



Instituto Tecnológico de Mexicali

Fundamentos de Base de datos

José Ramon Bogarin Valenzuela

Unidad 2

EXPLAIN ANALYZE

Genesis Edel Ortiz Pérez

Mexicali BC a 06 de marzo del 2025

EXPLAIN y EXPLAIN ANALYZE

Estas dos herramientas son la base principal al momento de buscar problemas en cuanto al rendimiento de la consulta que realizamos, a pesar de ser muy utilizadas, no son algo nuevo, ya que han estado disponibles desde los comienzos de PostgreSQL, estas herramientas han ido evolucionando conforme el paso de los años, convirtiéndose en herramientas capaces de proporcionar los reportes más detallados en cuanto a la ejecución de las consultas (query), a continuación se explicaran sus funciones y algunos ejemplos.

Explain

- Este es un comando que nos permite la extracción de información sobre el plan de ejecución generado para un query por el planificador.
- Muestra cómo se escanearán las tablas referenciadas, escaneo secuencial o escaneo por índice.
- Este comando solo nos da una idea de cómo se ejecutará la consulta, mas no hace la ejecución.
- Nos permite detectar ineficiencias en las consultas y optimizarlas antes de ejecutarlas en producción
- Ayuda a entender como PostgreSQL accede a los datos (índices, escaneos, joins etc.).

Explain Analyze

En PostgreSQL, el comando EXPLAIN ANALYZE proporciona una vista detallada de cómo se ejecuta una consulta, mostrando los detalles de ejecución planificados y reales. Mejora la comprensión del rendimiento de las consultas, así como identificar los famosos cuellos de botella y optimizar consultas complejas. Al analizar el plan de ejecución, los administradores y desarrolladores de bases de datos ´pueden realizar ajustes informados en los índices, las combinaciones y otros elementos de consulta para mejorar la eficiencia.

Comando	¿ejecuta la consulta?	¿Muestra tiempos reales
EXPLAIN	NO	NO
EXPLAIN ANALYZE	SI	SI

Tipos de escaneo en el plan de ejecución

Tipo de escaneo	Descripción	Rendimiento
Seq scan	Escaneo secuencial de toda la tabla	Lento en tablas grandes
Index Scan	Escaneo por índice para buscar datos	Más rápido que Seq scan si hay pocos resultados

El índice

Un índice es un objeto más de la base de datos. Vinculado a una tabla, está formado por una o más columnas de la tabla y por un puntero a la fila de la tabla que representa, con la interesante característica de que las filas dentro del índice mantienen siempre un orden especificado por algún algoritmo.

Cuando insertamos una fila nueva en la tabla, ésta se graba en un lugar indeterminado del espacio libre del fichero físico, sin embargo, en el índice, el valor de la columna indexada (o columnas) se insertarán en su lugar correspondiente según dicho algoritmo. El índice nos garantiza que se mantendrá el orden también en operaciones de eliminación y de actualización de la columna o columnas indexadas.

Sin índice, cualquier búsqueda en la tabla obligará al planificador de consultas a hacer un escaneo completo secuencial de la tabla que, para tablas grandes, supone una importante inversión de tiempo.

En pocas palabras:

- ◆ Permite al servidor de base de datos encontrar y recuperar filas mucho más rápido
- ◆ Su principal beneficio es el de agilizar las consultas a los datos
- ◆ Agregan sobrecarga al sistema de base de datos en su conjunto
- ◆ Al crear un PK(Primary Key) automáticamente se crea un índice
- ◆ Es una tabla con pocas filas es casi inapreciable

Seq scan

A diferencia del índice, el escaneo secuencial consiste en una operación que hace un recorrido por toda la tabla para buscar los registros de manera secuencial, es decir fila por fila desde el principio hasta el final.

Esto puede ocurrir por situaciones como:

- No hay índices definidos para las columnas que se consultan

- El optimizador de consultas determina que escanear toda la tabla es más rápido que usar un índice (por ejemplo, cuando la tabla es pequeña)

Ejemplos de EXPLAIN Y EXPLAIN ANALYZE

Ejemplo sin índice (usando seq scan)

EXPLAIN:

Como ya lo hemos mencionado si no hay un índice en la columna, por ejemplo, de edad, PostgreSQL hace un escaneo secuencial:

```
EXPLAIN SELECT * FROM usuarios WHERE edad > 25;
```

Salida esperada:

Seq Scan on usuarios (cost=0.00..35.50 rows=5 width=100)

Explicación:

Seq scan on usuarios: hará un escaneo secuencial en la tabla usuarios

Cost=0.00..35.50: el primer número es el costo estimado de inicio y el segundo el costo total.

Rows=5: se estima que la consulta devolverá 5 filas

Width=100: tamaño promedio en bytes de cada fila

EXPLAIN ANALYZE:

```
EXPLAIN ANALYZE SELECT * FROM usuarios WHERE edad > 25;
```

Salida esperada:

Seq Scan on usuarios (cost=0.00..35.50 rows=5 width=100)

(actual time=0.020..0.045 rows=5 loops=1)

Explicación:

Actual time=0.020..0.045: tiempo real de ejecución en milisegundos

Rows=5: PostgreSQL realmente encontró 5 filas que cumplen la condición

Loops=1: indica cuantas veces se ha ejecutado un nodo específico del plan de ejecución

Ejemplo con índice (usando Index Scan)

Explain:

```
CREATE INDEX idx_edad ON usuarios WHERE edad>25;
```

```
EXPLAIN SELECT*FROM usuarios WHERE edad>25;
```

Salida esperada:

Index Scan using idx_edad on usuarios

(cost=0.15..8.23 rows=5 width=100)

Ahora está usando un índice que en este caso es la edad

EXPLAIN ANALYZE:

```
EXPLAIN ANALYZE SELECT*FROM usuarios WHERE edad>25;
```

Salida esperada:

Index Scan using idx_edad on usuarios

(cost=0.15..8.23 rows=5 width =100)

(actual time=0.010..0.25 rows = 5 loops=1)

Explicacion:

actual time=0.010..0.25: la consulta ahora es más rápida gracias al índice, PostgreSQL evita escanear toda la tabla y busca directamente en el índice que se le asigno, como en este caso se le asigno el índice de edad>25, lo que hará es buscar solo a los usuarios que sean mayores de 25, en vez de buscarlos por todas las columnas

Referencias

Using EXPLAIN ANALYZE in PostgreSQL to Optimize Queries. (2024, 23 diciembre).

W3resource. <https://www.w3resource.com/PostgreSQL/snippets/postgresql-explain-analyze.php>

PostgreSQL - explain. (s. f.). Solvetic. <https://www.solvetic.com/tutoriales/article/382-postgresql-explain/>

<https://youtu.be/-h41xlv1550>