

数据库第六次作业

22373386 高铭

WEEK 8

什么是索引?

索引类似于书籍的目录，它们提供了一种快速查找数据的方式，而不必遍历整个数据库表。

- 是主表上的一种辅助数据结构
- 对数据库表中一个或多个列的值进行排序
- 用于快速定位和访问数据库表中的特定行或列，以提高主表的查询速度
- 如果想按特定列的值来查找数据，则与在表中搜索所有的行相比，索引有助于更快地获取信息。

简述B+树索引、Hash索引及聚簇索引的实现原理和优缺点

B+树索引

实现原理： B+树是一种多叉平衡树，它将所有的数据都存储在叶子节点，而内部节点不存储数据，只存储关键字和孩子指针。所有关键字的查找都会走一条从根节点到叶子节点的路径。

优点：

- 范围查找效率高：B+树索引有序性，可以进行范围查询，例如查找某一范围内的数据。
- 适合范围查询和排序：由于有序性，B+树索引在范围查询和排序方面性能良好。
- 支持高效的插入和删除操作：B+树的平衡性能够保证在插入和删除时维持较好的性能。

缺点：

- 插入、删除可能需要频繁的节点分裂、合并操作，影响性能。
- 数据分布不均匀时，B+树索引可能导致某些节点过度增长，造成不平衡。
- 不适用于查询的记录数占记录总数的百分比非常大的时候

Hash索引

实现原理： Hash索引基于哈希表实现，通过计算哈希值将记录映射到哈希槽中，然后指向对应记录行的地址。

1. 将索引列的值通过哈希函数映射为哈希值。
2. 哈希值作为索引，直接指向对应的数据位置。
3. 哈希索引没有顺序性，不支持范围查询。

优点：

- 查找效率高：哈希索引的查找复杂度为 $O(1)$ ，即常数时间，因此查找速度非常快。
- 适合等值查找：对于等值查找（例如根据主键查找）效率极高。

缺点：

- 不支持范围查询：由于哈希索引没有顺序性，无法进行范围查询和排序操作。
- 哈希冲突：如果哈希函数设计不当或者数据分布不均匀，容易导致哈希冲突，影响性能。
- 重构代价很大，不适合在更新频繁的表中建立哈希索引

聚簇索引

实现原理： 聚簇索引定义了数据表中行的物理存储顺序。当在表上创建聚簇索引时，数据库系统会根据指定的列或列组合，将数据行重新组织为一个物理上的顺序，通常基于聚簇索引列的值。这样，具有相似聚簇索引值的行会相邻存储，从而提高查询性能。一个表只能有一个聚簇索引，因为它定义了数据行的物理存储顺序。

优点：

- 查询性能高：由于数据行物理上相邻，因此范围查询、排序等操作性能较好。
- 避免了额外的查找步骤：由于索引叶子节点直接存储数据，不需要额外的指针查找步骤。

缺点：

- 更新成本较高：插入、删除等操作可能需要移动数据行，影响性能。
- 数据插入时需要保证键值的有序性，可能导致插入性能下降。
- 对于范围删除或者更新操作，可能需要大量的数据移动，开销较大。