



EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO



Derek Sainz Matinez

23490623

Base de datos

Jose Bogarin

ISC

5 de marzo del 2025

Uso del Índice con **EXPLAIN** y **EXPLAIN ANALYZE**

Los comandos **EXPLAIN** y **EXPLAIN ANALYZE** son herramientas de optimización de consultas en bases de datos relacionales, como PostgreSQL, MySQL y MariaDB. Sirven para analizar cómo el motor de la base de datos ejecutará una consulta y determinar si los índices están siendo utilizados correctamente.

EXPLAIN

Este comando muestra el plan de ejecución estimado de una consulta sin ejecutarla. Es útil para identificar si una consulta usará índices o si se debe optimizar.

Ejemplo en PostgreSQL o MySQL:

sql

```
EXPLAIN SELECT * FROM empleados WHERE id = 10;
```

Salida típica:

pgsql

```
Index Scan using empleados_pkey on empleados (cost=0.29..8.30  
rows=1 width=48)
```

Explicación:

- **Index Scan using empleados_pkey**: Indica que se usa un índice para buscar el registro.
- **cost=0.29..8.30**: Estima el costo de ejecutar la consulta.
- **rows=1**: Número estimado de filas devueltas.
- **width=48**: Tamaño en bytes de cada fila.

EXPLAIN ANALYZE

A diferencia de **EXPLAIN**, este comando ejecuta realmente la consulta y muestra estadísticas reales, incluyendo tiempos de ejecución y número de filas procesadas.

Ejemplo en PostgreSQL:

sql

```
EXPLAIN ANALYZE SELECT * FROM empleados WHERE id = 10;
```

Salida típica:

pgsql

```
Index Scan using empleados_pkey on empleados (cost=0.29..8.30  
rows=1 width=48) (actual time=0.023..0.025 rows=1 loops=1)
```

Diferencias clave:

- (actual time=0.023..0.025): Tiempo real de ejecución en milisegundos.
- rows=1 loops=1: Número real de filas y repeticiones.

Introducción - Utilizar MySQL EXPLAIN

Prefijo una consulta con SELECT con EXPLAIN

- MySQL no ejecutará la consulta, solo la analiza
- EXPLAIN nos ayuda a entender cómo y cuándo MySQL, utilizará índices
- EXPLAIN devuelve una tabla de datos de la que se identifica mejoras potenciales a Optimizar
- Las consultas de tres maneras
 1. Modificar o crea índices
 2. Modificar la estructura de la consulta
 3. Modificar la estructura de datos
- Consultas optimizadas = resultados más rápidos, menor carga del servidor

Introducción - Revisión de la indexación

- Estructura rápida y compacta para identificar las ubicaciones de las filas
- Mantenga los índices en la memoria recortando la grasa:
 1. ¿Puedo reducir los caracteres de ese índice VARCHAR?
 2. ¿Puedo usar un TINYINT en lugar de un BIGINT?
 3. ¿Puedo usar un INTEGER para describir un estado o una bandera (más bien que una descripción textual)?
- Corte el conjunto de resultados tan pronto como sea posible

- MySQL sólo utilizará un índice por consulta / tabla - no puede combinar dos índices separados para hacer útil su uso*

Comprensión y preparación trae consigo la *Estrategia de Indexación*

EXPLAIN – Ejemplo funcionando

```
EXPLAIN SELECT * FROM asistentes WHERE conference_id = 123 AND registration_status > 0
```

tabla	possible_keys	key	row
asistentes	NULL	NULL	14052

Las tres columnas más importantes devueltas por EXPLAIN

- Posibles claves
 - Todos los posibles índices que MySQL podría haber utilizado
 - Basado en una serie de búsquedas y cálculos muy rápidos
- Seleccionada de Clave
- Filas escaneadas
 - Indicación del esfuerzo requerido para identificar su conjunto de resultados

Utilidad de EXPLAIN

Con la ayuda de EXPLAIN,

- Es posible ver dónde se debe agregar índices a tablas para que la sentencia se ejecute más rápido mediante el uso de índices para buscar filas.
- También se puede utilizar EXPLAIN para comprobar si el optimizador une a las tablas (JOIN) en un orden óptimo. Para dar una sugerencia al optimizador de que utilice un orden de combinación correspondiente al orden en que se nombran las tablas en una sentencia SELECT, se debe comenzar la sentencia con SELECT STRAIGHT_JOIN en lugar de SELECT.

La traza del optimizador a veces puede proporcionar información complementaria a la de EXPLAIN. Sin embargo, el formato de rastreo del optimizador y el contenido están sujetos a cambios entre versiones.

Si tiene un problema con los índices que no se utilizan cuando cree que deberían hacerlo, ejecute ANALYZE TABLE para actualizar estadísticas de tabla, como la cardinalidad de las claves, que pueden afectar las opciones que el optimizador realiza.

Utilizando el índice (INDEX)"

Celebre, porque:

- MySQL obtuvo sus resultados sólo consultando el índice,
 - Que bien podría haber estado sentado en la memoria
- MySQL no necesita ni siquiera mirar a la mesa para obtener sus resultados
 - Abrir una mesa puede ser una operación costosa.
- MySQL puede responder a la siguiente consulta más rápidamente
 - ¿La forma más rápida de obtener sus datos?

Particularmente útil ...

- Cuando sólo está interesado en una sola fecha o un ID
- El COUNT (), SUM (), AVG () etc. de un campo

Usar el índice (INDEX)" – Ejemplo

```
EXPLAIN SELECT AVG (age) FROM asistentes WHERE conference_id = 123
```

tabla	possible_keys	key	row	extra
asistentes	conference_id	conference_id	331	

¡Nada es realmente incorrecto con esta consulta - podría ser más rápido!

```
ALTER TABLE asistentes ADD INDEX conf_age (conference_id, age);
```

```
EXPLAIN SELECT AVG (age) FROM asistentes WHERE conference_id = 123
```

tabla	possible_keys	key	row	extra
asistentes	conference_id, conf_surname	conf_surname	331	Using index

Fuera del almacenamiento en caché, la manera más rápida de obtener sus datos (No es una garantía)

<https://es.stackoverflow.com/questions/53779/mysql-para-qu%C3%A9-sirve-explain-en-una-consulta-y-c%C3%B3mo-interpretar-los-resultados>