

Módulo 6: Índices e Otimização de Consultas

Módulo 6: Índices e Otimização de Consultas

Os índices são fundamentais para melhorar a performance de consultas SQL, especialmente em bancos de dados com grande volume de dados. Este módulo aborda como criar índices, analisar a performance das consultas e adotar boas práticas para otimização.

1. Criação de Índices

O que são Índices e sua Importância

Um índice em SQL é uma estrutura de dados que melhora a velocidade das operações de busca em uma tabela. É semelhante ao índice de um livro, que permite localizar informações rapidamente sem percorrer todas as páginas.

● Por que usar índices?

- Aceleram operações de busca (**SELECT**) em colunas indexadas.
- Melhoram a eficiência em consultas com cláusulas **WHERE**, **JOIN**, e ordenações (**ORDER BY**).
- Reduzem o custo computacional em tabelas grandes.

● Cuidado com os índices:

- Criar muitos índices pode aumentar o tempo de inserção, atualização e exclusão (**INSERT**, **UPDATE**, **DELETE**), pois o índice também precisa ser ajustado.

Criação de Índices

CREATE INDEX

Cria um índice básico em uma ou mais colunas de uma tabela.

Exemplo Prático: Índice em uma coluna **nome** na tabela **clientes**:

```
CREATE INDEX idx_nome_cliente ON clientes(nome);
```

- **idx_nome_cliente:** Nome do índice.

- **clientes(nome):** Define que o índice será criado na coluna **nome**.

UNIQUE INDEX

Garante que os valores na coluna ou colunas indexadas sejam únicos.

Exemplo Prático: Índice único na coluna **email**:

```
CREATE UNIQUE INDEX idx_email_cliente ON clientes(email);
```

Aplicabilidade:

- Garantir unicidade em colunas críticas, como emails ou números de identificação.
-

Índices Compostos

Um índice composto cobre duas ou mais colunas, otimizando consultas que filtram ou ordenam com base em múltiplos campos.

Exemplo Prático: Criar um índice composto em **nome** e **cidade** na tabela **clientes**:

```
CREATE INDEX idx_nome_cidade ON clientes(nome, cidade);
```

Aplicabilidade:

- Consultas que utilizam ambas as colunas no filtro ou ordenação.
- Exemplo de consulta otimizada:

```
SELECT *  
FROM clientes  
WHERE nome = 'João Silva' AND cidade = 'São Paulo';
```

2. Análise de Performance

EXPLAIN: Análise de Consultas SQL

O comando **EXPLAIN** exibe o plano de execução de uma consulta, indicando como o banco de dados planeja executar a operação.

Como Usar

Exemplo Prático: Verificar a performance de uma consulta com **EXPLAIN**:

```
EXPLAIN SELECT * FROM clientes WHERE nome = 'João Silva';
```

Saída Interpretada:

- **type**: Tipo de busca (ideal: **index** ou **const**).
 - **key**: Índice usado na consulta.
 - **rows**: Número de linhas examinadas (quanto menor, melhor).
 - **Extra**: Informações adicionais (evite **Using full table scan**).
-

Uso de Índices em Consultas JOIN e WHERE

Índices são especialmente úteis em consultas complexas que utilizam **JOIN** ou **WHERE**.

Exemplo Prático:

Imagine duas tabelas, **clientes** e **pedidos**, relacionadas por **id_cliente**. Um índice na coluna **id_cliente** pode acelerar a consulta.

Sem Índice:

```
SELECT *  
FROM clientes  
JOIN pedidos ON clientes.id_cliente = pedidos.id_cliente;
```

Com Índice:

```
CREATE INDEX idx_id_cliente ON pedidos(id_cliente);
```

Resultado:

- O plano de execução mostrará uma redução no número de linhas analisadas.
-

3. Otimização de Consultas

Uso de Índices para Acelerar Consultas

Os índices são usados automaticamente pelo banco de dados em consultas que envolvem colunas indexadas. Isso reduz o número de registros analisados.

Exemplo Prático:

Consulta com índice na coluna **nome**:

```
SELECT * FROM clientes WHERE nome = 'Maria Souza';
```

- **Sem índice:** O banco faz uma **varredura completa da tabela** (*Full Table Scan*).
- **Com índice:** Apenas os registros correspondentes são acessados diretamente.

Como Evitar Full Table Scans

Um **Full Table Scan** ocorre quando o banco analisa cada linha de uma tabela para encontrar os registros desejados. Isso é ineficiente em tabelas grandes.

Estratégias para Evitar:

1. **Criação de Índices Apropriados:**

```
CREATE INDEX idx_cidade ON clientes(cidade);
```

2. **Uso Correto de Condições:** Sempre utilize colunas indexadas em filtros **WHERE**:

```
SELECT * FROM clientes WHERE cidade = 'São Paulo';
```

3. **Evitar Operações em Colunas:** Operações nas colunas podem impedir o uso do índice.
 - **Ineficiente:**

```
SELECT * FROM clientes WHERE UPPER(nome) = 'JOÃO SILVA';
```

- **Eficiente:**

```
SELECT * FROM clientes WHERE nome = 'João Silva';
```

4. **Limitar Registros Retornados:** Use **LIMIT** para reduzir o número de registros processados:

```
SELECT * FROM clientes LIMIT 10;
```

Boas Práticas de Escrita de Consultas SQL

1. Selecione Apenas as Colunas Necessárias:

- Ineficiente

```
SELECT * FROM clientes;
```

- Eficiente

```
SELECT nome, cidade FROM clientes;
```

2. Evite Subconsultas Desnecessárias:

- Substitua subconsultas por **JOINS** sempre que possível.

- Ineficiente:

```
SELECT nome FROM clientes WHERE id_cliente = (SELECT id_cliente FROM pedidos  
WHERE valor_total > 100);
```

- Eficiente:

```
SELECT clientes.nome  
FROM clientes  
JOIN pedidos ON clientes.id_cliente = pedidos.id_cliente  
WHERE pedidos.valor_total > 100;
```

3. Utilize Funções de Agregação Com Cuidado:

- Agrupe e filtre somente onde necessário:

```
SELECT cidade, COUNT(*) AS total_clientes
FROM clientes
GROUP BY cidade
HAVING total_clientes > 10;
```

4. **Analise Regularmente a Performance:** Use **EXPLAIN** para identificar gargalos e ajustar índices e consultas.

Aplicabilidade Prática

1. **Sistemas de Vendas:**
 - Criar índices em colunas como **id_cliente** e **data_pedido** para acelerar relatórios.
2. **E-commerce:**
 - Melhorar a performance de consultas em catálogos com índices em **categoria** e **preco**.
3. **Gestão de Projetos:**
 - Índices compostos em colunas como **id_projeto** e **id_funcionario** para relatórios de produtividade.

Com o uso correto de índices e práticas otimizadas, é possível obter consultas rápidas e eficientes, garantindo o desempenho de aplicações mesmo em grandes volumes de dados.

Vamos Praticar

Questões Práticas (10)

1. **Criação de Índices:**

Crie um índice básico na coluna `nome` da tabela `clientes` e explique como esse índice pode melhorar a performance em uma consulta `SELECT`.

2. **Índice Único:**

Crie um índice único na coluna `email` da tabela `clientes` para garantir que não existam emails duplicados. Explique a importância desse índice para a integridade dos dados.

3. **Índices Compostos:**

Crie um índice composto nas colunas `nome` e `cidade` da tabela `clientes`. Em seguida, explique como esse índice pode otimizar uma consulta que filtra clientes por nome e cidade simultaneamente.

4. **Análise de Performance com EXPLAIN:**

Execute o comando `EXPLAIN` em uma consulta que utiliza um `JOIN` entre as tabelas `clientes` e `pedidos`. Interprete os seguintes campos da saída: `type`, `key`, `rows` e `Extra`.

5. **Uso de Índices em JOINS:**

Crie um índice na coluna `id_cliente` da tabela `pedidos` e demonstre como ele melhora a performance de uma consulta que realiza um `JOIN` entre `clientes` e `pedidos`.

6. **Evitar Full Table Scans:**

Crie um índice na coluna `nome` da tabela `clientes` e compare o plano de execução de uma consulta `SELECT` com e sem índice.

7. **Limitação de Registros com LIMIT:**

Execute uma consulta que retorne apenas os 5 primeiros clientes da tabela `clientes`, ordenados por `nome`. Explique como o uso de `LIMIT` pode melhorar a performance.

8. **Evitar Operações em Colunas:**

Otimize uma consulta que filtra clientes usando `UPPER(nome)` para evitar operações que impeçam o uso de índices. Explique a mudança feita.

9. **Criação de Índices Apropriados:**

Crie índices nas colunas `categoria` e `preco` da tabela `produtos`. Explique como esses índices podem melhorar consultas que envolvem filtros nessas colunas.

10. **Identificação de Gargalos com EXPLAIN:**

Analise uma consulta complexa utilizando `EXPLAIN`, identifique possíveis gargalos no desempenho e proponha melhorias usando índices ou reestruturação da consulta.

Questões Teóricas (10)

1. **Conceito de Índices:**
Explique o que são índices em SQL e por que eles são importantes para a performance de consultas em tabelas grandes.
2. **Tipos de Índices:**
Diferencie entre um índice básico (`CREATE INDEX`) e um índice único (`CREATE UNIQUE INDEX`). Dê exemplos de aplicações práticas para cada tipo.
3. **Índices Compostos:**
Explique o que é um índice composto e em quais situações ele é mais útil.
4. **Impacto dos Índices em Operações DML:**
Discuta como índices podem afetar o desempenho de operações como `INSERT`, `UPDATE` e `DELETE`.
5. **EXPLAIN em Consultas SQL:**
Qual é o objetivo do comando `EXPLAIN` no MySQL? Quais informações ele fornece para ajudar na otimização de consultas?
6. **Full Table Scan:**
O que é um Full Table Scan e por que ele pode ser prejudicial ao desempenho? Como evitá-lo?
7. **Limitação de Registros em Consultas:**
Por que o uso do comando `LIMIT` pode melhorar a performance de uma consulta? Cite um exemplo prático.
8. **Índices e JOINS:**
Explique como os índices podem melhorar a performance de consultas que utilizam `JOINS`.
9. **Boas Práticas em Consultas SQL:**
Liste e explique três boas práticas para escrever consultas SQL eficientes.
10. **Funções de Agregação e Performance:**
Como o uso de funções de agregação (`COUNT`, `SUM`, `AVG`) pode impactar a performance de uma consulta? O que pode ser feito para otimizar consultas que utilizam essas funções?

Instruções para a Entrega das Atividades

1. **Elaboração e Envio do Arquivo**
 - Responda todas as questões de forma clara e objetiva.
 - Gere um arquivo no formato **.PDF** contendo as respostas de cada questão.
 - Envie o arquivo para os e-mails dos professores responsáveis.
2. **Validação da Atividade**
 - Após o envio do arquivo, procure o(s) professor(es) para realizar a validação da atividade.
 - **Não inicie a próxima atividade sem antes validar a anterior com o professor.**
3. **Forma de Validação**
 - **Explicação Verbal:** Explique cada resposta verbalmente ao(s) professor(es).
 - **Perguntas e Respostas:** Esteja preparado para responder aos questionamentos do(s) professor(es) sobre o conteúdo das respostas.
 - **Orientação:** Receba orientações sobre a apresentação do(s) tema(s).