**SQL优化（系列）- 建表优化**

**一、建表优化**

    以MySQL为例，建表优化是非常重要的，可以*有效提升数据库的性能和效率*。以下是一些建表优化的常见技巧和最佳实践：

**1.1、使用合适的数据类型：**

    选择合适的数据类型可以*节省存储空间并提高查询性能*。例如，只存储正整数时可以使用 INT 而不是 VARCHAR。

**1.2、设定合适的字段长度：**

    根据实际需求设定字段的长度，*不要设置过长，以节省存储空间*，并避免不必要的数据截断。

**1.3、为重要字段增加索引：**

    根据查询的需求为频繁进行查询的字段*建立索引，可以加快检索速度*。但是*不要为所有字段都建立索引，因为索引会增加写操作的开销*。

**1.4、考虑使用分区表：**

    如果表中的数据量很大，可以*考虑使用分区表来提高查询性能*，可以根据时间范围或其他标准对表进行分区。

**1.5、使用合适的存储引擎：**

    根据实际需求选择合适的存储引擎，例如*InnoDB 适合事务处理和频繁更新的场景，而 MyISAM 更适合读取和查询频繁的场景。*

**1.6、避免过度规范化：**

    合理设计表结构，*避免过度规范化，可以减少表连接的次数，提高查询效率。*

**1.7、设定适当的默认值：**

*为表中的字段设定适当的默认值，可以简化插入操作，并避免空值的情况。*

**1.8、定期清理无用数据：**

*定期清理表中过期或无用的数据，以提高数据库的性能。*

    以上是一些建表优化的常见技巧，我们可以根据实际情况灵活运用，以*提高 MySQL 数据库的性能和效率。*

**二、使用总结**

1、*在表中建立索引，优先考虑where、order by使用到的字段。*

2、*尽量使用数字型字段（如性别，男：1 女：2），若只含数值信息的字段尽量不要设计为字符型，这会降低查询和连接的性能，并会增加存储开销*。

这是因为引擎在处理查询和连接时会 *逐个比较字符串中每一个字符*，而对于数字型而言只需要比较一次就够了。

3、*查询数据量大的表 会造成查询缓慢。主要的原因是扫描行数过多。这个时候可以通过程序，分段分页进行查询，循环遍历，将结果合并处理进行展示。要查询100000到100050的数据，如下：*

SELECT \* FROM (SELECT ROW\_NUMBER() OVER(ORDER BY ID ASC) AS rowid,\* FROM infoTab)t WHERE t.rowid > 100000 AND t.rowid <= 100050

4、用varchar/nvarchar 代替 char/nchar

*尽可能的使用 varchar/nvarchar 代替 char/nchar ，因为首先变长字段存储空间小，可以节省存储空间，其次对于查询来说，在一个相对较小的字段内搜索效率显然要高些。*

*不要以为 NULL 不需要空间，比如：char(100) 型，在字段建立时，空间就固定了， 不管是否插入值（NULL也包含在内），都是占用 100个字符的空间的，如果是varchar这样的变长字段， null 不占用空间。*