

Sql Antipatterns Avoiding The Pitfalls Of Database Programming Pragmatic Programmers

This book presents recent research work and results in the area of communication and information technologies. The book includes the main results of the 11th International Conference on Computing and Information Technology (IC2IT) held during July 2nd-3rd, 2015 in Bangkok, Thailand. The book is divided into the two main parts Data Mining and Machine Learning as well as Data Network and Communications. New algorithms and methods of data mining are discussed as well as innovative applications and state-of-the-art technologies on data mining, machine learning and data networking. T-SQL insiders help you tackle your toughest queries and query-tuning problems Squeeze maximum performance and efficiency from every T-SQL query you write or tune. Four leading experts take an in-depth look at T-SQL's internal architecture and offer advanced practical techniques for optimizing response time and resource usage. Emphasizing a correct understanding of the language and its foundations, the authors present unique solutions they have spent years developing and refining. All code and techniques are fully updated to reflect new T-SQL enhancements in Microsoft SQL Server 2014 and SQL Server 2012. Write faster, more efficient T-SQL code: Move from procedural programming to the language of sets and logic Master an efficient top-down tuning methodology Assess algorithmic complexity to predict performance Compare data aggregation techniques, including new grouping sets Efficiently perform data-analysis calculations Make the most of T-SQL's optimized bulk import tools Avoid date/time pitfalls that lead to buggy, poorly performing code Create optimized BI statistical queries without additional software Use programmable objects to accelerate queries Unlock major performance improvements with In-Memory OLTP Master useful and elegant approaches to manipulating graphs About This Book For experienced T-SQL practitioners Includes coverage updated from Inside Microsoft SQL Server 2008 T-SQL Querying and Inside Microsoft SQL Server 2008 T-SQL Programming Valuable to developers, DBAs, BI professionals, and data scientists Covers many MCSE 70-464 and MCSA/MCSE 70-461 exam topics

If you create, manage, operate, or configure systems running in the cloud, you're a cloud engineer--even if you work as a system administrator, software developer, data scientist, or site reliability engineer. With this book, professionals from around the world provide valuable insight into today's cloud engineering role. These concise articles explore the entire cloud computing experience, including fundamentals, architecture, and migration. You'll delve into security and compliance, operations and reliability, and software development. And examine networking, organizational culture, and more. You're sure to find 1, 2, or 97 things that inspire you to dig deeper and expand your own career. "Three Keys to Making the Right Multicloud Decisions," Brendan O'Leary "Serverless Bad Practices," Manases Jesus Galindo Bello "Failing a Cloud Migration," Lee Atchison "Treat Your Cloud Environment as If It Were On Premises," Iyana Garry "What Is Toil, and Why Are SREs Obsessed with It?," Zachary Nickens "Lean QA: The QA Evolving in the DevOps World," Theresa Neate "How Economies of Scale Work in the Cloud," Jon Moore "The Cloud Is Not About the Cloud," Ken Corless "Data Gravity: The Importance of Data Management in the Cloud," Geoff Hughes "Even in the Cloud, the Network Is the Foundation," David Murray "Cloud Engineering Is About Culture, Not Containers," Holly Cummins

This book constitutes the refereed proceedings of the Software Engineering and Algorithms section of the 10th Computer Science On-line Conference 2021 (CSOC 2021), held on-line in April 2021. Software engineering research and its applications to intelligent algorithms take an essential role in computer science research. In this book, modern research methods, application of machine and statistical learning in the software engineering research are presented.

Best Practices for Scalability and Performance

Strategies for Building Successful Teams and Organizations

Seven Databases in Seven Weeks

Learn T-SQL Querying

Database Design for Mere Mortals

Infrastructure as Code

The Art of SQL

A guide to SQL covers such topics as retrieving records, metadata queries, working with strings, data arithmetic, date manipulation, reporting and warehousing, and hierarchical queries.

Effective SQL brings together the hands-on solutions and practical insights you need to solve a wide range of complex problems with SQL, and to design databases that make it far easier to manage. In the proven format of the best-selling Effective series, it focuses on providing clear, practical explanations, expert tips, and plenty of realistic examples -- all in full color. Drawing on their immense experience as database instructors, three world-class database experts identify specific challenges, and distill each solution into five pages or less. Throughout, they provide well-annotated SQL code designed for all leading database systems. For specific implementations ranging from SQL Server to Oracle and MySQL, wherever these vary or permit you to achieve your goal more efficiently. Going beyond mere syntax, the authors also show you how to design a database that makes it difficult to write effective SQL, how to improve suboptimal designs, and how to work around designs you can't change. You'll also find detailed sections on filtering, subqueries, and metadata, as well as specific solutions for everything from listing products to scheduling events and defining data hierarchies. Simply put, if you already know the basics of SQL, Effective SQL will help you become a world-class SQL problem-solver.

When it comes to choosing, using, and maintaining a database, understanding its internals is essential. But with so many distributed databases and tools available today, it's often difficult to understand how they differ. With this practical guide, Alex Petrov guides developers through the concepts behind modern database and storage engine internals. Throughout the book, you'll explore relevant research, books, papers, blog posts, and the source code of several open source databases. These resources are listed at the end of parts one and two. You'll discover that the most significant distinctions between databases reside in subsystems that determine how storage is organized and how data is distributed. This book examines: Storage engines: Explore storage classification and taxonomy, and dive into B-Tree and LSM-Tree. Structured storage engines, with differences and use-cases for each Storage building blocks: Learn how database files are organized to build efficient storage, using auxiliary data structures such as LSM-Tree. Write-Ahead Log Distributed systems: Learn step-by-step how nodes and processes connect and build complex communication patterns Database clusters: Which consistency models are common and how distributed storage systems achieve consistency

"This book takes the somewhat daunting process of database design and breaks it into completely manageable and understandable components. Mike's approach whilst simple is completely professional. I would recommend this book to any novice database designer." --Sandra Barker, Lecturer, University of South Australia, Australia "Databases are a critical infrastructure technology for information systems and today's organizations have never had it so easy to design a database. Mike has written a literate explanation of database technology--a topic that is intricate and often obscure. If you design databases yourself, this book will educate you about pitfalls and show you why

that use a database, the book explains the technology so that you can understand what the vendor is doing and assess their products better." --Michael Blaha, consultant and trainer, author of *Database Design for Mere Mortals* Technology "If you told me that Mike Hernandez could improve on the first edition of *Database Design for Mere Mortals* I wouldn't have believed you, but he did! The second edition is packed with detailed explanations, and even includes database-design tools on the CD-ROM! This is a must-read for anyone who is even remotely interested in relational database design, from the individual who wants to create a useful tool at work, to the seasoned professional who wants to brush up on the fundamentals. Simply put, if you want to do it right, read this book!" --Matt Greer, Process Control Design Company "Mike's approach to database design is totally common-sense based, yet he's adhered to all the rules of good relational database design. I use Mike's books in my starter database-design courses for anyone who's interested in learning how to design databases or how to write SQL queries." --Michelle Poolet, President, MVDS, Inc. "Slapping together sophisticated applications with poor database design is just as much now as when Mike wrote his first edition, perhaps even more. Whether you're just getting started developing with data or are a seasoned pro; whether you've read Mike's previous books you're happier letting someone else design your data or you love doing it yourself--this is the book for you. Mike's ability to explain these concepts in a way that's not only clear, but fun, continues to be a pleasure." Foreword by Ken Getz, MCW Technologies, coauthor *ASP.NET Developer's JumpStart* "The first edition of Mike Hernandez's book *Database Design for Mere Mortals* was one of the few books that I brought to my office to smaller quarters. The second edition expands and improves on the original in so many ways. It is not only a good, clear read, but contains a remarkable quantity of clear, concise things to do. It's a must for anyone interested in the subject of database design." --Malcolm C. Rubel, Performance Dynamics Associates "Mike's excellent guide to relational database design deserves a second edition as a tool for fledgling Microsoft Access and other desktop database developers, as well as for client/server pros. I recommend it highly to all my readers." --Roger Jennings, author of *Special Edition Unix Shell Scripts* silver bullets! Database technology has advanced dramatically, the newest crop of database servers perform operations faster than anyone could have imagined six years ago, but none of these technologies can fix a bad database design, or capture data that you forgot to include! *Database Design for Mere Mortals(TM)*, Second Edition, helps you design your database right in the first place!" --Matt Nunn, Microsoft Server, Microsoft Corporation "When my brother started his professional career as a developer, I gave him Mike's book to help him understand database concepts and make real-world applications. If you need a refresher on the finer points of database design, this is the book I pick up. I do not think that there is a better testimony to the value of a book than that it gets used. For this reason I have recommended it to my peers and students that they utilize this book in their day-to-day development tasks." --Chris Kunicki, Senior Consultant, OfficeZealot.com "Mike has always had an incredible knack for taking the most complex and breaking them down, and explaining them so that anyone can 'get it.' He has honed and polished his first very, very good edition and made it even better. If you're just starting out building databases, this is a must-read cover to cover. Expert designers will find Mike's approach fresh and enlightening and a source of great material for training others." --John Viescas, President, Viescas Consulting, Inc., author of *Microsoft Access 2000* and coauthor of *SQL Queries for Mere Mortals* "Whether you need to learn about relational database design in general, design a relational database, understand relational database design, or learn best practices for implementing a relational database, *Database Design for Mere Mortals(TM)*, Second Edition, is an indispensable book that you'll refer to often. With his many years of real-world experience with databases, Michael shows you how to analyze and improve existing databases, implement keys, define table relationships and business rules, and create data views, resulting in data integrity, uniformity, and no data-entry errors." --Paul Cornell, Site Editor, MSDN Office Developer Center Sound database design can save hours of development time and ensure functionality and reliability. *Database Design for Mere Mortals(TM)*, Second Edition, is a straightforward, platform-independent tutorial on the basic principles of relational database design. It provides a commonsense design methodology for developing databases. In this Second Edition, expert Michael J. Hernandez has expanded his best-selling first edition, maintaining its hands-on approach and accessibility while updating its coverage and including even more examples and illustrations. The new CD-ROM that includes diagrams of sample databases, as well as design guidelines, documentation forms, and examples of the database design process. This book will give you the knowledge and skills to design efficient and effective relational databases.

A Handbook of Agile Software Craftsmanship

Scaling Teams

Generating Optimal Execution Plans

Inside Microsoft SQL Server 2008

Embracing Microservices Design

A Geek's Guide to the Beauty of Numbers, Logic, and Computation

Best Practice Ruby on Rails Refactoring

Build a core level of competency in SQL so you can recognize the parts of queries and write simple SQL statements. SQL knowledge is essential for anyone involved in programming, data science, and data management. This book covers features of SQL that are standardized and common across most database vendors. You will gain a base of knowledge that will prepare you to go deeper into the specifics of any database product you might encounter. Examples in the book are worked in PostgreSQL and SQLite, but the bulk of the examples are platform agnostic and will work on any database platform supporting SQL. Early in the book you learn about table design, the importance of keys as row identifiers, and essential query operations. You then move into more advanced topics such as grouping and summarizing, creating calculated fields, joining data from multiple tables when it makes business sense to do so, and more. Throughout the book, you are exposed to a set-based approach to the language and are provided a good grounding in subtle but important topics such as the effects of null value on query results. With the explosion of data science, SQL has regained its prominence as a top skill to have for technologists and decision makers worldwide. *SQL Primer* will guide you from the very basics of SQL through to the mainstream features you need to have a solid, working knowledge of this important, data-oriented language. What You'll Learn Create and populate your own database tables Read SQL queries and understand what they are doing Execute queries that get correct results Bring together related rows from multiple tables Group and sort data in support of reporting applications Get a grip on nulls, normalization, and other key concepts Employ subqueries, unions, and other advanced features Who This Book Is For Anyone new to SQL who is looking for step-by-step guidance toward understanding and writing SQL queries. The book is aimed at those who encounter SQL statements often in their work, and provides a sound baseline useful across all SQL database systems. Programmers, database managers, data scientists, and business analysts all can benefit from the baseline of SQL knowledge provided in this book.

The study of software engineering and its applications to system engineering is critical in computer science research. Modern research methodologies, as well as the use of

machine and statistical learning in software engineering research, are covered in this book. This book contains the refereed proceedings of the Software Engineering Perspectives in Systems part of the 11th Computer Science On-line Conference 2022 (CSOC 2022), which was held in April 2022 online.

Offers tips for improving the performance of any SQL database, no matter what the platform. Written for experienced database administrators familiar with SQL, the book identifies the similarities and differences of eight DBMSs, including Oracle 9i, IBM DB2 7.2, and Microsoft SQL server 2000. It provides strategies for refining sorts, subqueries, columns, tables, indexes, constraints, and locks. Annotation copyrighted by Book News, Inc., Portland, OR

The Complete Guide to Avoiding and Fixing Common Rails 3 Code and Design Problems As developers worldwide have adopted the powerful Ruby on Rails web framework, many have fallen victim to common mistakes that reduce code quality, performance, reliability, stability, scalability, and maintainability. Rails™ AntiPatterns identifies these widespread Rails code and design problems, explains why they're bad and why they happen—and shows exactly what to do instead. The book is organized into concise, modular chapters—each outlines a single common AntiPattern and offers detailed, cookbook-style code solutions that were previously difficult or impossible to find. Leading Rails developers Chad Pytel and Tammer Saleh also offer specific guidance for refactoring existing bad code or design to reflect sound object-oriented principles and established Rails best practices. With their help, developers, architects, and testers can dramatically improve new and existing applications, avoid future problems, and establish superior Rails coding standards throughout their organizations. This book will help you understand, avoid, and solve problems with Model layer code, from general object-oriented programming violations to complex SQL and excessive redundancy Domain modeling, including schema and database issues such as normalization and serialization View layer tools and conventions Controller-layer code, including RESTful code Service-related APIs, including timeouts, exceptions, backgrounding, and response codes Third-party code, including plug-ins and gems Testing, from test suites to test-driven development processes Scaling and deployment Database issues, including migrations and validations System design for “graceful degradation” in the real world

SQL Queries for Mere Mortals

SQL Performance Tuning

SQL Cookbook

API Design for C++

A Step by Step Approach to Relational Database Design and Development

From Journeyman to Master

97 Things Every Cloud Engineer Should Know

Develop microservice-based enterprise applications with expert guidance to avoid failures and technological debt with the help of real-world examples Key FeaturesImplement the right microservices adoption strategy to transition from monoliths to microservicesExplore real-world use cases that explain anti-patterns and alternative practices in microservices developmentDiscover proven recommendations for avoiding architectural mistakes when designing microservicesBook DescriptionMicroservices have been widely adopted for designing distributed enterprise apps that are flexible, robust, and fine-grained into services that are independent of each other. There has been a paradigm shift where organizations are now either building new apps on microservices or transforming existing monolithic apps into microservices-based architecture. This book explores the importance of anti-patterns and the need to address flaws in them with alternative practices and patterns. You'll identify common mistakes caused by a lack of understanding when implementing microservices and cover topics such as organizational readiness to adopt microservices, domain-driven design, and resiliency and scalability of microservices. The book further demonstrates the anti-patterns involved in re-platforming brownfield apps and designing distributed data architecture. You'll also focus on how to avoid communication and deployment pitfalls and understand cross-cutting concerns such as logging, monitoring, and security. Finally, you'll explore testing pitfalls and establish a framework to address isolation, autonomy, and standardization. By the end of this book, you'll have understood critical mistakes to avoid while building microservices and the right practices to adopt early in the product life cycle to ensure the success of a microservices initiative. What you will learnDiscover the responsibilities of different individuals involved in a microservices initiativeAvoid the common mistakes in architecting microservices for scalability and resiliencyUnderstand the importance of domain-driven design when developing microservicesIdentify the common pitfalls involved in migrating monolithic applications to microservicesExplore communication strategies, along with their potential drawbacks and alternativesDiscover the importance of adopting governance, security, and monitoringUnderstand the role of CI/CD and testingWho this book is for This practical microservices book is for software architects, solution architects, and developers involved in designing microservices architecture and its development, who want to gain insights into avoiding pitfalls and drawbacks in distributed applications, and save time and money that might otherwise get wasted if microservices designs fail. Working knowledge of microservices is assumed to get the most out of this book.

This indispensable SQL reference book is the first of its kind to leverage the benefits of design patterns to relational database SQL queries; all common SQL structures and design patterns are clearly categorized and described. Emphasizing the theoretical foundation for almost every type of SQL query problem, accompanying figures are

included to help visualize the problem. Because SQL is a declarative language there are many ways to write any SQL query and professional database programmers must understand the correct way to write SQL for complicated database queries, and managers must institute formal SQL coding standards to improve productivity and maintainability. The SQL design patterns in this resource greatly improve the quality and productivity of systems development projects by forming a "best practices" foundation for all relational database queries.

SQL is the ubiquitous language for software developers working with structured data. Most developers who rely on SQL are experts in their favorite language (such as Java, Python, or Go), but they're not experts in SQL. They often depend on antipatterns - solutions that look right but become increasingly painful to work with as you uncover their hidden costs. Learn to identify and avoid many of these common blunders. Refactor an inherited nightmare into a data model that really works. Updated for the current versions of MySQL and Python, this new edition adds a dozen brand new mini-antipatterns for quick wins. No matter which platform, framework, or language you use, the database is the foundation of your application, and the SQL database language is the standard for working with it. Antipatterns are solutions that look simple at the surface, but soon mire you down with needless work. Learn to identify these traps, and craft better solutions for the often-asked questions in this book. Avoid the mistakes that lead to poor performance and quality, and master the principles that make SQL a powerful and flexible tool for handling data and logic. Dive deep into SQL and database design, and learn to recognize the most common missteps made by software developers in database modeling, SQL query logic, and code design of data-driven applications. See practical examples of misconceptions about SQL that can lure software projects astray. Find the greatest value in each group of data. Understand why an intersection table may be your new best friend. Store passwords securely and don't reinvent the wheel. Handle NULL values like a pro. Defend your web applications against the security weakness of SQL injection. Use SQL the right way - it can save you from headaches and needless work, and let your application really shine! What You Need: The SQL examples use the MySQL 8.0 flavor, but other popular brands of RDBMS are mentioned. Other code examples use Python 3.9+ or Ruby 2.7+.

API Design for C++ provides a comprehensive discussion of Application Programming Interface (API) development, from initial design through implementation, testing, documentation, release, versioning, maintenance, and deprecation. It is the only book that teaches the strategies of C++ API development, including interface design, versioning, scripting, and plug-in extensibility. Drawing from the author's experience on large scale, collaborative software projects, the text offers practical techniques of API design that produce robust code for the long term. It presents patterns and practices that provide real value to individual developers as well as organizations. API Design for C++ explores often overlooked issues, both technical and non-technical, contributing to successful design decisions that product high quality, robust, and long-lived APIs. It focuses on various API styles and patterns that will allow you to produce elegant and durable libraries. A discussion on testing strategies concentrates on automated API testing techniques rather than attempting to include end-user application testing techniques such as GUI testing, system testing, or manual testing. Each concept is illustrated with extensive C++ code examples, and fully functional examples and working source code for experimentation are available online. This book will be helpful to new programmers who understand the fundamentals of C++ and who want to advance their design skills, as well as to senior engineers and software architects seeking to gain new expertise to complement their existing talents. Three specific groups of readers are targeted: practicing software engineers and architects, technical managers, and students and educators. The only book that teaches the strategies of C++ API development, including design, versioning, documentation, testing, scripting, and extensibility. Extensive code examples illustrate each concept, with fully functional examples and working source code for experimentation available online. Covers various API styles and patterns with a focus on practical and efficient designs for large-scale long-term projects.

A Hands-on Guide to Relational Database Design

Proceedings of 11th Computer Science On-line Conference 2022, Vol. 1

An Accelerated Introduction to SQL Basics

A guide to developing efficient and elegant T-SQL code

SQL Antipatterns, Volume 1

SQL Primer

The demand for SQL information and training continues to grow with the need for a database behind every website capable of offering web-based information queries. SQL is the de facto language for database retrieval, and if you need to access, update, or utilize data in a modern database management system, you will need SQL to do it. The Second Edition of Joe Celko's Trees and Hierarchies in SQL for Smarties covers two new sets of extensions over three entirely new chapters and expounds upon the changes that have occurred in SQL standards since the first edition's publication. Benefit from mastering the challenging aspects of these database applications in SQL as taught by Joe Celko, one of the most-read SQL authors in the world. Expert advice from a SQL authority and award-winning columnist who has given 10 years of service to the ANSI SQL standards committee Teaches scores of advanced techniques that can be used with any SQL environment Offers graph theory and programming techniques for working around deficiencies and gives insight into real-world challenges

Six-Step Relational Database Design™ bridges the gaps between database theory, database modeling, and database implementation by outlining a simple but reliable six-step process for modeling user data on a Crow's Foot Relational Model Diagram, and then demonstrating how to implement this model on any relational database management system. The second edition is a new chapter on implementation that goes through the steps necessary to implement each of the case studies on a relational database management system, clearly relating the design and database theory. In addition, questions are also included at the end of each of the six steps and one of the previous case studies has been replaced, making the case study selection more relevant. Six-Step Relational Database Design™ uses three case studies and starts with a statement of the problem by the client and then goes through the six steps necessary to create a data model of the client's business requirements. This model can then be used to implement the database on any relational database management system. Six-Step Relational Database Design™ should be used as a handbook for students and professionals in the software-development field. The technique described in this book can be used by students for quickly developing databases for their applications, and by professionals for developing sturdy, reliable, and accurate relational database models for their software applications.

What others in the trenches say about The Pragmatic Programmer... "The cool thing about this book is that it's great for keeping the programming process fresh. The book helps you grow and clearly comes from people who have been there." —Kent Beck, author of Extreme Programming Explained: Embrace Change "I found this book to be a great mix of solid advice and wonderful analogies!" —Martin Fowler, author of Refactoring and UML Distilled "I would buy a copy, read it twice, then tell all my colleagues to run out and grab a copy. This is a book I will never loan because I would worry about it being lost." —Kevin Ruland, Management Science, MSG-Logistics "The wisdom and practical experience of the authors is obvious. The topics are relevant and useful.... By far its greatest strength for me has been the outstanding analogies—tracer bullets, broken windows, and the fabulous helicopter-based explanation of the concept of orthogonality, especially in a crisis situation. I have little doubt that this book will eventually become an excellent source of useful information for journeymen programmers and experienced programmers alike." —John Lakos, author of Large-Scale C++ Software Design "This is the sort of book I will buy a dozen copies of when it comes out so I can give it to my clients." —Eric Vought, Senior Software Engineer "Most modern books on software development fail to cover the basics of what makes a great software developer, instead spending their time on syntax or technology which has the greatest leverage possible for any software team is in having talented developers who really know their craft well. An excellent book." —Pete McBreen, Independent Consultant "Since I bought this book, I have implemented many of the practical suggestions and tips it contains. Across the board, they have saved my company time and money while helping me get my job done and I think it should be a desktop reference for everyone who works with code for a living." —Jared Richardson, Senior Software Developer, iRenaissance, Inc. "I would like to see this issued to every employee at my company...." —Chris Cleeland, Senior Software Engineer, Object Computing, Inc. "If I'm putting together a project, it's the authors of this book that I want. . . . And far from settling for people who've read their book." —Ward Cunningham Straight from the programming trenches, The Pragmatic Programmer cuts through the increasing specialization and technical complexity of modern software development to examine the core process--taking a requirement and producing working, maintainable code that delights its users. It covers topics ranging from personal responsibility and career development to architectural techniques for keeping your code flexible and easy to adapt and reuse. Read this book, and you'll learn how to Fight software development's trap of duplicating knowledge; Write flexible, dynamic, and adaptable code; Avoid programming by coincidence; Bullet-proof your code with contracts, assertions, and exceptions; Capture requirements; Test ruthlessly and effectively; Delight your users; Build teams of pragmatic programmers; and Make your developments more precise with automation. Written as a series of contained sections and filled with entertaining anecdotes, thoughtful examples, and interesting analogies, The Pragmatic Programmer illustrates the best practices and major pitfalls of software development from different aspects of software development. Whether you're a new coder, an experienced programmer, or a manager responsible for software projects, use these lessons daily, and you'll see improvements in personal productivity, accuracy, and job satisfaction. You'll learn skills and develop habits and attitudes that form the foundation for long-term success in your career and become a Pragmatic Programmer.

Explore the world of .NET design patterns and bring the benefits that the right patterns can offer to your toolkit today About This Book Dive into the powerful fundamentals of .NET for software development The code is explained piece by piece and the application of the pattern is also showcased. This fast-paced guide shows you how to implement the patterns in your applications Who This Book Is For This book is for those with familiarity with .NET development who would like to take their skills to the next level and be in the driver's seat when using modern development techniques. Basic object-oriented C# programming experience and an elementary familiarity with the .NET framework library is required. What You Will Learn Put design patterns and pattern catalogs into the right perspective Apply patterns for software development under C#/.NET Use GoF and other patterns in real-life development scenarios Be able to explain design patterns in your vocabulary and well articulate your design thoughts Leverage object/functional programming by mixing OOP and FP Understand the reactive programming model using Rx and RxJs Write composable code using C# LINQ constructs Be able to implement concurrent/parallel programming techniques using idioms under .NET Avoiding pitfalls when creating composable and maintainable code using imperative, functional, and reactive code. In Detail Knowing about design patterns enables developers to improve their code base, promoting code reuse and making their design more robust. This book focuses on the practical aspects of programming in .NET. You will learn about some of the relevant design patterns (and their application) that are commonly used. We start with classic object-oriented programming (OOP) techniques, evaluate parallel programming and concurrency models, enhance implementations by mixing OOP and functional programming, and finally to the reactive programming model where functional programming and OOP are used in synergy to write better code. Throughout this book, we'll show you how to work with architecture/design techniques, GoF patterns, relevant patterns from other catalogs, functional programming, and reactive programming techniques. After reading this book, you'll be able to convincingly leverage these design patterns (factory pattern, builder pattern, prototype pattern, adapter pattern, facade pattern, decorator pattern, observer pattern and so on) for your applications. You will also be able to write fluid functional code in .NET that would leverage concurrency and parallelism! Style and approach This tutorial-based book takes a step-by-step approach to each major pattern and explains them in a detailed manner along with code examples.

Six-Step Relational Database Design
SQL Design Patterns

.NET Design Patterns

Proceedings of 10th Computer Science On-line Conference 2021, Vol. 1

Foundations of Software and System Performance Engineering

How to Write Accurate SQL Code

Clean Code

SQL Antipatterns Avoiding the Pitfalls of Database Programming

Virtualization, cloud, containers, server automation, and software-defined networking are meant to simplify IT operations. But many organizations adopting these technologies have found that it only leads to a faster-growing sprawl of unmanageable systems. This is where infrastructure as code can help. With this practical guide, author Kief Morris of ThoughtWorks shows you how to effectively use principles, practices, and patterns pioneered through the DevOps movement to manage cloud age infrastructure. Ideal for system administrators, infrastructure engineers, team leads, and architects, this book demonstrates various tools, techniques, and patterns you can use to implement infrastructure as code. In three parts, you'll learn about the platforms and tooling involved in creating and configuring infrastructure elements, patterns for using these tools, and practices for making infrastructure as code work in your environment. Examine the pitfalls that organizations fall into when adopting the new generation of infrastructure technologies Understand the capabilities and service models of dynamic infrastructure platforms Learn about tools that provide, provision, and configure core infrastructure resources Explore services and tools for managing a dynamic infrastructure Learn specific patterns and practices for provisioning servers, building server templates, and updating running servers

Troubleshoot query performance issues, identify anti-patterns in code, and write efficient T-SQL queries Key Features Discover T-SQL functionalities and services that help you interact with relational databases Understand the roles, tasks and responsibilities of a T-SQL developer Explore solutions for carrying out database querying tasks, database administration, and troubleshooting Book Description Transact-SQL (T-SQL) is Microsoft's proprietary extension to the SQL language that is used with Microsoft SQL Server and Azure SQL Database. This book will be a useful guide to learning the art of writing efficient T-SQL code in modern SQL Server versions, as well as the Azure SQL Database. The book will get you started with query processing fundamentals to help you write powerful, performant T-SQL queries. You will then focus on query execution plans and learn how to leverage them for troubleshooting. In the later chapters, you will learn how to identify various T-SQL patterns and anti-patterns. This will help you analyze execution plans to gain insights into current performance, and determine whether or not a query is scalable. You will also learn to build diagnostic queries using dynamic management views (DMVs) and dynamic management functions (DMFs) to address various challenges in T-SQL execution. Next, you will study how to leverage the built-in tools of SQL Server to shorten the time taken to address query performance and scalability issues. In the concluding chapters, the book will guide you through implementing various features, such as Extended Events, Query Store, and Query Tuning Assistant using hands-on examples. By the end of this book, you will have the skills to determine query performance bottlenecks, avoid pitfalls, and discover the anti-patterns in use. Foreword by Conor Cunningham, Partner Architect – SQL Server and Azure SQL – Microsoft What you will learn Use Query Store to understand and easily change query performance Recognize and eliminate bottlenecks that lead to slow performance Deploy quick fixes and long-term solutions to improve query performance Implement best practices to minimize performance risk using T-SQL Achieve optimal performance by ensuring careful query and index design Use the latest performance optimization features in SQL Server 2017 and SQL Server 2019 Protect query performance during upgrades to newer versions of SQL Server Who this book is for This book is for database administrators, database developers, data analysts, data scientists, and T-SQL practitioners who want to get started with writing T-SQL code and troubleshooting query performance issues, through the help of practical examples. Previous knowledge of T-SQL querying is not required to get started on this book.

Looks at the principles and clean code, includes case studies showcasing the practices of writing clean code, and contains a list of heuristics and "smells" accumulated from the process of writing clean code.

Joe Celko's Trees and Hierarchies in SQL for Smarties

A Hands-on Guide to Data Manipulation in SQL

A Deep Dive into How Distributed Data Systems Work

T-SQL Querying

SQL and Relational Theory

Managing Servers in the Cloud

Query and manipulate databases from popular relational database servers using SQL

Learn effective and scalable database design techniques in SQL Server 2019 and other recent SQL Server versions. This book is revised to cover additions to SQL Server that include SQL graph enhancements, in-memory online transaction processing, temporal data storage, row-level security, and other design-related features. This book will help you design OLTP databases that are high-quality, protect the integrity of your data, and perform fast on-premises, in the cloud, or in hybrid configurations. Designing an effective and scalable database using SQL Server is a task requiring skills that have been around for well over 30 years, using technology that is constantly changing. This book covers everything from design logic that business users will understand to the physical implementation of design in a SQL Server database. Grounded in best practices and a solid understanding of the underlying theory, author Louis Davidson shows you how to "get it right" in SQL Server database design and lay a solid groundwork for the future use of valuable business data. What You Will Learn Develop conceptual models of client data using interviews and client documentation Implement designs that work on premises, in the cloud, or in a hybrid approach Recognize and apply common database design patterns Normalize data models to enhance integrity and scalability of your databases for the long-term use of valuable data Translate conceptual models into high-performing SQL Server databases Secure and protect data integrity as part of meeting regulatory requirements Create effective indexing to speed query performance Understand the concepts of concurrency Who This Book Is For Programmers and database administrators of all types who want to use SQL Server to store transactional data. The book is especially useful to those wanting to learn the latest database design features in SQL Server 2019 (features that include graph objects, in-memory OLTP, temporal data support, and more). Chapters on fundamental concepts, the language of database modeling, SQL implementation, and the normalization process lay a solid groundwork for readers who are just entering the field of database design. More advanced chapters serve the seasoned veteran by tackling the latest in physical implementation features that SQL Server has to offer. The book has been carefully revised to cover all the design-related features that are new in SQL Server 2019.

Illustrating some of the most common misconceptions and pitfalls software developers face using relational databases, this book helps readers use a database to produce the most efficient

results, and turn sluggish, inflexible code into high-quality, reliable solutions.

Are you an SQL programmer that, like many, came to SQL after learning and writing procedural or object-oriented code? Or have switched jobs to where a different brand of SQL is being used, or maybe even been told to learn SQL yourself? If even one answer is yes, then you need this book. A "Manual of Style" for the SQL programmer, this book is a collection of heuristics and rules, tips, and tricks that will help you improve SQL programming style and proficiency, and for formatting and writing portable, readable, maintainable SQL code. Based on many years of experience consulting in SQL shops, and gathering questions and resolving his students' SQL style issues, Joe Celko can help you become an even better SQL programmer. Help you write Standard SQL without an accent or a dialect that is used in another programming language or a specific flavor of SQL, code that can be maintained and used by other people. Enable you to give your group a coding standard for internal use, to enable programmers to use a consistent style. Give you the mental tools to approach a new problem with SQL as your tool, rather than another programming language — one that someone else might not know!

Leading a fast-growing team is a uniquely challenging experience. Startups with a hot product often double or triple in size quickly—a recipe for chaos if company leaders aren't prepared for the pitfalls of hyper-growth. If you're leading a startup or a new team between 10 and 150 people, this guide provides a practical approach to managing your way through these challenges. Each section covers essential strategies and tactics for managing growth, starting with a single team and exploring typical scaling points as the team grows in size and complexity. The book also provides many examples and lessons learned, based on the authors' experience and interviews with industry leaders. Learn how to make the most of: Hiring: Learn a scalable hiring process for growing your team People management: Use 1-on-1 mentorship, dispute resolution, and other techniques to ensure your team is happy and productive Organization: Motivate employees by applying five organizational design principles Culture: Build a culture that can evolve as you grow, while remaining connected to the team's core values Communication: Ensure that important information—and only the important stuff—gets through

Design and Deploy Production-Ready Software

61 Specific Ways to Write Better SQL

Good Math

Beginning Database Design Solutions

SQL Tuning

Everything Developers Need to Know about SQL Performance

Avoiding the Pitfalls of Database Programming

SQL is full of difficulties and traps for the unwary. You can avoid them if you understand relational theory, but only if you know how to put the theory into practice. In this insightful book, author C.J. Date explains relational theory in depth, and demonstrates through numerous examples and exercises how you can apply it directly to your use of SQL. This second edition includes new material on recursive queries, "missing information" without nulls, new update operators, and topics such as aggregate operators, grouping and ungrouping, and view updating. If you have a modest-to-advanced background in SQL, you'll learn how to deal with a host of common SQL dilemmas. Why is proper column naming so important? Nulls in your database are causing you to get wrong answers. Why? What can you do about it? Is it possible to write an SQL query to find employees who have never been in the same department for more than six months at a time? SQL supports "quantified comparisons," but they're better avoided. Why? How do you avoid them? Constraints are crucially important, but most SQL products don't support them properly. What can you do to resolve this situation? Database theory and practice have evolved since the relational model was developed more than 40 years ago. SQL and Relational Theory draws on decades of research to present the most up-to-date treatment of SQL available. C.J. Date has a stature that is unique within the database industry. A prolific writer well known for the bestselling textbook An Introduction to Database Systems (Addison-Wesley), he has an exceptionally clear style when writing about complex principles and theory.

Learn everything you need to know to build efficient SQL queries using this easy-to-follow beginner's guide Key FeaturesExplore all SQL statements in depth using a variety of examplesGet to grips with database querying, data aggregate, manipulation, and much moreUnderstand how to explore and process data of varying complexity to tell a storyBook Description SQL is a powerful querying language that's used to store, manipulate, and retrieve data, and it is one of the most popular languages used by developers to query and analyze data efficiently. If you're looking for a comprehensive introduction to SQL, Learn SQL Database Programming will help you to get up to speed with using SQL to streamline your work in no time. Starting with an overview of relational database management systems, this book will show you how to set up and use MySQL Workbench and design a database using practical examples. You'll also discover how to query and manipulate data with SQL programming using MySQL Workbench. As you advance, you'll create a database, query single and multiple tables, and modify data using SQL querying. This SQL book covers advanced SQL techniques, including aggregate functions, flow control statements, error handling, and subqueries, and helps you process your data to present your findings. Finally, you'll implement best practices for writing SQL and designing indexes and tables. By the end of this SQL

programming book, you'll have gained the confidence to use SQL queries to retrieve and manipulate data. What you will learn: Install, configure, and use MySQL Workbench to restore a database; Explore different data types such as string, numeric, and date and time; Query a single table using the basic SQL SELECT statement and the FROM, WHERE, and ORDER BY clauses; Query multiple tables by understanding various types of table relationships; Modify data in tables using the INSERT, UPDATE, and DELETE statements; Use aggregate functions to group and summarize data; Detect bad data, duplicates, and irrelevant values while processing data. Who this book is for: This book is for business analysts, SQL developers, database administrators, and students learning SQL. If you want to learn how to query and manipulate SQL data for database administration tasks or simply extract and organize relevant data for analysis, you'll find this book useful. No prior SQL experience is required.

"If this book had been available to Healthcare.gov's contractors, and they read and followed its life cycle performance processes, there would not have been the enormous problems apparent in that application. In my 40+ years of experience in building leading-edge products, poor performance is the single most frequent cause of the failure or cancellation of software-intensive projects. This book provides techniques and skills necessary to implement performance engineering at the beginning of a project and manage it throughout the product's life cycle. I cannot recommend it highly enough." – Don Shafer, CSDP, Technical Fellow, Athens Group, LLC

Poor performance is a frequent cause of software project failure. Performance engineering can be extremely challenging. In *Foundations of Software and System Performance Engineering*, leading software performance expert Dr. André Bondi helps you create effective performance requirements up front, and then architect, develop, test, and deliver systems that meet them. Drawing on many years of experience at Siemens, AT&T Labs, Bell Laboratories, and two startups, Bondi offers practical guidance for every software stakeholder and development team participant. He shows you how to define and use metrics; plan for diverse workloads; evaluate scalability, capacity, and responsiveness; and test both individual components and entire systems. Throughout, Bondi helps you link performance engineering with everything else you do in the software life cycle, so you can achieve the right performance—now and in the future—at lower cost and with less pain. This guide will help you

- Mitigate the business and engineering risk associated with poor system performance
- Specify system performance requirements in business and engineering terms
- Identify metrics for comparing performance requirements with actual performance
- Verify the accuracy of measurements
- Use simple mathematical models to make predictions, plan performance tests, and anticipate the impact of changes to the system or the load placed upon it
- Avoid common performance and scalability mistakes
- Clarify business and engineering needs to be satisfied by given levels of throughput and response time
- Incorporate performance engineering into agile processes
- Help stakeholders of a system make better performance-related decisions
- Manage stakeholders' expectations about system performance throughout the software life cycle, and deliver a software product with quality performance

André B. Bondi is a senior staff engineer at Siemens Corp., Corporate Technologies in Princeton, New Jersey. His specialties include performance requirements, performance analysis, modeling, simulation, and testing. Bondi has applied his industrial and academic experience to the solution of performance issues in many problem domains. In addition to holding a doctorate in computer science and a master's in statistics, he is a Certified Scrum Master.

Data is getting bigger and more complex by the day, and so are your choices in handling it. Explore some of the most cutting-edge databases available - from a traditional relational database to newer NoSQL approaches - and make informed decisions about challenging data storage problems. This is the only comprehensive guide to the world of NoSQL databases, with in-depth practical and conceptual introductions to seven different technologies: Redis, Neo4J, CouchDB, MongoDB, HBase, Postgres, and DynamoDB. This second edition includes a new chapter on DynamoDB and updated content for each chapter. While relational databases such as MySQL remain as relevant as ever, the alternative, NoSQL paradigm has opened up new horizons in performance and scalability and changed the way we approach data-centric problems. This book presents the essential concepts behind each database alongside hands-on examples that make each technology come alive. With each database, tackle a real-world problem that highlights the concepts and features that make it shine. Along the way, explore five database models - relational, key/value, columnar, document, and graph - from the perspective of challenges faced by real applications. Learn how MongoDB and CouchDB are strikingly different, make your applications faster with Redis and more connected with Neo4J, build a cluster of HBase servers using cloud services such as

*Amazon's Elastic MapReduce, and more. This new edition brings a brand new chapter on DynamoDB, updated code samples and exercises, and a more up-to-date account of each database's feature set. Whether you're a programmer building the next big thing, a data scientist seeking solutions to thorny problems, or a technology enthusiast venturing into new territory, you will find something to inspire you in this book. What You Need: You'll need a *nix shell (Mac OS or Linux preferred, Windows users will need Cygwin), Java 6 (or greater), and Ruby 1.8.7 (or greater). Each chapter will list the downloads required for that database.*

Effective SQL

A Guide to Modern Databases and the NoSQL Movement

Recent Advances in Information and Communication Technology 2015

SQL Performance Explained

Database Internals

Release It!

SQL Antipatterns

A single dramatic software failure can cost a company millions of dollars - but can be avoided with simple changes to design and architecture. This new edition of the best-selling industry standard shows you how to create systems that run longer, with fewer failures, and recover better when bad things happen. New coverage includes DevOps, microservices, and cloud-native architecture. Stability antipatterns have grown to include systemic problems in large-scale systems. This is a must-have pragmatic guide to engineering for production systems. If you're a software developer, and you don't want to get alerts every night for the rest of your life, help is here. With a combination of case studies about huge losses - lost revenue, lost reputation, lost time, lost opportunity - and practical, down-to-earth advice that was all gained through painful experience, this book helps you avoid the pitfalls that cost companies millions of dollars in downtime and reputation. Eighty percent of project life-cycle cost is in production, yet few books address this topic. This updated edition deals with the production of today's systems - larger, more complex, and heavily virtualized - and includes information on chaos engineering, the discipline of applying randomness and deliberate stress to reveal systematic problems. Build systems that survive the real world, avoid downtime, implement zero-downtime upgrades and continuous delivery, and make cloud-native applications resilient. Examine ways to architect, design, and build software - particularly distributed systems - that stands up to the typhoon winds of a flash mob, a Slashdotting, or a link on Reddit. Take a hard look at software that failed the test and find ways to make sure your software survives. To skip the pain and get the experience...get this book.

*For all the buzz about trendy IT techniques, data processing is still at the core of our systems, especially now that enterprises all over the world are confronted with exploding volumes of data. Database performance has become a major headache, and most IT departments believe that developers should provide simple SQL code to solve immediate problems and let DBAs tune any bad SQL later. In *The Art of SQL*, author and SQL expert Stephane Faroult argues that this safe approach only leads to disaster. His insightful book, named after *Art of War* by Sun Tzu, contends that writing quick inefficient code is sweeping the dirt under the rug. SQL code may run for 5 to 10 years, surviving several major releases of the database management system and on several generations of hardware. The code must be fast and sound from the start, and that requires a firm understanding of SQL and relational theory. *The Art of SQL* offers best practices that teach experienced SQL users to focus on strategy rather than specifics. Faroult's approach takes a page from Sun Tzu's classic treatise by viewing database design as a military campaign. You need knowledge, skills, and talent. Talent can't be taught, but every strategist from Sun Tzu to modern-day generals believed that it can be nurtured through the experience of others. They passed on their experience acquired in the field through basic principles that served as guiding stars amid the sound and fury of battle. This is what Faroult does with SQL. Like a successful battle plan, good architectural choices are based on contingencies. What if the volume of this or that table increases unexpectedly? What if, following a merger, the number of users doubles? What if you want to keep several years of data online? Faroult's way of looking at SQL performance may be unconventional and unique, but he's deadly serious about writing good SQL and using SQL well. *The Art of SQL* is not a cookbook, listing problems and giving recipes. The aim is to get you-and your manager-to raise good questions.*

Provides information on the architecture of the T-SQL programming language to create scalable code.

*A poorly performing database application not only costs users time, but also has an impact on other applications running on the same computer or the same network. *SQL Tuning* provides an essential next step for SQL developers and database administrators who want to extend their SQL tuning expertise and get the most from their database applications. There are two basic issues to focus on when tuning SQL: how to find and interpret the execution plan of an SQL statement and how to change SQL to get a specific alternate execution plan. *SQL Tuning* provides answers to these questions and addresses a third issue that's even more important: how to find the optimal execution plan for the query to use. Author Dan Tow outlines a timesaving method he's developed for finding the optimum execution plan--rapidly and systematically--regardless of the complexity of the SQL or the database platform being used. You'll learn how to understand and control SQL execution plans and how to diagram SQL queries to deduce the best execution plan for a query. Key chapters in the book include exercises to reinforce the concepts you've learned. *SQL Tuning* concludes by addressing special concerns and unique solutions to*

"unsolvable problems." Whether you are a programmer who develops SQL-based applications or a database administrator or other who troubleshoots poorly tuned applications, SQL Tuning will arm you with a reliable and deterministic method for tuning your SQL queries to gain optimal performance.

Software Engineering and Algorithms

Joe Celko's SQL Programming Style

Pragmatic Unit Testing in Java 8 with JUnit

The Expert Guide to SQL Programming

Learn SQL Database Programming

The Pragmatic Programmer

Pro SQL Server Relational Database Design and Implementation

The vast majority of software applications use relational databases that virtually every application developer must work with. This book introduces you to database design, whether you're a DBA or database developer. You'll discover what databases are, their goals, and why proper design is necessary to achieve those goals. Additionally, you'll master how to structure the database so it gives good performance while minimizing the chance for error. You will learn how to decide what should be in a database to meet the application's requirements.

Mathematics is beautiful--and it can be fun and exciting as well as practical. Good Math is your guide to some of the most intriguing topics from two thousand years of mathematics: from Egyptian fractions to Turing machines; from the real meaning of numbers to proof trees, group symmetry, and mechanical computation. If you've ever wondered what lay beyond the proofs you struggled to complete in high school geometry, or what limits the capabilities of computer on your desk, this is the book for you. Why do Roman numerals persist? How do we know that some infinities are larger than others? And how can we know for certain a program will ever finish? In this fast-paced tour of modern and not-so-modern math, computer scientist Mark Chu-Carroll explores some of the greatest breakthroughs and disappointments of more than two thousand years of mathematical thought. There is joy and beauty in mathematics, and in more than two dozen essays drawn from his popular "Good Math" blog, you'll find concepts, proofs, and examples that are often surprising, counterintuitive, or just plain weird. Mark begins his journey with the basics of numbers, with an entertaining trip through the integers and the natural, rational, irrational, and transcendental numbers. The voyage continues with a look at some of the oddest numbers in mathematics, including zero, the golden ratio, imaginary numbers, Roman numerals, and Egyptian and continuing fractions. After a deep dive into modern logic, including an introduction to linear logic and the logic-savvy Prolog language, the trip concludes with a tour of modern set theory and the advances and paradoxes of modern mechanical computing. If your high school or college math courses left you grasping for the inner meaning behind the numbers, Mark's book will both entertain and enlighten you.

Presents a guide to writing effective SQL queries, from simple data selection and filtering to joining multiple tables and modifying sets of data, with information on how to solve a variety of challenging SQL problems.

The Pragmatic Programmers classic is back! Freshly updated for modern software development, Pragmatic Unit Testing in Java 8 With JUnit teaches you how to write and run easily maintained unit tests in JUnit with confidence. You'll learn mnemonics to help you know what tests to write, how to remember all the boundary conditions, and what the qualities of a good test are. You'll see how unit tests can pay off by allowing you to keep your system code clean, and you'll learn how to handle the stuff that seems too tough to test. Pragmatic Unit Testing in Java 8 With JUnit steps you through all the important unit testing topics. If you've never written a unit test, you'll see screen shots from Eclipse, IntelliJ IDEA, and NetBeans that will help you get past the hard part--getting set up and started. Once past the basics, you'll learn why you want to write unit tests and how to effectively use JUnit. But the meaty part of the book is its collected unit testing wisdom from people who've been there, done that on production systems for at least 15 years: veteran author and developer Jeff Langr, building on the wisdom of Pragmatic Programmers Andy Hunt and Dave Thomas. You'll learn: How to craft your unit tests to minimize your effort in maintaining them. How to use unit tests to help keep your system clean. How to test the tough stuff. Memorable mnemonics to help you remember what's important when writing unit tests. How to help your team reap and sustain the benefits of unit testing. You won't just learn about unit testing in theory--you'll work through numerous code examples. When it comes to programming, hands-on is the only way to learn!

Software Engineering Perspectives in Systems

Rails AntiPatterns

Process, Performance Modeling, Requirements, Testing, Scalability, and Practice

A practical guide to revealing anti-patterns and architectural pitfalls to avoid microservices fallacies