Php 与 MySQL

1.服务器的概念

C/S架构：client/Server

                    (客户机/服务器)结构，是大家熟知的软件系统体系结构，通过将任务合理分配到Client端和Server端，降低了系统的通讯开销，

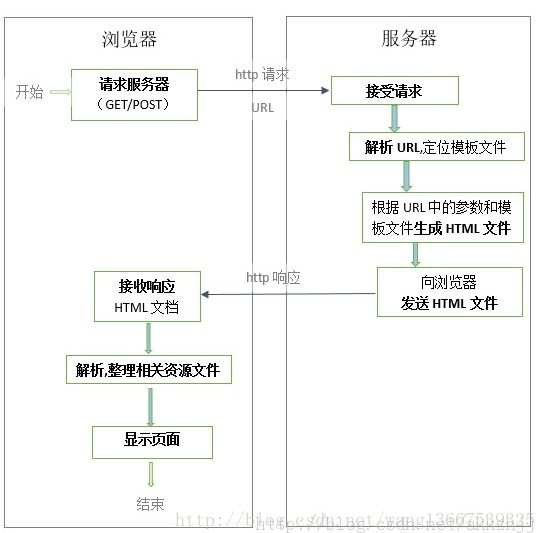
                    可以充分利用两端硬件环境的优势。 开发C/S架构可以采用多种语言，包括Java,C++,C#，以及Dephi等

B/S架构: Browser/Server

                    即Browser/Server(浏览器/服务器)结构，是随着技术Internet的兴起，对C/S结构的一种变化或者改进的结构。在这种结构下，

                    用户界面完全通过WWW浏览器实现。前端在大部分企业就是做浏览器端展现相关的工作，会用到html,css,js,ps,ai等等，而“后端”的主要工作室程序开发、

                    数据处理了，比如：php，asp，mysql，mssql。



2.PHP介绍

\* PHP是服务器端语言

\* PHP优点:

    \*    跨平台 ,同时支持多种数据库

    \*    安全性和效率好

    \*    使用成本低 ( linux   apache   mysql   PHP内核)

    \*    相对jsp ,和 asp.net 简单

    \*    开发源码(可以做二次开发) / 开源软件

    \*    在PHP4 PHP5 以及即将发布PHP6 中更好的支持 面向对象

\* 缺点:

    \*   安装比较复杂，配置比较多

    \*   太灵活，解释执行(编译执行),所以有些错误，到真的运行的时候才会暴露

    服务器端除了PHP还有，JSP（java），ASP，python，nodeJS，go等等。

    PHP：personal home page；

PHP环境搭建

\* phpStudy(PHP运行环境一键安装包)

   phpStudy是搭建服务器的工具，我们需要把php的代码放在phpStudy工具里的“www”目录下，“www”目录就是服务器的目录，即放置项目代码的目录。

   如：把index.html文件拷贝至www下，则用服务器的方式打开index.html的做法是：在浏览器中输入http://ip地址/index.html

假如：

  www的全路径是：D:\phpStudy\WWW，

  ip地址是：10.35.165.10/文件名

第一个HelloWorld

\* PHP的简单了解

    \* 文件扩展名  php

    \* 代码写在<?php   与  ?>之间

    \* PHP代码每句话以分号结束

    \* PHP的注释：

    \* 单行用  //  或 #  /\*内容\*/

如： <?php

        echo "hello,world";

     ?>

多行用 /\*\*/

环形地址：本机IP

http://localhost/testPhp/HelloWorld.php

http://127.0.0.1/testPhp/HelloWorld.php

//编码格式

中文问题：header("Content-type:text/html;charset=utf-8");

PHP语法：

\* php中，定义一个变量要以$符号打头，变量名区分大小写

\* php 的变量的数据类型，是变化的，php变量的数据类型是由运行时的上下文决定。

  如：$age = 250;

  如:   $a=90; //相当于我们定义了一个变量 $a,并赋值90

        $a=”hello”; //表示$a 的类型变化.

  数据类型：

  字符串的连接用.

  echo "Hello World" ."</br>". "大黄";

\* PHP的流程控制

\* 顺序控制(从上到下，从左到右)

\* 分支控制

 if(条件表达式){

   //n多语句;

 }else{

       //n多语句;

 }

 switch分支语句

 switch(表达式){

   case 常量1://n多语句;break;

   case 常量2://n多语句break;

   default:   //n多语句;break;

 }

    $a = 123;

    $b = 456;

    $c = 0;

    if($a>$b){

        $c = $a;

    }else{

        $c = $a;

    }

    echo $c;

\* PHP的流程控制

    \* 循环：（break，continue）

    for(循环初值; 循环的条件; 步长){

    }

    for($i=0; $i<10; $i++){

        echo "我今年".$i."岁了"."</br>";

    }

\* PHP的函数

<?php

    function add($a,$b){

        return $a+$b;

    }

    echo add(1,2);

?>

\* PHP的数组

  $arr = Array(11,22,33,44,55);

    //count 用来统计数组的长度

  for($i=0; $i<count($arr); $i++){

      echo $arr[$i]."</br>";

  }

PHP生成动态页面 .php文件中

<!DOCTYPE html>

<html>

    <head>

        <meta charset="utf-8">

        <title></title>

    </head>

    <body>

        <ul>

        <?php

            for($i=0; $i<10; $i++){

        ?>

            <li><?php echo $i ?></li>

        <?php

            }

        ?>

        </ul>

    </body>

</html>

\* PHP接收前端的数据

    \* $\_POST['参数名‘]

    \* $\_GET["参数名"]

    \* $\_REQUEST["参数名"];

    \* 请求参数有中文的解决方案：

        \* 增加以下代码：

        \* 使用iconv

如：

<form action="login.php" method="post">

            姓名:<input type="text" name="username"/></br>

            密码:<input type="password" name = "pwd"/></br>

            <input type="submit" value="登录" />

</form>

  $\_GET['username']获取username

  $\_GET['password']获取password

    header("Content-type: text/html; charset=utf-8");

<?php

    header("Content-type:text/html;charset=utf-8");

//     $name = $\_GET["username"];

//     $pwd = $\_GET["pwd"];

//    $name = $\_POST["username"];

//  $pwd = $\_POST["pwd"];

    $name = $\_REQUEST["username"];

     $pwd = $\_REQUEST["pwd"];

    echo $name.":".$pwd;

?>

3.数据库

\*  数据库概念：

\*  库：仓库

\*  表：一个仓库被分为了许多部分，很像类

\*  字段：很像类的每个属性。

\*  每个字段的数据类型：

     数据库(Database)是按照数据结构来组织、存储和管理数据的仓库。

      如：

  int ----> 整数 blob -----> 二进制数据

  varchar/char ----->字符串

  date -----> 日期

\* 常见的数据库

    \* 关系型数据库：Oracle、MySQL、SQLServer、DB2、sybase

    \* 非关系型的数据库:Redis，Tokyo Cabinet，Cassandra，Voldemort，MongoDB，Dynomite，

                      HBase，CouchDB，Hypertable， Riak，Ti，

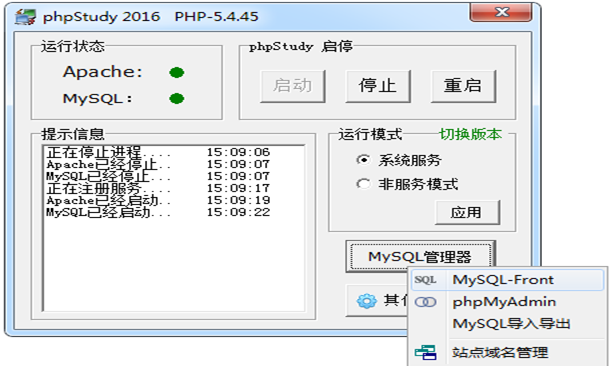
\* mySQL

     一种开源的数据库软件，市场占有率很高，现在属于Oracle公司。

\* 使用工具：

\*    点击 MySQL管理器--》选择 MySQL-Front

     phpStudy



\*  进入 SQL编辑器

    \* 创建数据库：

    \* 打开某一个库 use db20170203;

    \* 创建表：

创建数据库：

    可视化：点击鼠标右键

    代码： create database 数据库名

    如： create database db20170203

创建表：

create table Student

(

       stuId int,

       stuName varchar(20),

       stuGender char(4),

       stuAge int

)

sql语句：增删查改

create table Student

(

       stuId int,

       stuName varchar(20),

       stuGender char(4),

       stuAge int

)

--增

insert into 表名[字段1，字段2...字段N]

values(值1,值2,值N);

如：

insert into student(name, sex, age) values("孙丽华", "女", 21);

insert into Student

values(1,"大黄",666,18);

insert into Student

values(3,"小明",666,18);

insert into Student(stuId,stuName)

values(2,"laowang");

--删 直接删除整个表的内容，但是表还在

delete from 表名

where 条件;

如：

\*  delete from students; 删除表中的所有数据，

\*  delete from students where id=2; //删除表中，id是2的数据。

\*  delete from students where age<20;

delete from Student;

--整个表都删掉

drop table Student;

--where 条件 类似于if

delete from Student

where stuName = "大黄"

--OR类似于 ||

delete from Student

where stuId = 1

OR stuName = "小明";

--AND类似于&&

delete from Student

Where stuName = "大黄"

AND stuid = 1;

--改

update 表明 set 字段1=值1，字段2=值2...

WHERE 更新条件;

update student

set stuName = "啦啦啦",stuId = 12323

where stuname = "嘿嘿";

--查

  select 列名称 from 表名称 [查询条件];

\* select name, age from students;

\* select \* from students;

\* select \* from students where age > 21 ;

\* select \* from students where name like "%王%"; --模糊查询

\* select \* from students where id<5 and age>20;

如：

select stuAge,stuId,stuAge from Student;

select\*from Student;

数据库练习，要求都写一遍（作业中会有没有用到的语法，通过练习自己查询资料学会）

\* 练习

create table EMP

(

EMPNO numeric(4) PRIMARY KEY,

ENAME VARCHAR(10),

JOB VARCHAR(9),

MGR numeric(4),

HIREDATE DATE,

SAL numeric(7,2),

COMM numeric(7,2),

DEPNO numeric(4)

);

CREATE TABLE Dept(

DEPTNO numeric(4),

DNAME VARCHAR(14),

LOC VARCHAR(13)

);

CREATE TABLE Salgrade

(

GRADE numeric,

LOSAL numeric,

HISAL numeric

);

INSERT INTO Dept VALUES (10,'ACCOUNTING','NEW YORK');

INSERT INTO Dept VALUES (20,'RESEARCH','DALLAS');

INSERT INTO Dept VALUES (30,'SALES','CHICAGO');

INSERT INTO Dept VALUES (40,'OPERATIONS','BOSTON');

INSERT INTO EMP VALUES

(7369,'SMITH','CLERK',7902,'1980-12-17',800,null,20);

INSERT INTO EMP VALUES

(7499,'ALLEN','SALESMAN',7698,'1981-02-20',1600,300,30);

INSERT INTO EMP VALUES

(7521,'WARD','SALESMAN',7698,'1981-02-22',1250,500,30);

INSERT INTO EMP VALUES

(7566,'JONES','MANAGER',7839,'1981-04-02',2975,NULL,20);

INSERT INTO EMP VALUES

(7654,'MARTIN','SALESMAN',7698,'1981-09-28',1250,1400,30);

INSERT INTO EMP VALUES

(7698,'BLAKE','MANAGER',7839,'1981-05-01',2850,NULL,30);

INSERT INTO EMP VALUES

(7782,'CLARK','MANAGER',7839,'1981-06-09',2450,NULL,10);

INSERT INTO EMP VALUES

(7844,'TURNER','SALESMAN',7698,'1981-09-08',1500,0,30);

INSERT INTO EMP VALUES

(7900,'JAMES','CLERK',7698,'1981-12-03',950,NULL,30);

INSERT INTO EMP VALUES

(7902,'FORD','ANALYST',7566,'1981-12-03',3000,NULL,20);

INSERT INTO EMP VALUES

(7934,'MILLER','CLERK',7782,'1982-01-23',1300,NULL,10);

INSERT INTO SALGRADE VALUES (1,700,1200);

INSERT INTO SALGRADE VALUES (2,1201,1400);

INSERT INTO SALGRADE VALUES (3,1401,2000);

INSERT INTO SALGRADE VALUES (4,2001,3000);

INSERT INTO SALGRADE VALUES (5,3001,9999);

#要求列出每个雇员的姓名及年薪

SELECT ename,sal\*12 FROM emp;

#查看每月可以得到奖金的雇员信息

SELECT \* FROM emp WHERE comm is NOT NULL;

#要求基本工资大于1500，同时可以领取奖金的雇员信息

SELECT \* FROM emp WHERE sal>1500 AND comm is NOT NULL;

#查询在1981年雇佣的全部雇员信息，BETWEEN .. AND 包含等于的情况

SELECT \* FROM emp WHERE hiredate BETWEEN '01-JAN-81' AND '31-DEC-81';

#要求查询出雇员编号不是 7369、7499的雇员信息

SELECT \* FROM emp WHERE empno NOT IN(7369,7499);

#SQL中LIKE语句要注意通配符 % 和 \_  SELECT \* FROM emp WHERE hiredate LIKE '%81%‘;

#要求对雇员的工资由低到高进行排序，升序为默认(ASC)，降序(DESC)

SELECT \* FROM emp ORDER BY sal  desc;

#找出佣金高于薪金的60%的员工

SELECT \* FROM emp WHERE comm>sal\*0.6

#找出部门10中所有经理(MANAGER)和部门20中所有办事员(CLERK)的详细资料

SELECT \* FROM emp WHERE (deptno=20 AND job='CLERK') OR (deptno=10 AND job='MANAGER');

#找出既不是经理又不是办事员但其薪金大于或等于2000的所有员工的资料

SELECT \* FROM emp WHERE job NOT IN('MANAGER','CLERK') AND sal >= 2000;

#显示不带有"R"的员工的姓名

SELECT ename FROM emp WHERE ename NOT LIKE '%R%';

4.PHP连接MySql

mysql\_connect(servername,username,password);

参数描述

servername可选。规定要连接的服务器。默认是 "localhost:3306"。

username可选。规定登录所使用的用户名。默认值是拥有服务器进程的用户的名称。

password可选。规定登录所用的密码。默认是 ""。

<?php

    header("Content-type:text/html;charset=utf-8");

    //$conn = mysql\_connect("localhost","root","root");

    //创建数据库连接对象

    $conn = mysql\_connect("localhost","root","root");

    if($conn){

        echo "连接成功"."</br>";

        //选择数据库

        mysql\_select\_db("xa1901");

    }else{

        die("Could not connect:" . mysql\_error());

    }

        //数据库查询

    //mysql\_query(sql语句,连接对象);

    //增

    // mysql\_query("insert into student values(999,'大王','M','23')",$conn);

    //删

    //mysql\_query("delete from student where stuName = '小明'",$conn);

    //改

    //mysql\_query("update student set stuname='蝙蝠侠' where stuid = 999",$conn);

    //返回的是一个表格

    $result = mysql\_query("select \* from student where stuid = 2",$conn);

    //返回结果集中的字段数,也就是列数

    $result\_col = mysql\_num\_fields($result);

    //返回行数

    $result\_row = mysql\_num\_rows($result);

    echo $result\_row;

    if($result\_row == 1){

        echo "存在";

    }else{

        echo "不存在";

    //判断结果集的行数

    if(mysql\_num\_rows($result)>0){

                    //将结果集放到数组$rows

            while($rows = mysql\_fetch\_assoc($result)){

                echo $rows["stuName"]." ".$rows["stuAge"];

            }

        }

    }

    mysql\_close($conn);

?>

案例：模拟登陆

<?php

    header("Content-type:text/html;charset=utf-8");

    $username = $\_POST["username"];

    $pwd = $\_POST["pwd"];

    $conn = mysql\_connect("localhost","root","root");

    if($conn){

        echo "连接成功!";

        mysql\_select\_db("xa1901");

    }else{

        echo "连接失败!";

    }

    //切记加变量在字符串中加单引号

    $result = mysql\_query("select \* from student where stuid='$username' and stuName='$pwd'");

    $row = mysql\_num\_rows($result);

    if($row == 1){

        echo "登录成功";

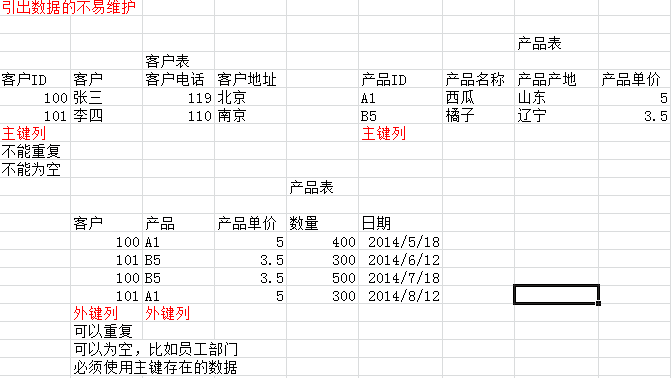
    }else{

        echo $row;

    }

?>

5.表的设计思想



create table Student

(

       stuId int primary key,

       stuName varchar(20) not null,

       stuGender char(4),

       stuAge int

)

表设计的三范式：

1.每个字段应该独立

2.每个表应该有主键

3.从表在访问主表的时候应该只出现主表的主键

作业：

1. 整理笔记，练习课堂代码
2. 预习cookie