**避免全表扫描**

当我们的where过滤的字段没有索引时，就会发生全表扫描，如下，当我们的Name字段没有索引时，就会发生全部扫描

select \*

from dbo.Pages

where Name = 'Home'

当字段拥有索引时，我们对字段使用函数或表达式也会发生全表扫描

-- 这里会发生全表扫描

select \*

from dbo.Pages

where SUBSTRING(Name, 1, 2) = 'Ho'

-- 改进：使用like

select \*

from dbo.Pages

where Name like 'Ho%'

-- 这里会发生全表扫描

select \*

from dbo.Pages

where PostNum/2 = 10

-- 改进

select \*

from dbo.Pages

where PostNum = 10\*2

**少用负向查询**

负向查询会引发全表扫描，如下是负向查询符号

Not, !=, <>, !<, Not Exists, Not In

非负向查询符号

=, <, >, >=, <=, Between, Link（同配符在后面是是非负，如’陈%’）

我们应该可以使用or和and将非负向查询符号替换负向查询符号

-- 这里会发生全表扫描

select \*

from dbo.PageDatas

where Click != 0

-- 改进：使用like

select \*

from dbo.PageDatas

where Click < 0 or Click > 0

**And和Or**

And连接的多个字段，如果有一个字段有索引，那么数据库会优先通过该字段筛选出数据，其他字段则在该筛选的基础上筛选，所以And连接的字段中只要有一个字段有索引，就可以大幅度提供搜索速度

OR不一样，OR连接的多个字段必须每个字段都有索引，否则会引发全表扫描

**Where的查询顺序**

当我们使用And或Or连接多个条件时，sqlserver得where会先查询最右边的条件，如果最右边的条件能筛选出一个较小的结果集，那么在执行接下来的条件时会遍历的很少

-- 几乎所有人的 age 都大于

-- 第一次筛选对整个表进行筛选，结果集也差不多是整个表

-- 第二次也是对整个表进行选项

select \*

from dbo.Person

where Name = 'Cql' and Age > 2

-- 改进

select \*

from dbo.Person

where Age > 2 and Name = 'Cql'

注：现在的数据库都有查询优化器，可以优化where的顺序，不改进问题也不大

**内嵌视图取代临时表**

-- 如下 t 就是内嵌视图

select \*

from (

    select top 5 \*

    from dbo.Pages

) as t

**使用存储过程代替sql语句**

服务器想要数据库执行sql语句，会发送sql语句给数据库，如果sql语句很大，应该考虑使用存储过程代替sql语句

**分表**

对于具有索引的查询，分表的性能优化不大，例如从200万的行数据查找某一行，通过主键最多执行21个步骤，而通过没有索引的字段查找则需200万次，如果将该表分为2个表，则通过主键最多执行20个步骤，而通过没有索引的字段查找则需100万步骤

虽然如此，但我们还是说一下得说一下分表

基于面向对象得建模，如“人”表有3个人群，分别为“小孩”、“中年”、“老年”，这3个实体均继承“人”这个实体，我们可以使用EF得TPC模式将这3个实体得数据分别映射到3个表中

**分库**

分库时间上就是划分子系统，将一个大系统分为多个子系统，每个子系统都拥有自己得数据库