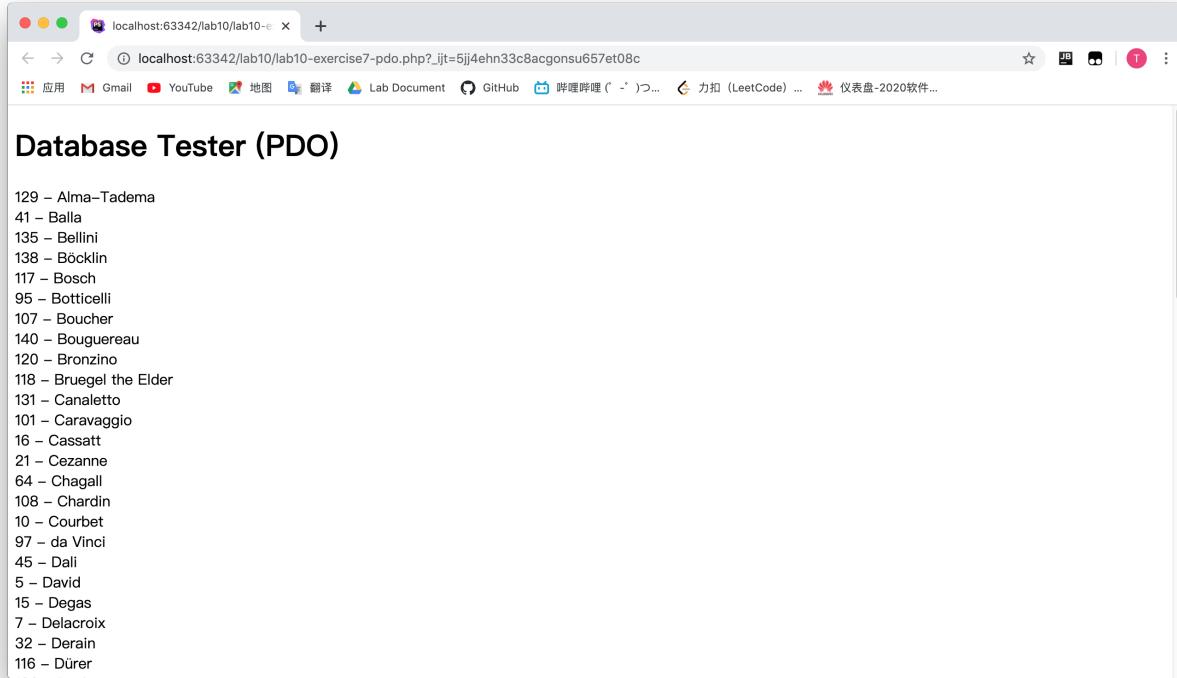


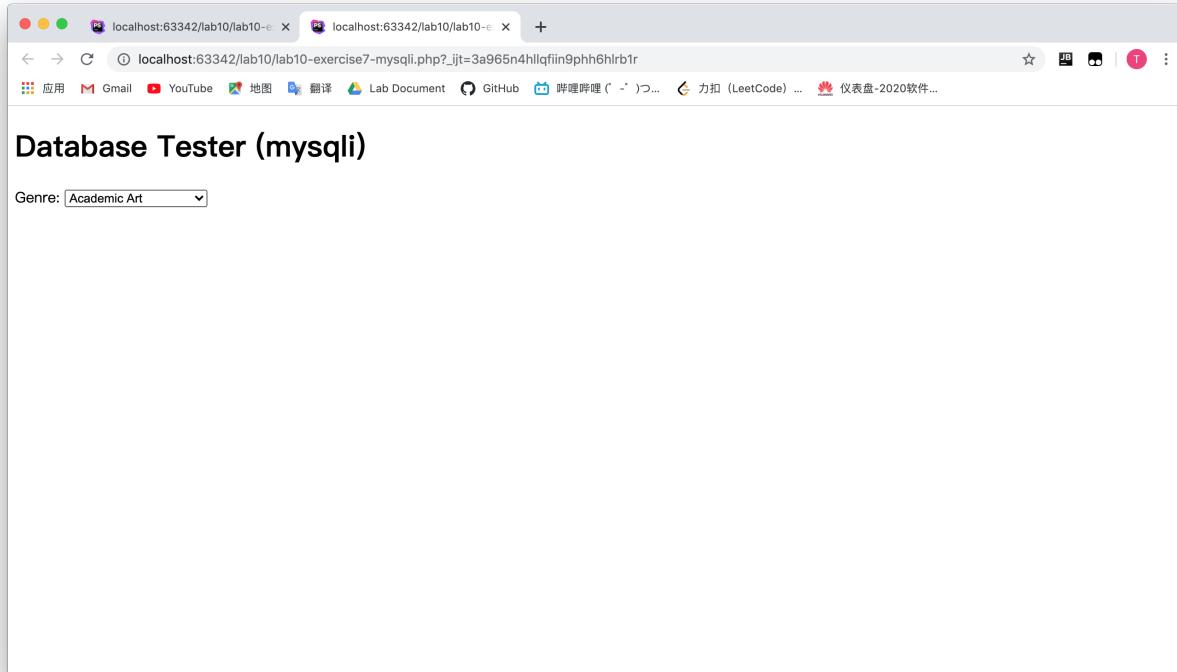
Lab10 Document

Exercise 7

- lab10-exercise7-pdo.php



- lab10-exercise7-mysqli.php

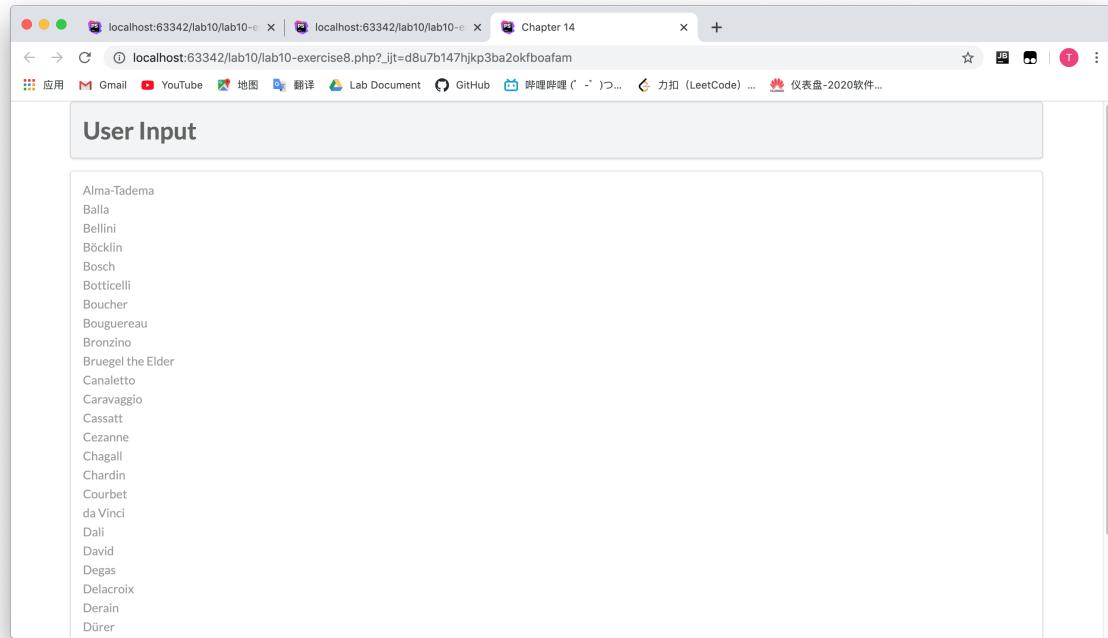


- two ways to access data 都是使用sql语句但是具体的句法api有所不同
 - pdo: 使用 \$pdo->query(\$sql) 从查询结果获取数据使用 \$row = \$result->fetch()

- o mysqli: 使用 mysqli_query(\$connection, \$sql) 从查询结果获取数据使用 \$row = mysqli_fetch_assoc(\$result)

Exercise 8

- lab10-exercise8.php



o

- outputArtists(): 使用sql语句和pdo方法获取LastName在0-30的Artist对象，再把他们打印出来
- outputPaintings(): 使用ArtistId查询到Painting对象，并在循环中调用outputSinglePainting()进行展示
- outputSinglePainting(\$row): 打印查询到的Painting对象

Exercise 9

- lab10-exercise9.php

- o exec()方法返回执行SQL 语句后受影响的行数，其语法格式如下：

```
int PDO::exec(string statement)
```

- o query()方法用于返回执行查询后的结果集，该函数的语法格式如下：

```
PDOStatement PDO::query(string statement)
```

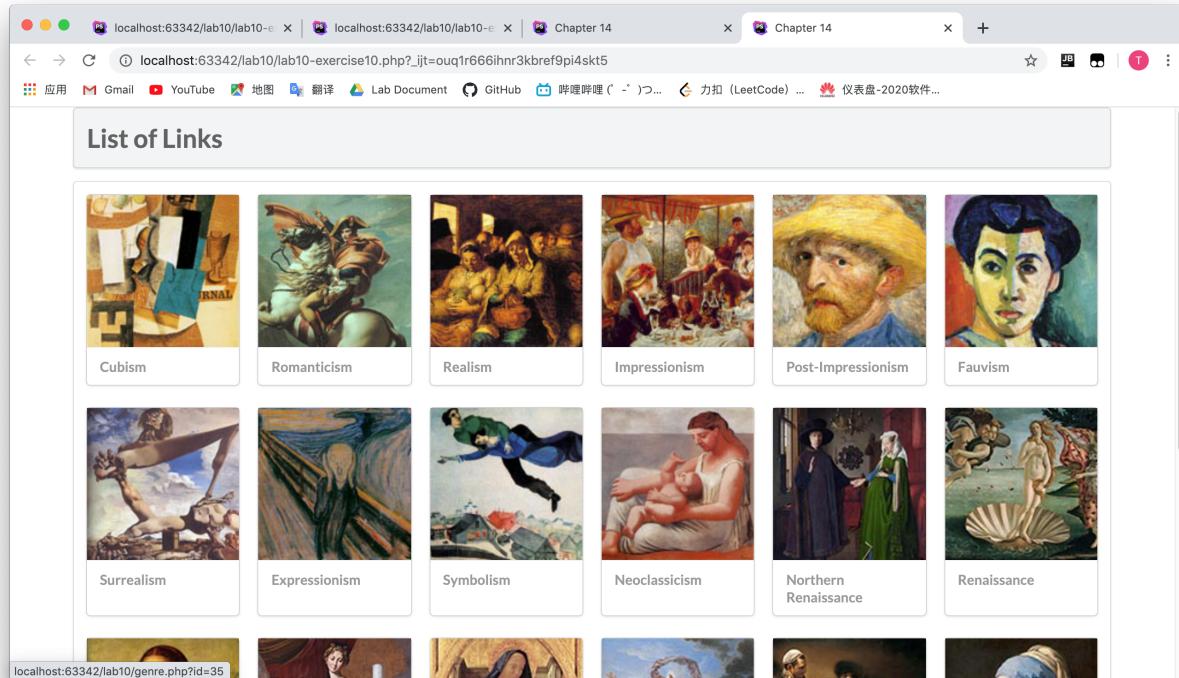
- o 预处理语句包括prepare()和execute()两种方法。首先，通过prepare()方法做查询准备工作，然后通过execute()方法执行查询，并且还可以通过bindParam()方法来绑定参数给execute()方法，语法如下：

```
PDOStatement PDO::prepare(string statement[,array driver_options]) bool
```

```
PDOStatement::execute([array input_parameters])
```

Exercise 10

- lab10-exercise10.php



- 查询到所有的GenreId, GenreName, Description, 再使用outputSingleGenre(\$row)逐一输出