



DeepL

订阅 DeepL Pro 以编辑此演示文稿。

访问 www.DeepL.com/pro，了解更多信息。

文件组织

R & G - 第九章



DBMS 的结构

已完成



SQL 客户端

我们将访问



查询解析
优化

关系操作符

文件和索引管理

缓冲区管理

你在这里



磁盘空间管理

数据库



召回。堆积文件

- 无序的记录集合
- 回顾一下 DBMS 高层的 API。
今天我们要问的是："如何？以什么为代价？"
 - 插入 / 删除 / 修改记录
 - 通过 **记录ID** 获取一个特定的记录 ...
 - 记录 id 是一对指针式编码 (**pageID** , 页面上的位置) 。
 - 扫描所有记录
 - 可能会对要检索的记录设置一些条件

回顾一下。多个文件组织

存在许多替代方案，每种方案在某些情况下都很好，而在另一些情况下则不那么好。

这是 DB 系统工作中的一个主题！

- **堆积文件。**当典型的访问是对所有记录进行全面扫描时适合。
- **排序的文件。**最适合按顺序检索，或需要一定范围的记录时。
- **集群文件和索引。**将数据分组，以实现快速查找和有效修改。
 - 更多关于这个问题的信息很快就会出现

更大的问题

- 什么是 " 最好的 " 文件组织？
 - 取决于访问模式 。
 - 如何？常见的访问模式到底是什么？
- 我们能不能对权衡进行量化？
 - 如果一个人更好 是多少？

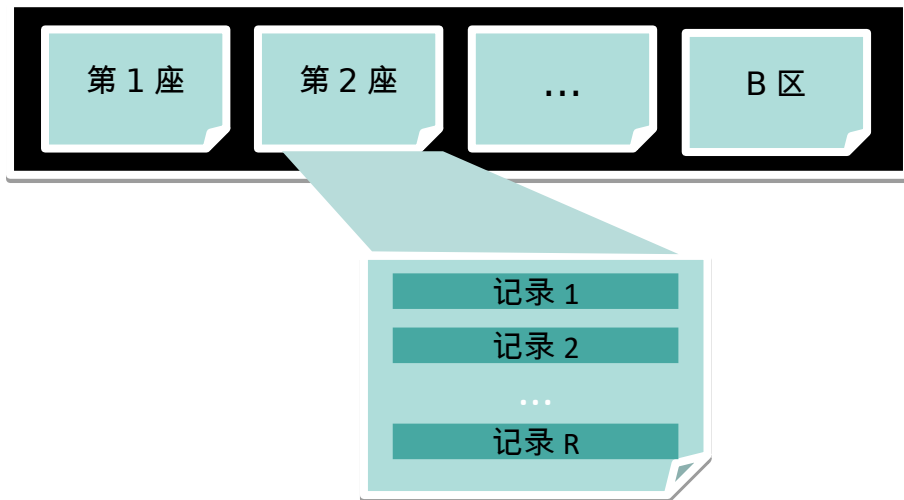
目标

- 数据访问的大画面开销
 - 我们会（过度）简化性能模型以提供洞察力，而不是为了获得完美的性能
 - 还是要有一点纪律。
 - 提前清楚地识别假设
 - 然后以一种有原则的方式估计成本
- 查询优化的基础
 - 没有对速度的估计，就不能选择最快的方案！

成本模型和分析

用于分析的成本模型

- **B** : 文件中数据块的数量
- **R** : 每块记录的数量
- **D**: 读 / 写磁盘块的 (平均) 时间
- 关注。均匀随机工作负载的平均案例分析
- 现在，我们将忽略
 - 顺序与随机 I/O
 - 预取
 - 任何内存中的费用
- 足够好地显示总体趋势



更多假设

- 单一记录的插入和删除
- 等价选择 -- 正好有一个匹配
- 对于堆积文件。
 - 插入总是附加到文件的末尾。
- 对于分类的文件。
 - 已打包。文件在删除后被压缩。
 - 根据搜索关键词排序

额外的挑战

- 在了解这些幻灯片后 ...
 - 你应该质疑所有这些假设，并重新制定
 - 为测试而学习，并产生想法的良好练习

堆积文件和分类文件

堆积文件



已分类的文件



为了说明问题，记录只是整数

- **B** : 数据块的数量 = 5
- **R** : 每块记录的数量 = 2
- **D** : 读 / 写磁盘块的 (平均) 时间 = 5ms

运营成本。扫描？

	堆积文件	已分类的文件
扫描所有记录		
平等搜索		
范围搜索		
插入		
删除		

- **B** : 数据块的数量 =5
- **R** : 每块记录的数量 =2
- **D** : 读 / 写磁盘块的 (平均) 时间 =5ms

扫描所有记录

堆积文件



已分类的文件



- **B** : 数据块的数量
- **R** : 每块记录的数量
- **D**: 读 / 写磁盘块的平均时间
- 触及的页面 : ?
- 阅读记录的时间 : ?

业务成本。扫描成本

	堆积文件	已分类的文件
扫描所有记录	$B * D$	$B * D$
平等搜索		
范围搜索		
插入		
删除		

- **B** : 数据块的数量
- **R** : 每块记录的数量
- **D**: 读 / 写磁盘区块的平均时间

运营成本。平等搜索？

	堆积文件	已分类的文件
扫描所有记录	$B \cdot D$	$B \cdot D$
平等搜索		
范围搜索		
插入		
删除		

- **B** : 数据块的数量
- **R** : 每块记录的数量
- **D**: 读 / 写磁盘块的平均时间

查找键 8 : 堆积文件

堆积文件

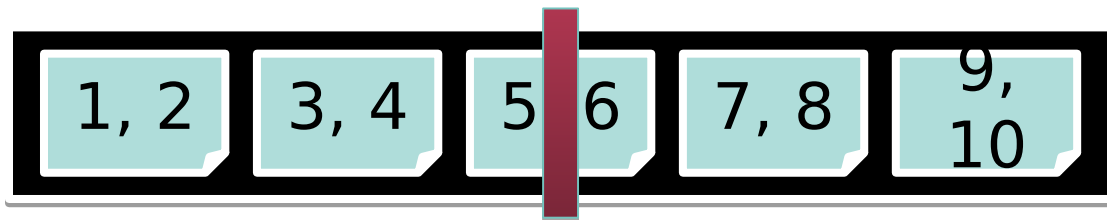


- **P(i)** : 钥匙在**第*i***页的概率为 **1/B**
- **T(i)**: 如果**第*i***页上的键是 **i** , 所触及的页面数
- 因此, 预计触及的页数
- 平均触及的页数?

$$\sum_{i=1}^B T(i)P(i) = \sum_{i=1}^B i \frac{1}{B} = \frac{B(B+1)}{2B} \approx \frac{B}{2}$$

查找键 8：分类的文件

已分类的文件



- **最坏的情况。**在二进制搜索中触及的页面
 - 对数₂ B
- **平均值的情况。**在二进制搜索中触及的页面
 - 日志₂ B?

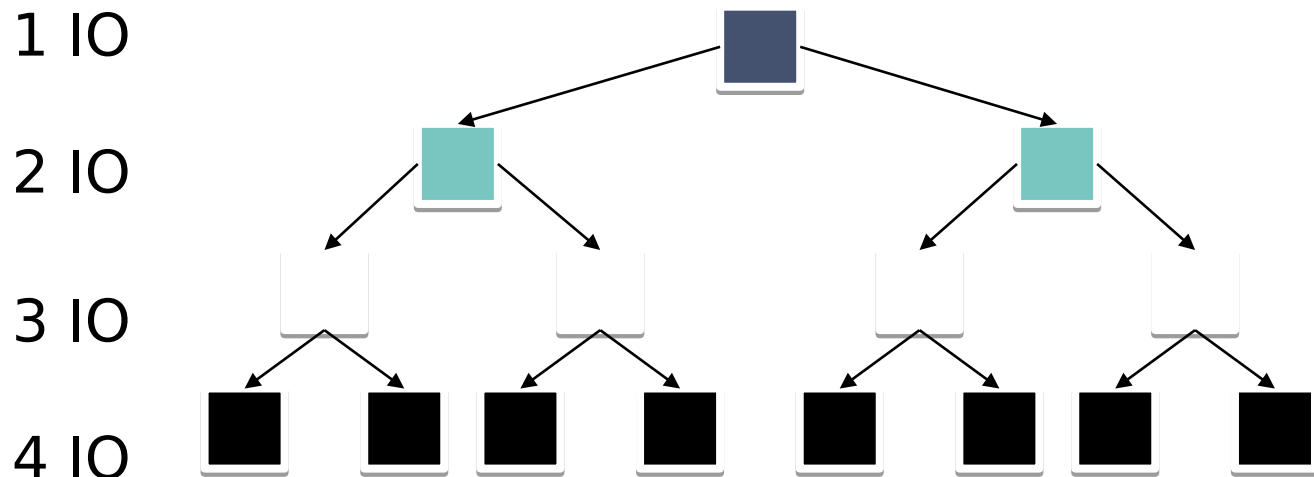
平均案例二进制搜索

预期的阅读数。 $1 (1 / b) + 2 (2 / b) + 3 (4 / b) + 4 (8 / b)$



平均案例二进制搜索 cont

预期的阅读数。 $1 (1 / b) + 2 (2 / b) + 3 (4 / b) + 4 (8 / b)$



$$\sum_{i=1}^{\log_2 B} i \frac{2^{i-1}}{B} = \frac{1}{B} \sum_{i=1}^{\log_2 B} i 2^{i-1} = \log_2 B - \frac{B-1}{B}$$

运营成本。方程式搜索成本

	堆积文件	已分类的文件
扫描所有记录	$B * D$	$B * D$
平等搜索	$0.5 * B * D$	$(\log_2 B) * D$
范围搜索		
插入		
删除		

- **B** : 数据块的数量
- **R** : 每块记录的数量
- **D**: 读 / 写磁盘块的平均时间

运营成本。范围搜索？

	堆积文件	已分类的文件
扫描所有记录	$B * D$	$B * D$
平等搜索	$0.5 * B * D$	$(\log_2 B) * D$
范围搜索		
插入		
删除		

- **B** : 数据块的数量
- **R** : 每块记录的数量
- **D**: 读 / 写磁盘块的平均时间

查找 7 和 9 之间的键：堆文件

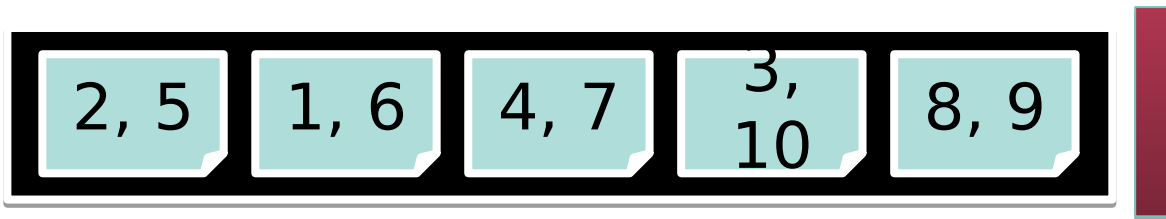
堆积文件



- 始终触摸所有区块。为什么？

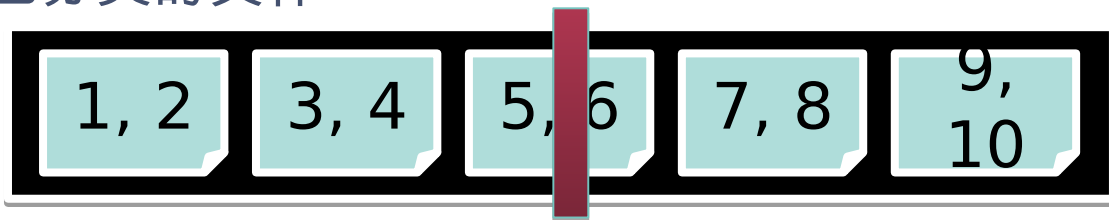
寻找 7 和 9 之间的钥匙：比较

堆积文件



- 寻找范围的开始

已分类的文件



- 搜索范围的起点
- 向右扫描

操作成本。范围搜索成本

	堆积文件	已分类的文件
扫描所有记录	$B * D$	$B * D$
平等搜索	$0.5 * B * D$	$(\log_2 B) * D$
范围搜索	$B * D$	$((\log_2 B) + \text{pages}) * D$
插入		
删除		

- **B** : 数据块的数量
- **R** : 每块记录的数量
- **D**: 读 / 写磁盘块的平均时间

业务成本。插入？

	堆积文件	已分类的文件
扫描所有记录	$B * D$	$B * D$
平等搜索	$0.5 * B * D$	$(\log_2 B) * D$
范围搜索	$B * D$	$((\log_2 B) + \text{pages}) * D$
插入		
删除		

- **B** : 数据块的数量
- **R** : 每块记录的数量
- **D**: 读 / 写磁盘块的平均时间

插入 4.5 : 堆积文件

堆积文件



- 粘在文件的末尾
- 成本 = $2 * D$
- 为什么是 2 ?

插入 4.5 : 堆 VS 分类文件

堆积文件



- 读取最后一页，追加，写入。成本 = $2 * D$

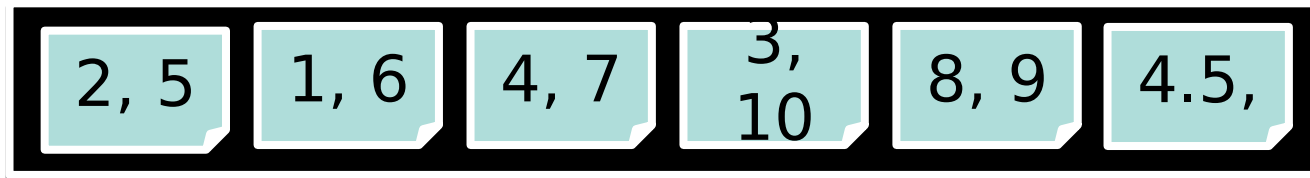
已分类的文件



- 找到记录的位置。成本 = $\log_2 BD$

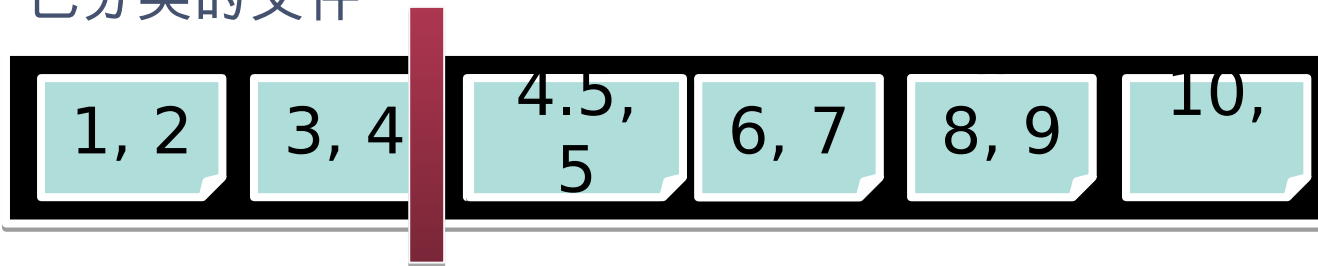
插件 4.5 : 堆与分类 Pt 2

堆积文件



- 读取最后一页，追加，写入。成本 = $2 * D$

已分类的文件



- 找到记录的位置。成本 = $\log_2 BD$
- 插入和转移文件的其余部分

业务成本。插入成本

	堆积文件	已分类的文件
扫描所有记录	$B * D$	$B * D$
平等搜索	$0.5 * B * D$	$(\log_2 B) * D$
范围搜索	$B * D$	$((\log_2 B) + \text{pages}) * D$
插入	$2 * D$	$((\log_2 B) + B) * D$
删除		

- **B** : 数据块的数量
- **R** : 每块记录的数量
- **D**: 读 / 写磁盘块的平均时间

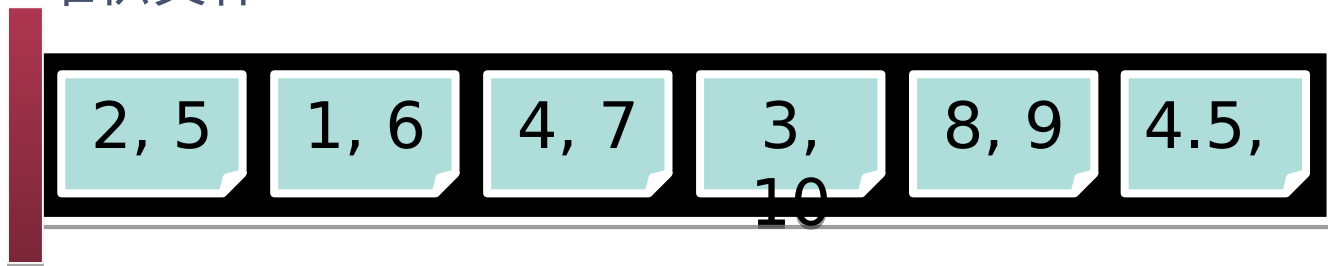
业务成本。删除？

	堆积文件	已分类的文件
扫描所有记录	$B * D$	$B * D$
平等搜索	$0.5 * B * D$	$(\log_2 B) * D$
范围搜索	$B * D$	$((\log_2 B) + \text{pages}) * D$
插入	$2 * D$	$((\log_2 B) + B) * D$
删除		

- **B** : 数据块的数量
- **R** : 每块记录的数量
- **D**: 读 / 写磁盘块的平均时间

删除 4.5 : 堆积文件

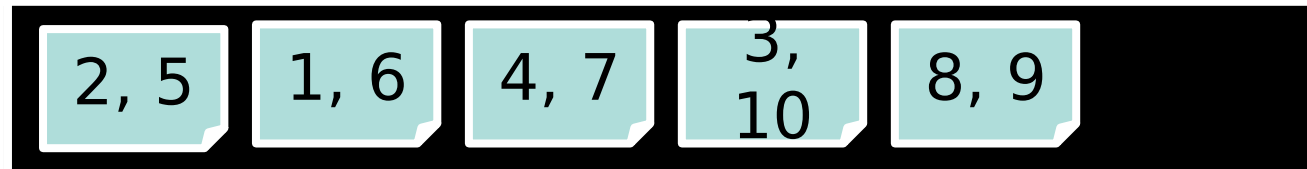
堆积文件



- 一般情况下，要找到记录。 **$B/2$ 读作**
- 删除页面上的记录
- 成本 = $(B/2 + 1) * D$
 - 为什么是 +1 ?

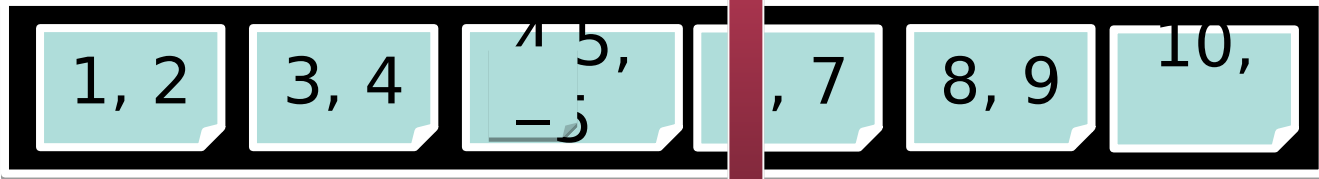
删除 4.5 : 堆积文件与排序文件

堆积文件



- 平均案例运行时间： $(B/2 + 1) * D$

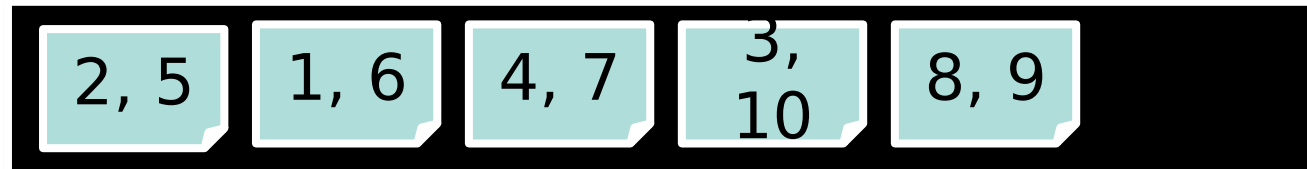
已分类的文件



- 查找记录的位置：日志 $\frac{B}{2}$
- 删除页面中的记录 Gap

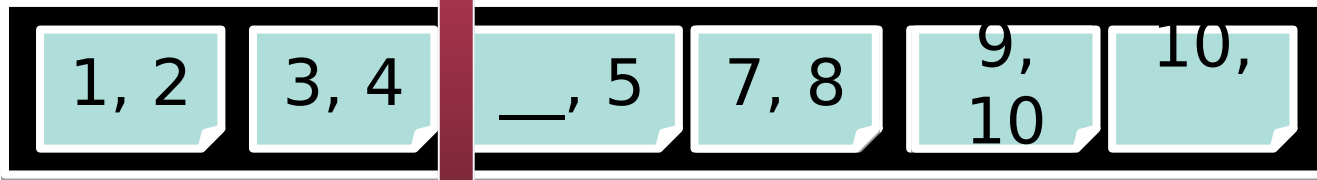
删除 4.5 : 堆积文件与分类文件的比较 Pt 2

堆积文件



- 平均案例运行时间： $(B/2 + 1) * D$

已分类的文件



- 查找记录的位置： $\log_2 B$
- 将剩下的部分移到 1 记录 $2 * (B/2)$

业务成本完成

	堆积文件	已分类的文件
扫描所有记录	$B * D$	$B * D$
平等搜索	$0.5 * B * D$	$(\log_2 B) * D$
范围搜索	$B * D$	$((\log_2 B) + \text{pages}) * D$
插入	$2 * D$	$((\log_2 B) + B) * D$
删除	$(0.5 * B + 1) * D$	$((\log_2 B) + B) * D$

- **B** : 数据块的数量
- **R** : 每块记录的数量
- **D**: 读 / 写磁盘块的平均时间

业务成本完整版 Pt 2

	堆积文件	已分类的文件
扫描所有记录	$B * D$	$B * D$
平等搜索	$0.5 * B * D$	$(\log_2 B) * D$
范围搜索	$B * D$	$((\log_2 B) + \text{pages}) * D$
插入	$2 * D$	$((\log_2 B) + B) * D$
删除	$(0.5 * B + 1) * D$	$((\log_2 B) + B) * D$

- **B** : 数据块的数量
- **R** : 每块记录的数量
- **D**: 读 / 写磁盘块的平均时间
- 我们能做得更好吗？
 - 索引！