

INSTITUTO FEDERAL
Paraíba
Campus João Pessoa

Aula

7

Banco de Dados II

Atualizada em 07/10/2019

Indexação



PostgreSQL

Professor:

Dr. Alex Sandro da Cunha Rêgo



alex@ifpb.edu.br



- **Definição de Índice**
 - Vantagens e desvantagens
 - Estrutura de um índice
- Tipos de índices no PostgreSQL
- A cláusula **Explain**
- Sintaxe:
 - CREATE INDEX
 - DROP INDEX
 - REINDEX
- Índice multicoluna
- Lista índices de uma tabela



- **Definição**

- ☐ Uma **estrutura de dados** que visa otimizar o desempenho de um banco de dados quando uma *query* é processada, minimizando o número de acessos a disco

- **Características**

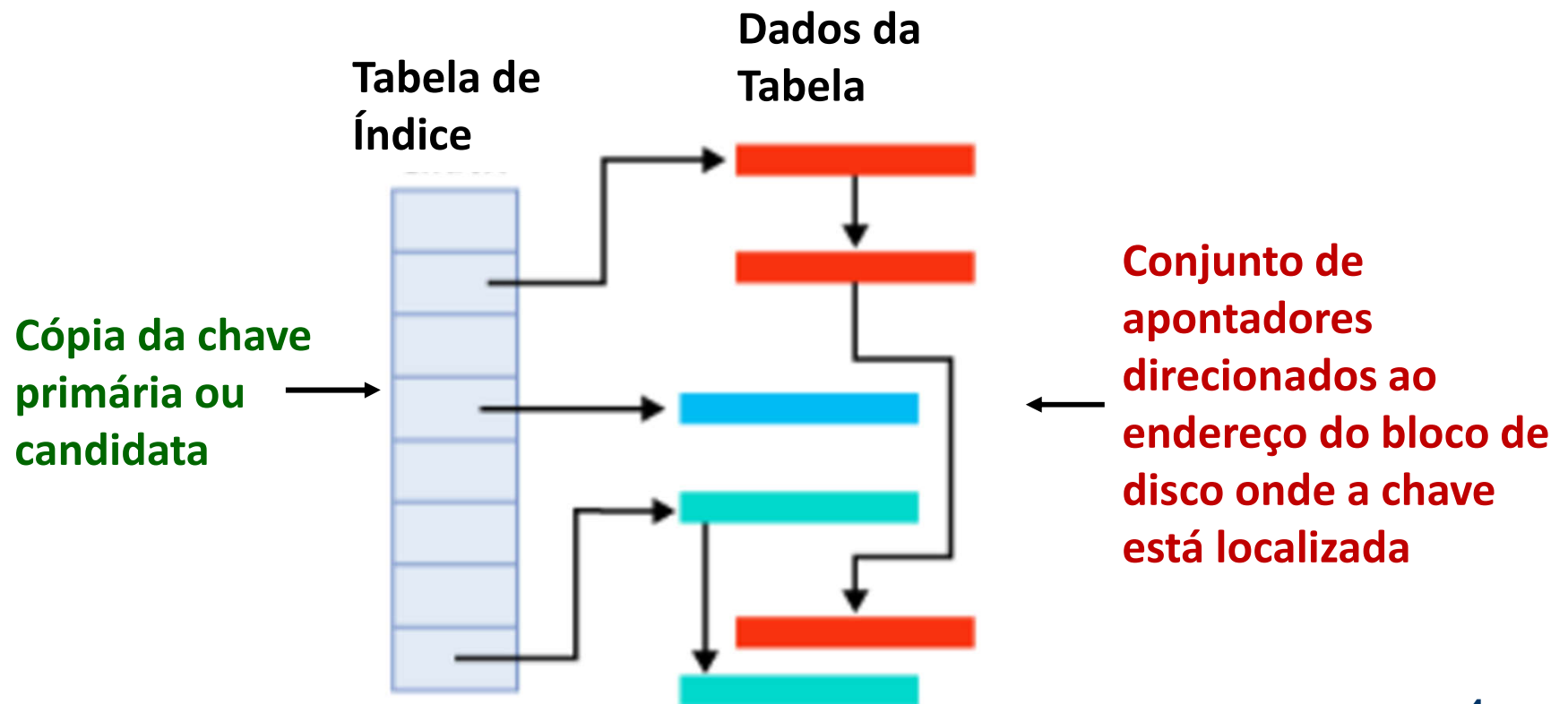
- ☐ Permite **localizar e acessar rapidamente** um item de dado específico em uma tabela
- ☐ Quando criados, fazem referência a uma ou mais colunas de uma tabela
- ☐ O valor da **chave do índice** é armazenada em uma ordem que facilita a localização do dado

Índices



- Estrutura de um índice

Chave do Índice	Referência do Dado
-----------------	--------------------





- **Tipos de índice no PostgreSQL**

- **B-tree (default)**

- ✓ Usado pelo **query planner** sempre que uma coluna indexada estiver envolvida com os operadores **<**, **<=**, **>**, **>=** e **=**

- **Hash**

- ✓ Lida apenas com comparações simples de igualdade

- **GIN**

- ✓ Valores com mais de uma chave (JSON e arrays)

- **GiST**

- ✓ busca textual e tipos de dados geoespaciais

- **SP-GiST e BRIN**

- ✓ datasets volumosos



- **Vantagens**

- ❑ Sem índice, o custo para encontrar o número de telefone de uma pessoa seria maior (sem ordenação)
 - ✓ **Varredura sequencial**
- ❑ Facilita a identificação de duplicatas

- **Desvantagens**

- ❑ Ocupam mais espaço em disco
- ❑ Desaceleram operações de INSERT, UPDATE e DELETE
 - ✓ Cada operação implica na **atualização dos índices**
 - ✓ O UPDATE é **acelerado** se a condição do WHERE indicar um campo indexado

Índices



• Sintaxe

```
CREATE INDEX nome_índice ON nome_tabela  
[USING method] (  
    nome_da_coluna [ASC|DESC]  
    [NULL{FIRST|LAST}],  
    ...  
)
```

Diagram illustrating the syntax for the `NULL{FIRST|LAST}` clause. An arrow points from the word `default` to the `FIRST` and `LAST` options, indicating they are the default values.

- ❑ Index method: **btree**, hash, gist, spgist, gin e brin
- ❑ **ASC** e **DESC**: critério de ordenação
- ❑ **NULL FIRST** e **NULL LAST**: campos NULL aparecem antes ou depois de campos **not-null**
- ❑ Colunas **primary key** têm índices criados automaticamente

Índices



- A cláusula **EXPLAIN**
 - Use a cláusula EXPLAIN (no SELECT) para verificar se a *query* utiliza ou não um índice

```
EXPLAIN SELECT * FROM address
WHERE phone = '223664661973';
```

QUERY PLAN

```
-----
Seq Scan on address (cost=0.00..15.54 rows=1 width=61)
  Filter: ((phone)::text = '223664661973'::text)
(2 registros)
```

- Observe que foi realizada uma **varredura sequencial**

Índices



- A cláusula **EXPLAIN**
 - Criando um índice para a coluna **phone**

```
CREATE INDEX idx_address_phone  
ON address(phone);
```

```
EXPLAIN SELECT * FROM address  
WHERE phone = '223664661973';
```

QUERY PLAN

```
-----  
Index Scan using idx_address_phone on address  
(cost=0.28..8.29 rows=1 width=61)  
Index Cond: ((phone)::text = '223664661973'::text)  
(2 registros)
```



- **Índices Multicoluna**

- Índice definido para duas ou mais colunas

```
CREATE INDEX nome_índice ON nome_tabela  
(coluna1, coluna2,...);
```

- Deve ser criado em condições de queries que envolvam qualquer subconjunto das colunas do índice

```
CREATE INDEX idx_actor_first_last  
ON actor (first_name,last_name);  
SELECT actor_id FROM actor  
WHERE first_name = 'valor1' AND  
      last_name  = 'valor2';
```



- **Índices Multicoluna**

- ❑ **NOTA:** ao definir um índice **multicoluna**, coloque as colunas mais utilizadas na cláusula WHERE no **início da lista de colunas**
- ❑ O otimizador de consulta considerará o índice nos seguintes casos (colunas a,b,c)

```
...WHERE a=v1 AND b=v2 AND c=v3;  
...WHERE a=v1 AND b=v2;  
...WHERE a=v1;
```

- ❑ **Não** utilizará o índice:

```
...WHERE b=v2 AND c=v2;  
...WHERE c=v3;
```

Apagando um Índice



- **DROP INDEX**

- ❑ Remove um índice existente em um banco de dados

DROP INDEX

```
[IF EXISTS] nome_índice  
[CASCADE | RESTRICT];
```

- ❑ **IF EXISTS**: para evitar remover um índice que não existe gera um erro
- ❑ **CASCADE**: elimina automaticamente objetos que dependem do índice
- ❑ **RESTRICT (default)**: recusa a eliminação do índice se houver objetos dependentes dele

Apagando um índice



- **DROP INDEX**

```
CREATE INDEX idx_actor_firstName  
ON actor(first_name);
```

```
EXPLAIN SELECT * FROM actor  
WHERE first_name = 'John';
```

QUERY PLAN

```
-----  
Seq Scan on actor (cost=0.00..4.50 rows=1 width=25)  
  Filter: ((first_name)::text = 'John'::text)  
(2 registros)
```



Às vezes, o otimizador de consulta não utiliza o índice

Apagando um índice



- **DROP INDEX**

- ❑ **JUSTIFICATIVA:** o otimizador de consulta entendeu que é mais prudente realizar uma varredura na tabela inteira para localizar o registro
- ❑ Se um índice não é útil, você pode **removê-lo**

```
DROP INDEX idx_actor_firstName;
```

Listando um índice



- **Alternativa 1:** visão **pg_indexes**

- **Colunas:**

- ✓ **schemaname:** fornece o nome do esquema que contém tabelas e índices
 - ✓ **tablename:** armazena o nome da tabela a qual o índice pertence
 - ✓ **indexname:** fornece o nome do índice
 - ✓ **tablespace:** fornece o table_space (*local no Sistema de arquivos em que os objetos de BD são armazenados*) em que se encontra o índice
 - ✓ **indexdef:** armazena o comando de definição de índice utilizado (CREATE INDEX)

Listando um índice



- **Alternativa 1:** visão `pg_indexes`

- ❑ Query: selecionar todos os índices (e definição de criação) para a tabela **actor**

```
SELECT indexname FROM pg_indexes  
WHERE tablename = 'actor';
```

```
indexname  
-----  
actor_pkey  
idx_actor_last_name  
idx_actor_firstname  
(3 registros)
```


Listando um índice



- **Alternativa 2:** comando **psql**
 - ❑ **Premissa:** conectar a um banco de dados

```
\d nome_tabela
```

- ❑ O comando retorna todas as informações da referida tabela, incluindo estrutura da tabela, **índices**, triggers e constraints

```
Table: public.actor
Coluna      |          Tipo          | Collation | Nullable |          Default          |
-----|-----|-----|-----|-----|
actor_id    | integer                |           | not null | nextval('actor_actor_id_seq'::regclass) |
first_name  | character varying(45)  |           | not null |                                     |
last_name   | character varying(45)  |           | not null |                                     |
last_update | timestamp without time zone |           | not null | now()                               |
=ndices:
  "actor_pkey" PRIMARY KEY, btree (actor_id)
  "idx_actor_firstname" btree (first_name)
  "idx_actor_last_name" btree (last_name)
Referenciada por:
  TABLE "film_actor" CONSTRAINT "film_actor_actor_id_fkey" FOREIGN KEY (actor_id) REFERENCES actor(actor_id)
Gatilhos:
  last_updated BEFORE UPDATE ON actor FOR EACH ROW EXECUTE PROCEDURE last_updated()
```

Corrupção de Índice



- **REINDEX**

- ❑ Um índice, com o passar do tempo, pode ter sua estrutura corrompida e não conter mais **dados válidos** (referência inconsistente)

- ✓ **Falhas de hardware** ou **bugs de software**

- ❑ Recriando um único **índice**

```
REINDEX INDEX nome_do_índice;
```

- ❑ Recriando todos os índices de uma **tabela**

```
REINDEX TABLE nome_da_tabela;
```

- ❑ Recriando todos os índices de um **database**

```
REINDEX DATABASE nome_do_database;
```



- **Veja também:**
 - ❑ UNIQUE Index
 - ❑ PARTIAL Index
 - ❑ MULTICOLUMN Index

Referências Bibliográficas



- PostgreSQL Create Index: Disponível em: <http://www.postgresqltutorial.com/postgresql-indexes/postgresql-create-index/>
- Index (PostgreSQL Documentation): Disponível em: <https://www.postgresql.org/docs/9.1/indexes.html>
- PostgreSQL-Indexes (TutorialsPoint): Disponível em: https://www.tutorialspoint.com/postgresql/postgresql_indexes.htm
- Indexing in databases (GeeksforGeeks): Disponível em: <https://www.geeksforgeeks.org/indexing-in-databases-set-1/>

Referências Bibliográficas



- Indexing and B-Trees—Applications in Postgres: Disponível em:
<https://medium.com/@rrfd/indexing-and-b-trees-applications-in-postgres-fdaf5cee5dba>
- Postgre EXPLAIN:
<http://www.postgresqltutorial.com/postgresql-explain/>