



# Banco de dados com PostgreSQL

## Aula 8



Antes de mais nada, precisamos entender o que são os índices para a base de dados, onde em geral, ele é uma estrutura de dados utilizada para melhorar o tempo de execução das consultas, ou seja, os índices são estruturas que organizam referências a localização dos dados reais das tabelas.

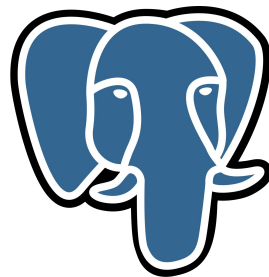
Quando estamos lidando com SGBD's (Sistemas Gerenciadores de Bancos de Dados), como é o caso do PostgreSQL, temos que o índice é uma "cópia" do item que desejamos combinar com uma referência à localização real dos dados. Quando realizamos buscas nas tabelas sem a utilização de índices, dependendo da quantidade de registros, podemos perceber que a busca é um pouco lenta, pois dessa forma, a pesquisa é realizada de forma sequencial. Quando dizemos que existe uma pesquisa sequencial, estamos nos referindo a uma busca linha a linha em toda a tabela (ou conjunto de tabelas) da base de dados com o intuito de obter a informação necessária. Para entendermos melhor os índices, vejamos primeiramente como os dados armazenados nas tabelas são organizados, onde existem duas formas, que são as tabelas heaps e as tabelas organizadas por índices.



Criando um novo índice

O código abaixo cria um novo índice na tabela Produto

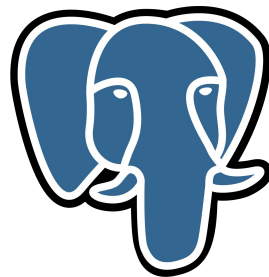
```
CREATE INDEX test1_id_index ON produto (id);
```





Existem várias ações para se otimizar uma consulta, independente do banco de dados utilizado. No caso do PostgreSQL existe um comando interessante, e simples de ser usado, que fornece dados para uma análise de custo de tempo que uma consulta irá necessitar para ser realizada. Este mesmo comando pode ser utilizado para comparar diferentes formas de escrever a mesma consulta, visando encontrar o modo que utilize menos recursos possíveis do servidor.

O PostgreSQL disponibiliza um comando que possibilita analisar previamente consultas SQL (nas operações de SELECT, INSERT, UPDATE, DELETE, EXECUTE, e DECLARE) tendo em vista saber como será o comportamento das mesmas, e permitindo também comparar diferentes expressões SQL para descobrir qual a melhor para se atingir um resultado mais rápido.

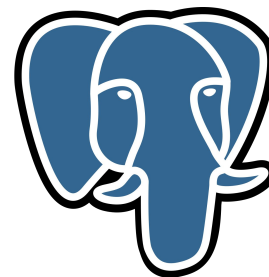




### EXPLAIN

Gera um relatório da execução da query.

```
EXPLAIN SELECT * FROM APLICATIVOS WHERE NOME = 'Softblue'
```

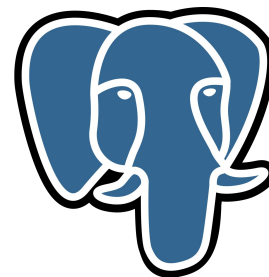




### EXPLAIN ANALYSE

Gera um relatório da execução da query, exibindo o custo em milisegundos

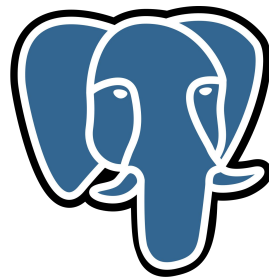
```
EXPLAIN ANALYSE SELECT * FROM APLICATIVOS WHERE NOME =  
'Softblue'
```





As organizações têm backups ou se preparam para backups para evitar a perda de dados e manter a continuidade. No entanto, raramente você investiga como uma perda de dados pode ocorrer em seu ambiente, o que é exclusivo para cada caso de uso. Há uma variedade de razões para a perda de dados que exige a necessidade de fazer backup de seus bancos de dados. Alguns dos motivos podem ser: corrupção de dados, falha de dispositivo, erros de software ou humanos e muito mais.

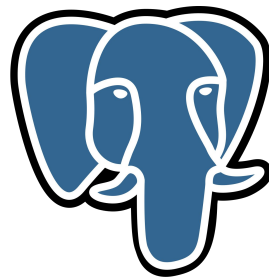
A seguir veremos os tipos diferentes de backups disponíveis no PostgreSQL





### 1º Dump SQL

A função `pg_dump` do PostgreSQL exporta um banco de dados e despeja seu conteúdo em um arquivo. Internamente, esses dumps são consistentes, o que significa que refletem o estado do banco de dados no momento em que o `pg_dump` foi iniciado.

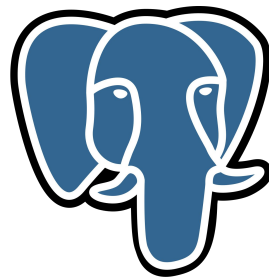






### 2º Backup no nível de sistemas de arquivos

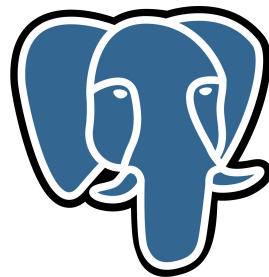
O backup no nível do sistema de arquivos é um dos métodos de backup mais rápidos. Isso envolve transferir os arquivos para um local de armazenamento e copiá-los de volta quando necessário. No entanto, essa abordagem funciona apenas para backups completos, não para backups de bancos de dados específicos.





### 3º Arquivamento contínuo/recuperação pontual

O PostgreSQL mantém um log chamado WAL (Write Ahead Log) no diretório `pg_xlog/`. Esse log é atualizado sempre que um update é efetuado e serve como uma cópia de emergência caso ocorra um crash no banco de dados.





ACID é um conceito que se refere às quatro propriedades de transação de um sistema de banco de dados: Atomicidade, Consistência, Isolamento e Durabilidade.

- Atomicidade: Em uma transação envolvendo duas ou mais partes de informações discretas, ou a transação será executada totalmente ou não será executada, garantindo assim que as transações sejam atômicas.
- Consistência: A transação cria um novo estado válido dos dados ou em caso de falha retorna todos os dados ao seu estado antes que a transação seja iniciada.
- Isolamento: Uma transação em andamento mas ainda não validada deve permanecer isolada de qualquer outra operação, ou seja, garantimos que a transação não será interferida por nenhuma outra transação concorrente.



- Durabilidade: Dados validados são registados pelo sistema de tal forma que mesmo no caso de uma falha e/ou reinício do sistema, os dados estão disponíveis em seu estado correto.

