

## 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 SQL92 and SQL99 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 8.4

## Reference

## Volume IV



By The PostgreSQL Global Development Group

PostgreSQL 8.4 • Reference





**PostgreSQL 8.4**  
**Official Documentation**

**Reference**

**Volume IV**



*Fultus™ Books*

# PostgreSQL



## **PostgreSQL 8.4 Official Documentation**

### **Reference**

### **Volume IV**

ISBN 1-59682-161-2

Copyright © 1996-2009 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

License.....	8
Abstract .....	9
Part VI Reference .....	10
I SQL Commands.....	11
ABORT.....	11
ALTER AGGREGATE .....	12
ALTER CONVERSION .....	13
ALTER DATABASE.....	14
ALTER DOMAIN.....	16
ALTER FOREIGN DATA WRAPPER.....	19
ALTER FUNCTION.....	20
ALTER GROUP .....	23
ALTER INDEX.....	24
ALTER LANGUAGE.....	26
ALTER OPERATOR .....	27
ALTER OPERATOR CLASS.....	28
ALTER OPERATOR FAMILY .....	29
ALTER ROLE.....	32
ALTER SCHEMA.....	35
ALTER SEQUENCE.....	36
ALTER SERVER .....	39
ALTER TABLE.....	40
ALTER TABLESPACE.....	50
ALTER TEXT SEARCH CONFIGURATION .....	51
ALTER TEXT SEARCH DICTIONARY .....	52
ALTER TEXT SEARCH PARSER.....	54
ALTER TEXT SEARCH TEMPLATE .....	54
ALTER TRIGGER.....	55
ALTER TYPE .....	56
ALTER USER.....	57
ALTER USER MAPPING.....	58
ALTER VIEW.....	59

## Volume IV

---

ANALYZE.....	60
BEGIN.....	62
CHECKPOINT .....	64
CLOSE.....	64
CLUSTER .....	66
COMMENT.....	68
COMMIT .....	71
COMMIT PREPARED .....	72
COPY .....	73
CREATE AGGREGATE .....	82
CREATE CAST .....	85
CREATE CONSTRAINT TRIGGER .....	90
CREATE CONVERSION .....	91
CREATE DATABASE.....	93
CREATE DOMAIN.....	96
CREATE FOREIGN DATA WRAPPER.....	98
CREATE FUNCTION.....	100
CREATE GROUP .....	108
CREATE INDEX.....	108
CREATE LANGUAGE.....	115
CREATE OPERATOR.....	118
CREATE OPERATOR CLASS.....	120
CREATE OPERATOR FAMILY .....	124
CREATE ROLE.....	125
CREATE RULE.....	129
CREATE SCHEMA .....	132
CREATE SEQUENCE.....	134
CREATE SERVER .....	138
CREATE TABLE.....	139
CREATE TABLE AS .....	153
CREATE TABLESPACE.....	156
CREATE TEXT SEARCH CONFIGURATION .....	157
CREATE TEXT SEARCH DICTIONARY .....	158
CREATE TEXT SEARCH PARSER.....	160
CREATE TEXT SEARCH TEMPLATE.....	161
CREATE TRIGGER.....	162
CREATE TYPE.....	165
CREATE USER .....	174
CREATE USER MAPPING.....	175

## Table of Contents

---

CREATE VIEW .....	176
DEALLOCATE .....	178
DECLARE .....	179
DELETE .....	183
DISCARD .....	186
DROP AGGREGATE.....	187
DROP CAST.....	188
DROP CONVERSION .....	189
DROP DATABASE .....	190
DROP DOMAIN .....	191
DROP FOREIGN DATA WRAPPER.....	192
DROP FUNCTION .....	193
DROP GROUP.....	194
DROP INDEX .....	194
DROP LANGUAGE.....	195
DROP OPERATOR .....	196
DROP OPERATOR CLASS.....	197
DROP OPERATOR FAMILY.....	199
DROP OWNED .....	200
DROP ROLE .....	201
DROP RULE.....	202
DROP SCHEMA.....	203
DROP SEQUENCE .....	204
DROP SERVER.....	205
DROP TABLE .....	205
DROP TABLESPACE .....	207
DROP TEXT SEARCH CONFIGURATION.....	208
DROP TEXT SEARCH DICTIONARY.....	209
DROP TEXT SEARCH PARSER .....	210
DROP TEXT SEARCH TEMPLATE .....	211
DROP TRIGGER.....	212
DROP TYPE .....	213
DROP USER.....	214
DROP USER MAPPING .....	214
DROP VIEW.....	215
END .....	216
EXECUTE .....	217
EXPLAIN.....	218
FETCH .....	221

---

## Volume IV

---

GRANT .....	225
INSERT .....	232
LISTEN .....	235
LOAD.....	236
LOCK.....	237
MOVE .....	240
NOTIFY .....	241
PREPARE .....	243
PREPARE TRANSACTION.....	245
REASSIGN OWNED .....	247
REINDEX .....	248
RELEASE SAVEPOINT.....	251
RESET .....	252
REVOKE.....	253
ROLLBACK.....	256
ROLLBACK PREPARED .....	257
ROLLBACK TO SAVEPOINT.....	258
SAVEPOINT .....	260
SELECT.....	261
SELECT INTO.....	279
SET .....	281
SET CONSTRAINTS.....	284
SET ROLE.....	285
SET SESSION AUTHORIZATION .....	287
SET TRANSACTION.....	288
SHOW .....	290
START TRANSACTION .....	292
TRUNCATE .....	293
UNLISTEN.....	295
UPDATE.....	296
VACUUM.....	300
VALUES .....	303
<b>II. PostgreSQL Client Applications .....</b>	<b>306</b>
clusterdb.....	306
createdb .....	308
createlang .....	311
createuser .....	313
dropdb .....	317
droplang.....	319

## Table of Contents

---

dropuser .....	322
ecpg .....	324
pg_config.....	326
pg_dump.....	328
pg_dumpall.....	338
pg_restore.....	343
psql .....	351
reindexdb .....	381
vacuumdb .....	384
<b>III. PostgreSQL Server Applications .....</b>	<b>388</b>
initdb .....	388
pg_controldata.....	391
pg_ctl .....	392
pg_resetxlog.....	396
postgres .....	398
postmaster.....	406
<b>List of Volumes .....</b>	<b>407</b>
<b>Index.....</b>	<b>410</b>

# License

## **PostgreSQL is released under the BSD license.**

PostgreSQL Database Management System  
(formerly known as Postgres, then as Postgres95)

Portions Copyright (c) 1996-2009, The PostgreSQL Global Development Group

Portions Copyright (c) 1994, 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 8.4 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.

## Part VI.

# Reference

The entries in this Reference are meant to provide in reasonable length an authoritative, complete, and formal summary about their respective subjects. More information about the use of PostgreSQL, in narrative, tutorial, or example form, can be found in other parts of this book. See the cross-references listed on each reference page.

The reference entries are also available as traditional "man" pages.

# I. SQL Commands

This part contains reference information for the SQL commands supported by PostgreSQL. By "SQL" the language in general is meant; information about the standards conformance and compatibility of each command can be found on the respective reference page.

---

## **ABORT**

### **Name**

ABORT -- abort the current transaction

### **Synopsis**

```
ABORT [ WORK | TRANSACTION ]
```

### **Description**

ABORT rolls back the current transaction and causes all the updates made by the transaction to be discarded. This command is identical in behavior to the standard SQL command ROLLBACK, and is present only for historical reasons.

### **Parameters**

WORK  
TRANSACTION

Optional key words. They have no effect.

### **Notes**

Use COMMIT to successfully terminate a transaction.

Issuing ABORT when not inside a transaction does no harm, but it will provoke a warning message.

### **Examples**

To abort all changes:

```
ABORT;
```

---

## Compatibility

This command is a PostgreSQL extension present for historical reasons. `ROLLBACK` is the equivalent standard SQL command.

### See Also

BEGIN, COMMIT, ROLLBACK

---

## ALTER AGGREGATE

### Name

ALTER AGGREGATE -- change the definition of an aggregate function

### Synopsis

```
ALTER AGGREGATE name ( type [ , ... ] ) RENAME TO new_name
ALTER AGGREGATE name ( type [ , ... ] ) OWNER TO new_owner
ALTER AGGREGATE name ( type [ , ... ] ) SET SCHEMA new_schema
```

### Description

ALTER AGGREGATE changes the definition of an aggregate function.

You must own the aggregate function to use ALTER AGGREGATE. To change the schema of an aggregate function, you must also have CREATE privilege on the new schema. To alter the owner, you must also be a direct or indirect member of the new owning role, and that role must have CREATE privilege on the aggregate function's schema. (These restrictions enforce that altering the owner doesn't do anything you couldn't do by dropping and recreating the aggregate function. However, a superuser can alter ownership of any aggregate function anyway.)

### Parameters

*name*

The name (optionally schema-qualified) of an existing aggregate function.

*type*

An input data type on which the aggregate function operates. To reference a zero-argument aggregate function, write `*` in place of the list of input data types.

*new\_name*

The new name of the aggregate function.

*new\_owner*

The new owner of the aggregate function.

---

*new\_schema*

The new schema for the aggregate function.

### Examples

To rename the aggregate function `myavg` for type `integer` to `my_average` :

```
ALTER AGGREGATE myavg(integer) RENAME TO my_average;
```

To change the owner of the aggregate function `myavg` for type `integer` to `joe`:

```
ALTER AGGREGATE myavg(integer) OWNER TO joe;
```

To move the aggregate function `myavg` for type `integer` into schema `myschema`:

```
ALTER AGGREGATE myavg(integer) SET SCHEMA myschema;
```

### Compatibility

There is no `ALTER AGGREGATE` statement in the SQL standard.

### See Also

`CREATE AGGREGATE`, `DROP AGGREGATE`

---

## ALTER CONVERSION

### Name

`ALTER CONVERSION` -- change the definition of a conversion

### Synopsis

```
ALTER CONVERSION name RENAME TO newname
```

```
ALTER CONVERSION name OWNER TO newowner
```

### Description

`ALTER CONVERSION` changes the definition of a conversion.

You must own the conversion to use `ALTER CONVERSION`. To alter the owner, you must also be a direct or indirect member of the new owning role, and that role must have `CREATE` privilege on the conversion's schema. (These restrictions enforce that altering the owner doesn't do anything you couldn't do by dropping and recreating the conversion. However, a superuser can alter ownership of any conversion anyway.)

### Parameters

*name*

The name (optionally schema-qualified) of an existing conversion.

---

*newname*

The new name of the conversion.

*newowner*

The new owner of the conversion.

### Examples

To rename the conversion `iso_8859_1_to_utf8` to `latin1_to_unicode`:

```
ALTER CONVERSION iso_8859_1_to_utf8 RENAME TO latin1_to_unicode;
```

To change the owner of the conversion `iso_8859_1_to_utf8` to `joe`:

```
ALTER CONVERSION iso_8859_1_to_utf8 OWNER TO joe;
```

### Compatibility

There is no `ALTER CONVERSION` statement in the SQL standard.

### See Also

CREATE CONVERSION, DROP CONVERSION

---

## ALTER DATABASE

### Name

ALTER DATABASE -- change a database

### Synopsis

```
ALTER DATABASE name [ [ WITH ] option [ ... ] ]
```

where *option* can be:

```
CONNECTION LIMIT conlimit
```

```
ALTER DATABASE name RENAME TO newname
```

```
ALTER DATABASE name OWNER TO new_owner
```

```
ALTER DATABASE name SET TABLESPACE new_tablespace
```

```
ALTER DATABASE name SET configuration_parameter { TO | = } { value | DEFAULT }
```

```
ALTER DATABASE name SET configuration_parameter FROM CURRENT
```

```
ALTER DATABASE name RESET configuration_parameter
```

```
ALTER DATABASE name RESET ALL
```

**Description**

`ALTER DATABASE` changes the attributes of a database.

The first form changes certain per-database settings. (See below for details.) Only the database owner or a superuser can change these settings.

The second form changes the name of the database. Only the database owner or a superuser can rename a database; non-superuser owners must also have the `CREATEDB` privilege. The current database cannot be renamed. (Connect to a different database if you need to do that.)

The third form changes the owner of the database. To alter the owner, you must own the database and also be a direct or indirect member of the new owning role, and you must have the `CREATEDB` privilege. (Note that superusers have all these privileges automatically.)

The fourth form changes the default tablespace of the database. Only the database owner or a superuser can do this; you must also have create privilege for the new tablespace. This command physically moves any tables or indexes in the database's old default tablespace to the new tablespace. Note that tables and indexes in non-default tablespaces are not affected.

The remaining forms change the session default for a run-time configuration variable for a PostgreSQL database. Whenever a new session is subsequently started in that database, the specified value becomes the session default value. The database-specific default overrides whatever setting is present in `postgresql.conf` or has been received from the `postgres` command line. Only the database owner or a superuser can change the session defaults for a database. Certain variables cannot be set this way, or can only be set by a superuser.

**Parameters**

*name*

The name of the database whose attributes are to be altered.

*connlimit*

How many concurrent connections can be made to this database. -1 means no limit.

*newname*

The new name of the database.

*new\_owner*

The new owner of the database.

*new\_tablespace*

The new default tablespace of the database.

---

*configuration\_parameter*  
*value*

Set this database's session default for the specified configuration parameter to the given value. If *value* is `DEFAULT` or, equivalently, `RESET` is used, the database-specific setting is removed, so the system-wide default setting will be inherited in new sessions. Use `RESET ALL` to clear all database-specific settings. `SET FROM CURRENT` saves the session's current value of the parameter as the database-specific value.

See `SET` and *Chapter 18* (Vol.II - page 77) for more information about allowed parameter names and values.

### Notes

It is also possible to tie a session default to a specific role rather than to a database; see `ALTER ROLE`. Role-specific settings override database-specific ones if there is a conflict.

### Examples

To disable index scans by default in the database `test`:

```
ALTER DATABASE test SET enable_indexscan TO off;
```

### Compatibility

The `ALTER DATABASE` statement is a PostgreSQL extension.

### See Also

`CREATE DATABASE`, `DROP DATABASE`, `SET`, `CREATE TABLESPACE`

---

## ALTER DOMAIN

### Name

`ALTER DOMAIN` -- change the definition of a domain

### Synopsis

```
ALTER DOMAIN name
    { SET DEFAULT expression | DROP DEFAULT }
ALTER DOMAIN name
    { SET | DROP } NOT NULL
ALTER DOMAIN name
    ADD domain_constraint
ALTER DOMAIN name
    DROP CONSTRAINT constraint_name [ RESTRICT | CASCADE ]
```

# List of Volumes

## **Volume I. The SQL Language**

### **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. Server Administration**

### **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. Monitoring Database Activity
27. Monitoring Disk Usage
28. Reliability and the Write-Ahead Log
29. Regression Tests

**Part IV. Client Interfaces**

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

**Volume III. Server Programming****Part V. Server Programming**

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

**Volume IV. Reference****Part VI. Reference**

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

## **Volume V. Internals and Appendixes**

### **Part VII. Internals**

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

DEALLOCATE .....	178	DROP TYPE .....	213
DECLARE .....	179	DROP USER.....	214
DISCARD .....	186	DROP USER MAPPING .....	214
DROP AGGREGATE.....	187	DROP VIEW .....	215
DROP CAST.....	188	dropdb .....	317
DROP CONVERSION .....	189	droplang.....	319
DROP DATABASE .....	190	dropuser .....	322
DROP DOMAIN .....	191	ecpg.....	324
DROP FOREIGN DATA WRAPPER.....	192	END .....	216
DROP FUNCTION .....	193	EXECUTE.....	217
DROP GROUP.....	194	EXPLAIN.....	218
DROP INDEX .....	194	FETCH.....	221
DROP LANGUAGE .....	195	GRANT.....	225
DROP OPERATOR.....	196	index	
DROP OPERATOR CLASS.....	197	building concurrently.....	112
DROP OPERATOR FAMILY.....	199	initdb.....	388
DROP OWNED .....	200	INSERT .....	232
DROP ROLE .....	201	LISTEN .....	235
DROP RULE .....	202	LOAD.....	236
DROP SCHEMA.....	203	LOCK.....	237
DROP SEQUENCE .....	204	MOVE .....	240
DROP SERVER.....	205	NOTIFY .....	241
DROP TABLE .....	205	patterns	
DROP TABLESPACE .....	207	in psql and pg_dump .....	370
DROP TEXT SEARCH		pg_config.....	326
CONFIGURATION .....	208	pg_controldata .....	391
DROP TEXT SEARCH DICTIONARY ..	209	pg_ctl .....	392
DROP TEXT SEARCH PARSER .....	210	pg_dump.....	328
DROP TEXT SEARCH TEMPLATE .....	211	pg_dumpall.....	338
DROP TRIGGER .....	212	pg_restore.....	343

---

postgres .....	398	rolling back .....	258
postmaster .....	406	SELECT .....	261
PREPARE .....	243	SELECT INTO .....	279
PREPARE TRANSACTION .....	245	SET .....	281
prepared statements		SET CONSTRAINTS .....	284
creating .....	243	SET ROLE .....	285
executing .....	217	SET SESSION AUTHORIZATION .....	287
removing .....	178	SET TRANSACTION .....	288
showing the query plan .....	218	SHOW .....	290
REASSIGN OWNED .....	247	START TRANSACTION .....	292
REINDEX .....	248	storage parameters .....	147
reindexdb .....	381	TABLE command .....	261
RELEASE SAVEPOINT .....	251	TOAST	
RESET .....	252	per-column storage settings .....	41
REVOKE .....	253	TRUNCATE .....	293
ROLLBACK .....	256	UNLISTEN .....	295
psql .....	371	UPDATE .....	296
ROLLBACK PREPARED .....	257	VACUUM .....	300
ROLLBACK TO SAVEPOINT .....	258	vacuumdb .....	384
SAVEPOINT .....	260	VALUES .....	303
savepoints		WITH	
defining .....	260	in SELECT .....	261
releasing .....	251		