

## What is PostgreSQL?

PostgreSQL is a powerful, open source object-relational database system. It has more than 15 years of active development and a proven architecture that has earned it a strong reputation for reliability, data integrity, and correctness. It runs on all major operating systems, including Linux and Windows. It includes most SQL:2008 data types. It also supports storage of binary large objects, including pictures, sounds, or video. It has native programming interfaces and exceptional documentation.



PostgreSQL  
[www.postgresql.org](http://www.postgresql.org)



PostgreSQL and the PostgreSQL logo are trademarks of The PostgreSQL Global Development Group

## About PostgreSQL

An enterprise class database, PostgreSQL boasts sophisticated features such as Multi-Version Concurrency Control (MVCC), point in time recovery, tablespaces, asynchronous replication, nested transactions (savepoints), online/hot backups, a sophisticated query planner/optimizer, and write ahead logging for fault tolerance. It supports international character sets, multibyte character encodings, Unicode, and it is locale-aware for sorting, case-sensitivity, and formatting. It is highly scalable both in the sheer quantity of data it can manage and in the number of concurrent users it can accommodate. There are active PostgreSQL systems in production environments that manage in excess of 4 terabytes of data.

**Fultus™**



Published by Fultus Corporation  
[www.fultus.com](http://www.fultus.com)



PostgreSQL  
[www.postgresql.org](http://www.postgresql.org)



# PostgreSQL 9.0

## The SQL Language

### Volume I



By The PostgreSQL Global Development Group

PostgreSQL 9.0 • The SQL Language





**PostgreSQL 9.0**  
**Official Documentation**

**Volume I**  
**The SQL Language**



*Fultus™ Books*

# PostgreSQL



## **PostgreSQL 9.0 Official Documentation**

### **Volume I**

### **The SQL Language**

ISBN-10: 1-59682-246-5

ISBN-13: 978-1-59682-246-7

Copyright © 1996-2011 The PostgreSQL Global Development Group

Cover design and book layout by Fultus Corporation



*Published by Fultus Corporation*

Publisher Web: *www.fultus.com*

Linbrary - Linux Library: *www.linbrary.com*

Online Bookstore: *store.fultus.com*

email: *production@fultus.com*



This material may only be distributed subject to the terms and conditions set forth in the BSD License (presently available at <http://www.postgresql.org/about/licence>).

PostgreSQL and PostgreSQL logo are trademarks or registered trademarks of The PostgreSQL Global Development Group, in the U.S. and other countries. All product names and services identified throughout this manual are trademarks or registered trademarks of their respective companies.

The author and publisher have made every effort in the preparation of this book to ensure the accuracy of the information. However, the information contained in this book is offered without warranty, either express or implied. Neither the author nor the publisher nor any dealer or distributor will be held liable for any damages caused or alleged to be caused either directly or indirectly by this book.

# Table of Contents

List of Tables.....	13
List of Examples .....	16
License.....	17
Abstract .....	18
Preface .....	19
What is PostgreSQL? .....	19
A Brief History of PostgreSQL.....	20
Conventions .....	23
Further Information.....	23
Bug Reporting Guidelines .....	24
<b>Part I Tutorial .....</b>	<b>29</b>
<b>Chapter 1. Getting Started .....</b>	<b>30</b>
1.1. Installation.....	30
1.2. Architectural Fundamentals .....	30
1.3. Creating a Database .....	31
1.4. Accessing a Database.....	33
<b>Chapter 2 The SQL Language .....</b>	<b>35</b>
2.1. Introduction .....	35
2.2. Concepts .....	35
2.3. Creating a New Table .....	36
2.4. Populating a Table With Rows.....	37
2.5. Querying a Table .....	38
2.6. Joins Between Tables .....	39
2.7. Aggregate Functions.....	42
2.8. Updates.....	44
<b>Chapter 3. Advanced Features.....</b>	<b>45</b>
3.1. Introduction .....	45
3.2. Views.....	45
3.3. Foreign Keys .....	45
3.4. Transactions .....	46
3.5. Window Functions.....	49

---

3.6. Inheritance .....	52
3.7. Conclusion .....	54
<b>Part II. The SQL Language.....</b>	<b>55</b>
<b>Chapter 4. SQL Syntax.....</b>	<b>56</b>
4.1. Lexical Structure.....	56
4.1.1. Identifiers and Key Words.....	57
4.1.2. Constants.....	58
4.1.2.1. String Constants.....	59
4.1.2.2. String Constants with C-Style Escapes.....	59
4.1.2.3. String Constants with Unicode Escapes.....	60
4.1.2.4. Dollar-Quoted String Constants.....	61
4.1.2.5. Bit-String Constants.....	62
4.1.2.6. Numeric Constants.....	63
4.1.2.7. Constants of Other Types.....	64
4.1.3. Operators.....	64
4.1.4. Special Characters .....	65
4.1.5. Comments .....	65
4.1.6. Lexical Precedence .....	66
4.2. Value Expressions.....	67
4.2.1. Column References .....	68
4.2.2. Positional Parameters .....	68
4.2.3. Subscripts .....	69
4.2.4. Field Selection.....	69
4.2.5. Operator Invocations.....	70
4.2.6. Function Calls.....	70
4.2.7. Aggregate Expressions.....	70
4.2.8. Window Function Calls.....	72
4.2.9. Type Casts.....	74
4.2.10. Scalar Subqueries .....	75
4.2.11. Array Constructors .....	75
4.2.12. Row Constructors.....	77
4.2.13. Expression Evaluation Rules .....	78
4.3. Calling Functions .....	79
4.3.1. Using positional notation.....	80
4.3.2. Using named notation .....	80
4.3.3. Using mixed notation .....	81
<b>Chapter 5. Data Definition .....</b>	<b>82</b>
5.1. Table Basics.....	82
5.2. Default Values .....	84

---

## Table of Contents

---

5.3. Constraints .....	85
5.3.1. Check Constraints .....	85
5.3.2. Not-Null Constraints .....	87
5.3.3. Unique Constraints .....	88
5.3.4. Primary Keys.....	89
5.3.5. Foreign Keys.....	90
5.3.6. Exclusion Constraints .....	93
5.4. System Columns.....	93
5.5. Modifying Tables .....	95
5.5.1. Adding a Column.....	95
5.5.2. Removing a Column .....	96
5.5.3. Adding a Constraint .....	96
5.5.4. Removing a Constraint.....	96
5.5.5. Changing a Column's Default Value.....	97
5.5.6. Changing a Column's Data Type .....	97
5.5.7. Renaming a Column .....	97
5.5.8. Renaming a Table .....	98
5.6. Privileges .....	98
5.7. Schemas .....	99
5.7.1. Creating a Schema.....	99
5.7.2. The Public Schema .....	100
5.7.3. The Schema Search Path.....	101
5.7.4. Schemas and Privileges .....	102
5.7.5. The System Catalog Schema .....	102
5.7.6. Usage Patterns .....	103
5.7.7. Portability .....	103
5.8. Inheritance.....	104
5.8.1. Caveats.....	107
5.9. Partitioning.....	108
5.9.1. Overview .....	108
5.9.2. Implementing Partitioning.....	109
5.9.3. Managing Partitions.....	112
5.9.4. Partitioning and Constraint Exclusion .....	113
5.9.5. Alternative Partitioning Methods.....	115
5.9.6. Caveats.....	115
5.10. Other Database Objects.....	116
5.11. Dependency Tracking .....	117
<b>Chapter 6. Data Manipulation.....</b>	<b>118</b>
6.1. Inserting Data .....	118

---

6.2. Updating Data .....	119
<b>Chapter 7. Queries.....</b>	<b>121</b>
7.1. Overview .....	121
7.2. Table Expressions.....	122
7.2.1. The FROM Clause .....	122
7.2.1.1. Joined Tables .....	122
7.2.1.2. Table and Column Aliases .....	126
7.2.1.3. Subqueries .....	127
7.2.1.4. Table Functions.....	127
7.2.2. The WHERE Clause.....	128
7.2.3. The GROUP BY and HAVING Clauses.....	130
7.2.4. Window Function Processing.....	132
7.3. Select Lists.....	132
7.3.1. Select-List Items.....	133
7.3.2. Column Labels.....	133
7.3.3. DISTINCT.....	134
7.4. Combining Queries.....	134
7.5. Sorting Rows.....	135
7.6. LIMIT and OFFSET.....	136
7.7. VALUES Lists.....	137
7.8. WITH Queries (Common Table Expressions).....	138
<b>Chapter 8. Data Types .....</b>	<b>142</b>
8.1. Numeric Types .....	143
8.1.1. Integer Types .....	144
8.1.2. Arbitrary Precision Numbers .....	144
8.1.3. Floating-Point Types.....	146
8.1.4. Serial Types.....	147
8.2. Monetary Types .....	148
8.3. Character Types .....	149
8.4. Binary Data Types.....	151
8.4.1. bytea hex format .....	152
8.4.2. bytea escape format.....	152
8.5. Date/Time Types.....	154
8.5.1. Date/Time Input.....	155
8.5.1.1. Dates.....	156
8.5.1.2. Times .....	157
8.5.1.3. Time Stamps.....	158
8.5.1.4. Special Values .....	159

---

## Table of Contents

---

8.5.2. Date/Time Output .....	159
8.5.3. Time Zones .....	160
8.5.4. Interval Input .....	162
8.5.5. Interval Output .....	165
8.5.6. Internals .....	165
8.6. Boolean Type .....	165
8.7. Enumerated Types .....	167
8.7.1. Declaration of Enumerated Types .....	167
8.7.2. Ordering .....	167
8.7.3. Type Safety .....	168
8.7.4. Implementation Details .....	168
8.8. Geometric Types.....	169
8.8.1. Points.....	169
8.8.2. Line Segments.....	169
8.8.3. Boxes.....	170
8.8.4. Paths.....	170
8.8.5. Polygons .....	170
8.8.6. Circles.....	171
8.9. Network Address Types .....	171
8.9.1. inet.....	171
8.9.2. cidr.....	172
8.9.3. inet vs. cidr .....	173
8.9.4. macaddr.....	173
8.10. Bit String Types.....	173
8.11. Text Search Types .....	174
8.11.1. tsvector.....	174
8.11.2. tsquery .....	176
8.12. UUID Type.....	177
8.13. XML Type .....	177
8.13.1. Creating XML Values.....	178
8.13.2. Encoding Handling.....	179
8.13.3. Accessing XML Values .....	179
8.14. Arrays .....	180
8.14.1. Declaration of Array Types.....	180
8.14.2. Array Value Input .....	181
8.14.3. Accessing Arrays.....	182
8.14.4. Modifying Arrays.....	184
8.14.5. Searching in Arrays.....	187
8.14.6. Array Input and Output Syntax.....	187

8.15. Composite Types .....	189
8.15.1. Declaration of Composite Types .....	189
8.15.2. Composite Value Input .....	190
8.15.3. Accessing Composite Types .....	191
8.15.4. Modifying Composite Types .....	192
8.15.5. Composite Type Input and Output Syntax .....	192
8.16. Object Identifier Types .....	193
8.17. Pseudo-Types .....	195
<b>Chapter 9. Functions and Operators .....</b>	<b>197</b>
9.1. Logical Operators .....	197
9.2. Comparison Operators .....	198
9.3. Mathematical Functions and Operators .....	200
9.4. String Functions and Operators .....	203
9.5. Binary String Functions and Operators .....	212
9.6. Bit String Functions and Operators .....	214
9.7. Pattern Matching .....	215
9.7.1. LIKE .....	215
9.7.2. SIMILAR TO Regular Expressions .....	216
9.7.3. POSIX Regular Expressions .....	217
9.7.3.1. Regular Expression Details .....	221
9.7.3.2. Bracket Expressions .....	223
9.7.3.3. Regular Expression Escapes .....	224
9.7.3.4. Regular Expression Metasyntax .....	227
9.7.3.5. Regular Expression Matching Rules .....	228
9.7.3.6. Limits and Compatibility .....	230
9.7.3.7. Basic Regular Expressions .....	231
9.8. Data Type Formatting Functions .....	231
9.9. Date/Time Functions and Operators .....	238
9.9.1. EXTRACT, date_part .....	241
9.9.2. date_trunc .....	245
9.9.3. AT TIME ZONE .....	246
9.9.4. Current Date/Time .....	247
9.9.5. Delaying Execution .....	249
9.10. Enum Support Functions .....	249
9.11. Geometric Functions and Operators .....	250
9.12. Network Address Functions and Operators .....	253
9.13. Text Search Functions and Operators .....	255
9.14. XML Functions .....	258
9.14.1. Producing XML Content .....	258

## Table of Contents

---

9.14.1.1. xmlcomment .....	258
9.14.1.2. xmlconcat .....	259
9.14.1.3. xmlelement .....	259
9.14.1.4. xmlforest.....	260
9.14.1.5. xmlpi.....	261
9.14.1.6. xmlroot .....	261
9.14.1.7. xmlagg .....	262
9.14.1.8. XML Predicates .....	262
9.14.2. Processing XML .....	263
9.14.3. Mapping Tables to XML.....	263
9.15. Sequence Manipulation Functions .....	267
9.16. Conditional Expressions .....	269
9.16.1. CASE .....	270
9.16.2. COALESCE.....	271
9.16.3. NULLIF .....	271
9.16.4. GREATEST and LEAST .....	272
9.17. Array Functions and Operators .....	272
9.18. Aggregate Functions.....	274
9.19. Window Functions.....	277
9.20. Subquery Expressions .....	279
9.20.1. EXISTS.....	279
9.20.2. IN .....	279
9.20.3. NOT IN .....	280
9.20.4. ANY/SOME.....	281
9.20.5. ALL.....	281
9.20.6. Row-wise Comparison .....	282
9.21. Row and Array Comparisons .....	282
9.21.1. IN .....	282
9.21.2. NOT IN .....	283
9.21.3. ANY/SOME (array).....	283
9.21.4. ALL (array).....	284
9.21.5. Row-wise Comparison .....	284
9.22. Set Returning Functions.....	285
9.23. System Information Functions.....	288
9.24. System Administration Functions .....	298
9.25. Trigger Functions.....	305
<b>Chapter 10. Type Conversion .....</b>	<b>306</b>
10.1. Overview.....	306
10.2. Operators.....	308

---

10.3. Functions.....	311
10.4. Value Storage .....	314
10.5. UNION, CASE, and Related Constructs .....	315
<b>Chapter 11. Indexes .....</b>	<b>317</b>
11.1. Introduction.....	317
11.2. Index Types .....	318
11.3. Multicolumn Indexes .....	320
11.4. Indexes and ORDER BY.....	321
11.5. Combining Multiple Indexes .....	322
11.6. Unique Indexes .....	323
11.7. Indexes on Expressions.....	324
11.8. Partial Indexes.....	325
11.9. Operator Classes and Operator Families .....	328
11.10. Examining Index Usage .....	329
<b>Chapter 12. Full Text Search .....</b>	<b>331</b>
12.1. Introduction.....	331
12.1.1. What Is a Document?.....	332
12.1.2. Basic Text Matching.....	333
12.1.3. Configurations.....	334
12.2. Tables and Indexes .....	335
12.2.1. Searching a Table .....	335
12.2.2. Creating Indexes.....	336
12.3. Controlling Text Search .....	337
12.3.1. Parsing Documents .....	338
12.3.2. Parsing Queries .....	339
12.3.3. Ranking Search Results .....	340
12.3.4. Highlighting Results.....	343
12.4. Additional Features.....	344
12.4.1. Manipulating Documents .....	344
12.4.2. Manipulating Queries.....	345
12.4.2.1. Query Rewriting.....	346
12.4.3. Triggers for Automatic Updates.....	348
12.4.4. Gathering Document Statistics.....	349
12.5. Parsers .....	350
12.6. Dictionaries.....	352
12.6.1. Stop Words.....	353
12.6.2. Simple Dictionary.....	354
12.6.3. Synonym Dictionary.....	355
12.6.4. Thesaurus Dictionary .....	357

---

## Table of Contents

---

12.6.4.1. Thesaurus Configuration.....	358
12.6.4.2. Thesaurus Example .....	358
12.6.5. Ispell Dictionary.....	359
12.6.6. Snowball Dictionary.....	361
12.7. Configuration Example.....	361
12.8. Testing and Debugging Text Search .....	363
12.8.1. Configuration Testing.....	363
12.8.2. Parser Testing.....	365
12.8.3. Dictionary Testing .....	366
12.9. GiST and GIN Index Types .....	367
12.10. psql Support.....	368
12.11. Limitations .....	371
12.12. Migration from Pre-8.3 Text Search.....	372
<b>Chapter 13. Concurrency Control.....</b>	<b>373</b>
13.1. Introduction.....	373
13.2. Transaction Isolation .....	373
13.2.1. Read Committed Isolation Level.....	374
13.2.2. Serializable Isolation Level .....	376
13.2.2.1. Serializable Isolation versus True Serializability.....	377
13.3. Explicit Locking.....	378
13.3.1. Table-Level Locks.....	378
13.3.2. Row-Level Locks .....	381
13.3.3. Deadlocks .....	382
13.3.4. Advisory Locks.....	383
13.4. Data Consistency Checks at the Application Level .....	384
13.5. Locking and Indexes.....	385
<b>Chapter 14. Performance Tips.....</b>	<b>386</b>
14.1. Using EXPLAIN .....	386
14.2. Statistics Used by the Planner .....	392
14.3. Controlling the Planner with Explicit JOIN Clauses .....	394
14.4. Populating a Database.....	396
14.4.1. Disable Autocommit .....	396
14.4.2. Use COPY .....	396
14.4.3. Remove Indexes.....	397
14.4.4. Remove Foreign Key Constraints .....	397
14.4.5. Increase maintenance_work_mem.....	398
14.4.6. Increase checkpoint_segments .....	398
14.4.7. Disable WAL archival and streaming replication .....	398
14.4.8. Run ANALYZE Afterwards .....	399

---

14.4.9. Some Notes About <code>pg_dump</code> .....	399
14.5. Non-Durable Settings.....	400
<b>List of Volumes</b> .....	<b>401</b>
<b>Index</b> .....	<b>404</b>
<b>Linbrary™ Advertising Club (LAC)</b> .....	<b>415</b>
Your Advertising Here.....	423

# List of Tables

Table 4.1. Backslash Escape Sequences.....	59
Table 4.2. Operator Precedence (decreasing) .....	67
Table 8.1. Data Types.....	143
Table 8.2. Numeric Types .....	144
Table 8.3. Monetary Types.....	149
Table 8.4. Character Types.....	149
Table 8.5. Special Character Types .....	151
Table 8.6. Binary Data Types .....	151
Table 8.7. <code>bytea</code> Literal Escaped Octets .....	153
Table 8.8. <code>bytea</code> Output Escaped Octets .....	153
Table 8.9. Date/Time Types .....	154
Table 8.10. Date Input.....	156
Table 8.11. Time Input.....	157
Table 8.12. Time Zone Input.....	157
Table 8.13. Special Date/Time Inputs .....	159
Table 8.14. Date/Time Output Styles.....	160
Table 8.15. Date Order Conventions .....	160
Table 8.16. ISO 8601 interval unit abbreviations .....	163
Table 8.17. Interval Input .....	164
Table 8.18. Interval Output Style Examples .....	165
Table 8.19. Boolean Data Type .....	166
Table 8.20. Geometric Types.....	169
Table 8.21. Network Address Types .....	171
Table 8.22. <code>cidr</code> Type Input Examples .....	172
Table 8.23. Object Identifier Types .....	194
Table 8.24. Pseudo-Types.....	196
Table 9.1. Comparison Operators .....	198
Table 9.2. Mathematical Operators.....	201
Table 9.3. Mathematical Functions .....	202
Table 9.4. Trigonometric Functions .....	203

---

Table 9.5. SQL String Functions and Operators .....	204
Table 9.6. Other String Functions .....	208
Table 9.7. Built-in Conversions .....	212
Table 9.8. SQL Binary String Functions and Operators.....	213
Table 9.9. Other Binary String Functions .....	214
Table 9.10. Bit String Operators .....	214
Table 9.11. Regular Expression Match Operators .....	217
Table 9.12. Regular Expression Atoms .....	222
Table 9.13. Regular Expression Quantifiers .....	223
Table 9.14. Regular Expression Constraints.....	223
Table 9.15. Regular Expression Character-Entry Escapes.....	226
Table 9.16. Regular Expression Class-Shorthand Escapes.....	226
Table 9.17. Regular Expression Constraint Escapes.....	226
Table 9.18. Regular Expression Back References.....	227
Table 9.19. ARE Embedded-Option Letters .....	228
Table 9.20. Formatting Functions .....	232
Table 9.21. Template Patterns for Date/Time Formatting.....	234
Table 9.22. Template Pattern Modifiers for Date/Time Formatting .....	234
Table 9.23. Template Patterns for Numeric Formatting.....	236
Table 9.24. Template Pattern Modifiers for Numeric Formatting .....	237
Table 9.25. to_char Examples.....	238
Table 9.26. Date/Time Operators .....	239
Table 9.27. Date/Time Functions .....	240
Table 9.28. AT TIME ZONE Variants.....	246
Table 9.29. Enum Support Functions .....	250
Table 9.30. Geometric Operators .....	251
Table 9.31. Geometric Functions.....	252
Table 9.32. Geometric Type Conversion Functions .....	252
Table 9.33. cidr and inet Operators.....	253
Table 9.34. cidr and inet Functions .....	254
Table 9.35. macaddr Functions .....	255
Table 9.36. Text Search Operators .....	255
Table 9.37. Text Search Functions.....	257
Table 9.38. Text Search Debugging Functions.....	258
Table 9.39. Sequence Functions .....	267
Table 9.40. Array Operators .....	272

---

## List of Tables

---

Table 9.41. Array Functions.....	273
Table 9.42. General-Purpose Aggregate Functions .....	275
Table 9.43. Aggregate Functions for Statistics .....	277
Table 9.44. General-Purpose Window Functions .....	278
Table 9.45. Series Generating Functions .....	286
Table 9.46. Subscript Generating Functions .....	287
Table 9.47. Session Information Functions .....	289
Table 9.48. Access Privilege Inquiry Functions .....	291
Table 9.49. Schema Visibility Inquiry Functions .....	293
Table 9.50. System Catalog Information Functions.....	295
Table 9.51. Comment Information Functions.....	296
Table 9.52. Transaction IDs and snapshots.....	297
Table 9.53. Snapshot components.....	297
Table 9.54. Configuration Settings Functions .....	298
Table 9.55. Server Signalling Functions .....	298
Table 9.56. Backup Control Functions.....	299
Table 9.57. Recovery Information Functions.....	301
Table 9.58. Database Object Size Functions.....	302
Table 9.59. Database Object Location Functions.....	302
Table 9.60. Generic File Access Functions .....	303
Table 9.61. Advisory Lock Functions .....	304
Table 12.1. Default Parser's Token Types .....	351
Table 13.1. SQL Transaction Isolation Levels.....	374
Table 13.2. Conflicting lock modes.....	381

## List of Examples

Example 8.1. Using the character types .....	151
Example 8.2. Using the boolean type .....	166
Example 8.3. Using the bit string types .....	174
Example 9.1. XSLT stylesheet for converting SQL/XML output to HTML .....	267
Example 10.1. Factorial Operator Type Resolution .....	309
Example 10.2. String Concatenation Operator Type Resolution .....	310
Example 10.3. Absolute-Value and Negation Operator Type Resolution .....	311
Example 10.4. Rounding Function Argument Type Resolution .....	313
Example 10.5. Substring Function Type Resolution .....	314
Example 10.6. character Storage Type Conversion.....	315
Example 10.7. Type Resolution with Underspecified Types in a Union .....	316
Example 10.8. Type Resolution in a Simple Union.....	316
Example 10.9. Type Resolution in a Transposed Union.....	316
Example 11.1. Setting up a Partial Index to Exclude Common Values.....	325
Example 11.2. Setting up a Partial Index to Exclude Uninteresting Values .....	326
Example 11.3. Setting up a Partial Unique Index.....	327

# License

PostgreSQL is Copyright © 1996-2011 by the PostgreSQL Global Development Group and is distributed under the terms of the license of the University of California below.

Postgres95 is Copyright © 1994-5 by the Regents of the University of California.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose, without fee, and without a written agreement is hereby granted, provided that the above copyright notice and this paragraph and the following two paragraphs appear in all copies.

IN NO EVENT SHALL THE UNIVERSITY OF CALIFORNIA BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF THE UNIVERSITY OF CALIFORNIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE UNIVERSITY OF CALIFORNIA SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON AN "AS-IS" BASIS, AND THE UNIVERSITY OF CALIFORNIA HAS NO OBLIGATIONS TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

## Abstract

Welcome to the *PostgreSQL 9.0 Official Documentation*! After many years of development, PostgreSQL has become feature-complete in many areas. This release shows a targeted approach to adding features (e.g., authentication, monitoring, space reuse), and adds capabilities defined in the later SQL standards.

# Preface

This book is the official documentation of PostgreSQL. It has been written by the PostgreSQL developers and other volunteers in parallel to the development of the PostgreSQL software. It describes all the functionality that the current version of PostgreSQL officially supports.

To make the large amount of information about PostgreSQL manageable, this book has been organized in several parts. Each part is targeted at a different class of users, or at users in different stages of their PostgreSQL experience:

- *Part I* (page 29) is an informal introduction for new users.
- *Part II* (page 55) documents the SQL query language environment, including data types and functions, as well as user-level performance tuning. Every PostgreSQL user should read this.
- *Part III* (Vol.II) describes the installation and administration of the server. Everyone who runs a PostgreSQL server, be it for private use or for others, should read this part.
- *Part IV* (Vol.II) describes the programming interfaces for PostgreSQL client programs.
- *Part V* (Vol.III) contains information for advanced users about the extensibility capabilities of the server. Topics include user-defined data types and functions.
- *Part VI* (Vol.IV) contains reference information about SQL commands, client and server programs. This part supports the other parts with structured information sorted by command or program.
- *Part VII* (Vol.V) contains assorted information that might be of use to PostgreSQL developers.

## What is PostgreSQL?

PostgreSQL is an object-relational database management system (ORDBMS) based on *POSTGRES, Version 4.2*<sup>1</sup>, developed at the University of California at Berkeley Computer

---

<sup>1</sup> <http://s2k-ftp.cs.berkeley.edu:8000/postgres/postgres.html>

Science Department. POSTGRES pioneered many concepts that only became available in some commercial database systems much later.

PostgreSQL is an open-source descendant of this original Berkeley code. It supports a large part of the SQL standard and offers many modern features:

- complex queries
- foreign keys
- triggers
- views
- transactional integrity
- multiversion concurrency control

Also, PostgreSQL can be extended by the user in many ways, for example by adding new

- data types
- functions
- operators
- aggregate functions
- index methods
- procedural languages

And because of the liberal license, PostgreSQL can be used, modified, and distributed by anyone free of charge for any purpose, be it private, commercial, or academic.

## A Brief History of PostgreSQL

The object-relational database management system now known as PostgreSQL is derived from the POSTGRES package written at the University of California at Berkeley. With over two decades of development behind it, PostgreSQL is now the most advanced open-source database available anywhere.

### The Berkeley POSTGRES Project

The POSTGRES project, led by Professor Michael Stonebraker, was sponsored by the Defense Advanced Research Projects Agency (DARPA), the Army Research Office (ARO), the National Science Foundation (NSF), and ESL, Inc. The implementation of POSTGRES began in 1986. The initial concepts for the system were presented in *The design of POSTGRES*<sup>2</sup>, and the definition of the initial data model appeared in *The POSTGRES data*

---

<sup>2</sup> <http://s2k-ftp.cs.berkeley.edu:8000/postgres/papers/ERL-M85-95.pdf>

## Preface

---

*model*<sup>3</sup>. The design of the rule system at that time was described in *The design of the POSTGRES rules system* (see Section "Bibliography" – Vol.V). The rationale and architecture of the storage manager were detailed in *The design of the POSTGRES storage system* <sup>4</sup>.

POSTGRES has undergone several major releases since then. The first "demoware" system became operational in 1987 and was shown at the 1988 ACM-SIGMOD Conference. Version 1, described in *The implementation of POSTGRES*<sup>5</sup>, was released to a few external users in June 1989. In response to a critique of the first rule system (*commentary on the POSTGRES rules system*<sup>6</sup>), the rule system was redesigned (*On Rules, Procedures, Caching and Views in Database Systems*<sup>7</sup>), and Version 2 was released in June 1990 with the new rule system. Version 3 appeared in 1991 and added support for multiple storage managers, an improved query executor, and a rewritten rule system. For the most part, subsequent releases until Postgres95 (see below) focused on portability and reliability.

POSTGRES has been used to implement many different research and production applications. These include: a financial data analysis system, a jet engine performance monitoring package, an asteroid tracking database, a medical information database, and several geographic information systems. POSTGRES has also been used as an educational tool at several universities. Finally, Illustra Information Technologies (later merged into *Informix*<sup>8</sup>, which is now owned by *IBM*<sup>9</sup>) picked up the code and commercialized it. In late 1992, POSTGRES became the primary data manager for the *Sequoia 2000 scientific computing project*<sup>10</sup>.

The size of the external user community nearly doubled during 1993. It became increasingly obvious that maintenance of the prototype code and support was taking up large amounts of time that should have been devoted to database research. In an effort to reduce this support burden, the Berkeley POSTGRES project officially ended with Version 4.2.

### Postgres95

In 1994, Andrew Yu and Jolly Chen added an SQL language interpreter to POSTGRES. Under a new name, Postgres95 was subsequently released to the web to find its own way in the world as an open-source descendant of the original POSTGRES Berkeley code.

---

<sup>3</sup> <http://s2k-ftp.cs.berkeley.edu:8000/postgres/papers/ERL-M87-13.pdf>

<sup>4</sup> <http://s2k-ftp.cs.berkeley.edu:8000/postgres/papers/ERL-M87-06.pdf>

<sup>5</sup> <http://s2k-ftp.cs.berkeley.edu:8000/postgres/papers/ERL-M90-34.pdf>

<sup>6</sup> <http://s2k-ftp.cs.berkeley.edu:8000/postgres/papers/ERL-M89-82.pdf>

<sup>7</sup> <http://s2k-ftp.cs.berkeley.edu:8000/postgres/papers/ERL-M90-36.pdf>

<sup>8</sup> <http://www.informix.com/>

<sup>9</sup> <http://www.ibm.com/>

<sup>10</sup> [http://meteora.ucsd.edu/s2k/s2k\\_home.html](http://meteora.ucsd.edu/s2k/s2k_home.html)

Postgres95 code was completely ANSI C and trimmed in size by 25%. Many internal changes improved performance and maintainability. Postgres95 release 1.0.x ran about 30-50% faster on the Wisconsin Benchmark compared to POSTGRES, Version 4.2. Apart from bug fixes, the following were the major enhancements:

- The query language PostQUEL was replaced with SQL (implemented in the server). Subqueries were not supported until PostgreSQL (see below), but they could be imitated in Postgres95 with user-defined SQL functions. Aggregate functions were re-implemented. Support for the `GROUP BY` query clause was also added.
- A new program (`psql`) was provided for interactive SQL queries, which used GNU Readline. This largely superseded the old monitor program.
- A new front-end library, `libpgtcl`, supported Tcl-based clients. A sample shell, `pgtclsh`, provided new Tcl commands to interface Tcl programs with the Postgres95 server.
- The large-object interface was overhauled. The inversion large objects were the only mechanism for storing large objects. (The inversion file system was removed.)
- The instance-level rule system was removed. Rules were still available as rewrite rules.
- A short tutorial introducing regular SQL features as well as those of Postgres95 was distributed with the source code
- GNU make (instead of BSD make) was used for the build. Also, Postgres95 could be compiled with an unpatched GCC (data alignment of doubles was fixed).

## PostgreSQL

By 1996, it became clear that the name "Postgres95" would not stand the test of time. We chose a new name, PostgreSQL, to reflect the relationship between the original POSTGRES and the more recent versions with SQL capability. At the same time, we set the version numbering to start at 6.0, putting the numbers back into the sequence originally begun by the Berkeley POSTGRES project.

Many people continue to refer to PostgreSQL as "Postgres" (now rarely in all capital letters) because of tradition or because it is easier to pronounce. This usage is widely accepted as a nickname or alias.

The emphasis during development of Postgres95 was on identifying and understanding existing problems in the server code. With PostgreSQL, the emphasis has shifted to augmenting features and capabilities, although work continues in all areas.

Details about what has happened in PostgreSQL since then can be found in *Appendix E - Vol.V*.

### Conventions

This book uses the following typographical conventions to mark certain portions of text: new terms, foreign phrases, and other important passages are emphasized in *italics*. Everything that represents input or output of the computer, in particular commands, program code, and screen output, is shown in a monospaced font (`example`). Within such passages, italics (*example*) indicate placeholders; you must insert an actual value instead of the placeholder. On occasion, parts of program code are emphasized in bold face (**example**), if they have been added or changed since the preceding example.

The following conventions are used in the synopsis of a command: brackets ([ and ]) indicate optional parts. (In the synopsis of a Tcl command, question marks (?) are used instead, as is usual in Tcl.) Braces ({ and }) and vertical lines (|) indicate that you must choose one alternative. Dots (. . .) mean that the preceding element can be repeated.

Where it enhances the clarity, SQL commands are preceded by the prompt =>, and shell commands are preceded by the prompt \$. Normally, prompts are not shown, though.

An *administrator* is generally a person who is in charge of installing and running the server. A *user* could be anyone who is using, or wants to use, any part of the PostgreSQL system. These terms should not be interpreted too narrowly; this book does not have fixed presumptions about system administration procedures.

### Further Information

Besides the documentation, that is, this book, there are other resources about PostgreSQL:  
Wiki

The PostgreSQL *wiki*<sup>11</sup> contains the project's *FAQ*<sup>12</sup> (Frequently Asked Questions) list, *TODO*<sup>13</sup> list, and detailed information about many more topics.

Web Site

The PostgreSQL *web site*<sup>14</sup> carries details on the latest release and other information to make your work or play with PostgreSQL more productive.

Mailing Lists

The mailing lists are a good place to have your questions answered, to share experiences with other users, and to contact the developers. Consult the PostgreSQL web site for details.

---

<sup>11</sup> <http://wiki.postgresql.org/>

<sup>12</sup> [http://wiki.postgresql.org/wiki/Frequently\\_Asked\\_Questions](http://wiki.postgresql.org/wiki/Frequently_Asked_Questions)

<sup>13</sup> <http://wiki.postgresql.org/wiki/ToDo>

<sup>14</sup> <http://www.postgresql.org/>

Yourself!

PostgreSQL is an open-source project. As such, it depends on the user community for ongoing support. As you begin to use PostgreSQL, you will rely on others for help, either through the documentation or through the mailing lists. Consider contributing your knowledge back. Read the mailing lists and answer questions. If you learn something which is not in the documentation, write it up and contribute it. If you add features to the code, contribute them.

## Bug Reporting Guidelines

When you find a bug in PostgreSQL we want to hear about it. Your bug reports play an important part in making PostgreSQL more reliable because even the utmost care cannot guarantee that every part of PostgreSQL will work on every platform under every circumstance.

The following suggestions are intended to assist you in forming bug reports that can be handled in an effective fashion. No one is required to follow them but doing so tends to be to everyone's advantage.

We cannot promise to fix every bug right away. If the bug is obvious, critical, or affects a lot of users, chances are good that someone will look into it. It could also happen that we tell you to update to a newer version to see if the bug happens there. Or we might decide that the bug cannot be fixed before some major rewrite we might be planning is done. Or perhaps it is simply too hard and there are more important things on the agenda. If you need help immediately, consider obtaining a commercial support contract.

## Identifying Bugs

Before you report a bug, please read and re-read the documentation to verify that you can really do whatever it is you are trying. If it is not clear from the documentation whether you can do something or not, please report that too; it is a bug in the documentation. If it turns out that a program does something different from what the documentation says, that is a bug. That might include, but is not limited to, the following circumstances:

- A program terminates with a fatal signal or an operating system error message that would point to a problem in the program. (A counterexample might be a "disk full" message, since you have to fix that yourself.)
- A program produces the wrong output for any given input.
- A program refuses to accept valid input (as defined in the documentation).
- A program accepts invalid input without a notice or error message. But keep in mind that your idea of invalid input might be our idea of an extension or compatibility with traditional practice.

- PostgreSQL fails to compile, build, or install according to the instructions on supported platforms.

Here "program" refers to any executable, not only the backend server.

Being slow or resource-hogging is not necessarily a bug. Read the documentation or ask on one of the mailing lists for help in tuning your applications. Failing to comply to the SQL standard is not necessarily a bug either, unless compliance for the specific feature is explicitly claimed.

Before you continue, check on the TODO list and in the FAQ to see if your bug is already known. If you cannot decode the information on the TODO list, report your problem. The least we can do is make the TODO list clearer.

### What to report

The most important thing to remember about bug reporting is to state all the facts and only facts. Do not speculate what you think went wrong, what "it seemed to do", or which part of the program has a fault. If you are not familiar with the implementation you would probably guess wrong and not help us a bit. And even if you are, educated explanations are a great supplement to but no substitute for facts. If we are going to fix the bug we still have to see it happen for ourselves first. Reporting the bare facts is relatively straightforward (you can probably copy and paste them from the screen) but all too often important details are left out because someone thought it does not matter or the report would be understood anyway.

The following items should be contained in every bug report:

- The exact sequence of steps *from program start-up* necessary to reproduce the problem. This should be self-contained; it is not enough to send in a bare `SELECT` statement without the preceding `CREATE TABLE` and `INSERT` statements, if the output should depend on the data in the tables. We do not have the time to reverse-engineer your database schema, and if we are supposed to make up our own data we would probably miss the problem.

The best format for a test case for SQL-related problems is a file that can be run through the `psql` frontend that shows the problem. (Be sure to not have anything in your `~/.psqlrc` start-up file.) An easy way to create this file is to use `pg_dump` to dump out the table declarations and data needed to set the scene, then add the problem query. You are encouraged to minimize the size of your example, but this is not absolutely necessary. If the bug is reproducible, we will find it either way.

If your application uses some other client interface, such as PHP, then please try to isolate the offending queries. We will probably not set up a web server to reproduce

your problem. In any case remember to provide the exact input files; do not guess that the problem happens for "large files" or "midsize databases", etc. since this information is too inexact to be of use.

- The output you got. Please do not say that it "didn't work" or "crashed". If there is an error message, show it, even if you do not understand it. If the program terminates with an operating system error, say which. If nothing at all happens, say so. Even if the result of your test case is a program crash or otherwise obvious it might not happen on our platform. The easiest thing is to copy the output from the terminal, if possible.

**Note**

If you are reporting an error message, please obtain the most verbose form of the message. In `psql`, say `\set VERBOSITY verbose` beforehand. If you are extracting the message from the server log, set the run-time parameter `log_error_verbosity` (see Section "18.7. Error Reporting and Logging" - Vol.II) to `verbose` so that all details are logged.

**Note**

In case of fatal errors, the error message reported by the client might not contain all the information available. Please also look at the log output of the database server. If you do not keep your server's log output, this would be a good time to start doing so.

- The output you expected is very important to state. If you just write "This command gives me that output." or "This is not what I expected.", we might run it ourselves, scan the output, and think it looks OK and is exactly what we expected. We should not have to spend the time to decode the exact semantics behind your commands. Especially refrain from merely saying that "This is not what SQL says/Oracle does." Digging out the correct behavior from SQL is not a fun undertaking, nor do we all know how all the other relational databases out there behave. (If your problem is a program crash, you can obviously omit this item.)
- Any command line options and other start-up options, including any relevant environment variables or configuration files that you changed from the default. Again, please provide exact information. If you are using a prepackaged distribution that starts the database server at boot time, you should try to find out how that is done.
- Anything you did at all differently from the installation instructions.
- The PostgreSQL version. You can run the command `SELECT version();` to find out the version of the server you are connected to. Most executable programs also

## Preface

---

support a `--version` option; at least `postgres --version` and `psql --version` should work. If the function or the options do not exist then your version is more than old enough to warrant an upgrade. If you run a prepackaged version, such as RPMs, say so, including any subversion the package might have. If you are talking about a Git snapshot, mention that, including the commit hash.

If your version is older than 9.0.3 we will almost certainly tell you to upgrade. There are many bug fixes and improvements in each new release, so it is quite possible that a bug you have encountered in an older release of PostgreSQL has already been fixed. We can only provide limited support for sites using older releases of PostgreSQL; if you require more than we can provide, consider acquiring a commercial support contract.

- Platform information. This includes the kernel name and version, C library, processor, memory information, and so on. In most cases it is sufficient to report the vendor and version, but do not assume everyone knows what exactly "Debian" contains or that everyone runs on i386s. If you have installation problems then information about the toolchain on your machine (compiler, make, and so on) is also necessary.

Do not be afraid if your bug report becomes rather lengthy. That is a fact of life. It is better to report everything the first time than us having to squeeze the facts out of you. On the other hand, if your input files are huge, it is fair to ask first whether somebody is interested in looking into it. Here is an *article*<sup>15</sup> that outlines some more tips on reporting bugs.

Do not spend all your time to figure out which changes in the input make the problem go away. This will probably not help solving it. If it turns out that the bug cannot be fixed right away, you will still have time to find and share your work-around. Also, once again, do not waste your time guessing why the bug exists. We will find that out soon enough.

When writing a bug report, please avoid confusing terminology. The software package in total is called "PostgreSQL", sometimes "Postgres" for short. If you are specifically talking about the backend server, mention that, do not just say "PostgreSQL crashes". A crash of a single backend server process is quite different from crash of the parent "postgres" process; please don't say "the server crashed" when you mean a single backend process went down, nor vice versa. Also, client programs such as the interactive frontend "psql" are completely separate from the backend. Please try to be specific about whether the problem is on the client or server side.

---

<sup>15</sup> <http://www.chiark.greenend.org.uk/~sgtatham/bugs.html>

## Where to report bugs

In general, send bug reports to the bug report mailing list at `<pgsql-bugs@postgresql.org>`. You are requested to use a descriptive subject for your email message, perhaps parts of the error message.

Another method is to fill in the bug report web-form available at the project's *web site*<sup>16</sup>. Entering a bug report this way causes it to be mailed to the `<pgsql-bugs@postgresql.org>` mailing list.

If your bug report has security implications and you'd prefer that it not become immediately visible in public archives, don't send it to `pgsql-bugs`. Security issues can be reported privately to `<security@postgresql.org>`.

Do not send bug reports to any of the user mailing lists, such as `<pgsql-sql@postgresql.org>` or `<pgsql-general@postgresql.org>`. These mailing lists are for answering user questions, and their subscribers normally do not wish to receive bug reports. More importantly, they are unlikely to fix them.

Also, please do *not* send reports to the developers' mailing list `<pgsql-hackers@postgresql.org>`. This list is for discussing the development of PostgreSQL, and it would be nice if we could keep the bug reports separate. We might choose to take up a discussion about your bug report on `pgsql-hackers`, if the problem needs more review.

If you have a problem with the documentation, the best place to report it is the documentation mailing list `<pgsql-docs@postgresql.org>`. Please be specific about what part of the documentation you are unhappy with.

If your bug is a portability problem on a non-supported platform, send mail to `<pgsql-hackers@postgresql.org>`, so we (and you) can work on porting PostgreSQL to your platform.



### Note

Due to the unfortunate amount of spam going around, all of the above email addresses are closed mailing lists. That is, you need to be subscribed to a list to be allowed to post on it. (You need not be subscribed to use the bug-report web form, however.) If you would like to send mail but do not want to receive list traffic, you can subscribe and set your subscription option to `nomail`. For more information send mail to `<majordomo@postgresql.org>` with the single word `help` in the body of the message.

---

<sup>16</sup> <http://www.postgresql.org/>

# Part I.

# Tutorial

Welcome to the PostgreSQL Tutorial. The following few chapters are intended to give a simple introduction to PostgreSQL, relational database concepts, and the SQL language to those who are new to any one of these aspects. We only assume some general knowledge about how to use computers. No particular Unix or programming experience is required. This part is mainly intended to give you some hands-on experience with important aspects of the PostgreSQL system. It makes no attempt to be a complete or thorough treatment of the topics it covers.

After you have worked through this tutorial you might want to move on to reading *Part II* (page 55) to gain a more formal knowledge of the SQL language, or *Part IV* (Vol.II) for information about developing applications for PostgreSQL. Those who set up and manage their own server should also read *Part III* (Vol.II).

# Chapter 1.

## Getting Started

### 1.1. Installation

Before you can use PostgreSQL you need to install it, of course. It is possible that PostgreSQL is already installed at your site, either because it was included in your operating system distribution or because the system administrator already installed it. If that is the case, you should obtain information from the operating system documentation or your system administrator about how to access PostgreSQL.

If you are not sure whether PostgreSQL is already available or whether you can use it for your experimentation then you can install it yourself. Doing so is not hard and it can be a good exercise. PostgreSQL can be installed by any unprivileged user; no superuser (root) access is required.

If you are installing PostgreSQL yourself, then refer to *Chapter 15 - Vol.II* for instructions on installation, and return to this guide when the installation is complete. Be sure to follow closely the section about setting up the appropriate environment variables.

If your site administrator has not set things up in the default way, you might have some more work to do. For example, if the database server machine is a remote machine, you will need to set the `PGHOST` environment variable to the name of the database server machine. The environment variable `PGPORT` might also have to be set. The bottom line is this: if you try to start an application program and it complains that it cannot connect to the database, you should consult your site administrator or, if that is you, the documentation to make sure that your environment is properly set up. If you did not understand the preceding paragraph then read the next section.

### 1.2. Architectural Fundamentals

Before we proceed, you should understand the basic PostgreSQL system architecture. Understanding how the parts of PostgreSQL interact will make this chapter somewhat clearer.

In database jargon, PostgreSQL uses a client/server model. A PostgreSQL session consists of the following cooperating processes (programs):

- A server process, which manages the database files, accepts connections to the database from client applications, and performs database actions on behalf of the clients. The database server program is called `postgres`.
- The user's client (frontend) application that wants to perform database operations. Client applications can be very diverse in nature: a client could be a text-oriented tool, a graphical application, a web server that accesses the database to display web pages, or a specialized database maintenance tool. Some client applications are supplied with the PostgreSQL distribution; most are developed by users.

As is typical of client/server applications, the client and the server can be on different hosts. In that case they communicate over a TCP/IP network connection. You should keep this in mind, because the files that can be accessed on a client machine might not be accessible (or might only be accessible using a different file name) on the database server machine.

The PostgreSQL server can handle multiple concurrent connections from clients. To achieve this it starts ("forks") a new process for each connection. From that point on, the client and the new server process communicate without intervention by the original `postgres` process. Thus, the master server process is always running, waiting for client connections, whereas client and associated server processes come and go. (All of this is of course invisible to the user. We only mention it here for completeness.)

### 1.3. Creating a Database

The first test to see whether you can access the database server is to try to create a database. A running PostgreSQL server can manage many databases. Typically, a separate database is used for each project or for each user.

Possibly, your site administrator has already created a database for your use. He should have told you what the name of your database is. In that case you can omit this step and skip ahead to the next section.

To create a new database, in this example named `mydb`, you use the following command:

```
$ createdb mydb
```

If this produces no response then this step was successful and you can skip over the remainder of this section.

If you see a message similar to:

```
createdb: command not found
```

then PostgreSQL was not installed properly. Either it was not installed at all or your shell's search path was not set to include it. Try calling the command with an absolute path instead:

```
$ /usr/local/pgsql/bin/createdb mydb
```

The path at your site might be different. Contact your site administrator or check the installation instructions to correct the situation.

Another response could be this:

```
createdb: could not connect to database postgres: could not connect to server:
No such file or directory
Is the server running locally and accepting
connections on Unix domain socket "/tmp/.s.PGSQL.5432"?
```

This means that the server was not started, or it was not started where `createdb` expected it. Again, check the installation instructions or consult the administrator.

Another response could be this:

```
createdb: could not connect to database postgres: FATAL:  role "joe" does not exist
```

where your own login name is mentioned. This will happen if the administrator has not created a PostgreSQL user account for you. (PostgreSQL user accounts are distinct from operating system user accounts.) If you are the administrator, see *Chapter 20 - Vol.II* for help creating accounts. You will need to become the operating system user under which PostgreSQL was installed (usually `postgres`) to create the first user account. It could also be that you were assigned a PostgreSQL user name that is different from your operating system user name; in that case you need to use the `-U` switch or set the `PGUSER` environment variable to specify your PostgreSQL user name.

If you have a user account but it does not have the privileges required to create a database, you will see the following:

```
createdb: database creation failed: ERROR:  permission denied to create database
```

Not every user has authorization to create new databases. If PostgreSQL refuses to create databases for you then the site administrator needs to grant you permission to create databases. Consult your site administrator if this occurs. If you installed PostgreSQL yourself then you should log in for the purposes of this tutorial under the user account that you started the server as<sup>1</sup>.

You can also create databases with other names. PostgreSQL allows you to create any number of databases at a given site. Database names must have an alphabetic first character

---

<sup>1</sup> As an explanation for why this works: PostgreSQL user names are separate from operating system user accounts. When you connect to a database, you can choose what PostgreSQL user name to connect as; if you don't, it will default to the same name as your current operating system account. As it happens, there will always be a PostgreSQL user account that has the same name as the operating system user that started the server, and it also happens that that user always has permission to create databases. Instead of logging in as that user you can also specify the `-U` option everywhere to select a PostgreSQL user name to connect as.

and are limited to 63 characters in length. A convenient choice is to create a database with the same name as your current user name. Many tools assume that database name as the default, so it can save you some typing. To create that database, simply type:

```
$ createdb
```

If you do not want to use your database anymore you can remove it. For example, if you are the owner (creator) of the database `mydb`, you can destroy it using the following command:

```
$ dropdb mydb
```

(For this command, the database name does not default to the user account name. You always need to specify it.) This action physically removes all files associated with the database and cannot be undone, so this should only be done with a great deal of forethought.

More about `createdb` and `dropdb` can be found in *createdb* (see "*createdb command*" section) and *dropdb* (see "*dropdb command*" section) respectively.

## 1.4. Accessing a Database

Once you have created a database, you can access it by:

- Running the PostgreSQL interactive terminal program, called *psql*, which allows you to interactively enter, edit, and execute SQL commands.
- Using an existing graphical frontend tool like pgAdmin or an office suite with ODBC or JDBC support to create and manipulate a database. These possibilities are not covered in this tutorial.
- Writing a custom application, using one of the several available language bindings. These possibilities are discussed further in *Part IV* (Vol.II).

You probably want to start up `psql` to try the examples in this tutorial. It can be activated for the `mydb` database by typing the command:

```
$ psql mydb
```

If you do not supply the database name then it will default to your user account name. You already discovered this scheme in the previous section using `createdb`.

In `psql`, you will be greeted with the following message:

```
psql (9.0.3)
Type "help" for help.
```

```
mydb=>
```

The last line could also be:

```
mydb=#
```

That would mean you are a database superuser, which is most likely the case if you installed PostgreSQL yourself. Being a superuser means that you are not subject to access controls. For the purposes of this tutorial that is not important.

If you encounter problems starting `psql` then go back to the previous section. The diagnostics of `createdb` and `psql` are similar, and if the former worked the latter should work as well.

The last line printed out by `psql` is the prompt, and it indicates that `psql` is listening to you and that you can type SQL queries into a work space maintained by `psql`. Try out these commands:

```
mydb=> SELECT version();
                version
-----
 PostgreSQL 9.0.3 on i586-pc-linux-gnu, compiled by GCC 2.96, 32-bit
(1 row)

mydb=> SELECT current_date;
      date
-----
 2002-08-31
(1 row)

mydb=> SELECT 2 + 2;
?column?
-----
         4
(1 row)
```

The `psql` program has a number of internal commands that are not SQL commands. They begin with the backslash character, "\". For example, you can get help on the syntax of various PostgreSQL SQL commands by typing:

```
mydb=> \h
```

To get out of `psql`, type:

```
mydb=> \q
```

and `psql` will quit and return you to your command shell. (For more internal commands, type `\?` at the `psql` prompt.) The full capabilities of `psql` are documented in `psql`. If PostgreSQL is installed correctly you can also type `man psql` at the operating system shell prompt to see the documentation. In this tutorial we will not use these features explicitly, but you can use them yourself when it is helpful.

# List of Volumes

## **Volume I.**

### **Preface**

- What is PostgreSQL?
- A Brief History of PostgreSQL
- Conventions
- Further Information
- Bug Reporting Guidelines

### **Part I. Tutorial**

1. Getting Started
2. The SQL Language
3. Advanced Features

### **Part II. The SQL Language**

4. SQL Syntax
5. Data Definition
6. Data Manipulation
7. Queries
8. Data Types
9. Functions and Operators
10. Type Conversion
11. Indexes
12. Full Text Search
13. Concurrency Control
14. Performance Tips

## **Volume II.**

### **Part III. Server Administration**

15. Installation from Source Code
16. Installation from Source Code on Windows
17. Server Setup and Operation

18. Server Configuration
19. Client Authentication
20. Database Roles and Privileges
21. Managing Databases
22. Localization
23. Routine Database Maintenance Tasks
24. Backup and Restore
25. High Availability, Load Balancing, and Replication
26. Recovery Configuration
27. Monitoring Database Activity
28. Monitoring Disk Usage
29. Reliability and the Write-Ahead Log
30. Regression Tests

#### **Part IV. Client Interfaces**

31. libpq - C Library
32. Large Objects
33. ECPG - Embedded SQL in C
34. The Information Schema

### **Volume III.**

#### **Part V. Server Programming**

35. Extending SQL
36. Triggers
37. The Rule System
38. Procedural Languages
39. PL/pgSQL - SQL Procedural Language
40. PL/Tcl - Tcl Procedural Language
41. PL/Perl - Perl Procedural Language
42. PL/Python - Python Procedural Language
43. Server Programming Interface

### **Volume IV.**

#### **Part VI. Reference**

- I. SQL Commands
- II. PostgreSQL Client Applications
- III. PostgreSQL Server Applications

**Volume V.**

**Part VII. Internals**

- 44. Overview of PostgreSQL Internals
- 45. System Catalogs
- 46. Frontend/Backend Protocol
- 47. PostgreSQL Coding Conventions
- 48. Native Language Support
- 49. Writing A Procedural Language Handler
- 50. Genetic Query Optimizer
- 51. Index Access Method Interface Definition
- 52. GiST Indexes
- 53. GIN Indexes
- 54. Database Physical Storage
- 55. BKI Backend Interface
- 56. How the Planner Uses Statistics

**Part VIII. Appendixes**

- A. PostgreSQL Error Codes
- B. Date/Time Support
- C. SQL Key Words
- D. SQL Conformance
- E. Release Notes
- F. Additional Supplied Modules
- G. External Projects
- H. The CVS Repository
- I. Documentation
- J. Acronyms

**Bibliography**

**Index**

# Index

\$ .....	68	array_agg .....	274
* .....	133	array_append .....	272
:: .....	74	array_cat.....	272
abs .....	200	array_dims.....	272
acos .....	200	array_fill.....	272
age.....	238	array_length .....	272
aggregate function.....	42	array_lower.....	272
built-in .....	274	array_ndims.....	272
invocation.....	70	array_prepend.....	272
alias		array_to_string.....	272
for table name in query .....	39	array_upper .....	272
in the FROM clause.....	126	ascii .....	203
in the select list .....	133	asin.....	200
ALL .....	279, 282	AT TIME ZONE .....	246
AND (operator).....	197	atan .....	200
any.....	195, 274, 279, 282	atan2 .....	200
anyarray .....	195	autocommit	
anyelement.....	195	bulk-loading data .....	396
anyenum .....	195	auto-increment .....	147
anynonarray .....	195	average .....	42, 274
arbitrary precision numbers.....	144	backslash escapes.....	59
area.....	250	backup .....	298
ARRAY.....	75, 180	BETWEEN.....	198
accessing.....	182	BETWEEN SYMMETRIC.....	198
constant.....	181	bigint.....	63, 144
constructor .....	75	bigserial.....	147
declaration.....	180	binary data .....	151
determination of result type .....	315	functions .....	212
I/O.....	187	binary string	
modifying.....	184	concatenation.....	212
searching.....	187	length .....	212

## Index

bit string		clock_timestamp .....	238
constant.....	62	CLUSTER	
data type .....	173	of databases	
bit strings		See database cluster.....	35
functions .....	214	cmax.....	93
bit_and.....	274	cmin.....	93
bit_length.....	203	COALESCE.....	271
bit_or.....	274	col_description.....	288
bitmap scan.....	322	column.....	35, 82
bool_and.....	274	adding.....	95
bool_or.....	274	removing.....	96
Boolean		renaming.....	97
data type.....	165	system column.....	93
operators		column data type	
See operators, logical.....	197	changing.....	97
box (data type).....	170	column reference.....	68
B-tree.....	317, 335, 336, 397	COMMENT	
btrim.....	203	about database objects.....	288
bytea.....	151	in SQL.....	65
CASCADE		common table expression	
foreign key action.....	90	See WITH.....	138
with DROP.....	117	comparison	
CASE.....	269	operators.....	198
determination of result type.....	315	row-wise.....	282
case sensitivity		subquery result row.....	279
of SQL commands.....	57	composite type.....	189, 191, 192
cbrt.....	200	constant.....	190
ceiling.....	200	constructor.....	77
center.....	250	concurrency.....	373
char_length.....	203	conditional expression.....	269
character string		configuration	
concatenation.....	203	of the server	
constant.....	59	functions.....	298
length.....	203	conjunction.....	197
check constraint.....	85	constant.....	58
chr.....	203	constraint.....	85, 397
cid.....	193	adding.....	96
cidr.....	172, 173	check.....	85
circle.....	171	exclusion.....	93

foreign key .....	90	enumerated (enum) .....	167
name .....	85	numeric .....	143
NOT NULL .....	87	type cast .....	74
primary key .....	89	database	
removing .....	96	creating .....	31
unique .....	88	database cluster .....	35
constraint exclusion .....	113	date .....	154, 156
convert .....	203	constants .....	159
convert_from .....	203	current .....	247
convert_to .....	203	output format .....	159
COPY .....	37	See also formatting .....	231
correlation .....	274	date_part .....	238, 241
cos .....	200	date_trunc .....	238, 245
cot .....	200	deadlock .....	382
count .....	42	decimal	
covariance		See numeric .....	63
population .....	274	decode .....	203
sample .....	274	default value .....	84
CREATE DATABASE .....	31	changing .....	97
CREATE TABLE .....	36	degrees .....	200
createdb .....	31	delay .....	249
cross join .....	122	dense_rank .....	277
cstring .....	195	diameter .....	250
ctid .....	93	dirty read .....	373
cume_dist .....	277	disjunction .....	197
current_catalog .....	288	DISTINCT .....	38, 134
current_database .....	288	div .....	200
current_date .....	238	document	
current_schema .....	288	text search .....	332
current_time .....	238	dollar quoting .....	61
current_timestamp .....	238	double precision .....	146
current_user .....	288	DROP TABLE .....	36
currval .....	267	DTD .....	178
data area		duplicate .....	38
See database cluster .....	35	duplicates .....	134
data type .....	142	encode .....	203
category .....	108, 121, 306	enumerated types .....	167
constant .....	64	escape string syntax .....	59
conversion .....	306	every .....	274

## Index

EXCEPT .....	134	GRANT .....	98
exclusion constraint .....	93	GREATEST .....	272
EXISTS .....	279	determination of result type .....	315
exp .....	200	GROUP BY .....	42, 130
expression		grouping .....	130
order of evaluation .....	78	GUID .....	177
syntax .....	67	has_any_column_privilege .....	288
extract .....	238, 241	has_column_privilege .....	288
FALSE .....	165	has_database_privilege .....	288
field selection .....	69	has_foreign_data_wrapper_privilege .....	288
first_value .....	277	has_function_privilege .....	288
float4		has_language_privilege .....	288
See real .....	146	has_schema_privilege .....	288
float8		has_sequence_privilege .....	288
See double precision .....	146	has_server_privilege .....	288
floating point .....	146	has_table_privilege .....	288
floor .....	200	has_tablespace_privilege .....	288
foreign key .....	45, 90	hash	
format_type .....	288	See index .....	317, 335, 336, 397
formatting .....	231	HAVING .....	42, 130
full text search .....	331	height .....	250
data types .....	174	hierarchical database .....	35
functions and operators .....	174	history	
function .....	197, 253, 255	of PostgreSQL .....	20, 401
in the FROM clause .....	127	identifier	
invocation .....	70	length .....	57
mixed notation .....	80	syntax of .....	57
named notation .....	80	IFNULL .....	271
positional notation .....	80	IN .....	279, 282
type resolution in an		index .....	317, 335, 336, 397
invocation .....	249, 258, 311	and ORDER BY .....	321
generate_series .....	285	B-tree .....	318
generate_subscripts .....	285	combining multiple indexes .....	322
get_bit .....	212	examining usage .....	329
get_byte .....	212	GIN .....	318
GIN		text search .....	367
See index .....	317, 335, 336, 397	GiST .....	318
GiST		text search .....	367
See index .....	317, 335, 336, 397	hash .....	318

locks.....	385	ISNULL .....	198
multicolumn.....	320	isopen .....	250
on expressions .....	324	join .....	39, 122
partial.....	325	controlling the order .....	394
unique .....	323	cross.....	122
inet (data type) .....	171	left.....	122
inet_client_addr .....	288	natural.....	122
inet_client_port .....	288	outer .....	39, 122
inet_server_addr .....	288	right.....	122
inet_server_port .....	288	self .....	39
inheritance .....	52, 104	justify_days.....	238
initcap .....	203	justify_hours.....	238
INSERT .....	37, 118	justify_interval .....	238
inserting .....	118	key word	
int2		syntax of .....	57
See smallint .....	144	label	
int4		See alias.....	39, 126, 133
See integer .....	63, 144	lag.....	277
int8		language_handler .....	195
See bigint .....	63, 144	last_value .....	277
integer.....	63, 144	lastval .....	267
internal .....	195	lead.....	277
INTERSECT .....	134	LEAST.....	272
interval .....	154, 162	determination of result type .....	315
output format.....	165	left join.....	122
See also formatting.....	231	length.....	250
IS DISTINCT FROM.....	198, 282	of a binary string	
IS DOCUMENT.....	262	See binary strings, length .....	212
IS FALSE .....	198	of a character string	
IS NOT DISTINCT FROM.....	198, 282	See character string, length .....	203
IS NOT FALSE .....	198	length(tsvector) .....	344
IS NOT NULL .....	198	LIKE.....	215
IS NOT TRUE .....	198	LIMIT.....	136
IS NOT UNKNOWN.....	198	line segment.....	169
IS NULL .....	198	linear regression.....	274
IS TRUE .....	198	ln .....	200
IS UNKNOWN.....	198	localtime.....	238
isclosed .....	250	localtimestamp .....	238
isfinite .....	238		

## Index

lock.....	378	nth_value.....	277
advisory .....	383	ntile .....	277
log.....	200	null value	
lower .....	203	comparing.....	198
lpad .....	203	default value .....	84
lseg .....	169	in DISTINCT .....	134
ltrim.....	203	with check constraints .....	85
MAC address		with unique constraints.....	88
See macaddr .....	173	NULLIF .....	271
macaddr (data type) .....	173	number	
max.....	42	constant.....	63
md5 .....	203	numeric.....	63
min .....	42	numeric (data type) .....	144
mod .....	200	numnode .....	345
MVCC .....	35, 45, 317, 331, 373	NVL.....	271
name		obj_description .....	288
qualified.....	99	object identifier	
syntax of.....	57	data type .....	193
unqualified .....	101	object-oriented database .....	35
NaN		octet_length.....	203
See not a number .....	144, 146	OFFSET.....	136
natural join.....	122	oid.....	193
negation.....	197	column .....	93
network		ONLY.....	122
data types.....	171	opaque .....	195
nextval .....	267	operator .....	197, 253, 255
non-durable.....	400	invocation .....	70
nonrepeatable read .....	373	logical .....	197
NOT (operator).....	197	precedence.....	66
not a number		syntax .....	64
double precision .....	146	type resolution in an invocation.....	308
numeric (data type) .....	144	operator class.....	328
NOT IN.....	279, 282	operator family.....	328
notation		OR (operator).....	197
functions .....	79	ORDER BY .....	38, 135
NOTNULL .....	198	outer join .....	122
not-null constraint.....	87	OVER clause .....	72
now.....	238	overlay .....	203
npoints .....	250	owner .....	98

parameter		pg_indexes_size .....	298
syntax.....	68	pg_is_in_recovery.....	298
parenthesis.....	67	pg_is_other_temp_schema .....	288
partitioning.....	108, 109	pg_last_xlog_receive_location .....	298
path (data type).....	170	pg_last_xlog_replay_location .....	298
pattern matching.....	215	pg_listening_channels.....	288
pclose.....	250	pg_ls_dir .....	298
percent_rank.....	277	pg_my_temp_schema .....	288
performance.....	386	pg_opclass_is_visible .....	288
permission		pg_operator_is_visible.....	288
See privilege.....	98	pg_postmaster_start_time .....	288
pg_advisory_lock.....	298	pg_read_file.....	298
pg_advisory_lock_shared.....	298	pg_relation_filinode .....	298
pg_advisory_unlock.....	298	pg_relation_filepath .....	298
pg_advisory_unlock_all.....	298	pg_relation_size .....	298
pg_advisory_unlock_shared .....	298	pg_reload_conf.....	298
pg_cancel_backend.....	298	pg_rotate_logfile.....	298
pg_client_encoding.....	203	pg_size_pretty .....	298
pg_column_size .....	298	pg_sleep .....	249
pg_conf_load_time .....	288	pg_start_backup.....	298
pg_conversion_is_visible.....	288	pg_stat_file.....	298
pg_current_xlog_insert_location.....	298	pg_statistic .....	392
pg_current_xlog_location.....	298	pg_stats .....	392
pg_database_size .....	298	pg_stop_backup.....	298
pg_function_is_visible .....	288	pg_switch_xlog .....	298
pg_get_constraintdef.....	288	pg_table_is_visible.....	288
pg_get_expr .....	288	pg_table_size .....	298
pg_get_function_arguments .....	288	pg_tablespace_databases.....	288
pg_get_function_identity_arguments... 288		pg_tablespace_size .....	298
pg_get_function_result .....	288	pg_terminate_backend.....	298
pg_get_functiondef.....	288	pg_total_relation_size .....	298
pg_get_indexdef.....	288	pg_try_advisory_lock.....	298
pg_get_keywords.....	288	pg_try_advisory_lock_shared.....	298
pg_get_ruledef .....	288	pg_ts_config_is_visible .....	288
pg_get_serial_sequence .....	288	pg_ts_dict_is_visible .....	288
pg_get_triggerdef .....	288	pg_ts_parser_is_visible .....	288
pg_get_userbyid .....	288	pg_ts_template_is_visible.....	288
pg_get_viewdef.....	288	pg_type_is_visible .....	288
pg_has_role.....	288	pg_typeof .....	288

## Index

pg_xlogfile_name.....	298	real.....	146
pg_xlogfile_name_offset.....	298	record.....	195
phantom read .....	373	rectangle .....	170
pi.....	200	referential integrity .....	45, 90
plainto_tsquery.....	339	regclass .....	193
plperl.on_init configuration		regconfig.....	193
parameter.....	358	regdictionary.....	193
plperl.on_plperl_init configuration		regexp_matches.....	217
parameter.....	358	regexp_replace.....	217
plperl.on_plperlu_init configuration		regexp_split_to_array.....	217
parameter.....	358	regexp_split_to_table.....	217
plperl.use_strict configuration		regoper.....	193
parameter.....	358	regoperator.....	193
point.....	169	regproc.....	193
polygon.....	170	regprocedure .....	193
popen .....	250	regression intercept .....	274
position.....	203	regression slope.....	274
POSTGRES .....	20, 30, 31	regtype.....	193
Postgres95 .....	21	regular expression.....	216, 217
power.....	200	See also pattern matching .....	215
predicate locking.....	377	relation.....	35
primary key.....	89	relational database .....	35
privilege.....	98	repeat .....	203
for schemas.....	102	replace.....	203
querying.....	288	RESTRICT	
psql.....	33	foreign key action .....	90
qualified name.....	99	with DROP .....	117
query .....	38, 121	REVOKE.....	98
querytree .....	345	right join .....	122
quotation marks		round .....	200
and identifiers.....	57	row .....	35, 77, 82
escaping .....	59	row type .....	189, 191, 192
quote_ident.....	203	constructor.....	77
quote_literal.....	203	row_number .....	277
quote_nullable.....	203	row-wise comparison.....	282
radians .....	200	rpad.....	203
radius .....	250	rtrim .....	203
random .....	200	scalar	
rank .....	277	See expression.....	67, 78

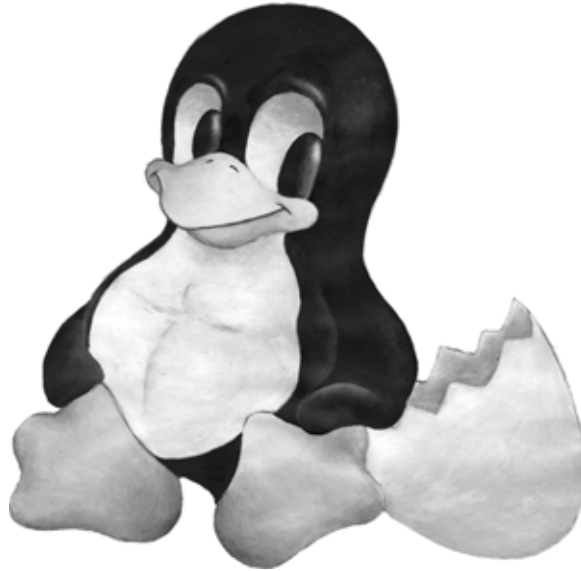
schema.....	99, 108, 121, 306	split_part.....	203
creating.....	99	sqrt.....	200
current.....	101, 288	standard deviation.....	274
public.....	100	population.....	274
removing.....	99	sample.....	274
search path.....	101	statement_timestamp.....	238
current.....	288	statistics.....	274
search_path.....	101	of the planner.....	392
SELECT.....	38, 121	string	
select list.....	132	See character string.....	59, 203
sequence.....	267	string_agg.....	274
and serial type.....	147	string_to_array.....	272
serial.....	147	strip.....	344
serial4.....	147	strpos.....	203
serial8.....	147	subquery.....	42, 75, 127, 279
serializability.....	377	subscript.....	69
SET.....	298	substr.....	203
set difference.....	134	substring.....	203, 212, 216, 217
set intersection.....	134	sum.....	42
set operation.....	134	superuser.....	33
set returning functions		suppress_redundant_updates_trigger..	305
functions.....	285	syntax	
set union.....	134	SQL.....	56
set_bit.....	212	system catalog	
set_byte.....	212	schema.....	102
setseed.....	200	table.....	35, 82
setval.....	267	creating.....	82
setweight.....	344	inheritance.....	104
shobj_description.....	288	modifying.....	95
SHOW.....	298	partitioning.....	108, 109
sign.....	200	removing.....	82
signal		renaming.....	98
backend processes.....	298	table expression.....	122
SIMILAR TO.....	216	table function.....	127
sin.....	200	tableoid.....	93
sleep.....	249	tan.....	200
smallint.....	144	template1.....	31
SOME.....	274, 279, 282	text search.....	331
sorting.....	135	data types.....	174



## Index

functions and operators.....	174	trim.....	203
indexes .....	367	TRUE.....	165
tid .....	193	trunc .....	200
time .....	154, 157	ts_debug.....	363
constants .....	159	ts_headline .....	343
current.....	247	ts_lexize .....	366
output format.....	159	ts_parse.....	365
See also formatting .....	231	ts_rank .....	340
time span.....	154	ts_rank_cd.....	340
time with time zone .....	154, 157	ts_rewrite.....	346
time without time zone .....	154, 157	ts_stat .....	349
time zone.....	160	ts_token_type.....	365
conversion .....	246	tsquery (data type).....	176
timeofday .....	238	tsvector (data type).....	174
timestamp.....	154, 158	tsvector concatenation.....	344
timestamp with time zone .....	154, 158	txid_current.....	288
timestamp without time zone .....	154, 158	txid_current_snapshot.....	288
to_ascii .....	203	txid_snapshot_xip.....	288
to_char .....	231	txid_snapshot_xmax.....	288
to_date .....	231	txid_snapshot_xmin .....	288
to_hex.....	203	txid_visible_in_snapshot .....	288
to_number.....	231	type	
to_timestamp .....	231	See data type .....	142
to_tsquery.....	339	type cast.....	63, 74
to_tsvector.....	338	UESCAPE.....	57, 60
TOAST		unaccent.....	249, 258, 311
versus large objects .....	35, 45, 317, 331, 373	Unicode escape	
token .....	56	in identifiers .....	57
transaction.....	46	in string constants .....	60
transaction isolation.....	373	UNION .....	134
transaction isolation level .....	373	determination of result type .....	315
read committed.....	374	unique constraint .....	88
serializable.....	376	unnest .....	272
transaction_timestamp .....	238	unqualified name.....	101
translate .....	203	UPDATE.....	44, 119
trigger .....	195	updating.....	119
for updating a derived tsvector column		upper.....	203
.....	348	user	
in PL/Python .....	305	current.....	288

UUID .....	177	WITH	
value expression .....	67	in SELECT .....	138
VALUES .....	137	xid .....	193
determination of result type .....	315	xmax .....	93
variance .....	274	xmin .....	93
population .....	274	XML .....	177
sample .....	274	XML export .....	263
version .....	33, 288	XML option .....	178
view .....	45	xmlagg .....	262, 274
void .....	195	xmlcomment .....	258
WHERE .....	128	xmlconcat .....	259
width .....	250	xmlelement .....	259
width_bucket .....	200	xmlforest .....	260
window function .....	49	xmlparse .....	178
built-in .....	277	xmlpi .....	261
invocation .....	72	xmlroot .....	261
order of execution .....	132	xmlserialize .....	178
		XPath .....	263

# Linbrary™ Advertising Club (LAC)



**Linbrary™**  Official Docs as a Real Books <http://www.linbrary.com>  **Linux Library**



Linbrary Advertising Club



**Linux Documentation Project - Machtelt Garrels**

*<http://www.tldp.org/>*

Version	Title	Edition	ISBN- 10	ISBN- 13
TLDP	Introduction to Linux (Third Edition)	paperback	1-59682-199-X	978-1-59682-199-6
		eBook (pdf)	1-59682-200-7	978-1-59682-200-9
	Bash Guide for Beginners (Second Edition)	paperback	1-59682-201-5	978-1-59682-201-6
		eBook (pdf)	1-59682-202-3	978-1-59682-202-3
<i><a href="http://www.linbrary.com/linux-tldp/">http://www.linbrary.com/linux-tldp/</a></i>				



Linbrary Advertising Club



**Fedora Project Official Documentation**

*<http://docs.fedoraproject.org>*

Version	Title	Edition	ISBN- 10	ISBN- 13	
Fedora 14	Fedora 14 <b>Installation Guide</b>	paperback	1-59682-228-7	978-1-59682-228-3	
		eBook (pdf)	1-59682-233-3	978-1-59682-233-7	
	Fedora 14 <b>User Guide</b>	paperback	1-59682-229-5	978-1-59682-229-0	
		eBook (pdf)	1-59682-234-1	978-1-59682-234-4	
	Fedora 14 <b>Security Guide</b>	paperback	1-59682-230-9	978-1-59682-230-6	
		eBook (pdf)	1-59682-235-X	978-1-59682-235-1	
	Fedora 14 <b>Storage Administration Guide</b>	paperback	1-59682-231-7	978-1-59682-231-3	
		eBook (pdf)	1-59682-236-8	978-1-59682-236-8	
	Fedora 14 <b>Musicians Guide</b>	paperback	1-59682-232-5	978-1-59682-232-0	
		eBook (pdf)	1-59682-237-6	978-1-59682-237-5	
	Fedora 13	Fedora 13 <b>Installation Guide</b>	paperback	1-59682-212-0	978-1-59682-212-2
			eBook (pdf)	1-59682-217-1	978-1-59682-217-7
Fedora 13 <b>User Guide</b>		paperback	1-59682-213-9	978-1-59682-213-9	
		eBook (pdf)	1-59682-218-X	978-1-59682-218-4	
Fedora 13 <b>Security Guide</b>		paperback	1-59682-214-7	978-1-59682-214-6	
		eBook (pdf)	1-59682-219-8	978-1-59682-219-1	
Fedora 13 <b>SE Linux User Guide</b>		paperback	1-59682-215-5	978-1-59682-215-3	
		eBook (pdf)	1-59682-220-1	978-1-59682-220-7	
Fedora 13 <b>Virtualization Guide</b>		paperback	1-59682-216-3	978-1-59682-216-0	
		eBook (pdf)	1-59682-221-X	978-1-59682-221-4	



Linbrary Advertising Club



**Fedora Project Official Documentation**

*<http://docs.fedoraproject.org>*

Version	Title	Edition	ISBN- 10	ISBN- 13
Fedora 12	Fedora 12 <b>Installation Guide</b>	paperback	1-59682-179-5	978-1-59682-179-8
		eBook (pdf)	1-59682-184-1	978-1-59682-184-2
	Fedora 12 <b>User Guide</b>	paperback	1-59682-180-9	978-1-59682-180-4
		eBook (pdf)	1-59682-185-X	978-1-59682-185-9
	Fedora 12 <b>Security Guide</b>	paperback	1-59682-181-7	978-1-59682-181-1
		eBook (pdf)	1-59682-186-8	978-1-59682-186-6
	Fedora 12 <b>SE Linux User Guide</b>	paperback	1-59682-182-5	978-1-59682-182-8
		eBook (pdf)	1-59682-187-6	978-1-59682-187-3
Fedora 12 <b>Virtualization Guide</b>	paperback	1-59682-183-3	978-1-59682-183-5	
	eBook (pdf)	1-59682-188-4	978-1-59682-188-0	
Fedora 11	Fedora 11 <b>Installation Guide</b>	paperback	1-59682-142-6	978-1-59682-142-2
		eBook (pdf)	1-59682-146-9	978-1-59682-146-0
	Fedora 11 <b>User Guide</b>	paperback	1-59682-143-4	978-1-59682-143-9
		eBook (pdf)	1-59682-147-7	978-1-59682-147-7
	Fedora 11 <b>Security Guide</b>	paperback	1-59682-144-2	978-1-59682-144-6
		eBook (pdf)	1-59682-148-5	978-1-59682-148-4
	Fedora 11 <b>SE Linux User Guide</b>	paperback	1-59682-145-0	978-1-59682-145-3
		eBook (pdf)	1-59682-149-3	978-1-59682-149-1
<i><a href="http://www.linbrary.com/fedora/">http://www.linbrary.com/fedora/</a></i>				



Linbrary Advertising Club



**Ubuntu Official Documentation**

*<http://www.ubuntu.com/>*

Version	Title	Edition	ISBN- 10	ISBN- 13
<b>Ubuntu 10.10</b>	Ubuntu 10.10 <b>Installation Guide</b>	paperback	1-59682-238-4	978-1-59682-238-2
		eBook (pdf)	1-59682-242-2	978-1-59682-242-9
	Ubuntu 10.10 <b>Desktop Guide</b>	paperback	1-59682-239-2	978-1-59682-239-9
		eBook (pdf)	1-59682-243-0	978-1-59682-243-6
	Ubuntu 10.10 <b>Server Guide</b>	paperback	1-59682-240-6	978-1-59682-240-5
		eBook (pdf)	1-59682-244-9	978-1-59682-244-3
Ubuntu 10.10 <b>Packaging Guide</b>	paperback	1-59682-241-4	978-1-59682-241-2	
	eBook (pdf)	1-59682-245-7	978-1-59682-245-0	
<b>Ubuntu 10.04 LTS</b>	Ubuntu 10.04 LTS <b>Installation Guide</b>	paperback	1-59682-203-1	978-1-59682-203-0
		eBook (pdf)	1-59682-207-4	978-1-59682-207-8
	Ubuntu 10.04 LTS <b>Desktop Guide</b>	paperback	1-59682-204-X	978-1-59682-204-7
		eBook (pdf)	1-59682-208-2	978-1-59682-208-5
	Ubuntu 10.04 LTS <b>Server Guide</b>	paperback	1-59682-205-8	978-1-59682-205-4
		eBook (pdf)	1-59682-209-0	978-1-59682-209-2
	Ubuntu 10.04 LTS <b>Packaging Guide</b>	paperback	1-59682-206-6	978-1-59682-206-1
		eBook (pdf)	1-59682-210-4	978-1-59682-210-8



Linbrary Advertising Club



**Ubuntu Official Documentation**

*<http://www.ubuntu.com/>*

Version	Title	Edition	ISBN- 10	ISBN- 13	
Ubuntu 9.10	Ubuntu 9.10 <b>Installation Guide</b>	paperback	1-59682-171-X	978-1-59682-171-2	
		eBook (pdf)	1-59682-175-2	978-1-59682-175-0	
	Ubuntu 9.10 <b>Desktop Guide</b>	paperback	1-59682-172-8	978-1-59682-172-9	
		eBook (pdf)	1-59682-176-0	978-1-59682-176-7	
	Ubuntu 9.10 <b>Server Guide</b>	paperback	1-59682-173-6	978-1-59682-173-6	
		eBook (pdf)	1-59682-177-9	978-1-59682-177-4	
	Ubuntu 9.10 <b>Packaging Guide</b>	paperback	1-59682-174-4	978-1-59682-174-3	
		eBook (pdf)	1-59682-178-7	978-1-59682-178-1	
	Ubuntu 9.04	Ubuntu 9.04 <b>Installation Guide</b>	paperback	1-59682-150-7	978-1-59682-150-7
			eBook (pdf)	1-59682-154-X	978-1-59682-154-5
Ubuntu 9.04 <b>Desktop Guide</b>		paperback	1-59682-151-5	978-1-59682-151-4	
		eBook (pdf)	1-59682-155-8	978-1-59682-155-2	
Ubuntu 9.04 <b>Server Guide</b>		paperback	1-59682-152-3	978-1-59682-152-1	
		eBook (pdf)	1-59682-156-6	978-1-59682-156-9	
Ubuntu 9.04 <b>Packaging Guide</b>		paperback	1-59682-153-1	978-1-59682-153-8	
		eBook (pdf)	1-59682-157-4	978-1-59682-157-6	
<i><a href="http://www.linbrary.com/ubuntu/">http://www.linbrary.com/ubuntu/</a></i>					



Linbrary Advertising Club



**PostgreSQL Official Documentation**

<http://www.postgresql.org/>

Version	Title	Edition	ISBN- 10	ISBN- 13
<b>PostgreSQL 9.0</b>	PostgreSQL 9.0 <b>Volume I. The SQL Language</b>	paperback	1-59682-246-5	978-1-59682-246-7
		eBook (pdf)	1-59682-251-1	978-1-59682-251-1
	PostgreSQL 9.0 <b>Volume II. Server Administration</b>	paperback	1-59682-247-3	978-1-59682-247-4
		eBook (pdf)	1-59682-252-X	978-1-59682-252-8
	PostgreSQL 9.0 <b>Volume III. Server Programming</b>	paperback	1-59682-248-1	978-1-59682-248-1
		eBook (pdf)	1-59682-253-8	978-1-59682-253-5
<b>PostgreSQL 8.04</b>	PostgreSQL 8.04 <b>Volume I. The SQL Language</b>	paperback	1-59682-158-2	978-1-59682-158-3
		eBook (pdf)	1-59682-163-9	978-1-59682-163-7
	PostgreSQL 8.04 <b>Volume II. Server Administration</b>	paperback	1-59682-159-0	978-1-59682-159-0
		eBook (pdf)	1-59682-164-7	978-1-59682-164-4
	PostgreSQL 8.04 <b>Volume III. Server Programming</b>	paperback	1-59682-160-4	978-1-59682-160-6
		eBook (pdf)	1-59682-165-5	978-1-59682-165-1
	PostgreSQL 8.04 <b>Volume IV. Reference</b>	paperback	1-59682-161-2	978-1-59682-161-3
		eBook (pdf)	1-59682-166-3	978-1-59682-166-8
	PostgreSQL 8.04 <b>Volume V. Internals &amp; Appendixes</b>	paperback	1-59682-162-0	978-1-59682-162-0
	eBook (pdf)	1-59682-167-1	978-1-59682-167-5	
<a href="http://www.linbrary.com/postgresql/">http://www.linbrary.com/postgresql/</a>				




Linbrary Advertising Club



**The Apache Software Foundation Official Documentation**

***<http://www.apache.org/>***

Version	Title	Edition	ISBN- 10	ISBN- 13
<b>Apache Web Server 2.2</b>	Apache HTTP Server 2.2 <b>Vol.I. Server Administration</b>	paperback	1-59682-191-4	978-1-59682-191-0
		eBook (pdf)	1-59682-195-7	978-1-59682-195-8
	Apache HTTP Server 2.2 <b>Vol.II. Security &amp; Server Programs</b>	paperback	1-59682-192-2	978-1-59682-192-7
		eBook (pdf)	1-59682-196-5	978-1-59682-196-5
	Apache HTTP Server 2.2 <b>Vol.III. Modules (A-H)</b>	paperback	1-59682-193-0	978-1-59682-193-4
		eBook (pdf)	1-59682-197-3	978-1-59682-197-2
	Apache HTTP Server 2.2 <b>Vol.IV. Modules (I-V)</b>	paperback	1-59682-194-9	978-1-59682-194-1
		eBook (pdf)	1-59682-198-1	978-1-59682-198-9
<i><a href="http://www.linbrary.com/apache-http/">http://www.linbrary.com/apache-http/</a></i>				

Version	Title	Edition	ISBN- 10	ISBN- 13
<b>Subversion 1.6</b>	Subversion 1.6 <b>Version Control with Subversion</b>	paperback	1-59682-169-8	978-1-59682-169-9
		eBook (pdf)	1-59682-170-1	978-1-59682-170-5
				
<i><a href="http://www.linbrary.com/subversion/">http://www.linbrary.com/subversion/</a></i>				

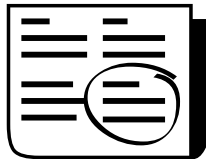


Linbrary Advertising Club

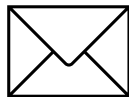


Linbrary Advertising Club

# Your Advertising Here



**More Books Coming Soon!!!**



Please Feel Free to Contact Us at

*[production@fultus.com](mailto:production@fultus.com)*