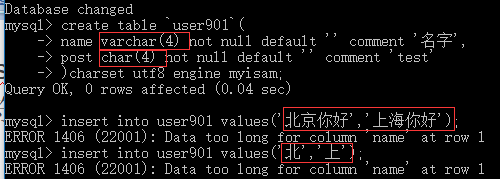
mysql中utf8编码中char与varchar插入中文字符越界报错问题

本人使用的mysql版本是5.7.20,使用平台是Windows10 +CMD

在创建一个表后,插入varchar与char数据时老是报错：

ERROR 1406 (22001): Data too long for column 'name' at row 1

如图：



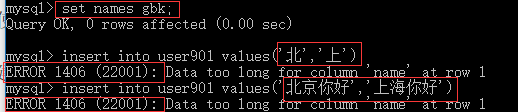
首先,在MySQL5.0以上的版本中,char(n)与varchar(n)中n都是表示字符的个数,通过insert插入的汉字应该只要不超过n的大小就时可行的,但是在实际操作过程中发现不管我插入n个数的汉字还是小于n个数汉字都是报错.

首先,主要的原因不是因为字符的长度,而是编码问题,我们都知道,windows上的CMD默认的编码是GBK：

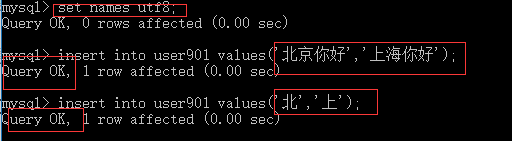


而我在创建表时设置的编码时utf8.

然后我想是不是可以用set names gbk;来解决一下编码的问题,但是操作过后发现不行：



那就换个思路,set names utf8;终于问题愉快的解决了：



对此,我特么专门,专门查了一下关于如何设置CMD编码的问题,用来防止下次出现类似的错误：

-----------转载如下-------------------------

网上有设置的方法，但说明确不够详细系统，说设置字体为：Lucida Console。问题是，在默认方式下，只有点阵字体，哪有什么Lucida Console。所以，在自己成功设置后，拿出来和大家分享下过程，下面就让我们找出Lucida Console来设置吧。

 　　这里需要先了解些内容：

**DHCP**

CHCP是MS DOS中的命令，用来显示或设置活动代码页编号的。用法是：

CHCP [nnn]

其中nnn指定的是代码页的编号。这个参数是可选的，在命令行下如果不指定这个代码页编号，那么默认是显示当前的代码页编号。比如，在默认的cmd窗口中，我们输入chcp，显示的将类似：

活动的代码页: 936

 这里的936表示当前使用的是简体中文（GB2312）编码。

**UTF8**

你也需要了解编码的一些知识，为了完成支持UTF8的工作，你至少需要知道UTF8代码页的编号：65001。更多关于编码的内容，这里不赘述，请自行查找相关内容。

有这两个知识点，接下来，让cmd支持UTF8就变得容易了。

1. 运行CMD；

2. 输入 CHCP，回车查看当前的编码；

3. 输入CHCP 65001，回车；

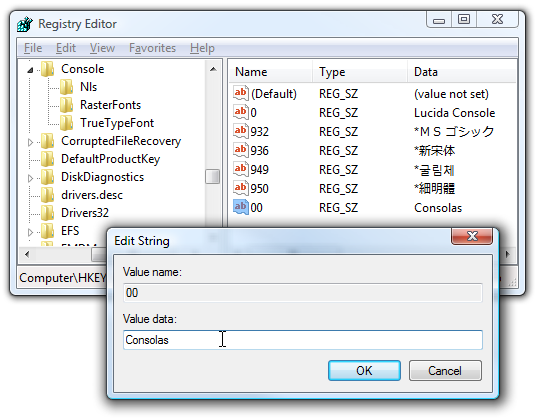
4. 仅如此，还是不能支持UTF8的正常显示，你还要在窗体上右键，选择属性，来设置字体；

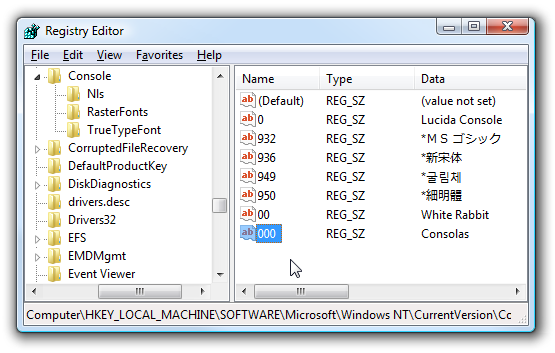
5. 操作完上面几步后，即使你原来的字体里面没有显示Lucida Console这个字体，现在应该也能看到了。选择它。如果原来就有，可以选上它先试试，不行在执行上述步骤（这里补充：至少我本机需要CHCP 65001下，有朋友说不要）；

6. 选择只应用到本窗体，确认。

这时候，你的Console里面，应该支持UTF8了。

有些朋友也想知道如何在CMD显示更多的字体，这个其实也是有办法的，只要在注册表（路径：HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Console\TrueTypeFont）中添加就好了。需要注意的是，添加的字体需要是定宽的(fixed-width)。操作示意图：





图片转载自：<https://www.howtogeek.com/howto/windows-vista/stupid-geek-tricks-enable-more-fonts-for-the-windows-command-prompt/>

文字部分转载自：

<http://blog.useasp.net/archive/2012/04/24/how_to_use_UTF8_encoding_in_Windows_CMD.aspx>

mysql中count()的使用细节以及distinct关键字的使用

---count()中可以写表达式,但是count()只会统计真和假,所以要加上判断条件,同时为了防止重复数据的干扰,可以使用count(distinct espr) 来去重

而且在处理一个大数据量数据库的时候mysql对于count(\*)的不同处理会造成不同的结果,比如执行:  
SELECT count(\*) from tablename  
即使对于千万级别的数据mysql也能非常迅速的返回结果  
而对于  
SELECT count(\*) FROM tablename WHERE.....  
mysql的查询时间开始攀升,这是因为当没有WHERE语句对于整个mysql的表进行count运算的时候,MyISAM类型的表中保存有总的行数,而当添加有WHERE限定语句的时候Mysql需要对整个表进行检索从而得出count的数值

用distinct的时候，如果它有索引，mysql会把它转成group by的方式执行。

mysql的distinct的关键字有很多用处  
1.不需要count()函数统计重复数据时可以使用:  
比如select count(distinct number) from table\_name;  
就是计算talbeb\_name表中number不同的记录有多少条

2,在需要返回记录不同的id的具体值的时候可以用  
比如select distinct id from table\_name;  
返回talbeb\_name表中不同的id的具体的值

3.下面的情况返回结果时可能会有歧义  
select distinct id, type from table\_name;  
实际上返回的是 id与type同时不相同的结果,也就是DISTINCT同时作用了两个字段，必须得id与tyoe都相同的才被排除了,与我们期望的结果不一样

可以利用group by ,concat()函数来进行排除,注意版本的支持mysql4.1以下不支持

select id, type from table\_name group by id;

**Mysql处理海量数据时的一些优化查询速度方法**

最近一段时间由于工作需要，开始关注针对Mysql数据库的select查询语句的相关优化方法。

由于在参与的实际项目中发现当mysql表的数据量达到百万级时，普通SQL查询效率呈直线下降，而且如果where中的查询条件较多时，其查询速度简直无法容忍。曾经测试对一个包含400多万条记录（有索引）的表执行一条条件查询，其查询时间竟然高达40几秒，相信这么高的查询延时，任何用户都会抓狂。因此如何提高sql语句查询效率，显得十分重要。以下是网上流传比较广泛的30种SQL查询语句优化方法：

1、应尽量避免在 where 子句中使用!=或<>操作符，否则将引擎放弃使用索引而进行全表扫描。

2、对查询进行优化，应尽量避免全表扫描，首先应考虑在 where 及 order by 涉及的列上建立索引。

3、应尽量避免在 where 子句中对字段进行 null 值判断，否则将导致引擎放弃使用索引而进行全表扫描，如：

select `id` from `test` where `num` is null

可以在num上设置默认值0，确保表中num列没有null值，然后这样查询：

select `id` from `test` where `num`=0

4、尽量避免在 where 子句中使用 or 来连接条件，否则将导致引擎放弃使用索引而进行全表扫描，如：

select id from t where num=10 or num=20

可以这样查询：

select id from t where num=10

union all

select id from t where num=20

5、下面的查询也将导致全表扫描：(不能前置百分号)

select id from t where name like '%abc%'

若要提高效率，可以考虑全文检索。

6、in 和 not in 也要慎用，否则会导致全表扫描，如：

select id from t where num in(1,2,3)

对于连续的数值，能用 between 就不要用 in 了：

select id from t where num in(1,2,3)

7、如果在 where 子句中使用参数，也会导致全表扫描。因为SQL只有在运行时才会解析局部变量，但优化程序不能将访问计划的选择推迟到运行时；它必须在编译时进行选择。然 而，如果在编译时建立访问计划，变量的值还是未知的，因而无法作为索引选择的输入项。如下面语句将进行全表扫描：

select `id` from `test` where `num`=@num

可以改为强制查询使用索引：

select `id` from `test` with(index(索引名)) where `num`=@num

8、应尽量避免在 where 子句中对字段进行表达式操作，这将导致引擎放弃使用索引而进行全表扫描。如：

select `id` from `test` where `num`/2=100

应改为:

select `id` from `test` where `num`=100\*2

9、应尽量避免在where子句中对字段进行函数操作，这将导致引擎放弃使用索引而进行全表扫描。如：

select `id` from `test` where substring(name,1,3)=’abc’–name以abc开头的id

select `id` from `test` where datediff(day,createdate,’2005-11-30′)=0–’2005-11-30′生成的id

应改为:

select id from t where name like ‘abc%’

select id from t where createdate >= ‘2005-11-30’ and createdate < ‘2005-12-1’;

10、不要在 where 子句中的“=”左边进行函数、算术运算或其他表达式运算，否则系统将可能无法正确使用索引。

11、在使用索引字段作为条件时，如果该索引是复合索引，那么必须使用到该索引中的第一个字段作为条件时才能保证系统使用该索引，否则该索引将不会被使用，并且应尽可能的让字段顺序与索引顺序相一致。

12、不要写一些没有意义的查询，如需要生成一个空表结构：

select col1,col2 into #t from t where 1=0

这类代码不会返回任何结果集，但是会消耗系统资源的，应改成这样：

create table #t(…)

13、**很多时候用 exists 代替 in 是一个好的选择**：

select num from a where num in(select num from b)

用下面的语句替换：

select num from a where exists(select 1 from b where num=a.num)

14、并不是所有索引对查询都有效，SQL是根据表中数据来进行查询优化的，当索引列有大量数据重复时，SQL查询可能不会去利用索引，如一表中有字段 sex，male、female几乎各一半，那么即使在sex上建了索引也对查询效率起不了作用。

15、索引并不是越多越好，索引固然可以提高相应的 select 的效率，但同时也降低了 insert 及 update 的效率，因为 insert 或 update 时有可能会重建索引，所以怎样建索引需要慎重考虑，视具体情况而定。一个表的索引数最好不要超过6个，若太多则应考虑一些不常使用到的列上建的索引是否有必要。

16.应尽可能的避免更新 clustered 索引数据列，因为 clustered 索引数据列的顺序就是表记录的物理存储顺序，一旦该列值改变将导致整个表记录的顺序的调整，会耗费相当大的资源。若应用系统需要频繁更新 clustered 索引数据列，那么需要考虑是否应将该索引建为 clustered 索引。

17、尽量使用数字型字段，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。

18、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。

19、**任何地方都不要使用 select \* from t ，用具体的字段列表代替“\*”，不要返回用不到的任何字段。**

20、尽量使用表变量来代替临时表。如果表变量包含大量数据，请注意索引非常有限（只有主键索引）。

21、避免频繁创建和删除临时表，以减少系统表资源的消耗。

22、临时表并不是不可使用，适当地使用它们可以使某些例程更有效，例如，当需要重复引用大型表或常用表中的某个数据集时。但是，对于一次性事件，最好使用导出表。

23、在新建临时表时，如果一次性插入数据量很大，那么可以使用 select into 代替 create table，避免造成大量 log ，以提高速度；如果数据量不大，为了缓和系统表的资源，应先create table，然后insert。

24、如果使用到了临时表，在存储过程的最后务必将所有的临时表显式删除，先 truncate table ，然后 drop table ，这样可以避免系统表的较长时间锁定。

25、尽量避免使用游标，因为游标的效率较差，如果游标操作的数据超过1万行，那么就应该考虑改写。

26、使用基于游标的方法或临时表方法之前，应先寻找基于集的解决方案来解决问题，基于集的方法通常更有效。

27、与临时表一样，游标并不是不可使用。对小型数据集使用 FAST\_FORWARD 游标通常要优于其他逐行处理方法，尤其是在必须引用几个表才能获得所需的数据时。在结果集中包括“合计”的例程通常要比使用游标执行的速度快。如果开发时 间允许，基于游标的方法和基于集的方法都可以尝试一下，看哪一种方法的效果更好。

28、尽量避免向客户端返回大数据量，若数据量过大，应该考虑相应需求是否合理。

29、尽量避免大事务操作，提高系统并发能力。

**创建用户时报错**

**ERROR 1290 (HY000): The MySQL server is running with the--skip-grant-tables option so it cannot execute thisstatement**

前段时间下载了最新的mysql版本5.7.20,由于5.7以上的版本都没有my.ini,就自己在网上搜了一个,在配置一切ok了打算创建一个用户给妹妹练练手,却发现老是报

The MySQL server is running with the--skip-grant-tables option so it cannot execute thisstatement,

看了一下网上的各种说法,最终发现特么问题就出在my.ini,在这个配置文件的最后特么多了一句

skip-grant-tables

坑爹,怒删之!!

随后在进行操作,顺利解决问题,不过特么我放在bin目录的貌似没有改.....

