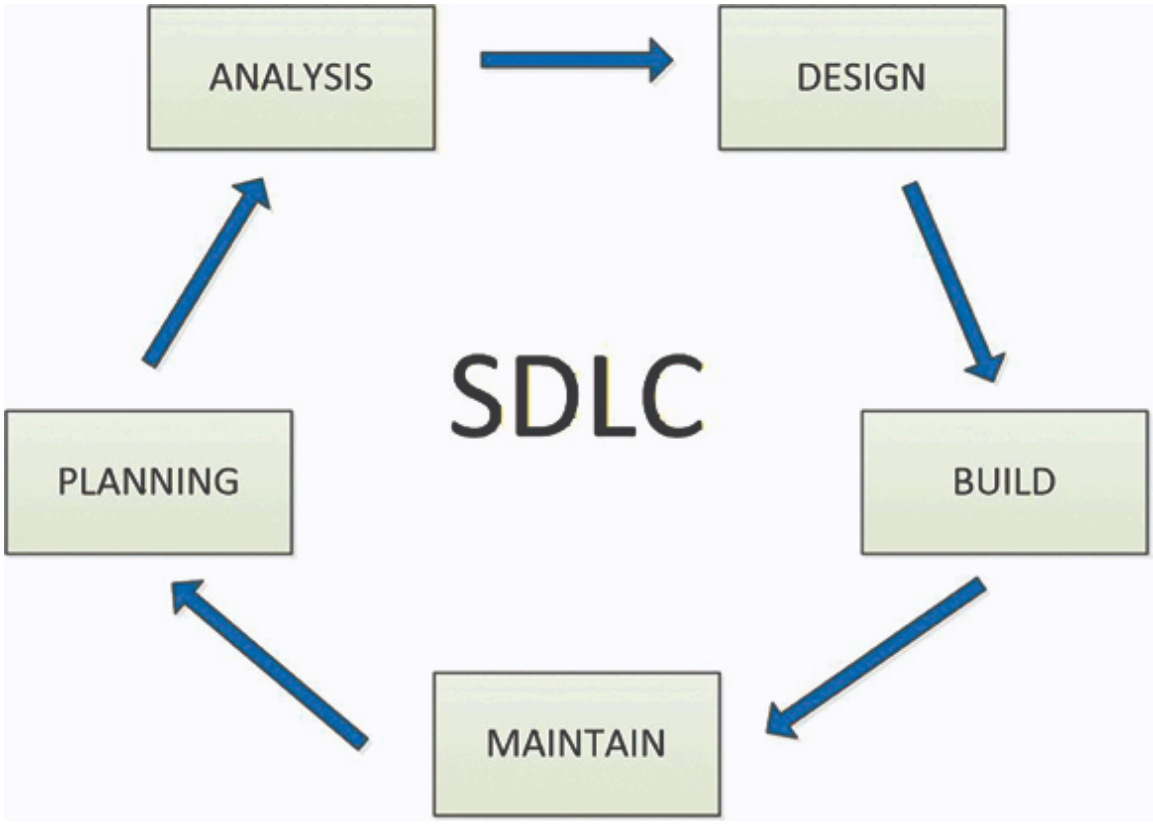


Figure 1.3: *Iterative phases in SDLC*

Therefore, it is important to understand why we use it, how it is followed, and how it is different from a methodology. Firstly, we use SDLC as a template to increase the probability of the success by the end of a project. Success refers to the timely delivery of features addressing user requirements within budget. 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 prepared to tackle hurdles when the project hits delays. Basically, this template of steps when building a system provides better information and leads to more opportunities for learning. This means that as a team iterates through multiple projects and therefore has multiple cycles of traversing the SDLC framework; this results in improved experience and anticipation of challenges. Due to SDLC, professionals become more flexible and confident in tackling invariable project glitches.

The SDLC is a template of phases, and consists of planning, analysis, design, implementation, and 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 from a cost/benefit perspective. It is important to remember that only a handful of projects are funded. Each organization must make sure that a project is going to eventually pay for itself by generating additional revenue or reducing expenses.

Once a project is approved, the analysis phase begins. This phase is where the current system is explored and 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 construction industry defined 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 prominent factors in the construction industry are the reluctance at issuing changes due to high risk and incurring the subsequent cost associated with them.

Methodology

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

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

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 downstream bucket. This methodology is known for being exhaustive, thorough, and somewhat slow. However, it is good for implementing the SDLC framework for industries that have many inflexible, deliberate, and dependent requirements which need approval or validation. This methodology is also good for industries that do not need nimbleness around rapid customer interaction; this means that 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 behind it are the changes that are exceptionally expensive or even prohibited. Imagine having built the third floor of a new apartment building designed to be four stories in height. A customer then decides that 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 is then 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, therefore 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 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 allowed 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 of continuous improvement and is worth considering for anyone who is interested in 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 colossal failures of large companies in the technology sector. Before Agile, software development teams often managed projects through the waterfall methodology, which was deliberate, precise, and slow. Moreover, changes to agreed-upon features are difficult to incorporate and customer interaction is diminished. The process often took three 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 deliver only a handful of features in a span of mere 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 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. Development companies win because they have happier customers and generate revenue quickly, with less bloat and risk. The coders

win because they can specialize, while finding interesting and challenging work that makes them feel valued.

Thus, Agile is the dominant methodology in the technology industry. Take the time to become familiar with this methodology if you intend on becoming a data science professional.

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) while identifying mistakes as well as successful corrections. This repetitive reflection over time allows each interested company better insight and accuracy in the following domains:

- Planning (Budgeting)
- Spending
- Task Management (Recognition of Dependencies, Prioritization)
- Avoiding Problematic Situations
- Troubleshooting
- Reacting Quickly and Efficiently to Emerging Issues

Again, each industry (and to a lesser extent, each organization) can develop a unique methodology tailored to their culture and approach to delivering their goods. This is accomplished by capturing data throughout the lifecycle of many projects and building a body of knowledge which analysts can learn from. This repetitive experience allows for continual improvement, establishing ‘best practices’, nurturing of specialists for each process, as well as being better able to on-board or train new employees 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 for the process. The sooner a risk is identified; the sooner it can be corrected. Future projects can be informed from the work and projects done previously, which allows the organization to make budgetary and timeline estimates more accurately.

This is an opportunity to iron out any confusion you may have about the relevance and application of 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.

Comparing SDLC with Organize or Die

Now that we have been exposed to "Organize or Die", as well as SDLC, let us compare the two. Organize or Die is 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 responds to "Organize or Die"; it is a method for becoming 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 to keep track of things. After World War II, the overreaching industrial goal was to automate a lot of

paper processes. With the advent of **relational theory in the 1970s**, we have come a long way.

Thus, new data analysts need to put data collection into context to gain competitive advantage and foster a culture of innovation. Data is at the core of all these motivations and thus needs extensive documentation of goals and objectives, step-by-step processes of each objective task, the results of these processes, as well as our observations and analyses of outcomes in relation to envisioned goals.

As we saw previously, for thousands of years, data had been managed manually with pen and paper. While this was easy to implement, it had obvious limitations in security, durability, and the ability to digest at any valuable scale.

These limitations when combined with the competitive nature of post-World War II enterprises led to the emergence of opportunities to gain significant advantages through computerizing processes. The primary objectives at this stage 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 was essentially no other way to manually organize physical items. We had to group physical items in similar clusters all the time.

Computers were effectively a new tool in midst 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 was structured in a top-down or hierarchical manner. 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. We realized that this rigid structure was difficult in operation when applied to real-world business scenarios. Not every relationship was hierarchical (strictly defined as one-parent only), especially when tracked over a fifty-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 caused designers to organize the hierarchy in the shape of an instantly readable report with all the data 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 many copies of similar data to artificially inflate the overall size and 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 it happens.

Insertion/Deletion Anomalies

The hierarchical model has insertion and deletion anomalies, which 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. Similarly, a deletion anomaly occurs when a parent record is set to be deleted, but there exist one or more child/dependent records underneath it that are still considered valuable, and not desired to be deleted.

These challenges are difficult and often affect how companies conducted business operations. Many times, companies force their practices into a rigid hierarchical model. An example can be how a delivery driver working at a pizza chain cannot work a shift at a different store in the same area, despite the outlet being operated by the same enterprise. In the pizza chain's hierarchical database system for payroll, employees are hired at a store and not by the company itself. It often means that the company would have to hire the same worker again if they are asked to cover a shift at a different location than their home store. This would also mean that the company will file official paperwork with the State of Washington and Internal Revenue Service (IRS). Since this was considered too much of a hassle to do things the right way back in the '80s, managers loaned their cars to workers for covering the shifts. These workarounds were common at that time as many

companies struggled with inflexible database systems that did not align easily with the demands of real-life business operations.

Another detailed example of a hierarchy is a bus transit system where a database is organized in a collection of several hierarchies. When we look at these hierarchies, we must consider that they follow a cabinet filing system of order forms, invoices, and paper receipts. In paper-based systems, each original record is 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, as shown here:

- **`Neighborhoods`**
- **`Drivers`**
- **`Vehicles`**
- **`Routes`**
- **`TRIPS/SCHEDULES`**

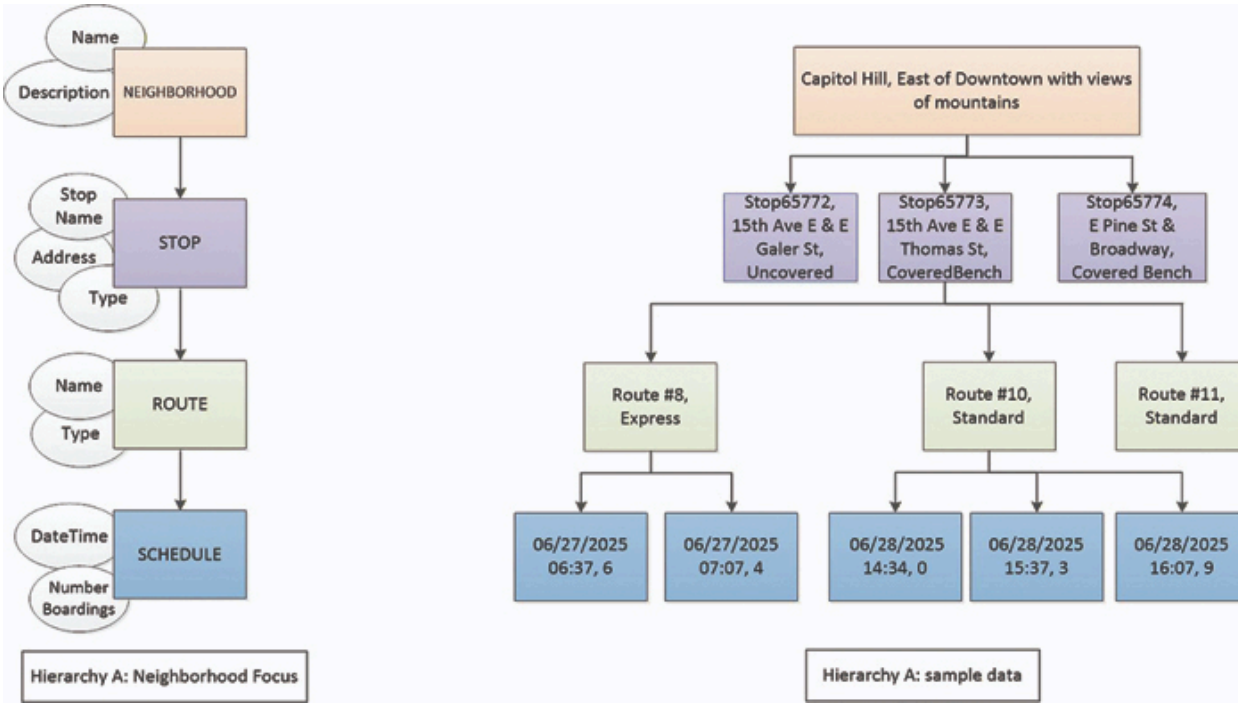


Figure 1.4: *A hierarchy of neighborhood trips for METRO_TRANSIT*

Let us see another example from **`METRO_TRANSIT`**:

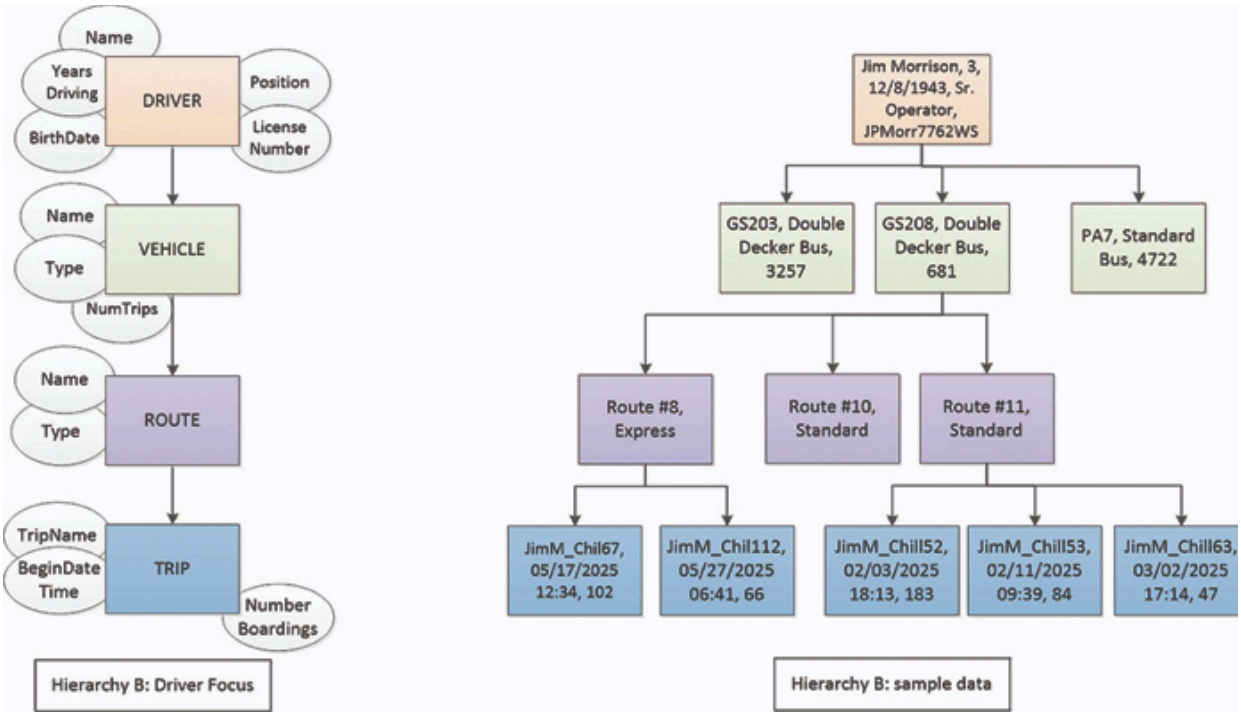


Figure 1.5: *Hierarchy of driver-centric data for* `METRO_TRANSIT`

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 bring an umbrella if the bus stop is uncovered in case of rain. 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 because they take up more storage space than reference and how they introduce the possibility 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 these values are easy and convenient for a user to read if they are seeking data regarding a single route, they 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 that 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 that 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 upon which hierarchy is engaged.

The 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, they first recognize which perspective or focus they need to adopt, 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.

Another example of a hierarchical structure could be how we track music from an online streaming service (services such as Amazon Music, Spotify, and Pandora come to mind). We may want 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! This is shown as follows:

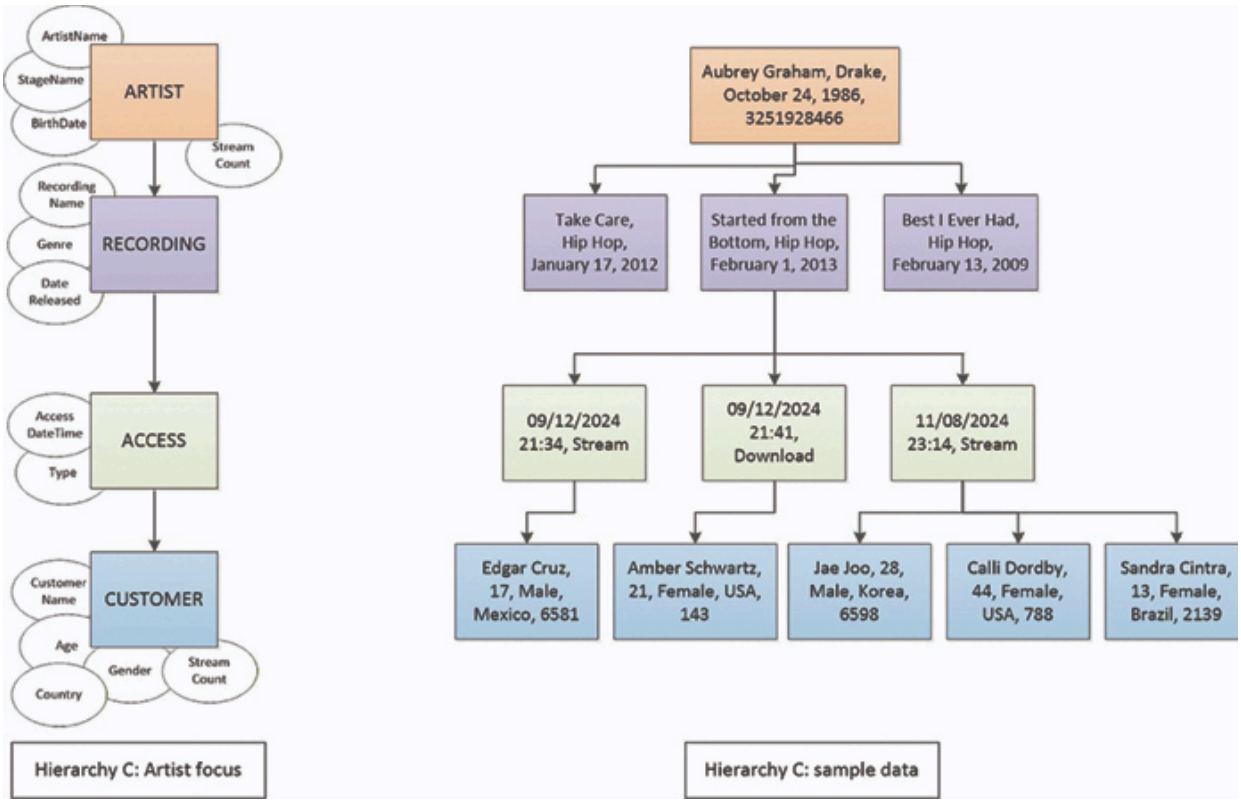


Figure 1.6: *Example hierarchy of artist-centric data for* `MUSIC_STREAM`

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 significant time to traverse each possible combination of the higher-level values, perhaps with hundreds or even thousands of iterations. Please notice the following 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 the usual duration).

- Redundant data requires detailed understanding of all hierarchies as well as a manual process to validate integrity throughout the group.
- A mistake of either missing an update or deletion can result in inaccurate data; this means that different answers are returned from the system depending on where the question is searched.

These alternate structures create redundant values that effectively double or triple data storage demands. More than being bloated, slow, and expensive, these duplicate structures require extra care to maintain accuracy. Again, each insert, update, and/or delete has to be performed in each hierarchy, which significantly slows each operation. Unfortunately, over a period of months and years, many operations are not completed in each hierarchy as required, which introduce several inaccuracies. The inaccurate data can lead to mistrust about retrievable data, which results in many people resorting to managing data offline and even retreating to paper-based systems.

Additional flaws of the hierarchical database model are effectively having computer systems dictate limitations to the business, where tasks or actions that did not fit neatly into the established set of hierarchies are not 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. The rigid hierarchical database designs of the 1980s did not allow drivers to be assigned a bus, route, or trip dynamically. The result was that a business had to rehire the driver for a single trip, thereby duplicating their personal data and creating downstream confusion or misinformation when it was not cleaned up properly in the computer system.

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 resulted in 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 the previous flaws for collection and retrieval of large data sets and remained unchanged for over 50 years. 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. This has changed every aspect of our lives from how goods and services are manufactured, delivered, and consumed throughout the world.

Flaws and Limitations of Spreadsheets

This topic explains when we can think of data management in terms of an easy-to-use spreadsheet and when to embrace more complicated relational theory. Simply put, early in the data management process, while capturing the creation of original transactional data for long-term storage, we will not work with it like we do with a spreadsheet. A spreadsheet is best for ‘last mile’ analysis of data and perhaps making projections or visualizations of potential scenarios. When designing robust data storage systems with millions of rows of data, it is important to not treat data as lines of text; instead, we need to treat it as the input for machines that will be processed and stored with great sophistication.

While Excel might be the most popular software application ever, we need to curb our excitement around using it to solve every data-related problem we encounter. Excel has its place, but it is time to venture into the realm of relational theory while being aware of its limitations.

Spreadsheets are perhaps the single most common computer application in use across the world after internet browsing and email. While they are tremendously valuable to organize simple tasks, estimate costs, or calculate future values, there are significant limitations that curtail their effectiveness 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 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, there may be insufficient protections and security concerns 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 that we must be diligent about how and when we share sensitive data. Additionally, not all data in a spreadsheet can be transactionally sound.

Consistency

The brilliance of spreadsheets lies in their flexibility and ease of use. Unfortunately, the integrity associated with transactional data is difficult to enforce in a spreadsheet. This ability to overwrite data on a whim may be considered a feature for ease of use when we are exploring data with 'what if' projections, but it introduces data anomalies and inconsistencies. The hallmark of relational theory and the role of database management systems to maintain consistency across hundreds of tables and many hundreds of columns is therefore, paramount.

Scalability/Automation

As stated previously, spreadsheets are fantastic analytical tools, but 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, and 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.

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

Here is an example of a transportation system in the form of a table that that contains example data which specifically tracks passengers who have boarded various vehicles and paid fares.

While reviewing the data in the spreadsheet, consider if it will be easy to read and learn from it.

DateTime	Passenger	RouteName	RouteNumber	RouteType	Driver	StopName	StopType	Destination	BusType	Fare	Neighborhood
2/13/25 7:36	Ivey Hazekamp	Capitol Hill-Downtown	32	Regular	Jimi Hendrix	Broadway Avenue and Cherry Street	Covered	Downtown	Articulated	$4.50	Capitol Hill
2/13/25 9:36	Daceel Eustache	Fremont-Waterfront-Downtown	78-S	Special	Meryl Streep	Hwy 99-N 36th	Covered	Downtown	Extra-long	$2.75	Fremont
2/14/25 6:36	Daceie Eustache	Fremont-Waterfront-Downtown	78-E	XP	Bruce Lee	Elliott Avenue and Mercer Street	Covered	Downtown	Extra-long	$2.75	Interbay
2/16/25 6:32	Kenyetta Terron	Sodo-Downtown Express	42-E	Express	Meryl Streep	First Avenue and Terry Street	Covered	Downtown	Doub	$2.75	SODO
2/19/25 15:39	Kenav Terron	Sodo-Downtown Express	42-E	Exoress	Bruce Lee	First Avenue and Terry Street	Uncovered	Fremont	Doubled	$4.50	South Downtown
2/21/25 6:13	Daceel Eustache	Fremont-Waterfront-Downtown	78-S	Special	Jimmy Hendricks	Elliott Avenue and Mercer Street	Cvd	Downtown	Extra-long	$2.75	Interbay
2/21/25 7:36	Kenyetta Terron	Sodo-Downtown Express	42-E	Express	Bruce Lee	First Avenue and Terry Street	UC	Downtown	Doubled	$4.50	South Downtown
2/22/25 8:32	Ivey Hazekamp	Capitol Hill-Downtown	32	Regular	Jim Morrison	Broadway Avenue and Cherry Street	Covered	Downtown	Articulated	$4.50	Capitol Hill
3/1/25 6:33	Ivey Hazekamp	Capitol Hill-Downtown	32	Regular	Bruce Lee	Broadway Avenue and Cherry Street	Covered	Downtown	Articulated	$4.50	Capitol Hill
3/1/25 6:36	Ivey Hazekamp	Capitol Hill-Downtown	32	Regular	Meryl Streep	Broadway Avenue and Cherry Street	Covered	Downtown	Articulated	$4.50	Capitol Hill
3/3/25 6:42	Daceel Eustache	Fremont-Downatown Commuter	70	Commute	Jim Morrison	Sixth Avenue and Battery Street	Regular	Fremont	Doubled	$4.50	Downtown
3/5/25 6:32	Daceel Eustache	Fremont-Waterfront-Downtown	78-E	XP	Meryl Streep	Elliott Avenue and Mercer Street	Covered	Downtown	Extra-long	$2.75	Interbay
3/9/25 7:52	Kenyetta Terron	Sodo-Downtown Express	42-E	Express	Jimi Hendrix	First Avenue and Terry Street	UC	Downtown	Doubled	$4.50	South Downtown
3/10/25 6:33	Daceel Eustache	Fremont-Downatown Commuter	70	Commute	jim	Elliott Avenue and Mercer Street	Covered	Downtown	Extra-long	$2.75	Interbay
3/11/25 6:32	Kenyetta Terron	Sodo-Downtown Express	42-E	Express	Bruce Lee	First Avenue and Terry Street	Uncovered	Downtown	Doubled	$4.50	South Downtown
3/13/25 8:32	Ivey Hazekamp	Capitol Hill-Downtown	32	Regular	Bruce Lee	Broadway Avenue and Cherry Street	Covered	Downtown	Articulated	$4.50	Capitol Hill
3/14/25 6:32	Ivey Hazekamp	Capitol Hill-Downtown	32	Express	Bruce Lee	Broadway Avenue and Cherry Street	Covered	Downtown	Articulated	$4.50	Capitol Hill
3/15/25 1:36	Janey Lundgren	Capitol Hill-Downtown	32	Reg	Jimmy Hendricks	Broadway Avenue and Cherry Street	Covered	Downtown	2 Decks	$2.75	Capitol Hill
3/16/25 6:32	Daceel Eustache	Fremont-Downatown Commuter	70	Commute	Jim Morrison	Sixth Avenue and Battery Street	Regular	Fremont	Doubled	$4.50	Downtown
3/17/25 11:33	Ivey Hazekamp	Capitol Hill-Downtown	32	Regular	Jimi Hendrix	Fourth Avenue and Seneca Street	Covered	Capitol Hill	Doubled	$2.75	Downtown
3/18/25 16:13	Jane Lundgren	Capitol Hill-Downtown	32	Reg	Meryl Streep	Fourth Avenue and Seneca Street	Covered	CH	2 Decks	$2.75	Downtown
3/19/25 6:32	Janey Lundgren	Capitol Hill-Downtown	32	Reg	Meryl Streep	Broadway Avenue and Cherry Street	Covered	Downtown	2 Decks	$2.75	Capitol Hill
3/21/25 6:32	Daceel Eustache	Fremont-Downatown Commuter	70	Commute	Meryl Streep	Sixth Avenue and Battery Street	Regular	Fremont	Doubled	$4.50	Downtown
3/26/25 9:04	Janie Lundgren	Capitol Hill-Downtown	32	Regular	Jimmy Hendricks	Broadway Avenue and Cherry Street	Covered	Downtown	2 Decks	$2.75	Capitol Hill
4/18/25 5:23	Daceie Eustache	Fremont-Downatown Commuter	70	Commute	Jimi Hendrix	Fourth Avenue and Seneca Street	Regular	Fremont	Doubled	$4.50	Downtown

Figure 1.7: *Example `METRO_TRANSIT` data in a spreadsheet*

Please note that the 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 will be eventually difficult and cumbersome as we add more and more data.

After considering this data for a few minutes, consider the following questions:

- Which route generated the most revenue from fares?
- Which driver had the most trips?
- Which passengers took the most trips to Capitol Hill?

If these questions cause frustration, you are beginning to see why merely eyeballing data is a losing proposition these days. Perhaps, the single-greatest reason companies transitioned from paper-based systems to computers was that organizational learning was severely restricted under a manual process. The few questions that could have been 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. There is probably not enough value or timely insight to discover patterns or trends in behavior, and make 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. We must break free of our reliance on ‘eyeballing’ data, and embrace the power of relational theory to support data-driven decision making.

Therefore, the crux of this book is to help each reader recognize how data is the core of 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 excel than those who do not.

Conclusion

It is a human urge to prioritize efficiency and seek competitive advantages while performing the most mundane tasks. We are habituated to improve

processes, produce more, save time, and prosper. The core of all innovation, discovery, and gains in product quality has always been based on data.

For centuries, innovations were slowly adopted because of manual processes and could only be shared with limited number of people. Nowadays, not only can problem-solving issues include anyone with access to the internet, but learnings can be shared across the globe in mere minutes. Collaboration, experimentation, and learning can occur at lightning speed through digital databases. Anyone with the right technical skills can participate in this exercise and help shape the future.

Next, we will become more familiar with the structure of relational databases and other concepts in this book. Relational databases are the birthplace of data meant for analytical processing. Virtually, every organization creates original data from many daily operations 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 complex relationships and behavioral patterns. When done correctly (and ethically), learning across industries can be truly groundbreaking.

Post-Chapter Challenges

Here are a few challenges for you to try based on the objectives stated in the chapter.

Track 1 (THINK): Data Tourist Seeking Ancillary Awareness

You must spend a total of 10 minutes reviewing the following questions and exploring your thoughts. These questions and your responses will capture the essence of this chapter.

- How important are data systems for societies, organizations, and individuals? Why?
- What are some 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 a few paragraphs in response to each of the following questions:

- Which data is required to track a passenger's travel habits and preferences from the perspectives of the passengers, drivers, and the municipal management office?
- What are the questions a typical passenger may want to ask while selecting an appropriate route to their workplace?
- What are the metrics/measurements to determine whether a route, driver, or stop is 'successful'? How frequently can 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 in data science. Let us 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 to track the metropolitan transit system.
- Assume that the users of your data set want to learn about route safety; begin tracking data on incidents of injuries, assaults, or aggressive panhandling. Which data needs to be captured? Give reasons for your answer.