

# **Guia del Administrador de PostgreSQL**

**El equipo de desarrollo de PostgreSQL**

**Editado por  
Thomas Lockhart**



**Guia del Administrador de PostgreSQL**  
por El equipo de desarrollo de PostgreSQL

Editado por Thomas Lockhart

PostgreSQL  
es marca registrada © 1996-9 por el Postgres Global Development Group.



# Tabla de contenidos

Sumario .....	??
<b>1. Introducción.....</b>	<b>??</b>
Recursos.....	??
11 .....	??
12 .....	??
Terminología .....	??
Notación .....	??
Y2K Statement (Informe sobre el efecto 2000).....	??
11 .....	??
Copyrights y Marcas Registradas .....	??
<b>2. Portes.....</b>	<b>??</b>
Plataformas actualmente soportadas .....	??
Plataformas no soportadas .....	??
<b>3. Opciones de Configuración.....</b>	<b>??</b>
Parámetros de configuración (configure).....	??
Parámetros de construcción (make) .....	??
Soporte Local .....	??
Cuales son los Beneficios? .....	??
Cuales son las Desventajas? .....	??
Autenticación Kerberos .....	??
Disponibilidad .....	??
Instalación .....	??
Operaciones .....	??
<b>4. Distribución del Sistema.....</b>	<b>??</b>
<b>5. Instalación .....</b>	<b>??</b>
Antes de comenzar.....	??
Procedimiento de Instalación .....	??
51 .....	??
<b>6. Instalacion en Win32.....</b>	<b>??</b>
Construccion de librerias .....	??
Instalacion de las librerias.....	??
Usando las librerias.....	??
<b>7. Entorno de tiempo de ejecución.....</b>	<b>??</b>
Utilizando Postgres desde Unix.....	??
Iniciando postmaster .....	??
Usando pg_options .....	??
Opciones reconocidas.....	??
<b>8. Seguridad .....</b>	<b>??</b>
Autentificacion de Usuarios .....	??
Nombres de usuario y grupos.....	??
Crear Usuarios .....	??
Crear Grupos .....	??
Asignar usuarios a los Grupos.....	??
Control de Acceso .....	??
Funciones y Reglas.....	??
Funciones .....	??
Reglas .....	??
Caveats .....	??
<b>9. Agregar y Eliminar Usuarios .....</b>	<b>??</b>
<b>10. Gestión de Disco .....</b>	<b>??</b>
Localizaciones Alternativas .....	??

<b>11. Gestión de una base de datos .....</b>	<b>??</b>
Creación de una base de datos .....	??
Acceso a la base de datos .....	??
Destrucción de una base de datos.....	??
Copia de seguridad y restauración.....	??
Bases de datos grandes .....	??
<b>12. Tratamiento de problemas.....</b>	<b>??</b>
Fallos de inicio de Postmaster .....	??
Problemas con la conexión del Cliente.....	??
Depuración de mensajes .....	??
pg_options .....	??
<b>13. Recuperación de bases de datos.....</b>	<b>??</b>
<b>14. Pruebas de regresión .....</b>	<b>??</b>
Entorno de regresión.....	??
Estructura de directorios .....	??
Procedimiento para el test de regresión.....	??
Análisis de Regresión .....	??
Diferencias en los mensajes de error.....	??
Diferencias en fechas y horas.....	??
Diferencias en punto flotante.....	??
Diferencias en polígonos .....	??
Diferencias aleatorias .....	??
Los archivos “expected” .....	??
Archivos de comparación específicos de la plataforma .....	??
<b>15. Notas de versiones .....</b>	<b>??</b>
Version 6.5.3 .....	??
Migracion a v6.5.3.....	??
Lista Detallada de Cambios .....	??
Version 6.5.2 .....	??
Migracion to v6.5.2.....	??
Lista Detallada de Cambios .....	??
Version 6.5.1 .....	??
Migracion to v6.5.1.....	??
Lista Detallada de Cambios .....	??
Version 6.5 .....	??
Migracion to v6.5.....	??
Lista Detalla de Cambios.....	??
Version 6.4.2 .....	??
Migracion a v6.4.2.....	??
Lista Detallada de Cambios .....	??
Version 6.4.1 .....	??
Migracion a v6.4.1.....	??
Lista Detallada de Cambios .....	??
Version 6.4 .....	??
Migracion a v6.4.....	??
Lista Detallada de Cambios .....	??
Version 6.3.2 .....	??
Lista Detallada de Cambios .....	??
Version 6.3.1 .....	??
Lista Detallada de Cambios .....	??
Version 6.3 .....	??
Migracion a v6.3.....	??
Lista Detallada de Cambios .....	??
Version 6.2.1 .....	??
Migracion desde v6.2 a v6.2.1 .....	??

Lista Detalladas de Cambios.....	??
Version 6.2 .....	??
Migracion desde v6.1 a v6.2 .....	??
Migracion desde v1.x a v6.2 .....	??
Lista Detallada de Cambios .....	??
Version 6.1.1 .....	??
Migración desde v6.1 a v6.1.1 .....	??
Lista Detallada de Cambios .....	??
Version 6.1 .....	??
Migración a v6.1.....	??
Lista Detallada de Cambios .....	??
Version v6.0 .....	??
Migración desde v1.09 a v6.0 .....	??
Migración desde versiones previas a v1.09 hasta v6.0.....	??
Lista Detallada de Cambios .....	??
Version v1.09 .....	??
Version v1.02 .....	??
Migración de v1.02 a v1.02.1 .....	??
Procedimiento de Volcado/Recarga Procedure .....	??
Lista Detallada de Cambios .....	??
Version v1.01 .....	??
Migración desde v1.0 a v1.01 .....	??
Lista Detallada de Cambios .....	??
Version v1.0 .....	??
Lista Detallada de Cambios .....	??
Postgres95 Beta 0.03 .....	??
Lista Detallada de Cambios .....	??
Postgres95 Beta 0.02 .....	??
Lista Detallada de Cambios .....	??
Postgres95 Beta 0.01 .....	??
Tiempos Resultantes .....	??
v6.5.....	??
v6.4beta .....	??
v6.3.....	??
v6.1.....	??
<b>Bibliografía .....</b>	<b>??</b>





## Summario

Postgres, desarrollada originalmente en el Departamento de Ciencias de la Computación de la Universidad de California en Berkeley, fué pionera en muchos de los conceptos de bases de datos relacionales orientadas a objetos que ahora empiezan a estar disponibles en algunas bases de datos comerciales. Ofrece soporte al language SQL92/SQL3, integridad de transacciones, y extensibilidad de tipos de datos. PostgreSQL es un descendiente de dominio público y código abierto del código original de Berkeley.



# Capítulo 1. Introducción

Este documento es el manual del programador para el gestor de bases de datos PostgreSQL<sup>1</sup>, desarrollado inicialmente en la Universidad de California en Berkeley. PostgreSQL se basa en Postgres versión 4.2<sup>2</sup>. El proyecto Postgres, liderado por el Profesor Michael Stonebraker, ha sido financiado por la Agencia de Proyectos de Investigación de Defensa Avanzados (DARPA), la Oficina de Investigación del Ejército (ARO), la Fundación Nacional de Ciencia (NSF), y ESL, Inc.

## Recursos

Este manual está organizado en diferentes partes:

### Tutorial

Introducción para nuevos usuarios. No cubre características avanzadas.

### Guía del Usuario

Información general para el usuario, incluye comandos y tipos de datos.

### Guía del Programador

Información avanzada para programadores de aplicaciones. Incluyendo tipos y extensión de funciones, librería de interfaces y lo referido al diseño de aplicaciones.

### Guía del Administrador

Información sobre instalación y administración. Lista de equipo soportado.

### Guía del Desarrollador

Información para desarrolladores de Postgres. Este documento es para aquellas personas que están contribuyendo al proyecto de Postgres; la información referida al desarrollo de aplicaciones aparece en la *Guía del Programador*. Actualmente incluido en la *Guía del Programador*.

### Manual de Referencia

Información detallada sobre los comandos. Actualmente incluido en la *Guía del Usuario*.

Además de este manual, hay otros recursos que le servirán de ayuda para la instalación y el uso de Postgres:

### man pages

Las páginas de manual(man pages) contienen más información sobre los comandos.

### FAQs(Preguntas Frecuentes)

La sección de Preguntas Frecuentes(FAQ) contiene respuestas a preguntas generales y otros asuntos que tienen que ver con la plataforma en que se desarrolle.

### LEAME(READMEs)

Los archivos llamados LEAME(README) están disponibles para algunas contribuciones.

### Web Site

El sitio web de Postgres<sup>3</sup> contiene información que algunas distribuciones no incluyen. Hay un catálogo llamado *mhonarc* que contiene el histórico de las listas de correo electrónico. Aquí podrá encontrar bastante información.

### Listas de Correo

La lista de correo *pgsql-general*<sup>4</sup> (*archive*<sup>5</sup>) es un buen lugar para contestar sus preguntas.

### Usted!

Postgres es un producto de código abierto. Como tal, depende de la comunidad de usuarios para su soporte. A medida que empiece a usar Postgres, empezará a depender de otros para que le ayuden, ya sea por medio de documentación o en las listas de correo. Considere contribuir lo que aprenda. Si aprende o descubre algo que no esté documentado, escríbalo y contribuya. Si añade nuevas características al código, hágalas saber.

Aun aquellos con poca o ninguna experiencia pueden proporcionar correcciones y cambios menores a la documentación, lo que es una buena forma de empezar. El *pgsql-docs*<sup>6</sup> (*archivo*<sup>7</sup>) de la lista de correos es un buen lugar para comenzar sus pesquisas.

## Terminología

En la documentación siguiente, *sitio* (o *site*) se puede interpretar como la máquina en la que está instalada Postgres. Dado que es posible instalar más de un conjunto de bases de datos Postgres en una misma máquina, este término denota, de forma más precisa, cualquier conjunto concreto de programas binarios y bases de datos de Postgres instalados.

El *superusuario* de Postgres es el usuario llamado *postgres* que es dueño de los ficheros de la bases de datos y binarios de Postgres. Como superusuario de la base de datos, no le es aplicable ninguno de los mecanismos de protección y puede acceder a cualquiera de los datos de forma arbitraria. Además, al superusuario de Postgres se le permite ejecutar programas de soporte que generalmente no están disponibles para todos los usuarios. Tenga en cuenta que el superusuario de Postgres *no* es el mismo que el superusuario de Unix (que es conocido como *root*). El superusuario debería tener un identificador de usuario (*UID*) distinto de cero por razones de seguridad.

El *administrador de la base de datos* (*database administrator*) o DBA, es la persona responsable de instalar Postgres con mecanismos para hacer cumplir una política de seguridad para un *site*. El DBA puede añadir nuevos usuarios por el método descrito más adelante y mantener un conjunto de bases de datos plantilla para usar *concreatedb*.

El *postmaster* es el proceso que actúa como una puerta de control (*clearing-house*) para las peticiones al sistema Postgres. Las aplicaciones frontend se conectan al *postmaster*, que mantiene registros de los errores del sistema y de la comunicación entre los procesos backend. El *postmaster* puede aceptar varios argumentos desde la línea de órdenes para poner a punto su comportamiento. Sin embargo, el proporcionar argumentos es necesario sólo si se intenta trabajar con varios sitios o con uno que no se ejecuta a la manera por defecto.

El backend de Postgres (el programa ejecutable *postgres real*) lo puede ejecutar el superusuario directamente desde el intérprete de órdenes de usuario de Postgres (con el nombre de la base de datos como un argumento). Sin embargo, hacer esto eli-

mina el buffer pool compartido y bloquea la tabla asociada con un postmaster/sitio, por ello esto no está recomendado en un sitio multiusuario.

## Notación

“...” o `/usr/local/pgsql/` delante de un nombre de fichero se usa para representar el camino (path) al directorio home del superusuario de Postgres.

En la sinopsis, los corchetes (“[” y “]”) indican una expresión o palabra clave opcional. Cualquier cosa entre llaves (“{” y “}”) y que contenga barras verticales (“|”) indica que debe elegir una de las opciones que separan las barras verticales.

En los ejemplos, los paréntesis (“(” y “)”) se usan para agrupar expresiones booleanas. “|” es el operador booleano OR.

Los ejemplos mostrarán órdenes ejecutadas desde varias cuentas y programas. Las órdenes ejecutadas desde la cuenta del root estarán precedidas por “>”. Las órdenes ejecutadas desde la cuenta del superusuario de Postgres estarán precedidas por “%”, mientras que las órdenes ejecutadas desde la cuenta de un usuario sin privilegios estarán precedidas por “\$”. Las órdenes de SQL estarán precedidas por “=>” o no estarán precedidas por ningún prompt, dependiendo del contexto.

**Nota:** En el momento de escribir (Postgres v6.5) la notación de las órdenes flagging (o flojos) no es universalmente estable o congruente en todo el conjunto de la documentación. Por favor, envíe los problemas a la Lista de Correo de la Documentación (o Documentation Mailing List)<sup>8</sup>.

## Y2K Statement (Informe sobre el efecto 2000)

**Autor:** Escrito por Thomas Lockhart<sup>9</sup> el 22-10-1998.

El Equipo de Desarrollo Global (o Global Development Team) de PostgreSQL proporciona el árbol de código de software de Postgres como un servicio público, sin garantía y sin responsabilidad por su comportamiento o rendimiento. Sin embargo, en el momento de la escritura:

- El autor de este texto, voluntario en el equipo de soporte de Postgres desde Noviembre de 1996, no tiene constancia de ningún problema en el código de Postgres relacionado con los cambios de fecha en torno al 1 de Enero de 2000 (Y2K).
- El autor de este informe no tiene constancia de la existencia de informes sobre el problema del efecto 2000 no cubiertos en las pruebas de regresión, o en otro campo de uso, sobre versiones de Postgres recientes o de la versión actual. Podríamos haber esperado oír algo sobre problemas si existiesen, dada la base que hay instalada y dada la participación activa de los usuarios en las listas de correo de soporte.
- Por lo que el autor sabe, las suposiciones que Postgres hace sobre las fechas que se escriben usando dos números para el año están documentadas en la Guía del Usuario<sup>10</sup> en el capítulo de los tipos de datos. Para años escritos con dos números, la transición significativa es 1970, no el año 2000; ej. “70-01-01” se interpreta como “1970-01-01”, mientras que “69-01-01” se interpreta como “2069-01-01”.

- Los problemas relativos al efecto 2000 en el SO (sistema operativo) sobre el que esté instalado Postgres relacionados con la obtención de "la fecha actual" se pueden propagar y llegar a parecer problemas sobre el efecto 2000 producidos por Postgres.

Diríjase a The Gnu Project<sup>11</sup> y a The Perl Institute<sup>12</sup> para leer una discusión más profunda sobre el asunto del efecto 2000, particularmente en lo que tiene que ver con el open source o código abierto, código por el que no hay que pagar.

## Copyrights y Marcas Registradas

*La traducción de los textos de copyright se presenta aquí únicamente a modo de aclaración y no ha sido aprobada por sus autores originales. Los únicos textos de copyright, garantías, derechos y demás legalismos que tienen validez son los originales en inglés o una traducción aprobada por los autores y/o sus representantes legales. .*

PostgreSQL tiene Copyright © 1996-2000 por PostgreSQL Inc. y se distribuye bajo los términos de la licencia de Berkeley.

Postgres95 tiene Copyright © 1994-5 por los Regentes de la Universidad de California. Se autoriza el uso, copia, modificación y distribución de este software y su documentación para cualquier propósito, sin ningún pago, y sin un acuerdo por escrito, siempre que se mantengan el copyright del párrafo anterior, este párrafo y los dos párrafos siguientes en todas las copias.

En ningún caso la Universidad de California se hará responsable de daños, causados a cualquier persona o entidad, sean estos directos, indirectos, especiales, accidentales o consiguientes, incluyendo lucro cesante que resulten del uso de este software y su documentación, incluso si la Universidad ha sido notificada de la posibilidad de tales daños.

La Universidad de California rehusa específicamente ofrecer cualquier garantía, incluyendo, pero no limitada únicamente a, la garantía implícita de comerciabilidad y capacidad para cumplir un determinado propósito. El software que se distribuye aquí se entrega "tal y cual", y la Universidad de California no tiene ninguna obligación de mantenimiento, apoyo, actualización, mejoramiento o modificación.

Unix es una marca registrada de X/Open, Ltd. Sun4, SPARC, SunOS y Solaris son marcas registradas de Sun Microsystems, Inc. DEC, DECstation, Alpha AXP y ULTRIX son marcas registradas de Digital Equipment Corp. PA-RISC y HP-UX son marcas registradas de Hewlett-Packard Co. OSF/1 es marca registrada de Open Software Foundation.

## Notas

1. <http://postgresql.org/>
2. <http://s2k-ftp.CS.Berkeley.EDU:8000/postgres/postgres.html>
3. [postgresql.org](http://postgresql.org)
4. <mailto:pgsql-general@postgresql.org>
5. <http://www.PostgreSQL.ORG/mhonarc/pgsql-general/>
6. <mailto:pgsql-docs@postgresql.org>
7. <http://www.PostgreSQL.ORG/mhonarc/pgsql-docs/>
8. <mailto:docs@postgresql.org>
9. <mailto:lockhart@alumni.caltech.edu>
10. <http://www.postgresql.org/docs/user/datatype.htm>

11. <http://www.gnu.org/software/year2000.html>
12. <http://language.perl.com/news/y2k.html>





## Capítulo 2. Portes

Este manual describe la versión 6.5 Postgres. La comunidad de desarrollo de Postgres ha compilado y probado Postgres en varias plataformas. Visita esta página web<sup>1</sup> para tener la última información.

### Plataformas actualmente soportadas

En el momento de esta publicación, las siguientes plataformas han sido probadas:

Tabla 2-1. plataformas soportadas

OS	Procesador	Versión	Enviado	Apuntes
AIX 4.3.2	RS6000	v6.5	1999-05-26	(Andreas Zeugswetter)
BSDI	x86	v6.5	1999-05-25	(Bruce Momjian)
FreeBSD 2.2.x-4.0	x86	v6.5	1999-05-25	(Tatsuo Ishii, Marc Fournier)
DGUX 5.4R4.11	m88k	v6.3	1998-03-01	v6.4 probablemente con soporte
Digital Unix 4.0	Alpha	v6.5.3	1999-11-04	(Pedro J. Lobo)
HPUX	PA-RISC	v6.4	1998-10-25	Both 9.0x and 10.20 (Tom Szybist)
IRIX 6.5.6f	MIPS	v6.5.3	2000-02-18	MIPSPPro 7.3.1.1m; full support
linux 2.0.x	Alpha	v6.5.3	1999-11-05	(Ryan Kirkpatrick)
linux 2.2.x	arm41	v6.5.3	1999-11-05	(Mark Knox)
linux 2.2.x/ glibc2	x86	v6.5.3	1999-11-05	(Lamar Owens)
linux 2.0.x	MIPS	v6.4	1998-12-16	Cobalt Qube (Tatsuo Ishii)
linux 2.0.x	Sparc	v6.4	1998-10-25	(Tom Szybist)
linuxPPC 2.1.24	PPC603e	v6.4	1998-10-26	Powerbook 2400c (Tatsuo Ishii)
mklinux DR3	PPC750	v6.4	1998-09-16	PowerMac 7600 (Tatsuo Ishii)
NetBSD	arm32	v6.5	1999-04-14	(Andrew McMurry)
NetBSD 1.3.2	x86	v6.4	1998-10-25	(Brook Milligan)
NetBSD	m68k	v6.4.2	1998-12-28	Mac SE/30 (Mr. Mutsaers)
NetBSD-current	NS32532	v6.4	1998-10-27	pequeños problemas con el compilador
NetBSD/sparc 1.3H	Sparc	v6.4	1998-10-27	(Tom I Helbekkmo)
NetBSD 1.3	VAX	v6.3	1998-03-01	(Tom I Helbekkmo)
QNX-4.25	x86	v6.5.2	1999-11-08	requiere pequeños parches
SCO OpenServer 5	x86	v6.5	1999-05-25	(Andrew Merrill)
SCO UnixWare 7	x86	v6.5	1999-05-25	(Andrew Merrill)
Solaris	x86	v6.4	1998-10-28	(Marc Fournier)
Solaris 2.6-2.7	Sparc	v6.4	1998-10-28	(Tom Szybist, Frank Fournier)
SunOS 4.1.4	Sparc	v6.3	1998-03-01	Patches submitted (Tatsuo Ishii)
SVR4	MIPS	v6.4	1998-10-28	Sin soporte para compilador
Windows	x86	v6.4	1999-01-06	Librerías del lado del cliente
Windows NT	x86	v6.5	1999-05-26	Trabajando con la librería

Plataformas listadas para v6.3.x y v6.4.x también trabajan con la v6.5, pero no hemos recibido confirmación explícita de la misma en el momento de la creación de la lista.

**Nota:** Para Windows NT, el porte de la parte del servidor Postgres se ha conseguido recientemente Postgres Se requiere de la librería Cygnus para compilarlo.

## Plataformas no soportadas

Hay pocas plataformas con las cuales se haya intentado y se haya informado que no trabaja con la distribución estándar. Otras listadas aquí no proveen de suficientes librerías para intentarlo.

**Tabla 2-2. Posiblemente Plataformas incompatibles**

OS	Procesador	Versión	Enviado	Apuntes
MacOS	all	v6.x	1998-03-01	Sin librerías compatib
NextStep	x86	v6.x	1998-03-01	Sólo soporte cliente; v
SVR4 4.4	m88k	v6.2.1	1998-03-01	Confirmado con parch

## Notas

1. <http://www.postgresql.org/docs/admin/ports.htm>

## Capítulo 3. Opciones de Configuración

### Parámetros de configuración (configure)

El conjunto de parámetros disponibles en configure se puede obtener escribiendo

```
$ ./configure -help
```

Los siguientes parámetros pueden ser de interés para los instaladores:

Nombre de directorios y ficheros:

-prefix=PREFIX	ficheros de instalación independiente de la arquitectura en PREFIX
-bindir=DIR	[/usr/local/pgsql] ejecutables de usuario en el DIR [EPREFIX/bin]
-libdir=DIR	librerías de código objeto en el DIR [EPREFIX/lib]
-includedir=DIR	ficheros de cabeceras C en el in DIR [PREFIX/include]
-mandir=DIR	documentación man en el DIR [PREFIX/man]

Características y paquetes:

-disable-FEATURE	no incluir la FEATURE (lo mismo que -enable-FEATURE=no)
-enable-FEATURE[=ARG]	incluir FEATURE [ARG=yes]
-with-PACKAGE[=ARG]	usar PACKAGE [ARG=yes]
-without-PACKAGE	no usar PACKAGE (lo mismo que -with-PACKAGE=no)

-enable y -with opciones reconocidas:

-with-template=template	usar el fichero plantilla del sistema operativo
-with-includes=incdir	ver directorio plantilla sitio donde están los ficheros cabecera para tk/tcl, etc. en el DIR
-with-libs=incdir	buscar librerías también en DIR
-with-libraries=libdir	buscar librerías también en DIR
-enable-locale	activa el soporte local
-enable-recode	activa el soporte de codificación cirílica
-with-mb=encoding	activa el soporte para multi-byte
-with-pgport=portnum	cambia el puerto de inicio por defecto
-with-maxbackends=n	define el número máximo por defecto de procesos servidores
-with-tcl	construye interfaces Tcl y pgtclsh
-with-tclconfig=tcldir	tclConfig.sh y tkConfig.sh están en DIR
-with-perl	construye interfaces con Perl
-with-odbc	construye el paquete del driver ODBC
-with-odbcinst=odbcdir	cambia el directorio por defecto de odbcinst.ini
-enable-cassert	activa los chequeos de afirmación (depurando)
-with-CC=compiler	usa el compilador de C especificado
-with-CXX=compiler	usa el compilador de C++ especificado
-without-CXX	previene la construcción de código C++

Algunos sistemas pueden tener problemas de construcción con algunas características específicas de Postgres. Por ejemplo, sistemas con el compilador de C++ dañado

pueden necesitar especificar `-without-cxx` para el proceso de construcción para saltarse la construcción de `libpq++`.

## Parámetros de construcción (make)

Muchos parámetros relacionados con la instalación pueden activar en la etapa de construcción de la instalación de Postgres.

En muchos casos, estos parámetros deben colocarse en un fichero, `Makefile.custom`, utilizado para este propósito. La distribución por defecto no contiene este fichero opcional, pero puedes crearlo con el editor de texto que tu elijas. Cuando actualizas una instalación, tu puedes simplemente copiar tu viejo `Makefile.custom` a la nueva instalación antes que hagas la construcción.

```
make [ variable=value [,...] ]
```

Unas pocas de las muchas variables que puedes especificar son:

### POSTGRES DIR

Lo más alto en el árbol de la instalación.

### BINDIR

Localización de las aplicaciones y utilidades.

### LIBDIR

localización de las librerías, incluyendo las librerías compartidas.

### HEADERDIR

Localización de los ficheros include.

### ODBCINST

localización de las librerías, incluyendo las librerías compartidas `psqlODBC` (ODBC)

.

Hay otros parámetros opcionales que no se utilizan comúnmente. Muchos de las que listan debajo son apropiadas cuando se estaba desarrollando el código del servidor Postgres.

### CFLAGS

Establece los flags para el compilador de C. Debe ser especificado con `"+="` para conservar los parámetros por defecto.

### YFLAGS

Establece los flags para el parser yacc/bison. Puede usarse `-v` para ayudar a diagnosticar problemas de construcción de un nuevo parser. Debe ser especificado con `"+="` para conservar los parámetros por defecto.

### USE\_TCL

Activa el constructor del interfaces Tcl.

**HSTYLE**

Páginas HTML estilo DocBook para construir la documentación de partida. No usar a menos que tu estés desarrollando nueva documentación de documentos fuente SGML compatibles con DocBook en `doc/src/sgml/`.

**PSTYLE**

Páginas estilo DocBook para construir la documentación impresa de partida. No usar a menos que tu estés desarrollando nueva documentación de documentos fuente SGML compatibles con DocBook en `doc/src/sgml/`.

Aquí hay un ejemplo de `Makefile.custom` para un sistema Linux PentiumPro:

```
# Makefile.custom
# Thomas Lockhart 1999-06-01

POSTGRES_DIR= /opt/postgres/current
CFLAGS+= -m486 -O2

# documentation

HSTYLE= /home/tgl/SGML/db118.d/docbook/html
PSTYLE= /home/tgl/SGML/db118.d/docbook/print
```

## Soporte Local

**Nota:** Escrito por Oleg Bartunov. Ver Oleg's web page<sup>1</sup> para más información sobre el soporte de lengua local y Rusa.

Mientras que estaba en un proyecto para una compañía en Moscú, Rusia, Me encontré con el problema que postgresql no tenía soporte para alfabetos nacionales. Después de mirar posibles soluciones alternativas decidí desarrollar un soporte local yo mismo. No soy un programador en C pero ya había tenido experiencia con la programación local cuando trabajo en perl (depurando) y glimpse. Después de bastantes días sumergido por el árbol de fuente de Postgres Realice muy pocas correcciones en `src/backend/utils/adt/varlena.c` and `src/backend/main/main.c` para conseguir lo que quería! Di soporte sólo para `LC_CTYPE` and `LC_COLLATE`, pero más tarde otros lo añadieron para `LC_MONETARY`. Tuve muchos mensajes de la gente a cerca de este parche por eso decidí enviarselo a los desarrolladores y (sorprendentemente) lo incorporaron dentro de la distribución Postgres.

La gente a veces se queja que el soporte local no funciona para ellos. Hay algunos errores comunes:

- No configurar debidamente postgresql antes de compilarlo. Tu debes ejecutar la configuración con la opción `--enable-locale` para activar el soporte local. No iniciar el entorno correctamente cuando se inicia postmaster. Tu debes definir las variables de entorno `LC_CTYPE` and `LC_COLLATE` antes de ejecutar postmaster porque por detrás coge información local del entorno. Yo uso el siguiente shell script (`runpostgres`):

```
#!/bin/sh

export LC_CTYPE=koi8-r
export LC_COLLATE=koi8-r
postmaster -B 1024 -S -D/usr/local/pgsql/data/ -o '-Fe'
```

y lo ejecuto en rc.local así

```
/bin/su - postgres -c "/home/postgres/runpostgres"
```

- Un soporte local estropeado en un OS (por ejemplo, el soporte local en libc bajo Linux algunas veces ha sido cambiado y esto ha causado muchos problemas) El más reciente perl tiene también soporte local y si el soporte local es defectuoso **perl -v** da un aviso parecido a esto:

```
8:17[mira]:~/WWW/postgres>setenv LC_CTYPE not_exist
8:18[mira]:~/WWW/postgres>perl -v
perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
LC_ALL = (unset),
  LC_CTYPE = "not_exist",
  LANG = (unset)
are supported and installed on your system.
perl: warning: Falling back to the standard locale ("C").
```

- Localización errónea de los ficheros locales! Las posibles localizaciones son: /usr/lib/locale (Linux, Solaris), /usr/share/locale (Linux), /usr/lib/nls/loc (DUX 4.0). Chequea **man locale** para encontrar la localización correcta. Bajo Linux yo hice un enlace simbólico entre /usr/lib/locale y /usr/share/locale para estar seguro que la próxima libc no estropea mi soporte local.

## Cuales son los Beneficios?

Tu puedes usar ~\* y el operador order by para cadenas que contienen caracteres de alfabetos nacionales. Los usuarios no Ingleses definitivamente lo necesitan. Si tu no quieres usar el soporte local libera la variable USE\_LOCALE.

## Cuales son las Desventajas?

Hay una evidente desventaja si utilizamos el soporte local - que el la velocidad! Por eso, utilízalo sólo si verdaderamente lo necesitas.

## Autenticación Kerberos

Kerberos es un sistema de autenticación de seguridad estándar industrial indicado para ser distribuido bajo redes publicas.

## Disponibilidad

El sistema de autenticación Kerberos no se distribuye con Postgres. Las versiones de Kerberos están típicamente disponibles como software opcional para los vendedores de sistemas operativos. Además, la distribución del código fuente se puede obtener a través de MIT Project Athena<sup>2</sup>.

**Nota:** Puedes desear obtener la versión del MIT si tu proveedor te proporciona una versión, ya que en algunos puntos de venta está deliberadamente capado o proporcionado sin interoperatividad con la versión del MIT.

Los usuarios que se encuentran fuera de los Estados Unidos de América o Canadá están avisados de que la distribución del código actual de encriptación de Kerberos está restringida por las leyes de exportación de Gobierno U. S.

Las preguntas acerca de tu Kerberos deben estar dirigidas a tu proveedor o a MIT Project Athena<sup>3</sup>. Notar que las FAQLs (Lista de preguntas frecuentes) se envían periódicamente a la Kerberos mailing list<sup>4</sup> (envía to subscribe<sup>5</sup>), y USENET news group<sup>6</sup>.

## Instalación

Instalación de Kerberos se trata en detalle en las *Notas de instalación de Kerberos*. Tener cuidado que el fichero llave servidor (the `srvtab` o `keytab`) es de alguna manera accesible para la lectura por la cuenta Postgres.

Postgres y sus clientes pueden ser compilables para usar la versión 4 o la versión 5 de los protocolos Kerberos del MIT configurando la variable `KRBVERS` en el fichero `src/Makefile.global` al valor apropiado. Puedes también cambiar la localización donde espera encontrar Postgres las librerías asociadas, los ficheros de cabecera y el fichero llave del servidor.

Cuando la compilación se haya completado, Postgres debe registrarse como un servicio Kerberos. Mirar las *Notas de Operaciones de Kerberos* y las páginas del manual relacionadas para más detalle del registro de servicios.

## Operaciones

Después de la instalación inicial, Postgres debe operar en todos los sentidos como un servicio normal Kerberos. Para más detalles en el uso de la autenticación, ver la *Guía del Usuario PostgreSQL* en las secciones referentes a `postmaster` y `psql`.

En los comentarios de la versión 5 de Kerberos, las siguientes suposiciones están hechas para el usuario y el nombrado de servicio:

- Los nombres de los usuarios principales (anames) se asemen para contener el nombre del usuario actual Unix/Postgres en el primer componente.
- El servicio Postgres se asume para ser tenido como dos componentes, el nombre de servicio y el nombre de host, canonizada en la versión 4 (p.e., con todos los sufijos de dominio borrados).

**Tabla 3-1. Ejemplos de Parámetros de Kerberos**

Parámetro	Ejemplo
user	frew@S2K.ORG
user	ao- ki/HOST=miyu.S2K.Berkeley.EDU@S2K.ORG
host	postgres_dbms/ucbvax@S2K.ORG

El soporte de la versión 4 desaparecerá después de algún tiempo tras la puesta en producción de la revisión de la versión 5 del MIT.

## Notas

1. <http://www.sai.msu.su/~megera/postgres/>
2. <ftp://athena-dist.mit.edu>
3. [info-kerberos@athena.mit.edu](mailto:info-kerberos@athena.mit.edu)
4. <mailto:kerberos@ATHENA.MIT.EDU>
5. <mailto:kerberos-request@ATHENA.MIT.EDU>
6. <news:comp.protocols.kerberos>



## Capítulo 4. Distribución del Sistema

### Figura 4-1. Distribución de los archivos de Postgres

*Distribución de los archivos de Postgres* muestra cómo se ordena la distribución de Postgres en su disco, cuando es instalada de la forma usual. Para simplificar, asumiremos que Postgres se ha instalado en el directorio `/usr/local/pgsql`. Por lo tanto, siempre que se mencione el directorio `/usr/local/pgsql`, usted deberá sustituirlo por aquél en el que haya instalado Postgres en su sistema. Todas los órdenes de Postgres se instalan en en el directorio `/usr/local/pgsql/bin`. Por consiguiente, debería incluir este directorio en su variable de entorno de path. Si utiliza un intérprete de órdenes derivado del Berkeley C, como `csh` o `tcsh`, incluya la línea

```
set path = ( /usr/local/pgsql/bin path )
```

en el archivo `.login` de su directorio particular (`home`). Si, en cambio, utiliza un intérprete de órdenes derivado del Bourne, como `sh`, `ksh` o `bash`, deberá agregar las líneas:

```
PATH=/usr/local/pgsql/bin:$PATH
export PATH
```

al archivo `.profile` ubicado en su directorio de inicio. De aquí en adelante se asumirá que usted ha agregado el directorio `/usr/local/pgsql/bin` a su path. Además, en este documento se hará referencia frecuentemente a «fijar una variable del intérprete de órdenes» o «fijar una variable de entorno». Si no comprendió totalmente el último párrafo sobre la modificación de la variable `PATH`, debería consultar la documentación sobre el funcionamiento de su intérprete de órdenes antes de continuar.

Si no ha realizado la instalación con las opciones por defecto, tal vez tenga algo de trabajo extra. Por ejemplo, si el servidor de base de datos está ubicado en una máquina remota, necesitará colocar el nombre del servidor en la variable de entorno `PGHOST`. Tal vez también necesite fijar la variable de entorno `PGPORT`. Básicamente, si intenta ejecutar una aplicación y ésta se queja por no poder conectarse con el proceso principal (`postmaster`), deberá volver un poco sobre sus pasos y asegurarse de que ha establecido todas las variables de entorno necesarias con los valores correctos.



## Capítulo 5. Instalación

Instrucciones para la instalación de PostgreSQL 7.0.

Los comandos que se mencionan a continuación fueron probados utilizando el shell `bash` sobre la distribución RedHat 5.2 de Linux. A menos que se indique lo contrario, estos comandos serán igualmente aplicables para la mayoría de los sistemas. Comandos como `ps` y `tar` pueden variar entre las distintas plataformas en cuanto a las opciones que deben usarse. *Utilice su sentido común* antes de teclear cualquiera de estos comandos.

Si aún no tiene la distribución de PostgreSQL, puede obtenerla en [ftp.postgresql.org](http://ftp.postgresql.org)<sup>1</sup>. Una vez obtenida, desempaquetela utilizando los siguientes comandos:

```
$ gunzip postgresql-7.0.tar.gz
$ tar -xf postgresql-7.0.tar
$ mv postgresql-7.0 /usr/src
```

Nuevamente, estos comandos pueden ser distintos en su sistema.

### Antes de comenzar

Para compilar PostgreSQL se requiere la utilidad GNU `make`. Otras utilidades similares *no funcionarán* en este caso. En los sistemas GNU/Linux, GNU `make` es a herramienta por defecto. En otros sistemas puede que encuentre que la herramienta GNU `make` se encuentre instalada con el nombre “`gmake`”. De aquí en adelante, utilizaremos este nombre para referirnos a GNU `make`, sin importar cuál sea el nombre que tiene en su sistema. Para probar GNU `make` teclee:

```
$ gmake -version
```

Si necesita obtener GNU `make`, lo puede encontrar en [ftp://ftp.gnu.org](http://ftp.gnu.org)<sup>2</sup>.

En <http://www.postgresql.org/docs/admin/ports.htm><sup>3</sup> puede encontrar información actualizada sobre las plataformas soportadas. En general, la mayoría de la plataformas compatibles con Unix que utilicen bibliotecas actualizadas debería ser capaz de ejecutar PostgreSQL. En el subdirectorio `doc` de la distribución existen varios documentos del tipo LEAME y otros con Preguntas de Uso Frecuente (FAQ en inglés) específicos para esa distribución, que pueden resultarle útiles si está teniendo problemas.

La cantidad mínima de memoria que se requiere para ejecutar PostgreSQL es de sólo 8 MB. Sin embargo, se verifica una notable mejora en la velocidad cuando ésta se expande a 96 MB o más. La regla es que, por más memoria que usted instale en su sistema, nunca será demasiada.

Verifique que exista suficiente espacio libre en el disco. Necesitará alrededor de 30 MB para los archivos con el código fuente durante la compilación, y unos 5 MB para el directorio de instalación. Una base de datos vacía ocupa aproximadamente 1 MB. De no estar vacía, la base ocupará unas cinco veces el espacio que ocuparía un archivo de texto que contuviera los mismos datos. Si ejecuta las pruebas de regresión, necesitará alrededor de 20 MB extra como espacio temporal.

Para revisar el espacio libre en el disco, utilice:

```
$ df -k
```

Dados los precios actuales de los discos duros, debería considerar conseguir uno grande y rápido antes de poner a trabajar una base de datos.

## Procedimiento de Instalación

### Instalación de PostgreSQL

Para una instalación de nuevas, o una actualización desde una versión previa de PostgreSQL:

1. Cree la cuenta de superusuario PostgreSQL. Éste es el usuario bajo el que corre el servidor. Para el uso en producción, deberá usted crear una cuenta de usuario diferente, sin privilegios (habitualmente se utiliza `postgres`). Si no tiene usted acceso como `root`, o quiere evitarse este paso, su propia cuenta de usuario es suficiente.

Ejecutar PostgreSQL como `root`, `bin`, o cualquier otra cuenta con permisos de acceso especiales es un riesgo de seguridad, y por ello está permitido.

No necesitará usted compilar e instalar bajo esta cuenta (aunque puede hacerlo). Se le dirá cuando necesite conectarse como el superusuario de la base de datos.

2. Si no está usted actualizando un sistema existente, salte a paso 4.

Ahora necesitará usted realizar una copia de seguridad (backup) de su base de datos existente. Si tiene una instalación razonablemente reciente de su base de datos (posterior a la 6.0), conseguirá un vaciado de la misma tecleando:

```
$ pg_dumpall > db.out
```

Si quiere usted conservar las identificación de los objetos (oids), utilice la opción `-o` al ejecutar `pg_dumpall`. Sin embargo, a no ser que tenga usted una razón especial para hacer esto, como podría ser utilizar estos identificadores como claves en las tablas, no lo haga.

Asegurese de utilizar el comando `pg_dumpall` de la versión que está usted ejecutando actualmente. Pero no utilice el script de 6.0 o el superusuario PostgreSQL tomará la propiedad de *todo*. Si es esta la versión que tiene usted, debería usted utilizar el comando `pg_dumpall` de una versión 6.x.x posterior. El correspondiente a la versión 7.0 no trabajará en bases de datos anteriores. Si está usted actualizando desde una versión previa a Postgres95 v1.09, deberá usted realizar un backup de su base de datos, instalar Postgres95 v1.09, restaurar su base de datos, y realizar el backup de nuevo.

#### Atención

Debe usted asegurarse de que su base de datos no se actualiza durante su backup. Si es necesario, pare el `postmaster`, edite los permisos del fichero `/usr/local/pgsql/data/pg_hba.conf` para permitirle a usted sólo su uso, y relance de nuevo `postmaster`.

3. Si está usted actualizando un sistema existente, mate ahora el servidor de la base de datos. Teclee:

```
$ ps ax | grep postmaster
```

Esto debería listar los números de proceso para una serie de procesos, de un modo similar a:

```
263 ? SW 0:00 (postmaster)
777 p1 S 0:00 grep postmaster
```

Teclee la siguiente línea, reemplazando *pid* con el identificador (id) del proceso *postmaster* (263 en el caso anterior). (No utilice el id del proceso "grep postmaster".) (N. del T. también puede hacerlo con la línea

```
$ ps ax | grep postmaster | grep -v grep
```

que le dará la misma salida, pero sin incluir la línea correspondiente al mismo proceso "grep". Fin de la N. del T.)

```
$ kill pid
```

**Sugerencia:** En sistemas que arrancan PostgreSQL en el durante la secuencia de arranque de la máquina, probablemente se encontrara un fichero startup que cumplirá el mismo cometido. Por ejemplo, en un sistema Linux RedHat, se debería encontrar que

```
$ /etc/rc.d/init.d/postgres.init stop
```

funcione correctamente para parar la base.

También deberá trasladar los directorios anteriores a otro sitio. Teclee lo siguiente:

```
$ mv /usr/local/pgsql /usr/local/pgsql.old
```

con sus propias rutas particulares.

4. Configure el código fuente para su sistema. Este es el paso en el que puede usted especificar su ruta de instalación actual para el proceso de construcción, y hacer elecciones sobre lo que tenga usted instalado. Cambiese al subdirectorio *src* y teclee:

```
$ ./configure
```

seguido de todas las opciones que desee usted dar. Para una primera instalación, debería ir todo bien sin dar ninguna. Para obtener una lista completa de las opciones, teclee:

```
./configure -help
```

Algunas de las opciones que se utilizan más a menudo son:

**-prefix=BASEDIR**

Selecciona un directorio base diferente para la instalación de PostgreSQL. La opción de defecto es */usr/local/pgsql*.

**-enable-locale**

Si quiere usted utilizar locales.

**-enable-multibyte**

Le permitirá utilizar páginas de caracteres multibyte. Se emplea principalmente para lenguajes como japonés, coreano o chino.

**-with-perl**

Construye la interface Perl. Note por favor que la interface Perl se instalará en el lugar habitual de los módulos Perl (habitualmente bajo `/usr/lib/perl`), de modo que deberá usted tener acceso root para realizar esta opción correctamente.

**-with-odbc**

Construye el paquete del driver ODBC.

**-with-tcl**

Construye las librerías de interface y los programas que requieren Tcl/Tk, incluyendo libpgtcl, pgctlsh y pgtksh.

## 5. Compile el programa. Teclee:

```
$ make
```

El proceso de compilación ocupará entre 10 minutos y una hora, variando en función de la máquina y de las opciones elegidas.

La última línea que se muestre por el proceso debería ser:

```
All of PostgreSQL is successfully made. Ready to install.
```

Recuerde que “make” se puede llamar “make” en su sistema.

## 6. Instale el programa. Teclee:

```
$ make install
```

7. Dígame a su sistema como encontrar las nuevas librerías compartidas. Cómo hacer esto varía de unas plataformas a otras. Lo que tiende a trabajar en todas partes es fijar la variable de entorno `LD_LIBRARY_PATH`:

```
$ LD_LIBRARY_PATH=/usr/local/pgsql/lib
$ export LD_LIBRARY_PATH
```

Quizá quiera usted poner estas dos líneas en un script de arranque de su shell, como `~/.bash_profile`.

En algunos sistemas se prefiere el siguiente método, pero debe usted tener acceso root. Edite el fichero `/etc/ld.so.conf` y añada una línea

```
/usr/local/pgsql/lib
```

Y ahora corra el comando `/sbin/ldconfig`.

En la duda, diríjase a las páginas de manual de su sistema. Si recibe usted más tarde un mensaje como

```
./psql: error in loading shared libraries
libpq.so.2.1: cannot open shared object file: No such file or directory
```

entonces es que todo lo anterior era necesario. Simplemente realice este paso de nuevo.

## 8. Cree la instalación de la base de datos. Para hacer esto, debe usted conectarse como su cuenta de superusuario de PostgreSQL. No trabajará como root.

```
$ mkdir /usr/local/pgsql/data
$ chown postgres /usr/local/pgsql/data
$ su - postgres
$ /usr/local/pgsql/initdb -D /usr/local/pgsql/data
```

La opción `-D` especifica la situación donde se almacenarán los datos. Puede usted utilizar cualquier otro path, porque no tiene porqué estar bajo el directorio de la instalación. Sólo asegúrese de que la cuenta del superusuario puede escribir en el directorio (o crearlo) antes de arrancar **initdb**. (Si estaba usted siguiendo los pasos de la instalación hasta ahora como el superusuario de PostgreSQL, puede que tenga usted que conectarse como root temporalmente para crear el directorio de datos.)

9. Los pasos previos deberían haberle indicado como arrancar el servidor de la base de datos. Ahagamos ahora:

```
$ /usr/local/pgsql/bin/postmaster -D /usr/local/pgsql/data
```

Esto arrancará el servidor en primer término. Para mandarlo a segundo plano, utilice la opción `-s`.

10. Si está usted actualizando desde una instalación anterior, extraiga sus datos con:

```
$ /usr/local/pgsql/bin/psql < db.out
```

Y haga también una copia de seguridad de su anterior fichero `pg_hba.conf`, así como de todos los demás ficheros que pueda usted haber creado para la autenticación, tales como ficheros de claves de acceso.

Con todo esto concluimos la instalación propiamente dicha. Para hacer su vida más productiva y agradable, debería mirar los siguientes pasos y sugerencias opcionales.

- La vida será más conveniente si fija usted algunas variables de entorno. Primero, probablemente quiera usted incluir `/usr/local/pgsql/bin` (o su equivalente) en su PATH. Para hacer esto, añada lo siguiente en su fichero de arranque de la shell, tal como `~/.bash_profile` (o `/etc/profile`, si quiere usted que afecte a todos los usuarios):

```
PATH=$PATH:/usr/local/pgsql/bin
```

Aún más, si usted fija la variable PGDATA en el entorno del superusuario de PostgreSQL, podrá usted omitir la opción `-D` para `postmaster` y `initdb`.

- Probablemente quiera usted instalar la documentación man y HTML. Teclee

```
$ cd /usr/src/pgsql/postgresql-7.0/doc
$ gmake install
```

Esto instalará ficheros bajo `/usr/local/pgsql/doc` y `/usr/local/pgsql/man`. Para permitir a su sistema encontrar la documentación man, necesitará añadir una línea como la siguiente en el fichero de arranque de la shell:

```
MANPATH=$MANPATH:/usr/local/pgsql/man
```

La documentación está también disponible en formato Postscript. Si tiene usted una impresora Postscript, o tiene su impresora ya preparada para aceptar ficheros Postscript utilizando un filtro de impresión, podrá imprimir la Guía de Usuario simplemente tecleando

```
$ cd /usr/local/pgsql/doc
$ gunzip -c user.ps.gz | lpr
```

Aquí tiene lo que debería hacer usted si tiene Ghostscript en su sistema y está escribiendo en una impresora laserjet:

```
$ alias gshp='gs -sDEVICE=laserjet -r300 -dNOPAUSE'
$ export GS_LIB=/usr/share/ghostscript:/usr/share/ghostscript/fonts
```

```
$ gunzip user.ps.gz
$ gshp -sOUTPUTFILE=user.hp user.ps
$ gzip user.ps
$ lpr -l -s -r manpage.hp
```

En caso de dudas, refierase a sus manuales o a su experto local.

Probablemente debería empezar por leer la Guía del Administrador si es usted completamente nuevo en PostgreSQL, porque contiene información sobre como declarar usuarios y la autenticación a la base de datos.

- Habitualmente, querrá usted modificar su computadora de modo que arranque el servidor de base de datos siempre que se ponga en marcha. Esto no es necesario; el servidor PostgreSQL se puede ejecutar normalmente desde cuentas no privilegiadas sin intervención de root.

Diferentes sistemas tienen diferentes convenciones para arrancar demonios en el momento de la puesta en marcha, de modo que deberá usted familiarizarse primer con ellos. La mayoría de los sistemas tienen un fichero `/etc/rc.local` o `/etc/rc.d/rc.local` que en la mayoría de los casos no es un mal lugar para situar este comando. Siempre que lo haga, el postmaster deberá ser ejecutado por el superusuario de PostgreSQL (`postgres`) *y no por root* o cualquier otro usuario. Por ello, probablemente quiera usted formar las líneas de comando iniciandolas con su `-c '...' postgres`.

Podría ser interesante mantener un registro de las salidas del servidor. Para arrancar de esta forma el servidor, intente:

```
nohup su -c 'postmaster -D /usr/local/pgsql/data > server.log 2>&1' postgres &
```

Aquí tenemos algunas otras sugerencias específicas del sistema operativo:

- Edite el fichero `rc.local` en NetBSD o el fichero `rc2.d` en SPARC Solaris 2.5.1 para que contenga la siguiente línea:

```
su postgres -c "/usr/local/pgsql/bin/postmaster -S -D /usr/local/pgsql/data"
```

- En FreeBSD RELEASE-2.2 editE `/usr/local/etc/rc.d/pgsql.sh` para que contenga las siguientes líneas y hégale `chmod 755` y `chown root:bin`.

```
#!/bin/sh
[ -x /usr/local/pgsql/bin/postmaster ] && {
    su -l postgres -c 'exec /usr/local/pgsql/bin/postmaster
        -D/usr/local/pgsql/data
        -S -o -F > /usr/local/pgsql/errlog' &
    echo -n ' postgres'
}
```

Usted puede colocar las rupturas de líneas como se muestra antes. La shell es capaz de seguir traduciendo más allá del final de la línea si no se ha terminado una expresión. El `exec` salva un nivel de shell bajo el proceso postmaster, de modo que el padre es `init`.

- En Linux RedHat, añada un fichero `/etc/rc.d/init.d/postgres.init` que se basará en el ejemplo que se encuentra en `contrib/linux/`. Y a continuación haga un link simbólico a este fichero desde `/etc/rc.d/rc5.d/S98postgres.init`.



- Ejecute los test de regresión. Los test de regresión son un conjunto de pruebas que verifican que PostgreSQL corre en su máquina en la forma en que los desarrolladores esperan que lo haga. Debería hacer esto definitivamente antes de poner un servidor en uso en producción. El fichero `/usr/src/pgsql/postgresql-7.0/src/test/regress/README` contiene instrucciones detalladas para correr e interpretar los tests de regresión.

## Notas

1. <ftp://ftp.postgresql.org>
2. <ftp://ftp.gnu.org>
3. <http://www.postgresql.org/docs/admin/ports.htm>



## Capítulo 6. Instalacion en Win32

Instrucciones para la instalacion de Postgres version 6.4(libreria de clientes) en Win32.

### Construccion de librerias

Los "makefiles" incluidos en Postgres estan escritos para Microsoft Visual C++ y probablemente no trabajen en otros sistemas. Es posible compilar las librerias manualmente en otros casos.

Para contruir las librerias, cambie al directorio `src` y escriba los comandos

```
copy include\config.h.win32 include\config.h
nmake /f win32.mak
```

Esto asume que usted tiene Visual C++ en su camino.

Los siguientes archivos seran creados:

- `interfaces\libpq\Release\libpq.dll` - La libreria dinamica enlazable
- `interfaces\libpq\Release\libpqdll.lib` - Libreria Importada para conectar el programa a `libpq.dll`
- `interfaces\libpq\Release\libpq.lib` - Version Estatica de la libreria
- `bin\psql\Release\psql.exe` - El monitor interactivo de SQL

### Instalacion de las librerias

La unica parte de la libreria que sera realmente instalada es la libreria de `libpq.dll`. El archivo en la mayoría de los casos debe ser puesto en el directorio `WINNT\SYSTEM32` (o en `WINDOWS\SYSTEM` en un sistema Windows 95/98). Si este archivo es instalado usando un programa de instalacion, debe ser instalado con un examinador de versiones usando `VERSIONINFO` que esta incluido en el archivo, para asegurar que una version mas reciente de la libreria no sea sobre escrita.

Si planea desarrollar usando `libpq` en esta maquina, tendra que anadir los directorios `src\include` y `src\interfaces\libpq` al camino en sus configuraciones.

### Usando las librerias

Para usar las librerias, debe anadir los archivos `libpqdll.lib` a su proyecto (en Visual C++, solo haga un click con el boton derecho en el proyecto y escoja anadirlo).

Una vez que esto esta hecho, sera posible usar la libreria como lo haria en cualquiera plataforma basada en UNIX.



## Capítulo 7. Entorno de tiempo de ejecución

En este capítulo se detalla la interacción entre Postgres y el sistema operativo.

### Utilizando Postgres desde Unix

Todas las órdenes de Postgres que se ejecutan directamente desde un shell de Unix se encuentran en el directorio ".../bin". Incluir este directorio en su variable de entorno path facilitará mucho la ejecución de los mismos.

Existe una colección de catálogos del sistema en cada servidor. La misma incluye una clase (pg\_user) que contiene una instancia para cada usuario válido en Postgres. La instancia especifica un conjunto de privilegios sobre Postgres, como la posibilidad de actuar como superusuario en Postgres, la posibilidad de crear/destruir bases de datos y la posibilidad de actualizar los catálogos del sistema. Un usuario de Unix no puede hacer nada con Postgres hasta que se instale una instancia apropiada en dicha clase. Puede encontrarse más información sobre los catálogos del sistema ejecutando consultas sobre las clases apropiadas.

### Iniciando postmaster

No le puede suceder nada a una base de datos a menos que esté corriendo el proceso postmaster. Como administrador, hay una serie de cosas que debe recordar antes de iniciar postmaster. Vea las secciones de instalación y configuración en este mismo manual. De todos modos, si ha unсталado Postgres siguiendo las instrucciones de instalación al pie de la letra, lo único que tendrá que hacer para iniciar el proceso postmaster es introducir esta simple orden:

```
% postmaster
```

Ocasionalmente, postmaster escribe mensajes que le serán de ayuda para resolver problemas. Si desea ver los mensajes de diagnóstico de postmaster, puede iniciarlo con la opción -d y redirigir la salida al archivo de registro:

```
% postmaster -d > pm.log 2>&1 &
```

Si no desea ver los mensajes, inícielo de la forma

```
% postmaster -S
```

y postmaster será "S"ilencioso. Observe que al no haber el signo ampersand ("&") al final del último ejemplo, no se ejecuta como proceso de fondo.

### Usando pg\_options

**Nota:** Contribución de Massimo Dal Zotto<sup>1</sup>

El archivo opcional `data/pg_options` contiene opciones usadas por el backend para controlar los mensajes de trazado y otros parámetros ajustables del mismo. El archivo se vuelve a leer cuando el backend recibe la señal `SIGHUP`, permitiendo cambiar las opciones de tiempo de ejecución al vuelo, sin que sea preciso reiniciar Postgres. En este archivo se pueden incluir opciones de depuración usadas por el paquete de trazado (`backend/utls/misc/trace.c`) o parámetros numéricos usados por el backend para controlar su comportamiento.

Todas las `pg_options` se inicializan a cero al iniciar el backend. Si se añaden o se modifican opciones, serán leídas por todos los backend que se inicien a continuación. Para que cualquier cambio tome efecto en los backend que están activos, es preciso enviar una señal `SIGHUP` al postmaster, quien reenviará la señal a todos los backend activos. Se pueden activar los cambios para un backend específico enviándole directamente una señal `SIGHUP`.

Las `pg_options` pueden especificarse también con la opción `-T` de Postgres:

```
postgres opciones -T "verbose=2,query,hostlookup-"
```

Las funciones usadas para indicar errores y mensajes de depuración pueden usar la facilidad `syslog(2)`. Los mensajes que se escriben en `stdout` o `stderr` incluyen un prefijo con la fecha, la hora y el número de proceso del backend que las genera.

```
#timestamp      #pid      #message
980127.17:52:14.173 [29271] StartTransactionCommand
980127.17:52:14.174 [29271] ProcessUtility: drop table t;
980127.17:52:14.186 [29271] SIIncNumEntries: table is 70% full
980127.17:52:14.186 [29286] Async_NotifyHandler
980127.17:52:14.186 [29286] Waking up sleeping backend process
980127.19:52:14.292 [29286] Async_NotifyFrontEnd
980127.19:52:14.413 [29286] Async_NotifyFrontEnd done
980127.19:52:14.466 [29286] Async_NotifyHandler done
```

Este formato facilita la lectura de los mensajes y permite saber exactamente lo que hace cada backend y en qué momento. También facilita la escritura de scripts `awk` o `perl` sencillos que analicen los historiales para detectar errores o problemas en la base de datos o para calcular estadísticas temporales de transacciones.

Los mensajes escritos a través de `syslog` utilizan la facilidad `LOG_LOCAL0`. El uso de `syslog` se controla mediante la opción `syslog` en `pg_options`. Por desgracia, muchas funciones llaman directamente a `printf()` para escribir sus mensajes a `stdout` o `stderr`, cuya salida no se puede controlar mediante la opción `syslog` ni puede incluir fecha y hora. Sería recomendable sustituir todas las llamadas a `printf` con la macro `PRINTF`, y todas las escrituras a `stderr` por la macro `EPRINTF`, para poder controlar toda la salida de un modo uniforme.

El formato del archivo `pg_options` es como sigue:

```
# comentario
option=valor_entero # Establece el valor de option
option              # establece option = 1
option+             # establece option = 1
option-             # establece option = 0
```

Observe que *keyword* puede ser una abreviatura de un nombre de opción de los definidos en `backend/utils/misc/trace.c`.

### Ejemplo 7-1. Archivo `pg_options`

Por ejemplo, mi archivo `pg_options` contiene los siguientes valores:

```
verbose=2
query
hostlookup
showportnumber
```

## Opciones reconocidas

Actualmente están definidas las opciones:

`all`

Marca de traza global. Los valores permitidos son:

0

Mensajes de trazado activados individualmente

1

Activar todos los mensajes de trazado

-1

Inhibir todos los mensajes de trazado

`verbose`

Marca de verbosidad. Valores permitidos:

0

Sin mensajes, éste es el valor por omisión.

1

Escribir mensajes de información.

2

Escribir más mensajes de información.

`query`

Trazar peticiones. Valores permitidos:

0

No escribir la petición.

1

Escribir una versión condensada de la petición en una línea.

4

Escribir la consulta completa.

plan

Escribir el plan de consulta.

parse

Escribir la salida del traductor de consultas.

rewritten

Escribir la consulta reescrita.

parserstats

Escribir las estadísticas del traductor de consultas.

plannerstats

Escribir las estadísticas del planificador.

executorstats

Escribir las estadísticas de ejecución.

shortlocks

De momento no se usa, pero se precisa para habilitar nuevas características en el futuro.

locks

Trazar bloqueos.

userlocks

Trazar bloqueos de usuario.

spinlocks

Trazar 'spin locks'.

notify

Trazar funciones de notificación.

malloc

Sin uso por el momento.

palloc

Sin uso por el momento.



lock\_debug\_oidmin

Minimum relation oid traced by locks.

lock\_debug\_relid

oid, if not zero, of relation traced by locks.

lock\_read\_priority

Sin uso por el momento.

deadlock\_timeout

Temporizador de comprobación de bloqueos circulares..

syslog

Marca de syslog. Valores permitidos:

0

Mensajes a stdout/stderr.

1

Mensajes a stdout/stderr y syslog.

2

Mensajes solamente a syslog.

hostlookup

Habilitar la consulta de nombre de host en ps\_status.

showportnumber

Mostrar el número de puerto en ps\_status.

notifyunlock

Desbloqueo de pg\_listener después de notify.

notifyhack

Borrar tuplas duplicadas de pg\_listener.

## Notas

1. <mailto:dz@cs.unitn.it>



## Capítulo 8. Seguridad

La seguridad de la base de datos esta implementada en varios niveles:

- Protección de los ficheros de la base de datos. Todos los ficheros almacenados en la base de datos estan protegidos contra escritura por cualquier cuenta que no sea la del superusuario de Postgres.
- Las conexiones de los clientes al servidor de la base de datos estan permitidas, por defecto, únicamente mediante sockets Unix locales y no mediante sockets TCP/IP. Ha de arrancarse el demonio con la opcion `-i` para permitir la conexion de clientes no locales.
- Las conexiones de los clientes se pueden restringir por dirección IP y/o por nombre de usuario mediante el fichero `pg_hba.conf` situado en `PG_DATA`.
- Las conexiones de los clientes pueden ser autenticadas mediante otros paquetes externos.
- A cada usuario de Postgres se le asigna un nombre de usuario y (opcionalmente) una contraseña. Por defecto, los usuarios no tienen permiso de escritura a bases de datos que no hayan creado.
- Los usuarios pueden ser incluidos en *grupos*, y el acceso a las tablas puede restringirse en base a esos grupos.

### Autenticacion de Usuarios

*Autenticacion* es el proceso mediante el cual el servidor de la base de datos y el `postmaster` se aseguran de que el usuario que está solicitando acceso a la base de datos es en realidad quien dice ser. Todos los usuarios que quieren utilizar Postgres se comprueban en la tabla `pg_user` para asegurarse que están autorizados a hacerlo. Actualmente, la verificación de la identidad del usuario se realiza de distintas formas:

Desde la shell del usuario

Un demonio que se lanza desde la shell del usuario anota el id original del usuario antes de realizar un `setuid` al id del usuario `postgres`. El id original del usuario se emplea como base para todo tipo de comprobaciones.

Desde la red

Si Postgres se instala como distribuido, el acceso al puerto TCP del `postmaster` está disponible para todo el mundo. El ABD configura el fichero `pg_hba.conf` situado en el directorio `PG_DATA` especificando el sistema de autenticacion a utilizar en base al equipo que realiza la conexión y la base de datos a la que se conecta. Ver `pg_hba.conf(5)` para obtener una descripción de los sistemas de autenticación disponibles. Por supuesto la autenticación basada en equipos no es perfecta incluso en los sistemas Unix. Es posible, para determinados intrusos, enmascarar el equipo de origen. Estos temas de seguridad están fuera del alcance de Postgres.

## Nombres de usuario y grupos

Para crear un nuevo usuario, ejecute el programa `createuser`.

Para añadir un usuario o un grupo de usuarios a un nuevo grupo uno de los usuarios debe crear el grupo y añadir al resto a ese grupo. En `Postgres` estos pasos no pueden realizarse actualmente mediante el comando **`create group`**. Los grupos se definen añadiendo los valores a la tabla `pg_group`, y usando el comando **`grant`** para asignar privilegios al grupo.

## Crear Usuarios

### Crear Grupos

Actualmente no hay una forma fácil de crear grupos de usuarios. Hay que añadirlos/actualizarlos uno a uno en la tabla `pg_group` table. Por ejemplo: `jolly=> insert into pg_group (groname, grosysid, grolist) jolly=> values ('posthackers', '1234', '{5443, 8261}')`; `INSERT 548224 jolly=> grant insert on foo to group posthackers; CHANGE jolly=>` Los campos de `pg_group` son: \* `groname`: El nombre del grupo. Este campo debe de ser unicamente alfanumérico. No añadas subrayados u otros signos de puntuación. \* `grosysid`: El id del grupo. El tipo del campo es `int4` y debe de ser único para cada grupo. \* `grolist`: La lista de id de usuarios que pertenecen al grupo. Este campo es de tipo `int4`.

## Asignar usuarios a los Grupos

## Control de Acceso

`Postgres` proporciona mecanismos para permitir a los usuarios limitar el acceso que otros usuarios tendrán a sus datos.

### SuperUsuarios de la Base de Datos

Los SuperUsuarios de la base de datos (aquellos que tienen el campo `pg_user . usesuper` activado) ignoran todos los controles de acceso descritos anteriormente con dos excepciones: las actualizaciones del catálogo del sistema no están permitidas si el usuario no tiene el campo `pg_user . usecatupd` activado, y nunca se permite la destrucción del catálogo del sistema (o la modificación de sus estructuras).

### Privilegios de acceso

El uso de los privilegios de acceso para limitar la lectura, escritura y la puesta de reglas a las clases se trata en *`grant/revoke(1)`*.

### Borrado de clases y modificación de estructuras.

Los comandos que borran o modifican la estructura de una clase, como **`alter`**, **`drop table`**, y **`drop index`**, solo funcionan con el propietario de la clase. Como hemos dicho antes, estas operaciones no están permitidas *nunca* en los catálogos del sistema.

## Funciones y Reglas

Las funciones y las reglas permiten a los usuarios insertar código en el servidor de la base de datos que otros usuarios pueden ejecutar sin saberlo. Ambos mecanismos permiten a los usuarios alojar *caballos de troya* con relativa impunidad. La única protección efectiva es un estrecho control sobre quien puede definir funciones (por ejemplo, escribir en tablas con campos SQL) y reglas. También se recomienda auditar seguimientos y alertas en `pg_class`, `pg_user` y `pg_group`.

### Funciones

Las funciones escritas en cualquier lenguaje excepto SQL se ejecutan por el servidor de la base de datos con el mismo permiso que el usuario *postgres* (el servidor de la base de datos funciona con el user-id de *postgres*). Es posible cambiar las estructuras de datos internas del servidor por los usuarios, desde dentro de funciones de confianza. Es por ello que este tipo de funciones pueden, entre otras cosas, evitar cualquier sistema de control de acceso. Este es un problema inherente a las funciones definidas por los usuarios en C.

### Reglas

Como en las funciones SQL, las reglas también se ejecutan con la identidad y los permisos del usuario que llamó al servidor de la base de datos.

### Caveats

There are no plans to explicitly support encrypted data inside of Postgres (though there is nothing to prevent users from encrypting data within user-defined functions). There are no plans to explicitly support encrypted network connections, either, pending a total rewrite of the frontend/backend protocol.

User names, group names and associated system identifiers (e.g., the contents of `pg_user.usesysid`) are assumed to be unique throughout a database. Unpredictable results may occur if they are not.



## Capítulo 9. Agregar y Eliminar Usuarios

`createuser` permite que usuarios específicos accedan a Postgres. `dropuser` elimina usuarios y previene que éstos accedan a Postgres.

Estas órdenes sólo afectan a los usuarios con respecto a Postgres; no tienen efecto en otros privilegios del usuario o en su estado con respecto al sistema operativo subyacente.





## Capítulo 10. Gestión de Disco

### Localizaciones Alternativas

Se puede crear una base de datos en una localización diferente a la establecida por defecto durante la instalación. Recuerde que todos los accesos a base de datos ocurren realmente a través del proceso en segundo plano, así que éste debe poder acceder a cualquier especificación.

Se crean localizaciones alternativas y referencias mediante una variable de entorno que da el path absoluto hasta la situación de almacenamiento deseada. Esta variable de entorno debe estar definida antes de que el proceso en segundo plano sea arrancado y debe ser modificable mediante la cuenta del administrador de postgres. Cualquier variable de entorno puede ser utilizada para referirse a una localización alternativa, si bien se recomienda la utilización de un nombre de variable con prefijo PGDATA para evitar confusión y conflicto con otras variables.

**Nota:** En versiones previas de Postgres, también estaba permitido utilizar un nombre de path absoluto para especificar una localización de almacenamiento alternativa. Se prefiere el método de especificación de variables de entorno, puesto que concede al administrador del sistema más flexibilidad en la gestión del almacenamiento en disco. Si prefiere utilizar paths absolutos, puede hacerlo definiendo "ALLOW\_ABSOLUTE\_DBPATHS" y recompilando Postgres. Para hacer esto, añada cualquiera de estas líneas

```
#define ALLOW_ABSOLUTE_DBPATHS 1
```

al archivo `src/include/config.h`, o especifique

```
CFLAGS+= -DALLOW_ABSOLUTE_DBPATHS
```

en su `Makefile.custom`.

Recuerde que la creación de una base de datos la ejecuta realmente un proceso de la base de datos en segundo plano. Por lo tanto, cualquier variable de entorno que especifique una localización alternativa debe ser definida antes de que el proceso en segundo plano sea arrancado. Para definir una localización alternativa apuntando a PGDATA2 `/home/postgres/data`, primero escriba

```
% setenv PGDATA2 /home/postgres/data
```

para definir la variable de entorno que será utilizada con las órdenes siguientes. Normalmente, querrá definir esta variable en el fichero de inicialización del super usuario de Postgres, `.profile` o `.cshrc` para asegurar que está definido al arrancar el sistema. Se puede utilizar cualquier variable de entorno para referirse a una localización alternativa, aunque se prefiere que las variables estén prefijadas con "PGDATA" para eliminar confusiones y la posibilidad de conflictos con otras variables, o su reescritura.

Para crear un área de almacenamiento de datos en PGDATA2, asegúrese de que `/home/postgres` ya existe y puede ser escrito por el administrador de postgres. Después desde la línea de órdenes, escriba

```
% setenv PGDATA2 /home/postgres/data
% initlocation $PGDATA2
```

```
Creating Postgres database system directory /home/postgres/data  
Creating Postgres database system directory /home/postgres/data/base
```

Para comprobar la nueva localización, cree una base de datos test escribiendo

```
% createdb -D PGDATA2 test  
% dropdb test
```

## Capítulo 11. Gestión de una base de datos

Si el programa `postmaster` de Postgres está cargado y en ejecución, podemos crear algunas bases de datos con las que experimentar. En este documento describiremos las órdenes básicas para gestionar una base de datos.

### Creación de una base de datos

Supongamos que quiere crear una base de datos llamada `mibase`. Puede hacerlo con el siguiente orden:

```
% createdb nombredb
```

Postgres le permite crear cualquier número de bases de datos en una máquina dada, y usted se convierte automáticamente en el administrador de la base de datos que acaba de crear. Los nombres de las bases de datos han de comenzar por un carácter alfabético, y su longitud está limitada a 31 caracteres. No todos los usuarios están autorizados para convertirse en administradores de bases de datos. Si Postgres rechaza la orden de crear bases de datos, es que necesita que el administrador del sistema le garantice derechos para crear las bases de datos. Consulte a su administrador de sistemas si le ocurre eso.

### Acceso a la base de datos

Una vez que ha construido la base de datos, puede acceder a ella por los siguientes medios:

- Ejecutando el programa monitor de terminal de Postgres (`psql`) que le permite introducir, editar y ejecutar órdenes SQL de un modo interactivo.
- Escribiendo un programa en C que use la biblioteca de rutinas `libpq`. Esto le permite enviar órdenes SQL desde C y obtener las respuestas y mensajes de estado en su programa. Esta interfaz se discute en el documento *Guía de programación en PostgreSQL*.

Puede que quiera ejecutar el programa `psql`, para probar los ejemplos de este manual. Puede activarlo para la base de datos `nombredb` escribiendo la orden:

```
% psql nombredb
```

Recibirá como respuesta el siguiente mensaje:

```
Welcome to the Postgres interactive sql monitor:
```

```
type \? for help on slash commands
type \q to quit
type \g or terminate with semicolon to execute query
You are currently connected to the database: nombredb
```

```
nombredb=>
```

Este indicativo le informa de que el monitor de terminal se encuentra dispuesto y que puede escribir consultas SQL en el espacio de trabajo creado por el citado monitor de terminal. El programa `psql` responde a códigos de escape que comienzan con la barra invertida, `"\"`. Por ejemplo, puede obtener ayuda sobre la sintaxis de varias órdenes SQL de Postgres escribiendo:

```
nombredb=> \h
```

Una vez que ha terminado de introducir sus consultas en el espacio de trabajo, puede pasar el contenido de éste al servidor de Postgres escribiendo:

```
nombredb=> \g
```

Esto le dice al servidor que procese la consulta. Si termina su consulta con punto y coma, no es necesario que introduzca la secuencia `\g`. `psql` procesará automáticamente consultas terminadas en punto y coma. Para leer las consultas de un fichero, en lugar de introducirlas interactivamente, escriba:

```
nombredb=> \i fichero
```

Para salir de `psql` y volver a Unix, escriba:

```
nombredb=> \q
```

y `psql` terminará y le devolverá el intérprete de órdenes. (Para conocer más órdenes de escape, escriba `\h` en la entrada del monitor.) Los espacios en blanco (espacios, tabuladores y saltos de línea) pueden usarse libremente en las consultas SQL. Los comentarios en una sola línea se indican mediante dos guiones (`"--"`). Todo lo que vaya desde los guiones hasta el fin de la línea será ignorado. Los comentarios que abarcan múltiples líneas, o que están dentro de una línea se encierran entre `"/* ... */"`, una convención que se tomó prestada de Ingres.

## Destrucción de una base de datos

Si usted es el administrador de la base de datos `mibase`, puede destruirla usando el siguiente orden de Unix:

```
% dropdb nombredb
```

Esta acción elimina físicamente todos los ficheros asociados con la base de datos y no puede ser invertida, por lo que sólo debe ser usada con gran precaución.

Es también posible destruir una base de datos desde una sesión SQL usando:

```
> drop database nombredb
```

## Copia de seguridad y restauración

### Atención

Deben realizarse copias de seguridad de las bases de datos regularmente. Dado que Postgres gestiona sus propios ficheros en el sistema, *no se recomienda* confiar en los sistemas de copia de seguridad del sistema para las copias de respaldo de las bases de datos; no hay garantía de que los ficheros estén en un estado consistente que permita su uso después de la restauración.

Postgres proporciona dos utilidades para realizar las copias de seguridad de su sistema: `pg_dump` para copias de seguridad de bases de datos individuales y `pg_dumpall` para realizar copias de seguridad de toda la instalación de una sola vez.

La copia de seguridad de una sola base de datos puede realizarse usando la siguiente orden:

```
% pg_dump nombredb > nombredb.pgdump
```

y puede ser restaurada usando

```
cat nombredb.pgdump | psql nombredb
```

Esta técnica puede usarse para mover bases de datos a una nueva localización y para renombrar bases de datos existentes..

## Bases de datos grandes

**Autor:** Escrito por Hannu Krosing<sup>1</sup> on 1999-06-19.

Dado que Postgres permite tablas de mayor tamaño que el permitido por el sistema de ficheros, puede resultar problemático el volcado de una tabla a un fichero, ya que el fichero resultante seguramente superará el tamaño máximo permitido.

Como `pg_dump` escribe en stdout, puede usar las herramientas \*nix para sortear estos posibles problemas:

- Uso de volcados comprimidos:

```
% pg_dump nombredb | gzip > nombrefichero.dump.gz
```

la recuperamos con:

```
% createdb nombredb
% gunzip -c nombrefichero.dump.gz | psql nombredb
```

```
o
```

```
% cat nombrefichero.dump.gz | gunzip | psql nombredb
```

- Use split:

```
% pg_dump nombredb | split -b 1m - nombrefichero.dump.
```

y lo recuperamos con:

```
% createdb nombredb  
% cat nombrefichero.dump.* | psql nombredb
```

Por supuesto, el nombre del fichero (*nombrefichero*) y el contenido de la salida de `pg_dump` no tiene por qué coincidir con el nombre de la base de datos. Además, la base de datos restaurada puede tener un nombre distinto, por lo que este mecanismo también es efectivo para renombrar bases de datos.

## Notas

1. hannu@trust.ee

## Capítulo 12. Tratamiento de problemas

### Fallos de inicio de Postmaster

Hay varias posibles razones para que postmaster no pueda inicializarse. Compruebe el fichero de registro de postmaster, o inícielo manualmente (sin redirigir la salida estándar o la de errores) para ver los mensajes que aparecen. Alguno de los posibles mensajes de error son autoexplicativos, pero los hay que pueden no serlos tanto:

```
FATAL: StreamServerPort: bind() failed: Address already in use
       Is another postmaster already running on that port?
```

Esto normalmente significa lo que sugiere: accidentalmente ha iniciado una segunda instancia de postmaster en el mismo puerto en el que ya se está ejecutando uno. Sin embargo, si el mensaje de error del núcleo no es "Address already in use" o alguna variante, puede estar ocurriendo otro problema. Por ejemplo, el tratar de iniciar una sesión de postmaster en un puerto de error reservado puede producir algo como:

```
$ postmaster -i -p 666
FATAL: StreamServerPort: bind() failed: Permission denied
       Is another postmaster already running on that port?
```

```
IpcMemoryCreate: shmget failed (Invalid argument) key=5440001, size=83918612, per-
mission=600
FATAL 1: ShmemCreate: cannot create region
```

Un mensaje como éste posiblemente indica que el límite impuesto al tamaño de las zonas de memoria compartidas es menor que área de «buffer» que Postgres está intentando crear. (O puede significar que no dispone de soporte para la memoria compartida de tipo SysV configurado en su núcleo.) Como arreglo temporal puede tratar de iniciar postmaster con un número de «buffers» menor de lo normal (parámetro -B). Sin embargo, debería reconfigurar su núcleo para incrementar el tamaño permitido para la memoria compartida. Este mensaje puede aparecer cuando trate de iniciar varias sesiones de postmaster en la misma máquina, si el total de espacio necesario excede el límite impuesto por el núcleo.

```
IpcSemaphoreCreate: semget failed (No space left on device) key=5440026, num=16, per-
mission=600
```

Un mensaje como éste *no* significa que se haya quedado sin espacio en el disco; significa que la cantidad máxima de semáforos permitidos por el núcleo para el SysV es menor que la cantidad que Postgres intenta crear. Como antes, puede evitar este problema iniciando el postmaster con un número de procesos «backend» menor (parámetro -N), pero sería mejor que incrementara el límite impuesto por el núcleo.

### Problemas con la conexión del Cliente

Una vez que tiene el postmaster en ejecución, al tratar de conectar con él mediante una aplicación cliente puede producirse un fallo por varias razones. Los ejemplos de

mensajes de error mostrados aquí son para clientes basados en las versiones recientes de libpq; los clientes basados en otras bibliotecas de interfaz pueden producir otros mensajes, con más o menos información.

```
connectDB() - connect() failed: Connection refused
Is the postmaster running (with -i) at 'server.joe.com' and accepting con-
nections on TCP/IP port '5432'?
```

Este es el fallo genérico de 'No puedo encontrar un postmaster con el que comunicarme'. Puede ocurrir algo así cuando se intenta una comunicación TCP/IP o mediante socket Unix con un postmaster local:

```
connectDB() - connect() failed: No such file or directory
Is the postmaster running at 'localhost' and accepting connections on Unix soc-
ket '5432'?
```

La última línea es útil para verificar que el cliente está tratando de conectar donde se supone que debe hacerlo. Si en realidad no hay ningún postmaster ejecutándose allí, el mensaje de error del núcleo será del tipo de 'Conexión rehusada' o de 'No existe el fichero o directorio', como los anteriores. (Es particularmente importante tener en cuenta que 'Conexión rehusada' en este contexto *no* significa que el postmaster haya recibido la petición de conexión y la haya rechazado; en este caso se produce un mensaje diferente, como se verá.) Otros mensajes de error, como el de "Connection timed out" sí indican problemas más importantes, como la falta de conectividad en la red.

```
No pg_hba.conf entry for host 123.123.123.123, user joeblow, database testdb
```

Esto es lo más probable que obtenga si consigue contactar con un postmaster, pero éste no quiere hablar con usted. Como sugiere el mensaje, el postmaster rehúsa la petición de conexión porque no encuentra un renglón de autorización en su fichero de configuración pg\_hba.conf

```
Password authentication failed for user 'joeblow'
```

Los mensajes como éste indican que ha contactado con el postmaster, y éste está dispuesto a hablar con usted, pero no lo hará hasta que supere el método de autorización especificado en el fichero pg\_hba.conf. Compruebe la clave que está enviando, o su programa IDENT o Kerberos, si el mensaje de error menciona alguno de esos tipos de autenticación.

```
FATAL 1: SetUserId: user 'joeblow' is not in 'pg_shadow'
```

Esta es otra variante de fallo de autenticación: no se ha ejecutado la orden de Postgres 'create\_user' para el nombre de usuario indicado.

```
FATAL 1: Database testdb does not exist in pg_database
```

No hay base de datos con ese nombre bajo el control de ese postmaster. Nótese que si no especifica el nombre de la base de datos, se aplica por defecto su nombre de usuario en Postgres, lo que puede no ser lo correcto.



## Depuración de mensajes

El `postmaster` presenta ocasionalmente mensajes que pueden ser de ayuda en la solución de problemas. Si desea ver mensajes de depuración de `postmaster`, puede iniciarlo con la opción `-d` y redirigir la salida a un fichero de registro:

```
% postmaster -d > pm.log 2>&1 &
```

Si no desea ver estos mensajes, puede escribir

```
% postmaster -S
```

y el `postmaster` entrará en modo 'S'ilencioso. Nótese que no se incluye el símbolo `'&'` en el último ejemplo, ya que el `postmaster` se ejecutará en segundo plano.

## pg\_options

**Nota:** Contribución de Massimo Dal Zotto<sup>1</sup>

El fichero opcional `data/pg_options` contiene opciones de ejecución usadas por el backend para controlar mensajes de ejecución y otros parámetros ajustables. Lo que hace interesante a este fichero es el hecho de que es releído por el backend cuando recibe una señal `SIGHUP`, haciendo así posible cambiar opciones de ejecución sin tener que reiniciar Postgres. Las opciones especificadas en este fichero pueden incluir puntos de depuración usados por el paquete `trace` (`backend/utls/misc/trace.c`) o parámetros numéricos que puede usar el backend para controlar su comportamiento. Se pueden definir nuevas opciones y parámetros en `backend/utls/misc/trace.c` y en `backend/include/utls/trace.h`.

Las opciones de `pg_option` pueden especificarse con el parámetro `-T` de Postgres:

```
postgres opciones -T "verbose=2,query,hostlookup-"
```

Las funciones usadas para imprimir errores y mensajes de depuración pueden ahora usar la utilidad `syslog(2)`. Los mensajes impresos en `stdout` o `stderr` son precedidos por una etiqueta informativa que incluye la fecha y hora y el pid del backend:

```
#timestamp          #pid    #message
980127.17:52:14.173 [29271] StartTransactionCommand
980127.17:52:14.174 [29271] ProcessUtility: drop table t;
980127.17:52:14.186 [29271] SIIncNumEntries: table is 70% full
980127.17:52:14.186 [29286] Async_NotifyHandler
980127.17:52:14.186 [29286] Waking up sleeping backend process
980127.19:52:14.292 [29286] Async_NotifyFrontEnd
980127.19:52:14.413 [29286] Async_NotifyFrontEnd done
980127.19:52:14.466 [29286] Async_NotifyHandler done
```

Este formato mejora la legibilidad de los registros, y permite comprender qué «backend» concreto está haciendo qué y en qué momento. También hace más fácil escribir guiones (scripts) en awk o perl que monitoricen el fichero de registro para detectar errores o problemas en la base de datos, o para contabilizar estadísticas temporales de las transacciones.

Los mensajes impresos por syslog usan la utilidad de registro LOG\_LOCAL0. El uso de syslog puede ser controlado por las opciones referentes a él en syslog. Desgraciadamente, muchas funciones llaman directamente a `printf()` para mostrar sus mensajes en stdout o stderr y esta salida no puede ser redirigida a syslog o incluir información sobre fecha y hora. Sería aconsejable que todas las llamadas a `printf` pudieran ser reemplazadas por la macro `PRINTF` y la salida a stderr se cambiaran para que usaran `EPRINTF` en su lugar, de modo que se pudieran controlar todas las salidas de un modo uniforme.

El formato del fichero `pg_options` es como sigue:

```
# comentario
opción=valor_entero # set value for opción
opción              # set opción = 1
opción+             # set opción = 1
opción-             # set opción = 0
```

Nótese que *palabra\_clave* puede ser también una abreviación del nombre de opción definido en `backend/utls/misc/trace.c`.

Véase *Usando pg\_options* para una lista completa de las opciones y sus posible valores.

## Notas

1. <mailto:dz@cs.unitn.it>

## Capítulo 13. Recuperación de bases de datos

Esta sección necesita ser escrita. ¿Algún voluntario?



## Capítulo 14. Pruebas de regresión

### Instrucciones y análisis del test de regresión

Los tests de regresión de PostgreSQL son un conjunto completo de pruebas para la implementación de SQL embebidos en PostgreSQL. Realizan pruebas tanto sobre operaciones SQL estándar, como también sobre las capacidades añadidas por PostgreSQL.

Los tests de regresión fueron desarrollados originalmente por Jolly Chen y Andrew Yu, y fueron extensamente repasados/reempaquetados por Fournier y Thomas Lockhart. Para PostgreSQL v6.1 en adelante los tests de regresión forman parte de cada release oficial.

Algunas bases de datos PostgreSQL correctamente instaladas y totalmente funcionales pueden fallar en alguno de estos test de regresión debido a problemas con la representación del punto flotante y el soporte de zona horaria. Los tests actuales son evaluados usando un sencillo algoritmo "diff", y son muy sensibles a pequeñas diferencias en el sistema. Para tests aparentemente fallidos, si se examinan estas diferencias, pueden revelar no ser significativas.

Las notas sobre tests de regresión de abajo asumen lo siguiente (excepto en casos indicados):

- Las instrucciones son compatibles con Unix. Vea la nota abajo.
- Se usan las opciones por defecto excepto donde se indica.
- El usuario postgres es el superusuario Postgres.
- La ruta de las fuentes es /usr/src/pgsql (son posibles otras rutas).
- La ruta de los ejecutables es /usr/local/pgsql (son posibles otras rutas).

### Entorno de regresión

Para preparar los tests de regresión, haga **make all** en el directorio de los tests de regresión. Esto compila un programa C con funciones extendidas PostgreSQL en un librería compartida. Se generan algunos guiones (scripts) SQL localizados y archivos de salida comparativos para los tests que los necesiten. La localización reemplaza macros en los archivos de fuentes con rutas absolutas y nombres de usuario.

Normalmente, los tests de regresión deben ser ejecutados por el usuario postgres ya que el directorio 'src/test/regress' y subdirectorios son de su propiedad. Si ejecuta los test de regresión con otro usuario el directorio 'src/test/regress' debe tener permisos de escritura para ese usuario.

Antes era estrictamente necesario ejecutar el postmaster con la zona horaria del sistema establecida en PST, pero ya no es necesario. Puede ejecutar los tests de regresión sobre su configuración habitual del postmaster. El guión (script) del test establecerá la variable de entorno PGTZ para asegurar que los tests dependientes de la zona horaria produzcan los resultados esperados. De todas formas, su sistema debe proporcionar librerías de soporte para la zona horaria PST8PDT, o los tests dependientes de la zona horaria fallarán. Para comprobar que su equipo soporta esto, escriba lo siguiente:

```
setenv TZ PST8PDT
date
```

La orden "date" de arriba tiene que devolver la hora actual del sistema en la zona horaria PST8PDT. Si la base de datos PST8PDT no está disponible, entonces el sistema tiene que devolver la hora en GMT. Si la zona horaria PST8PDT no está disponible, puede establecer las reglas para esa zona horaria explícitamente.

```
setenv PGTZ PST8PDT7,M04.01.0,M10.05.03
```

## Estructura de directorios

**Nota:** Esto debería ser una tabla en la sección anterior.

```
input/ .... .Archivos fuente que son convertidos, usando 'make all', en
             alguno de los archivos .sql en el subdirectorío 'sql'

output/ ... .Archivos fuente que son convertidos, usando 'make all', en
           archivos .out en el subdirectorío 'expected'

sql/ ..... .Archivos sql usados para ejecutar los tests de regresión

expected/ . .Archivos .out que representan lo que *esperamos* que parezcan
            los resultados

results/ .. .Archivos .out que contienen lo que los resultados *realmente*
            parecen. Además es usado como almacén temporal para el test de
            copia de tablas.
```

## Procedimiento para el test de regresión

Las instrucciones están probadas en un RedHat Linux versión 4.2 usando el intérprete de órdenes bash. Excepto donde se indique, seguramente funcione en la mayoría de sistemas. Instrucciones como `ps` y `tar` cambian mucho las opciones que debe usar en cada plataforma. *Use el sentido común* antes de escribir estas instrucciones.

Para una instalación nueva o si está actualizando una versión anterior de Postgres:

### Configuración de la Regresión de Postgres

1. El archivo `/usr/src/pgsql/src/test/regress/README` tiene instrucciones detalladas para la ejecución e interpretación de los tests de regresión. Lo que sigue es una versión más corta:

Si el postmaster no se está ejecutando ya, inicie el postmaster en una ventana que esté disponible escribiendo

```
postmaster
```

o inicie el demonio postmaster en segundo plano escribiendo

```
cd
nohup postmaster > regress.log 2>&1 &
```

Ejecute postmaster desde la cuenta de superusuario de Postgres (normalmente la cuenta postgres).

**Nota:** No ejecute `postmaster` desde la cuenta de `root`.

2. Si ha ejecutado anteriormente los tests de regresión, borre el directorio de trabajo con:

```
cd /usr/src/pgsql/src/test/regress
gmake clean
```

No necesita escribir "gmake clean" si es la primera vez que está ejecutando los tests.

3. Ejecute los tests de regresión. Escriba

```
cd /usr/src/pgsql/src/test/regress
gmake all
```

4. Ejecute los tests de regresión. Escriba

```
cd /usr/src/pgsql/src/test/regress
gmake runtest
```

5. Debería obtener en la pantalla (y además escrito en el archivo `./regress.out`) una serie de líneas indicando qué tests han pasado y qué tests han fallado. Tenga en cuenta que puede ser normal que alguno de los tests falle. Para los tests fallidos, use `diff` para comparar los archivos de los directorios `./results` y `./expected`. Si falla `float8`, escriba algo como esto:

```
cd /usr/src/pgsql/src/test/regress
diff -w expected/float8.out results
```

6. Después de ejecutar los tests y examinar los resultados, escriba

```
dropdb regression
cd /usr/src/pgsql/src/test/regress
gmake clean
```

para recuperar el espacio en disco temporal usado por los tests.

## Análisis de Regresión

Los resultados se encuentran en los archivos del directorio `./results`. Estos resultados pueden ser comparados con los resultados del directorio `./expected` usando `'diff'`. (El guión (script) del test hace esto por usted, y deja las diferencias en `./regression.diffs`.)

Los archivos pueden no corresponderse de forma exacta. El guión del test informará de una diferencia como "failure" (fallo), pero la diferencia puede deberse a pequeñas variaciones entre plataformas en los mensajes de error, comportamiento de la librería matemática, etc. "Fallos" de este estilo no indican necesariamente un problema con Postgres.

Por tanto, es necesario examinar las diferencias de cada test "fallido" con el fin de determinar si existe un problema realmente. Los siguientes puntos intentan proporcionar una guía para determinar si una diferencia es significativa o no.

### Diferencias en los mensajes de error

Alguno de los tests de regresión incluyen intencionadamente valores de entrada no válidos. Los mensajes de error pueden venir tanto del código de Postgres como de las rutinas de sistema de la plataforma en la que nos encontremos. En el último caso, los mensajes pueden variar entre plataformas, pero deben reflejar información similar. Estas diferencias en los mensajes darán como resultado un test "fallido" que puede ser validado mediante una inspección.

### Diferencias en fechas y horas

Muchos de los resultados de fecha y hora son dependientes del entorno de la zona horaria. Los archivos de referencia están generados para la zona horaria PST8PDT (Berkeley, California) y aparentemente pueden parecer fallos si los tests no se ejecutan con esta zona horaria establecida. El programa que ejecuta los tests de regresión establece la variable de entorno PGTZ a PST8PDT para asegurar resultados parecidos.

Parece que algunos sistemas no aceptan la sintaxis recomendada para establecer explícitamente las reglas de la zona horaria local; puede ser que necesite usar una forma distinta para establecer PGTZ en estas máquinas.

Algunos sistemas que usan librerías antiguas de zonas horarias fallan al aplicar las correcciones de ahorro de luz solar en las fechas anteriores a 1970, causando que las horas de esas fechas aparezcan en PST a pesar de todo. Esto dará pie a diferencias localizadas en los resultados de los tests.

### Diferencias en punto flotante

Algunos de los tests implican calcular números de 64-bits (float8) a partir de las columnas de una tabla. Se han observado diferencias en los resultados que devuelven funciones matemáticas en columnas de tipo float8. Los tests con float8 y de geometría son particularmente propensos a pequeñas diferencias entre plataformas. Se precisa una comparación con lupa por parte humana para determinar diferencias que normalmente se encuentran 10 posiciones a la derecha del punto decimal.

Algunos errores de señales del sistema con `pow()` y `exp()` difieren de los mecanismos que espera el actual código de Postgres.



## Diferencias en polígonos

Varios de los tests incluyen operaciones con coordenadas sobre el callejero de Oakland/Berkley CA. Los datos de este mapa vienen expresados como polígonos cuyos vértices están representados en pares de números float8 (latitud y longitud decimal). Inicialmente, se crean y llenan algunas tablas con coordenadas, después se crean algunas vistas (Views) haciendo el Join de dos tablas usando el operador de intersección de polígonos (##), y después se realiza un Select sobre la vista. Cuando comparamos los resultados de diferentes plataformas, las diferencias aparecen en el segundo o tercer lugar a la derecha del punto decimal. Las instrucciones SQL donde se dan estos problemas son las siguientes:

```
QUERY: SELECT * from street;
QUERY: SELECT * from iexit;
```

## Diferencias aleatorias

Hay al menos un caso de test en random.out que esta diseñado para producir resultados aleatorios. Esto causa que random falle el test de regresión cada vez. Escribir

```
diff results/random.out expected/random.out
```

debe producir una o unas pocas líneas de diferencias por esta razón, pero otras variaciones en punto flotante o en arquitecturas distintas pueden causar más diferencias.

## Los archivos “expected”

Los archivos ./expected/\*.out fueron adaptados del monolítico archivo original expected.input proporcionado por Jolly Chen. Versiones más modernas de estos archivos generadas en varias máquinas de desarrollo han sido sustituidas después de una cuidadosa (?) inspección. Muchas de estas máquinas de desarrollo están ejecutando variantes del Unix OS (FreeBSD, Linux, etc) en hardware Ix86. El archivo original expected.input fue creado en un sistema SPARC Solaris 2.4 usando el código de postgres-1.02a5.tar.gz. Fue comparado con un archivo creado en un sistema I386 Solaris 2.4 y las diferencias fueron solamente en los polígonos de punto flotante en el tercer dígito a la derecha del punto decimal. (vea más arriba) El archivo original sample.regress.out se obtuvo de la entrega 1.01 de postgres construida por Jolly Chen y se incluye aquí para referencia. Tendría que haberse ejecutado con una máquina DEC ALPHA ya que el Makefile.global en la version 1.01 de postgres tiene PORTNAME=alpha.

## Archivos de comparación específicos de la plataforma

Como alguno de los tests producen resultados inherentes a la plataforma usada, hemos proporcionamos una forma para suplir los archivos de comparación específicos para cada plataforma. Frecuentemente se da la misma variación en múltiples plataformas; en vez de dar un archivo de comparación separado para cada plataforma, existe un archivo guía que define qué archivo de comparación usar. De forma que, para eliminar fallos tontos de una plataforma en particular, debe elegir o crear un

fichero de resultados variantes, y añadir una línea al archivo guía, que es "mapa de resultados".

Cada línea del archivo guía es de la siguiente forma

```
testname/platformnamepattern=comparisonfilename
```

El nombre del test (testname) es sencillamente el nombre del módulo de regresión de ese test en particular. El patrón del nombre de la plataforma (platformnamepattern) está generado al estilo de `expr(1)` (que es una expresión regular con el símbolo `^` implícito al principio). Esta se comprueba con el nombre de la plataforma tal como viene escrito en `config.guess`. El nombre del fichero de comparación (comparisonfilename) es el nombre del sustituto del fichero de resultados de comparación.

Por ejemplo: el test de regresión `int2` incluye una entrada deliberada de un valor que es demasiado largo para caber en un `int2`. El mensaje de error específico que es producido es dependiente de la plataforma; nuestra plataforma de referencia saca

```
ERROR: pg_atoi: error reading "100000": Numerical result out of range
```

pero en un buen número de otras plataformas Unix saca

```
ERROR: pg_atoi: error reading "100000": Result too large
```

En este caso, proporcionamos una variante del archivo de comparación, `int2-too-large.out`, que incluye la sintaxis de este mensaje de error. Para no mostrar estos "fallos" tontos en las plataformas HPPA, el `resultmap` (mapa de resultados) incluye

```
int2/hppa=int2-too-large
```

que se activará en cualquier máquina en el que la salida de `config.guess` comience por `'hppa'`. Otras líneas en el `resultmap` seleccionan la variante del archivo de comparación para otras plataformas donde sea apropiado.

## Capítulo 15. Notas de versiones

### Version 6.5.3

Esta es basicamente una limpieza de la version 6.5.2. Hemos añadido un nuevo pgacces que se perdio en la 6.5.2, e instalado una correccion especifica para NT.

### Migracion a v6.5.3

No se requiere un dump/restores para aquellos que esten ejecutando una 6.5.\*.

### Lista Detallada de Cambios

Version actualizada de pgacces 0.98  
Parche especifico para NT  
Correccion para reglas de volcado en tablas heredadas

### Version 6.5.2

Esta es basicamente una limpieza de la version 6.5.1. Hemos corregio una variedad de problemas reportados por usuarios de 6.5.1.

### Migracion to v6.5.2

No se requiere un dump/restores para aquellos que esten ejecutando una 6.5.\*.

### Lista Detallada de Cambios

corregidas las subselect+CASE (Tom)  
Añadida configuracion de SHLIB\_LINK para los portes de solaris\_i386 y solaris\_sparc(Daren Sefcik)  
Correcciones para CASE y WHERE en clausulas "join"(Tom)  
Correccion para aborto en BTScan(Tom)  
Reparada la comprobacion para UNIQUE redundante e indices PRIMARY KEY(Thomas)  
Mejorado para que se compruebe las restricciones en multi-columns(Thomas)  
Correccion para Win32 que tenia problemas con MB habilitado(Hiroki Kataoka)  
Permite a yacc de BSD y a bison compilar codigo pl(Bruce)  
Corregido el trabajo con SET NAMES  
corregidos los int8 (Thomas)  
Correccion del consumo de memoria de "vacuum"(Hiroshi,Tatsuo)  
Reduccion del consumo total de memoria de "vacuum"(Tom)  
Correccion para timestamp(datetime)  
Correcciones de problemas en las reglas de paso al analizador sintactico(Tom)  
Correccion del problema de cuotas en mkMakefile.tcldefs.sh.in y mkMakefile.tkdefs.sh.in  
This is to re-use space on index pages freed by vacuum(Vadim)  
documentado -x para pg\_dump(Bruce)  
Correcin para operadores unarios en la regla de paso al analizador sintactico(Tom)  
Comentado el FileUnlink de exceso de segmentos durante mdtruncate() (Tom)  
Enlazamiento en Irix corregido por Yu Cao >yucao@falcon.kla-tencor.com<

Reparado el error logico en LIKE: no debia devolver un LIKE\_ABORT cuando alcanza el final de un patron antes del final del texto(Tom)  
 Reparada limpieza incorrecta de montones de memoria reservadas durante aborto de transaccion(Tom)  
 Version actualizada de pgaccess 0.98

## Version 6.5.1

Esta es basicamente una limpieza de la version 6.5. Hemos corregio una variedad de problemas reportados por usuarios de 6.5.

## Migracion to v6.5.1

No se requiere un dump/restores para aquellos que esten ejecutando una 6.5.

## Lista Detallada de Cambios

Añadido un fichero NT LEAME  
 Correcciones de portabilidad para linux\_ppc, Irix, linux\_alpha, OpenBSD, alpha  
 Eliminado QUERY\_LIMIT, utilizar SELECT...LIMIT  
 Correccion para EXPLAIN en herencia(Tom)  
 Parche para permitir "vacuum" en tablas con multi-segmentos(Hirosi)  
 Correccion para la selectividad del optimizador R-Tree(Tom)  
 Corregida laguna el descriptor de ficheros ACL(Atsushi Ogawa)  
 Nueva expresion del codigo de sub arboles(Tom)  
 Se evitan escrituras en disco para transacciones de solo-lectura(Vadim)  
 Correccion para eliminacion de tablas temporales si la ultima transaccion fue abortada (Bruce)  
 Correccion para prevenir que sean creadas tuplas demasiado largas(Bruce)  
 correcciones en plpgsql  
 Se permiten numeros de puerto de 32k - 64k(Bruce)  
 Añadido ^ precedence(Bruce)  
 Renombrados ficheros ordendo llamados pg\_temp a pg\_sorttemp(Bruce)  
 Correccion para microsegundos en valores temporales(Tom)  
 Limpieza de la fuente del Tutorial  
 Nuevo porte a linux\_m68k  
 Correccion para la ordenacion de NULL's en algunos casos(Tom)  
 Corregidas las dependencias para librerias compartidas(Tom)  
 Corregido fallos tecnicos que afectaban a GROUP BY en subselects(Tom)  
 Correccion para algunas alarmas del compilador (Tomoaki Nishiyama)  
 Añadido soporte para Win1250 (Checo) (Pavel Behal)

## Version 6.5

Esta version marca un avance grande en el conocimiento que el equipo de desarrollo tiene del codigo fuente que heredamos de Berkeley. Veras que podemos añadir mayores features mas facilmente, gracias al incremento en tamaño y experiencia de nuestro mundialmente extenso equipo de desarrollo.

He aquí un conciso resumen de los más notables cambios:

#### Control de concurrencia multi-version(MVCC en inglés Multi-version concurrency control)

Esto elimina nuestro viejo bloqueo a nivel de tabla, y lo reemplaza con un bloqueo del sistema que es superior a la mayoría de los sistemas de bases de datos comerciales. En un sistema tradicional, cada registro que es modificado se bloquea hasta que se confirma la transacción, previniendo lecturas por otros usuarios. MVCC utiliza de modo natural el carácter multi-version de PostgreSQL para permitir que las lecturas continúen leyendo datos consistentes durante la actividad de escritura. Las escrituras continúan utilizando el sistema de transacciones compacto `pg_log`. Todo esto se realiza sin tener que reservar un bloqueo para cada registro como en los sistemas tradicionales de base de datos. Así que, básicamente, ya no estaremos restringidos por más tiempo por el bloqueo simple a nivel de tabla; tenemos algo mejor que el bloqueo a nivel de registro.

#### Copias de seguridad en caliente con `pg_dump`

`pg_dump` se beneficia de las nuevas prestaciones de MVCC para dar consistencia a un `dump/backup` mientras la base de datos permanece en línea y disponible para ser consultada.

#### Tipos de datos numéricos

Ahora tenemos tipos de datos verdaderamente numéricos, con precisión especificada por el usuario.

#### Tablas temporales

Se garantiza que las tablas temporales tienen nombres únicos durante una sesión en la base de datos, y que son destruidas al salir de la sesión.

#### Nuevas prestaciones SQL

Ahora tenemos soporte para declaraciones `CASE`, `INTERSECT`, and `EXCEPT`. Tenemos nuevos `LIMIT/OFFSET`, `SET TRANSACTION ISOLATION LEVEL`, `SELECT ... FOR UPDATE`, y un mejorado comando `LOCK TABLE`.

#### Aceleramiento

Continuamos acelerando PostgreSQL, gracias a la variedad de talentos que hay en nuestro equipo. Hemos acelerado la asignación de memoria, la optimización, las uniones de tablas (`table join`), y las rutinas de transferencias de registros.

#### Portes

Continuamos ampliando nuestra lista de portes, esta vez incluimos WinNT/ix86 y NetBSD/arm32.

#### Interfaces

La mayoría de interfaces tienen una nueva versión, y la funcionalidad existente ha sido mejorada.

#### Documentación

Nuevo y actualizado material está presente por toda la documentación. Se han aportado nuevas FAQs para las plataformas SGI y AIX. El *Tutorial* tiene información introductoria sobre SQL de Stefan Simkovics. Para la *Guía del Usuario*, hay páginas de referencia cubriendo la utilidad `postmaster` y más programas de

utilidad, un apendice nuevo contiene detalles sobre el comportamiento de `date/time`. La *Guia del Administrador* tiene un nuevo capitulo sobre resolucion de problemas de Tom Lane. Y la *Guia del Programador* tiene una descripcion del proceso de interrogacion, tambien de Stefan, y detalles acerca de como obtener el arbol del codigo fuente de Postgres por CVS anonimo y CVSup.

## Migracion to v6.5

Un `dump/restore` utilizando `pg_dump` es necesario para aquellos que deseen migrar datos de cualquier version previa de Postgres. `pg_upgrade` *not* puede ser utilizado para actualizar esta version porque la estructura en disco de las tablas ha cambiado comparada con versiones precedentes.

La nueva caracteristica de Control de Concurrencia Multi-Version (MVCC, en ingles) puede dar comportamientos un poco diferentes en entornos multiusuarios. *Lea y comprenda la seccion siguiente para asegurar que sus aplicaciones existentes le proporcionaran el comportamiento que necesita.*

### Control de Concurrencia Multi-Version

A causa de que las lecturas en 6.5 no bloquean los datos, a pesar del nivel de aislamiento de transaccion, los datos leidos por una transaccion pueden ser sobre escritos por otra. En otras palabras, si un registro es devuelto por **SELECT** eso no significa que este registro exista realmente en el momento en que es devuelto. (i.e algunas veces despues de que la sentencia o la transaccion comience) ni tampoco que el registro este protegido de ser borrado o actualizado por una transaccion en concurrente antes de que la transccion en curso haga commit o rollback.

Para asegurar la existencia actual de un registro y protegerlo contra actualizaciones concurrentes se debe utilizar **SELECT FOR UPDATE** o una sentencia **LOCK TABLE** apropiada. Esto deberia ser tenido en cuenta cuando se porten aplicaciones desde versiones precedentes de Postgres y otros entornos.

Tenga todo lo anterior en mente su utiliza triggers `contrib/refint.*` para integridad referencial. Ahora se requieren tecnicas adicionales. Un modo es utilizar el comando **LOCK parent\_table IN SHARE ROW EXCLUSIVE MODE** si una transaccion va a actualizar/borrar una clave primaria y utilizar el comando **LOCK parent\_table IN SHARE MODE** si una transaccion va a actualizar/insertar una clave foranea.

**Nota:** Notese que si ejecuta una transaccion en modo **SERIALIZABLE** entonces debe ejecutar el comando **LOCK** anterior antes de la ejecucion de cualquier sentencia DML (**SELECT/INSERT/DELETE/UPDATE/FETCH/COPY\_TO**) en la transaccion.

Esto inconvenientes desapareceran en el futuro cuando la habilidad para leer datos sucios (no confirmados) (a pesar del nivel de aislamiento) y la verdadera integridad referencial sea implementada.

## Lista Detalla de Cambios

Correcciones de errores

-----

Correccion de las funciones de conversion text<->float8 y text<->float4(Thomas)  
 Correccion para creacion de tablas con constraints con mixed-case(Billy)  
 Cambiado comportamiento de exp()/pow() para generar error en underflow/overflow(Jan)  
 Correccion de error en pg\_dump -z  
 Limpiezas de invasiones de memoria(Tatsuo)  
 Correccion para abortos de lo\_import(Tatsuo)  
 Ajustes en el manejo de nombres de tipo de datos para suprimir dobles comillas(Thomas)  
 Uso de coercion de tipo para emparejar columnas y DEFAULT(Thomas)  
 Correccion de deadlock de este modo solo verifica una vez despues de un segundo de espera(Bruce)  
 Correcciones para agregaciones y PL/pgsql(Hiroshi)  
 Correccion para aborto de subquery(Vadim)  
 Correccion de libpq para la funcion PQfnnumber y nombres que no distinguen mayusculas-minusculas(Bahman Rafatjoo)  
 Correccion para objetos grandes escritos-en-medio, no bloques extra, consumo de memoria(Tatsuo)  
 Correccion para pg\_dump -d o -D y entrecomillado de caracteres especiales en INSERT  
 Reparados serios problemas con dynahash(Tom)  
 Correjidos problemas de portabilidad para INET/CIDR  
 Correccion de problema con error de selectividad en ALTER TABLE ADD COLUMN(Bruce)  
 Correccion del ejecutor de ese modo trabaja merzejoin de diferentes tipos de columnas(Tom)  
 Correccion de error para selectividad de OR en Alpha  
 Correccion para problema de selectividad de indice OR(Bruce)  
 Correccion \d para que muestre de ese modo la extension apropiada para char()/varchar()(I  
 Correccion en el codigo del tutorial(Clark)  
 Mejoras en la comprobacion de destroyuser(Oliver)  
 Correccion para Kerberos(Rodney McDuff)  
 Correccion para borrado de la base de datos mientras los buffers no se han limpiado(Bruce)  
 Correccion la secuencia nextval() para que pueda asi distinguir mayusculas-minusculas  
 Correccion para operador !=  
 Borrado de buffers antes de destruir los ficheros de las base de datos(Bruce)  
 Correccion del caso en que el ejecutor evalua las funciones dos veces(Tatsuo)  
 Se permite a las acciones de secuencia netxval distinguir mayusculas-minusculas(Bruce)  
 Correccion de optimizador de indexacion para que no trabaje para numeros negativos(Bruce)  
 Correccion de perdidas de memoria en ejecuciones con fjisNull  
 Correccion de perdidas de memoria para aggregate(Erik Riedel)  
 Se permite que username contenga una almohadilla (#, dash en ingles) en los permisos GRANT  
 Limpieza de NULL en los tipos inet  
 Limpieza de errores en tablas del sistema(Tom)  
 Correccion de problemas de PAGER y del comando \?(Masaaki Sakaida)  
 Reducido el tamaño de fichero de multi-segment por defecto a 1GB(Peter)  
 Correccion del volcado de CREATE OPERATOR(Tom)  
 Correccion para escaneo hacia atras de cursores(Hiroshi Inoue)  
 Correccion para COPY FROM STDIN cuando se utiliza \i(Tom)  
 Correccion para una subselect cuando es comparada dentro de una expresion(Jan)  
 Correccion para manejo de devolucion de error mientras se duelen registros(Tom)  
 Correccion de problemas con referencia a tipos de array(Tom,Jan)  
 Se previene oid de UPDATE SET (Jan)  
 Correccion de pg\_dump asi ña opcion -t puede manejar nombres de tabla en mayusculas-minusculas  
 Correcciones para GROUP BY es casos especiales(Tom, Jan)  
 Correccion de perdidas de memoria en queries falladas(Tom)  
 DEFAULT soporta ahora identificadores mixed-case(Tom)  
 Correccion para que multi-segment utilice DROP/RENAME de tabla, de indice(Ole Gjerde)  
 Dehabilitacion de uso de pg\_dump con las dos opciones -o y -d(Bruce)

Se permite a pg\_dump volver adecuadamente permisos de GROUP(Bruce)  
 Correccion para GROYP BY en INSERT INTO table SELECT \* FROM table2(Jan)  
 Correccion para computaciones en vistas(Jan)  
 Correcciones para agregaciones en array con indices(Tom)  
 Correccion para como maneja DEFAULT entrecomillado simple en valores que requieren demasiadas comillas  
 Correccion de problema de seguridad con no super-usuarios que importan/exportan objetos de gran tamaño.(Tom)  
 Vuelta atras de transaccion que crea table la limpia adecuadamente(Tom)  
 Correccion para permitir que tablas largas y nombres de columnas generen nombres en serie adecuados(Tom)

Mejoras  
 -----

Añadida la utilidad "vacuumdb"  
 Se acelera libpq por mejor asignacion de memoria(Tom)  
 EXPLAIN utiliza todos los indices(Tom)  
 Implementadas las expresiones CASE, COALESCE, NULLIF(Thomas)  
 Nuevo formato de salida de pg\_dump(Constantin)  
 Añadida la cadena min()/max() a las funciones(Thomas)  
 Extendidas nuevo tipo de coersiones para agregaciones(Thomas)  
 Nueva contribucion moddatetime (Terry)  
 Actualizacion a pgaccess 0.96(Constantin)  
 Añadida rutina para byte unico en tipo de caracter "char"(Thomas)  
 Mejorada la funcion substr()(Thomas)  
 Mejorado el manejo de multi-byte (Tatsuo)  
 Control de concurrencia Multi-version /MVCC(Vadim)  
 Nuevo modo Serialized(Vadim)  
 Correccion para tablas por encima de 2gigs(Peter)  
 Nuevo SET TRANSACTION ISOLATION LEVEL(Vadim)  
 Nuevo LOCK TABLE IN ... MODE(Vadim)  
 Actualizado el driver ODBC(Byron)  
 Nuevo tipo de datos NUMERIC(Jan)  
 Nueva SELECT FOR UPDATE(Vadim)  
 Manejo de "NaN" e "Infinity" para valores de entrada(Jan)  
 Mejorado el manejo de date/year(Thomas)  
 Mejorado el manejo de conexiones con el motor de base de datos(backend)(Magnus)  
 Nuevas opciones ELOG\_TIMESTAMPS y la opcion USE\_SYSLOG para ficheros de registro(Massimo)  
 Nueva opcion TCL\_ARRAYS (Massimo)  
 Nueva INTERSECT y EXCEPT(Stefan)  
 Nuevo pg\_index.indisprimary para restro de claves primarias(D'Arcy)  
 Nuevas opcion pg\_dump para permitir el borrado de tablas antes de su creacion(Brook)  
 Acelaracion de las rutinas de salida de registro(Tom)  
 Nuevo nivel de aislamiento de READ COMMITTED (Vadim)  
 Nuevas tablas/indices TEMP (Bruce)  
 Se evita el ordenamiento si el resultado ya esta ordenado(Jan)  
 Nueva optimizacin para la asignacion de memoria(Jan)  
 Se permite a psql ejecutar \p\g(Bruce)  
 Se permiten multiples reglas de acciones(Jan)  
 Añadida funcionalidad LIMIT/OFFSET (Jan)  
 Mejorado el optimizador cuando se unen un numero grande de tablas(Bruce)  
 Nueva introduccion a SQL de La Tesis de Doctorado de S. Simkovics(Stefan, Thomas)  
 Nueva introduccion a procesamiento del motor de base de datos (backend) de la Tesis de Doctorado de S. Simkovics(Stefan)  
 Mejorado el soporte para int8(Ryan Bradetich, Thomas, Tom)  
 Nuevas rutinas para convertir entre tipos int8 y text/varchar(Thomas)  
 Nuevos planes arboreos, donde se unen meta-tablas(Bruce)  
 Habilitadas consultas por la mano derecha por defecto(Bruce)  
 Se permite que el numero maximo de procesos en servidor (backends) se parametrice en el momento de la configuracion  
     (-with-maxbackends and postmaster switch (-N backends))(Tom)  
 GEQO por defecto tenga ahora 10 tablas porque el optimizador se acelera(Tom)  
 Se permite NULL=Var para MS-SQL portabilidad(Michael, Bruce)



Modificados contrib check\_primary\_key() y consecuentemente "automatic" or "dependent"(Ar)  
 Se permite que psql \d en una vista muestre la consulta(Ryan)  
 Se acelera el LIKE(Bruce)  
 Correcciones/prestaciones EcpG , vease fichero src/interfaces/ecpg/ChangeLog(Michael)  
 Correcciones/prestaciones JDBC , vease src/interfaces/jdbc/CHANGELOG(Peter)  
 Se hace que el operador % tenga precedencia como /(Bruce)  
 Añadida la nueva opcion postgres -O para permitir cambios en la estructura de tablas del sistema(Bruce)  
 Actualizado el script contrib/pginterface/findoidjoins (Tom)  
 Mayor aceleracion en vacuum de lineas borradas con indices(Vadim)  
 Se permite la ejecucion de diferentes versiones de funciones no-SQL basadas en argumentos(Tom)  
 Añadida la opcion -E que muestra las consultas actuales enviadas pro \dt y sus amigos(Masaaki Sakaida)  
 Añadido numero de version en los banners de arranque de psql(Masaaki Sakaida)  
 Nuevo contrib/vacuumlo que elimina objetos grandes no referenciados(Peter)  
 Nueva inicializacion para tamaños de tablas, asi las tablas no vacuumadas se ejecutan mejor(Tom)  
 Mejorados los mensajes de error cuando una conexion es rechazada(Tom)  
 Soporte para array de campos char() y varchar() (Massimo)  
 Revision del codigo has para incrementar fiabilidad y prestaciones(Tom)  
 Actualizacion a PyGreSQL 2.4(D'Arcy)  
 Cambiadas las opciones de depuracion asi -d4 y -d5 producen diferentes displays de nodos(Jan)  
 nuevas opciones: pretty\_plan, pretty\_parse, pretty\_rewritten(Jan)  
 Mejor optimizacion de estadisticas para los accesos a tablas del sistema(Tom)  
 Mejor manejo de tamaño de bloque no por defecto(Massimo)  
 Mejorado el optimizador de consumo de memoria GEQO(Tom)  
 UNION soporta ahora ORDER BY de columnas que no estan en la lista de target(Jan)  
 Mejoras grandes en libpq++ (Vince Vielhaber)  
 pg\_dump utiliza ahora -z(ACL's) por omision(Bruce)  
 cache de procesos en servidor (backend), aceleracion de memoria(Tom)  
 Se hace que pg\_dump lo haga todo en una transaccion snapshot(Vadim)  
 correccion de perdidas de memoria para objetos grandes, correccion para pg\_dumping(Tom)  
 INET escribe ahora respecto a la netmask para comparaciones  
 Se hace que VACUUM ANALYZE solo utilice un readlock(Vadim)  
 Se permiten vistas (VIEW) en UNIONS(Jan)  
 pg\_dump puede generar ahora snapshots consistentes en bases de datos activas(Vadim)

#### Cambios en el Arbol Fuente

-----

Mejorado el emparejamiento en el porte(Tom)  
 Correcciones de portabilidad para SunOS  
 Añadido porte para NT/Win32 del proceso en el servidor (backend) y se habilita carga dinamica(Magnus and Daniel Horak)  
 Nuevo porte a Cobalt Qube(Mips) ejecutando Linux(Tatsuo)  
 Porte a NetBSD/m68k(Mr. Mutsuki Nakajima)  
 Porte a NetBSD/sun3(Mr. Mutsuki Nakajima)  
 Porte a NetBSD/macppc(Toshimi Aoki)  
 Correccion para configuracion de tcl/tk(Vince)  
 Eliminada la clave CURRENT para consultas por regla(Jan)  
 carga dinamica en NT ahora funciona(Daniel Horak)  
 Añadido soporte para ARM32(Andrew McMurry)  
 Mejor soporte para HP/UX 11 y Unixware  
 Mejorado el manejo de ficheros para ser mas uniforme, previene lagunas en los descriptors de ficheros.(Tom)  
 Nuevos comandos de instalacion para plpgsql(Jan)

## Version 6.4.2

La version 6.4.1 fue incorrectamente empaquetada. Esta tiene tambien una correccion adicional para un bug

## Migracion a v6.4.2

No se requiere un dump/restores para aquellos que esten ejecutando una 6.4.\*.

## Lista Detallada de Cambios

Correccion para problema de constante de fecha y hora en algunas plataformas (Thomas)

## Version 6.4.1

Esta es basicamente una limpieza de la version 6.4. Hemos corregio una variedad de problemas reportados por usuarios de 6.4

## Migracion a v6.4.1

No se requiere un dump/restores para aquellos que esten ejecutando una 6.4.

## Lista Detallada de Cambios

Añadida la opcion -N a pg\_dump para forzar dobles comillas alrededor de los identificadores. Este es la opcion por omision (Thomas)  
Correccion para NOT in donde la clausula causa abortos (Bruce)  
Correccion para el coredump de EXPLAIN VERBOSE (Vadim)  
Correccion para problemas de shared-library en Linux  
Correccion del test de existencia de tabla para permitir combinacion de mayusculas y minusculas y  
espacios en blanco en el nombre de tabla (Thomas)  
Fijacion de un par de problemas de pg\_dump  
La configuracion empareja mejor entradas similares en la platilla template/. (Tom)  
Cambios en la construccion de los nombre de la funcion de SPI\_\* a spi\_\*  
Correccion para la clausula OR WHERE (Vadim)  
Correcciones para nombres de tablas con mayusculas y minusculas (Billy)  
Correccion para contrib/linux/postgres.init.csh/sh (Thomas)  
rebasamiento de memoria en libpq corregido  
Correcciones para SunOS (Tom)  
Se cambia el comportamiento de exp() para generar error en negativos (Thomas)  
pg\_dump fixes for memory leak, inheritance constraints, layout change  
Actualizado pgaccess a 0.93  
Se corrige el prototipo para plataformas de 64-bit  
Correcciones para Multi-byte (Tatsuo)  
Nueva pagina de manual ecpg  
Corregidos rebasamientos de memoria (Tatsuo)  
Correccion para fallos en lo\_import() (Bruce)  
Mejores busquedas para el programa de instalacion (Tom)

Correcciones para zona horaria(Tom)  
 Correcciones para HPUX(Tom)  
 Se utiliza coersiones de tipo implicito para emparejar valores DEFAULT(Thomas)  
 Anadidas rutinas para ayudar a tipo de caracter bytes-solo(single-byte) (internal)(Thomas)  
 Correcciones para compilacion de libpq en Win32(Magnus)  
 Actualizacion a PyGreSQL 2.2(D'Arcy)

## Version 6.4

Hay *muchas* prestaciones nuevas y mejoras en esta version. Gracias a unestros desarrolladores y mantenedores, casi todos los aspectos del sistema han recibido alguna atencion desde la version anterior. He aqui un resumen, sumario incompleto:

- Las vistas y las reglas son ahora funcionales gracias al extensivo nuevo codigo en las reglas de re escritura de Jan Wieck. Tambien escribio un capitulo sobre ello en la *Guia del Programador*.
- Jan tambien contribuyo al segundo lenguaje procedural, PL/pgSQL, para ir con el original lenguaje procedural PL/pgTCL con el que el contribuyo la ultima version.
- Tenemos soporte opcional de caracter multiple-byte de Tatsuo Iisho para complementar nuestro soporte local.
- Las comunicaciones cliente/servidor han sido depuradas, con mejor soporte para mensajes asincronos e interrupciones, gracias a Tom Lane.
- El depurador de sintactico ejecuta ahora coersiones de tipo automatico para emparejar argumentos a los operadores y funciones disponibles, y para emparejar columnas y expresiones con columnas destino. Esto utiliza un mecanismo generico que soporta las prestaciones de extensibilidad de tipo de Postgres. Hay un nuevo capitulo en la *Guia de Usuario* que cubre este asunto.
- Tres nuevos tipos de datos han sido anadidos. Dos tipos, inet y cidr, soportan varias formas de trabajo en red IP, subred, y direccionamiento por maquina. Ahora hay un tipo entero de 8 byte disponible para algunas plataformas. Vease el capitulo de tipos de datos en la *Guia del Usuario* para mas detalles. Un cuarto tipo, serial, se soporta ahora por el depurador de sintactico como una amalgama de tipo int4, una secuencia, y un indice unico.
- Han sido añadidas varias prestaciones mas sintacticas compatibles con SQL92, incluyendo **INSERT DEFAULT VALUES** Several more SQL92-compatible syntax features have been added, including **INSERT DEFAULT VALUES**
- La instalacion y configuracion automatica del sistema ha recibido alguna atencion, y deberia ser mas robusta para mas plataformas de lo que nunca ha sido.

## Migracion a v6.4

Se requiere un dump/restore utilizando `pg_dump` o `pg_dumpall` para aquellas que desen migrar datos desde cualquier version anterior de Postgres.

## Lista Detallada de Cambios

### Correcciones de errores

-----

Correccion para una minuscula perdida en PQsetdb/PQfinish(Bryan)  
 Se elimina char2-16 de los tipos de datos, se utiliza char/varchar(Darren)  
 Pqfn no maneja un mensaje de NOTICE(Anders)  
 Reducidas elevadas esperas por ocupacion a causa de bloqueos en transacciones con muchos procesos en servidor(backends) (dg)  
 Deteccion de bloqueos de transacciones atascadas (dg)  
 Correccion para masrcas de tiempo en estilo "ISO" en decodificacion y codificacion(Thomas)  
 Correccion del problema con borrado de tabla (drop) despues de deshacer (rollback) una transaccion(Vadin)  
 Cambiado mensaje de error y eliminado mensaje actualizado no funcional(Vadim)  
 Correccion para verificacion de la matriz (array) de COPY  
 Correccion para SELECT 1 UNION SELECT NULL  
 Correccion para perdidas de buffer en llamadas a objetos grandes(Pascal)  
 Cambio de propietario de tipo oid a int4(Bruce)  
 Correccion de error en la compatibilidad con oracle de las funciones btrim() ltrim() y rtrim()  
 Correccion de invalidacion en rebasamientos de cache compartida(Massimo)  
 Prevencion de perdidas en descriptores de ficheros en COPY's fallidos(Bruce)  
 Correccion para perdidas en la pg\_select de libpgtcl(Constantin)  
 Correccion de problemas con usuario/contrasena de mas de 8 caracteres(Tom)  
 Correccion de problemas con manejo de NOTIFY asincronos en el proceso en servidor(backend)  
 Correccion de muchas entradas de sistema malas(Tom)

### Mejoras

-----

Actualizacion de ecpg y ecpglib, vease src/interfaces/ecpg/ChangeLog(Michael)  
 Se muestra en indice utilizado en un EXPLAIN(Zeugswette)  
 EXPLAIN invoca una regla de sistema y muestra plan(es) para la reescritura de consultas(Jan)  
 Conocimiento multi-byte de muchos tipos de datos y funciones, via configure(Tatsuo)  
 Nuevo configure con la opcion -with-mb(Tatsuo)  
 Nueva opcion initdb -pgencoding(Tatsuo)  
 Nueva opcion createdb -E multibyte(Tatsuo)  
 Select version(); ahora devuelve la version de PostgreSQL(Jeroen)  
 Libpq permite ahora clientes asincronos(Tom)  
 Se permite la cancelacion desde el cliente de una consulta en el proceso en servidor(backend)(Tom)  
 Psql cancelas las consultas ahora con Control-C(Tom)  
 Usuarios de Libpq no necesitan dar consultas dummy para obtener mensajes NOTIFY(Tom)  
 NOTIFY envia ahora al PID del emisor, asi que puedes decir si eras tu mismo(Tom)  
 La estructura de PGresult ahora incluye un mensaje de error asociado, si lo hay(Tom)  
 Se definen los argumentos "tz\_hour" y "tz\_minute" como date\_part()(Thomas)  
 Se anaden rutinas para convertir entre varchar y bpchar(Thomas)  
 Se anaden rutinas para permitir el dimensionamiento de varchar y bpchar dentro de las columnas de destino(Thomas)  
 Se anade un bit a las etiquetas (flags) oara soportar zona horaria y minutos en la devolucion de fecha(Thomas)  
 Se permiten mas variaciones en numeros de coma flotante (por ej. ".1", "1e6")(Thomas)  
 Se corrigen el analisis sintactico de menores unarios empezando con espacios(Thomas)  
 Se implementa TIMEZONE\_HOUR, TIMEZONE\_MINUTE por especificaciones SQL92(Thomas)  
 Se verifica i se ignora adecuadamente constraints de columna FOREIGN KEY(Thomas)  
 SE defina USER como sinonimo de CURRENT\_USER por especificacines de SQL92(Thomas)  
 Se habilita HAVING en clausulas pero no se corrije en ningun otro lugar aun.  
 Se hace el tipo "char" un sinonimo de "char(1)" (actualmente implementado como bpchar)(Thomas)  
 Se guarda el tipo de cadena si esta especificado por el manejo de la clausula DEFAULT(Thomas)

Operaciones de coercion abarcan diferentes tipos de datos(Thomas)  
 Se permite a algunos indices utilizar columnas de diferentes tipos(Thomas)  
 Anadidas capacidades para coersiones de tipo automatico(Thomas)  
 Depuraciones para objetos grandes, de este modo un fichero es truncado en su apertura(Peter)  
 Depuraciones de la lectura de lineas(Tom)  
 Se permite que psql \f \ tomen los espacios en blanco como delimitadores(Bruce)  
 Se pasa el pg\_attribute.atttypmod al frontal de la aplicacion para las longitudes de los campos(Tom,Bruce)  
 Libreria de compatibilidad con Msql en /contrib(Aldrin)  
 Se elimina el requerimiento de que las clausulas identificadoras ORDER/GROUP BY estuvieran incluidas en la lista de la busqueda(David)  
 Se convierten columnas para emparejarlas en las clausulas de UNION(Thomas)  
 Se elimina fork()/ecec() y solo se hace fork()(Bruce)  
 Depuraciones en Jdbc(Peter)  
 Se muestra el estado del proceso en el servidor (backend) en la linea de comandos de ps(solo funciona en algunas plataformas)(Bruce)  
 Pg\_hba.conf tiene ahora una opcion sameuser en el campo de la base de datos  
 Se hace que lo\_unlink tome el parametro oid, no el int4  
 Nueva DISABLE\_COMPLEX\_MACRO para compiladores que no pueden manejar nuestras macros(Bruce)  
 Libpgtcl maneja los NOTIFY ahora como un evento Tcl, no se necesitan enviar consultas tontas(Tom)  
 Depuraciones en libpgtcl(Tom)  
 Anadida una opcion -error al comando pg\_result de libpgtcl(Tom)  
 Anadido parche locale, vease docs/README/locale(Oleg)  
 Correccion para pg\_dump con ella la sintaxis de CONSTRAINT y CHECK es correcta(ccb)  
 Nuevo codigo contrib/lo para eliminar grandes objetos huérfanos(Peter)  
 Nuevp comando psql "SET CLIENT\_ENCODING TO 'encoding'" para prestaciones multi.byte, vease /doc/README.mb(Tatsuo)  
 codigo /contrib/noupdate para revocar permisos de actualizacion en una columna  
 Libpq puede ser compilada ahora en win32(Magnus)  
 Anadido PQsetdbLogin() en libpq  
 Nuevo tipo entero 8-byte, comprobado por el configure del soporte para OS(Thomas)  
 Mejor soporte para nombres entrecomillados de tabla/columnas(Thomas)  
 Se rodean los nombres de tabla y columnas con dobles comillas en pg\_dump(Thomas)  
 PQreset() trabaja ahora con contraseñas(Tom)  
 Handle case of GROUP BY target list column number out of range(David)  
 Se permite UNION en las subconsultas  
 Anadido auto-dimensionamiento a la pantalla a los comandos \d?(Bruce)  
 Se utiliza UNION para mostrar todos los resultados de \d? en una consulta(Bruce)  
 Se anade la prestacion de busqueda de campo \d?(Bruce)  
 Pg\_dump utiliza menos peticiones \connect(Tom)  
 Se hace que la opcion -z de pg\_dump trabaje mejor, se documenta en la pagina de manual(Tom)  
 Se anade la clausula HAVING con total soporte para subconsultas y uniones(Stephan)  
 Texto completo de las rutinas de indexado en contrib/fulltextindex(Maarten)  
 Los ids de transacciones se almacenan ahora en memoria compartida(Vadim)  
 Nuevo PGCLIENTENCODING cuando ejecutan el comando COPY(Tatsuo)  
 Soporte para la sintaxis SQL92 "SET NAMES"(Tatsuo)  
 Soporte para LATIN2-5(Tatsuo)  
 Anadido el caso UNICODE los test de refresion(Tatsuo)  
 Depuracion de gestor de bloqueos, nuevos modos de bloqueos para LLL(Vadim)  
 Se permite el uso de indice en clausulas OR(Bruce)  
 Se permite "SELECT NULL ORDER BY 1;"  
 La explicacion VERBOSE del plan lo imprime, y ahora imprime en bonito el plan al fichero de log del postmaster(Bruce)  
 Se anaden indices al display para el comando \d(Bruce)  
 Se permite el GROUP BY en funciones(David)  
 Nuevo pg\_class.relkind para objetos grandes(Bruce)  
 Nuevo modo de enviar libpq mensajes NOTICE a diferentes localizaciones(Tom)  
 Nuevo comando de escritura \w para psql(Bruce)

Nuevo /contrib/findoidjoins escanea columnas oid para encontrar relaciones de union(Bruce)  
 Se permite que sean considerados indices compatibles binarios cuando se verifican indices validos para una clausula de restriccion conteniendo una constante(Thomas)  
 Nuevo codigo ISBN/ISSN en /contrib/isbn\_issn  
 Se permite NOT LIKE, IN, NOT IN, BETWEEN, y NOT BETWEEN constraint(Thomas)  
 Nuevo sistema de reescritura corrige muchos problemas con reglas y vistas(Jan)
 

- \* Reglas en trabajos relacionados
- \* Cualificaciones de eventos en trabajos de inserciones/actualizaciones/borrados
- \* Nueva variable OLD para referirse a CURRENT, CURRENT se elimina en un futuro
- \* Las reglas de actualizacion se pueden referir a NEW y OLD en la regla cualificacion/accion
- \* Inserciones/actualizaciones/borrados en vistas de trabajo
- \* Reglas multiples de accion se soportan ahora, rodeadas entre parentesis
- \* Usuarios normales pueden crear vistas/reglas en las tablas en las que tengan permisos de RULE
- \* Las reglas y las vistas heredan los permisos del creador
- \* No hay reglas a nivel de columna
- \* No hay reglas de ACTUALIZACION NUEVA/VIEJA (UPDATE NEW/OLD)
- \* Nuevas vistas de sistema pg\_tables, pg\_indexes, pg\_rules y pg\_views
- \* Solo se puede ejecutar una accion en las reglas de SELECT
- \* Re escritura total revisada, tal vez para la 6.5
- \* manejo de subselects
- \* manejo de agregaciones en vistas
- \* manejo de inserciones dentro de una seleccion desde una vista ahora funciona

 Los indices del sistema ahora son multi-clave(Bruce)  
 Los tipos Oidint2, oidint4, y oidname types se eliminan(Bruce)  
 Se utiliza cache del sistema para mas busquedas en tablas del sistema(Bruce)  
 Nueva lenguaje de programacion en el proceso en servidor(backend) PL/pgSQL en backend/p  
 El nuevo tipo de datos SERIAL, auto crea la secuencia/indice(Thomas)  
 Se permite la comprobacion de declaraciones sin recompilar(Massimo)  
 Mejoras en el bloqueo de usuario(Massimo)  
 Nuevo comando setval() para configurar el valor de una secuencia(Massimo)  
 Auto eliminacion del fichero de socket de unixsi no hay un postmaster ejecutandose(Massimo)  
 Paquete de trazo condicional(Massimo)  
 Nuevo comando UNLISTEN(Massimo)  
 Psql y libpq se compilan ahora bajo win32 utilizando win32.mak(Magnus)  
 Lo\_read ya no almacena rastros NULL (Bruce)  
 Los identificadores son truncados ahora internamente a 31 caracteres(Bruce)  
 Opciones de createuser estan disponibles ahora en la linea de comando  
 Anadido soporte para codigo de enteros de 64-bit, configuracion comprobada, tipos de int8(Thomas)  
 Se previene la perdida de un descriptor a causa de un COPY fallido(Bruce)  
 Nuevo comando pg\_upgrade(Bruce)  
 Directorios Updated /contrib (Massimo)  
 Nueva sentencia CREATE TABLE DEFAULT VALUES disponible(Thomas)  
 Nueva sentencia INSERT INTO TABLE DEFAULT VALUES disponible(Thomas)  
 Nueva prestacin DECLARE y FETCH(Thomas)  
 Estructuras internas de libpq ahora no se exportan (Tom)  
 Se permiten indices con mas de 8 claves(Bruce)  
 Se elimina el teclado ARCHIVE, que ya no se utiliza(Thomas)  
 La opcion -n de pg\_dump para suprimir comillas alrededor de los identificadores se deshabilita las columnas del sistema para las vistas(Jan)  
 nuevos tipos INET y CIDR para direcciones de red(TomH, Paul)  
 No mas comillas en las salidas de psql  
 pg\_dump ahora vuelca las vistas(Terry)  
 nuevo SET QUERY\_LIMIT(Tatsuo,Jan)

Cambios en el Arbol Fuente  
 -----  
 Limpieza de /contrib (Jun)

Enlazadas algunas pequenas funciones llamadas para cada registro(Bruce)  
 Inline some small functions called for every row(Bruce)  
 Correcciones paraAlpha/linux  
 Limpiezas para Hp/UX (Tom)  
 Test de regresion para Multi-byte (Soonmyung.)  
 Se elimina la opcion -disabled del configure  
 Se define PGDOC para que utilice POSTGRES DIR por defecto  
 Se hace la regresion opcional  
 Se eliminan corchetes extras en el codigo de pgindent(Bruce)  
 Se anade soporte para la libreria compartida bsdi(Bruce)  
 Nueva soporte para opcion de configuracion -without-CXX support(Brook)  
 Nueva FAQ\_CVS  
 Actualizado flowchart de proceso de servidor en tools/backend(backend)  
 Cambiado atttymod de int16 a int32(Bruce, Tom)  
 Se corrige Getrusage() para plataformas que no lo tienen(Tom)  
 Se anade PQconnectdb, PGUSER, PGPASSWORD a la pagina de manual de libpq  
 NS32K platform fixes(Phil Nelson, John Buller)  
 Correcciones para Sco 7/UnixWare 2.x (Billy,others)  
 Correccines para Sparc/Solaris 2.5 (Ryan)  
 Pgbuiltin.3 esta obsoleto, movido a los ficheros de documentacion(Thomas)  
 Aun mas documentacion(Thomas)  
 Soporte para Nexstep(Jacek)  
 Soporte para Aix (David)  
 Pagina de manual para pginterface(Bruce)  
 Todas las librerias compartidas tienen numero de version  
 Unidas todos los defines de las librerias compartidas de SO-especificos dentro de un unico fichero  
 Comprobacion de configuracion TCL/Tk mas inteligente(Billy)  
 Configuracion de perl mas inteligente(Brook)  
 confdigure utiliza install-sh facilitado si no se encuentra script de instalacion(Tom)  
 nueva Makefile.shlib para configuracion de librerias compartidas(Tom)

## Version 6.3.2

Esta es una version de correccion de errores para 6.3.x. Consultese las notas de la version v6.3 para un sumario completo de las nuevas prestaciones.

Sumario:

- Se repara soporte para configuracion automatica para algunas plataformas, incluyendo Linux, a causa de fallos introducidos sin advertirlo en v6.3.1.
- Se manejan correctamente las llamadas a funciones en la parte izquierda de las clausulas BETWEEN y LIKE.

No se requiere un dump/restore para aquellos que esten ejecutando 6.3 o 6.3.1. Un 'make distclean', 'make', y 'make install' es todo lo requerido. Este ultimo paso deberia ser ejecutado mientras que la postmaster no este ejecutandose. Deberia re-enlazar (re-link) cualquier aplicacion cliente que utilice librerias Postgres.

Para actualizaciones desde instalaciones pre-v6.3, consultese las instrucciones de instalacion y migracion para v6.3

## Lista Detallada de Cambios

```

Cambios
-----
Mejoras en la deteccion del configure para tcl/tk(Brook Milligan, Alvin)
Mejoras en las paginas de manual(Bruce)
correcciones para BETWEEN y LIKE (Thomas)
correccion para psql \connect utilizado por pg_dump(Oliver Elphick)
Nuevo driver odbc
pgaccess, version 0.86
eliminado qsort, se utiliza ahora la version de libc, depuraciones(Jeroen)
correcciones para buffer excedidos que se han detectado(Maurice Gittens)
correcciones para buffer excedidos en libpgtcl(Randy Kunkee)
correccion para UNION con DISTICNT en ORDER BY(Bruce)
se comprueba en el configure el gettimeofday (Doug Winterburn)
Correccion para el error "indices no usados"(Vadim)
Adicciones a la documentcion(Thomas)
Correcciones para perdida de memoria en el proceso en el servidor(backend)(Bruce)
limpiezas en libreadline (Erwan MAS)
Eliminado DISTDIR(Bruce)
limpieza de las dependencias de Makefile (Jeroen van Vianen)
correcciones para ASSERT (Bruce)

```

## Version 6.3.1

Sumario:

- Soporte adicional para conjuntos de caracteres multi-byte.
- Reparacion de la ordenacion de bytes para clientes y servidores de frontal mixto.
- Actualizaciones menores para permitir sintaxis SQL.
- Mejoras a la autodeteccion de la configuracion durante la instalacion.

No se requiere un dump/restore para aquellos que ejecuten 6.3. Un 'make distclean', 'make', y 'make install' es todo lo que se requiere. Deberia re-enlazar (re-link) cualquier aplicacion cliente que utilice librerias Postgres.

Para actualizaciones desde instalaciones pre-v6.3, consultese las instrucciones de instalacion y migracion para v6.3

## Lista Detallada de Cambios

```

Cambios
-----
depuracion/correcciones para ecpg, ahora en version 1.1(Michael Meskes)
depuracion en pg_user(Bruce)
correccion para objetos largos para pg_dump y tclsh(alvin)
correccion de LIKE para multiples subrayados adyacentes
correccion para redefiniciones en la construccion de funciones(Thomas)
depuraciones para ultrix4
actualizacion a pg_access 0.83
actualizacion de la pagina de manual CLUSTER

```



Soporte para juego de caracteres multi-byte, vease doc/README.mb(Tatsuo)  
 correccion para configure -with-pgport  
 correccion para pg\_ident  
 correccion de big-endian para comunicaciones en el proceso servidor(backend)(Kataoka)  
 correccion para SUBSTR() y substring() (Jan)  
 varias correcciones para jdbc (Peter)  
 libpgtcl improvements, see libpgtcl/README(Randy Kunkee)  
 Correccion del error "Datasize = 0" (Vadim)  
 Se previene la ocultacion de \do(Bruce)  
 Eliminadas entradas de juegos de caracteres Ruso duplicadas  
 depuracion para Sunos4  
 Se permiten claves opcionales de TABLA en LOCK y SELECT INTO(Thomas)  
 Opciones de CREATE SEQUENCE para permitir enteros negativos(Thomas)  
 Se anade "PASSWORD" como un identificador de columna valido(Thomas)  
 Se anade comprobacion de los campos objeto en UNION (Bruce)  
 Correccion para el porte a Alpha(Dwayne Bailey)  
 Correccion para matrices con texto contenido comillas(Doug Gibson)  
 Correccion para la compilacion en Solaris(Albert Chin-A-Young)  
 Se indentifica mejor librerias e includes de tcl y tk(Bruce)

## Version 6.3

Hay *muchas* prestaciones nuevas y mejoras en esta version. He aqui un breve, incompleto resumen:

- Muchas prestaciones nuevas, incluyendo capacitacion total para subconsultas SQL92 (todo esta aqui excepto la lista destino en la subconsulta)
- Soporte para variables de entorno del lado del cliente para especificar zona horaria y estilo de fecha.
- Interfaz de conexion (socket) para conexiones cliente/servidor. Esto es por ahora defecto asi que necesitas iniciar postmaster con la opcion "-i"
- Mejores mecanismos de autorizacion con contrasena. Los permisos de tabla por defecto han cambiado.
- El viejo estilo de "time travel" ha sido eliminado. Se han mejorado los resultados.

**Nota:** Bruce Momjian escribio las siguientes notas para presentar la nueva version.

Hay algunas cosas generales en la 6.3 que quiero mencionar. Son solo los grandes items que no pueden ser descritos en una frase. Se aun necesita una revision de la lista de cambios detallado.

Primero, ahora tenemos subconsultas. Ahora que las tenemos, quiero decir que sin subconsultas, SQL es un lenguaje muy limitado. Las subconsultas son una prestacion mayor, y deberia revisar su codigo en los lugares en los que una subconsulta le provea de una mejor solucion para sus consultas. Creo que encontrara que hay mas usos para las subconsultas de los que usted cree. Vadim nos ha puesto en el gran mapa de SQL con las subconsultas, y con ellas funcionales totalmente. La unica cosa que no puede hacer con las subconsultas es utilizarlas en lista a obtener(target list).

Segundo, la 6.3 utiliza conexión de dominio unix(unix domain sockets) en vez TCP/IP por defecto. Para permitir conexiones desde otras máquinas, tiene que utilizar la nueva opción `postmaster -i`, y por supuesto editar `pg_hba.conf`. También por esta razón el formato de `pg_hba.conf` ha cambiado.

Tercero, los campos `char()` permitirán ahora un acceso más rápido que `varchar()` o que campos de texto. Específicamente, el texto y el `varchar()` tienen una penalización para el acceso a cualquier columna después de la primera columna de este tipo. `char()` utilizó también este acceso penalizado, pero ya no existe por más tiempo. Esto puede sugerir que rediseñe algunas de sus tablas, específicamente si tiene columnas de caracteres cortos que haya definido como `varchar()` o texto. Este y otros cambios hacen la 6.3 aún más rápida que versiones precedentes.

Tenemos ahora contraseñas definibles independientemente de cualquier fichero Unix. Hay nuevos comandos SQL `USER`. Véase la página de manual `pg_hba.conf` para más información. Hay una tabla nueva, `pg_shadow`, que es utilizada para almacenar información de los usuarios y las contraseñas de los usuarios, y que es, por defecto solo consultable (SELECT-able) por el super usuario de postgres. `pg_user` es ahora una vista de `pg_shadow`, y es consultable (SELECT-able) por `PUBLIC`. Debe seguir utilizando `pg_user` en sus aplicaciones sin cambios.

Las tablas creadas por usuarios ahora ya no tienen permisos de consulta (SELECT) para `PUBLIC` por defecto. Esto se hizo porque los estándares ANSI lo requieren. Puede por supuesto conceder (GRANT) cualquier permiso que quiera después de que la tabla sea creada. Las tablas del sistema continúan siendo consultables (SELECT-able) por `PUBLIC`.

Tenemos también verdaderos códigos de detección de bloqueos (deadlock). No más sesenta segundos de tiempo de expiración (timeouts). Y el nuevo código implementa un FIFO mejor, así que debería haber un menor agotamiento de recursos durante un uso fuerte.

Muchas quejas han sido hechas acerca de la inadecuada documentación en versiones anteriores. Thomas ha puesto mucho esfuerzo en muchos manuales nuevos para esta versión. Compruebe en directorio `/doc`.

Por razones de rendimiento, `time travel` se elimina, pero puede ser implementado utilizando disparadores (triggers) (véase `pgsql/contrib/spi/README`). Por favor, compruebe el nuevo comando `\d` para tipos, operadores, etc. También las vistas tienen ahora sus propios permisos, no basados en las tablas subyacentes, así que los permisos para ellas deben ser configurados separadamente. Compruebe `pgsql/interfaces` para ver algunas maneras de dialogar con Postgres.

Esta es la primera versión que realmente requiere una explicación para los usuarios existentes. De muchas maneras, fue necesario esto porque la nueva versión elimina muchas limitaciones, y las soluciones provisionales (work-around) que eran la gente estaba utilizando ya no se necesitan.

## Migración a v6.3

Se requiere la utilización de un `pg_dump` o de un `pg_dumpall` para todos aquellos que deseen migrar datos de cualquier versión anterior de Postgres.

## Lista Detallada de Cambios

Corrección de errores

-----

Corrección de cursores binarios rotos por implementación de MOVE (Vadim)

Correccion para fallos de la libreria tcl(Jan)  
 Correccion para el manejo de matrices (arrays), por Gerhard Hintermayer  
 Correccion de error en acl, y se elimina pqtrace duplicado (Bruce)  
 Correccion para psql \e con fichero vacio (Bruce)  
 Correccion para texcat en campos varchar() (Bruce)  
 Correccion para DBT Sendproc (Zeugswetter Andres)  
 Correccion para problema de sintaxis en vacuum analyze(Bruce)  
 Correccion para indentificadores internacionales(Tatsuo)  
 Correccion para agregaciones en tablas heredadas (Bruce)  
 Correccion fr substr() para datos fuera de rango  
 Correccion para select 1=1 o 2=2, select 1=1 and 2=2, y selec sum (2+) (Bruce)  
 Correccion para mostrar estado de del resultado cuando no hay tty de salida. La opcion -q  
     aun no lo devuelve (Bruce)  
 Correccion para count(\*), argumentos con vistas y tablas multiples y sum(3)(Bruce)  
 Correccion para cluster (Bruce)  
 Correccion para PQtrace start/stop muchas veces (Bruce)  
 Correccion para una variedad de problemas como bloqueos más recientes en espera  
     que tienen bloqueo antes que procesos en espera mas viejos, y teniendo gente bloqueada  
     en lectura no se comparte bloqueo si proceso que escribe esta esperando para cojer el                      bloqueo, y los procesos escritores que esperan no obtienen prioridad sobre los  
     procesos lectores que esperan(Bruce)  
 Correccion para abortos en psql cuando se ejecutan consultas desde ficheros externos (James)  
 Correccion para el problema de multiples ordenaciones por columnas, cuando la primera  
     tiene valores NULL (Jeroen)  
 Utilización correcto soporte de funciones de la tabla hash para float8 e int4(Thomas)  
 Re-enable JOIN= option in CREATE OPERATOR statement (Thomas)  
 Se cambia la precedencia de los operadores booleanos para emparejarlos con el comportamiento esperado(Thomas)  
 Se genera elog(ERORR) en enteros que sobre-crecen (Bruce)  
 Se permite argumentos multiples en las funciones de clausulas de constraint(Thomas)  
 Se comprueba los literales booleanos de entrada para 'true', 'false', 'yes', 'no', '1', '0'  
     y se envian a elog(ERROR) si no se reconocen(Thomas)  
 Correccion para objetos de gran tamaño importante  
 Correccion para GROUP BY mostrando duplicados(Vadim)  
 Correccion para exploracion de indice en MergeJion(Vadim)

## Mejoras

-----

Sub consultas con las palabras claves EXISTS, IN, ALL y ANY (Vadim, Bruce, Thomas)  
 Nuevo Manual del Usuario(Thomas, others)  
 Aceleracion por alineamiento de algunas funciones frecuentemente llamadas  
 Deteccion real de bloqueos(deadlock), no mas expiraciones por tiempo(timeout)(Bruce)  
 se anaden las "constantes" SQL92 CURRENT\_DATE, CURRENT\_TIME, CURRENT\_TIMESTAMP, CURRENT\_USER(Thomas)  
 Se modifica la sintaxis para ser compatibles con SQL-92(Thomas)  
 Se implementa las clausulas SQL PRIMARY KEY y UNIQUE utilizando indices(Thomas)  
 Se reconoce la sintaxis SQL92 para CLAVES FORANEAS (FOREIGN KEYS). Se pone una  
     aviso (notice) en el elog (Thomas)  
 Se permite la clausula de restriccion NOT NULL UNIQUE (cada una de ellas permitidas separadamente antes)(Thomas)  
 Se permite el estilo de arrojar no constantes (::") Postgres (Thomas)  
 Se anade soporte para las constantes booleanas de SQL3 TRUE y FALSE (Thomas)  
 Se soporta la sintaxis SQL92 para IS TRUE/IS FALSE/IS NOT TRUE/IS NOT FALSE(Thomas)  
 Se permiten cadenas mas cortas para literales booleanos(por ej. "t", "tr", "tru")(Thomas)  
 Se permiten identificadores de delimitacion SQL92(Thomas)  
 Se implementa decodificacion para cadenas binarias y hexadecimales SQL92 (b'10' and x'11

Support SQL92 syntax for type coercion of literal strings  
 (e.g. "DATETIME 'now'")(Thomas)

Se añade conversiones para tipos int2, int4, y OID a y desde texto (Thomas)

Se utilizan bloqueos compartidos cuando se crean índices (Vadim)

La memoria libre reservada para una consulta de usuario dentro de un bloque de transacción  
 después de esa consulta sea hecha, se había desactivado <= 6.2.1(Vadim)

Nueva declaración SQL, CREATE PROCEDURAL LANGUAGE (Jan)

Nuevo interfaz para el motor de base de datos en el servidor (backend)Lenguaje Procedural (PL)

Postgres(Jan)

Se renombra la opción pg\_dump -H a -h(Bruce)

Se añade soporte Java para contraseñas, fechas europeas (Peter)

Se utilizan índices para operaciones LIKE y ~, !~ (Bruce)

Se añade funciones hash para fecha (datetime) y marca de tiempo (timespan) (Thomas)

Se elimina Time Travel(Vadim, Bruce)

Se añade paginación para \d y \z, y se corrige \i (Bruce)

Se añade soporte para conexiones de dominio Unix (Unix domain socket) al motor de base de datos y a  
 la librería del frontal(Goran)

Se implementa CREATE DATABASE/WITH LOCATION y la utilidad initlocation(Thomas)

Se permiten más palabras reservadas SQL92 y/o Postgres como identificadores de  
 columnas(Thomas)

Aumenta el soporte para SQL92 SET TIME ZONE...(Thomas)

SET/SHOW/RESET TIME ZONE utilizan la variable de entorno del motor de base de datos (backend) TZ(Thomas)

Se implementa SET keyword = DEFAULT y SET TIME ZONE DEFAULT(Thomas)

Se habilita SET TIME ZONE utilizando la variable de entorno (Thomas)

Se añade la variable de entorno PGDATESTYLE a la inicialización del frontal y del motor de base de  
 datos(backend)(Thomas)

Se añaden las variables de entorno PGTZ, PGCOSTHEAP, PGCOSTINDEX, PGRPLANS, PGGEQO a la librería de  
 inicialización del frontal(Thomas)

Se configura automáticamente la zona horaria en los test de regresión con "setenv PGTZ PST8PDT"(Thomas)

Se añade la tabla pg\_descripcion para información sobre tablas, columnas, operadores, tipos y  
 agregaciones (Bruce)

Se incrementa el límite de caracteres de 16 a 32 en los nombres de tablas/índices de sistema(Bruce)

Se renombran los índices del sistema(Bruce)

Se añade la opción 'GERMAN' para SET DATESTYLE(Thomas)

Se define un "ISO-style" formato de salida de fecha extendida con los campos "hh:mm:ss"(Thomas)

Se permite valores fraccionales para tiempos delta (por ej. '2.5 días')(Thomas)

Se valida la entrada de números más cuidadosamente para tiempos delta(Thomas)

Se implementa el día del año como una entrada posible para date\_part()(Thomas)

Se definen las funciones timespan\_finite() y text\_timespan() (Thomas)

Se elimina materia archivada(Bruce)

Se habilita una base de datos de autenticación pg\_password que esta separada del fichero de contraseñas  
 del sistema(Todd)

Se vuelcan los permisos ACLs, GRANT, REVOKE(Matt)

Se definen funciones de longitud de cadena text, varchar y bpchar(Thomas)

Corrección para el manejo de herencia en consultas, y computos de coste(Bruce)

Se implementa CREATE TABLE/AS SELECT (una alternativa a SELECT/INTO)(Thomas)

Se permite NOT, IS NULL, IS NOT NULL en restricciones(Thomas)

Se implementa UNIONes para SELECT(Bruce)

Se añade UNION, GROUP, DISTINCT a INSERT(Bruce)

varchar() almacena solo los bytes necesarios en disco(Bruce)

Correccion para BLOBs(Peter)  
 Mega-Parche para JDBC... vease README\_6.3 para lista de cambios(Peter)  
 Se elimina "opcion" no utilizada de PQconnectdb()  
 Nuevo comando LOCK y pagina de manual describiendo bloqueos(deadlocks)(Bruce)  
 Se anaden nuevos comandos psql \da, \dd, \df, \do, \dS, and \dT(Bruce)  
 Mejora de psql \z para mostrar secuencias(Bruce)  
 Se muestran NOT NULL y DEFAULT en tabla psql \d (Bruce)  
 Nuevo fichero de arranque de psql .psqlrc(Andrew)  
 Se modifican el script de muestra del arranque en contrib/linux para mostrar syslog(Thomas)  
 Nuevos tipos para direcciones IP y MAC en contrib/ip\_and\_mac(TomH)  
 Conversiones de tiempo en sistema Unix con tipos date/time en contrib/unixdate(Thomas)  
 Actualizacion del material contrib(Massimo)  
 Se anade soporte para conexion Unix (Unix socket) para DBD::Pg(Goran)  
 Nueva interfaz para python (PyGreSQL 2.0)(D'Arcy)  
 El nuevo protocolo para frontal/motor de base de datos(backend) tiene un numero version, byte de orden de  
     red (Phil)  
 Caracteristicas de seguridad en pg\_hba.conf mejorada y documentada, muchas depuraciones(Phil)  
 CHAR() tiene ahora acceso mas rapido que VARCHAR() o TEXT  
 preprocesador SQL embebido en ecpg  
 Se reduce la carga de columnas del sistemas(Vadim)  
 Se elimina la tabla pg\_time(Vadim)  
 Se anade el atributo pg\_type para indentificar tipos que necesitan longitud (bpchar, varchar)  
 Se anade informe de final de linea fuera de termino cuando el comando COPY falla  
 Se permiten que los permisos en VIEW se configuren separadamente de las tabla subyacentes.  
     Por seguridad, se utiliza como apropiado GRANT/REVOKE en vistas(Jan)  
 Las tablas ahora no tienen por omision GRANT SELECT TO PUBLIC. Debes conceder explicitamente esos permisos.  
 Limpieza de ejemplos en el tutorial(Darren)

#### Cambios en el Arbol Fuente

-----  
 Se anaden nuevas herramientas de desarrollo html, y caracteres flotantes en /tools/backend  
 Correccion para compilacion en SCO  
 Porte a computador Stratus, Robert Gillies  
 Anadido soporte para shlib para BSD44\_derived & i386\_solaris  
 Se hace la configuracion de Make mas automatizada(Brook)  
 Se anade script para comprobar los resultados del test de regresion  
 Se separan las funciones de verificacion sintactica en ficheros mas pequenos, agrupados juntos(Bruce)  
 Se renombran heap\_create a heap\_create\_and\_catalog, rename heap\_create a heap\_create()(Bruce)  
 Parche para bloqueos en Sparc/Linux (TomS)  
 Se elimina PORTNAME y se reorganiza el material especifico de puertos(Marc)  
 Se anade fichero de optimizador README(Bruce)  
 Se eliminan algunas recursiones en el optimizados y se limpia algo de codigo ahi(Bruce)  
 Correccion para bloqueo en NetBSD(Henry)  
 Correccion para el make libptcl(Tatsuo)  
 Parche para AIX(Darren)  
 Cambio IS TRUE, IS FALSE, ... a la expresiones utilizando "=" mas que a llamadas a funcion  
     istrue() o isfalse() para permitir optimizacion(Thomas)  
 Varias correcciones relacionadas con NetBSD/Sparc(TomH)  
 Bloqueos para linux Alpha(Travis,Ryan)  
 Se cambian elog(ALARMAS)(WARM) a elog(ERROR)(Bruce)  
 FAQ para FreeBSD(Marc)

Bring in the PostODBC source tree as part of our standard distribution(Marc)  
 Un parche menor para HP/UX vs 9(Stan)  
 New pg\_attribute.atttypmod for type-specific info like varchar length(Bruce)  
 Parches para Unixware(Billy)  
 Nuevo 'bloqueo' i386 para giro de bloqueo en asm(Billy)

Soporte para procesos en servidor (backends) multiplexados se elimina

Comienza un porte a OpenBSD  
 Comienza un porte a AUX  
 Comienza un porte a Cygnus  
 Se anaden funciones de cadena para la suite de regresion(Thomas)  
 Se expanden unos pocos nombres de funcion anteriormente truncados a 16 caracteres(Thomas)  
 Emilinda llamadas a malloc() no necesitadas y reemplazadas con palloc()(Bruce)

## Version 6.2.1

La v6.2.1 es una version de corrección de error y de usabilidad sobre la v6.2

Summary:

- Se permite a las cadenas extender líneas, por SQL92.
- Se incluyen ejemplos de funciones desencadenadoras(triggers) para insertar nombres de usuario en actualizaciones de tablas.

esta es una version menor de correccion de error sobre la v6.2 Para actualizaciones desde sistemas pre-v6.2, se requiere un dump/reload completo. Mire las notas de la version v6.2 para instrucciones.

## Migracion desde v6.2 a v6.2.1

Esta es una version menor de correccion de error. No se requiere un dump/reload desde la v6.2, pero se requiere desde cualquier version anterior a la v6.2

Al actualizar desde v6.2, si escojes un dump/reload encontraras que avg(money) se calcula ahora correctamente. Todos las otras correcciones de errores tendran efecto al actualizar los ejecutables.

Otra manera de evitar el dump/reload es utilizar el siguiente comando SQL desde psql para actualizar la tabla de sistema existente:

```
update pg_aggregate set aggfinalfn = 'cash_div_flt8'
where aggrname = 'avg' and aggbasetype = 790;
```

Se necesita hacer esto a todas las bases de datos existentes, incluyendo template1.

## Lista Detalladas de Cambios

Cambios en esta version

-----

Se permite bombres de columna TIME y TYPE (Thomas)

Se permite un rango grande de true/false como valores booleanos(Thomas)

Se soporta la salida de "now" y "current"(Thomas)

Manejo de DEFAULT con INSERT de NULL adecuado(Vadim)

Correccion para el proble de contar relaciones de referencia en el gestor de buffer(Vadim)

Se permite que las cadenas (strings) se extiendan a lineas, como ANSI(Thomas)

Correccion para cursor recursivo con ORDER BY(Vadim)

Correccion de computacion de avg(cash)(Thomas)

Correccion para especificacion de una columna dos veces en un ORDER/GROUP BY(Vadim)

Documentada nueva funcion de libpq que devuelve las lineas afectadas, PQcmdTuples(Bruce)

Funcion desencadenadora (trigger) para insertar nombres de usuario para INSERT/UPDATE(Brook Milligan)

## Version 6.2

Se requiere un dump /restore para aquellos que deseen migrar datos desde versiones anteriores de Postgres.

## Migracion desde v6.1 a v6.2

Esta version requiere un volcado (dump) completo de la base de datos en 6.1 y una retauracion de la base de datos en 6.2

Notese que la utilidad pg\_dump y pg\_dumpall de 6.2 deberia utilizarse para volcar (dump) la base de datos 6.1.

## Migracion desde v1.x a v6.2

Aquellas migraciones desde versiones anteriores a 1.\* deberian actualizarse primero a 1.09 porque el formato de salida de COPY fue mejorado desde la version 1.02.

## Lista Detallada de Cambios

Correccion de errores

-----

Correccion de problemas con pg\_dump para herencia, secuencias, tablas archivo(Bruce)

Correccion de errores de compilacion en desbordamientos para shifts, unsigned y prototipos malos

de Solaris(Diab Jerius)

Correccion de errores en lineas aritmeticas geometricas (malos calculos de intersecciones)(Thomas)

Comprobacion de intersecciones geometricas en los puntos de finalizacion para evitar lo feo del redondeo(Thomas)

Se cojen tentativas no-funcionales de borrado(Vadim)

Se cambia los nombres de la funcion de tiempo para ser mas consistentes(Michael Reifenberg)

Se comprueba las divisiones por cero(Michael Reifenberg)

Correccion para error muy antiguo que hacia que tuplas cambiadas/insertadas por un comando fueran visibles para el propio comando (de ese modo teniamos multiples actualizaciones

de tuplas actualizadas, etc)(Vadim)

Correccion para SELECT null, 'fail' FROM pg\_am(Patrick)

No se permite SELECT NULL como EMPTY\_FIELD(Patrick)

Se elimina senas innecesario de contrib/pginterface

Correccion OR(donde x!= 1 o x sea nulo no devolvera tuplas con x NULL)(Vadim)

Correccion para la funcion time\_cmp (Vadim)

Correccion en el manejo de funciones con argumentos que no tiene el atributo primero en

las clausulas WHERE (Vadim)

Correccion de GROUP BY cuando el orden de las entradas es diferente del orden en la lista de

seleccion (Vadim)

Correccion de pg\_dump para agrgaciones sin sfunc1(Vadim)

Mejoras

-----

El parametro del optimizador genetico por defecto GEQO es ahora 8(Bruce)

Se permite el uso de parametros en la lista de consulta teniendo agrgaciones en las funciones(Vadim)

Se anade driver JDBC como una interfaz(Adrian & Peter)

Utilidad pg\_password

Se devuelve el numero de tuplas insertadas/afectadas por una INSERT/UPDATE/DELETE etc.(Vadim)

Los disparadores (triggers) son implementados con CREATE TRIGGER (SQL3) (Vadim)

SPI (Interfaz de Programacion en el Servidor) permite la ejecución de consultas dentro de

funciones C (Vadim)

Implementado NOT NULL (SQL92) (Robson Paniago de Miranda)

Se incluyen palabras reservadas para el manejo de cadenas (strings), uniones externas (outer

joins) y uniones (unions) (Thomas)

Implementados comentarios extendidos ("/\* ... \*/") utilizando estados exclusivos(Thomas)

Anadido comentarios de una sola linea "///" (Bruce)

Eliminadas algunas restricciones de caracteres en nombres de operadores(Thomas)



Implementado DEFAULT y CONSTRAINT para tablas (SQL92)(Vadim & Thomas)

Anadido operador de concatenacion de texto y funcion (SQL92) (Thomas)

Soporte para sintaxis WITH TIME ZONE (SQL92) (Thomas)

Soporte para sintaxis INTERVAL unidad TO unidad (SQL92)(Thomas)

Se definen los tipos (de datos) DOUBLE PRECISION, INTERVAL, CHARACTER,

y CHARACTER VARYING (SQL92)(Thomas)

Se define el tipo FLOAT(p) y DECIMAL(p,s), NUMERIC(p,s) rudimentaria-  
mente (SQL92)(Thomas)

Se define EXTRACT(), POSITION(), SUBSTRING(), y TRIM() (SQL92)(Thomas)

Se define CURRENT\_DATE, CURRENT\_TIME, CURRENT\_TIMESTAMP (SQL92)(Thomas)

Se anade sintaxis y avisos para UNION, HAVING, INNER y OUTER JOIN (SQL92)(Thomas)

Se anaden mas palabras reservadas, la mayoría para cumplir con SQL92(Thomas)

Se permite la entrada de tiempo hh:mm:ss para tipos de marca de tiempo/tiempo

relativo (timespan/reltime) (Thomas)

Se anaden rutinas center() para lseg, path, polygon(Thomas)

Se anaden rutinas distance() para circle-polygon, polygon-polygon(Thomas)

Se verifica explicitidad para puntos y poliginis contenidos en los poligonos

utilizando un algoritmo de cruzamiento de ejes(Thomas)

Se anade rutina para convertir circle-box(Thomas)

Se unen operadores conflictivos para diferentes tipos de datos geometricos(Thomas)

Se reemplaza el operador de distancia "<==>" con "<->"(Thomas)

Se reemplaza el operador "above", "!^" con ">^", y el operador "below", "!" con "<^"(Thomas)

Se anaden rutinas para recorte de texto en ambos extremos, subcadenas y posiciones de cadena(Thomas)

Se han anadido rutinas de conversion circle(box) and poly(circle)(Thomas)

Se permite que las ordenaciones se almacenen en memoria antes que en ficheros(Bruce & Vadim)

Se permite que funciones y operadores con tipos internamente identicos finalicen con exito(Bruce)

Speed up backend startup after profiling analysis(Bruce)

Se alinean frecuentes llamadas a funciones para mejorar las prestaciones(Bruce)

Se reducen las llamadas open() (Bruce)

psql: Se anade correccion para PAGER \h y \?,\C

Correccion para el paginador psql cuando no hay tty(Bruce)

Nueva utilidad entab (Bruce)

Funciones generales de disparadores para integridad referencial (Vadim)

Funciones generales de disparadores para tiempo transcurrido (Vadim)

Funciones generales de disparadores para la prestacion AUTOINCREMENT/IDENTITY (Vadim)

Se implemente MOVE (Vadim)

Cambios en el Arbol Fuente

-----

Parches para HPUX 10 (Vladimir Turin)

Anadido soporte para SCO (Daniel Harris)

parches para mkLinux(Tatsuo Ishii)

Se cambia la terminologia geometrica de caja de "length" a "width"(Thomas)

Se desaprueban campos incrementales temporales no almacenados en el codigo geometrico(Thomas)

Se eliminan instrucciones de reinicio de INSTALL(Bruce)

Se mira en /usr/ucb antes de instalar(Bruce)

Correccion para el codigo de ejemplo de c++(Thomas)

Anadido -o a la pagina de manual de psql(Bruce)

Se previene que la longitud de la cadena relname no asignada sea copiada dentro de la base de datos (Bruce)

Depuracion del uso de NAMEDATALEN (Bruce)

Correccion para nombres pg\_proc por encima de los 15 caracteres de salida(Bruce)

Anadida la funcion strNcpy() (Bruce)

eliminados algunos (void) repartidos que son innecesarias(Bruce)  
 nuevo directorio de interfaces(Marc)  
 Se remplazan las llamadas fopen() con llamadas a las funciones fd.c(Bruce)  
 Se hacen las funciones estaticas donde es posible(Bruce)  
 Se encierran las funciones no usadas con #ifdef NOT USED(Bruce)  
 Se eliminan las llamadas a difftime() en el soporte de estampacion de fecha para corregir SunOS(Bruce & Thomas)  
 Cambios para Digital Unix  
 Correccion de portabilidad para pg\_dumpall(Bruce)  
 Se renombra pg\_attribute.attnvals a attdisbursion(Bruce)  
 Pagina de manual "intro/unix" ahora es "pgintro"(Bruce)  
 Pagina de manual "built-in" ahora es "pgbuiltin"(Bruce)  
 Pagina de manual "drop" ahora es "drop\_table"(Bruce)  
 Anadidas paginas de manual "create\_trigger", "drop\_trigger"(Thomas)  
 Anadidas constraints a los test de regresion(Vadin 6 Thomas)  
 Anadidos comentarios a la sintaxis del test de regresion(Thomas)  
 Anadido PGIDENT y programa de soporte(Bruce)  
 Commit masivo al ajecutar PGIDENT en todos los ficheros \*.c y \*.h(Bruce)  
 Ficheros movidos al directorio /src/tools(Bruce)  
 Guias de programacion de SPI y disparadores(triggers)(Vadim & D'Arcy)

## Version 6.1.1

### Migración desde v6.1 a v6.1.1

Esta es una versión de correcciones menores. No se necesita volcado/recarga desde la versión 6.1, pero sí para cualquier versión previa. Diríjase a las notas de versión de v6.1 para conseguir más detalles.

### Lista Detallada de Cambios

Cambios en esta versión

-----

Correcciones para SET con opciones (Thomas)  
 se permite a pg\_dump/pg\_dumpall preservar la propiedad de todos los objetos/tablas (Bruce)  
 la nueva opción \connect de psql permite cambiar el código de usuario sin cambiar de base de datos.  
 corrección de la opción -debug de initdb (Yoshihiko Ichikawa)  
 limpieza de lexttest(Bruce)  
 correcciones en la rutina hash (Vadim)  
 corrección en la aritmética de construcción del mes en fecha/hora (Thomas)  
 corrección en la manipulación del horario en zonas horarias en algunas migraciones (Thomas, Bruce, Tatsuo)  
 timestamp sobrecargado al utilizar funciones standard (Thomas)  
 otras limpiezas del código en rutinas de fecha y hora (Thomas)  
 \d de psql ahora es insensible a las mayúsculas (Bruce)  
 los comandos con \ de psql ahora pueden tener ; final (Bruce)  
 corrección de huecos de memoria en psql cuando se utiliza \g (Bruce)  
 correcciones mayores para la manipulación final de la comunicación con el servidor (Thomas, Tatsuo)  
 Correcciones para el ensamblador de Solaris y los ficheros de include (Yoshihiko Ichikawa)

pg\_dumpall devuelve ahora el status correcto, corregida la portabilidad (Bruce)

## Version 6.1

Las pruebas de regresión se han adaptado y modificado extensamente para la versión v6.1 de Postgres.

Se han añadido tres nuevos tipos de datos (datetime, timespan, and circle) al conjunto nativo de tipos de Postgres. Puntos, cajas, rutas y polígonos tienen sus formatos de salida consistentes por encima de los tipos de datos. La salida polígono en misc.out sólo se ha revisado para hacer correcciones relativas a la salida de regresión original.

Postgres v6.1 introduce un nuevo optimizador alternativo que utiliza algoritmos *genéticos*. Estos algoritmos introducen un comportamiento aleatorio en la ordenación de los resultados de la consulta cuando la consulta contiene múltiples calificadores o múltiples tablas (dando el optimizador una elección en el orden de evaluación). Se han modificado varias pruebas de regresión para ordenar explícitamente el resultado, y hacerlo así insensible a las elecciones del optimizador. Unas pocas pruebas de regresión corresponden a tipos de datos que son inherentemente desordenados (como puntos o intervalos de tiempo), y las pruebas que involucran estos tipos se fuerzan explícitamente con **set geqo to 'off'** y **reset geqo**.

La interpretación de los especificadores de vectores (los corchetes alrededor de valores atómicos) parece haber cambiado alguna vez tras las pruebas de regresión originales. Los ficheros actuales `./expected/*.out` reflejan esta nueva interpretación, ¡que puede no ser correcta!

Las pruebas de regresión con float8 fallan al menos en algunas plataformas. Esto se debe a las diferencias en las implementaciones de `pow()` y `exp()` y a los mecanismos de señalización utilizados para las condiciones de sobrecarga y subcarga.

Los resultados "aleatorios" en la prueba aleatoria deberían provocar que la prueba "aleatoria" resultase "fallida", puesto que los test de regresión se evalúan con un simple diff. Sin embargo, la prueba "aleatoria" no parece producir resultados aleatorios en mi máquina de pruebas (Linux/gcc/i686).

## Migración a v6.1

Esta migración requiere un volcado completo de la base de datos 6.0 y su restauración en la base de datos en 6.1.

Aquellos que quieran migrar desde versiones iniciales 1.\* deberían primero actualizarse a 1.09, porque el formato de salida de COPY se mejoró en la versión 1.02.

## Lista Detallada de Cambios

Corrección de errores

-----

comprobaciones en la longitud empaquetada en rutinas de la librería  
 parche de prioridad en el gestor de bloqueos  
 comprobaciones para sub/sobrecarga de float8 (Bruce)  
 correcciones en cruces de múltiples tablas (Vadim)  
 corrección de un aborto de SIGPIPE (Darren)

correcciones sobre objetos grandes (Sven)  
 se permite que los índices btree manipulen NULL,s (Vadim)  
 correcciones en timezone (D'Arcy)  
 select SUM(x) puede devolver NULL cuando no hay filas (Thomas)  
 se corrigen errores del optimizador interno y del ejecutor (Vadim)  
 se corrigen problemas cuando bucles internos en < o <= no tienen filas (Vadim)  
 se previene la re-inversión de cláusulas join de índices (Vadim)  
 corregida la clausula join para múltiples tablas (Vadim)  
 corregidos hash y hashjoin para vectores (Vadim)  
 corrección en btree para el tipo abstime (Vadim)  
 correcciones para objetos grandes (Raymond)  
 corregidos huecos en el buffer en índices hash (Vadim)  
 corregido rtree para el uso de barridos internos (Vadim)  
 correcciones esenciales para el uso de barridos internos, limpiezas (Vadim, Andrea)  
 se impide la colocación de buffers locales innecesarios (Vadim, Massimo)  
 corregidos huecos de buffers locales en abortos de transacciones (Vadim)  
 corregidos huecos de memoria del gestor de ficheros, limpiados (Vadim, Massimo)  
 corregidos huecos de memoria del gestor de almacenamiento (Vadim)  
 corregido que btree duplique la manipulación (Vadim)  
 corregida la reencarnación de tuplas borradas causada por vacuum (Vadim)  
 corregido SELECT varchar()/char() INTO TABLE hace campos de longitud cero (Bruce)  
 corregidas muchos huecos de memoria de psql, pg\_dump y libpq utilizando Purify (Igor)

#### Mejoras

-----

estadísticas de optimización de atributos (Bruce)  
 nuevo código de carga masivo btree mucho más rápido (Paul)  
 añadido BTREE UNIQUE para código de carga masiva (Vadim)  
 nuevo aspecto del código de depuración (Massimo)  
 cambios masivos en libpq++ (Leo)  
 el nuevo optimizador GEQO acelera la optimización en tablas multi-tabla (Martin)  
 nuevo mensaje de alarma para inserciones no únicas en claves únicas (Marc)  
 update x=-1, sin espacios, ahora es válido (Bruce)  
 se elimina la manipulación de identificadores sensibles a las mayúsculas (Bruce, Thomas, Dan)  
 la depuración del servidor ahora imprime un árbol agradable (Darren)  
 nuevas funciones de caracteres de Oracle (Edmund)  
 nuevas funciones de palabra clave con texto plano (Dan)  
 se cambian a mensajes diferentes no tal clase y insuficientes privilegios (Dan)  
 nueva función ANSI timestamp (Dan)  
 nuevos tipos ANSI Time y Date (Thomas)  
 mueve grandes grupos de datos en el servidor (Martin)  
 índices btree multicolumna (Vadim)  
 nuevo comando SET var TO valor (Martin)  
 status de transacción de actualización en las lecturas (Dan)  
 nuevos ajustes locales para tipos de caracteres (Oleg)  
 nuevo generador de series de números SEQUENCE (Vadim)  
 ahora es posible GROUP BY una función (Vadim)  
 reorganizada la prueba de regresión (Thomas, Marc)  
 nuevos pesos de operaciones en el optimizador (Vadim)  
 nueva opción psql \z grant/permit (Marc)  
 nuevo tipo de datos MONEY (D'Arcy, Thomas)  
 incrementada la velocidad de comunicación por el socket tcp (Vadim)  
 nueva opción VACUUM para estadísticas de atributos, y para ciertas columnas (Vadim)  
 muchas potenciaciones de tipos geométricos (Thomas, Keith)  
 Pruebas de regresión adicionales (Thomas)  
 nuevas variables de estilo de datos (Thomas, Vadim, Martin)

más operadores de comparación para ordenar tipos (Thomas)  
 nuevas funciones de conversión (Thomas)  
 no más formato btree compacto (Thomas)  
 se permite a pg\_dumpall preservar la propiedad de las bases de datos (Bruce)  
 nuevas variables SET GEQO=# y R\_PLANS (Vadim)  
 el viejo optimizador (!GEQO) puede utilizar planes del lado derecho (VADIM)  
 mejorado el control de tipos en el traductor de SQL (Bruce)  
 nuevos comandos SET, SHOW y RESET (Thomas, Vadim)  
 nueva opción USER \connect basededatos  
 nueva opción destroydb -i (Igor)  
 nuevos comandos de psql \dt y \di (Darren)  
 SELECT "\n" ahora genera nueva línea (A. Duursma)  
 nuevas funciones de conversión de la geometría desde el formato anterior (Thomas)

#### Cambios en el árbol fuente

-----

nuevo guión de configuración (Marc)  
 añadida la opción de configuración de lectura de línea (Marc)  
 nuevos ficheros de plantillas específicas del Sistema Operativo (Marc)  
 ya no se necesita editar Makefile.global (Marc)  
 se reordenan los ficheros de include (Marc)  
 parche para nextstep (Gregor Hoffleit)  
 se elimina código específico de WIN32 (Bruce)  
 se elimina la opción -e de postmaster, se mantiene sólo la opción postgres -e (Bruce)  
 se mezcla el código de librerías duplicadas en clientes y servidores (Martin)  
 ahora trabaja con eBones, Kerberos internacional (Jun)  
 más soporte a librerías compartidas  
 c++ incluye limpieza de fichero (Bruce)  
 Aviso sobre flex erróneo (Bruce)  
 Correcciones en la portabilidad para DG-UX, Ultrix, Irix, AIX.

## Version v6.0

Aquellos que quieran migrar datos desde versiones previas de Postgres necesitarán hacer un volcado/recuperación.

### Migración desde v1.09 a v6.0

Esta migración necesita un volcado completo de la base de datos 1.09 y una recuperación de los datos en 6.0.

### Migración desde versiones previas a v1.09 hasta v6.0

Quienes quieran migrar desde las iniciales versiones 1.\* deberían actualizarse primero a la versión 1.09, ya que el formato de salida de COPY se mejoró a partir de la versión 1.02.

## Lista Detallada de Cambios

### Corrección de errores

-----

Error ALTER TABLE - corriendo el proceso postgres se necesita re-leer la definición de la tabla.

Se permite que vacuum se ejecute sobre una tabla o sobre la base entera (Bruce)

Correcciones en tablas.

Corregido una sobre-escritura en tabla en escritura de memoria (Kurt)

Corregido un error en btree elusivo en rango/no en rango (Dan)

Correcciones en los índices hash para algunos tipos como time y date.

Correcciones para la explosión del tamaño de pg\_log.

Corregidos los permisos en lo\_export()(Bruce).

Corregidas lecturas no inicializadas de memoria (Kurt).

Corregido un error ALTER TABLE ... char(3) (Bruce)

Corregidas una pocas lagunas de memoria pequeñas.

Corregida la manipulación de EXPLAIN de opciones y cambiado el nombre de opción del path completo.

Corregida la salida de permisos de grupos de acl

Lagunas en la memoria (localizadas y eliminadas con herramientas como Purify (Kurt))

Mejoras menores de las reglas del sistema.

Se corrige NOTIFY

Nuevas instrucciones para ejecutar-comprobar.

Repaso general del código del analizador/traductor para informar correctamente de los errores e incrementar la velocidad.

Pg\_dump -d ahora manipula correctamente los NULL (Bruce)

Se evita que SELECT NULL mate el servidor (Bruce)

Se informan adecuadamente errores cuando las columnas de INSERT ... SELECT no casan.

Se informan adecuadamente errores cuando se están insertando nombres de columna que no son correctos.

Psql \g nombrefichero ahora trabaja (Bruce)

Corregido un problema de psql con instrucciones múltiples en una línea con múltiples salidas.

Eliminados oid's de sistema duplicados

SELECT \* INTO TABLE . GROUP/ORDER BY daba un error de enlace si la tabla existía (Bruce)

Varias correcciones a consultas que mataban el servidor

Las comillas al principio de una cadena a insertar produce un error (Bruce)

El lanzamiento de una consulta vacía ahora devuelve un status de vacío, no sólo la consulta " " (Bruce)

### Mejoras

-----

Se añade una página de manual para EXPLAIN (Bruce)

Se añade la capacidad de índice UNIQUE (Dan)

Se añade control de acceso vigilando nombre\_host/usuario, más que sólo nombre\_host y usuario.

Se añade el sinónimo != para <> (Bruce)

Se permite "select oid,\* from table"

Se permite a ORDER BY especificar columnas por número, o por tabla.columna que no son alias (Bruce)

Se permite el comando COPY desde la aplicación cliente (Bryan)

Se permite a GROUP BY que utilice alias de nombres de columnas (Bruce)

Se permite la compresión actual, no sólo en la misma página (Vadim)

Se permite la opción de instalación-configuración para auto-ayudar a todos los usuarios locales (Bryan)

Se permite a libpq que distinga entre textos con valor " y nulo (Bruce)  
 Se permite a los usuarios diferentes de postgres con privilegios de createdb ejecutar destroydb.  
 Se permiten restricciones sobre quién puede crear funciones C (Bryan)  
 Se permiten restricciones sobre quien puede hacer COPY del servidor (Bryan)  
 Se pueden reducir tablas, pg\_timer y pg\_log (Vadim & Erich)  
 Cambiado el nivel de debug 2 para imprimir sólo consultas, cambiado el formato de  
 la cabecera del debug (Bruce)  
 Se cambia la representación de las constantes decimales desde float4 a float8 (Bruce)  
 Ahora se fija el formato de fecha europeo cuando se arranca el postmaster.  
 Se ejecutan las funciones con el nombre en minúscula si no se encuentran con el nombre exacto.  
 Las correcciones del procesamiento de agregados/GROUP, permiten  
 'select sum(func(x),sum(x+y) from z'  
 Gist está ahora incluido en la distribución (Marc)  
 Autenticación Idendde usuarios locales (Bryan)  
 Se implementa el calificador BETWEEN (Bruce)  
 Se implementa el calificador IN (Bruce)  
 Libpq tiene PQgetisnull()(Bruce)  
 Mejoras de Libpq++  
 Nuevas opciones en initdb (Bryan)  
 Pg\_dump permite volcar los oid's (Bruce)  
 Pg\_dump crea los índices tras cargar las tablas, por velocidad (Bruce)  
 Pg\_dumpall vuelca todas las bases de datos, y la tabla de usuarios.  
 Adiciones a Pginterface para los valores NULL (Bruce)  
 Se previene la ejecución de postmaster como root  
 \h y \? son ahora legibles (Bruce)  
 Psql permite punto y coma escapados (\;) en cualquier parte de la línea (Bruce)  
 Se cambia el prompt de comandos de Psql para líneas intermedias en consultas o en líneas entre comillas (Bruce)  
 Las variables char(3) de Psql se muestran ahora como (bp)char en salidas \d (Bruce)  
 El código de retorno de Psql es ahora más ajustado (Bryan?)  
 Se actualiza la sintaxis de la ayuda de Psql (Bruce)  
 Se re-visita y corrige vacuum (Vadim)  
 Se reduce el tamaño de las diferencias de regresión, se elimina la diferencia del nombre de la zona horaria (Bruce)  
 Se eliminan parámetros de tiempo de compilación para capacitar distribuciones binarias (Bryan)  
 Gestión inversa de máscaras HBA (Bryan)  
 Autenticación segura de usuarios locales (Bryan)  
 Se incrementa la seguridad de vacuum (Vadim)  
 Vacuum ahora tiene opción VERBOSE (Bruce)

#### Cambios en el árbol fuente

-----

Todas las funciones tienen ahora prototipos que se comparan contra las llamadas.  
 Se permite inhabilitar fácilmente las declaraciones en Makefile.global (Bruce).  
 Se cambian las constantes oid utilizadas en el código para los nombres de #define  
 Se desacoplan las defines de sparc y solaris (Kurt)  
 gcc -Wall compila limpiamente con avisos (warnings) sólo a partir de construcciones no corregidas.  
 Gran reorganización/reducción del fichero de include (Marc).  
 Make ahora para en fallos de compilación (Bryan)  
 Reestructuración del Makefile (Bryan, Marc).  
 Se mezcla bsd\_2\_1 con bsd (Bruce)  
 Se elimina el programa Monitor-  
 Se cambia el nombre de Postgres95 a PostgreSQL  
 Nuevo fichero config.h (Marc, Bryan)  
 PG\_VERSION se fija ahora a 6.0 y lo utiliza el postmaster.  
 Adiciones a la portabilidad, incluyendo Ultrix, DG/UX, AIX, y Solaris

Se reduce el número de #define's, se centralizan las #define's  
 Se eliminan OIDS duplicadas en las tablas del sistema(Dan)  
 Se elimina información duplicada en el catálogo del sistema o errores de informe(Dan)  
 Se eliminan muchas #define's específicas del sistema operativo.  
 Generación/localización del fichero de objetos reestructurada(Bryan, Marc)  
 Reestructuradas las localizaciones de ficheros específicas de la  
 migración(Bryan, Marc)  
 Corregidas variables no utilizadas/no inicializadas.

## Version v1.09

Lo siento, hemos parado de registrar los cambios desde 1.02 a 1.09. Algunos de los cambios listados en 6.0 estaban ya incluidos en las versiones 1.02.1 a 1.09.

## Version v1.02

### Migración de v1.02 a v1.02.1

Aquí tenemos un nuevo fichero de migración para 1.02.1. Este el cambio de 'copy' y un guión para convertir los antiguos ficheros ascii.

**Nota:** Las siguientes notas son para el beneficio de los usuarios que quieren migrar bases de datos desde postgres95 1.01 y 1.02 a postgres95 1.02.1.

Si está usted arrancando con postgres95 1.02.1 de nuevas y no necesita migrar una base de datos antigua, no necesita leer lo que sigue.

Para actualizar anteriores bases de datos postgres95 versiones 1.01 o 1.02 a la versión 1.02.1, se requieren los siguientes pasos:

1. Arranque un nuevo postmaster 1.02.1
2. Añadi las nuevas funciones y operadores incluidos de 1.02.1 a bases de datos 1.02 o 1.02. Esto se hace ejecutando el nuevo servidor 1.02.1 contra su propia base de datos 1.01 o 1.02, y aplicando las consultas incluidas al final de este fichero. Se puede hacer esto muy facilmente con psql. Si su base de datos 1.01 o 1.02 se llama "testdb", y ha cortado los comandos del final de este fichero y los ha salvado en addfunc.sql:

```
% psql testdb -f addfunc.sql
```

Aquellos que estén actualizado bases de datos 1.02 obtendrán un aviso cuando ejecuten las dos últimas instrucciones en el fichero, pues ya están presentes en 1.02. No hay motivo para preocuparse.

### Procedimiento de Volcado/Recarga Procedure

Si está intentando recargar un pg\_dump o 'copy tablename to stdout' en modo texto generados con una versión previa, necesitará ejecutar el guión de sed siguiente sobre el fichero ASCII antes de cargarlo en la base de datos. El formato antiguo utilizaba ';



como end-of-data, mientras que el nuevo marcador de end-of-data (fin de los datos) es '\.'. También, las cadenas vacías se cargan ahora como "" en lugar de como NULL. Vea la página del manual de copy para obtener detalles completos.

```
sed 's/^\.$/\\./g' <in_file >out_file
```

Si está usted cargando una copia binaria más vieja, o una copia que no procede de stdout, no hay caracter end-of-data, y por ello no se necesita conversión.

```
- following lines added by agc to reflect the case-insensitive
- añadidas las siguientes líneas por agc para que no sea sensible a las mayúsculas
- regexp searching for varchar (in 1.02), and bpchar (in 1.02.1)
- regexp buscando varchar (en 1.02) y bpchar (en 1.02.1)
create operator ~* (leftarg = bpchar, rightarg = text, procedure = texticregexeq);
create operator !~* (leftarg = bpchar, rightarg = text, procedure = texticregexne);
create operator ~* (leftarg = varchar, rightarg = text, procedure = texticregexeq);
create operator !~* (leftarg = varchar, rightarg = text, procedure = texticregexne);
```

## Lista Detallada de Cambios

Mantenimiento y desarrollo del código fuente.

- \* equipo de voluntarios extendido por todo el mundo.
- \* el árbol fuente se encuentra ahora en CVS en ftp.ki.net

Mejoras

- \* psql (y la librería libpq que subyace) tiene ahora muchas más opciones para dar forma a sus salidas, incluyendo HTML
- \* pg\_dump ahora extrae el esquema y/o los datos, con muchas mejoras para asegurar que se complete.
- \* se utiliza psql en lugar de monitor en los guiones de órdenes de administración. monitor será despreciado en la siguiente versión.
- \* mejoradas las funciones de fecha/hora
- \* la comparación/inserción/actualización de NULL corregidos/potenciados.
- \* la librería y el interprete de órdenes de TCL/TK corregidos para que trabajen tanto con tcl7.4/tk4.0 como con tcl7.5/tk4.1

Errores corregidos (aunque demasiados numerosos para mencionarlos)

- \* índices
- \* gestión de almacenamiento
- \* comprobación de punteros a NULL antes de dereferenciarlos
- \* Correcciones en el Makefile.

Nuevas Migraciones

- \* añadida la migración a SolarisX86
- \* añadida la migración a BSDI 2.1
- \* añadida la migración a DGUX

## Version v1.01

### Migración desde v1.0 a v1.01

Las siguientes notas son para beneficio de los usuarios que quieren migrar bases de datos de postgres95 1.0 a postgres95 1.01

Si está usted arrancando de nuevas con postgres95 1.01 y no necesita migrar bases de datos anteriores, no necesita usted leer lo siguiente.

Para que postgres95 versión 1.01 funcione con bases de datos creadas con postgres95 versión 1.0, se requieren los siguientes pasos:

1. Fije la definición de NAMEDATALEN en src/Makefile.global a 16 y OIDNAMELEN a 20.
2. Decida si quiere usted autenticación basada en el ordenador.
  - a. Si lo hace, debe usted crear un fichero llamado "pg\_hba" en su directorio de datos de nivel superior (típicamente el valor de su \$PGDATA). src/libpq/pg\_hba muestra un ejemplo de sintaxis.
  - b. Si no quiere autenticación basada en el ordenador, puede usted comentar la línea

```
HBA = 1
```

en src/Makefile.global

Compruebe que la autenticación basada en el ordenador se ha activado por defecto, y si no sigue los pasos A o B anteriores, el out-of-the-box 1.01 no le permitirá conectar a las bases de datos 1.0

3. Compile e instale 1.01, pero NO ejecute el paso initdb.
4. Antes de hacer ninguna otra cosa, pare su postmaster 1.0, y respalde su directorio \$PGDATA existente.
5. Fije su variable de entorno PGDATA a sus bases de datos 1.0, pero fijela de forma que los binarios 1.01 sean los que se utilizan.
6. Modifique el fichero \$PGDATA/PG\_VERSION de 5.0 a 5.1
7. Arranque un nuevo postmaster 1.01
8. Añada las nuevas funciones y operadores incluidas en 1.01 sobre bases de datos 1.0. Esto se hace ejecutando el nuevo servidor 1.01 contra su propia base de datos 1.0, y aplicando las consultas unidas y salvadas en el fichero 1.0\_to\_1.01.sql. Se puede hacer facilmente desde psql. Si su base de datos se llama "testdb":

```
% psql testdb -f 1.0_to_1.01.sql
```

y entonces ejecute los siguientes comandos(copiar y pegar desde aquí):

- funciones incluidas añadidas que son nuevas en 1.01

```
create function int4eqoid (int4, oid) returns bool as 'foo'
language 'internal';
create function oideqint4 (oid, int4) returns bool as 'foo'
language 'internal';
create function char2icregexeq (char2, text) returns bool as 'foo'
language 'internal';
create function char2icregexne (char2, text) returns bool as 'foo'
language 'internal';
create function char4icregexeq (char4, text) returns bool as 'foo'
```

```

language 'internal';
create function char4icregexne (char4, text) returns bool as 'foo'
language 'internal';
create function char8icregexeq (char8, text) returns bool as 'foo'
language 'internal';
create function char8icregexne (char8, text) returns bool as 'foo'
language 'internal';
create function char16icregexeq (char16, text) returns bool as 'foo'
language 'internal';
create function char16icregexne (char16, text) returns bool as 'foo'
language 'internal';
create function texticregexeq (text, text) returns bool as 'foo'
language 'internal';
create function texticregexne (text, text) returns bool as 'foo'
language 'internal';

```

- funciones incluidas añadidas que son nuevas en 1.01

```

create operator = (leftarg = int4, rightarg = oid, procedure = int4eqoid);
create operator = (leftarg = oid, rightarg = int4, procedure = oideqint4);
create operator ~* (leftarg = char2, rightarg = text, procedure = char2icregexeq);
create operator !~* (leftarg = char2, rightarg = text, procedure = char2icregexne);
create operator ~* (leftarg = char4, rightarg = text, procedure = char4icregexeq);
create operator !~* (leftarg = char4, rightarg = text, procedure = char4icregexne);
create operator ~* (leftarg = char8, rightarg = text, procedure = char8icregexeq);
create operator !~* (leftarg = char8, rightarg = text, procedure = char8icregexne);
create operator ~* (leftarg = char16, rightarg = text, procedure = char16icregexeq);
create operator !~* (leftarg = char16, rightarg = text, procedure = char16icregexne);
create operator ~* (leftarg = text, rightarg = text, procedure = texticregexeq);
create operator !~* (leftarg = text, rightarg = text, procedure = texticregexne);

```

## Lista Detallada de Cambios

### Incompatibilidades:

- \* 1.01 es compatible hacia atrás con 1.0 si el usuario sigue los pasos marcados en el fichero `MIGRATION_from_1.0_to_1.01`.  
En caso contrario, 1.01 no es compatible con la base de datos 1.0

### Potenciaciones:

- \* se añade `PQdisplayTuples()` a `libpq` y se modifican `monitor` y `psql` para que lo utilicen.
- \* se añade la migración a `NEXT` (requiere la implementación de `SysVIPC`)
- \* se añade la sintaxis `CAST .. AS ...`
- \* se añaden las palabras clave `ASC` y `DESC`
- \* se añade `'internal'` como un posible lenguaje para `CREATE FUNCTION`  
las funciones internas son funciones en C que se han linkado estáticamente en el servidor de postgres.
- \* se ha añadido un nuevo tipo `"name"` para identificadores de sistema (nombres de tablas, nombres de atributos, etc.) Esto reemplaza al viejo tipo `char16`. La longitud del nombre se fija en la definición (`#define`) `NAMEDATELEN` de `src/Makefile.global`
- \* un manual de referencia legible que describe el lenguaje de consulta.
- \* se ha añadido un control de acceso basado en el ordenador. Se utiliza un fichero de configuración (`$PGDATA/pg_hba`) para almacenar los datos de configuración.

Si el control de acceso basado en el ordenador no es deseable, comente HBA=1 en src/Makefile.global.

- \* cambiada la manipulación de reges para hacerla uniforme con el código regex de Henry Spencer sin mirar la plataforma. Es código regex está incluido en la distribución.
- \* añadidas funciones y operadores para expresiones regulares insensibles a las mayúsculas.
- Los operadores son ~\* y !~\*.
- \* pg\_dump utiliza COPY en lugar de un bucle de SELECT para incrementar el rendimiento.

Errores corregidos:

- \* corregido un error del optimizador que provocaba volcados de memoria cuando se utilizaban llamadas a funciones en comparaciones en la cláusula WHERE.
- \* se cambian todos los usos de getuid a geteuid se modo que se utilicen los uids efectivos.
- \* psql devuelve ahora status distintos de cero en los errores cuando se usa -c
- \* se aplican los parches públicos 1-14

## Version v1.0

### Lista Detallada de Cambios

Cambio en el Copyright:

- \* Se ha aflojado el copyright de Postgres 1.0 para que sea libremente modificable para cualquier propósito. Lea por favor el fichero COPYRIGHT.
- Gracias al Profesor Michael Stonebraker por hacerlo posible.

Incompatibilidades:

- \* los formatos de fecha tienen que ser MM-DD-YYYY (o DD-MM-YYYY si está usted utilizando EUROPEAN STYLE). Esto sigue las especificaciones SQL-92.
- \* "delimiters" es ahora una palabra clave.

Potenciaciones:

- \* se ha añadido la sintaxis LIKE de sql
- \* el comando copy ahora recibe una especificación opcional USING DELIMITER. Los delimitadores pueden ser cualquier cadena de un único carácter.
- \* se ha añadido la migración a IRIX 5.3.
- Gracias a Paul Walmsley y otros.
- \* se ha actualizado pg\_dump para trabajar con una nueva libpq.
- \* se ha añadido \d a psql
- Gracias a Keith Parks
- \* se ha incrementado el rendimiento de regexp para arquitecturas que utilizan regex de POSIX regex debido a la memorización (caching) de patrones precompilados.
- Gracias a Alistair Crooks
- \* una nueva versión de libpq++
- Gracias a William Wanders

## Errores corregidos:

- \* se pueden especificar usuarios arbitrarios en el guión (script) createuser.
- \* ya funciona \c para conectar a otras bases de datos.
- \* se ha corregido una mala entrada en pg\_proc para float4inc()
- \* los usuarios con el campo usecreatedb fijado, ya pueden crear bases de datos sin ser el superusuario.
- \* se eliminan las entradas del control de acceso cuando la entrada ya no tiene ningún permiso.
- \* corregida la implementación de formatos de fecha/hora no portables.
- \* añadidas marcas (banderas=flags) kerberos en src/backend/Makefile
- \* libpq ya trabaja con kerberos.
- \* se han corregido errores tipográficos en el manual de usuario.
- \* btree con índices múltiples no ha trabajado nunca, pero ahora le decimos que no trabajan cuando intenta utilizarlo.

## Postgres95 Beta 0.03

### Lista Detallada de Cambios

## Cambios incompatibles:

- \* BETA-0.3 ES INCOMPATIBLE CON BASES DE DATOS CREADAS CON VERSIONES PREVIAS (debido a cambios en el catálogo del sistema y a cambios en la estructura de los índices).
- \* las dobles comillas (") se desprecian como un carácter de limitación para cadenas literales; necesitarás convertirlas a apostrofes (').
- \* el nombre de los agregados (como int4sum) se renombran de acuerdo con el SQL estándar (por ejemplo sum).
- \* la sintaxis CHANGE ACL se reemplaza por la sintaxis GRANT/REVOKE.
- \* los literales flotantes (como 3.14) son ahora del tipo float4 (en lugar del float8 de versiones anteriores); deberá realizar un transformado de tipos si su instalación depende de que siga siendo float8.  
¡Si rechaza realizar el transformado de tipos, y asigna un literal flotante a un campo de tipo float8, es posible que los valores almacenados sean incorrectos!
- \* se ha recompuesto totalmente LIBPQ para que las aplicaciones cliente (frontend) puedan conectarse a múltiples servidores (backend).
- \* el campo usesysid de pg\_user se ha cambiado de int2 a int4 para permitir mayores rangos de identificadores de usuarios de Unix.
- \* los portes a los sistemas operativos netbsd/freebsd/bsd se han consolidado en un único port derivado de BSD44. (Gracias a Alistair Crooks).

Cumplimento del estándar SQL (los siguientes detalles cambian para hacer a postgres95

más ajustado al estándar SQL-92):

- \* se han incluido los siguientes tipos SQL: smallint, int(eger), float, real, char(N), varchar(N), date y time.

Los siguientes son alias de los tipos postgres existentes:

```

        smallint -> int2
        integer, int -> int4
        float, real -> float4
    char(N) y varchar(N) se han implementado como tipos text truncados.
    Además, char(N) rellena a blancos el espacio no utilizado.
    * se utiliza el apóstrofe (') para limitar cadenas literales; " (además de \') se soportan
    para permitir insertar un único límite en una cadena.
    * se utilizan los nombres de agregados de SQL estándar (MAX, MIN, AVG, SUM, COUNT)
    (También, se pueden ahora sobrecargar los agregados, es decir, puede usted definir
    su propio agregado MAX para disponer de un tipo definido por el usuario).
    * se ha eliminado CHANGE ACL. Se añade la sintaxis GRANT/REVOKE.
    - Se pueden dar privilegios a un grupo utilizando la palabra clave "GROUP".
      Por ejemplo:
          GRANT SELECT ON mi_tabla TO GROUP mi_grupo;
    La palabra clave 'PUBLIC' también está soportada para autorizar a
    todos los usuarios.

```

Sólo se pueden otorgar o retirar privilegios a un usuario o grupo cada vez.

"WITH GRANT OPTION" no está soportado. Sólo los propietarios de clases pueden cambiar el control de acceso.

- El control de acceso de defecto es autorizar a los usuarios sólo a leer. Deberá usted autorizar explícitamente el acceso en inserción/actualización a los usuarios. Para cambiar esto, deberá modificar la línea
 

```
src/backend/utils/acl.h
```

 que define ACL\_WORLD\_DEFAULT

Errores corregidos:

- \* se ha corregido el error según el cual los agregados de tablas vacías no trabajaban adecuadamente. Ahora, los agregados ejecutados sobre tablas vacías devuelven las condiciones iniciales de los agregados. Así, COUNT de una tabla vacía devuelve correctamente el valor 0. MAX/MIN de una tabla vacía devolverá una tupla de valor NULL.
- \* se permite el uso de \; dentro del monitor.
- \* el mecanismo de notificación asíncrono LISTEN/NOTIFY ya trabaja.
- \* NOTIFY en el cuerpo de las reglas de acción ya trabaja.
- \* ya funcionan los índices hash, y los métodos de acceso en general deberían funcionar mejor. La creación de grandes índices btree debería ser mucho más rápida. (Gracias a Paul Aoki).

Otros cambios y potenciaciones:

- \* se añade la instrucción EXPLAIN utilizada para analizar el plan de ejecución de una consulta. (es decir: "EXPLAIN SELECT \* FROM EMP" muestra el plan de ejecución de la consulta).
- \* los mensajes WARN y NOTICE ya no muestran tiempo de ejecución en sí mismos. Para activar el tiempo de ejecución en los mensajes de error, descomente en
 

```
src/backend/utils/elog.h:
```

 la línea
 

```
/* define ELOG_TIMESTAMPS */
```
- \* En una violación del control de acceso, se dará el mensaje
 

```
"Either no such class or insufficient privilege"
```

 Este es el mismo mensaje que se devuelve cuando no se encuentra una clase. Esto

disuade a los usuarios no privilegiados de sospechar la existencia de clases privilegiadas.

- \* se han hecho algunos cambios adicionales en el catálogo del sistema que no son visibles para el usuario.

cambios en libpgtcl:

- \* se añade la opción `-oid` al comando de tcl `pg_result`. `pg_result -oid` devolverá el oid de la última tupla insertada. Si el último comando no fue una inserción, `pg_result -oid` devuelve `""`.
- \* el interface de objetos largos está utilizable como comandos tcl `pg_lo*`: `pg_lo_open`, `pg_lo_close`, `pg_lo_creat`, etc.

Potenciaciones de la Portabilidad y Nuevas Migraciones:

- \* Se han limpiado problemas con flex/lex. Ahora, se debería poder utilizar flex en lugar de lex en cualquier plataforma. Ya se harán asunciones sobre la forma de analizador sintáctico elegido basadas en la plataforma que utilice.
- \* Ahora se soporta la migración a Linux-ELF. Se han probado varias configuraciones: Se sabe que la siguiente configuración funciona:  
kernel 1.2.10, gcc 2.6.3, libc 4.7.2, flex 2.5.2, bison 1.24  
todo en formato ELF.

Nuevas utilidades:

- \* `ipcclean` añadido a la distribución  
habitualmente no se necesita ejecutar `ipcclean`, pero si cae su servidor y deja segmentos de memoria ocupados, `ipcclean` los limpiara para usted.

Nueva documentación:

- \* se ha revisado el manual del usuario y se ha añadido la documentación de `libpq`.

## Postgres95 Beta 0.02

### Lista Detallada de Cambios

Cambios incompatibles:

- \* La declaracion SQL para crear una base de datos en `'CREATE DATABASE'` en lugar de `'CREATEDB'`. De modo similar, el borrado de una base de datos en `'DROP DATABASE'` en lugar de `'DESTROYDB'`. Sin embargo los nombres de los ejecutables `'createdb'` y `'destroydb'` permanecen igual.

Nuevas herramientas:

- \* `pgperl` - una interfaz Perl (4.036) para Postgres95
- \* `pg_dump` - una utilidad para volcar una base de datos postgres en un fichero  
fichero de script conteniendo los comandos de creacion. Los ficheros script estan en formato ASCII y pueden ser usados para reconstruir una base de datos, incluso

en otras maquinas y otras arquitecturas. (Tambien es bueno para convertir una base de datos Postgres 4.2 a base de datos Postgres95.)

Los siguientes portes han sido incorporados en postgres95-beta-0.02:

- \* el porte a NetBSD por Alistair Crooks
- \* el porte a AIX por Mike Tung
- \* el porte a Windows NT por Jon Forrest (hay mas gente pero aun no han hecho nada)
- \* el porte a Linux ELF por Brian Gallew

Los siguientes errores han sido corregidos en postgres95-beta-0.02:

- \* lineas nuevas no acaban en escape en COPY OUT y problemas con COPY OUT cuando la primera linea es un '.'
- \* no se puede escribir un "retur" para utilizar el id del usuario por omision en "createuser"
- \* SELECT DISTINCT en tablas grandes aborta
- \* Problemas en la instalacion en Linux
- \* monitor no acepta el uso de 'localhost' como PGHOST
- \* psql genera un volcado de memoria cuando ejecuta \c or \l
- \* la etiqueta "pgtclsh" desaparecida de src/bin/pgtclsh/Makefile
- \* libpgtcl tiene un número de puerto de defecto codificado en duro
- \* SELECT DISTINCT INTO TABLE se cuelga
- \* CREATE TYPE no acepta 'variable' como longitud interna ("internallenght")
- \* resultados incorrectos utilizando mas de 1 agregado en una SELECT

## Postgres95 Beta 0.01

Version inicial.

## Tiempos Resultantes

Estos son los tiempos resultantes de ejecutar los test de regresion con los comandos

```
% cd src/test/regress
% make all
% time make runtest
```

Los tiempos bajo Linux 2.0.27 parecen tener una variacion de aproximadamente 5% de ejecucion a ejecucion, presumiblemente debido a vaguedades de planificacion en los sistemas multitareas.

## v6.5

Como ha sido el caso para versiones precedentes, los tiempos entre versiones no son directamente comparables puesto que se han añadido nuevos test de regresion. En general, la v6.5 es mas rapida que versiones precedentes.

Tiempo con fsync() desabilitado:

Tiempo	Sistema
--------	---------



```

02:00 Dual Pentium Pro 180, 224MB, UW-SCSI, Linux 2.0.36, gcc 2.7.2.3 -
O2 -m486
04:38 Sparc Ultra 1 143MHz, 64MB, Solaris 2.6

```

Tiempos con `fsync()` habilitado:

```

Tiempo Sistema
04:21 Dual Pentium Pro 180, 224MB, UW-SCSI, Linux 2.0.36, gcc 2.7.2.3 -
O2 -m486

```

Para el sistema Linux anterior, la utilizacion de discos UW-SCSI mejores que los (viejos) IDE conducen a una mejora de un 50% en la velocidad de los test de regresion.

## v6.4beta

Los tiempos para esta version no son directamente comparables a aquellos de versiones precedentes puesto que algunos test de regresion adicionales han sido incluidos. No obstante, en general, la v6.4 puede ser levemente mas rapida que versiones precedentes (gracias, Bruce!).

```

Tiempo Sistema
02:26 Dual Pentium Pro 180, 96MB, UW-SCSI, Linux 2.0.30, gcc 2.7.2.1 -
O2 -m486

```

## v6.3

Los tiempos para esta version no son directamente comparables a aquellos de versiones precedentes puesto que algunos test de regresion adicionales han sido incluidos y algunos test obsoletos incluyendo los tiempos de viaje han sido eliminados. No obstante, en general, la v6.3 es sustancialmente mas rapida que versiones precedentes (gracias, Bruce!).

```

Tiempo Sistema
02:30 Dual Pentium Pro 180, 96MB, UW-SCSI, Linux 2.0.30, gcc 2.7.2.1 -
O2 -m486
04:12 Dual Pentium Pro 180, 96MB, EIDE, Linux 2.0.30, gcc 2.7.2.1 -
O2 -m486

```

## v6.1

```

Tiempo Sistema
06:12 Pentium Pro 180, 32MB, EIDE, Linux 2.0.30, gcc 2.7.2 -O2 -m486
12:06 P-100, 48MB, Linux 2.0.29, gcc
39:58 Sparc IPC 32MB, Solaris 2.5, gcc 2.7.2.1 -O -g

```



## **Bibliografía**

Selección de referencias y lecturas sobre SQL y Postgres.

### **Libros de referencia sobre SQL**

### **Documentación específica sobre PostgreSQL**

### **Procedimientos y Artículos**

### **Notas**

1. <http://simon.cs.cornell.edu/home/praveen/papers/partindex.de95.ps.Z>
2. <http://s2k-ftp.CS.Berkeley.EDU:8000/postgres/papers/ERL-M89-17.pdf>

