

Database Optimization:

Desvendando o Poder da
Performance em MySQL e
PostgreSQL

SAMUEL MOLLING





Whoami



25 anos
Campo Bom, RS
Staff DBRE @ PicPay
Community Creator @
MongoDB



samuelmolling@gmail.com

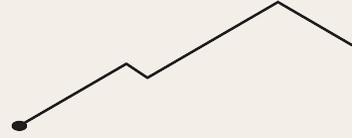


[linkedin.com/in/samuelmolling](https://www.linkedin.com/in/samuelmolling)

- Amante dos esportes
- Cloud, OpenSource e muito código

- 4x MongoDB Certified
- 2x AWS Certified
- 2x Datadog Certified
- 1x Terraform Certified
- 1x Cassandra Certified
- 1x Neo4j Certified
- 2x Scrum Certified

Agenda



- **Introdução**
- **Princípios Básicos da Performance em Bancos de Dados**
- **Monitoramento e Análise de Performance**
- **Indexação**
- **Otimização de Consultas SQL**
- **Encerramento**
- **Q&A**



PostgreSQL

VS



MySQL™

Princípios Básicos da Performance em Bancos de Dados Relacionais

- **Indexação**
- **Normalização**
- **Otimização de consultas**
- **Caching e Pooling**
- **Particionamento de Dados**
- **Dimensionamento Horizontal e Vertical**
- **Monitoramento e Análise de Performance**
- **Escolha do Tipo de Armazenamento**
- **Gerenciamento de Transações e Concorrência**
- **Arquitetura e Design do Sistema**

Princípios Básicos da Performance em Bancos de Dados Relacionais

- **Indexação** ←
- **Normalização**
- **Otimização de consultas** ←
- **Caching e Pooling**
- **Particionamento de Dados**
- **Dimensionamento Horizontal e Vertical**
- **Monitoramento e Análise de Performance** ←
- **Escolha do Tipo de Armazenamento**
- **Gerenciamento de Transações e Concorrência**
- **Arquitetura e Design do Sistema**

Monitoramento e Análise de Performance



PERCONA

Monitoring and Management

Monitoramento e Análise de Performance



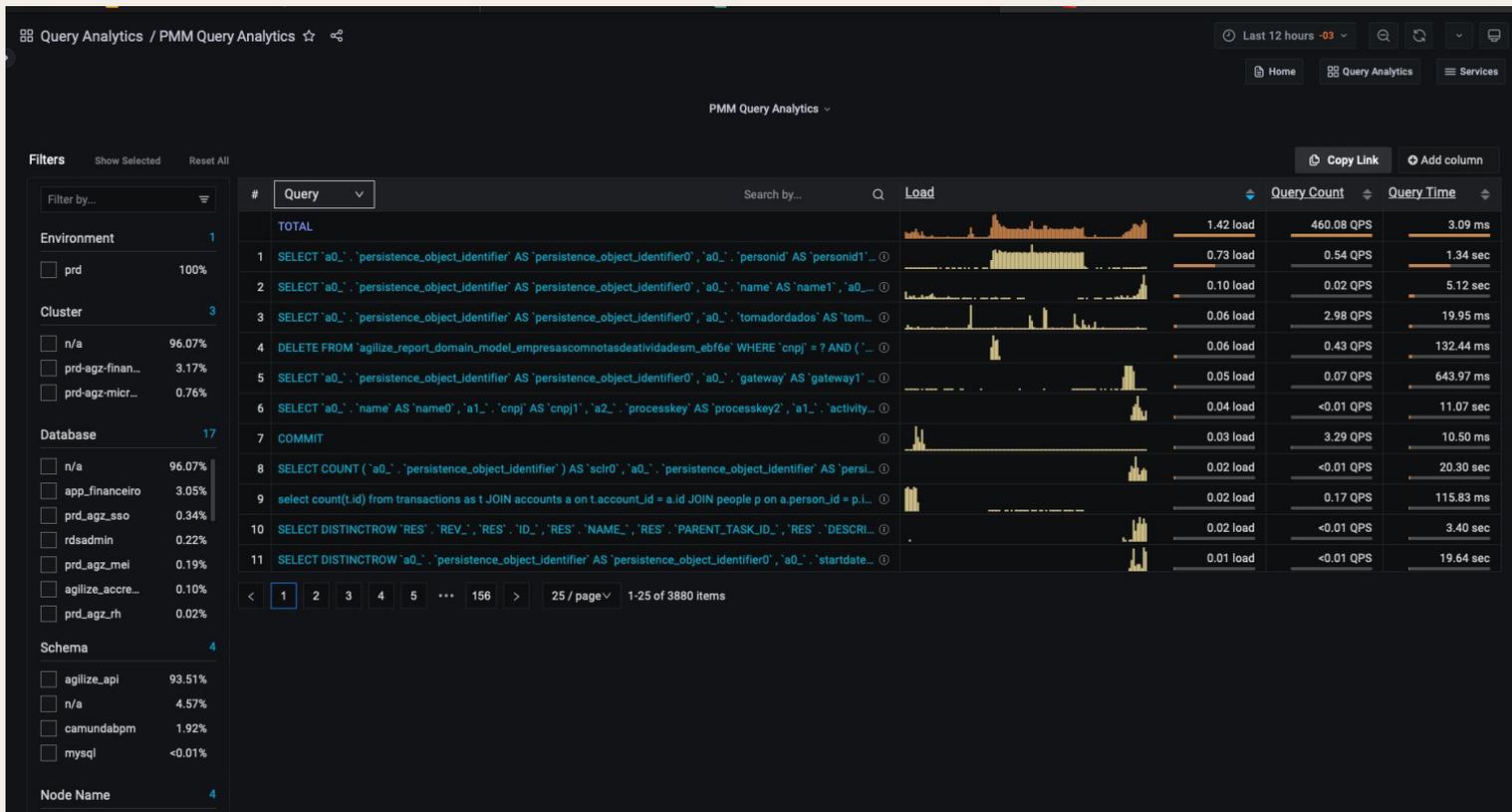
Monitoramento e Análise de Performance

The screenshot displays the 'Advisor Checks / All Checks' interface. On the left, a table lists various checks with columns for Name, Description, Status, and Interval. On the right, a detailed view of three checks is shown, each with a title, description, status, and category, along with 'Run' and 'Disable' buttons and an edit icon.

Filter	Name
Check if newer version of PostgreSQL	Check for newer version of PostgreSQL
Check for unsupported PostgreSQL	Check for unsupported PostgreSQL
Check if PostgreSQL version is EOL	Check if PostgreSQL version is EOL
PostgreSQL Archiver is failing	PostgreSQL Archiver is failing
PostgreSQL Autovacuum Logging Is Disabled	PostgreSQL Autovacuum Logging Is Disabled
PostgreSQL cache hit ratio	PostgreSQL cache hit ratio
PostgreSQL Checkpoints Logging is Disabled	PostgreSQL Checkpoints Logging is Disabled
PostgreSQL fsync is set to off	PostgreSQL fsync is set to off

Check if PostgreSQL version is EOL	PostgreSQL Archiver is failing	PostgreSQL Autovacuum Logging Is Disabled
Checks to see if the currently installed PostgreSQL version is end of life and no longer supported	This check verify if the archiver has failed.	This check returns a notice if the log_autovacuum_min_duration configuration option is set to -1 (disabled). It is recommended to enable the logging of autovacuum run information, as that provides a lot of useful information with almost no drawbacks.
Enabled	Enabled	Enabled
Standard	Standard	Standard
Run Disable	Run Disable	Run Disable

Monitoramento e Análise de Performance



Indexação

- Chave primária
- Chave Estrangeira
- Constraint Unique
 - Índice Clusterizado
 - Índice Não-Clusterizado
 - Índice B-Tree
 - Índice Bitmap
 - Índice Hash
 - Índice Funcional



Indexação

- **Compound Index**
- **Unique Index**
- **Partial Index**
- **Wildcard Index**
- **Text search Index**
- **Geospatial Index**



Otimização de consultas SQL



Ciclo de vida de uma consulta SQL



O famoso Explain



```
EXPLAIN SELECT * FROM tablename WHERE columnname = 'value';
```

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	tablename	ref	idx_column	idx_column	4	const	1	Using where

O famoso Explain



```
EXPLAIN ANALYZE SELECT * FROM tablename WHERE columnname = 'value';
```

```
Seq Scan on tablename (cost=0.00..19.00 rows=1 width=132) (actual time=0.015..0.020 rows=1 loops=1)
  Filter: (columnname = 'value')
  Rows Removed by Filter: 999
Planning Time: 0.050 ms
Execution Time: 0.045 ms
```

Uso Ineficiente de Funções em Colunas

```
SELECT
  *
FROM
  pedidos
WHERE
  YEAR(data_pedido) = 2022;
```



Uso Ineficiente de Funções em Colunas



```
SELECT
  *
FROM
  pedidos
WHERE
  data_pedido ≥ '2022-01-01'
  AND data_pedido < '2023-01-01';
```

Uso Inadequado de OFFSET para Paginação

```
SELECT
  *
FROM
  clientes
ORDER BY
  nome
LIMIT
  10 OFFSET 10000;
```



Uso Inadequado de OFFSET para Paginação



```
SELECT
|   *
FROM
|   clientes
WHERE
|   id > [último id na página anterior]
ORDER BY
|   nome
LIMIT
|   10;
```

Uso Excessivo de Subconsultas

```
SELECT
  nome
FROM
  clientes
WHERE
  id_cliente IN (
    SELECT
      id_cliente
    FROM
      pedidos
    WHERE
      valor_total > 1000
  );
```



Uso Excesivo de Subconsultas



```
SELECT
    DISTINCT c.nome
FROM
    clientes c
    JOIN pedidos p ON c.id_cliente = p.id_cliente
WHERE
    p.valor_total > 1000;
```

Seleção Excessiva de Colunas

```
SELECT
  *
FROM
  pedidos;
```



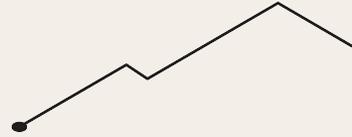
Seleção Excessiva de Colunas

```
SELECT  
    id_pedido,  
    data_pedido,  
    valor_total  
FROM  
    pedidos;
```



Não Considerar a Cardinalidade

```
CREATE INDEX idx_status ON clientes(status);
```



Não Considerar a Cardinalidade

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

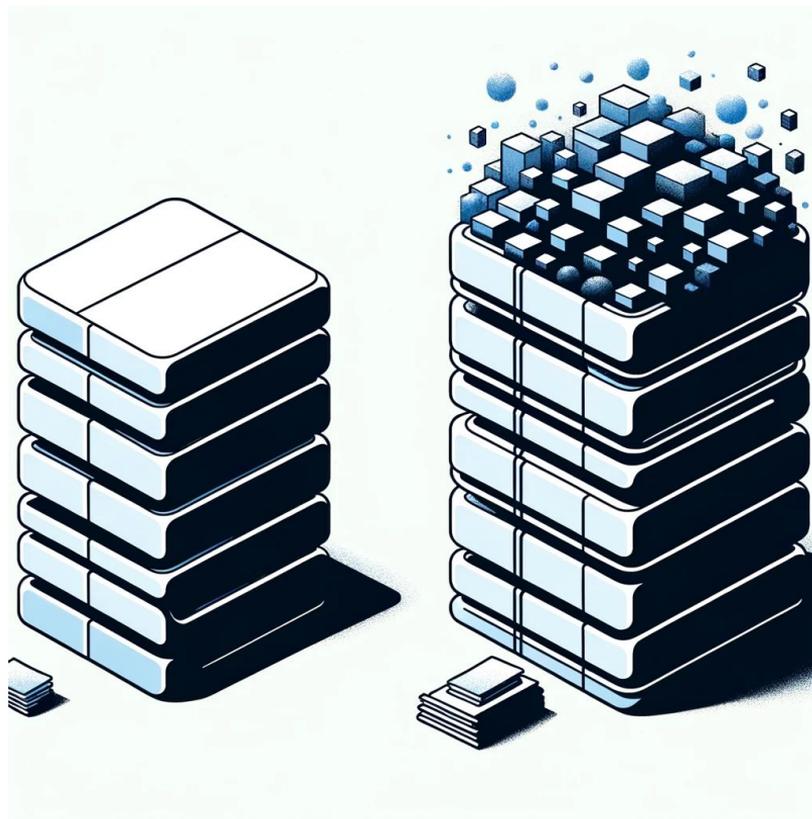


Tabelas Grandes

- **Arquivamento de Dados**
- **Indices Eficientes**
- **Particionamento de tabelas**



Inchaço de tabelas



Praticando...



```
SELECT
  COUNT(1) AS `total`
FROM
  (
    SELECT
      COUNT(1)
    FROM
      `agilize_payment_contract_domain_model_invoice` AS `invoice`
    JOIN `agilize_payment_contract_domain_model_invoiceentry` `invoiceentry` ON `invoice`.`persistence_object_identifier` = `invoiceentry`.`invoice`
    WHERE
      YEAR(`invoice`.`competence`) ≥ 2022
    GROUP BY
      `invoice`
    HAVING
      SUM(`invoiceentry`.`amount`) ≥ 10
  ) AS `total`
```

Praticando...



id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	PRIMARY	<derived2>	NULL	ALL	NULL	NULL	NULL	NULL	312597	100.00	NULL
2	DERIVED	invoice	NULL	ALL	PRIMARY	NULL	NULL	NULL	176943	100.00	Using where; Using temporary; Using filesort
2	DERIVED	invoiceentry	NULL	ref	IDX_BB9DF02190651744	IDX_BB9DF02190651744	123	agilize_api.invoice.persistence_object_identifier	1	100.00	NULL

Praticando...



```
SELECT
  COUNT(*) AS total
FROM
  (
    SELECT
      invoice.persistance_object_identifier
    FROM
      agilize_payment_contract_domain_model_invoice AS invoice
      JOIN agilize_payment_contract_domain_model_invoiceentry AS invoiceentry ON invoice.persistance_object_identifier = invoiceentry.invoice
    WHERE
      invoice.competence >= '2022-01-01'
    GROUP BY
      invoice.persistance_object_identifier
    HAVING
      SUM(invoiceentry.amount) >= 10
  ) AS subquery
```

Praticando...



id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	PRIMARY	<derived2>	NULL	ALL	NULL	NULL	NULL	NULL	104188	100.00	NULL
2	DERIVED	invoice	NULL	index	PRIMARY,flow_model_invoice	PRIMARY	122	NULL	176943	33.33	Using where
2	DERIVED	invoiceentry	NULL	ref	IDX_BB9DF0219	IDX_BB9DF02190651744	123	agilize_api.invoice.persistance_object_identifier	1	100.00	NULL

Praticando...

```
SELECT
  c.nome,
  COUNT(p.id_pedido) AS num_pedidos
FROM
  clientes c
  JOIN pedidos p ON c.id_cliente = p.id_cliente
WHERE
  p.data_pedido ≥ '2022-01-01'
  AND p.data_pedido < '2023-01-01'
GROUP BY
  c.nome
ORDER BY
  num_pedidos DESC
LIMIT
  10;
```



Praticando...



```
SELECT
    id_pedido,
    id_cliente,
    data_pedido,
    data_entrega_prevista,
    status
FROM
    pedidos
WHERE
    status = 'Em Atraso'
    AND data_entrega_prevista < CURRENT_DATE;
```

Praticando...



```
CREATE INDEX idx_pedidos_em_atraso ON pedidos (data_entrega_prevista)
WHERE
    status = 'Em Atraso';
```

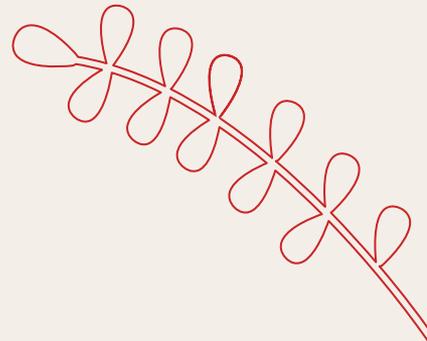
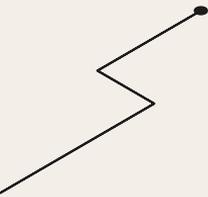
Encerramento

- **Normalização**
- **Otimização de consultas**
- **Caching e Pooling**
- **Particionamento de Dados**
- **Dimensionamento Horizontal e Vertical**
- **Monitoramento e Análise de Performance**
- **Escolha do Tipo de Armazenamento**
- **Gerenciamento de Transações e Concorrência**
- **Arquitetura e Design do Sistema**



Q&A





OBRIGADO

