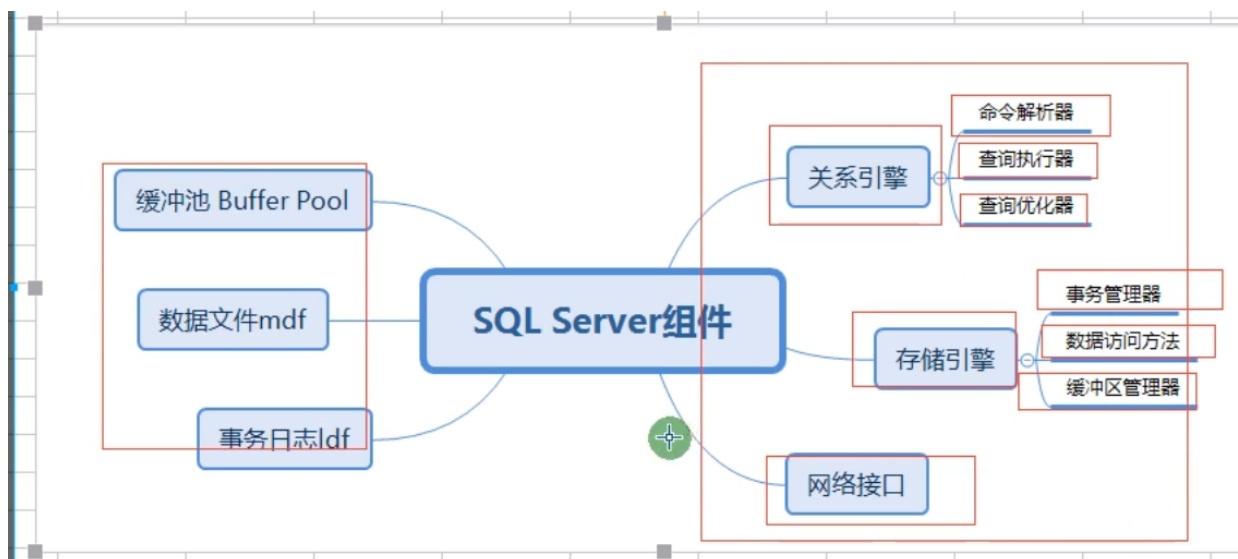


## 1.数据库是什么



## 2.数据库存储

Page 1 8kb

8个Page组成Extent,所有的Extent组成Heap

IAM Page: 管理数据页，记录数据页在硬盘的具体位置

文本/图现象页：从Page中指向过来的

Index Page：索引页：记录数据和位置

## 3.索引

### 聚集索引

定义：数据行的物理顺序与列值（一般是主键的那一列）的逻辑顺序相同，一个表中只能拥有一个聚集索引。

### 非聚集索引

定义：就是数据行的物理顺序和列值的逻辑顺序不同，就像是一个指针指向对应的真实数据，非聚集索引是可以有多个的。

非聚集索引最多的问题是二次查找的问题，要解决二次找到的问题最好是用复合索引，把需要查找的数据全部都写道索引中来。

聚集索引最大的问题是修改，因为其本身的有序性，所以修改起来成本比较高。

聚集索引和非聚集索引（辅助索引）的区别：

1. 聚集索引和非聚集索引的根本区别是表记录的排列顺序与索引的排列顺序是否一致。

2. 聚集索引表记录的排列顺序和索引的排列顺序一致，所以查询效率高，只要找到第一个索引值记录，其余就连续性的记录在物理也一样连续存放。聚集索引对应的缺点就是修改慢，因为为了保证表中记录的物理和索引顺序一致，在记录插入的时候，会对数据页重新排序。

3. 非聚集索引制定了表中记录的逻辑顺序，但是记录的物理和索引不一定一致，两种索引都采用B+树结构，非聚集索引的叶子层并不和实际数据页相重叠，而采用叶子层包含一个指向表中的记录在数据页中的指针方式。非聚集索引层次多，不会造成数据重排。

数据在磁盘中排列的顺序和索引的顺序是否一致是区分，聚集索引和非聚集索引最大的区别。因为是有顺序的所以能够快速通过有序去查找到对应的所有记录

## 4.索引失效

### B-Tree索引有效和失效的情况

翻译 星坠竹空 2018-12-04 17:09:12 199 收藏  
分类专栏: mysql

以索引key(last\_name, first\_name, dob)为例

**索引有效的情况:**

1.全位匹配:

全值匹配指的是和索引中的所有列进行匹配，例如前面提到的索引可用于查找姓名为CubaAllen、出生于1960-01-01 的人。

2.匹配最左前缀

前面提到的索引可用于查找所有姓为Allen 的人，即只使用索引的第一列。

3.匹配到前缀

也可以只匹配某一列的值的开头部分。例如前面提到的索引可用于查找所有以J开头的姓的人。这里也只使用了索引的第一列。

4.匹配范围值

例如前面提到的索引可用于查找姓在Allen 和B arrymore之间的人。这里也只使用了索引的第一列。

5.精确匹配第一列并范围匹配另外一列

前面提到的索引也可用于查找所有姓为Allen，并且名字是字母K开头（比如Kim、Karl等）的人。即第一列last\_name也全匹配，第二列first\_name范围匹配。

6.只访问索引的查询

B-Tree通常可以支持“只访问索引的查询”，即查询只需要访问索引，而无须访问数据行。后面我们将单独讨论这种“覆盖索引”的优化。

**索引失效的情况:**

1.如果不是按照索引的最左列开始查找，则无站使用索引。例如上面例子中的索引无法用于查找名字为B ill的人，也无法查找某个特定生日的人，因为这两列都不是最

左数据列。类似地，也无站查找姓氏以某个字母结尾的人。

2.不能跳过索引中的列。也就是说，前面所述的索引无法用于查找姓为Smith 并且在某个特定日期出生的人。如果不指定名（first\_name），则MySQL 只能使用索引的第一列。

只用到第一列:

```
EXPLAIN select * from applicant_info where applicant_info.caseCategoryCode = 'SZ201809290001' and idCardNo = '362226198908241213';
```

其实我给人的理解，所谓的索引失效即是索引的B+树，不能很好的去匹配对于的数据了。只能通过扫描的方式一个一个节点的去比较。