

数据库笔记

1. 脏读，不可重复读，幻读：

- 脏读：一进程读到另一进程尚未commit的内容，另一进程可能之后rollback，所以变成dirty data。数据库隔离切换到READ_COMMITTED可防止。
- 不可重复读：一进程读取数据期间，另一进程修改了该数据，于是原进程两次读取的内容发生变化。数据库隔离级别为REPEATABLE_READ可防止。
- 幻读：一进程处理数据时，另一进程插入或删除了新的记录，导致原事务结束后还有新数据，像发生幻觉一样。数据库级别升为SERIALIZABLE_READ。
- 不可重复读出现多是因为修改，幻读重点是新增、删除。

2. B+ 树和 B 树的区别？

- B 树非叶子结点和叶子结点都存储数据,因此查询数据时，时间复杂度最好为 $O(1)$,最坏为 $O(\log n)$ 。
- B+ 树只在叶子结点存储数据，非叶子结点存储关键字，且不同非叶子结点的关键字可能重复，因此查询数据时，时间复杂度固定为 $O(\log n)$ 。
- B+ 树叶子结点之间用链表相互连接，因而只需扫描叶子结点的链表就可以完成一次遍历操作，B树只能通过中序遍历。

3. 为什么 B+ 树比 B 树更适合应用于数据库索引？

- B+ 树更加适应磁盘的特性，相比 B 树减少了 I/O 读写的次数。由于索引文件很大因此索引文件存储在磁盘上，B+ 树的非叶子结点只存关键字不存数据，因而单个页可以存储更多的关键字，即一次性读入内存的需要查找的关键字也就越多，磁盘的随机 I/O 读取次数相对就减少了。
- B+ 树的查询效率相比B树更加稳定，由于数据只存在在叶子结点上，所以查找效率固定为 $O(\log n)$ 。
- B+ 树叶子结点之间用链表有序连接，所以扫描全部数据只需扫描一遍叶子结点，利于扫库和范围查询；B 树由于非叶子结点也存数据，所以只能通过中序遍历按序来扫。也就是说，对于范围查询和有序遍历而言，B+ 树的效率更高。

4. 什么时候会全表扫描

- 模糊查询效率很低，%like%无法使用索引，但右模糊查询（like%）还是可以使用的，左模糊查询可以反转称为右模糊，全模糊无法优化
- 查询条件中含有is null的select语句执行慢。优化：对于is null可以建立组合索引。
- 查询条件中使用了不等于操作符（<>、!=）的select语句执行慢。优化：!= 改为or，可以使用索引避免全表扫描（例如，把column<>'aaa'，改成column<'aaa' or column>'aaa'，就可以使用索引了）
- or语句使用不当会引起全表扫描。where子句中比较的两个条件，一个有索引，一个没索引，使用or则会引起全表扫描。例如：where A1 or B2，A上有索引，B上没索引，则比较B=: 2 时会重新开始全表扫描。
- select count(*) from table；这样不带任何条件的count会引起全表扫描，并且没有任何业务意义，是一定要杜绝的。