

supporte efficacement les correspondances exactes, par plage, par clé partielle et par motif. les fichiers volatiles sont traités efficacement car l'index est dynamique - il s'étend et se contracte au fur et à mesure que la table s'agrandit et se réduit. moins bien adapté aux fichiers relativement stables - dans ce cas, ISAM est plus efficace.

Fonctionnement des B-Tree Visualisation Implémenter B-TREE en java pour les opérations suivantes :

Recherche Insertion Suppression Vous avez 15 minutes...

voir <https://www.programiz.com/dsa/b-tree>

Variante B+Tree :

Parcours séquentiel + recherche indexée

Fonctionnement des B+Tree Visualisation ISAM ISAM (acronyme de l'anglais Indexed Sequential Access Method)

Organisation séquentielle indexée : Permet un accès séquentiel ou direct (par index)

ISAM, comme les fichiers séquentiels physiques, stocke les enregistrements les uns à la suite des autres, ce qui permet une utilisation très efficace de l'espace disque. Mais contrairement aux fichiers séquentiels physiques, les données ISAM doivent être stockés sur le disque, car les adresses du disque sont nécessaires pour créer les index. L'emplacement physique des enregistrements dans ISAM n'est pas important puisque les index se chargent de l'accès aux enregistrements. Un seul fichier ISAM peut comporter plusieurs dizaines d'index, chacun permettant de retrouver les fichiers dans un ordre prédéfini. Dans certains cas, la taille des index dépassera la taille du fichier de base. MyISAM utilise B-Tree

Mysql ⇒ MyISAM PostGreSQL ⇒ Unlogged tables Oracle ⇒ Tablespaces MyISAM ne gère ni les clés étrangères, ni les transactions. Il n'y a pas à vérifier la validité des enregistrements. Cela permet donc un précieux gain de temps sur des tables très fréquemment ouvertes en écriture/lecture.

Lors de suppressions sur des champs de type VARCHAR, CHAR, BLOB ou TEXT, le moteur supprime le contenu mais la place précédemment supprimée est conservée et peut être réutilisée ultérieurement. L'utilisation de OPTIMIZE peut défragmenter la table afin de gagner de la place et ainsi faciliter l'accès aux données.

MyISAM ⇒ très utilisé pour le Web vitrine.

Une table MyISAM utilise trois fichiers :

maTable.FRM : Fichier de définition de la table maTable.MYD : Fichier contenant les données de la table maTable.MYI : Fichier d'index Permet les recherches en FULL-TEXT ⇒ Recherche avec retour de pertinence

Exemple :

```
SELECT *, MATCH (Titre, Article) AGAINST ('base de données') AS Score FROM Article ORDER BY score DESC;
Retour :
```

IdArticle | Titre | Article | Score 7 | Federated | Le moteur Federated permet de déporter... | 0,21985759722453
6 | Exemple | Ce type de table est assez particulier... | 0,15944281721118 8 | InnoDB | C'est le moteur
transactionnel le plus utilisé... | 0,14023887676722 1 | Mysam | Ce moteur est une version évoluée d'ISAM
avec... | 0,068685997386924 2 | Memory | Les tables de type Memory stockent les... | 0 Avantages Moteur
rapide. Possibilité d'écrire et lire en même temps sans risque de verrouillage de table. Verrouillage de table
manuel. La mise en cache des clés. Gain de place sur le disque. Gain de mémoire lors des mises à jour. Gestion
de la recherche FULL-TEXT. Inconvénients Pas de gestion des contraintes de clés étrangères Pas de gestion des
transactions (pas de COMMIT / ROLLBACK possible). InnoDB InnoDB est le moteur par défaut à partir de MySQL
5.5.52. Son principal avantage par rapport aux autres moteurs de stockage de MySQL est qu'il permet des
transactions ACID (Atomiques, Cohérentes, Isolées et Durables), ainsi que la gestion des clés étrangères (avec
vérification de la cohérence). Autrement dit, InnoDB est un moteur de bases de données relationnelles et
transactionnelles, comparable à celui utilisé par PostgreSQL.

Toutes les bases de données sont stockées au même endroit. Par défaut dans le fichier ibdata qui se trouve généralement dans le dossier /var/lib/mysql/. Il est également possible d'utiliser plusieurs fichiers ou même d'utiliser directement une ou plusieurs partitions sur le disque en mode RAW.

Les fichiers de définitions de table .frm sont également dans un dossier au nom de la base comme pour MyISAM.

Les fichiers données de InnoDB ne peuvent pas être sauvegardés par copie de fichiers : il en résulterait une corruption de données. La seule façon de faire une copie des fichiers à chaud est d'utiliser Percona Xtrabackup pour MySQL et Mariabackup pour MariaDB, ou d'utiliser la commande mysqldump en mode "single transaction".

Depuis la version 5.6, une extension memcached peut être associée à InnoDB, permettant d'améliorer les performances.

Avantages Verrouillage de ligne. Gestion du COMMIT/ROLLBACK Gestion de gros volumes de données. Gestion des clés étrangères. Grande panoplie d'éléments de configuration du moteur. Gestion du backup sans bloquer une base en production. Couramment disponible chez les hébergeurs en mutualisé. Inconvénients Lenteur de certaines opérations telles que le SELECT COUNT(*) FROM maTable. Statistiques envoyées ne sont pas forcément précises : ce ne sont que des estimations. Choix d'organisation séquentiel (HEAP) Conditions favorables :

chargement de données dans la base petites tables (occupation de peu de pages) requêtes manipulant des tables entières pour récupérer de l'espace dû à des DELETE Eviter :

accès à 1 ou plusieurs tuples grosses tables HASH Conditions favorables :

recherche sur valeur exacte de clé (le plus rapide) Eviter :

recherche sur pattern matching (partie de clé) traitement de table entière joints naturels (systématique sans restriction) ISAM Conditions favorables :

requêtes nécessitant pattern matching ++ la table grossit lentement (peu de réorg.) clé large Eviter :

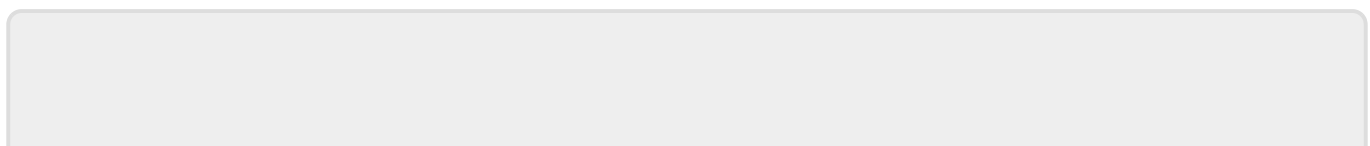
si recherche sur clé complète (→HASH) grosse table à croissance rapide BTREE Conditions favorables :

besoin de pattern matching + la table grossit vite table trop grosse pour être souvent réorganisée (Modify) joints de tables entières Eviter :

table statique ou à croissance faible large clé si ajout de nouveaux tuples seulement en fin de table (plus grand risque de DEAD LOCK) Résumé Fonctionnalité B-Tree ISAM Hash Heap chargement de table ++ -- + recherche sur clé complète ++ ++ + - intervalles/pattern matching + + -- recherches séquentielles ++ ++ - + recherche sur clé partielle + + -- accès à données triées + --- joints sur larges tables + + -- index croît comme table + --- très petite table + --- très grande table + --- Source : Bernard Espinasse

Application Objectif :

Benchmark des moteurs de stockage MySQL (InnoDB, MyISAM) et PostgreSQL. Tests sur petits (100) et + gros volumes de données (10 000) sur les 6 primitives de base Etablir un protocole de test (nb de requêtes, concurrency level, méthode/outils utilisés) Créer les bases de données : Pour chaque moteur (X3) Pour chaque volume (X2) Incorporer les données avec GenerateData Mettre en place le protocole établi Réaliser les tests Présenter les résultats



From:

<http://slamwiki2.kobject.net/> - **SlamWiki 2.1**

Permanent link:

<http://slamwiki2.kobject.net/cnam/nfp107/seance8?rev=1682171139>

Last update: **2023/04/22 15:45**

