

# Programação e Otimização em SQL (CET 0604)



A large orange circle on the left side of the slide.

Discussão  
Artigo

## **Performance Impact of Optimization Methods on MySQL Document-Based and Relational Databases**

Comentários?



# Instalação Mysql

Instalaram?

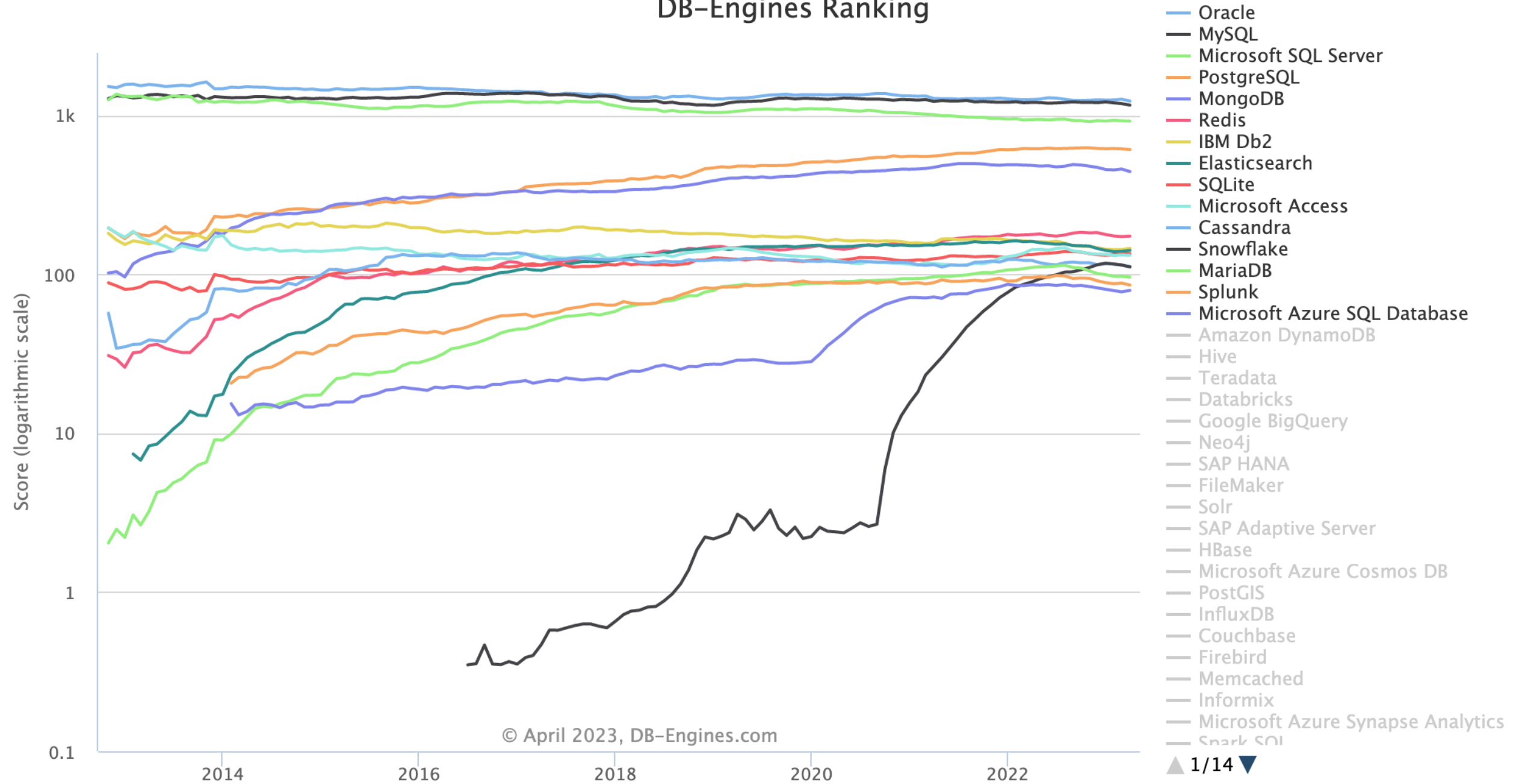
Mysql Server

<https://dev.mysql.com/doc/mysql-getting-started/en/#mysql-getting-started-installing>

Mysql Workbench:

<https://www.mysql.com/products/workbench/>

# DB-Engines Ranking



# Objetivos da Aula Prática

- Utilização Prática do Mysql
- Modelagem de dados
- Criação de Banco de Dados e Tabelas
- Geração de Massa de Dados
- Execução de Querys
- Análise de Querys
- Otimização de Querys
- Comparativo de Performance



# Utilização Prática do Mysql

- Login (mysql -u root -p)
- show processlist;
- show databases;
- use <DATABASE>;
- show tables



# Configuração de Logs

---

```
SET global general_log = 1;  
SET global log_output = 'table';
```

```
USE mysql;  
SELECT *, CONVERT(argument USING utf8 ) FROM  
mysql.general_log;
```

Caso as tabelas não estejam criadas:

<https://tableplus.com/blog/2018/10/how-to-show-queries-log-in-mysql.html>

\*log\_queries.sql

# Log

- Baixar o arquivo no formato .csv
- Utilizar o Excel para uma limpeza dos dados
- Separar as colunas por vírgula
- Ordenar para ver a repetição entre as consultas
- Otimizar as mais requisitadas



# Estrutura Banco de Dados

schema.sql

Criação do Banco de  
Dados (CREATE DATABASE)

Criação das Tabelas  
(CREATE TABLE)

Descrição das tabelas  
criadas (DESC)



# Exercício de Modelagem

## Parte 1



Façam a modelagem de um E-commerce, no mínimo: Categoria, Produto, Cliente e Pedido.



Modelo Entidade-Relacionamento e o Arquivo com instruções SQL de criação.



Tabelas, chaves, relacionamentos.



Bônus: +1 ponto no trabalho

# Tabela Categoria

---

```
CREATE TABLE Categoria (  
  cat_id INT UNSIGNED AUTO_INCREMENT NOT NULL,  
  cat_nome VARCHAR (255),  
  cat_descricao VARCHAR (255),  
  PRIMARY KEY(cat_id)  
) ENGINE=INNODB;
```

# Tabela Produto

---

```
CREATE TABLE Produto (  
  pro_id INT UNSIGNED AUTO_INCREMENT NOT NULL,  
  pro_nome VARCHAR (255),  
  pro_descricao VARCHAR (255),  
  pro_preco FLOAT,  
  pro_cat_id INT UNSIGNED NOT NULL,  
  PRIMARY KEY(pro_id),  
  FOREIGN KEY (pro_cat_id) REFERENCES Categoria(cat_id) ON DELETE RESTRICT  
) ENGINE=INNODB;
```

# Tabela Cliente

---

```
CREATE TABLE Cliente (  
  cli_id INT UNSIGNED AUTO_INCREMENT NOT NULL,  
  cli_nome VARCHAR (255),  
  cli_email VARCHAR (255),  
  PRIMARY KEY(cli_id)  
) ENGINE=INNODB;
```

# Tabela Pedido

---

```
CREATE TABLE Pedido (  
  ped_id INT UNSIGNED AUTO_INCREMENT NOT NULL,  
  ped_data DATETIME,  
  ped_endereço VARCHAR (255),  
  ped_pro_id INT UNSIGNED NOT NULL,  
  ped_cli_id INT UNSIGNED NOT NULL,  
  PRIMARY KEY (ped_id),  
  FOREIGN KEY (ped_pro_id) REFERENCES Produto(pro_id) ON DELETE RESTRICT,  
  FOREIGN KEY (ped_cli_id) REFERENCES Cliente(cli_id) ON DELETE RESTRICT  
) ENGINE=INNODB;
```



# Geração da Massa de Dados

---

- <https://generatedata.com/>
- <https://www.mockaroo.com/>
- Configuração dos tipos de dados
- Escolha do nome das tabelas
- Formato de exportação
- Download do dataset Gerado
- Dados regionalizados

# CREATE INDEX

```
CREATE TABLE t1 (  
  col1 VARCHAR(10),  
  col2 VARCHAR(20),  
  INDEX (col1, col2(10))  
);  
CREATE INDEX idx2 ON t1 (col1, col2(10));  
SHOW INDEX from t1
```



# Exercício de Otimização Parte 2

Massa de Dados:

Categoria  $\geq$  500 itens

Produto  $\geq$  500 itens

Cliente  $\geq$  500 itens

Pedido  $\geq$  5.000 itens

# Exercício de Otimização Parte 2



Consultas com e sem índice:



Categoria: Listagem de todas as categorias com pelo menos 3 produtos cadastrados (JOIN)



Clientes: Listagem dos clientes cujo nome contenham sequencia de letras "ca" (TEXT SEARCH)



Produto: Listagem de todos os produtos com valor entre 100 e 200 (FILTRAGEM)



Pedidos: Listagem dos 10 clientes que fizeram a maior quantidade de compras. (AGRUPAMENTO)



Bônus: +1 ponto na prova teórica.

# Dump e Restore

- Backup do Banco de Dados:
- `mysqldump -u [user] -p [database_name] > [filename].sql`
- Restore Banco de Dados
- `mysql -u [user] -p [database_name] < [filename].sql`
- Compactar o arquivo .sql com rar ou zip e o tamanho pode ser reduzido em 95%.



# Métricas

---

- Reduzir o tempo de respostas das queries mais utilizadas
- Aumentar a capacidade de executar mais queries por segundo
- Caso não seja o foco do projeto, não olhar para redução de recursos, eles estão lá para serem utilizados.



# Otimizações Modelagem

---

Usar char ao invés  
de varchar

Reduzir o tamanho  
dos campos para o  
máximo que ele irá  
armazenar

Evitar que os  
campos tenham  
dados null

Usar enum para  
representar dados  
que sejam um  
subconjunto

# EXPLAIN

	id	select_ty...	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
►	1	SIMPLE	Produto	NULL	index	pro_cat_id	pro_cat_id	4	NULL	1500	100.00	Using index; Using temporary; Using filesort

Table 8.1 EXPLAIN Output Columns

Column	JSON Name	Meaning
<u>id</u>	select_id	The SELECT identifier
<u>select_type</u>	None	The SELECT type
<u>table</u>	table_name	The table for the output row
<u>partitions</u>	partitions	The matching partitions
<u>type</u>	access_type	The join type
<u>possible_keys</u>	possible_keys	The possible indexes to choose
<u>key</u>	key	The index actually chosen
<u>key_len</u>	key_length	The length of the chosen key
<u>ref</u>	ref	The columns compared to the index
<u>rows</u>	rows	Estimate of rows to be examined
<u>filtered</u>	filtered	Percentage of rows filtered by table condition
<u>Extra</u>	None	Additional information



# Otimizações Querys

---

- Reduzir as linhas do resultado. (LIMIT)
- Reduzir as colunas do resultado (Não usar \*)
- Em alguns casos é possível trocar uma query complexa por algumas mais simples e agrupar os dados externamente (aplicação ou relatório)
- Inspecionar com o EXPLAIN

# Índices

---

Colocar índices para todos os campos que são usados nas filtrações (WHERE e ORDER BY)

Colocar índice em todas as Foreign Keys

Índices simples: somente uma coluna

Índices compostos: mais de uma coluna utilizada em uma mesma query

Primary key tem um índice automático.

Quando a tabela tem poucos registros o impacto é pequeno, mas quando ela cresce é MUITO grande.



# HEAP TABLE

```
CREATE TABLE relatorio ENGINE=MEMORY SELECT  
pro_id, pro_nome, pva_peso FROM tbl_produto JOIN  
tbl_produto_variante ON pva_pro_id = pro_id WHERE  
pro_id < 10;
```

```
SELECT * FROM relatorio;
```

```
DROP TABLE relatorio;
```

```
*heap-table-query-vitesse.sql
```



Artigo  
Leitura  
Extra  
(opcional)

**SmartIX: A database indexing agent based on reinforcement learning**





# Trabalhos Avaliativos

Análise crítica dos 3 artigos científicos:

- Resumo
- Como seria possível avançar mais no tema?
- Prazo: 19/04/2024 23:59

Avaliação:  $(TI * 0,2) + (PP * 0,4) + (AV * 0,4)$

- TI – Trabalho Individual
- PP – Projeto Prático
- AV – Avaliação Teórica
- \* Bônus: até +1 ponto no Trabalho e +1 ponto na Prova



# Projeto Prático em Grupo

- Divisão dos Grupos
- Temas dos Projetos
- Conhecimento deve ser compartilhado
- Tabela Fato > 24.000 registros (simular 2 anos de operação de uma empresa)
- 5 Tabelas Dimensões
- Dimensões Importantes: responsável, item utilizado, instante, local, tipo de fato.
- Prazo: 05/05/2024 até 23:59

# Entregáveis

- Descrição DETALHADA do projeto
- Modelagem (anexar modelo)
- Explicação e justificativa de cada tabela e campo
- Análises sobre as escolhas de campos e queries que refletem em otimizações (tipos de dados, escrita de consultas, índices).
- Um relatório gerencial online ou offline com 5 indicadores (KPI) via consultas SQL que poderiam ser utilizadas em um dashboard de acompanhamento estratégico da operação comercial. Demonstrar o funcionamento delas e o EXPLAIN para justificar a correta construção da consulta.

*Optimization, Backups,  
Replication, and more*

**3rd Edition**  
Covers Version 5.5



# High Performance MySQL

O'REILLY®

*Baron Schwartz,  
Peter Zaitsev &  
Vadim Tkachenko*

## Aprofundar mais

High Performance MySQL:  
Optimization, Backups, and  
Replication

Baron Schwartz