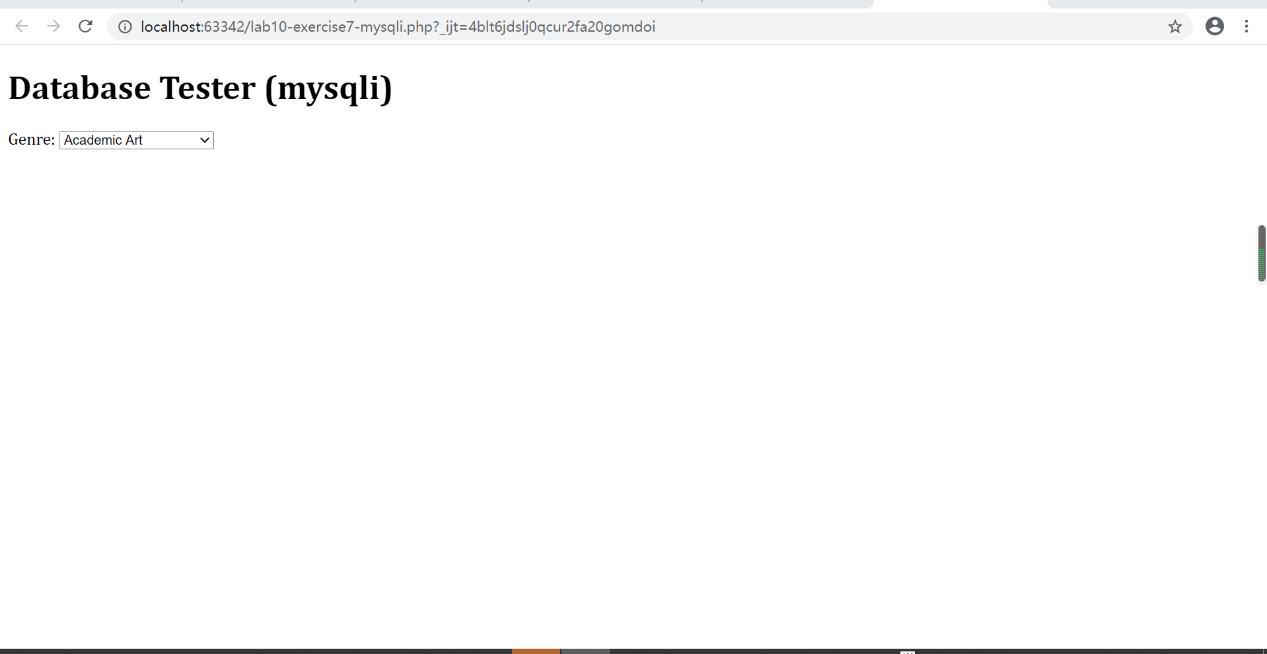
EXERCISE7





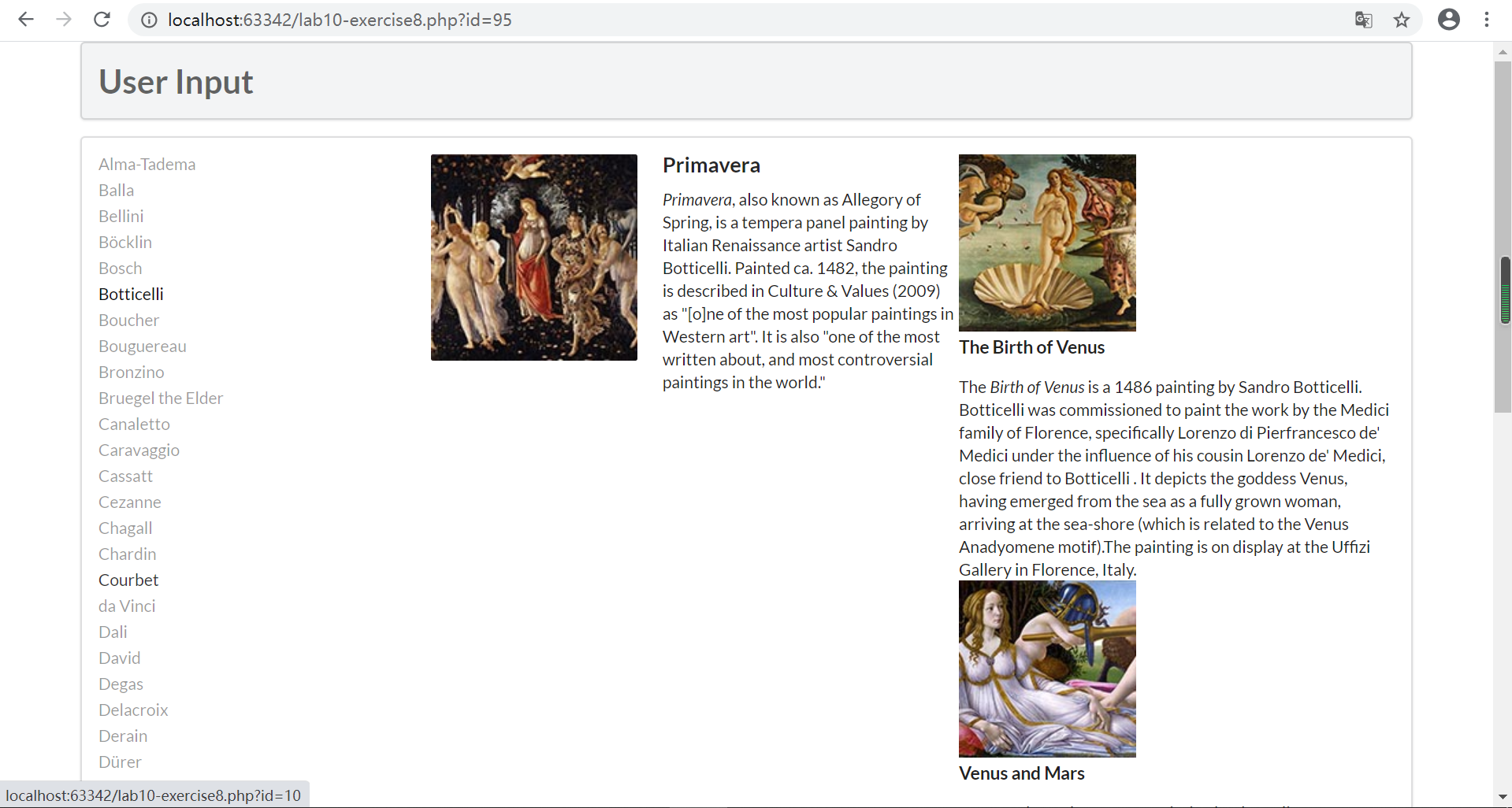
Config.php定义了之前设置的用户，定义了它的用户名、密码等。

lab10-exercise7-pdo.php的功能是将艺术家们按last name的字母表顺序排列：用config中定义的用户名、密码等创建了一个pdo类的实例，接着通过这个pdo来对数据库进行操作，$sql是数据库的筛选方式，这里就是选出artist中的所有人并按last name排序。$result是pdo根据$sql筛选后返回的结果数组，用循环打印出来就得到了页面显示的结果。

lab10-exercise7-mysqli.php通过mysqli与数据库连接，通过mysqli\_connect函数 创建了连接$connection，后续的操作要用到这个连接，最后在用完后关闭这个连接。这里的$sql筛选的条件是Genres中的全部按GenreName排序，$result是通过上述的连接和筛选方式，利用mysqli\_query函数得到的结果们，通过循环将他们依次加入select中的option。做完后将result的内存清空，并断开连接。

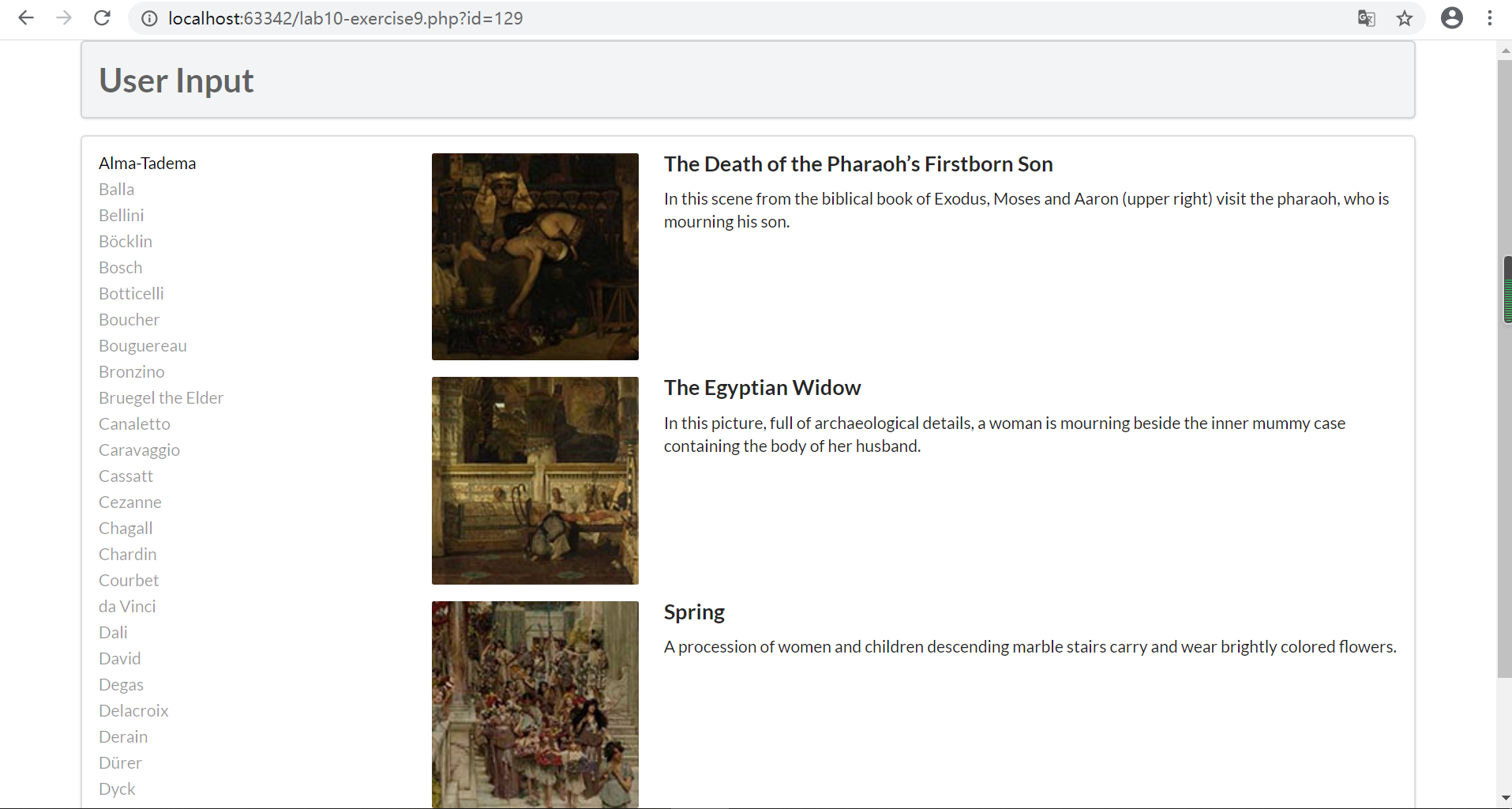
两个方法的区别就是一个是创建pdo类的实例，通过对它操作来操作数据库，另一个是创建一个与数据库的连接，操作数据库需要用到这个连接。

EXERCISE8



这里用的是exercise7中第一个利用pdo类的方式操作数据库，outputArtists()函数通过$筛选到Artist中、按lastName进行排序结果的前30个，通过循环将每个艺术家的名字设置为一个<a>，放在下面的html部分中的<div class="ui link list"></div>中。而outputPaintings()函数是用来在点击左边艺术家名字后显示对应的图片的。它通过$sql筛选所点击的艺术家id为条件索引其名下的所有图片，得到的结果，并按行循环依次输出图片。输出图片的方法为调用outputSinglePainting($row)函数，通过echo函数在页面上输出一块内容，具体结构是class为item的div中包含了图片div（通过$row的属性得到位置和名称），包含名字和描述的内容div。这一整块在点击左边链接后显示的图片位于<div class="ui items"></div>中，动态生成。

EXERCISE9



执行sql语句的方法有4种，E9中用的是prepare()

Exec()函数：$res=$pdo->exec($query)，返回值是受影响的行数; 通常进行的是insert、delete等操作。

Query()函数：$res=$pdo->query($sql);$sql是筛选条件（如select \* from Artists order by LastName），返回值是搜索的结果。Query()可以实现所有exec()的功能。

预处理语句：包括prepare()和execute()两步。prepare()方法负责查询准备，execute()方法负责执行查询，此外bindParam()方法可以绑定参数给execute()方法

$res=$pdo->prepare($query)

$res->execute();

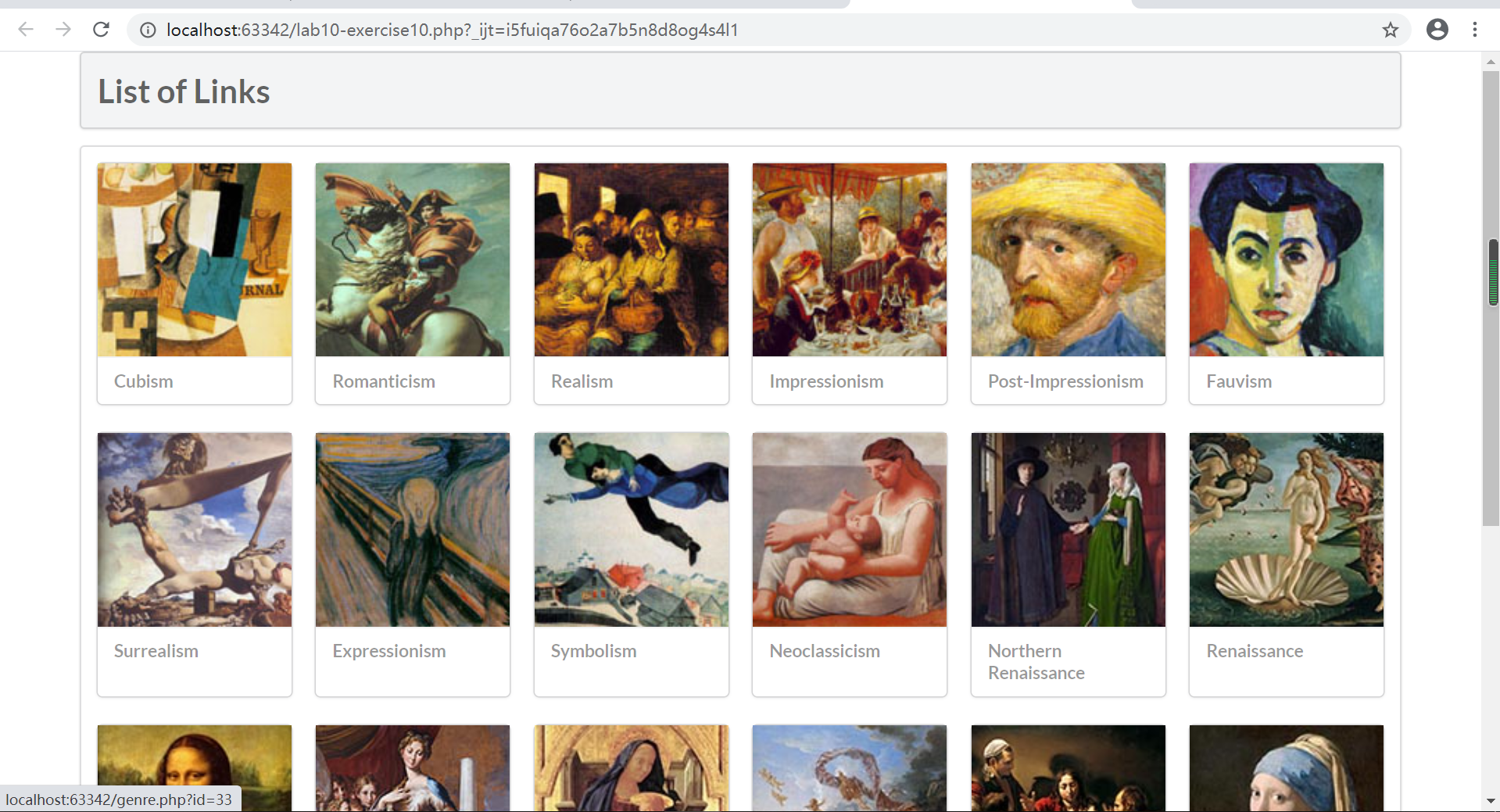
而使用预处理语句的好处在于：

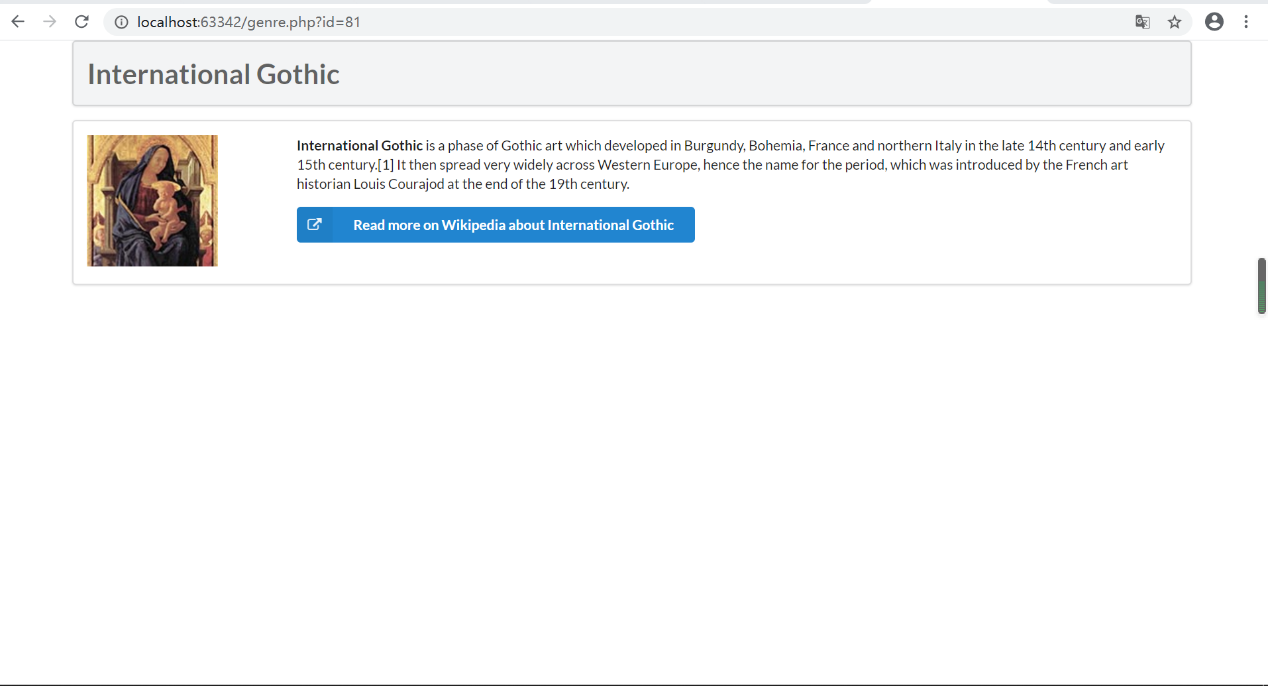
1、只解析查询一次，然后可以使用相同或不同的参数执行多次。Prepare()函数执行之后，数据库就会分析、编译并优化它要执行查询的计划。如果是结构相同而参数不同的复杂查询，那多次解析就会很占时间、执行速度较慢，而通过使用一个预处理语句就可以避免重复分析、编译、优化的环节。

2、传给预处理语句的参数不需要使用引号。

3、减少sql注入的机会。

EXERCISE10





outputGenres()函数用上述方法搜索出Genres中的 GenreId, GenreName, Description 并用GenreId进行排序。得到的结果通过循环以outputSingleGenre()的函数输出。outputSingleGenre()函数的功能则是动态生成一个包括图片和文字信息的div，形成的过程和结构与exercise8、9中的差不多，通过constructGenreLink($id, $label)动态连接id、label等成分，生成对应图片的链接，实现点击图片和下方的文字都能转到对应详情页面的效果。