



# OTIMIZAÇÃO ÍNDICES

GRUPO: AUGUSTO SAVI, DARLAN URBANO ANDRÉ, FÁBIO HENRIQUE PEREIRA, JOÃO PEDRO GOETTEM SAGGIORATO, MATTHEW HEIDMANN HELEODORO, MATEUS BALTAZAR DA ROCHA, NATALIE DOS SANTOS LIMA.

# PROBLEMA

Criamos um banco de dados simples e adicionamos alguns dados fictícios e queremos saber a quantidade de livros lançados no mesmo dia.

```
+-----+  
| Tables_in_livraria |  
+-----+  
| autores             |  
| capitulos          |  
| livros              |  
| vendas              |  
+-----+
```

```
select count(id) from livros;
```

```
+-----+  
| count(id) |  
+-----+  
|      20000 |  
+-----+
```

# PROBLEMA

- desc livros;

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
nome	varchar(100)	NO		NULL	
data_de_lancamento	date	NO		NULL	
autor_id	int(11)	NO		NULL	
preco	decimal(10,2)	NO		NULL	

# PROBLEMA

```
select *, (select count(livros2.data_de_lancamento) from livros as livros2 where  
livros2.data_de_lancamento = livros1.data_de_lancamento) as anteriores from livros as livros1  
order by livros1.data_de_lancamento;
```

Pegaremos os dados da tabela livro junto com um sub select para contar os livros com mesma data de lançamento ordenada por data de lançamento.

Essa query durou mais ou menos **2 minutos**.

Ai você deve estar se perguntando porque demorou eternos 2 minutos se só temos 20 mil registros

# PROBLEMA

```
explain select *, (select count(livros2.data_de_lancamento) from livros as livros2 where  
livros2.data_de_lancamento = livros1.data_de_lancamento) as anteriores from livros as livros1  
order by livros1.data_de_lancamento;
```

```
mysql> explain select *, (select count(l2.data_de_lancamento) from livros as l2 where l2.data_de_lancamento = l.data_de_lancamento) as anteriores from livros as  
l order by l.data_de_lancamento;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	PRIMARY	l	NULL	ALL	NULL	NULL	NULL	NULL	20235	100.00	Using filesort
2	DEPENDENT SUBQUERY	l2	NULL	ALL	NULL	NULL	NULL	NULL	20235	10.00	Using where

2 rows in set, 2 warnings (0,01 sec)

# ÍNDICE NO MYSQL

- No banco de dados MySQL os índices podem ser criados com considerável facilidade, tanto no momento da concepção da tabela quanto em uma tabela já existente.

Criando o índice junto com a tabela

```
CREATE TABLE CLIENTES  
(  
  Codigo INT,  
  Nome VARCHAR(50),  
  INDEX (Codigo)  
);
```

Criando o índice separadamente

```
CREATE INDEX idx_CLIENTES_CODIGO ON CLIENTES(Codigo);
```

# DESVANTAGENS

AO CRIARMOS UM ÍNDICE PARA UMA COLUNA, A QUERY FICA ABSURDAMENTE MAIS RÁPIDA, PORÉM NEM TUDO SÃO FLORES QUANDO FALAMOS DA CRIAÇÃO DE ÍNDICES. ALGUMAS DESVANTAGENS SÃO:

- AUMENTO DE ESPAÇO EM DISCO, POIS O ÍNDICE É SALVO E ISSO AUMENTA O TAMANHO DO SEU BANCO;
- PERDA DE PERFORMANCE NA HORA DE INSERIR, ATUALIZAR E APAGAR O DADO;
- ALÉM DE CRIAR MAIS UM ARTEFATO PARA SER MANTIDO E CUIDADO NO SEU BANCO DE DADOS.

# REFERÊNCIAS

- Modelagem de Banco de Dados Relacional: Índices e Normalização. alura, 2021. Disponível em: <https://cursos.alura.com.br/course/modelagem-banco-relacional>. Acesso em: 08, 06 e 2021.