

Armazenamento SGBD Explicado

Como os SGBDs como PostgreSQL ou MySQL fazem para armazenar os dados?

Sistemas Gerenciadores de Banco de Dados (SGBDs) como **PostgreSQL** e **MySQL** armazenam dados utilizando uma combinação de estruturas de arquivos, buffers em memória, e técnicas de organização interna que garantem desempenho, consistência e segurança dos dados. Vamos detalhar os principais aspectos:

1. Armazenamento em Arquivos

Os dados de um banco são fisicamente armazenados em **arquivos binários no disco**:

PostgreSQL:

- Cada **tabela** e **índice** é armazenado em um ou mais arquivos no sistema de arquivos, geralmente dentro do diretório `base/` do cluster.
- Cada **coluna TOASTed** (dados grandes como textos longos ou blobs) é armazenada separadamente em tabelas auxiliares.
- O sistema também armazena arquivos para o **Write-Ahead Log (WAL)**, essenciais para recuperação em caso de falha.

MySQL (com InnoDB):

- Usa arquivos como:
 - `*.ibd` para armazenar dados e índices por tabela (modo *file-per-table*).
 - `ibdata1`, se configurado com armazenamento compartilhado.
 - Também possui arquivos de **log binário**, **undo logs**, e **redos** para garantir consistência e recuperação.
-

2. Estrutura Interna dos Arquivos

Os dados são organizados em **páginas de dados** (tipicamente de 8KB), que contêm **tuplas (linhas)** da tabela. Essas páginas formam:

- **Tabelas (heap ou clustered)**: conjunto de páginas com dados.
 - **Índices (geralmente B-Trees)**: estrutura para busca eficiente.
-

3. Buffer Pool (Memória)

Antes de ler/escrever no disco, os SGBDs mantêm um cache de páginas de dados em memória:

- **Buffer Pool (InnoDB)** ou **Shared Buffers (PostgreSQL)** armazenam páginas acessadas recentemente.
 - Isso reduz o número de acessos ao disco, que são lentos.
 - As modificações são feitas em memória e posteriormente escritas em disco (**checkpoint** ou **flush**).
-

4. Controle de Concorrência e Recuperação

- Os SGBDs mantêm **logs de transações** (WAL no PostgreSQL, redo/undo logs no InnoDB) para garantir:
 - **Durabilidade (D do ACID)**: após um commit, os dados podem ser recuperados.
 - **Recuperação de falhas**: refazer ou desfazer operações incompletas.
-

5. Sistema de Catálogo

- Metadados como nomes de tabelas, esquemas, colunas, índices, permissões, etc. são armazenados em **tabelas internas do sistema**.
-

Exemplo de Fluxo de Armazenamento:

1. Usuário executa `INSERT INTO clientes VALUES (...)`.
2. A linha é armazenada em uma página de memória no **buffer pool**.
3. O **WAL**/log de redo é atualizado com a operação.
4. Após o commit, o log é forçado ao disco.
5. Periodicamente, o **checkpoint** grava as páginas modificadas no disco físico.

Se quiser, posso te mostrar onde ficam fisicamente os arquivos desses SGBDs em um sistema Linux ou Windows. Também posso comparar a estratégia de armazenamento entre PostgreSQL e MySQL. Deseja isso?

Desejo

Perfeito! Vamos dividir em duas partes:

1. Localização Física dos Arquivos (Linux e Windows)

PostgreSQL

Linux (padrão via `apt`, `dnf` ou compilação)

- Diretório base dos dados:

```
/var/lib/postgresql/<versão>/main/
```

ou

```
/usr/local/pgsql/data/
```

- Estrutura comum:

```
pgsql
```

```
data/
```

```
├─ base/           # Dados reais das tabelas
├─ global/         # Dados globais (ex: roles, databases)
├─ pg_wal/         # Write-Ahead Logs (WAL)
```

```
└─ pg_stat/           # Estatísticas de execução
└─ pg_tblspc/         # Tablespaces (links para outros locais)
└─ postgresql.conf    # Arquivo de configuração principal
```

Windows

- Diretório configurado durante a instalação, como:

```
C:\Program Files\PostgreSQL\<versão>\data\
```

MySQL (InnoDB Engine)

Linux

- Diretório de dados padrão:

```
/var/lib/mysql/
```

- Estrutura comum:

```
pgsql

/var/lib/mysql/
└─ nome_do_banco/
  └─ tabela.ibd      # Arquivo com dados e índices da tabela
└─ ibdata1           # Tablespace compartilhado (se configurado)
└─ ib_logfile0,1     # Redo logs
└─ mysql/            # Dados do sistema (usuários, permissões)
└─ binlog.*          # Binários para replicação
```

Windows

- Normalmente fica em:

```
C:\ProgramData\MySQL\MySQL Server <versão>\Data\
```

2. Comparativo de Estratégias de Armazenamento

Característica	PostgreSQL	MySQL (InnoDB)
Formato das tabelas	Heap (sem ordem física)	Clustered (dados organizados pelo índice primário)
Tabelas separadas por arquivo?	Sim, cada tabela tem um OID e arquivos binários separados	Sim (modo file-per-table habilitado)
Índices	B-Tree (padrão), Hash, GiST, GIN, BRIN	B-Tree (padrão), Full-text
Tamanho da página	8KB padrão	16KB padrão (configurável)
Buffer em memória	Shared Buffers + WAL buffers	InnoDB Buffer Pool
Log de transações	Write-Ahead Logging (pg_wal/)	Redo/Undo Logs (ib_logfile*, rollback segments)
Undo/Redo separado?	Não, o undo é implementado via MVCC + tuplas antigas	Sim, undo e redo são separados
Multiversão (MVCC)	Sim, eficiente, usado por padrão	Sim, mas um pouco mais pesado
TOAST (dados grandes)	Sim, separa colunas grandes em arquivos auxiliares	Armazena dentro do .ibd, dependendo do tamanho
Recovery	WAL + checkpoints	Redo logs + undo logs + double write buffer

Se você quiser, posso te mostrar como navegar nesses diretórios e até ler os arquivos brutos com ferramentas como `pg_filedump` ou `hexdump`, ou ainda gerar um exemplo de tabela e mostrar onde e como ela será persistida fisicamente. Deseja algo mais prático?