

反馈文档

###Exercise 7 使用 PDO 与数据库建立连接。代码如下（解释见注释部分）：

```
try {
    $pdo = new PDO(DBCONNSTRING,DBUSER,DBPASS);
    //创建PDO对象，PDO的构造函数接受3个字符串参数，都在config.php文件中定义过
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    //将错误处理模式设置为ERRMODE_EXCEPTION
    //ERRMODE_EXCEPTION: 除设置错误码之外，PDO 还将抛出一个 PDOException异常类并设置它的属性来反射错误码和错误信息
    $sql = "select * from Artists order by LastName";
    //这是查询的语句，意为把Artists表中的行按照LastName这一列的内容的首字母升序排列
    $result = $pdo->query($sql);
    //$result是查询结果（对象）
    while ($row = $result->fetch()) {
        //一条条地获取查询结果，直到没有了（返回FALSE）为止
        echo $row['ArtistID'] . " - " . $row['LastName'] . "<br/>";
    }
    $pdo = null;
    //把对象设为null，断开连接，释放资源
} catch (PDOException $e) {
    die( $e->getMessage() );
    //若连接不成功，则处理异常
}
```

Database Tester (PDO)

```
129 - Alma-Tadema
41 - Balla
135 - Bellini
138 - Böcklin
117 - Bosch
95 - Botticelli
107 - Boucher
140 - Bouguereau
120 - Bronzino
118 - Bruegel the Elder
131 - Canaletto
101 - Caravaggio
16 - Cassatt
21 - Cezanne
64 - Chagall
108 - Chardin
10 - Courbet
97 - da Vinci
45 - Dali
5 - David
15 - Degas
7 - Delacroix
32 - Derain
116 - Dürer
103 - Dyck
93 - Francesca
137 - Friedrich
141 - Gaddi
110 - Gainsborough
18 - Gauguin
111 - Gentileschi
114 - Giotto
8 - Goya
121 - Greco
123 - Hals
136 - Hayez
126 - Hooch
6 - Ingres
40 - Kandinsky
60 - Klee
29 - Klimt
125 - Kruseman
... ..
```

使用 MySQLi 与数据库建立连接。代码如下（解释见注释部分）：

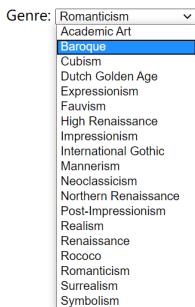
```
$connection = mysqli_connect(DBHOST, DBUSER, DBPASS, DBNAME);
if ( mysqli_connect_errno() ) {
    die( mysqli_connect_error() );
    //若连接不成功，处理异常
}
$sql = "select * from Genres order by GenreName";
//在Genres表中按照GenreName列的内容的首字母升序排列Genres表中的行
if ($result = mysqli_query($connection, $sql)) {
```

```

// loop through the data
while($row = mysqli_fetch_assoc($result)) {
    echo '<option value="' . $row['GenreID'] . '">';
    echo $row['GenreName'];
    echo "</option>";
//将选出的、排序好的内容依次加入select的option中
}
// release the memory used by the result set
mysqli_free_result($result);
}
// close the database connection
mysqli_close($connection);

```

Database Tester (mysqli)



###Exercise 8 代码如下（解释见注释部分）：

```

function outputArtists() {
    try {
        $pdo = new PDO(DBCONNSTRING,DBUSER,DBPASS);
//创建PDO实例
        $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
//将错误处理模式设置为ERRMODE_EXCEPTION
//ERRMODE_EXCEPTION: 除设置错误码之外, PDO 还将抛出一个 PDOException异常类并设置它的属性来反射错误码和错误信息
        $sql = "select * from Artists order by LastName limit 0,30";
//这是查询的语句, 意为把Artists表中的行按照LastName这一列的内容的首字母升序排列, 显示前30条
        $result = $pdo->query($sql);
//$result是查询结果 (对象)
        while ($row = $result->fetch()) {
//遍历查询结果
            echo '<a href="' . $_SERVER["SCRIPT_NAME"] . '?id=' . $row['ArtistID'] . '" class="';
//将查询结果所得的艺术家名字设成a元素
            if (isset($_GET['id']) && $_GET['id'] == $row['ArtistID']) echo 'active ';
//获取鼠标点击的a元素 (? 添加active的类名
            echo 'item">';
//每个元素都添加item的类名
            echo $row['LastName'] . '</a>';
        }
        $pdo = null;
//断开连接, 释放缓存
    }
    catch (PDOException $e) {
        die( $e->getMessage() );
    }
}

```

```
    }
}
```

这段代码和上面几乎全都重复

```
function outputPaintings() {
    try {
        if (isset($_GET['id']) && $_GET['id'] > 0) {
            $pdo = new PDO(DBCONNSTRING,DBUSER,DBPASS);
            $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
            $sql = 'select * from Paintings where ArtistId=' . $_GET['id'];
            //查找点击的艺术家（名字）对应的作品
            $result = $pdo->query($sql);
            while ($row = $result->fetch()) {
                outputSinglePainting($row);
            }
            //显示查询结果中艺术家的代表作、作品标题和描述
            $pdo = null;
        }
    } catch (PDOException $e) {
        die( $e->getMessage() );
    }
}

function outputSinglePainting($row) {
    echo '<div class="item">';
    echo '<div class="image">';
    echo '';
    //将img元素的src属性根据图片分别设成不同的值
    echo '</div>';
    echo '<div class="content">';
    echo '<h4 class="header">';
    echo $row['Title'];
    //获取查询结果的标题
    echo '</h4>';
    echo '<p class="description">';
    echo $row['Excerpt'];
    //获取查询结果中的描述部分
    echo '</p>';
    echo '</div>'; // end class=content
    echo '</div>'; // end class=item
}
```

User Input

Alma-Tadema

Balla

Bellini

Böcklin

Bosch

Botticelli

Boucher

Bouguereau

Bronzino

Bruegel the Elder

Canaletto

Caravaggio

Cassatt

Cezanne

Chagall

Chardin

Courbet

da Vinci

Dali

David

Degas

Delacroix

Derain

Dürer

Dyck


Francesca

Friedrich

Gaddi


Gainsborough

Gauguin




Portrait of Elizabeth and Philadelphia Wharton

Anthony van Dyck in this portrait of the daughters of Philip, 4th Lord Wharton, produced a genuinely appealing image.




Portrait of Henry Danvers, Earl of Danby

From 1632 to 1641 van Dyck worked at the court of the English King Charles I, where he produced a marvellous series of portraits of the English aristocracy, of which this is perhaps one of the most outstanding.



Portrait of Agostino Pallavicini

Agostino Pallavicini, a member of the Genoese branch of the Pallavicini family and the future doge of Genoa, sits enveloped by the sumptuous, flowing red robes worn in his role as ambassador to the Pope.



William II, Prince of Orange, and his Bride, Mary Stuart

William's father, Frederick Henry, commissioned the celebrated Flemish painter Van Dyck to portray the young Dutch prince and English princess on the occasion of their marriage in London.

Exercise 9

执行SQL语句的方式

DriverManager：用于管理 JDBC 驱动的服务类。主要功能是获取 Connection 对象。

Connection：代表数据库连接对象，每个 Connection 代表一个物理连接会话。

Statement：用于执行SQL语句的工具接口。常用方法：

- ResultSet executeQuery(String sql)throws SQLException：专用于查询。
- int executeUpdate(String sql)throws SQLException：执行 DDL、DML 语句，前者返回0，后者返回受影响行数。
- boolean execute(String sql)throws SQLException：可执行任何 SQL 语句。如果执行后第一个结果为 ResultSet（即执行了查询语句），则返回 true；如果执行了 DDL、DML 语句，则返回 false。返回结果为 true，则随后可通过该 Statement 对象的 getResultSet() 方法获取结果集对象（ResultSet 类型），返回结果为 false，则可通过 Statement 对象的 getUpdateCount() 方法获得受影响的行数。

PreparedStatement：为Statement的子接口，可预编译SQL 语句，常用语执行多条结构相同，仅值不同的SQL 语句，见下例。同样具有Statement对象常用的三个方法，但用法不同，因为已经预编译了SQL 月，所以无需再在方法中写sql语句，只需 setString方法设置参数值即可，如代码中。

好处

使用 PreparedStatement 比使用 Statement 多了三个好处：

- PreparedStatement 预编译 SQL 语句，性能更好，执行更快。
- PreparedStatement 无须“拼接” SQL 语句，编程更简单。
- PreparedStatement 可以防止 SQL 注入（如将输入的 true 当成直接量，导致判断直接通过，从而降低了安全性），安全性更好。