IBM PRESS

DB2 Developer's Guide

A Solutions-Oriented Approach to Learning the Foundation and Capabilities of DB2 for z/OS

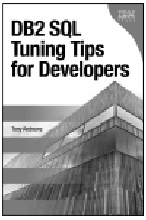
Sixth Edition

Craig S. Mullins

Common DB2 SQLCODE Values

SQLCODE	SQLSTATE	Description
+000	00000	The SQL statement finished successfully.
+100	02000	No rows found to satisfy the SQL statement.
+117	01525	Number of values being inserted does not equal number of columns in the table.
-101	54001	SQL statement is too complex.
-104	42601	Illegal symbol encountered in SQL statement. Usually, this means you have a syntax error somewhere in your SQL statement.
-122	42803	Column function used illegally; all columns not applied to the column function must be in the GROUP BY.
-150	42807	Invalid view UPDATE requested; or an invalid INSERT, UPDATE, or DELETE was requested on a transition table during a triggered action.
-305	22002	A null was returned but no indicator variable is available to assign null to the host variable.
-501	24501	Must open a cursor before attempting to fetch from it or close it.
-502	24502	Cannot open a cursor twice without first closing it.
-510	42828	The table specified by the cursor of the UPDATE or DELETE statement cannot be modified as requested.
-530	23503	Invalid foreign key value supplied for the specified constraint name.
-532	23504	Deletion violates the named referential constraint.
-545	23513	INSERT or UPDATE caused a check constraint violation.
-552	42502	User is attempting to perform an operation for which he or she is not authorized.
-803	23505	Insert violates uniqueness constraint.
-805	51002	The DBRM or package name was not found in the plan.
-811	21000	Must use a cursor when more than one row is returned as the result of an embedded SELECT statement.
-818	51003	Plan/Package vs. load module timestamp mismatch. The DBRM in the executing plan or package was not created from the same precompilation as the load module.
-904	57011	The specified resource is unavailable. Determine why, and retry the request.
-911	40001	The current unit of work has been rolled back.
-913	57033	Unsuccessful execution caused by deadlock or timeout.
-922	42505	The user is not authorized to perform the task

Related Books of Interest



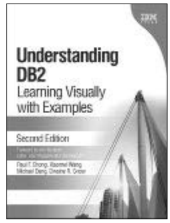
DB2 SQL Tuning Tips for Developers

by Tony Andrews

ISBN: 0-13-303846-7

This well-organized, easy-to-understand reference brings together 102 SQL-related skills and techniques any developer can use to build DB2® applications that deliver consistently superior performance. Legendary DB2 tuning expert Tony Andrews ("Tony the Tuner") draws on more than 23 years of DB2-related experience, empowering developers to take performance into their own hands—whether they're writing new software or tuning existing systems.

Andrews reveals the hidden truth about why DB2 queries, programs, and applications often perform poorly, and shows developers exactly how to clear the bottlenecks and resolve the problems. He fully reflects the latest DB2 SQL programming best practices up to and including DB2 V9 and DB2 V10 on z/OS®—techniques that are taught in no other book and are rarely covered in typical DB2 training courses.



Understanding DB2

Learning Visually with Examples, Second Edition

By Raul F. Chong, Xiaomei Wang, Michael Dang, and Dwaine R. Snow

ISBN: 0-13-158018-3

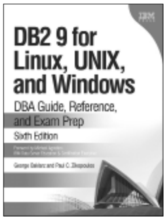
IBM® DB2® 9 and DB2 9.5 provide breakthrough capabilities for providing Information on Demand, implementing Web services and Service Oriented Architecture, and streamlining information management. *Understanding DB2: Learning Visually with Examples, Second Edition*, is the easiest way to master the latest versions of DB2 and apply their full power to your business challenges.

Written by four IBM DB2 experts, this book introduces key concepts with dozens of examples drawn from the authors' experiences working with DB2 in enterprise environments. Thoroughly updated for DB2 9.5, it covers new innovations ranging from manageability to performance and XML support to API integration. Each concept is presented with easy-to-understand screenshots, diagrams, charts, and tables. This book is for everyone who works with DB2: database administrators, system administrators, developers, and consultants. With hundreds of well-designed review questions and answers, it will also help professionals prepare for the IBM DB2 Certification Exams 730, 731, or 736.

Listen to the author's podcast at:
ibmpressbooks.com/podcasts

Sign up for the monthly IBM Press newsletter at
ibmpressbooks/newsletters

Related Books of Interest



DB2 9 for Linux, UNIX, and Windows

DBA Guide, Reference, and Exam Prep, Sixth Edition

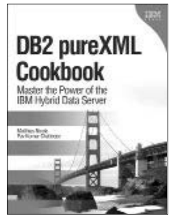
by George Baklarz and Paul C. Zikopoulos

ISBN: 0-13-185514-X

The sixth edition of this classic offers complete, start-to-finish coverage of DB2® 9 administration and development for Linux®, UNIX®, and Windows® platforms, as well as authoritative preparation for the latest IBM® DB2 certification exam. Written for both DBAs and developers, this definitive reference and self-study guide covers all aspects of deploying and managing DB2 9, including DB2 database design and development; day-to-day administration and backup; deployment of networked, Internet-centered, and SOA-based applications; migration; and much more.

You'll also find an unparalleled collection of expert tips for optimizing performance, availability, and value. Download Complete DB2 V9 Trial Version. Visit ibm.com/db2/9/download.html to download a complete trial version of DB2, which enables you to try out dozens of the most powerful features of DB2 for yourself—everything from pureXML™ support to automated administration and optimization.

Listen to the author's podcast at:
ibmpressbooks.com/podcasts



DB2 pureXML Cookbook

Master the Power of the IBM Hybrid Data Server

By Matthias Nicola and Pav Kumar-Chatterjee

ISBN: 0-13-815047-8

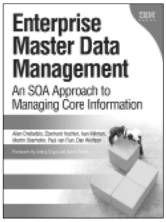
DB2® pureXML® Cookbook provides hands-on solutions and best practices for developing and managing XML database applications with DB2.

More and more database developers and DBAs are being asked to develop applications and manage databases that involve XML data. Many are utilizing the highly praised DB2 pureXML technology from IBM®. In *DB2 pureXML Cookbook*, two leading experts from IBM offer the practical solutions and proven code samples that database professionals need to build better XML solutions faster. Organized by task, this book is packed with more than 700 easy-to-adapt "recipe-style" examples covering the entire application lifecycle–from planning and design through coding, optimization, and troubleshooting.

IBM Press™

Visit ibmpressbooks.com
for all product information

Related Books of Interest



Enterprise Master Data Management

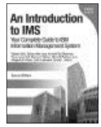
An SOA Approach to Managing Core Information

by Allen Dreibelbis, Eberhard Hechler, Ivan Milman, Martin Oberhofer, Paul van Run, and Dan Wolfson

ISBN: 0-13-236625-8

Enterprise Master Data Management provides an authoritative, vendor-independent MDM technical reference for practitioners: architects, technical analysts, consultants, solution designers, and senior IT decision makers. Written by the IBM® data management innovators who are pioneering MDM, this book systematically introduces MDM's key concepts and technical themes, explains its business case, and illuminates how it interrelates with and enables SOA.

Drawing on their experience with cutting-edge projects, the authors introduce MDM patterns, blueprints, solutions, and best practices published nowhere else—everything you need to establish a consistent, manageable set of master data, and use it for competitive advantage.



An Introduction to IMS

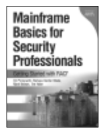
Klein, Long, Blackman, Goff, Nathan, Lanyi, Wilson, Butterweck, Sherrill

ISBN: 0-13-288687-1

IBM Cognos 10 Report Studio: Practical Examples

Draskovic, Johnson

ISBN: 0-13-265675-2



Mainframe Basics for Security Professionals

Pomerantz, Vander, Weele, Nelson, Hahn

ISBN: 0-13-173856-9

Service-Oriented Architecture (SOA) Compass

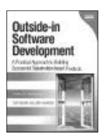
Bieberstein, Bose, Fiammante, Jones, Shah

ISBN: 0-13-187002-5

WebSphere Business Integration Primer

Iyengar, Jessani, Chilanti

ISBN: 0-13-224831-X



Outside-in Software Development

Kessler, Sweitzer

ISBN: 0-13-157551-1

Sign up for the monthly IBM Press newsletter at ibmpressbooks/newsletters

Accolades for *DB2 Developer's Guide*

"Once you've picked up and read *DB2 Developer's Guide,* you will know why people on the DB2 List Serve forum refer to this book as the BIBLE. You will find that the *DB2 Developer's Guide* is a comprehensive guide for both the beginner and experienced in DB2 and relational database technology…I cannot say enough about the *DB2 Developer's Guide*."

—Troy Coleman
Data Administration Newsletter

"*DB2 Developer's Guide* has the potential to pay for itself many times over if you follow its useful design and performance advice. If you use DB2 in your workplace, the most recent edition of *DB2 Developer's Guide* should definitely be on your bookshelf. Read it to save yourself an enormous amount of pain and suffering."

—Ron Shirey
Relational Database Journal

"…the book is not only the size of a small encyclopedia, it is also just about as comprehensive."
Books & Bytes News & Reviews

"*DB2 Developer's Guide* is a must buy for both inexperienced and DB2 experts alike. I am amazed at the amount of information Craig covers in the *DB2 Developer's Guide*."

—Chris Foot
Data Administration Newsletter

"*DB2 Developer's Guide* is a complete reference for the DB2 professional. It is a perfect tool for finding the options available to the DB2 developer, and steering you to the right method."

—Gregory Amov
Computing News & Review

"*DB2 Developer's Guide* presents literally everything programmers and DBAs need to know about advanced DB2…This is an excellent book…It is chock full of DB2 technical information, design and tuning approaches, and database administration guidelines…In an organized and coherent way, Mullins seems to have dumped his entire DB2 life experience into *DB2 Developer's Guide*."

—Jonathon Sayles
Relational Database Journal

"Enormous amount of priceless information. I don't think there has ever been any other publication that managed to cover so much. And the book is not just a 'developer's guide'—the book is 'The DB2 professional's guide.'"

—Daniela Guentcheva
On amazon.com

"With more than 25 years experience as an application developer on IBM mainframes, including 15 years with DB2, I thought that there was little I didn't know until reading Craig Mullins' book. It goes into the depth required (and beyond) for professional developers and even deeper into the domain of the DBA's territory."

—Keith A. Marsh
On amazon.com

DB2 Developer's Guide, Sixth Edition

DB2 Developer's Guide, Sixth Edition

A Solutions-Oriented Approach to Learning the Foundation and Capabilities of DB2 for z/OS

Craig S. Mullins

IBM Press
Pearson plc
Upper Saddle River, NJ • Boston • Indianapolis • San Francisco
New York • Toronto • Montreal • London • Munich • Paris • Madrid
Cape Town • Sydney • Tokyo • Singapore • Mexico City

ibmpressbooks.com

IBM Press Program Managers: Steven M. Stansel, Ellice Uffer
Cover design: IBM Corporation

Associate Publisher: Dave Dusthimer
Marketing Manager: Stephane Nakib
Executive Editor: Mary Beth Ray
Publicist: Andrea Bledsoe
Editorial Assistant: Vanessa Evans
Senior Development Editor: Christopher Cleveland
Technical Reviewers: Willie Favero, Chuck Kosin
Managing Editor: Kristy Hart
Senior Project Editor: Lori Lyons
Copy Editor: Apostrophe Editing Services
Indexer: Brad Herriman
Compositor: Nonie Ratcliff
Proofreaders: Katherine Ruiz, Chrissy White/Language Logistics, LLC
Manufacturing Buyer: Dan Uhrig

Published by Pearson plc
Publishing as IBM Press

IBM Press offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact

U. S. Corporate and Government Sales
1-800-382-3419
corpsales@pearsontechgroup.com.

For sales outside the U. S., please contact

International Sales
international@pearson.com.

Library of Congress Cataloging-in-Publication Data

Mullins, Craig.
Db2 developer's guide : a solutions-oriented approach to learning the foundation and capabilities of Db2 for z/OS / Craig S. Mullins. — 6th ed.
p. cm.
Includes index.
ISBN 978-0-13-283642-5 (pbk.)
1. Database management. 2. IBM Database 2. I. Title.
QA76.9.D3M84 2012
005.75'65—dc23
2012006815

Pearson Education, Inc
Rights and Contracts Department
501 Boylston Street, Suite 900
Boston, MA 02116
Fax (617) 671 3447

ISBN-13: 978-0-13-3283642-5
ISBN-10: 0-13-283642-4

Text printed in the United States on recycled paper at Edwards Brothers Malloy, Ann Arbor, Michigan.
First printing May 2012

This book is dedicated to my mom, Donna Mullins,
and to the memory of my father, Giles R. Mullins.

Without the constant support and guidance my parents provided,
I would not have the success I enjoy today.

Contents at a Glance

Introduction ... 1

Part I SQL Techniques, Tips, and Tricks

1 The Magic Words ... 3
2 Data Manipulation Guidelines ... 56
3 Using DB2 Functions ... 135
4 Using DB2 User-Defined Functions and Data Types ... 167
5 Data Definition Guidelines ... 200
6 DB2 Indexing and Hashing Guidelines ... 324
7 Database Change Management, Schema Evolution, and Database Definition On Demand ... 353
8 Using DB2 Triggers ... 373
9 Large Objects and Object/Relational Databases ... 393
10 pureXML: Using XML in DB2 for z/OS ... 408
11 Supporting Temporal Data in DB2 for z/OS ... 428
12 DB2 Security, Authorization, and Auditing ... 448

Part II DB2 Application Development

13 Using DB2 in an Application Program ... 486
14 Dynamic SQL Programming ... 567
15 Program Preparation ... 601
16 Using DB2 Stored Procedures ... 656
17 DB2 and the Internet ... 689

Part III DB2 In-Depth

18 The Doors to DB2 ... 704
19 Data Sharing ... 772
20 DB2 Behind the Scenes ... 792
21 The Optimizer ... 816
22 The Table-Based Infrastructure of DB2 ... 874
23 Locking DB2 Data ... 889

Part IV DB2 Performance Monitoring

24 DB2 Performance Monitoring ... 928
25 Using `EXPLAIN` ... 980
26 The Five R's ... 1014
27 DB2 Object Monitoring Using the DB2 Catalog and RTS ... 1021

Part V DB2 Performance Tuning

28 Tuning DB2's Environment ... 1064
29 Tuning DB2's Components ... 1089
30 DB2 Resource Governing ... 1143

Part VI DB2 Utilities and Commands

31 An Introduction to DB2 Utilities 1152
32 Data Consistency Utilities 1176
33 Backup and Recovery Utilities 1201
34 Data Movement and Organization Utilities 1240
35 Catalog Manipulation Utilities 1289
36 Stand-Alone Utilities and Sample Programs 1314
37 DB2 Commands 1340
38 DB2 Utility and Command Guidelines 1366
39 DB2 Contingency Planning 1376

Part VII The Ideal DB2 Environment

40 Components of a Total DB2 Solution 1394
41 Organizational Issues 1423

Part VII Distributed DB2

42 DRDA 1448
43 Distributed DB2 1458
44 DB2 Connect 1473
45 Distribution Guidelines 1485
46 Data Warehousing with DB2 1506
Index 1541

Contents

Preface xxiii

Part I SQL Techniques, Tips, and Tricks

1 The Magic Words 3

An Overview of SQL 4
SQL Tools of the Trade 13
Static SQL 42
Dynamic SQL 44
SQL Performance Factors 45

2 Data Manipulation Guidelines 56

A Bag of Tricks 56
SQL Access Guidelines 58
Complex SQL Guidelines 90
Common Table Expressions and Recursion 110
Working with Nulls 115
Date and Time Guidelines 119
Data Modification Guidelines 125

3 Using DB2 Functions 135

Aggregate Functions 135
Scalar Functions 141
Table Functions 159
MQSeries Built-In Functions 159
XML Built-In Functions 161
The `RAISE_ERROR` Function 162
The `CAST` Operation 163
Built-In Function Guidelines 163

4 Using DB2 User-Defined Functions and Data Types 167

What Is a User-Defined Function? 167
Types of User-Defined Functions (UDFs) 168
What Is a User-Defined Data Type? 190
User-Defined Data Types (UDTs) and Strong Typing 191

5 Data Definition Guidelines 200

An Overview of DB2 Database Objects ... 200
DB2 Databases ... 201
Creating and Using DB2 Table Spaces ... 204
DB2 Storage and STOGROUPs ... 239
Table Guidelines ... 244
General Table Guidelines ... 275
Normalization and Denormalization ... 278
Assuring Data Integrity in DB2 ... 290
Referential Integrity ... 290
Views, Aliases, and Synonyms ... 302
Index Guidelines ... 313
Naming Conventions ... 313
Miscellaneous DDL Guidelines ... 322

6 DB2 Indexing and Hashing Guidelines 324

How an Index Works ... 324
Creating Indexes ... 326
DB2 Hashing and Hash Organized Tables ... 337
Index and Hash Guidelines ... 341

7 Database Change Management, Schema Evolution, and Database Definition On Demand 353

Online Schema Changes ... 354
Versioning for Online Schema Changes ... 370

8 Using DB2 Triggers 373

What Is a Trigger? ... 373
Trigger Guidelines ... 388

9 Large Objects and Object/Relational Databases 393

Defining the Term "Object/Relational" ... 393
What Is a Large Object? ... 394
LOB Guidelines ... 403
DB2 Extenders ... 407

10 pureXML: Using XML in DB2 for z/OS 408

What Is XML? ... 408
pureXML ... 412
XML-DB2 Guidelines ... 425

11 Supporting Temporal Data in DB2 for z/OS 428

The Need for Temporal Data ... 428
DB2 Temporal Support ... 430
Temporal Data Guidelines ... 446
Summary ... 447

12 DB2 Security, Authorization, and Auditing 448

Authorization and Privileges ... 448
Database Auditing ... 476
Using External Security (for Example, RACF, ACF2, and Top Secret) ... 480

Part II DB2 Application Development

13 Using DB2 in an Application Program 486

Embedded SQL Basics ... 487
Embedded SQL Guidelines ... 489
Host Variables ... 504
Programming with Cursors ... 511
Modifying Data with Embedded SQL ... 525
Application Development Guidelines ... 527
Batch Programming Guidelines ... 536
Online Programming Guidelines ... 547
General SQL Coding Guidelines ... 552
Introduction to Java ... 554
Using REXX and DB2 ... 563
Developing Applications Using Only SQL ... 565

14 Dynamic SQL Programming 567

What Is Dynamic SQL? ... 567
Dynamic SQL Versus Static SQL ... 569
The Four Classes of Dynamic SQL ... 576
pureQuery ... 588
Making Dynamic SQL More Static and Vice Versa ... 589
Dynamic SQL Guidelines ... 594

15 Program Preparation 601

Program Preparation Steps ... 601
Running a DB2 Program ... 608
Preparing a DB2 Program ... 609

What Is a DBRM? 622
What Is a Plan? 622
What Is a Package? 623
What Is a Collection? 628
Versions 629
Converting DBRM-Based Plans in DB2 V10 630
Program Preparation Objects 631
Program Preparation Guidelines 632

16 Using DB2 Stored Procedures 656

What Is a Stored Procedure? 657
Implementing DB2 Stored Procedures 661
Procedural SQL 678
The Procedural DBA 683
IBM Data Studio 687

17 DB2 and the Internet 689

The Internet Phenomenon 689
Accessing DB2 over the Internet 692
Finding DB2 Information Using the Internet 695

Part III DB2 In-Depth

18 The Doors to DB2 704

DB2 Program Execution Basics 704
TSO (Time-Sharing Option) 706
CICS (Customer Information Control System) 726
IMS (Information Management System) 751
CAF (Call Attach Facility) 763
RRSAF (Recoverable Resource Manager Services Attach Facility) 767
Comparison of the Environments 768

19 Data Sharing 772

Data Sharing Benefits 772
Data Sharing Requirements 774
The DB2 Coupling Facility 778
Data Sharing Naming Conventions 782
Data Sharing Administration 783
Data Sharing Application Development Guidelines 787
Data Sharing Administration Guidelines 788

20 DB2 Behind the Scenes 792

The Physical Storage of Data 792
What Makes DB2 Tick 808
Specialty Processors 812

21 The Optimizer 816

Physical Data Independence 817
How the Optimizer Works 818
Filter Factors 821
Screening 823
Access Path Strategies 824
Other Operations Performed by the Optimizer 868

22 The Table-Based Infrastructure of DB2 874

The DB2 Catalog 874
The DB2 Directory 886

23 Locking DB2 Data 889

How DB2 Manages Locking 889
Locks Versus Latches 892
Lock Duration 892
Table Space Locks 895
Table Locks 897
Page Locks 898
Row Locks 899
Lock Suspensions, Timeouts, and Deadlocks 901
Partition Independence 904
Lock Avoidance 908
Data Sharing Global Lock Management 911
LOBs and Locking 914
DB2 Locking Guidelines 916
Other DB2 Components 921
The Big Picture 922

Part IV DB2 Performance Monitoring

Defining DB2 Performance 926
Types of DB2 Performance Monitoring 927

24 DB2 Performance Monitoring 928

DB2 Traces 929
Trace Destinations 936

Using IFCIDs 937
Tracing Guidelines 938
Performance Monitoring and Reporting: Online and Batch 940
Monitoring and Reporting Strategy 967
Performance Profiles 970
Viewing DB2 Console Messages 972
Displaying the Status of DB2 Resources 977
Monitoring z/OS 979

25 Using EXPLAIN 980

How `EXPLAIN` Works 980
Access Paths and the `PLAN_TABLE` 982
Cost Estimates and the `DSN_STATEMNT_TABLE` 998
Function Resolution and the `DSN_FUNCTION_TABLE` 1001
Additional Explain Tables 1002
Explaining the Dynamic Statement Cache 1003
`EXPLAIN` Guidelines 1005
Additional Tools for Managing Access Paths 1012

26 The Five R's 1014

Approaches to Rebinding 1014
A Best Practice Approach to Rebinding 1016

27 DB2 Object Monitoring Using the DB2 Catalog and RTS 1021

DB2 Catalog Queries 1021
Real Time Statistics 1048
Reviewing the Rules for an Effective Monitoring Strategy 1058

Part V DB2 Performance Tuning

28 Tuning DB2's Environment 1064

Tuning the z/OS Environment 1064
Tuning the Teleprocessing Environment 1087

29 Tuning DB2's Components 1089

Tuning the DB2 Subsystem 1089
Tuning the Database Design 1114
Tuning the Application 1116
The Causes of DB2 Performance Problems 1137

30 DB2 Resource Governing 1143

The Resource Limit Facility 1143

Part VI DB2 Utilities and Commands

31 An Introduction to DB2 Utilities 1152

Generating Utility JCL 1152
Monitoring DB2 Utilities 1156
The IBM DB2 Utilities 1158
Using LISTDEF and TEMPLATE 1159
Issuing SQL Statements in DB2 Utilities 1173

32 Data Consistency Utilities 1176

The CHECK Utility 1177
The CHECK DATA Option 1177
The CHECK LOB Option 1186
The CHECK INDEX Option 1188
The REPAIR Utility 1191
The REPAIR DBD Option 1192
The REPAIR LOCATE Option 1193
The REPAIR SET Option 1196
REPAIR and Versions 1198
The REPORT Utility 1198
The DIAGNOSE Utility 1200

33 Backup and Recovery Utilities 1201

The COPY Utility 1202
The COPYTOCOPY Utility 1215
The MERGECOPY Utility 1218
The QUIESCE Utility 1220
The RECOVER Utility 1224
The REBUILD INDEX Utility 1232
The REPAIR Utility 1235
The REPORT RECOVERY Utility 1235
Backing Up and Restoring the System 1236

34 Data Movement and Organization Utilities 1240

The LOAD Utility 1240
The UNLOAD Utility 1260
The REORG Utility 1265

35 Catalog Manipulation Utilities 1289

The CATENFM Utility 1289
The CATMAINT Utility 1289

The DSNJCNVB Utility ... 1290
The MODIFY RECOVERY Utility ... 1290
The MODIFY STATISTICS Utility ... 1293
The RUNSTATS Utility ... 1295
The STOSPACE Utility ... 1311

36 Stand-Alone Utilities and Sample Programs 1314

The Stand-Alone Utilities ... 1314
DB2 Sample Programs ... 1332

37 DB2 Commands 1340

DB2 Environment Commands ... 1340
Information-Gathering Commands ... 1343
Administrative Commands ... 1353
Environment Control Commands ... 1358
DSN Commands ... 1359
IMS Commands ... 1361
CICS Commands ... 1362
TSO Commands ... 1364
IRLM Commands ... 1364

38 DB2 Utility and Command Guidelines 1366

Utility Guidelines ... 1366
The Pending States ... 1372

39 DB2 Contingency Planning 1376

What Is a Disaster? ... 1376
DB2 Recovery Basics ... 1380
Additional DB2 Disaster Recovery Technologies ... 1387
DB2 Environmental Considerations ... 1388
DB2 Contingency Planning Guidelines ... 1390

Part VII The Ideal DB2 Environment

40 Components of a Total DB2 Solution 1394

DB2 Tools ... 1394
DB2 Tools Vendors ... 1420

41 Organizational Issues 1423

Education ... 1423
Standards and Procedures ... 1429

Operational Support ... 1440
Political Issues ... 1441
Environmental Support ... 1443
Tool Requirements ... 1443

Part VIII Distributed DB2

The Advantages of Data Distribution ... 1446
DB2 Data Distribution ... 1446
DB2 Data Warehousing ... 1447

42 DRDA 1448

What Is DRDA? ... 1448
DRDA Functions ... 1449
DRDA Architectures and Standards ... 1451
The Five DRDA Levels ... 1453
Putting It All Together ... 1455

43 Distributed DB2 1458

Distributing Data Using DB2 ... 1458
DB2 Support for the DRDA Levels ... 1460
Methods of Accessing Distributed Data ... 1460
Packages for Static SQL ... 1465
Two-Phase Commit ... 1466
Miscellaneous Distributed Topics ... 1470

44 DB2 Connect 1473

An Overview of IBM DB2 Connect ... 1473

45 Distribution Guidelines 1485

Distribution Behind the Scenes ... 1485
Block Fetch ... 1487
Dynamic Cursor Pre-Open ... 1491
Distributed Performance Problems ... 1491
Distributed Database Design Issues ... 1496
Distributed Data Placement ... 1499
Distributed Optimization ... 1500
Distributed Security Guidelines ... 1501
Miscellaneous Distributed Guidelines ... 1502

46 Data Warehousing with DB2 1506

Defining the Basic Terms 1507
Designing a Data Warehouse 1510
Populating a Data Warehouse 1513
Accessing the Data Warehouse 1519
Managing the Data Warehouse 1520
The Big Picture 1520
IBM Data Warehousing Solutions 1521
Materialized Query Tables 1522
General Data Warehouse Guidelines 1533
DB2-Specific Data Warehousing Guidelines 1538

Index 1541

Preface: A Short History of DB2 for z/OS

Let's go back in time…almost three decades ago…back to the wild and woolly 1980s! And watch as our favorite DBMS, DB2, grows up over time.

Version 1 Release 1 was announced on June 7, 1983. And it became generally available on Tuesday, April 2, 1985. I wonder if it was ready on April 1st but not released because of April Fool's Day? Initial DB2 development focused on the basics of making a relational DBMS work. Early releases of DB2 were viewed by many as an "information center" DBMS, not for production work like IMS.

Version 1 Release 2 was announced on February 4, 1986 and was released for general availability a month later on March 7, 1986. Wow! Can you imagine waiting only a month for a new release of DB2 these days? But that is how it happened back then. Same thing for Version 1 Release 3, which was announced on May 19, 1987 and became GA on June 26, 1987. DB2 V1R3 saw the introduction of date data types.

You might notice that IBM delivered "releases" of DB2 in the 1980s, whereas today (and ever since V3) there have been only versions. Versions are major, whereas releases are not quite as significant as a version.

Version 2 of DB2 became a reality in 1988. Version 2 Release 1 was announced in April 1988 and delivered in September 1988. Here we start to see the gap widening again between announcement and delivery. V2R1 was a significant release in the history of DB2. Some mark it as the bellwether for when DB2 began to be viewed as a DBMS capable of supporting mission critical, transaction processing workloads. Not only did V2R1 provide many performance enhancements, but it also signaled the introduction of declarative Referential Integrity (RI) constraints. RI was important for the acceptance of DB2 because it helps to assure data integrity within the DBMS.

No sooner than V2R1 became GA than IBM announced Version 2 Release 2 on October 4, 1988. But it was not until a year later that it became generally available on September 23, 1988. DB2 V2R2 again bolstered performance in many ways. It also saw the introduction of distributed database support (private protocol) across MVS systems.

Version 2 Release 3 was announced on September 5, 1990 and became generally available on October 25, 1991. Two significant features were added in V2R3: segmented table spaces and packages. Segmented table spaces quickly became the de facto standard for most DB2 data, and packages made DB2 application programs easier to support. DB2 V2R3 is also the version that beefed up distributed support with Distributed Relational Database Architecture (DRDA). Remote unit of work distribution was not available in the initial GA version, but IBM came out with RUOW support for DB2 V2R3 in March 1992.

DB2 Version 3 was announced in November 1993 and GA in December 1993. Now it may look like things sped up again here, but not really. This is when the QPP program for early support of DB2 started. QPP was announced in March 1993 and delivered to customers in June 1993. Still though, this is a fairly rapid version turnaround by today's standards.

V3 greatly expanded the number of buffer pool options available (from 5 pools to 80). There were many advances made in DB2 V3 to take better advantage of the System 390 environment: V3 introduced support for hardware-assisted compression and hiperpools. It was also V3 that introduced I/O parallelism for the first time.

Version 4 was a significant milestone in the history of DB2. It was highlighted by the introduction of Type 2 indexes, which removed the need to lock index pages (or subpages, which are now obsolete). Prior to V4, index locking was a particularly thorny performance problem that vexed many shops.

And, of course, I'd be remiss if I did not discuss data sharing, which made its debut in V4. With data sharing, DB2 achieved new heights of scalability and availability unmatched within the realm of DBMS; it afforded users the ability to upgrade without an outage and to add new subsystems to a group on-the-fly. The new capabilities did not stop there; V4 also introduced stored procedures, CP parallelism, performance improvements, and more. DB2 V4 was, indeed, a major milestone in the history of mainframe DB2.

In June 1997, DB2 Version 5 became generally available. It was the first DB2 version to be referred to as DB2 for OS/390 (previously it was DB2 for MVS). Not as significant as V4, we see the trend of even-numbered releases being bigger and more significant than odd-numbered releases. (Of course, this is just my opinion.) V5 was touted by IBM as the e-business and BI version. It included Sysplex parallelism, prepared statement caching, reoptimization, online REORG, and conformance to the SQL-92 standard.

Version 6 brings us to 1999 and the introduction of the Universal Database term to the DB2 moniker. The "official" name of the product is now DB2 Universal Database for OS/390. And the Release Guide swelled to more than 600 pages! Six categories of improvements were introduced with V6 spanning:

- Object-relational extensions and active data
- Network computing
- Performance and availability
- Capacity improvements
- Data sharing enhancements
- User productivity

The biggest of the new features were SQLJ, inline statistics, triggers, large objects (LOBs), user-defined functions, and distinct types.

Version 6 is also somewhat unique in that there was this "thing" typically referred to as the V6 refresh. It added functionality to DB2 without there being a new release or version. The new functionality in the refresh included `SAVEPOINT`s, identity columns, declared temporary tables, and performance enhancements (including star join). I wonder why IBM did not just issue a point release like in the past?

March 2001 brings us to DB2 Version 7, another "smaller" version of DB2. Developed and released around the time of the Year 2000 hubbub, it offered much improved utilities and some nice new SQL functionality, including scrollable cursors, limited FETCH, and row expressions. Unicode support was also introduced in DB2 V7. For a more detailed overview of V7 (and the V6 refresh) consult my web site at

```
http://www.craigsmullins.com/dbaz-DB2v7.htm.
```

DB2 Version 8 followed, but not immediately. IBM took advantage of Y2K and the general desire of shops to avoid change during this period to take its time and deliver the most significant and feature-laden version of DB2 ever. V8 had more new lines of code than DB2 V1R1 had total lines of code.

I don't want to get bogged down in recent history here, outlining the features and functionality of DB2 releases that should be fresh in our memory (V8 and V9). If you want some details on those, I refer you to the web again and the following links:

```
V8: http://www.craigsmullins.com/zjdp_001.htm
V9: http://www.craigsmullins.com/zjdp_025.htm
```

Which brings us to today. Most shops should be either running Version 10 in production or planning their migration to V10 from either V8 or V9.

Let this book be your guide to DB2 V10!

Acknowledgments

Writing and producing a technical book is a time-consuming and laborious task. Luckily, I had many understanding and helpful friends and associates who made the process much easier. First, I need to thank my wife, Beth, for her understanding, thoughtfulness, and care during the time it took to produce this edition of the book. You're the best, Bethie!

I would also like to thank the many folks who have reviewed and commented upon the text for each of this book's six editions. Chuck Kosin has served as a technical editor for the book since the second edition, and I am sure it is a much better text thanks to his eagle eye, technical acumen, and excellent suggestions. Chuck also helped with several screen shots and testing out all the SQL. I would also like to thank Willie Favero at IBM Corporation for his review and suggestions for this edition, as well as the past few editions of the book. Roger Miller, recently retired from IBM Corporation, has offered guidance, material, and suggestions for improving the book in each of its many editions. Sheryl Larsen (SML, Inc.) has been especially helpful in reviewing the access path and complex SQL components of the book; and I additionally want to thank her for her kind permission allowing me to include her access path diagrams in my book. Thanks also to Bayard "Tink" Tysor, who reviewed the pureXML chapter, and Rebecca Bond, who reviewed the security chapter; both of these friends and experts in their field offered helpful suggestions and the book is better because of their contributions. Bill Arledge (of CA Technologies) was also helpful pulling together screen shots for various sections of the book, as well as reading through several chapters of the manuscript. I'd also like to thank the following people for their friendship, support, and contributions to various editions of this book: Bill Backs, Troy Coleman, Bernard Klopfer, and Dan Pizzica.

A big hearty "Thank You" goes out to the many readers who provided suggestions for improvements on each of the previous editions of the book (either in person or by e-mail). I do read the e-mail suggestions and comments sent to me by readers, so keep them coming.

Special thanks to Dan Wardman, Vice President, IM Mainframe Software and Site Executive, IBM Silicon Valley Lab, who acted as the Executive Champion for the sixth edition of this book.

In addition, many thanks to the understanding and patient folks at IBM Press who have worked with me on this edition of the book, specifically Mary Beth Ray, Steven Stansel, Ellice Uffer, Christopher Cleveland, and all the editorial and production staff who were involved in producing the sixth edition of the book.

I'd also like to thank the editors and production team at SAMS Publishing, which published the first five editions of this book, specifically Carol Ackerman, Andy Beaster, Charlie Dresser, Rosemarie Graham, Patricia Kinyon, Susan Pink, Beverly Scherf, Loretta Yates, and everyone who had a hand in any of the previous editions of this book.

And finally, a big thank-you to all the people with whom I have worked and come in contact with professionally over the years. I'd specifically like to thank my coworkers at

USX Corporation, Mellon Bank, ASSET, Inc., Barnett Technologies, Duquesne Light Company, Gartner Group, PLATINUM Technology, Inc., BMC Software, NEON Enterprise Software, and SoftwareOnZ LLC. This book is surely a better one due to the fine quality of my coworkers, each of whom has expanded my horizons in many different and satisfying ways.

If you have any questions or comments about this text, you can contact me at `craig@craigsmullins.com` or via my web site at `http://www.CraigSMullins.com`. You can also write to me in care of the publisher.

About the Author

Craig S. Mullins is a data management strategist, researcher, and consultant. He is president and principal consultant of Mullins Consulting, Inc. and the publisher and editor of *The Database Site* (`http://www.TheDatabaseSite.com`). Craig has also been appointed as an Information Champion by IBM.

Craig has extensive experience in all facets of database systems development, including systems analysis and design, database and system administration, data analysis, and developing and teaching DB2 and database development classes. He has worked with DB2 since Version 1 and has experience in multiple roles, including programmer, DBA, instructor, and analyst. His experience spans industries, having worked for companies in the following fields: manufacturing (USX Corporation), banking (Mellon Bank), utilities (Duquesne Light Company), commercial software development (BMC Software, NEON Enterprise Software, and PLATINUM Technology, Inc.), consulting (ASSET, Inc. and Mullins Consulting, Inc.), and computer industry analysis (Gartner Group). In addition, Craig authored many of the popular “Platinum Monthly DB2 Tips” and worked on Platinum’s DB2 system catalog and access path posters.

Craig is a regular lecturer at industry conferences. You may have seen him present at such events as the International DB2 Users Group (IDUG), the IBM Information on Demand (IOD) Conference, the IBM DB2 Technical Conference, SHARE, DAMA, CMG, or at one of many regional user groups throughout the world. Craig is a member of the IDUG Volunteers Hall of Fame.

Craig is also the author of *Database Administration: The Complete Guide to Practices and Procedures* (ISBN 0-201-74129-6). This book offers the industry’s only comprehensive guide to heterogeneous database administration.

Craig is a frequent contributor to computer industry publications, with hundreds of articles published over the past couple decades. His articles have been published in *Byte*, *DB2 Update*, *Database Programming & Design*, *DBMS*, *Data Management Review*, *zJournal*, and many others. Craig writes four regular columns, including “The DBA Corner” for *Database Trends and Applications*, “The Database Report” for *The Data Administration Newsletter*, “z/Data Perspectives” for *zJournal*, and “The Buffer Pool” for *IDUG Solutions Journal*. He also writes a blog focusing on DB2 topics at `http://db2portal.blogspot.com`. Complete information on Craig’s published articles and books can be found on his website at `http://www.craigsmullins.com`.

Craig graduated *cum laude* from the University of Pittsburgh with a B.S. degree and a dual major in computer science and economics.

Follow Craig on Twitter at `http://www.twitter.com/craigmullins`.

IN THIS CHAPTER

- What Is XML?
- pureXML
- XML-DB2 Guidelines

CHAPTER **10**

pureXML: Using XML in DB2 for z/OS

XML is gaining popularity for persisting complex data and is frequently used in web-enabled applications and as a means of data transmission. This chapter introduces you to the basics of XML and provides an overview of pureXML, IBM's implementation of XML support embedded in DB2.

A comprehensive treatment of DB2 pureXML would require a book length treatment, and it is not the intent of this book, or this chapter, to provide an exhaustive treatment of DB2's support for XML. If you are looking for an introduction, this chapter is a good starting place. References for additional pureXML research are provided at the end of the chapter.

What Is XML?

XML stands for Extensible Markup Language. You may be familiar with HTML, the markup language used to create web pages. Like HTML, XML is based upon Standard Generalized Markup Language (SGML). SGML is a language for defining markup languages that was developed and standardized by the International Organization for Standardization (ISO).

Whereas HTML uses tags to describe *how* data appears on a web page, XML is designed to transport and store data. In other words, XML uses tags to describe the *what*—that is, the data itself. XML retains the key SGML advantage of self-description, while avoiding the complexity of full-blown SGML. XML allows tags to be defined by users that describe the data stored in the document. This capability gives users a means for describing the structure and nature of the data in the document. In essence, the document becomes self-describing.

The simple syntax of XML makes it easy to process by machine while remaining understandable to people. Again, use HTML as a metaphor to help you understand XML. HTML uses tags to describe the appearance of data on a page. For example the tag, “text”, would specify that the “text” data should appear in bold face. XML uses tags to describe the data itself, instead of its appearance. For example, consider the following XML describing a customer address:

```
<CUSTOMER>
   <first_name>Craig</first_name>
   <middle_initial>S.</middle_initial>
   <last_name>Mullins</last_name>
   <company_name>Mullins Consulting, Inc.</company_name>
   <street_address>15 Coventry Ct.</street_address>
   <city>Sugar Land</city>
   <state>TX</state>
   <zip_code>77479</zip_code>
   <country>USA</country>
</CUSTOMER>
```

XML is actually a meta-language—that is, a language for defining other markup languages. These languages are collected in dictionaries called Document Type Definitions (DTDs). The DTD stores definitions of tags for specific industries or fields of knowledge. So, the meaning of a tag must be defined in a “document type declaration” (DTD), such as the following:

```
<!DOCTYPE CUSTOMER [
  <!ELEMENT CUSTOMER (first_name, middle_initial, last_name,
             street_address, city, state, zip_code, country)>
  <!ELEMENT first_name (#PCDATA)>
  <!ELEMENT middle_initial (#PCDATA)>
  <!ELEMENT last_name (#PCDATA)>
  <!ELEMENT street_address (#PCDATA)>
  <!ELEMENT city (#PCDATA)>
  <!ELEMENT state (#PCDATA)>
  <!ELEMENT zip_code (#PCDATA)>
  <!ELEMENT country (#PCDATA)>
]
```

The DTD for an XML document can either be part of the document or stored in an external file. The XML code samples shown are meant to be examples only. By examining them, you can quickly see how the document describes its contents.

For data management professionals, this is a plus because it eliminates the trouble to track down the meaning of data elements. One of the biggest problems associated with database management and processing is finding and maintaining the meaning of stored data. If the data can be stored in documents using XML, the documents themselves will describe their data content. Of course, the DTD is a rudimentary vehicle for defining data semantics.

More recently, the XML Schema has been introduced to describe the structure of an XML document. An XML Schema has better support for applications, document structure, attributes, and data-typing. For example, here is the previous DTD transformed into an XML Schema:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="customer">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="first_name" type="xs:string"/>
        <xs:element name="middle_initial" type="xs:string"/>
        <xs:element name="last_name" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>

    <xs:complexType name="USAddress">
      <xs:sequence>
        <xs:element name="street" type="xs:string"/>
        <xs:element name="city"   type="xs:string"/>
        <xs:element name="state"  type="xs:string"/>
        <xs:element name="zip"    type="xs:decimal"/>
      </xs:sequence>
      <xs:attribute name="country" type="xs:NMTOKEN"
       fixed="US"/>
    </xs:complexType>

   </xs:element>

</xs:schema>
```

Given the benefits of XML Schema over DTDs, more XML documents are adopting them for use in modern XML applications.

NOTE

You can use Extensible Stylesheet Language (XSL) with XML to format XML data for display.

The important thing to remember about XML is that it solves a different problem than HTML. HTML is a markup language, but XML is a meta-language. In other words, XML is a language that generates other kinds of languages. The idea is to use XML to generate a language specifically tailored to each requirement you encounter. In short, XML enables designers to create their own customized tags, thereby enabling the definition, transmission, validation, and interpretation of data between applications and between organizations. So, the most important reason to learn XML is that it is quickly becoming the de facto standard for application interfaces.

There are, however, some issues with XML, the most troubling of which is market hype. There is plenty of confusion surrounding XML. Some believe that XML provides metadata where none currently exists, or that XML replaces SQL as a data access method for relational data. Neither of these assertions is true.

There is no way that any technology, XML included, can conjure up information that does not exist. People must create the metadata tags in XML for the data to be described. XML enables self-describing documents; it doesn't describe your data for you.

Moreover, XML doesn't perform the same functions as SQL. As a result, XML can't replace it. As the standard access method for relational data, SQL DML is used to "telling" a relational DBMS what data is to be retrieved. XML, on the other hand, is a document description language that describes the basic contents of data. An XML schema can be associated with an XML document to type XML data. XML might be useful for defining databases but not for accessing them.

DB2, as well as most of the other popular DBMS products, now provides built-in support for native XML. By integrating XML into DB2 databases, you can more directly and quickly access the XML documents, as well as search and store entire XML documents using SQL. Integration can involve simply storing XML in a large `VARCHAR` or `CLOB` column, breaking down the XML into multiple columns in one or more DB2 tables, or more commonly to store XML data natively within a column of a DB2 table. As of DB2 V9, you can store XML data natively in a DB2 using pureXML, which allows you to define columns with a data type of `XML`.

Storing Data Relationally Versus as XML

Before delving into a discussion of how to store and access XML data natively in DB2 for z/OS, first take a moment to contrast the traditional row and column data of DB2 (and other relational database systems) versus XML data.

The self-describing data format of XML enables complex data to be stored in a single document without giving up the ability to query, search, or aggregate the data. And the XML definition (DTD or XML schema) can be modified without requiring any changes to the database schema. Of course, this may be viewed as a pro by some and a con by others. Developers will likely enjoy the added flexibility, whereas DBAs will likely lament the lack of control.

The flexible nature of XML can require more resources (CPU and I/O) to examine and interpret the XML data as opposed to accessing the same data in a traditional row and column format. But the complexity of the schema must be taken into consideration.

XML is often more suitable for applications with complex and variable data structures, and for combining structured and unstructured information. Relational row and column data is most suitable for stable data structures. A complex hierarchy stored in XML, for example, may be queried more efficiently than a similar hierarchy stored in traditional DB2 form. Relational data, though, can offer more query flexibility and optimization choices.

Keeping an XML document intact in an XML column has the advantage of maximum flexibility but can negatively impact performance.

Fully shredding an XML document into a relational format has the advantage of making high volume transactional processes run faster. But shredding has several disadvantages, as well: Converting the XML documents to the equivalent relational model can consume a lot of resources; converting relational data back into an XML document is expensive; and it can be difficult to keep up with changing requirements.

A hybrid approach in which some portions of the document are maintained as XML and other portions are shredded into relational can be a best practice approach both for performance and maintainability.

Objects having sparse attributes are another area in which XML offers an advantage over a traditional relational data format. When a large number of attributes are possible but most are not used by every instance, XML may be a better choice because every attribute would need to be defined and stored with traditional relational data. In relational data, columns might be NULL, but in XML those data items are just not present.

Consider, for example, a product catalog where each product may have a different number of attributes: size, color, weight, length, height, material, style, watts, voltage, resolution, and so on, depending upon the product. A relational approach with one column per attribute would require a large number of columns requiring NULL, which is not ideal. Alternative approaches, such as a table per product or a three-column table that stores name/value pairs for each product ID are equally unappealing. With an XML solution, elements and attributes can be optional, so they would be omitted when they do not apply for a specific product.

Often there are XML schemas already defined by some industry standard organization (for example, ISO). Using those schemas can greatly reduce the time and effort for design and data modeling. Also these schemata are often used for defining the data received or sent by an application. It is often desirable to store XML data redundantly (which is against good relational design) in a relational column or even a subset of the document in an XML column. If schema validation, a relatively costly process, can be done once, then portions of the XML document can be used in other tables without revalidating.

Parsing and serializing XML data also can be a costly process. If XML data is required to be passed as output, it can be beneficial to store those portions of the document redundantly as XML documents. Using some hybrid design approach combining relational and XML can be useful: Highly referenced fields are best in relational columns, whereas sparsely populated or seldom-referenced fields may be better left in XML columns.

At any rate, XML data has its place, and DB2 users are lucky that they can store XML data natively within DB2 databases.

pureXML

V9 As of DB2 V9, XML data can be stored natively in DB2 databases. This implementation is known as pureXML. With pureXML, you can treat XML as another data type that can be managed by DB2. This means that you can CREATE tables with XML columns, ALTER existing tables to add XML columns, INSERT XML data (optionally validated against XML schemas), CREATE indexes on XML data, search XML data, and generally manipulate XML as part of your DB2 databases.

DB2's support for XML with pureXML is novel in that the XML data is integrated into the DB2 database system enabling access and management of the XML using DB2 functions and capabilities.

Creating a Table with an XML Column

Similar to LOB data, XML data is physically stored in separate table spaces from the base tables that contain the XML columns. But unlike with LOBs, the storage structures are transparent to the user. You do not need to define and manage the storage mechanisms used by DB2 for XML data.

For example, the following SQL creates a new table with an XML column, which can be used to store the customer XML example shown previously:

```
CREATE TABLE CUST
(CUSTNO   INTEGER NOT NULL,
 STATUS   CHAR(1),
 XMLCUST  XML)
IN DB.TS;
```

When a table is defined with an XML column, DB2 generates a hidden column in the base table called a `DOCID`, which contains a unique identifier for the XML column. There is a single `DOCID` column even if the table contains more than one XML column. The `DOCID` column is purely internal; it does not show up in a `SELECT *`, for example. In addition, DB2 automatically creates a document ID index for the XML column, too.

The XML data is not stored directly in the XML column. Instead, a separate internal table in a separate table space is created for each XML column in the base table. A single XML document may be physically split across multiple rows in the XML table, but it logically still belongs to a single row in the base table, which is the table created with the XML column definition. The internal XML table is composed of three columns:

- `DOCID (BIGINT)`
- `MIN_NODEID (VARCHAR)`
- `XMLDATA (VARBINARY)`

The `DOCID` column is used as a pointer between the base table and the XML table. In any given row, the `MIN_NODEID` column stores the lowest node stored in the `XMLDATA` column of the row. This information optimizes DB2's capability to process XML documents. The `XMLDATA` column contains a region of an XML document formatted as a parsed tree. The internal table is clustered by `DOCID` and `MIN_NODEID`.

NOTE

An XML node is the smallest unit of a valid, complete structure in a document. For example, a node can represent an element, an attribute, or a text string.

An XML node ID is an identifier for XML nodes and facilitates navigation among multiple XML data rows in the same document.

The internal XML table always has a 16-KB page size. This is so regardless of the page size of the base table containing the XML column. The table space used by the internal XML table is a Universal table space. If the base table resides in a partitioned table space, the XML table space will be range-partitioned; if not, the XML table space will be partition-by-growth.

NOTE

The internal XML table space inherits the `COMPRESS` parameter specification from the base table space. Of course, the `COMPRESS` attribute for an XML internal table space can be altered if wanted.

The TBSBPXML DSNZPARM system parameter is available to specify the default buffer pool for XML table spaces. The default is BP16K0; however, because the DB2 Catalog uses this buffer pool, it is a good idea to use different 16-K buffer pools for your XML table spaces.

CAUTION

Be sure to GRANT USE authority to the buffer pool to be used for XML table spaces to any DBA (or user) who must CREATE tables with XML columns.

XML Document Trees

DB2's pureXML implementation follows the XPath 2.0 and the XQuery 1.0 data model, which provides an abstract representation of XML documents. By using the data model, all permissible values of expressions in XPath can be defined, including values used during intermediate calculations. The pureXML data model is described in terms of sequences and items, atomic values, and nodes.

The primary difference between XPath and XQuery is that XPath is a subset of XQuery for addressing parts of a document. XPath cannot be used to construct new XML documents, nor does it support complex join, grouping, and ordering.

XML is hierarchic in nature and, as such, every XML document can be represented as a node tree. When XML data is queried or modified, the hierarchical structure of the document must be traversed. To assure that XML data is accessed as efficiently as possible, DB2 physically stores XML documents in hierarchical format, as trees of nodes with parent-child relationships between the nodes.

To better understand this concept of hierarchic trees, work through an example. To start, examine the following sample XML:

```
<customer>
   <custname>
       <first_name>Craig</first_name>
       <last_name>Mullins</last_name>
   </custname>
   <addr country="US">
       <street>100 Easy St</street>
       <city>Pittsburgh</city>
       <state>PA</state>
       <zip_code>15215</zip_code>
   </addr>
   <phone type="work">412-555-1000</phone>
   <phone type="mobile">972-555-8174</phone>
</customer>
```

This basic XML document contains customer data. At the root of the tree is the root element, customer. There are various direct children elements as well: first_name, last_name, addr, and two occurrences of phone.

The element addr is composed of multiple elements as well: street, city, state, and zip_code. And addr also has an attribute for the country. The element phone has an attribute, type, associated with it, as well.

Figure 10.1 illustrates a representation of this XML document as a hierarchical tree. You can build these trees by parsing an XML document using an XML parser.

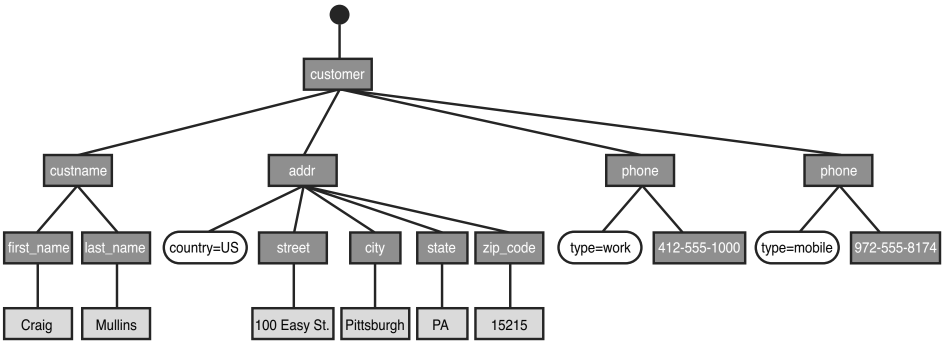


FIGURE 10.1 An XML document tree.

XML data is stored in DB2 as an XML document tree. When XML is inserted or loaded into an XML column, DB2 parses the XML document to produce the hierarchical format, and that is what is stored in the XML table space. If one or more columns are of type XML, and you select the entire XML column, DB2 retrieves the XML document tree and converts it back into the text XML document. This process is called *serialization*.

When the XML data is larger than a page, the document tree is divided into groups of nodes; each group is usually a subtree of nodes. The divided data is stored in the `XMLDATA` column of the internal XML table. Nodes are grouped bottom up, up to the largest row size of a page. One document can be physically stored across many rows and pages.

DB2 stores XML data as an XML document tree to optimize performance when accessing XML data. The document tree is further optimized by replacing tag names with 4-byte identifiers. So the internal storage does not look exactly like the tree shown in Figure 10.1. Tag names are mapped to `stringIDs` and stored in the DB2 Catalog in the table `SYSIBM.SYSXMLSTRINGS`. DB2 caches the mapping table to optimize performance. Furthermore, XML query evaluation and traversal of XML documents can operate on integers, which is typically much faster than operating on strings. There will be only one 4-byte identifier for a particular string, even if that string (that is, “name”) is used in many different XML columns.

The exact shape of the document tree can vary from document to document; it depends upon the content of each XML document. The document tree is not built or predefined based on the XML Schema. The actual XML data is stored as `VARBINARY` data, composed of the XML document tree (or a sequence of subtrees) with context path.

Serializing and Parsing XML Data

An XML value can be transformed into a textual XML value that represents the XML document by using the `XMLSERIALIZE` function or by retrieving the value into an application variable of an XML, string, or binary data type.

The inverse can also be achieved. The textual XML document can be turned into the XML value using the `XMLPARSE` function or by storing a value from a string, binary, or XML data type to an XML column.

Altering a Table to Contain XML Data

Modifying a table to add an XML column is also a simple endeavor. All that is necessary is a simple `ALTER`, such as the following:

```
ALTER TABLE DSN81010.EMP
  ADD JOBHISTORY XML;
```

This statement adds the `JOBHISTORY` column as XML to the `EMP` table. Of course, when an XML column is added to an existing table, DB2 creates the same internal table and index to support the XML data as it would for a new table.

Schema Validation

As you have seen, XML is flexible. XML has no rigorously enforced schema applied when data is added, such as with a DBMS such as DB2. But you can validate XML against a schema with a process known as XML schema validation. XML schema validation can determine whether the structure, content, and data types of an XML document are valid according to an XML schema. In addition, XML schema validation removes whitespace that can be ignored from the XML document being validated.

You can validate an XML document in DB2 in two ways:

- Automatically, by specifying an XML type modifier in the XML column definition of the `CREATE` or `ALTER TABLE` statement. When a column has an XML type modifier, DB2 implicitly validates documents that are inserted into the column or documents in the column that are updated.
- Manually, by executing the `DSN_XMLVALIDATE` built-in function when you `INSERT` a document into an XML column or `UPDATE` a document in an XML column.

If you perform XML schema validation using DB2, you need to set up an XML Schema Repository. A DB2 for z/OS XML schema repository (XSR) is a set of DB2 tables where you can store XML schemas. XSR requires additional software: WLM, z/OS XML System Services, Java 2 Technology Edition 31 bit (V5 or later), and IBM Data Server Driver for JDBC and SQLJ.

Refer to the IBM manual, *pureXML Guide* (SC19-2981) for additional information on using an XML Schema Repository, as well as additional information on using type modifiers.

NOTE

An XML schema is not required when using DB2 and pureXML. XML columns can store any well-formed documents. You can also validate XML documents against many schemas, if needed.

XML Namespaces

XML namespaces are a W3C XML standard for providing uniquely named elements and attributes in an XML document. XML documents may contain elements and attributes that have the same name but belong to different domains. A namespace can be used to remove the ambiguity when referencing such elements and attributes.

Namespaces are supported by all DB2 pureXML features including SQL/XML, XML indexes, and XML Schema validation.

A namespace must be declared with a prefix assigned to a Universal Resource Identifier (URI) before it can be used. Now augment your example XML with a namespace:

```
<c:customer xmlns:c="http://ddgsample.org">
   <c:custname>
       <c:first_name>Craig</c:first_name>
       <c:last_name>Mullins</c:last_name>
   </c:custname>
   <c:addr country="US">
       <c:street>100 Easy St</c:street>
       <c:city>Pittsburgh</c:city>
       <c:state>PA</c:state>
       <c:zip_code>15215</c:zip_code>
   </c:addr>
   <c:phone type="work">412-555-1000</c:phone>
   <c:phone type="mobile">972-555-8174</c:phone>
</c:customer>
```

The attribute `xmlns:c` declares that `c` is a namespace prefix bound to the URI `http://ddgsample.org`. The prefix `c` can be used for the `customer` element and all other elements or attributes in the document that are descendants of `customer`.

CAUTION

Be careful when specifying namespace URIs. There is no requirement that a namespace be a valid URI. For example, URIs with blanks are not valid, but they do not affect whether an XML document is well-formed. You can insert XML documents with spaces in their namespace URIs, but URIs with spaces cannot be declared in a query. So take care to specify valid namespace URIs.

Not every node in a document must belong to the same namespace. Furthermore, an XML document can contain multiple namespaces. Specify namespace declarations as needed within your XML documents according to your needs for accessing elements of the XML document.

If all elements in a document belong to the same namespace, you can declare a default namespace and avoid the use of prefixes:


```
<customer xmlns="http://ddgsample.org">
   <custname>
       <first_name>Craig</first_name>
       <last_name>Mullins</last_name>
   </custname>
   <addr country="US">
       <street>100 Easy St</street>
```

```
        <city>Pittsburgh</city>
        <state>PA</state>
        <zip_code>15215</zip_code>
    </addr>
    <phone type="work">412-555-1000</phone>
    <phone type="mobile">972-555-8174</phone>
</customer>
```

In this case, all elements belong to the declared `http://ddgsample.org` namespace.

Indexing XML Data

You can build indexes on data stored in XML columns to improve the efficiency of queries against XML documents. Indexes on XML data differ from DB2 indexes on relational data. A typical DB2 index lists a column, or series of columns, and the index is built upon those columns. An XML index is based upon a part of the data in the XML column, not the entire column.

A typical DB2 index has one entry for each row in the table, even if the value is NULL, whereas an XML index does not have an entry for a row if the XML document in that row does not contain that element.

An XML index uses an XML pattern expression to index paths and values in XML documents stored within a single XML column. The index entries in XML indexes provide access to nodes within the XML document. Because multiple parts of a XML document can satisfy an XML pattern, DB2 might generate multiple index keys when it inserts values for a single document into the index.

XML indexes are built using the `CREATE INDEX` statement (and dropped using `DROP INDEX`). Instead of listing columns, the `GENERATE KEY USING XMLPATTERN` clause is used to indicate what portion of the XML document you want to index:

```
CREATE INDEX CSTLNMX1
  ON   CUST(XMLCUST)
GENERATE KEY USING XMLPATTERN '/customerinfo/custname/last_name'
AS SQL VARCHAR(20)
```

The `GENERATE KEY USING XMLPATTERN` clause provides information about what you want to index. This clause is called an XML index specification. The XML index specification contains an XML pattern clause. The XML pattern clause in this example indicates that you want to index the values of the `last_name` attribute of each `customer` element. The index entries are to be stored as `VARCHAR(20)`.

Every XML pattern expression specified in index `CREATE` statement must be associated with a data type. The only supported data types are `VARCHAR`, `DECFLOAT`, `DATE`, and `TIMESTAMP`.

NOTE

V10 DATE and TIMESTAMP support was added for DB2 V10.

Only one index specification clause is allowed in each `CREATE INDEX` statement, but it is permissible to create multiple XML indexes on each XML column.

To identify the portion of the XML to be indexed, you specify an XML pattern to identify a set of nodes in the XML document. This pattern expression is similar to an XPath expression. (But only a subset of the XPath language is supported.)

CAUTION

If you validate your XML documents against an XML schema, be sure to verify that the data type specifications in the XML schema match the data types used in your indexes.

Namespace Declarations in XML Indexes

In the `XMLPATTERN` clause of the `CREATE INDEX` statement, you can specify an optional namespace declaration that maps a URI to a namespace prefix. Then you can use the namespace prefix when you refer to element and attribute names in the XML pattern expression.

For example, consider the following index creation statement:

```
CREATE INDEX CSTPHNX2
  ON   CUST(XMLCUST)
GENERATE KEY USING XMLPATTERN 'declare namespace s="http://ddgsample.org/";
/s:customer/s:phone/@s:type' AS SQL VARCHAR(12)
```

The `namespace` declaration maps the namespace URI `http://ddgsample.org` to the character `s`. It can then be used to qualify all elements and attributes with that namespace prefix.

XML Indexes and `UNIQUE`

Specifying `UNIQUE` in an XML index definition is a little bit different than in traditional, relational indexes. For a traditional index, the `UNIQUE` keyword enforces uniqueness across all rows in the table. For XML indexes, the `UNIQUE` keyword enforces uniqueness across all documents in an XML column. This means the index can ensure uniqueness. not only across all rows in the table, but also within a single document within a row.

For an XML index, DB2 enforces uniqueness for the following:

- Data type of the index.
- XML path to a node.
- Value of the node after the XML value has been cast to the SQL data type specified for the index.

CAUTION

Because rounding can occur during conversion of an index key value to the specified data type for the index, multiple values that appear to be unique in the XML document might result in duplicate key errors.

Querying XML Data

Of course, after you create tables with XML data in them, you want to access that data in your DB2 applications. Doing so requires a basic understanding of XPath.

XPath

XPath is a programming language designed by the World Wide Web Consortium (W3C) for querying and modifying XML data. DB2 supports a subset of the language constructs in the XPath 2.0 specification. In XPath, there are seven kinds of nodes: element, attribute, text, namespace, processing-instruction, comment, and document nodes.

XML documents are treated as trees of nodes. The top element of the tree is called the root element. Now look at some sample XML again:

```
<customer>
   <custname>
       <first_name>Craig</first_name>
       <last_name>Mullins</last_name>
   </custname>
   <addr country="US">
       <street>100 Easy St</street>
       <city>Pittsburgh</city>
       <state>PA</state>
       <zip_code>15215</zip_code>
   </addr>
   <phone type="work">412-555-1000</phone>
   <phone type="mobile">972-555-8174</phone>
</customer>
```

Examples of nodes in this XML document include `<customer>`, which is the root element node, `<city>Pittsburgh</city>`, which is an element node, and `Type="work"`, which is an attribute node.

There is a relationship among the nodes that you need to understand for XML processing. Each element and attribute has one parent. In the example, XML the `addr` element is the parent of `street`, `city`, `state`, and `zip_code`. Element nodes can have zero, one, or more children. Nodes that have the same parent are siblings. The `street`, `city`, `state`, and `zip_code` nodes are siblings of each other, as well as children of `addr`.

Descendants are any children of a node as well as the children's children, and so on. So in your example, the descendants of customer include `custname`, `first_name`, `last_name`, `addr`, `street`, `city`, `state`, `zip_code`, and `phone`.

XPath uses path expressions to select nodes or node-sets in an XML document. The node is selected by following a path or steps. If you are familiar with Windows® or UNIX® file structure, an XPath specification will look familiar to you. Table 10.1 outlines some useful path expressions.

TABLE 10.1 XPath Expressions

Expression	Description
Nodename	Selects all child nodes of the named node
/	Selects from the root node
//	Selects nodes in the document from the current node that match the selection no matter where they are
.	Selects the current node
..	Selects the parent of the current node
@	Selects attributes

To select the `last_name` element of the `customer` XML document using XPath you would specify the following:

```
customer/addr/last_name,
```

Or to select all attributes that are named phone using XPath, you can specify the following:

```
//@phone
```

Use DB2 XPath in the following contexts:

- As an argument to the `XMLQUERY` built-in function, which extracts data from an XML column.
- As an argument to the `XMLEXISTS` predicate, which is used for evaluation of data in an XML column.
- In an XML index, to determine the nodes in an XML document to index. (Only a subset of XPath, called an XML pattern, is valid in this context.)

The `XMLQUERY` Function

You can use the `XMLQUERY` function to execute an XPath expression from within SQL. You can pass variables to the XPath expression specified in `XMLQUERY`. `XMLQUERY` returns an XML value, which is an XML sequence. This sequence can be empty or can contain one or more items.

When you execute XPath expressions from within an `XMLQUERY` function, you can allow XML data to participate in SQL queries. Furthermore, you can retrieve parts of XML documents, instead of the entire XML document. This gives you the ability to operate on both relational and XML data in the same SQL statement. And you can apply additional SQL to the returned XML values after using `XMLCAST` to cast the results to a non-XML type.

CAUTION

XPath is case-sensitive. The case of any variables you specify in an `XMLQUERY` function must match the XPath expression.

Consider the following query, which returns the phone number information extracted from the XML for each customer in the CUST table:

```
SELECT CUSTNO,
  XMLQUERY ('declare default element namespace "http://ddgsample.org";
  /customer/phone' passing XMLCUST)
  AS "PHONE FROM XMLCUST"
FROM   CUST
```

You can use the XMLEXISTS predicate to restrict the result set based on values in XML columns. To do so, an XPath expression is specified to the XMLEXISTS predicate. If the XPath expression returns an empty sequence, the value of the XMLEXISTS predicate is false. Otherwise, XMLEXISTS returns true and the row is returned.

For example, consider this SQL statement:

```
SELECT CUSTNO, STATUS, XMLCUST
FROM   CUST
WHERE  XMLEXISTS ('declare default element namespace "http://ddgsample.org";
  //addr[city="Pittsburgh"]' passing XMLCUST)
AND XMLEXISTS ('declare default element namespace "http://ddgsample.org";
/customer[last_name="Mullins"]' passing XMLCUST)
```

This matches your sample XML document in which customer `city` is `Pittsburgh` and `last_name` is `Mullins`, so that row will be returned (as would any other where both of these conditions are true).

The `XMLTABLE()` Function

The `XMLTABLE()` function though can be used to produce XML query results comparable to XQuery. The following example uses the `XMLTABLE()` function to query your sample table and XML document, returning the `city`, `street`, `state`, and `zip_code` as columns in a result table:

```
SELECT X.*
FROM   CUST,
       XMLTABLE (XMLNAMESPACES(DEFAULT 'http://ddgsample.org'),
         '$x/customer/addr[@zip_code=15215]'
         PASSING XMLCUST as "x"
       COLUMNS
         ZIP_CODE INT         PATH '@zip_code',
         STREET   VARCHAR(50) PATH 'street',
         CITY     VARCHAR(30) PATH 'city',
         STATE    VARCHAR(2)  PATH 'state') AS X
```

Recall that the XML column in the `CUST` table is named `XMLCUST`, therefore you code `PASSING XMLCUST`.

To enable this query to use an XML index, can add a predicate using `XMLEXISTS`, for example:

```
WHERE XMLEXISTS('$x/customer/addr[@zip=15215]'PASSING XMLCUST AS "x")
```

The XMLTABLE() function can be used when converting XML data into a relational result set. It can be beneficial to use XMLTABLE() in views where the SQL referencing the view does not have to code XPath expressions. The optimizer can still use XML indexes when view is referenced.

Access Methods

DB2 supports several access methods for XML data. The basic access method is the DocScan, which traverses XML data and evaluates XPath expressions using an IBM-patented technique called QuickXScan (see Note). There is no access type indicator for DocScan in the PLAN_TABLE because it is part of a scan if there is a predicate on an XML column involved.

NOTE

QuickXScan, an industrial strength streaming XPath algorithm implemented for native XML support in DB2 for z/OS. QuickXScan evaluates XPath expressions with predicates by one sequential scan of XML data with high efficiency. Because QuickXScan does not rely on relational techniques, it is well-suited for Internet applications that favor an efficient streaming algorithm.

XML indexes are used only for the XMLEXISTS predicate and XMLTABLE function evaluation. There are three access types for XML index-based access. Similar to RID list access, ANDing, and ORing, they include the following:

- DocID list access (DX)
- DocID list ANDing (DI for DocID list Intersection)
- DocID list ORing (DU for DocID list Union)

Inserting XML Data

Inserting XML data is as simple as inserting any other type of data. For example, assume that you want to use the CUSTOMER table described earlier in this chapter to INSERT an XML document. The following SQL achieves this goal:

```
INSERT INTO CUSTOMER
  (CUSTNO, STATUS, XMLCUST)
VALUES (1000, 'Y',
'<customer>
   <custname>
       <first_name>Craig</first_name>
       <last_name>Mullins</last_name>
   </custname>
   <addr country="US">
       <street>100 Easy St</street>
       <city>Pittsburgh</city>
       <state>PA</state>
       <zip_code>15215</zip_code>
   </addr>
   <phone type="work">412-555-1000</phone>
```

```
   <phone type="mobile">972-555-8174</phone>
</customer>'
);
```

This SQL statement inserts a new customer into the CUSTOMER table, giving it a CUSTNO of 1000 and a STATUS of Y. The XML data is inserted into the XML column defined as XMLCUST.

Deleting XML Data

You can use the SQL DELETE statement to delete rows that contain XML documents. Nothing special is required, just code the DELETE as normal including any WHERE clause that you want. When the row is deleted, the XML data for that row is deleted, too.

Of course, you might want to DELETE based upon values in the XML document. You can do this using XPath expressions within XMLEXISTS, for example, to DELETE rows from CUST table for which the value of the city element is Pittsburgh (in the XMLCUST column):

```
DELETE FROM CUST
WHERE XMLEXISTS ('declare default element namespace "http://ddgsample.org";
//addr[city="Pittsburgh"]' passing XMLCUST)
```

Updating XML Data

You can also update XML data, either the entire XML document, or a portion of the XML document.

Updating an Entire XML Document

To update an entire XML document in an XML column, supply the XML data to the UPDATE statement, being sure to specify a WHERE clause for the rows to be updated. The input to the XML column must be a well-formed XML document (as defined in the XML 1.0 specification).

> **NOTE**
>
> When you update an XML column, you might also want to validate the input XML document against a registered XML schema.

Updating a Portion of an XML Document

V10 You can also use the UPDATE statement with the XMLMODIFY function to update a portion of an XML document in an XML column. The XMLMODIFY function specifies a basic updating expression that you can use to insert nodes, delete nodes, replace nodes, or replace the values of a node in XML documents stored in XML columns.

> **CAUTION**
>
> Before you can use XMLMODIFY to UPDATE part of an XML document, the column containing the XML document must support XML versions.

The following UPDATE statement modifies the document for CUSTNO 100 changing the value of the city element to “Houston”:

```
UPDATE CUST
SET XMLCUST=XMLMODIFY('replace value of node /customer/addr/city with "Houston" ' )
WHERE CUSTNO = 100
```

The following update expressions are supported in the XMLMODIFY function:

- **delete expressions**: To remove elements or attributes from a document
- **insert expressions**: To add elements or attributes to a document
- **replace value of node expressions**: To change the value of an element or attribute
- **replace node expressions**: To replace an existing element or attribute with a different one

XML-DB2 Guidelines

Consider the following guidelines as you embark on using XML with your DB2 databases.

Learn All You Can About XML Before you begin to mix XML and DB2, be sure that you have a solid grasp of XML. The short introduction in this chapter is merely the tip of the iceberg. You must understand that XML is hierarchical and, as such, cannot match up exactly with your relational, DB2 way of thinking and processing data.

For in-depth coverage of pureXML support in DB2 for z/OS, refer to the *IBM DB2 for z/OS pureXML Guide* (SC19-2981) and the IBM RedBook *Extremely pureXML in DB2 10 for z/OS* (SG24-7915).

Consider augmenting the information in the pureXML manual with additional sources. The DB2 for z/OS pureXML section of IBM developerWorks® website at http://www.ibm.com/developerworks/wikis/display/db2xml/DB2+for+zOS+pureXML contains many useful examples and articles. Another useful reference is the IBM Press book titled *DB2 pureXML Cookbook*, which covers pureXML for both DB2 for LUW and DB2 for z/OS. You can find additional information on XML at the following websites:

http://www.oasis-open.org
http://www.w3schools.com
http://www.xml.org

Find XMLEXISTS Predicates for Indexing Because XML indexes are used only on the XMLEXISTS predicate, it is a good idea to find the predicates within XMLEXISTS clauses before doing any XML indexing. Look for predicates such as [@id = xxx] or [price > 100.00].

Favor Creating Lean XML Indexes Assume your queries often search for customer documents by last_name. In that case, an index on the last_name element can improve the performance of such queries, for example:

```
CREATE INDEX CUSTLNX1
 ON CUST(XMLCUST)
generate key using xmlpattern '/customer/custname/last_name' as sql varchar(20);
```

Use Caution Before Indexing Everything As a general rule of thumb, avoid indexing everything (also known as a heavy index) because it is costly to maintain during `INSERT`, `UPDATE`, and `DELETE` processing. An additional concern is that a heavy index requires a lot of storage, which might be better used for more targeted indexes. For example, consider the following heavy index:

```
CREATE INDEX HEAVYIX
  ON CUST(XMLCUST)
generate key using xmlpattern '//*' as sql varchar(100);
```

When using `xmlpattern "//*"` to create an XML index, the generated index key value could contain entries from every text node in every XML document in the XML column. Due to the creation and maintenance overhead, avoid such heavy indexes.

An exception to avoiding heavy indexes might be made for applications with low write activity and an unpredictable query workload making specific indexes hard to anticipate and define.

Favor XPath Expressions with Fully Specified Paths Avoid using `*` and `//` in your path expressions; instead, use fully specified paths whenever possible. For example, assume that you need to retrieve customers' ZIP codes. There are multiple path expressions you could code to get to the appropriate data. Both `/customer/addr/zip_code` and `/customer/*/zip_code` return the ZIP code. But for optimal performance, the fully specified path should be preferred over using `*` or `//` because it enables DB2 to navigate directly to the wanted elements, skipping over non-relevant parts of the document.

CAUTION

Sometimes using * and `//` can lead to unwanted query results. For example, if some of the `customer` documents also contained `spouse` information, the path `/customer/*/zip_code` could return the ZIP code of both the customer and their spouse. This may, or may not, be the intent of the query, so be careful when using * and //.

Use `RUNSTATS` to Gather Statistics on XML Data and Indexes The `RUNSTATS` utility has been extended to collect statistics on XML data and XML indexes. The DB2 Optimizer uses these statistics to generate efficient execution plans for SQL/XML queries. Thus, continue to use `RUNSTATS` as you would for relational data. Simply stated, DB2 generates better access plans if XML statistics are available.

NOTE

`RUNSTATS TABLESPACE` does not collect histogram statistics for XML table spaces; `RUNSTATS INDEX` does not collect histogram statistics for XML node ID indexes or XML indexes.

Use `CHECK DATA` Consider running the `CHECK DATA` utility periodically to check the consistency between the XML document data and its associated XML schema and its XML index data.

Use `REPORT TABLESPACE SET` Use the `REPORT TABLESPACE SET` utility to identify the underlying XML objects that are automatically created.

V10 **Consider Deferring the Creation of XML Table Spaces** As of DB2 V10 you can defer the actual physical creation of XML table spaces and their associated indexes to optimize your space management requirements.

By specifying `DEFINE(NO)`, the underlying VSAM data sets are not created until the first `INSERT` or `LOAD` operation. The undefined XML table spaces and dependent index spaces are registered in the DB2 Catalog but are considered empty when access is attempted before data is inserted or loaded.

DSNZPARMS: `XMLVALA` and `XMLVALS` The `XMLVALA` subsystem parameter specifies an upper limit for the amount of storage that each user is to have available for storing XML values. The default is 200 MB. DB2 performs streaming, so you might be able to insert and select XML documents larger than the limit. However, it is a good idea to check the value and set an appropriate value based on your expected XML processing needs.

If you construct XML documents, set `XMLVALA` to at least twice the maximum length of documents generated. If you query XML data, set `XMLVALA` at least four times the maximum document size.

`XMLVALS` is the virtual storage limit allowed for XML processing for the DB2 subsystem. The default value is 10 GB.

Summary

This chapter is not intended to be a comprehensive treatment of DB2's XML support. After reading this chapter you should understand the basics of XML, how it differs from traditional relational data, and how DB2 for z/OS supports integrated XML data with pureXML.

Index

Numbers

3GLs (third-generation languages), 486

4GLs (fourth-generation languages), 486

tools, 1419-1420

4KB pages, 797

A

abbreviations, standardized, 252

ABEND command, 1360

abends, 1360

UDFs (user-defined functions), 184

ABSVAL function, 141

Academic Initiative System, 1443

access control

columns, 466-469

LBAC (label-based access control), 461-464

referential integrity, 465

restrictions, 466

row level granularity, 462-464

rows, 466-469

access guidelines, SQL (Structured Query Language), 58-90

access methods, XML (Extensible Markup Language), 423

access paths, 5

analyzing, 1116-1118

changing, 88-89

influencing, 46-48

managing, 1013

EXPLAIN, 980-1002, 1006

IBM Data Studio, 1012

optimization, 1130-1135

reviewing, 1018-1019

SQL statements, 980

stability, 654

strategies, 846-860

single table, 824-845

subqueries, optimization, 869-871

ACCESSCREATOR column (PLAN_TABLE), 987

accessing

data warehouses, 1519

distributed data, 1460-1465

LOB columns, 399-402

ACCESSNAME column (PLAN_TABLE), 987
ACCESSTYPE column (PLAN_TABLE), 987
ACCESS_DEGREE column (PLAN_TABLE), 989
ACCESS_PGROUP_ID column (PLAN_TABLE), 989
Accounting Report Additional Information listing (24.7), 951-952
Accounting Report Buffer Pool Information listing (24.9), 953-954
Accounting Report Database Code Usage listing (24.8), 952
Accounting Report Locking Activity listing (24.5), 950-951
Accounting Report Program Terminations listing (24.6), 951
Accounting Report RID List Details listing (24.10), 956
accounting reports, 943-945
- long, 946-956
  - buffer pool information, 952-955
  - database code usage information, 952
  - locking activity, 950-951
  - program status, 951
  - SQL activity, 949-950
- trace, 956

accounting trace, 930-931
ACOS function, 141
ACTION parameter (BIND), 639
active database constructs, application development, 530
ad hoc SQL, 12
Adamson, Christopher, 1513
ADD MONTHS function, 141
address spaces, 810
administration
- commands, 1353-1358
- data, 1432-1433
- data sharing, 783-791
- packages, 624-626

administrative commands, 1353-1358
ADMIN_TASK_LIST function, 159
ADMIN_TASK_STATUS function, 159
Advanced Program-to-Program Communication (APPC), DRDA (Distributed Relational Database Architecture), 1452
AFTER triggers, 379, 390
aggregate functions, 34, 135-136
- AVG, 136
- CORRELATION, 137
- COUNT BIG, 138
- COUNT, 137-138
- COVARIANCE, 138-139
- COVARIANCE SAMP, 138-139
- MAX, 139
- MIN, 139-140
- searching results, 165
- STDDEV, 140
- SUM, 140
- VARIANCE, 141
- VARIANCE SAMP, 141

ALIAS database objects, 38
aliases, 201, 313
- databases, 1471-1472
- server location, 1472

ALT tools, 1396-1398
ALTER BUFFERPOOL command, 1355
ALTER TABLE statement, 1532
ALTER UTILITY command, 1282, 1355
analysis
- access paths, 1116-1118
- distributed throughput, 1491-1493

analysis tools, queries, 64-65
ANSI SQL reserved words, 504
APPC (Advanced Program-to-Program Communication), DRDA (Distributed Relational Database Architecture), 1452
applets. Java, 555-557
Application Assist Processor (zAAP), 1076

application development, 486-487, 527-536
active database constructs, 530
batch programming, 536, 546-547
clustered access, 536
COMMIT statements, 539-541
lock strategies, 538-539
LOCK TABLE command, 538
parallel processing, 536-538
restartable, 543-546
SAFEPOINTs, 542-543
units of work, 541-542
"black boxes," 528-529
code modular, 529
cursors, 511-513
data modifications, 513-515
rowset positioning, 518-520
scrollable, 515-518
SELECT statement, 521-525
data filtering, 531
data sharing, 787-788
dynamic SQL, 567-569
classes, 576-589, 594
versus static, 569-576
embedded SQL statements, 487-490
data modification, 525-527
delimiting, 490
GET DIAGNOSTICS, 494, 497
SQLCA (SQL Communication Area), 491-493
SQLCODE, 493
SQLSTATE, 493
WHENEVER, 500-501
error handling, standardizing, 497-499
host variables, 504-506, 509-511
host structures, 506
null indicators, 507-509
Java, 554-563
naming conventions, 501-504
online programming, 547-552
program preparation, 601, 609, 632-655
background, 632
batch procedures, 616-618
BIND, 637-654
BIND command, 605-606
CICS processors, 632
CLIST, 618-619
collections, 628-629
compiling programs, 606-607
converting DBRM-based plans, 630-631
DB2I, 609-616
Declarations Generator, 601-604, 632-634
default names, 632
Java, 607
linkage editor, 655
linking programs, 607
multiple methods, 619-622
objects, 631-632
packages, 623-627
plans, 622
precompiling, 634-637
precompiling programs, 604-605
REXX EXEC, 618-619
running programs, 608
versions, 629-630
REXX, 563-565
SQL (Structured Query Language), 565-566
coding, 552-554
stored procedures, 530
unqualified SQL (Structured Query Language), 530
user-defined functions, 530
application development guide, 1434
application efficiency queries, Catalog, 1041-1042
application I/O, 1078-1079
application requester (AR) function (DRDA), 1450

application server (AS) function (DRDA), 1450

application-directed data access, 1461-1462

application-level changes, 57

applications

- batch, 705-706
- data sharing, impact, 776-778
- executing, 704-706
- Internet, 695
  - network traffic minimization, 695
- Java, 555-557
- online, 705-706
- rebinding, 1014-1016
  - best practices, 1016-1018
  - gathering statistics, 1019-1020
  - periodic maintenance, 1015
  - regular maintenance, 1015
  - reviewing access paths, 1018-1019
  - system maintenance, 1015-1016
- tuning, 1116-1137

APPLNAME column (DSN_STATEMENT_TABLE), 999

APPLNAME column (PLAN_TABLE), 986

AR (application requester) function (DRDA), 1450

ARC tools, 1398

architectures, DRDA (Distributed Relational Database Architecture), 1451-1452

ARCHIVE LOG command, 1355

archives, mailing lists, 700

arithmetic

- columns, minimizing, 80
- date and time, 84-85, 122-123
- precision, 78-80

arithmetic expressions, 78-80

AS (application server) function (DRDA), 1450

ASCII function, 142

ASENSITIVE scrollable cursors, 516

ASIN function, 142

assigning values, columns and UDTs (user-defined data types), 196

ASUTIME column (RLST), 1146

ATAN function, 142

Attach Facility

- CAF (Call Attach Facility), 763-765
  - benefits, 766
  - creating threads, 764-766
  - drawbacks, 766
  - vendor tools, 766-767
- CICS (Customer Information Control System), 730-731
- IMS, 752-755
- RRSAF (Recoverable Resource Manager Services Attach Facility), 767-768

attachment interfaces, data sharing, impact, 776-777

attribute copy options, MQTs (Materialized Query Tables), 1526

AUD tools, 1399-1400

audit reports, 956-957

Audit Summary Report listing (24.11), 956-957

audit trace, 477-479, 931-932

auditing, 476-481

- policies, 479-481
- tables, forcing, 267
- trace-based, 1400

auditing tools, 1399-1400

AUTHID column (RLST), 1146

AUTHID traces, 971

authids, translation, 1501-1502

authorization, 448, 453-461

- BINDAGENT, 456
- DISPLAY, 453
- group-level, 41
- IDs, 448-449
- privileges
  - granting and revoking, 449-450
  - group, 451-452
- PUBLIC access, 452-453

queries, Catalog, 1044-1046
repeating, 453
SECADM, 450-451, 455
SELECT, 453
synonyms, 458
SYSADM, 455, 459-460
auto rebind, 64
automatic query rewrite, MQTs (Materialized Query Tables), 1528-1532
AUTOSIZE parameter (buffer pool), 1105-1106
AUXERROR parameter, CHECK_DATA utility, 1181
auxiliary tables, indexing, 351
AUXONLY option (SCOPE parameter), 1180
availability, resources, 769
AVG function, 136
avoidance, locks, 908-911

B

backup and recovery
contingency planning, 1390-1392
disaster recovery technologies, 1387-1388
DSN1COPY strategy, 1384-1385
environmental considerations, 1388-1390
FlashCopy strategy, 1385-1387
Scalpel strategy, 1381-1384
Sledgehammer strategy, 1380-1381
backup and recovery utilities, 1201-1202
BACKUP_SYSTEM, 1236-1238
COPY, 1202-1215
copying index spaces, 1204-1205
copying table spaces, 1203-1204
full image copy, 1202-1203
incremental image copy, 1202-1203
phases, 1205-1206
COPYTOCOPY, 1215-1218
execution, 1216-1217
phases, 1216
MERGECOPY, 1218-1220
phases, 1219
QUIESCE, 1220-1223
phases, 1222
REBUILD_INDEX, 1232-1235
index versions, 1234
phases, 1234
RECOVER, 1224-1232
phases, 1228
recover table spaces, 1226-1227
RECOVER_INDEX, 1228
REPAIR, 1235
REPORT_RECOVERY, 1235-1236
RESTORE_SYSTEM, 1238-1239
BACKUP SYSTEM utility, 1387
backups, data sharing, 784
BACKUP_SYSTEM utility, 1236-1237
phases, 1237-1238
Basic Row Format (BRF), 799, 1081
batch applications, 705-706
Batch JCL for a TSO/DB2 Program listing (18.1), 708-709
batch performance monitoring, 940-943
batch procedures, program preparation, 616-618
batch processing, 11
batch programming, 536, 546-547
clustered access, 536
COMMIT statements, 539-541
foreground TSO, 725
lock strategies, 538-539
LOCK TABLE command, 538
parallel processing, 536-538
restartable, 543-546
SAFEPOINTs, 542-543
TSO (Time-Sharing Options), 708-709
units of work, 541-542

batch reports, 928, 968

BEFORE triggers, 390

BETWEEN predicate, 74

BIGINT data type, 39, 244

BIGINT function, 142

BINARY data type, 39-40, 245

BINARY function, 142

BIND, 637-654, 1014
- ACTION parameter, 639
- BIND PACKAGE Parameter, 640
- BIND PLAN parameter, 639
- CACHESIZE parameter, 647
- CONCURRENTACCESS-RESOLUTION parameter, 643
- DYNAMICRULES parameter, 652-653
- IMMEDWRITE parameter, 654
- OPTHINT parameter, 654
- parameters
  - row locks, 893-895
  - table space locks, 892-893
- PATH parameter, 652
- QUALIFIER parameter, 638

Bind CLIST listing (15.4), 621-622

BIND command, 590, 605-606, 650

BIND PACKAGE command, 605, 887, 1360

Bind Package panel (DB2I), 613

BIND PACKAGE parameter (BIND), 640

BIND parameters, 618, 916

BIND PLAN command, 605, 887, 1360

Bind Plan panel (DB2I), 613

BIND PLAN parameter (BIND), 639

BIND QUERY command, 1360

bind-time authorization checking, dynamic SQL, 596

Bind/Rebind/Free option (TSO), 717-720

BINDAGENT authority, 456

BINDAGENT group-level authorization, 41

binding, 590, 605-606, 650
- packages, 640
- plans, 605, 887, 1360
- queries, 1360
- rebinding, 1014-1016
  - best practices, 1016-1018
  - gathering statistics, 1019-1020
  - periodic maintenance, 1015
  - regular maintenance, 1015
  - reviewing access paths, 1018-1019
  - system maintenance, 1015-1016

BIND_EXPLAIN_ONLY column (PLAN_TABLE), 991

BIND_RO_TYPE column (DSN_STATEMENT_CACHE_TABLE), 1005

BIND_TIME column (PLAN_TABLE), 990

bitemporal tables, implementing, 443-445

“black boxes,” application development, 528-529

BLOB data type, 39, 245

BLOB function, 142

BLOBs (Binary Large Objects), 394

block fetches, 66, 1487-1488
- coding cursors, 1487-1489
- continuous, 1490-1491
- data currency, 1489-1490
- limited, 1490

block style, SQL statements, 552-554

blogs, 699, 1428
- Twitter, 1428-1429

bookmarking web pages, 702

Boot Strap Data Set (BSDS), 921, 1316

boundaries, partitions, changing, 367

BRF (Basic Row Format), 799, 1081

browsers, 689

browsing, cursor-free, 522

BSDS (Boot Strap Data Set), 921, 1316

Buffer Manager, 892
buffer pool general information statistics report, 961
buffer pool read information statistics report, 962
buffer pool usage information, long accounting reports, 952-955
buffer pool write information statistics report, 962
buffer pools, 1108-1110
data sharing, 780
data sharing group, 1110-1114
determining sizes, 1106-1108
indexes, 349
memory structures, 1066-1067
parameters, 1096-1101
AUTOSIZE, 1105-1106
DWQT, 1104
PGFIX, 1105
PGSTEAL, 1104-1105
VPSEQT, 1102-1103
BUFFERPOOL parameter, table space, 229
BUILD phase
LOAD utility, 1243
REBUILD_INDEX utility, 1234
REORG INDEX utility, 1273
REORG TABLESPACE utility, 1274
built-in functions, 85-86, 163. ***See also*** **functions**
business requirements, UDTs (user-defined data types), 193-195
business time, 430
modifying data, 434-441
period data types, 446
querying data, 432-434
WITHOUT OVERLAPS, 446-447
BUSINESS TIME tables, 431
bytecodes, JVMs (Java Virtual Machines), 555

C

C/S tools, 1402
CACHED_TS column (DSN_STATEMENT_CACHE_TABLE), 1005
caches
DSC (dynamic statement cache), 1003
in-memory table, 858-859
statements, sharing, 591-592
CACHESIZE parameter (BIND), 647
caching, dynamic statement, 590-591
CAF (Call Attach Facility), 763-765, 1152
batch processing, 771
benefits, 766
creating threads, 764-766
drawbacks, 766
feasibility, 770-771
resource availability, 769
vendor tools, 766-767
calling stored procedures, 672
calls, stored procedures, nesting, 663-664, 674
canned reporting, 11
cardinality
columns, 335
UDFs (user-defined functions), 190
Cartesian products, 20
avoiding, 102
SQL joins, 19-20
Cartesian star joins, 856
CASE (computer-aided software engineering) tools, 487
CASE expressions, 32-34, 98-99
case sensitivity, XPath, 421
CAST function, 184
CAST operation, 163
CAT tools, 1400-1401
Catalog, 874-882
Communication area, 881
contention, 885-886, 1367, 1370
Environmental area, 881
Objects area, 880
Performance area, 881
posters, 885
Programs area, 880
queries, 1047
application efficiency, 1041-1042

authorization, 1044-1046
creating formatted reports, 1047
historical, 1043-1044
monitoring objects, 1021-1048
navigational, 1023-1031
partition statistics, 1036-1037
physical analysis, 1031-1036
programmer's aid, 1037-1041
query and analysis tools, 1400-1401
relationships, 883-885
remote management access tables, 1459
RTS (Real Time Statistics), 1048, 1054-1058
DSNACCOX, 1054
externalization, 1053-1054
tables, 1048-1052
Security area, 881
statistics, 819-821
changing, 1124-1130
tuning, 1089-1092
utilities, 1289
CATENFM, 1289
CATMAINT, 1289
DSNJCNVB, 1290
MODIFY RECOVERY, 1290-1293
MODIFY STATISTICS, 1293-1295
RUNSTATS, 1295-1310
STOSPACE, 1311-1313
Utility area, 881
XML area, 880

Catalog integrity verification utility, 1318-1319

categories of DB2 products, 1394

CATENFM utility, 1289

CATMAINT utility, 1289

CCSID ENCODING function, 142

CCSID parameter, table space, 233

CDB (Communication Database) table, 879

CDR (Conceptual Design Review), 1437

CDRA (Character Data Representation Architecture), DRDA (Distributed Relational Database Architecture), 1452

CEILING function, 142

CEMT options, CICS interface, 740

change log inventory utility, 1316-1317

change support, online schema, 355-357

changed columns, updating, 126-127

CHANGELIMIT parameter, COPY utility, 1208

CHAR function, 38, 142, 244

Character Data Representation Architecture (CCRA), DRDA (Distributed Relational Database Architecture), 1452

CHARACTER LENGTH function, 142

check constraint checking, CHECK_DATA utility, 1179

check constraints, 297-300
semantics, 299
versus triggers, 300, 374-375

CHECK DATA JCL (for LOB References) listing (32.2), 1180

CHECK DATA JCL (for XML References) listing (32.3), 1182

CHECK DATA JCL listing (32.1), 1178

CHECK DATA utility, 427

CHECK INDEX JCL listing (32.5), 1188-1189

CHECK INDEX utility, 1165

CHECK LOB JCL listing (32.4), 1186-1187

CHECK utility, 1177

CHECKDAT phase (CHECK_DATA), 1183

Checking for DB2 Availability listing (18.3), 750

CHECKXML phase (CHECK_DATA), 1182

CHECK_DATA utility, 1177-1186, 1190-1191
AUXERROR parameter, 1181
check constraint checking, 1179
data integrity, 1183-1184
defining exception tables, 1184-1185
LOB reference checking, 1179-1181
locking considerations, 1183
phases, 1182
referential integrity checking, 1177
SCOPE parameter, 1180, 1184
XML reference checking, 1181-1182
CHECK_INDEX utility, 1188
locking, 1189
phases, 1189
CHECK_LOB utility, 1186
EXCEPTIONS parameter, 1188
locking considerations, 1187
phases, 1187
CICS (Customer Information Control System), 726-727
Attach Facility, 730-731
batch processing, 771
commands, 1362-1364
COMMIT, 746-747
connections, 740-745
feasibility, 770-771
file compression commands, 748
grouping transactions, 740
managing interface, 740
manuals, 750
operations, 726-727
plan management, 745
program preparation, 728-729
RDO (Resource Definition Online), 728
parameters, 732-739
resource availability, 769
subsystem IDs, 741
tables, 727-728
terminology, 726-727
threads, 732
transactions, designing, 746-750
claims, resources, registering, 904-905
classes
DFSMS (Data Facility Management System), 240
disaster recovery, 1378
dynamic SQL, 576-589, 594
EXECUTE IMMEDIATELY, 576-578
fixed-list SELECT, 581-584
non-SELECT, 578-581
pureQuery, 588-589
varying-list SELECT, 584-587
security, 40
SQL, 576-589, 594
STOGROUPS, 244
classic partitioned table spaces, 216
clauses
FETCH FIRST n ROWS ONLY, 95-96
FOR EACH ROW, 379
FOR FETCH ONLY, 66
FOR READ ONLY, 66
FOR UPDATE OF, 899
GROUP BY, 105
HAVING, 165
IN, 74
IN DATABASE, 278
KEEP UPDATE LOCKS, 90
LIKE, 275
OPTIMIZE FOR 1 ROW, 87
OPTIMIZE FOR n ROWS, 86, 1494
ORDER BY, 30, 68, 105
PIECESIZE, 350-351
WHEN, 382
WHERE, 526, 531
scalar functions, 86
WHERE NOT NULL, 343
WITH HOLD, 919
WITH NO DATA, 1527
client/server tools, 1402

CLIENT_APPLNAME traces, 972

CLIENT_USERID traces, 972

CLIENT_WKRSTNNAME traces, 972

CLIST, program preparation, 618-619

CLOBs (Character Large OBjects), 394

- data type,39, 245
- function, 143
- programs, 536

clone tables, 274-275, 301-302

cloning programs, 536

CLOSE parameter, table space, 229-231

CLSN (Commit Log Sequence Number), 908

clustered access, batch programming, 536

clustered columns, joins, 101

clustering, 327

- changing, 363-364

CM (conversation manager), 1486

COALESCE function, 103, 143

COBOL, 577-579, 662

- compilers, upgrading, 487
- error handling, 497

COBOL Program Using EXECUTE IMMEDIATE listing (14.1), 577

COBOL Program Using Non-SELECT Dynamic SQL listing (14.2), 578-579

COBOL Stored Procedure Shell listing (16.1), 662

Codd, E.F., 4

code appropriate existence checking, 96-98

code listings

- Accounting Report Additional Information, 951-952
- Accounting Report Buffer Pool Information, 953-954
- Accounting Report Database Code Usage, 952
- Accounting Report Locking Activity, 950-951
- Accounting Report Program Terminations, 951
- Accounting Report RID List Details, 956
- Audit Summary Report, 956-957
- Batch JCL for a TSO/DB2 Program, 708-709
- Bind CLIST, 621-622
- CHECK DATA JCL, 1178
- CHECK DATA JCL (for LOB References), 1180
- CHECK DATA JCL (for XML References), 1182
- CHECK INDEX JCL, 1188-1189
- CHECK LOB JCL, 1186-1187
- Checking for DB2 Availability, 750
- COBOL Program Using EXECUTE IMMEDIATE, 577
- COBOL Program Using Non-SELECT Dynamic SQL, 578-579
- COBOL Stored Procedure Shell, 662
- COPYTOCOPY JCL, 1217
- Cursor Processing, 512-513
- DB2 Accounting Report, 944
- DB2 Statistics Buffer Pool General Information, 961
- DB2 Statistics Buffer Pool Read Information, 962
- DB2 Statistics Buffer Pool Write Information, 963
- DB2 Statistics Common Storage Usage, 965
- DB2 Statistics EDM Pool Activity Information, 965-966
- DB2 Statistics Locking Activity Information, 964-965
- DB2 Statistics Log Activity Information, 963-964
- DDL to Create the DSN_FUNCTION_TABLE, 1002

DDL to Create the DSN_STATEMNT_TABLE, 998
DDL to Create the PLAN_TABLE, 982-984
DIAGNOSE JCL, 1200
DSN1CHKR JCL, 1318-1319
DSN1COMP JCL, 1320
DSN1COPY JCL, 1324
DSN1COPY JCL (Using the OBIDXLAT Option), 1325-1326
DSN1LOGP JCL, 1330
DSN1PRNT JCL, 1331
DSN1SDMP JCL, 1329
DSNJLOGF JCL, 1315-1316
DSNJU003 JCL (Change Log Inventory), 1316
DSNJU004 JCL (Print Log Map), 1318
DSNTEP4 JCL, 1332-1333
DSNTIAD JCL, 1335
DSNTIAUL JCL, 1336-1337
Fixed-List SELECT Dynamic SQL, 581
I/O Activity Summary Report, 957-959
Image copy JCL, 1167, 1203-1204
Incremental Image Copy JCL, 1204
Index Copy JCL, 1205
JCL for Full Recovery, 1227
JCL for Partial Recovery, 1227
JCL to issue I DB2 Command in Batch, 1341
JCL to Run a DL/I Batch DB2 Program, 761-762
JDBC Code Fragment, 558
LOAD JCL (Nonrestartable), 1242-1243
LOAD JCL (Restartable), 1241-1242
Long Accounting Report, 946-947
Long Accounting Report Highlights, 948
Long Accounting Report SQL Activity, 949
MERGECOPY JCL, 1218-1219
MODIFY RECOVERY JCL, 1291
MODIFY STATISTICS JCL, 1294-1295
Non-SELECT Dynamic SQL Using Parameter Markers, 579-580
Precompile, Compile, and Link CLIST, 620
Pseudo-code for Retrieving Data from an Application Join, 532
Pseudo-code for Retrieving Data from an SQL Join, 531-532
QMF Form to be Used with the DBID/PSID/OBID Query, 1389
QMF Form to be Used with the SYSCOPY Query, 1382
QUIESCE JCL, 1220-1221
REBUILD INDEX JCL, 1233
RECOVER INDEXSPACE JCL, 1228
REORG JCL (Nonrestartable), 1267-1268
REORG JCL (Restartable), 1266-1267
REPAIR DBD JCL, 1192-1193
REPAIR LOCATE JCL, 1194
REPAIR SET JCL, 1196-1197
REPORT RECOVERY JCL, 1236
REPORT TABLESPACESET JCL, 1199
Results of the DISPLAY GROUP Command, 786
Running a DB2 Program in TSO Batch, 609
RUNSTATS INDEX JCL, 1298
RUNSTATS TABLESPACE JCL, 1296-1297
Sample COBOL Error-Handling Paragraph, 497-499

Sample Program Preparation Procedure, 616-618
Sample Results of DISPLAY BUFFERPOOL, 1346
Sample Results of DISPLAY LOG, 1347
SQLDA, 582-583
SQLJ Code Fragment, 558
STOSPACE JCL, 1311
Typical Processing Scenario, 890
UNLOAD JCL, 1260
Updating with a Cursor, 514-515
Varying-List SELECT Dynamic SQL, 585
Varying-List SELECT Dynamic SQL with Minimum SQLDA, 587

code modular application development, 529

code predicates

indexable, 68-71
indexed columns, 68
nonindexable, 70
Stage 1, 69-71
table range columns, 68

code predicates. *See* predicates

coding

predicates, 76-77
SQL, 552, 554
statements, 60

COEXIST command, auto rebind, 64

Cognos, 1522

COLLATION KEY function, 143

COLLECTION column (DSN_USERQUERY_TABLE), 1133

COLLECTION privilege class, 449

collection program preparation objects, 632

COLLECTION security class, 40

collections, DBRMs, 628-629

COLLID column (DSN_STATEMENT_CACHE_TABLE), 1005

COLLID column (DSN_STATEMENT_TABLE), 999

COLLID column (PLAN_TABLE), 988

COLLID traces, 971

COLUMN database objects, 38

column functions, 34, 135-136

AVG, 136
CORRELATION, 137
COUNT BIG, 138
COUNT, 137-138
COVARIANCE, 138-139
COVARIANCE SAMP, 138-139
MAX, 139
MIN, 139-140
nulls, 165
STDDEV, 140
SUM, 140
VARIANCE, 141
VARIANCE SAMP, 141

columns, 200, 244-245

access control, 466-469
cardinality, indexes, 335
changed, updating, 126-127
clustered, joins, 101
data types, 244-245, 260-261
ROWID, 245-246
DATE and TIME, 259-260
DECIMAL, 264
defaults, 261-262
defining, 252, 257
details, online schema, 357-359
DSN_STATEMENT_CACHE_TABLE, 1004-1005
DSN_STATEMENT_TABLE, 999-1001
DSN_USERQUERY_TABLE, 1133
FLOAT, 1023
identity, 246-251, 549
IMPLICITLY HIDDEN, 265-266
indexed
code predicates, 68
joins, 101
indexes, 313, 344
adding, 343, 362-363
uniqueness, 342
INTEGER, 264-265

labels, specifying, 277-278
limiting grouping, 105
LOB, 395-399
  accessing, 399-402
minimizing arithmetic, 80
minimum required, 58-59
naming, 252
  homonyms, 257
  standardized abbreviations, 252
  synonyms, 257
nullable, 254-257
online schema, changing, 361-362
PLAN_TABLE, 62, 986-993, 1006
prefixing, 603-604
renaming, 361
  arithmetic expressions, 80
  views, 306-307
RLSTs (resource limit specification tables), 1146
ROW CHANGE TIMESTAMP, 269-270
ROWID, 403
sequence objects, 246-251
sequencing, 253-254
SYSIBM.SYSINDEX-SPACESTATS, 1051-1052
SYSIBM.SYSTABLE-SPACESTATS, 1049
table range, code predicates, 68
TIMESTAMP, 259-260
values
  assigning, 196
  computing average, 136
VARCHAR, 263, 395-396, 404
VARGRAPHIC, 395-396, 404
variable, 262-263
  indexing, 329-330
  monitoring, 263-264
XML, 412-414

COLUMN_FN_EVAL column (PLAN_TABLE), 988

COM tools, 1401-1402

combined tables, denormalization, 283

come-from checking, 1501

command thread attributes, DB2CONN parameter (RDO), 736-737

commands, 1340
  administrative, 1353-1358
  ALTER UTILITY, 1282
  BIND, 590, 605-606
  BIND PACKAGE, 605, 887
  BIND PLAN, 605, 887
  CICS, 1362-1364
  COEXIST, 64
  DCLGEN, 601-604, 611, 632-634
  DISPLAY GROUP, 786
  DSN, 1359-1361
  environment, 1340-1342
  environment control, 1358-1359
  EXPLAIN, 61, 64
  IMS, 1361-1362
  information-gathering, 1343-1355
  IRLM, 1364-1365
  LOCK TABLE, 538
  SELECT, 58-59
  TSO, 1364

COMMENT ON statement, 277

comments, tables, 277

COMMIT statement, 788, 907
  CICS (Customer Information Control System), 746-747
  IMS/TM, 756-759
  two-phase, 1466-1470
    coordinators, 1468
    multi-site updating, 1468
    participants, 1468

Commit Log Sequence Number (CLSN), 908

COMMIT statements, 539-541, 548

common storage statistics report, 965

common table expressions (CTEs), 105, 110-111
  recursion, 111-115

Communication area (Catalog), 881

communication database, 1459-1460

Communication Database (CDB) tables, 879

COMPARE DECFLOAT function, 143

compilers, COBOL, upgrading, 487

compiling programs, 606-607

compression
- indexes, 336-337
- table space, 231-233

compression analyzer utility, 1320-1322

compression tools, 1401-1402

computer-aided software engineering (CASE) tools, 487

CONCAT function, 143

Conceptual Design Review (CDR), 1437

concurrency, LOAD utility, 1250-1251

concurrent copying, DFSMS, 1381

CONCURRENTACCESS-RESOLUTION parameter (BIND), 643

Connect (DB2), 1473-1474, 1480-1484
- editions, 1475-1479
- EE thread pooling, 1479-1480
- supported platforms, 1475

connection attributes, DB2CONN parameter (RDO), 733-734

connection pooling, databases, 1469-1470

connections
- CICS, 740-745
- releasing, 1462

console messages, viewing, 972-977

consolidation, extents, 226

constants, UDTs (user-defined data types), 197-198

constraint, referential, 291

constraints
- check, 297-300
  - semantics, 299
  - versus triggers, 300
- informational, 296
- referential, 291-294
- self-referencing, 296

CONTAINS function, 143

contention
- Catalog, 1367, 1370
- DB2 Catalog, 885-886
- utilities, 1367, 1370

contingency planning, 1376-1377, 1390-1392
- disaster recovery, 1379-1380
  - requirements, 1379
  - technologies, 1387-1388
- DSN1COPY strategy, 1384-1385
- environmental considerations, 1388-1390
- FlashCopy strategy, 1385-1387
- risk, determining and managing, 1377-1379
- Scalpel strategy, 1381-1384
- Sledgehammer strategy, 1380-1381

continuous block fetches, 1490-1491

continuous monitoring, 967

conversation manager (CM), 1486

Coordinated Universal Time (UTC), 121

coordinators, two-phase commit, 1468

COPY utility, 1165, 1202-1215, 1372
- copying index spaces, 1204-1205
- copying table spaces, 1203-1204
- full image copy, 1202-1203
- incremental image copy, 1202-1203
- phases, 1205-1206

COPYCHANGES column (SYSIBM.SYSINDEX-SPACESTATS), 1052

COPYCHANGES column (SYSIBM.SYSTABLESPACE-STATS), 1050

copying
- index spaces, 1204-1205
- table spaces, 1203-1204

COPYLASTTIME column (SYSIBM.SYSINDEX-SPACESTATS), 1052
COPYLASTTIME column (SYSIBM.SYSTABLESPACE-STATS), 1050
COPYTOCOPY JCL listing (33.4), 1217
COPYTOCOPY utility, 1215-1218
- execution, 1216-1217
- phases, 1216

COPYUPDATEDPAGES column (SYSIBM. SYSINDEXSPACESTATS), 1052
COPYUPDATEDPAGES column (SYSIBM. SYSTABLESPACESTATS), 1050
COPYUPDATELRSN column (SYSIBM.SYSINDEX-SPACESTATS), 1052
COPYUPDATELRSN column (SYSIBM.SYSTABLESPACE-STATS), 1050
COPYUPDATETIME column (SYSIBM.SYSINDEX-SPACESTATS), 1052
COPYUPDATETIME column (SYSIBM.SYSTABLESPACE-STATS), 1050
correcting pending states, 1374
CORRELATION function, 137
CORRELATION_NAME column (PLAN_TABLE), 989
COS function, 143
COSH function, 143
COST_CATEGORY column (DSN_STATEMENT_TABLE), 1000
COUNT BIG function, 138
COUNT function, 137-138
coupling facilities
- data sharing, 778-779
- multiple, 788
- preventing failures, 789
- recovery, 786

COVARIANCE function, 138-139
COVARIANCE SAMP function, 138-139
CPUs (central processing units)
- costs, 818
- specialty, 812-815
- usage, tuning, 1074-1076

CPY2CPY phase (COPYTO-COPY), 1216
CREATE AS, 276
CREATE FUNCTION statement, 606
CREATE INDEX statement, 330
CREATE LIKE, 276
CREATE statement, 1524
CREATE TRIGGER statement, 606
CREATOR column (PLAN_TABLE), 986
CREATOR column (SYSIBM.SYSINDEX-SPACESTATS), 1052
cross-posting to newsgroups, 701
Cross-system Coupling Facility (XCF) groups, 774
CTEREF column (PLAN_TABLE), 991
CTEs (common table expressions), 105, 110-111
- recursion, 111-115

CURRENT PACKAGE PATH special register, 629
currently committer data, using, 911
Cursor Processing listing (13.2), 512-513
cursors
- avoiding, 535
- coding, block fetches, 1487-1489
- data modification, 513-515, 522
- declaring, 522-523
- DELETE, 526
- dynamic pre-open, 1491
- holding, 546
- multi-row fetch, 518-520
- multiple rows, retrieving, 535
- programming with, 511-513
- rowset positioning, 518-519
  - data modification, 519-520
  - inserting multiple rows, 520

scrollable, 515-518
ASENSITIVE, 516
fetching data, 515
INSENSITIVE, 515-516
moving within result sets, 535
SENSITIVE, 515-517
SELECT statement, 521-525
UPDATE, 526
CURSQLID column (DSN_STATEMENT_CACHE_TABLE), 1005
Customer Information Control System (CICS). ***See*** **CICS (Customer Information Control System)**

D

Dashboard solutions including Clarity, 1522
data administration, 1432-1433
Data Administration Newsletter, The, **1427**
data buffering, 779
DATA CAPTURE CHANGES option, 922
data cleaning, data warehouses, 1516-1519
data compression, 51
table spaces, 231-233
data consistency utilities, 1176-1177
CHECK, 1177
CHECK_DATA, 1177-1182, 1185-1186
check constraint checking, 1179
data integrity, 1183-1184
defining exception tables, 1184-1185
LOB referencing checking, 1179-1181
locking considerations, 1183
phases, 1182
referential integrity checking, 1177
XML referencing checking, 1181-1182
CHECK_INDEX, 1188-1191
locking, 1189
phases, 1189
CHECK_LOB, 1186
EXCEPTIONS parameter, 1188
locking considerations, 1187
phases, 1187
DIAGNOSE, 1200
REPAIR, 1191, 1198
REPAIR_DBD, 1192-1193
REPAIR_LOCATE, 1193-1195
REPAIR_SET, 1196-1198
REPORT, 1198-1200
REPORT_TABLESPACESET, 1199-1200
Data Control Language (DCL), 11
data currency, block fetches, 1489-1490
data definition control, 471
Data Definition Language (DDL), 11
data derivation, views, 303
data encryption, 473-476
Data Facility Storage Management System (DFSMS), 239-242
classes, 240
data filtering, application development, 531
data integrity, 290
enforcing, 566
FOR UPDATE OF, 526
triggers, 300
data integrity, CHECK_DATA utility, 1183-1184
Data Manipulation Language (DML), 11
data marts versus data warehousing, 1508
data modification
cursors, 513-515, 522
embedded SQL statements, 525-527
rowset positioning cursors, 519-520

data modification guidelines, 125-134

data movement tools, 1406

data movement utilities

LOAD, 1240-1243, 1252-1259

concurrency, 1250-1251

creating flash copy, 1245

creating inline copy, 1245

gathering inline statistics, 1245

loading delimited input data sets, 1246

locking, 1250-1251

phases, 1243-1244

rerun/restart procedures, 1246-1250

sorting, 1251-1252

versus INSERT, 1244-1245

REORG, 1265-1272, 1283-1288

gathering inline statistics, 1278-1279

job streams, 1272-1273

online reorganization, 1279-1283

phases, 1273-1275

reorganization frequency, 1268-1269

rerun/restart procedures, 1275-1278

RTS (Real Time Statistics), 1269-1272

SHRLEVEL parameter, 1279

UNLOAD, 1260-1265

locking, 1262

phases, 1261

restarting, 1261

termination, 1261

versus DSNTIAUL, 1262

data partitioned secondary indexes (DPSIs), 54, 332-334

data processing, SQL (Structured Query Language), 11

data set dump creator utility, 1330-1332

data sets

BSDS (bootstrap data set), 1316

delimited input, loading, 1246

DSN1SDMP, 1329

RUNSTATS utility, 1298-1299

table spaces, 210

VSAM, 795-797

data sharing, 51, 772, 780

administration, 783-791

application development, 787-788

application impact, 776-778

backup and recovery, 784

benefits, 772-774

buffer pools, 780

coupling facility, 778-779

database statuses, 785

global data buffering, 779

global inter-system communication, 779

global lock management, 779, 911-914

group buffer pool duplexing, 781

group buffer pools, 1110-1114

group creation, 783

groups, 775-776

monitoring, 786

naming conventions, 782-783

Parallel Sysplex, 773-774

requirements, 774-775

subsystem availability, 785

Sysplex, 773-774

data storage, physical, 792-808

data structures, 37-38

security controls, 40-42

Data Studio, 1396

Data Studio (IBM), 687

Data Transaction Manager (DTM), 1487

data transformation, data warehouses, 1515-1516

data types, 38-40
changing, online schema, 359-361
choosing, 258-259
columns, 244-245
renaming, 361
optimization, 260-261
parameters, UDFs (user-defined functions), 185
rows, ROWID, 245-246
UDFs (user-defined functions), 190-191
UDTs (user-defined data types), 191-192
assigning values, 196
business requirements, 193-195
constants, 197-198
host variables, 197-198
LOBs, 192-193
naming, 197
set operations, 198-199
using equivalent, 73-74
data warehouses, 1506-1507, 1520-1522, 1533-1539
accessing, 1519
designing, 1510-1513
IBM solutions, 1521-1522
managing, 1520
metadata, 1511
MQTs (Materialized Query Tables), 1522-1523, 1527, 1532-1533
attribute copy options, 1526
automatic query rewrite, 1528-1532
benefits, 1523
converting tables into, 1527-1528
creating, 1523-1524
population and maintenance, 1528
query optimization options, 1524-1526
refreshable options, 1524
ODS (Operational Data Store), 1508-1509
OLAP (On-Line Analytical Processing), 1509-1510
populating, 1513-1519
data cleansing, 1516-1519
data transformation, 1515-1516
propagation, 1514
replication, 1514
snapshots, 1514
star schema, 1511-1513
versus data marts and operational data, 1508
data-element definitions (SQLDA), 584
data-partitioned secondary index (DPSI), 827
database access threads (DBATs), inactive, 1469
database administration guide, 1433
database analysis tools, 1402-1403
database archiving tools, 1398
database auditing, 476-481
database code usage information, long accounting reports, 952
DATABASE database objects, 38
database descriptors (DBDs), 201-204
database management systems (DBMs), 4
database modeling and design tools, 1403-1404
database object lists, 1160
database object types, 37-38
database objects, 200-201
database request modules (DBRMs). *See* DBRMs (database request modules)
DATABASE security class, 40
database server (DS) function (DRDA), 1450
database services (DBAS), 809-812
Database Services Address Space (DBAS), 809-810
database-level changes, 57
databases, 201-202
aliases, 1471-1472
communication, 1459-1460
connection pooling, 1469-1470
data integrity, 290

denormalization, 279-281
avoiding, 289
combined tables, 283
derivable data, 285
hierarchies, 285-289
mirror tables, 282
pre-joined tables, 281
redundant data, 284
repeating groups, 284-285
report tables, 282
split tables, 282-283
testing validity, 289-290
design, tuning, 1114-1116
distributed design, 1496-1499
normalization, 278-279
parameters, specifying, 203
referential integrity, 290-292
avoiding, 295-296
check constraints, 297-300
implementation restrictions, 296
informational constraints, 296
programmatic, 295
referential constraints, 291-294
referential sets, 294-295
self-referencing constraints, 296

DataPropagator, 1521
DataQuant, 1522
DATASIZE column (SYSIBM.SYSTABLESPACESTATS), 1050
DATE, 119-120
date and time, 119-121
arithmetic, 122-123
DATE, 119-120
displaying, 120-121
irregular date sorting, 124-125
mixing, 123
non-standard dates, 121-122
TIME, 119-120
time zones, 121
TIMESTAMP, 119-120
total number of days, returning, 123
date and time arithmetic, 84-85
DATE and TIME columns, 259-260
DATE data type, 39, 245
DATE function, 143
DAY function, 143
DAYOFMONTH function, 143
DAYOFWEEK function, 143
DAYOFWEEK ISO function, 143
DAYOFYEAR function, 144
DAYS function, 144
DB2, 3-4
accounting report, 944
Internet access, 692
Microsoft .NET, 694
Net.Data, 695
WebSphere, 693-694
DB2I (Interactive), 609-616
DB2 Accounting Report listing (24.1), 944
***DB2 Administration Guide*, 1379**
***DB2 Application Programming and SQL Guide*, 585, 635**
DB2 Catalog, 874-882
Communication area, 881
contention, 885-886
Environmental area, 881
Objects area, 880
Performance area, 881
posters, 885
Programs area, 880
relationships, 883-885
Security area, 881
Utility area, 881
XML area, 880
DB2 Connect, 1473-1474, 1480-1484
editions, 1475-1479
EE thread pooling, 1479-1480
supported platforms, 1475
DB2 Directory, 886-888
***DB2 for z/OS: Data Sharing in a Nutshell*, 774**
***DB2 pureXML Cookbook*, 425**
DB2 Statistics Buffer Pool General Information listing (24.13), 961

DB2 Statistics Buffer Pool Read Information listing (24.14), 962

DB2 Statistics Buffer Pool Write Information listing (24.15), 963

DB2 Statistics Common Storage Usage listing (24.18), 965

DB2 Statistics EDM Pool Activity Information listing (24.19), 965-966

DB2 Statistics Locking Activity Information listing (24.17), 964-965

DB2 Statistics Log Activity Information listing (24.16), 963-964

DB2CONN parameter (RDO), 733
- command thread attributes, 736-737
- connection attributes, 733-734
- general attributes, 733
- pool thread attributes, 735-736

DB2ENTRY parameter (RDO), 737-739

DB2I Commands option (TSO), 720

DB2TRAN parameters (RDO), 739

DBA tools, 1402-1403
- procedural, 683-687

DBADM group-level authorization, 41

DBADM security, 451-452

DBAS (Database Services Address Space), 809-812

DBATs (database access threads)
- high performance, 1496
- inactive, 1469

DBCLOBs (Double Byte Character Large OBjects), 39, 144, 245, 394

DBCTRL group-level authorization, 41

DBD01 structure (Directory), 887

DBDs (database descriptors), 201-204

DBID column (SYSIBM.SYSINDEXSPACESTATS), 1052

DBID column (SYSIBM.SYSTABLESPACESTATS), 1050

DBIDs, 794

DBMAINT group-level authorization, 41

DBMs (database management systems), 4

DBNAME column (SYSIBM.SYSINDEXSPACESTATS), 1052

DBNAME column (SYSIBM.SYSTABLESPACESTATS), 1050

DBRM program preparation objects, 631

DBRMs (database request modules), 622-624
- collections, 628-629
- packages, 623-627
  - administration, 624-626
  - benefits, 624
  - list size considerations, 627
  - performance, 627
  - versions, 629-630
- plans, 622
  - converting, 630-631

DCE (Distributed Computing Environment) security, 482

DCL (Data Control Language), 11
- data sharing, impact, 777

DCLGEN command, 510, 601-604, 611, 632-634, 1360
- TSO (Time-Sharing Options), 717

DCRM (Distributed Communication Resource Manager), 1485-1486

DDF (Distributed Data Facility), 1485-1486
- DCRM (Distributed Communication Resource Manager), 1485-1486
- DDIS (Distributed Data Interchange System), 1485-1487

DRDS (Distributed Relational Data System), 1485
DTM (Data Transaction Manager), 1487
DTM (Distributed Transaction Manager), 1485

DDF (Distributed Data Facility), 1458

DDF DRDA
modifying, 1471
suspend and resume, 1470

DDFS (Distributed Data Facility Services), 809

DDIS (Distributed Data Interchange System), 1485-1487

DDL (Data Definition Language), 11
coding for performance, 1115
data sharing, impact, 777
DSN_STATEMENT_TABLE, 998
statements, 322

DDL to Create the DSN_FUNCTION_TABLE listing (25.3), 1002

DDL to Create the DSN_STATEMNT_TABLE listing (25.2), 998

DDL to Create the PLAN_TABLE listing (25.1), 982-984

DDM (Distributed Data Management), DRDA, 1452

deadlocks, 901-904
IMS, 759-760

DECFLOAT data type, 39, 245

DECFLOAT function, 144

DECFLOAT SORTKEY function, 144

DECIMAL columns, 264

DECIMAL data type, 39, 244

DECIMAL function, 144

decimal precision, 80

Declarations Generator, 601-604, 632-634

declarative RI, triggers, 391

DECLARE, explicit declaring tables, 490

DECLARE CURSOR statement, 523

declared temporary tables, 271-273

declaring cursors, 522-523

DECRYPT BINARY function, 144

DECRYPT BIT function, 144

DECRYPT CHAR function, 144

DECRYPT DB function, 144

defaults
columns, 261-262
rows, 261-262

Defaults option (TSO), 722

defining
columns, 252, 257
RLSTs (resource limit specification tables), 1146-1147
table schemas, 276
useful storage groups, 239

definition changes, online schema, 369-370

definitions, table spaces, 237

DEF_PAR_DEGREE column (DSN_USERQUERY_TABLE), 1133

DEGREES function, 144

delimited input data sets, loading, 1246

DELETE NO ACTION statement, 296

DELETE parameter, MODIFY STATISTICS utility, 1294

DELETE statements, 37, 126-127, 526, 548, 577

deleting
non-uniform distribution statistics, 1135-1136
XML data, 424

delimiting SQL statements, 490

denormalization, 99, 279-281
avoiding, 289
combined tables, 283
deliverable data, 285

derivable data, 285
distributed
fragmentation, 1497-1498
replication, 1498
snapshots, 1498-1499
hierarchies, 285-289
mirror tables, 282
pre-joined tables, 281
redundant data, 284
repeating groups, 284-285
report tables, 282
split tables, 282-283
testing validity, 289-290

DES tools, 1403-1404

design
databases, tuning, 1114-1116
denormalization, 279-281
avoiding, 289
combined tables, 283
derivable data, 285
hierarchies, 285-289
mirror tables, 282
pre-joined tables, 281
redundant data, 284
repeating groups, 284-285
report tables, 282
split tables, 282-283
testing validity, 289-290
normalization, 278-279

design reviews, 1436-1440

designing data warehouses, 1510-1513

destinations, traces, 936

DETERMINISTIC parameter, UDFs (user-defined functions), 188

development, applications, 486-487, 527-536
active database constructs, 530
batch programming, 536-547
“black boxes,” 528-529
code modular, 529
cursors, 511-525
data filtering, 531
data sharing, 787-788
dynamic SQL, 567-589, 594
embedded SQL statements, 487-497, 500-501, 525-527
error handling, 497-499
host variables, 504-511
Java, 554-563
naming conventions, 501-504
online programming, 547-552
program preparation, 601-605, 632-634
REXX, 563-565
SQL, 565-566
coding, 552-554
stored procedures, 530
unqualified SQL, 530
user-defined functions, 530

development tools, 1411-1412

DFSMS (Data Facility Storage Management System), 239-242
classes, 240
concurrent copying, 1381

DFSORT statement, 1081

DIAGNOSE JCL listing (32.10), 1200

DIAGNOSE utility, 1200

DIFFERENCE function, 144

DIGITS function, 144

direct index lookup, 833-834

direct row access, 550, 831-832

Directory, 886-888

dirty reads, 52

disabling query parallelism, 868

DISALLOW PARALLEL parameter, UDFs (user-defined functions), 188

disaster recovery, 1379-1380
classes, 1378
contingency planning, 1376-1377, 1390-1392
determining and managing risk, 1377-1379
DSN1COPY strategy, 1384-1385
FlashCopy strategy, 1385-1387

Scalpel strategy, 1381-1384
Sledgehammer strategy, 1380-1381
environmental considerations, 1388-1390
requirements, 1379
technologies, 1387-1388
disasters, 1376-1377
DISCARD phase (LOAD utility), 1244
DISCONNECT parameters, 1462
disk and space management tools, 1404
disk volumes, STOGROUPS, 244
DISPLAY authority, 453
DISPLAY BUFFERPOOL command, 1345-1346
DISPLAY command, 1343-1345, 1350-1351
DISPLAY DATABASE command, 1347-1350
DISPLAY GROUP command, 786
DISPLAY LOG command, 1347
DISPLAY UTILITY command, 1352-1353
displaying
date and times, 120-121
resource status, 977-979
DISTINCT, 67-68
DISTINCT TYPE security class, 40
Distributed Communication Resource Manager (DCRM), 1485-1486
Distributed Computing Environment (DCE) security, 482
distributed data, workstation DB2, 1470
Distributed Data Facility (DDF). *See* DDF (Distributed Data Facility), 1485
Distributed Data Facility Services (DDFS), 809
Distributed Data Interchange System (DDIS), 1485-1487
Distributed Data Management (DDM), DRDA, 1452
distributed data placement, 1499-1500
distributed database design, 1496-1499
distributed denormalization
fragmentation, 1497-1498
replication, 1498
snapshots, 1498-1499
distributed optimization, 1500-1501
distributed query blocks, controlling, 86
Distributed Relational Data System (DRDS), 1485
Distributed Relational Database Architecture (DRDA). *See* DRDA (Distributed Relational Database Architecture)
distributed requests, DRDA (Distributed Relational Database Architecture), 1454, 1457
distributed response time, analyzing, 1493
distributed security, 1501-1502
distributed throughput, analyzing, 1491-1493
Distributed Transaction Manager (DTM), 1485
distributed unit of work (DUW), 1460
DRDA (Distributed Relational Database Architecture), 1454-1456
distributing data
accessing, 1460-1465
DRDA (Distributed Relational Database Architecture), 1458-1460
distribution, 1502-1505
advantages, 1446
block fetches, 1487-1489
continuous, 1490-1491
data currency, 1489-1490
limited, 1490
dynamic cursor pre-open, 1491
performance problems, 1491-1496
DL/I batch interface, IMS, 761-762
DML (Data Manipulation Language), 11

document trees, XML, 414-415

DOUBLE function, 144

downtime, causes, 353

DPSIs (data partitioned secondary indexes), 54, 332-334, 827

drains, resources, 905-906

DRDA (Distributed Relational Database Architecture), 1448-1458

- APPC (Advanced Program-to-Program Communication), 1452
- architectures, 1451-1452
- benefits, 1449
- CDRA (Character Data Representation Architecture), 1452
- DDF
  - modifying, 1471
  - suspend and resume, 1470
- DDM (Distributed Data Management), 1452
- distributed data
  - accessing, 1460-1465
  - workstation DB2, 1470
- distributed requests, 1454, 1457
- distributing data, 1458-1460
- DUW (distributed unit of work), 1454-1456
- FD:OCA (Formatted Data: Object Content Architecture), 1452
- functions, 1449-1451
- levels, 1453-1455, 1460
- RDA (Remote Database Access), 1449
- remote requests, 1454-1455
- RUW (remote unit of work), 1454, 1456
- standards, 1451-1452
- three-part name support, 1463-1464
- user-assisted distribution, 1453

DRDS (Distributed Relational Data System), 1485

DRIVETYPE column (SYSIBM.SYSINDEX-SPACESTATS), 1052

DRIVETYPE column (SYSIBM.SYSTABLESPACE-STATS), 1051

DS (database server) function (DRDA), 1450

DSC (dynamic statement cache), 1003, 1495

DSD tools, 1404

DSN commands, 1359-1361

DSN FUNCTION TABLE, 184

DSN XMLVALIDATE function, 145

DSN1CHKR JCL listing (36.4), 1318-1319

DSN1CHKR utility, 1318-1319

DSN1COMP JCL listing (36.5), 1320

DSN1COMP utility, 1320-1322

DSN1COPY JCL (Using the OBIDXLAT Option) listing (36.7), 1325-1326

DSN1COPY JCL listing (36.6), 1324

DSN1COPY strategy, backup and recovery, 1384-1385

DSN1COPY utility, 1322-1328

DSN1LOGP JCL listing (36.9), 1330

DSN1LOGP utility, 1330

DSN1PRNT JCL listing (36.10), 1331

DSN1PRNT utility, 1330-1332

DSN1SDMP JCL listing (36.8), 1329

DSN1SDMP utility, 1328-1329

DSNACCOX, RTS (Real Time Statistics), 1054

DSNC command, 1362

DSNC DISCONNECT command, 1362

DSNC DISPLAY command, 1362

DSNC DISPLAY PLAN command, 1363

DSNC DISPLAY STATISTICS command, 1363

DSNC MODIFY command, 1362

DSNC STOP command, 1362

DSNC STRT command, 1362

DSNC transactions, CICS interface, 740
DSNDB04, avoiding, 203
DSNDDF, 879
DSNJCNVB utility, 1290
DSNJLOGF JCL listing (36.1), 1315-1316
DSNJLOGF utility, 1315-1316
DSNJU003 JCL (Change Log Inventory) listing (36.2), 1316
DSNJU003 utility, 1316-1317
DSNJU004 JCL (Print Log Map) listing (36.3), 1318
DSNJU004 utility, 1317-1318
DSNTEP2 program, 1332-1334
DSNTEP4 JCL listing (36.11), 1332-1333
DSNTEP4 program, 1332-1334
DSNTIAD JCL listing (36.12), 1335
DSNTIAD program, 1334-1336
DSNTIAUL versus UNLOAD utility, 1262
DSNTIAUL JCL listing (36.13), 1336-1337
DSNTIAUL program, 1336-1339
DSNUPROC parameters, 1156
DSNZPARM, 1081
DSNZPARM parameters, 917
  active users, 1095
  tuning, 1092-1096
DSN_COLDIST_TABLE, 62
DSN_COLIST_TABLE (EXPLAIN), 1002
DSN_DETCOST_TABLE (EXPLAIN), 1003
DSN_FILTER_TABLE, 62
DSN_FILTER_TABLE (EXPLAIN), 1003
DSN_FUNCTION_TABLE, 63, 176
DSN_FUNCTION_TABLE (EXPLAIN), 1001-1002
DSN_KEYTGTDIST_TABLE, 63
DSN_KEYTGTDIST_TABLE (EXPLAIN), 1002
DSN_PGRANGE_TABLE, 63
DSN_PGROUP_TABLE, 63
DSN_PGROUP_TABLE (EXPLAIN), 1003
DSN_PREDICAT_TABLE, 63
DSN_PREDICAT_TABLE (EXPLAIN), 1002
DSN_PTASK_TABLE, 63
DSN_PTASK_TABLE (EXPLAIN), 1003
DSN_QUERY_TABLE, 63
DSN_QUERY_TABLE (EXPLAIN), 1003
DSN_SORTKEY_TABLE, 63
DSN_SORTKEY_TABLE (EXPLAIN), 1003
DSN_SORT_TABLE, 63
DSN_SORT_TABLE (EXPLAIN), 1003
DSN_STATEMENT_CACHE_TABLE, 63
DSN_STATEMENT_CACHE_TABLE (EXPLAIN), 1002-1005
DSN_STATEMENT_TABLE (EXPLAIN), 998-1001
DSN_STATEMNT_TABLE, 63
DSN_STRUCT_TABLE, 63
DSN_STRUCT_TABLE (EXPLAIN), 1003
DSN_USERQUERY_TABLE, 1133
DSN_VIEWREF_TABLE, 63
DSN_VIEWREF_TABLE (EXPLAIN), 1003
DSSIZE parameter, table space, 217-218
DTM (Data Transaction Manager), 1487
DTM (Distributed Transaction Manager), 1485
dummy tables, 81-82
dump and trace utility, 1328-1329
duplicate rows, avoiding, 257
duplicating table schema, 275
duration
  date and time, 122
  LOBs, 915-916
  locks, 892-895
DUW (distributed unit of work), 1460
  DRDA, 1454-1456
DWQT parameter (buffer pool), 1104

dynamic cursor pre-open, 1491

dynamic SENSITIVE scrollable cursors, 516-517

dynamic SQL, 44-45, 567-569, 574, 594, 597-600
- bind-time authorization checking, 596
- caching prepared statements, 596
- classes, 576-589, 594
  - EXECUTE IMMEDIATE, 576-578
  - fixed-list SELECT, 581-584
  - non-SELECT, 578-581
  - varying-list SELECT, 584-587
- data uniformity, 571-572
- host variables, 574-575
- KEEPDYNAMIC parameter, 591
- making more static, 589-593
- monitoring, 575-576
- parallelism, 596
- parameter markers, 595-596
- performance sensitivity, 571
- program examples, 576
- pureQuery, 588-589
- range predicates, 572
- REOPT parameter, 592-593
- repetitions execution, 573
- RUNSTATS, 574
- runtime environment, 574
- statements, 594
- tuning, 575-576
- versus static, 569-576

dynamic SQL processor, 1332-1334

dynamic SQL update program, 1334-1336

dynamic statement cache (DSC), 1003, 1495

dynamic statement caching, 590-591

dynamic system parameters, 53

DYNAMICRULES parameter (BIND), 652-653

E

EBCDIC CHR function, 145

EBCDIC STR function, 145

Eclipse IDE (Integrated Development Environment), 588

editing ROWID columns, 403

editors (table), 1405-1406

EDITPROCs, 300

EDM DBD caches, 202

EDM pool, 202

EDM pool activity statistics report, 965-966

EDM pools, 1069-1073

EDM skeleton pool, 202

EDM statement caches, 202

EDT tools, 1405-1406

education, 1423-1427
- blogs, 1428
  - Twitter, 1428-1429
- industry periodicals, 1427-1428
- mailing lists, 1429
- webinars, 1429

EE thread pooling, DB2 Connect, 1479-1480

EJBs (Enterprise Java Beans), 557

embedded SELECT statements, cursors, 521-525

embedded SQL statements, 487-490
- data modification, 525-527
- delimiting, 490
- GET DIAGNOSTICS, 494, 497
- pretesting, 61
- SQLCA (SQL Communication Area), 491, 493
- SQLCODE, 493
- SQLSTATE, 493
- WHENEVER, 500-501

emptying tables, 546-547

enclaves, 815

ENCRYPT STR function, 145

encryption, 473-476

END command, 1360

ENFORCE phase (LOAD utility), 1244

enforcing naming conventions, 314-316

Enterprise Java Beans (EJBs), 557

Enterprise Systems Connection (ESCON), 1077

enterprise-wide changes, 57

environment

- teleprocessing, 1087-1088
- tuning, 1064
  - teleprocessing, 1087-1088
  - z/OS, 1064-1087

environment commands, 1340-1342

environment control commands, 1358-1359

Environmental area (Catalog), 881

environmental support, 1443

equivalent data types, 73-74

ERASE parameter, table space, 231

error handling, standardizing, 497, 499

escalation, locks, 917

ESCON (Enterprise Systems Connection), 1077

ESCR (extended storage constraint relief), 1065

ETL tools, 1406

EXCEPT set operations, 26

exception reporting, 103

exception-based monitoring, 967

EXCEPTIONS parameter, CHECK_LOB utility, 1188

EXEC SQL utility, SQL statements, issuing, 1173-1175

EXECUTE IMMEDIATELY class (dynamic SQL), 576-578

EXECUTE security class, 40

execution, external UDFs, 173-178

execution environments, 488

existential SQL, 12

EXISTS, 73

EXP function, 145

EXPLAIN, 176, 980-982, 1005-1011

- access paths, 993-998
- tables, 1002
  - DSN_COLDIST_TABLE, 1002
  - DSN_DETCOST_TABLE, 1003
  - DSN_FILTER_TABLE, 1003
  - DSN_FUNCTION_TABLE, 1001-1002
  - DSN_KEYTGTDIST_TABLE, 1002
  - DSN_PGRANGE_TABLE, 1003
  - DSN_PGROUP_TABLE, 1003
  - DSN_PREDICAT_TABLE, 1002
  - DSN_PTASK_TABLE, 1003
  - DSN_QUERY_TABLE, 1003
  - DSN_SORTKEY_TABLE, 1003
  - DSN_SORT_TABLE, 1003
  - DSN_STATEMENT_CACHE_TABLE, 1002-1005
  - DSN_STATEMENT_TABLE, 998-1001
  - DSN_STRUCT_TABLE, 1003
  - DSN_VIEWREF_TABLE, 1003
  - PLAN_TABLE, 982-993, 1006

EXPLAIN command, 61

- auto rebind, 64
- table creation, 62-63

EXPLAIN privilege, 456-457

explain reports, 957

EXPLAIN_TIME column (DSN_STATEMENT_TABLE), 1000

EXPLAIN_TIME column (PLAN_TABLE), 991

explicitly coding literals, 533

explicitly declaring tables, 490

expressions

- arithmetic, 78-80
- CASE, 32-34, 98-99
- common table expressions (CTEs), 105, 110-115
- index on, 54

indexing, 330-331
row, 107-108
table, 105-106
improving performance, 106-107
tables, 91
XPath, 420
extended storage constraint relief (ESCR), 1065
extenders, 407
Extensible Markup Language (XML). *See* XML (Extensible Markup Language)
extents, 226
EXTENTS column (SYSIBM.SYSINDEX-SPACESTATS), 1051
EXTENTS column (SYSIBM.SYSTABLESPACE-STATS), 1049
EXTERNAL ACTION UDFs (user-defined functions), 189
external scalar UDFs, 168-171
external security, 481
external stored procedure, 676-677
external table UDFs, 168
creating, 171-173
external UDFs, 169
externalization, RTS (Real Time Statistics), 1053-1054
EXTRACT function, 145
extract tools, 1406

F

failures, stored procedures, controlling, 669-670
FAQs (Frequently Asked Questions), 701
FD:OCA (Formatted Data: Object Content Architecture), DRDA, 1452
FETCH, 899
FETCH FIRST, 108
FETCH FIRST n ROWS ONLY clause, 95-96
FETCH operation, multi-row, 54
FETCH statement, 109-110
FETCH WITH CONTINUE, 402
fetches
block, 1487-1488
coding cursors, 1487-1489
continuous, 1490-1491
data currency, 1489-1490
limited, 1490
multi-row, 109-110
fetching data, scrollable data, 515
FICON (Fibre Connectivity), 1077
field definitions, SQLCA (SQL Communication Area), 492
FIELDPROCs, 300
inheriting, 1527
file compression commands, CICS, 748
file reference variables, LOB, 402
filter factor, formulas, 822
Financial Modernization Act of 1999, 1399
firing triggers, 377-378, 385
fixed-list SELECT class (dynamic SQL), 581-584
Fixed-List SELECT Dynamic SQL listing (14.4), 581
flames, newsgroups, 701
FlashCopy, 1214
FlashCopy strategy, backup and recovery, 1385-1387
flat file programming versus DB2 programming, 489
FLOAT columns, 1023
FLOAT data type, 39, 244
FLOAT function, 144-145
FLOOR function, 145
FOR EACH ROW clause, 379
FOR FETCH ONLY clause, 66
FOR READ ONLY clause, 66, 533
FOR UPDATE OF clause, 899
column list, 126
data integrity, 526
foreground TSO
batch programs, 725
Time-Sharing Options, 709
foreign keys, indexes, 343
Formatted Data: Object Content Architecture (FD:OCA), DRDA, 1452

formatted reports, creating, 1047

formulas, filter factor, 822

fourth-generation languages (4GLs), 486

tools, 1419-1420

fragmentation, 1497-1498

FREE PACKAGE command, 1360

FREE PLAN command, 1360

FREE QUERY command, 1360

free space, indexes, specifying, 347-348

FREEPAGE parameter (DSN1COMP utility), 1321

table space, 227-229

Frequently Asked Questions (FAQs), 701

full image copy, 1202-1203

FULL OUTER JOIN, 28

FULLCOPY parameter (DSN1COMP utility), 1321

fullselect, table schemas, defining, 276

functions, 135

aggregate, 34, 135-136

AVG, 136

CORRELATION, 137

COUNT BIG, 138

COUNT, 137-138

COVARIANCE, 138-139

COVARIANCE SAMP, 138-139

MAX, 139

MIN, 139-140

searching results, 165

STDDEV, 140

SUM, 140

VARIANCE, 141

VARIANCE SAMP, 141

built-in, 85-86

CAST, 184

column, 34

DRDA, 1449-1451

hash, 337

MQSeries scalar, 159-161

scalar, 34, 141

ABSVAL, 141

ACOS, 141

ADD MONTHS, 141

ASCII, 142

ASCII CHAR, 142

ASCII STR, 142

ASIN, 142

ATAN, 142

ATAN2, 142

ATANH, 142

BIGNIT, 142

BINARY, 142

BLOB, 142

CCSID ENCODING, 142

CEILING, 142

CHAR, 142

CHARACTER LENGTH, 142

CLOB, 143

COALESCE, 103, 143

COLLATION KEY, 143

COMPARE DECFLOAT, 143

CONCAT, 143

CONTAINS, 143

COS, 143

COSH, 143

DATE, 143

DAY, 143

DAYOFMONTH, 143

DAYOFWEEK, 143

DAYOFWEEK ISO, 143

DAYOFYEAR, 144

DAYS, 144

DBCLOB, 144

DECFLOAT, 144

DECFLOAT SORTKEY, 144

DECIMAL, 144

DECRYPT BINARY, 144

DECRYPT BIT, 144

DECRYPT CHAR, 144

DECRYPT DB, 144

DEGREES, 144

DIFFERENCE, 144

DIGITS, 144

DOUBLE, 144

DSN XMLVALIDATE, 145

EBCDIC CHR, 145

EBCDIC STR, 145

ENCRYPT STR, 145

EXP, 145

EXTRACT, 145

FLOAT, 144-145
FLOOR, 145
GENERATE UNIQUE, 145
GETHINT, 145
GETVARIABLE, 145
GRAPHIC, 145
GREATEST, 146
HEX, 146
HOUR, 146
IDENTY VAL LOCAL(), 146
INFULL, 146
INSERT, 146
INTEGER, 147
JULIAN DAY, 147
LAST DAY, 147
LEAST, 147
LEFT, 147
LENGTH, 147
LOCATE, 147-148
LOCATE IN STRING, 148
LOG10, 148
LOWER, 148
LPAD, 148
LTRIM, 148
MAX, 148
MICROSECOND, 148
MIDNIGHT SECONDS, 148
MIN, 148
MINUTE, 149
MOD, 149
MONTH, 149
MONTHS BETWEEN, 149
MULTIPLY ALT, 149
NEXT DAY, 149
NORMALIZE DECFLOAT, 149
NORMALIZE STRING, 149
NULLIF, 149
OVERLAY, 149
POSITION, 150
POSSTR, 150
POWER, 150
QUANTIZE, 150
QUARTER, 150
RADIANS, 150
RAISE ERROR, 150, 162-163
RAND, 82, 151
REAL, 151
REPEAT, 151
REPLACE, 151
RID, 151
RIGHT, 152
ROUND, 152
ROUND TIMESTAMP, 152
ROWID, 152
RPAD, 152
RTRIM, 152
SCORE, 152
SECOND, 152
SIGN, 153
SIN, 153
SINH, 153
SMALLINT, 153
SOAPHTTPC, 153
SOAPHTTPV, 153
SOUNDEX, 153
SPACE, 153
SQRT, 153
STRIP, 153
SUBSTR, 154
SUBSTRING, 154
TAN, 154
TANH, 154
TIME, 154
TIMESTAMP, 154
TIMESTAMP FORMAT, 155
TIMESTAMP ISO, 155
TIMESTAMP TZ, 155
TIMESTAMPADD, 154
TIMESTAMPDIFF, 154
TOTALORDER, 155
TRANSLATE, 155-156, 165
TRUNC TIMESTAMP, 156-157
TRUNCATE, 156
trusted context, 158
UNICODE, 157
UNICODE STR, 157
UPPER, 157, 165
VALUE, 157
VARBINARY, 157
VARCHAR, 157
VARCHAR FORMAT, 158
VARGRAPHIC, 158

WEEK, 158
WEEK ISO, 158
WHERE clauses, 86
YEAR, 158
SQL, 34
string units, 166
synonyms, 164-165
table, 159
user-defined, 135, 167-168
abends, 184
cardinality, 190
creating, 169-171, 173
data types, 190-191
DETERMINISTIC parameter, 188
DISALLOW PARALLEL parameter, 188
DSN FUNCTION TABLE, 184
execution, 173-178
external, 169
EXTERNAL ACTION, 189
external scalar, 168
external table, 168
invoking, 184
naming, 180
NOT DETERMINISTIC parameter, 188
null input arguments, 189
parameter data types, 185
parameters, 185
program restrictions, 181
programming languages, 173
reusablility, 184
schema, 169
scratchpads, 189
SECURITY parameter, 188-189
SET CURRENT PACKAGE PATH, 184
sourced, 168, 178
SQL scalar, 168, 178-179
SQL table, 168, 179-180
SQL within, 186-187
starting and stopping, 182-183
templates, 185-186
versus program logic, 163
XML, 161-162
XMLQUERY, 421
XMLTABLE(), 422

G

GBPCACHE option, 777
GBPCACHE parameter, buffer pools, 1111
GDPS (Geographically Dispersed Parallel Sysplex), 1387
***GDPS Family - An Introduction to Concepts and Capabilities*, 1388**
general attributes
DB2CONN parameter (RDO), 733
DB2ENTRY parameter (RDO), 737
GENERATE UNIQUE function, 145
generating utility JCL, 1152-1156
Geographically Dispersed Parallel Sysplex (GDPS), 1387
GET DIAGNOSTICS statement, 494, 497
GETHINT function, 145
GETVARIABLE function, 145
global data buffering, data sharing, 779
global inter-system communication, data sharing, 779
global lock management, data sharing, 779, 911-914
global trace, 933
GMT (Greenwich Mean Time), 121
Gonzales, Ralph Michael L., 1522
Google Groups, 691-692
governing resources, Resource Limit Facility (RLF), 1143
defining RLSTs, 1146-1147
predictive governing, 1144, 1150
reactive governing, 1144, 1150

Gramm-Leach-Bliley Act, 1399

granting privileges, 449-450

GRAPHIC data type, 38, 244

GRAPHIC function, 145

greater than or equal to predicate, 74

GREATEST function, 146

Greenwich Mean Time (GMT), 121

group buffer pool duplexing, 781, 788

GROUP BY clause, 105

group privileges, 451-452

group-level authorizations, 41

grouping

- retrieved data, 29-30
- transactions, 740

groups

- Cross-system Coupling Facility (XCF), 774
- data sharing, 775-776
  - creating, 783
  - monitoring, 786
- naming conventions, 782

GROUP_MEMBER column (DSN_STATEMENT_TABLE), 1000

GROUP_MEMBER column (PLAN_TABLE), 989

GROUP_MEMBER traces, 971

GUI-based programming languages, 486

H

hash overflow, 339

hash spaces, 338-339

hash-organized tables, creating, 339-341, 347

hashes, 337

- MAXROWS parameter, avoiding, 352
- monitoring usage, 351
- space, altering, 352

hashing, 324-326, 338

- indexes, 55

HASHLASTUSED column (SYSIBM.SYSTABLESPACE-STATS), 1050

HAVING clause, 165

- versus WHERE clause, 30-31

Health Insurance Portability and Accountability Act (HIPAA), 478, 1398

HEX function, 146

hierarchical locking, 913

hierarchies, denormalization, 285-289

high-level qualifiers, STOGROUPS, 243

hints, optimizer, 48-49

HINT_SCOPE column (DSN_USERQUERY_TABLE), 1133

HINT_USED column (PLAN_TABLE), 990

HIPAA (Health Insurance Portability and Accountability Act), 478, 1398-1399

historical queries, Catalog, 1043-1044

historical statistics, 53

holding cursors, 546

homonyms, columns, 257

host structures, 506, 533

host variable arrays, 527

host variables, 504-506, 509-511

- dynamic SQL, 574-575
- host structures, 506
- null indicators, 507-509
- simulating, temporary tables, 533-534
- UDTs (user-defined data types), 197-198

HOUR function, 146

HTTP (HyperText Transfer Protocol), 690

hybrid joins, 51, 851-854, 859

- parallelism, 867

HyperText Transfer Protocol (HTTP), 690

I

I/O

- activity reports, 957-959
- application, 1078-1079
- cost, 818
- internal, 1080-1081
- log, 1082-1083
- paging, 1083-1084
- sort, 1081-1082
- tuning, 1076-1084

I/O Activity Summary Report listing (24.12), 957-959

IBM Academic Initiative System, 1443

***IBM Data Management* magazine, 1427**

IBM Data Studio, 687

access paths, managing, 1012

***IBM Data Warehousing*, 1522**

***IBM DB2 Application Programming and SQL Guide* manual, 576**

***IBM DB2 Codes* manual, 493**

IBM DB2 for z/OS page, 698

***IBM DB2 for z/OS pureXML Guide*, 425**

***IBM DB2 Installation Guide*, 1358**

***IBM DB2 SQL Reference Manual*, 576, 982, 1003**

***IBM DFSORT Application Programming Guide*, 1371**

IBM hints, 49

***IBM RedBook Extremely pureXML in DB2 10 for z/OS*, 425**

IBM SQL reserved words, 503-504

IBM SWL procedural SQL, 678-680

***IBM System Journal*, 1427**

***IBM Systems* magazine, 1427**

IBM utilities, 1158-1159

IBMREQD column (SYSIBM.SYSINDEX-SPACESTATS), 1050-1052

IBM_SERVICE_DATA column (PLAN_TABLE), 989

IDENTITY column, 250-251

identity columns, 246-251, 549, 1030

IDENTY VAL LOCAL() function, 146

IDs, authorization, 448-449

IDUG (International DB2 Users Group) conferences, 1427

***IDUG Solutions Journal*, 1427**

IFCIDs (Instrumentation Facility Component Identifiers), 937-938

IFI (Instrumentation Facility Interface), 53, 942

image copies, indexes, 351

Image copy JCL listing (31.1), 1167

Image Copy JCL listing (33.1), 1203-1204

IMMEDWRITE parameter (BIND), 654

impedance mismatches, 44

implementation restrictions, referential integrity, 296

implicit table spaces, avoiding, 278

IMPLICITLY HIDDEN columns, 265-266

IMS (Information Management System), 751

Attach Facility, 752-755

deadlocks, 759-760

DL/I batch interface, 761-762

programs, 751-752

RTT (resource translation table), 756

SYSGEN, 760

IMS commands, 1361-1362

IMS/TM, 751-752

batch processing, 771

COMMIT, 756-759

feasibility, 770-771

resource availability, 769

restart capabilities, 759

threads, 756-757

transaction design, 762-763

IN clause, 74

IN DATABASE clause, 278

IN lists, 91-92

in-memory table caches, 858-859

inactive DBATs (database access threads), 1469

INCLUDE statement, 602

incremental image copy, 1202-1203

Incremental Image Copy JCL listing (33.2), 1204

Independent Software Vendors (ISVs), 1424

index access

eliminating, 87

multiple, 839-840

index compression, 51

Index Copy JCL listing (32.3), 1205

INDEX database objects, 38

index keys, controlling, 344-345

index lookaside, 50, 845

index pages, 802-808

index scans

matching, 834-835

non-matching, 835-837

index screening, 837-838

index spaces, copying, 1204-1205

index versions, REBUILD_INDEX utility, 1234

index-only access, 50, 345, 838-839

indexable predicates, 68-71, 78-80

indexed access, 832-833

indexed columns

code predicates, 68

joins, 101

indexes, 313, 324-326

buffer pools, 349

column cardinality, 335

columns, 344

adding, 343, 362-363

uniqueness, 342

compression, 336-337

creating, 326-335, 348-349

Data Partitioned Secondary Indexes (DPSIs), 332-334

direct index lookup, 833-834

DPSIs (data partitioned secondary indexes), 54

expressions, 54

foreign keys, 343

hashing, 55

image copies, 351

levels, 335

modifying, 335-336

multi-index access, 346-347

multi-column indexes, 345-346

Non-Partitioned Secondary Indexes (NPSIs), 332-333

NPIs (non-partitioned indexes), 793

online schema, changing, 362-365

page sizes, 336-337

altering, 364

primary keys, 343

renaming, 365

specifying free space for, 347-348

table space data, 349

variable index keys, changing treatment of, 364-365

virtual, creating, 349

XML, creating, 425-426

indexing

auxiliary tables, 351

data modification, 342

detriments, 343

expressions, 330-331

partitioning, 331-334

tables, 344

VARCHAR columns, 263

variable columns, 329-330

workloads, 341-342

XML data, 418-419

INDEXONLY column (PLAN_TABLE), 987

INDEXSPACE column (SYSIBM.SYSINDEX-SPACESTATS), 1052

INDEXVAL phase (LOAD utility), 1244

indicator variables, 116-117, 255

industry periodicals, 1427-1428

***Information Management*, 1427**

Information Management System (IMS). *See* IMS (Information Management System)

Information On Demand (IOD) Conference, 1427

Information Server (IBM), 1521

information-gathering commands, 1343-1355

informational constraints, 296

InfoSphere, 1521

InfoSphere Replication Server, 1521

INFULL function, 146

inheritance, FIELDPROC, 1527

inline LOBs (large objects), 396

inline SQL scalar functions, 178

inline views, 105

INSENSITIVE scrollable cursors, 515-516

INSERT, 126, 129

versus LOAD utility, 1244-1245

INSERT function, 146

INSERT operation, multi-row, 54

INSERT statement, 35-36, 130-133

INSERT statements, 526

inserting XML data, 423-424

INSERTs, 527

INSTALL SYSADM group-level authorization, 41

INSTALL SYSOPR group-level authorization, 41

INSTANCE column (SYSIBM.SYSINDEX-SPACESTATS), 1052

INSTANCE column (SYSIBM.SYSTABLESPACE-STATS), 1050

INSTEAD OF triggers, 386, 392

examples, 387-388

restrictions, 386-387

Instrumentation Facility Component Identifiers (IFCIDs), 937-938

Instrumentation Facility Interface (IFI), 53, 942

INSZERT, 527

INT tools, 1406

INTEGER columns, 264-265

INTEGER data type, 38, 244

INTEGER function, 147

Integrated Information Processor (zIIP), 1075

integrity

data, 290

triggers, 300

referential, 290-292

avoiding, 295-296

check constraints, 297-300

implementation restrictions, 296

informational constraints, 296

programmatic, 295

referential constraints, 291-294

referential sets, 294-295

self-referencing constraints, 296

integrity tools, 1406

Internet applications, 695

interfaces, CICS, managing, 740

internal I/O, 1080-1081

International DB2 Users Group (IDUG), 698-699, 1427

International Technical Support Organization (ITSO), 698

Internet, 689

DB2, access, 692-695

Google Groups, 691-692

mailing lists, 691-692

archives, 700

digesting, 700

privacy, 700

resources, 696-697

search engines, 702

Usenet Newsgroups, 691-692, 701-702

WWW (World Wide Web), 689-690

Internet enabling tools, 1408

Internet-based programming languages, 486

INTERSECT set operations, 26

INVALIDATE option (AUXERROR), 1181

invoking UDFs (user-defined functions), 184

IOD (Information On Demand) Conference, 1427

IPLIST table (Catalog), 875

IPNAMES table (Catalog), 875

IRLM, 1073

tuning, 1114

IRLM commands, 1364-1365

IRLM parameters, 918

ISOBID column (SYSIBM.SYSINDEX-SPACESTATS), 1052

isolation level, SQL statement, 535-536

ISPF tables, 724-725

ISV tools, third-party, 354

ISVs (Independent Software Vendors), 1424

ITSO (International Technical Support Organization), 698

J

Java, 488, 555
- applets, 555-557
- application development, 554-563
- applications, 555-557
- Enterprise Java Beans (EJBs), 557
- Java Database Connectivity (JDBC) versus SQLJ, 557-559
- Java Virtual Machines (JVMs), 554
  - bytecodes, 555
- program preparation, 607
- servlets, 555-557

Java Database Connectivity (JDBC) versus SQLJ, 557-559

Java Virtual Machines (JVMs), 554
- bytecodes, 555

JCL for Full Recovery listing (33.7), 1227

JCL for Partial Recovery listing (33.8), 1227

JCL to issue I DB2 Command in Batch listing (37.1), 1341

JCL to Run a DL/I Batch DB2 Program listing (18.4), 761-762

JDBC (Java Database Connectivity) versus SQLJ, 557-559

JDBC Code Fragment listing (11.6), 558

job streams, reorganization, 1272-1273

joining
- non-relational data, 534-535
- tables, 15-20, 531-533

joins, 91, 100
- clustered columns, 101
- hybrid, 51, 851-854, 859
  - parallelism, 867
- indexed columns, 101
- LEFT OUTER JOIN, 103
- merge scan, 850-851, 859
  - parallelism, 867
- methods, 51
- nested loop, 51, 846-849, 859
  - parallelism, 867
- ORDER BY, specifying, 101
- outer, 26-29, 102
- parallel, 866-867
- reducing, 99
- RIGHT OUTER JOIN, 103
- SQL, 15-19, 22-23
  - Cartesian products, 19-20
- star, 854-858
- versus subqueries, 100-101

JOIN_DEGREE column (PLAN_TABLE), 989

JOIN_PGROUP_ID column (PLAN_TABLE), 989

JOIN_TYPE column (PLAN_TABLE), 989

JULIAN DAY function, 147

JVMs (Java Virtual Machines), 554
- bytecodes, 555

K–L

KEEP UPDATE LOCKS clause, 90

KEEPDYNAMIC parameter, dynamic SQL, 591

Kerberos security, 482-483

key ranges, table space partitioning, 89-90

L-locks, 914

label-based access control (LBAC), 461-466
- referential integrity, 465
- restrictions, 465-466
- row level granularity, 462-464

labels, columns, specifying, 277-278

languages. *See* programming languages

large objects (LOBs), 394-395, 403-407

columns, 395-396

FETCH WITH CONTINUE, 402

file reference variables, 402

inline, 396

locking, 403

logging, 395

materialization, 402

size considerations, 395

variable declarations, 401

large partitioned table spaces, 208-209

Larsen, Sheryl, 698

LAST DAY function, 147

LASTUSED column (SYSIBM.SYSINDEX-SPACESTATS), 1052

latches versus locks, 892

LBAC (label-based access control), 461-466

referential integrity, 465

restrictions, 465-466

row level granularity, 462-464

LDR (Logical Design Review), 1438

LEAST function, 147

LEFT function, 147

LEFT OUTER JOIN, 28, 103

LENGTH function, 147

less than or equal to predicate, 74

levels

DRDA, 1453-1455, 1460

indexes, 335

LIB parameter (DSNUPROC), 1156

LIKE, 74

LIKE clause, duplicating table schema, 275

LIKE predicate, 75-77

limitations, triggers, 389

limited block fetches, 1490

limited block protocols, 1451

limited scanning, partitions, 52

linkage editor, 655

linking programs, 607

list prefetch, 841-842

list prefetch performance factor, 50

LISTDEF control statement, 1370

LISTDEF utility, 1159-1160, 1165-1166

creating lists, 1160-1162

list expansion, 1163-1165

wildcarding, 1162-1163

listings

Accounting Report Additional Information, 951-952

Accounting Report Buffer Pool Information, 953-954

Accounting Report Database Code Usage, 952

Accounting Report Locking Activity, 950-951

Accounting Report Program Terminations, 951

Accounting Report RID List Details, 956

Audit Summary Report, 956-957

Batch JCL for a TSO/DB2 Program, 708-709

Bind CLIST, 621-622

CHECK DATA JCL, 1178

CHECK DATA JCL (for LOB References), 1180

CHECK DATA JCL (for XML References), 1182

CHECK INDEX JCL, 1188-1189

CHECK LOB JCL, 1186-1187

Checking for DB2 Availability, 750

COBOL Program Using EXECUTE IMMEDIATE, 577

COBOL Program Using Non-SELECT Dynamic SQL, 578-579

COBOL Stored Procedure Shell, 662

COPYTOCOPY JCL, 1217

Cursor Processing, 512-513

DB2 Accounting Report, 944
DB2 Statistics Buffer Pool General Information, 961
DB2 Statistics Buffer Pool Read Information, 962
DB2 Statistics Buffer Pool Write Information, 963
DB2 Statistics Common Storage Usage, 965
DB2 Statistics EDM Pool Activity Information, 965-966
DB2 Statistics Locking Activity Information, 964-965
DB2 Statistics Log Activity Information, 963-964
DDL to Create the DSN_FUNCTION_TABLE, 1002
DDL to Create the DSN_STATEMNT_TABLE, 998
DDL to Create the PLAN_TABLE, 982-984
DIAGNOSE JCL, 1200
DSN1CHKR JCL, 1318-1319
DSN1COMP JCL, 1320
DSN1COPY JCL, 1324
DSN1COPY JCL (Using the OBIDXLAT Option), 1325-1326
DSN1LOGP JCL, 1330
DSN1PRNT JCL, 1331
DSN1SDMP JCL, 1329
DSNJLOGF JCL, 1315-1316
DSNJU003 JCL (Change Log Inventory), 1316
DSNJU004 JCL (Print Log Map), 1318
DSNTEP4 JCL, 1332-1333
DSNTIAD JCL, 1335
DSNTIAUL JCL, 1336-1337
Fixed-List SELECT Dynamic SQL, 581
I/O Activity Summary Report, 957-959
Image copy JCL, 1167, 1203-1204
Incremental Image Copy JCL, 1204
Index Copy JCL, 1205
JCL for Full Recovery, 1227
JCL for Partial Recovery, 1227
JCL to issue I DB2 Command in Batch, 1341
JCL to Run a DL/I Batch DB2 Program, 761-762
JDBC Code Fragment, 558
LOAD JCL (Nonrestartable), 1242-1243
LOAD JCL (Restartable), 1241-1242
Long Accounting Report, 946-947
Long Accounting Report Highlights, 948
Long Accounting Report SQL Activity, 949
MERGECOPY JCL, 1218-1219
MODIFY RECOVERY JCL, 1291
MODIFY STATISTICS JCL, 1294-1295
Non-SELECT Dynamic SQL Using Parameter Markers, 579-580
Precompile, Compile, and Link CLIST, 620
Pseudo-code for Retrieving Data from an Application Join, 532
Pseudo-code for Retrieving Data from an SQL Join, 531-532
QMF Form to be Used with the DBID/PSID/OBID Query, 1389
QMF Form to be Used with the SYSCOPY Query, 1382
QUIESCE JCL, 1220-1221
REBUILD INDEX JCL, 1233
RECOVER INDEXSPACE JCL, 1228
REORG JCL (Nonrestartable), 1267-1268
REORG JCL (Restartable), 1266-1267

REPAIR DBD JCL, 1192-1193

REPAIR LOCATE JCL, 1194

REPAIR SET JCL, 1196-1197

REPORT RECOVERY JCL, 1236

REPORT TABLESPACESET JCL, 1199

Results of the DISPLAY GROUP Command, 786

Running a DB2 Program in TSO Batch, 609

RUNSTATS INDEX JCL, 1298

RUNSTATS TABLESPACE JCL, 1296-1297

Sample COBOL Error-Handling Paragraph, 497-499

Sample Program Preparation Procedure, 616-618

Sample Results of DISPLAY BUFFERPOOL, 1346

Sample Results of DISPLAY LOG, 1347

SQLDA, 582-583

SQLJ Code Fragment, 558

STOSPACE JCL, 1311

Typical Processing Scenario, 890

UNLOAD JCL, 1260

Updating with a Cursor, 514-515

Varying-List SELECT Dynamic SQL, 585

Varying-List SELECT Dynamic SQL with Minimum SQLDA, 587

lists

creating, LISTDEF utility, 1160-1162

expanding, LISTDEF utility, 1163-1165

IN, 91-92

literals

explicitly coding, 533

UNION query, 25

LITERAL_REPL column (DSN_STATEMENT_CACHE_TABLE), 1005

LOAD JCL (Nonrestartable) listing (33.2), 1242-1243

LOAD JCL (Restartable) listing (34.1), 1241-1242

load module program preparation objects, 631

load tools, 1406

LOAD utility, 348, 527, 1240-1243, 1252-1259

concurrency, 1250-1251

creating flash copy, 1245

creating inline copy, 1245

gathering inline statistics, 1245

loading delimited input data sets, 1246

locking, 1250-1251

phases, 1243-1244

rerun/restart procedures, 1246-1250

sorting, 1251-1252

versus INSERT, 1244-1245

loading tables, 1240-1243

LOADRLASTTIME column (SYSIBM.SYSINDEX-SPACESTATS), 1051

LOADRLASTTIME column (SYSIBM.SYSTABLESPACE-STATS), 1049

LOBs (large objects), 265, 394-395, 403-407

accessing, 399-402

columns, 395-399

FETCH WITH CONTINUE, 402

file reference variables, 402

inline, 396

locking, 403, 914-915

duration, 915-916

logging, 395

materialization, 402

pages, 801-802

reference checking, 1179-1181

RTS (Real Time Statistics), 1271-1272

size consideration, 395

table spaces, 215-216

locks, 916

UDTs (user-defined data types), 192-193

variable declarations, 401

LOCATE function, 147-148

LOCATE IN STRING function, 148

LOCATION traces, 971
LOCATIONS table (Catalog), 875
LOCK TABLE command, 919
batch programming, 538
locked data, skipping, 133-134
locked rows, skipping, 909-911
locking, 889-892
CHECK_DATA utility, 1183
CHECK_INDEX utility, 1189
CHECK_LOB utility, 1187
global, management, 911-914
hierarchical, 913
LOAD utility, 1250-1251
LOB (large objects), 403, 914-916
long accounting reports, activity, 950-951
UNLOAD utility, 1262
locking activity information statistics report, 964
locking considerations, CHECK_DATA utility, 1183
locking reports, 959
LOCKMAX parameter, 918, 220
LOCKPART parameter, 220
locks, 917-923
avoidance, 51, 788, 908-911, 919
deadlocks, 901-904
duration, 892-895
escalation, 51, 917
L-locks, 914
P-locks, 913
pages, 898-901
promotions, 917
rows, 899-901
S-locks, 914
structures, 913
suspensions, 901-904
table spaces, 895-897
LOBs, 916
tables, 897-898
timeouts, 901-904
user escalation, 917
versus latches, 892
X-locks, 914
LOCKSIZE ANY, 918
LOCKSIZE parameter, table space, 218-220
log activity information statistics report, 963-964
log I/O, 1082-1083
log offloading, 922
LOG phase (REORG INDEX utility), 1273
LOG phase (REORG TABLESPACE utility), 1274
log preformat utility, 1315-1316
Log Record Sequence Numbers (LRSN), 784
LOG10 function, 148
LOGAPPLY phase (COPY), 1205
LOGAPPLY phase (LOAD utility), 1244
LOGAPPLY phase (RECOVER utility), 1229
LOGAPPLY phase (RESTORE_SYSTEM utility), 1239
LOGCSR phase (LOAD utility), 1244
LOGCSR phase (RECOVER utility), 1229
logging LOBs (large objects), 395
logic, triggers, testing, 389
Logical Design Review (LDR), 1438
logs, archiving to disk, 789
LOGSCR phase (COPY), 1206
LOGUNDO phase
COPY utility, 1206
LOAD utility, 1244
RECOVER utility, 1229
Long Accounting Report Highlights listing (24.3), 948
Long Accounting Report listing (24.2), 946-947
Long Accounting Report SQL Activity listing (24.4), 949
long accounting reports, 946-949, 951-952, 956
buffer pool information, 952, 954-955
database code usage information, 952

locking activity, 950-951
program status, 951
SQL activity, 949-950
LONG VARCHAR data type, 675
LONG VARGRAPHIC data type, 675
loop joins, nested, 846-849
LOWER function, 148
LPAD function, 148
LPFACILITY column (SYSIBM.SYSTABLESPACE-STATS), 1051
LPL pages, recovery, 785
LRSN (Log Record Sequence Numbers), 784
LTRIM function, 148
LULIST table (Catalog), 875
LUMODES table (Catalog), 875
LUNAME column (RLST), 1146
LUNAMES table (Catalog), 875

M

mailing lists, 691-692, 696-697, 1429
archives, 700
digesting, 700
maintenance, rebinding, 1015
managing data warehouses, 1520
MATCHCOLS column (PLAN_TABLE), 987
matching index scans, 834-835
materialization, LOB (large objects), 402
Materialized Query Tables (MQTs), 53, 201, 1522-1523, 1527, 1532-1533
attribute copy options, 1526
automatic query rewrite, 1528-1532
benefits, 1523
converting tables into, 1527-1528
creating, 1523-1524
population and maintenance, 1528
query optimization options, 1524-1526
refreshable options, 1524
MAX function, 139, 148
MAXROWS parameter, hashing, 352
MAXROWS parameter (DSN1COMP utility), 1321
MAXROWS parameter, table space, 233
MAX_PAR__DEGREE column (DSN_USERQUERY_TABLE), 1133
MEMBER CLUSTER, 778
MEMBER CLUSTER parameter, table space, 234
memory structures, 1066-1067
buffer pools, 1066-1067
EDM pools, 1069-1073
IRLM, 1073
open data sets, 1073-1074
RID pools, 1067
sort pools, 1067-1069
total memory requirements, 1074
working storage, 1073
memory usage, tuning, 1064-1074
MERGC column (PLAN_TABLE), 992
merge scan joins, 850-851, 859
parallelism, 867
merge scans, 51
MERGE statement, 129-132
MERGECOPY JCL listing (33.5), 1218-1219
MERGECOPY utility, 1165, 1218-1220
phases, 1219
MERGE_JOIN_COLS column (PLAN_TABLE), 989
MERGN column (PLAN_TABLE), 992
messages, console, viewing, 972-977
metadata, 1414-1415
data warehouses, 1511
METHOD column (PLAN_TABLE), 986
methods, program preparation, 619-622
MICROSECOND function, 148
Microsoft .NET, 694

MIDAW (Modified Indirect Access Word), 1077

middleware resource usage, limiting, 1147-1149

MIDNIGHT SECONDS function, 148

MIG tools, 1407

migration procedures, 1436

migration tools, 1407

MIN function, 139-140, 148

MINUTE function, 149

mirror tables, denormalization, 282

miscellaneous tools, 1407

MIXOPSEQ column (PLAN_TABLE), 988

MLS (multi-level security), 461-466

- LBAC, 465-466
- row level granularity, 462-464

MOD function, 149

MODESELECT table (Catalog), 875

Modified Indirect Access Word (MIDAW), 1077

modified source program preparation objects, 631

MODIFY RECOVERY JCL listing (35.1), 1291

MODIFY RECOVERY utility, 1165, 1290-1291

- phases, 1292-1293

MODIFY STATISTICS JCL listing (35.2), 1294-1295

MODIFY STATISTICS utility, 1165, 1293-1295

modifying data, SQL, 35-37

monitor trace, 933-934

monitoring

- data sharing groups, 786
- dynamic SQL, 575-576
- objects, 1021
  - Catalog, 1021-1048
  - RTS (Real Time Statistics), 1048-1058
  - rules, 1058-1059
- utilities, 1156-1158
- variable columns, 263-264

monitoring performance, 928-929

- continuous monitoring, 967
- displaying resource status, 977-979
- exception-based monitoring, 967
- periodic monitoring, 967
- profiles, 970-972
- reports, 940-943
  - accounting, 943-956
  - audit, 956-957
  - explain, 957
  - I/O activity, 957, 959
  - locking, 959
  - record trace, 959
  - SQL trace, 959-960
  - statistics, 960-966
  - system parameters, 966-967
- strategies, 967-970
- traces, 929-930
  - accounting, 930-931
  - audit, 931-932
  - destinations, 936
  - global, 933
  - guidelines, 938-940
  - IFCIDs (Instrumentation Facility Component Identifiers), 937-938
  - monitor, 933-934
  - performance, 934-935
  - statistics, 935-936
- viewing console messages, 972-977
- z/OS, 979

MONTH function, 149

MONTHS BETWEEN function, 149

MOTs (materialized query tables), 53

MQSeries scalar functions, 159-161

MQTs (Materialized Query Tables), 1522-1523, 1527, 1532-1533

- attribute copy options, 1526
- automatic query rewrite, 1528-1532
- benefits, 1523
- converting tables into, 1527-1528
- creating, 1523-1524
- population and maintenance, 1528

query optimization options, 1524-1526

refreshable options, 1524

MSC tools, 1407

Mullins, Craig S., 690

multi-index access, 346-347

multi-row fetch cursors, 518-520

multi-row fetches, 109-110

multi-table access path strategies, 846-860

multi-table table spaces, 235-237

multi-tier processing, 1472

multi-column indexes, 345-346

multi-level security

label-based access control (LBAC), 461-466

referential integrity, 465

restrictions, 465-466

row level granularity, 462-464

multiple index access, 839-840

multiple rows

inserting, rowset positioning cursors, 520

retrieving, 535

updating, 126

multiple tables, retrieving data from, 91-110

MULTIPLY ALT function, 149

multi-row FETCH operation, 54

multi-row INSERT operation, 54

N

NACTIVE column (SYSIBM.SYSINDEX-SPACESTATS), 1051

NACTIVE column (SYSIBM.SYSTABLESPACE-STATS), 1049

NAME column (SYSIBM.SYSINDEX-SPACESTATS), 1052

NAME column (SYSIBM.SYS-TABLESPACESTATS), 1050

namespace declarations, XML indexes, 419

namespace URIs, specifying, 417

namespaces, XML, 417-418

naming

columns, 252

packages, 501-504

plans, 501-504

programs, 501-504

triggers, 388

UDFs (user-defined functions), 180

UDTs (user-defined data types), 197

variables, 501-504

naming conventions, 313-314, 321, 1435-1436

compliance, 321

data sharing, 782-783

developing, 314-316

enforcing, 314-316

samples, 316-320

native SQL stored procedures, 672-674

navigational queries, Catalog, 1023-1031

nested loop joins, 51, 846-849, 859

parallelism, 867

nesting stored procedure calls, 663-664, 674

.NET, 694

NET tools, 1408

Net.Data, 695

Netezza (IBM), 1521

network traffic, minimizing, 695

newsgroups, 691-692, 696, 701-702

NEXT DAY function, 149

NGBPDEP, 791

NLEAF column (SYSIBM.SYSINDEX-SPACESTATS), 1051

NLEVELS column (SYSIBM.SYSINDEX-SPACESTATS), 1051

non-inline SQL scalar functions, 179

non-matching index scans, 835-837

non-partitioned indexes (NPIs), 793

non-partitioned secondary indexes (NPSIs), 332-333

non-relational data

accessing, SQL, 274

joining, 534-535

non-SELECT class (dynamic SQL), 578-581

Non-SELECT Dynamic SQL Using Parameter Markers listing (14.3), 579-580

non-shared objects, 776

non-standard dates, 121-122

non-uniform distribution statistics, deleting, 1135-1136

non-updatable views, 312

nonindexable predicates, 70

nonsargable processing, 50

normalization, 278-279

NORMALIZE DECFLOAT function, 149

NORMALIZE STRING function, 149

NOT, 73

NOT DETERMINISTIC parameter, UDFs (user-defined functions), 188

NOT EXISTS statement, 91

NOT IN statement, 91

NPAGES column (SYSIBM.SYSINDEX-SPACESTATS), 1051

NPAGES column (SYSIBM.SYSTABLESPACE-STATS), 1049

NPIs (non-partitioned indexes), 793

NPSIs (non-partitioned secondary indexes), 332-333

NULL, 117-118

null indicators, 604

- host variables, 507-509

null input arguments, UDFs (user-defined functions), 189

nullable columns, 254-257

NULLIF function, 149

nulls, 115, 165

- detriments, 115
- indicator variables, 116-117
- NULL, 117-118

NUMPARTS parameter, table space, 231

O

OBID parameter, tables, 267-268

OBIDs, 794

object migration tools, 1407

object-oriented (OO) technology, 393-394

object-oriented programming (OOP), 393

object-relational mapping frameworks, 486

object/relational, 393-394

objects, 37-38, 200-201

- LOBs (large objects), 265, 394-395, 403-407
  - accessing, 399
  - columns, 395-396
  - FETCH WITH CONTINUE, 402
  - file reference variables, 402
  - inline, 396
  - locking, 403, 914-916
  - logging, 395
  - materialization, 402
  - pages, 801-802
  - reference checking, 1179-1181
  - RTS (Real Time Statistics), 1271-1272
  - size consideration, 395
  - table spaces, 215-216, 916
  - UDTs (user-defined data types), 192-193
  - variable declarations, 401
- monitoring, 1021
  - Catalog, 1021-1048
  - RTS (Real Time Statistics), 1048-1058
  - rules, 1058-1059
- non-shared, 776
- object-relational mapping frameworks, 486
- program preparation, 631-632
- security controls, 40-42
- sequence, 246-251, 549
- shared, 776
- triggers, 374-376

Objects area (Catalog), 880

ODR (Organization Design Review), 1438

ODS (Operational Data Store), 1508-1509

offline copy utility, 1322-1328

offloading logs, 922

OLAP (On-Line Analytical Processing), 1509-1510

OLAP Server, 1522

OLTP (Online Transaction Processing) stored procedures, 11, 670

On-Line Analytical Processing (OLAP), 1509-1510

one-fetch index access, 842-843

online applications, 705-706

online design techniques, TSO (Time-Sharing Options), 709-712

online performance monitoring, 940-943

online programming, 547-552

online reorganization, REORG, 1279-1283

online reporting, 969-970

online return codes, utilities, 1366

online schema, 354-357

- change support, 355-357
- changes, versioning, 370-372
- changing column details, 357-359
- changing columns, 361-362
- changing data types, 359-361
- changing indexes, 362-365
- changing table space partitioning, 365-368
- changing table space type, 368
- pending definition changes, 369-370
- renaming columns, 361

Online Transaction Processing (OLTP) stored procedures, 11, 670

online utilities, monitoring, 1156-1158

OO (object-oriented) technology, 393-394

OOP (object-oriented programming), 393

open data sets, 1073-1074

OpenPages, 1522

operation system exploitation, 50

operational data versus data warehousing, 1508

Operational Data Store (ODS), 1508-1509

operational support, 1440-1441

operational support tools, 1408

operations

- CAST, 163
- FETCH, 54
- INSERT, 54
- set, 23, 91
  - EXCEPT, 26
  - INTERSECT, 26
  - union, 23-26
- SQL, 13-14

OPR tools, 1408

OPTHINT column (PLAN_TABLE), 990

OPTHINT parameter (BIND), 654

optimization

- data types, 260-261
- distributed, 1500-1501
- safe queries, 54

OPTIMIZE FOR 1 ROW clause, 87

OPTIMIZE FOR n ROWS clause, 86, 1494

- SELECT statement, 1123-1124

optimizer, 45-46, 816-818, 868-869

- access paths
  - influencing, 46-48
  - strategies, 824-860
- CPU costs, 818
- EXPLAIN, 980-982, 1002, 1005-1011
  - access paths, 993-998
  - DSN_COLDIST_TABLE, 1002
  - DSN_DETCOST_TABLE, 1003
  - DSN_FILTER_TABLE, 1003
  - DSN_FUNCTION_TABLE, 1001-1002
  - DSN_KEYTGTDIST_TABLE, 1002
  - DSN_PGRANGE_TABLE, 1003

DSN_PGROUP_TABLE, 1003
DSN_PREDICAT_TABLE, 1002
DSN_PTASK_TABLE, 1003
DSN_QUERY_TABLE, 1003
DSN_SORTKEY_TABLE, 1003
DSN_SORT_TABLE, 1003
DSN_STATEMENT_CACHE_TABLE, 1002-1005
DSN_STATEMENT_TABLE, 998-1001
DSN_STRUCT_TABLE, 1003
DSN_VIEWREF_TABLE, 1003
PLAN_TABLE, 982-993, 1006
hints, 48-49
I/O cost, 818
influencing, 1118-1123
physical data independence, 817-818
predicates, filter factor, 821-823
query parallelism, 861-868
screening, 823
SQL statement, 821
statistics, 816
Catalog, 819-821
subqueries, 869-871
viewing optimization, 871-873

options, z/OS, 1084-1087

OPTIONS utility, 1172

ORDER BY, 108, 533
specifying, 101

ORDER BY clause, 30, 68, 105

Organization Design Review (ODR), 1438

organization utilities
LOAD, 1240-1243, 1252-1259
concurrency, 1250-1251
creating flash copy, 1245
creating inline copy, 1245
gathering inline statistics, 1245
loading delimited input data sets, 1246
locking, 1250-1251
phases, 1243-1244
rerun/restart procedures, 1246-1250
sorting, 1251-1252
versus INSERT, 1244-1245
REORG, 1265-1272, 1283-1288
gathering inline statistics, 1278-1279
job streams, 1272-1273
online reorganization, 1279-1283
phases, 1273-1275
reorganization frequency, 1268-1269
rerun/restart procedures, 1275-1278
RTS (Real Time Statistics), 1269-1272
SHRLEVEL parameter, 1279
UNLOAD, 1260-1265
locking, 1262
phases, 1261
restarting, 1261
termination, 1261
versus DSNTIAUL, 1262

OTHER_OPTIONS column (DSN_USERQUERY_TABLE), 1133

OTHER_PARMS column (DSN_USERQUERY_TABLE), 1134

outer joins, 26-28, 102
types, 28-29

overactive data areas, 548-549

OVERLAY function, 149

P

P-locks, 913

PACKADM group-level authorization, 41

PACKAGE column (DSN_USERQUERY_TABLE), 1133
PACKAGE privilege class, 450
package program preparation objects, 631
PACKAGE security class, 40
packages
administration, 624-626
benefits, 624
DBRMs, 623-627
list size considerations, 627
naming, 501, 503-504
performance, 627
static SQL, 1465
systematic rebinding, 625
triggers, 384-385
version maintenance, 625-626
versions, 629-630
pages
index, 336-337, 802-808
altering sizes, 364
LOB, 801-802
locks, 898-899, 901
ranges, screening, 827-828
structures, 797-798
table space data pages, 799-801
types, 798-799
PAGESIZE parameter (DSN1COPY), 1327
PAGE_RANGE column (PLAN_TABLE), 989
paging I/O, 1083-1084
Pair Wise Joins, 857
Pair Wise Star Join, 856-857
parallel joins, 866-867
parallel processing, batch programming, 536-538
Parallel Sysplex, 773-774
parallelism
dynamic SQL, 596
hybrid joins, 867
merge scan joins, 867
nested loop joins, 867
optimization, 865
queries, 52, 859-867
disabling, 868
restrictions, 866
sysplex, 788
zIIPs, 865-866
PARALLELISM_MODE column (PLAN_TABLE), 989
parameter data types, UDFs (user-defined functions), 185
parameter markers, dynamic SQL, 595-596
parameters
BIND, 618, 916
ACTION, 639
BIND PACKAGE, 640
BIND PLAN, 639
CACHESIZE, 647
CONCURRENT-ACCESSRESOLUTION, 643
DYNAMICRULES, 652-653
IMMEDWRITE, 654
OPTHINT, 654
PATH, 652
row locks, 893-895
table space locks, 892-893
buffer pool, 1096-1101
AUTOSIZE, 1105-1106
DWQT, 1104
PGFIX, 1105
PGSTEAL, 1104-1105
sequential steal, 1102
VPSEQT, 1102-1103
CHECK_DATA utility
AUXERROR, 1181
SCOPE, 1180, 1184
CHECK_LOB utility, 1188
databases, specifying, 203
DISCONNECT, 1462
DSNUPROC, 1156
DSNZPARM, 917
MAXROWS, 352
QUALIFIER, 638
RDO (Resource Definition Online), 732-739
SEQUENCE objects, 249-250
SSM, 753
table spaces, 217-235
BUFFERPOOL, 229
CCSID, 233
CLOSE, 229-231
DSSIZE, 217-218

ERASE, 231
FREEPAGE, 227-229
LOCKMAX, 220
LOCKPART, 220
LOCKSIZE, 218-220
MAXROWS, 233
MEMBER CLUSTER, 234
NUMPARTS, 231
PCTFREE, 227-229
PRIQTY, 222-226
SECQTY, 222-226
SEGSIZE, 231
TRACKMOD, 234
USING, 221-222
tables, 266-270
TRACKMOD, 777
TSO (Time-Sharing Options), 707-708
UDFs (user-defined functions), 185
z/OS, 1084-1087

PARENT_PLANNO column (PLAN_TABLE), 991

PARENT_QBLOCKNO column (PLAN_TABLE), 991

parsing data, XML, 415

participants, two-phase commit, 1468

PARTITION column (SYSIBM.SYSINDEXSPACESTATS), 1052

PARTITION column (SYSIBM.SYSTABLESPACESTATS), 1050

partition statistics queries (Catalog), 1036-1037

partition-by-growth (PBG) universal table space, 207

partition-by-range (PBR) universal table space, 207

partitioned table spaces, 207-208, 238
advantages, 213-214
changing, 210
classic, 216
data sets, 210
disadvantages, 214-215
large, 208-209
scans, 826
SMS, 241-242
versus multiple tables, 211-213

partitioning
indexing, 331-334
table space, changing, 365-368

partitions
boundaries, changing, 367
DPSIs (data partitioned secondary indexes), 54
independence, 52, 904-907
limited scanning, 52
rebalancing, 367-368
rotating, 366-367
tables, adding to, 365-366

PATH parameter, 652

Payment Card Industry Data Security Standard (PCI DSS), 1399

PBG (partition-by-growth) universal table space, 207

PBR (partition-by-range) universal table space, 207

PC tools, 1408

PCI DSS (Payment Card Industry Data Security Standard), 1399

PCTFREE parameter (DSN1COMP utility), 1321

PCTFREE parameter, table space, 227-229

PDR (Physical Design Review), 1438

Peer-to-Peer Remote Copy (PPRC), 1387

pending definition changes, online schema, 369-370

PENDING option (SCOPE parameter), 1180

pending states, 1372-1373
correcting, 1374
reasons, 1373

performance
distribution, performance problems, 1491-1496
packages, 627

Performance area (Catalog), 881

performance factors
queries, 57-58
SQL, 45-55

performance monitoring, 928-929
continuous monitoring, 967
exception-based monitoring, 967

periodic monitoring, 967
profiles, 970-972
reports, 940-943
accounting, 943-956
audit, 956-957
explain, 957
I/O activity, 957-959
locking, 959
record trace, 959
SQL trace, 959-960
statistics, 960-966
system parameters, 966-967
resources, displaying status, 977-979
strategies, 967-970
traces, 929-930
accounting, 930-931
audit, 931-932
destinations, 936
global, 933
guidelines, 938-940
IFCIDs (Instrumentation Facility Component Identifiers), 937-938
monitor, 933-934
performance, 934-935
statistics, 935-936
viewing console messages, 972-977
z/OS, 979

performance monitors, 1410-1411

performance problems, troubleshooting, 1137-1142

performance tools, 1411

performance trace, 934-935

periodic maintenance, rebinding, 1015

periodic monitoring, 967

periodicals, 1427-1428

personally identifiable information (PII), 429

PGFIX parameter (buffer pool), 1105

PGSTEAL parameter (buffer pool), 1104-1105

phases
BACKUP_SYSTEM utility, 1237-1238
CHECK_DATA utility, 1182
CHECK_INDEX, 1189
CHECK_LOB utility, 1187
COPY utility, 1205-1206
COPYTOCOPY utility, 1216
LOAD utility, 1243-1244
MERGECOPY utility, 1219
MODIFY RECOVERY utility, 1292-1293
MODIFY STATISTICS utility, 1294-1295
QUIESCE utility, 1222
REBUILD_INDEX utility, 1234
RECOVER utility, 1228
REORG INDEX, 1273-1275
REPAIR utility, 1191
RUNSTATS utility, 1298
STOSPACE utility, 1311
UNLOAD utility, 1261

physical analysis queries, Catalog, 1031-1036

physical data independence, 817-818

Physical Design Review (PDR), 1438

physical storage of data, 792-808

PIECESIZE clause, 350-351

PII (personally identifiable information), 429

PKGNAME traces, 971

plan and package analysis tools, 1409-1410

PLAN privilege class, 450

plan program preparation objects, 632

PLAN security class, 40

plan stored procedures, 675

PLANNAME column (RLST), 1146

PLANNAME traces, 971

PLANNO column (PLAN_TABLE), 986

plans
CICS, 745
DBRMs, 622
naming, 501, 503-504

PLAN_TABLE (EXPLAIN), 62, 982-984
columns, 986-993, 1006
querying, 984-986
PLN tools, 1409-1410
PM tools, 1410-1411
political issues, 1441-1443
pool thread attributes, DB2CONN parameter (RDO), 735-736
pools
EDM, 1069-1073
sort, 1067-1069
populating
data warehouses, 1513
data cleansing, 1516-1519
data transformation, 1515-1516
propagation, 1514
replication, 1514
snapshots, 1514
MQTs (Materialized Query Tables), 1528
POSITION function, 150
positioned DELETE statement, 37
positioned UPDATE statement, 36
POSSTR function, 150
Post-Implementation Design Review (Post-IDR), 1440
posters, DB2 Catalog, 885
POWER function, 150
PPRC (Peer-to-Peer Remote Copy), 1387
PRDID traces, 972
Pre-Implementation Design Review (Pre-IDR), 1439
pre-joined tables, denormalization, 281
precision, arithmetic, 78-80
Precompile option (TSO), 717
Precompile panel (DB2I), 612
Precompile, Compile, and Link CLIST listing (15.3), 620
precompiling programs, 604-605, 620, 634-637, 717
predicates, 77-78
BETWEEN, 74
coding, 76-77
filter factor, 821-823
greater than or equal to predicate, 74
indexable, 68-71, 78-80
less than or equal to, 74
LIKE, 75-77
nonindexable, 70
range, dynamic SQL, 572
screening, 823
Stage 1, 69-71
transitive closure rules, 92-93
XMLEXISTS, 425
predictive governing, RLF (Resource Limit Facility), 1144, 1150
predictive resource governing, 597
prefetch
list, 841-842
sequential, 825, 828-831
skip sequential, 841
PREFETCH column (PLAN_TABLE), 988
prefixing columns, 603-604
PRELOGA phase (RECOVER utility), 1229
PRELOGC phase (RECOVER utility), 1229
pretesting embedded SQL, 61
PRF tools, 1411
PRG tools, 1411-1412
primary authids, 449
primary keys
indexes, 343
tables, 258
PRIMARY_ACCESSTYPE column (PLAN_TABLE), 990
PRIMAUTH column (DSN_STATEMENT_CACHE_TABLE), 1005
print log map utility, 1317-1318
PRIQTY parameter, 350
table space, 222-226
privileges, 448
EXPLAIN, 456-457
granting and revoking, 449-450
REFERENCES, 460
SQLADM, 452
Procedural DBA, 683-684
tasks, 684-687

procedural SQL, 6, 678
- benefits, 682
- drawbacks, 682-683
- IBM, 678-680

procedures, 1429
- application development guide, 1434
- batch, program preparation, 616, 618
- data administration, 1432-1433
- database administration guide, 1433
- design reviews, 1436-1440
- migration, 1436
- naming conventions, 1435-1436
- query tool guide, 1435
- responsibilities, 1430-1432
- roles, 1430-1432
- security guide, 1434-1435
- SQL performance guide, 1435
- stored, 656-657, 674-677
  - atomic parameters, 675
  - benefits, 659-661
  - calling, 672
  - coding parameters, 662-663
  - controlling failures, 669-670
  - creating, 666-668
  - developing, 661
  - executing, 670-672
  - external, 676-677
  - IBM Data Studio, 687
  - implementation, 657-666
  - LE/370, 675
  - managing, 668-670
  - native SQL, 672-674
  - nesting, 663-664, 674
  - Online Transaction Processing (OLTP), 670
  - plan, 675
  - procedural DBA, 683-687
  - procedural SQL, 678-683
  - program preparation, 666
  - returning result sets, 664-666
  - reusability, 676
  - SQL, 680-681
  - subprograms, 674
  - supported languages, 661
  - temporary tables, 676
  - versioning, 673-674
  - WLM (Work Load Manager), 670-672
- system administration guide, 1433-1434
- turnover, 1436
- vendor tool guides, 1435

procedures, stored, 1496

processor usage, tuning, 1074-1076

processors, specialty, 812-815

PROCMS column (DSN_STATEMENT_TABLE), 1000

PROCSU column (DSN_STATEMENT_TABLE), 1000

production environment, 11

productivity tools, 487

PROFILEID traces, 971

profiles
- performance monitoring, 970-972
- RUNSTATs, 1302-1303

PROFILE_ENABLED traces, 971

PROFILE_TIMESTAMP traces, 971

PROGNAME column (DSN_STATEMENT_TABLE), 999

PROGNAME column (PLAN_TABLE), 986

program cloning, 536

program logic, 100, 163

program preparation, 601, 609, 632-655
- background, 632
- batch procedures, 616-618
- BIND, 637-654
- BIND command, 605-606
- CICS, 728-729
- CICS processors, 632

CLIST, 618-619
collections, 628-629
compiling programs, 606-607
DB2I, 609-616
DBRM-based plans, converting, 630-631
Declarations Generator, 601-604, 632-634
default names, 632
Java, 607
linkage editor, 655
linking programs, 607
multiple methods, 619-622
objects, 631-632
packages, 623-627
versions, 629-630
plans, 622
precompiling, 604-605, 634-637
REXX EXEC, 618-619
stored procedures, 666

Program Preparation option (TSO), 717

Program Preparation panel (DB2I), 614

program restrictions, external UDFs, 181

program status, long accounting reports, 951

programmatic RI (referential integrity), 295

programmer's aid queries, Catalog, 1037-1041

programming
application development, 486-487, 527-536
active database constructs, 530
"black boxes," 528-529
code modular, 529
cursors, 511-525
data filtering, 531
embedded SQL statements, 487-504, 525-527
error handling, 497-499
host variables, 504-511
stored procedures, 530
unqualified SQL, 530
user-defined functions, 530
batch, 536, 546-547
clustered access, 536
COMMIT statements, 539-541
lock strategies, 538-539
LOCK TABLE command, 538
parallel processing, 536-538
restartable, 543-546
SAFEPOINTs, 542-543
units of work, 541-542
dynamic SQL, 567-569
classes, 576-589, 594
versus static, 569-576
online, 547-552

programming languages
external UDFs, 173
fourth-generation languages, 486
GUI-based, 486
Internet-based, 486
third-generation languages, 486
Web-based, 486

programming tools, 1411-1412

programs
batch, 705-706
TSO (Time-Sharing Options), 708-709
compiling, 606-607
executing, 704-706
IMS, 751-752
linking, 607
naming, 501, 503-504
online, 705-706
precompiling, 604-605
running, 608
sample
dynamic SQL processor, 1332-1334
dynamic SQL update program, 1334-1336
sample unload program, 1336-1339

Programs area (Catalog), 880

PROGRAM_NAME column (DSN_STATEMENT_CACHE _TABLE), 1005

projection operation, SQL, 13-14

promotions, locks, 917

propagation, populating data warehouses, 1514

propagation tools, 1406

Pseudo-code for Retrieving Data from an Application Join listing (13.5), 532

Pseudo-code for Retrieving Data from an SQL Join listing (13.4), 531-532

PSID column (SYSIBM.SYSINDEX-SPACESTATS), 1052

PSID column (SYSIBM.SYS-TABLESPACESTATS), 1050

PSIDs, 794

PUBLIC access authority, 452-453

PUBLIC AT ALL LOCATIONS, 1502

PUNC (Possibly UNCommitted) bit, 908

pureQuery, dynamic SQL, 588-589

pureXML, 408, 412-415

***pureXML Guide*, 416**

Q

QBLOCKNO column (PLAN_TABLE), 986

QBLOCK_TYPE column (PLAN_TABLE), 990, 1006

QM (queue manager), 1486

QMF (Query Management Facility), 722-723, 1522

- forms, 1382-1389
- tools, 1412-1413

QMF Form to be Used with the DBID/PSID/OBID Query listing (39.2), 1389

QMF Form to be Used with the SYSCOPY Query listing (39.1), 1382

QUALIFIER parameter (BIND), 638

QUANTIZE function, 150

QUARTER function, 150

queries

- analysis tools, 64-65
- Catalog, 1047
  - application efficiency, 1041-1042
  - authorization, 1044-1046
  - creating formatted reports, 1047
  - historical, 1043-1044
  - monitoring objects, 1021-1048
  - navigational, 1023-1031
  - partition statistics, 1036-1037
  - physical analysis, 1031-1036
  - programmer's aid, 1037-1041
- distributed query blocks, controlling, 86
- materialized query tables (MQTs), 53
- parallelism, 52, 536, 859-867
  - disabling, 868
  - restrictions, 866
- performance factors, 57-58
- predicates
  - filter factor, 821-823
  - screening, 823
- safe optimization, 54
- subqueries, 20-23, 91
- tweaking, 87-88

Query Management Facility (QMF). *See* QMF (Query Management Facility)

query optimization options, MQTs (Materialized Query Tables), 1524-1526

query tool guide, 1435

query tools, 1412-1413

QUERYID column (DSN_USERQUERY_TABLE), 1134

querying

- business time data, 432-434
- PLAN_TABLE, 984-986
- XML data, 420-424

QUERYNO column (DSN_STATEMENT_TABLE), 999

QUERYNO column (DSN_USERQUERY_TABLE), 1133

QUERYNO column (PLAN_TABLE), 986

QUERY_TEXT column (DSN_USERQUERY_TABLE), 1133

queue manager (QM), 1486

QuickXScan, 423

QUIESCE JCL listing (33.6), 1220-1221

QUIESCE utility, 1165, 1220-1223

phases, 1222

R

RACF, 481

RADIANS function, 150

RAID storage devices, 1084

RAISE_ERROR function, 150, 162-163

RAND function, 82, 151

range predicates, dynamic SQL, 572

RCV tools, 1413-1414

RDA (Relational Data Services), 50

RDA (Remote Database Access), 1449

RDBMS (relational database management system), 4, 58

RDO (Resource Definition Online), 728

parameters, 732-739

RDS (Relational Data Services), 1068

reactive governing, RLF (Resource Limit Facility), 1144, 1150

reactive resource governing, 597

read engines, 49

read-only systems, RI (referential integrity), 296

reading data, 780

REAL data type, 244

REAL function, 151, 244

Real Time Statistics (RTS). *See* RTS (Real Time Statistics)

REASON column (DSN_STATEMENT_TABLE), 1000

rebalancing partitions, 367-368

REBIND command, 1014-1018

REBIND PACKAGE command, 1360

REBIND PLAN command, 1360

REBIND TRIGGER PACKAGE command, 1360

rebinding, 1014-1016

best practices, 1016-1018

gathering statistics, 1019-1020

packages, 625

periodic maintenance, 1015

regular maintenance, 1015

reviewing access paths, 1018-1019

system maintenance, 1015-1016

REBUILD INDEX JCL listing (33.10), 1233

REBUILD utility, 351, 1165

REBUILDLASTTIME column (SYSIBM.SYSINDEX-SPACESTATS), 1051

REBUILD_INDEX utility, 1232-1235

record identifier (RID), 50

record trace reports, 959

RECOVER INDEXSPACE JCL listing (33.9), 1228

RECOVER INDOUBT command, 1356

RECOVER utility, 1165, 1224-1226, 1229-1232

phases, 1228

recovering table spaces, 1226-1227

Recoverable Resource Manager Services Attach Facility (RRSAF). *See* RRSAF (Recoverable Resource Manager Services Attach Facility)

recovery

contingency planning, 1376-1377, 1390-1392

determining and managing risk, 1377-1379

DSN1COPY strategy, 1384-1385

environmental considerations, 1388-1390

FlashCopy strategy, 1385-1387

Scalpel strategy, 1381-1384

Sledgehammer strategy, 1380-1381

technologies, 1387-1388

coupling facility, 786
data sharing, 784
disaster, 1379-1380
disasters, classes, 1378
recovery log extractor utility, 1330
recovery management and assistance tools, 1413-1414
RECOVER_INDEX utility, 1228
RECOVER_TABLESPACE utility, 1225-1226
recursion, common table expressions (CTEs), 111-115
Redbook site, 698
reducing joins, 99
redundant data, denormalization, 284
REFERENCES privilege, 460
referential constraints, 291-294
referential integrity (RI), 290-292, 1136-1137
avoiding, 295-296
check constraints, 297-300
semantics, 299
implementation restrictions, 296
checking, CHECK_DATA utility, 1177
informational constraints, 296
programmatic, 295
referential constraints, 291-294
referential sets, 294-295
self-referencing constraints, 296
triggers, supplementing, 390-391
referential sets, 294-295
REFONLY option (SCOPE parameter), 1181
reformulating SQL, 71-73
refreshable options, MQTs (Materialized Query Tables), 1524
registers, special, 82-83
regular maintenance, rebinding, 1015
regulatory compliance, 478
Relational Data Services (RDS), 50, 1068
relational database management systems (RDBMSs), 4, 58
relational division, tables, 31-32
Relational Resource Adapters (RRAs), 693
relationships, DB2 Catalog, 883-885
releasing connections, 1462
RELOAD phase (LOAD utility), 1243
RELOAD phase (REORG TABLESPACE utility), 1274
REMARKS column (PLAN_TABLE), 988, 1006
REMARKS traces, 971
remote requests, 1460
DRDA, 1454-1455
remote unit of work (RUW), 1460
DRDA, 1454-1456
RENAME statement, 278
renaming
columns
arithmetic expressions, 80
views, 306-307
indexes, 365
tables, 278
REOPT
changing access paths, 88-89
dynamic SQL, 592-593
REOPT column (DSN_USER-QUERY_TABLE), 1133
reoptimization, run-time, 52
reordered format, rows, 254
Reordered Row Format (RRF), 799, 1081
REORG, 1014, 1016
REORG JCL (Nonrestartable) listing (34.5), 1267-1268
REORG JCL (Restartable) listing (34.4), 1266-1267
REORG parameter (DSN1COMP utility), 1321
REORG utility, 1166, 1265-1272, 1283-1288
gathering inline statistics, 1278-1279
job streams, 1272-1273
online reorganization, 1279-1283
phases, 1273-1275

reorganization frequency, 1268-1269
rerun/restart procedures, 1275-1278
RTS (Real Time Statistics), 1269-1272
SHRLEVEL parameter, 1279

reorganizing table spaces, 1018

REORGAPPENDINSERT column (SYSIBM.SYSINDEXSPACESTATS), 1051

REORGCLUSTERSENS column (SYSIBM.SYSTABLESPACESTATS), 1050

REORGDELETES column (SYSIBM.SYSINDEXSPACESTATS), 1051

REORGDELETES column (SYSIBM.SYSTABLESPACESTATS), 1049

REORGDISORGLOB column (SYSIBM.SYSTABLESPACESTATS), 1049

REORGFARINDREF column (SYSIBM.SYSTABLESPACESTATS), 1049

REORGHASHACCESS column (SYSIBM.SYSTABLESPACESTATS), 1050

REORGINDEXACCESS column (SYSIBM.SYSINDEXSPACESTATS), 1052

REORGINSERTS column (SYSIBM.SYSINDEXSPACESTATS), 1051

REORGINSERTS column (SYSIBM.SYSTABLESPACESTATS), 1049

REORGLASTTIME column (SYSIBM.SYSINDEXSPACESTATS), 1051

REORGLASTTIME column (SYSIBM.SYSTABLESPACESTATS), 1049

REORGLEAFFAR column (SYSIBM.SYSINDEXSPACESTATS), 1052

REORGLEAFNEAR column (SYSIBM.SYSINDEXSPACESTATS), 1051

REORGMASSDELETE column (SYSIBM.SYSINDEXSPACESTATS), 1051

REORGMASSDELETE column (SYSIBM.SYSTABLESPACESTATS), 1049

REORGNEARINDREF column (SYSIBM.SYSTABLESPACESTATS), 1049

REORGNUMLEVELS column (SYSIBM.SYSINDEXSPACESTATS), 1052

REORGPSEUDODELETES column (SYSIBM.SYSINDEXSPACESTATS), 1051

REORGSCANACCESS column (SYSIBM.SYSTABLESPACESTATS), 1050

REORGUNCLUSTINS column (SYSIBM.SYSTABLESPACESTATS), 1049

REORGUPDATES column (SYSIBM.SYSTABLESPACESTATS), 1049

REP tools, 1414-1415

REPAIR DBD JCL listing (32.6), 1192-1193

REPAIR LOCATE JCL listing (32.7), 1194

REPAIR phase (REPAIR utility), 1192

REPAIR SET JCL listing (32.8), 1196-1197

REPAIR utility, 1191, 1198, 1235

REPAIR_DBD utility, 1192-1193

REPAIR_LOCATE utility, 1193-1195

REPAIR_SET utility, 1196-1198

REPEAT function, 151

repeating groups, denormalization, 284-285

REPLACE function, 151

replication, 1498
populating data warehouses, 1514

replication tools, 1406

REPORT option (AUXERROR), 1181

REPORT phase (LOAD utility), 1244

REPORT RECOVERY JCL listing (33.11), 1236

report tables, denormalization, 282

REPORT TABLESPACESET utility, 427, 1199

REPORT TABLESPACESET JCL listing (32.9), 1199

REPORT utility, 1166, 1198-1200

REPORTCK phase (CHECK_DATA), 1183

reports, 968
batch, 928, 968
formatted, creating, 1047
online, 969-970
performance monitoring, 940-943
accounting, 943-956
audit, 956-957
explain, 957
I/O activity, 957-959
locking, 959
record trace, 959
SQL trace, 959-960
statistics, 960-966
strategies, 967-970
system parameters, 966-967
REPORT_RECOVERY utility, 1235-1236
REPORT_TABLESPACESET utility, 1199-1200
repositories, 1414-1415
requirements
data sharing, 774-775
disaster recovery, 1379
rerun procedures
LOAD utility, 1246-1250
REORG, 1275-1278
reserved words, SQL, 501-504
Resource Limit Facility (RLF), 1143, 1464
defining RLSTs, 1146-1147
limiting middleware resource usage, 1147-1149
predictive governing, 1144, 1150
reactive governing, 1144, 1150
resource limit specification tables (RLSTs), 1143, 1149
defining, 1146-1147
resource translation table (RTT), 756
resources, 1423-1426
availability, 769
blogs, 1428-1429
governing
limiting middleware resource usage, 1147-1149
Resource Limit Facility (RLF), 1143-1150
industry periodicals, 1427-1428
Internet, 696
mailing lists, 696-697, 1429
status, displaying, 977-979
webinars, 1429
response time, distributed, analyzing, 1493
responsibilities, 1430-1432
restart capabilities, IMS/TM, 759
restart procedures
LOAD utility, 1246-1250
REORG, 1275-1279
restartable programs, creating, 543-546
restarting UNLOAD utility, 1261-1262
RESTORE_SYSTEM utility, 1238-1239, 1387
result sets, stored procedures, returning, 664-666
Results of the DISPLAY GROUP Command listing (19.1), 786
retrieved data, SQL, grouping and sorting, 29-30
retrieving data, multiple tables, 91-110
reusability
stored procedures, 676
UDFs (user-defined functions), 184
revoking privileges, 449-450
REXX, application development, 563-565
REXX EXEC, program preparation, 618-619
RI (referential integrity), 290-292, 1136-1137
referential constraints, 291-294
referential sets, 294-295
RID (record identifiers), 50, 151, 808
pools, 1067
RIGHT function, 152
RIGHT OUTER JOIN, 28, 103
risks, determining and managing, 1377-1379

RLF (Resource Limit Facility), 1143, 1464
defining RLSTs, 1146-1147
limiting middleware resource usage, 1147-1149
predictive governing, 1144, 1150
reactive governing, 1144, 1150
RLFASUERR column (RLST), 1146
RLFASUWARN column (RLST), 1147
RLFBIND column (RLST), 1146
RLFCOLLN column (RLST), 1146
RLFFUNC column (RLST), 1146
RLFPKG column (RLST), 1146
RLF_CATEGORY_B column (RLST), 1147
RLSTs (resource limit specification tables), 1143, 1146-1149
ROLE traces, 972
roles, 471-473, 1430-1432
rotating partitions, 366-367
ROUND function, 152
ROUND TIMESTAMP function, 152
ROUTINE_ID column (PLAN_TABLE), 991
ROW CHANGE TIMESTAMP column, 269-270
row expressions, 107-108
row level granularity, LBAC (label-based access control), 462-464
ROWID columns, 403
ROWID data type, 39, 245
ROWID function, 152, 550
ROWLIMIT parameter (DSN1COMP utility), 1321
rows, 244-245
access control, 466-469
avoiding duplicate, 257
defaults, 261-262
direct access, 550, 831-832
locked, skipping, 909-911
locks, 899-901
BIND parameters, 893-895
minimum required, 58-59
multiple
retrieving, 535
updating, 126
reordered format, 254
ROWID data type, 245-246
sequence objects, 246-251
tables, counting, 137
rowset positioning cursors, 518-519
data modification, 519-520
inserting multiple rows, 520
RPAD function, 152
RRAs (Relational Resource Adapters), 693
RRF (Reordered Row Format), 799, 1081
RRSAF (Recoverable Resource Manager Services Attach Facility), 767-768
batch processing, 771
feasibility, 770-771
resource availability, 769
RTRIM function, 152
RTS (Real Time Statistics), 53, 1048, 1053-1058, 1371
DSNACCOX, 1054
examining, 1018
externalization, 1053-1054
REORG utility, 1269-1272
tables, 1048-1052
versus RUNSTATs, 1302
RTT (resource translation table), IMS, 756
RUN command, 1360
Run option (TSO), 720
run-time reoptimization, 52
running programs, 608
Running a DB2 Program in TSO Batch listing (15.1), 609
RUNSTATS INDEX JCL listing (34.5), 1298
RUNSTATS INDEX utility, 1166, 1298, 1301
RUNSTATS phase (RUNSTATS utility), 1298
RUNSTATS TABLESPACE JCL listing (35.3), 1296-1297

RUNSTATS TABLESPACE utility, 1166, 1296-1297, 1301

RUNSTATS utility, 426, 574, 1014-1019, 1056-1057, 1090, 1295-1310

data sets, 1298-1299

histogram statistics, 1301

phases, 1298

profiles, 1302-1303

updating Catalog tables, 1299-1300

versus RTS (Real Time Statistics), 1302

runtime environment, dynamic SQL, 574

RUW (remote unit of work), 1460

DRDA, 1454-1456

S

S-locks, 914

SAFEPOINT statements, 542-543

Sample COBOL Error-Handling Paragraph listing (11.1), 497-499

Sample Program Preparation Procedure listing (15.2), 616-618

sample programs, 1332

dynamic SQL processor, 1332-1334

dynamic SQL update program, 1334-1336

sample unload program, 1336-1339

Sample Results of DISPLAY BUFFERPOOL listing (37.2), 1346

Sample Results of DISPLAY LOG listing (37.3), 1347

sample unload program, 1336-1339

Sarbanes-Oxley Act, 478, 1398-1399

sargable processing, 50

scalar fullselect, 108-109

scalar functions, 34, 141

ABSVAL, 141

ACOS, 141

ADD MONTHS, 141

ASCII, 142

ASCII CHAR, 142

ASCII STR, 142

ASIN, 142

ATAN, 142

ATAN2, 142

ATANH, 142

BIGNIT, 142

BINARY, 142

BLOB, 142

CCSID ENCODING, 142

CEILING, 142

CHAR, 142

CHARACTER LENGTH, 142

CLOB, 143

COALESCE, 143

COLLATION KEY, 143

COMPARE DECFLOAT, 143

CONCAT, 143

CONTAINS, 143

COS, 143

COSH, 143

DATE, 143

DAY, 143

DAYOFMONTH, 143

DAYOFWEEK, 143

DAYOFWEEK ISO, 143

DAYOFYEAR, 144

DAYS, 144

DBCLOB, 144

DECFLOAT, 144

DECFLOAT SORTKEY, 144

DECIMAL, 144

DECRYPT BINARY, 144

DECRYPT BIT, 144

DECRYPT CHAR, 144

DECRYPT DB, 144

DEGREES, 144

DIFFERENCE, 144

DIGITS, 144

DOUBLE, 144

DSN XMLVALIDATE, 145

EBCDIC CHR, 145

EBCDIC STR, 145

ENCRYPT STR, 145

EXP, 145

EXTRACT, 145

FLOAT, 144-145

FLOOR, 145

GENERATE UNIQUE, 145

GETHINT, 145

GETVARIABLE, 145

GRAPHIC, 145
GREATEST, 146
HEX, 146
HOUR, 146
IDENTITY VAL LOCAL(), 146
INFULL, 146
INSERT, 146
INTEGER, 147
JULIAN DAY, 147
LAST DAY, 147
LEAST, 147
LEFT, 147
LENGTH, 147
LOCATE, 147-148
LOCATE IN STRING, 148
LOG10, 148
LOWER, 148
LPAD, 148
LTRIM, 148
MAX, 148
MICROSECOND, 148
MIDNIGHT SECONDS, 148
MIN, 148
MINUTE, 149
MOD, 149
MONTH, 149
MONTHS BETWEEN, 149
MQSeries, 159-161
MULTIPLY ALT, 149
NEXT DAY, 149
NORMALIZE DECFLOAT, 149
NORMALIZE STRING, 149
NULLIF, 149
OVERLAY, 149
POSITION, 150
POSSTR, 150
POWER, 150
QUANTIZE, 150
QUARTER, 150
RADIANS, 150
RAISE ERROR, 150
RAND, 151
REAL, 151
REPEAT, 151
REPLACE, 151
RID, 151
RIGHT, 152
ROUND, 152
ROUND TIMESTAMP, 152
ROWID, 152
RPAD, 152
RTRIM, 152
SCORE, 152
SECOND, 152
SIGN, 153
SIN, 153
SINH, 153
SMALLINT, 153
SOAPHTTPC, 153
SOAPHTTPV, 153
SOUNDEX, 153
SPACE, 153
SQRT, 153
STRIP, 153
SUBSTR, 154
SUBSTRING, 154
TAN, 154
TANH, 154
TIME, 154
TIMESTAMP, 154
TIMESTAMP FORMAT, 155
TIMESTAMP ISO, 155
TIMESTAMP TZ, 155
TIMESTAMPADD, 154
TIMESTAMPDIFF, 154
TOTALORDER, 155
TRANSLATE, 155-156
TRUNC TIMESTAMP, 156-157
TRUNCATE, 156
trusted context, 158
UNICODE, 157
UNICODE STR, 157
UPPER, 157
VALUE, 157
VARBINARY, 157
VARCHAR, 157
VARCHAR FORMAT, 158
VARGRAPHIC, 158
versus CAST operation, 163
WEEK, 158
WEEK ISO, 158
WHERE clauses, 86
XML, 161-162
YEAR, 158

scans, table spaces, 825-831

SCANTAB phase (CHECK_DATA), 1183

Scalpel strategy, backup and recovery, 1381-1384
SCHEMA column (DSN_USERQUERY_TABLE), 1133
SCHEMA privilege class, 450
SCHEMA security class, 40
schema validation, XML, 416
schemas, tables, defining, 276
SCOPE parameter, CHECK_DATA utility, 1180, 1184
SCORE function, 152
scratchpads, UDFs (user-defined functions), 189
screen input, validating, 724
screening
  index, 837-838
  page ranges, 827-828
  predicates, 823
scrollable cursors, 515-518
  ASENSITIVE, 516
  fetching data, 515
  INSENSITIVE, 515-516
  moving within result set, 535
  SENSITIVE, 515-517
SCT02 structure (Directory), 887
SDR (SQL Design Review), 1439
search engines, 702
SEC tools, 1416-1417
SECADM authority, 455
SECADM authorization, 450-451
SECADM group-level authorization, 41
SECOND function, 152
secondary authids, 449
SECQTY, 222-226, 350
SECTNOI column (DSN_STATEMENT_TABLE), 1001
SECTNOI column (PLAN_TABLE), 991
security, 482-484
  access control
    columns, 466-469
    rows, 466-469
  authorization, 448, 453-461
    IDs, 448-449
    PUBLIC access, 452-453
    repeating, 453
    SECADM, 450-451
  data definition control, 471
  database, 476-481
  distributed, 1501-1502
  encryption, 473-476
  external, 481
  Kerberos, 482-483
  label-based access control (LBAC), 461-466
    referential integrity, 465
    restrictions, 465-466
    row level granularity, 462-464
  MLS (multilevel security), 461-466
  privileges, 448
    granting and revoking, 449-450
    group, 451-452
  roles, 471-473
  session variables, 469-471
  special registers, 483-484
  trusted context, 471-473
  views, 303
Security area (Catalog), 881
security controls, data structures, 40-42
security guide, 1434-1435
SECURITY parameter, UDFs (user-defined functions), 188-189
security tools, 1416-1417
segmented table spaces, 205-206
SEGSIZE parameter, table space, 231
SELCT authority, 453
SELECT command
  avoiding, 59
  minimum number of rows and columns required, 58-59
SELECT statement, 14-15, 130-132, 1527, 1532
  cursors, 521-525
  OPTIMIZE FOR n ROWS clause, 1123-1124
SELECT statements, 213, 510
  aggregate functions, 135
  joining tables, 15-20
  singleton, 521-522
  subqueries, 20-23

selection operation, SQL, 13-14

self-referencing constraints, 296

semantics, 299

SENSITIVE scrollable cursors, 515-517

SEQCOPY phase (COPY), 1205

sequence objects, 246-251, 549

SEQUENCE objects, parameters, 249-250

SEQUENCE privilege class, 450

SEQUENCE security class, 40

sequencing columns, 253-254

sequential detection, 49

sequential prefetch performance feature, 49, 825, 828-831

sequential steal parameter (buffer pool), 1102

serializing data, XML, 415

server location aliases, 1472

service level agreements (SLAs), 929

service level management (SLM), 929

Service Request Blocks (SRBs), 814

servlets, Java, 555-557

session variables, 469-471

SET CURRENT PACKAGE PATH, UDFs (user-defined functions), 184

set operations, 23, 91

- EXCEPT, 26
- INTERSECT, 26
- UDTs (user-defined data types), 198-199
- union, 23-26

set-at-a-time processing, SQL, 7-8, 10-11

sets, referential, 294-295

shared objects, 776

sharing cached statements, 591-592

SHRLEVEL parameter, REORG, 1279

SIGN function, 153

simulating host variables, 533-534

SIN function, 153

single table access path strategies, 824-845

single-table table spaces, 235

singleton SELECT statements, 65-66, 521-522

SINH function, 153

SJTABLES column (DSN_USERQUERY_TABLE), 1133

SKCTs (skeleton cursor tables), 202, 887

skip sequential prefetch, 841

skipping

- lock rows, 909-911
- locked data, 133-134

SLAs (service level agreements), 929

Sledgehammer strategy, backup and recovery, 1380-1381

sliding scale extents, 226

SLM (service level management), 929

SMALLINT data type, 38, 153, 244

Smart Analytics System (IBM), 1522

snapshots, 1498-1499

- populating data warehouses, 1514

SOAPHTTPC function, 153

SOAPHTTPV function, 153

sort I/O, 1081-1082

SORT phase

- CHECK_DATA, 1183
- LOAD utility, 1243
- REBUILD_INDEX utility, 1234
- REORG TABLESPACE utility, 1274

sort pools, 1067-1069

SORTBLD phase (LOAD utility), 1244

SORTBLD phase (REBUILD_INDEX utility), 1234

SORTC_GROUP column (PLAN_TABLE), 988

SORTC_JOIN column (PLAN_TABLE), 988

SORTC_ORDERBY column (PLAN_TABLE), 988

SORTC_PGROUP_ID column (PLAN_TABLE), 989

SORTC_UNIQ column (PLAN_TABLE), 988

sorting

LOAD utility, 1251-1252

retrieved data, 29-30

SORTN_GROUPBY column (PLAN_TABLE), 987

SORTN_JOIN column (PLAN_TABLE), 987

SORTN_ORDERBY column (PLAN_TABLE), 987

SORTN_PGROUP_ID column (PLAN_TABLE), 989

SORTN_UNIQ column (PLAN_TABLE), 987

SOUNDEX function, 153

source program preparation objects, 631

sourced UDFs, 168, 178

SPACE column (SYSIBM.SYSINDEX-SPACESTATS), 1051

SPACE column (SYSIBM.SYS-TABLESPACESTATS), 1050

SPACE function, 153

space management tools, 1404

spam, 701

sparse index access, 844-845

special registers, 82-83, 483-484

CURRENT PACKAGE PATH, 629

specialty processors, 812-815

split tables, denormalization, 282-283

SPSS (Statistical Package for the Social Science), 1522

SPT01 structure (Directory), 887

SPUFI command, 1360

TSO (Time-Sharing Options), 712-722

SQL (Structured Query Language), 4-5, 13

access guidelines, 58, 82-90

ad hoc, 12

application development, 565-566

CASE expressions, 32-34

code appropriate existence checking, 96-98

coding, 552, 554

data processing, 11

data structures, 37-38

security controls, 40-42

DCL (Data Control Language), 11

DDL (Data Definition Language), 11

DELETE statement, 37

DML (Data Manipulation Language), 11

dynamic, 44-45, 567-569, 574, 594, 597-600

bind-time authorization checking, 596

caching prepared statements, 596

classes, 576-589, 594

data uniformity, 571-572

host variables, 574-575

KEEPDYNAMIC parameter, 591

making more static, 589-593

monitoring, 575-576

parallelism, 596

parameter markers, 595-596

performance sensitivity, 571

program examples, 576

range predicates, 572

REOPT parameter, 592-593

repetitions execution, 573

RUNSTATS, 574

runtime environment, 574

statements, 594

tuning, 575-576

versus static, 569-576

embedded statements, 487-490

delimiting, 490

GET DIAGNOSTICS, 494, 497

pretesting, 61

SQLCA (SQL Communication Area), 491-493

SQLCODE, 493

SQLSTATE, 493

WHENEVER, 500-501

existential, 12
flexibility, 5-6
functions, 34
HAVING clause, 30-31
INSERT statement, 35-36
joins, 15-23
  Cartesian products, 19-20
  outer, 26-29
long accounting reports, activity, 949-950
modifying data, 35-37
nature of, 7
non-relational data
  accessing, 274
  joining, 534-535
performance factors, 45-55
procedural, 678
  benefits, 682
  drawbacks, 682-683
  IBM, 678-680
procedural SQL, 6
projection operation, 13-14
queries
  analysis tools, 64-65
  parallelism, 52
  subqueries, 20-23
  tweaking, 87-88
reformulating, 71-73
reserved words, 501-503
  ANSI, 504
  IBM, 503-504
retrieved data, grouping and sorting, 29-30
scalar functions, 178-179
SELECT statement, 14-15
selection operation, 13-14
set-at-a-time processing, 7-11
set operations, 23-26
Stage 1, achieving, 71-72
statement types, 11
statements, 90-91
  coding simply, 60
  isolation level, 535-536
  performance potential, 61
static, 42, 44, 567
  making more dynamic, 593-594
  packages, 1465
stored procedures, 672-674, 680-681
tables
  joining, 19-20, 531-533
  relational division, 31-32
Top Ten problem, 93-95
types, 11-12
UDFs (user-defined functions), 186-187
UPDATE statement, 36
versions, 11
WHERE clause, 30-31

SQL DBMSs, 4

SQL Design Review (SDR), 1439

SQL performance guide, 1435

SQL scalar UDFs, 168

SQL statements, 821
  access paths, 980, 993-1002
  block style, 552-554
  embedded, data modification, 525-527
  host variables, 504-506, 509-511
    host structures, 506
    null indicators, 507-509
  issuing, 1173-1175
  tweaking, 1121-1123

SQL table functions, 179-180

SQL table UDFs, 168

SQL trace reports, 959-960

SQLADM group-level authorization, 41

SQLADM privilege, 452

SQLCA (SQL Communication Area), 491-493

SQLCODE statement, 493

SQLDA, 582-584

SQLDA listing (14.5), 582-583

SQLJ versus Java Database Connectivity (JDBC), 557-559

SQLJ Code Fragment listing (11.7), 558

SQLSTATE statement, 493

SQRT function, 153

SRBs (Service Request Blocks), 814

SRM (Systems Resource Manager), 1084

SSAS (system services), 809

SSM parameters, 753

Stage 1

achieving, 71-72

predicates, 69-71

stage processing, 50

stand-alone utilities, 1314-1315

DSN1CHKR, 1318-1319

DSN1COMP, 1320-1322

DSN1COPY, 1322-1328

DSN1LOGP, 1330

DSN1PRNT, 1330-1332

DSN1SDMP, 1328-1329

DSNJLOGF, 1315-1316

DSNJU003, 1316-1317

DSNJU004, 1317-1318

standardized abbreviations, 252

standardizing error handling, 497-499

standards, 1429

application development guide, 1434

data administration, 1432-1433

database administration guide, 1433

design reviews, 1436-1440

DRDA, 1451-1452

naming conventions, 1435-1436

query tool guide, 1435

responsibilities, 1430-1432

roles, 1430-1432

security guide, 1434-1435

SQL performance guide, 1435

system administration guide, 1433-1434

vendor tool guides, 1435

star joins, 854-858

star schema, data warehouses, 1511-1513

***Star Schema: The Complete Reference*, 1513**

STARJOIN column (DSN_USERQUERY_TABLE), 1133

START DATABASE command, 1356

START RLIMIT command, 1357

START TRACE command, 1357

starting UDFs, 182-183

statement types, SQL, 11

statements

ALTER TABLE, 1532

cached, sharing, 591-592

COMMENT ON, 277

COMMIT, 539-541, 548, 788

CREATE, 1524

CREATE FUNCTION, 606

CREATE INDEX, 330

CREATE TRIGGER, 606

DDL, 322

DECLARE CURSOR, 523

DELETE, 37, 127, 548, 577

DSC (dynamic statement cache), 1495

dynamic, caching, 590-591

dynamic SQL, 594

embedded SQL, 487-490

data modification, 525-527

delimiting, 490

GET DIAGNOSTICS, 494, 497

SQLCA (SQL Communication Area), 491

SQLCODE, 493

SQLSTATE, 493

WHENEVER, 500-501

FETCH, 109-110

INCLUDE, 602

INSERT, 35-36, 130-133, 526

LOCK STABLE, 919

MERGE, 129-132

NOT EXISTS, 91

NOT IN, 91

performance potential, checking, 61

physical data independence, 817-818

RENAME, 278

RUNSTATS, 1019

SAFEPOINT, 542-543

SELECT, 14-15, 130-132, 135, 213, 1527, 1532
cursors, 521-525
singleton, 521-522
SQL, 90-91, 821
block style, 552-554
coding simply, 60
host variables, 504-511
isolation level, 535-536
issuing, 1173-1175
tweaking, 1121-1123
static SQL statements, 42
TEMPLATE, 1167-1171
TIMESTAMP, 549
TRUNCATE, 379, 546-547
UPDATE, 36, 127, 131
VALUES, triggers, 391

static SENSITIVE scrollable cursors, 516-517

static SQL, 42-44, 567
making more dynamic, 593-594
packages, 1465
versus dynamic SQL, 569-576

statistics
Catalog, 819-821
changing, 1124-1130
gathering, 1019-1020
LOAD utility, 1245
REORG, 1278-1279
histogram, 1301
historical, 53
non-uniform distribution statistics, deleting, 1135-1136
optimizer, 816
real-time, 53
RTS (Real Time Statistics), 1048, 1054-1058
DSNACCOX, 1054
externalization, 1053-1054
tables, 1048-1052

statistics reports, 960-966
buffer pool general information, 961
buffer pool read information, 962
buffer pool write information, 962
common storage usage, 965
EDM pool activity information, 965-966
locking activity information, 964
log activity information, 963-964

statistics trace, 935-936

STATSDELETES column (SYSIBM.SYSINDEXSPACESTATS), 1052

STATSDELETES column (SYSIBM.SYSTABLESPACESTATS), 1049

STATSINSERTS column (SYSIBM.SYSINDEXSPACESTATS), 1052

STATSINSERTS column (SYSIBM.SYSTABLESPACESTATS), 1049

STATSLASTTIME column (SYSIBM.SYSINDEXSPACESTATS), 1052

STATSLASTTIME column (SYSIBM.SYSTABLESPACESTATS), 1049

STATSMASSDELETE column (SYSIBM.SYSINDEXSPACESTATS), 1052

STATSMASSDELETE column (SYSIBM.SYSTABLESPACESTATS), 1050

STATSUPDATES column (SYSIBM.SYSTABLESPACESTATS), 1050

STDDEV function, 140

STMTTOKEN column (PLAN_TABLE), 991

STMT_ENCODE column (DSN_STATEMENT_TABLE), 1001

STMT_ID column (DSN_STATEMENT_CACHE_TABLE), 1005

STMT_TEXT column (DSN_STATEMENT_CACHE_TABLE), 1005

STMT_TOKEN column (DSN_STATEMENT_CACHE_TABLE), 1005

STMT_TYPE column (DSN_STATEMENT_TABLE), 1000

STOGROUPs, 38, 239
classes, 244
Data Facility Storage Management System (DFSMS), 239-242
disk volumes, 244
high-level qualifiers, 243
SYSDEFLT, 242
user-defined VSAM, 242-243
stopping UDFs, 182-183
storage groups, 201, 239
classes, 244
Data Facility Storage Management System (DFSMS), 239-242
defining useful, 239
disk volumes, 244
high-level qualifiers, 243
SYSDEFLT, 242
user-defined VSAM, 242-243
stored procedures, 656-657, 674-677, 1496
application development, 530
atomic parameters, 675
benefits, 659-661
calling, 672
coding parameters, 662-663
controlling failures, 669-670
creating, 666-668
developing, 661
executing, 670-672
external, 676-677
IBM Data Studio, 687
implementation, 657-659
implementing, 661-666
LE/370, 675
managing, 668-670
native SQL, 672-674
nesting, 663-664, 674
nesting calls, 664
OLTP (Online Transaction Processing), 670
plan, 675
procedural DBA, 683-687
procedural SQL, 678-683
program preparation, 666
returning result sets, 664-666
reusability, 676
SQL, 680-681
subprograms, 674
supported languages, 661
temporary tables, 676
versioning, 673-674
versus triggers, 374
WLM (Work Load Manager), 670-672
stored temporary tables, 274
storing data relationally, 411
STOSPACE JCL listing (35.5), 1311
STOSPACE phase (STOSPACE utility), 1311
STOSPACE utility, 1311-1313
string units, 166
STRIP function, 153
strong types, 191-192
Structured Query Language (SQL). *See* SQL (Structured Query Language)
structures
host, 533
locks, 913
pages, 797-798
sub-SELECT, 129
subprograms, stored procedures, 674
subqueries, 91
optimization, 869-871
SQL, 20-23
versus joins, 100-101
versus subselects, 20
subselects versus subqueries, 20
SUBSTR function, 154
SUBSTRING function, 154
subsystem availability, data sharing, 785
subsystem IDs, specifying, 741
subsystems, tuning, 1089-1114
buffer pools, 1096-1114
Catalog, 1089-1092
DSNZPARMs, 1092-1096
SUM function, 140
surrogate keys, tables, 258
suspensions, locks, 901-904

SWITCH phase (REORG INDEX utility), 1273

SWITCH phase (REORG TABLESPACE utility), 1274

synchronization rule, views, 311

SYNONYM database objects, 38

synonyms, 201, 313

- authorization, 458
- columns, 257
- functions, 164-165

SYSADM authority, 455, 459-460

SYSADM group-level authorization, 41

SYSAUDITPOLICIES (V10) table (Catalog), 875

SYSAUTOALERTS (V10) table (Catalog), 875

SYSAUTOALERTS_OUT (V10) table (Catalog), 875

SYSAUTORUNS_HIST (V10) table (Catalog), 875

SYSAUTORUNS_HISTOU (V10) table (Catalog), 875

SYSAUTOTIMEWINDOWS (V10) table (Catalog), 875

SYSAUXRELS table (Catalog), 875

SYSCHECKDEP table (Catalog), 875

SYSCHECKS table (Catalog), 875

SYSCHECKS2 table (Catalog), 875

SYSCOLAUTH table (Catalog), 875

SYSCOLDIST table (Catalog), 875

SYSCOLDISTSTATS table (Catalog), 875

SYSCOLDIST_HIST table (Catalog), 875

SYSCOLSTATS table (Catalog), 875

SYSCOLUMNS table (Catalog), 875

SYSCOLUMNS_HIST table (Catalog), 875

SYSCONSTDEP table (Catalog), 875

SYSCONTEXT (V9) table (Catalog), 875

SYSCONTEXTAUTHIDS (V9) table (Catalog), 875

SYSCONTROLS (V10) table (Catalog), 875

SYSCOPY table (Catalog), 876

SYSCTRL group-level authorization, 41

SYSCTXTTRUSTATTRS (V9) table (Catalog), 876

SYSDATABASE table (Catalog), 876

SYSDATATYPES table (Catalog), 876

SYSDBAUTH table (Catalog), 876

SYSDBRM table (Catalog), 876

SYSDEFLT, 242

SYSDEPENDENCIES (V9) table (Catalog), 876

SYSDUMMY1 table (Catalog), 876

SYSDUMMYA table (Catalog), 876

SYSDUMMYE table (Catalog), 876

SYSDUMMYU table (Catalog), 876

SYSENVIRONMENT (V9) table (Catalog), 876

SYSFIELDS table (Catalog), 876

SYSFOREIGNKEYS table (Catalog), 876

SYSGEN (IMS), 760

SYSIBM.SYSINDEXSPACE-STATS columns, 1051-1052

SYSIBM.SYSTABLESPACE-STATS columns, 1049

SYSINDEXES table (Catalog), 876

SYSINDEXES_HIST table (Catalog), 876

SYSINDEXES_RTSECT (V10) table (Catalog), 876

SYSINDEXES_TREE (V10) table (Catalog), 876

SYSINDEXPART table (Catalog), 876

SYSINDEXPART_HIST table (Catalog), 876

SYSINDEXSPACESTATS (V9) table (Catalog), 876

SYSINDEXSTATS table (Catalog), 876

SYSINDEXSTATS_HIST table (Catalog), 876

SYSJARCLASS_SOURCE table (Catalog), 876
SYSJARCONTENTS table (Catalog), 876
SYSJARDATA table (Catalog), 876
SYSJAROBJECTS table (Catalog), 876
SYSJAVAOPTS table (Catalog), 876
SYSJAVAPATHS (V9) table (Catalog), 876
SYSKEYCOLUSE table (Catalog), 877
SYSKEYS table (Catalog), 877
SYSKEYTARGETS (V9) table (Catalog), 877
SYSKEYTARGETSTATS (V9) table (Catalog), 877
SYSKEYTARGETS_HIST (V9) table (Catalog), 877
SYSKEYTGTDIST (V9) table (Catalog), 877
SYSKEYTGTDISTSTATS (V9) table (Catalog), 877
SYSKEYTGTDISTSTATS_HIST (V9) table (Catalog), 877
SYSLGRNX structure (Directory), 888
SYSLOBSTATS table (Catalog), 877
SYSLOBSTATS_HIST table (Catalog), 877
SYSOBDS table (Catalog), 877
SYSOBJROLEDEP (V9) table (Catalog), 877
SYSOPR group-level authorization, 41
SYSPACKAGE table (Catalog), 877
SYSPACKAUTH table (Catalog), 877
SYSPACKCOPY (V10) table (Catalog), 877
SYSPACKDEP table (Catalog), 877
SYSPACKLIST table (Catalog), 877
SYSPACKSTMT table (Catalog), 877
SYSPARMS table (Catalog), 877
SYSPENDINGDDL (V10) table (Catalog), 877
SYSPENDINGOBJECTS (V10) table (Catalog), 877
SYSPKSYSTEM table (Catalog), 877
SYSPLAN table (Catalog), 877
SYSPLANAUTH table (Catalog), 877
SYSPLANDEP table (Catalog), 877
Sysplex, 773-774
  parallelism, 788
SYSPLSYSTEM table (Catalog), 877
SYSQUERY (V10) table (Catalog), 877
SYSQUERYOPTS (V10) table (Catalog), 877
SYSQUERYPLAN (V10) table (Catalog), 877
SYSQUERY_AUX (V10) table (Catalog), 877
SYSRELS table (Catalog), 877
SYSRESAUTH table (Catalog), 878
SYSROLES (V9) table (Catalog), 878
SYSROUTINEAUTH table (Catalog), 878
SYSROUTINES table (Catalog), 878
SYSROUTINESTEXT (V9) table (Catalog), 878
SYSROUTINES_OPTS table (Catalog), 878
SYSROUTINES_PTREE (V10) table (Catalog), 878
SYSROUTINES_SRC table (Catalog), 878
SYSSCHEMAAUTH table (Catalog), 878
SYSSEQUENCEAUTH table (Catalog), 878
SYSSEQUENCES table (Catalog), 878
SYSSEQUENCESDEP table (Catalog), 878
SYSSTMT table (Catalog), 878
SYSSTOGROUP table (Catalog), 878
SYSSTRINGS table (Catalog), 878
SYSSYNONYMS table (Catalog), 878
SYSTABAUTH table (Catalog), 878

SYSTABCONST table (Catalog), 878

SYSTABLEPART table (Catalog), 878

SYSTABLEPART_HIST table (Catalog), 878

SYSTABLES table (Catalog), 878

SYSTABLESPACE table (Catalog), 878

SYSTABLESPACESTATS (V9) table (Catalog), 878

SYSTABLES_HIST table (Catalog), 878

SYSTABLES_PROFILES (V10) table (Catalog), 878

SYSTABLES_PROFILES_TEXT (V10) table (Catalog), 878

SYSTABSTATS table (Catalog), 878

SYSTABSTATS_HIST table (Catalog), 878

system administration guide, 1433-1434

system DBADM authority, 451-452

system maintenance, rebinding, 1015-1016

SYSTEM parameter (DSNUPROC), 1156

system parameters reports, 966-967

SYSTEM privilege class, 450

SYSTEM security class, 40

system services (SSAS), 809

system time, 430

SYSTEM TIME tables, implementing, 437-438

system-directed data access, 1463

system-level changes, 57

Systems Resource Manager (SRM), 1084

SYSTRIGGERS table (Catalog), 878

SYSTRIGGERS_STMT (V10) table (Catalog), 878

SYSUSERAUTH table (Catalog), 878

SYSUTILX structure (Directory), 887

SYSVIEWDEP table (Catalog), 878

SYSVIEWS table (Catalog), 878

SYSVIEWS_STMT table (Catalog), 879

SYSVIEWS_TREE table (Catalog), 879

SYSVOLUMES table (Catalog), 879

SYSXMLRELS (V9) table (Catalog), 879

SYSXMLSTRINGS (V9) table (Catalog), 879

SYSXMLTYPMOD (V9) table (Catalog), 879

SYSXMLTYPMSCHEMA (V9) table (Catalog), 879

T

table altering tools, 1396-1398

TABLE database objects, 38

table editors, 1405-1406

table expressions, 105-106

- improving performance, 106-107
- versus views, 107

table functions, 159

TABLE privilege class, 450

table range columns, code predicates, 68

TABLE security class, 40

table space type, online schema

table spaces, 200, 204-205

- copying, 1203-1204
- data compression, 231-233
- data pages, 799-801
- definitions, 237
- DSN1COMP utility, parameters, 1321
- implicit, avoiding, 278
- LOBs (large objects), 215-216
  - locks, 916
- locks, 895-897
  - BIND parameters, 892-893
- multi-table, 235-237

parameters, 217-235
BUFFERPOOL, 229
CCSID, 233
CLOSE, 229-231
DSSIZE, 217-218
ERASE, 231
FREEPAGE, 227-229
LOCKMAX, 220
LOCKPART, 220
LOCKSIZE, 218-220
MAXROWS, 233
MEMBER CLUSTER, 234
NUMPARTS, 231
PCTFREE, 227-229
PRIQTY, 222-226
SECQTY, 222-226
SEGSIZE, 231
TRACKMOD, 234
USING, 221-222
partitioned, 207-208, 238
advantages, 213-214
changing, 210
classic, 216
data sets, 210
disadvantages, 214-215
key ranges, 89-90
large, 208-209
online schema, 365-368
scans, 826
SMS, 241-242
versus multiple tables, 211-213
recovering, RECOVER utility, 1226-1227
reorganizing, 1018
RTS (Real Time Statistics), 1270-1272
scans, 825-831
segmented, 205-206
single, 238
single-table, 235
specifying type, 216-217
universal, 206-207, 237-238
XML, 215

tables, 244
aliases, 313
auditing, forcing, 267
auxiliary, indexing, 351
bitemporal, implementing, 443-445
BUSINESS TIME, 431
Catalog, remote management access tables, 1459
CDB (Communication Database), 879
CICS, 727-728
clone, 274-275, 301-302
columns, 244-245
data types, 244-246
DATE and TIME, 259-260
defaults, 261-262
defining, 252, 257
homonyms, 257
identity, 246-251
IMPLICITLY HIDDEN, 265-266
labels, 277-278
LOB, 396-402
naming, 252
nullable, 254-257
sequence objects, 246-251
sequencing, 253-254
synonyms, 257
TIMESTAMP, 259-260
VARCHAR, 263
variable, 262-263
XML, 412-414
combined, denormalization, 283
comments, 277
creating
EXPLAIN command, 62-63
data types, 260-261
DB2 Catalog, 874-876, 878-879
DECIMAL, 264
dummy, 81-82
emptying, 546-547
EXPLAIN, 1002
DSN_COLDIST_TABLE, 62, 1002
DSN_DETCOST_TABLE, 1003
DSN_FILTER_TABLE, 1003
DSN_FUNCTION_TABLE, 63, 1001-1002

DSN_KEYTGTDIST_TABLE, 63, 1002
DSN_PGRANGE_TABLE, 63, 1003
DSN_PGROUP_TABLE, 63, 1003
DSN_PREDICAT_TABLE, 63, 1002
DSN_PTASK_TABLE, 63, 1003
DSN_QUERY_TABLE, 63, 1003
DSN_SORTKEY_TABLE, 63, 1003
DSN_SORT_TABLE, 63, 1003
DSN_STATEMENT_CACHE_TABLE, 63, 1002-1005
DSN_STATEMENT_TABLE, 998-1001
DSN_STRUCT_TABLE, 63, 1003
DSN_VIEWREF_TABLE, 63, 1003
PLAN_TABLE, 982-993, 1006
explicitly declaring, 490
expressions, 91
fragmentation, 1497-1498
hash spaces, 338-339
hash-organized, creating, 339-341, 347
in-memory table caches, 858-859
indexing, 344
INTEGER, 264-265
ISPF, 724-725
joining, 15-20, 51, 531-533
loading, 1240-1243
locks, 897-898
Materialized Query Tables (MQTs), 53, 1522-1523, 1527, 1532-1533
attribute copy options, 1526
automatic query rewrite, 1528-1532
benefits, 1523
converting into, 1527-1528
creating, 1523-1524
population and maintenance, 1528
query optimization options, 1524-1526
refreshable options, 1524
mirror, denormalization, 282
multiple, retrieving data from, 91-110
parameters, 266-270
partitions
adding to, 365-366
rotating, 366-367
PLAN_TABLE, 62
pre-joined, denormalization, 281
primary keys, 258
renaming, 278
report, denormalization, 282
RLSTs (resource limit specification tables), 1143, 1147-1149
defining, 1146-1147
rows, 244-245
counting, 137
defaults, 261-262
direct access, 831-832
duplicating, 257
reordered format, 254
ROWID data type, 245-246
sequence objects, 246-251
RTS (Real Time Statistics), 1048-1052
RTT (resource translation table), 756
scans, 825-831
schema, defining and duplicating, 275-276
skeleton package tables (SKPTs), 202
split, denormalization, 282-283
SQL, relational division, 31-32
surrogate keys, 258
synonyms, 313
SYSTEM TIME, implementing, 437-438
temporal, time-based transaction data, 266
temporary, 270, 274
accessing non-relational data, 274
benefits, 270

creating, 271
declared, 271-273
stored, 274
stored procedures, 676
updating, RUNSTATS utility, 1299-1300
views, 302-304, 307-313
data derivation, 303
mimicing domain support, 304-306
non-updatable, 312
renaming columns, 306-307
restrictions, 312
security, 303
specifying column names, 312
synchronization rule, 311
usage rule, 303
XML data, altering to contain, 416
TABLESPACE database objects, 38
TABLE_DCCSID column (PLAN_TABLE), 991
TABLE_ENCODE column (PLAN_TABLE), 991
TABLE_MCCSID column (PLAN_TABLE), 991
TABLE_SCCSID column (PLAN_TABLE), 991
TABLE_TYPE column (PLAN_TABLE), 991
TABNO column (PLAN_TABLE), 987
TAN function, 154
TANH function, 154
Task Control Blocks (TCBs), 814
tasks, procedural DBA, 684-687
TCBs (Task Control Blocks), 814
TCP/IP, native support, 1472
teleprocessing, tuning, 1087-1088
TEMPLATE control statement, 1370
TEMPLATE statement, 1167-1171
TEMPLATE utility, 1159-1160
templates, UDFs (user-defined functions), 185-186
temporal data, 446-447
business time, 430
support, 430-446
system time, 430
temporal tables, time-based transaction data, 266
temporary tables, 270, 274
benefits, 270
creating, 271
declared, 271-273
non-relational data, accessing, 274
stored, 274
stored procedures, 676
Terminal Monitor Program (TMP), 725
TERMINATE UTILITY command, 1358
termination, UNLOAD utility, 1261
testing trigger logic, 389
third-generation languages (3GLs), 486
third-party tools
auditing, 1399-1400
Catalog and analysis, 1400-1401
client/server, 1402
compression, 1401-1402
data movement, 1406
database analysis tools, 1402-1403
database archiving tools, 1398
database modeling and design tools, 1403-1404
disk and space management, 1404
extract/transformation/load tools, 1406
fourth-generation languages, 1419-1420
integrity tools, 1406
Internet enabling, 1408
miscellaneous, 1407
object migration tools, 1407
operational support, 1408
PC-based emulation, 1408
performance, 1411
performance monitors, 1410-1411

plan and package analysis, 1409-1410
programming and development, 1411-1412
QMF, 1412
query, 1412-1413
recovery management and assistance, 1413-1414
repositories, 1414-1415
security, 1416-1417
table altering tools, 1396-1398
table editors, 1405-1406
utility enhancement, 1417, 1419
vendors, 699, 1420-1422

thread operation attributes, DB2ENTRY parameter (RDO), 737-739

thread selection attributes, DB2ENTRY parameter (RDO), 737

threads, 705
CAF (Call Attach Facility), 764-766
CICS, 732
DBATs (database access threads), inactive, 1469
IMS/TM, 756-757
specifying, transactions, 743-744

three-part name support, DRDA, 1463-1464

TIME data type, 39, 119-120, 245

TIME function, 154

time zones, non-standard dates, 121-122

time-based transaction data, temporal tables, 266

Time-Sharing Option (TSO). *See* TSO (Time-Sharing Option)

timeouts, locks, 901-904

TIMESTAMP, 119-120

TIMESTAMP columns, 259-260, 988

TIMESTAMP data type, 39, 245

TIMESTAMP FORMAT function, 155

TIMESTAMP function, 154

TIMESTAMP ISO function, 155

TIMESTAMP statement, 549

TIMESTAMP TZ function, 155

TIMESTAMPADD function, 154

TIMESTAMPDIFF function, 154

TMP (Terminal Monitor Program), 725

TNAME column (PLAN_TABLE), 986

tools, 1394, 1396
auditing, 1399-1400
Catalog and analysis query, 1400-1401
client/server, 1402
compression, 1401-1402
Data Studio, 1396
database analysis tools, 1402-1403
database archiving, 1398
database modeling and design, 1403-1404
disk and space management, 1404
extract/transformation/load tools, 1406
fourth-generation languages, 1419-1420
integrity, 1406
Internet enabling, 1408
miscellaneous, 1407
object migration, 1407
operational support, 1408
PC-based emulation, 1408
performance, 1411
performance monitors, 1410-1411
plan and package analysis, 1409-1410
programming and development, 1411-1412
QMF, 1412
query, 1412-1413
recovery management and assistance, 1413-1414
repositories, 1414-1415
required, 1443
security, 1416-1417
table altering, 1396-1398
table editors, 1405-1406
third-party, 699, 1420-1422

TSO, 723
utility enhancement, 1417, 1419
"Top Ten" problem, 93-95
total number of days, returning, 123
TOTALENTRIES column (SYSIBM.SYSINDEXSPACESTATS), 1052
TOTALORDER function, 155
TOTALROWS column (SYSIBM.SYSTABLESPACESTATS), 1050
TOTAL_COST column (DSN_STATEMENT_TABLE), 1001
tournament sorts, 68
trace, auditing, 477-478
trace-based auditing, 1400
traces, 971
accounting reports, 956
performance monitoring, 929-930
accounting, 930-931
audit, 931-932
destinations, 936
global, 933
guidelines, 938-940
IFCIDs (Instrumentation Facility Component Identifiers), 937-938
monitor, 933-934
performance, 934-935
statistics, 935-936
record trace reports, 959
SQL trace reports, 959-960
tracker site, disaster recovery, 1388
TRACKMOD parameter, 777
TRACKMOD parameter, table space, 234
transaction recovery, 1414
transactions
CICS, designing, 746-750
grouping, 740
IMS/TM, design, 762-763
threads, specifying, 743-744
transformation tools, 1406
transition variables, triggers, 391
transitive closure rules, predicates, 92-93
TRANSLATE function, 155-156, 165
translation, authids, 1501-1502
triggers, 373-376, 382-384, 1539
AFTER, 379, 390
BEFORE, 390
benefits, 375-376
creating, 378-382, 389
data integrity, 300
declarative RI, 391
firing, 377-378
firing other triggers, 385
implementing, 389
INSTEAD OF, 386, 392
examples, 387-388
restrictions, 386-387
limitations, 389
logic, testing, 389
naming, 388
packages, 384-385
referential integrity, supplementing, 390-391
transition variables, 391
types, 375
VALUES statement, 391
versus check constraints, 300, 374-375
versus stored procedures, 374
WHEN clause, 382
troubleshooting performance problems, 1137-1142
TRUNC_TIMESTAMP function, 156-157
TRUNCATE function, 156, 379
emptying tables, 546-547
trusted context, 471-473
TSLOCKMODE column (PLAN_TABLE), 988
TSO (Time-Sharing Option), 706-707, 723-724
batch programs, 708-709
Bind/Rebind/Free option, 717-720
DB2I Commands option, 720

DCLGEN option, 717
Defaults option, 722
foreground, 709
batch programs, 725
ISPF panels and tables, 724-725
online design techniques, 709-712
parameters, 707-708
Precompile option, 717
Program Preparation option, 717
QMF (Query Management Facility), 722-723
resource integration, 724
Run option, 720
SPUFI, 712-722
TMP (Terminal Monitor Program), 725
tools, 723
Utilities option, 722
TSO commands, 1364
tuning, 929
80-20 rule, 1061-1062
applications, 1116-1137
buffer pool parameters, 1096-1101
AUTOSIZE, 1105-1106
DWQT, 1104
PGFIX, 1105
PGSTEAL, 1104-1105
VPSEQT, 1102-1103
buffer pools, 1108-1110
data sharing group, 1110-1114
sizes, 1106-1108
Catalog, 1089-1092
database design, 1114-1116
DSNZPARMs, 1092-1096
dynamic SQL, 575-576
IRLM, 1114
subsystems, 1089-1102, 1104-1114
tuning environment, 1064
teleprocessing, 1087-1088
z/OS, 1064
CPU usage, 1074-1076
I/O, 1076-1084
memory usage, 1064-1074
parameters and options, 1084-1087
turnover procedures, 1436
tweaking queries, 87-88
Twitter, blogs, 1428-1429
two-phase commit, 1466-1470
coordinators, 1468
multi-site updating, 1468
participants, 1468
types
outer joins, 28-29
SQL, 11-12
Typical Processing Scenario listing (23.1), 890

U

UDFs (user-defined functions), 167, 168
abends, 184
cardinality, 190
data types, 190-191
DETERMINISTIC parameter, 188
DISALLOW PARALLEL parameter, 188
DSN FUNCTION TABLE, 184
execution, external, 173-178
external, 169
EXTERNAL ACTION, 189
external scalar, 168-171
external table, 168-173
invoking, 184
naming, 180
NOT DETERMINISTIC parameter, 188
null input arguments, 189
parameter data types, 185
parameters, 185
program restrictions, 181
programming languages, external, 173
reusablility, 184
schema, 169
scratchpads, 189
SECURITY parameter, 188-189

SET CURRENT PACKAGE PATH, 184
sourced, 168, 178
SQL scalar, 168, 178-179
SQL table, 168, 179-180
SQL within, 186-187
starting and starting, 182-183
templates, 185-186
UDFs (user-defined functions), 63
UDTs (user-defined data types), 190-192, 1025
assigning values, 196
business requirements, 193-195
constants, 197-198
host variables, 197-198
LOBs, 192-193
naming, 197
set operations, 198-199
UID parameter (DSNUPROC), 1156
uncommitted reads, 52
UNCOMPRESSEDDATASIZE column (SYSIBM.SYSTABLESPACESTATS), 1050
UNICODE function, 157
UNICODE STR function, 157
Uniform Resource Locators (URLs), 689
UNION query, literals, 25
union set operations, 23-26
UNIQUE, XML indexes, 419
unique index columns, 342
units of work, batch programming, 541-542
universal table spaces, 206-207, 237-238
UNLOAD JCL listing (34.3), 1260
UNLOAD phase
REBUILD_INDEX utility, 1234
REORG INDEX utility, 1273
REORG TABLESPACE utility, 1273
UNLOAD utility, 1261
UNLOAD utility, 1260-1265
locking, 1262
phases, 1261
restarting, 1261
termination, 1261
versus DSNTIAUL, 1262
unqualified SQL, application development, 530
UPDATE statement, 36, 127-128, 131, 526
UPDATESTATSTIME column (SYSIBM.SYSINDEXSPACESTATS), 1051
UPDATESTATSTIME column (SYSIBM.SYSTABLESPACESTATS), 1049
updating
changed columns, 126-127
XML data, 424-425
updating data, 780
Updating with a Cursor listing (13.3), 514-515
upgrading compilers, COBOL, 487
UPPER function, 157, 165
URLs (Uniform Resource Locators), 689
usage, hashes, monitoring, 351
USE privilege class, 450
USE security class, 40
Usenet newsgroups, 691-692, 696, 701-702
user escalation, locks, 917
user-assisted distribution, DRDA, 1453
user-defined distinct types (UDTs), 1025
user-defined functions, 135, 167-168
abends, 184
application development, 530
cardinality, 190
data types, 190-191
DETERMINISTIC parameter, 188
DISALLOW PARALLEL parameter, 188
DSN FUNCTION TABLE, 184
execution, external UDFs, 173-178
external, 169
EXTERNAL ACTION, 189
external scalar, 168
creating, 169-171

external table, 168
creating, 171-173
invoking, 184
naming, 180
NOT DETERMINISTIC parameter, 188
null input arguments, 189
parameter data types, 185
parameters, 185
program restrictions, 181
programming languages, external UDFs, 173
reusablility, 184
schema, 169
scratchpads, 189
SECURITY parameter, 188-189
SET CURRENT PACKAGE PATH, 184
sourced, 168, 178
SQL scalar, 168, 178-179
SQL table, 168, 179-180
SQL within, 186-187
starting and stopping, 182-183
templates, 185-186

user-defined functions (UDFs), 63

user-defined VSAM, STOGROUPS, 242-243

USERFILTER column (DSN_USERQUERY_TABLE), 1133

USERNAMES table (Catalog), 879

USING parameter, table space, 221-222

UTC (Coordinated Universal Time), 121

UTILINIT phase
CHECK_DATA, 1182
COPY, 1205
COPYTOCOPY, 1216
LOAD utility, 1243
MERGECOPY, 1219
REBUILD_INDEX utility, 1234
RECOVER utility, 1228
REORG INDEX utility, 1273
REORG TABLESPACE utility, 1273-1274
RESTORE_SYSTEM utility, 1239
RUNSTATS utility, 1298
STOSPACE utility, 1311
UNLOAD utility, 1261

utilities, 1366-1367
backup and recovery, 1201-1202
BACKUP_SYSTEM, 1236-1238
COPY, 1165, 1202-1215, 1372
COPYTOCOPY, 1215-1218
MERGECOPY, 1165, 1218-1220
QUIESCE, 1220, 1222-1223
REBUILD_INDEX, 1232-1235
RECOVER, 1224-1232
RECOVER_INDEX, 1228
REPAIR, 1235
REPORT_RECOVERY, 1235-1236
RESTORE_SYSTEM, 1238-1239
Catalog, 1289
CATENFM, 1289
CATMAINT, 1289
DSNJCNVB, 1290
MODIFY RECOVERY, 1165, 1290-1293
MODIFY STATISTICS, 1165, 1293-1295
RUNSTATS, 1295-1310
STOSPACE, 1311-1313
contention, 1367, 1370
data consistency, 1176-1177
CHECK, 1177
CHECK_DATA, 1177-1186
CHECK_INDEX, 1165, 1188-1191
CHECK_LOB, 1186-1188
DIAGNOSE, 1200
REPAIR, 1191, 1198
REPAIR_DBD, 1192-1193
REPAIR_LOCATE, 1193-1195
REPAIR_SET, 1196-1198

REPORT, 1166, 1198-1200
REPORT_TABLESPACESET, 1199-1200
data movement
LOAD, 1240-1259
REORG, 1265-1288
UNLOAD, 1260-1265
disaster recovery, 1387
IBM, 1158-1159
LISTDEF, 1159-1160, 1165-1166
creating lists, 1160-1162
list expansion, 1163-1165
wildcarding, 1162-1163
monitoring, 1156-1158
online return codes, 1366
OPTIONS, 1172
QUIESCE, 1165
REBUILD, 1165
RECOVER, 1165
RECOVER_TABLESPACE, 1225-1226
REORG, 1166
RUNSTATS INDEX, 1166
RUNSTATS TABLESPACE, 1166
SQL statements
issuing, 1173-1175
stand-alone, 1314-1315
DSN1CHKR, 1318-1319
DSN1COMP, 1320-1322
DSN1COPY, 1322-1328
DSN1LOGP, 1330
DSN1PRNT, 1330-1332
DSN1SDMP, 1328-1329
DSNJLOGF, 1315-1316
DSNJU003, 1316-1317
DSNJU004, 1317-1318
TEMPLATE, 1159-1160
work data sets, 1367
Utilities option (TSO), 722
Utility area (Catalog), 881
utility enhancement tools, 1417-1419
utility JCL, generating, 1152-1156
UTILTERM phase
CHECK_DATA, 1183
COPY, 1206
COPYTOCOPY, 1216
LOAD utility, 1244
MERGECOPY, 1219
REBUILD_INDEX utility, 1234
RECOVER utility, 1229
REORG INDEX utility, 1273
REORG TABLESPACE utility, 1274
REPAIR utility, 1192
RESTORE_SYSTEM utility, 1239
RUNSTATS utility, 1298
STOSPACE utility, 1311
UNLOAD utility, 1261
UTL tools, 1417, 1419
UTPROC parameter (DSNUPROC), 1156
UTSs (universal table spaces), 206-207

V

validating screen input, 724
validity, denormalization, testing, 289-290
VALIDPROCs, 300
VALUE function, 157
values
columns, computing average, 136
UDTs (user-defined data types), assigning, 196
VALUES statement, triggers, 391
VARBINARY data type, 39-40, 245
VARBINARY function, 157
VARCHAR columns, 263, 395-396, 404
VARCHAR data type, 38, 244
VARCHAR FORMAT function, 158
VARCHAR function, 157
VARGRAPHIC columns, 395-396, 404
VARGRAPHIC data type, 38, 244
VARGRAPHIC function, 158
variable columns, 262-263
indexing, 329-330
monitoring, 263-264

variable declarations, LOB, 401

variable index keys, changing treatment, 364-365

variables

host, 504-506, 509-511

dynamic SQL, 574-575

host structures, 506

null indicators, 507-509

simulating, 533-534

indicator, 116-117, 255

naming, 501, 503-504

null indicator, 604

session, 469-471

VARIANCE function, 141

VARIANCE SAMP function, 141

varying-list SELECT class (dynamic SQL), 584-587

Varying-List SELECT Dynamic SQL listing (14.6), 585

Varying-List SELECT Dynamic SQL with Minimum SQLDA listing (14.7), 587

vendor tool guides, 1435

vendor tools, CAF (Call Attach Facility), 766-767

vendors

ISVs (Independent Software Vendors), 1424

third-party, 699

tools, 1420-1422

VERIFY_GROUP_FOR_USER function, 158

VERIFY_ROLE_FOR_USER function, 158

VERIFY_TRUSTED_ CONTEXT_FOR_ROLE_USER function, 159

VERSION column (DSN_STATEMENT_TABLE), 1001

VERSION column (DSN_USERQUERY_TABLE), 1133

VERSION column (PLAN_TABLE), 988

version maintenance, packages, 625-626

version program preparation objects, 632

versioning

changes, online schema, 370-372

stored procedures, 673-674

versions

packages, 629-630

SQL, 11

VIEW database objects, 38

VIEW privilege class, 450

VIEWs, 98, 302-304, 307-313

views

data derivation, 303

inline, 105

mimicking domain support, 304-306

non-updatable, 312

renaming columns, 306-307

restrictions, 312

specifying column names, 312

synchronization rule, 311

usage rule, 303

versus table views, 107

virtual indexes, creating, 349

virtual tables

aliases, 313

synonyms, 313

views, 302-304, 307-313

data derivation, 303

mimicking domain support, 304-306

non-updatable, 312

renaming columns, 306-307

restrictions, 312

security, 303

specifying column names, 312

synchronization rule, 311

usage rule, 303

visual Explain, query analysis, 64-66

VPSEQT parameter (buffer pool), 1102-1103

VSAM, data sets, 795-797

W

WAS (WebSphere Application Server), 693
Web browsers, 689
Web pages, 690
bookmarking, 702
Web sites, 690
search engines, 702
Web-based programming languages, 486
webinars, 1429
WebSphere, 693-694
WEEK function, 158
WEEK ISO function, 158
WHEN clause, 382
WHENEVER statement, avoiding, 500-501
WHEN_OPTIMIZE column (PLAN_TABLE), 990
WHERE clause, 526, 531
versus HAVING clause, 30-31
WHERE NOT NULL clause, 343
wildcarding LISTDEF utility, 1162-1163
WITH GRANT OPTION, 458
WITH HOLD clause, 919
WITH NO DATA clause, 1527
WLM (Workload Manager), 670-672, 1086
work data sets, utilities, 1367
Work Load Manager (WLM), 670-672, 1086
working storage, 1073
Workload Manager (WLM), 1086
workloads, indexing, 341-342
workstation DB2, 1470
WWW (World Wide Web), 689-690

X

X-locks, 914
XCF (Cross-system Coupling Facility) groups, 774
XML (Extensible Markup Language), 408-412, 425
access methods, 423
CHECK DATA utility, 427
deleting data, 424
document trees, 414-415
indexes, creating, 418-419, 425-426
inserting data, 423-424
namespaces, 417-418
parsing data, 415
pureXML, 408, 412-415
querying data, 420-424
REPORT TABLESPACE SET utility, 427
RUNSTATS utility, 426
schema validation, 416
serializing data, 415
updating data, 424-425
XMLEXISTS predicate, 425
XML area (Catalog), 880
XML data type, 39, 245
XML reference checking, CHECK_DATA utility, 1181-1182
XML scalar functions, 161-162
XML table spaces, 215
XMLEXISTS predicate, 425
XMLQUERY function, 421
XMLSCHEMAONLY option (SCOPE parameter), 1181
XMLTABLE() function, 422
XMLVALA subsystem parameter, 427
XPath, 420-421
XSRANNOTATIONINFO (V9) table (Catalog), 879
XSRCOMPONENT (V9) table (Catalog), 879
XSROBJECTCOMPONENTS (V9) table (Catalog), 879
XSROBJECTGRAMMAR (V9) table (Catalog), 879
XSROBJECTHIERARCHIES (V9) table (Catalog), 879
XSROBJECTPROPERTY (V9) table (Catalog), 879
XSROBJECTS (V9) table (Catalog), 879
XSRPROPERTY (V9) table (Catalog), 879

Y-Z

YEAR function, 158

Yevich, Lawson, and Associates website, 698

***z/Journal*, 1428**

z/OS, 411

monitoring, 979

parameters and options, 1084-1087

pureXML, 412-416, 425-427

deleting XML data, 424

indexing XML data, 418-419

inserting XML data, 423-424

namespaces, 417-418

querying XML data, 420-423

schema validation, 416

updating XML data, 424-425

system-level changes, 57

temporal data, 428-430, 446-447

business time, 430

support, 430-446

system time, 430

tuning, 1064

CPU usage, 1074-1076

I/O, 1076-1084

memory usage, 1064-1074

parameters and options, 1084-1087

XML, 408, 410-412

zAAP (Application Assist Processor), 1076

zIIP (Integrated Information Processor), 813-814, 1075

parallelism, 865-866