



escola  
britânica de  
artes criativas  
& tecnologia

# Desenvolvedor Full Stack Python

Melhorando a Performance de Leitura com Índices

# AGENDA

- **Introdução a Índices**
- **Exercício**

# Opcional

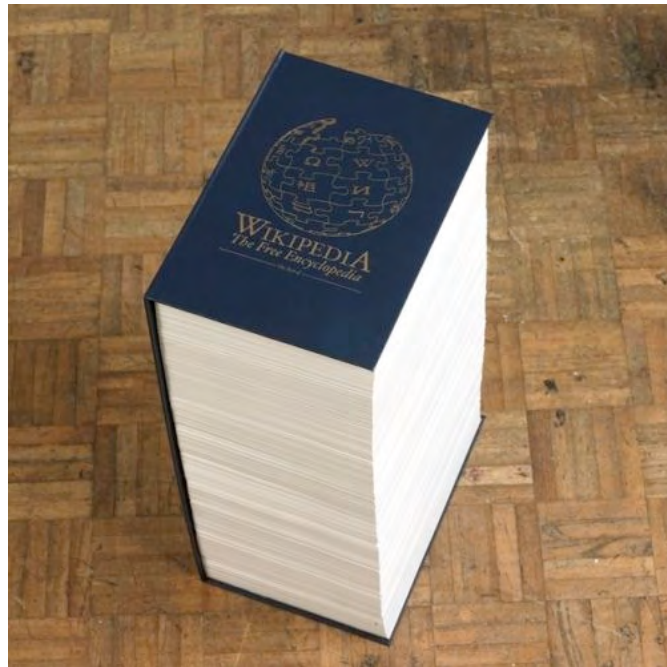
O aluno conseguirá ver índices normalmente, **mas terá uma vantagem tanto no curso quanto no mercado se souber o conteúdo extra de SQL.**

# Introdução a índices

Como procurar por algum assunto em um livro grande?

Ou para os mais velhos como procurar um telefone na lista telefônica?

**Índices nos bancos de dados** são utilizados para facilitar a busca de informações em uma tabela com o menor número possível de operações de leituras, tornado assim a busca mais rápida e eficiente.



# Otimizando a performance de um banco de dados

## Arquitetura

- Read Replica (AWS)
- Multi-AZ (Amazon RDS) - Failover Support
- Master Slave Replication

## Banco de Dados

- Indexes

# Como o SQL armazena os dados

Os registros são armazenados em páginas de dados, páginas estas que compõem o que chamamos de **pilha**, que por sua vez é uma coleção de páginas de dados que contém os registros de uma tabela.

H	DADOS	H	DADOS
H	DADOS	H	DADOS
H	DADOS	H	DADOS
H	DADOS	H	DADOS

Uma extensão é formada por oito páginas no máximo totalizando 64 Kb.

# Como o SQL armazena os dados

Os registros de dados não são armazenados em uma **ordem específica**.

As páginas de dados não estão vinculadas a uma lista, pois implementam diretamente o conceito de **pilhas**.

Quando são inseridos registros em uma página de dados e ela se encontra quase cheia, as páginas de dados são divididas em um link é estabelecido para marcações e ligações entre elas.

# Utilizando o comando **Explain**

O Postgres possui uma funcionalidade bem útil quando precisamos verificar o tempo e o custo de um determinado SELECT, esse comando se chama **EXPLAIN ANALYZE**.

Com o **EXPLAIN ANALYZE** teremos o retorno detalhado da execução de um SELECT:

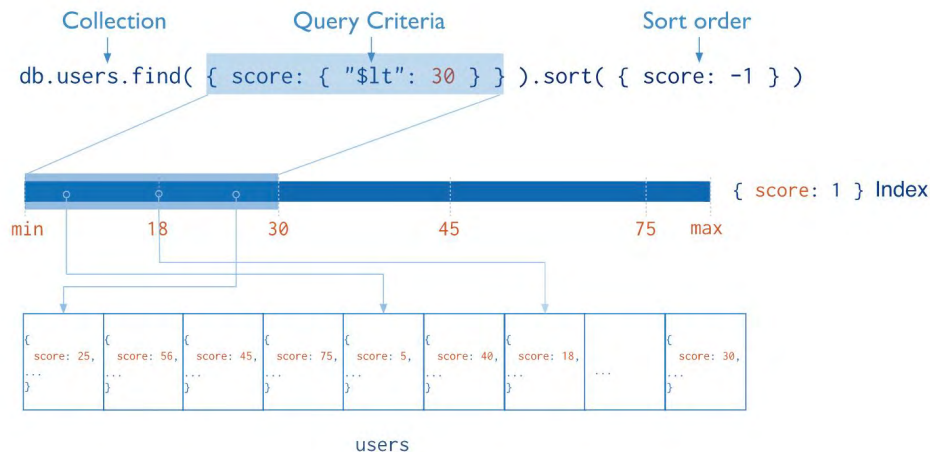
```
postgres=# EXPLAIN ANALYZE select * from customer;  
              QUERY PLAN  
-----  
Seq Scan on customer (cost=0.00..1.00 rows=1 width=238) (actual time=1.868..1.872 rows=4 loops=1)  
Planning Time: 3.453 ms  
Execution Time: 3.140 ms  
(3 rows)
```



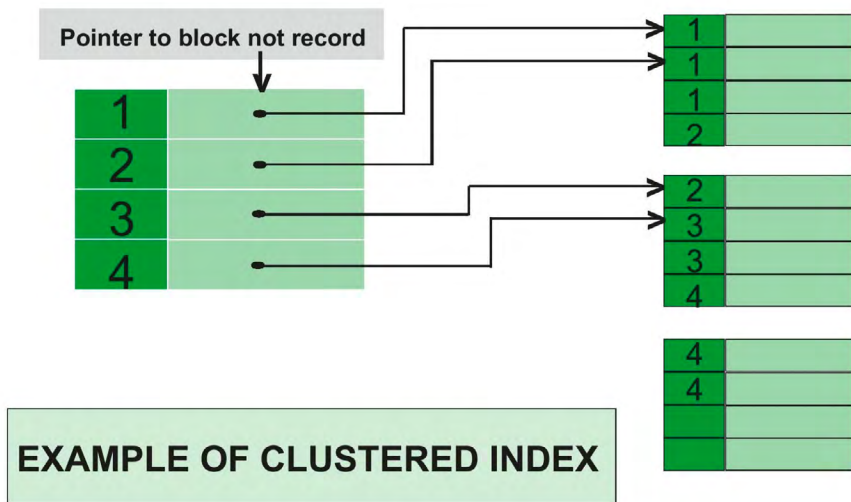
# Índices no SQL

Índices são sempre bem vindos em colunas de grande **seletividade**.

## Indices no NOSQL (Mongo, MariaDB...)



# Índice Clusterizado



# Índice Clusterizado

Um índice clusterizado determina a ordem em que as linhas de uma tabela são armazenadas no disco.

Suponha que temos uma tabela chamada “Livro” que tem uma coluna de chave primária “livroID” e que criamos um índice clusterizado para essa mesma coluna. Ao fazer isso, todas as linhas dentro da tabela Livro serão fisicamente ordenadas (Através de um **Grupo**).

Numa tabela, quando é definida uma **chave primária (PK)**, um índice **clusterizado** é criado automaticamente pelo banco para a mesma. E o que ele faz basicamente é ordenar as informações pela coluna de chave daquele índice em disco

**Postgres não possui suporte para índices clusterizados.**

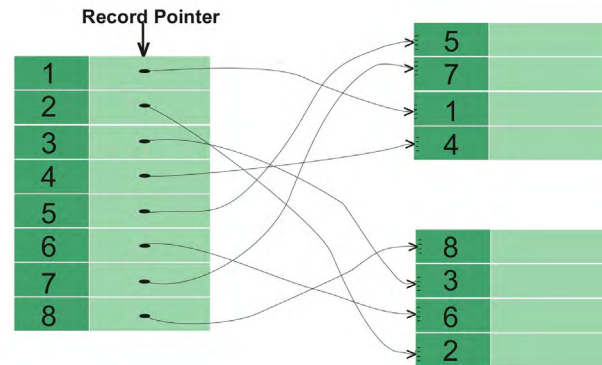
# Índice Não Clusterizado

O índice não clusterizado é semelhante ao índice de um livro.

O índice de um livro consiste em um nome de capítulo e número de página, se você quiser ler qualquer tópico ou capítulo, você pode ir diretamente para essa página usando o **índice desse livro**.

Não há necessidade de passar por cada página de um livro.

**CLUSTER** employees **USING** employees\_ind;



EXAMPLE OF NON-CLUSTERED INDEX

# B-tree e Sorting

As entradas do índice B-Tree são classificadas em ordem ascendente por padrão. Em alguns casos, faz sentido fornecer uma ordem de tipo diferente para um índice.

## Para Criar um Índice B-Tree em uma Coluna Única

```
CREATE INDEX customer_name_index ON customer(name DESC NULLS LAST);
```

## Para Criar um Índice B-Tree em diversas colunas

```
CREATE INDEX customer_name_email_index ON customer(name, email);
```

# Vantagens e Desvantagens de Índices

## Vantagens

- Melhoria de Performance
- Pode Trazer dados específicos mais rápidos

## Desvantagens

- Piora a Performance de Escritas em BD
- Aumenta o Consumo de Espaço

# Resumo

Aprendemos:

- Índices
- Arquiteturas para Otimização e Melhoria de Performance de BD
- Indices Clusterizados e Não Clusterizados
- Explain Analyze
- Criar Indices
- Pontos Negativos e Positivos

Script inicial:

<https://gist.github.com/drsantos20/3a75e0385c015db1c1f5144178778ca0>

Agora você já sabe como índices funcionam e como otimizar buscas em bancos com grande volume de dados, neste exercício vamos executar um SELECT utilizando o analyze na tabela de clientes salvar o resultado, vamos criar um índice em um campo na tabela clientes e executar um novo SELECT e comparar o resultado.

Você pode compartilhar os arquivos de criação de índice na plataforma da EBAC.

Entregar no formato .sql