

Nome: Carlos Eduardo Rodrigues

Matrícula: 221031265

Atividade: *Benchmarking* na base de dados **marketplace**

CONSULTA 1

```
SELECT SQL_NO_CACHE COUNT(*) AS `Número de compras em 2015`  
FROM sale  
WHERE YEAR(date) = 2015  
GROUP BY YEAR(date);
```

Método de Análise (registre suas ações sobre a consulta original)

Modelagem: **sem necessidade de alteração.**

Instrução SELECT: não existe necessidade de Group By e pode ser usado o operador between para condição do ano inteiro, sem necessidade de retirada somente do ano e conversão para inteiro para comparação com 2015.

Indexação: criação do índice adicional apresentado a seguir sobre a chave de procura desta consulta (*date*) que não possuía nenhum índice implementado.

```
CREATE INDEX sale_date_idx ON sale(date);
```

NOVA CONSULTA 1 (otimizada)

```
SELECT SQL_NO_CACHE COUNT(*) AS `Compras em 2015 otimizada`  
FROM sale  
WHERE date BETWEEN `2015-01-01` AND `2015-12-31`;
```

EXECUÇÕES

- **Antes** (original)
 - Transações: **58**
 - Latência média: **3690.78**
- **Depois** (otimizada)
 - Transações: **72**
 - Latência média: **3178.13**
- **Resultado** (apuração de melhora percentual da Consulta 1)
 - Transações: $((72 / 58) - 1) * 100 = 24,13\%$ (melhoria aumentou)

- Latência média: $((3178,13 / 3690,78) - 1) * 100 = -13,89\%$ (melhoria diminuiu)

CONSULTA 2 (original)

```
SELECT SQL_NO_CACHE COUNT(*) AS `Quantidade`  
FROM product  
WHERE price BETWEEN 0 AND 1000  
AND stock <= 100;
```

Método de Análise (registre suas ações sobre a consulta original)

Modelagem: Sem necessidade de alteração.

Instrução SELECT: A ordem das condições na cláusula WHERE foi ajustada, colocando stock <= 100 antes de price BETWEEN 0 AND 1000

Indexação: Criado um índice composto sobre price e stock

CREATE INDEX product_price_stock_idx ON product(price, stock);

NOVA CONSULTA 2 (otimizada)

```
SELECT SQL_NO_CACHE COUNT(*) AS `Quantidade`  
FROM product  
WHERE stock <= 100  
AND price BETWEEN 0 AND 1000;
```

EXECUÇÕES

- **Antes** (original)
 - Transações: 338
 - Latência média: 481.17
- **Depois** (otimizada)
 - Transações: 607
 - Latência média: 264.88
- **Resultado** (apuração de melhora percentual da Consulta 2)

Aplicar fórmula: $((\text{valorOtimizado} / \text{valorOriginal}) - 1) * 100$

 - Transações: $((607 / 338) - 1) * 100 = 79,58\%$ (melhoria aumentou)
 - Latência média: $((264.88 / 481.17) - 1) * 100 = -44,96\%$ (melhoria diminuiu)

CONSULTA 3 (original)

```
SELECT SQL_NO_CACHE name AS `Nome da categoria`  
FROM category  
LIMIT 10;
```

Método de Análise (registre suas ações sobre a consulta original)

Modelagem: Sem necessidade de alteração.

Instrução SELECT: Sem necessidade de alteração.

Indexação: Criado um índice sobre o atributo name.

CREATE INDEX category_name_idx ON category(name);

NOVA CONSULTA 3 (otimizada)

```
SELECT SQL_NO_CACHE name AS `Nome da categoria`  
FROM category  
LIMIT 10;
```

EXECUÇÕES

- **Antes** (original)
 - Transações: 116884
 - Latência média: 1.37
- **Depois** (otimizada)
 - Transações: 117989
 - Latência média: 1.36
- **Resultado** (apuração de melhora percentual da Consulta 2)

Aplicar fórmula: $((\text{valorOtimizado} / \text{valorOriginal}) - 1) * 100$

 - Transações: $((117989 / 116884) - 1) * 100 = 0,95\%$ (melhoria aumentou)
 - Latência média: $((1.36 / 1.37) - 1) * 100 = -0,73\%$ (melhoria diminuiu)

CONSULTA 4 (original)

```
SELECT SQL_NO_CACHE *  
FROM sale s
```

INNER JOIN customer c ON s.customerId = c.customerId
WHERE date > CURDATE() - INTERVAL 1 YEAR;

Método de Análise (registre suas ações sobre a consulta original)

Modelagem: Sem necessidade de alteração.

Instrução SELECT: Sem necessidade de alteração.

Indexação: Criado um índice sobre o atributo date.

CREATE INDEX sale_date_idx ON sale(date);

NOVA CONSULTA 4 (otimizada)

SELECT SQL_NO_CACHE *
FROM sale s
INNER JOIN customer c ON s.customerId = c.customerId
WHERE date > CURDATE() - INTERVAL 1 YEAR;

EXECUÇÕES

- **Antes** (original)
 - Transações: 32
 - Latência média: 7157.83
- **Depois** (otimizada)
 - Transações: 69867
 - Latência média: 2.29
- **Resultado** (apuração de melhora percentual da Consulta 4)

Aplicar fórmula: $((\text{valorOtimizado} / \text{valorOriginal}) - 1) * 100$

 - Transações: $((69867 / 32) - 1) * 100 = 218234,38\%$ (melhoria aumentou)
 - Latência média: $((2.29 / 7157.83) - 1) * 100 = -99,97\%$ (melhoria diminuiu)

CONSULTA 5 (original)

SELECT SQL_NO_CACHE login, password
FROM user
WHERE userId = 11100;

Método de Análise (registre suas ações sobre a consulta original)

Modelagem: Sem necessidade de alteração.

Instrução SELECT: Sem necessidade de alteração.

Indexação: Sem necessidade de alteração.

EXECUÇÕES

- **Antes** (original)
 - Transações: 122069
 - Latência média: 1.31
- Sem necessidade adicional para otimização.
- Ao solicitar um explain, foi verificado que o MySQL está usando a PK para acessar o registro (key: PRIMARY). O tipo de acesso é const, o que significa que o MySQL encontrou o registro em tempo constante.

#	id	select_type	table	partition	type
1	1	SIMPLE	user	HULL	const
key	key_len		ref		
PRIMARY	4		const		

CONSULTA 6 (original)

SELECT SQL_NO_CACHE *
FROM store;

Método de Análise (registre suas ações sobre a consulta original)

Modelagem: Sem necessidade de alteração.

Instrução SELECT: Sem necessidade de alteração.

Indexação: Sem necessidade de alteração.

EXECUÇÕES

- **Antes** (original)
 - Transações: 4272
 - Latência média: 37.49
- Sem necessidade adicional para otimização.