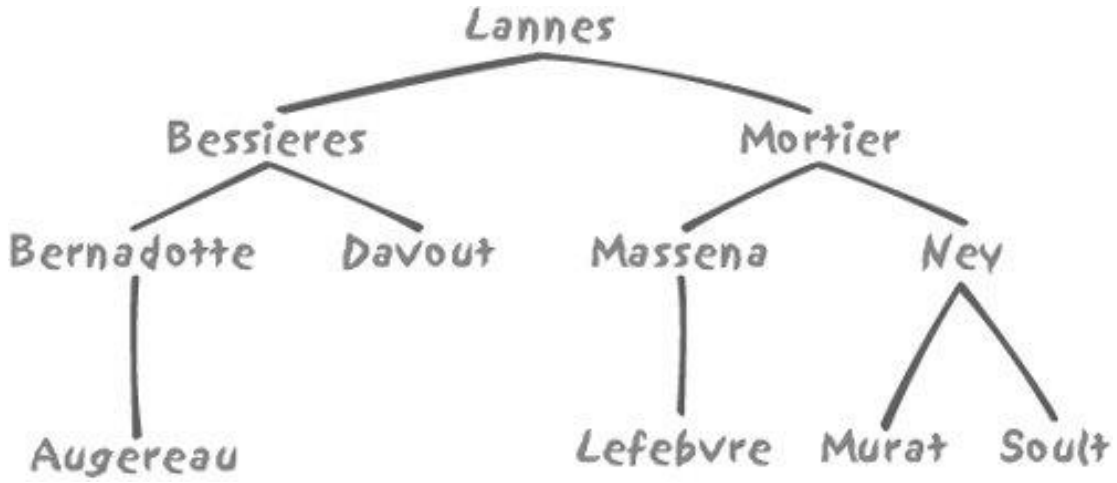
索引使用的典型问题

- 函数和类型转化对索引的影响
- 索引和外键
- 同一个字段，多个索引
- 系统生成键
- 总结，为什么没有使用我的索引?

函数和类型转换对索引的影响

- Where f (indexed_col) = ‘ some value‘
- 这种检索条件会使索引无法发挥作用
  - 日期函数
  - 隐式类型转换

字符串和日期的例子



where name = 'MASSENA'

where substr(name, 3, 1) = 'R'

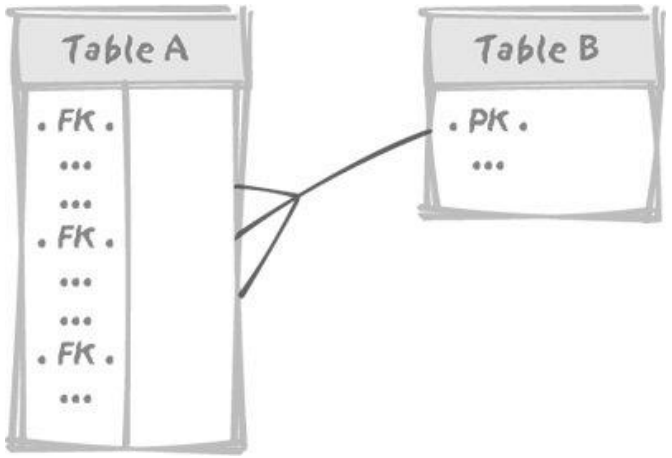
where date_entered = to_date('18-JUN-1815', 'DD-MON-YYYY')

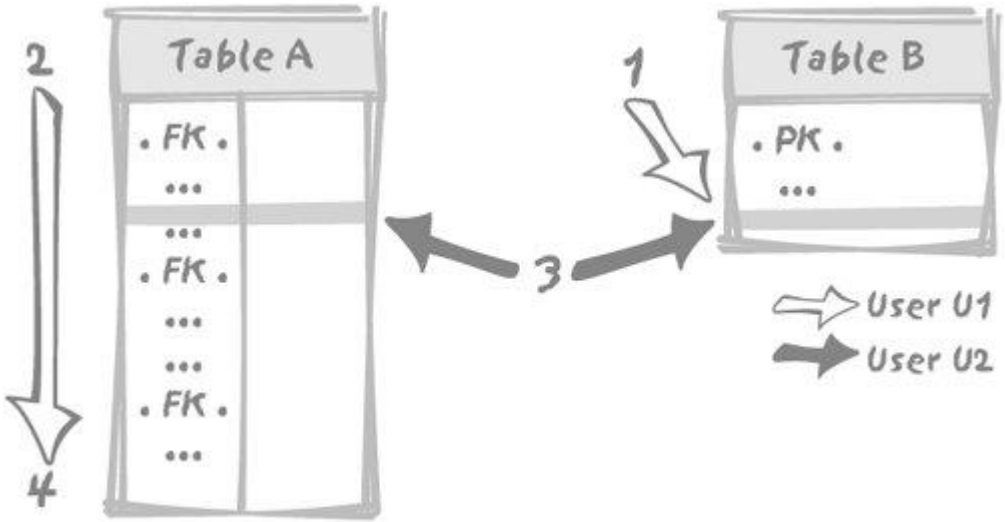
where trunc(date_entered) = to_date('18-JUN-1815', 'DD-MON-YYYY')

where date_entered >= to_date('18-JUN-1815', 'DD-MON-YYYY') and
date_entered < to_date('19-JUN-1815', 'DD-MON-YYYY')

索引与外键

- 系统地对表的外键加上索引的做法非常普遍
  - *但是为什么呢?*
  - *有例外吗?*
- 建立索引必须有理由
  - 无论是对外键，或是其他字段都是如此





同一字段，多个索引

- 如果系统为外键自动增加索引，常常会导致同一字段属于多个索引的情况

Order_Details (Order_id,article_id)

Order_Details (article_id,Order_id,)

当不需要为Article_id构建索引的时候，会引入额外索引

- 为每个外键建立索引，可能会导致多余索引

系统生成键

- 系统生产序列号，远好于
  - 寻找当前最大值并加1
  - 用一个专用表保存”下一个值“且加锁更新
- 但如果插入并发性过高，在主键索引的创建操作上会发生十分严重的资源竞争
- 解决方案
  - 反向键索引或叫逆向索引（reverse index）
  - 哈希索引（hash indexing）

为什么没有使用我的索引?

- 情况1：我们在使用B+树索引，而且谓词中没有使用索引的最前列
  - T，T(X,Y)上有索引，做SELECT * FROM T WHERE Y=5
- 跳跃式索引（仅CBO）

为什么没有使用我的索引？（cont’）

- 情况2：使用SELECT COUNT(*) FROM T，而且T上有索引，但是优化器仍然全表扫描
- 情况3：对于一个有索引的列作出函数查询
  - Select * from t where f(indexed_col) = value
- 情况4：隐形函数查询

为什么没有使用我的索引？（cont’）

- 情况5：此时如果用了索引，实际反而会更慢
- 情况6：没有正确的统计信息，造成CBO无法做出正确的选择
- 总结：归根到底，不使用索引的通常愿意就是“不能使用索引，使用索引会返回不正确的结果”，或者“不该使用索引，如果使用了索引就会变得更慢”

总结：索引访问的不同特点

- “查询使用了索引就万事大吉了”—误解啊～～
- 索引只是访问数据的一种方式
- “通过索引定位记录”只是查询工作的一部分
- 优化器有更多的选择权利
- 总结：索引不是万灵药。充分理解要处理的数据，做出合理的判断，才能获得高效方案

思考题

- 请研究你手上使用的数据库，比如，MySQL or Oracle，请研究数据库管理系统提供的其它索引形式，并阅读相关的文档

- **欢迎将你的成果用最简单且达意的方式在视频下留言**

End

第一部分，索引的优化结束了

让我们等待第二部分 SQL的优化