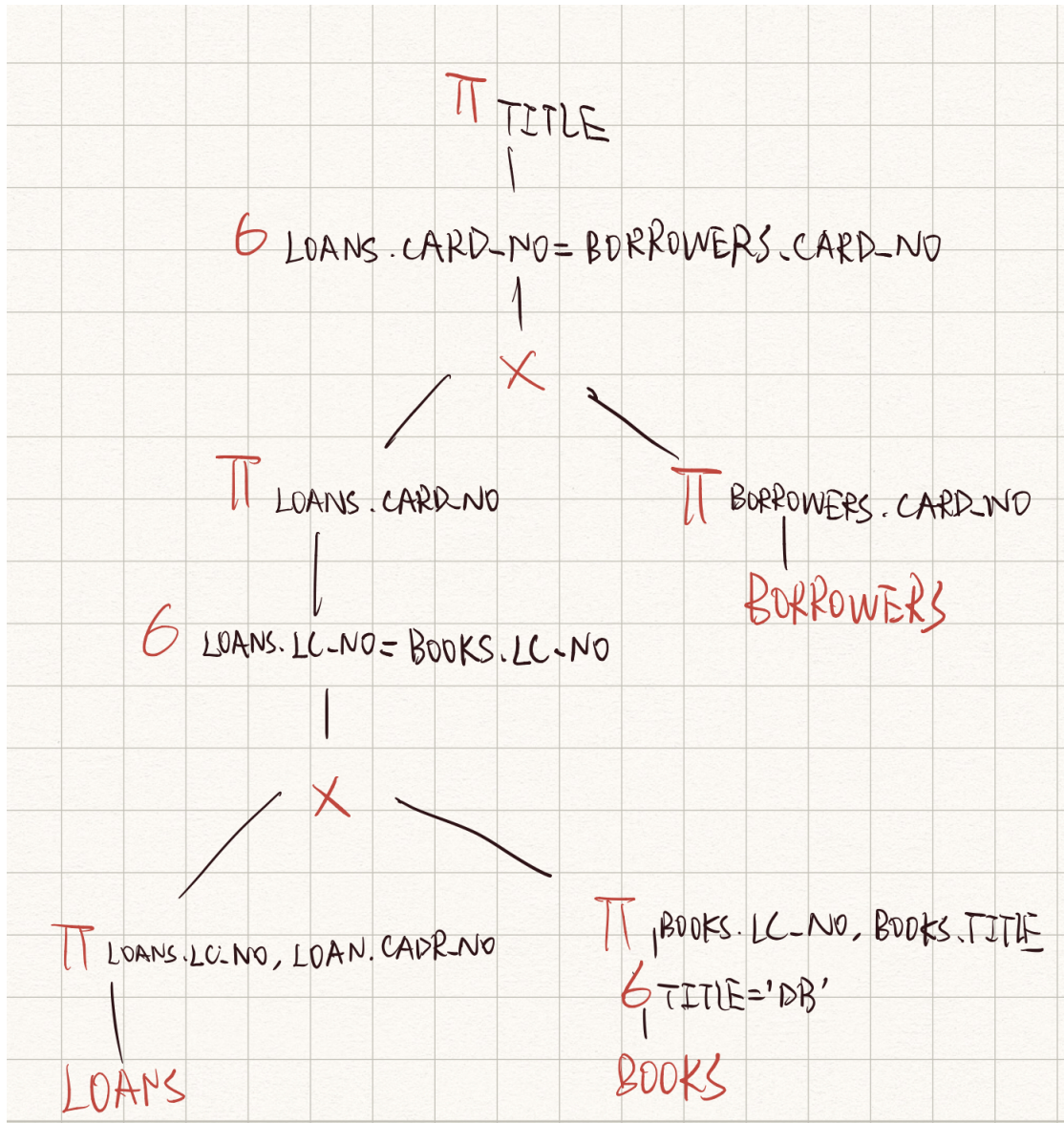


查询过程 优化 索引 存储 建模 算法实现过程优化树 实现过程 存储结构

查询过程 优化 索引 分类 容量 硬盘容量计算 IDEF1X图

- 查询优化



- 索引相关

1. (简答题)为什么MySQL、Oracle等关系型数据库采用B+树作为索引，而不是B树，Hash等索引？

**我的答案：**

B树能够在非叶节点中存储数据，但也会导致在查询连续数据时可能会带来更多随机I/O。使用B+树，所有叶节点可以通过指针相互连接，可以减少顺序遍历时产生的额外随机I/O。

Hash虽然能够提供O(1)的时间复杂度，但是需要将数据全部加在到内存中，当数据量大时，不可能全部存在内存中，需要存储在磁盘中。而使用B+树，基于按照节点分段加载，会减少内存消耗；此外，数据库中经常需要查询多条数据，B+树数据的有序性，与叶子节点有链表相连，查询效率要比Hash高。

## 2. (简答题)MySQL的InnoDB以及MyISAM引擎在实现B+树索引时有哪些不同点?

### 我的答案:

主索引的区别: InnoDB的数据文件本身就是索引文件, data存的是数据本身, 索引也是数据; MyISAM的索引和数据是分开的, data存的是数据地址, 索引时索引, 数据是数据。

辅助索引的区别: InnoDB的辅助索引data域存储相应记录主键的值而不是地址; MyISAM的辅助索引和主索引区别不大。

MyISAM 的每个索引都是单独的一棵树, 每个索引都存储有真实的数据区地址, 而 InnoDB 只有主键索引树才存储有真实地址, 而辅助索引树的叶子节点存储的是主键的关键字。

MyISAM 每个索引树都可以独当一面, 而 InnoDB 的辅助索引树就算找到了对应的关键字, 也还是要到叶子节点拿到主键的关键字, 然后再去主键索引树遍历。

MyISAM 没有默认的主键索引, 而 InnoDB 有默认的主键索引(聚集索引)(不明确指定的情况下), InnoDB 除了主键索引是聚集索引, 其他都是非聚集索引。

## 3. (简答题)要实现千万级别的记录存储, InnoDB中B+索引树的高度至少为几? 请说明理由

### 我的答案:

假设表的记录数为N, 每个树节点平均有B个索引KEY, B+索引树的高度为 $\log B / \log N$ 。

假设表中有1600w条记录, 若每个节点保存64个索引KEY, 索引的高度为 $\log_2^{24} / \log_2 64 \approx 24/6=4$ 。

以BIGINT为例, 存储大小为8byte。INT存储大小为4byte。索引树的每个节点除了存储KEY, 还要存储指针, 所以每个节点保存的KEY的数量:  $\text{pagesize} / (\text{keysize} + \text{pointsize})$ 。假设平均指针大小为4byte, 每个节点可以存储 $16k / ((8+4)*8) \approx 171$ , 则拥有3000w记录的数据, 索引树的高度为

$\log_2^{25} / \log_2 171 \approx 25/7.4 \approx 3.38$ ;

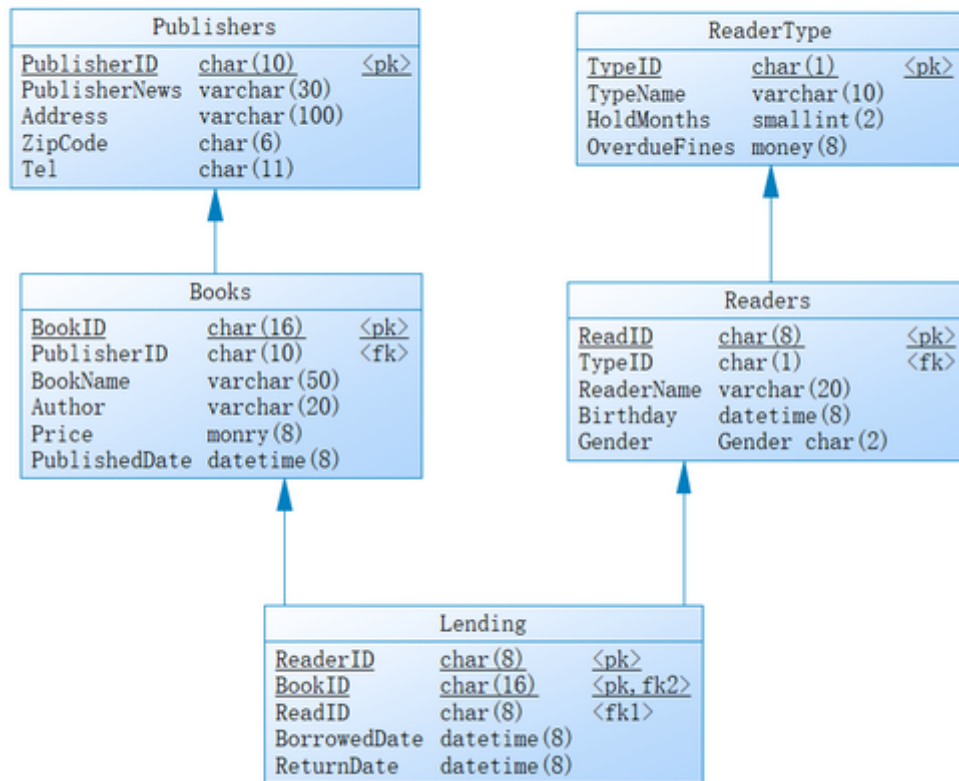
假设平均指针大小为8byte, 每个节点可以存储 $16k / ((8+8)*8) \approx 128$ , 则拥有3000w记录的数据, 索引树的高度为

$\log_2^{25} / \log_2 128 \approx 25/7 \approx 3.57$ ;

因此要实现千万级别的记录存储, InnoDB中B+索引树高度至少为3。

## • 图书管理数据库设计

我的答案：



- 建模

见建模pdf、IDEF1X习题pdf和数据建模习题docx

- 存储

第四讲ppt：存储与索引

第五讲ppt：查询与优化