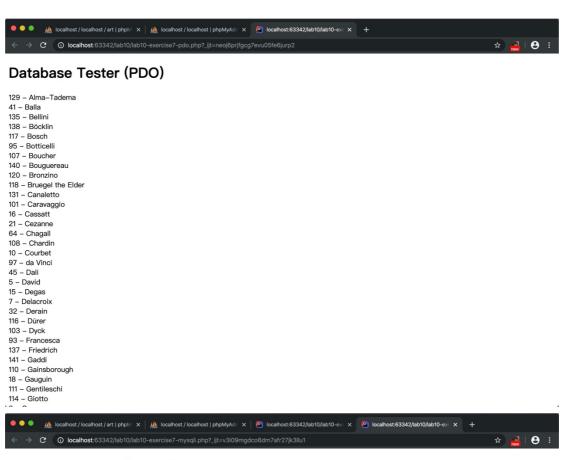
# Exercise7



#### Database Tester (mysqli)



#### 代码解释:

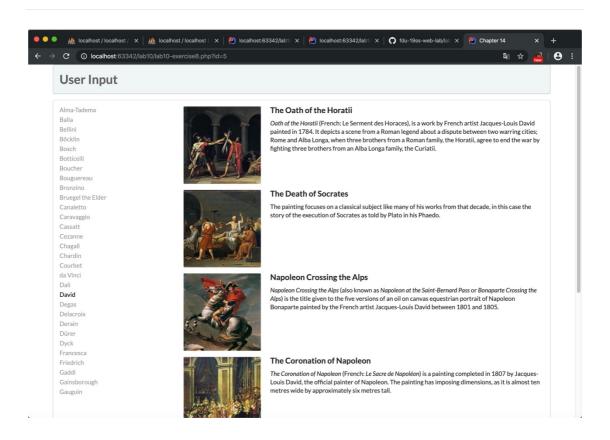
#### lab10-exercise7-pdo.php

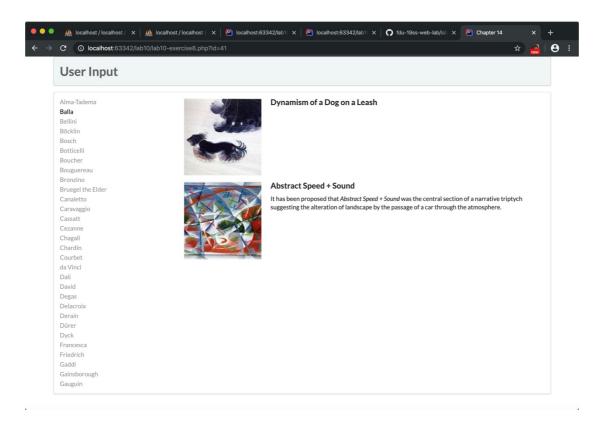
采用了 PDO 的方式连接 MySQL, 首先创建 PDO 对象, 然后设置 PDO 错误模式为异常, \$sql 为数据库检索对象(字符串), 当\$result 为匹配结果, 用 while 语句, 当匹配到时, 将结果返回给\$row, 然后打印\$row 的相关信息;最后关闭连接

#### lab10-exercise7-mysqli.php

使用 MySQLi 进行连接,首先创建连接,然后与上述方法相同,同样是通过匹配检索对象(\$spl),通过使用循环,匹配到后打印\$row 的相关信息,最后关闭连接。

## **Exercise8:**





#### 代码解释:

outputPaintings 方法与 outputSinglePainting 方法::

outputPaintings 方法与上述使用 PDO 思路相同,先于数据库创建连接,然后根据匹配语句进行匹配,然后打印获取的对象,打印对象使 outputSinglePainting 方法。

# **Exercise9:**

#### 使用预处理语句:

预处理语句工作原理:

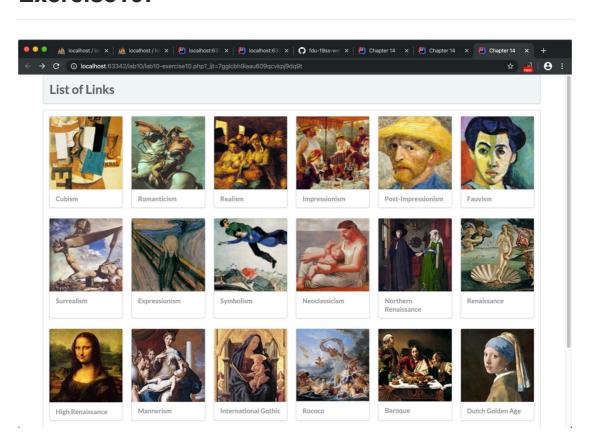
- 1. 预处理:创建 SQL 语句模版并发送到数据库。预留的值使用参数 "?"标记;
- 2. 数据库解析,编译,对 SQL 语句模版进行查询优化,并存储结果不输出;

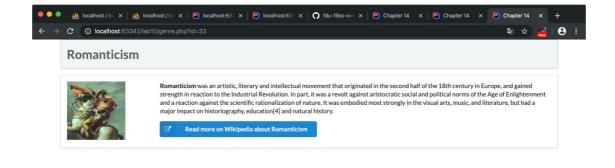
3. 执行:最后,将应用绑定的值传递给参数("?"标记),数据库执行语句。

### 相对于直接执行 SQL 语句, 预处理语句有两个主要的优点:

- · 预处理语句大大减少了分析时间,制作了一次查询(虽然语句多次执行);
- · 绑定的参数减少了服务器宽带,只需要发送查询的参数,而不是整个语句;
- · 预处理语句针对 SQL 注入是非常有用的,因为参数值发送后使用不同的协议,保证了数据的合法性。

## Exercise 10:





### 代码解释:

创建 PDO 的对象连接到数据库,然后进行检索,将获取到的 Genres 的内容,用 outputSingleGenre 方法打印出来,打印的为 html 内容,constructGenreLike 方法为返回所需要的 a 标签。