

# mysql 第一课 数据库这玩意

## 所谓 sql

### 名词解释

Sql = Structured Query Language 资料查询语言

我们来看几个 sql 语句

```
Select * from table;
```

```
Update table set name='sql' where id =1;
```

```
Delete from table where name != 'sql';
```

....

原来 sql 也是一种语言,我们知道,语言不过也是一种协议罢了!

那 sql 和 http 有什么关系吗?

其实是一样的,都是一种协议而已!!

### 问题

但凡协议,都须两端,http 的两端是浏览器和 apache,那么 sql 的两端是什么?

人(或 sql 客户端)和数据库服务器(如 mysql)!!

就像 http 中 apache 要完成 http 协议逻辑一样,mysql 也要完成 sql 定义的语义,而且必须完成!

这就是 sql 与 mysql 的关系

## 所谓数据库(database)

回忆

Access, sqlserver, mysql 都是大名鼎鼎数据库

问题

为什么要有数据库？

举个表格的例子

		D	E	F	G	H	I	J	K	L	M
届高三年级第一次校统测—各班各科平均											
科类	班级	各科平均分						班级总平均分	排序	班主任	
理科	1	语文	数学	英语	物理	化学	生物		#N/A		显示最新数据
	2								#N/A		
	3								#N/A		
	4								#N/A		
	5								#N/A		
	6								#N/A		
	7								#N/A		
	8								#N/A		
	9								#N/A		
	10								#N/A		
	11								#N/A		
	12								#N/A		
	13								#N/A		
年级平均分											
科类	班级	各科平均分						班级总平均分	排序	班主任	
文科	16	语文	数学	英语	政治	历史	地理		#N/A		
	17								#N/A		
	18								#N/A		
	19								#N/A		
	20								#N/A		
	21								#N/A		
	22								#N/A		
	23								#N/A		
年级平均分											

这是个简单的成绩表格的例子，大家知道每个行是一个班级，每个列是一种学科，而现实生活中，这样的表格例子比比皆是！表格往往是一些同类事物的各个属性组成的一个集合，就像这个成绩表！

试想，如果学校增加到 1 万，班级增加到 10 万个，人类还能维护这个表格吗？于是，数据库应运而生，为了方便的解决，存储，管理和准确的维护该类型的数据表格，产生了关系型数据库；每一个行在数据库里是一个记录，每个列在数据库里是表格一个的属性列，而每个同类型的数据集合，就是一个数据库里的表！

这就是数据库！

## mysql 也是一种数据库

mysql 也不过是一种数据库；

当然，作为 sql 的服务端，mysql 同时要完成数据库的存储功能和提供 sql 接口的功能，所以 mysql 在物理上是一个服务器，一个软件，也是一个程序！

[视频下载地址]

<http://linux-e.googlecode.com/files/LAMP%E4%B9%8Bmysql01-%E6%95%B0%E6%8D%AE%E5%BA%93%E8%BF%99%E7%8E%A9%E6%84%8F%E5%84%BF.rar>

# 安装 mysql

## 回忆

回忆一下 apache 的安装过程！

## 示例

### 1. download

<http://mysql.mirror.tw/Downloads/MySQL-5.1/>

### 2. configure

./configure

### 3. make

make

### 4. make install

make install

# 启动 mysql

## 思考

mysql 程序在哪里？

find / -name mysql -ctime 0

mysql 配置在哪里？

find / -name my.cnf

mysql 数据库的数据存在哪里？

## 示例

### 1. mysql 初始化

mysql\_install\_db --user=mysql --group=mysql

这个初始化作了什么？

我们来看看都有哪些变化？

发现多了些目录;我们删除这些目录;再初始化一遍看看!!

### 2. 启动脚本启动

启动脚本做了什么？

**mysqld\_safe &**

## 思考

**Ps -ef | grep mysql**

我们发现, 除了 mysqld, mysqld\_safe 也一直运行着,为什么?

找找默认的配置是哪个? /etc/my.cnf

我们修改这个默认配置;重新启动看有什么效果

```
Ps -ef | grep mysql
```

### 3.关闭 mysql

```
mysqladmin shutdown
```

#### 总结

```
Ps -ef | grep mysql
```

```
/usr/local/libexec/mysqld --basedir=/usr/local --datadir=/var/lib/mysql --user=mysql  
--log-error=/var/lib/mysql/vdev.sina.com.cn.err --pid-file=/var/run/mysqld/mysqld.pid  
--socket=/var/lib/mysql/mysql.sock
```

我们发现启动脚本其实是读了默认的配置文件的/etc/my.cnf，然后将配置文件里的配置项写成mysqld 的参数启动

#### 关键参数

--datadir=/var/lib/mysql [提示] mysql 数据路径，每个 **mysql 实例**都对应一个相应的 datadir  
这也说明 mysql\_install\_db 做的工作就是为 **mysql 实例**初始化 datadir

--log-error=/var/lib/mysql/vdev.sina.com.cn.err #**mysql 实例**的错误日志

最重要的参数就是 datadir

#### 思考

一个服务器上能运行多个这种 mysql 实例吗?

#### [视频下载地址]

<http://linux-e.googlecode.com/files/LAMP%E4%B9%8Bmysql02-%E5%90%AF%E5%8A%A8.rar>

# 访问 mysql

## 示例

通过 mysql shell 访问？通过 php 语言客户端访问？

mysql -uroot -p

执行一些 sql 语句

```
Show databases;
```

```
Use database;
```

```
Show tables;
```

## 试验

如果我们删除一些东西, 还能正常访问吗?

删除 mysql datadir?

删除 db 目录?

删除 默认的 mysql 目录?

删除默认配置?

杀了进程?....

[视频下载地址]

<http://linux-e.googlecode.com/files/LAMP%E4%B9%8Bmysql03-%E8%AE%BF%E9%97%AEmysql.rar>

# mysql 的权限问题

## 举例

root 和它的密码

一个很有意思的问题. mysqlroot 密码丢失怎么办?

网上肯定一堆这种帖子, 我们来看看他们这些解决方法

1. 使用 `mysqladmin, mysqladmin -u root -p password mypasswd`

2. `mysql> Insert INTO mysql.user (Host,User,Password)`

`VALUES('%','jeffrey',PASSWORD('biscuit'));`

`mysql> FLUSH PRIVILEGES`

3.`bin/safe_mysqld --skip-grant-tables &` 不检查权限启动

`>use mysql`

`>update user set password=password("new_pass") where user="root";`

`>flush privileges;`

4. 直接用十六进制编辑器编辑 `user.MYD` 文件,修改密码字符串....

.....

OK,我们看到这些方法有一个共同点,是什么呢,他们都要在忘记密码,没有权限的情况下,绕过权限检查(不管是无权限启动,还是直接修改文件), 然后第二步的话,都是在 `mysql` 库的 `user` 表上做文章,也就是说,最终 `root` 的权限(包括其他用户)存储在 `user` 表上,我们只要修改这个表,就能修改权限!

好啦,通过这个我想大家对 `mysql` 权限系统已经料了如执掌了!

## 思考

根据以上思路,你还能自己设计另一种全新的找回密码方法吗?

**[视频下载地址]**

**<http://linux-e.googlecode.com/files/LAMP%E4%B9%8Bmysql04-mysql%E6%9D%83%E9%99%90.rar>**

# mysql 第二课 库和表的操作

## 数据库和文件系统

### 问题

我们重启了 mysql,数据还在,那么数据到底存在哪?

[回忆] datadir

我们之前也了解到,用户表对应着 user 文件!其他数据表也是存在文件里!!!  
所以,mysql 和文件系统是有关系的,mysql 就像寄生虫一样,依赖文件系统

### Mysql 的库和目录

1. Show databases 看看库
  2. 在 mysql 数据目录下 list 目录
- 我们发现每个目录对应一个数据库

### 试验

1. 我们新建一个目录  
发现 database 多了一个
2. 我们删除一个目录玩玩就知道了  
发现 database 少了一个
3. 我们新建一个库  
发现目录多了一个

### Mysql 的表和文件

- 1.Show tables 看看表
  - 2.在对应的目录下 list 文件
- 我们发现一个数据表,对应几个相关的文件

### 试验

将一个 mysql 数据库目录移开, 再移回来, 看数据有没有变化?

[视频下载地址]

<http://linux-e.googlecode.com/files/LAMP%E4%B9%8Bmysql05-mysql%E6%96%87%E4%BB%B6.rar>

# Create/drop

## 思考

真的可以随便把目录,文件搞来搞去吗???

## 试验

我们先做大量插入操作,然后马上移开目录, 关闭 mysql  
当我们重新移回数据, 启动 mysql

再看看出问题没有?

## 结论

还是走正道!

Create 命令, 创建命令, 包括创建 database, 创建 table  
Drop 命令, 删除命令, 包括删除 database, 删除 table

## 示例

```
Create database test_db;  
Create database test_db2;  
Use test_db;  
Create table test_tbl(  
    Id INT,  
    Name VARCHAR(100)  
);
```

```
Use test_db2;  
Create table test_tbl(  
    Id INT,  
    Name VARCHAR(100)  
);
```

```
Use test_db;  
Drop table test_tbl;  
Drop database test_db2;
```

[视频下载地址]

<http://linux-e.googlecode.com/files/LAMP%E4%B9%8Bmysql06-db.rar>



# Insert

## 回忆

数据库有三个元素,database, table, record

现在关于 database, table 的创建, 修改, 删除命令已经介绍完了  
开始介绍 record 相关的 sql 命令

## 问题

行/记录(record) 这是 mysql 数据表的基本组成,怎么新建一个记录呢?

Insert 插入,新建

## 示例

```
Use test_db
```

```
Insert into test_tbl values(1, 'lanjinwei');
```

```
Insert into test_tbl values(2, 'lanjinwei');
```

```
Insert into test_tbl values(3, 'lanjinwei');
```

```
Insert into test_tbl values(4, 'lanjinwei');
```

```
Insert into test_tbl values(5, 'lanjinwei');
```

# Select

## 问题

怎么查找我要的记录呢?

Select

## 示例

```
Select * from test_tbl; //全部记录
```

```
Select * from test_tbl where id = 1; //某个条件的记录
```

```
Select * from test_tbl where id < 4 and id > 2; //某些个记录
```

```
Select id from test_tbl where name = 'lanjinwei'; //只查某些列
```

# Update

## 问题

怎么改变一个记录呢?

Update

## 示例

```
Update test_tbl set name = 'mysql' where id = 1;
```

```
Update test_tbl set name = 'user' where id >=4;
```

```
...
```

# Delete

## 问题

怎么删除一个记录呢?

delete

## 示例

```
Delete from test_tbl where id=3;
```

[视频下载地址]

<http://linux-e.googlecode.com/files/LAMP%E4%B9%8Bmysql07-sql%E8%AF%AD%E5%8F%A5.rar>

# mysql 第三课 DBA 的问题

## 我要一个权限!!!

### 情景

当你是 mysql 管理员的时候，肯定会有人找你要分配一个权限？

### 方案

GRANT \*\* ON \* TO test@localhost IDENTIFIED BY 'password';

示例

```
grant select,insert,update,delete,create,drop on vtdc.employee to joe@10.163.225.87 identified by '123';
```

## 查询好慢哦!!

### 情景

最近我的 mysql 数据库上了个新业务，有个查询怎么这么慢？

### 方案

1. 长期慢查询日志;

修改配置，

```
long_query_time=2  
log-slow-queries= /usr/var/slowquery.log
```

然后，我们看/usr/var/slowquery.log 这个 log 就可以查出哪些 sql 比较慢了

2. 当下, Show full process;

```
mysql> show full processlist;
```

# 数据丢了!!!

## 情景

昨天我磁盘坏了，正好 mysql 数据在那上面，全丢了,怎么办!!!

## 方案

数据要备份,

数据导出

1. mysqldump

```
mysqldump -uroot -p test_db > backup.sql
```

2. 备份目录

```
Cp -rp test_db /backup/
```

数据导入

1. 管道

```
Mysql -uroot -p < backup.sql
```

2. 移回目录

```
Cp -rp /backup/test_db ./
```

# 数据库扛不住啦!!

## 情景

我们的 mysql 压力越来越大，现在一个库已经扛不住访问拉？

## 方案

为 mysql 建立辅库，搭建主辅库分担压力

## 实践

为当前数据库配置两个辅库！

## [视频下载地址]

<http://linux-e.googlecode.com/files/LAMP%E4%B9%8Bmysql08-DBA%E7%9A%84%E9%97%AE%E9%A2%98.rar>