



MySQL 数据库

认识什么是库？

京东仓库：



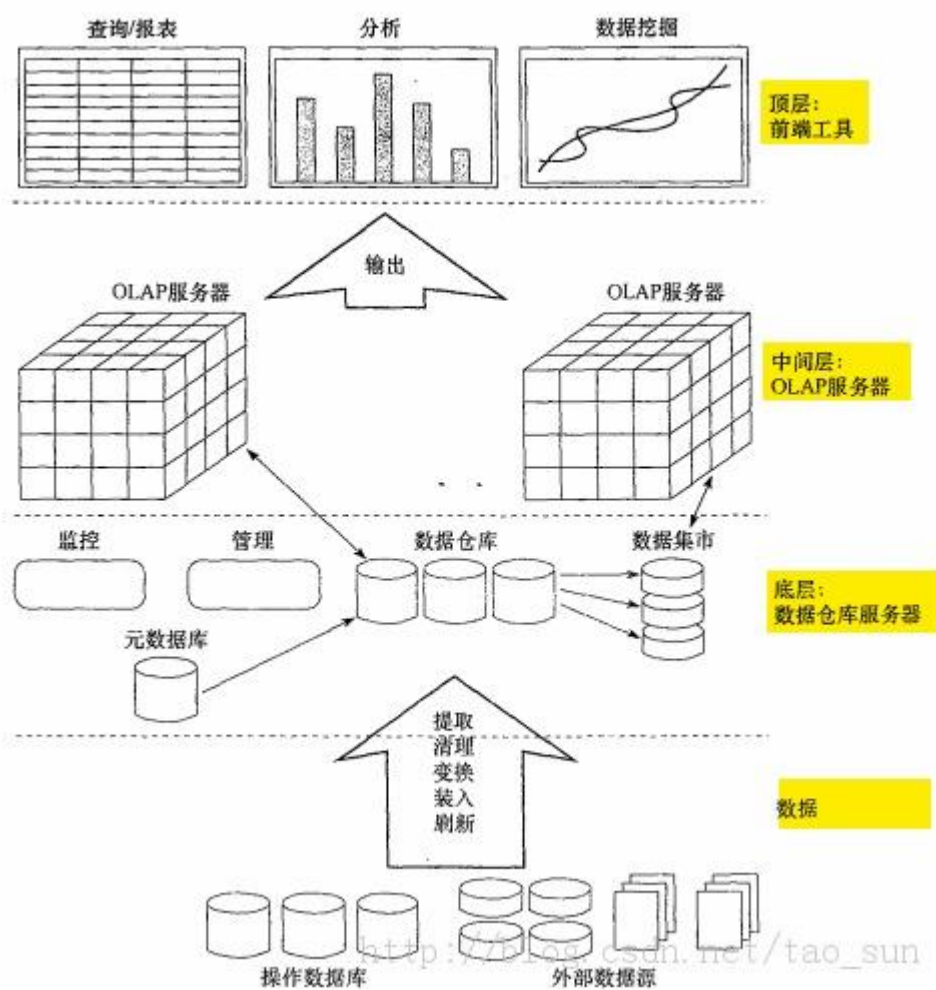
弹药库：



金库：



数据仓库：



1. 数据库概念介绍

数据库(Database)是按照一定的结构在电脑上保存和管理数据的软件。

数据库：存储数据的仓库。



玩家数量超过 2 亿！



在没有计算机之前，我们通过人工来管理数据，随着数据量的不断增长，通过人工来管理数据效率极其的低下，有了计算机之后，人们开始使用计算机来管理数据，并开发了专门用来管理、组织数据的软件，数据库。

1.1 数据库分类

1.1.1 关系数据库

关系数据库，是建立在关系模型基础上的数据库，借助于集合代数等数学概念和方法来处理数据库中的数据。

数据库的另外一种区分方式：基于存储介质
存储介质分为两种：磁盘和内存

关系型数据库：存储在磁盘中
非关系型数据库：存储在内存中

1.1.2 非关系数据库

非关系型数据库严格上不是一种数据库，应该是一种数据结构化存储方法的集合。
必须强调的是，数据的持久存储，尤其是海量数据的持久存储，还是需要关系数据库这员老将

由于关系型数据库本身天然的多样性，以及出现的时间较短，因此，不像关系型数据库，有几种数据库能够一统江山，关系型数据库的非常多，并且大部分都是开源的，这里列出一些：

Redis,Tokyo

Cabinet,Cassandra,Voldemort,MongoDB,Dynomite,Hbase,CouchDB,Hypertable,Riak,Tin,Flare,Lightcloud,KiokuDB,Scalaris,Kai,ThruDB

2. 关系型数据库

2.1 基本概念

关系数据库，是建立在关系模型基础上的数据库，借助于集合代数等数学概念和方法来处理数据库中的数据。现实世界中的各种实体以及实体之间的各种联系均用关系模型来表示。关系模型是由埃德加·科德于 1970 年首先提出的，并配合“科德十二定律”。现如今虽然对此模型有一些批评意见，但它还是数据存储的传统标准。关系模型由**关系数据结构**、**关系操作集合**、**关系完整性约束**三部分组成。

关系数据结构：指的数据以什么方式来存储，是一种二维表的形式存储
本质：二维表

编号	姓名	性别	年龄	电话	家庭住址
1	张三丰	男	12	138 3838 1438	花果山 1 号坑
2	黑山老妖	女	10	138 3838 1438	花果山 2 号坑
3	东方不败	男	10	138 3838 1438	花果山 3 号坑

关系操作集合：如何来关联和管理对应的存储数据，SQL 指令

获取张三的年纪：已知条件为姓名

Select 年龄 from 二维表 where 姓名 = 张三;

关系完整性约束：数据内部有对应的关联关系，以及数据与数据之间也有对应的关联关系

编号	姓名	性别	年龄	电话	家庭住址
1	张三丰	男	12	138 3838 1438	花果山 1 号坑
2	黑山老妖	女	18	138 3838 1438	花果山 2 号坑
3	东方不败	男	10	138 3838 1438	花果山 3 号坑

表内约束：对应的具体列只能放对应的数据（不能乱放）

表间约束：自然界各实体都是有着对应的关联关系（外键）

2.2 典型关系型数据库

Oracle、DB2、Microsoft SQL Server、Microsoft Access、MySQL、SQLite

小型关系型数据库：Microsoft Access，SQLite

中型关系型数据库：SQL Server，Mysql

大型关系型数据库：Oracle，DB2

Mysql 当前跟 Oracle 是一个公司的：隶属于 Oracle

2.3 库、表、字段、记录

先来搞明白三者之间的关系



想了解同学们的信息：

班级学生信息表

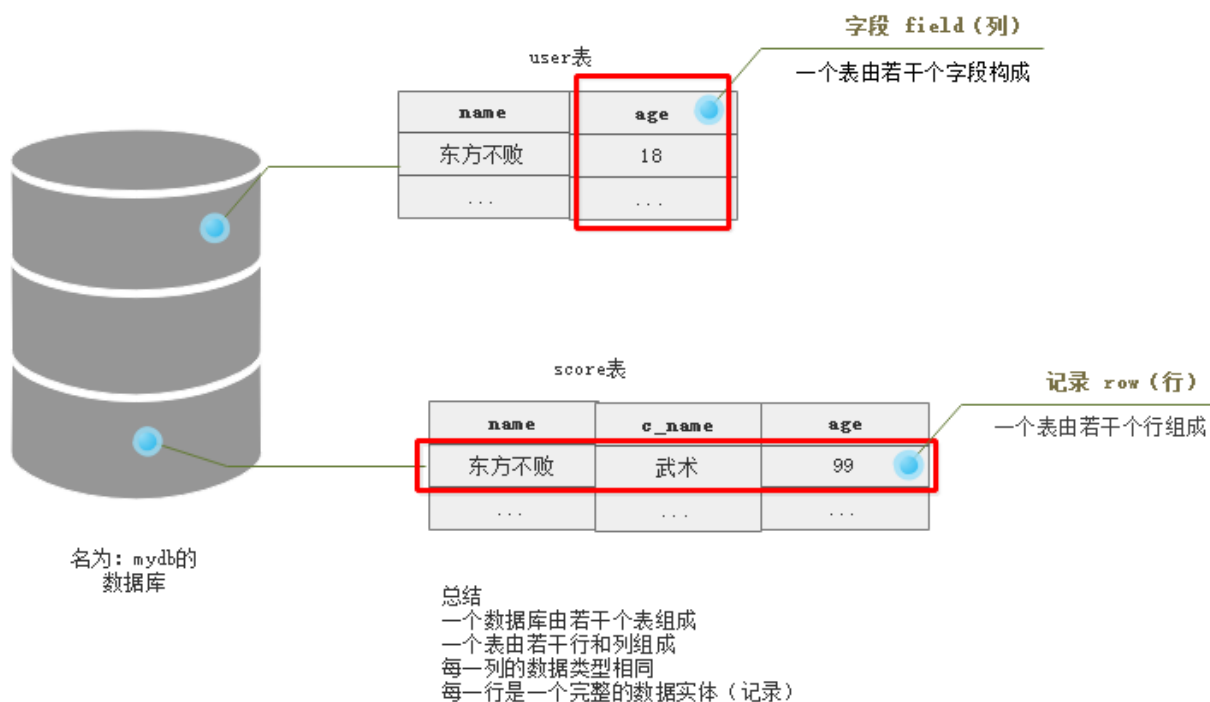
name	age	sex	address
王二小	18	男	5#250
张三丰	19	男	5#438
凤姐	17	女	2#202

宠物表

name	age	sex	address
小蛤	1	母	狗窝

表：
把所有的同学的信息放到一起，就是一个学生信息表
表用来存储同一类数据，由字段（列）和记录（行）来构成的

字段：



数据库软件中保存的数据是以表格的形式管理的，
比如学生信息：

编号	姓名	性别	年龄	电话	家庭住址
1	张三丰	男	12	138 3838 1438	花果山 1 号坑
2	黑山老妖	女	10	138 3838 1438	花果山 2 号坑
3	东方不败	男	10	138 3838 1438	花果山 3 号坑

在数据库中保存以上数据需要以下三步：

1. 创建数据库



2. 在数据库中创建学生表
3. 向学生表中插入三条记录数据

3. MySQL

3.1 基本介绍

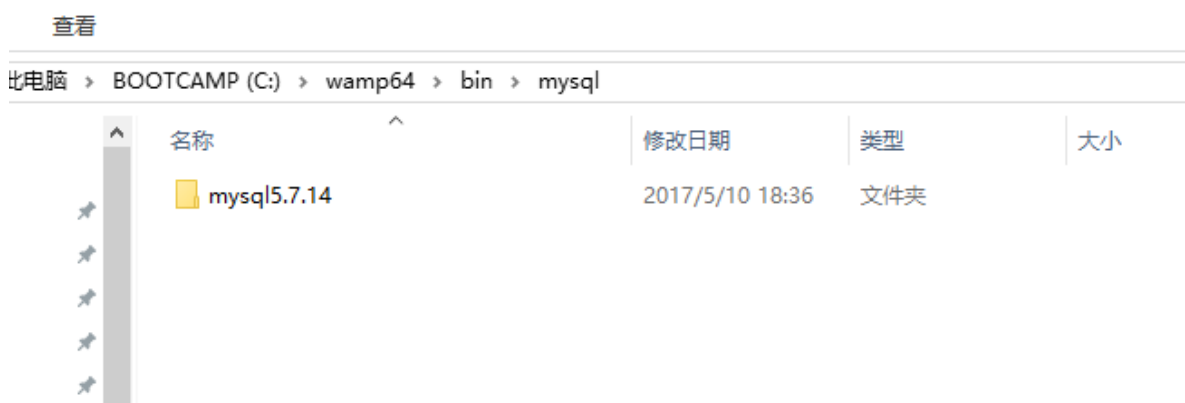
MySQL 是一个关系型数据库管理系统，由瑞典 MySQL AB 公司开发，目前属于 Oracle 旗下产品。MySQL 是最流行的关系型数据库管理系统之一，在 WEB 应用方面，MySQL 是最好的 RDBMS (Relational Database Management System，关系数据库管理系统) 应用软件。

AB 公司被 Sun 公司收购→Sun 公司又被 Oracle 公司收购了，所以现在 MySQL 和 Oracle 现在同属于一家公司。

- 1) MySQL 是一种开源免费的数据库产品
- 2) MySQL 对 PHP 的支持是最好 (wamp 或者 lamp)

MySQL 中用到的操作指令就是 SQL 指令

3.2 wampserver 中的 MySQL



名称	修改日期	类型	大小
bin	2017/5/10 18:35	文件夹	
data	2017/7/31 11:08	文件夹	
lib	2017/5/10 18:35	文件夹	
share	2017/5/10 18:36	文件夹	
COPYING	2016/7/12 13:55	文件	18 KB
my.ini	2017/5/10 18:36	配置设置	7 KB
my-default.ini	2016/7/12 14:41	配置设置	2 KB
README	2016/7/12 13:55	文件	3 KB
wampserver.conf	2015/12/11 12:55	CONF 文件	1 KB

3.2.1 MySQL 启动、关闭、重启

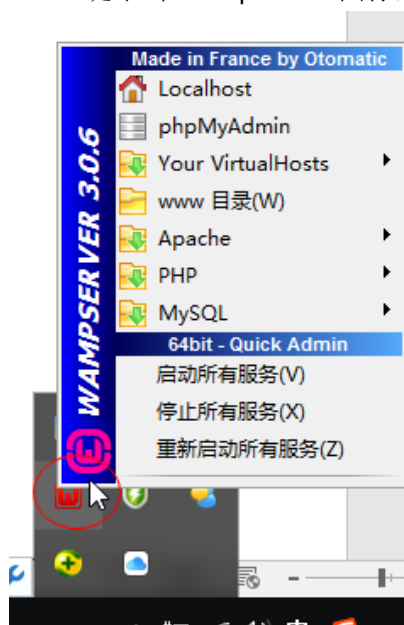
MySQL 是一种 C/S 结构：客户端和服务端，服务端对应的软件：mysqld.exe

Mysql 服务端架构有以下几层构成：

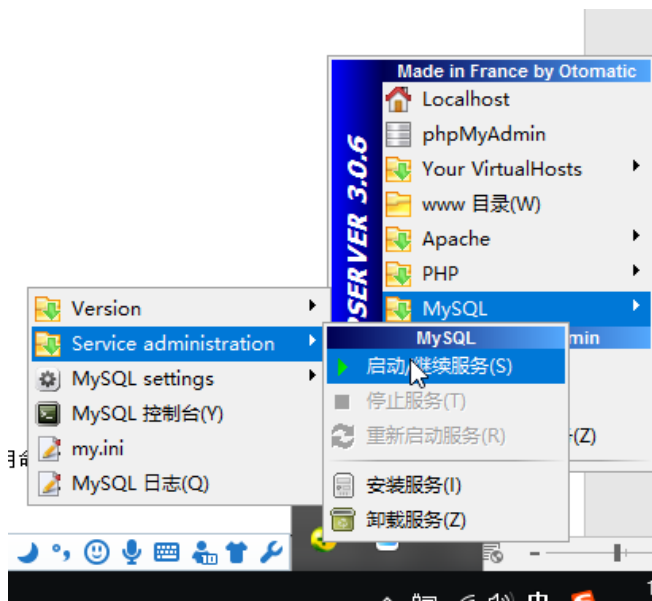
- 1、数据库管理系统（最外层）：DBMS，专门管理服务器端的所有内容
- 2、数据库（第二层）：DB，专门用于存储数据的仓库（可以有很多个）
- 3、二维数据表（第三层）：Table，专门用于存储具体实体的数据
- 4、字段（第四层）：Field，具体存储某种类型的数据（实际存储单元）

3.2.2 通过 wamp 控制台启动 MySQL

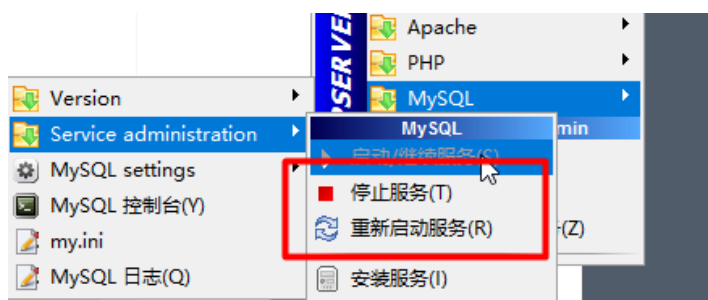
- 左键单击 wamp server 图标，弹出如下菜单



- 选择 MySQL → Service administration → 启动继续服务



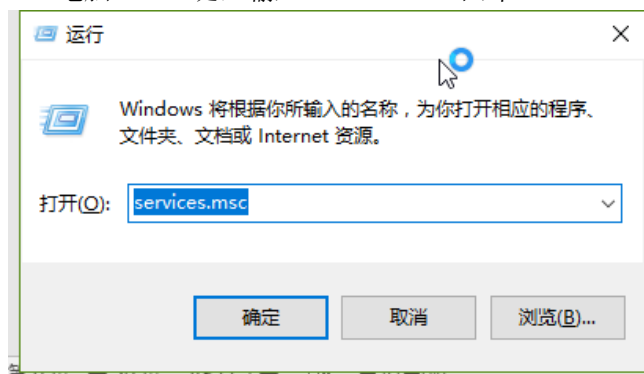
启动成功后，状态如下：



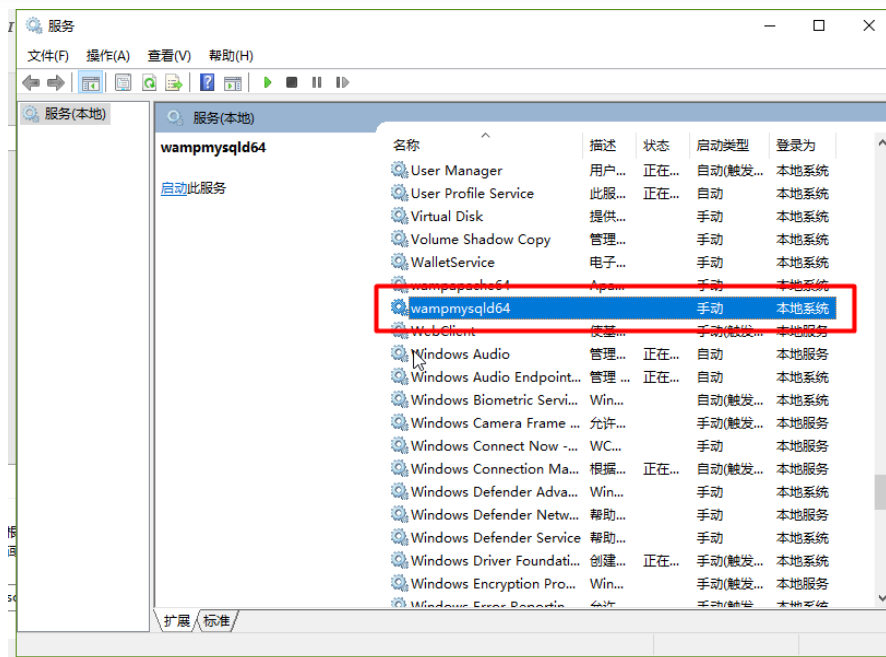
3.2.3 通过“服务”启动 MySQL

通过 Windows 下打开 cmd 命令窗口，然后使用命令进行管理

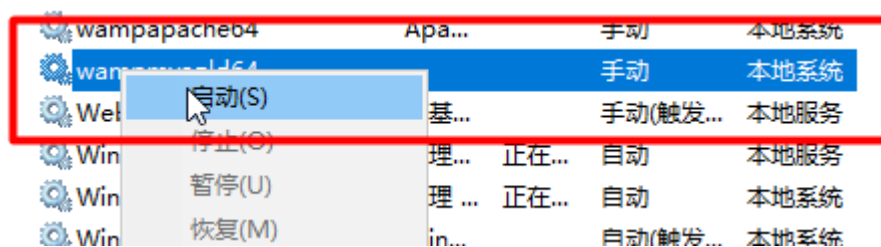
- 电脑 win+r 键，输入 services.msc 回车



打开系统服务管理器，并找到 wampmysqld64 (MySQL) 服务



- 右键→启动,启动服务即可



3.3 操作 MySQL 的方式

操作 MySQL 数据库有两种方式:

1. 在命令行中直接输入 SQL 语句 (下次课会学习)
2. 使用图形界面管理数据库, 如: phpMyAdmin、Navicat for MySQL 等。

以下学习如何使用 phpMyAdmin 图形界面来管理数据。

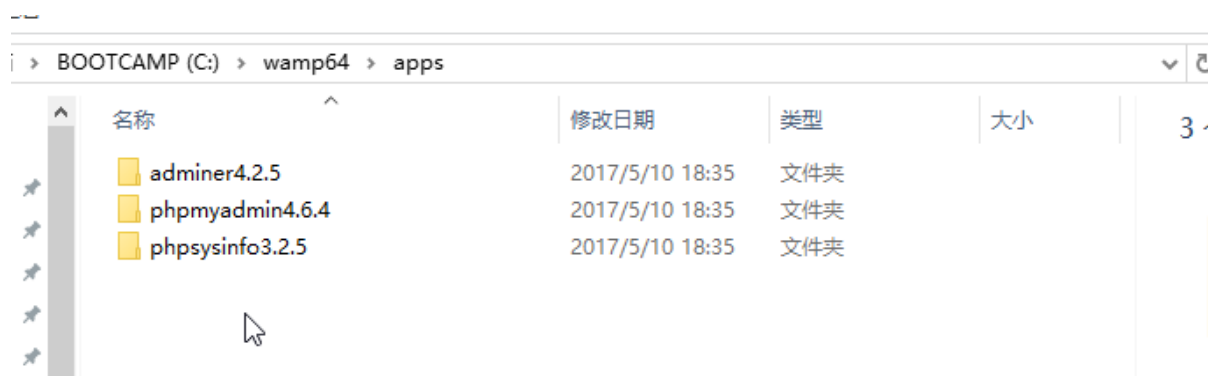
4. 使用 phpMyAdmin

4.1 基本介绍



phpMyAdmin, 简称 PMA 是一个以 PHP 为基础, 以 Web-Base 方式架构在网站主机上的 MySQL 的数据库管理工具, 让管理者可用 Web 接口管理 MySQL 数据库。

phpMyAdmin 在安装 wampserver 的时候已经集成安装了, 具体位置在
C:\{安装路径}\wamp64\apps

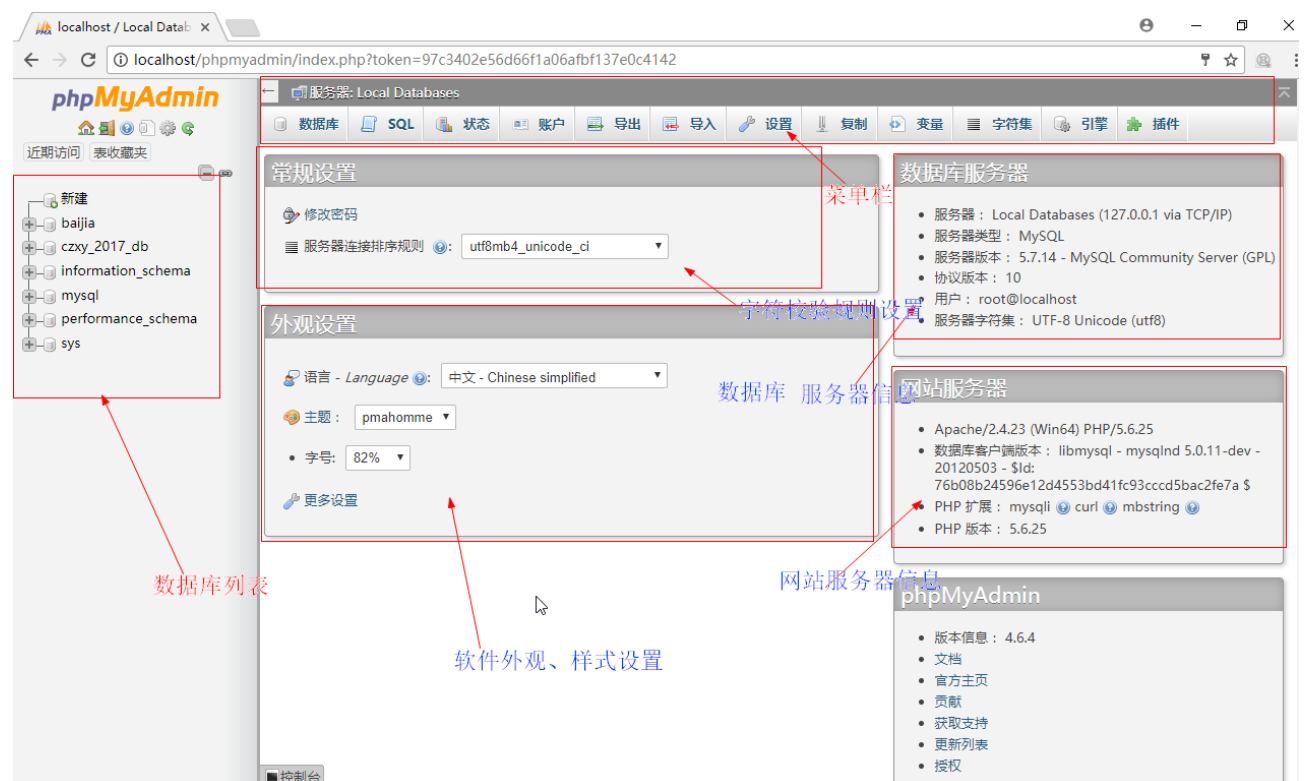


4.2 开始使用 phpMyAdmin

注意: 因为 phpMyAdmin 是使用 php 开发, 需要 php 运行环境, 所以使用之前必须开启 apache 服务



输入用户名为: root 密码: 默认为空 (不用写), 点击执行即可登录成功



4.3 创建数据库

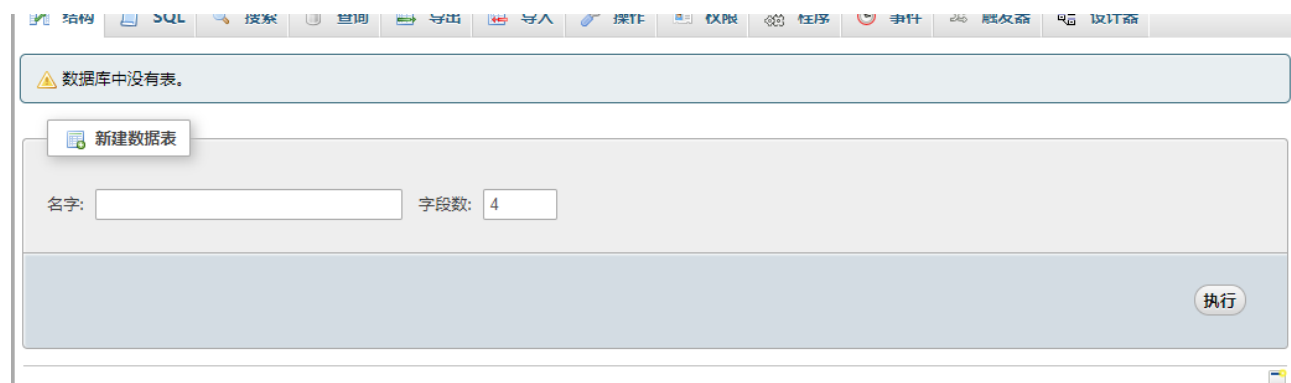
如果要保存数据 (学生信息表) 到我们的数据库中

- 1、创建一个数据库, 名称为: mydb
- 2、在数据库中创建一张数据表, 名称为: student
- 3、在表中创建字段信息 (姓名、性别、年龄...)

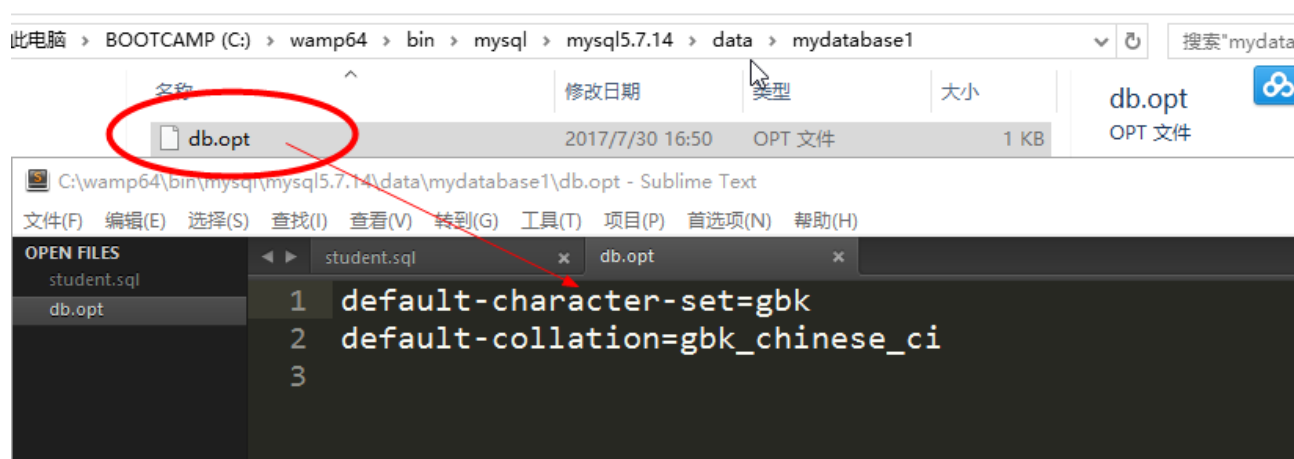
4、插入一条一条的记录（行）



创建成功后显示如下：



创建完成后，会在 mysql 下的 data 目录生成数据库相关的文件



练习 1：创建一个名为 czxy 的数据库，设置编码为 utf8_general_ci

字符集

练习 2：创建一个名为 mydb 的数据库，设置编码为 gbk_chinese_ci

字符集

4.4 创建数据表

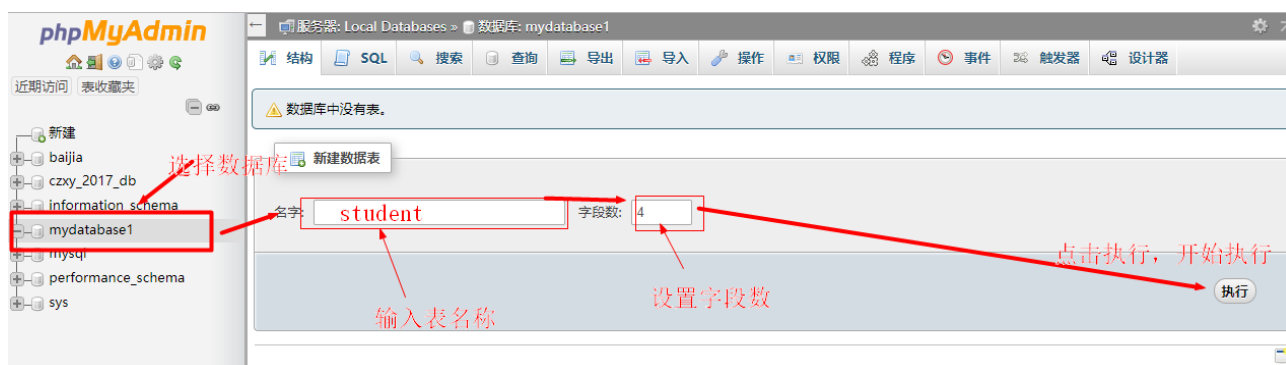
编号	姓名	性别	年龄	电话	家庭住址
1	张三丰	男	12	138 3838 1438	花果山 1 号坑
2	黑山老妖	女	10	138 3838 1438	花果山 2 号坑
3	东方不败	男	10	138 3838 1438	花果山 3 号坑

第一步：点击选择数据库

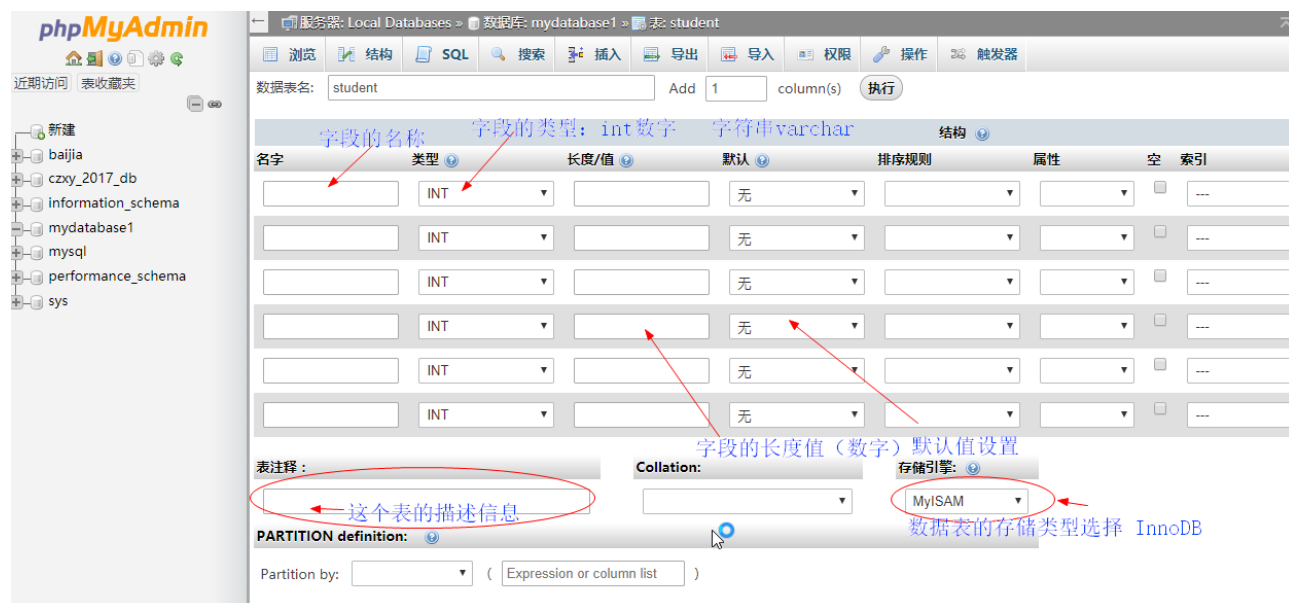
第二步：输入数据表的名称，如：student，表示这个表中存放的都是学生的相关信息

第三步：设置字段数，如 4，表示有 4 个字段（Feild）

第四步：点击“执行”按钮，开始执行



4.5 为表设置字段



字段的名称 字段的类型: int 数字 字符串 varchar

字段的长度值 (数字) 默认值设置

这个表的描述信息

数据表的存储类型选择 InnoDB

设置内容如下:

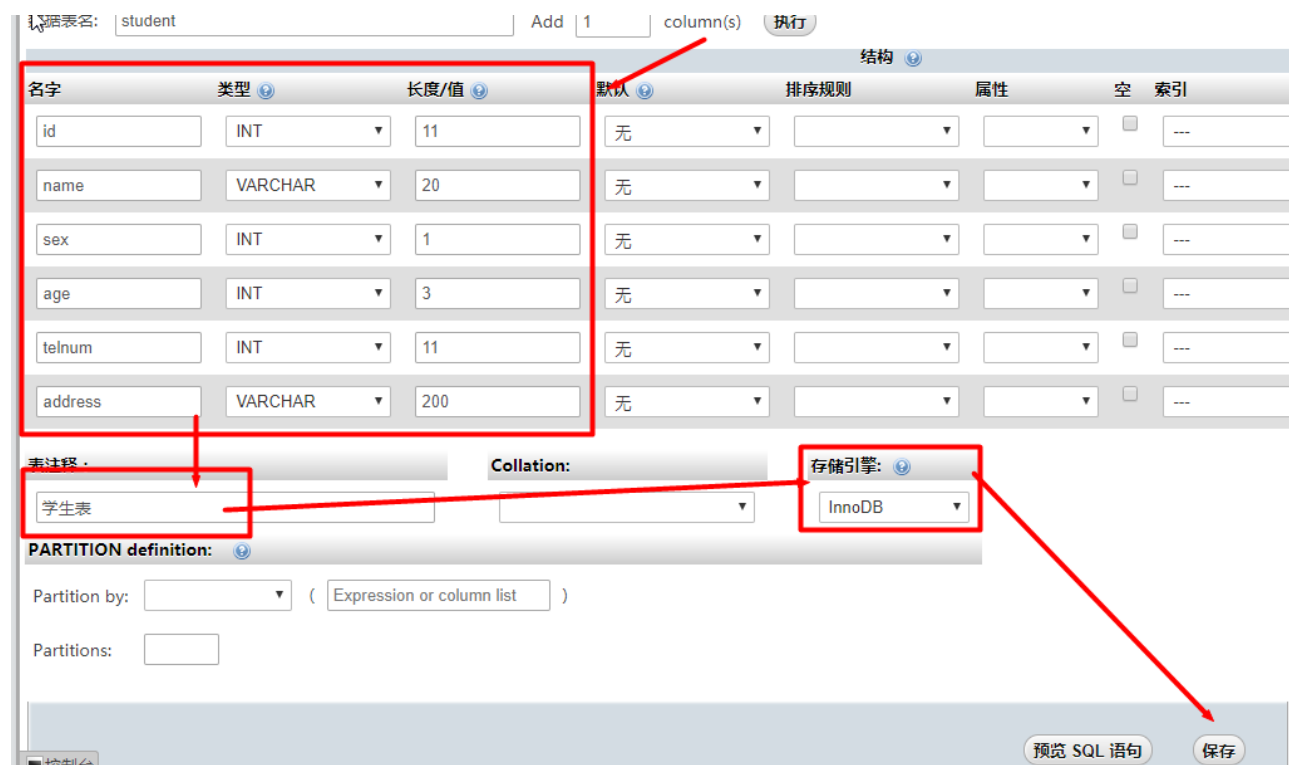


Table comment: 学生表

Storage engine: InnoDB

保存

创建完成后的效果:

#	名字	类型	排序规则	属性	空	默认	注释	额外	操作
1	id	int(11)			否	无			修改 删除 主键 唯一 索引 空间 全文搜索 更多
2	name	varchar(20)	gbk_chinese_ci		否	无			修改 删除 主键 唯一 索引 空间 全文搜索 更多
3	sex	int(1)			否	无			修改 删除 主键 唯一 索引 空间 全文搜索 更多
4	age	int(1)			否	无			修改 删除 主键 唯一 索引 空间 全文搜索 更多
5	telnum	int(11)			否	无			修改 删除 主键 唯一 索引 空间 全文搜索 更多
6	address	varchar(200)	gbk_chinese_ci		否	无			修改 删除 主键 唯一 索引 空间 全文搜索 更多

☐ 全选 选中项: ☐ 浏览 ☐ 修改 ☐ 删除 ☐ 主键 ☐ 唯一 ☐ 索引

打印 规划表结构 移动字段 Improve table structure

添加 1 个字段 于 address 之后 执行

4.6 修改表结构

localhost/phpmyadmin/tbl_structure.php?db=mydb&table=student&token=8a0ab9c181965565274565e9156dd7b7

phpMyAdmin

服务器: Local Databases » 数据库: mydb » 表: student "学生表"

浏览 结构 SQL 搜索 插入 导出 导入 权限 操作 触发器

表结构 关联视图

#	名字	类型	排序规则	属性	空	默认	注释	额外	操作
1	id	int(10)			否	无			修改 删除 主键 唯一 索引 空间 全文搜索
2	name	varchar(30)	utf8_general_ci	是	NULL				修改 删除 主键 唯一 索引 空间 全文搜索
3	sex	varchar(3)	utf8_general_ci	是	NULL				修改 删除 主键 唯一 索引 空间 全文搜索
4	age	int(10)		是	NULL				修改 删除 主键 唯一 索引 空间 全文搜索
5	telnum	bigint(11)		是	NULL				修改 删除 主键 唯一 索引 空间 全文搜索
6	address	varchar(200)	utf8_general_ci	是	NULL				修改 删除 主键 唯一 索引 空间 全文搜索

☐ 全选 选中项: ☐ 浏览 ☐ 修改 ☐ 删除 ☐ 主键 ☐ 唯一 ☐ 索引

打印 规划表结构 移动字段 Improve table structure

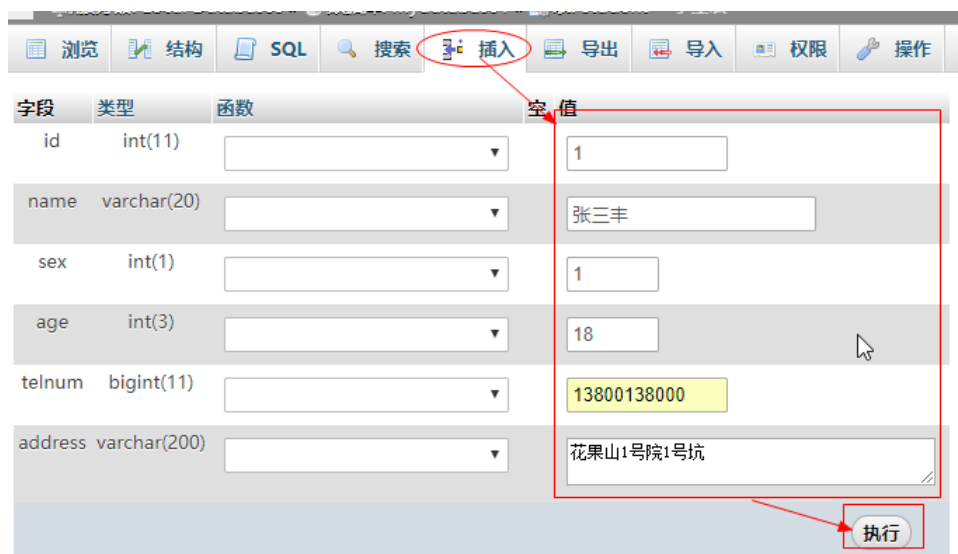
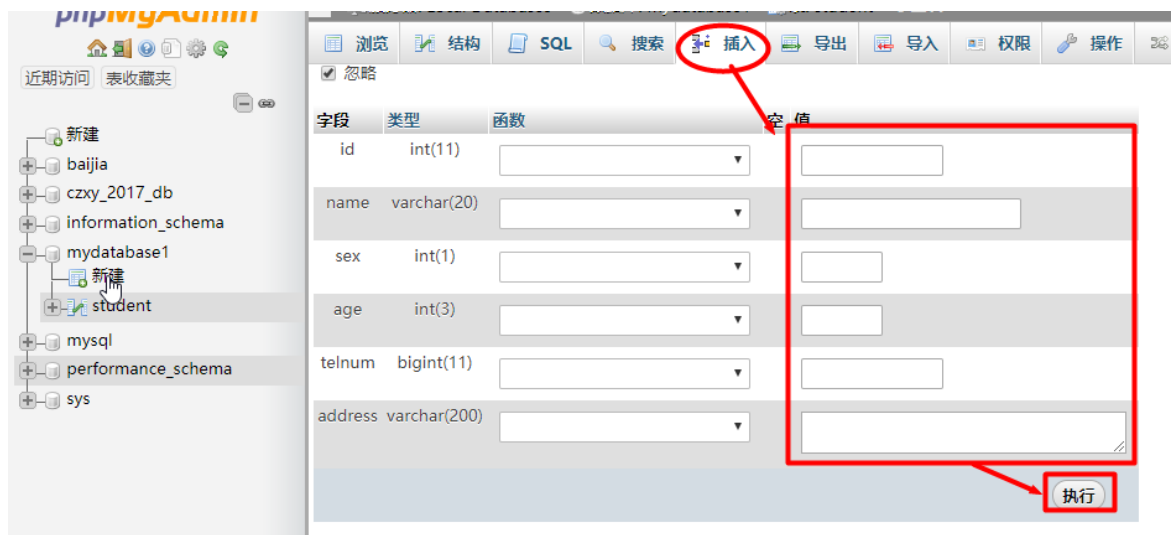
添加 1 个字段 于 address 之后 执行

4.7 插入数据

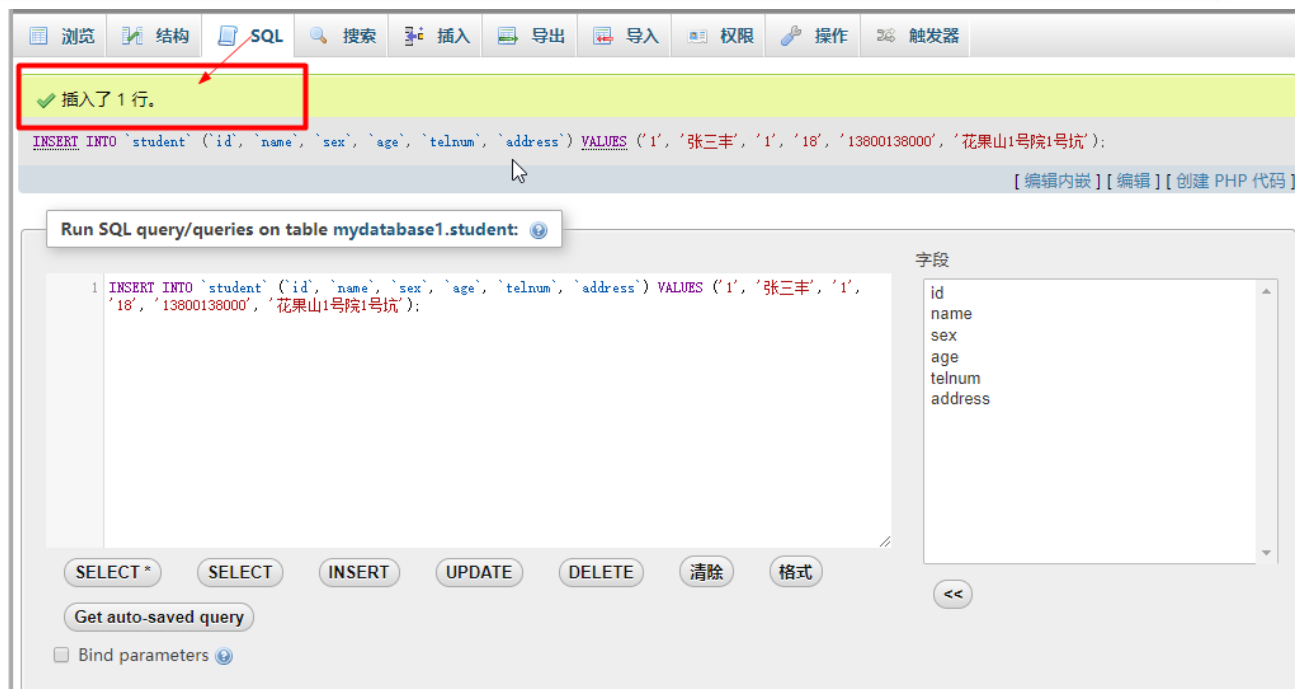
第一步：点击插入按钮：



第二步：填写字段对应的值

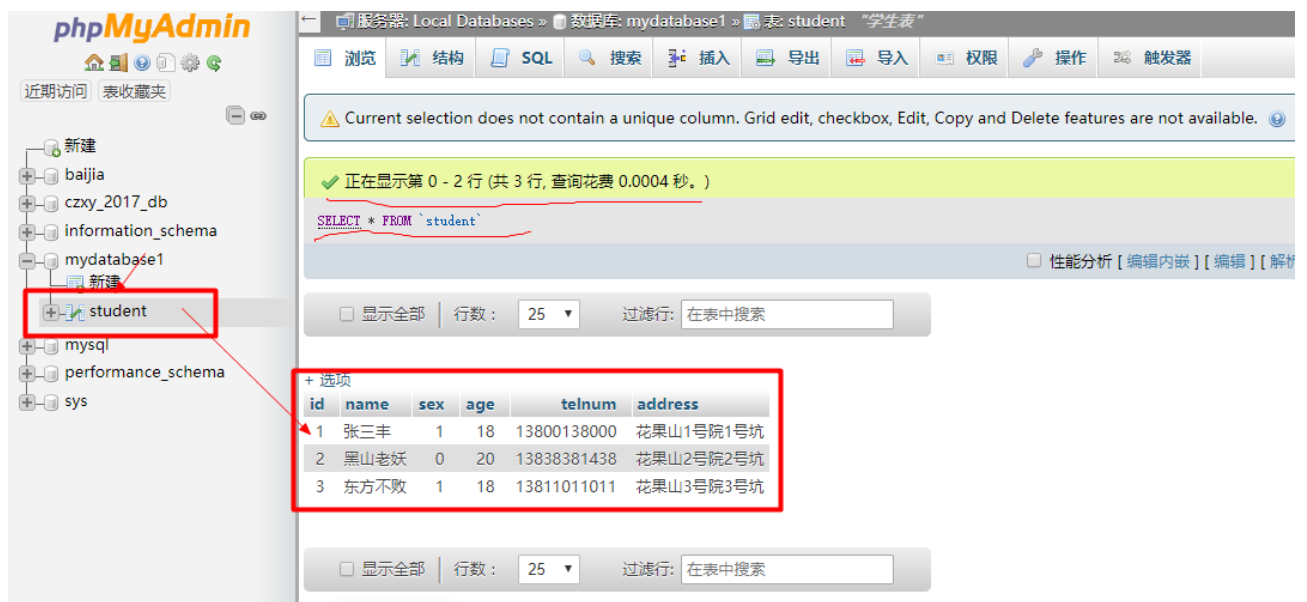


第三步：保存



4.8 查看数据

- 在左侧导航，点击表名称则可以查询表中的数据

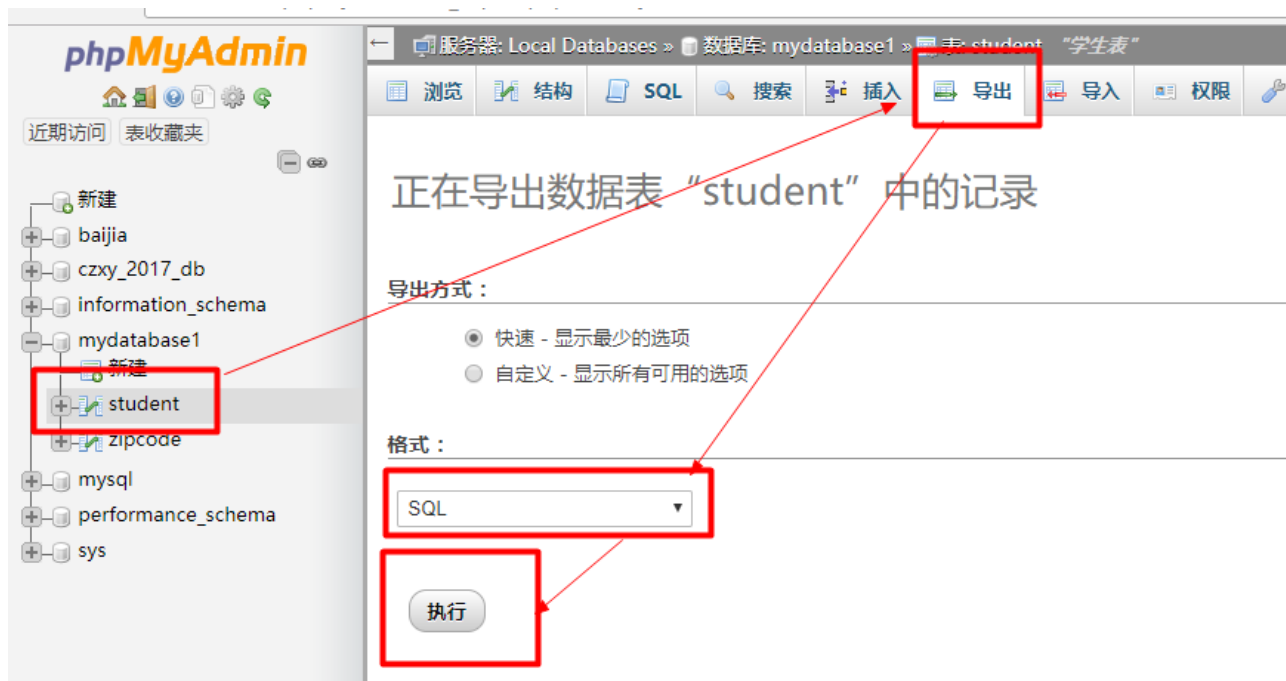


- 数据比较多的时候，可以过滤以便显示所需要的数据

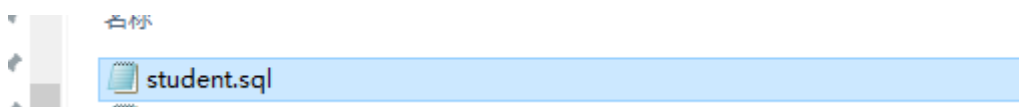


4.9 导出数据

导出数据的主要目的是进行数据备份，确保数据安全。



导出文件后缀为.sql 的数据库脚本文件



文件内容就是数据库指令



```
student.sql x
1  -- phpMyAdmin SQL Dump
2  -- version 4.6.4
3  -- https://www.phpmyadmin.net/
4  --
5  -- Host: 127.0.0.1
6  -- Generation Time: 2017-07-30 12:45:01
7  -- 服务器版本: 5.7.14
8  -- PHP Version: 5.6.25
9
10 SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
11 SET time_zone = "+00:00";
12
13
14 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
15 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
16 /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
17 /*!40101 SET NAMES utf8mb4 */;
18
19 --
20 -- Database: `mydatabase1`
21 --
22
23 -- -----
24
25 --
26 -- 表的结构 `student`
27 --
28
29 CREATE TABLE `student` (
30   `id` int(11) NOT NULL,
31   `name` varchar(20) NOT NULL,
32   `sex` int(1) NOT NULL,
33   `age` int(3) NOT NULL,
34   `telnum` bigint(11) NOT NULL,
35   `address` varchar(200) NOT NULL
36 ) ENGINE=InnoDB DEFAULT CHARSET=gbk COMMENT='学生表';
37
38 --
39 -- 转存表中的数据 `student`
40 --
41
42 INSERT INTO `student` (`id`, `name`, `sex`, `age`, `telnum`, `address`) VALUES
43 (1, '张三丰', 1, 18, 13800138000, '花果山1号院1号坑'),
44 (2, '黑山老妖', 0, 20, 13838381438, '花果山2号院2号坑'),
45 (3, '东方不败', 1, 18, 13811011011, '花果山3号院3号坑');
46
47 /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
```

```
-- phpMyAdmin SQL Dump
-- version 4.6.4
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Generation Time: 2017-07-30 12:45:01
-- 服务器版本: 5.7.14
```




```
-- PHP Version: 5.6.25

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Database: `mydatabase1`
--

--
-- -----
--
-- 表的结构 `student`
--

CREATE TABLE `student` (
  `id` int(11) NOT NULL,
  `name` varchar(20) NOT NULL,
  `sex` int(1) NOT NULL,
  `age` int(3) NOT NULL,
  `telnum` bigint(11) NOT NULL,
  `address` varchar(200) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=gbk COMMENT='学生表';

--
-- 转存表中的数据 `student`
--

INSERT INTO `student` (`id`, `name`, `sex`, `age`, `telnum`, `address`) VALUES
(1, '张三丰', 1, 18, 13800138000, '花果山 1 号院 1 号坑'),
(2, '黑山老妖', 0, 20, 13838381438, '花果山 2 号院 2 号坑'),
(3, '东方不败', 1, 18, 13811011011, '花果山 3 号院 3 号坑');

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

4.10 导入数据

一条一条的输入比较麻烦，可以使用导入功能，批量添加数据

实例：导入全国所有的省市县的数据

- 选择数据库，并点击菜单上的“导入”



- 选择.sql 文件



- 点击执行按钮，开始执行



- 导入成功!

✓ 导入成功, 执行了 11 个查询。 (student.sql)

✓ MySQL 返回的查询结果为空 (即零行)。 (查询花费 0.0007 秒。)

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO"
```

[编辑内嵌] [编辑] [创]

✓ MySQL 返回的查询结果为空 (即零行)。 (查询花费 0.0003 秒。)

```
SET time_zone = "+00:00"
```

[编辑内嵌] [编辑] [创]

✓ MySQL 返回的查询结果为空 (即零行)。 (查询花费 0.0003 秒。)

```
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */
```

[编辑内嵌] [编辑] [创]

✓ MySQL 返回的查询结果为空 (即零行)。 (查询花费 0.0003 秒。)

```
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */
```

[编辑内嵌] [编辑] [创]

✓ MySQL 返回的查询结果为空 (即零行)。 (查询花费 0.0003 秒。)

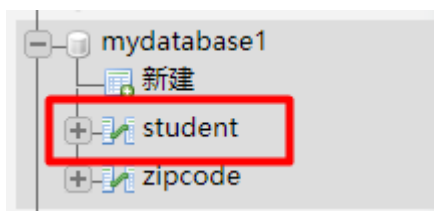
```
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */
```

[编辑内嵌] [编辑] [创]

MySQL 返回的查询结果为空 (即零行)。 (查询花费 0.0003 秒。)

控制台

导入成功后, 数据表显示在数据库中

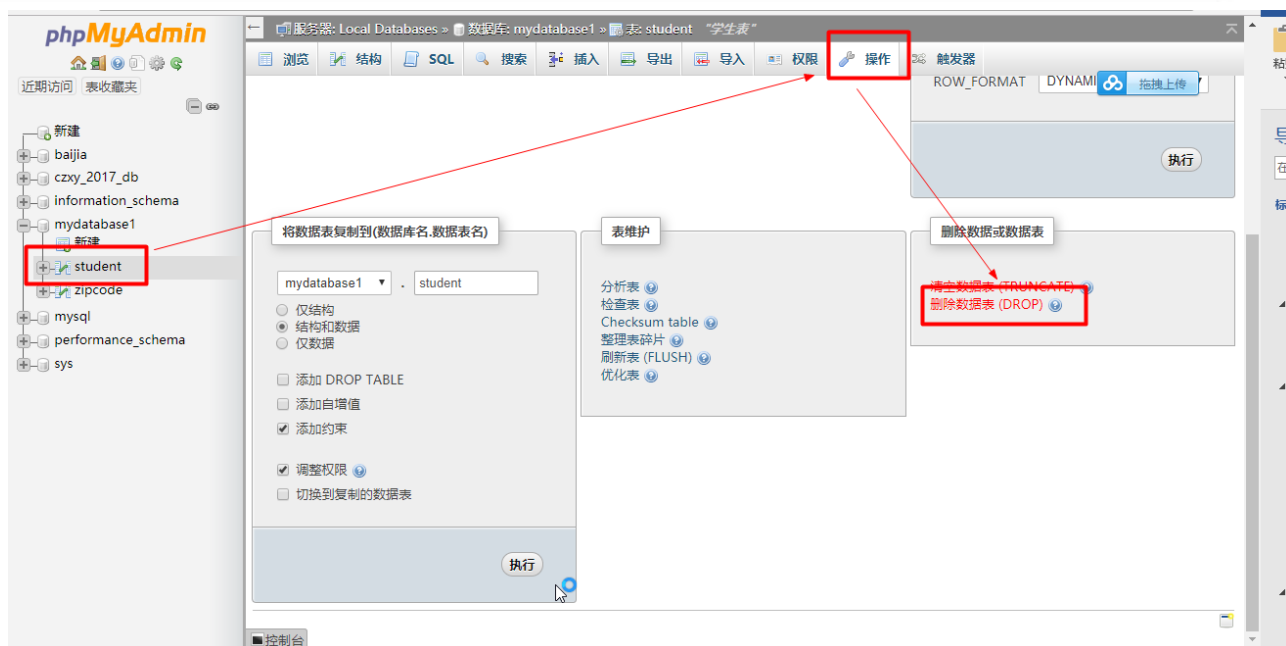


4.11 删除数据表

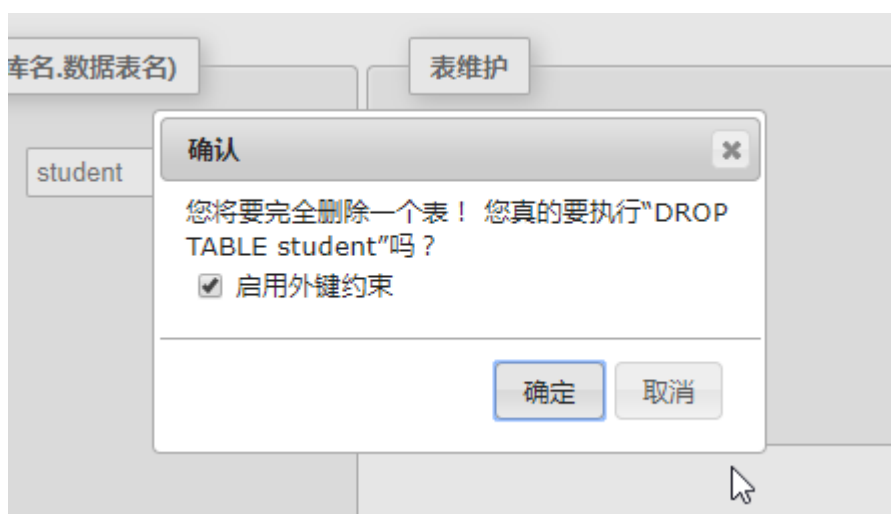
如果一个数据表想要进行删除, 可以使用如下方法进行删除

清除数据表: 清空表中的数据, 但是结构还在 (慎用)

删除数据表: 清空表中的数据, 同时删除表结构 (慎用)



弹出提示，确认是否真的要删除



点击确定即可删除！

4.12 删除数据库

删除数据库：同时删除数据库中的所有表结构及表中的数据（非常危险，慎用！慎用！慎用！）

5. 数值类型

5.1 整数类型

整型类型名称及表示的范围

类型	存储	最小值	最大值
	(Bytes)	(Signed/Unsigned)	(Signed/Unsigned)
TINYINT	1	-128	127
		0	255
SMALLINT	2	-32768	32767
		0	65535
MEDIUMINT	3	-8388608	8388607
		0	16777215
INT	4	-2147483648	2147483647
		0	4294967295
BIGINT	8	-9223372036854775808	9223372036854775807
		0	18446744073709551615

5.1.1 通过 tinyint 看数据长度问题

tinyint 表示迷你整型，字段长度默认为 4，表示数字范围：-127 ~ 128

字段 类型 函数 空 值

int_1	tinyint(4)			255	请输入一个小于或等于 127 的值
int_2	smallint(6)			1000	
int_3	mediumint(9)			10000	
int_4	int(11)			100000	
int_5	bigint(20)			1000000	
int_6	tinyint(4)			-255	请输入一个大于或等于 -128 的值

tinyint

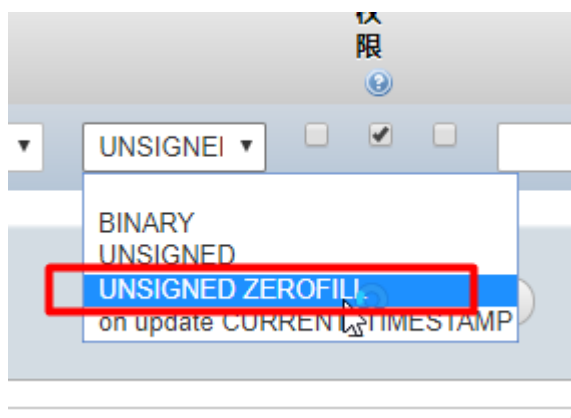
执行

-127 含“-”，总长度为 4，可以通过设置属性为 unsigned 来查看、验证。

#	名字	类型	排序规则	属性	空	默认	注释	额外	操作
1	int_1	tinyint(4)			是	NULL			修改 删除 主键 U 唯一 索引 空间 全文搜索 非重复值 (DISTINCT)
2	int_2	smallint(6)			是	NULL			修改 删除 主键 U 唯一 索引 空间 全文搜索 非重复值 (DISTINCT)
3	int_3	mediumint(9)			是	NULL			修改 删除 主键 U 唯一 索引 空间 全文搜索 非重复值 (DISTINCT)
4	int_4	int(11)			是	NULL			修改 删除 主键 U 唯一 索引 空间 全文搜索 非重复值 (DISTINCT)
5	int_5	bigint(20)			是	NULL			修改 删除 主键 U 唯一 索引 空间 全文搜索 非重复值 (DISTINCT)
6	int_6	tinyint(4)			是	NULL			修改 删除 主键 U 唯一 索引 空间 全文搜索 非重复值 (DISTINCT)

5.1.2 关于整型的补零问题

zerofill 0 填充



5.1.3 整型数据的常见应用场景

保存人的年龄（1-100）：tinyint
保存某个状态值（0、1）：tinyint
小型项目的编号：int

5.1.4 关于 id 字段的自增 auto_increment 问题

第一步：点击表名，并点击结构



第二步：设置表的主键为 id

第三步：设置主键为自动增长



然后，勾选 A_I 复选框，点击保存即可。



