# 2018.04.03

## 关系型数据库：

使用二维表的结构，去实现实体信息与实体之间的联系。

## 常用的数据库术语：

database 数据库

table表

SQL: structured query language结构化查询语言

row行

column列

record记录

field字段

relational关系型

## 命令行管理

检查端口占用DOS命令：netstat -ano

mysql -hlocalhost -uroot -proot

## MySQL数据库基础指令

### 创建数据库：create database dbName;

查看建库信息：show create database dbName;

### 查看数据库：show databases;

### 删除数据库：drop database dbName;

### 修改数据库：alter database dbName;

### 使用数据库：use dbName;

名字不能是关键字，严格区分大小写（除非用了反引号``,不建议使用这种方式）

## MySQL数据表基础指令

### 创建数据表：create table tableName

create table tableName(

field1 type1 propsAdj1,

field2 type2 propsAdj2

)charset=utf-8;

注意：可以跟上字符集，但是默认是随数据库。

create table student (

id int primary key(主键) auto\_increment

name varchar(20) not null,

age tinyint,

gender char(1)

);

### 查询数据表：show tables

desc (description) tableName 查看数据结构

show create table tableName 查看建表语句

### 修改数据表：alter table tableName 配置=值

alter table student charset=utf8 修改数据表编码

如果要写入中文数据，建议使用set names gbk进行编码

### 删除数据表：drop table tableName

## 记录操作

### 增加记录 insert into tableName (field,list) values (value,list);

如果使用insert tableName values(default,all value,list) 必须给id一个占位符null 或者default

### 删除记录delete from tableName; //全部清空

### 查询记录select field\_list from tableName where condition

select \* from tableName; 某个表的所有字段

select \* from tableName,anotherTableName; 多个表的所有字段

select name,gender from tableName 指定字段

select name as 别名, gender from tableName 给指定字段设置别名

where子句

= select \* from tableName where name=’xx’;

>

<

>=

<=

in 是否在集合中

select \* from tableName where name in (‘this’,’this’,’this’);

and 并且

select \* from tableName where age>18 and id<4;

or 或者

select \* from tableName where age>18 or id<4;

like 模糊查询

select \* from tableName where name like ‘X%’;

其中 %匹配个或者多个字符 \_匹配1个字符

order by 子句

order by field asc 升序

order by field desc 降序

order by field asc, field desc 按第一个字段的升序，第二个字段的降序

select \* from tableName order by desc;

limit

limit n; 限制输出N条数据

limit n,m; 跳过n条，限制输出M条

添加字段：

alter table student add column classid int default 1; 需要跟上数据类型和默认值。

### 删除数据记录delete from tableName where condition

delete from student where id=7; //id是唯一的标志。

注意：删除和修改数据都会对数据造成不可恢复的影响，所以一定要谨慎。

### 修改记录：update tableName set field1 = new value1, field2 = new value2 where id=num;

# 2018.4.4

## PHP：hypertext preprocessor超文本预处理器。

开源脚本语言。是服务器端用来生成HTML代码的编程语言，PHP是处理PHP编程语言的一个软件（超文本预处理器）。适用于中小型项目的web开发领域。

## PHP的优势

开源免费

较好的安全性

跨平台（支持所有流行的数据库及操作系统）

效率高 易用性好 高效 PHP消耗非常少系统资源

可以用C C++进行程序扩展

面向过程/面向对象

## PHP语法

### 找到编写代码的位置

F:\itsource\software\php\php\_teacher\WWW

声明：

<?php ....?>

语句结束：

;

申明头信息：

header(‘Content-Type:text/html; charset=utf-8’);

注释：

/\*\*/ //

变量：

$varName=value; 定义变量

$varName=new value; 修改变量

unset($varName); 删除变量

isset($varName); 检测变量是否定义，即使没有定义不会报错

var\_dump(isset($varName)/$varName); 调试输出变量的值和类型

echo $varName 输出变量值

拼接变量不用+而是用点.

预定义变量：

$\_GET[‘parasName’] 接收GET方式传的值

$\_POST[‘parasName’] 接收POST方式传的值

$\_SERVER 输出系统的所有的服务器端或客户端的内置信息

var\_dump($\_SERVER);

$\_SESSION 保存用户会话信息

数据类型：

integer

double

string

resource

null

### 流程控制语句：

条件判断：if(contion) {...}else{....}

状态分支：switch(status){

case value1:....break;

case value2:....break;

case value3:....break;

case value4:....break;

default:

}

for循环

for($i=0,$i<coun($arr), i++){....}

while循环

while(condition){...}

break,condition

break 中断语句块执行

continue 中断当前循环，执行下一次循环

### 函数

定义函数

function fnName(形参){

return...

}

函数调用

fnName(实际参数);

系统函数

PHP提供了非常多系统函数，可以直接使用不用定义

count() 统计数组的元素个数

ceil() 向上取整

mysqli\_query() 将sql语句发送到MySQL服务器执行，并将执行的结果作为函数的返回值。

### 数组

数组的分类:

根key的类型进行分类：

关联数组：key是一个字符串

$arr=[‘name’=>’zhangsan’, ‘age’=>19];

$arr=[‘name’];

索引数组：key是一个数字

$arr=[‘zhangsan’, ‘18’];

$arr[0];

根据值的类型进行划分

一维数组：

$arr=[[‘zhangsan’,18],[‘sam’,20]];

多维数组：

$arr=[[‘zhangsan’,18],[‘sam’,20]];

$arr[0][0];

数组的遍历

for($i=0,$i<coun($arr), i++){$arr[i]}

foreach($arr as key=>value){...}

文件加载

include(url);

require(url);

php操作数据库

链接数据库

$conn=mysqli\_connect(‘host’,’username’,’password’,’dbname’);

发送查询命令到MySQL

$results=mysqli\_query($conn,’sql语句’);

结果集处理

读取数据----关联数组

$row=mysqli\_fetch\_assoc($results); //读取到的结果是一行数据（一维数组）

读取数据----索引数组

$row=mysqli\_fetch\_row($results);

读取数据----关联数组+索引数组

$row=mysqli\_fetch\_arry($results);

获取错误提示

mysqli\_error($conn);

获取记录总条数

mysql\_num\_rows($results);

释放结果集

mysqli\_free\_result($results);

# 超市后台项目

## 网站的基本概念：

前台：任何人都可以直接打开网站查看信息的平台

后台：管理员通过登陆实现身份验证以后，对网站进行管理的平台

前端：前端是指浏览器端，所有在浏览器端执行的程序都是前端程序，从事前端程序的人叫前端程序员，前端技术： HTML CSS JS Jquery AJAX VUE ElementUI ...

后端：后端是指服务器端，所有在服务器端挂靠的程序都是后端程序，从事后端程序的人也称为后端程序员，后端技术：Node.js PHP JSP Python Mysql Mongodb...

## 数据库分析：

数据库名： market

create database market;

数据表： admin(id,username,password,group)

create table admin(

id int primary key auto\_increment,

username varchar(20),

password varchar(32),

usergroup varchar(20)

);

insert into admin(username,password,usergroup) values('admin','123456','超级管理员');

insert into admin(username,password,usergroup) values('guest','000000','超级管理员');

## 开发步骤：

1. 实现验证用户名和密码是否正确

### 1) VUE数据双向绑定

v-model="变量名"

### 2) VUE绑定事件

在登陆按钮上设置属性 v-on:click="事件处理函数"

在Vue实例中设置 methods属性，定义事件处理函数

### 3) 发起AJAX请求

获取用户输入的用户名和密码 this.username this.password

构造 AJAX请求，并且将用户名和密码数据发送到服务器端 /api/checklogin.php

### 4) 在服务器端创建文件 /api/checklogin.php

连接数据库

接收数据

验证数据有效性

### 5) 构造SQL语句查询用户名和密码是否存在

用户名和密码是 并且关系

### 6) 将查询的结果返回给客户端

mysqli\_num\_rows()

有数据：登陆成功

无数据：登陆失败

### 7) 在浏览器端将结果显示出来

1: 登陆成功

0: 登陆失败

导出数据库：

mysqldump -uroot -proot 数据库 > xxx.sql

## session技术

session是一种会话控制，不同于http请求，实现用户数据在服务器进行多文件之间共享。session是服务器端技术，基于cookie，由服务器端进行设置和保存，客户端以cookie形式存储标识phpsessid

理解：session常用与保存登陆状态，可以在多文件之间共享

session操作

session\_start(); 开启session机制

$\_session[‘name’]=value; 赋值

$\_session[‘name’] 查询

$\_session[‘name’]=value; 修改