[PDO 3](#_Toc488833834)

[手册位置 3](#_Toc488833835)

[作用： 3](#_Toc488833836)

[扩展-外部扩展的使用 4](#_Toc488833837)

[PDO的使用 6](#_Toc488833838)

[数据库的使用步骤 6](#_Toc488833839)

[$pdo = new PDO(dsn); 7](#_Toc488833840)

[提交 8](#_Toc488833841)

[Pdo连接属性的设置和获取 10](#_Toc488833842)

[事务处理 11](#_Toc488833843)

[关闭自动提交 11](#_Toc488833844)

[预处理 12](#_Toc488833845)

[定义|准备模版 12](#_Toc488833846)

[绑定参数 12](#_Toc488833847)

[准备变量 13](#_Toc488833848)

[提交 13](#_Toc488833849)

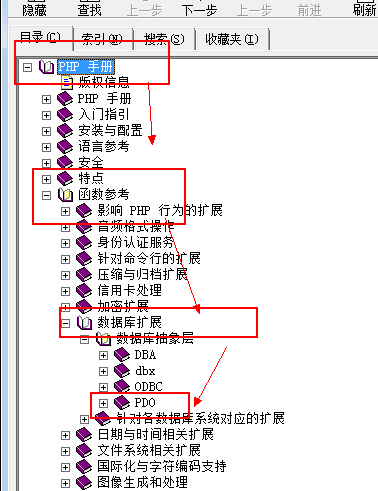
[Model类的封装 13](#_Toc488833850)

# 回顾

* 数据表设计
  + 原则：字段的选择要合理
  + 字段类型
    - 数值型
    - 字符串型
    - 日期时间型
      * 通常用int来存时间戳
  + 字段属性关键词
    - Not null 非null,非空
    - Unsigned 无符号，都是正数
    - Auto\_increment 通常用在主键上
    - Default：默认值
    - Zerofill：零填充
  + 索引
    - 为了提高检索|查询速率，索引并不是越多越好
    - 索引是附加在字段上的
    - Primary key
      * 1.一个表只能有1个
    - Unique
      * 1.一个表允许有多个
    - Index
    - Full-text
    - Foreign key
  + 引擎
    - 设置：
      * 建表的时候在结束小括号后面使用engine=xxx
    - 常见的引擎：
      * Myisam
        + 1.不支持外键索引
        + 2.不支持事务处理
        + 3.有3个文件
      * Innodb
        + 1.支持外键索引
        + 2.支持事务处理
        + 3.有2个文件
  + 字符集
    - Utf8
    - 对于字符集，建库的时候就改为utf8，对应的表就是自动生成的utf8
* 数据语句设计
  + As 别名
  + Where条件
    - 逻辑判断
      * And
      * Or
      * !
    - 比较判断
      * >
      * <
      * =
      * ....
    - 运算
    - Between x and y
      * 包含x,y
    - Not between x and y
      * 不包含x,y
    - In(x,y,z)
      * 只有xyz
    - Not in(x,y,z)
      * 除了xyz
    - Like
      * 通配符：
        + \_:匹配单一任意字符
        + %：匹配任意字符，可有可无
    - Not like
    - Null
    - Not null
  + Group by
    - 分组，通常用来做统计
  + Order by
    - 排序
    - 可选值
      * Asc:默认的，升序
      * Desc,降序
  + Limit
    - Limit x
      * 结果集中取前x条
    - Limit x,y
      * 结果集中从x+1开始取，取y条
      * 就是分页
  + 常用的函数
    - Count():求数目
    - Sum():求和
    - Avg():求平均
    - Max():求最大
    - Min():求最小

# PDO

## 手册位置

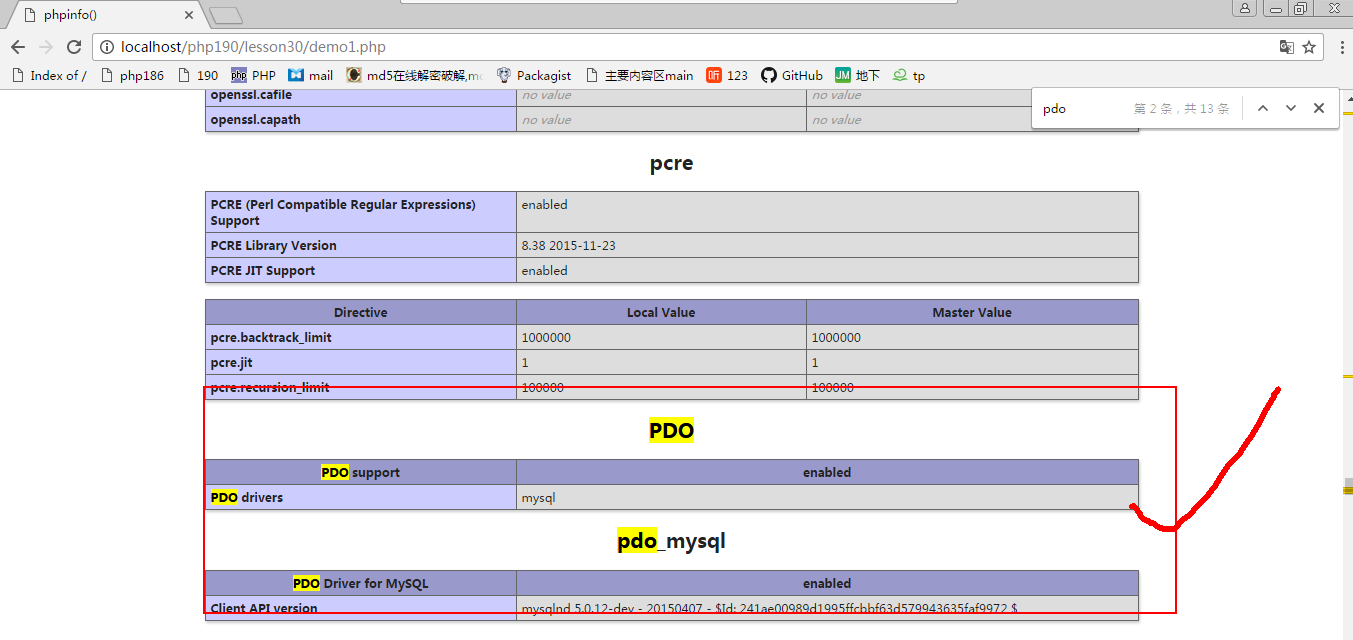


## 作用：

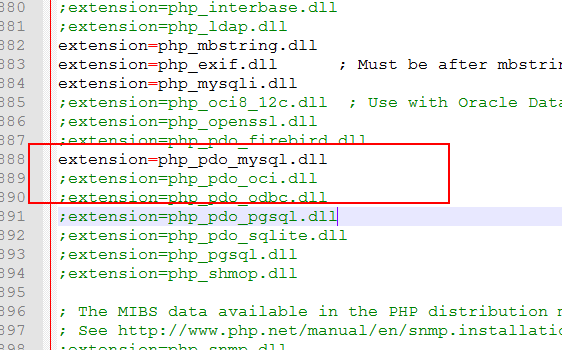
* Php和mysql做交互的一种方式，可以通过php来操作mysql

# 扩展-外部扩展的使用

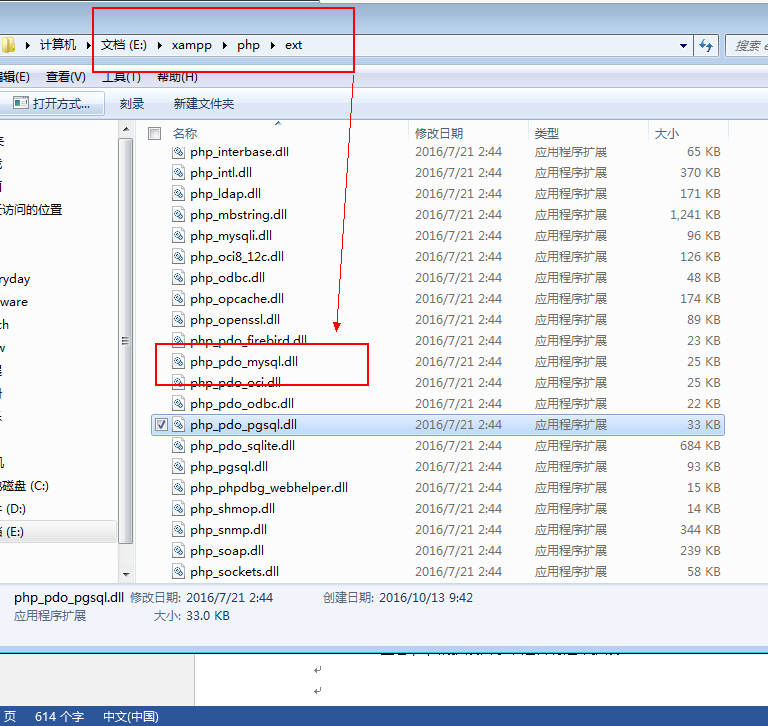
* 注意：pdo是php的外部扩展，所以使用之前先得判断是否加载了这个扩展，如何查看



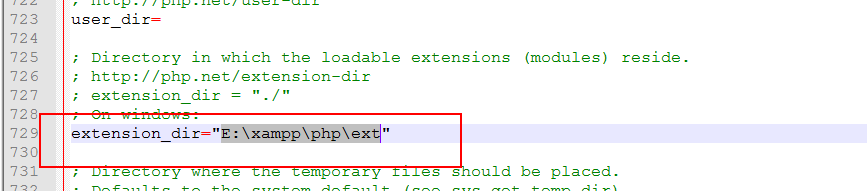
* 如果发现没有，可参考以下步骤
  + 1.打开php.ini文件，看是否开启



* + 2.查看php的扩展目录下是否有这个扩展



* 3.检查ext的目录路径是否正确（所有的扩展都没有）



## PDO的使用



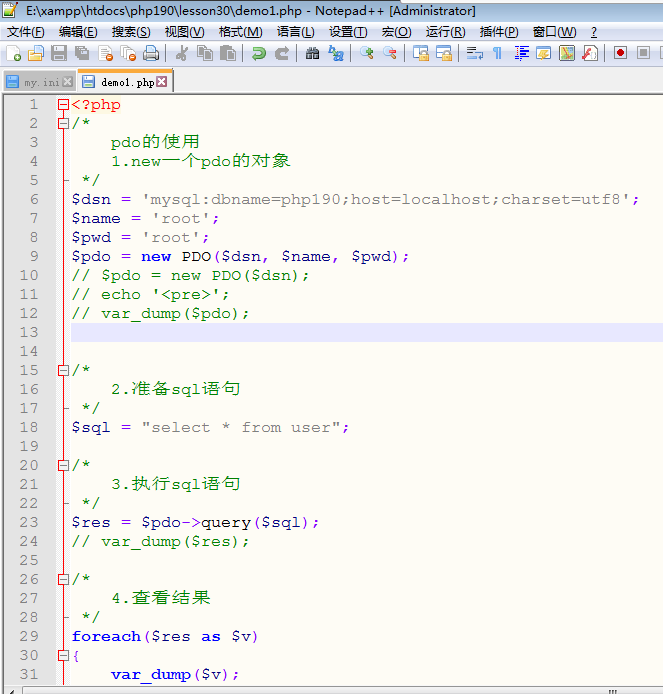
### 数据库的使用步骤

* 1.连接数据库
* 2.设置字符集，选择数据库
* 3.准备sql语句
* 4.执行sql语句
* 5.看结果
* 6.关闭数据库连接

### $pdo = new PDO(dsn);



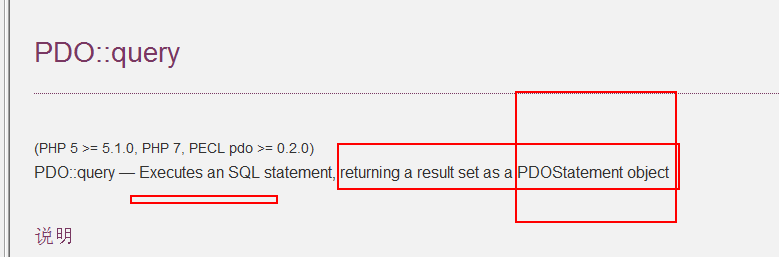
* Dsn数据源名，data source name
* 示例



### 提交

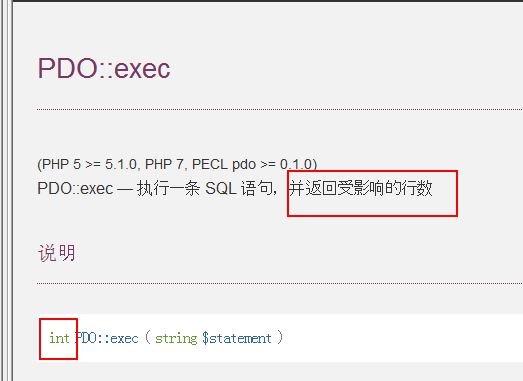
**提交分为两种，**

* **查询的提交**
  + **特点：返回结果集**



**$pdo->query($sql);**

* **使用query（）默认结果是PDOStatement的对象，可以变为数组**
* **2，增，删，改的提交**
  + **特点：返回影响行数**

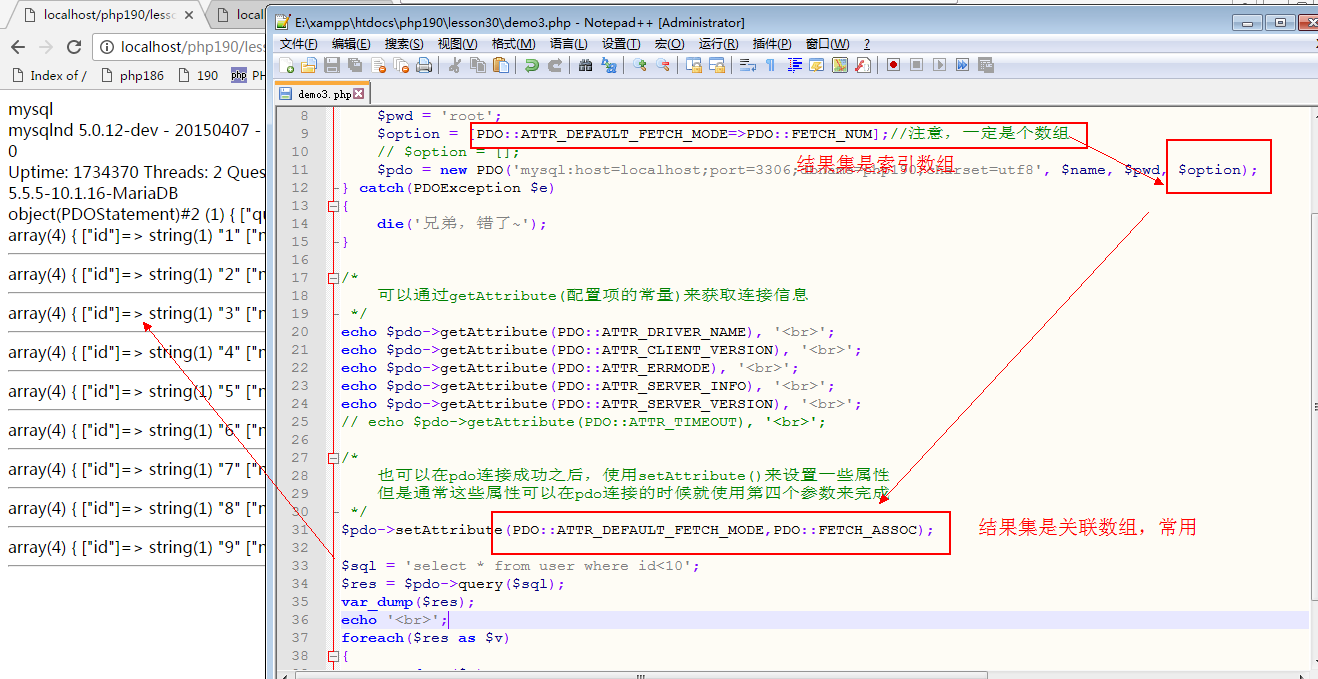


**示例：**



### Pdo连接属性的设置和获取

* 属性的获取：
  + **$pdo->getAttribute(名)**



* **属性的配置**
  + **$pdo->setAttribute(配置名，值)**
* **注意：通常这些配置都可以在pdo连接的时候使用第四个参数来设置。比如说：结果集的格式是关联数组还是索引数组**

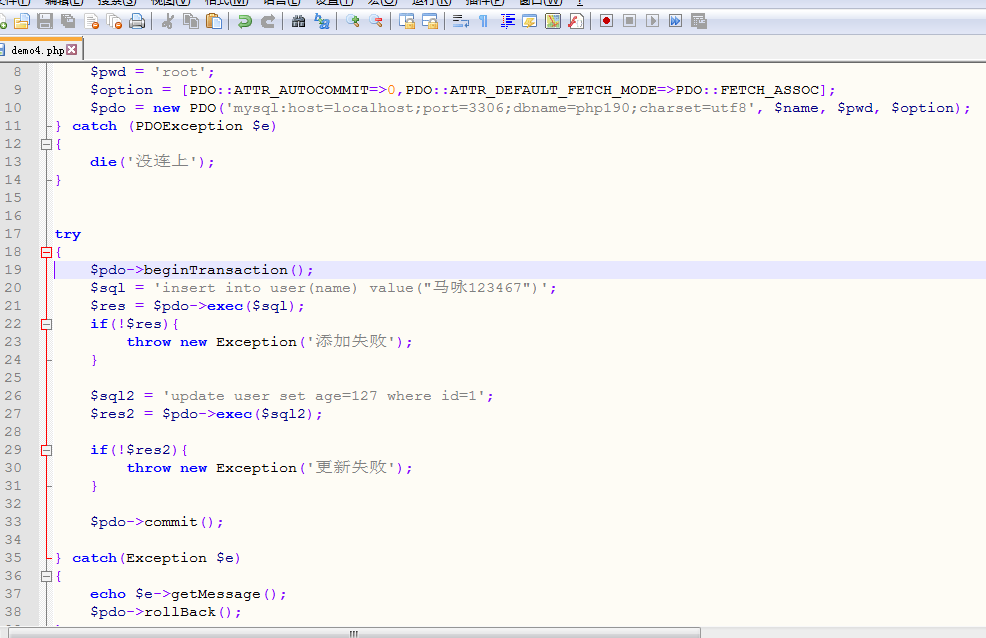
## 事务处理

* 场景：
  + 银行转账，A给B1000，银行从A账户扣1000，加到B账户上，如果银行扣了但是没加上，怎么办？得撤销，此处就可以用到事务处理
* 特点：
  + 原子性
  + 一致性
  + 隔离性
  + 持久性
* 也就是说：数据的操作必须得同时成功，才算成功，否则的话，就撤销恢复到以前

### 关闭自动提交

setAttribute(PDO::AUTO\_COMMIT, 0);

* 示例：demo4.php



## 预处理

* 好处
  + 1.sql语句只需要定义一次就够了，需要写值的地方，临时赋值
  + 2.参数无需引号括起来，防止sql注入

### 定义|准备模版

* $st = $pdo->prepare(sql语句的模版)

### 绑定参数

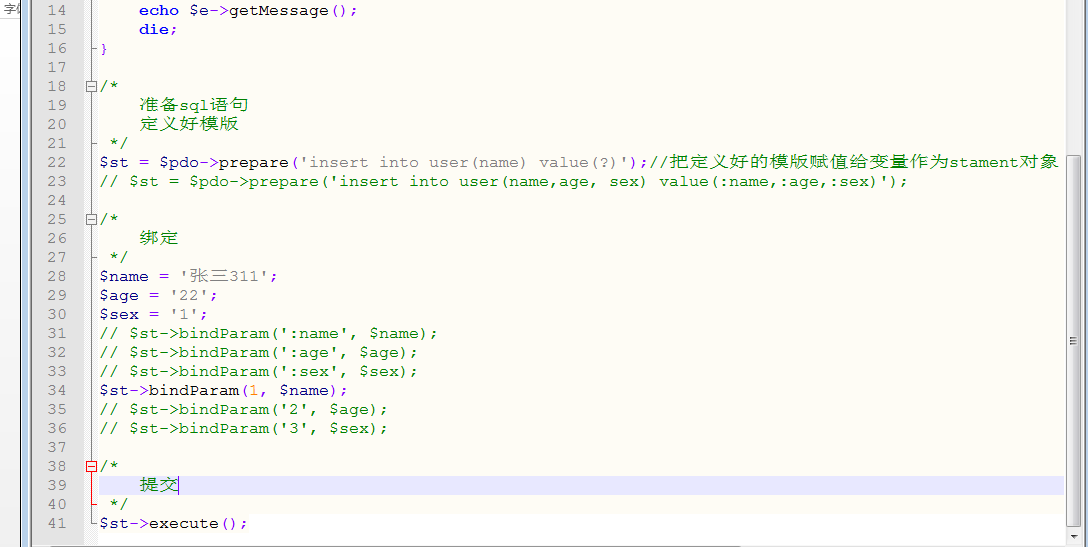
* **模版中使用:name**
  + **$st->bindParam(‘:name’, $name)**
* **模版中使用？**
  + **$st->bindParam(1, $name)**

### 准备变量

### 提交

**$st->execute()**

**示例：**



# Model类的封装

* 封装增删改查的方法