# 系统默认字符集

show variables like 'character\_set\_%' 查看mysql是不是utf-8

character\_set\_connection | utf8   连接层字符集

character\_set\_database   | latin1           当前选中数据库的默认字符集

character\_set\_filesystem | binary

character\_set\_results    | utf8       查询结果字符集

character\_set\_server    | latin1       默认的内部操作字符集

character\_set\_system  | utf8   系统元数据(字段名等)字符集

character\_sets\_dir       | /usr/local/mysql-5.5.23-osx10.6-x86\_64/share/charsets/

查看到：

character\_set\_database和character\_set\_server依然是latin1的字符集，也就是说mysql后续创建的表都是latin1字符集的，不是utf8，会造成一些麻烦。所以有必要修改my.cnf，在修改my.cnf之前一定要关闭mysql进程，不然会遇到mysql的sock不能连接的问题。

## 设置默认字符集

临时设置：会话级别

SET NAMES 'utf8';

它相当于下面的三句指令：

SET character\_set\_client = utf8;

SET character\_set\_results = utf8;

SET character\_set\_connection = utf8;

# databases

 数据库(DataBase):简称DB，按照某一特定数据结构来组织，存储数据的仓库。

show databases; 显示所有的db；

show CREATE database wwl; 查询wwl数据库详细的创建情况

## 创建数据库

有三个属性：字符集和排序规则

IF NOT EXISTS ：这关键字用来判断数据库名字，有了这关子健：有同名也不会报错，（可有可无）

CHARACTER SET: 字符集

COLLATE： 排序规则 (后续有兴趣继续深究这里不写了)

CREATE DATABASE IF NOT EXISTS `wwl3` CHARACTER SET 'utf8' COLLATE 'utf8\_general\_ci';

# table

MySQL的数据表类型很多，其中比较重要的是MyISAM，InnoDB这两种。

## 表类型

### **InnoDB**

InnoDB是MySQL4.0之后推出的一种比较新的数据表类型，这种类型是事务安全的。它与BDB类型具有相同的特性，它们还支持外健。InnoDB表格速度很快具有比BDB还丰富的特性，因此如果需要一个事物安全的存储引擎，建议使用它。如果你的数据执行大量的insert或者update，处于性能方面的考虑，同样应该使用InnoDb表。对于支持事物的InnoDB类型的表来说，影响速度的主要原因是autcommit设置是打开的，而且程序没有显式调用begin开始事物，导致每插入一条都自动提交，严重影响了速度。可以在执行sql前调用begin，多条sql想成一个事物，将大大提高性能。

### MyISAM

MyISAM基于ISAM代码，可以说是ISAM的衍生品，不过增加了不少好用的扩展。它是MySQL的默认数据表类型，基于了传统的ISAM类型，ISAM是Indexed Sequential Access Method(有索引的顺序访问方法)的缩写，一般来说，它是存储记录和文件的标准方法。与其他存储引擎相比，MyISAM具有检查和修复表格的大多数工具。ISAM表格可以被压缩，而且它们支持全文索引，不过它们不是事务安全的，也不支持外健。如果事务会滚将会造成不完全会滚，从而不具备原子性。假如忽略事物以及访问并发性的话，并且需要执行大量的select检索语句的话，MyISAM将是最好的选择。

### 区别

（1）事物处理：

MyISAM是非事物安全型的，而InnoDB是事物安全型的（支持事物处理等高级处理）；

（2）锁机制不同：MyISAM是表级锁，而InnoDB是行级锁。

表级锁：锁住整个表，行级锁：只锁表中字段一行

（3）select，update，insert，delete操作：

MyISAM：如果执行大量的select，MyISAM是更好的选择

InnoDB：如果你的数据执行大量的insert或update，出于性能方面的考虑，应该使用InnoDB

（4）查询表的行数不同：

MyISAM：select count(\*) from table,MyISAM只是简单的读出保存好的行数，注意的是，当count（\*）语句包含where条件时，两种表的操作是一样的。

InnoDB：InnoDB中不保存表的具体行数，也就是说，执行select count（\*）from table时，InnoDB要扫描一遍整个表来计算有多少行

5）外健支持：

MyISAM表不支持外健，而InnoDB支持

为什么MyISAM会比InnoDB的查询速度快？

InnoDB在做select的时候，要维护的东西比MyISAM引擎多很多；

1）数据块，InnoDB要缓慢，MyISAM只缓存索引快，这中间还有换进换出的减少；

2）InnoDB寻址要映射到块，再到行，MyISAM记录的直接是文件的offset，定位比InnoDB要快

3）InnoDB还需要维护MVCC一致；虽然你的场景没有，但他还是需要去检查和维护

MVCC（Muti-Version Concurrency Control）多版本并发控制

### 应用场景

MyISAM适合：（1）做很多count的计算；（2）插入不频繁，查询非常频繁；（3）没有事务。

InnoDB适合：（1）可靠性要求比较高，或者要求事务；（2）表更新和查询都相当频繁，并且行锁定的机会比较大的情况。

## 创建表

use wwl; //先进入数据库

CREATE TABLE `user` (

`id` int(10) NOT NULL AUTO\_INCREMENT , //not null :不能为空，auto\_increment:自动增长

`name` varchar(10) NOT NULL DEFAULT '默认内容' COMMENT '这是注释关键字' ,

`user\_id` int(10) NOT NULL

PRIMARY KEY (`id`) // PRIMARY KEY：设置主键

CONSTRAINT `ccc` FOREIGN KEY (`user\_id `) REFERENCES `users` (`userid`) //设置字段user\_id和users表的userid外键关联名字：ccc

)

ENGINE=InnoDB //引擎(表类型)

DEFAULT CHARACTER SET=utf8 //字符集编码

AUTO\_INCREMENT=1; //上面设置的自动增长：这里设置自动增长次数

注意：标点符号的使用：`user`这里的标点符号不是单引号而是键盘数字1旁边的

## 约束

唯一索引：此表中唯一的，不能有重复，一般用于外键