

Mysql

搭建环境 WAMP

W windows 系统

A apache 服务器

M mysql 数据库

P php

数据库的简介

1.1 数据库

存储,管理数据的仓库. **Datebase**. 实际项目中,一般将项目相关的数据全部存储在指定的数据库中.

存储管理数据的工具有很多,比如 **txt.excle.word** 为什么要存储数据库.

数据量较大,数据与数据之间的较为复杂的时候,数据库有专门的,高效的管理系统—数据库管理系统 **DBMS**.

1.2 数据库分类

数据库大小 :

大型数据库: **oracle** (甲骨文): 大型,维护费用昂贵;

DB2: 数据库中的黄金,一般用于金融产品;

中型数据库: **SQL Server** 微软产品

MySQL

发展历程, 1. 是瑞典的 **AB** 公司产品

2. 2008 年 1 月被 **sun** 公司收购

3. 2009 年 4 月被 **oracle** 公司收购

4. 2010 年发布了 **mysql 5.5**

5. **oracle** 对 **mysql** 版本重新进行了划分,分成了社区版和企业版

小型数据库 : **access** 数据库

1.3 关系型数据库

1 实体集,实体,属性

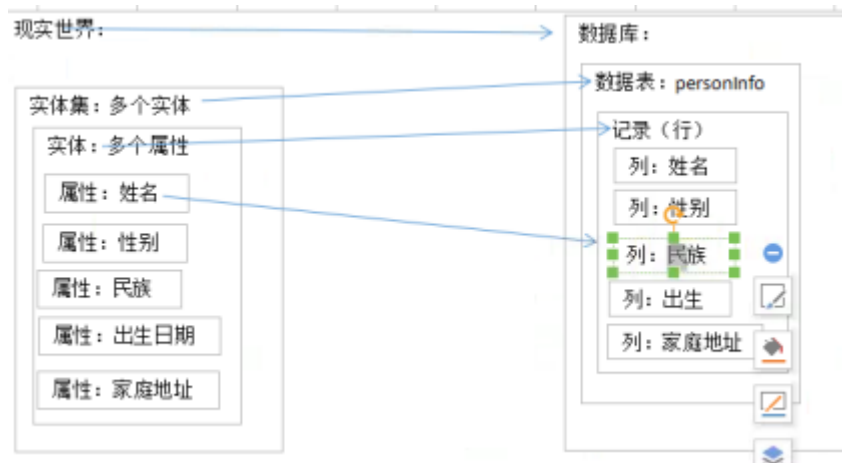
基于关系模型而组织的数据库.一个关系模型就是一个二维表(同 **excle** 表中的行和列);

实体 : 描述客观事物的概念,具体的事物(学生,人,手机),抽象的事物(订单,购物车);

实体集: 同一类的所有实体组成的整体.

实体集在数据中表示一张二维表,数据表.

实体具有很多属性, 对应数据表中的列.



1.4 数据库的架构

数据表(table):

表示以行和列的形式组织起来的数据的集合.存储某一类实体的数据,如:学生信息,员工信息,文章信息.一个数据库中可以有多个数据表.

记录: (行,row,record)

一个有意信息的集合,(一行信息表示一个实体)

一个数据表中可以有 N 条记录

字段: (列,column,field)

某一个列的值的名称.

4. SQL

SQL: structured Query Language 结构化查询语言.

SQL 语句 : 它相当于代码或者发送的命令.

1.5 简介

关系型数据库使用通用语言.

数据库脚本语言文件的后缀.

1.6 分类

根据操作(数据库,数据表,字段) 或者层次的不同,分为:

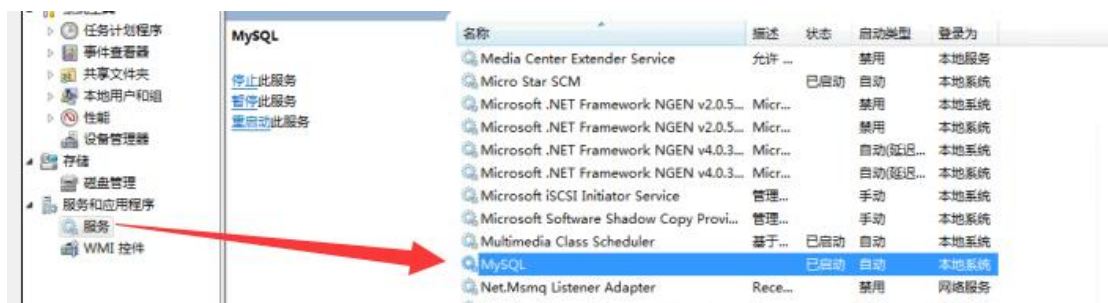
1. 数据查询语言(DQL : Data Query Language) select(查询数据);
2. 数据定义语言 (DDL: Data Definition language) create drop alter
3. 数据操作语言 (DML: Data Manipulation Language) insert 添加 update 修改 delete 删除
4. 数据控制语言 (DCL: Data Controler Language) grant,revoke(用户管理)

访问数据库服务器

1.7 访问步骤

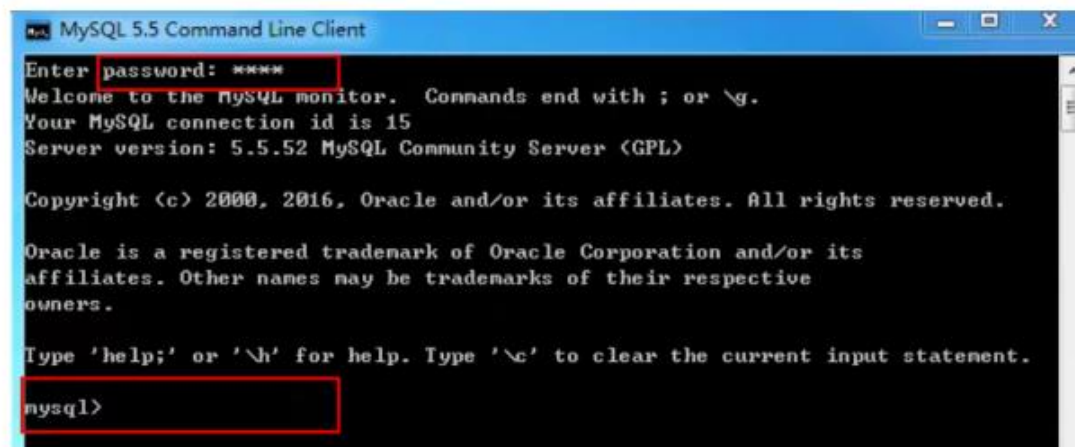
1. 开启数据库服务
2. 通过客户端发送命令到数据库

1.8 开启数据库服务



MySQL 5.5 Command Line Client

访问成功:



计算机-->右键-->管理-->服务和应用程序-->服务

1.9 命令行方式启动服务

Win 键+R --> 运行（左下角）--> 输入：net start MySQL



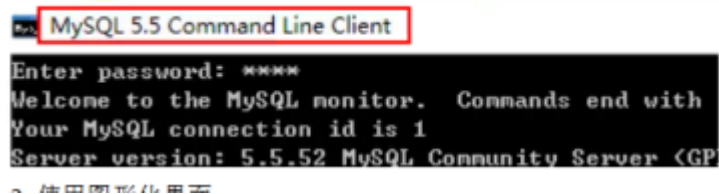
启动服务：net start MySQL

停止服务: net stop MySQL

1.10 客户端连接数据库服务器

数据库结构 C(client)/S(Server)架构通过客户端访问数据库服务器

1 命令行(mysql command line client)



Php 可以直接操作数据库

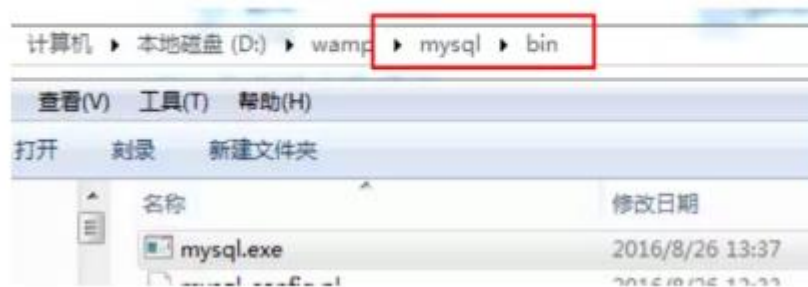
PhpMyAdmin(Web 程序操作数据库)

1.11 命令行连接数据库

CMD 访问数据库:

1 配置环境变量

MySQL 数据库安装目录 bin 目录



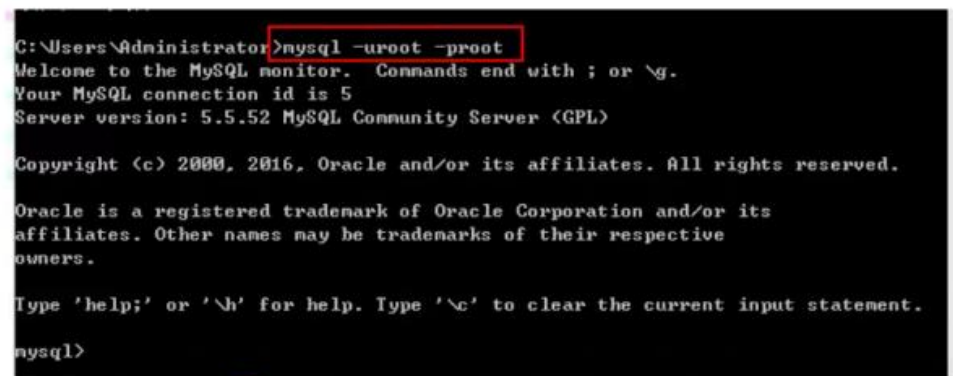
效果: 输入 mysql



说明: mysql mysql 数据库 bin 目录下的 mysql.exe (配置环境变量起的作用)

- h, host 主机名, 主机地址 (ip 地址)。数据库默认主机名为 localhost (本机), 可以省略
- p, port 端口号 (安装数据库默认端口号: 3306), 可以省略
- u, user 用户名
- p, password 密码。可以直接输入, 也可以回车之后输入 (保护账号信息安全)

直接输入密码:



1.11 退出登录

1 exit

```
mysql> exit
Bye
```

2 quit

```
mysql> quit
Bye
```

3 或者 \q

```
mysql> \q;
Bye
```

1.13 查看数据库

查看数据库服务器全部的数据库

语法: `show databases;`

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| test |
+-----+
4 rows in set (0.00 sec)

mysql>
```

注意: 1 不要把 data 写成了 date;
2 databases 不要少些了 s;

1.14 查找错误方法

从 near 位置开始找

```
mysql> show database ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MySQL server version for the right syntax to use near 'datab
ase' at line 1
mysql>
```

故意少了*, 出现了错误
near位置开始, 后面就是错误提示

1.15 查看数据库的创建语句

```
语法: show create database 数据库名称 ;

mysql> show create database test ;
+-----+-----+
| Database | Create Database |
+-----+-----+
| test     | CREATE DATABASE `test` /*!40100 DEFAULT CHARACTER SET utf8 */ |
+-----+-----+
1 row in set (0.00 sec)
```

1.16 创建数据库

```
语法: create database [if not exists] 数据库名称;

创建 study 数据库

mysql> create database study;
Query OK, 1 row affected (0.00 sec)

mysql> show databases ;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| study |
| test |
+-----+
5 rows in set (0.00 sec)
```

数据库名称规范

- 1 字母,数字,下划线
- 2 不能使用纯数字的名称
- 3 命名 见名知意
- 4 使用系统关键字,特殊字符时候,需要使用` (esc 下面的键,数字 1 左边的键)包括起来
- 5 长度不超过 64 个字符.

1.17 删除数据库

语法: drop database [if exists] 数据库名称;

删除数据库:

```
mysql> show databases ;
+-----+
| Database |
+-----+
| information_schema |
| mysql      |
| performance_schema |
| study      |
| test       |
+-----+
5 rows in set (0.00 sec)

mysql> drop database study;
Query OK, 0 rows affected (0.00 sec)

mysql> show databases ;
+-----+
| Database |
+-----+
| information_schema |
| mysql      |
| performance_schema |
| test       |
+-----+
```

删除数据库

1.18 修改数据库

主要数据库字符集

主要数据库字符集。

语法: alter database 数据库名称 charset 字符集;

修改数据库编码:

```
mysql> alter database study charset gbk;
Query OK, 1 row affected (0.00 sec)
```

1.19 切换数据库(选择数据库)

一般先选择数据库, 然后对其居中的数据表进行操作。

语法: use 数据库名称;

```
mysql> use study;
Database changed
mysql>
```


查看当前选择的数据库

```
语法: \s;

mysql> \s;

-----
mysql  Ver 14.14 Distrib 5.5.52, for Win64 (<x86>)

Connection id:          8
Current database:       study
Current user:           root@localhost
SSL:                    Not in use
Using delimiter:        ;
Server version:         5.5.52 MySQL Community Server (GPL)
Protocol version:       10
Connection:             localhost via TCP/IP
Server characterset:    utf8
Db      characterset:    gbk
Client characterset:    utf8
Conn.  characterset:    utf8
TCP port:               3306
Uptime:                 51 min 5 sec

Threads: 3  Questions: 44  Slow queries: 0  Opens: 35  Flush tables:
les: 2  Queries per second avg: 0.014
-----
```

数据表操作

1.20 查看数据表

查看当前数据库中的全部数据表

```
语法: show tables;

mysql> show tables;

+-----+
| Tables_in_mysql |
+-----+
| columns_priv    |
| db              |
| event           |
| func            |
| general_log     |
| help_category   |
| help_keyword    |
| help_relation   |
| help_topic      |
| host            |
```

1.21 查看数据表结构

查看表结构，表结构（字段，数据类型，数据属性）

语法: desc 数据表名称;

```
mysql> desc db;
+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default |
+-----+-----+-----+-----+
| Host           | char(60)      | NO   | PRI |          |
| Db             | char(64)      | NO   | PRI |          |
| User           | char(16)      | NO   | PRI |          |
| Select_priv    | enum('N','Y') | NO   |     | N        |
| Insert_priv    | enum('N','Y') | NO   |     | N        |
| Update_priv    | enum('N','Y') | NO   |     | N        |
| Delete_priv    | enum('N','Y') | NO   |     | N        |
| Create_priv    | enum('N','Y') | NO   |     | N        |
| Drop_priv      | enum('N','Y') | NO   |     | N        |
| Grant_priv     | enum('N','Y') | NO   |     | N        |
| References_priv| enum('N','Y') | NO   |     | N        |
| Index_priv     | enum('N','Y') | NO   |     | N        |
| Alter_priv     | enum('N','Y') | NO   |     | N        |
| Create_tmp_table_priv| enum('N','Y')| NO   |     | N        |
```

1.22 创建表

语法: create table 表名 (
 字段名称 (列) 数据类型,
 字段名称 (列) 数据类型
);

注意 : 数据中有字段,字段定义离不开数据库表的定义,数据表定义离不开字段的定义.

不同字段之间使用英文的逗号分隔.

最后一个字段与括号之间不能有逗号.

创建学生表: 字段有, 学号, 姓名, 年龄, 性别等

```
mysql> create table student(
-> stuno varchar(10),
-> stuname varchar(50),
-> age int ,
-> sex char(2)
-> );
Query OK, 0 rows affected (0.01 sec)

mysql> show tables;
+-----+
| Tables_in_study |
+-----+
| student         |
+-----+
1 row in set (0.00 sec)
```

```
mysql> desc student;
```

Field	Type	Null	Key	Default	Extra
stuno	varchar(10)	YES		NULL	
stuname	varchar(50)	YES		NULL	
age	int(11)	YES		NULL	
sex	char(2)	YES		NULL	

1.23 查看数据表的创建语句

语法: show create table 表名称;

Show create table 表名称 \G

```
mysql> show create table student;
```

Table	Create Table
student	CREATE TABLE `student` (`stuno` varchar(10) DEFAULT NULL, `stunane` varchar(50) DEFAULT NULL, `age` int(11) DEFAULT NULL, `sex` char(2) DEFAULT NULL) ENGINE=MyISAM DEFAULT CHARSET=gbk ;

1 row in set (0.00 sec)

可以修改后面的分号 (;) 为 \G, 查看起来更清晰

1.24 删除数据表

语法: `drop table [if exists] 数据表名称;`

```
mysql> drop table student;
Query OK, 0 rows affected (0.00 sec)

mysql> drop table student;
ERROR 1051 (42S02): Unknown table 'student'
mysql> show tables;
Empty set (0.00 sec)
```

重复删除

```
mysql> drop table student;
Query OK, 0 rows affected (0.00 sec)

mysql> drop table student;
ERROR 1051 (42S02): Unknown table 'student'

mysql> drop table if exists student;
Query OK, 0 rows affected, 1 warning (0.00 sec)
```

数据操作

1.25 添加数据

语法: `insert into 表(字段1, 字段2, 字段3) values (值1, 值2, 值3)`

注意: 插入数据时, 字段列表和数据要一一对应

添加一条数据到 student 表

```
mysql> insert into student(stuno,stuname,age,sex) values('001','PPX',20,'F');
Query OK, 1 row affected (0.00 sec)

mysql> select * from student;
+-----+-----+-----+-----+
| stuno | stuname | age | sex |
+-----+-----+-----+-----+
| 001   | PPX     | 20  | F   |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

1.26 查询数据

语法: `select */字段列表 from 表;`

* 代表全部列

```
mysql> select * from student;
+-----+-----+-----+-----+
| stuno | stuname | age | sex |
+-----+-----+-----+-----+
| 001   | PPX    | 20  | F   |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

可以查询指定的列

```
mysql> select stuno, stuname from student;
+-----+-----+
| stuno | stuname |
+-----+-----+
| 001   | PPX    |
+-----+-----+
```

1.27 修改数据

语法: `update 表 set 字段=值, 字段=值 where 条件 ;`

将学号 001 的学生年龄修改为 25

```
mysql> update student set age=25 where stuno = '001';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from student;
+-----+-----+-----+-----+
| stuno | stuname | age | sex |
+-----+-----+-----+-----+
| 001   | PPX    | 25  | F   |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

1.28 删除数据

语法: `delete from 表 where 条件;`

删除数据:

```
mysql> delete from student ;
Query OK, 1 row affected (0.00 sec)

mysql> select * from student;
Empty set (0.00 sec)
```