INSTITUTO TECNOLÓGICO SUPERIOR DE JEREZ

JEREZ, ZACATECAS

NOMBRE:

ADRIANA DE JESUS MARQUEZ MENDOZA

NÚMERO DE CONTROL:

S17070161

CORREO ELECTRONICO:

marquez98709@gmail.com

CARRERA:

INGENIERIA EN SISTEMAS COMPUTACIONALES

SEMESTRE:

Sexto Semestre

ACTIVIDAD:

Actividad 2 - Cuestionario y Mapa conceptual Particionamiento

DOCENTE:

M.T.I, I.S.C. SALVADOR ACEVEDO SANDOVAL

MATERIA:

Administración de Base de Datos

Fecha:

20/03/2020

PARTICIONAMIENTO (MySQL y GBBD Postgre)

Asignación de espacio en disco para base de datos.

9.2 Gigas

Asignación de espacio en disco para tablas.

Depende del tipo de tabla.

Asignación de espacio en disco para usuarios.

Puede ser en kilobytes o megabytes. Todo depende de la base de datos.

¿Qué es y para qué sirve?

En postgre el tipo de particionamiento soportado se denomina particionado mediante herencia de tablas. Cada partición puede ser creada como una tabla hija de una única tabla padre.

Para que sirve:

- Reduce la cantidad de datos a recorrer en cada consulta SQL.
- Aumenta el rendimiento (menos datos que recorrer).

2.-Tipos

Partición o segmentación de tablas e índices

Existen dos tipos de particionado más comunes en postgre, que son:

- **Particionar por rangos:** La tabla es particionada mediante rangos definidos en base a la columna de llave primary o cualquier columna que no se solape entre los rangos de valores asignados a diferentes tablas hijas.
- **Particionar por lista:** La tabla es particionada listando los valores de cada una de las llaves en cada partición.

3.- Limitaciones

El tamaño máximo de una tabla puede ser de hasta 32 de TB (si una sola tabla, sin particionar, solamente una tabla) siendo esto (mucho, demasiado) más que

suficiente, sin el mantenimiento adecuado (índices y vacuums) y sin una estrategia de adecuada de particionamiento, esta tabla desde que pese 1 TB será casi inutilizable.

4.- instrucciones para el particionamiento

Para la creación de particiones se tomó en cuenta en base a un ejemplo.

- Crear las tablas hijas: ***test=# CREATE TABLE prueba_1 (CHECK (col >0 AND col <1000001)) INHERITS (prueba);***
- Crear reglas adicionales: ***test=# CREATE RULE prueba_3_rule AS ON INSERT TO prueba WHERE (col >2000000 AND col <3000001) DO INSTEAD INSERT INTO prueba_3 VALUES (NEW.*);***
- Insertar valores: ***test=# INSERT INTO prueba_6 SELECT * FROM prueba WHERE (col >5000000 AND col <6000001);***
- Eliminar datos de la tabla padre sin usar la cláusula ONLY: **test=# DELETE FROM prueba;**
  DELETE 12000000

  test=# SELECT count(*) FROM prueba;
  count
  ——-
  0
  (1 row)
- Eliminar datos de la tabla padre usando la cláusula ONLY: ***test=# DELETE FROM ONLY prueba;***
  DELETE 6000000

  test=# SELECT count(*) FROM ONLY prueba;
  count
  ——-
  0
  (1 row)

Podemos observar que al realizar una consulta sin usar dicha cláusula, el resultado es la suma de las filas de las tablas hijas.

test=# SELECT count(*) FROM prueba;

count

———

6000000

(1 row)

MySQL

Asignación de espacio en disco para base de datos.

25M

Asignación de espacio en disco para tablas.

16KB

Asignación de espacio en disco para usuarios.

No es tan simple debido a que son el usuario mysql y el grupo mysql los que adquieren la propiedad de todos los ficheros que utiliza, y sólo con esa información no podemos conseguir que el sistema de cuotas sea capaz de distinguir entre diversas bases de datos.

1.- ¿Qué es y para qué sirve?

El particionado de tablas es una técnica que se usa para reducir la cantidad de lecturas físicas a la base de datos cuando ejecutamos consultas, es decir, es el proceso donde tablas muy grandes son divididas en múltiples partes más pequeñas.

Al separar una tabla grande en tablas individuales más pequeñas, las consultas que acceden sólo a una fracción de los datos pueden ejecutarse más rápido debido a que hay menos datos que escanear.

El objetivo principal de particionar es ayudar en el mantenimiento de tablas grandes y reducir el tiempo de respuesta general para leer y cargar datos para operaciones SQL particulares.

2.- Tipos

Según quien realice la gestión del particionado, podemos distinguir dos tipos de particionado:

Manual: El particionado lo podríamos realizar nosotros en nuestra lógica de procesos de carga ETL. El problema es que se tendrá que gestionar este particionado para saber de qué tabla se tienen que leer los datos que le estemos pidiendo.

Automático: Las diferentes porciones de la tabla podrán ser almacenadas en diferentes ubicaciones del sistema de forma automática según nos permita el SGBDR que estemos utilizando.

La gestión la realiza de forma automática el motor de base de datos tanto a la hora de insertar registros como a la hora de leerlos.

MySQL implementa el particionado horizontal que consiste en tener varias tablas con las mismas columnas en cada una de ellas y distribuir la cantidad de registros en estas tablas (generalmente se particiona separando los datos en años, meses, etc.).

RANGE: La asignación de los registros de la tabla a las diferentes particiones se realiza según un rango de valores definido sobre una determinada columna de la

tabla o expresión. Es decir, nosotros indicaremos el número de particiones a crear, y para cada partición, el rango de valores que serán la condición para insertar en ella.

LIST: La asignación de los registros de la tabla a las diferentes particiones se realiza según una lista de valores definida sobre una determinada columna de la tabla o expresión. Es decir, nosotros indicaremos el número de particiones a crear, y para cada partición, la lista de valores que serán la condición para insertar en ella.

HASH: Está pensado para repartir de forma equitativa los registros de la tabla entre las diferentes particiones. Es decir, es MySQL quien hace ese trabajo. Para definir este tipo de particionado, deberemos de indicarle una columna del tipo integer o una función de usuario que devuelve un integer.

KEY: Similar al HASH, pero la función para el particionado la proporciona MySQL automáticamente. Se pueden indicar los campos para el particionado, pero siempre han de ser de la clave primaria de la tabla o de un índice único.

SUBPARTITIONS: MySQL permite además realizar subparticionado. Permite la división de cada partición en múltiples subparticiones.

Además, hemos de tener en cuenta que la definición de particiones no es estática. Es decir, MySQL tiene herramientas para poder cambiar la configuración del particionado para añadir o suprimir particiones existentes, fusionar particiones en otras, dividir una partición en varias, etc.

3.- Limitaciones

El particionado tiene sus limitaciones y sus restricciones, pues no se puede realizar sobre cualquier tipo de columna o expresión, tenemos un límite de particiones a definir y habrá que tener en cuenta algunas cosas para mejorar el rendimiento de las consultas y evitar que estas se recorran todas las particiones de una.

Construcciones prohibidas:

Procedimientos almacenados, funciones almacenadas, UDF o complementos.

Variables declaradas o variables de usuario.

Operadores aritméticos y lógicos.

El uso de los operadores aritméticos +, - y * se permite en la partición de expresiones. Sin embargo, el resultado debe ser un valor entero o NULL (excepto en el caso de [LINEAR]).

El operador DIV también es compatible; El operador "/"no está permitido.

Los operadores de bits |, &, ^, <<, >>, y ~ no se permiten.

Número máximo de particiones: El número máximo posible de particiones para una tabla dada que no usa el motor de almacenamiento NDB es de 8192 (este número incluye subparticiones).

Claves foráneas no admitidas: Las tablas particionadas que usan el motor de almacenamiento InnoDB no admiten claves foráneas.

4.- Instrucciones para el particionamiento

Crear partición:

ALTER TABLE [nombre_tabla] PARTITION BY RANGE(TO_DAYS(date))(

PARTITION [nombre_particion] VALUES LESS THAN (TO_DAYS("2011-12-01")),

PARTITION [nombre_particion] VALUES LESS THAN (TO_DAYS("2012-01-01")),

PARTITION [nombre_particion] VALUES LESS THAN (TO_DAYS("2012-02-01")),

PARTITION [nombre_particion_default] VALUES LESS THAN MAXVALUE);

Borrar partición:

ALTER TABLE [nombre_tabla] DROP PARTITION [nombre_particion];

Añadir particiones:

Si se tiene una particion default:

ALTER TABLE [nombre_tabla] REORGANIZE PARTITION [nombre_particion_default] INTO (

PARTITION [nombre_particion] VALUES LESS THAN (TO_DAYS("2012-07-01"),

PARTITION [nombre_particion_default] VALUES LESS THAN MAXVALUE));

Sin partición default:

ALTER TABLE [nombre_tabla] ADD PARTITION (

PARTITION [nombre_particion] VALUES LESS THAN (TO_DAYS("2012-07-01")));

Consultar partición:

SELECT PARTITION_NAME,TABLE_ROWS FROM information_schema.PARTITIONS WHERE TABLE_NAME='[nombre_tabla]';

Referencias

Blog. (7 de febrero de 2007). Obtenido de Blog: http://enriqueplace.blogspot.com/2006/02/postgresql-la-base-de-datos-empez.html

Cuotas en MySQL. (s.f.). Obtenido de Cuotas en MySQL: http://www.bdat.net/trucos/quotas_en_mysql/

gloogle sites. (s.f.). Obtenido de gloogle sites: https://sites.google.com/site/itjabd23/home/asignatura/plan-de-estudios/unidad-3-configuracion-y-administracion-del-espacio-en-disco

MySQL. (s.f.). Obtenido de MySQL: http://download.nust.na/pub6/mysql/doc/refman/5.0/es/innodb-file-space.html

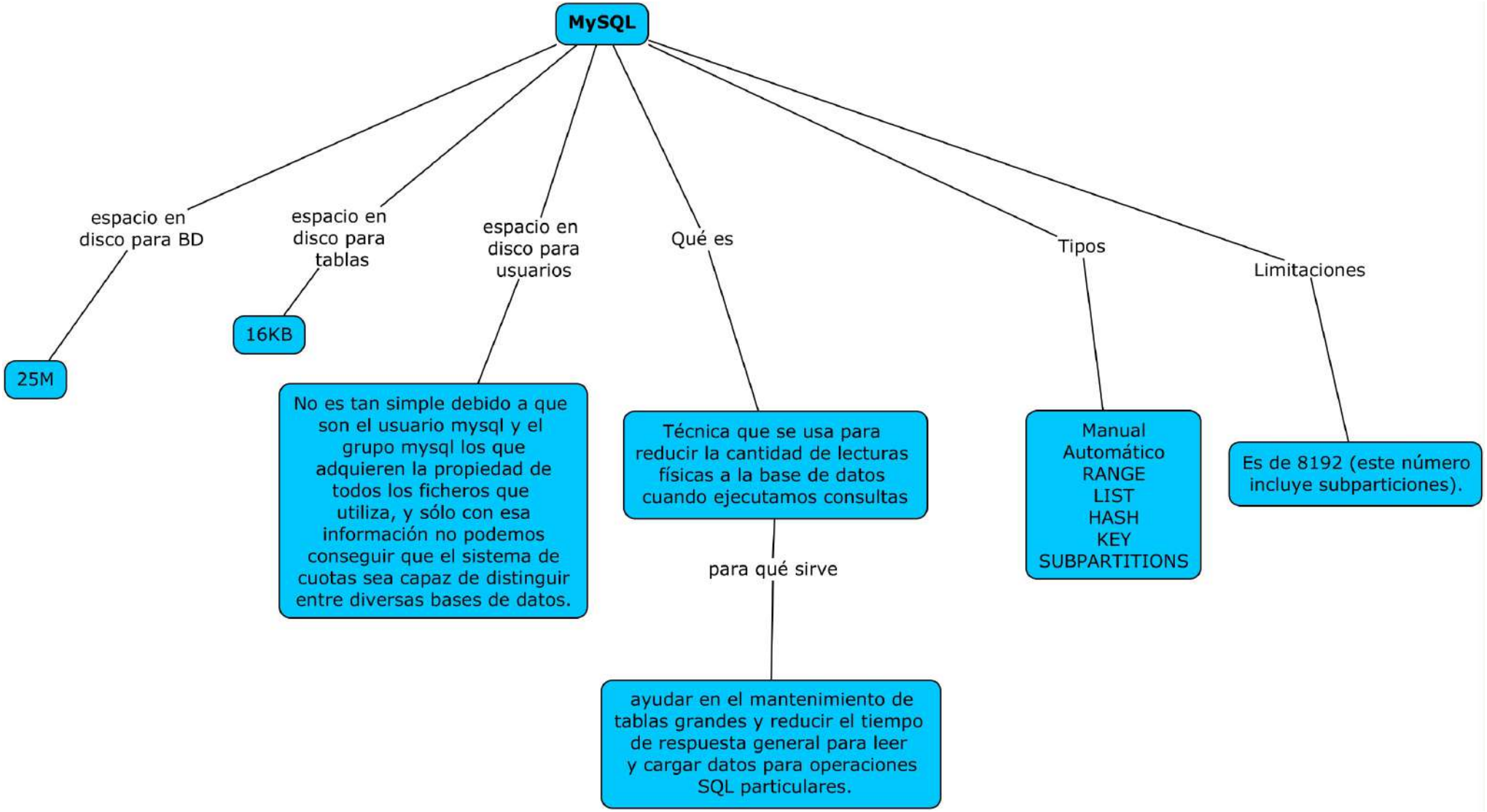
PostgreSQL. (27 de noviembre de 2009). Obtenido de postgreSQL: https://beastieux.com/2009/11/27/postgresql-particionamiento-de-tablas/

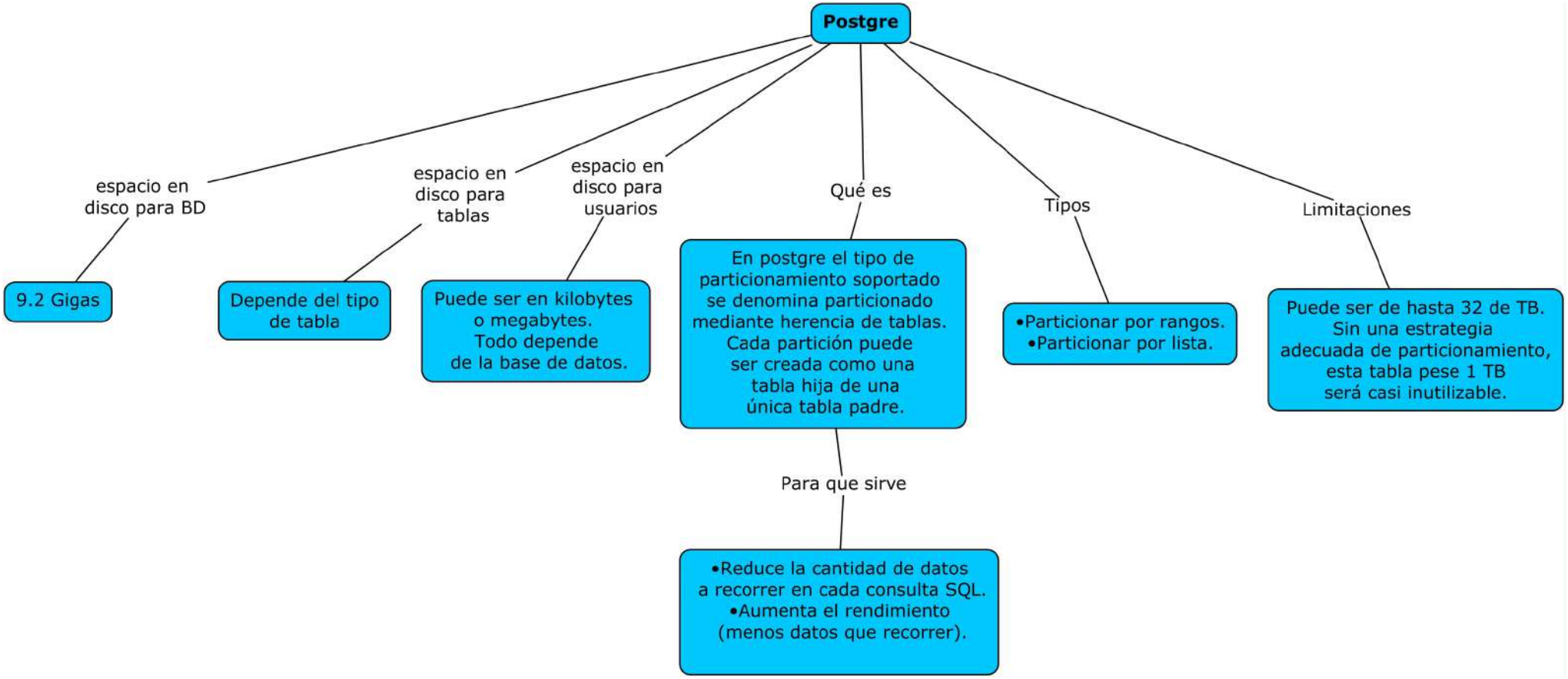
rozvodb. (18 de mayo de 2018). Obtenido de rozvodb: http://rozvodb.com/node/8

uneweb. (25 de agosto de 2014). Obtenido de uneweb: http://tecnologiaenvivo.com/aprende-sobre-el-espacio-de-tablas/

Referencias

- NE. (NE). Particiones de base de datos. 2020, de ibm.com Sitio web: https://www.ibm.com/support/knowledgecenter/es/SSH2TE_1.0.0/com.ibm.7700.r2.common.doc/doc/c00000092.html
- Leonardo De Seta. (2009). Particionado de tablas para aumentar el rendimiento. 2020, de dosideas.com Sitio web: https://dosideas.com/noticias/base-de-datos/576-incrementar-rendimiento-de-bbdd-con-qparticionado-de-tablasq
- NE. (NE). Chapter 23 Partitioning. 2020, de mysql.com Sitio web: https://dev.mysql.com/doc/refman/8.0/en/partitioning.html
- NE. (NE). Tabla Hechos Venta. Particionado en MySql.. 2020, de dataprix.com Sitio web: https://www.dataprix.com/es/blog-it/respinosamilla/tabla-hechos-venta-particionado-mysql
- Daniel Santana. (NE). Particiones en MySQL y Oracle. 2020, de blogspot.com Sitio web: http://dan1456bd.blogspot.com/p/particiones-en-mysql-y-oracle.html
- NE. (2019). FRAGMENTACIÓN HORIZONTAL EN MYSQL. 2020, de blogprog.gonzalolopez.es Sitio web: https://blogprog.gonzalolopez.es/articulos/fragmentacion-horizontal-en-mysql.html


MySQL
espacio en disco para BD
25M
espacio en disco para tablas
16KB
espacio en disco para usuarios
No es tan simple debido a que son el usuario mysql y el grupo mysql los que adquieren la propiedad de todos los ficheros que utiliza, y sólo con esa información no podemos conseguir que el sistema de cuotas sea capaz de distinguir entre diversas bases de datos.
Qué es
Técnica que se usa para reducir la cantidad de lecturas físicas a la base de datos cuando ejecutamos consultas
para qué sirve
ayudar en el mantenimiento de tablas grandes y reducir el tiempo de respuesta general para leer y cargar datos para operaciones SQL particulares.
Tipos
Manual
Automático
RANGE
LIST
HASH
KEY
SUBPARTITIONS
Limitaciones
Es de 8192 (este número incluye subparticiones).


Postgre
espacio en disco para BD
9.2 Gigas
espacio en disco para tablas
Depende del tipo de tabla
espacio en disco para usuarios
Puede ser en kilobytes o megabytes. Todo depende de la base de datos.
Qué es
En postgre el tipo de particionamiento soportado se denomina particionado mediante herencia de tablas. Cada partición puede ser creada como una tabla hija de una única tabla padre.
Para que sirve
•Reduce la cantidad de datos a recorrer en cada consulta SQL.
•Aumenta el rendimiento (menos datos que recorrer).
Tipos
•Particionar por rangos.
•Particionar por lista.
Limitaciones
Puede ser de hasta 32 de TB. Sin una estrategia adecuada de particionamiento, esta tabla pese 1 TB será casi inutilizable.