



INSTITUTO FEDERAL
PIAUÍ
Campus Parnaíba

MySQL

Prof. Msc Denival A. dos Santos

Métodos de funcionamento e armazenamento

- Antes da criação de um banco de dados, tabelas e outros recursos no MySQL, é fundamental conhecer e optar pelos métodos de funcionamento e armazenamento disponibilizados pelo MySQL;
- Cada método possui características próprias, e devem ser utilizadas de acordo com cada situação. Uns são rápidos em relação ao processamento, outros ocupam menos espaço em disco, outros permitem o uso de transações e outros podem outras características.
- Para escolher o mais adequado para a sua necessidade é necessário conhecer todos os tipos de armazenamento e escolher de acordo com sua necessidade e que melhor se encaixa.



Connectors

Native C API, JDBC, ODBC, .NET, PHP, Python, Perl, Ruby, Cobol



MySQL Server

Management Services & Utilities

Backup & Recovery, Security, Replication, Cluster, Administration, Configuration, Migration, & Metadata



Connection Pool

Authentication - Thread Reuse - Connection Limits - Check Memory - Caches



SQL Interface

DML, DDL, Stored Procedures, Views, Triggers, etc.



Parser

Query Translation, Object Privilege



Optimizer

Access Paths, Statistics



Caches & Buffers

Global and Engine Specific Caches & Buffers



Pluggable Storage Engines Memory, Index & Storage Management



MyISAM



InnoDB



Archive



Federated



Memory



Merge



Cluster



BDB



Custom



File System

NTFS - NFS
SAN - NAS

Files & Logs

Redo, Undo, Data, Index, Binary, Error, Query, and Slow



- É um método de armazenamento muito rápido, de bom armazenamento em disco, sem restrições de uso de tipos de dados, que permite o uso de todos os recursos do MySQL, **com exceção do suporte a transações**;
- Seu nível de bloqueio (LOCK / UNLOCK) é de tabela, não tão prático quanto os de níveis mais aprofundados. Características do MyISAM:
 - Nome de uso: **MyISAM**
 - Suporte a índices: **Sim**
 - Suporte a transações: **Não**
 - Tipos de dados não suportados: **Nenhum**
 - Nível de bloqueio: **Tabela**

- É o método de armazenamento padrão configurado pelo MySQL, é o mais recomendado para grandes e complexos bancos de dados, pois além de oferecer todos os recursos disponibilizados pelo MyISAM, ainda permite o uso de transações com as propriedades ACID (Atomicidade, Consistência, Isolamento e Durabilidade);
- Além do armazenamento dos dados e índices em disco, algumas dessas informações são armazenadas também na memória enquanto o servidor está ativo, fazendo com que o seu processamento seja ainda mais veloz;
- Outra característica que o diferencia do método MyISAM é que seu nível de bloqueio (LOCKS) é de linha, sendo mais eficiente e aumentando sua disponibilidade, pois apenas os registros comprometidos em uma transação são bloqueados, e não a tabela toda;
- Recomendado para todos os tamanhos de banco de dados, especialmente os de grande porte. Possui as mesmas características do MyISAM com **exceção do suporte a transações**, que o MyISAM não possui.

MERGE

- Diferentemente do MyISAM e do InnoDB, o MERGE possibilita combinar duas ou mais tabelas, desde que idênticas em níveis estruturais em uma mesma expressão SQL de consulta;
- Este tipo de procedimento normalmente é associado a banco de dados cujas tabelas possuam estruturas idênticas, contudo cada uma é responsável pelo armazenamento de informações referentes a um determinado período (horas, dias, etc), o que caracteriza, de certa forma, esses bancos de dados em repositórios de informações de logs;
- Este método é apenas de funcionamento, e não de armazenamento. Ou seja, é possível definir o seu uso apenas em um banco de dados, contudo as tabelas utilizadas devem ser do tipo MyISAM obrigatoriamente, caso contrário poderão ocorrer falhas em seu mecanismo;
- O MERGE **não tem suporte a índices e transações** e seu nível de bloqueio é de tabela.

MEMORY (HEAP)

- Como o seu próprio nome informa, o armazenamento das informações de dados é realizada na memória RAM do computador em questão.
- Esse mecanismo de armazenamento fornece uma grande vantagem e uma grande desvantagem: a velocidade de processamento é a mais rápida, pois não há busca em disco, e, em contrapartida, os dados são perdidos quando o servidor é desligado ou reiniciado;
- Seu uso é indicado em aplicações cujas informações precisam ser armazenadas apenas temporariamente, como sessões, por exemplo. Desta forma, não há necessidade de armazená-las, e tira-se proveito apenas de sua velocidade de processamento;
- O MEMORY **não tem suporte a índices e transações**, não suporta tipo de dados BLOB e TEXT (por serem muito grandes) e o nível de bloqueio e de tabela.

BerkeleyDB (BDB)

- Este método de armazenamento se baseia na geração de um código HASH (identificador único) que é armazenado com cada registro inserido no banco de dados, funcionando como uma espécie de chave primária de registros do bando de dados;
- Sua uso mostra-se eficaz quando há recuperação de dados baseada em chaves únicas (HASH), fazendo com que a recuperação seja muitas vezes mais rápida que outros métodos de armazenamento. Contudo, a recuperação de múltiplos registros sequenciais pode ser mais lenta que o normal;
- A grande desvantagem é que se houver corrompimento das informações do HASH gerado para um determinado registro, todas as informações ligadas a ele poderão perde-se;
- O BDB **possui suporte a índices e transações** porém seu nível de bloqueio é de página (blocos de 8.192 bytes de memória).

ARCHIVE

- As informações neste método de armazenamento são gravadas em arquivos-texto, no formato de tabelas sequencias (ordem de gravação). Seu mecanismo de funcionamento é bastante limitado, pois suporta apenas os comandos de inserção (INSERT) e recuperação (SELECT) de dados;
- Como não há suporte a índices, cada SELECT necessita percorrer a tabela inteira para garantir que todos os resultados foram obtidos;
- Dependendo do como for usado (aplicações de log, por exemplo), seu uso pode ser eficiente em grandes bancos de dados, que geram um volume de informações muito grande, podendo ser armazenado posteriormente em mídias digitais ou unidades de backup;
- O ARCHIVE **não possui suporte a índices e transações e não existe níveis de bloqueio.**

CSV - Command Separated Values

- Similar ao método de armazenamento ARCHIVE, o CSV armazena os dados em um arquivo-texto, cujos valores são separados por vírgula ou outro caractere definido previamente;
- Seu uso normalmente é o de migração de backup de informações, pois meios de armazenamento do tipo texto não são apropriados para serem utilizados como banco de dados;
- Ao utilizar o CSV, suas informações tornam-se portáteis para outras aplicações de forma simples e rápida.
- Uma recomendação para este método de armazenamento é não utilizar formatos binários ou do tipo BLOB e TEXT muito grandes, pois podem comprometer sua portabilidade dependendo de onde serão utilizados posteriormente;
- O CSV não possui suporte a índices e transações e o nível de bloqueio é de tabela.

Na prática - exemplos da utilização das engines

```
1 create table tabela_modelo (  
2   id int,  
3   modelo varchar(50),  
4   data date  
5 ) ENGINE = MYISAM;  
6  
7 create table tabela_modelo2 (  
8   id int not null,  
9   modelo varchar(50),  
10  data date,  
11  primary key(id)  
12 ) ENGINE = InnoDB;
```

- **show engines** - mostra as engines suportadas por seu servidor e qual a padrão;
- **show table status** - exibe as informações de qual engine cada tabela do seu banco esta utilizando, além de outras informações;
- **alter table <nome_da_tabela> engine = <engine_desejado>** - muda o engine da tabela selecionada;
- **set storage_engine = <engine_desejado>** - alterar o padrão de storage da sessão.