
Sample Database Design Document

This is likewise one of the factors by obtaining the soft documents of this **Sample Database Design Document** by online. You might not require more become old to spend to go to the ebook opening as without difficulty as search for them. In some cases, you likewise realize not discover the pronouncement Sample Database Design Document that you are looking for. It will certainly squander the time.

However below, like you visit this web page, it will be appropriately categorically easy to get as well as download lead Sample Database Design Document

It will not consent many times as we tell before. You can realize it while play a role something else at home and even in your workplace. appropriately easy! So, are you question? Just exercise just what we meet the expense of under as skillfully as evaluation **Sample Database Design Document** what you later to read!



PC Learning Labs Teaches Lotus Notes Release 4 CRC Press

"Riordan covers core skills for any developer--database design and development--in a perfect amount of detail.

This book should be on every professional developer's reading list." --Duncan Mackenzie, developer, Microsoft (MSDN)"Designing a database is not a trivial subject. Riordan brings experience and clear explanations to a fundamental part of software development." --Patrick Birch, database and technical writing consultant"If you buy only one book on database design, make it this one. Riordan has a talent for

explaining technical issues in simple language, without over simplifying." --Brendan Reynolds, developer, Dataset IT Systems and Microsoft Access MVP"A book that will expertly guide you in how to develop a database for a client-- and how to do it right the first time!" --Kenneth D. Snell, Ph.D., ACCESS developer and Microsoft Access MVP "Riordan has produced a unique book that brings together a formal, yet commonsense,

approach to relational database design...and then goes further! Many database designers will find immense value in the steps to developing practical data warehouse designs. If you are seeking a framework for designing transactional databases, or want to step out into the world of analytical databases, Riordan's book excels at bridging both worlds." --Paul Irvine, vice president, engineering, Via Training"Riordan takes a complex subject and makes it easy. If you're over your head on a database design project, this book will help bail you out!" --Mike Gunderloy, contributing editor, Application Development Trends "This book covers a wide range of database design and data modeling topics in a well-

organized, easy to understand format." --Amy Sticksel, Sticksel Data Systems, Inc."In Designing Effective Database Systems, Riordan's style, wit, and attention to detail are outstanding." --Sandra Daigle, Microsoft Access MVP The Software Developer's Step-by-Step Guide to Database Design World-renowned expert Rebecca M. Riordan has written the definitive database design book for working developers who aren't database experts. No matter how messy or complex your data challenge, Designing Effective Database Systems shows you how to design an effective, high-performance database to solve it. Riordan begins by thoroughly demystifying the principles of relational design, making them accessible to every professional developer. Next,

she offers the field's clearest introduction to dimensional database modeling--practical insight for designing today's increasingly important analytical applications. One task at a time, the author illuminates every facet of database analysis and design for both traditional databases and the dimensional databases used for data warehousing, showing how to avoid common architectural pitfalls that complicate development and reduce extensibility. The book concludes with comprehensive, expert guidance on designing databases for maximum usability. This book will teach you to Understand relational database models, structures, relationships, and data integrity principles Define database system goals, criteria, scope, and work

processes Construct accurate conceptual models: relationships, entities, domain analysis, and normalization Build efficient, secure database schema Master the elements of online analytical processing (OLAP) design: fact tables, dimension tables, snowflaking, and more Architect and construct easy, efficient interfaces for querying and reporting Learn from practice examples based on Microsoft's Northwind sample database Riordan has helped thousands of professionals master database design and development, earning Microsoft's coveted MVP honor for her exceptional contributions. Nobody is more qualified to help you master database design and apply it in your real-world environment.

High-Performance Web Databases "O'Reilly

Media, Inc."

Fully revised and updated, Relational Database Design, Second Edition is the most lucid and effective introduction to relational database design available. Here, you'll find the conceptual and practical information you need to develop a design that ensures data accuracy and user satisfaction while optimizing performance, regardless of your experience level or choice of DBMS. Supporting the book's step-by-step instruction are three case studies illustrating the planning, analysis, and design steps involved in arriving at a sound design. These real-world examples include object-relational design techniques, which are addressed in greater detail in a new chapter devoted entirely to this timely subject. * Concepts you need to master to put the book's practical instruction to work. * Methods for tailoring your design to the environment in which the database will run and the uses to which it will be put. * Design approaches that ensure data accuracy and consistency. * Examples of how design can inhibit or boost database application performance. * Object-relational design techniques, benefits, and examples. * Instructions on how to choose and use a

normalization technique. * Guidelines for understanding and applying Codd's rules. * Tools to implement a relational design using SQL. * Techniques for using CASE tools for database design.

Relational Database Design

Springer Science & Business Media Fully updated and expanded from the previous edition, A Practical Guide to Database Design, Second Edition is intended for those involved in the design or development of a database system or application. It begins by illustrating how to develop a Third Normal Form data model where data is placed "where it belongs". The reader is taken step-by-step through the Normalization process, first using a simple then a more complex set of data requirements. Next, usage analysis for each Logical Data Model is reviewed and a Physical Data Model is produced that will satisfy user performance requirements. Finally, each Physical Data Model is used as input to create databases using

both Microsoft Access and SQL Server. The book next shows how to use an industry-leading data modeling tool to define and manage logical and physical data models, and how to create Data Definition Language statements to create or update a database running in SQL Server, Oracle, or other type of DBMS. One chapter is devoted to illustrating how Microsoft Access can be used to create user interfaces to review and update underlying tables in that database as well as tables residing in SQL Server or Oracle. For users involved with Cyber activity or support, one chapter illustrates how to extract records of interest from a log file using PERL, then shows how to load these extracted records into one or more SQL Server “tracking” tables adding status flags for analysts to use when reviewing activity of interest. These status flags are used to flag/mark collected records as “Reviewed”, “Pending” (currently being

analyzed) and “Resolved”. The last chapter then shows how to build a web-based GUI using PHP to query these tracking tables and allow an analyst to review new activity, flag items that need to be investigated, and finally flag items that have been investigated and resolved. Note that the book has complete code/scripts for both PERL and the PHP GUI. Semistructured Database Design Packt Publishing Ltd

The financial markets industry is at the same crossroads as the automotive industry in the late 1970s. Margins are collapsing and customization is rapidly increasing. The automotive industry turned to quality and its no coincidence that in the money management industry many of the spectacular failures have been due largely to problems in quality control. The financial industry is on the verge of a quality revolution. New and old firms alike are creating new investment vehicles and new strategies that are radically changing the nature of the industry. To compete, mutual funds, hedge fund industries, banks and proprietary trading firms are being

forced to quickly research, test and implement trade selection and execution systems. And, just as in the early stages of factory automation, quality suffers and leads to defects. Many financial firms fall short of quality, lacking processes and methodologies for proper development and evaluation of trading and investment systems. Authors Kumiega and Van Vliet present a new step-by-step methodology for such development. Their methodology (called K|V) has been presented in numerous journal articles and at academic and industry conferences and is rapidly being accepted as the preferred business process for the institutional trading and hedge fund industries for development, presentation, and evaluation of trading and investment systems. The K|V model for trading system development combines new product development, project management and software development methodologies into one robust system. After four stages, the methodology requires repeating the entire waterfall for continuous improvement. The discussion quality and its applications to the front office is presented using lessons learned by the authors after using the

methodology in the real world. As a result, it is flexible and modifiable to fit various projects in finance in different types of firms. Their methodology works equally well for short-term trading systems, longer-term portfolio management or mutual fund style investment strategies as well as more sophisticated ones employing derivative instruments in hedge funds. Additionally, readers will be able to quickly modify the standard K|V methodology to meet their unique needs and to quickly build other quantitatively drive applications for finance. At the beginning and the end of Quality Money Management the authors pose a key question: Are you willing to change and embrace quality for the 21st century or are willing to accept extinction? The real gem in this book is that the concepts give the reader a road map to avoid extinction. Presents a robust process engineering framework for developing and evaluating trading and investment systems Best practices along the step-by-step process will mitigate project risk, model risk, and ensure data quality Includes a quality model for backtesting and managing market risk of working systems

Guide on Logical Database Design Addison-Wesley Professional

CouchDB is a new breed of database for the Internet, geared to meet the needs of today's dynamic web applications. With this concise introduction, you'll learn how CouchDB's simple model for storing, processing, and accessing data makes it ideal for the type of data and rapid response users now demand from your applications--and how easy CouchDB is to set up, deploy, maintain, and scale. The code-packed examples in this book will help you learn how to work with documents, populate a simple database, replicate data from one database to another, and a host of other tasks. Install CouchDB on Linux, Mac OS X, Windows, or (if you must) from the source code Interact with data through CouchDB's RESTful API, and use standard HTTP operations, such as PUT, GET, POST, and DELETE Use Futon--CouchDB's web-based interface-- to manage databases and documents, and to configure replications Learn how to create, update, and delete documents in JSON format, and how to create and delete databases Work with design documents to get the formatting and indexing your application requires

Database Design CRC Press

Create database designs that scale, meet business requirements, and inherently work toward keeping your data structured and usable in the face of changing business models and software systems. This book is about database

design theory. Design theory is the scientific foundation for database design, just as the relational model is the scientific foundation for database technology in general. Databases lie at the heart of so much of what we do in the computing world that negative impacts of poor design can be extraordinarily widespread. This second edition includes greatly expanded coverage of exotic and little understood normal forms such as: essential tuple normal form (ETNF), redundancy free normal form (RFNF), superkey normal form (SKNF), sixth normal form (6NF), and domain key normal form (DKNF). Also included are new appendixes, including one that provides an in-depth look into the crucial notion of data consistency. Sequencing of topics has been improved, and many explanations and examples have been rewritten and clarified based upon the author's teaching of the content in instructor-led courses. This book aims to be different from other books on design by bridging the gap between the theory of design and the practice of design. The book explains theory in a way that practitioners should be able to understand, and it explains why that theory is of considerable practical importance. Reading this book provides you with an important theoretical grounding on which to do the practical work of database design. Reading the

book also helps you in going to and understanding the more academic texts as you build your base of knowledge and expertise. Anyone with a professional interest in database design can benefit from using this book as a stepping-stone toward a more rigorous design approach and more lasting database models.

What You Will Learn Understand what design theory is and is not Be aware of the two different goals of normalization Know which normal forms are truly significant Apply design theory in practice Be familiar with techniques for dealing with redundancy Understand what consistency is and why it is crucially important

Who This Book Is For Those having a professional interest in database design, including data and database administrators; educators and students specializing in database matters; information modelers and database designers; DBMS designers, implementers, and other database vendor personnel; and database consultants. The book is product independent.

Beginning Database Design Solutions
National Academies Press

This block is concerned with the database lifecycle, which describes the stages a database goes through, from the time the need for a database is established until it is withdrawn from use. This block applies the

practice developed in Block 3 to systematically develop, implement and maintain a database design that supports the information requirements of an enterprise. It presents a simple framework for database development and maintenance. This is a very practical block and will require you to write and execute SQL statements for which you will need access to a computer installed with the course software (order code M359/CDR01) and database cards

Scenarios and Hospital conceptual data model (order code M359/DBCARDS)
SQL Server 2012 Database Design
Complete Publishing

Semistructured Database Design provides an essential reference for anyone interested in the effective management of semistructured data. Since many new and advanced web applications consume a huge amount of such data, there is a growing need to properly design efficient databases. This volume responds to that need by describing a semantically rich data model for semistructured data, called Object-Relationship-Attribute model for Semistructured data (ORA-SS). Focusing on this new model, the book discuss

problems and present solutions for a number of topics, including schema extraction, the design of non-redundant storage organizations for semistructured data, and physical semistructured database design, among others. **Semistructured Database Design**, presents researchers and professionals with the most complete and up-to-date research in this fast-growing field.

Relational Database Design Clearly

Explained Createspace Independent

Publishing Platform

With about 200,000 entries, **StarBriefs Plus** represents the most comprehensive and accurately validated collection of abbreviations, acronyms, contractions and symbols within astronomy, related space sciences and other related fields. As such, this invaluable reference source (and its companion volume, **StarGuides Plus**) should be on the reference shelf of every library, organization or individual with any interest in these areas. Besides astronomy and associated space sciences, related fields such as aeronautics, aeronomy, astronautics, atmospheric sciences, chemistry, communications, computer sciences, data processing, education, electronics, engineering, energetics,

environment, geodesy, geophysics, information handling, management, mathematics, meteorology, optics, physics, remote sensing, and so on, are also covered when justified. Terms in common use and/or of general interest have also been included where appropriate.

[A Practical Guide to Database Design](#) Sams Publishing

A concise introduction to database design concepts, methods, and techniques in and out of the cloud In the newly revised second edition of *Beginning Database Design Solutions: Understanding and Implementing Database Design Concepts for the Cloud and Beyond*, Second Edition, award-winning programming instructor and mathematician Rod Stephens delivers an easy-to-understand guide to designing and implementing databases both in and out of the cloud. Without assuming any prior database design knowledge, the author walks you through the steps you'll need to take to understand, analyze, design, and build databases. In the book, you'll find clear coverage of foundational database concepts along with hands-on examples that help you practice important techniques so you can apply them to your own database designs, as well as: Downloadable source code that illustrates the

concepts discussed in the book *Best practices for reliable, platform-agnostic database design Strategies for digital transformation driven by universally accessible database design* An essential resource for database administrators, data management specialists, and database developers seeking expertise in relational, NoSQL, and hybrid database design both in and out of the cloud, *Beginning Database Design Solutions* is a hands-on guide ideal for students and practicing professionals alike.

[The Data and Analytics Playbook](#) Createspace Independent Publishing Platform

Beginning Database Design, Second Edition provides short, easy-to-read explanations of how to get database design right the first time. This book offers numerous examples to help you avoid the many pitfalls that entrap new and not-so-new database designers. Through the help of use cases and class diagrams modeled in the UML, you'll learn to discover and represent the details and scope of any design problem you choose to attack. Database design is not an exact science. Many are surprised to find that problems with their databases are caused by poor design rather than by difficulties in using

the database management software. *Beginning Database Design, Second Edition* helps you ask and answer important questions about your data so you can understand the problem you are trying to solve and create a pragmatic design capturing the essentials while leaving the door open for refinements and extension at a later stage. Solid database design principles and examples help demonstrate the consequences of simplifications and pragmatic decisions. The rationale is to try to keep a design simple, but allow room for development as situations change or resources permit. Provides solid design principles by which to avoid pitfalls and support changing needs Includes numerous examples of good and bad design decisions and their consequences Shows a modern method for documenting design using the Unified Modeling Language *Quality Money Management* Morgan Kaufmann A short guide for everyone on how to structure your data and set-up your MySQL database tables efficiently and easily. [Beginning Database Design Solutions](#) CRC Press This course has two teaching challenges - good

database design and a skills-development approach. This text contains worked examples along with projects that need manipulation of sample databases. Entity relationship diagrams are also included, as well as a software design tool, ER-Assistant.

Relational Database Design Clearly Explained CRC Press

PC Learning Labs presents a hands-on instruction book designed to help users of Lotus Notes 4.0 attain a high level of fluency in the shortest time possible. Throughout the book, theory is blended with practice by presenting new techniques and then applying them in hands-on activities. The disk offers sample Lotus Notes files.

z/OS: WebSphere Business Process Management V7 Production Topologies

Morgan Kaufmann

The rapidly increasing volume of information contained in relational databases places a strain on databases, performance, and maintainability: DBAs are under greater pressure than ever to optimize database structure for system performance and administration. Physical Database Design discusses the concept of how physical structures of databases affect performance, including specific examples, guidelines, and

best and worst practices for a variety of DBMSs and configurations. Something as simple as improving the table index design has a profound impact on performance. Every form of relational database, such as Online Transaction Processing (OLTP), Enterprise Resource Management (ERP), Data Mining (DM), or Management Resource Planning (MRP), can be improved using the methods provided in the book. The first complete treatment on physical database design, written by the authors of the seminal, Database Modeling and Design: Logical Design, Fourth Edition Includes an introduction to the major concepts of physical database design as well as detailed examples, using methodologies and tools most popular for relational databases today: Oracle, DB2 (IBM), and SQL Server (Microsoft) Focuses on physical database design for exploiting B+tree indexing, clustered indexes, multidimensional clustering (MDC), range partitioning, shared nothing partitioning, shared disk data placement, materialized views, bitmap indexes, automated design tools, and more!

Getting Started with CouchDB CUP Archive Fully revised and updated, Relational Database Design, Second Edition is the most lucid and effective introduction to relational database design available. Here, you'll find the conceptual and

practical information you need to develop a design that ensures data accuracy and user satisfaction while optimizing performance, regardless of your experience level or choice of DBMS. Supporting the book's step-by-step instruction are three case studies illustrating the planning, analysis, and design steps involved in arriving at a sound design. These real-world examples include object-relational design techniques, which are addressed in greater detail in a new chapter devoted entirely to this timely subject. * Concepts you need to master to put the book's practical instruction to work.* Methods for tailoring your design to the environment in which the database will run and the uses to which it will be put.* Design approaches that ensure data accuracy and consistency.* Examples of how design can inhibit or boost database application performance.* Object-relational design techniques, benefits, and examples.* Instructions on how to choose and use a normalization technique.* Guidelines for understanding and applying Codd's rules.* Tools to implement a relational design using SQL.* Techniques for using CASE tools for database design.

Data Analysis for Database Design Elsevier

Fully revised, updated, and expanded, Relational Database Design and Implementation, Third Edition is the most lucid and effective introduction to the subject available for IT/IS professionals

interested in honing their skills in database design, implementation, and administration. This book provides the conceptual and practical information necessary to develop a design and management scheme that ensures data accuracy and user satisfaction while optimizing performance, regardless of experience level or choice of DBMS. The book begins by reviewing basic concepts of databases and database design, then briefly reviews the SQL one would use to create databases. Topics such as the relational data model, normalization, data entities and Codd's Rules (and why they are important) are covered clearly and concisely but without resorting to "Dummies"-style talking down to the reader. Supporting the book's step-by-step instruction are three NEW case studies illustrating database planning, analysis, design, and management practices. In addition to these real-world examples, which include object-relational design techniques, an entirely NEW section consisting of three chapters is devoted to database implementation and management issues. * Principles needed to understand the basis of good relational database design and implementation practices. * Examples

to illustrate core concepts for enhanced comprehension and to put the book's practical instruction to work. * Methods for tailoring DB design to the environment in which the database will run and the uses to which it will be put. * Design approaches that ensure data accuracy and consistency. * Examples of how design can inhibit or boost database application performance. * Object-relational design techniques, benefits, and examples. * Instructions on how to choose and use a normalization technique. * Guidelines for understanding and applying Codd's rules. * Tools to implement a relational design using SQL. * Techniques for using CASE tools for database design.

StarBriefs Plus Lulu.com

Using client-proven methods and over 15 years of experience with the topic, Dr. Jones provides the reader with an understanding of the practical art of good database design, regardless of database type. Readers who have been confused by the high-end, academic style of current database design books will benefit from the approachable, tutorial style.

A Sane Approach to Database Design IBM

Redbooks

Relational Database Design and Implementation:

Clearly Explained, Fourth Edition, provides the conceptual and practical information necessary to develop a database design and management scheme that ensures data accuracy and user satisfaction while optimizing performance. Database systems underlie the large majority of business information systems. Most of those in use today are based on the relational data model, a way of representing data and data relationships using only two-dimensional tables. This book covers relational database theory as well as providing a solid introduction to SQL, the international standard for the relational database data manipulation language. The book begins by reviewing basic concepts of databases and database design, then turns to creating, populating, and retrieving data using SQL. Topics such as the relational data model, normalization, data entities, and Codd's Rules (and why they are important) are covered clearly and concisely. In addition, the book looks at the impact of big data on relational databases and the option of using NoSQL databases for that purpose. Features updated and expanded coverage of SQL and new material on big data, cloud computing, and object-relational databases Presents design approaches that ensure data accuracy and consistency and help boost performance Includes three case studies, each illustrating a different database design challenge Reviews the basic concepts of databases and database design, then turns to creating, populating, and retrieving data using SQL

Database Design: Know It All Academic Press

Language Integrated Query (LINQ), as well as the C# 3.0 and VB 9.0 language extensions to support it, is the most important single new feature of Visual Studio 2008 and the .NET Framework 3.x. LINQ is Microsoft's first attempt to define a universal query language for a diverse set of in-memory collections of generic objects, entities persisted in relational database tables, and element and attributes of XML documents or fragments, as well as a wide variety of other data types, such as RSS and Atom syndication feeds. Microsoft invested millions of dollars in Anders Hejlsberg and his C# design and development groups to add new features to C# 3.0—such as lambda expressions, anonymous types, and extension methods—specifically to support LINQ Standard Query Operators (SQOs) and query expressions as a part of the language itself. Corresponding additions to VB 9.0 followed the C# team's lead, but VB's implementation of LINQ to XML offers a remarkable new addition to the language: XML literals. VB's LINQ to XML implementation includes XML literals, which treat well-formed XML documents or fragments as part of the VB language, rather than requiring translation of element and attribute names and values from strings to XML DOM nodes and values. This book concentrates on hands-on development of practical Windows and Web applications that demonstrate C# and VB programming techniques to bring you up to speed on LINQ technologies. The first half of the book covers LINQ Standard Query Operators (SQOs) and the concrete implementations of LINQ for querying collections that implement generic IEnumerable, IQueryable, or both interfaces. The second half is devoted to the ADO.NET Entity Framework, Entity Data Model, Entity SQL (eSQL) and LINQ to Entities. Most code examples emulate real-world data sources, such as the Northwind sample database running on SQL Server 2005 or 2008 Express Edition, and collections derived from its tables. Code examples are C# and VB Windows form or Web site/application projects not, except in the first chapter, simple command-line projects. You can't gain a feel for the behavior or performance of LINQ queries with "Hello World" projects that process arrays of a few integers or a few first and last names. This book is intended for experienced .NET developers using C# or VB who want to gain the maximum advantage from the query-processing capabilities of LINQ implementations in Visual Studio 2008—LINQ to Objects, LINQ to SQL, LINQ to DataSets, and LINQ to XML—as well as the object/relational mapping (O/RM) features of VS 2008 SP1's Entity Framework/Entity Data Model and LINQ to Entities and the increasing number of open-source LINQ implementations by third-party developers. Basic familiarity with generics and other language features introduced by .NET 2.0, the Visual Studio integrated development environment (IDE), and relational database management systems (RDBMSs), especially Microsoft SQL Server 200x, is assumed. Experience with SQL Server's Transact-SQL (T-SQL) query language and stored procedures will be helpful but is not required. Proficiency with VS 2005, .NET 2.0, C# 2.0, or VB 8.0 will aid your initial understanding of the book's C# 3.0 or VB 9.0 code samples but isn't a prerequisite. Microsoft's .NET code samples are primarily written in C#. All code samples in this book's chapters and sample projects have C# and VB versions unless they're written in T-SQL or JavaScript. Professional ADO.NET 3.5: LINQ and the Entity Framework concentrates on programming the System.Linq and System.Linq.Expressions namespaces for LINQ to Objects, System.Data.Linq for LINQ to SQL, System.Data.Linq for LINQ to DataSet, System.Xml.Linq for LINQ to XML, and System.Data.Entity and System.Web.Entity for EF's Entity SQL. "Taking a New Approach to Data Access in

ADO.NET 3.5," uses simple C# and VB code examples to demonstrate LINQ to Objects queries against in-memory objects and databinding with LINQ-populated generic List collections, object/relational mapping (O/RM) with LINQ to SQL, joining DataTables with LINQ to DataSets, creating EntitySets with LINQ to Entities, querying and manipulating XML InfoSets with LINQ to XML, and performing queries against strongly typed XML documents with LINQ to XSD. "Understanding LINQ Architecture and Implementation," begins with the namespaces and C# and VB language extensions to support LINQ, LINQ Standard Query Operators (SQOs), expression trees and compiled queries, and a preview of domain-specific implementations. C# and VB sample projects demonstrate object, array, and collection initializers, extension methods, anonymous types, predicates, lambda expressions, and simple query expressions. "Executing LINQ Query Expressions with LINQ to Objects," classifies the 50 SQOs into operator groups: Restriction, Projection, Partitioning, Join, Concatenation, Ordering, Grouping, Set, Conversion, and Equality, and then lists their keywords in C# and VB. VS 2008 SP1 includes C# and VB versions of the LINQ Project Sample Query Explorer, but the two Explorers

don't use real-world collections as data sources. This describes a LINQ in-memory object generator (LIMOG) utility program that writes C# 3.0 or VB 9.0 class declarations for representative business objects that are more complex than those used by the LINQ Project Sample Query Explorers. Sample C# and VB queries with these business objects as data sources are more expressive than those using arrays of a few integers or last names. "Working with Advanced Query Operators and Expressions," introduces LINQ queries against object graphs with entities that have related (associated) entities. This begins with examples of aggregate operators, explains use of the Let temporary local variable operator, shows you how to use Group By with aggregate queries, conduct the equivalent of left outer joins, and take advantage of the Contains() SQO to emulate SQL's IN() function. You learn how to compile queries for improved performance, and create mock object classes for testing without the overhead of queries against relational persistence stores. "Using LINQ to SQL and the LinqDataSource," introduces LINQ to SQL as Microsoft's first O/RM tool to reach released products status and shows you how to autogenerate class files for entity types with the graphical O/R Designer or command-line SqlMetal.exe. This also explains how to edit

*.dbml mapping files in the Designer or XML Editor, instantiate DataContext objects, and use LINQ to SQL as a Data Access Layer (DAL) with T-SQL queries or stored procedures. Closes with a tutorial for using the ASP.NET LinqDataSource control with Web sites or applications. "Querying DataTables with LINQ to DataSets," begins with a comparison of DataSet and DataContext objects and features, followed by a description of the DataSetExtensions. Next comes querying untyped and typed DataSets, creating lookup lists, and generating LinqDataViews for databinding with the AsDataView() method. This ends with a tutorial that shows you how to copy LINQ query results to DataTables. "Manipulating Documents with LINQ to XML," describes one of LINQ most powerful capabilities: managing XML Infosets. This demonstrates that LINQ to XML has query and navigation capabilities that equal or surpasses XQuery 1.0 and XPath 2.0. It also shows LINQ to XML document transformation can replace XQuery and XSLT 1.0+ in the majority of common use cases. You learn how to use VB 9.0's XML literals to constructs XML documents, use GroupJoin() to produce hierarchical documents, and work with XML namespaces in C# and VB. "Exploring Third-Party and Emerging LINQ Implementations,"

describes Microsoft's Parallel LINQ (also called PLINQ) for taking advantage of multiple CPU cores in LINQ to Objects queries, LINQ to REST for translating LINQ queries into Representational State Transfer URLs that define requests to a Web service with the HTML GET, POST, PUT, and DELETE methods, and Bart De Smet's LINQ to Active Directory and LINQ to SharePoint third-party implementations. "Raising the Level of Data Abstraction with the Entity Data Model," starts with a guided tour of the development of EDM and EF as an O/RM tool and heir apparent to ADO.NET DataSets, provides a brief description of the entity-relationship (E-R) data model and diagrams, and then delivers a detailed analysis of EF architecture. Next comes an introduction to the Entity SQL (eSQL) language, eSQL queries, client views, and Object Services, including theObjectContext, MetadataWorkspace, and ObjectStateManager. Later chapters describe eSQL and these objects in greater detail. Two C# and VB sample projects expand on the eSQL query and Object Services sample code. "Defining Conceptual, Mapping, and Storage Schema Layers," provides detailed insight into the structure of the *.edmx file that generates the *.ssdl (storage schema data language), *.msl (mapping schema language), and *.csdl

files at runtime. You learn how to edit the *.edmx file manually to accommodate modifications that the graphic EDM Designer can't handle. You learn how to implement the Table-per-Hierarchy (TPH) inheritance model and traverse the MetadataWorkspace to obtain property values. Four C# and VB sample projects demonstrate mapping, substituting stored procedures for queries, and TPH inheritance. "Introducing Entity SQL," examines EF's new eSQL dialect that adds keywords to address the differences between querying entities and relational tables. You learn to use Zlatko Michaelov's eBlast utility to write and analyze eSQL queries, then dig into differences between eSQL and T-SQL SELECT queries. (eSQL v1 doesn't support INSERT, UPDATE, DELETE and other SQL Data Manipulation Language constructs). You execute eSQL queries against the EntityClient, measure the performance hit of eSQL compared to T-SQL, execute parameterize eSQL queries, and use SQL Server Compact 3.5 as a data store. C# and VB Sample projects demonstrate the programming techniques. "Taking Advantage of Object Services and LINQ to Entities," concentrates manipulating the Object Services API's ObjectContext. It continues with demonstrating use of partial classes for the ModelNameEntities and

EntityName objects, executing eSQL ObjectQuerys, and deferred or eager loading of associated entities, including ordering and filtering the associated entities. Also covers instructions for composing QueryBuilder methods for ObjectQuerys, LINQ to Entities queries, and parameterizing ObjectQuerys. "Updating Entities and Complex Types," shows you how to perform create, update, and delete (CRUD) operations on EntitySets and manage optimistic concurrency conflicts. It starts with a detailed description of the ObjectContext.ObjectStateManager and its child objects, which perform object identification and change tracking operations with EntityKeys. This also covers validation of create and update operations, optimizing the DataContext lifetime, performing updates with stored procedures, and working with complex types. "Binding Data Controls to the ObjectContext", describes creating design-time data sources from ObjectContext.EntitySet instances, drag-and-drop addition of BindingNavigator, BindingSource, bound TextBox, and DataGridView controls to Windows forms. You also learn how to update EntityReference and EntitySet values with ComboBox columns in DataGridView controls. (You can't update EntitySet values directly; you must delete and add a new member having

the required value). This concludes with a demonstration of the use of the ASP.NET EntityDataSource control bound to GridView and DropDownList controls. "Using the Entity Framework As a Data Source," concentrates on using EF as a data source for the ADO.NET Data Services Framework (the former codename "Project Astoria" remains in common use), which is the preferred method for deploying EF v1 as a Web service provider. (EF v2 is expected to be able to support n-tier data access with Windows Communication Foundation [WCF] directly). A Windows form example uses Astoria's .NET 3.5 Client Library to display and update entity instances with the Atom Publication (AtomPub or APP) wire format. The Web form project uses the AJAX Client Library and JavaScript Object Notation (JSON) as the wire format.