应用系统体系架构 — 作业6

# 学号： 522031910213 姓名： 朱涵 得分：

A.在聚簇索引上执行查询，尤其是范围查询时，它的执行效率会非常高。请问，为什么聚簇索引的效率高？

答：

聚簇索引是指直接建立在数据表的索引 其叶子节点的值就是数据行 键一般默认是主键 而非聚簇索引是指在聚簇索引的基础上建立的二次索引 其叶子节点的值是主键值 键则是某几个字段的数据

在查询聚簇索引时 根据主键直接定位到了数据表的记录 而查询非聚簇索引则需要先根据字段数据定位到主键 然后在根据主键在聚簇索引里查询到数据记录

可以看到查询聚簇索引是直接查询到磁盘中的数据记录 然后可以通过叶子节点的指针直接顺序访问这些数据 磁盘的顺序访问是比较快的 因此范围查询的效率很高

相比之下非聚簇索引需要二次查询 而且查询到范围内的数据在聚簇索引中通常需要回表进行随机访问 就会慢很多

1. 如果要将你的图书封面以Base64的形式存储在MySQL中，那么你在数据表的设计中，是以VARCHAR类型来存储，还是以LONGBLOB形式存储？请详细解释你的理由。

答：

LONGBLOB类型。

Base64编码将二进制数据编码为文本形式 每3个字节转换为4个字符 因此编码后的数据比原始二进制数据要大约33% 图片本身数据大小就比较大 转换为Base64形式的文本后只会更大

VARCHAR类型是直接存储在数据库中的 而LONGBLOB类型则是存储在其他位置 数据库中存放指针 我们知道一个数据行的大小如果超过页大小的一半 查询时效率就会非常低 为此InnoDB也设置了限制 一行数据声明不能超过64KB 而实际存储时不能超过页的一半（默认是8kb）若使用VARCHAR存储base64文本串 很有可能出现超出限制的情况 就算不超出限制 也会因为数据行太大而影响查询性能 BLOB类型的数据则不会有这种问题 查询时仅包含指向实际数据的指针 避免了字符编码的额外开销

另外 从语义上来说 VARCHAR是为了存储字符串文本类型的 而BLOB用于存储图像等二进制数据 使用LONGBLOB在语义上也

1. 当你的图书表上建立复合索引以加速涉及多个字段的查询时，请用SQL语句给出你建立该复合索引的方式，并详细解释你为什么构建这样的索引，包括字段顺序、升降序排序等因素是如何确定的。

答：

以author列和price列为例建立符合索引 SQL语句应该为**“CREATE INDEX idx\_author\_price ON books (author ASC, publish\_year ASC);”**

首先是字段顺序 由于书店中经常需要查询某一个作者的所有书 但是不会查询某一个价格的所有书籍 因此我们需要的查询是类似于where author=A AND price<B的查询 根据最左匹配原则 如果price在前 这个索引将无法匹配 因此等值查询的author在前

然后是升降序 对于author字段 如果在展示所有书籍时需要按照author的字典序升序展示 可以采用升序来优化这一排序过程（只需要遍历这个复合索引即可得到按author升序的数据）；对于price字段 通常用户希望查询结果中价格越低的排在越前面 因此可以考虑升序 在返回where author=A AND price<B的查询结果时按照索引中访问的顺序即可返回price升序的数据

如果有其他需求（比如查询某一价格范围内的书籍） 应该针对不同场景的查询建立其他索引（比如建立价格的单字段索引）

1. 你认为你的订单表的主键使用自增主键还是UUID好？为什么？

答： **自增主键**

自增主键的优点是占用空间较小 在单一数据库中插入数据行更快（只需要+1） 建立聚簇索引也比较高效 而缺点是分布式数据库不适用或是需要额外的开销（比如锁）来保证主键的唯一性和一致性

UUID的优点是分布式数据库中不需要额外开销来保证主键的唯一性 可以直接生成 缺点是占用空间大（通常是128位）在单一数据库中插入数据行需要更多的计算开销 且建立聚簇索引比较繁琐

我的项目目前是单机部署 没有分表分库 适合简单的自增主键 性能更好 如果未来需要扩展成分布式数据库 可以考虑使用UUID来保证主键唯一性

E.请你搜索参考文献，总结InnoDB和MyISAM两种存储引擎的主要差异。

## 答：

## 1. 存储结构：

## MyISAM的数据表在磁盘上存储为三个文件，.frm文件存储表结构定义，.MYD文件存储数据文件，.MYI存储索引文件；而InnoDB的数据表在磁盘上存储为两个文件，.frm文件存储表结构定义，.ibd存储数据和索引。

## 也就是说MyISAM的数据表中索引和数据是分开存储的 ，其主键索引中的叶子节点存储的是数据的地址 ，而InnoDB的聚簇索引则会直接存储数据内容。同样的原因，InnoDB必须有一个唯一索引字段，如果用户不声明则会自动生成隐藏的row\_id列，而MyISAM则不需要。

## 2. 存储空间：

## MyISAM可被压缩，存储空间较小。MyISAM支持三种不同的存储格式：静态表（默认）、 动态表、压缩表；而InnoDB需要更多的内存和存储，它会在主内存中建立缓冲池用于高速缓冲数据和索引。

## 3. 事务：

## InnoDB支持事务，MyISAM不支持。其中InnoDB每一条SQL语言都会默认封装成事务，自动提交， 最好的选择是手动使用begin与commit指令来开启与提交事务。

## 4. 外键：

## InnoDB支持外键，而MyISAM不支持。

## 5. 锁机制：

## InnoDB支持表、行级锁，而MyISAM支持表级锁。对于大量删除插入操作，MyISAM由于整表上锁会更慢

## 上述解答参考了：

## <https://www.php.cn/faq/497537.html>

## <https://blog.csdn.net/weixin_40954799/article/details/140950714>