安装配置教程：http://jingyan.baidu.com/article/642c9d34aa809a644a46f717.html

1.安装：

选择Typical典型安装、自定义安装路径

2.配置(步骤)：

Detailed Configuration详细配置

Developer machine 服务器类型，占用资源少

选择数据库用途：多功能/事务型/非事务型 一般选择多功能型

选择数据库文件存放的目录：InnoDB tablespace setting

选择数据库连接的个数 ： 20/500 甚至手工设置更多

Enable TCP/IP networking ： 启用tcp连接 ，设定端口，允许其他用户远程访问数据库；如果不启用，就只能在自己的机器上访问mysql 数据库了

设置编码：Best Support for multilingualism UTF-8编码

设置将MySQL安装路径配置到系统的path环境变量中

创建超级管理员用户，设置用户名和密码

配置完成

打开MySQL Command line Client ，输入管理员用户名和密码验证配置是否成功

## 数据库操作

MySQL在window平台下不区分大小写，但在Linux平台下区分大小写

 MySQL在Linux下数据库名、表名、列名、别名大小写规则是这样的：

　　    1、数据库名与表名是严格区分大小写的；

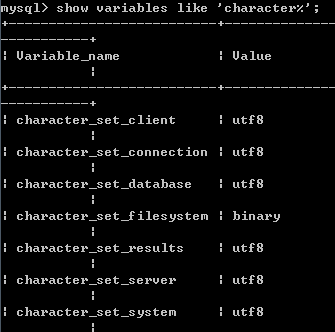
　　    2、表的别名是严格区分大小写的；

　　    3、列名与列的别名在所有的情况下均是忽略大小写的；

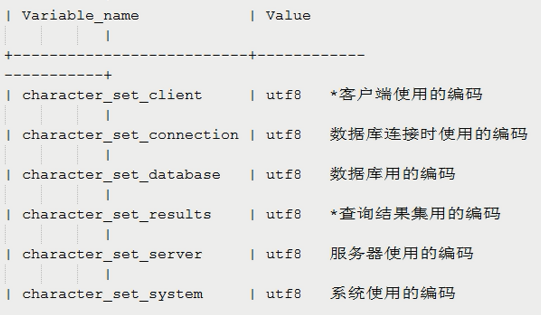
　　    4、变量名也是严格区分大小写的；    MySQL在Windows下都不区分大小写

查看系统各个部分的编码格式：

show variables like ‘ character%’ ;



含义：



1.创建数据库

create database 数据库名 [character set 字符集] [ collate 校验方式]

----创建一个名称为mydb1的数据库。

create database mydb1;

----创建一个使用utf8字符集的mydb2数据库。

create database mydb2 character set utf8;

----创建一个使用utf8字符集，并带校对规则的mydb3数据库。

create database mydb3 character set utf8 collate utf8\_bin ;

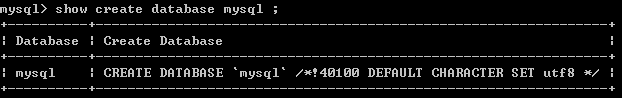
2.查看数据库

----查看当前数据库服务器中所有的数据库

show databases ;

----查看创建的数据库的类型

Show create database 数据库名 ;

 为数据库设置的字符编码

3.修改数据库的字符集（前提是被修改的数据库已经创建好了）

Alter database 数据库名character set utf8/其他的字符编码 ;

4.删除数据库

Drop database 数据库名 ；

5.查看当前正在使用的数据库

Select database();

6.切换到当前正在使用的数据库

Use 数据库名称 ；

## 数据库表操作

### 1.查看表

查看表结构：desc tabName ；

查看当前数据库中所有表：show tables ；

查看当前数据库表建表语句 show create table tabName;

### 2.创建表

**CREATE TABLE table\_name**

**(**

**field1 datatype,**

**field2 datatype,**

**field3 datatype,**

**)character set 字符集 collate 校对规则**

Character set 字符集 和 collate校对规则可以不写，则默认为该表所在的数据库设置的字符集和校对规则。

约束:

primary key

unique

not null

auto\_increment 主键字段必须是数字类型。

外键约束

### 3.修改表结构

ALTER TABLE table ADD/MODIFY/DROP/CHARACTER SET/CHANGE (column datatype [DEFAULT expr][, column datatype]...);

(1)修改表的名称：rename table 表名 to 新表名;

(2)增加一个字段（一列）

alter table 表名 add 字段名 字段值类型 ;

如增加一个二进制的图片字段：alter table 表名 add image blob;

(3)修改表中某个字段的值的类型

alter table 表名 modify 已有字段名 字段类型 ;

(4)修改表中某个字段的名称及值的类型

alter table 表名change 原字段名称 新字段名称 字段值类型;

(5)删除某一列

alter table 表名 drop 列名 ；

(6)修改表的字符集

alter table 表名 character set utf8/gbk ;

### 4.删除表

drop table 表名 ;

### 5.插入数据

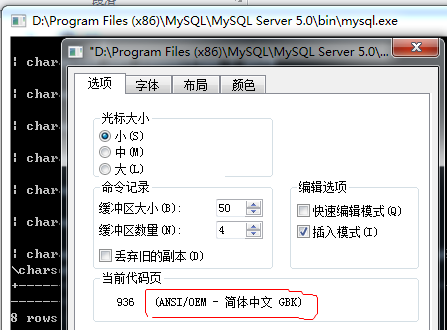
INSERT INTO 表名 [(列名1,列名2，列名3……)] VALUES ( 值1，值2，值3);

insert into user\_info value(1,'michael','beijing','15779683263');

插入中文字符出错：

错误一：原因：客户端使用的是GBK编码，而数据库认为客户端使用的是utf8编码，所以报错

查看MySQL客户端编码格式：



解决方法：

问题：客户端中的编码格式是GBK的，无法修改（如上图红色圈中部分所示）

方法：告诉服务器，客户端使用的编码格式

SET character\_set\_client=gbk ;

错误二：

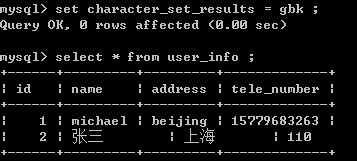
查询出显示的结果为乱码：



原因：数据库中用的是utf8编码，显示从服务器查询到的结果集用的也是utf8编码，但是客户端显示的是用gbk编码

解决方案：将结果集的编码设置为gbk

SET character\_set\_results = gbk ;



### 6.更新数据

**UPDATE *tbl\_name***

**SET *col\_name1*=*expr1* [, *col\_name2*=*expr2* ...]**

**[WHERE *where\_definition*]**

UPDATE 表名 SET 字段名 = 新数据值 WHERE … …;

* UPDATE语法可以用新值更新原有表行中的各列。
* SET子句指示要修改哪些列和要给予哪些值。
* WHERE子句指定应更新哪些行。如没有WHERE子句，则更新所有的行。

UPDATE user\_info SET username = ‘zhangsan’ where username = ‘lisi’ ;

### 7.删除一条数据（记录）

DELETE FROM 表名 WHERE 字段名 = 要求的值 ；

删除表中所有的记录

DELETE FROM 表名 ；----🡪逐条删除

摧毁整张表结构，重新建立表结构

TRUNCATE TABLE 表名；

### 数据库查询操作（DQL）：

**SELECT [DISTINCT] \*|{*column*1, *column*2. *column*3..}**

**FROM *table where 条件;***

* select 指定查询哪些列的数据。
* column指定列名。
* \*号代表查询所有列。
* from指定查询哪张表。
* DISTINCT可选，指显示结果时，是否剔除重复数据

### 8.查询数据库表中的数据

8.1查询出一条记录的所有字段

SELECT \* FROM 表名 ；

8.2查询某条记录指定的几个字段属性值

SELECT 字段名1，字段名2 FROM 表名 ；

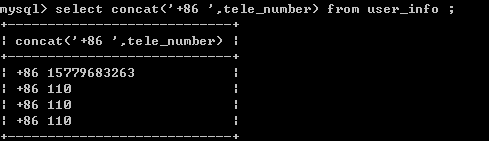
8.3查询出符合条件的记录，并过滤掉相同的数据，如查询一个班级里所有的学生的英语成绩，显示时过滤掉相同成绩即可加上distinct

SELECT DISTINCT 字段名1，字段名2…… FROM 表名 ；

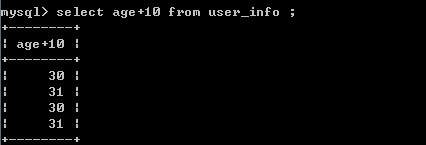
8.4 在SELECT 语句中使用表达式对查询的列进行运算（数据库中的数据不发生变化，只是执行SQL查询语句操作时，执行相应的加减、重命名等操作并显示出来）

8.4.1如，查询表中所有的电话号码，并在其前面加上 +86 两个字符，最终显示到命令行中

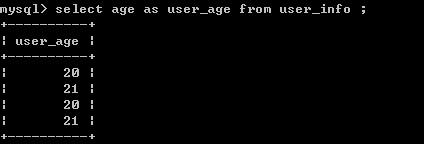
concat(a,b) 是MySQL中进行字符拼接的操作



8.4.2 查询所有用户的年龄，并+10后显示出来



8.4.3 查询所有用户的年龄，并以别名显示出来，通过关键字as指定别名 ： select 字段名1 as 字段名2 from 表名 ;



8.4.4 区间段查询

SELECT \* FROM 表名 WHERE 字段名 BETWEEN value1 AND value2 ；

查询年龄在20到22岁之间的用户

select \* from user\_info where age between 20 and 22 ;

8.4.5 集合数据匹配查询

SELECT \* FROM 表名WHERE 字段名 IN (value1，value2，value3);

查询年龄为20,21,22的所有用户

select \* from user\_info where age IN (20,21,22);

8.4.6 查询和含有某个关键字符的属性

% ：匹配多个字符

\_ : 匹配一个字符

如：查询所有姓李的学生成绩。

SELECT \* FROM student WHERE name LIKE '李%';

查询出表中的address以 “安”字结束的那条记录

SELECT \* FROM 表名 WHERE address LIKE ‘\_安’;

8.4.7 对查询出的结果进行排序输出 ORDER BY 字段名 ASC/DESC

ASC : 升序排列

DESC : 降序排列

对数学成绩排序后输出。

SELECT name,math FROM student ORDER by math;

对总分排序后输出，然后再按从高到低的顺序输出

SELECT name AS 姓名,chinese+english+math 总分 FROM student ORDER BY 总分 DESC;

对姓李的学生成绩排序输出

SELECT \* FROM student WHERE name LIKE '李%' ORDER BY chinese;

MySQL 基本数据类型



Select 的WHERE 语句的常用逻辑、数值操作

