2-7数据库编程(MySqli)下

1. 课堂案例

课堂讲解

一、上节回顾

mysql命令

工具的使用

自由主题

二、学习目标

数据库操作

三、教学过程描述

1、数据库操作

1-1、数据库函数

1、mysqli_connect

打开一个到 MySQL 服务器的新的连接。

语法:

mysqli_connect(host,username,password,dbname,port,socket);

host:可选。规定主机名或 IP 地址。

username： 可选。规定 MySQL 用户名。

password： 可选。规定 MySQL 密码。

dbname： 可选。规定默认使用的数据库。

port： 可选。规定尝试连接到 MySQL 服务器的端口号。

socket :可选。规定 socket 或要使用的已命名 pipe。

2、mysqli_select_db

mysqli_select_db() 函数用于更改连接的默认数据库。

语法:

mysqli_select_db(connection,dbname);

connection：必需。规定要使用的 MySQL 连接。

dbname：必需，规定要使用的默认数据库。

3、mysqli_set_charset

设置默认字符编码。

bool mysqli_set_charset (mysqli $link , string $charset)

设置在数据库间传输字符时所用的默认字符编码。

参数

link

仅以过程化样式：由mysqli_connect() 或 mysqli_init() 返回的链接标识。

charset

被设为默认的字符编码名。

4、mysqli_query

执行某个针对数据库的查询。

语法：

mysqli_query(connection,query,resultmode);

connection：必需。规定要使用的 MySQL 连接。

query：必需，规定查询字符串。

resultmode：可选。一个常量。可以是下列值中的任意一个：

MYSQLI_USE_RESULT（如果需要检索大量数据，请使用这个）

MYSQLI_STORE_RESULT（默认）

5、mysqli_num_rows

返回结果集中行的数量。

语法：

mysqli_num_rows(result);

result：必需。规定由 mysqli_query()、mysqli_store_result() 或 mysqli_use_result() 返回的结果集标识符。

6、mysqli_multi_query

执行一个或多个针对数据库的查询。多个查询用分号进行分隔。

语法：

mysqli_multi_query(connection,query);

7、mysqli_fetch_assoc

mysqli_fetch_assoc() 函数从结果集中取得一行作为关联数组。

注释：该函数返回的字段名是区分大小写的。

语法：

mysqli_fetch_assoc(result);

result 必需。规定由 mysqli_query()、mysqli_store_result() 或 mysqli_use_result() 返回的结果集标识符。

8、mysqli_connect_error

返回上一次连接错误的错误描述。

语法：

mysqli_connect_error();

9、mysqli_close

关闭先前打开的数据库连接。

语法：

mysqli_close(connection);

connection：必需。规定要关闭的 MySQL 连接。

1-2、连接 MySQL

定义和用法

mysqli_connect() 函数打开一个到 MySQL 服务器的新的连接。

语法：

mysqli_connect(host,username,password,dbname,port,socket);

参数	描述
host	可选。规定主机名或 IP 地址。
username	可选。规定 MySQL 用户名。
password	可选。规定 MySQL 密码。
dbname	可选。规定默认使用的数据库。
port	可选。规定尝试连接到 MySQL 服务器的端口号。
socket	可选。规定 socket 或要使用的已命名 pipe。

你可以使用PHP的 mysqli_close() 函数来断开与MySQL数据库的链接。

该函数只有一个参数为 mysqli_connect() 函数创建连接成功后返回的 MySQL 连接标识符。

语法:bool mysqli_close ($link)

本函数关闭指定的连接标识所关联的到 MySQL 服务器的非持久连接。如果没有指定 link_identifier, 则关闭上一个打开的连接。

提示:通常不需要使用 mysqli_close(), 因为已打开的非持久连接会在脚本执行完毕后自动关闭。

1-3、插入

1、单条插入

1、PHP 中 SQL 查询语句必须使用引号

2、在 SQL 查询语句中的字符串值必须加引号

3、数值的值不需要引号

4、NULL 值不需要引号

INSERT INTO 语句通常用于向 MySQL 表添加新的记录:

INSERT INTO table_name (column1, column2, column3,...)

VALUES (value1, value2, value3,...)

注意： 如果列设置 AUTO_INCREMENT (如 "id" 列) 或 TIMESTAMP (如 "reg_date" 列),，我们就不需要在 SQL 查询语句中指定值；MySQL 会自动为该列添加值。

2、多条插入

1、多次执行：

$sql = "INSERT INTO table_name (column1, column2, column3,...) VALUES (value1, value2, value3,...) ;"

$sql .= "INSERT INTO table_name (column1, column2, column3,...) VALUES (value1, value2, value3,...) ;"

2、一次执行

$sql = "INSERT INTO table_name (column1, column2, column3,...)

VALUES (value1, value2, value3,...) ,VALUES (value1, value2, value3,...) ;"

3、预处理语句

mysqli 扩展提供了第二种方式用于插入语句。

我们可以预处理语句及绑定参数。

mysql 扩展可以不带数据发送语句或查询到mysql数据库。你可以向列关联或 "绑定" 变量。

"INSERT INTO MyGuests (firstname, lastname, email) VALUES(?, ?, ?)"

mysqli_stmt_bind_param($stmt, 's', $name);

可以是以下四种参数:

- i - 整数
- d - 双精度浮点数
- s - 字符串
- b - 布尔值

每个参数必须指定类型，来保证数据的安全性。通过类型的判断可以减少SQL注入漏洞带来的风险。

1-4、查询

SELECT 语句用于从数据表中读取数据:

SELECT column_name(s) FROM able_name

我们可以使用 * 号来读取所有数据表中的字段：SELECT * FROM table_name

注意：在实际开发者表的字段过多的时候不要使用 *

1-4-1、Where 子句

where 子句用于提取满足指定标准的的记录。

查询语句中你可以使用一个或者多个表，表之间使用逗号，分割，并使用WHERE语句来设定查询条件。

你可以在 WHERE 子句中指定任何条件。

你可以使用 AND 或者 OR 指定一个或多个条件。

WHERE 子句也可以运用于 SQL 的 DELETE 或者 UPDATE 命令。

WHERE 子句类似于程序语言中的 if 条件，根据 MySQL 表中的字段值来读取指定的数据。

操作符	描述	实例
=	等号，检测两个值是否相等，如果相等返回true	(A = B) 返回false。
<>, !=	不等于，检测两个值是否相等，如果不相等返回true	(A != B) 返回 true。
>	大于号，检测左边的值是否大于右边的值, 如果左边的值大于右边的值返回true	(A > B) 返回false。
<	小于号，检测左边的值是否小于右边的值, 如果左边的值小于右边的值返回true	(A < B) 返回 true。
>=	大于等于号，检测左边的值是否大于或等于右边的值, 如果左边的值大于或等于右边的值返回true	(A >= B) 返回false。
<=	小于等于号，检测左边的值是否小于于或等于右边的值, 如果左边的值小于或等于右边的值返回true	(A <= B) 返回 true。

使用主键来作为 WHERE 子句的条件查询是非常快速的。

select * from table_name where 条件

1-4-2、AND & OR

如果第一个条件和第二个条件都成立，则 AND 运算符显示一条记录。

```php
$sql = "select "."* from user where id =1 and name ='小夏'";
$result =  mysqli_query( $conn, $sql );
if (mysqli_num_rows($result) > 0){
// 输出数据
    while($row = mysqli_fetch_assoc($result)) {
        echo "id: " . $row["id"]. " - Name: " . $row["name"]. "<br>";
    }
}else{
    echo '没有内容';
}
```

如果第一个条件和第二个条件中只要有一个成立，则 OR 运算符显示一条记录

```php
$sql = "select "."* from user where id =1 or name ='小冬'";
$result =  mysqli_query( $conn, $sql );
if (mysqli_num_rows($result) > 0){
// 输出数据
    while($row = mysqli_fetch_assoc($result)) {
        echo "id: " . $row["id"]. " - Name: " . $row["name"]. "<br>";
    }
}else{
    echo '没有内容';
}
```

1-4-3、count

返回匹配指定条件的行数

SQL COUNT(*) 语法

COUNT(*) 函数返回表中的记录数：

select count(*) from table_name;

as：设置一个别名

1-4-4、limit

在我们使用查询语句的时候，经常要返回前几条或者中间某几行数据。LIMIT 子句可以被用于强制 SELECT 语句返回指定的记录数。LIMIT 接受一个或两个数字参数。参数必须是一个整数常量。如果给定两个参数，第一个参数指定第一个返回记录行的偏移量，第二个参数指定返回记录行的最大数目。初始记录行的偏移量是 0(而不是 1)：为了与 PostgreSQL 兼容，MySQL 也支持句法：LIMIT # OFFSET #。

SELECT * FROM table LIMIT 5,10; // 检索记录行 6-1

SELECT * FROM table LIMIT 5; //检索前 5 个记录行

//换句话说，LIMIT n 等价于 LIMIT 0,n

1-4-5、like

们可以在 SELECT 语句中使用 WHERE 子句来获取指定的记录。但是我们也需要搜索包含此类字符的情况的条件，此时我们就可以使用like来模糊搜索

```php
$sql = "select "."* from user where name like '%小夏'";
//$sql = "select "."* from user where name like '小夏%'";
//$sql = "select "."* from user where name like '%小夏%'";
```

1-4-5-1、%

表示任意个或多个字符。可匹配任意类型和长度的字符。

1、%前

可以查出我是小夏，而不能查出小夏是帅哥。

select * from user where name like '%小夏'

2、%后

可以查出小夏是帅哥，而不能查出我是小夏

select * from user where name like '小夏%'

3、%%

这样就能搜出帅气的小夏和我是小夏

select * from user where name like '%小夏%'

1-4-5-2、_

表示任意单个字符。匹配单个任意字符，它常用来限制表达式的字符长度语句:（可以代表一个中文字符）

这样我就查找到小夏，小冬，小秋，小春的姓名

select * from user where name like '小_';

1-4-6、NULL

我们已经知道 MySQL 使用 SQL SELECT 命令及 WHERE 条件来读取数据表中的数据,但是当提供的查询条件字段为 NULL 时，该命令可能就无法正常工作。

为了处理这种情况，MySQL提供了三大运算符:

1、IS NULL: 当列的值是 NULL,此运算符返回 true。

2、IS NOT NULL: 当列的值不为 NULL, 运算符返回 true。

3<=>: 比较操作符（不同于=运算符），当比较的的两个值为 NULL 时返回 true。

关于 NULL 的条件比较运算是比较特殊的。你不能使用 = NULL 或 != NULL 在列中查找 NULL 值 。

在 MySQL 中，NULL 值与任何其它值的比较（即使是 NULL）永远返回 false，即 NULL = NULL 返回false 。

MySQL 中处理 NULL 使用 IS NULL 和 IS NOT NULL 运算符。

1-5、更新

UPDATE 语句用于更新数据库表中已存在的记录。

语法：

```
UPDATE table_name
SET column1=value, column2=value2,...
WHERE some_column=some_value
```

1、你可以同时更新一个或多个字段。

2、你可以在 WHERE 条件中指定任何条件。

3、你可以在一个单独表中同时更新数据。

注释：请注意 UPDATE 语法中的 WHERE 条件。WHERE子句规定了哪些记录需要更新。如果您想省去 WHERE 条件，所有的记录都会被更新！

1-6、删除

MySQL 数据表中删除数据的通用语法：delete from table_name where 条件

如果没有指定 WHERE 子句，table_name表中的所有记录将被删除。

你可以在 WHERE 子句中指定任何条件

你可以在单个表中一次性删除记录。

当你想删除数据表中指定的记录时 WHERE 条件是非常有用的。

2、PHP与mysql的结合

使用PHP查出mysql的数据后，在前端 展示

```php
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
<table border="1">
    <?php foreach ($put as $item): ?>
    <tr>
        <td><?=$item['id'] ?></td>
        <td><?=$item['name'] ?></td>
    </tr>
    <?php endforeach; ?>
</table>
</body>
</html>
```

四、小结

创建数据库

设计数据表

PHP增删改查的操作

2. 课堂练习

了解数字类型

安装数据库工具navicat并创建一张用户表

使用phpmyadmin创建一张用户表

3. 课后练习

1、使用mysql向用户表插入一条信息

2、使用mysql向用户表删除一条信息

3.、使用mysql向用户表更新一条数据

4. 资料扩展

1、命令

1-1、创建数据库

CREATE DATABASE test DEFAULT CHARSET utf8 COLLATE utf8_general_ci;

1-2、创建表

CREATE TABLE `user` (
`id` int(11) NOT NULL ,
`name` varchar(35) NULL COMMENT '用户名' ,
`sex` tinyint(1) NULL COMMENT '性别:1男2女' ,
PRIMARY KEY (`id`)
);
注意:MySQL命令终止符为分号 (;) 。

2、mysql 语法

2-1、sum

返回数值列的总数。
SQL SUM() 语法
select sum(column_name) from table_name;

2-2、max

返回指定列的最大值。

2-3、min

返回指定列的最小值。

2-4、in

IN 操作符允许您在 WHERE 子句中规定多个值。
语法
select * from table_name
where 字段 in (value1,value2,...);

```php
$sql = "select "."* from user where id in (1,2,3)";
$result =  mysqli_query( $conn, $sql );
if (mysqli_num_rows($result) > 0){
// 输出数据
    while($row = mysqli_fetch_assoc($result)) {
        echo "id: " . $row["id"]. " - Name: " . $row["name"]. "<br>";
    }
}else{
    echo '没有内容';
}
```

2-5、between

BETWEEN 运算符用于 WHERE 表达式中，选取介于两个值之间的数据范围。BETWEEN 同 AND 一起搭配使用，语法如下：

WHERE column BETWEEN value1 AND value2

WHERE column NOT BETWEEN value1 AND value2

通常 value1 应该小于 value2。当 BETWEEN 前面加上 NOT 运算符时，表示与 BETWEEN 相反的意思，即选取这个范围之外的值。

```php
//查询2-5的id
$sql = "select "."* from user where id BETWEEN 2 AND 5";
$result =  mysqli_query( $conn, $sql );
if (mysqli_num_rows($result) > 0){
// 输出数据
    while($row = mysqli_fetch_assoc($result)) {
        echo "id: " . $row["id"]. " - Name: " . $row["name"]. "<br>";
    }
}else{
    echo '没有内容';
}
```

2-6、别名

可以为表名称或列名称指定别名

语法

列的 sql 别名语法

select 字段 as alias_name from table_name;

表的 sql 别名语法

select 字段 from table_name as alias_name;

3、mysql函数

3-1、mysqli_free_result

3-2、mysqli_insert_id

返回插入中自动生成的 ID（通过 AUTO_INCREMENT 生成）。

语法

mysqli_insert_id(connection);

connection　必需。规定要使用的 MySQL 连接。

3-3、mysqli_affected_rows

返回前一次 MySQL 操作（SELECT、INSERT、UPDATE、REPLACE、DELETE）所影响的记录行数。

语法

mysqli_affected_rows(connection);

connection　必需。规定要使用的 MySQL 连接。

4、另外两种操作方式

4-1、pdo

4-2、mysql