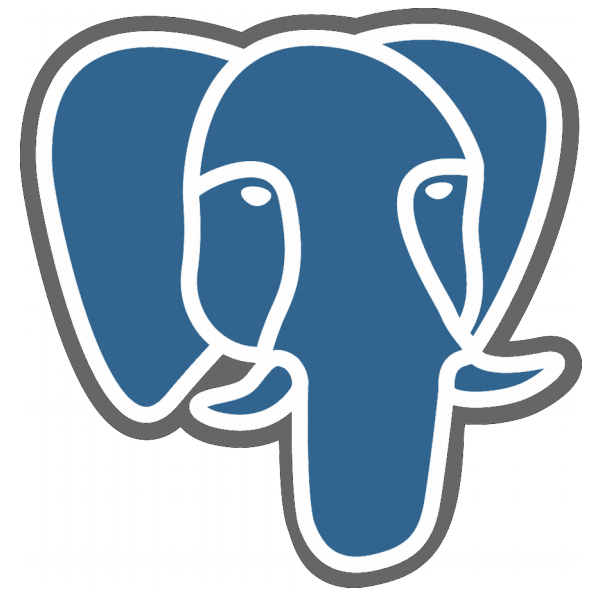


Consultas Seguras y No Seguras

INSTRUCTOR:

José Segovia

<info@todopostgresql.com>



Todo PostgreSQL

**Consultas
Seguras**

Añadir una nueva columna

Esta operación no bloqueará la tabla y se puede hacer con seguridad. Pero en algunos casos, pueden bloquear la tabla.

Todo PostgreSQL

Eliminar una columna

Eliminar una columna requiere muy poco tiempo, pero PostgreSQL no recuperará el espacio de disco hasta que se ejecute un "VACUUM FULL".

Añadir un valor predeterminado a una columna existente

Esta operación no bloqueará la tabla y se puede hacer con seguridad.

Eliminar una restricción

Esta operación no bloqueará la tabla y se puede hacer con seguridad.

Todo PostgreSQL

**Consultas
No Seguras**

Añadir una columna con un valor predeterminado

Añadir una columna con un valor por defecto requiere la actualización de cada fila de la tabla (para almacenar el nuevo valor de la columna). Para grandes tablas ejecutará una operación que mantendrá la tabla bloqueada.

Todo PostgreSQL

Añadir una columna con un valor predeterminado

Por tanto, en casos en los que se quiera rellenar la columna, en la que la mayoría de los valores no son por defecto, lo más aconsejable es añadir la columna sin valores predeterminados, e introducir los valores correctos usando **UPDATE** y luego añadir cualquier valor predeterminado.

Todo PostgreSQL

Añadir una columna con un valor predeterminado

La forma correcta es hacer “UPDATE” por lotes, por ejemplo, uno de 1000 filas a la vez, porque en mayores cantidades podrá causar el bloqueo de toda la tabla.

Todo PostgreSQL

Añadir una columna con not null

Esto tendrá el mismo problema que “Añadir una columna con un valor por defecto”. Para realizar esta operación sin bloqueo, puede crear una nueva tabla añadiendo la columna con not null.

Todo PostgreSQL

Añadir una columna con not null

Escribir en ambas tablas, rellenar y luego cambiar a la nueva tabla.

Esta solución es muy costosa y usa el doble de espacio de una tabla.

Todo PostgreSQL

Cambiar el tipo de dato de una columna

No es estrictamente inseguro para todos los cambios. Cambiar la longitud de un varchar, por ejemplo, bloquea una tabla.

Cambiar el tipo de dato de una columna

Si el cambio de tipo de columna requiere una reescritura o no, depende del tipo de dato, en el caso de que requiera una reescritura se tendrá que hacer la actualización de cada fila de la tabla.

Cambiar el tipo de dato de una columna

Como solución, puede agregar una nueva columna con el tipo de dato necesario, cambiar el código para escribir en ambas columnas, y rellenar la nueva columna.

Todo PostgreSQL

Añadir un índice

Normalmente PostgreSQL bloquea la tabla contra escritura para hacer los índices y realiza la construcción del índice con un sola exploración de la tabla.

Añadir un índice

Otras operaciones todavía pueden leer la tabla, pero si tratan de insertar, actualizar o eliminar filas de la tabla, no se podrán hacer puesto que están bloqueadas hasta que finalice la generación del índice.

Todo PostgreSQL

Añadir un índice

PostgreSQL soporta construcción de índices sin bloquear las escrituras. Este método se invoca especificando la opción **CONCURRENTLY** de **CREATE INDEX**.

Añadir un índice

Cuando se utiliza esta opción, PostgreSQL debe realizar dos exploraciones de la tabla, y además se debe esperar a que todas las transacciones existentes que podrían modificar o utilizar el índice sin interrumpir.

Añadir un índice

Así, este método requiere más trabajo total que una generación del índice estándar y tarda mucho más tiempo en completarse.

Añadir un índice

Sin embargo, ya que permite a las operaciones normales que puedan continuar mientras que el índice está siendo construido, este método es útil para añadir nuevos índices en un entorno de producción.

Todo PostgreSQL

Añadir un índice

Por supuesto, la CPU y las cargas de entrada y salida estarán ralentizadas por la creación del índice.

Todo PostgreSQL

Añadir un índice

Si surge un problema durante la exploración de la tabla, como una violación de una unique key, el comando **CREATE INDEX** fallará pero deja detrás un índice “invalid”.

Todo PostgreSQL

Añadir un índice

Este índice no se tendrá en cuenta para propósitos de consulta, ya que podría estar incompleta; sin embargo, seguirá generando gasto de recursos.

El meta comando `\d` informará de un índice como INVÁLIDO:

Todo PostgreSQL

Añadir un índice

```
postgres=# \d tab
          Table "public.tab"
   Column |      Type      | Modifiers
-----+-----+-----
   col    | integer        |
Indexes:
    "idx" btree (col) INVALID
```

Todo PostgreSQL

Añadir un índice

El método de recuperación recomendado en tales casos es eliminar el índice y volver a intentar realizarlo con **CREATE INDEX CONCURRENTLY**.

Todo PostgreSQL

Añadir un índice

Otra diferencia habitual en el comando `CREATE INDEX` es que puede ejecutarse dentro de un bloque de transacción, pero `CREATE INDEX CONCURRENTLY` no puede.

Añadir una columna con una unique constraint

Esta operación bloqueará la tabla. Como solución, puede agregar la columna, añadir índice único al mismo tiempo, y luego añadir la restricción a la tabla:

Todo PostgreSQL

Añadir una columna con una unique constraint

```
CREATE UNIQUE INDEX CONCURRENTLY token_is_unique ON large_table(token);  
ALTER TABLE large_table ADD CONSTRAINT token UNIQUE USING INDEX token_is_unique;
```

Todo PostgreSQL

VACUUM FULL

VACUUM recupera almacenamiento ocupado por tuplas muertas. En el funcionamiento normal de PostgreSQL, las tuplas que se eliminan o quedan obsoletas por una actualización no se eliminan físicamente de su tabla, siguen presentes hasta que se realice un **VACUUM**.

Todo PostgreSQL

VACUUM FULL

VACUUM FULL reescribe todo el contenido de la tabla en un nuevo archivo de disco con espacio extra, lo que permite que el espacio no utilizado se devuelva al sistema operativo. Esta forma es mucho más lenta y requiere un bloqueo exclusivo en cada tabla mientras se está procesando.

Todo PostgreSQL

VACUUM FULL

Con la extensión `pg_repack` podemos realizar un **VACÍO COMPLETO** casi en línea (podría haber un bloqueo momentáneo).

VACUUM FULL

1. Crear una tabla de registro para registrar los cambios realizados en la tabla original;
2. Añadir un disparador en la tabla original, cargando INSERTs, UPDATEs y DELETE en nuestra tabla de registro;

Todo PostgreSQL

VACUUM FULL

3. Crear una nueva tabla que contiene todas las filas de la tabla antigua;
4. Construir índices en esta nueva tabla;

Todo PostgreSQL

VACUUM FULL

5. Aplicar todos los cambios que hayan podido surgir de la tabla de registro a la nueva tabla;
6. Intercambiar las tablas, incluyendo índices y tablas toast, utilizando los catálogos del sistema;

VACUUM FULL

7. Eliminar la tabla original;

Pg_repack sólo mantendrá un **ACCESS EXCLUSIVE**, un bloqueo por un corto período durante la configuración inicial (pasos 1 y 2) y durante la fase final de intercambio y eliminación (pasos 6 y 7).

Todo PostgreSQL

VACUUM FULL

Para el resto de su tiempo, `pg_repack` sólo necesita llevar a cabo un **ACCESS SHARE**, bloqueo en la tabla original, es decir, los INSERTs, UPDATEs y DELETE puede proceder como de costumbre.

VACUUM FULL

La realización de un empaquetado de la tabla completa requiere un espacio libre en el disco alrededor de dos veces tan grande como la tabla (s) de destino y sus índices.

ALTER TABLE SET TABLESPACE

Normalmente todos los datos de PostgreSQL residen en un solo directorio. Sin embargo, es posible que tenga algunos discos SSD adicionales, o todo lo contrario, algunos discos lentos, pero muy grandes.

ALTER TABLE SET TABLESPACE

Por lo que quieres poner algunos de los datos en otro conjunto de discos. Esto son los espacios de tabla (tablespace).

Todo PostgreSQL

ALTER TABLE SET TABLESPACE

El directorio predeterminado de la Tabla es `$PGDATA/base`. Pero es posible tener otros tablespace.

```
CREATE TABLESPACE xxx LOCATION '/wherver';
```

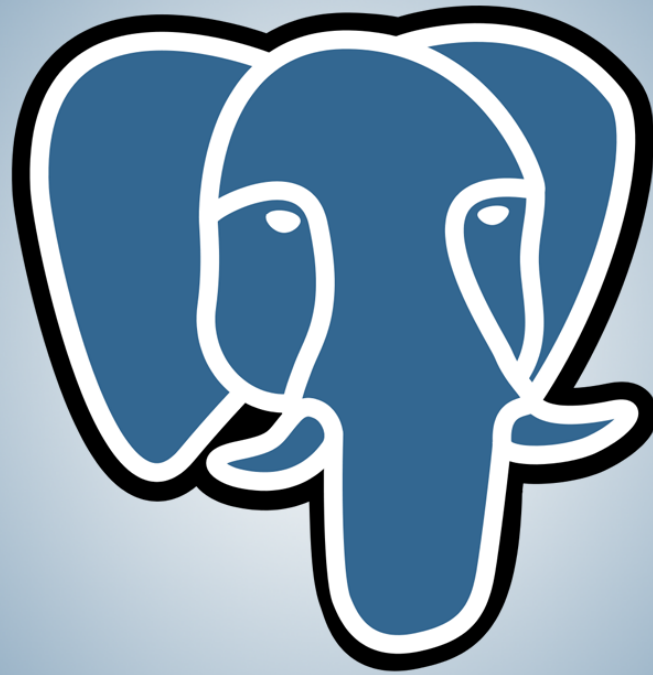
ALTER TABLE SET TABLESPACE

Después se pueden mover algunas tablas / índices a este nuevo espacio de tablas con el comando:

```
ALTER TABLE/INDEX whatever SET TABLESPACE xxx;
```

ALTER TABLE SET TABLESPACE

Esta es la operación de bloqueo. Para resolver este problema, puede utilizar `pg_repack` con la opción `--tablespace`.



todopostgresql.com