1. 索引
2. 概述

如果没有索引，我们在搜索记录时直接通过遍历所有的页，再遍历所有的记录来命中最终的记录，这样效率非常低。因此，我们需要使用索引提高查找的效率。

1. InnoDB中的索引方案
2. 索引规则

由于各个页中的记录的存储没有规律，因此查找起来没有头绪，不知道从哪里入手。我们要找到某个记录，就必须首先找到该记录在哪个页中，通过什么条件来定位该记录呢？主键是一个很好的选择，仿效页中页目录的做法，设定下一个索引页中用户记录的主键值必须大于上一个页中用户记录的主键值，这样就能够通过主键来确定每个页的范围了。要维持这种状态，需要在页中记录进行增删改的过程中，通过一些记录移动的操作保持不同页的顺序排序。然后，给所有的页也建立一个目录，每个页对应一个目录项，每个目录项中包括两个部分：页的用户记录中最小的主键值key和页号page\_no。

那么，我们在查找记录时直接通过记录的主键值确定该记录所在的页，进而进入该页中通过页目录来查找该记录所在的槽，然后遍历槽中的记录。这样，查找的效率会高很多。

1. InnoDB中的索引方案

InnoDB使用页作为管理存储空间的基本单位，但是最多保证16KB的连续存储空间，如果表中页的数量特别多，总会有超出16KB的那一天。且如果删除了一个页，要删除目录项中该页对应的项，或者作为冗余存放在目录项中，浪费存储空间。这些问题都需要解决。

InnoDB通过复用存储用户记录的索引页来存储目录项，为了区分，这些用来表示目录项的记录称为目录项记录，他们的记录头信息中的record\_type属性为1，普通用户记录为0。且目录项记录只有主键值和页的编号两个列。

那么问题又来了，我们如何定位这些存储目录项纪录的页呢？

这些页在存储空间中也可能不是挨着的，可能会产生很多存储目录项记录的页，如何根据主键值快速定位一个存储目录项记录的页呢？答案是，为这些存储目录项记录的页再生成一个更高级的目录，像多级目录一样。这种结构就像一棵树，名为B+树，用户记录都存放在叶子节点中，非叶子节点都是目录项记录。第0层是叶子节点所在的层，一般情况下B+树都不会超过4层。

1. 聚簇索引

聚簇索引是满足以下条件的B+树：

1. 页内的记录按照主键的大小顺序排成一个单向链表，记录被划分为若干个组，每个组中主键值最大的记录在页内的偏移量会作为槽依次存放在页目录中。
2. 各个存放用户目录的页根据主键大小顺序排成双向链表
3. 存放目录项纪录的页分为不同的层级，在同一层级中的页也是根据页中目录项记录的主键大小顺序排成双向链表
4. B+树的叶子节点存储的是完整的用户记录，即包括隐藏列。

聚簇索引不需要我们在MySQL语句中显示地使用INDEX语句来创建，InnoDB中，就是这种存储方式。

1. 二级索引

聚簇索引只有在搜索条件是主键时才能够发挥作用，对于其他的列，如何是好呢？

我们可以多建几棵B+树，不同的B+中的数据采用不同的排序规则。这种B+树与聚簇索引的区别是什么呢？

1. 叶子节点存储的不是完整的用户记录，只是索引列+主键，当我们找到对应的索引记录时，通过主键到聚簇索引中查找完整的用户记录，这个操作称为回表。然后再返回到这棵B+树的叶子节点处，沿着单向链表将继续搜索。这种方案的优点是节省空间，这也是称为二级索引的原因，必须执行进行回表的二次操作。
2. 目录项记录是索引列+主键+页号，如果不添加主键，那么对于该索引列中有多个相同的值的情况，且目录项中不同的页的索引列都是相同的，在页分裂时就无法判断先插入的记录应该放入哪一个页中的困境，因此需要添加主键来保证查找的唯一性。

当我们为某个列或列组合声明UNIQUE时，便会为这个列或者列组合建立二级索引。但是即使有UNIQUE属性加持，也可能有多条记录键值相同的情况，如都是NULL值，或者MVCC服务。

1. 联合索引

可以同时以多个列的大小作为排序规则，同时为多个列建立索引，如让B+树按照c2和c3列的大小进行排序，则意思是：

1. 先把各个记录和页按照c2列排序
2. 在记录的c2列相同的情况下，再采用c3列进行排序

每条目录项都由c2列, c3列，页号三部分组成，用户记录有c2列，c3列和主键组成。

1. 注意事项
2. 根页面不会移动

为某个表创建一个B+树索引，都会为这个索引创建一个根节点页面，初始时表中没有数据，则对应的根页面中没有用户记录和目录项记录。随后插入用户记录，在根节点中的可用空间用完时，会将根节点中的所有记录复制到一个新分配的页中，对这个新页进行页分裂操作，得到另一个新的页，根节点此时便升级为存储目录项记录的页，也就需要把用户记录的目录项记录插入到根节点中。

B+ 树索引的根节点始终不会移动，即页号不会改变，会记录到某个地方，凡是InnoDB需要使用这个索引，从固定的地方找出根节点的页号，访问这个索引。

1. 一个页面至少两条记录

如果一个大的目录只存放一个子目录，那么目录层级会非常地多，因此，InnoDB规定一个页面最少两条记录。

1. MyISAM中的索引方案

MyISAM将索引和数据分开存储，表中的记录按照记录的插入顺序（非主键排序，因此无法二分）单独存储在一个文件中，称为数据文件，不划分数据页，直接通过行号查询记录。

索引信息单独存储到索引文件中，为表的主键单独创建一个索引，在索引的叶子节点中存储的不是完整的用户记录，而是主键+行号，也就是先通过索引找到对应的行号，再通过行号来查找用户记录。因此MyISAM中的索引都是二级索引。

对于MyISAM记录的行格式，有定长记录格式，变长记录格式，压缩记录格式。定长记录格式可以通过行号计算出在某条记录在数据文件中的地址偏移量，但是变长记录格式不行，因此需要在索引叶子节点处存储该条记录在数据文件中的地址偏移量。这一点，MyISAM的效率更高，因为它直接拿到地址偏移量到文件中取记录，而InnoDB是通过主键到聚簇索引中找记录。

1. MySQL中创建和删除索引

InnoDB和MyISAM会自动会主键或者UNIQUE的列建立索引，其他的列需要显示指定。

1. 创建索引

CREATE TABLE 表名(

(KEY | INDEX) 索引名 单个列或多个列

);

ALTER TABLE 表名 ADD (KEY | INDEX) 索引名 单个列或多个列;

1. 删除索引

ALTER TABLE 表名 DROP (KEY | INDEX) 索引名;

1. 联合索引

CREATE TABLE 表名(

(KEY | INDEX) 索引名 (多个列)

);

联合索引的索引名尽量以idx\_为前缀，后面是多个列的名称。