



PostgreSQL Overview

from www.postgresql.org/about

PostgreSQL is a powerful, open source 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, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), and Windows. It is fully ACID compliant, has full support for foreign keys, joins, views, triggers, and stored procedures (in multiple languages). It includes most SQL92 and SQL99 data types, including INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, and TIMESTAMP. It also supports storage of binary large objects, including pictures, sounds, or video. It has native programming interfaces for C/C++, Java, Perl, Python, Ruby, Tcl, ODBC, among others, and [exceptional documentation](#).

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. Some general PostgreSQL limits are included in the table below.

Limit	Value
Maximum Database Size	Unlimited
Maximum Table Size	32 TB
Maximum Row Size	1.6 TB
Maximum Field Size	1 GB
Maximum Rows per Table	Unlimited
Maximum Columns per Table	250 - 1600 depending on column types
Maximum Indexes per Table	Unlimited

PostgreSQL has won [praise from its users](#) and [industry recognition](#), including the Linux New Media Award for Best Database System and three time winner of the The Linux Journal Editors' Choice Award for best DBMS.

Featureful and Standards Compliant

PostgreSQL prides itself in standards compliance. Its SQL implementation strongly conforms to the ANSI-SQL 92/99 standards. It has full support for subqueries (including subselects in the FROM clause), read-committed and serializable transaction isolation levels. And while PostgreSQL has a fully relational system catalog which itself supports multiple schemas per database, its catalog is also accessible through the Information Schema as defined in the SQL standard.

Data integrity features include (compound) primary keys, foreign keys with restricting and cascading updates/deletes, check constraints, unique constraints, and not null constraints.

It also has a host of extensions and advanced features. Among the conveniences are auto-increment columns through sequences, and LIMIT/OFFSET allowing the return of partial result sets. PostgreSQL supports compound, unique, partial, and functional indexes which can use any of its B-tree, R-tree, hash, or GiST storage methods.

[GiST](#) (*Generalized Search Tree*) indexing is an advanced system which brings together a wide array of different sorting and searching algorithms including B-tree, B+-tree, R-tree, partial sum trees, ranked B+-trees and many others. It also provides an interface which allows both the creation of custom data types as well as extensible query methods with which to search them. Thus, GiST offers the flexibility to specify *what* you store, *how* you store it, and *the ability to define new ways* to search through it --- ways that far exceed those offered by standard B-tree, R-tree and other generalized search algorithms.

GiST serves as a foundation for many public projects that use PostgreSQL such as [OpenFTS](#) and [PostGIS](#). OpenFTS (Open Source Full Text Search engine) provides online indexing of data and relevance ranking for database searching. PostGIS is a project which adds support for geographic objects in PostgreSQL, allowing it to be used as a spatial database for geographic information systems (GIS), much like ESRI's SDE or Oracle's Spatial extension.

Other advanced features include table inheritance, a rules systems, and database events. Table inheritance puts an object oriented slant on table creation, allowing database designers to *derive* new tables from other tables, treating them as base classes. Even better, PostgreSQL supports both single and multiple inheritance in this manner.

The rules system, also called the *query rewrite system*, allows the database designer to create rules which identify specific operations for a given table or view, and dynamically transform them into alternate operations when they are processed.

The events system is an interprocess communication system in which messages and events can be transmitted between clients using the LISTEN and NOTIFY commands, allowing both simple peer to peer communication and advanced coordination on database events. Since notifications can be issued from triggers and stored procedures, PostgreSQL clients can monitor database events such as table updates, inserts, or deletes as they happen.

Highly Customizable

PostgreSQL runs stored procedures in more than a dozen programming languages, including Java, Perl, Python, Ruby, Tcl, C/C++, and its own PL/pgSQL, which is similar to Oracle's PL/SQL. Included with its standard function library are hundreds of built-in functions that range from basic math and string operations to cryptography and Oracle compatibility. Triggers and stored procedures can be written in C and loaded into the database as a library, allowing great flexibility in extending its capabilities. Similarly, PostgreSQL includes a framework that allows developers to define and create their own custom data types along with supporting functions and operators that define their behavior. As a result, a host of advanced data types have been created that range from geometric and spatial primitives to network addresses to even ISBN/ISSN (International Standard Book Number/International Standard Serial Number) data types, all of which can be optionally added to the system.

Just as there are many procedure languages supported by PostgreSQL, there are also many library interfaces as well, allowing various languages both compiled and interpreted to interface with Post-

greSQL. There are interfaces for Java (JDBC), ODBC, Perl, Python, Ruby, C, C++, PHP, Lisp, Scheme, and Qt just to name a few.

Best of all, PostgreSQL's source code is available under the most liberal open source license: the BSD license. This license gives you the freedom to use, modify and distribute PostgreSQL in any form you like, open or closed source. Any modifications, enhancements, or changes you make are yours to do with as you please. As such, PostgreSQL is not only a powerful database system capable of running the enterprise, it is a development platform upon which to develop in-house, web, or commercial software products that require a capable RDBMS.

Advantages

PostgreSQL offers many advantages for your company or business over other database systems.

Immunity to over-deployment

Over-deployment is what some proprietary database vendors regard as their #1 licence compliance problem. With PostgreSQL, no-one can sue you for breaking licensing agreements, as there is **no associated licensing cost for the software**.

This has several additional advantages:

- More profitable business models with wide-scale deployment.
- No possibility of being audited for license compliance at any stage.
- Flexibility to do concept research and trial deployments without needing to include additional licensing costs.

Better support than the proprietary vendors

In addition to our strong support offerings, we have a vibrant community of PostgreSQL professionals and enthusiasts that your staff can draw upon and contribute to.

Significant saving on staffing costs

Our software has been designed and created to have much lower maintenance and tuning requirements than the leading proprietary databases, yet still retain all of the features, stability, and performance.

In addition to this our training programs are generally regarded as being far more cost effective, manageable, and practical in the real world than that of the leading proprietary database vendors.

Legendary reliability and stability

Unlike many proprietary databases, it is extremely common for companies to report that PostgreSQL has never, ever crashed for them in several years of high activity operation. Not even once. It just works.

Extensible

The source code is available to all at no charge. If your staff have a need to customise or extend PostgreSQL in any way then they are able to do so with a minimum of effort, and with no attached

costs. This is complemented by the community of PostgreSQL professionals and enthusiasts around the globe that also actively extend PostgreSQL on a daily basis.

Cross platform

PostgreSQL is available for almost every brand of Unix (34 platforms with the latest stable release), and Windows compatibility is available via the Cygwin framework. Native Windows compatibility is also available with version 8.0 and above.

Designed for high volume environments

We use a multiple row data storage strategy called MVCC to make PostgreSQL extremely responsive in high volume environments. The leading proprietary database vendor uses this technology as well, for the same reasons.

GUI database design and administration tools

There are many high-quality GUI Tools available for PostgreSQL from both open source developers and commercial providers. A nice list is available from our community documentation system with [information about applications, licenses, and their features](#).

Technical Features

- Fully ACID compliant.
- ANSI SQL compliant.
- Referential Integrity.
- Replication (non-commercial and commercial solutions) allowing the duplication of the master database to multiple slave machines.
- Native interfaces for ODBC, JDBC, C, C++, PHP, Perl, TCL, ECPG, Python, and Ruby.
- Rules.
- Views.
- Triggers.
- Unicode.
- Sequences.
- Inheritance.
- Outer Joins.
- Sub-selects.
- An open API.
- Stored Procedures.
- Native SSL support.
- Procedural languages.
- Hot stand-by (commercial solutions).
- Better than row-level locking.
- Functional and Partial indexes.
- Native Kerberos authentication.
- Support for UNION, UNION ALL and EXCEPT queries.
- Loadable extensions offering SHA1, MD5, XML, and other functionality.
- Tools for generating portable SQL to share with other SQL-compliant systems.
- Extensible data type system providing for custom, user-defined datatypes and rapid development of new datatypes.

- Cross-database compatibility functions for easing the transition from other, less SQL-compliant RDBMS.

Featured Users

Here are a few of the hundreds of companies who have built products, solutions, web sites and tools using the world's most advanced open source database system.

BioPharm

- American Chemical Society
- BASF, Agricultural Product Division ([case study](#))
- Institut de Biologie et Chimie des Proteines ([article](#))

Telecom

- Cisco ([documentation](#))
- NTT Data
- Optus
- Skype
- Telstra

Education

- Sternberg Astronomical Institute, Russia
- University of Alabama, Birmingham ([documentation](#))
- University of California, Berkeley
- University of New South Wales
- University of Oslo, Norway
- University of Sydney

Government

- U.S. Agency for International Development
- U.S. Centers For Disease Control and Prevention
- U.S. Department of Labor
- U.S. General Services Administration
- U.S. State Department
- City of Garden Grove, California ([quote](#))
- National Physical Laboratory of India
- The National Weather Service
- United Nations Children's Fund

Technology

- Apple ([article](#))
- Fujitsu
- Sun Microsystems