

## BIG DATA - ATIVIDADE 08

**Aluno:** Arthur Vieira Toscano da Silva

**Atividade:** *Elaborar um material que possua imagens, animações, infográficos, timelines etc (não pode ser apenas textos) sobre um dos seguintes tópicos: Motores baseado em B-Tree (escolher um: MMAPv1, MyISAM, Aria, InnoDB, XtraDB e TokuDB) Motores baseado em LSM (escolher um: LevelDB, RocksDB, BerkeleyDB, DocDB, BadgerDB WiredTiger) Motores baseados em hashing (escolher um: Memory Engine, LMDB, Redis, Amazon DynamoDB ) Enviar tópico escolhido no Discord.*

- **Tópico escolhido:** B-Tree - InnoDB

**Observação:** Todas as fontes de onde tirei textos e imagens estão na parte de “Referências” ao fim do documento.

### O que é o InnoDB?

InnoDB é um mecanismo de armazenamento de uso geral que equilibra alta confiabilidade e alto desempenho. No MySQL 8.4, InnoDB é o mecanismo de armazenamento padrão do MySQL. A menos que você tenha configurado um mecanismo de armazenamento padrão diferente, emitir uma instrução CREATE TABLE sem uma cláusula ENGINE cria uma tabela InnoDB.

### Principais vantagens do InnoDB:

- Suas operações DML seguem o modelo ACID, com transações que incluem recursos de confirmação, reversão e recuperação de falhas para proteger os dados do usuário.
- Bloqueios em nível de linha e leituras consistentes no estilo Oracle aumentam a simultaneidade e o desempenho multiusuário.
- As tabelas InnoDB organizam seus dados no disco para otimizar consultas com base em chaves primárias. Cada tabela InnoDB possui um índice de chave primária, denominado índice clusterizado, que organiza os dados para minimizar a E/S para consultas de chave primária.
- Para manter a integridade dos dados, InnoDB suporta FOREIGN KEY restrições. Com chaves estrangeiras, inserções, atualizações e exclusões são verificadas para garantir que não resultem em inconsistências entre tabelas relacionadas.

A seguir uma tabela que mostra os recursos disponíveis ao usar InnoDB:

Tabela 17.1 Recursos do mecanismo de armazenamento InnoDB

Recurso	Apoiar
<i>Índices B-tree</i>	Sim
<i>Backup/recuperação de ponto no tempo</i> (implementado no servidor, em vez de no mecanismo de armazenamento.)	Sim
<i>Suporte a banco de dados de cluster</i>	Não
<i>Índices agrupados</i>	Sim
<i>Dados compactados</i>	Sim
<i>Caches de dados</i>	Sim
<i>Dados criptografados</i>	Sim (Implementado no servidor por meio de funções de criptografia; no MySQL 5.7 e posteriores, a criptografia de dados em repouso é suportada.)
<i>Suporte a chaves estrangeiras</i>	Sim
<i>Índices de pesquisa de texto completo</i>	Sim (o suporte para índices FULLTEXT está disponível no MySQL 5.6 e posteriores.)
<i>Suporte a tipos de dados geoespaciais</i>	Sim
<i>Suporte de indexação geoespacial</i>	Sim (o suporte para indexação geoespacial está disponível no MySQL 5.7 e posteriores.)
<i>Índices de hash</i>	Não (o InnoDB utiliza índices de hash internamente para seu recurso Índice de Hash Adaptável.)
<i>Caches de índice</i>	Sim
<i>Granularidade de bloqueio</i>	Linha
<i>MVCC</i>	Sim
<i>Suporte à replicação</i> (implementado no servidor, em vez de no mecanismo de armazenamento.)	Sim
<i>Limites de armazenamento</i>	64 TB
<i>Índices T-tree</i>	Não
<i>Transações</i>	Sim
<i>Atualizar estatísticas para dicionário de dados</i>	Sim

## Algumas características do InnoDB:

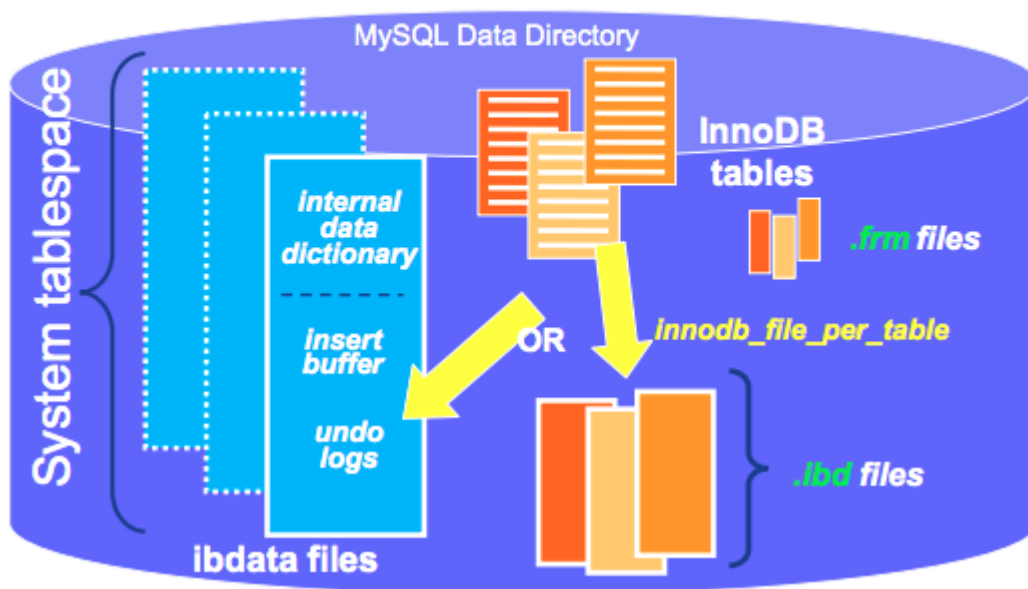
- Todas as tabelas INNODB são representadas em disco por um arquivo de extensão .frm no diretório da banco de dados e armazena dados e índices em tablespaces. Tais tablespaces são padronizadas e são localizadas no diretório data ou datadir. Também por padrão, todas as tabelas INNODB de um servidor compartilham um mesmo tablespace. O INNODB ainda nos permite configurar que cada tabela tenha sua própria tablespace, evitando problemas com arquivos grandes
- INNODB tem multi-versionamento e bloqueio em nível de linha, ou seja, caso uma transação A esteja atualizando uma determinada linha de uma tabela, esta será bloqueada até que haja um COMMIT ou ROLLBACK. O multi-versionamento provê à transação uma visão própria do estado do banco de dados, combinado com o bloqueio em nível de linha, provê o mínimo possível de contenções
- O INNODB trabalha basicamente utilizando dois recursos em disco, os tablespaces que armazenam dados e índices e um conjunto de arquivos de

log que registram todas as transações que manipulam dados em tabelas INNODB no servidor MySQL

### Funcionamento do tablespace no InnoDB:

O tablespace no InnoDB funciona como um grande espaço lógico de armazenamento que pode estar em um único arquivo ou em vários arquivos dependendo da configuração do banco de dados dentro desse espaço ficam guardados os dados das tabelas os índices e informações auxiliares cada tabela ou índice é organizado em segmentos e cada segmento é dividido em extensões chamadas extents cada extent ocupa 1 megabyte e dentro dele estão várias páginas que são as menores unidades de armazenamento do InnoDB cada página tem 16 kilobytes e pode guardar registros de dados índices ou informações de controle dessa forma quando você cria uma tabela no InnoDB ela ganha segmentos próprios que são preenchidos por extents e páginas conforme você insere linhas e cria índices o InnoDB vai alocando mais espaço dentro do tablespace garantindo a organização e a eficiência do acesso aos dados

## InnoDB Database Files



## **InnoDB + B-Tree:**

O InnoDB utiliza uma estrutura B+Tree para seus índices. Uma B+Tree é particularmente eficiente quando os dados não cabem na memória e precisam ser lidos do disco, pois garante que um número máximo fixo de leituras seja necessário para acessar quaisquer dados solicitados, com base apenas na profundidade da árvore, que é escalável.

Uma árvore de índices começa em uma página "raiz", cuja localização é fixa (e armazenada permanentemente no dicionário de dados do InnoDB) como ponto de partida para acessar a árvore. A árvore pode ser tão pequena quanto uma única página raiz ou tão grande quanto milhões de páginas em uma árvore de vários níveis.

As páginas são chamadas de páginas "folha" ou páginas "não-folha" (também chamadas de páginas "internas" ou "nós" em alguns contextos). As páginas folha contêm dados de linha reais. As páginas não-folha contêm apenas ponteiros para outras páginas não-folha ou para páginas folha. A árvore é balanceada, de modo que todos os ramos da árvore têm a mesma profundidade.

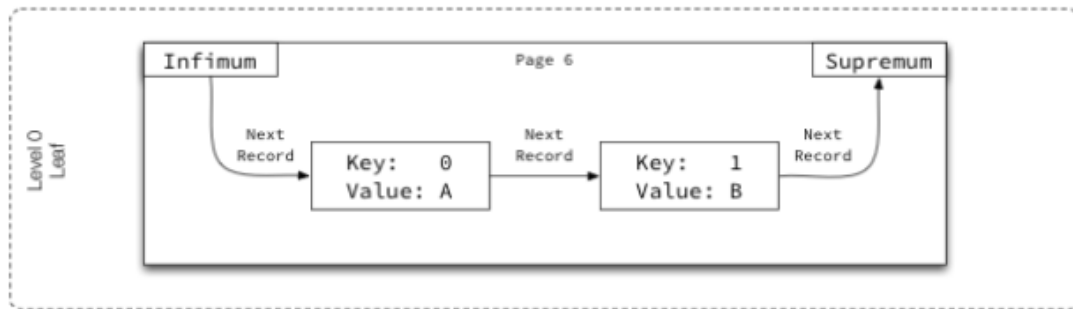
O InnoDB atribui um "nível" a cada página na árvore: páginas-folha recebem o nível 0, e o nível aumenta conforme a árvore sobe. O nível da página raiz é baseado na profundidade da árvore. Todas as páginas que não são folhas nem raiz também podem ser chamadas de páginas "internas", se uma distinção for importante.

### **Páginas de folhas e não folhas**

Tanto para páginas folha quanto para páginas não folha, cada registro (incluindo os registros do sistema ínfimo e supremo) contém um ponteiro para "próximo registro", que armazena um deslocamento (dentro da página) para o próximo registro. A lista encadeada começa no ínfimo e vincula todos os registros em ordem crescente por chave, terminando no supremo. Os registros não são fisicamente ordenados dentro da página (eles ocupam qualquer espaço disponível no momento da inserção); sua única ordem vem de sua posição na lista encadeada.

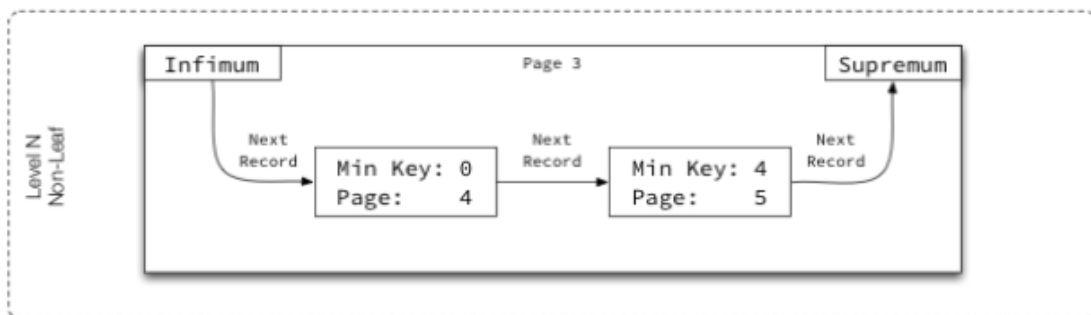
As páginas de folha contêm os valores não-chave como parte dos "dados" contidos em cada registro:

## B+Tree Simplified Leaf Page



As páginas não-folha têm uma estrutura idêntica, mas em vez de campos não-chave, seus “dados” são o número da página da página filha e, em vez de uma chave exata, representam a chave mínima na página filha para a qual apontam:

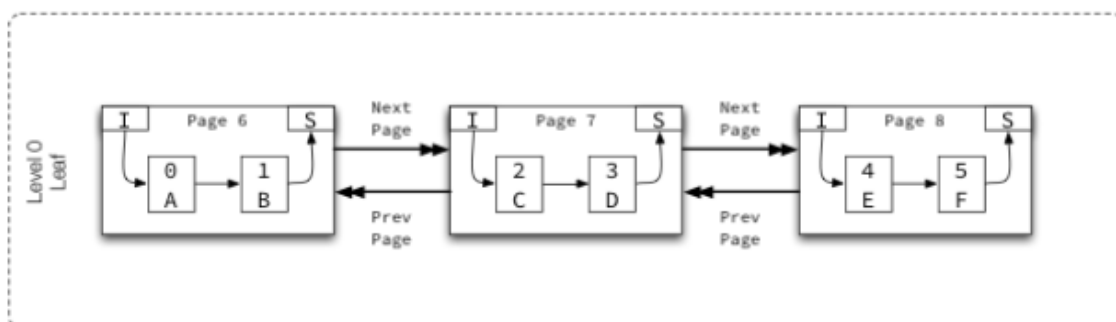
## B+Tree Simplified Non-Leaf Page



### Páginas no mesmo nível

A maioria dos índices contém mais de uma página, portanto, várias páginas são vinculadas em ordem crescente e decrescente:

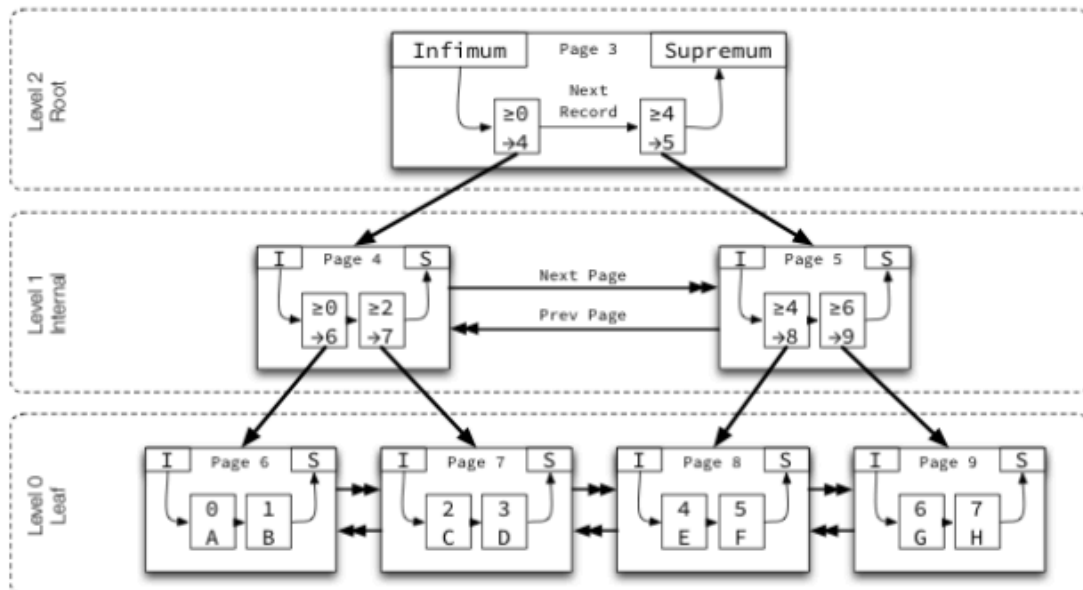
## B+Tree Simplified Level



Cada página contém ponteiros (no cabeçalho FIL ) para “página anterior” e “próxima página”, que para páginas INDEX são usados para formar uma lista duplamente vinculada de páginas no mesmo nível (por exemplo, páginas folha, no nível 0 formam uma lista, páginas de nível 1 formam uma lista separada, etc.).

### Estrutura de uma B-Tree no InnoDB:

## B+Tree Structure



**Referências:**

<https://dev.mysql.com/doc/refman/8.4/en/innodb-introduction.html>

<https://imasters.com.br/banco-de-dados/mysql/innodb-introducao-e-principais-caracteristicas>

[https://jackyfk.github.io/cs/data\\_store/mysql/innodb.html](https://jackyfk.github.io/cs/data_store/mysql/innodb.html)

<https://blog.jcole.us/2013/01/10/btree-index-structures-in-innodb/>