



Atividade realizada no Mysql

```
CREATE DATABASE livraria;
USE livraria;

-- Criando a tabela de autores

CREATE TABLE autores (
    id INT PRIMARY KEY AUTO_INCREMENT,
    nome VARCHAR(100),
    nacionalidade VARCHAR(50)
);

-- Criando a tabela livros
CREATE TABLE livros (
    id INT PRIMARY KEY AUTO_INCREMENT,
    titulo VARCHAR(100),
    autor_id INT,
    editora VARCHAR(50),
    ano_publicacao INT,
    preco DECIMAL(10,2),
    FOREIGN KEY (autor_id) REFERENCES autores(id)
);

--
-----

INSERT INTO autores(nome, nacionalidade)
VALUES ('Stephen King','Estados Unidos'),
('J.K.Rowling','Reino Unido'),
('Agatha Cristie','Reino Unido');

INSERT INTO livros(titulo, autor_id, editora, ano_publicacao,
preco)
VALUES ('It', 1 , 'Suma', 1986, 49.99),
('Harry Potter e a Pedra Filosofal', 2 , 'Rocco', 1997, 39.99),
('Assassinato no Expresso Oriente', 3 , 'Intrínseca', 1934,
29.99)
;

-----
```



```
=====
-- Atividades:
=====
```

```
=====
-- 1. Criação de Índices:
=====
```

-- Índice simples: Crie um índice em uma coluna que é frequentemente utilizada em cláusulas WHERE.

```
CREATE INDEX idx_Autores ON autores(nome);
SHOW INDEX FROM autores;
```

-- Índice composto: Crie um índice composto em duas ou mais colunas que são frequentemente utilizadas juntas em consultas.

```
CREATE INDEX idx_Livros ON livros(titulo,preco);
SHOW INDEX FROM livros;
```

-- Índice único: Crie um índice único para garantir a unicidade dos valores em uma coluna.

```
CREATE UNIQUE INDEX idx_IDlivros ON livros(id);
```

```
=====
-- 2. Análise de Desempenho:
=====
```

-- Consultas sem índice: Execute uma consulta que não utilize o índice e meça o tempo de execução.

```
SELECT * FROM Livros;
SELECT * FROM Livros WHERE ano_publicacao = 1997;
```

```
EXPLAIN ANALYZE SELECT * FROM livros WHERE ano_publicacao = 1997 ;
-- 1| SIMPLE| livros | ALL| rows:
3| 33.33| Using where
```

```
-- -> Filter: (livros.ano_publicacao = 1997) (cost=0.55 rows=1)
(actual time=0.0285..0.0316 rows=1 loops=1)
-- -> Table scan on livros (cost=0.55 rows=3) (actual
time=0.0268..0.0305 rows=3 loops=1)
```



-- Consultas com índice: Execute a mesma consulta após criar o índice e compare o tempo de execução.

```
CREATE INDEX idx_anoPubli ON livros (ano_publicacao);  
SELECT * FROM Livros WHERE ano_publicacao = 1997;
```

```
EXPLAIN ANALYZE SELECT * FROM livros WHERE ano_publicacao = 1997 ;  
--      ->   Index   lookup   on   livros   using   idx_anoPubli  
(ano_publicacao=1997)   (cost=0.35   rows=1)   (actual time=0.024..0.0256  
rows=1 loops=1)
```

-- Variação de carga: Aumente gradualmente a quantidade de dados na tabela e repita as consultas para observar o impacto do índice em diferentes tamanhos de dados.

```
SELECT * FROM Livros WHERE ano_publicacao = 1997;  
EXPLAIN ANALYZE SELECT * FROM livros WHERE ano_publicacao = 1997 ;  
--      '->   Index   lookup   on   livros   using   idx_anoPubli  
(ano_publicacao=1997)           (cost=0.35   rows=1)           (actual  
time=0.0213..0.0227 rows=1 loops=1)\n'
```

```
SELECT * FROM autores;  
INSERT INTO livros(titulo, autor_id, editora, ano_publicacao, preco)  
VALUES ('Harry Potter e a Câmara Secreta', 2, 'Suma', 1998,  
49.99),  
      ('Harry Potter e o Prisioneiro de Azkaban', 2  
, 'Suma', 1999, 39.99),  
      ('Harry Potter e o Cálice de Fogo', 2 , 'Suma', 2000, 29.99)  
;
```

```
SELECT * FROM Livros WHERE ano_publicacao = 1997;  
EXPLAIN ANALYZE SELECT * FROM livros WHERE ano_publicacao = 1997 ;  
--      ->   Index   lookup   on   livros   using   idx_anoPubli  
(ano_publicacao=1997)           (cost=0.35   rows=1)           (actual  
time=0.0848..0.0915 rows=1 loops=1)
```

```
SELECT * FROM livros;  
SELECT * FROM autores;
```



```
SELECT l.titulo, a.nome
FROM livros l
JOIN autores a ON a.id = l.id;
```

```
EXPLAIN SELECT l.titulo, a.nome
FROM livros l
JOIN autores a ON a.id = l.id
;
-- -> Nested loop inner join      (cost=1.6  rows=3)  (actual
time=0.032..0.0396 rows=3 loops=1)
-- -> Covering index scan on a using idx_Autores  (cost=0.55 rows=3)
(actual time=0.0194..0.0227 rows=3 loops=1)
-- -> Single-row index lookup on l using PRIMARY (i...
```

```
=====
-- 3. Otimização de Consultas:
=====
```

-- Escolha do índice: Analise o plano de execução de uma consulta para identificar qual índice está sendo utilizado.

```
EXPLAIN SELECT titulo, preco FROM livros;
-- 1  SIMPLE      livros      index      idx_Livros 409      6
      100.00      Using index
```

-- Criação de índices faltantes: Identifique consultas que poderiam ser otimizadas com a criação de um novo índice.

```
SELECT * FROM livros;
SELECT * FROM autores;
EXPLAIN SELECT nome FROM autores WHERE nacionalidade = 'Estados
Unidos';
-- 1 SIMPLE      autores      ALL      3
33.33 Using where
```

```
CREATE INDEX idx_nacionalidade ON autores(nacionalidade);
```

```
SELECT nome FROM autores WHERE nacionalidade = 'Estados Unidos';
EXPLAIN SELECT nome FROM autores WHERE nacionalidade = 'Estados
Unidos';
```



```
-- 1 SIMPLE      autores      ref  idx_nacionalidadeae
idx_nacionalidadeae  203  const 1    100.00
```

-- Remoção de índices não utilizados: Identifique índices que não estão sendo utilizados e remova-os para melhorar o desempenho de inserções, atualizações e deleções.

```
SHOW INDEX FROM livros;
```

```
SHOW INDEX FROM autores;
```

```
DROP INDEX idx_Livros ON livros;
```

```
=====
```

-- 4. Fragmentação de Índices:

```
=====
```

-- Simulação de fragmentação: Inserir ou excluir um grande número de registros para fragmentar um índice.

```
SHOW INDEX FROM livros;
```

```
EXPLAIN ANALYZE SELECT * FROM Livros WHERE preco < 40;
```

```
-- -> Filter: (livros.preco < 40.00)  (cost=0.55 rows=1) (actual
time=0.0362..0.0415 rows=2 loops=1)
```

```
      -- -> Table scan on Livros  (cost=0.55 rows=3) (actual
time=0.0326..0.0389 rows=3 loops=1)
```

```
SET SQL_SAFE_UPDATES = 0;
```

```
DELETE FROM livros WHERE preco < 30.00;
```

```
SELECT * FROM livros;
```

-- INSERINDO DADOS, DELETANDO E ATUALIZANDO PARA FRAGMENTAR O INDICE

```
INSERT INTO livros (titulo, autor_id, editora, ano_publicacao,
preco)
```

```
VALUES
```

```
('Livro A', 1, 'Editora X', 2001, 50.00),
```

```
('Livro B', 2, 'Editora Y', 2002, 45.00),
```

```
('Livro C', 2, 'Editora Z', 1999, 60.00),
```

```
('Livro D', 2, 'Editora Z', 1999, 60.00),
```

```
('Livro E', 2, 'Editora Z', 1999, 60.00),
```

```
('Livro F', 2, 'Editora Z', 1999, 60.00),
```

```
('Livro G', 2, 'Editora Z', 1999, 60.00),
```

```
('Livro H', 2, 'Editora Z', 1999, 60.00),
```

```
('Livro I', 2, 'Editora Z', 1999, 60.00),
```



```
('Livro J', 3, 'Editora Z', 1999, 60.00),
('Livro L', 1, 'Editora Z', 1999, 60.00),
('Livro M', 1, 'Editora Z', 1999, 60.00),
('Livro N', 1, 'Editora Z', 2000, 60.00),
('Livro O', 3, 'Editora Z', 1999, 60.00),
('Livro P', 3, 'Editora Z', 1999, 60.00),
('Livro Q', 3, 'Editora Z', 1999, 60.00),
('Livro R', 3, 'Editora Z', 1999, 60.00),
('Livro S', 3, 'Editora Z', 1999, 60.00),
('Livro T', 3, 'Editora Z', 1999, 60.00),
('Livro U', 3, 'Editora Z', 2021, 60.00),
('Livro V', 1, 'Editora Z', 2010, 60.00),
('Livro Z', 3, 'Editora K', 2010, 75.00);
```

```
EXPLAIN ANALYZE SELECT * FROM Livros WHERE preco < 40;
-- '-> Filter: (livros.preco < 40.00) (cost=2.65 rows=8) (actual
time=0.0889..0.104 rows=1 loops=1)\n      -> Table scan on Livros
(cost=2.65 rows=24) (actual time=0.0841..0.0982 rows=24 loops=1)\n'
```

```
DELETE FROM livros WHERE preco = 60.00;
```

```
SHOW STATUS LIKE 'Handler_read_key';
```

-- Reorganização: Reorganize o índice fragmentado e compare o desempenho das consultas antes e depois da reorganização.

Não foi possível

```
=====
-- 5. Índices e Transações:
=====
```

-- Índices e bloqueios: Analise o impacto da criação de índices em transações concorrentes.

-- Índices e locking: Observe como os índices afetam o mecanismo de locking do SGBD.

```
=====
-- 7. Índices e Materiais:
=====
```



-- Criação de materiais: Crie materiais (views materializadas) e índices sobre eles.

```
SHOW INDEX FROM livros;  
SHOW INDEX FROM autores;
```

```
CREATE VIEW vwProduto AS  
SELECT t.titulo, a.nome  
FROM livros t  
JOIN autores a ON a.id = t.autor_id;  
  
SELECT * FROM vwProduto;
```

-- Comparação de desempenho: Compare o desempenho de consultas utilizando índices em tabelas e em materiais.

=====

-- Questões para Reflexão:

=====

-- Qual o impacto da criação de um índice no desempenho de operações de inserção, atualização e deleção?

Poderá ocorrer uma sobrecarga no desempenho deste índice, o mesmo pode vir a ser corrompido. Resultando em uma consulta não satisfatória.

-- Quando é recomendado criar um índice composto?

Na utilização de varias colunas usadas frequentemente usadas em clausulas WHERE, JOIN, ORDER BY e Group By

-- Como a escolha do tipo de índice (clusterizado ou não-clusterizado) afeta o desempenho?

Os índices clusterizados são mais rápidos em relação aos não-clusterizados. Devido ao fato de as informações do índice e sua coluna respectiva estarem ordenadas e próximas na memória física do banco. Enquanto os não clusterizados: O índice mantém uma cópia das colunas indexadas, mas apenas como um **ponteiro** para as localizações físicas dos dados reais.



-- Quais são as melhores práticas para a manutenção de índices em um ambiente de produção?

Monitoramento dos índices através do EXPLAIN E EXPLAIN ANALYZE, verificar os índices fragmentados e remover índices não utilizados, dentre outros.

-- Como você pode identificar um índice que precisa ser reorganizado?

Analizando a fragmentação do índice.