



UNIVERSIDADE FEDERAL DO PIAUÍ - UFPI
CURSO: BACHARELADO EM SISTEMAS DE INFORMAÇÃO
PROFESSOR: Glauber Dias Gonçalves
C.H.: 60h CRÉDITOS: 3.1.0 PERÍODO: 2022.2



MY SQL

Milena Lira de Andrade Sousa
Luís Eduardo Silva Brito,
Víctor Macêdo Carvalho,

1. Introdução

O MySQL é um banco de dados conhecido por fornecer uma fácil acessibilidade e utilização, por aderir a linguagem SQL como interface. Grandes empresas como a NASA, HP, Bradesco e Sony fazem o uso desse banco de dados, assim como várias outras não listadas aqui.

Sua interface simples e também sua capacidade de funcionar em vários sistemas operacionais, são alguns dos motivos para este programa ser tão usado atualmente, e seu uso só tende a crescer cada vez mais. O MySQL é protegido por uma licença de software livre, desenvolvida pela GNU.

Ele surgiu na Suécia através de dois suecos e um finlandês chamados David Axmark, Allan Larsson e Michael Monty Widenius na década de 1980. Hoje seu desenvolvimento e manutenção empregam aproximadamente 400 profissionais no mundo inteiro, e mais de mil contribuem testando o software, integrando-o a outros produtos, e escrevendo (atualizando) a respeito dele.

2. Metodologia

A otimização de consultas no MySQL é uma tarefa crítica para garantir que seu aplicativo seja rápido e responsivo. Para a demonstração dessa otimização foram utilizados alguns passos para seguir na metodologia de otimização de consultas no MySQL:

- **Análise de consultas lentas:** identifique quais consultas estão levando muito tempo e precisam ser otimizadas.
- **Análise de índices:** verifique se os índices estão sendo usados corretamente e se existem oportunidades para adicionar ou remover índices.
- **Análise de consultas:** verifique a estrutura e o desempenho das consultas, procurando otimizar a escrita da consulta e evitar funções ineficientes no filtro WHERE.

A otimização de consultas é um processo contínuo e pode exigir muitos ajustes e testes antes de chegar a resultados satisfatórios. No entanto, seguir

uma metodologia sólida e documentada pode ajudar a garantir que suas consultas sejam otimizadas de forma eficiente.

3. Resultados

A diretiva de “Explain” é usada para obter um plano de execução da consulta, ela aponta como o MySQL realiza uma determinada consulta, possibilitando ter uma redução no gasto de memória, e uma melhora no desempenho do SGBD.

Como por exemplo, utilizando duas tabelas de um banco de dados, e supondo que a tabela de clientes possui 2000 registros e que a tabela de pedidos possui 500 mil registros.

Tabela 1. Clientes


#	Nome	Tipo
1	id 	int(11)
2	nome	varchar(100)
3	email	varchar(100)
4	senha	varchar(100)
5	cpf	varchar(100)

Tabela 2. Pedidos

#	Nome	Tipo
1	id 	int(11)
2	id_cliente	int(11)
3	numero	int(10)
4	descricao	varchar(100)
5	valor	decimal(12,2)
6	status	varchar(20)

No exemplo abaixo, predigamos que seja necessário realizar a junção dessas duas tabelas pelo id do cliente,

```
select * from clientes, pedidos
where clientes.id=pedidos.id_cliente and clientes.id='23'.
```

Utilizando a diretiva explain para calcular o desempenho do SGBD,

```
explain select * from clientes, pedidos
where clientes.id=pedidos.id_cliente and clientes.id='23';
```

Tabela 1: Explain da junção das tabelas de cliente e pedido

I d	table	type	Possible_ke ys	Key	Key_le n	Rowls	Ref	Filtere d
--------	-------	------	-------------------	-----	-------------	-------	-----	--------------

1	cliente	Const	Primary	Primary	4	1	Const	100.00
1	pedido	ALL	NULL	NULL	NULL	500000	NULL	10.00

O valor de “Rows” é 500 mil, ou seja, o MySQL percorre todos os registros da tabela de pedido para realizar o junção das tabelas, e é possível notar que não foi utilizado índices pois os valores de “Key” e “Possible_keys” são nulos.

Com o uso de índices é possível acelerar as consultas em banco de dados e aumentar o desempenho em relação a recuperação dos dados, possibilitando o acesso de forma direta aos atributos da tabela.

Criando um índice para a coluna “id_cliente”,

alter table pedidos add index (id_cliente);

Utilizando a Tabela 2 como exemplo da diretiva explain para calcular o desempenho do SGBD,

Tabela 2: Explain da junção das tabelas de cliente e pedido

Id	table	type	Possible_keys	Key	Key_len	Rows	Ref	Filtered
1	cliente	Const	Primary	Primary	4	1	Const	100.00
1	pedido	Ref	Id_cliente	Id_cliente	4	15	Const	100.00

O valor de “Rows” é 15 e podemos observar que o MySQL fez 16 acessos a tabela, consequentemente, melhorando o desempenho e diminuindo o número de acessos.

4. Conclusão

Depreende-se, diante do exposto, que é necessário para a otimização a utilização de índices adequados para as colunas utilizadas nas consultas. Evitar consultas com JOINS desnecessários, ocorrer a limitação do número de linhas retornadas pelas consultas. Utilização de consultas bem escritas e a evitação de subconsultas complexas. Desse modo, é necessário que mantenha as tabelas organizadas, excluindo linhas desnecessárias e otimizando a estrutura de dados.

Referências:

DUVIDAS, site. Disponível em: <https://duvidas.bem-vindo.net/348/otimizando-consultas-com-o-explain-do-mysql>

