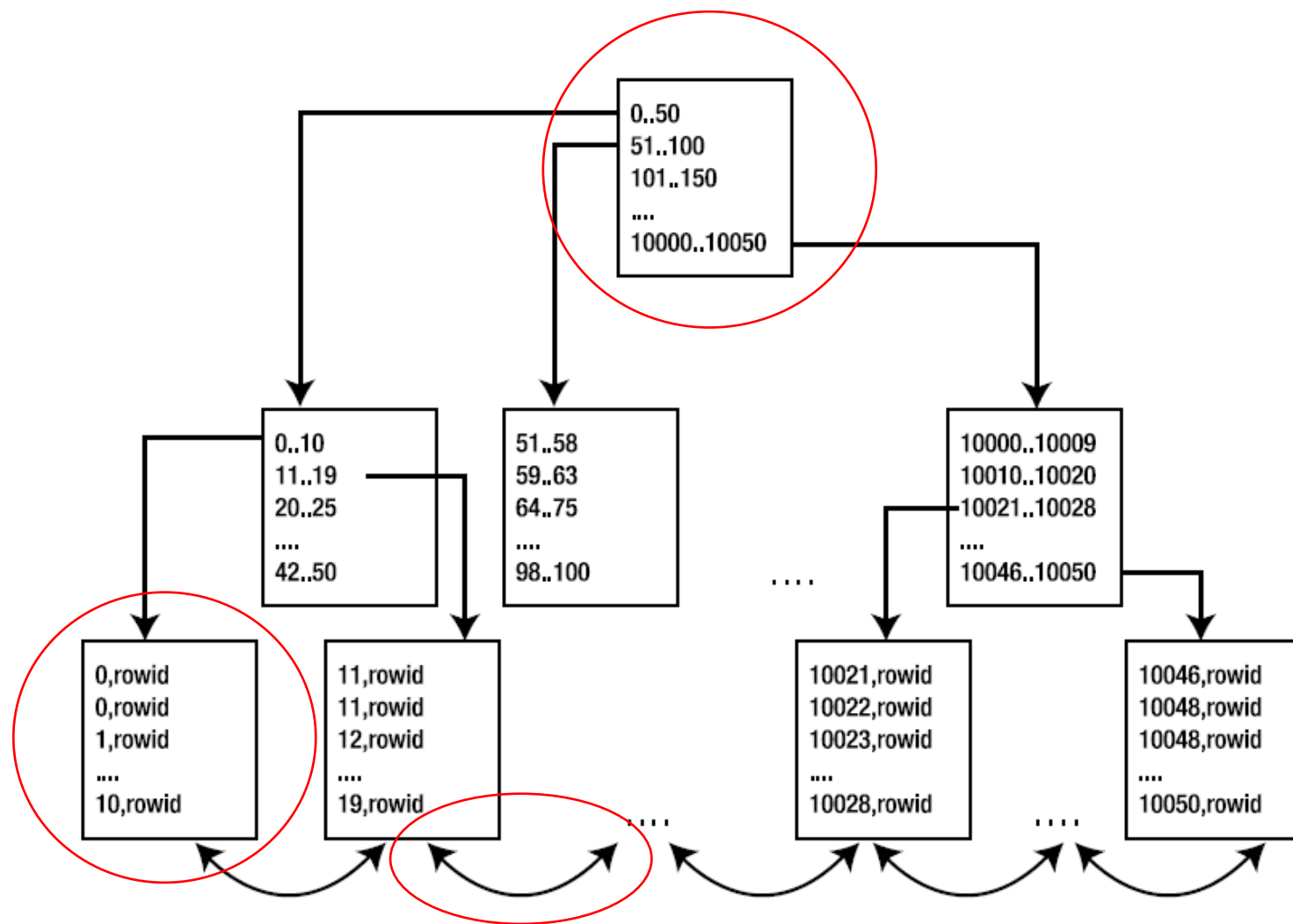


# 03 索引结构及使用

南京大学软件学院

# B-Tree (B+Tree) 的结构



# B+树索引能做什么？

- 充分理解B+树索引的结构，你就能充分B+树能做什么不能做什么

- 能做的

- 全键值

Where  $x=123$

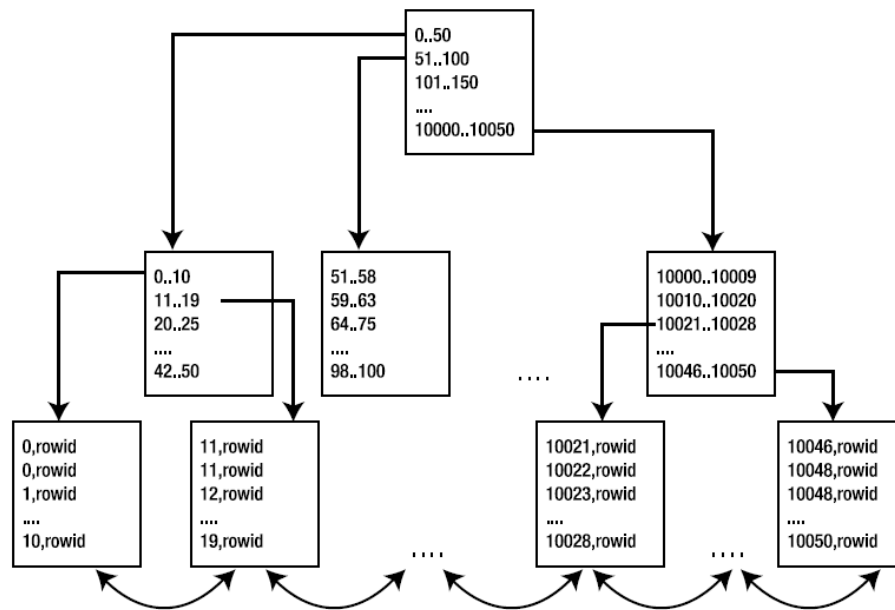
- 键值范围

Where  $45 < x < 123$

- 键前缀查找

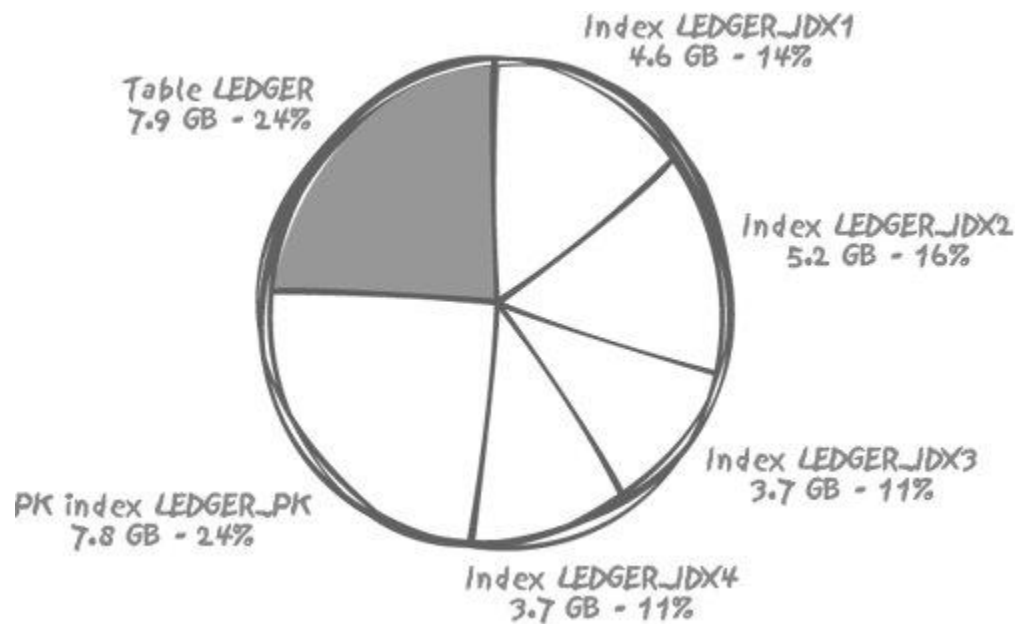
where  $x$  LIKE 'J%'

- 根据结构，请思考B+树索引不能做的有哪些？



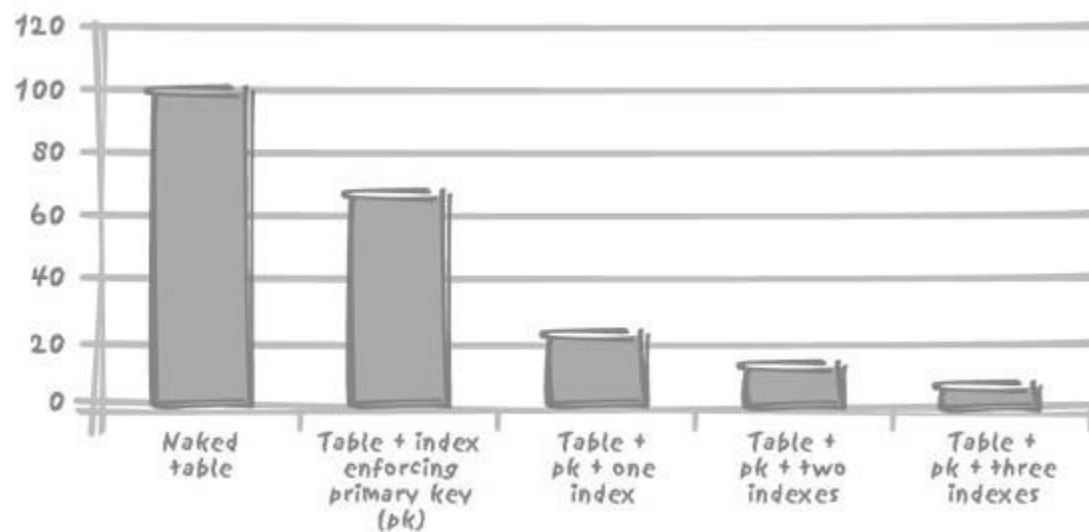
# 索引的另一面 (问题)

- 磁盘空间的开销

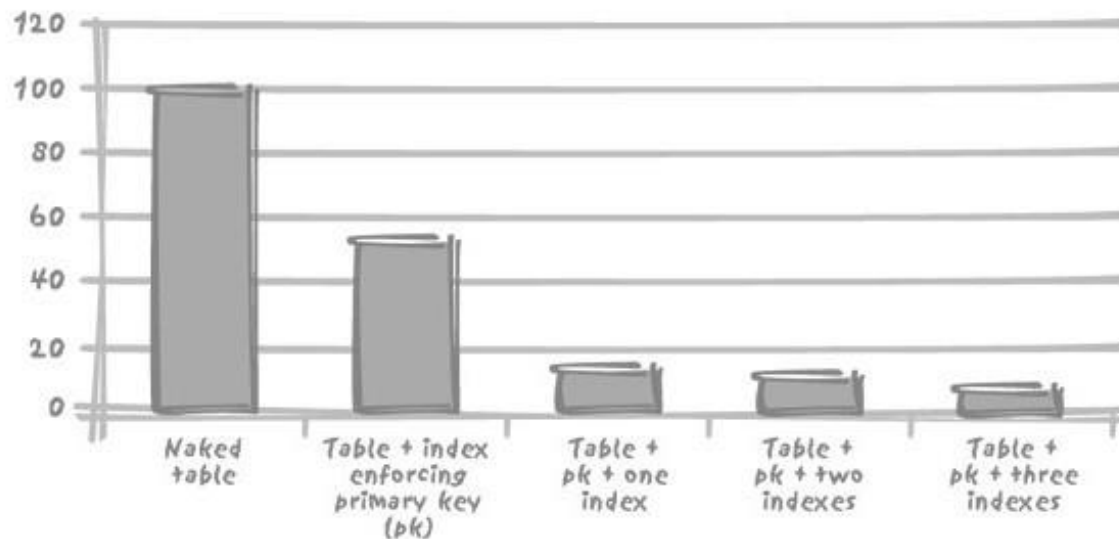


# 索引的另一面（问题）

- 磁盘空间的开销、处理的开销



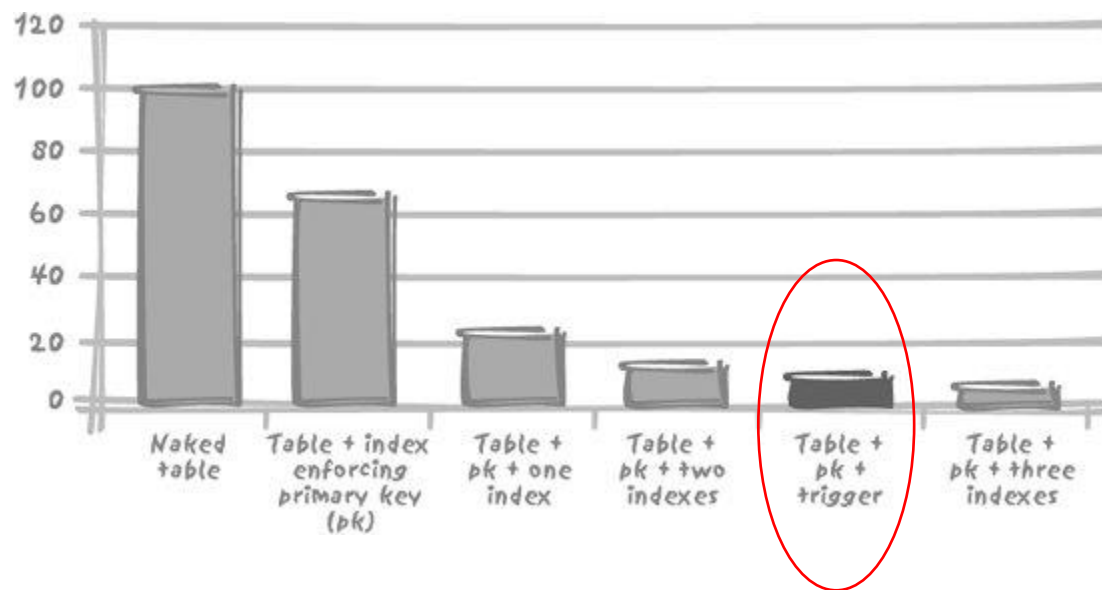
Oracle



MySQL

# 索引的另一面（问题）

- 数据库系统处理的开销



# Practice in class 3-1

- 那么不管怎么样，但，它至少能够提升查询效率不是吗？
- 对于B+树索引，不少数据库都有自己的处理方式，比如，MySQL中不同的存储引擎使用了不同的方式把索引保存到磁盘上，他们会影响性能。
  - MyISAM使用前缀压缩以减少索引，而InnoDB不会压缩索引，（有啥差别？）
  - MyISAM索引按照行存储的物理位置引用被索引的行，但是InnoDB按照主键值引用行，（有啥差别？）
- 请有兴趣的同学尝试去看一下你所用数据库索引的官方参考

# 索引有可能降低查询效率嘛？

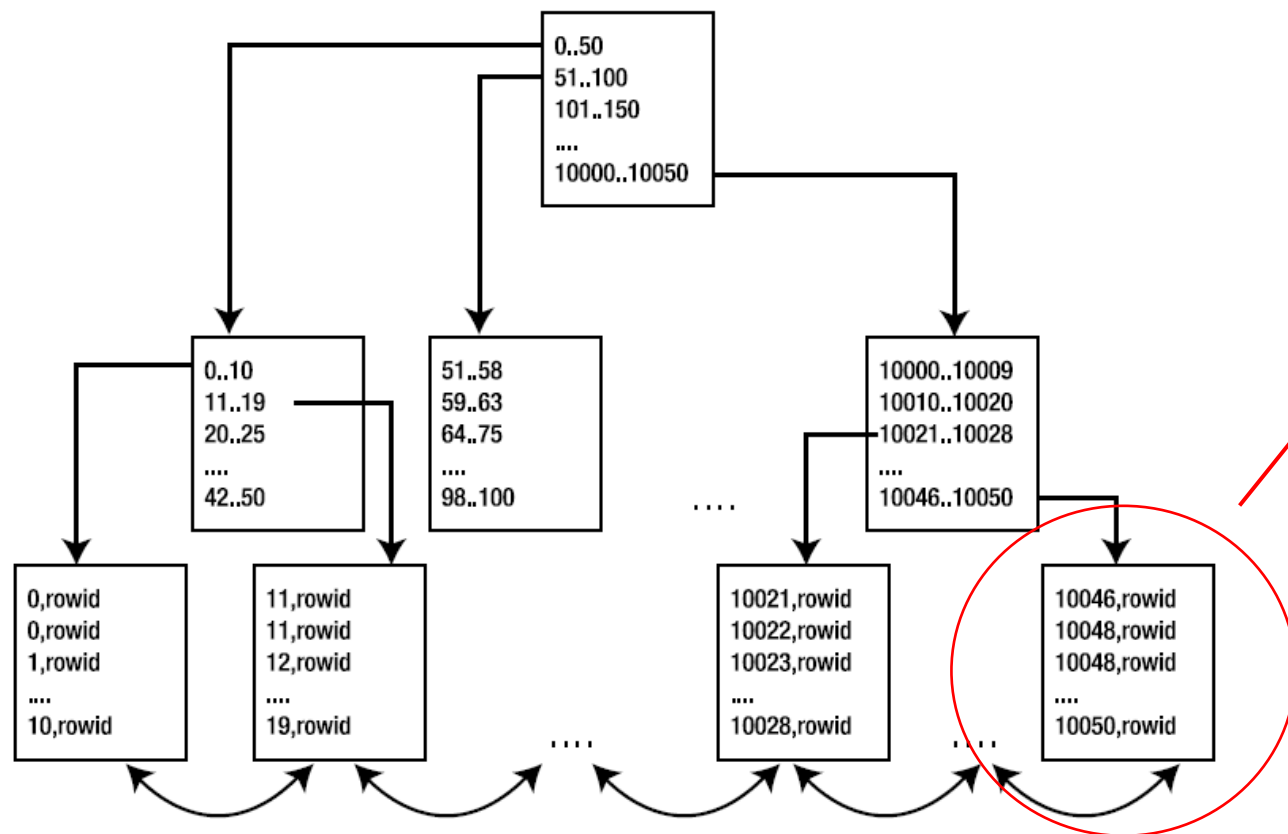
- 有可能.....
- 索引和目录
  - 索引和目录是两种完全不同的机制
  - 索引是一种以原子粒度访问数据库的手段而不是为了检索大量数据的



# 让索引发挥作用

- 索引的使用是否合理，首先取决于它是否有用
- 判断索引适用性的依据是检索比例（retrieval ratios）
- 什么时候应该使用B树索引
  - 仅需要通过索引访问基本表的很少一部分行
  - 如果要处理表中的多行，可以使用索引而不使用表

# 只使用索引，不使用表



如果是Index (x, y)

10046,2345,rowid  
10048,245 ,rowid  
10048,290 ,rowid  
10048,1356 ,rowed  
.....

那么

Select x,y

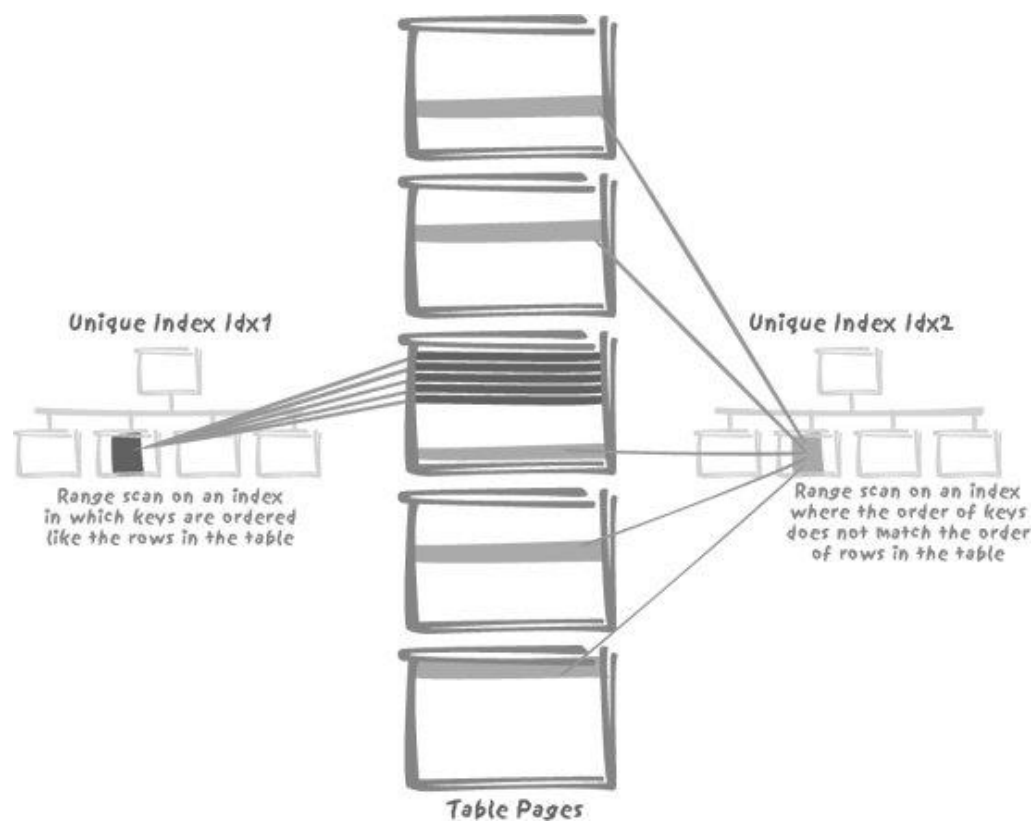
From T

Where x.....

就可以只使用索引不使用表

复合键索引，本质上索引是按照排名第一的字段进行的索引

# 让索引发挥作用 (cont')



- 索引只是查询工作的第一步
- 读取基本表中的数据才是查询的结束
- 同样的索引，但不同的物理结构，可能会引起查询效率的千差万别
  - 磁盘访问的速率
  - 物理I/O很可能是内存访问
  - 记录存储

# 索引无论怎样，都是数据库的重要组成部分

- 索引始终是数据库中极重要的组成部分
  - 通用目的或事务处理型数据库系统
  - 决策支持系统
- 事务处理型数据库中 “太多索引≈设计不够稳定”

# Practice in class 3-2

- 既然，使用复合键索引，在select子句中，如果所有字段都在复合键索引所包括的字段之中的时候，查询可以只使用索引不使用表
- 那么，为什么不可以针对表T (x,y,z) 这样的表，直接构建一个索引index (x,y,z) ，这样所有对这个表的访问就可以直接使用索引不使用表了，这会不会大幅度地提升查询效率呢？

# End

除了B树索引之外，数据库还有哪些索引呢？下一讲再见！