

# data<sub>docs</sub>

Lucas Cunegundes

April 16, 2025

## Contents

<b>1</b>	<b>Retenção de Dados</b>	<b>1</b>
1.1	GDPR . . . . .	2
<b>2</b>	<b>Data Archiving</b>	<b>2</b>
2.1	Data Archiving X Backup . . . . .	2
2.2	Técnicas Comuns . . . . .	2
2.2.1	Tabela de Histórico . . . . .	2
2.2.2	Particionamento por Tempo . . . . .	2
2.2.3	Cold Storage / Data Lakes . . . . .	3
2.2.4	Compreensão e indexação espacial . . . . .	3
<b>3</b>	<b>Liberação de Espaço em Disco</b>	<b>3</b>
3.1	PostgreSQL . . . . .	3
3.1.1	Vacuuming . . . . .	3

## 1 Retenção de Dados

Melhora o desempenho, escalabilidade, reduz custo e minimiza riscos de segurança Armazenar os dados por um tempo específico

- Identificação dos Dados
- Métodos de Armazenamento
- Formato dos Dados
- Períodos de Retenção
- Procedimentos de Descarte

- Arquivamento e Backup
- Funções e Responsabilidades

## 1.1 GDPR

Dados devem ser mantidos de uma forma que permita a identificação dos titulares desses dados somente pelo período necessário para a finalidade dos dados

## 2 Data Archiving

Mover dados para áreas seguras e fora da infraestrutura principal É diferente de um Backup, pois archiving cumpre a finalidade de reter dados e backup é uma cópia de segurança para proteger e recuperar de desastres

### 2.1 Data Archiving X Backup

Data Archiving cumpre a finalidade de reter dados, os arquivos armazenados devem ser tratados como parte de um repositório de dados

Backups, por outro lado, fazem parte de um mecanismo voltado para recuperação de dados do sistema primário de armazenamento

### 2.2 Técnicas Comuns

#### 2.2.1 Tabela de Histórico

Mover dados antigos da tabela principal para uma tabela \*\_history ou \*\_archive

```
INSERT INTO tabela_historico
SELECT * FROM tabela1 WHERE tabela1.data < CURRENT_DATE - INTERVAL '2 years';
DELETE FROM tabela1 WHERE tabela1.data < CURRENT_DATE - INTERVAL '2 years';
```

#### 2.2.2 Particionamento por Tempo

Permite dividir dados por mês / ano e truncar partições antigas rapidamente

```
CREATE TABLE tabela (
    id SERIAL,
    data DATE,
    valor NUMERIC
) PARTITION BY RANGE (tabela.data);
```

### 2.2.3 Cold Storage / Data Lakes

Exportar dados inativos em CSV / Parquet para buckets em nuvem (S3, Azure Blob). Ferramentas como Apache NiFi, Airflow ou Glue são comuns aqui

### 2.2.4 Comprensão e indexação espacial

Dados arquivados podem usar compreensão agressiva (ex: ZSTD, LZMA), menos índices e armazenamento mais barato

## 3 Liberação de Espaço em Disco

Jobs agendados com cron, como por exemplo:

- Jobs noturnos para arquivar dados
- Jobs semanais para apagar sessões antigas ou logs
- Jobs mensais para compactar dados arquivados

Políticas de ciclo de vida em storage (cloud)

- AWS S3: regra de expiração de objeto após x dias
- Azure: Arquivamento automático de dados inativos

### 3.1 PostgreSQL

Manutenções periódicas conhecidas como **vacuuming**. Na maioria dos casos, é suficiente deixar que o **autovacuum** daemon tome conta disso. Alguns administradores de banco de dados poderão querer suplementar ou substituir as atividades desse daemon com comandos manuais **VACUUM** o que tipicamente são executados de acordo com um agendamento pelo **chron** ou outros scripts

**VACUUM FULL** no máximo aproximadamente 30 minutos na janela de 14h40 às 15h10, considerar também janelas de sincronização dos bancos de dados dos raspberries com o banco de dados da aplicação

A utilização recomendada do **VACUUM** para quem não preferir desativar o daemon **autovacuum**, é de uma vez por dia durante os períodos de baixo uso do banco, complementados com limpezas mais frequentes nas tabelas mais utilizadas conforme necessário

#### 3.1.1 Vacuuming