INSTITUTO TECNOLÓCIO SUPERIOR DE JEREZ

INGENIERÍA EN SISTEMAS COMPUTACIONALES

ADMINISTRACIÓN DE BASES DE DATOS

6° SEMESTRE

I.S.C. SALVADOR ACEVEDO SANDOVAL

“MAPA CONCEPTUAL: PARTICIONAMIENTO”

ALBAR DE LA TORRE GARCÍA

LUIZENRIQUE GONZÁLEZ VILLA

No. Control:

16070122

16070125

JEREZ ZACATECAS

14 DE FEBRERO DEL 2019

Particionamiento (Mysql y PostgreSQL)

MYSQL

1.Asignación de espacio en disco para base de datos.

Administra las E / S de disco para evitar que el subsistema de E / S se sature y administrar el espacio en disco para evitar llenar los dispositivos de almacenamiento. El modelo de diseño ACID requiere una cierta cantidad de E / S que puede parecer redundante, pero ayuda a garantizar la fiabilidad de los datos. Dentro de estas restricciones, InnoDB intenta optimizar el trabajo de la base de datos y la organización de los archivos de disco para minimizar la cantidad de E / S de disco.

Los archivos de datos que se definan en la configuración usando el innodb_data_file_path, los archivos son concatenados para formar el espacio de tabla del sistema.

Las bases de datos se almacenan en ficheros o archivos. Existen diferentes formas de organizaciones primarias de archivos que determinan la forma en que los registros de un archivo se colocan físicamente en el disco y, por lo tanto, cómo se accede a éstos.

Las distintas formas de organizaciones primarias de archivos son:

- Archivos de Montículos (o no Ordenados): esta técnica coloca los registros en el disco sin un orden específico, añadiendo nuevos registros al final del archivo.
- Archivos Ordenados (o Secuenciales): mantiene el orden de los registros con respecto a algún valor de algún campo (clave de ordenación).
- Archivos de Direccionamiento Calculado: utilizan una función de direccionamiento calculado aplicada a un campo específico para determinar la colocación de los registros en disco.
- Árboles B: se vale de la estructura de árbol para las colocaciones de registros.
- Organización Secundaria o Estructura de Acceso Auxiliar: Estas permiten que los accesos a los registros de un archivo basado en campos alternativos, sean más eficientes que los que han sido utilizados para la organización primaria de archivos.

2. Asignación de espacio en disco para tablas.

Para evitar los problemas que vienen con el almacenamiento de todas las tablas e índices dentro del espacio de tablas del sistema, puede habilitar la innodb_file_per_table opción de configuración (la predeterminada), que almacena cada tabla recién creada en un archivo de espacio de tabla separado (con extensión. ibd). Para las tablas almacenadas

de esta manera, hay menos fragmentación dentro del archivo de disco, y cuando la tabla se trunca, el espacio se devuelve al sistema operativo en lugar de que InnoDB aún lo reserve dentro del espacio de tablas del sistema.

3. Asignación de espacio en disco para usuarios.

Cuando elimina datos de una tabla, InnoDB contrae los índices de árbol B correspondientes. El hecho de que el espacio liberado esté disponible para otros usuarios depende de si el patrón de eliminaciones libera páginas individuales o extensiones al espacio de tabla. Si se descarta una tabla o se eliminan todas las filas, se garantiza que liberará el espacio a otros usuarios, pero recuerde que las filas eliminadas se eliminan físicamente solo por la operación, que ocurre automáticamente un tiempo después de que ya no son necesarias para las reversiones de transacciones o lecturas consistentes.

Al detectar la liberación de espacio por medio de una eliminación de tablas o filas para otros usuarios

1: ¿Qué es y para qué sirve?

El particionamiento de tabla en las bases de datos es una técnica para dividir en segmentos lógicos las tablas de datos para que al momento de hacer búsquedas estas se puedan realizar sobre grupos más pequeños de datos.

2: Tipos

- **Particionamiento de key.-** Se proporcionan una o más columnas que se van a evaluar y el servidor MySQL proporciona su propia función de hash. Estas columnas pueden contener valores distintos de los enteros, ya que la función hash proporcionada por MySQL garantiza un resultado entero independientemente del tipo de datos de columna.

- **Particionamiento de range.-** Este tipo de particionamiento asigna filas a particiones en función de los valores de columna que se encuentran dentro de un intervalo determinado.

- **Particionamiento de Hash.-** Con este tipo de particionamiento, se selecciona una partición en función del valor devuelto por una expresión definida por el usuario que funciona en valores de columna en filas que se insertarán en la tabla. La función puede constar de cualquier expresión válida en MySQL que produzca un valor entero no negativo.

- **Particionamiento de List.-** La partición está seleccionada en función de columnas que coinciden con uno de un conjunto de valores discretos.

3: Limitaciones

Número máximo de particiones. El número máximo posible de particiones para una tabla determinada que no utiliza el motor de almacenamiento NDB es 8192. Este número incluye subparticiones.

Accesibilidad a un table. A veces, un cambio en el modo SQL del servidor puede hacer que las tablas particionadas no se puedan usar.

4: Instrucciones para el particionamiento

```
CREATE TABLE reports (
 id int(10) NOT NULL AUTO_INCREMENT,
 date datetime NOT NULL,
 report TEXT,
 PRIMARY KEY (id,date)
) ENGINE=InnoDB DEFAULT CHARSET=utf8

ALTER TABLE reports PARTITION BY RANGE(TO_DAYS(date))(
 PARTITION p201111 VALUES LESS THAN (TO_DAYS("2011-12-01")),
 PARTITION p201112 VALUES LESS THAN (TO_DAYS("2012-01-01")),
 PARTITION p201201 VALUES LESS THAN (TO_DAYS("2012-02-01")),
 PARTITION p201202 VALUES LESS THAN (TO_DAYS("2012-03-01")),
```

```
 PARTITION p201203 VALUES LESS THAN (TO_DAYS("2012-04-01")),
 PARTITION p201204 VALUES LESS THAN (TO_DAYS("2012-05-01")),
 PARTITION p201205 VALUES LESS THAN (TO_DAYS("2012-06-01")),
 PARTITION pDefault VALUES LESS THAN MAXVALUE
);
```

PostgreSQL

¿Qué Es y Para Q Sirve?

Consiste en subdividir una tabla de gran tamaño y con millones de registros en tablas hijas más pequeñas y vaciar el contenido de la tabla padre a las hijas. Esto con la finalidad de optimizar el rendimiento de la tabla.

Tipos

Los dos tipos de particionado más comunes en postgresql son:

Particionamiento de rango

La tabla se divide en "rangos" definidos por una columna clave o un conjunto de columnas, sin superposición entre los rangos de valores asignados a diferentes particiones. Por ejemplo, uno podría dividir por rangos de fechas o por rangos de identificadores para objetos comerciales particulares.

Particionamiento de lista

La tabla se divide mediante una lista explícita de los valores clave que aparecen en cada partición.

Limitaciones

NO FUNCIONA EL UPDATE EN CAMPOS CON RESTRICCIONES:

–Cuando queremos actualizar los idperacad=’2005-1′ a ‘2009-1’ se produce un error de violación de restricciones: <<ERROR: new row for relation “postulante_2005_1” violates check constraint “postulante_2005_1_idperacad_check”>>

–Esto sucede porque el idperacad ‘2005-1’ intenta actualizarse a ‘2009-1’ sobre la misma partición postulante_2005_1, y obviamente esta partición tiene restringido cualquier otro idperacad que no sea ‘2005-1’

Instrucciones

Crear las tablas hijas

crear las tablas hijas heredadas de la tabla padre «prueba» con una restricción para que chequee el intervalo de filas cuando se inserten los valores de la tabla padre en la tabla hija correspondiente. La sentencia SQL es algo similar a esta.

```
CREATE TABLE prueba_1 (CHECK (col >0 AND col <1000001)) INHERITS (prueba);
```

Crear reglas adicionales

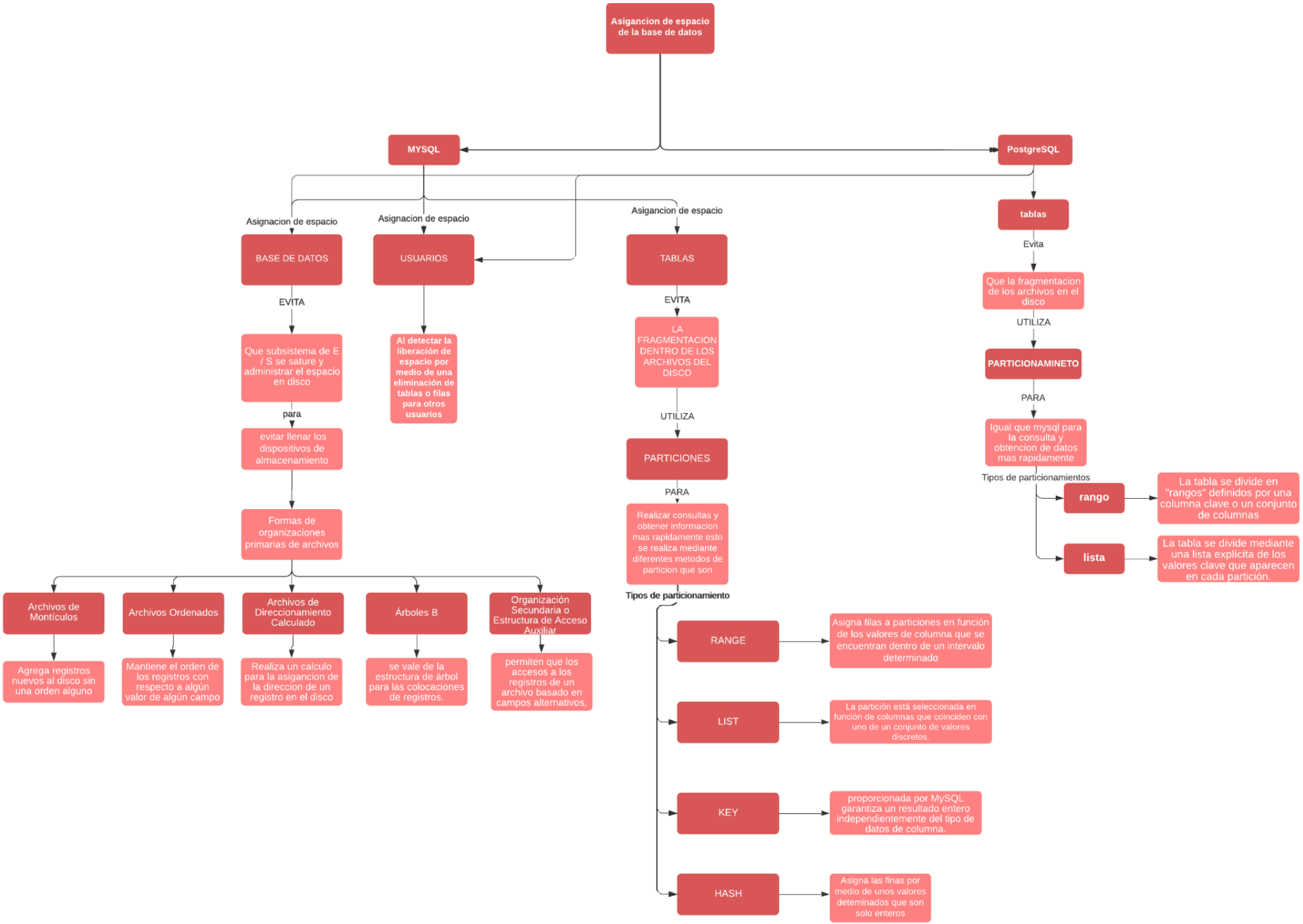
Para **prevenir que no se realicen inserciones en la tabla padre** y se redirijan a las tablas hijas correspondientes, se crean reglas de forma similar a esta.

```
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

inserción en cada tabla hija, realizando una subconsulta de la tabla padre con el rango de datos correspondiente. La sentencia SQL quedaría así.

```
INSERT INTO prueba_6 SELECT * FROM prueba WHERE (col >5000000 AND col <6000001);
```


Asigancion de espacio de la base de datos
MYSQL
PostgreSQL
Asignacion de espacio
BASE DE DATOS
EVITA
Que subsistema de E / S se sature y administrar el espacio en disco
para
evitar llenar los dispositivos de almacenamiento
Formas de organizaciones primarias de archivos
Archivos de Montículos
Agrega registros nuevos al disco sin una orden alguno
Archivos Ordenados
Mantiene el orden de los registros con respecto a algún valor de algún campo
Archivos de Direccionamiento Calculado
Realiza un calculo para la asigancion de la direccion de un registro en el disco
Árboles B
se vale de la estructura de árbol para las colocaciones de registros.
Organización Secundaria o Estructura de Acceso Auxiliar
permiten que los accesos a los registros de un archivo basado en campos alternativos.
Asignacion de espacio
USUARIOS
Al detectar la liberación de espacio por medio de una eliminación de tablas o filas para otros usuarios
Asigancion de espacio
TABLAS
EVITA
LA FRAGMENTACION DENTRO DE LOS ARCHIVOS DEL DISCO
UTILIZA
PARTICIONES
PARA
Realizar consultas y obtener informacion mas rapidamente esto se realiza mediante diferentes metodos de particion que son
Tipos de particionamiento
RANGE
Asigna filas a particiones en función de los valores de columna que se encuentran dentro de un intervalo determinado
LIST
La partición está seleccionada en función de columnas que coinciden con uno de un conjunto de valores discretos.
KEY
proporcionada por MySQL garantiza un resultado entero independientemente del tipo de datos de columna.
HASH
Asigna las finas por medio de unos valores deteminados que son solo enteros
tablas
Evita
Que la fragmentacion de los archivos en el disco
UTILIZA
PARTICIONAMINETO
PARA
Igual que mysql para la consulta y obtencion de datos mas rapidamente
Tipos de particionamientos
rango
La tabla se divide en "rangos" definidos por una columna clave o un conjunto de columnas
lista
La tabla se divide mediante una lista explícita de los valores clave que aparecen en cada partición.

Referencias

https://todopostgresql.com/particionado-de-tablas-de-postgres/

https://www.postgresql.org/docs/10/ddl-partitioning.html

https://beastieux.com/2009/11/27/postgresql-particionamiento-de-tablas/

https://dev.mysql.com/doc/mysql-infoschema-excerpt/5.6/en/partitions-table.html

https://dev.mysql.com/doc/refman/8.0/en/partitioning-limitations.html