Accounting Analytics

Mauricio Codesso

2026-03-30

Welcome

Accounting Analytics

A Practical Approach

Excel SQL Power BI

Northwind Traders AdventureWorks Cycles ERPNext Demo Company

Open Educational Resource

Accounting Analytics: A Practical Approach

This is an open educational resource textbook that teaches accounting students to extract, prepare, analyze, and visualize financial and operational data using three tools widely adopted in professional practice. The book is designed for undergraduate and graduate accounting and business students with no prior analytics or programming experience.

What This Book Covers

The book is organized into twenty chapters across five parts. Part I establishes the conceptual foundations of accounting analytics, including data types, data quality, and the relational database model. Part II teaches data preparation, descriptive analytics, statistical modeling, and audit testing in Microsoft Excel. Part III introduces SQL for querying relational databases, joining tables, performing aggregations, and conducting population-level audit analytics. Part IV covers data visualization principles and interactive dashboard design in Microsoft Power BI. Part V integrates all three tools in applied projects spanning financial reporting analytics, cost and management accounting, forensic accounting, and emerging technologies.

Every chapter follows a consistent structure that includes learning objectives, a professional scenario, conceptual narrative with embedded guided tutorials, and end-of-chapter assessments. Applied exercises are organized across three accounting perspectives: financial accounting, managerial accounting, and auditing. Five comprehensive cases, one per part, provide extended multi-tool investigations. The Preface describes the pedagogical design in detail.

The Datasets

Three realistic datasets accompany this book, provided in both Excel workbook and SQLite database formats. Students use the same data across all three tools throughout all twenty chapters.

Northwind Traders is a small wholesale food distribution company with eight core tables. It serves as the primary dataset for foundational chapters due to its simplicity and accessibility.

Adventure Works Cycles is a mid-size multinational bicycle manufacturer with approximately seventy tables spanning sales, production, purchasing, and human resources. It supports cost accounting, production analysis, and purchasing cycle exercises.

ERPNext Demo Company is a full enterprise resource planning environment with a complete accounting module including a chart of accounts, general ledger, journal entries, cost centers, and budgets. It supports financial statement preparation, audit testing, and integrated reporting.

Open Access

This textbook is an open educational resource distributed at no cost. All software tools used in the exercises are free or included with standard institutional licenses. The datasets are freely available in both Excel and SQLite formats. SQLite requires no server installation and runs on any operating system. No student should be prevented from developing analytics skills because of software cost or platform limitations.

About the Author

Mauricio Codesso is Assistant Teaching Professor at Northeastern Unversity, where he teaches courses in accounting, data analytics and conducts research in accounting information systems, information management, and emerging technologies.

Contributing

This book is a living project. Contributions from instructors, students, and practitioners are welcome and encouraged. If you find an error, have a suggestion for improving an exercise, or want to propose a new example, you can open an issue or submit a pull request on the book's GitHub repository. Feedback on clarity, accuracy, and coverage helps make this resource better for everyone who uses it.

If you encounter an error in the text, examples, datasets, or links, please report it on the GitHub Issues page. Issue reports help us correct problems quickly and keep the book reliable for everyone using it.

If you are an instructor using this book in your course and would like to share your experience or adapted materials, please reach out through the repository. The OER model works best when the community around the resource is active.

Citation

If you use this book in your teaching or research, please cite it as:

> Codesso, M. (2026). *Accounting Analytics: A Practical Approach*.

Preface

This book exists because the accounting profession has changed faster than most accounting curricula have adapted. Organizations now generate financial and operational data at a scale that makes traditional manual methods of analysis insufficient. Enterprise resource planning systems capture every transaction, every journal entry, every payment, and every production order in relational databases that contain millions of records. Auditors are expected to test entire populations rather than small samples. Management accountants are asked to explain variances, forecast performance, and identify operational drivers using data that lives in systems they were never trained to access. Financial reporting teams must reconcile, validate, and analyze data that flows from dozens of interconnected tables before it reaches the financial statements. The profession needs practitioners who can work directly with data, and accounting education needs textbooks that teach them how.

This textbook teaches accounting students to extract, prepare, analyze, and visualize data using three tools that are widely used in professional practice. Microsoft Excel serves as the foundation for data preparation, descriptive analytics, statistical modeling, and audit testing. SQL provides the ability to query relational databases directly, joining tables, aggregating results, and performing population-level analysis without relying on pre-built reports. Microsoft Power BI enables the creation of interactive dashboards and reports that communicate analytical findings to diverse audiences. These three tools cover the full analytics workflow from data access through communication of results, and they represent the toolkit that employers consistently identify as most relevant for entry-level accounting professionals (Sledgianowski, Gomaa, and Tan, 2017).

The book is built around three realistic datasets that students use throughout all twenty chapters, progressing from a simple wholesale distributor (Northwind Traders) through a mid-size manufacturer (Adventure Works Cycles) to a full enterprise resource planning environment with a complete accounting module (ERPNext Demo Company). All three are provided in both Excel workbook and SQLite database formats, so students work with the same underlying data regardless of which tool a given chapter uses. The About the Datasets section that follows this preface provides full descriptions of each database, including table counts, functional coverage, and a chapter-by-chapter appearance map.

Why This Book

Several features distinguish this textbook from other analytics resources available to accounting instructors.

The first is its focus on accounting. General-purpose data analytics textbooks teach techniques using data from marketing, healthcare, sports, or other domains. Those examples are interesting but do not help accounting students see how analytics connects to the work they will actually do. Every example, exercise, and case in this book uses accounting data and addresses accounting questions. Students analyze revenue trends, prepare trial balances, test for duplicate payments, calculate cost variances, perform Benford's Law analysis, and build financial reporting dashboards. The analytical techniques are the same ones taught in general analytics courses, but the context is always accounting, which helps students transfer what they learn to professional practice.

The second is its three-perspective exercise structure. Every chapter includes applied exercises organized into three sections by accounting perspective. Financial Accounting exercises address reporting, disclosure, and compliance questions. Managerial Accounting exercises address costing, budgeting, performance measurement, and decision support questions. Auditing exercises address assurance, control testing, anomaly detection, and risk assessment questions. This structure ensures that students see how the same analytical technique serves different purposes depending on the professional role. An aging analysis, for example, appears as a financial reporting exercise (estimating the allowance for doubtful accounts), a managerial accounting exercise (assessing collection efficiency), and an auditing exercise (evaluating management estimates and selecting balances for confirmation). Students who encounter all three perspectives develop a broader understanding of how analytics creates value across the profession.

The third is its integrated tool progression. Many textbooks teach Excel, SQL, and visualization tools in isolation. This book teaches them as complementary stages of a single workflow. Part II covers Excel. Part III covers SQL. Part IV covers Power BI. Part V brings all three tools together in integrated projects where students extract data using SQL, analyze it in Excel, and present results in Power BI within a single engagement. This progression mirrors how analytics projects work in practice, where no single tool handles every stage.

The fourth is its design as an open educational resource. The book and all companion datasets are freely available, and every tool used in the exercises is either free or included with standard institutional licenses. The Open Access section on the book's landing page describes the licensing terms in detail.

Pedagogical Design

The book follows a consistent pedagogical structure that reflects research on how students learn technical skills most effectively. Worked examples, immediate practice opportunities, and progressive complexity have been shown to support skill acquisition in technology-intensive courses (Borthick and Jones, 2000). Every chapter in this book applies these principles through a structured sequence.

Each chapter opens with four to six learning objectives written in measurable terms using action verbs drawn from Bloom's taxonomy. The objectives span multiple cognitive levels, from foundational understanding through application and analysis, so that both undergraduate and graduate students find appropriate challenges. Following the objectives, an opening scenario places the student in a realistic professional situation drawn from one of the three datasets. The scenario names a role, describes a concrete task, and motivates the material that follows by showing students why it matters before they learn how to do it.

The body of each chapter presents concepts in narrative paragraph form, supported by figures, tables, and diagrams. One to three guided tutorials are embedded within the conceptual content at the point where the relevant technique is introduced, so students read about a concept and immediately practice it before moving to the next topic. Each tutorial includes numbered steps, expected outputs, and a checkpoint that allows students to verify their work. Three types of callout boxes appear throughout the narrative. "In Practice" notes describe how the technique is used in professional settings. "Watch Out" notes warn about common errors and pitfalls. "Connecting the Dots" notes link the current topic to material in other chapters or other tools, helping students see the book as an integrated whole.

Each chapter closes with a summary, a list of key terms with definitions, ten to fifteen multiple choice questions spanning recall, application, and judgment, and applied exercises organized by the three accounting perspectives. Five comprehensive cases, one at the end of each of the book's five parts, provide extended multi-tool investigations that integrate material from all chapters in the part.

In-text citations in APA format appear throughout the narrative to support claims about professional practice, technique effectiveness, and adoption trends. Each chapter closes with a Further Reading section containing five to eight annotated references drawn from peer-reviewed journals, professional standards, and practitioner publications.

Audience and Course Design

This textbook serves both undergraduate and graduate four-credit courses in accounting analytics or accounting information systems. A single text serves both audiences. The instructor controls the depth and rigor of classroom discussions and analyses to match the course level. Undergraduate courses can focus on the guided tutorials and foundational exercises.

Graduate courses can emphasize the analytical judgment required by the applied exercises and comprehensive cases, assign additional readings from the Further Reading sections, and incorporate extended discussion of professional standards and research findings.

The book assumes no prior analytics or programming experience. Students need only the accounting knowledge gained from introductory financial and managerial accounting courses. Every technical concept is introduced from the ground up, and every tool is taught through step-by-step instruction before students are asked to work independently. Students who have prior experience with Excel, SQL, or Power BI will move through the early chapters faster and can focus their effort on the accounting applications and the more advanced techniques in later chapters.

How This Book Is Organized

The book contains twenty chapters organized into five parts. Parts I through IV follow a deliberate progression. Part I (Chapters 1 through 3) builds the conceptual foundation without introducing any tools. Parts II, III, and IV each teach one tool in depth: Excel in Chapters 4 through 8, SQL in Chapters 9 through 12, and Power BI in Chapters 13 through 16. Part V (Chapters 17 through 20) integrates all three tools in applied projects that span financial reporting, cost accounting, forensic analytics, and emerging technologies. A comprehensive case closes each part, requiring students to combine material from all chapters in that part into a multi-component deliverable.

Six appendices provide reference material including a software installation guide, complete dataset documentation with Entity-Relationship diagrams, and quick reference guides for Excel functions, SQL syntax, and DAX functions. Appendix F maps every exercise in the book to the relevant competency areas in the AICPA, IMA, and IFAC frameworks, supporting instructors who need to align their course with accreditation requirements.

A Note on Professional Standards and Research

This textbook references professional standards and peer-reviewed research throughout its chapters. These references serve two purposes. First, they ground the analytical techniques in the professional context where students will apply them. When a chapter on audit analytics references the AICPA's guidance on data analytics in auditing (AICPA, 2017), students see that the techniques they are learning are not academic exercises but tools that professional standards expect them to use. Second, the references connect the practical instruction to the broader body of knowledge in accounting and information systems. Students who read the annotated Further Reading sections will find pathways into the research literature that informs and extends what the textbook teaches.

The profession's integration of analytics into its competency frameworks has accelerated in recent years. The AICPA has embedded data analytics across its pre-certification curriculum. The IMA has emphasized technology and analytics in the Certified Management Accountant examination content. The IFAC has published guidance on the technology competencies that accounting graduates need (IFAC, 2019). These developments confirm that the skills taught in this book are not supplementary. They are foundational to the practice of accounting in the current environment.

To the Student

You are beginning a book that will change how you work with accounting data. The accounting courses you have taken so far taught you to understand financial statements, apply standards, calculate ratios, and interpret results. Those skills remain essential. What this book adds is the ability to work directly with the data that produces those statements, ratios, and results. Instead of receiving a finished trial balance and analyzing it, you will learn to query a database, extract the general ledger entries, and build the trial balance yourself. Instead of reading a variance report, you will learn to calculate the variances from production data and present them in an interactive dashboard. Instead of reviewing a sample of transactions that someone else selected, you will learn to test the entire population and let the data reveal the anomalies.

This book assumes no prior experience with analytics or programming. If you have never written a SQL query, never built a PivotTable, and never opened Power BI, you are exactly the audience this book was written for. Every technique is introduced from the ground up with step-by-step guided tutorials that you can follow at your own pace. The tutorials build on one another, so the output of one often serves as the input for the next, and the complexity increases gradually across chapters.

What to Expect

The book follows a consistent structure that will become familiar after the first few chapters. Each chapter opens with learning objectives that tell you what you will be able to do after completing the chapter. An opening scenario places you in a professional role and describes a task that motivates the material. The body of the chapter alternates between conceptual explanation and hands-on tutorials. Callout boxes provide practical tips, warnings, and connections to other parts of the book. Each chapter closes with a summary, key terms, review questions, and applied exercises.

The applied exercises at the end of each chapter are organized into three sections by accounting perspective. Financial Accounting exercises focus on reporting and disclosure tasks. Managerial Accounting exercises focus on costing, budgeting, and performance measurement. Auditing exercises focus on testing, anomaly detection, and assurance procedures. You will complete exercises in all three perspectives regardless of which area of accounting interests you most. This breadth is intentional. Analytics skills transfer across roles, and understanding how the same technique serves different purposes will make you a more versatile professional.

The Three Tools

You will learn three tools in this book, each suited to a different stage of the analytics workflow.

Microsoft Excel is the tool you will use first. Chapters 4 through 8 teach you to organize data in structured tables, clean and prepare messy data, build PivotTables for summarization, run regression models, and perform audit analytics procedures. Excel is most powerful for ad hoc analysis, financial modeling, and workpaper preparation.

SQL is introduced in Chapters 9 through 12. SQL stands for Structured Query Language, and it is the standard language for retrieving data from relational databases. You will use a free, lightweight database system called SQLite that requires no server and runs on any operating system. SQL is most powerful when working with large datasets, when data spans multiple related tables that must be combined, and when you want to save and reuse your analytical procedures.

Microsoft Power BI is introduced in Chapters 13 through 16. Power BI is a business intelligence platform that lets you build interactive dashboards and reports. You will connect Power BI to the same datasets you used in Excel and SQL, create data models, write DAX formulas for calculated measures, and design dashboards that stakeholders can explore on their own.

In Part V of the book, you will use all three tools together. You will extract data with SQL, analyze it in Excel, and present results in Power BI within a single integrated project.

The Three Datasets

You will work with three datasets throughout this book. All three are provided as both Excel workbooks and SQLite databases. You will become deeply familiar with them because you use them repeatedly across chapters, building cumulative knowledge of their structures, contents, and quirks. The datasets progress from a compact wholesale distributor (Northwind Traders, eight tables) through a mid-size manufacturer (Adventure Works Cycles, approximately seventy tables) to a full enterprise resource planning environment with a complete accounting module (ERPNext Demo Company). The About the Datasets section provides full descriptions of each database, including the specific chapters where each one appears.

How to Succeed

The most important habit you can develop is to do the tutorials yourself rather than reading through them passively. Open the dataset, follow the steps, and verify your results at each checkpoint. When you make an error, diagnosing and correcting it teaches you more than

getting it right the first time. The guided tutorials are designed so that you can complete them independently at your own pace, and they prepare you directly for the applied exercises that follow.

A second important habit is to read the conceptual material before jumping to the tutorials. The concepts explain why a technique works and when it is appropriate. The tutorials show you how to execute it. Both are necessary. A student who can execute a Benford's Law analysis without understanding what the results mean or when the test is appropriate has a technical skill but not an analytical one. This book aims to develop both.

Finally, pay attention to the connections across chapters and tools. The "Connecting the Dots" callout boxes link the current topic to material elsewhere in the book. The three-perspective exercise structure shows you how the same technique applies in different contexts. The comprehensive cases at the end of each part ask you to integrate everything you have learned. These connections are where the deepest learning happens, because they move you from knowing how to use a tool to understanding how to solve a problem.

About the Datasets

This textbook is built around three datasets that students use throughout all twenty chapters. All three are provided in both Microsoft Excel workbook format and SQLite database format, so students work with the same underlying data regardless of whether a given chapter uses Excel, SQL, or Power BI. The datasets represent three different types of businesses at three levels of complexity, and the textbook introduces them from simplest to most complex.

Northwind Traders is a fictional small wholesale food distribution company. It buys specialty food products from suppliers around the world and sells them to retail and restaurant customers. The Northwind database contains eight core tables covering customers, orders, order details, products, product categories, suppliers, employees, and shippers. The database includes approximately 830 orders, 2,100 order line items, 77 products, and 91 customers. Northwind is compact and easy to understand, which makes it the primary dataset for the foundational chapters where students are learning new tools and techniques. It supports exercises in sales analysis, customer analytics, purchasing evaluation, and inventory review.

Northwind appears in Chapters 1 through 10, 12, 13, 14, 16, and 19.

Adventure Works Cycles is a fictional mid-size multinational bicycle manufacturer that designs, produces, and sells bicycles, components, clothing, and accessories through multiple sales territories. The AdventureWorks database contains approximately seventy tables organized across five functional areas covering sales, production, purchasing, human resources, and person management. The database includes manufacturing cost data such as bills of materials, work orders, production routing, and scrap tracking, along with multi-territory sales transactions and vendor purchase orders. AdventureWorks introduces the data complexity that students will encounter in manufacturing and multi-division organizations, and it supports exercises in cost accounting, production analysis, sales performance evaluation, and purchasing cycle testing.

AdventureWorks appears in Chapters 1, 3, 5 through 7, 10 through 13, 15 through 19.

ERPNext Demo Company is a fictional company operating within a full enterprise resource planning environment that includes a complete accounting module. The ERPNext database contains a chart of accounts, general ledger entries, journal entries, sales and purchase invoices, cost centers, budgets, payment entries, bank reconciliation records, stock ledger entries, and asset registers. ERPNext is the most accounting-rich dataset of the three. Students use it for financial statement preparation, general ledger analysis, budgetary control, audit testing, and integrated financial reporting. Because ERPNext mirrors the structure of a production ERP

system, working with it gives students familiarity with the kind of data environment they will encounter in professional practice.

ERPNext appears in Chapters 1 through 3, 8, 11 through 13, 15 through 20.

The textbook introduces Northwind first because it is the simplest. AdventureWorks enters in the middle chapters as exercises grow more complex and students have built enough skill to work with richer data. ERPNext appears once students have enough analytical experience to work with a full accounting system. By the final chapters, students move fluently across all three datasets in integrated exercises. Complete documentation for every table in all three databases, including column names, data types, primary and foreign key relationships, and Entity-Relationship diagrams, appears in Appendix B.

Acknowledgments

[This section will acknowledge individuals and institutions that contributed to the development of this textbook, including reviewers, colleagues who class-tested draft chapters, students who provided feedback, and the open-source communities that maintain the tools and datasets used throughout the book.]

References

AICPA (American Institute of Certified Public Accountants). (2017). *Guide to audit data analytics.* AICPA.

Borthick, A. F., and Jones, D. R. (2000). The motivation for collaborative discovery learning online and its application in an information systems assurance course. *Issues in Accounting Education*, 15(2), 181-210.

IFAC (International Federation of Accountants). (2019). *Technology and the profession: A guide for professional accountancy organizations.* IFAC.

Sledgianowski, D., Gomaa, M., and Tan, C. (2017). Toward integration of Big Data, technology, and information systems competencies into the accounting curriculum. *Journal of Accounting Education*, 38, 81-93.

Downloads

This page provides access to the latest version of the book and the datasets used throughout the chapters. As the project evolves, we will add versioned releases for major editions.

Latest Book Version

These files are automatically updated whenever the book is rebuilt.

- Download PDF (latest)
- Download EPUB (latest)
- Download DOCX (latest)

Datasets

The datasets below are hosted on GitHub Releases to support large file sizes and fast global downloads.

Northwind

- Northwind (SQLite)
- Northwind (Excel)

AdventureWorks

- AdventureWorks (SQLite)
- AdventureWorks (Excel)

Versioned Editions (coming soon)

In future major releases, we will archive:

- Edition 1.0 (PDF, EPUB, DOCX)
- Edition 2.0 (PDF, EPUB, DOCX)
- Dataset versions (v1.0, v2.0, etc.)

These will remain permanently accessible for citation and reproducibility.

Part I: Foundations of Accounting Analytics

The Role of Analytics in Accounting

Conceptual Chapter Northwind AdventureWorks ERPNext

Learning Objectives

After completing this chapter, you will be able to:

1. Explain how data analytics has changed the work accountants perform and the skills the profession demands.
2. Define descriptive, predictive, and prescriptive analytics and identify examples of each in financial accounting, managerial accounting, and auditing.
3. Describe the purpose and capabilities of Excel, SQL, and Power BI as complementary tools for accounting analytics.
4. Compare the structure and business context of the three datasets used throughout this textbook.
5. Map the stages of an accounting analytics workflow from question formulation through communication of results.

Opening Scenario

You have just started a position as a staff accountant at a mid-size distribution company. During your first week, the controller asks you to investigate why gross margins declined in the most recent quarter. She hands you login credentials to the company's enterprise resource planning system and tells you the data you need is "in the database." Your accounting coursework prepared you to interpret financial statements, calculate ratios, and apply standards, but you have never pulled data directly from a database, cleaned it for analysis, or built an interactive report that the management team can explore on their own. You realize that answering the controller's question will require more than accounting knowledge. It will require the ability

to extract, prepare, analyze, and communicate data. This chapter introduces the analytical mindset and the toolkit you will develop throughout this book to handle exactly this kind of challenge.

The Changing Landscape of Accounting Work

The accounting profession has always been built on data. Accountants record transactions, classify them into accounts, summarize them in financial statements, and interpret the results for stakeholders. For most of the profession's history, the volume of data involved in these tasks was manageable through manual methods and later through basic spreadsheets. A general ledger might contain a few thousand entries per year, and an auditor could review a reasonable portion of those entries by hand. That era has ended.

Modern organizations generate transaction data at a scale that makes manual review impractical. A single retail company may process millions of sales transactions per month, each one recorded with a timestamp, a product identifier, a customer identifier, a store location, a payment method, and a discount code. An enterprise resource planning system at a manufacturing firm captures not only financial transactions but also production orders, material movements, quality inspections, and employee time records. The data that accountants must work with has grown by orders of magnitude, and it continues to grow (Vasarhelyi, Kogan, and Tuttle, 2015).

This growth in data has created both a challenge and an opportunity for the profession. The challenge is that traditional methods of sampling and manual review cannot keep pace with the volume and complexity of modern datasets. An auditor who selects 50 transactions from a population of 500,000 is examining one hundredth of one percent of the available evidence. The opportunity is that the same technology that generates massive datasets also provides tools to analyze entire populations rather than small samples. Accountants who can use these tools effectively bring more evidence to their conclusions, identify patterns that sampling would miss, and deliver insights that go beyond compliance and reporting (Earley, 2015).

Professional bodies have recognized this shift and responded by updating their competency frameworks. The American Institute of Certified Public Accountants has integrated data analytics into its pre-certification curriculum and its continuing education requirements. The Institute of Management Accountants has emphasized technology and analytics in its Certified Management Accountant examination content. The International Federation of Accountants has published guidance on the skills that accounting graduates need in a data-rich environment (IFAC, 2019). These developments signal that analytics is not a niche specialization within accounting. It is a core competency that every practitioner will need.

In Practice

Accounting firms of all sizes now recruit for analytics skills alongside traditional accounting knowledge. Major firms have established dedicated data analytics practices, and even small and mid-size firms use data analysis tools for audit testing, tax planning, and advisory engagements. A 2019 survey by the Institute of Management Accountants found that data analytics ranked among the top five skills employers seek when hiring management accountants (IMA, 2019).

The purpose of this textbook is to help you develop these skills in a structured and practical way. You will learn to work with data using three tools that are widely used in practice: Microsoft Excel, SQL, and Microsoft Power BI. You will apply these tools to realistic accounting data drawn from three carefully designed datasets. And you will practice solving problems from three accounting perspectives: financial accounting, managerial accounting, and auditing. By the end of this book, you will be able to extract data from a database, prepare it for analysis, summarize and model it, visualize the results, and communicate your findings to professional audiences.

Defining Accounting Analytics

Analytics is the process of examining data to draw conclusions, identify patterns, and support decisions. In accounting, analytics applies this process to financial and operational data for purposes such as reporting, planning, control, and assurance. The term "accounting analytics" as used in this textbook refers to the application of data extraction, preparation, analysis, and visualization techniques to accounting data for the purpose of producing information that supports financial reporting, managerial decision-making, and audit assurance.

It is important to distinguish accounting analytics from related terms that students may encounter. "Data analytics" is a broad term that encompasses analytical work in any domain, from marketing to healthcare to logistics. "Business intelligence" typically refers to the tools and infrastructure that organizations use to collect, store, and present data for operational and strategic decisions. "Big data" refers to datasets that are too large or complex for traditional data processing methods and is often associated with technologies such as Hadoop and cloud computing. Accounting analytics draws on all of these concepts but applies them specifically to the data, questions, and standards that define the accounting profession (Richins, Stapleton, Stratopoulos, and Wong, 2017).

The Three Types of Analytics

A useful framework for understanding the scope of analytics divides it into three categories: descriptive, predictive, and prescriptive. These categories represent increasing levels of

analytical complexity and are not mutually exclusive. Most real-world accounting analytics projects involve more than one type.

Figure 1: The Analytics Continuum.

Descriptive analytics answers the question "what happened?" It involves summarizing historical data to understand past performance, identify trends, and describe the current state of affairs. Descriptive analytics is the foundation of most accounting work. When a financial analyst prepares a comparative income statement showing revenue by product line for the past four quarters, that is descriptive analytics. When an auditor stratifies accounts receivable by aging bucket to assess the adequacy of the allowance for doubtful accounts, that is also descriptive analytics. The tools of descriptive analytics include aggregation, grouping, sorting, filtering, and basic statistical measures such as totals, averages, and counts.

In the context of this textbook, descriptive analytics appears most prominently in the Excel chapters (Part II) and the introductory SQL chapters (Part III). Students will build PivotTables that summarize sales by category and quarter, write SQL queries that calculate average order values by customer, and create Power BI dashboards that display key performance indicators. These are all descriptive tasks, and they form the base on which more advanced analysis is built.

Predictive analytics answers the question "what might happen?" It involves using historical data to make forecasts, estimate probabilities, and identify the factors that drive outcomes. Predictive analytics moves beyond describing the past to projecting the future. When a management accountant builds a regression model to forecast next quarter's revenue based on historical trends, that is predictive analytics. When an auditor uses statistical analysis to set an expectation for an account balance and then compares the actual balance to the expectation as a substantive analytical procedure, that too involves prediction. The tools of predictive analytics include regression analysis, trend extrapolation, time series modeling, and classification techniques.

In this textbook, predictive analytics appears in the modeling chapter of the Excel section (Chapter 7), in the intermediate SQL chapters where students use window functions for period-over-period analysis (Chapter 11), and in the Power BI chapters where DAX time intelligence functions enable rolling averages and year-over-year comparisons (Chapter 15). Students will

build forecasting models, conduct sensitivity analyses, and set analytical expectations that mirror the procedures used in professional practice.

Prescriptive analytics answers the question "what should we do?" It involves using data analysis to recommend a course of action, often by optimizing an objective subject to constraints. Prescriptive analytics is the most advanced of the three types and is less common in current accounting practice than descriptive and predictive analytics, although it is growing. When a cost accountant uses Excel's Solver tool to determine the optimal product mix that maximizes contribution margin given production capacity constraints, that is prescriptive analytics. When a tax advisor uses scenario modeling to recommend a filing strategy that minimizes a client's tax liability across multiple jurisdictions, that also has prescriptive elements.

This textbook introduces prescriptive analytics through optimization exercises in Chapter 7 (using Solver and Scenario Manager in Excel) and through scenario analysis in the integrated chapters of Part V. The emphasis throughout the book, however, is on descriptive and predictive analytics because these represent the analytical work that accounting professionals perform most frequently.

> Watch Out
>
> Students sometimes assume that descriptive analytics is "basic" and that the goal is to reach prescriptive analytics as quickly as possible. In practice, most accounting analytics work is descriptive, and doing it well requires substantial skill. A poorly constructed summary can mislead decision makers just as easily as a sophisticated model can. The value of analytics lies not in its complexity but in the quality of the questions it answers and the rigor of the process behind it.

Why Accountants Need Analytics Skills

The demand for analytics skills in accounting is driven by several developments that have reshaped the profession over the past two decades. Understanding these drivers helps explain why this textbook exists and why the skills it teaches matter for your career.

The first driver is the increasing availability of data. As organizations have adopted enterprise resource planning systems, cloud-based accounting platforms, and automated transaction processing, the volume of financial and operational data available for analysis has expanded dramatically. Accountants who can access and analyze this data directly, rather than waiting for IT departments to produce reports, work more efficiently and can respond to questions in real time (Appelbaum, Kogan, and Vasarhelyi, 2017).

The second driver is the evolution of audit methodology. Auditing standards increasingly encourage or require the use of data analytics. The ability to test an entire population of transactions rather than a sample changes the nature of audit evidence. Auditors who use analytics

can identify anomalies that sampling would miss, perform more precise risk assessments, and provide more relevant findings to audit committees. The Public Company Accounting Oversight Board has emphasized the use of technology and data analysis in its inspection findings, and firms have responded by investing in analytics training for their audit staff (PCAOB, 2017).

The third driver is the expansion of the accountant's role from preparer and verifier to advisor and analyst. Organizations expect their accounting and finance teams to provide forward-looking analysis, scenario planning, and strategic insight, not just historical reports. Management accountants, in particular, are asked to explain why results differ from plan, to forecast future performance under different assumptions, and to identify the operational drivers behind financial outcomes. These tasks require analytical skills that go beyond debits and credits (Cokins, 2013).

The fourth driver is competitive pressure in the labor market. Graduates who can combine accounting knowledge with data analysis skills are more attractive to employers than those with accounting knowledge alone. This is true across all areas of the profession, from public accounting to industry to government. The ability to write a SQL query, build a PivotTable, or create an interactive dashboard distinguishes a candidate in a hiring process and opens career paths that were previously reserved for specialists in information systems or data science.

> In Practice
>
> A survey of accounting professionals conducted by Kokina and Dagiliene (2017) found that respondents identified data analysis and technology skills as the area of greatest change in the competencies required of accountants. Respondents across audit, tax, and advisory roles reported that they spend more time working with data tools than they did five years earlier and expected that trend to continue.

Table 1: Examples of Analytics Applications by Accounting Role

Role	Descriptive Example	Predictive Example
Financial Reporting	Comparative income statement showing revenue by product line for the past four quarters	Revenue forecast for the next fiscal quarter using trend extrapolation of historical sales data
Managerial Accounting	Product profitability ranking by category based on unit margins and sales volume	Cost-volume-profit model projecting break-even points under three pricing scenarios

Role	Descriptive Example	Predictive Example
External Audit	Stratification of accounts receivable by aging bucket to evaluate the allowance for doubtful accounts	Regression-based expectation for monthly revenue used as a substantive analytical procedure
Internal Audit	Frequency distribution of payment amounts to identify concentrations near approval thresholds	Risk scoring of journal entries based on multiple indicators to prioritize selections for testing

The Accounting Analytics Workflow

Before learning specific tools, it is helpful to understand the general process that all accounting analytics projects follow. Whether you are building a revenue summary in Excel, writing an audit query in SQL, or designing a dashboard in Power BI, the work progresses through the same sequence of stages. This textbook is organized around these stages, and understanding them now will help you see how the chapters connect.

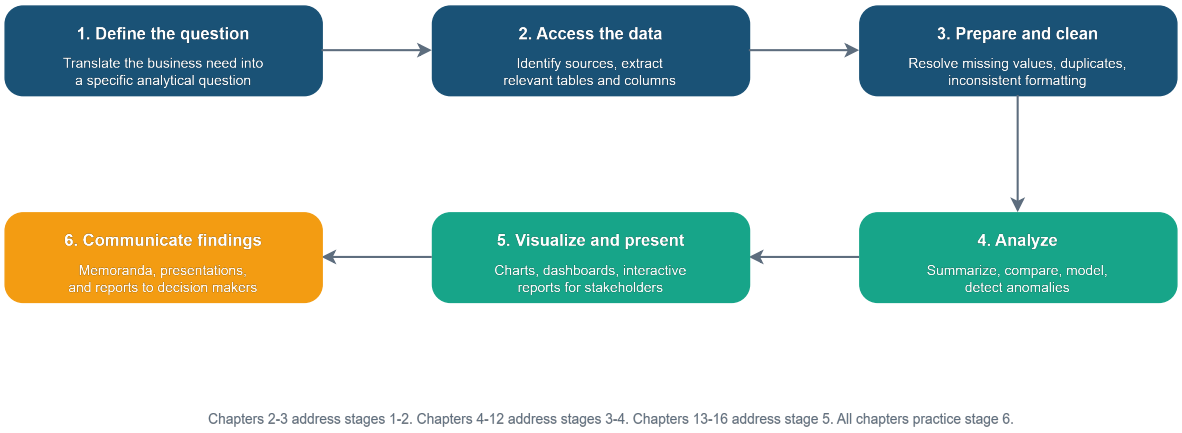


Figure 2: The Accounting Analytics Workflow.

The first stage is defining the question. Every analytics project begins with a clear statement of the problem or question to be addressed. In the opening scenario of this chapter, the controller's question was specific: why did gross margins decline in the most recent quarter? A well-defined question guides every subsequent decision, from which data to collect to which visualizations to create. Vague questions produce vague answers. The ability to translate a business concern into an analytical question is one of the most important skills an accountant can develop.

The second stage is identifying and accessing the data. Once the question is defined, the accountant must determine which data is needed and where it resides. In many organizations, the relevant data is stored in databases that are part of the company's ERP system or accounting software. Accessing this data may require writing SQL queries to extract the relevant tables and columns. In other cases, the data may be available in spreadsheet exports or flat files. Chapter 2 and Chapter 3 of this textbook address this stage by teaching students how accounting data is structured and stored.

The third stage is preparing and cleaning the data. Raw data almost always contains problems that must be resolved before analysis can begin. Missing values, duplicate records, inconsistent formatting, and incorrect data types are common in accounting datasets. Data preparation is often the most time-consuming stage of an analytics project, and it is also the most important. Analysis performed on dirty data produces unreliable results. Chapters 4 and 5 cover data preparation in Excel, and Chapters 9 and 10 cover data extraction and joining in SQL.

The fourth stage is analyzing the data. This is where the accountant applies techniques such as summarization, comparison, trend analysis, regression, and anomaly detection to answer the question defined in the first stage. The choice of technique depends on the question and the type of data available. Chapters 6 through 8 cover analysis in Excel, Chapters 10 through 12 cover analysis in SQL, and Chapters 15 and 16 cover analysis in Power BI.

The fifth stage is visualizing and presenting the results. Data visualization transforms analytical results into visual formats that audiences can interpret quickly and accurately. A well-designed chart, table, or dashboard communicates the answer to the original question more effectively than a spreadsheet full of numbers. Chapter 13 covers visualization principles, and Chapters 14 through 16 teach students to build interactive reports and dashboards in Power BI.

The sixth stage is communicating findings and recommendations. The final product of an analytics project is not a spreadsheet or a dashboard but a communication to decision makers. That communication might take the form of a written memorandum to an audit committee, a presentation to management, or an interactive report that stakeholders can explore. The ability to translate analytical results into clear, professional language is essential. This skill is practiced throughout the textbook in the applied exercises and comprehensive cases, each of which asks students to produce a written interpretation alongside their technical work.

> **Connecting the Dots**
>
> The six-stage workflow described here is not unique to accounting. It closely parallels the data analysis process described in data science and business intelligence literature (Provost and Fawcett, 2013). What makes accounting analytics distinctive is not the process itself but the data it works with (financial and operational transactions), the questions it addresses (reporting, control, assurance), and the professional standards that govern the work (GAAP, GAAS, IMA ethical standards). Throughout this book, the

> workflow provides the thread that connects every chapter and every tool.

The Tools of This Textbook

This textbook uses three tools: Microsoft Excel, SQL (Structured Query Language), and Microsoft Power BI. These tools were selected because they are widely used in accounting practice, they are accessible to students without programming backgrounds, and together they cover the full analytics workflow from data preparation through visualization.

Microsoft Excel

Excel is the most widely used analytical tool in the accounting profession. Nearly every accountant works with Excel on a daily basis, and most accounting graduates enter the workforce with at least basic spreadsheet skills. This textbook builds on that foundation and extends it into areas that many students have not explored, including structured Excel Tables, PivotTables, statistical functions, Power Query for data preparation, the Data Analysis ToolPak for regression, and Solver for optimization.

Excel is most powerful when working with datasets that fit comfortably in a spreadsheet, typically up to a few hundred thousand rows. It excels at ad hoc analysis, where the accountant needs to explore data interactively, try different approaches, and produce results quickly. Excel is also the primary tool for building financial models, conducting what-if analysis, and preparing workpapers that document analytical procedures.

Part II of this textbook (Chapters 4 through 8) is devoted to Excel. Students will learn to import, clean, summarize, model, and test accounting data using features that go well beyond basic formulas and formatting.

SQL (Structured Query Language)

SQL is the standard language for accessing data stored in relational databases. It allows accountants to extract exactly the data they need from large, complex databases without relying on pre-built reports or IT intermediaries. SQL is particularly important for accountants who work with ERP systems, because the underlying data in these systems is stored in relational databases that SQL can query directly.

SQL is most powerful when working with large datasets, when the data spans multiple related tables that must be joined together, and when the same analysis needs to be performed repeatedly. An auditor who writes a SQL query to identify duplicate payments can run that same query every quarter with no additional effort. A financial analyst who writes a query to

produce a revenue summary by territory and product line can reuse that query whenever the data is updated.

Part III of this textbook (Chapters 9 through 12) is devoted to SQL. Students will learn to write queries that retrieve, filter, join, aggregate, and analyze accounting data stored in SQLite databases.

Microsoft Power BI

Power BI is a business intelligence platform that enables accountants to build interactive dashboards and reports. It connects to a wide range of data sources (including Excel files and SQL databases), provides a data modeling layer for defining relationships and calculations, and offers a rich set of visualization tools for presenting results.

Power BI is most powerful when the goal is to create a reusable, interactive report that multiple stakeholders can explore. A management accountant who builds a budget-versus-actual dashboard in Power BI gives every department manager the ability to filter the report to their own cost center, drill down into specific line items, and compare results across periods. An audit team that builds an anomaly detection dashboard creates a monitoring tool that can be refreshed with new data each audit cycle.

Part IV of this textbook (Chapters 13 through 16) is devoted to Power BI. Students will learn visualization principles, the Power BI interface, data modeling with DAX, and interactive dashboard design.

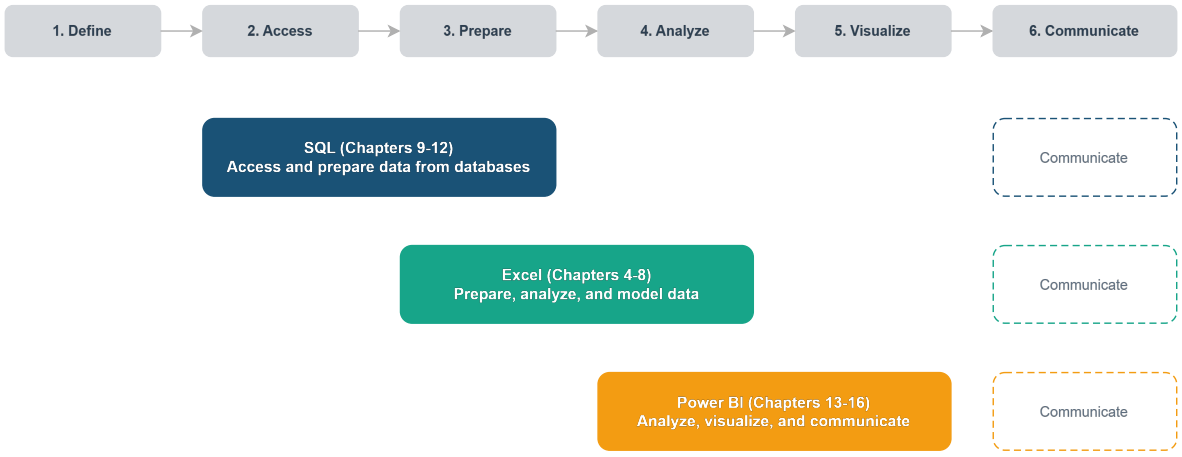


Figure 3: The Three Tools and Their Roles in the Analytics Workflow.

Connecting the Dots

Part V of this textbook (Chapters 17 through 20) integrates all three tools. In those chapters, you will work through scenarios that begin with SQL to extract data from a database, move to Excel for detailed analysis and modeling, and finish in Power BI for visualization and presentation. This integrated approach mirrors how analytics projects work in practice, where no single tool handles every stage of the workflow.

Introduction to the Three Datasets

This textbook is built around three datasets that you will use throughout the entire book. All three datasets are provided in two formats: Microsoft Excel workbooks and SQLite database files. This means you will work with the same underlying data regardless of whether you are using Excel, SQL, or Power BI in a given chapter. The datasets represent three different types of businesses at three different levels of complexity, and the textbook introduces them in order from simplest to most complex.

Northwind Traders

Northwind Traders is a fictional small wholesale food distribution company. It buys specialty food products from suppliers around the world and sells them to retail and restaurant customers. The Northwind database is compact and straightforward. It contains eight core tables: Customers, Orders, OrderDetails, Products, Categories, Suppliers, Employees, and Shippers.

The Northwind dataset is the first one you will work with in this book. Its simplicity makes it ideal for learning new tools and techniques without being distracted by data complexity. When you write your first SQL query in Chapter 9 or build your first PivotTable in Chapter 6, you will use Northwind data. The database contains approximately 830 orders, 2,100 order line items, 77 products, and 91 customers. These numbers are small enough that you can inspect the data manually to verify your analytical results, which is a valuable habit to develop.

From a financial accounting perspective, Northwind provides sales revenue data, customer receivables information, and product inventory records that support exercises in revenue analysis, receivables aging, and inventory valuation. From a managerial accounting perspective, Northwind provides data on product pricing, shipping costs, and sales performance by employee and region. From an auditing perspective, Northwind provides transaction-level data that supports exercises in duplicate detection, completeness testing, and anomaly identification.

Adventure Works Cycles

Adventure Works Cycles is a fictional mid-size multinational bicycle manufacturer. It designs, produces, and sells bicycles, bicycle components, clothing, and accessories through multiple sales territories. The AdventureWorks database is substantially larger and more complex than Northwind. It contains approximately 70 tables organized across five functional areas: Sales, Production, Purchasing, Human Resources, and Person management.

The AdventureWorks dataset enters the textbook in the middle chapters as exercises grow more complex and students have built enough skill to handle a richer data environment. Its manufacturing data is particularly valuable for managerial accounting exercises. The database contains work orders with routing and scrap tracking, bills of materials, standard and actual cost records, vendor purchase orders, and multi-territory sales transactions. These data elements support exercises in variance analysis, product costing, production performance measurement, and purchasing evaluation that are not possible with the simpler Northwind dataset.

From a financial accounting perspective, AdventureWorks provides multi-territory revenue data and product cost data that support exercises in disaggregated revenue disclosure and cost of goods sold analysis. From an auditing perspective, AdventureWorks provides purchasing cycle data that supports exercises in vendor analysis, segregation of duties testing, and purchase price variance evaluation.

ERPNext Demo Company

ERPNext is a fictional company operating within a full enterprise resource planning environment. Unlike Northwind and AdventureWorks, which focus on specific operational areas, the ERPNext database includes a complete accounting module with a chart of accounts, general ledger entries, journal entries, sales and purchase invoices, cost centers, budgets, payment entries, bank reconciliation records, and asset registers. This makes ERPNext the most accounting-rich dataset of the three.

The ERPNext dataset appears in the textbook once students have developed enough analytical skill to work with a full accounting system. It is the primary dataset for financial statement preparation, general ledger analysis, budgetary control, audit testing, and integrated financial reporting. Because ERPNext mirrors the structure of a production ERP system, working with it gives students familiarity with the kind of data environment they will encounter in professional practice.

From a financial accounting perspective, ERPNext provides everything needed to prepare financial statements from the general ledger, perform ratio analysis, and conduct period-over-period comparisons. From a managerial accounting perspective, ERPNext provides cost center data and budget records that support exercises in budget-versus-actual analysis and departmental performance measurement. From an auditing perspective, ERPNext provides

journal entry data that supports exercises in management override testing, Benford's Law analysis, duplicate payment detection, and continuous monitoring.

Table 2: Comparison of the Three Textbook Datasets

Attribute	Northwind Traders	Adventure Works Cycles	ERPNext Demo Company
Company Type	Small wholesale food distributor	Mid-size multinational bicycle manufacturer	Full enterprise resource planning environment
Number of Core Tables	8 tables	Approximately 70 tables across 5 schemas	20+ accounting and operational tables
Key Functional Areas	Sales, customers, products, suppliers	Sales, production, purchasing, human resources, person management	General ledger, journal entries, invoices, payments, budgets, cost centers
Accounting Module Depth	Transaction-level sales data only (no general ledger or chart of accounts)	Product cost and manufacturing data (no complete accounting module)	Full chart of accounts, general ledger, journal entries, and financial reporting capability
Primary Use in This Book	Introductory chapters for learning new tools and techniques	Mid-level chapters for cost accounting, production analysis, and purchasing evaluation	Financial reporting, audit analytics, budgetary control, and integrated analysis
Chapters Where Featured	1 through 10, 12 through 14, 16, and 19	1, 3, 5 through 7, 10 through 13, 15 through 19	1 through 3, 8, 11 through 13, 15 through 20

Watch Out

These three datasets are fictional companies created for educational purposes. They are designed to illustrate realistic data structures and analytical challenges, but they do not represent the full complexity or messiness of real-world data. Real ERP systems may contain hundreds of tables, custom fields, and data quality problems that exceed what you will encounter in these datasets. The skills you learn here will transfer to real-world environments, but you should expect that professional practice will require additional adaptation and judgment.

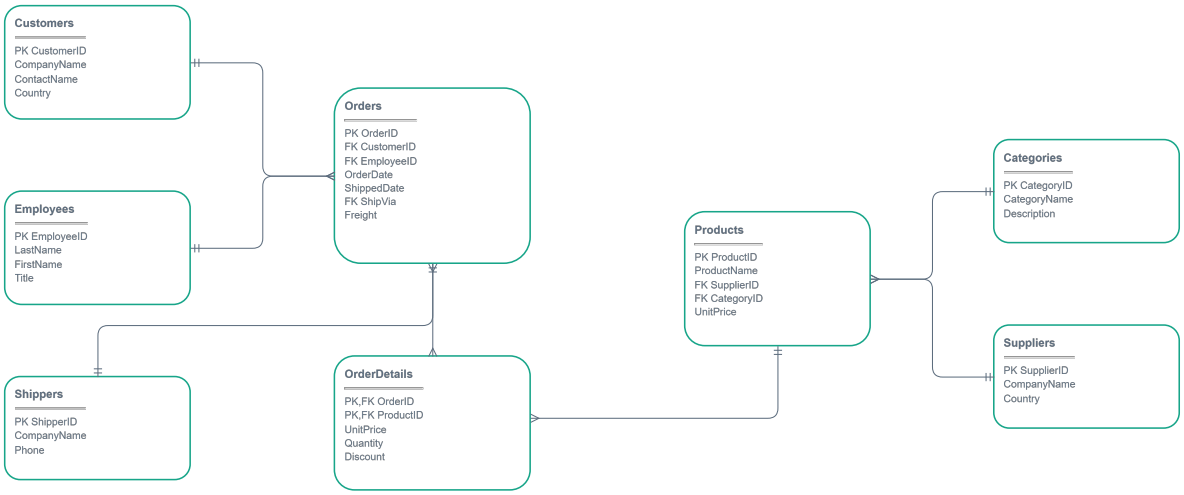


Figure 4: implified Entity-Relationship Diagram for Northwind Traders.

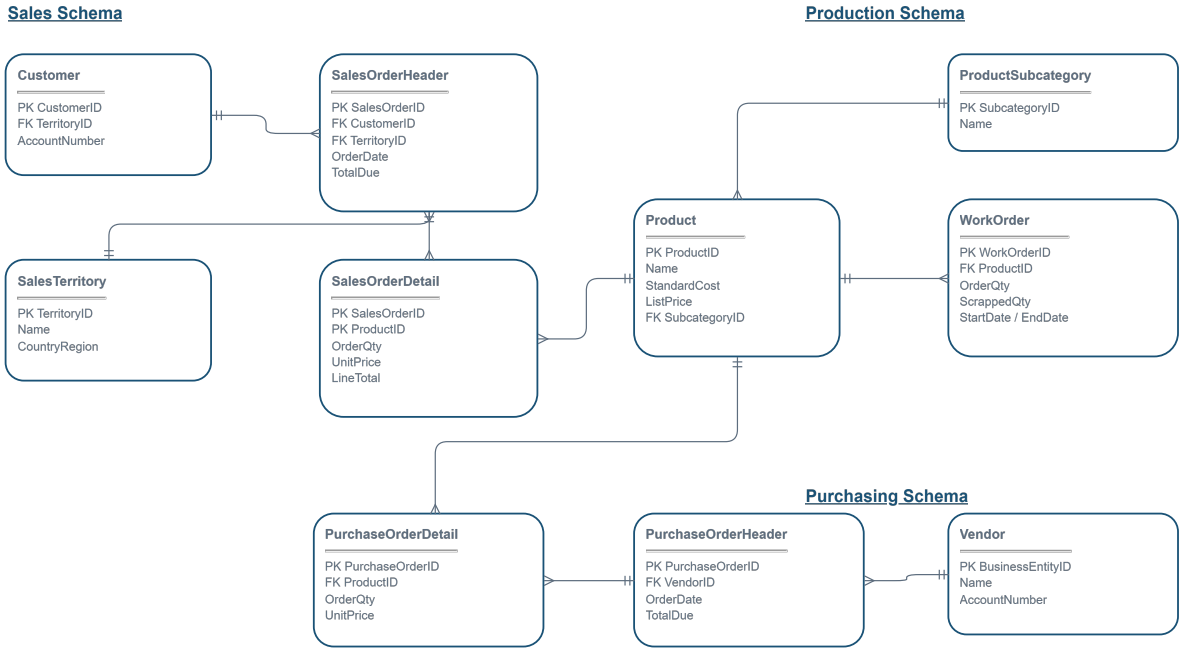


Figure 5: Simplified Entity-Relationship Diagram for Adventure Works Cycles.

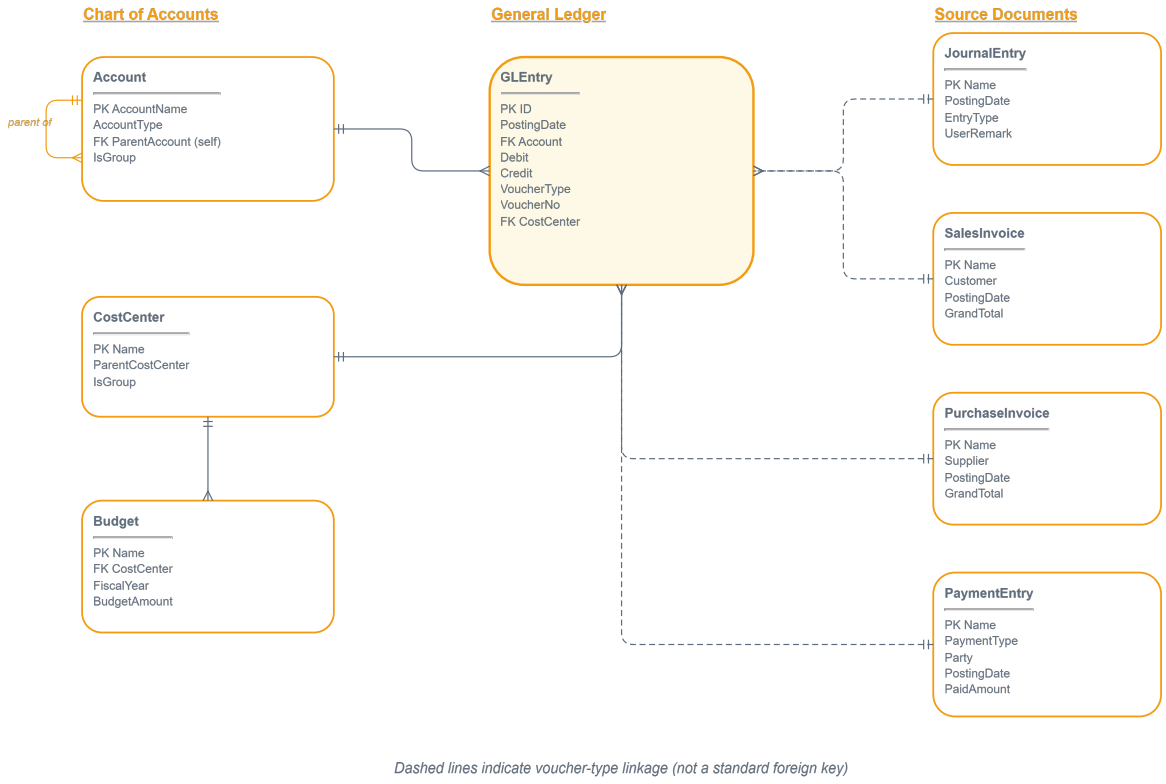


Figure 6: Simplified Entity-Relationship Diagram for ERPNext Demo Company.

Guided Tutorial 1.1: Exploring the Three Datasets

Context and objective. In this tutorial, you will open each of the three datasets in Excel and examine their structure. The goal is not to perform any analysis but to build familiarity with the tables, columns, and business context of each dataset. By the end of this tutorial, you will know what data is available in each database and where to find it.

Prerequisites. You need Microsoft Excel installed on your computer and access to the three Excel workbook files provided with this textbook: Northwind.xlsx, AdventureWorks.xlsx, and ERPNext.xlsx.

Step-by-step instructions.

Step 1. Open the file Northwind.xlsx in Excel. Notice that the workbook contains multiple worksheets, each representing one table in the Northwind database. Click through each worksheet tab at the bottom of the screen and note the table name. You should see worksheets labeled Customers, Employees, Categories, Suppliers, Shippers, Products, Orders, and OrderDetails.

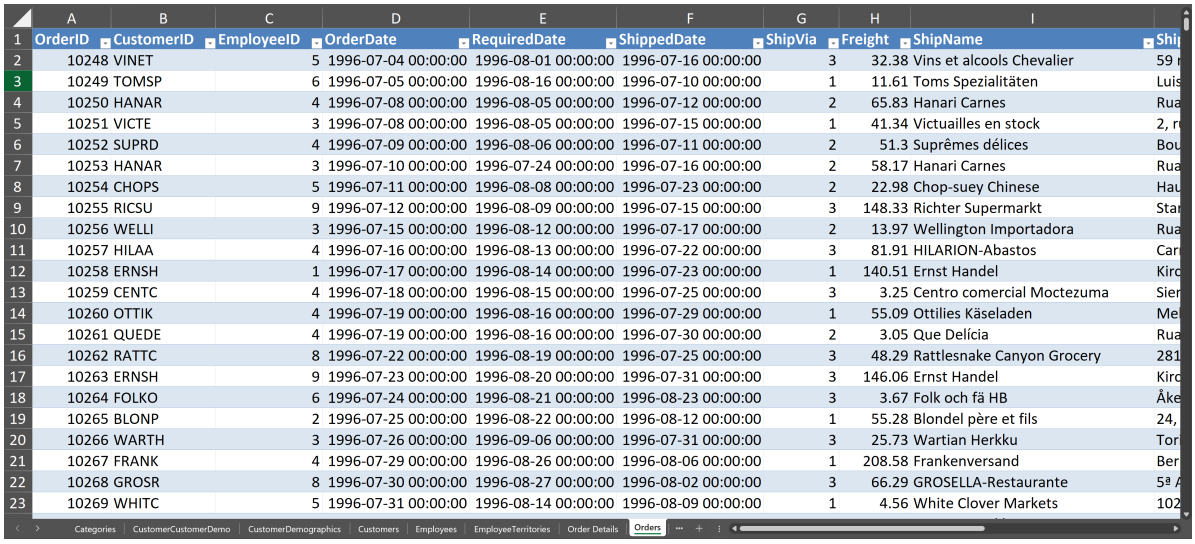
Step 2. Click on the Orders worksheet. Examine the column headers in row 1. You should see columns including OrderID, CustomerID, EmployeeID, OrderDate, RequiredDate, ShippedDate, ShipVia, and Freight, among others. Scroll down to get a sense of the number of records. The Orders table contains approximately 830 rows of data.

Step 3. Click on the OrderDetails worksheet. Examine the columns, which include OrderID, ProductID, UnitPrice, Quantity, and Discount. Notice that OrderID appears in both the Orders and OrderDetails worksheets. This column links the two tables together, a concept you will study in depth in Chapter 3 and use extensively when writing SQL joins in Chapters 10 and 11.

Step 4. Close Northwind.xlsx and open AdventureWorks.xlsx. This workbook contains significantly more worksheets than Northwind. Click through several worksheet tabs and note the variety of data available. Look for worksheets related to sales (such as SalesOrderHeader and SalesOrderDetail), production (such as WorkOrder and Product), purchasing (such as PurchaseOrderHeader and Vendor), and human resources (such as Employee and Department).

Step 5. Click on the Product worksheet in AdventureWorks.xlsx. Examine the columns, which include ProductID, Name, ProductNumber, StandardCost, ListPrice, ProductSubcategoryID, and several others. Notice that the Product table includes both cost and pricing information, which will be essential for profitability analysis in later chapters.

Step 6. Close AdventureWorks.xlsx and open ERPNext.xlsx. Look for worksheets that represent core accounting tables. You should find worksheets such as Account (the chart of accounts), GLEntry (general ledger entries), JournalEntry, SalesInvoice, PurchaseInvoice, PaymentEntry, and CostCenter. Click on the Account worksheet and examine the account



	A	B	C	D	E	F	G	H	I
1	OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	ShippedDate	ShipVia	Freight	ShipName
2	10248	VINET	5	1996-07-04 00:00:00	1996-08-01 00:00:00	1996-07-16 00:00:00	3	32.38	Vins et alcools Chevalier
3	10249	TOMSP	6	1996-07-05 00:00:00	1996-08-16 00:00:00	1996-07-10 00:00:00	1	11.61	Toms Spezialitäten
4	10250	HANAR	4	1996-07-08 00:00:00	1996-08-05 00:00:00	1996-07-12 00:00:00	2	65.83	Hanari Carnes
5	10251	VICTE	3	1996-07-08 00:00:00	1996-08-05 00:00:00	1996-07-15 00:00:00	1	41.34	Victuailles en stock
6	10252	SUPRD	4	1996-07-09 00:00:00	1996-08-06 00:00:00	1996-07-11 00:00:00	2	51.3	Suprêmes délices
7	10253	HANAR	3	1996-07-10 00:00:00	1996-07-24 00:00:00	1996-07-16 00:00:00	2	58.17	Hanari Carnes
8	10254	CHOPS	5	1996-07-11 00:00:00	1996-08-08 00:00:00	1996-07-23 00:00:00	2	22.98	Chop-suey Chinese
9	10255	RICSU	9	1996-07-12 00:00:00	1996-08-09 00:00:00	1996-07-15 00:00:00	3	148.33	Richter Supermarkt
10	10256	WELLI	3	1996-07-15 00:00:00	1996-08-12 00:00:00	1996-07-17 00:00:00	2	13.97	Wellington Importadora
11	10257	HILAA	4	1996-07-16 00:00:00	1996-08-13 00:00:00	1996-07-22 00:00:00	3	81.91	HILARION-Abastos
12	10258	ERNSH	1	1996-07-17 00:00:00	1996-08-14 00:00:00	1996-07-23 00:00:00	1	140.51	Ernst Handel
13	10259	CENTC	4	1996-07-18 00:00:00	1996-08-15 00:00:00	1996-07-25 00:00:00	3	3.25	Centro comercial Moctezuma
14	10260	OTTIK	4	1996-07-19 00:00:00	1996-08-16 00:00:00	1996-07-29 00:00:00	1	55.09	Ottilies Käseladen
15	10261	QUEDE	4	1996-07-19 00:00:00	1996-08-16 00:00:00	1996-07-30 00:00:00	2	3.05	Que Delícia
16	10262	RATTC	8	1996-07-22 00:00:00	1996-08-19 00:00:00	1996-07-25 00:00:00	3	48.29	Rattlesnake Canyon Grocery
17	10263	ERNSH	9	1996-07-23 00:00:00	1996-08-20 00:00:00	1996-07-31 00:00:00	3	146.06	Ernst Handel
18	10264	FOLKO	6	1996-07-24 00:00:00	1996-08-21 00:00:00	1996-08-23 00:00:00	3	3.67	Folk och fä HB
19	10265	BLONP	2	1996-07-25 00:00:00	1996-08-22 00:00:00	1996-08-12 00:00:00	1	55.28	Blondel père et fils
20	10266	WARTH	3	1996-07-26 00:00:00	1996-09-06 00:00:00	1996-07-31 00:00:00	3	25.73	Wartian Herkku
21	10267	FRANK	4	1996-07-29 00:00:00	1996-08-26 00:00:00	1996-08-06 00:00:00	1	208.58	Frankenversand
22	10268	GROSR	8	1996-07-30 00:00:00	1996-08-27 00:00:00	1996-08-02 00:00:00	3	66.29	GROSELLA-Restaurante
23	10269	WHITC	5	1996-07-31 00:00:00	1996-08-14 00:00:00	1996-08-09 00:00:00	1	4.56	White Clover Markets

Categories | CustomerCustomerDemo | CustomerDemographics | Customers | Employees | EmployeeTerritories | Order Details | Orders

Figure 7: Screenshot of the Northwind Orders worksheet in Excel.

names, account types, and parent account relationships. This is the chart of accounts for the ERPNext company, and it forms the backbone of the accounting module.

Step 7. Click on the GLEntry worksheet. Examine the columns, which should include fields such as PostingDate, Account, Debit, Credit, Voucher Type, and Voucher Number. This table contains the individual debit and credit entries that make up the general ledger. Every financial transaction in the ERPNext system is recorded here, and you will use this table extensively in the audit and financial reporting chapters.

Checkpoint. At this point, you should have opened all three workbooks and examined at least two worksheets in each one. You should be able to answer the following questions: Which dataset has the fewest tables? (Northwind.) Which dataset has tables related to manufacturing and production? (AdventureWorks.) Which dataset contains a chart of accounts and general ledger entries? (ERPNext.) If you can answer these three questions, you are ready to proceed.

How This Book Is Organized

This textbook is organized into five parts that follow the progression of the analytics workflow and introduce tools in a sequence designed for students with no prior analytics or programming experience.

Part I, Foundations of Accounting Analytics (Chapters 1 through 3), establishes the conceptual groundwork. You are in Part I now. Chapter 2 will teach you to think critically about accounting data, including its types, sources, and quality dimensions. Chapter 3 will introduce

the relational database model and show you how accounting data is organized within the three textbook databases.

Part II, Accounting Analytics with Excel (Chapters 4 through 8), teaches you to use Excel as an analytical tool. These chapters move from data organization and cleaning through summarization, modeling, and audit-specific analytics. By the end of Part II, you will be able to build PivotTables, run regressions, perform Benford's Law analysis, and prepare professional analytical workpapers.

Part III, Accounting Analytics with SQL (Chapters 9 through 12), teaches you to query relational databases using SQL. These chapters start with the basic SELECT statement and progress through joins, aggregation, window functions, and audit analytics queries. By the end of Part III, you will be able to write SQL queries that extract, join, summarize, and test accounting data directly from a database.

Part IV, Data Visualization and Power BI (Chapters 13 through 16), teaches you to build interactive reports and dashboards. These chapters begin with visualization principles and progress through Power BI basics, data modeling with DAX, and full dashboard design. By the end of Part IV, you will be able to build professional dashboards for financial reporting, operational management, and audit monitoring.

Part V, Integrated and Applied Topics (Chapters 17 through 20), brings everything together. These chapters apply the full toolkit to financial reporting analytics, cost and management accounting analytics, forensic accounting and fraud detection, and emerging technologies. The chapters in Part V require you to move between Excel, SQL, and Power BI within a single project, just as you would in professional practice.

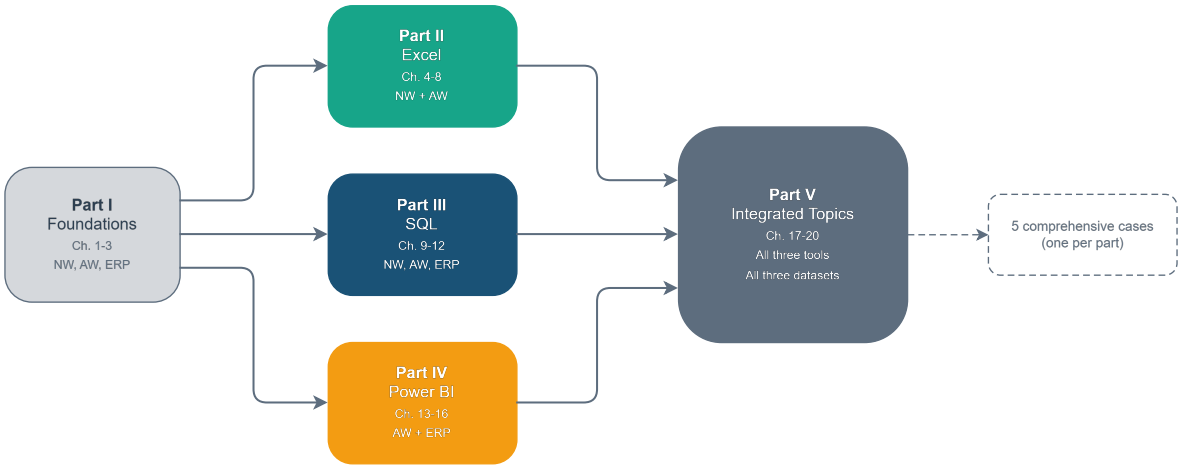


Figure 8: Book Organization Map.

Each chapter follows a consistent structure. It opens with learning objectives and a motivating scenario, presents concepts through narrative explanation and guided tutorials, and closes

with a summary, key terms, review questions, and applied exercises organized by three accounting perspectives: Financial Accounting, Managerial Accounting, and Auditing. Five comprehensive cases, one at the end of each part, provide extended multi-tool investigations that integrate the material from all chapters in that part.

> In Practice
>
> The three-perspective exercise structure in this book reflects how analytics is used across the accounting profession. The same technique, such as aging analysis, serves different purposes depending on the role. A financial accountant uses aging analysis to estimate the allowance for doubtful accounts. A managerial accountant uses it to assess collection efficiency and cash flow timing. An auditor uses it to evaluate management's estimates and to identify balances for confirmation testing. Seeing these connections will help you understand why a single analytical skill has broad professional value.

Looking Ahead

This chapter has introduced the concept of accounting analytics, defined the three types of analytics, described the tools and datasets you will use throughout the book, and outlined the analytical workflow that connects every chapter. In the next chapter, you will take a closer look at accounting data itself. You will learn to distinguish different data types, identify common data quality problems, and understand why the quality of your data determines the quality of your analysis. The practical skills begin in Chapter 2, and they build steadily from that point forward.

Chapter Summary

The accounting profession is undergoing a fundamental shift in the skills it demands from practitioners. The growth of transaction data, the evolution of audit methodology, and the expansion of the accountant's role from preparer to analyst have created strong demand for data analytics competencies. Professional bodies including the AICPA, IMA, and IFAC have responded by integrating analytics into their competency frameworks, signaling that these skills are essential rather than optional.

Analytics in accounting takes three forms. Descriptive analytics summarizes historical data to explain what happened. Predictive analytics uses historical patterns to forecast future outcomes. Prescriptive analytics recommends actions by optimizing objectives subject to

constraints. Most accounting analytics work is descriptive or predictive, and doing this work well requires both technical skill and professional judgment.

This textbook teaches accounting analytics using three tools (Excel, SQL, and Power BI) applied to three datasets (Northwind Traders, Adventure Works Cycles, and ERPNext Demo Company). The tools cover the full analytics workflow from data access through visualization and communication. The datasets represent three levels of complexity and three types of business environments, giving students a range of experience that prepares them for the variety they will encounter in practice.

Every analytical project follows a consistent workflow: define the question, access the data, prepare and clean the data, analyze it, visualize the results, and communicate findings. This workflow provides the organizing thread for the entire book. The chapters are arranged so that students develop foundational skills in Parts I through IV and then apply them in integrated, multi-tool projects in Part V.

Key Terms

Accounting analytics. The application of data extraction, preparation, analysis, and visualization techniques to financial and operational data for the purpose of supporting financial reporting, managerial decision-making, and audit assurance.

Analytics workflow. The sequence of stages that an accounting analytics project follows, from defining the question through communicating findings. The six stages are question definition, data identification, data preparation, analysis, visualization, and communication.

Chart of accounts. A structured list of all accounts used by an organization to record financial transactions, organized by account type (assets, liabilities, equity, revenue, expenses). In the ERPNext dataset, the chart of accounts is stored in the Account table.

Data analytics. A broad term for the process of examining data to draw conclusions, identify patterns, and support decisions. Accounting analytics is a specific application of data analytics to accounting data and questions.

Descriptive analytics. The type of analytics that summarizes and describes historical data to answer the question "what happened?" Examples in accounting include financial statement preparation, variance reports, and aging analyses.

Enterprise resource planning (ERP) system. An integrated software platform that organizations use to manage business processes across departments, including accounting, sales, purchasing, production, and human resources. ERPNext is the ERP-based dataset used in this textbook.

Entity-Relationship (ER) diagram. A visual representation of the tables in a database and the relationships between them. ER diagrams show how tables are connected through primary and foreign keys.

Foreign key. A column in a database table that references the primary key of another table, establishing a relationship between the two tables. For example, the CustomerID column in the Northwind Orders table is a foreign key that references the CustomerID column in the Customers table.

General ledger. The complete record of all financial transactions for an organization, organized by account. In the ERPNext dataset, the general ledger is represented by the GLEntry table.

Power BI. A business intelligence platform developed by Microsoft that enables users to connect to data sources, build data models, and create interactive visualizations and dashboards.

Predictive analytics. The type of analytics that uses historical data to forecast future outcomes and estimate probabilities. Examples in accounting include revenue forecasting, regression-based analytical procedures, and credit risk assessment.

Prescriptive analytics. The type of analytics that recommends actions by optimizing an objective subject to constraints. Examples in accounting include product mix optimization, tax strategy planning, and resource allocation modeling.

Primary key. A column (or combination of columns) in a database table that uniquely identifies each row. For example, OrderID is the primary key of the Northwind Orders table.

Relational database. A system for storing data in tables that are connected to one another through defined relationships. The three datasets in this textbook are provided as relational databases in SQLite format.

SQL (Structured Query Language). The standard language for accessing, querying, and manipulating data stored in relational databases. Part III of this textbook teaches SQL using SQLite databases.

SQLite. A lightweight relational database system that stores the entire database in a single file. This textbook uses SQLite because it requires no server installation and runs on any operating system.

Multiple Choice Questions

1. Which of the following best describes the primary reason that analytics skills have become essential for accounting professionals?

A. Accounting standards now require the use of data visualization tools in all financial reports.

B. The volume and complexity of financial and operational data have grown beyond what traditional manual methods can effectively analyze.

C. Analytics skills have replaced the need for accountants to understand generally accepted accounting principles.

D. Employers prefer to hire data scientists rather than accountants for financial reporting roles.

2. A management accountant prepares a report showing total manufacturing costs by product line for the past four quarters and calculates the percentage change from quarter to quarter. This activity is best classified as which type of analytics?

A. Prescriptive analytics

B. Predictive analytics

C. Descriptive analytics

D. Diagnostic analytics

3. An auditor builds a regression model using monthly revenue data from the prior year to establish an expected range for current-year monthly revenue. The auditor then compares actual monthly revenue to the expected range and investigates months where the difference exceeds the threshold. This procedure involves which type of analytics?

A. Descriptive analytics only

B. Predictive analytics

C. Prescriptive analytics

D. None of the above, because regression is a statistical technique rather than an analytics type

4. A cost accountant uses Excel's Solver tool to determine the combination of products that maximizes total contribution margin given constraints on machine hours and raw material availability. This activity is best classified as which type of analytics?

A. Descriptive analytics

B. Predictive analytics

C. Prescriptive analytics

D. Exploratory analytics

5. Which of the following is the correct sequence of stages in the accounting analytics workflow as described in this chapter?

A. Access data, define the question, analyze, clean, visualize, communicate

B. Define the question, access data, clean and prepare data, analyze, visualize, communicate

C. Analyze data, define the question, visualize, clean, communicate, access data

D. Define the question, visualize, access data, analyze, clean, communicate

6. Which of the three textbook datasets contains a chart of accounts and general ledger entries?

A. Northwind Traders

B. Adventure Works Cycles

C. ERPNext Demo Company

D. All three datasets contain a chart of accounts

7. Which tool covered in this textbook is most appropriate for extracting data directly from a relational database?

A. Microsoft Excel

B. SQL

C. Microsoft Power BI

D. The Data Analysis ToolPak

8. The Northwind dataset is introduced first in this textbook primarily because it:

A. Contains the most realistic accounting data of the three datasets

B. Is the only dataset that includes both Excel and SQLite formats

C. Is compact and simple enough for students to learn new tools without being overwhelmed by data complexity

D. Includes manufacturing and production data needed for early exercises

9. Which professional organization's competency framework has been updated to include data analytics as a core skill for management accountants?

A. The Financial Accounting Standards Board (FASB)

B. The Institute of Management Accountants (IMA)

C. The Securities and Exchange Commission (SEC)

D. The International Accounting Standards Board (IASB)

10. An accountant needs to create an interactive dashboard that allows multiple department managers to filter financial results to their own cost center and drill down into individual line items. Which tool is best suited for this purpose?

A. A SQL query

B. A static Excel spreadsheet

C. Microsoft Power BI

D. The Excel Data Analysis ToolPak

11. In the context of a relational database, a foreign key is best described as:

A. A column that uniquely identifies each row in its own table

B. A column in one table that references the primary key of another table, establishing a relationship between them

C. A password or security credential used to access the database

D. A column that stores encrypted financial data for confidentiality purposes

12. According to this chapter, which stage of the analytics workflow is typically the most time-consuming?

A. Defining the question

B. Data preparation and cleaning

C. Visualization

D. Communication of findings

13. The Adventure Works Cycles dataset is most valuable for managerial accounting exercises because it includes:

A. A complete chart of accounts and general ledger

B. Manufacturing cost data including work orders, bills of materials, and scrap tracking

C. Only sales and customer data

D. Budget records and cost center allocations

14. Which of the following best describes why this textbook provides datasets in both Excel and SQLite formats?

A. Excel files are used only for financial accounting exercises and SQLite files are used only for auditing exercises

B. Students can work with the same underlying data regardless of whether a given chapter uses Excel, SQL, or Power BI

C. SQLite databases are smaller than Excel files and require less storage space

D. Excel format is for undergraduate students and SQLite format is for graduate students

15. A financial analyst uses PivotTables to summarize total revenue by product category and fiscal quarter. The analyst notices that revenue in one category declined sharply in the third quarter and investigates the underlying transactions to determine the cause. The summarization step is an example of descriptive analytics. What type of analytics would applying a regression model to forecast fourth-quarter revenue in that category represent?

A. Descriptive analytics

B. Predictive analytics

C. Prescriptive analytics

D. The regression would not be considered analytics because it involves statistics

---

Applied Exercises

Financial Accounting Exercises

Exercise 1.1 (Financial Accounting): Mapping Datasets to the Financial Reporting Cycle

Dataset: All three (Northwind, AdventureWorks, ERPNext)

Scenario. You are a financial reporting analyst preparing a memorandum for your supervisor that assesses the analytical capabilities of three data sources your company has available. Your supervisor wants to understand which data source is best suited for specific financial reporting tasks.

Requirements. (1) Open each of the three textbook datasets in Excel and examine the available tables. For each dataset, identify which tables contain data that relates to revenue recognition, accounts receivable, inventory, and cost of goods sold. (2) For each of the four financial reporting areas listed above, determine which dataset provides the most complete data for analysis. Write a brief explanation (two to three sentences) for each area, identifying the specific tables you would use. (3) Identify one financial reporting task that can be performed with ERPNext data but cannot be performed with either Northwind or AdventureWorks data, and explain why. (4) Prepare a one-page summary memorandum presenting your findings in a format suitable for your supervisor.

Deliverable. A one-page written memorandum that compares the three datasets from a financial reporting perspective, identifying specific tables and explaining your reasoning.

Exercise 1.2 (Financial Accounting): Identifying Revenue Data Across Datasets

Dataset: Northwind and AdventureWorks

Scenario. You are a staff accountant asked to determine what revenue-related information is available in two of your company's data systems so that the team can plan a revenue trend analysis for the upcoming quarterly review.

Requirements. (1) Open both the Northwind and AdventureWorks datasets in Excel. Identify all tables that contain revenue or sales-related data. Record the table names and the column names that are most relevant to revenue measurement (such as amounts, dates, and product identifiers). (2) For each dataset, determine the total number of sales transactions recorded. Note the date range covered by the sales data. (3) Compare the two datasets in terms of the level of detail available for revenue analysis. Which dataset provides geographic (territory) information about sales? Which dataset provides product cost information alongside revenue? (4) Write a brief comparison (three to five sentences) explaining which dataset you would recommend for a detailed revenue trend analysis and why.

Deliverable. A written comparison of the revenue data available in each dataset, with specific table and column references.

Managerial Accounting Exercises

Exercise 1.3 (Managerial Accounting): Evaluating Datasets for Cost Analysis

Dataset: AdventureWorks and ERPNext

Scenario. You are a management accountant at a manufacturing company. Your controller has asked you to assess which of the company's data systems contains the information needed for a product cost analysis and a departmental budget review.

Requirements. (1) Open the AdventureWorks dataset in Excel and identify all tables that contain cost-related information. Look for tables related to product costs, work orders, production, and purchasing. List the table names and the columns most relevant to cost analysis. (2) Open the ERPNext dataset and identify all tables that contain budget and cost center information. List the table names and columns. (3) Determine which dataset is better suited for product-level cost analysis (comparing standard cost to actual cost for specific products) and which is better suited for departmental budget analysis (comparing budgeted amounts to actual spending by department). Explain your reasoning in three to five sentences. (4) Identify one managerial accounting question that requires data from both datasets to answer fully, and explain what tables from each dataset you would combine.

Deliverable. A written assessment that compares the managerial accounting capabilities of the two datasets, with specific references to tables and columns.

Exercise 1.4 (Managerial Accounting): Understanding Operational Data Structures

Dataset: AdventureWorks

Scenario. You have been asked to prepare a brief overview of the operational data available in the AdventureWorks database for a new manager who will be using this data to evaluate production performance.

Requirements. (1) Open AdventureWorks.xlsx and examine the worksheets related to production and manufacturing. Identify at least four tables that a production manager would find relevant. (2) For each table you identified, write one sentence describing what information it contains and how it might be used to evaluate production performance. (3) Identify the columns that appear in more than one table (these are the keys that connect the tables). Write two to three sentences explaining how these shared columns would allow you to combine information across tables for a more complete analysis. (4) Prepare a brief written overview (one half to one page) that the new manager could use as a reference when requesting reports.

Deliverable. A written overview of the production-related data in AdventureWorks, suitable for a non-technical manager audience.

Auditing Exercises

Exercise 1.5 (Auditing): Assessing Data Availability for Audit Testing

Dataset: All three (Northwind, AdventureWorks, ERPNext)

Scenario. You are a first-year auditor who has been assigned to plan the data analytics procedures for an upcoming engagement. Your senior has asked you to assess what data is available for three common audit tests: aging of receivables, testing for duplicate transactions, and journal entry testing for management override of controls.

Requirements. (1) For each of the three audit tests listed above, identify which of the three textbook datasets contains the data needed to perform the test. Be specific about which tables and columns you would use. (2) Determine which of the three datasets supports all three audit tests using a single data source. Explain your answer by identifying the specific tables. (3) Identify one limitation of the Northwind dataset for audit analytics purposes. What type of audit test cannot be performed with Northwind data, and why? (4) Write a planning memorandum (one page) that documents the data available for each test and recommends which dataset should be used for each purpose.

Deliverable. A one-page audit planning memorandum documenting data availability for three specified audit procedures.

Exercise 1.6 (Auditing): Evaluating Journal Entry Data for Audit Purposes

Dataset: ERPNext

Scenario. You are preparing for journal entry testing as part of an audit engagement. Before writing any queries or performing any analysis, you need to understand the structure of the journal entry data in the company's system.

Requirements. (1) Open ERPNext.xlsx and examine the JournalEntry and GLEntry worksheets. Identify all columns available in each table. (2) Determine which columns would be most useful for identifying journal entries that warrant audit attention. Consider columns related to the person who posted the entry, the date and time of posting, the amount, and any descriptions or references. (3) Write a brief assessment (three to five sentences) of whether the ERPNext data provides sufficient information to test for three common fraud risk indicators: entries posted outside of normal business hours, entries with round-dollar amounts, and entries posted just below an approval threshold. For any indicator where the data may be insufficient, explain what additional information would be needed. (4) Prepare a one-paragraph summary of your assessment for discussion with your audit senior.

Deliverable. A written assessment of the journal entry data structure in ERPNext, evaluating its suitability for audit testing of management override of controls.

Further Reading

Appelbaum, D., Kogan, A., and Vasarhelyi, M. A. (2017). Big data and analytics in the modern audit engagement: Research needs. *Auditing: A Journal of Practice and Theory*, 36(4), 1-27. This paper outlines a research agenda for data analytics in auditing and provides a framework for understanding how the analytical techniques taught in this textbook relate to the evolving audit process. The paper's discussion of population-based testing versus sampling is particularly relevant to the arguments presented in this chapter.

Earley, C. E. (2015). Data analytics in auditing: Opportunities and challenges. *Business Horizons*, 58(5), 493-500. This article provides an accessible overview of how audit practice is changing due to data analytics and identifies specific opportunities for auditors who develop analytical skills. It is useful reading for students interested in the audit perspective.

Kokina, J., and Dagiliene, L. (2017). The digitization of the accounting profession: Perceived threats and opportunities. *Journal of Emerging Technologies in Accounting*, 14(1), 1-13. This survey-based study examines how accounting professionals perceive the impact of digitization and analytics on their work. The findings support the claim made in this chapter that analytics skills are increasingly demanded across all areas of the profession.

Provost, F., and Fawcett, T. (2013). Data science and its relationship to big data and data-driven decision making. *Big Data*, 1(1), 51-59. This paper defines data science as a discipline and

describes the analytical workflow that underlies data-driven decision making. Although it addresses data science broadly rather than accounting specifically, the workflow it describes closely parallels the accounting analytics workflow presented in this chapter.

Richins, G., Stapleton, A., Stratopoulos, T. C., and Wong, C. (2017). Big data analytics: Opportunity or threat for the accounting profession? *Journal of Information Systems*, 31(3), 63-79. This paper examines the implications of big data for the accounting profession and discusses both the opportunities for accountants who develop analytics skills and the threats for those who do not. It provides evidence supporting the labor market arguments presented in this chapter.

Vasarhelyi, M. A., Kogan, A., and Tuttle, B. M. (2015). Big data in accounting: An overview. *Accounting Horizons*, 29(2), 381-396. This paper provides a comprehensive overview of how big data and analytics are affecting accounting practice, research, and education. It is one of the most cited papers on accounting analytics and offers valuable context for understanding why this textbook exists.

IFAC (International Federation of Accountants). (2019). *Technology and the profession: A guide for professional accountancy organizations.* IFAC. This guidance document describes the technology-related competencies that accounting graduates need and how professional bodies can integrate these competencies into their qualification frameworks. It supports the claim that analytics is recognized as a core competency by international standard-setting bodies.

Understanding Data in Accounting

Conceptual Chapter Northwind ERPNext

Learning Objectives

After completing this chapter, you will be able to:

1. Distinguish between quantitative and qualitative data and between structured and unstructured data, and identify examples of each in accounting contexts.
2. Describe the primary sources of accounting data, including general ledgers, sub-ledgers, trial balances, enterprise resource planning systems, and external data feeds.
3. Evaluate accounting data against four quality dimensions: accuracy, completeness, consistency, and timeliness.
4. Explain the concept of tidy data and why it matters for accounting analytics.
5. Identify common data quality problems in the Northwind and ERPNext datasets, including missing values, duplicates, inconsistent formatting, and outliers.

Opening Scenario

You are a newly hired internal auditor at a mid-size wholesale distribution company that recently migrated its accounting records from a legacy system to a new enterprise resource planning platform. Your supervisor has asked you to verify that the migrated data is reliable enough to support the upcoming year-end audit. When you begin examining the general ledger entries, you notice that some account names use abbreviations while others are spelled out in full. Several transactions are missing posting dates. A handful of customer records appear more than once under slightly different names. Before the audit team can use this data to test account balances, perform analytical procedures, or identify anomalies, someone needs to assess the quality of the data itself. Your supervisor has asked you to prepare that

assessment. This chapter will give you the vocabulary, the framework, and the practical skills to evaluate whether accounting data is fit for analysis.

Why Data Quality Comes First

Chapter 1 introduced the six-stage accounting analytics workflow, beginning with defining the question and ending with communicating findings. This chapter focuses on a concern that cuts across every stage of that workflow: the quality of the underlying data. No analytical technique, no matter how sophisticated, can produce reliable results if the data it operates on contains errors, gaps, or inconsistencies. A PivotTable built on revenue data with missing transaction dates will produce period totals that understate actual performance. A SQL query designed to detect duplicate payments will miss duplicates if vendor names are recorded inconsistently across tables. A Power BI dashboard showing budget-versus-actual comparisons will mislead managers if the budget data and the actual data use different account classifications.

The principle is straightforward. The quality of every analytical output is constrained by the quality of the data that feeds it. Researchers in information systems have formalized this idea by studying data quality as a measurable property of datasets, with dimensions that can be assessed, monitored, and improved (Wang and Strong, 1996). For accountants, data quality is not an abstract concern. It is a professional responsibility. Auditing standards require auditors to evaluate the reliability of data used in analytical procedures. Management accountants who present cost analyses to executives are implicitly vouching for the integrity of the numbers. Financial reporting analysts who prepare disclosures from database extracts must ensure that the underlying data supports the figures reported. Understanding what data quality means, how to assess it, and what problems to look for is therefore a prerequisite for everything else in this textbook.

Data Types in Accounting

Before you can evaluate the quality of a dataset, you need to understand the types of data it contains. Accounting data takes many forms, and the appropriate way to store, analyze, and interpret a piece of data depends on what type it is. Two classification schemes are particularly useful for accounting analytics: the distinction between quantitative and qualitative data, and the distinction between structured and unstructured data.

Quantitative and Qualitative Data

Quantitative data consists of values that represent measurements or counts and that can be meaningfully subjected to arithmetic operations. In accounting, the most familiar quantitative data includes transaction amounts (debits, credits, invoice totals, payment amounts), quantities (units produced, units sold, units in inventory), and calculated values (ratios, percentages, variances). Quantitative data can be further divided into continuous data, which can take any value within a range (such as a transaction amount of $1,247.83), and discrete data, which takes only specific values (such as the number of journal entries posted in a given month).

Qualitative data consists of values that represent categories, labels, or descriptions rather than measurements. In accounting, qualitative data includes account names, customer names, vendor classifications, transaction descriptions, product categories, and status codes such as "paid" or "outstanding." Qualitative data can be nominal, meaning the categories have no natural order (such as the names of cost centers), or ordinal, meaning the categories have a meaningful sequence (such as a credit rating scale from AAA to D).

The distinction matters because it determines which analytical techniques are appropriate. You can calculate an average transaction amount, but you cannot calculate an average account name. You can count the number of transactions in each product category, but the category labels themselves are not quantities. Many data quality problems arise when qualitative data is treated as quantitative or when quantitative data is stored in a format that prevents arithmetic operations. A common example is a transaction amount stored as text rather than as a number because the original data source included a currency symbol or a comma in the value. The number looks correct to a human reader, but Excel and SQL cannot perform calculations on it until the formatting is corrected.

[Table 2.1: Examples of Quantitative and Qualitative Data in Accounting. A table with three columns (Data Type, Accounting Example, Typical Source) and six rows showing examples such as transaction amounts from the general ledger (continuous quantitative), unit quantities from inventory records (discrete quantitative), account names from the chart of accounts (nominal qualitative), and aging categories from receivables schedules (ordinal qualitative).]

Structured and Unstructured Data

A second important distinction is between structured and unstructured data. Structured data is organized into a predefined format, typically rows and columns in a table, where each column has a defined data type and each row represents a single observation or transaction. All three datasets used in this textbook contain structured data. The Northwind Orders table, for example, has columns for OrderID, CustomerID, OrderDate, and Freight, and each row represents one customer order. Structured data is the primary input for the analytical techniques covered in this book, including PivotTables in Excel, SQL queries, and Power BI data models.

Unstructured data lacks a predefined tabular format. In accounting, unstructured data includes the text of contracts, the narrative sections of annual reports, email correspondence between auditors and clients, scanned images of invoices, and the notes that accountants attach to journal entries. Unstructured data often contains valuable information, but extracting that information requires different tools and techniques than those used for structured data. Natural language processing, a branch of artificial intelligence, is increasingly used to analyze unstructured accounting documents such as lease agreements and regulatory filings (Fisher, Garnsey, and Hughes, 2016). Chapter 20 discusses these emerging technologies in more detail.

Between these two extremes lies semi-structured data, which has some organizational elements but does not fit neatly into rows and columns. An example relevant to accounting is an XBRL (eXtensible Business Reporting Language) filing, which contains financial data tagged with standardized labels but organized in a hierarchical rather than tabular structure. Another example is a bank statement in PDF format that contains tabular transaction data embedded within an unstructured document layout.

[Figure 2.1: The Spectrum of Data Structure. A horizontal diagram showing three zones from left to right: Structured Data (example: a general ledger table with defined columns), Semi-Structured Data (example: an XBRL filing with tagged financial elements), and Unstructured Data (example: a contract document in plain text). Each zone includes a brief description and an accounting example.]

> **IN PRACTICE** Most of the data that accountants work with in practice is structured. General ledgers, trial balances, accounts receivable aging schedules, and payroll registers are all structured datasets stored in rows and columns within ERP systems or accounting software. However, the proportion of unstructured data in accounting work is growing as firms adopt tools for contract analysis, disclosure review, and automated document processing. Developing strong skills with structured data, as this textbook teaches, provides the foundation for working with less structured data as your career progresses.

This textbook focuses almost entirely on structured data because it represents the majority of what accountants analyze and because the tools covered here (Excel, SQL, and Power BI) are designed for structured datasets. The datasets you will work with throughout the book are all structured, with clearly defined tables, columns, and data types. Understanding this foundation is essential before you can evaluate whether a dataset is ready for analysis.

Sources of Accounting Data

Accounting data originates from many places within and outside an organization. Understanding where data comes from helps you anticipate its strengths and limitations, which in turn helps you assess its quality and suitability for a given analytical task. This section describes the most common sources of accounting data that you will encounter in practice.

The General Ledger and Sub-Ledgers

The general ledger is the central repository of an organization's financial transactions. Every debit and credit entry that affects the financial statements passes through the general ledger. In a database environment, the general ledger is typically stored as a table where each row represents one side of a journal entry (a single debit or credit), and columns record the posting date, the account number, the amount, a description, and a reference to the source document or voucher. The ERPNext dataset used in this textbook stores its general ledger in the GLEntry table, which follows exactly this structure.

Sub-ledgers provide detailed records for specific account categories. The accounts receivable sub-ledger, for example, contains individual customer balances and the invoices and payments that make up each balance. The accounts payable sub-ledger contains individual vendor balances. The inventory sub-ledger contains records of individual stock items, their quantities, and their costs. Sub-ledgers feed summary totals to the general ledger, and the balances in the two should agree. When they do not, the discrepancy becomes an audit finding. In the Northwind dataset, the Orders and OrderDetails tables function as a sales sub-ledger, recording individual transactions at a level of detail that a general ledger summary would not capture.

Trial Balances

A trial balance is a listing of all general ledger account balances at a specific point in time. It serves as a control to verify that total debits equal total credits and as the starting point for preparing financial statements. In an analytics context, the trial balance is useful as a summary dataset that provides a high-level view of the organization's financial position. Auditors frequently use the trial balance as the population from which they select accounts for testing. The trial balance is not a separate table in most databases but rather a report generated by summing the general ledger entries for each account.

Enterprise Resource Planning Systems

An enterprise resource planning system integrates data from across an organization's functional areas into a single database. A typical ERP system captures not only financial transactions but also sales orders, purchase orders, production schedules, inventory movements, human resources records, and customer relationship management data. This integration means that an accountant with access to the ERP database can trace a sales transaction from the customer order through the shipment, the invoice, the revenue recognition entry in the general ledger, and the cash receipt. The ERPNext dataset in this textbook simulates this integrated environment, with tables spanning the full cycle from sales invoices through general ledger entries and payment records.

ERP systems are the dominant data source in large and mid-size organizations. They produce structured data with defined fields, data types, and validation rules. However, ERP data is not immune to quality problems. Data entry errors, system configuration mistakes, and gaps in validation rules can all introduce inaccuracies that propagate through the system. One of the advantages of the analytical skills taught in this textbook is the ability to identify these problems by examining the data directly rather than relying solely on the reports the ERP system generates (Grabski, Leech, and Schmidt, 2011).

[Figure 2.2: Common Sources of Accounting Data. A diagram showing five data sources (General Ledger, Sub-Ledgers, Trial Balance, ERP System, and External Feeds) connected by arrows to a central box labeled "Accounting Analytics." Each source box includes a one-sentence description of the type of data it provides.]

External Data Sources

Accountants also work with data that originates outside the organization. External data sources include bank statements used for cash reconciliation, market price feeds used for fair value measurements, credit rating data used for impairment assessments, tax rate tables used for compliance calculations, and industry benchmarking data used for comparative analysis. External data introduces additional quality concerns because the accountant has less control over how the data was collected, formatted, and maintained. When external data is combined with internal data for analysis, differences in formatting, time periods, or classification schemes can create inconsistencies that must be resolved before the analysis can proceed.

> **CONNECTING THE DOTS** In Chapter 5, you will use Excel lookup functions and Power Query to merge data from different sources, such as combining Northwind order data with customer details from a separate table. The challenges you encounter in that chapter, including mismatched identifiers and inconsistent formatting, are practical examples of the data quality issues discussed here. Understanding the sources of accounting data now will help you anticipate those challenges when you begin working with the tools.

Data Quality Dimensions

Data quality is not a single characteristic that data either has or lacks. Researchers have identified multiple dimensions of data quality, each describing a different aspect of what makes data fit for its intended use. The framework developed by Wang and Strong (1996) identified more than a dozen dimensions, but four are particularly important for accounting analytics: accuracy, completeness, consistency, and timeliness. These four dimensions provide a practical vocabulary for describing data problems and assessing whether a dataset is suitable for a given analytical task.

Accuracy

Accuracy means that the recorded values correspond to the true values they are intended to represent. A transaction amount of $5,000 is accurate if the actual transaction was indeed $5,000. An account classification of "current asset" is accurate if the account genuinely meets the definition of a current asset under the applicable accounting standards. Accuracy is the most fundamental quality dimension because inaccurate data leads directly to incorrect analytical results.

In accounting datasets, accuracy problems arise from data entry errors (typing $50,000 instead of $5,000), calculation errors (applying the wrong exchange rate to a foreign currency transaction), classification errors (posting an expense to the wrong account), and measurement errors (using an incorrect cost allocation formula). Some accuracy problems are obvious when you inspect the data, such as a negative value in a field that should always be positive. Others are subtle and can only be detected by comparing the data to an independent source or by applying analytical tests such as reasonableness checks.

Completeness

Completeness means that all expected data values are present. A dataset is complete if it contains all the records that should be there and if every field within each record has a value where one is expected. Completeness problems take two forms. The first is missing records, where entire transactions or entities are absent from the dataset. The second is missing values, where individual fields within existing records are blank or null.

In accounting, completeness is a significant concern for both preparers and auditors. A revenue dataset that is missing the last three days of the fiscal period will understate total revenue. An accounts payable listing that omits recently recorded invoices will understate total liabilities. Auditors specifically test for completeness because management has an incentive to understate liabilities and overstate assets, and missing data can serve that purpose either intentionally or accidentally. In the Northwind dataset, you will discover that some orders have missing shipped dates, which raises the question of whether those orders were ever fulfilled.

Consistency

Consistency means that the same fact is represented in the same way wherever it appears. A customer named "Acme Corporation" in the accounts receivable table should not appear as "ACME Corp." in the sales order table and "Acme Corp" in the payment records table. An account coded as "4100" in the chart of accounts should be coded as "4100" everywhere it is referenced in the general ledger, not as "4100.0" in some entries and "41-00" in others.

Consistency problems are among the most common data quality issues in accounting datasets, and they are particularly troublesome because they can cause analytical results to be silently wrong rather than obviously wrong. If a SQL query groups revenue by customer name and the same customer appears under three different name variations, the query will produce three separate line items instead of one. The total revenue figure will be correct, but the per-customer breakdown will be misleading. Consistency problems are especially prevalent when data comes from multiple sources or when an organization has undergone a system migration (Redman, 2001).

Timeliness

Timeliness means that the data reflects the current state of the business as of the relevant reporting date. A balance sheet analysis performed on general ledger data that has not been updated to include the final adjusting entries of the period will produce inaccurate results. A receivables aging analysis based on data that is two weeks old may overstate the number of delinquent accounts if payments have been received in the interim.

Timeliness is a quality dimension that depends on context. Data that is perfectly timely for a monthly management report may be inadequate for a daily cash position analysis. The analytical question determines what level of timeliness is required. Accountants must assess whether the data they are working with is current enough for the purpose at hand and, if it is not, either obtain more current data or disclose the limitation in their analysis.

[Table 2.2: The Four Data Quality Dimensions with Accounting Examples. A table with four rows (Accuracy, Completeness, Consistency, Timeliness) and three columns (Dimension, Definition, Accounting Example). Each row includes a concrete example of a quality problem drawn from common accounting scenarios.]

> **WATCH OUT** A dataset can score well on one quality dimension while failing on another. The general ledger may be perfectly accurate (every amount matches the source document) and perfectly consistent (every account code follows the same format) but incomplete because transactions from a subsidiary were not yet consolidated. Evaluating data quality requires examining all four dimensions, not just the one that is easiest to check.

Common Data Quality Problems in Accounting Datasets

The four quality dimensions provide a framework for thinking about data quality, but in practice, accountants encounter specific types of problems that recur across organizations and systems. Learning to recognize these problems is the first step toward resolving them. This section describes the most common data quality problems you will encounter when working with accounting data.

Missing Values

Missing values occur when a field that should contain data is blank, null, or contains a placeholder such as "N/A" or "TBD." In accounting datasets, missing values often appear in date fields (a payment date that was never recorded), in descriptive fields (a journal entry with no memo), and in classification fields (a transaction that was never assigned to a cost center). Missing values can affect analysis in several ways. Some analytical tools exclude records with missing values from calculations, which can distort totals and averages. Other tools treat missing values as zeros, which produces a different type of distortion.

Duplicate Records

Duplicate records occur when the same transaction, entity, or event appears more than once in a dataset. In accounts payable, a duplicate payment means the organization paid the same invoice twice. In a customer master file, a duplicate customer record means the same customer has two identities in the system, which splits their transaction history and makes it difficult to assess total purchasing volume or outstanding receivables. Duplicates can result from data entry errors, system integration problems, or the failure of validation controls. Detecting duplicates is an important audit analytics procedure, and you will perform duplicate detection exercises in several chapters of this textbook.

Inconsistent Formatting

Inconsistent formatting occurs when the same type of data is recorded in different formats across records or across tables. Date formatting is a frequent source of inconsistency. One table may store dates as "2024-03-15" while another stores them as "03/15/2024" or "15-Mar-2024." Name formatting is another common issue. Vendor names may appear with or without legal suffixes ("Inc.", "LLC"), with different capitalization, or with abbreviated words. Currency amounts may include or exclude currency symbols, and numeric values may use commas or periods as decimal separators depending on regional settings.

Inconsistent formatting is more than a cosmetic problem. When data from two tables must be matched or merged, formatting differences can prevent correct joins and lookups. An XLOOKUP formula in Excel or a JOIN clause in SQL that matches on customer name will fail to connect records if the name is spelled differently in the two tables. Chapters 5 and 10 of this textbook teach specific techniques for standardizing formatted data before analysis.

Outliers

Outliers are values that fall far outside the expected range for a given field. A single invoice for $2,000,000 in a dataset where the average invoice is $5,000 may be a legitimate large transaction,

or it may be a data entry error where extra zeros were added accidentally. A journal entry posted at 3:00 AM may be a legitimate automated posting, or it may indicate unauthorized activity. Outliers require investigation rather than automatic removal. The appropriate response depends on the context and on the accountant's professional judgment.

In audit analytics, outliers are often the transactions of greatest interest. Auditors look for unusual transactions because they may indicate errors, fraud, or control failures. The statistical and analytical techniques covered in later chapters, including stratification in Chapter 6, Benford's Law analysis in Chapter 8, and anomaly detection queries in Chapter 12, are all designed to help you identify and evaluate outliers systematically.

[Figure 2.3: Common Data Quality Problems Illustrated. A four-panel diagram with one panel for each problem type (Missing Values, Duplicates, Inconsistent Formatting, Outliers). Each panel shows a small sample of data with the problem highlighted and a brief annotation explaining what is wrong. The examples use Northwind-style data to maintain continuity with the textbook datasets.]

> **IN PRACTICE** In professional practice, data quality assessment is often the first task in any analytics engagement. Audit teams perform data quality checks before running analytical procedures to ensure that the results will be reliable. Management accountants validate the data behind cost reports before presenting them to executives. The AICPA's Guide to Audit Data Analytics recommends that auditors evaluate the completeness and accuracy of data obtained from client systems before using it in substantive procedures (AICPA, 2017). The data quality skills you develop in this chapter apply directly to that professional requirement.

The Concept of Tidy Data

In 2014, the statistician Hadley Wickham published a paper that formalized a set of principles for organizing datasets in a way that makes them easy to analyze. He called datasets that follow these principles "tidy" and those that violate them "messy" (Wickham, 2014). The concept of tidy data has since become widely adopted in data analysis across many fields, and it is directly relevant to accounting analytics.

A tidy dataset has three properties. First, each variable occupies its own column. A variable is any attribute that is measured or recorded, such as transaction date, account number, or debit amount. Second, each observation occupies its own row. An observation is a single instance of what is being measured, such as one line item in a journal entry or one sales transaction. Third, each type of observational unit forms its own table. Customer information belongs in a customer table, order information belongs in an order table, and product information belongs in a product table.

These principles may seem obvious, but violations are surprisingly common in accounting data. A frequent violation is storing multiple variables in a single column. For example, a

spreadsheet might have a column called "Account" that contains entries like "4100-Sales-Domestic" where the account number, account name, and geographic segment are all packed into one field. Analyzing sales by geographic segment requires splitting this column into three separate columns before any aggregation can be performed.

Another common violation is storing observations in column headers rather than in rows. A budget spreadsheet might have columns labeled "January," "February," "March," and so on, with each row representing a department and the values representing that department's budgeted amount for each month. This layout is easy for humans to read but difficult for analytical tools to process. A tidy version of the same data would have three columns (Department, Month, and Budget Amount) with one row for each department-month combination. This format, sometimes called "long" format as opposed to the "wide" format of the original, is what PivotTables, SQL GROUP BY queries, and Power BI data models require.

[Figure 2.4: Messy Versus Tidy Data. A side-by-side comparison showing the same budget data in two layouts. The left panel shows the "wide" format with months as column headers (messy). The right panel shows the "long" format with Department, Month, and Budget Amount as columns (tidy). Annotations explain why the tidy format is preferred for analysis.]

> **WATCH OUT** Students sometimes confuse "tidy" with "clean." A tidy dataset is one that follows the structural principles described above: each variable in its own column, each observation in its own row, each type of observational unit in its own table. A clean dataset is one that is free of quality problems such as missing values, duplicates, and inconsistencies. A dataset can be tidy but dirty (properly structured but containing errors) or clean but messy (free of errors but organized in a way that makes analysis difficult). Both structure and quality must be addressed before analysis can proceed.

The three textbook datasets are, for the most part, already tidy. They are stored as relational database tables where each column represents a single variable and each row represents a single observation. This is by design: relational databases enforce the structural principles of tidy data because their table definitions require each column to have a name, a data type, and a single purpose. When you import these datasets into Excel, however, you may encounter situations where the data needs to be restructured, particularly when working with summary reports that were designed for human readers rather than analytical tools. Chapter 5 introduces Power Query, which provides a systematic way to reshape data from messy to tidy formats.

> **CONNECTING THE DOTS** The tidy data principles described here are closely related to the concept of database normalization, which you will study in Chapter 3. Both tidy data and normalization seek to organize data so that each fact is stored once, in a single defined location, with no redundancy or ambiguity. Understanding tidy data now will make the transition to relational database concepts in the next chapter feel natural.

Mapping the Textbook Datasets to Real-World Data Sources

The three datasets used in this textbook represent different types of accounting data environments, and understanding how they map to real-world sources will help you apply the concepts from this chapter in practice.

The Northwind Traders dataset represents the type of data you would find in a small company's order management system. It records sales transactions, customer information, product details, and supplier records. In a real company of similar size, this data might reside in a standalone accounting package or a small-business ERP system. The data is relatively simple: a few hundred customers, a few thousand orders, and straightforward relationships between tables. Northwind does not include a general ledger or a chart of accounts, which means it cannot support financial statement preparation on its own. It functions as a sub-ledger system focused on the revenue and purchasing cycles.

The AdventureWorks Cycles dataset represents the type of data you would find in a mid-size manufacturing company's departmental databases. It includes tables for sales, production, purchasing, and human resources, each organized within its own schema (a logical grouping of related tables within a database). In a real company, these data might come from separate modules within an ERP system or from several integrated applications. The AdventureWorks data is more complex than Northwind because it captures manufacturing processes, including work orders, bills of materials, production routing, and scrap records. This complexity introduces additional data quality challenges, such as ensuring that cost records are consistent across the production and accounting modules.

The ERPNext Demo Company dataset represents the type of data you would find in a full enterprise resource planning system with an integrated accounting module. It includes a chart of accounts, general ledger entries, journal entries, sales and purchase invoices, cost centers, and budget records. In a real company, this data would serve as the authoritative source for financial reporting. The ERPNext dataset is the most accounting-rich of the three, and it mirrors the data structures that auditors and financial analysts encounter when working with production ERP systems. The general ledger entries in ERPNext record every debit and credit that affects the financial statements, and the chart of accounts provides the classification structure that organizes those entries into meaningful categories.

[Table 2.3: Mapping Textbook Datasets to Real-World Data Sources. A table with four columns (Feature, Northwind Traders, AdventureWorks Cycles, ERPNext Demo Company) and five rows (Real-World Equivalent, Accounting Module Depth, Primary Data Quality Challenges, Most Relevant For, Chapters Where Quality Issues Are Explored). Each cell contains a brief description.]

Understanding these mappings helps you connect the exercises in this textbook to the data environments you will encounter after graduation. When you clean and validate Northwind order data in the tutorials that follow, you are practicing the same skills you would use when working with transaction exports from a client's accounting system. When you assess the

completeness and consistency of ERPNext general ledger data, you are performing the same type of evaluation that an audit team performs at the start of every engagement.

Guided Tutorial 2.1: Inspecting the Northwind Dataset for Data Quality Problems

Context and objective. In this tutorial, you will examine the Northwind dataset in Excel and identify specific data quality problems across several tables. The goal is to practice applying the four quality dimensions (accuracy, completeness, consistency, and timeliness) to a real dataset and to document what you find. This tutorial connects to the opening scenario by showing you how a data quality assessment works in practice.

Prerequisites. You need Microsoft Excel installed on your computer and access to the Northwind.xlsx file provided with this textbook.

Step-by-step instructions.

Step 1. Open Northwind.xlsx in Excel and click on the Orders worksheet. Scroll across the columns to familiarize yourself with the available fields. You should see columns including OrderID, CustomerID, EmployeeID, OrderDate, RequiredDate, ShippedDate, ShipVia, Freight, ShipName, ShipAddress, ShipCity, ShipRegion, ShipPostalCode, and ShipCountry.

Step 2. Click on the ShippedDate column header to select the entire column. Use Excel's Find and Select feature (Ctrl+F on Windows, Cmd+F on Mac) to search for blank cells, or use the filter dropdown to look for blanks. Count the number of orders that have no ShippedDate value. These missing values represent a completeness problem: orders that were placed but for which no shipment date was recorded. Record the count.

[Figure 2.5: Screenshot of the Northwind Orders table in Excel with the ShippedDate column filtered to show blank cells. An annotation highlights the missing values and identifies this as a completeness problem.]

Step 3. With the Orders worksheet still active, examine the ShipRegion column using the same approach. Filter for blank values. You will find that many orders have no value in the ShipRegion field. Consider whether this represents a true completeness problem (the data should exist but was not entered) or whether some countries do not use region designations. This distinction matters because the appropriate response differs. If the data should exist, it needs to be obtained. If the field is not applicable for certain records, the blank is acceptable.

Step 4. Click on the Customers worksheet. Examine the CompanyName column. Look for potential consistency problems by scanning for variations in how company names are formatted. Check whether any company names appear to be near-duplicates (similar names with slight differences in punctuation, abbreviation, or spacing). You can sort the column alphabetically to make near-duplicates easier to spot.

Step 5. Return to the Orders worksheet. Create a simple check for potential accuracy problems by examining the Freight column. Sort the Freight column from largest to smallest. Examine the highest values. Are any freight amounts unusually large relative to the others? Make a note of any values that appear to be outliers. In a real engagement, you would investigate these further to determine whether they are legitimate transactions or data entry errors.

Step 6. Examine the relationship between OrderDate, RequiredDate, and ShippedDate. For records that have all three dates populated, check whether the dates are in a logical sequence: OrderDate should come first, followed by ShippedDate, and RequiredDate should not precede OrderDate. You can create a simple formula in a new column to test this. In cell P2, enter a formula that returns "Check" if ShippedDate is earlier than OrderDate, and "OK" otherwise. Copy the formula down the column and filter for any "Check" results. These would indicate a potential accuracy problem in the date fields.

[Figure 2.6: Screenshot showing a helper column added to the Northwind Orders table that flags records where ShippedDate precedes OrderDate. The formula is visible in the formula bar, and the annotation identifies this as an accuracy check.]

Step 7. Click on the Products worksheet. Examine the UnitPrice column. Sort by UnitPrice from smallest to largest. Check whether any products have a unit price of zero, which could indicate a missing value recorded as zero rather than left blank. Also check the UnitsInStock column for any negative values, which would be physically impossible and would represent an accuracy problem.

Checkpoint. At this point, you should have identified at least three specific data quality problems in the Northwind dataset: missing ShippedDate values (completeness), potential inconsistencies in company names or regional data (consistency), and at least one instance worth investigating further in the date sequence or freight amounts (accuracy). You should have a written list of findings, noting the table, the column, the type of problem, and the number of affected records. This list is a simple version of the data quality assessment that auditors perform at the start of every analytics engagement.

Guided Tutorial 2.2: Examining ERPNext General Ledger Data for Completeness and Consistency

Context and objective. In this tutorial, you will examine the ERPNext dataset to assess whether the general ledger entries and chart of accounts are complete and consistent. This is a more accounting-specific data quality assessment than the Northwind tutorial, because it involves checking whether the accounting records themselves are internally coherent. The skills practiced here connect directly to the audit procedures you will learn in Chapter 8 and Chapter 12.

Prerequisites. You need Microsoft Excel installed on your computer, access to the ERPNext.xlsx file, and completion of Tutorial 2.1.

Step-by-step instructions.

Step 1. Open ERPNext.xlsx in Excel and click on the Account worksheet (the chart of accounts). Examine the columns available. You should see fields including account name, account number or code, account type (such as Asset, Liability, Equity, Income, or Expense), and a parent account field that defines the account hierarchy. Scroll through the list to get a sense of how many accounts exist and how they are organized.

Step 2. Check the chart of accounts for completeness. Every account type should be represented. Sort or filter the account type column and verify that the chart includes accounts classified as assets, liabilities, equity, income, and expenses. If any major category is missing, that is a completeness problem. Record your findings.

Step 3. Click on the GLEntry worksheet (the general ledger). Examine the columns available. You should see fields including a posting date, an account reference, debit and credit amounts, a voucher type (indicating the source document, such as "Journal Entry" or "Sales Invoice"), and a voucher number. Scroll down to get a sense of the volume of data.

Step 4. Check the general ledger for a fundamental accounting consistency requirement: for every transaction, total debits should equal total credits. To test this at the aggregate level, use the SUM function to calculate the total of the Debit column and the total of the Credit column. The two totals should be equal, or very nearly so (small differences may arise from rounding). If there is a significant difference, that indicates a serious consistency problem in the ledger.

[Figure 2.7: Screenshot of the ERPNext GLEntry worksheet with SUM formulas calculating total debits and total credits at the bottom of the respective columns. An annotation highlights the comparison and explains that equal totals confirm the basic debit-equals-credit consistency of the ledger.]

Step 5. Check for accounts referenced in the general ledger that do not exist in the chart of accounts. This is a consistency check between two related tables. To perform this check manually, use the COUNTIFS function. In a new column adjacent to the GLEntry data, enter a formula that counts how many times the account value in the current GL row appears in the Account worksheet's account column. If any GLEntry row returns a count of zero, that means the entry references an account that does not exist in the chart of accounts, which is a consistency problem. Scan for any zeros and record the affected entries.

Step 6. Examine the PostingDate column in the GLEntry worksheet. Sort by posting date and check the date range. Note the earliest and latest dates. Check whether there are any gaps in the sequence of months, which could indicate missing periods. Also check whether any entries have dates that fall outside the expected fiscal year, which would be an accuracy concern.

Step 7. Examine the Debit and Credit columns for missing or unusual values. Filter for rows where both the Debit and Credit fields are zero or blank. An entry with no debit and no credit

amount serves no accounting purpose and may indicate a data entry error or a system artifact. Record the count of such entries if any exist.

Checkpoint. At this point, you should have assessed the ERPNext data against three quality dimensions: completeness (all account types present in the chart of accounts, no gaps in the posting date range), consistency (debits equal credits in the aggregate, all GL accounts exist in the chart of accounts), and accuracy (no entries outside the expected date range, no zero-amount entries without explanation). Your findings should be documented in a brief written list that identifies each test performed, the result, and any issues discovered. This exercise previews the audit analytics work you will perform in Chapters 8 and 12, where you will use SQL to automate these same tests across the entire population.

Looking Ahead

This chapter has provided the vocabulary and the framework you need to think critically about accounting data before you begin analyzing it. You can now distinguish between quantitative and qualitative data, between structured and unstructured data, and between data that is tidy and data that needs restructuring. You know the primary sources of accounting data and the four dimensions against which data quality is measured. You have practiced inspecting two of the textbook datasets for specific quality problems and documenting your findings.

In the next chapter, you will build on this foundation by studying how accounting data is organized within relational databases. Chapter 3 introduces the concepts of tables, keys, and relationships that make it possible to connect data across different parts of an accounting system. Understanding these structures will prepare you for the SQL chapters in Part III and for the data modeling work you will do in Power BI in Part IV.

Chapter Summary

Every accounting analytics project depends on the quality of the data that feeds it. Before applying any analytical technique, accountants must understand what types of data they are working with, where the data came from, and whether it is fit for the intended purpose. This chapter established the conceptual foundation for that assessment.

Accounting data can be classified along two dimensions. The quantitative versus qualitative distinction determines whether arithmetic operations are appropriate. The structured versus unstructured distinction determines which analytical tools can be applied. The datasets used in this textbook contain structured data organized in rows and columns, which is the format

required by Excel, SQL, and Power BI. Accounting data originates from general ledgers, subledgers, trial balances, enterprise resource planning systems, and external sources. Each source has characteristic strengths and limitations that affect data quality.

Data quality is assessed along four dimensions: accuracy, completeness, consistency, and timeliness. Accuracy means that recorded values correspond to true values. Completeness means that all expected records and fields are present. Consistency means that the same fact is represented the same way wherever it appears. Timeliness means that the data reflects the relevant reporting period. Common data quality problems include missing values, duplicate records, inconsistent formatting, and outliers. Each of these problems can distort analytical results if not identified and addressed before analysis begins.

The concept of tidy data provides a structural standard for how datasets should be organized: each variable in its own column, each observation in its own row, and each type of observational unit in its own table. Tidy data is easier to analyze than messy data because it aligns with the input requirements of PivotTables, SQL queries, and Power BI data models. The three textbook datasets follow tidy principles because they are stored as relational database tables, but real-world data often requires restructuring before it reaches this standard.

The practical skills introduced in the two guided tutorials, inspecting the Northwind dataset for missing values, date inconsistencies, and outliers, and examining the ERPNext general ledger for completeness and consistency, represent the first steps in a data quality assessment process that you will apply throughout this textbook and throughout your career.

Key Terms

Accuracy. A data quality dimension indicating that recorded values correspond to the true values they are intended to represent. In accounting, accuracy problems include data entry errors, misclassifications, and incorrect calculations.

Completeness. A data quality dimension indicating that all expected records and field values are present in the dataset. Missing transactions, blank fields, and omitted time periods are examples of completeness problems.

Consistency. A data quality dimension indicating that the same fact is represented in the same way wherever it appears in a dataset or across related datasets. Inconsistent customer names, account codes, or date formats are common consistency problems.

Data quality. The degree to which a dataset is fit for its intended use, assessed across multiple dimensions including accuracy, completeness, consistency, and timeliness.

Duplicate record. A record that appears more than once in a dataset, representing the same underlying transaction, entity, or event. Duplicate payments and duplicate customer entries are common examples in accounting data.

External data. Data originating from outside the organization, such as bank statements, market price feeds, credit ratings, and industry benchmarks. External data introduces additional quality concerns because the accountant has less control over how it was collected and formatted.

General ledger. The central repository of an organization's financial transactions, containing debit and credit entries organized by account. In the ERPNext dataset, the general ledger is stored in the GLEntry table.

Missing value. A field within a record that is blank, null, or contains a placeholder rather than a substantive value. Missing values represent a completeness problem and can distort calculations if not handled properly.

Outlier. A data value that falls far outside the expected range for its field. Outliers may represent legitimate unusual transactions or data entry errors and require investigation to determine the appropriate treatment.

Qualitative data. Data consisting of categories, labels, or descriptions rather than numerical measurements. Account names, transaction descriptions, and vendor classifications are examples of qualitative data in accounting.

Quantitative data. Data consisting of numerical values that represent measurements or counts and that can be subjected to arithmetic operations. Transaction amounts, unit quantities, and financial ratios are examples of quantitative data in accounting.

Semi-structured data. Data that has some organizational elements but does not fit neatly into rows and columns. XBRL filings and PDF bank statements are examples relevant to accounting.

Structured data. Data organized into a predefined format of rows and columns, where each column has a defined data type. All three textbook datasets contain structured data.

Sub-ledger. A detailed record for a specific category of accounts that feeds summary totals to the general ledger. The accounts receivable sub-ledger and accounts payable sub-ledger are common examples.

Tidy data. A dataset organized so that each variable occupies its own column, each observation occupies its own row, and each type of observational unit forms its own table. Tidy data aligns with the input requirements of analytical tools such as PivotTables, SQL, and Power BI.

Timeliness. A data quality dimension indicating that the data reflects the current state of the business as of the relevant reporting date. Data that is not current enough for the intended analysis has a timeliness problem.

Unstructured data. Data that lacks a predefined tabular format, such as the text of contracts, email correspondence, or scanned invoice images.

Multiple Choice Questions

1. A transaction amount of $15,000 is recorded in the general ledger as $1,500 due to a data entry error. This is an example of a problem with which data quality dimension?

A. Completeness B. Timeliness C. Accuracy D. Consistency

2. An auditor discovers that 47 orders in a sales database have no value in the ShippedDate field. This situation represents a problem with which data quality dimension?

A. Accuracy B. Completeness C. Consistency D. Timeliness

3. A company's accounts receivable sub-ledger records a customer as "Johnson & Johnson Inc." while the payment records list the same customer as "Johnson and Johnson." An analyst attempting to match payments to receivables will encounter errors because of this difference. This is an example of a problem with which data quality dimension?

A. Accuracy B. Completeness C. Consistency D. Timeliness

4. Which of the following is an example of qualitative data in an accounting context?

A. The total amount of a sales invoice B. The number of units of inventory on hand C. The name of the cost center to which an expense is assigned D. The gross profit margin expressed as a percentage

5. According to the concept of tidy data, a budget spreadsheet with months as column headers (January, February, March) and departments as rows is considered "messy" because it violates which tidy data principle?

A. Each variable should occupy its own column B. Each observation should occupy its own row C. Each type of observational unit should form its own table D. Both A and B

6. An auditor is performing analytical procedures using general ledger data extracted from the client's ERP system on March 15 for a fiscal year ending December 31. The data includes all adjusting entries through March 10 but not the final three adjusting entries posted on March 12. This situation represents a concern about which data quality dimension?

A. Accuracy B. Completeness C. Consistency D. Timeliness

7. Which of the following best describes the relationship between structured and unstructured data in accounting practice?

A. Accountants work exclusively with structured data and never encounter unstructured data. B. Most accounting analytical work uses structured data, but the proportion of unstructured data in accounting is growing. C. Unstructured data has replaced structured data as the primary input for accounting analytics. D. Structured and unstructured data require the same analytical tools and techniques.

8. A management accountant discovers that the same product appears in the inventory database under two different product codes because it was entered separately by two warehouse locations. This is an example of which data quality problem?

A. Missing value B. Outlier C. Duplicate record D. Timeliness problem

9. Which of the following sources of accounting data provides the most complete picture of an organization's financial transactions?

A. The accounts receivable sub-ledger B. The general ledger C. An external bank statement D. A product inventory listing

10. The ERPNext dataset is described as the most "accounting-rich" of the three textbook datasets. Which feature primarily distinguishes it from Northwind and AdventureWorks?

A. It contains sales transaction data. B. It includes a chart of accounts, general ledger entries, journal entries, and a complete accounting module. C. It contains more tables than the other two datasets combined. D. It records data in an unstructured format.

11. A dataset is organized so that each variable occupies its own column and each observation occupies its own row, but several records contain data entry errors in the amount field. This dataset is best described as:

A. Tidy and clean B. Tidy but not clean C. Clean but not tidy D. Neither tidy nor clean

12. A financial analyst creates a revenue summary using a SQL query that groups sales by customer name. The query returns 150 distinct customers, but the company's customer master file contains only 142 customers. The most likely explanation for the discrepancy is:

A. The SQL query contains a syntax error. B. Some customers made purchases but were never entered into the customer master file. C. Inconsistent formatting of customer names caused the same customer to appear under multiple name variations. D. The customer master file is more current than the sales data.

13. An enterprise resource planning system is distinct from a standalone accounting package primarily because an ERP system:

A. Uses a relational database to store data B. Integrates data from multiple functional areas such as sales, production, purchasing, and human resources into a single database C. Produces financial statements automatically D. Eliminates all data quality problems through built-in validation rules

14. Which of the following best explains why data quality assessment should be performed before any analysis begins?

A. Software tools cannot process data that contains quality problems. B. Data quality problems can cause analytical results to be unreliable or misleading, and identifying problems in advance allows the accountant to address them. C. Professional standards prohibit accountants from using any data that contains missing values. D. Data quality assessment is only required for auditing engagements and not for managerial accounting or financial reporting.

15. An auditor finds that a general ledger entry references account code "5200" but this code does not appear anywhere in the company's chart of accounts. This finding represents a problem with:

A. Accuracy only B. Completeness only C. Consistency between the general ledger and the chart of accounts D. Timeliness of the chart of accounts

---

Applied Exercises

Financial Accounting Exercises

Exercise 2.1 (Financial Accounting): Assessing Revenue Data Quality in Northwind

Dataset: Northwind (Orders, OrderDetails, Products, Customers)

Scenario. You are a financial reporting analyst preparing to analyze Northwind Traders' revenue for the most recent fiscal year. Before building any reports, you need to verify that the underlying sales data is reliable enough to support accurate revenue figures.

Requirements. (1) Open the Northwind dataset in Excel and examine the Orders, OrderDetails, Products, and Customers tables. For each table, identify and document any data quality problems you observe, classifying each problem by quality dimension (accuracy, completeness, consistency, or timeliness). (2) Assess whether the OrderDetails table contains the information needed to calculate revenue for each order. Identify the columns involved and note any values that appear problematic (such as zero or negative quantities, missing prices, or discount values outside the expected zero-to-one range). (3) Examine whether every CustomerID in the Orders table corresponds to a customer in the Customers table. Use COUNTIFS or VLOOKUP to test this relationship. Document any orphaned records where an order references a customer that does not exist. (4) Prepare a brief written data quality assessment memorandum (one page) that summarizes your findings and recommends whether the data is suitable for revenue reporting or whether specific problems must be resolved first.

Deliverable. A one-page data quality assessment memorandum with specific findings organized by quality dimension.

Exercise 2.2 (Financial Accounting): Evaluating General Ledger Completeness in ERPNext

Dataset: ERPNext (GLEntry, Account)

Scenario. You are a financial reporting analyst who has been asked to verify that the ERPNext general ledger is complete before the team begins preparing the year-end financial statements. Your supervisor is concerned that some transactions may not have been posted.

Requirements. (1) Open the ERPNext dataset and examine the GLEntry table. Calculate the total debits and total credits across all entries. Determine whether the ledger is in balance. (2) Identify the date range covered by the general ledger entries. Check whether entries are present for every month within the fiscal year. Note any months that appear to have an unusually low number of entries compared to other months, which could indicate missing transactions. (3) Examine the voucher type column to understand what types of transactions are recorded. Determine whether the ledger includes entries from all expected source types (such as journal entries, sales invoices, purchase invoices, and payment entries). If any expected source type is absent, document this as a potential completeness concern. (4) Write a brief assessment (one page) of the general ledger's completeness, suitable for review by the financial reporting manager.

Deliverable. A one-page completeness assessment memorandum with quantitative evidence supporting each finding.

Managerial Accounting Exercises

Exercise 2.3 (Managerial Accounting): Data Quality Assessment for Cost Analysis

Dataset: AdventureWorks (Product, WorkOrder)

Scenario. You are a cost analyst who has been asked to perform a product cost analysis using AdventureWorks manufacturing data. Before starting the analysis, you need to assess whether the data is clean enough to produce reliable cost figures.

Requirements. (1) Open the AdventureWorks dataset in Excel and examine the Product table. Identify any products where the StandardCost field is zero, blank, or appears unreasonable. Assess whether these entries represent data quality problems or legitimate situations (such as products that have been discontinued and are no longer costed). (2) Compare the StandardCost and ListPrice columns. Identify any products where the ListPrice is less than the StandardCost, which would imply a negative margin. Determine whether these represent accuracy problems or deliberate pricing decisions. (3) Examine the WorkOrder table if available. Check for missing values in key fields such as quantity ordered, quantity produced, and dates. (4) Write a brief assessment (one half to one page) of the data's suitability for product cost analysis, identifying any problems that would need to be resolved before proceeding.

Deliverable. A written data quality assessment focused on cost analysis readiness, with specific product-level findings.

Exercise 2.4 (Managerial Accounting): Identifying Tidy Data Problems in Budget Reports

Dataset: ERPNext (Budget, CostCenter, or equivalent tables)

Scenario. You are a management accountant who needs to compare budgeted amounts to actual spending by cost center. The budget data is available in the ERPNext system, but you suspect it may not be in a format suitable for analysis.

Requirements. (1) Open the ERPNext dataset and locate any budget-related tables. Examine the structure. Determine whether the budget data follows tidy data principles (each variable in its own column, each observation in its own row). If not, describe specifically which tidy data principle is violated and how the data would need to be restructured. (2) Examine the cost center references in the budget data and in the GLEntry table. Assess whether the cost center names or codes are consistent between the two sources. Identify any inconsistencies that would prevent a straightforward comparison. (3) Document your findings in a brief memorandum (one half to one page) that recommends specific data preparation steps needed before a budget-versus-actual analysis can be performed.

Deliverable. A written memorandum identifying structural and quality issues in the budget data and recommending preparation steps.

Auditing Exercises

Exercise 2.5 (Auditing): Data Quality Evaluation for Audit Planning

Dataset: Northwind (Orders, OrderDetails, Customers, Suppliers) and ERPNext (GLEntry, Account, PaymentEntry)

Scenario. You are a first-year auditor assigned to evaluate the data that will be used for analytics procedures during the upcoming audit. Your senior has asked you to perform a preliminary data quality assessment of both the Northwind and ERPNext datasets to determine which data sources are reliable enough for audit testing.

Requirements. (1) For the Northwind dataset, perform the following data quality checks on the Orders and OrderDetails tables: test for missing values in critical fields (OrderDate, CustomerID, UnitPrice, Quantity), test for records where ShippedDate precedes OrderDate, and check for order line items with quantities of zero or negative values. Document the number of records affected by each problem. (2) For the ERPNext dataset, perform the following checks on the GLEntry and PaymentEntry tables: verify that total debits equal total credits in the general ledger, check for payment entries with missing dates or missing vendor references, and test for duplicate entries by looking for records that match on date, amount, and account. (3) For each problem identified, classify it by quality dimension and assess its potential impact on audit procedures. A missing date, for example, would prevent the auditor from testing

the transaction in the correct period. (4) Prepare a one-page audit data quality assessment memorandum that your senior could use to decide which datasets and tables are suitable for audit analytics.

Deliverable. A one-page audit data quality assessment memorandum organized by dataset and quality dimension.

Exercise 2.6 (Auditing): Testing for Duplicate Records

Dataset: ERPNext (PaymentEntry or equivalent payment table)

Scenario. Your audit team has identified duplicate payments as a risk area for the current engagement. Before performing a full duplicate payment analysis (which you will learn in Chapter 12 using SQL), you have been asked to perform a preliminary manual check using Excel.

Requirements. (1) Open the ERPNext dataset and locate the payment-related table(s). Identify the fields that would be relevant for detecting duplicate payments, such as vendor or payee, amount, and date. (2) Sort the data by payee and then by amount to bring potential duplicates into adjacent rows. Scan for records where the same payee received the same amount on the same date or within a narrow date range. Document any potential duplicates you identify. (3) Assess the reliability of your manual approach. Consider what limitations exist when searching for duplicates by visual inspection versus using automated techniques. Write two to three sentences explaining why a SQL-based approach (which you will learn later in the book) would be more effective for this task. (4) Prepare a brief memorandum (one half page) documenting the potential duplicates identified and your assessment of the method's limitations.

Deliverable. A written memorandum documenting potential duplicate payments and the limitations of manual detection methods.

Further Reading

Wang, R. Y., and Strong, D. M. (1996). Beyond accuracy: What data quality means to data consumers. *Journal of Management Information Systems*, 12(4), 5-33. This paper established the foundational framework for understanding data quality as a multidimensional concept defined by the needs of data users. The four quality dimensions discussed in this chapter (accuracy, completeness, consistency, and timeliness) draw on this framework, and the paper remains essential reading for anyone who works with data quality assessment in any professional context.

Wickham, H. (2014). Tidy data. *Journal of Statistical Software*, 59(10), 1-23. This paper formalized the principles of tidy data that are discussed in this chapter. Although the examples use the R programming language, the structural principles apply to any data analysis environment, including Excel, SQL, and Power BI. The paper's clear explanation of why each variable should occupy its own column and each observation its own row is directly relevant to the data preparation challenges students will encounter throughout this textbook.

Grabski, S. V., Leech, S. A., and Schmidt, P. J. (2011). A review of ERP research: A future agenda for accounting information systems. *Journal of Information Systems*, 25(1), 37-78. This review paper examines the research literature on enterprise resource planning systems and their implications for accounting. It provides useful context for understanding the role of ERP systems as data sources in accounting analytics and discusses the data quality challenges that arise within ERP environments.

Redman, T. C. (2001). *Data quality: The field guide.* Digital Press. This practitioner-oriented book provides a comprehensive treatment of data quality concepts, assessment methods, and improvement strategies. Although it addresses data quality broadly rather than in an accounting-specific context, the practical techniques it describes for identifying and resolving data quality problems are directly applicable to the exercises in this chapter and throughout the textbook.

AICPA (American Institute of Certified Public Accountants). (2017). *Guide to audit data analytics.* AICPA. This professional guidance document describes how auditors should use data analytics in audit engagements, including specific recommendations for evaluating the reliability and completeness of data before performing analytical procedures. The guide's discussion of data quality evaluation aligns with the quality dimensions covered in this chapter and supports the audit-oriented exercises throughout the textbook.

Fisher, I. E., Garnsey, M. R., and Hughes, M. E. (2016). Natural language processing in accounting, auditing and finance: A synthesis of the literature with a roadmap for future research. *Intelligent Systems in Accounting, Finance and Management*, 23(3), 157-214. This paper reviews the emerging application of natural language processing to unstructured accounting data such as contracts, disclosures, and regulatory filings. It provides useful context for the brief discussion of unstructured data in this chapter and points toward the emerging technologies covered in Chapter 20.

Bovee, M., Srivastava, R. P., and Mak, B. (2003). A conceptual framework and belief-function approach to assessing overall information quality. *International Journal of Intelligent Systems*, 18(1), 51-74. This paper presents a formal framework for assessing information quality that extends the foundational work of Wang and Strong. The paper's approach to evaluating quality dimensions is relevant to the data quality assessment exercises in this chapter, particularly for students interested in the theoretical underpinnings of data quality measurement.

Dai, J., and Vasarhelyi, M. A. (2017). Toward blockchain-based accounting and assurance. *Journal of Information Systems*, 31(3), 5-21. While this paper focuses primarily on blockchain technology, it includes a useful discussion of how data integrity and quality assurance are

handled in current accounting information systems versus emerging distributed ledger approaches. It provides forward-looking context that connects the data quality concepts in this chapter to the emerging technologies discussed in Chapter 20.

The Accounting Data Environment

Conceptual Chapter Northwind AdventureWorks ERPNext

Learning Objectives

After completing this chapter, you will be able to:

1. Explain the structure of a relational database and describe how tables, rows, columns, and data types organize accounting information.
2. Distinguish between primary keys and foreign keys and explain how they establish relationships between tables in an accounting database.
3. Read and interpret an Entity-Relationship diagram using crow's foot notation to identify tables, relationships, and cardinality.
4. Map familiar accounting structures such as the chart of accounts, general ledger, and sub-ledgers to their representations as related tables in a database.
5. Compare the database designs of Northwind, AdventureWorks, and ERPNext and evaluate how each design reflects the complexity of the underlying business.

Opening Scenario

You are a junior auditor assigned to a new client that uses an enterprise resource planning system to manage its accounting records. During the planning meeting, the senior auditor mentions that the team will extract data directly from the client's database to perform analytical procedures. She pulls up a diagram showing dozens of interconnected boxes, each representing a table in the database, and explains that understanding this diagram is essential before writing any queries. You recognize some of the table names from your accounting courses, such as "GeneralLedger," "SalesInvoice," and "ChartOfAccounts," but the lines connecting the tables and the symbols at their endpoints are unfamiliar. You realize that to work effectively with

this data, you need to understand not just accounting concepts but also how those concepts are represented in a database. This chapter provides that understanding. It introduces the relational database model, explains how accounting data is organized within it, and walks you through the database designs of the three datasets you will use throughout this book.

From Accounting Records to Database Tables

In your introductory accounting courses, you learned to work with familiar structures such as the chart of accounts, the general journal, the general ledger, and various sub-ledgers. You recorded transactions as debits and credits, posted them to T-accounts, prepared trial balances, and produced financial statements. These structures have served the accounting profession for centuries, and they remain the conceptual foundation of financial record-keeping.

What has changed is how organizations store and access these structures. In a manual system, the general journal is a physical book. In a spreadsheet-based system, it is a worksheet. In a modern organization, it is a table in a relational database. The chart of accounts is another table. The accounts receivable sub-ledger is yet another table. These tables are connected to one another through defined relationships that mirror the connections you already understand from accounting. The CustomerID on an invoice connects that invoice to a specific customer record. The AccountCode on a journal entry connects that entry to a specific account in the chart of accounts. These connections exist in manual systems as cross-references and posting notations. In a database, they are formalized as structural features that the system enforces and that analytical tools can use.

Understanding how accounting data is organized in a database is a prerequisite for everything that follows in this textbook. When you write SQL queries in Part III, you will need to know which tables contain the data you need and how those tables connect to one another. When you build data models in Power BI in Part IV, you will define relationships between tables that determine how your calculations and visualizations behave. Even in the Excel chapters of Part II, understanding the source structure helps you anticipate what the data will look like when you import it and what preparation steps will be necessary. The concepts in this chapter are not abstract. They are the practical foundation for every analytical task you will perform.

> **IN PRACTICE** Accounting firms increasingly expect new hires to understand relational database concepts. Auditors who can read a database diagram and identify the tables relevant to a particular audit procedure work more efficiently during fieldwork. A 2018 survey of public accounting professionals found that understanding data structures and database concepts ranked among the most valuable technical skills for entry-level staff, second only to spreadsheet proficiency (Sledgianowski, Gomaa, and Tan, 2017).

The Relational Database Model

A relational database stores data in a collection of tables that are related to one another through shared values. The relational model was proposed by Edgar F. Codd in 1970 and has become the dominant approach to data storage in business information systems (Codd, 1970). Virtually every enterprise resource planning system, accounting software package, and transaction processing system in use today stores its data in a relational database. Understanding this model is therefore essential for any accountant who needs to access data directly rather than relying on pre-built reports.

Tables, Rows, and Columns

The fundamental unit of a relational database is the table. A table stores information about one type of entity or event. In an accounting context, a table might store information about customers, products, sales orders, journal entries, or accounts. Each table has a name that identifies what it contains. The Northwind database, for example, has a table called Customers that stores information about every customer the company does business with, and a table called Orders that records every sales order the company has received.

A table is organized into rows and columns. Each row represents a single instance of the entity or event that the table describes. In the Customers table, each row represents one customer. In the Orders table, each row represents one order. Each column represents a single attribute of that entity or event. In the Customers table, columns include CompanyName, ContactName, City, Country, and Phone. In the Orders table, columns include OrderID, CustomerID, OrderDate, ShippedDate, and Freight.

[Figure 3.1: Anatomy of a Database Table. An annotated diagram showing the Northwind Customers table with labels identifying the table name, column headers, rows, and individual cell values. The annotation explains that each row is one customer and each column is one attribute of that customer.]

If this structure looks familiar, it should. A database table is structurally identical to a well-organized spreadsheet or to the tidy data format described in Chapter 2. Each variable occupies its own column, and each observation occupies its own row. The difference is that a database enforces this structure. When a table is created, the database requires a definition that specifies the name, the data type, and the constraints for every column. This definition prevents the kind of structural drift that occurs in spreadsheets, where a user might insert a comment in a data column or merge cells in ways that break the tabular format.

Data Types

Every column in a database table has a defined data type that specifies what kind of values the column can hold. The most common data types in accounting databases include integers

(whole numbers used for identifiers and counts), decimal or numeric values (numbers with fractional parts used for monetary amounts, prices, and quantities), text or character strings (used for names, descriptions, and codes), dates (used for transaction dates, due dates, and posting dates), and Boolean values (true or false values used for status flags such as whether an invoice has been paid).

Data types matter for accounting analytics because they determine what operations are possible. You can calculate the sum of a column defined as a decimal but not the sum of a column defined as text. You can sort dates chronologically but only if the database recognizes the values as dates rather than text strings. Many of the data quality problems discussed in Chapter 2, such as a transaction amount stored as text because it includes a currency symbol, are essentially data type problems. When you work with the textbook datasets in later chapters, the data types are already defined correctly because the datasets were created as proper databases. In professional practice, however, you will encounter situations where data types are misconfigured, and understanding what the correct types should be will help you identify and resolve those problems.

[Table 3.1: Common Data Types in Accounting Databases. A table with three columns (Data Type, Description, Accounting Example) and five rows covering INTEGER, DECIMAL/NUMERIC, TEXT/VARCHAR, DATE, and BOOLEAN. Each row includes an example from the textbook datasets.]

> **WATCH OUT** A column that contains numbers is not necessarily a numeric data type. Account codes such as "4100" and "5200" are often stored as text rather than as numbers because they are identifiers, not quantities. You would never add two account codes together or calculate the average account code. Storing them as text prevents accidental arithmetic and preserves leading zeros that a numeric type would drop. When you encounter a column of numbers in a database, consider whether the values represent quantities to be calculated or codes to be matched.

Keys and Relationships

The power of a relational database comes not from individual tables but from the connections between them. In an accounting system, transactions do not exist in isolation. An order is placed by a customer, fulfilled by an employee, and contains products purchased from suppliers. A journal entry debits one account and credits another, and both accounts appear in the chart of accounts. These connections are represented in a database through keys.

Primary Keys

A primary key is a column, or a combination of columns, that uniquely identifies each row in a table. No two rows in the same table can have the same primary key value, and the primary

key cannot be blank. In the Northwind Customers table, the primary key is CustomerID. Every customer has a unique CustomerID, and no two customers share the same value. In the Northwind Orders table, the primary key is OrderID. Every order has a unique OrderID that distinguishes it from every other order in the table.

Primary keys serve the same purpose in a database that unique identifiers serve in accounting. An invoice number uniquely identifies an invoice. A check number uniquely identifies a payment. An employee number uniquely identifies an employee. When these documents and records are stored in a database, the identifying number becomes the primary key of the corresponding table. The database enforces uniqueness automatically, meaning it will reject any attempt to insert a second row with an existing primary key value. This enforcement is a form of data integrity control that prevents the duplication problems discussed in Chapter 2.

Foreign Keys

A foreign key is a column in one table that references the primary key of another table. Foreign keys create the connections between tables that make a relational database relational. In the Northwind Orders table, the column CustomerID is a foreign key that references the CustomerID primary key in the Customers table. This relationship means that every order is linked to a specific customer. If you know the CustomerID on an order, you can look up that customer's name, address, and contact information in the Customers table.

Foreign keys mirror the cross-references that exist in accounting records. When you post a journal entry to the general ledger, the account number on the entry refers to a specific account in the chart of accounts. That account number is functioning as a foreign key. When a sales invoice references a customer number, that customer number links the invoice to the customer master file. The database formalizes these references as structural constraints. A well-designed database will reject an order that references a CustomerID that does not exist in the Customers table, just as a well-designed accounting system will reject a journal entry that references an account code that does not appear in the chart of accounts.

[Figure 3.2: Primary Keys and Foreign Keys Illustrated. A diagram showing the Northwind Customers table and the Orders table side by side. The CustomerID column in Customers is labeled "PK" (primary key). The CustomerID column in Orders is labeled "FK" (foreign key). An arrow connects the two columns, showing that each FK value in Orders points to a PK value in Customers. Sample data rows illustrate the connection.]

The relationship between a primary key and a foreign key establishes what database designers call referential integrity. This means that every foreign key value must correspond to an existing primary key value in the referenced table. If a customer is deleted from the Customers table while orders referencing that customer still exist in the Orders table, the database has a referential integrity violation. The orders now reference a customer that does not exist, and any analysis that joins the two tables will produce incomplete results. Referential

integrity constraints prevent this by blocking deletions or updates that would create orphaned references. In Chapter 2, the exercise that checked whether every CustomerID in the Orders table existed in the Customers table was essentially a manual referential integrity test.

> **CONNECTING THE DOTS** In Chapter 10, you will write SQL JOIN statements that use exactly these key relationships to combine data from multiple tables. A JOIN on Orders.CustomerID = Customers.CustomerID retrieves the customer name for each order. Understanding keys now will make JOIN syntax intuitive rather than mysterious when you encounter it in Part III.

Relationships and Cardinality

The relationship between two tables has a direction and a cardinality, meaning it specifies how many rows in one table can be associated with a single row in the other table. Three types of cardinality are common in accounting databases.

A one-to-many relationship means that one row in the first table can be associated with many rows in the second table, but each row in the second table is associated with only one row in the first table. This is the most common relationship type in accounting data. One customer can place many orders, but each order belongs to one customer. One account in the chart of accounts can have many general ledger entries, but each entry is posted to one account. One product category can contain many products, but each product belongs to one category. In the Northwind database, the relationship between Customers and Orders is one-to-many. Customer "ALFKI" has placed multiple orders, and each of those orders has "ALFKI" as its CustomerID.

A one-to-one relationship means that each row in the first table is associated with exactly one row in the second table, and vice versa. One-to-one relationships are less common but do occur. In AdventureWorks, each employee row in the Employee table has a corresponding row in the Person table that stores personal details such as name and contact information. The two tables are linked by a shared identifier, and each employee has exactly one person record.

A many-to-many relationship means that each row in the first table can be associated with many rows in the second table, and each row in the second table can also be associated with many rows in the first table. In accounting, consider the relationship between orders and products. One order can contain many products, and one product can appear on many orders. Relational databases cannot directly represent a many-to-many relationship. Instead, they use a junction table (also called a bridge table or associative table) that sits between the two tables and converts the many-to-many relationship into two one-to-many relationships. In Northwind, the OrderDetails table serves this purpose. It sits between Orders and Products, with each row representing one product line on one order. The OrderDetails table has a foreign key to Orders (OrderID) and a foreign key to Products (ProductID).

[Figure 3.3: The Three Types of Cardinality. A three-panel diagram showing one-to-one, one-to-many, and many-to-many relationships. Each panel uses simple labeled boxes and connecting lines with crow's foot symbols. The one-to-many panel uses the Northwind Customers-to-Orders example. The many-to-many panel shows the Orders-OrderDetails-Products pattern with the junction table clearly labeled.]

Understanding cardinality is important for analytics because it affects the results you get when you combine tables. When you join a one-to-many relationship in SQL or in a Power BI data model, the many side of the relationship can multiply rows in your results. If you join the Customers table to the Orders table and a customer has 15 orders, that customer's information will appear 15 times in the output, once for each order. This is correct and expected behavior, but if you are not aware of the cardinality, you might accidentally count the customer 15 times when calculating the number of customers. Many analytical errors in accounting result from misunderstanding the cardinality of a relationship, and the ER diagrams described in the next section help you avoid those errors by making cardinality visible.

> **WATCH OUT** The most common mistake students make when working with related tables is double-counting. If you join a customer table to an orders table to an order details table, a single customer with 5 orders and 20 line items across those orders will appear 20 times in the result set. Summing revenue at the line-item level is correct. Counting the customer at the line-item level overstates the customer count by a factor of 20. Always consider the cardinality of each join before performing aggregations.

Entity-Relationship Diagrams

An Entity-Relationship diagram, commonly called an ER diagram, is a visual representation of the tables in a database and the relationships between them. ER diagrams are the maps that database designers, analysts, and auditors use to understand the structure of a data environment. Reading an ER diagram is a skill you will use repeatedly in this textbook and in professional practice.

How to Read an ER Diagram

In the notation used throughout this textbook, each table appears as a rectangle. The rectangle has a header bar containing the table name and a body listing the column names. Primary key columns are marked with "PK" and foreign key columns are marked with "FK." Relationships between tables are shown as lines connecting the foreign key in one table to the primary key it references in another table.

The endpoints of each relationship line use crow's foot notation to indicate cardinality. A single line ending in a perpendicular bar indicates "one." A line ending in a three-pronged

fork (resembling a crow's foot) indicates "many." A relationship line with a bar on one end and a crow's foot on the other represents a one-to-many relationship. The bar end points to the "one" table, and the crow's foot end points to the "many" table.

[Figure 3.4: Reading Crow's Foot Notation. A reference diagram showing the four common endpoint symbols used in crow's foot ER diagrams: "one and only one" (bar with bar), "one or many" (bar with crow's foot), "zero or one" (circle with bar), and "zero or many" (circle with crow's foot). Each symbol includes a brief label and an accounting example of when it applies.]

Consider a concrete example. The Northwind ER diagram shows a line between the Customers table and the Orders table. The Customers end of the line has a single bar, meaning "one." The Orders end has a crow's foot, meaning "many." Reading the relationship aloud, you would say "one customer can have many orders." Looking at the columns, you can see that CustomerID is the primary key of the Customers table and a foreign key in the Orders table. This tells you that the connection between the two tables is made through the CustomerID column, and that to retrieve a customer's name alongside their orders, you would match these two columns.

ER diagrams serve the same purpose for a database that a chart of accounts serves for an accounting system. The chart of accounts tells you how the organization classifies its financial transactions. The ER diagram tells you how the database organizes its data. Just as you would consult the chart of accounts before preparing a journal entry, you should consult the ER diagram before writing a query or building a data model.

> **IN PRACTICE** Auditors routinely request ER diagrams or data dictionaries from clients during the planning phase of an engagement. Understanding the client's database structure helps the audit team identify which tables contain the data needed for analytical procedures, which relationships need to be traversed to connect transactions to their supporting details, and where potential data integrity risks exist. The AICPA's Guide to Audit Data Analytics recommends that auditors document their understanding of the client's data structure before performing data extraction (AICPA, 2017).

The Accounting Data Model

The relational concepts described so far apply to databases in any domain. What makes an accounting database distinctive is the specific set of entities it stores and the relationships between them. This section shows how the familiar structures of accounting map to tables and relationships in a database. Seeing these connections will help you move between accounting concepts and database structures without difficulty.

The Chart of Accounts as a Table

The chart of accounts is the organizational backbone of any accounting system. It defines every account the organization uses to classify its financial transactions, typically organized into five major categories: assets, liabilities, equity, revenues, and expenses. In a database, the chart of accounts is stored as a table where each row represents one account. Columns typically include an account code or number, an account name, an account type (asset, liability, equity, revenue, or expense), and a parent account reference that defines the hierarchical structure.

In the ERPNext dataset, the chart of accounts is stored in the Account table. Each row contains an account name, an account type classification, and a parent account reference that places the account within the hierarchy. The parent account reference is a foreign key that points to another row in the same table, creating what database designers call a self-referencing relationship or recursive relationship. The "Cash and Cash Equivalents" account, for example, might have "Current Assets" as its parent account, and "Current Assets" might have "Assets" as its parent. This hierarchical structure allows the database to represent the multi-level account groupings that appear on financial statements.

[Figure 3.5: The Chart of Accounts as a Database Table. An annotated diagram showing a portion of the ERPNext Account table with columns for AccountName, AccountType, and ParentAccount. Arrows illustrate the self-referencing relationship where child accounts point to their parent accounts. The annotation explains how this structure represents the account hierarchy.]

The General Ledger as a Related Table

The general ledger records every financial transaction as a series of debit and credit entries. In a database, the general ledger is typically stored as a table where each row represents one side of a journal entry: a single debit or a single credit. Columns include the posting date, the account reference, the debit amount, the credit amount, a voucher type identifying the source document, and a voucher number linking the entry to the originating transaction.

The relationship between the general ledger table and the chart of accounts table is one-to-many. One account can appear in many general ledger entries, but each general ledger entry is posted to one account. The account reference in the general ledger table is a foreign key that points to the primary key of the chart of accounts table. This relationship is the database equivalent of posting a journal entry to a T-account. When you sum the debits and credits for a given account across all general ledger entries, you calculate that account's balance, which is precisely what a trial balance reports.

In the ERPNext dataset, the GLEntry table stores the general ledger. Each row has a field that references the Account table, establishing the connection between individual entries and the chart of accounts. This relationship is what makes it possible to produce a trial balance from the database: you group the GLEntry rows by account and sum the debit and credit columns for each group.

Sub-Ledgers and Their Relationship to the General Ledger

Sub-ledgers provide transaction-level detail for specific account categories. The accounts receivable sub-ledger, for example, records individual customer invoices and payments. The accounts payable sub-ledger records individual vendor invoices and payments. In a manual system, the sub-ledger detail rolls up to a control account in the general ledger, and the control account balance should equal the sum of all sub-ledger balances.

In a database, sub-ledgers are represented as tables that connect to the general ledger through shared identifiers. In the ERPNext dataset, the SalesInvoice table functions as part of the accounts receivable sub-ledger. Each sales invoice generates general ledger entries in the GLEntry table, and the voucher number recorded in both tables links the summary posting to its source document. This link allows an analyst or auditor to start with a general ledger balance, identify the journal entries that compose it, and trace each entry back to the original invoice.

The Northwind dataset demonstrates a simpler version of this pattern. The Orders and OrderDetails tables function as a sales sub-ledger, recording the details of each sales transaction. Northwind does not have a general ledger table, so the sub-ledger is the most detailed record available. In practice, a company like Northwind would have a general ledger in its accounting system, and the order data in Northwind would feed summary entries into that ledger. Understanding that the three datasets represent different levels of accounting completeness, as discussed in Chapter 1, helps you select the right dataset for a given analytical task.

[Figure 3.6: The Relationship Between the General Ledger and Sub-Ledgers. A diagram showing the ERPNext Account table at the top, connected to the GLEntry table in the middle, which is connected to the SalesInvoice and PurchaseInvoice tables at the bottom. Arrows and cardinality symbols show that one account has many GL entries, and each GL entry can be traced to a source document in a sub-ledger table. Annotations label the general ledger layer and the sub-ledger layer.]

> **CONNECTING THE DOTS** In Chapter 12, you will write SQL queries that trace transactions from the ERPNext general ledger back to their source documents in the sub-ledger tables. This trace is one of the most important procedures in audit analytics because it tests whether the amounts reported in the financial statements are supported by underlying transactions. The database structure you are learning here makes that trace possible.

The Northwind Database Design

Now that you understand the components of a relational database, it is time to examine the designs of the three textbook datasets in detail. Each database reflects the complexity of the business it represents. Studying all three together will show you how the same relational principles apply at different scales and in different business contexts.

Northwind Traders is a small wholesale food distribution company, and its database reflects that simplicity. The database contains eight core tables organized around two main business activities: selling products to customers and purchasing products from suppliers.

The sales side of the database centers on three tables. The Orders table records each sales order with columns for the order date, required date, shipped date, and freight amount. The OrderDetails table records the individual line items on each order, with columns for the product, quantity, unit price, and discount. The Customers table stores information about each customer, including company name, contact name, and address. The relationship between Customers and Orders is one-to-many (one customer, many orders), and the relationship between Orders and OrderDetails is also one-to-many (one order, many line items). As discussed earlier, OrderDetails also connects to the Products table through the ProductID foreign key, resolving the many-to-many relationship between orders and products.

The product side of the database includes the Products table, which records each product's name, unit price, units in stock, and units on order, and the Categories table, which classifies products into groups such as "Beverages," "Condiments," and "Seafood." Each product belongs to one category, and one category contains many products.

The supply side includes the Suppliers table, which stores information about the companies that provide products to Northwind. Each product has one supplier, recorded through the SupplierID foreign key in the Products table.

Two additional tables support operations. The Employees table records the sales representatives who handle orders. The Shippers table records the shipping companies that deliver orders. Both connect to the Orders table through foreign keys.

[Figure 3.7: Full Entity-Relationship Diagram for Northwind Traders. A complete ER diagram showing all eight tables (Customers, Orders, OrderDetails, Products, Categories, Suppliers, Employees, Shippers) with their columns, primary keys, foreign keys, and relationships displayed in crow's foot notation. Tables use teal header bars per the visual design specification.]

The Northwind design is clean and compact. With only eight tables and straightforward one-to-many relationships, it is easy to understand and navigate. This simplicity is why Northwind serves as the primary dataset for the foundational chapters of this textbook. When you learn new tools and techniques, working with a database you can hold in your head lets you focus on the technique itself rather than struggling with data complexity.

The AdventureWorks Database Design

Adventure Works Cycles is a mid-size multinational bicycle manufacturer, and its database reflects the complexity of a manufacturing operation. The database contains approximately 70 tables organized across five functional areas, or schemas: Sales, Production, Purchasing, Human Resources, and Person.

A schema is a logical grouping of related tables within a database. Schemas serve the same organizational purpose that departments serve in a company: they group related activities together and make it easier to find what you need. In AdventureWorks, the Sales schema contains tables related to customer orders, territories, and sales representatives. The Production schema contains tables related to products, work orders, bills of materials, and scrap tracking. The Purchasing schema contains tables for vendor management and purchase orders. The Human Resources schema stores employee and department information. The Person schema stores personal details shared across functional areas.

The Sales schema follows a pattern similar to Northwind but with additional complexity. The SalesOrderHeader table records order-level information such as the order date, customer, territory, and total amount due. The SalesOrderDetail table records individual line items. The Customer table stores customer information, and the SalesTerritory table defines the geographic regions where sales occur. These tables are connected through foreign keys in the same way as their Northwind counterparts, but with additional attributes and relationships that support multi-territory, multi-currency operations.

The Production schema introduces tables that have no equivalent in Northwind. The Product table stores product information including standard cost and list price. The WorkOrder table records manufacturing jobs, including the quantity ordered, quantity produced, quantity scrapped, and dates. The BillOfMaterials table defines the components required to manufacture each product. The ScrapReason table classifies why production units were rejected. These tables allow the database to track not just what was sold but how it was made, at what cost, and with what level of waste.

The Purchasing schema includes the PurchaseOrderHeader and PurchaseOrderDetail tables, which record orders placed with vendors, and the Vendor table, which stores vendor information. These tables support exercises in purchasing cycle analysis, vendor evaluation, and procurement testing.

[Figure 3.8: Focused Entity-Relationship Diagram for AdventureWorks Sales and Production Schemas. A diagram showing a selected subset of AdventureWorks tables from the Sales schema (SalesOrderHeader, SalesOrderDetail, Customer, SalesTerritory) and the Production schema (Product, WorkOrder, BillOfMaterials, ProductSubcategory). Tables use primary blue header bars. Relationships use crow's foot notation. The full diagram appears in Appendix B.]

The AdventureWorks design demonstrates how a relational database scales to accommodate a more complex business. The same principles of primary keys, foreign keys, and one-to-many relationships that you saw in Northwind apply here, but they connect a much larger network of tables. Navigating this network requires an ER diagram, which is why the ability to read such diagrams is an essential skill for accounting analytics.

> **IN PRACTICE** Manufacturing companies typically have the most complex database structures among the organizations that accountants serve. Production data introduces tables for bills of materials, work centers, routing steps, scrap

tracking, and quality control, all of which connect to the financial data through cost records. Cost accountants and auditors working in manufacturing frequently need to traverse relationships that span multiple schemas to answer questions such as "What was the actual cost of producing this product, and how does it compare to the standard cost?" The AdventureWorks dataset prepares you for exactly this type of analysis.

The ERPNext Database Design

The ERPNext Demo Company operates within a full enterprise resource planning environment, and its database design reflects the integrated nature of an ERP system. Unlike Northwind, which captures order-level transactions, and AdventureWorks, which adds manufacturing data, ERPNext includes a complete accounting module. This makes ERPNext the most relevant dataset for financial reporting, audit analytics, and any exercise that requires working directly with the general ledger.

The core of the ERPNext accounting module is the relationship between three tables. The Account table stores the chart of accounts with its hierarchical structure. The GLEntry table stores every debit and credit entry that makes up the general ledger. The JournalEntry table stores the header information for manually created journal entries, including the posting date, the entry type, and any descriptive notes. The GLEntry table connects to the Account table through an account reference foreign key, and it connects to source documents such as journal entries, sales invoices, and purchase invoices through a voucher type and voucher number pair.

Beyond the accounting core, ERPNext includes tables for sales invoices (SalesInvoice), purchase invoices (PurchaseInvoice), payment entries (PaymentEntry), cost centers (CostCenter), and budgets (Budget). The SalesInvoice table records revenue transactions and connects to the GLEntry table through the voucher reference, allowing analysts to trace a revenue balance in the general ledger back to the individual invoices that compose it. The CostCenter table defines the organizational units used for management reporting, and general ledger entries are tagged with cost center references that enable departmental analysis.

[Figure 3.9: Focused Entity-Relationship Diagram for ERPNext Accounting Module. A diagram showing the Account table, GLEntry table, JournalEntry table, SalesInvoice table, PurchaseInvoice table, and PaymentEntry table with their key columns and relationships. Tables use amber header bars. The voucher type and voucher number linkage between GLEntry and the source document tables is highlighted with an annotation. The full diagram appears in Appendix B.]

The ERPNext design illustrates a key feature of ERP databases: every transaction, regardless of its origin, ultimately flows into the general ledger. A sales invoice creates GL entries that debit accounts receivable and credit revenue. A purchase invoice creates GL entries that debit an expense or inventory account and credit accounts payable. A payment entry creates GL entries

that debit or credit cash and the corresponding receivable or payable. This design means that the GLEntry table is the single source of truth for the organization's financial position, and every balance reported on the financial statements can be verified by querying that table.

The ERPNext database also illustrates the concept of a polymorphic relationship, though you do not need to know that term for this course. The GLEntry table uses two columns, voucher type and voucher number, to link entries to their source documents. The voucher type column identifies which source table to look in (SalesInvoice, PurchaseInvoice, JournalEntry, PaymentEntry), and the voucher number identifies the specific record within that table. This design is common in ERP systems and is more flexible than having a separate foreign key column for each possible source table, but it requires you to filter by voucher type before following the reference.

> **WATCH OUT** The voucher type and voucher number pattern in ERPNext is different from a standard foreign key because the database cannot enforce referential integrity across multiple target tables. This means that a GLEntry could theoretically reference a voucher number that does not exist in the corresponding source table. In Chapter 12, you will write SQL queries to test for exactly this kind of referential integrity problem as part of an audit analytics exercise.

Comparing the Three Database Designs

Looking at all three databases together reveals how the relational model adapts to businesses of different sizes and complexities. Several patterns emerge from the comparison.

The first pattern is that database complexity increases with business complexity. Northwind has 8 tables and represents a straightforward buy-and-sell operation. AdventureWorks has approximately 70 tables and represents a manufacturing company with multiple departments. ERPNext has dozens of tables spanning a full ERP environment. The number of tables is not the important metric. What matters is the range of business activities captured and the depth of relationships between them. Each additional business process, whether it is manufacturing, multi-territory sales, or integrated budgeting, adds tables and relationships to the database.

The second pattern is that the accounting depth of the database varies. Northwind has no general ledger or chart of accounts. Its data supports sales and purchasing analysis but not financial statement preparation. AdventureWorks adds cost and production data but still lacks a complete accounting module. ERPNext provides the full accounting cycle from chart of accounts through general ledger to financial statements. This progression is deliberate. The textbook introduces Northwind first for its simplicity, adds AdventureWorks when cost and production analysis become relevant, and brings in ERPNext when the exercises require working with the general ledger and financial statements.

The third pattern is that the same relational principles apply at every scale. A foreign key in Northwind works exactly the same way as a foreign key in ERPNext. A one-to-many relationship between customers and orders in a small database follows the same rules as a one-to-many relationship between accounts and GL entries in a large one. The SQL syntax, the Power BI modeling techniques, and the Excel lookup functions you will learn all operate on these same relational foundations regardless of database size.

[Table 3.2: Comparison of the Three Textbook Database Designs. A table with rows for Number of Core Tables, Schema or Functional Organization, Accounting Module Presence, Key Relationship Examples, Junction Table Examples, and Self-Referencing Relationships. Columns are Northwind Traders, Adventure Works Cycles, and ERPNext Demo Company.]

Guided Tutorial 3.1: Reading the Northwind ER Diagram

Context and objective. In this tutorial, you will study the full Northwind ER diagram and practice identifying tables, keys, relationships, and cardinality. The goal is to develop fluency in reading ER diagrams so that you can navigate any database structure you encounter in later chapters or in professional practice.

Prerequisites. You need access to the Northwind ER diagram provided with this textbook (printed in this chapter as Figure 3.7 and also available as a separate file in the companion materials).

Step-by-step instructions.

Step 1. Locate Figure 3.7, the full Northwind ER diagram. Begin by counting the tables. You should identify eight tables: Customers, Orders, OrderDetails, Products, Categories, Suppliers, Employees, and Shippers. Each table is represented as a rectangle with a header bar and a list of columns.

Step 2. Identify the primary key of each table. Look for the column marked "PK" in each table. The primary keys are CustomerID (Customers), OrderID (Orders), ProductID (Products), CategoryID (Categories), SupplierID (Suppliers), EmployeeID (Employees), and ShipperID (Shippers). The OrderDetails table has a composite primary key consisting of both OrderID and ProductID together, because a single order can include a given product only once, and the combination of order and product uniquely identifies each line item.

Step 3. Identify the foreign keys in the Orders table. The Orders table contains three foreign keys: CustomerID (referencing Customers), EmployeeID (referencing Employees), and ShipVia (referencing Shippers). Each foreign key links the order to a related entity. Trace each foreign key to the table it references and confirm that the referenced column is the primary key of that table.

Step 4. Read the cardinality of the relationship between Customers and Orders. Find the line connecting the two tables. The Customers end should show a single bar (meaning "one"), and

the Orders end should show a crow's foot (meaning "many"). Read the relationship aloud: "One customer can have many orders. Each order belongs to one customer."

Step 5. Find the OrderDetails table. Notice that it sits between Orders and Products. Trace its two foreign keys: OrderID connects to the Orders table, and ProductID connects to the Products table. Both relationships are one-to-many (one order has many line items, one product appears on many line items). The OrderDetails table is the junction table that resolves the many-to-many relationship between orders and products.

Step 6. Trace a complete path through the diagram. Start at the Categories table. Follow the relationship to the Products table (one category has many products). From Products, follow the relationship to OrderDetails (one product appears on many order line items). From OrderDetails, follow the relationship to Orders (many line items belong to one order). From Orders, follow the relationship to Customers (many orders belong to one customer). You have now traced a path from product category to customer, crossing four tables and three relationships. This is the kind of path you will traverse in SQL queries when you want to answer questions such as "Which customers purchased products in the Seafood category?"

[Figure 3.10: Annotated Northwind ER Diagram with Traced Path. The same ER diagram as Figure 3.7, with a highlighted path from Categories through Products, OrderDetails, Orders, and Customers. Numbered callout markers identify each step in the path, and annotations explain the cardinality at each connection.]

Checkpoint. You should now be able to answer the following questions by reading the Northwind ER diagram. How many foreign keys does the Orders table contain? (Three: CustomerID, EmployeeID, and ShipVia.) Which table serves as the junction table between Orders and Products? (OrderDetails.) What is the cardinality of the relationship between Categories and Products? (One-to-many: one category, many products.) If you can answer all three questions correctly, you have the ER diagram reading skills you will need for the chapters ahead.

Guided Tutorial 3.2: Comparing Database Designs Across All Three Datasets

Context and objective. In this tutorial, you will compare the ER diagrams of all three textbook databases to understand how different business environments produce different data structures. The goal is to build your ability to assess a database design and determine what analytical questions it can support.

Prerequisites. You need access to the ER diagrams for all three datasets (Figures 3.7, 3.8, and 3.9 in this chapter, with full versions in Appendix B).

Step-by-step instructions.

Step 1. Examine the Northwind ER diagram (Figure 3.7) and identify the tables that store revenue-related data. The relevant tables are Orders and OrderDetails, which together record what was sold, to whom, in what quantity, at what price, and with what discount. Note that calculating total revenue requires joining these two tables because the Orders table has the order-level information while the OrderDetails table has the line-item amounts.

Step 2. Examine the AdventureWorks focused ER diagram (Figure 3.8) and identify the tables that store revenue-related data. The relevant tables are SalesOrderHeader and SalesOrderDetail, which follow the same header-detail pattern as Northwind. Additionally, the SalesTerritory table adds geographic context, and the Product table adds cost data (StandardCost and ListPrice) that Northwind does not provide. Note that AdventureWorks enables profitability analysis (revenue minus cost) while Northwind provides only the revenue side.

Step 3. Examine the ERPNext focused ER diagram (Figure 3.9) and identify the tables that store revenue-related data. In ERPNext, revenue appears in two places. The SalesInvoice table records individual revenue transactions at the invoice level. The GLEntry table records the corresponding general ledger entries, including the revenue account credits. The Account table classifies the revenue accounts within the chart of accounts hierarchy. Note that ERPNext enables financial statement preparation because it has the full accounting chain from source document (SalesInvoice) through general ledger (GLEntry) to account classification (Account).

Step 4. For each dataset, determine whether you could prepare a trial balance from the available data. A trial balance requires a list of all accounts and their balances, calculated by summing debits and credits from the general ledger. Northwind does not have a general ledger or chart of accounts, so a trial balance is not possible. AdventureWorks has product cost data but no general ledger, so a trial balance is not possible. ERPNext has both a chart of accounts (Account table) and a general ledger (GLEntry table), so a trial balance can be prepared by grouping GL entries by account and summing debits and credits.

Step 5. For each dataset, identify one analytical question that the database can answer and one that it cannot. Write these down in a brief comparison. For example, Northwind can answer "What is the total revenue by product category?" but cannot answer "What is the company's net income?" because it lacks expense and equity data. AdventureWorks can answer "What is the scrap rate by product line?" but cannot answer "What is the balance in the accounts payable control account?" because it lacks a general ledger. ERPNext can answer "What is the balance of every account on the trial balance?" but may have limited product-level manufacturing detail compared to AdventureWorks.

Checkpoint. You should now have a written comparison that identifies the analytical strengths and limitations of each database design. You should be able to explain why a database with more tables is not necessarily better than one with fewer tables. The right database for a given task is the one whose tables and relationships contain the specific data that the task requires.

Looking Ahead

This chapter has introduced the relational database model and shown you how accounting data is organized within it. You have learned how tables, rows, columns, and data types store information, how primary keys and foreign keys create relationships between tables, and how Entity-Relationship diagrams make these structures visible. You have examined the database designs of all three textbook datasets and compared their ability to support different accounting tasks.

In the next chapter, you will begin working with data directly. Chapter 4 introduces Excel as an analytical tool and teaches you to organize accounting data in Excel Tables, apply sorting and filtering, and use conditional formatting to identify patterns. The relational concepts from this chapter will remain in the background as you work, because the data you import into Excel comes from the same tables and relationships you have just studied.

Chapter Summary

Accounting data in modern organizations is stored in relational databases, which organize information into tables connected by defined relationships. Understanding this structure is a prerequisite for writing SQL queries, building Power BI data models, and preparing data for analysis in Excel. The relational model, first proposed by Codd (1970), has become the standard approach to data storage in business information systems and underlies virtually every ERP system and accounting software package in use today.

A relational database consists of tables, each storing information about one type of entity or event. Tables are organized into rows (one per instance) and columns (one per attribute), and every column has a defined data type that determines what values it can hold and what operations are possible. Primary keys uniquely identify each row in a table, and foreign keys create connections between tables by referencing the primary key of a related table. These keys formalize the cross-references that have always existed in accounting records: an invoice references a customer, a journal entry references an account, and a line item references a product.

Relationships between tables have a cardinality that specifies how many rows on each side of the relationship can be associated. One-to-many relationships are the most common in accounting data: one customer has many orders, one account has many GL entries, one category contains many products. Many-to-many relationships are resolved through junction tables, such as Northwind's OrderDetails table that connects orders to products. Understanding cardinality is essential for avoiding analytical errors such as double-counting when combining data from multiple tables.

Entity-Relationship diagrams provide a visual map of a database's structure. They show tables as labeled rectangles, relationships as connecting lines, and cardinality through crow's foot notation. Reading an ER diagram is a practical skill that auditors, analysts, and accountants use when planning data extractions, designing queries, and building data models. The three ER diagrams studied in this chapter, for Northwind, AdventureWorks, and ERPNext, illustrate how database complexity scales with business complexity while the underlying relational principles remain constant.

The familiar structures of accounting, including the chart of accounts, the general ledger, and the sub-ledgers, map directly to tables and relationships in a database. The chart of accounts is a table with a self-referencing hierarchy. The general ledger is a table of debit and credit entries linked to the chart of accounts through a foreign key. Sub-ledgers are tables of detailed transaction records that connect to the general ledger through voucher references. These mappings make it possible to perform financial reporting, audit testing, and managerial analysis directly from the database, which is exactly what you will do in the chapters ahead.

Key Terms

Cardinality. The specification of how many rows on each side of a relationship can be associated with one another. The three types are one-to-one, one-to-many, and many-to-many. In accounting databases, one-to-many is the most common cardinality.

Column. A single attribute within a database table, representing one piece of information recorded for every row. Examples include OrderDate, AccountName, and UnitPrice.

Composite primary key. A primary key consisting of two or more columns together, where no single column alone is sufficient to uniquely identify a row. The Northwind OrderDetails table uses a composite primary key of OrderID and ProductID.

Crow's foot notation. A visual notation system used in Entity-Relationship diagrams to indicate cardinality. A single bar represents "one" and a three-pronged fork represents "many."

Data type. The defined category of values that a column can hold, such as integer, decimal, text, date, or Boolean. Data types determine which operations can be performed on a column's values.

Entity-Relationship (ER) diagram. A visual representation of the tables in a database and the relationships between them, showing table names, column names, keys, and cardinality using standardized notation.

Foreign key. A column in one table that references the primary key of another table, creating a defined relationship between the two tables. For example, CustomerID in the Northwind Orders table is a foreign key referencing the Customers table.

Junction table. A table that resolves a many-to-many relationship between two other tables by converting it into two one-to-many relationships. The Northwind OrderDetails table is a junction table between Orders and Products.

Many-to-many relationship. A relationship in which each row in either table can be associated with multiple rows in the other table. Relational databases represent many-to-many relationships through junction tables.

One-to-many relationship. A relationship in which one row in the first table can be associated with many rows in the second table, but each row in the second table is associated with only one row in the first table. This is the most common relationship type in accounting databases.

Primary key. A column or combination of columns that uniquely identifies each row in a table. No two rows can share the same primary key value, and the value cannot be blank.

Referential integrity. The property of a database in which every foreign key value corresponds to an existing primary key value in the referenced table. Referential integrity prevents orphaned records that reference nonexistent related entries.

Relational database. A data storage system that organizes information into tables connected by defined relationships through primary and foreign keys. The three textbook datasets are all relational databases provided in SQLite format.

Row. A single record within a database table, representing one instance of the entity or event the table describes. Each row in the Northwind Orders table represents one customer order.

Schema. A logical grouping of related tables within a database. AdventureWorks uses schemas such as Sales, Production, Purchasing, and Human Resources to organize its approximately 70 tables.

Self-referencing relationship. A relationship in which a foreign key in a table references the primary key of the same table, creating a hierarchy within a single table. The ERPNext Account table uses this pattern to define the parent-child structure of the chart of accounts.

Table. The fundamental storage unit in a relational database, organized into rows and columns. Each table stores information about one type of entity or event, such as customers, orders, or general ledger entries.

Multiple Choice Questions

1. In a relational database, which of the following best describes the role of a primary key?

A. It stores the most important data value in the table. B. It uniquely identifies each row in a table and cannot be duplicated or left blank. C. It connects one table to another by referencing a column in a different table. D. It defines the data type for all columns in the table.

2. The CustomerID column appears in both the Northwind Customers table and the Northwind Orders table. In the Orders table, this column functions as a:

A. Primary key B. Foreign key C. Composite key D. Data type

3. An ER diagram shows a line between two tables. The line has a single bar on one end and a crow's foot on the other end. This notation represents which type of relationship?

A. One-to-one B. One-to-many C. Many-to-many D. The notation does not indicate cardinality

4. The Northwind OrderDetails table contains the columns OrderID and ProductID, both of which are foreign keys. This table exists because:

A. It stores the primary key of the Orders table. B. It resolves a many-to-many relationship between orders and products. C. It is required for every database to have at least one table with two foreign keys. D. It stores customer contact information for each order.

5. A cost accountant wants to compare standard product costs to actual production costs. Which textbook database is best suited for this analysis?

A. Northwind, because it contains product pricing data B. AdventureWorks, because it contains both standard cost data in the Product table and production data in the WorkOrder table C. ERPNext, because it contains a complete general ledger D. Any of the three databases would work equally well for this analysis

6. In the ERPNext database, the Account table contains a column called ParentAccount that references another row in the same Account table. This is an example of a:

A. Foreign key referencing a different table B. Composite primary key C. Self-referencing relationship D. Many-to-many relationship

7. An auditor wants to prepare a trial balance directly from database tables. Which textbook dataset provides the necessary data?

A. Northwind, using the Orders and OrderDetails tables B. AdventureWorks, using the SalesOrderHeader and Product tables C. ERPNext, using the Account table and the GLEntry table D. None of the three datasets supports trial balance preparation

8. In the AdventureWorks database, tables are organized into schemas such as Sales, Production, and Purchasing. The primary purpose of schemas is to:

A. Restrict access to certain tables based on user roles B. Group related tables together for logical organization and easier navigation C. Increase the speed of database queries D. Store backup copies of tables in case of data loss

9. A relational database enforces referential integrity. This means that:

A. Every table must contain at least one foreign key. B. Primary key values can be duplicated as long as the duplicates are in different tables. C. Every foreign key value must correspond to an existing primary key value in the referenced table. D. Tables without relationships are automatically deleted from the database.

10. An analyst joins the Northwind Customers table to the Orders table and then to the OrderDetails table. One customer has 10 orders, and those 10 orders contain a total of 45 line items. How many rows will this customer contribute to the joined result?

A. 1 row B. 10 rows C. 45 rows D. 55 rows

11. Which of the following accounting structures is represented in a database as a self-referencing table where child rows point to parent rows within the same table?

A. The general ledger B. The trial balance C. The chart of accounts hierarchy D. The accounts receivable sub-ledger

12. The ERPNext GLEntry table uses a voucher type and voucher number to link general ledger entries to their source documents. This design differs from a standard foreign key because:

A. It uses two columns instead of one. B. It can reference multiple different tables depending on the voucher type, which prevents the database from enforcing referential integrity automatically. C. It does not require the referenced record to exist. D. It is only used for journal entries and not for other transaction types.

13. Which of the following is the best definition of a relational database?

A. A database that stores all data in a single large table B. A data storage system that organizes information into tables connected through defined relationships using primary and foreign keys C. A spreadsheet with multiple worksheets D. A database that requires SQL to access its data

14. A database table has a column defined as the INTEGER data type. Which of the following values could this column store?

A. “Accounts Receivable” B. 2024-03-15 C. 4500 D. 99.73

15. The Northwind database has 8 tables and the AdventureWorks database has approximately 70 tables. Which of the following best explains this difference?

A. AdventureWorks is a newer database and uses more modern design principles. B. Northwind contains errors that should have been stored in additional tables. C. AdventureWorks captures more business processes, including manufacturing, multi-territory sales, and human

resources, each of which requires its own set of tables. D. The number of tables is arbitrary and has no relationship to the complexity of the business.

Applied Exercises

Financial Accounting Exercises

Exercise 3.1 (Financial Accounting): Tracing a Revenue Transaction Through the ERPNext Database

Dataset: ERPNext (Account, GLEntry, SalesInvoice)

Scenario. You are a financial reporting analyst preparing to verify that revenue transactions recorded in the ERPNext system are properly reflected in the general ledger. Before writing any queries or performing any calculations, you need to understand the path a revenue transaction follows through the database, from the source document to the account balance.

Requirements. (1) Open the ERPNext dataset in Excel and examine the SalesInvoice, GLEntry, and Account tables. For a revenue transaction, identify the specific columns in each table that would allow you to trace the transaction from the invoice, through the general ledger entry, to the account in the chart of accounts. Write down the column names that serve as the connecting links between these three tables. (2) Select one sales invoice from the SalesInvoice table. Record its voucher number. Then search the GLEntry table for all entries that reference this voucher number. Identify the accounts that were debited and credited. Verify that the debit and credit amounts for this transaction are equal. (3) For each account referenced in the general ledger entries you found, locate the account in the Account table and note its account type (asset, liability, equity, revenue, or expense). Confirm that the account classifications make sense for a revenue transaction (for example, the credit should be to a revenue account and the debit should be to a receivable or cash account). (4) Prepare a brief written trace (one page) that documents the path of the transaction through the three tables, identifying each table, the relevant columns, the connecting keys, and the amounts involved. This trace represents the type of vouching procedure that auditors perform to verify financial statement assertions.

Deliverable. A one-page written transaction trace documenting the path of a revenue transaction through the ERPNext database.

Exercise 3.2 (Financial Accounting): Mapping the Northwind Database to the Revenue Cycle

Dataset: Northwind (Customers, Orders, OrderDetails, Products)

Scenario. You are a financial reporting analyst who has been asked to document how the Northwind database captures the revenue cycle. Your documentation will be used by the team as a reference when preparing revenue analyses in later chapters.

Requirements. (1) Using the Northwind ER diagram (Figure 3.7), identify all tables involved in recording a sale from the initial order through the line-item details to the product and customer information. List each table and describe its role in the revenue cycle in one to two sentences. (2) For each pair of connected tables in the revenue cycle path, identify the foreign key that creates the connection and state the cardinality of the relationship. (3) Identify what revenue cycle information is missing from the Northwind database. Consider whether the database records cash receipts, tracks accounts receivable balances, or includes a general ledger. For each gap you identify, explain what additional table or tables would be needed and what columns they would contain. (4) Prepare a brief written summary (one page) that could serve as a data dictionary excerpt for the revenue-related tables in Northwind.

Deliverable. A one-page data dictionary excerpt covering the Northwind revenue cycle tables, their relationships, and their limitations.

Managerial Accounting Exercises

Exercise 3.3 (Managerial Accounting): Mapping the AdventureWorks Production Database

Dataset: AdventureWorks (Product, WorkOrder, BillOfMaterials, ProductSubcategory)

Scenario. You are a cost analyst at Adventure Works Cycles. Your manager has asked you to document the database tables that support manufacturing cost analysis so that the team can plan a product costing project for the upcoming quarter.

Requirements. (1) Using the AdventureWorks ER diagram and the Excel workbook, identify all tables that store production-related data. For each table, write one to two sentences describing what it contains and how it relates to manufacturing cost analysis. (2) Trace the path through the database that connects a finished product to its component materials. Start at the Product table and follow the relationships through the BillOfMaterials table. Explain how a cost analyst would use this path to calculate the material cost of a product. (3) Identify the table that records actual production activity (WorkOrder). Examine its columns and determine what information is available for comparing planned production to actual production, including quantities ordered, quantities produced, and quantities scrapped. (4) Determine whether the AdventureWorks database allows you to calculate the variance between standard cost and actual cost for a specific product. Identify which tables and columns would be involved and note any gaps where the data may be insufficient. (5) Write a brief memorandum (one page) summarizing the production-related data available in AdventureWorks and assessing its suitability for a product cost analysis project.

Deliverable. A one-page memorandum documenting the AdventureWorks production data and its suitability for cost analysis.

Exercise 3.4 (Managerial Accounting): Evaluating Cost Center Data in ERPNext

Dataset: ERPNext (CostCenter, GLEntry, Budget)

Scenario. You are a management accountant preparing to build a departmental performance report that compares budgeted expenses to actual expenses by cost center. Before starting the analysis, you need to understand how cost center data is structured in the ERPNext database.

Requirements. (1) Open the ERPNext dataset and examine the CostCenter table. Identify the columns available and describe the hierarchical structure if one exists. (2) Examine the GLEntry table and determine how general ledger entries are associated with cost centers. Identify the column that links GL entries to the CostCenter table and state the cardinality of the relationship. (3) Examine the Budget table, if available, and determine how budgeted amounts are associated with cost centers and accounts. Identify the columns that connect budget records to cost centers and to the chart of accounts. (4) Assess whether the ERPNext database provides sufficient data to compare budgeted versus actual spending for each cost center. Identify any gaps, such as missing cost center assignments on GL entries or budget records that do not correspond to actual GL account codes. Write a brief assessment (one half to one page) documenting your findings.

Deliverable. A written assessment of the ERPNext cost center data structure and its readiness for budget-versus-actual analysis.

Auditing Exercises

Exercise 3.5 (Auditing): Planning a Data Extraction Strategy

Dataset: All three (Northwind, AdventureWorks, ERPNext)

Scenario. You are an IT auditor responsible for planning the data extraction needed to support three audit procedures: a revenue completeness test, a purchasing cycle test, and a journal entry analysis for management override of controls. For each procedure, you need to identify which database to use, which tables to extract, and which relationships you will need to traverse.

Requirements. (1) For the revenue completeness test, identify the database and tables you would use. Explain which table contains the population of revenue transactions and which related tables you would need to join in order to verify that each transaction includes a valid customer, a valid product, and an amount. State the keys that connect the tables. (2) For the purchasing cycle test, identify the database and tables you would use. The test requires you to verify that every purchase order was placed with an approved vendor and that the quantities and prices on the purchase order match the quantities and prices recorded in the receiving

records. Identify which tables are relevant and note any gaps in the available data. (3) For the journal entry analysis, identify the database and tables you would use. The analysis requires access to every manually created journal entry, including the posting date, the amounts, the accounts debited and credited, and the person who created the entry. Identify which tables contain this information and which columns would be useful for risk-based filtering (such as entries posted on weekends or entries with round-dollar amounts). (4) Prepare a data extraction plan (one to two pages) that documents your selections for all three procedures, including the database, tables, key columns, and relationships for each.

Deliverable. A one-to-two page data extraction plan for three audit procedures, identifying specific tables and relationships in the textbook databases.

Exercise 3.6 (Auditing): Assessing Referential Integrity Across the Northwind Database

Dataset: Northwind (all tables)

Scenario. Your audit senior has asked you to perform a preliminary check on the referential integrity of the Northwind database. Referential integrity problems, where a foreign key references a primary key value that does not exist, can indicate data migration errors, deletion of master records without updating related transactions, or system configuration problems. Any integrity violations you find will be flagged for further investigation.

Requirements. (1) Using the Northwind ER diagram, identify all foreign key relationships in the database. List each relationship by specifying the child table, the foreign key column, the parent table, and the primary key column. (2) For each relationship you identified, use Excel lookup or counting functions to test whether every foreign key value in the child table exists as a primary key value in the parent table. Record the number of orphaned references (if any) for each relationship. (3) Pay particular attention to the relationship between Orders and Customers, between OrderDetails and Products, and between Products and Suppliers. For any orphaned references you find, record the specific foreign key values that have no match and the number of records affected. (4) Prepare a brief memorandum (one page) documenting your referential integrity testing results. For each relationship tested, state whether integrity was maintained or violated, and for any violations, describe the potential impact on financial analyses that depend on those relationships.

Deliverable. A one-page referential integrity testing memorandum documenting the results for all foreign key relationships in the Northwind database.

Further Reading

Codd, E. F. (1970). A relational model of data for large shared data banks. *Communications of the ACM*, 13(6), 377-387. This paper introduced the relational database model that underlies

virtually every modern business information system. Although the paper is technical, the core ideas of organizing data into tables with defined relationships are accessible, and understanding their origin helps explain why relational databases work the way they do. This is one of the most influential papers in the history of computing.

Dunn, C. L., Cherrington, J. O., and Hollander, A. S. (2005). *Enterprise information systems: A pattern-based approach* (3rd ed.). McGraw-Hill/Irwin. This textbook provides a comprehensive treatment of how accounting information systems are designed using relational database concepts. The chapters on REA (Resources, Events, Agents) modeling are particularly useful for understanding how accounting transactions map to database tables and relationships.

Gelinas, U. J., Dull, R. B., and Wheeler, P. R. (2018). *Accounting information systems* (11th ed.). Cengage Learning. This widely used accounting information systems textbook includes detailed coverage of relational databases, ER diagrams, and their application to accounting data. The chapters on database design and data modeling provide additional depth on the concepts introduced in this chapter.

Sledgianowski, D., Gomaa, M., and Tan, C. (2017). Toward integration of Big Data, technology, and information systems competencies into the accounting curriculum. *Journal of Accounting Education*, 38, 81-93. This paper reports on a survey of accounting professionals about the technology competencies needed by entry-level accountants. The findings support the claim that understanding database concepts is valued by employers and that accounting curricula should include instruction on relational data structures.

AICPA (American Institute of Certified Public Accountants). (2017). *Guide to audit data analytics.* AICPA. This professional guidance document includes recommendations for auditors on understanding client data structures before performing data analytics procedures. The guide's discussion of data extraction planning and relationship mapping aligns with the ER diagram reading skills taught in this chapter.

McCarthy, W. E. (1982). The REA accounting model: A generalized framework for accounting systems in a shared data environment. *The Accounting Review*, 57(3), 554-578. This paper introduced the REA model, which applies entity-relationship concepts specifically to accounting system design. The framework describes accounting data in terms of resources, events, and agents rather than debits and credits, providing an alternative perspective on how accounting transactions are represented in databases. Understanding the REA model deepens the connection between the database concepts in this chapter and the accounting concepts students already know.

Romney, M. B., and Steinbart, P. J. (2018). *Accounting information systems* (14th ed.). Pearson. This textbook provides thorough coverage of relational databases and ER modeling in the context of accounting information systems. The chapters on database design, data modeling, and REA diagrams offer additional practice with the concepts introduced in this chapter, and the book includes numerous exercises using accounting scenarios.

Part II: Accounting Analytics with Excel

Part III: Accounting Analytics with SQL

Part IV: Data Visualization and Power BI

Part V: Integrated and Applied Topics