**数据类型优化：**

* 1. **可以正确存储数据的最小数据类型**（占用存储空间小）
  2. **列值尽量避免null**

当可为NULL的列被索引时，每个索引记录需要一个额外的字节

* 1. **尽量使用相同的数据类型存储相似或相关的值**：尤其是关联条件中的列
  2. **避免使用MYSQL已经遗弃的特性**：例如指定浮点数的精度，或者整数显示宽度

**整数类型**

**Int(11)11代表的并不是长度，而是字符的显示宽度**

**在字段类型为int时,无论你显示宽度设置为多少，int类型范围是固定的**

有符号和无符号类型使用相同的存储空间，并具有相同的性能

**TINYINT、SMALLINT、MEDIUMINT、INT、BIGINT，**

**分别对应的存储空间是8、16、24、32、64位**

**同时还有可选的UNSIGNED属性，可以使正数的上限提高一倍**

**实数类型**

**DECIMAL（小数）只是一种存储格式，在计算中DECIMAL会转换为DOUBLE类型**

**字符串类型**

**VARCHAR：可变长字符串，需要使用1或2个额外字节记录字符串的长度**

**如果列的最大长度小于等于255字节,只是用1个字节表示,否则使用2个字节**

**适合使用VARCHAR的场景：**

**字符串列的最大长度比平均长度大很多；列的更新少；**

**使用UTF8这样每个字符使用不同字节数存储的字符集**

**CHAR: 适合存储很短的字符串、且对于经常变更的数据**

**char比varchar好（涉及碎片问题）。例如非常适合存储密码的MD5值**

BINARY和VARBINARY

存储的是二进制字符串，存储的是字节码而不是字符，填充采用\0而不是空格，检索时也不会去掉填充值

BLOB和TEXT

前者采用二进制，后者采用字符的方式存储数据。

**日期和时间类型**

DATETIME

存储从1001到9999年的日期数据，**精确到秒，使用8个字节的存储空间。**

TIMESTAMP

存储从1970到2038年的数据，**精确到秒，使用4个字节的存储空间。**

**尽量使用TIMESTAMP，因为它比DATETIME空间效率更高**

**如果需要存储比秒更小粒度的日期和时间值，可以使用BIGINT类型存储**，或者DOUBLE存储秒之后的小数部分

**查询优化：**

(关键在于避免全表扫描，尽量使用索引)

1. ：在where和order by涉及的列上建立索引
2. ：在where子句中

(对字段进行null值判断)

(使用!=或<>操作符)

(使用or来连接条件，改为union)

(使用in和not in)

(连续的数最好用between别用in)

(字段进行表达式操作 where num/2 = 100)改为 where num = 100\*2

(使用参数 num = @num)会导致引擎放弃使用索引而进行全表扫描

1. ：用exists代替in