Trình tự làm CRUD

Thứ tự các bước làm bài tập

* Viết Comment cẩn thận tên, mssv, ngày làm, mô tả file code
* Phân tích mã để biết mình sẽ có 2 phần lớn là Soạn DB và Soạn UI
* Phân tích mã để biết thứ tự hoạt động của function (store) SQL sẽ có ít nhất 6 việc chính là
  1. Tạo Câu SQL;
  2. Có thể có ít nhất một tham số cần truyền bind\_param hoặc không có tham số nào;
  3. Kiểm tra duyệt excecute câu SQL
  4. Đóng SQL -> close();
  5. Có thể có một giá trị trả về hoặc không có giá trị nào được trả về. Cần trả về mã lỗi để biết SQL thực thi thành công hay thất bại
     + -1: Lỗi đầu vào
     + -2: Lỗi truy vấn
     + 0: Thực thi truy vấn thành công
     + -3: select Không tồn tại ID trong csdl
     + -3: insert Tồn tại ID trong csdl
  6. Xử lý ngoại lệ trước khi trả về mã lỗi.

[Link sao chép thư mục tham khảo CRUD của cô Hồng Mỹ](https://drive.google.com/drive/folders/1LEaDqfylJ87reigERIEYDardiDOP6XgW?usp=sharing)

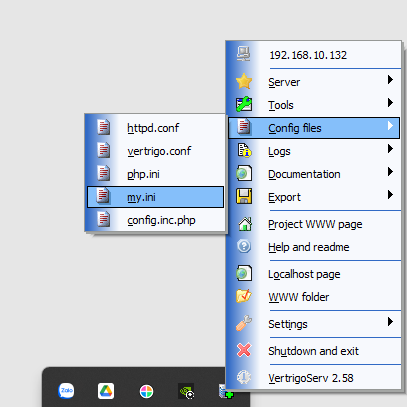
Phần 1: Soạn DB

# Đăng nhập vào DATABASE

Mật khẩu **root**

Password **vertrigo**

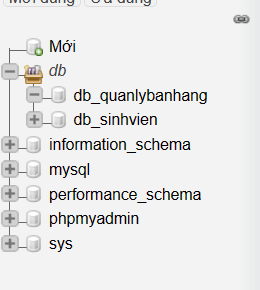
Trường hợp dính lỗi hy/0002 thì sửa lại cổng mặc định là 3306



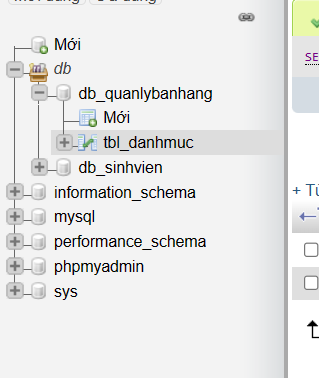
# Tạo database mới tên là db\_quanlybanhang

#### Import hoặc tạo mới database

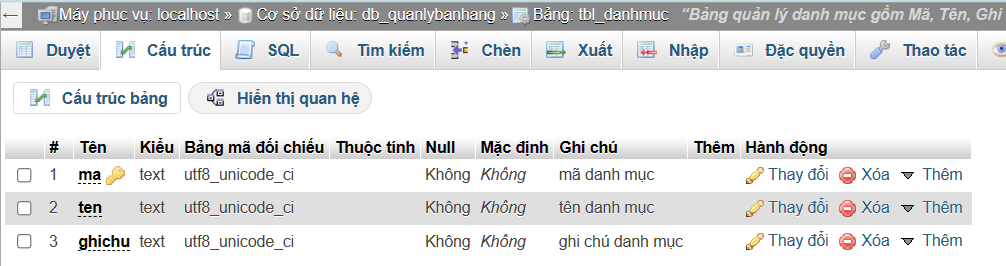




## Tạo bảng mới tên là tbl\_danhmuc



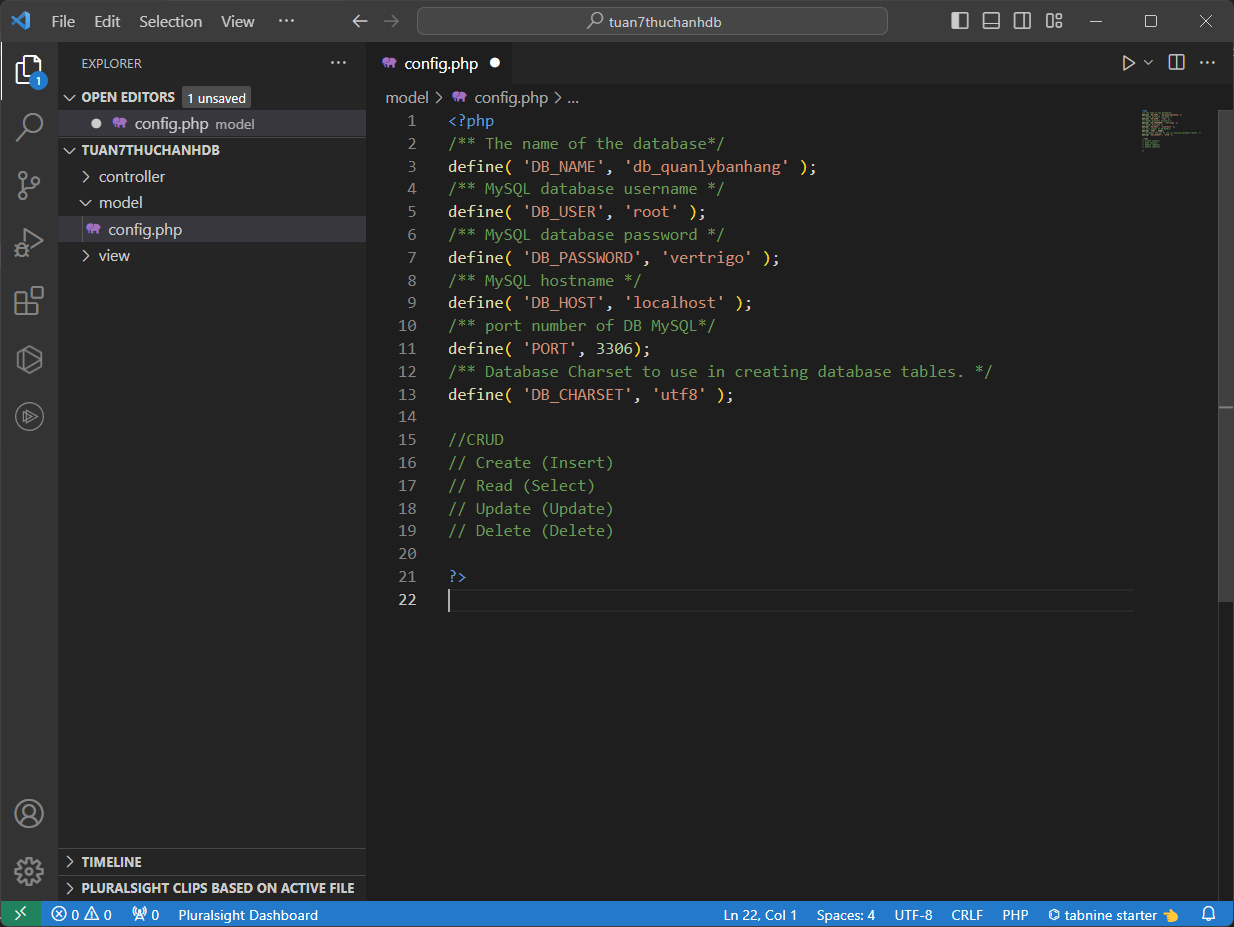
## Thêm 3 trường mã, tên, ghichu



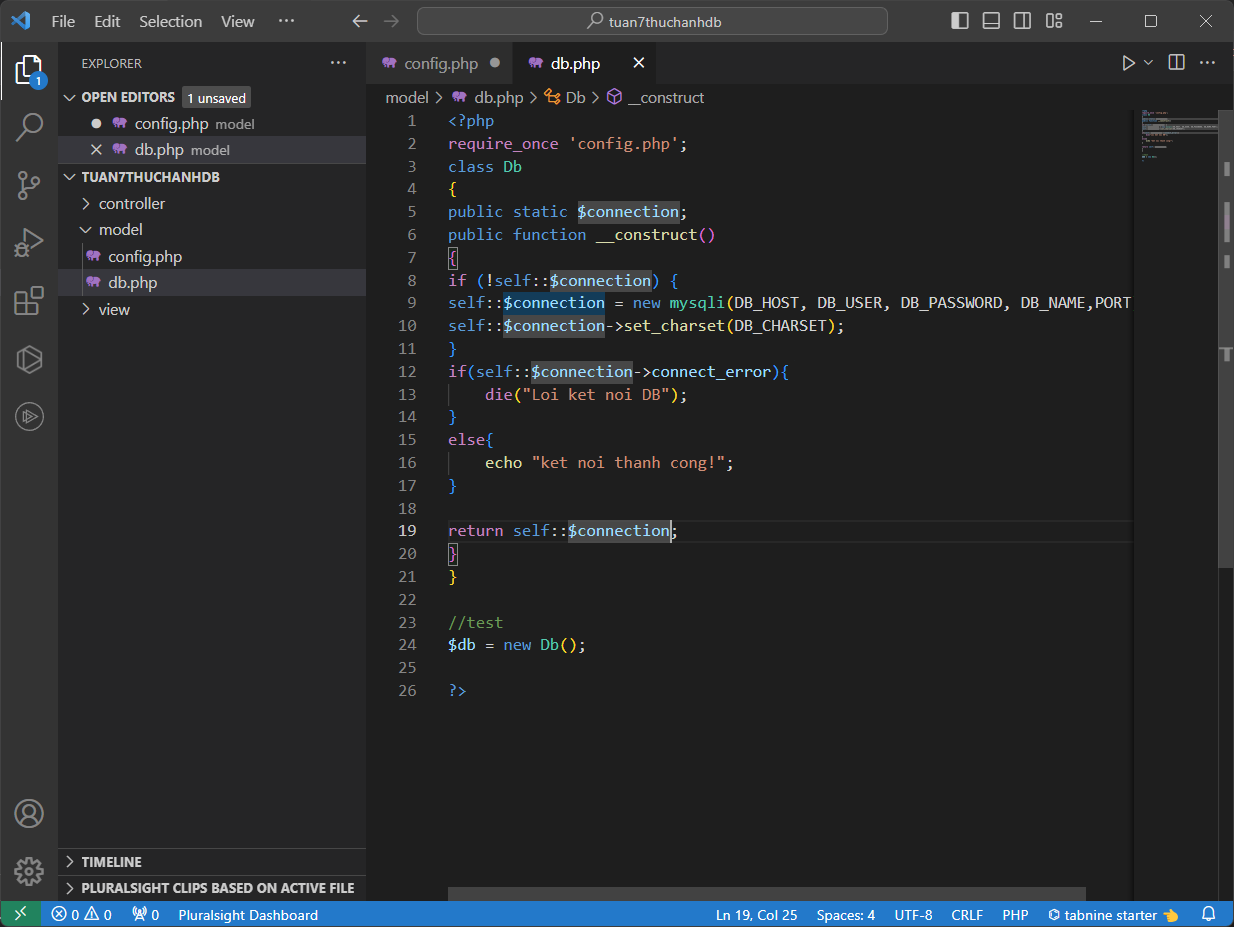
## Sao chép config kết nối tới database

[Link sao chép thư mục tham khảo CRUD của cô Hồng Mỹ](https://drive.google.com/drive/folders/1LEaDqfylJ87reigERIEYDardiDOP6XgW?usp=sharing)

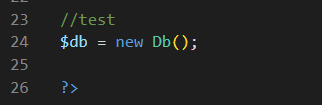
Sửa tên database lại cho đúng với cái đã tạo tên là db\_quanlybanhang

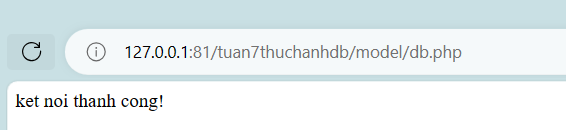


## Sao chép Class DB kết nối tới database



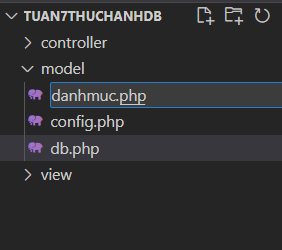
## Test kết nối





Kết nối thành công!

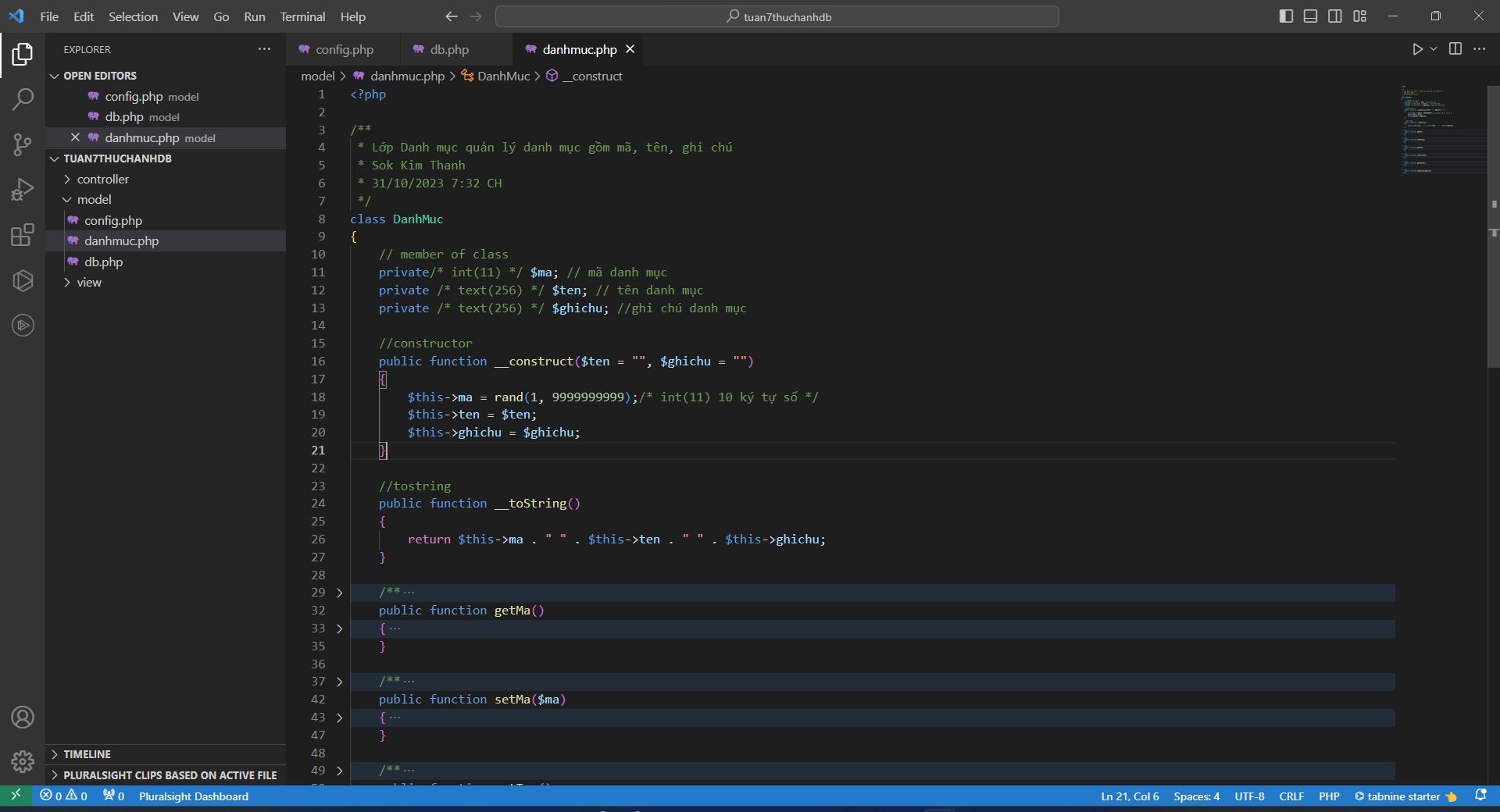
## Tạo lớp danh mục danhmuc.php



## Tạo cấu trúc cho lớp danh mục

#### Copy

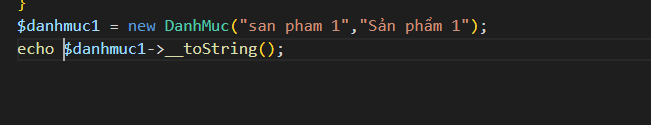
|  |
| --- |
| <?php  /\*\*   \* Lớp Danh mục quản lý danh mục gồm mã, tên, ghi chú   \* Sok Kim Thanh   \* 31/10/2023 7:32 CH   \*/  class DanhMuc  {      // member of class      private/\* text(11) \*/ $ma; // mã danh mục      private /\* text(256) \*/ $ten; // tên danh mục      private /\* text(256) \*/ $ghichu; //ghi chú danh mục      //constructor      public function \_\_construct($ma = "", $ten = "", $ghichu = "")      {          $this->ma = $ma;/\* text(11) 10 ký tự số chuỗi \*/          $this->ten = $ten;          $this->ghichu = $ghichu;      }      //tostring      public function \_\_toString()      {          return $this->ma . " " . $this->ten . " " . $this->ghichu;      }      /\*\*       \* Get the value of ma       \*/      public function getMa()      {          return $this->ma;      }      // /\*\*      //  \* Set the value of ma      //  \*      //  \* @return  self      //  \*/      // public function setMa($ma)      // {      //     $this->ma = $ma;      //     return $this;      // }      /\*\*       \* Get the value of ten       \*/      public function getTen()      {          return $this->ten;      }      /\*\*       \* Set the value of ten       \*       \* @return  self       \*/      public function setTen($ten)      {          $this->ten = $ten;          return $this;      }      /\*\*       \* Get the value of ghichu       \*/      public function getGhichu()      {          return $this->ghichu;      }      /\*\*       \* Set the value of ghichu       \*       \* @return  self       \*/      public function setGhichu($ghichu)      {          $this->ghichu = $ghichu;          return $this;      }  } |

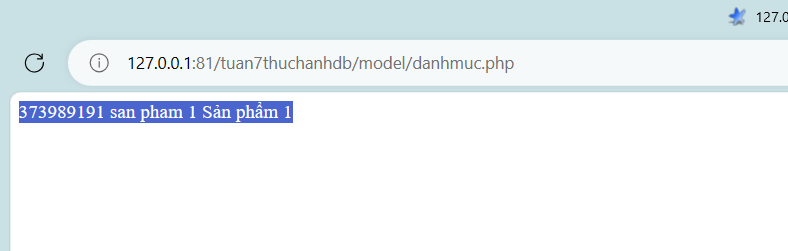


## Kiểm tra lớp danh mục

#### Copy

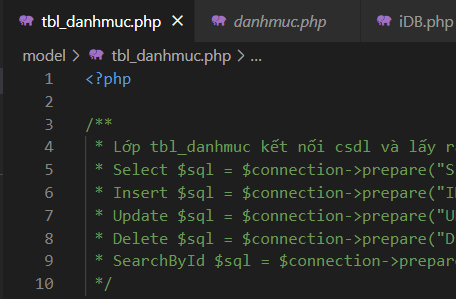
|  |
| --- |
| $danhmuc1 = new DanhMuc("san pham 1", "Sản phẩm 1");  echo $danhmuc1->\_\_toString(); |





## Viết mã CRUD (store cho tbl\_danhmuc) để copy vào lớp tbl\_danhmuc.php

Thao tác trên db để lấy ra chuỗi sql dán vào code sẵn để sử dụng dần



#### Copy

|  |
| --- |
| /\*\*       \* Lớp tbl\_danhmuc kết nối csdl và lấy ra toàn bộ thông tin của bảng       \* Select $sql = $connection->prepare("SELECT \* FROM tbl\_danhmuc");       \* Insert $sql = $connection->prepare("INSERT INTO `tbl\_danhmuc` (`ma`, `ten`, `ghichu`) VALUES ('2156465453', 'Sản phẩm', 'Danh mục sản phẩm'), ('2156465454', 'Loại Sản phẩm', 'Danh mục loại sản phẩm');");       \* Update $sql = $connection->prepare("UPDATE `tbl\_danhmuc` SET `ten` = 'Sản phẩm aaa' WHERE `tbl\_danhmuc`.`ma` = '2156465453';");       \* Delete $sql = $connection->prepare("DELETE FROM `tbl\_danhmuc` WHERE `tbl\_danhmuc`.`ma` = '2156465453';");       \* SearchById $sql = $connection->prepare("SELECT \* FROM `tbl\_danhmuc` WHERE `ma` LIKE '2156465453'");       \*/ |

## Lưu ý:

### chuyển số thành chuỗi trong php

|  |
| --- |
| [Bạn có thể sử dụng toán tử dấu chấm và nối một chuỗi với nó (và nó sẽ được gõ thành một chuỗi): $number = 123; $string = $number . '';](https://bing.com/search?q=chuy%e1%bb%83n+s%e1%bb%91+th%c3%a0nh+chu%e1%bb%97i+trong+php)[1](https://bing.com/search?q=chuy%E1%BB%83n+s%E1%BB%91+th%C3%A0nh+chu%E1%BB%97i+trong+php)  [Ngoài ra, bạn có thể sử dụng hàm strval() để chuyển đổi một số thành một chuỗi: $number = 123; $string = strval($number);](https://bing.com/search?q=chuy%e1%bb%83n+s%e1%bb%91+th%c3%a0nh+chu%e1%bb%97i+trong+php)[1](https://bing.com/search?q=chuy%E1%BB%83n+s%E1%BB%91+th%C3%A0nh+chu%E1%BB%97i+trong+php) |

### ➢ number\_format(number) - định dạng một giá trị số là gì?

|  |
| --- |
| Hàm number\_format() trong PHP được sử dụng để định dạng một giá trị số với các phân cách hàng nghìn và phân cách thập phân. Cú pháp của hàm như sau: number\_format(number,decimals,decimalpoint,separator). Trong đó:   * number: Giá trị số cần định dạng. * decimals: Số chữ số thập phân muốn hiển thị. Mặc định là 0. * decimalpoint: Dấu chấm thập phân muốn sử dụng. Mặc định là “.”. * separator: Dấu phân cách hàng nghìn muốn sử dụng. Mặc định là “,”.   Ví dụ: number\_format(1234567890.123456, 2, '.', ',') sẽ trả về chuỗi "1,234,567,890.12". |

## Tạo interface iDB.php

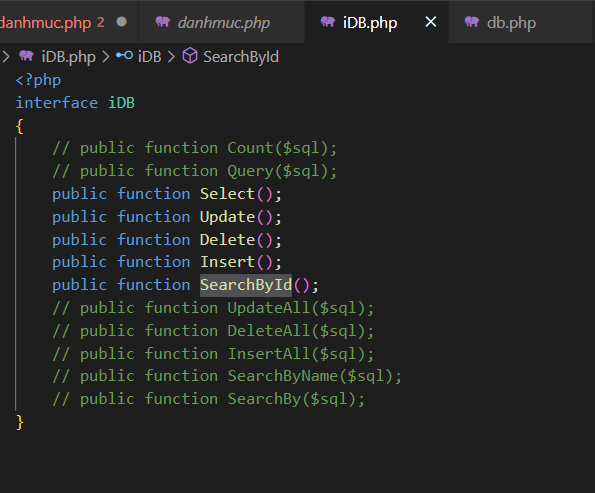
Tạo sẵn lớp ảo phương thức định nghĩa mà không thực thi

Lớp này định nghĩa sẵn các phương thức để không quên mình cần làm gì

#### Copy

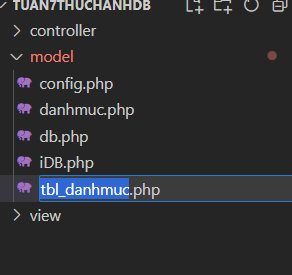
|  |
| --- |
| <?php  interface iDB  {      public function SelectAll();      public function SelectOne($id);      public function Insert($object);      public function Update($object);      public function Delete($id);  } |

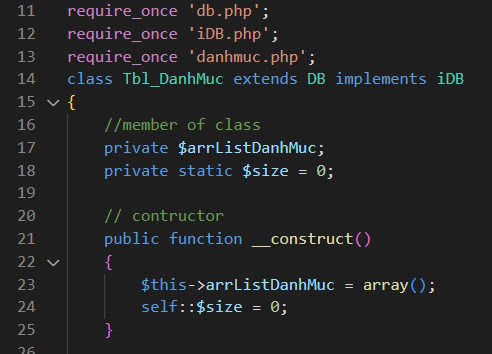
#### Ví dụ



## Tạo lớp tbl\_danhmuc.php

Đặt tên cho giống với database để dễ sử dụng



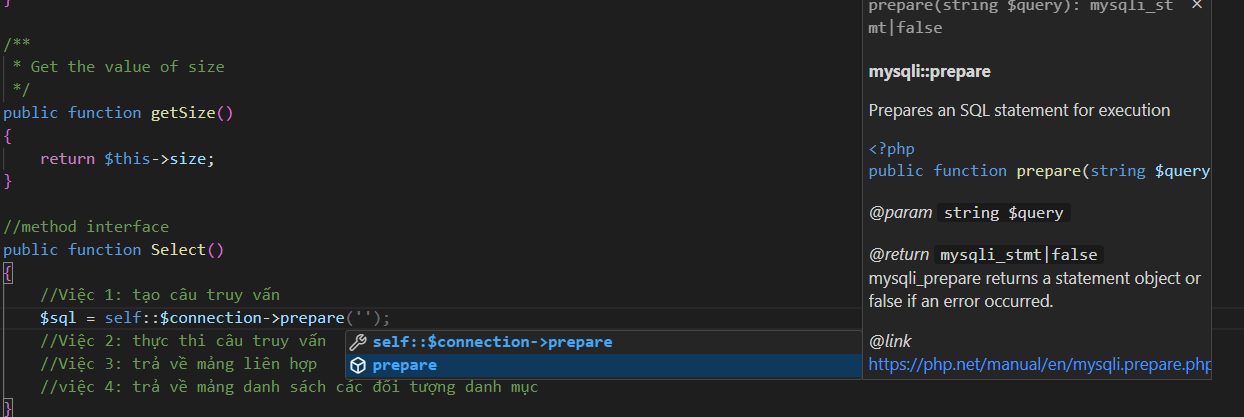


#### Copy

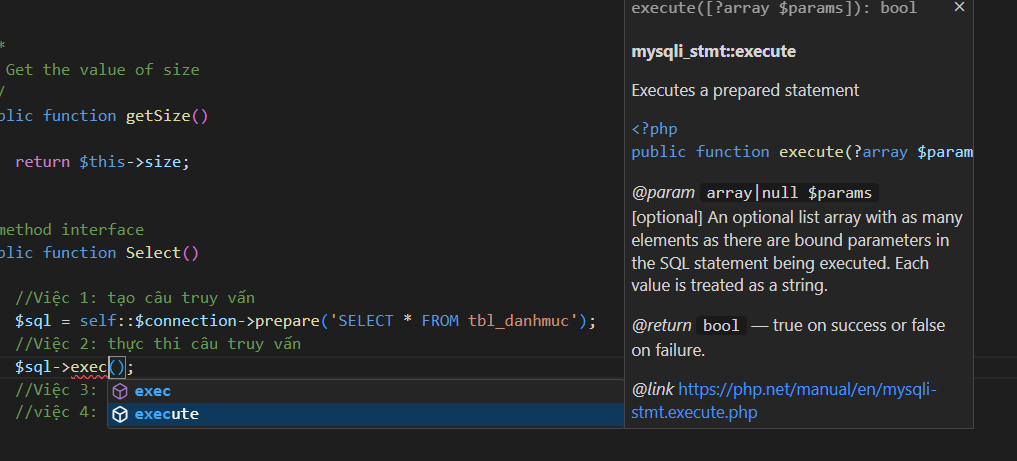
|  |
| --- |
| <?php  /\*\*   \* Lớp tbl\_danhmuc kết nối csdl và lấy ra toàn bộ thông tin của bảng(Thao tác trước trên phpmyadmin)   \* Select $sql = $connection->prepare("SELECT \* FROM tbl\_danhmuc");   \* Insert $sql = $connection->prepare("INSERT INTO `tbl\_danhmuc` (`ma`, `ten`, `ghichu`) VALUES ('2156465453', 'Sản phẩm', 'Danh mục sản phẩm'), ('2156465454', 'Loại Sản phẩm', 'Danh mục loại sản phẩm')");   \* Update $sql = $connection->prepare("UPDATE `tbl\_danhmuc` SET `ten` = 'Sản phẩm aaa' WHERE `tbl\_danhmuc`.`ma` = '2156465453'");   \* Delete $sql = $connection->prepare("DELETE FROM `tbl\_danhmuc` WHERE `tbl\_danhmuc`.`ma` = '2156465453'");   \* SearchById $sql = $connection->prepare("SELECT \* FROM `tbl\_danhmuc` WHERE `ma` LIKE '2156465453'");   \*   \*/  require\_once 'db.php';  require\_once 'iDB.php';  require\_once 'danhmuc.php';  class Tbl\_DanhMuc extends DB implements iDB  {      //member of class      private $arrListDanhMuc;      private static $size = 0;      private $table\_name;      // contructor      public function \_\_construct()      {          $this->arrListDanhMuc = array();          self::$size = 0;          $this->table\_name = 'tbl\_danhmuc';      }      // properties      /\*\*       \* Get the value of arrListDanhMuc       \*/      public function getArrListDanhMuc()      {          return $this->arrListDanhMuc;      }      /\*\*       \* Set the value of arrListDanhMuc       \*       \* @return  self       \*/      public function setArrListDanhMuc($arrListDanhMuc)      {          $this->arrListDanhMuc = $arrListDanhMuc;          return $this;      }      /\*\*       \* Get the value of size       \*/      public function getSize()      {          return self::$size;      }      //method interface      public function SelectAll()      {          //Việc 1: tạo câu truy vấn          $sql = self::$connection->prepare("SELECT \* FROM {$this->table\_name}");          //Việc 2: thực thi câu truy vấn          if (!$sql->execute()) {              throw new Exception("Lệnh truy vấn lỗi");              return -1; //xử lý lỗi          }          //Việc 3: trả về mảng liên hợp          $array = array();          $array = $sql->get\_result()->fetch\_all(MYSQLI\_ASSOC);          //việc 4: trả về mảng danh sách các đối tượng danh mục          foreach ($array as $key => $value/\* mảng value \*/) {              //4.1 Khởi tạo đối tượng DanhMuc với các giá trị từ mảng $value              $danhmuc = new DanhMuc($value['ten'], $value['ghichu']);              //4.2 Thêm đối tượng DanhMuc vào mảng arrListDanhMuc              array\_push($this->arrListDanhMuc, $danhmuc);          }          //việc 5: ngắt kết nối câu lệnh sql          $sql->close();          return $this->arrListDanhMuc; //Trả về thành công      }      function isNullOrEmptyString($str)      {          return (!isset($str) || trim($str) === '');      }      /\*\*       \* Lấy ra 1 danh mục       \*/      public function SelectOne($id)      {          if ($this->isNullOrEmptyString($id)) {              throw new Exception("Không tồn tại id");              return -1;          }          //Việc 1: Tạo câu sql          $sql = self::$connection->prepare("SELECT \* FROM /\* tbl\_danhmuc \*/{$this->table\_name} WHERE ma LIKE /\* '2156465453' \*/ ?");          // $sql->bind\_param("type", $varibles);          // Type:          // ● i - Integer          // ● d - Double          // ● s - String          // ● b - Blob          //Việc 2: Thực thi sql với param          $sql->bind\_param(/\* type \*/"s", /\* $variable \*/ $id);          // việc 3: kiểm tra tồn tại đối tượng này trong db          if (!$sql->execute()) {              throw new Exception("Lệnh truy vấn lỗi");              return -2;          }          //Việc 4: trả về một dòng          $row = $sql->get\_result()->fetch\_assoc();          // việc 5: Tạo một đối tượng để chứa giá trị          $danhmuc = new DanhMuc($row['ma'], $row['ten'], $row['ghichu']);          // việc 6: Giải phóng bộ nhớ          $sql->close();          return $danhmuc; // Tìm thấy danh mục;      }      /\*\*       \* Thêm 1 danh mục       \* return int 0: ok -1:null $danhmuc; -2 sai cau truy van       \* mấy cái chấm hỏi gọi là biến tham chiếu       \*/      public function Insert(/\* DanhMuc \*/$bienThamChieu)      {          if ($this->isNullOrEmptyString($bienThamChieu)) {              throw new Exception("Không tồn tại id");              return -1;          }          // Việc 1: Tạo câu truy vấn Sql          $sql = self::$connection->prepare("INSERT INTO {$this->table\_name} VALUES (/\* '2156465453' \*/?,/\*  'Sản phẩm' \*/?, /\* 'Danh mục sản phẩm' \*/?)");          // Việc 1.1 Tạo biến để gán vào chấm hỏi   (ghi chú: hàm bind\_param chỉ chấp nhận biến tham chiếu)          $ma = $bienThamChieu->getMa();          $ten = $bienThamChieu->getTen();          $ghichu = $bienThamChieu->getGhichu();          // Việc 2: Thêm param          $sql->bind\_param("sss", $ma, $ten, $ghichu);          // Việc 3: Thực thi câu truy vấn          if (!$sql->execute()) {              throw new Exception("Lệnh truy vấn lỗi");              return -2;          }          // Giải phóng bộ nhớ          $sql->close();          return 0;      }      /\*\*       \* Cập nhật một danh mục       \*/      public function Update($bienThamChieu)      {          if ($this->isNullOrEmptyString($bienThamChieu)) {              throw new Exception("Không tồn tại id");              return -1;          }          // Việc 1: Tạo câu truy vấn Sql          $sql = self::$connection->prepare("UPDATE {$this->table\_name} SET ten = /\* 'Sản phẩm aaa' \*/?, ghichu = /\* Sản phẩm bbbb \*/ ? WHERE {$this->table\_name}.ma = /\* '2156465453' \*/?");          // Việc 1.1 Tạo biến để gán vào chấm hỏi   (ghi chú: hàm bind\_param chỉ chấp nhận biến tham chiếu)          $ma = $bienThamChieu->getMa();          $ten = $bienThamChieu->getTen();          $ghichu = $bienThamChieu->getGhichu();          // Việc 2: Thêm param          $sql->bind\_param("sss", $ten, $ghichu, $ma);          // Việc 3: Thực thi câu truy vấn          if (!$sql->execute()) {              throw new Exception("Lệnh truy vấn lỗi");              return -2;          }          // Giải phóng bộ nhớ          $sql->close();          return 0;      }      /\*\*       \* Xóa một danh mục       \*/      public function Delete($id)      {          if ($this->isNullOrEmptyString($id)) {              throw new Exception("Không tồn tại id");              return -1; // không tồn tại id          }          // Việc 1: Tạo câu truy vấn Sql          $sql = self::$connection->prepare("DELETE FROM {$this->table\_name} WHERE {$this->table\_name}.ma = ?/\* '2156465453' \*/");          // Việc 2: Thêm param          $sql->bind\_param("s", $id);          // Việc 3: Thực thi câu truy vấn          if (!$sql->execute()) {              throw new Exception("Lệnh truy vấn lỗi");              return -2; // lệnh truy vấn lỗi          }          // Giải phóng bộ nhớ          $sql->close();          return 0;      }  }  // test  // 31/10/2023 9:00 PM  $tbl\_danhmuc = new Tbl\_DanhMuc();  // print\_r($tbl\_danhmuc->SelectAll());  // 31/10/2023 10:00 PM  // $danhmuc = $tbl\_danhmuc->SelectOne(/\* $id \*/'2156465453');  // echo $danhmuc->\_\_toString();  // 31/10/2023 11:20pm  // $danhmuc = new DanhMuc("1234567895", "Danh mục số 3", "Ghi chú danh mục");  // $tbl\_danhmuc->Insert($danhmuc);  // 31/10/2023 11:40PM  // echo $tbl\_danhmuc->SelectOne("1234567895")->\_\_toString(); //danhmuc so 4 ghichu danh muc4  // $danhmuc = new DanhMuc("1234567895", "Danh mục số 5", "Ghi chú danh mục 5");  // $tbl\_danhmuc->Update($danhmuc);  // echo $tbl\_danhmuc->SelectOne("1234567895")->\_\_toString();  // 01/11/2023 00:00 SA  // $tbl\_danhmuc->Delete("1234567895");  try {      $result = $tbl\_danhmuc->SelectOne("1234567895");      if ($tbl\_danhmuc->Delete("sfa") == 0) {          echo "Xóa thành công";      } else {          echo "Xóa không thành công!";      }  } catch (Exception $ex) {      echo "Lỗi: " . $ex->getMessage();  } |

### Phân tích viết hàm Select

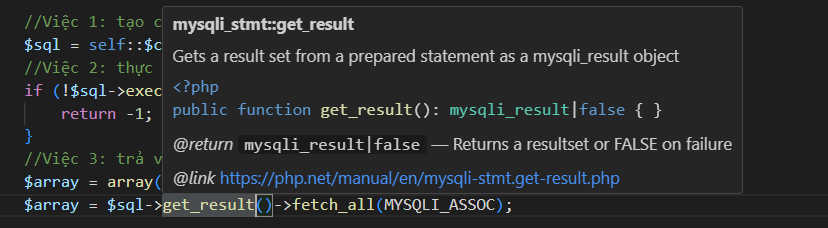
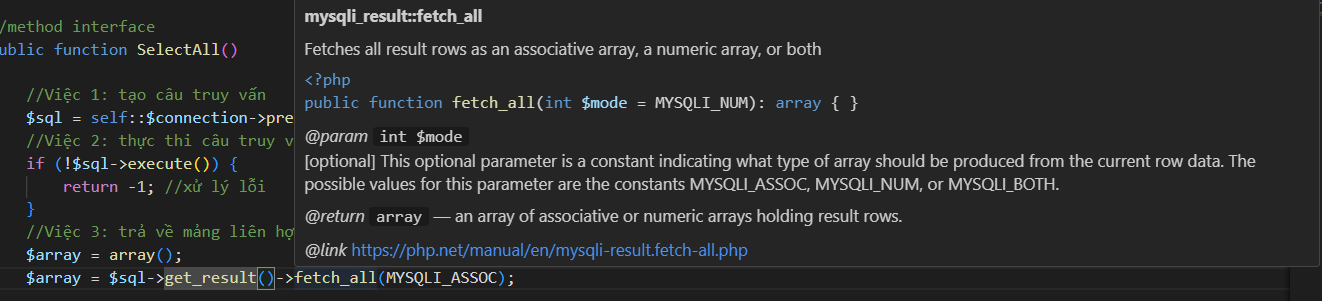
1. Tạo kết nối



1. Thực thi câu truy vấn



1. Trả về mảng liên hợp acsocciative

Copy

|  |
| --- |
| //method interface      public function SelectAll()      {          //Việc 1: tạo câu truy vấn          $sql = self::$connection->prepare('SELECT \* FROM tbl\_danhmuc');          //Việc 2: thực thi câu truy vấn          if (!$sql->execute()) {              return -1; //xử lý lỗi          }          //Việc 3: trả về mảng liên hợp          $array = array();          $array = $sql->get\_result()->fetch\_all(MYSQLI\_ASSOC); |

1. Trả về mảng danh sách các đối tượng danh mục

### Chú ý: $this->arrListDanhMuc = array(); là gì?

|  |
| --- |
| $this->arrListDanhMuc = array(); là một câu lệnh trong PHP để khởi tạo một mảng rỗng. Mảng là một kiểu dữ liệu trong PHP cho phép lưu trữ nhiều giá trị trong một biến duy nhất. Bạn có thể thêm các phần tử vào mảng này bằng cách sử dụng các hàm như array\_push() hoặc chỉ định trực tiếp các phần tử vào mảng.  Chúc bạn thành công! [1][2] |

### viết các ví dụ thêm, xóa, sửa, tìm kiếm cho mảng rỗng này.

|  |
| --- |
| Để thêm phần tử vào mảng, bạn có thể sử dụng hàm array\_push() hoặc chỉ định trực tiếp các phần tử vào mảng. Ví dụ:  <?php  $this->arrListDanhMuc = array();  array\_push($this->arrListDanhMuc, "Phần tử 1");  array\_push($this->arrListDanhMuc, "Phần tử 2");  $this->arrListDanhMuc[] = "Phần tử 3";  ?>  Để sửa giá trị của phần tử trong mảng, bạn có thể sử dụng chỉ số của phần tử đó. Ví dụ:  <?php  $this->arrListDanhMuc[0] = "Phần tử 1 đã được sửa";  ?>  Để xóa phần tử trong mảng, bạn có thể sử dụng hàm unset(). Ví dụ:  <?php  unset($this->arrListDanhMuc[0]);  ?>  Để tìm kiếm phần tử trong mảng, bạn có thể sử dụng hàm in\_array(). Ví dụ:  <?php  if (in\_array("Phần tử 2", $this->arrListDanhMuc)) {  echo "Tìm thấy phần tử 2 trong mảng";  } else {  echo "Không tìm thấy phần tử 2 trong mảng";  }  ?>  [Chúc bạn thành công](https://www.codehub.com.vn/PHP-xoa-phan-tu-khoi-mang)[1](https://www.codehub.com.vn/PHP-xoa-phan-tu-khoi-mang) |

Copy

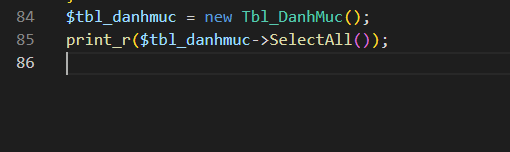
|  |
| --- |
| //việc 4: trả về mảng danh sách các đối tượng danh mục          foreach ($array as $key => $value/\* mảng value \*/) {              //4.1 Khởi tạo đối tượng DanhMuc với các giá trị từ mảng $value              $danhmuc = new DanhMuc($value['ten'], $value['ghichu']);              //4.2 Thêm đối tượng DanhMuc vào mảng arrListDanhMuc              array\_push($this->arrListDanhMuc, $danhmuc);          } |

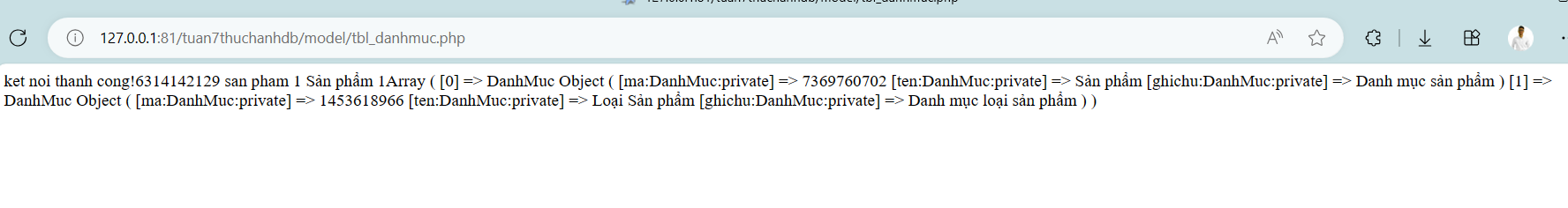
1. Đóng kết nối

Copy

|  |
| --- |
| //việc 5: ngắt kết nối câu lệnh sql          $sql->close();          return $this->arrListDanhMuc; //Trả về thành công      } |

## Test SelectAll()





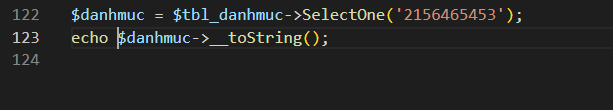
Kết nối thành công!

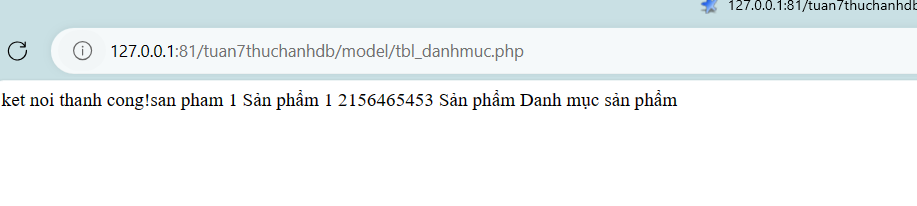
# SelectOne

Copy

|  |
| --- |
| /\*\*       \* Lấy ra 1 danh mục       \*/      public function SelectOne($id)      {          //Việc 1: Tạo câu sql          $sql = self::$connection->prepare("SELECT \* FROM tbl\_danhmuc WHERE ma LIKE /\* '2156465453' \*/ ?");          // $sql->bind\_param("type", $varibles);          // Type:          // ● i - Integer          // ● d - Double          // ● s - String          // ● b - Blob          //Việc 2: Thực thi sql với param          $sql->bind\_param(/\* type \*/"s", /\* $variable \*/ $id);          // việc 3: kiểm tra tồn tại đối tượng này trong db          if (!$sql->execute()) {              return -1;          }          //Việc 4: trả về một dòng          $row = $sql->get\_result()->fetch\_assoc();          // việc 5: Tạo một đối tượng để chứa giá trị          $danhmuc = new DanhMuc($row['ma'], $row['ten'], $row['ghichu']);          // việc 6: Giải phóng bộ nhớ          $sql->close();          return $danhmuc; // Tìm thấy danh mục;      } |

## Test SelectOne





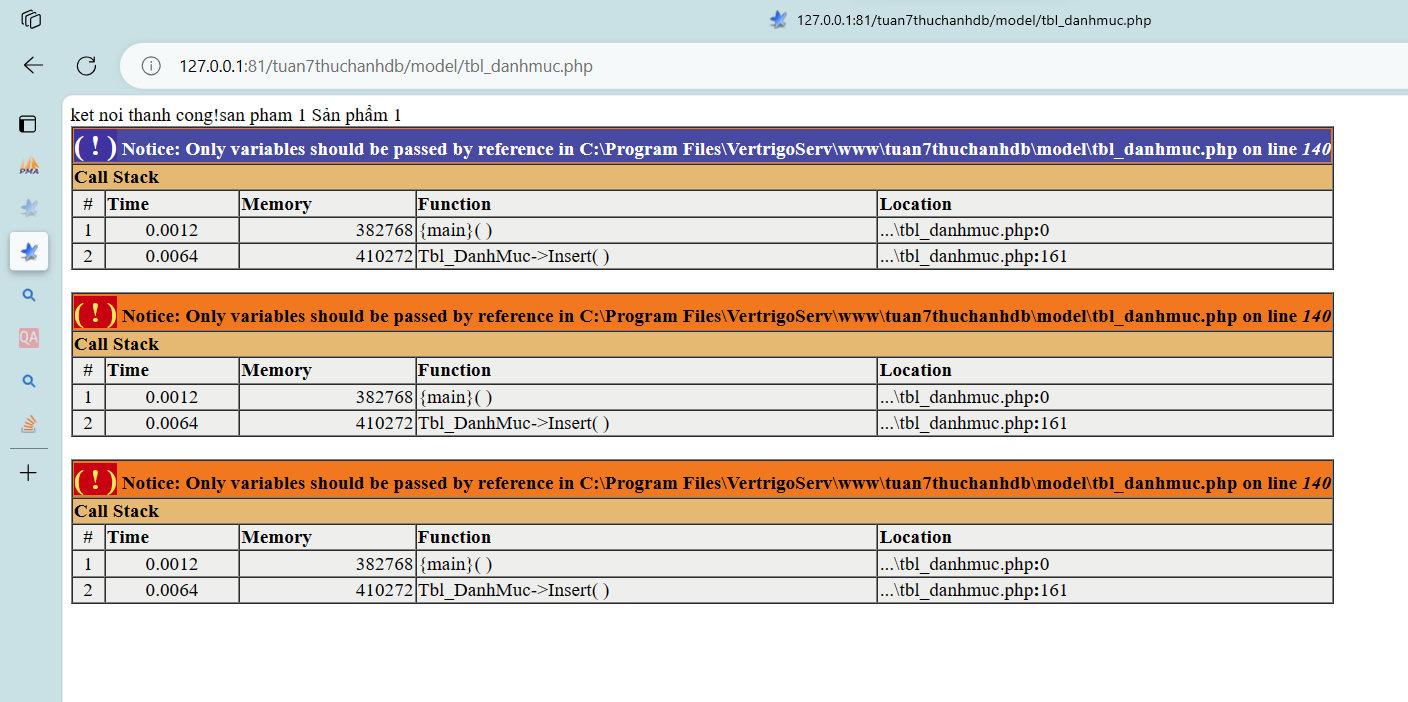
Kết nối thành công

# Insert

Copy

|  |
| --- |
| /\*\*       \* Thêm 1 danh mục       \*/      public function Insert(/\* object \*/DanhMuc $danhmuc)      {          // Việc 1: Tạo câu truy vấn Sql          $sql = self::$connection->prepare("INSERT INTO $this->table\_name VALUES (/\* '2156465453' \*/?,/\*  'Sản phẩm' \*/?, /\* 'Danh mục sản phẩm' \*/?)");          // Việc 2: Thêm param          $sql->bind\_param("sss", $danhmuc->getMa(), $danhmuc->getTen(), $danhmuc->getGhichu());          // Việc 3: Thực thi câu truy vấn          if (!$sql->execute()) {              return;          }            // Giải phóng bộ nhớ          $sql->close();      } |

Chú ý : Xử lý lỗi bind\_param không nhận phương thức



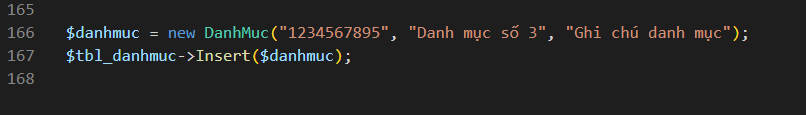
|  |
| --- |
| // Việc 1: Tạo câu truy vấn Sql  $sql = self::$connection->prepare("INSERT INTO $this->table\_name VALUES (/\* '2156465453' \*/?,/\* 'Sản phẩm' \*/?, /\* 'Danh mục sản phẩm' \*/?)");  // Việc 2: Thêm param  $sql->bind\_param("sss", $danhmuc->getMa(), $danhmuc->getTen(), $danhmuc->getGhichu()); Tại sao toi lại nhìn thấy lỗi này:"( ! ) Notice: Only variables should be passed by reference in C:\Program Files\VertrigoServ\www\tuan7thuchanhdb\model\tbl\_danhmuc.php on line 140  " |

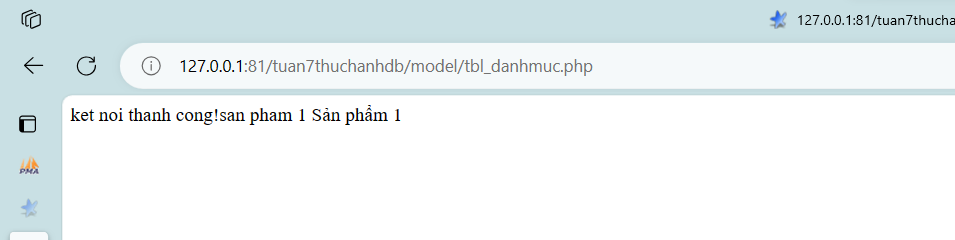
|  |
| --- |
| Thông báo lỗi này xuất hiện khi bạn truyền một giá trị không phải là biến vào phương thức bind\_param() của đối tượng $sql. Phương thức này yêu cầu các tham số được truyền vào phải là các biến tham chiếu, chứ không phải là các giá trị.  Để khắc phục lỗi này, bạn có thể tạo các biến để lưu giá trị của $danhmuc->getMa(), $danhmuc->getTen(), và $danhmuc->getGhichu() và sau đó truyền các biến này vào phương thức bind\_param(). Ví dụ:  $ma = $danhmuc->getMa();  $ten = $danhmuc->getTen();  $ghichu = $danhmuc->getGhichu();  $sql->bind\_param("sss", $ma, $ten, $ghichu);  Hy vọng điều này giúp ích cho bạn! |

Copy fix code Insert($danhmuc)

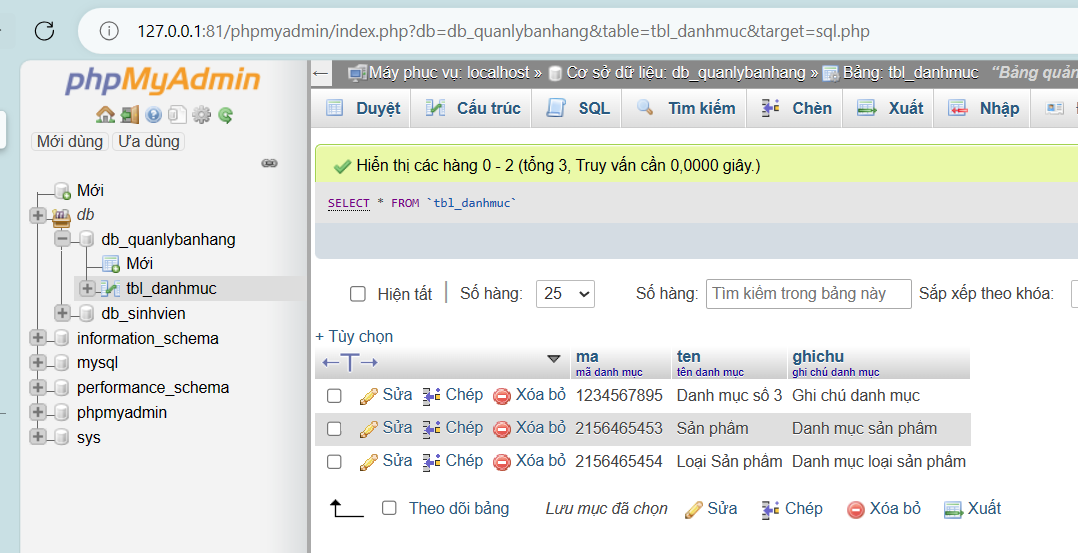
|  |
| --- |
| /\*\*       \* Thêm 1 danh mục       \* return int 0: ok -1:null $danhmuc; -2 sai cau truy van       \*/      public function Insert(/\* object \*/$danhmuc)      {          if ($danhmuc == null) {              return -1;          }          // Việc 1: Tạo câu truy vấn Sql          $sql = self::$connection->prepare("INSERT INTO $this->table\_name VALUES (/\* '2156465453' \*/?,/\*  'Sản phẩm' \*/?, /\* 'Danh mục sản phẩm' \*/?)");          // Việc 1.1 Tạo biến để gán vào chấm hỏi   (ghi chú: hàm bind\_param chỉ chấp nhận biến truyền vào)          $ma = $danhmuc->getMa();          $ten = $danhmuc->getTen();          $ghichu = $danhmuc->getGhichu();          // Việc 2: Thêm param          $sql->bind\_param("sss", $ma, $ten, $ghichu);          // Việc 3: Thực thi câu truy vấn          if (!$sql->execute()) {              return -2;          }          // Giải phóng bộ nhớ          $sql->close();          return 0;      } |

## Test Insert(/\*DanhMuc\*/$bienThamChieu)





Kết nối thành công

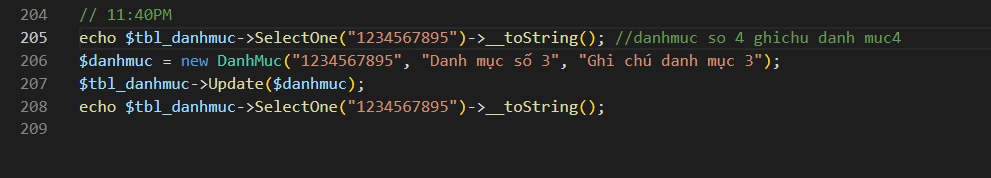


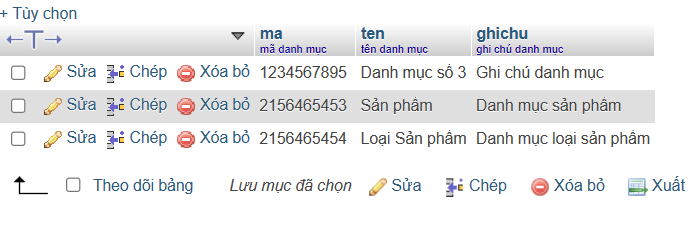
# Update

Copy

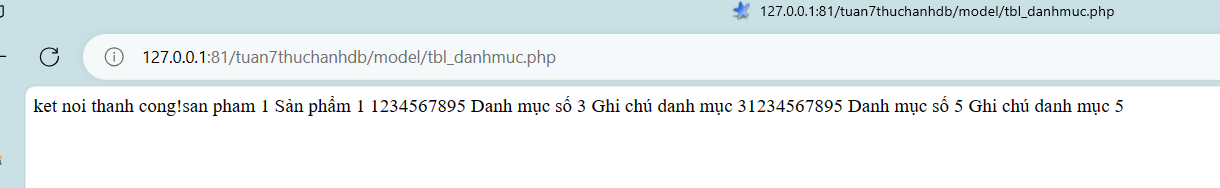
|  |
| --- |
| /\*\*       \* Cập nhật một danh mục       \*       \*/      public function Update($bienThamChieu)      {          if ($bienThamChieu == null) {              return -1;          }          // Việc 1: Tạo câu truy vấn Sql          $sql = self::$connection->prepare("UPDATE {$this->table\_name} SET ten = /\* 'Sản phẩm aaa' \*/?, ghichu = /\* Sản phẩm bbbb \*/ ? WHERE {$this->table\_name}.ma = /\* '2156465453' \*/?");          // Việc 1.1 Tạo biến để gán vào chấm hỏi   (ghi chú: hàm bind\_param chỉ chấp nhận biến tham chiếu)          $ma = $bienThamChieu->getMa();          $ten = $bienThamChieu->getTen();          $ghichu = $bienThamChieu->getGhichu();          // Việc 2: Thêm param          $sql->bind\_param("sss", $ten, $ghichu, $ma);          // Việc 3: Thực thi câu truy vấn          if (!$sql->execute()) {              return -2;          }          // Giải phóng bộ nhớ          $sql->close();          return 0;      } |

## Test Update









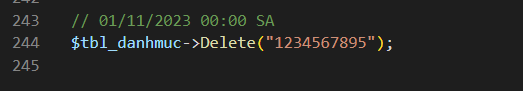
# Delete

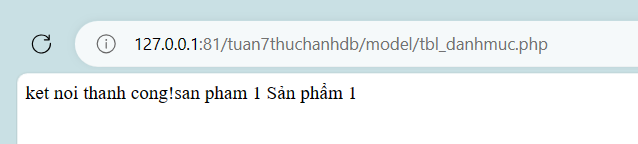
#### Copy

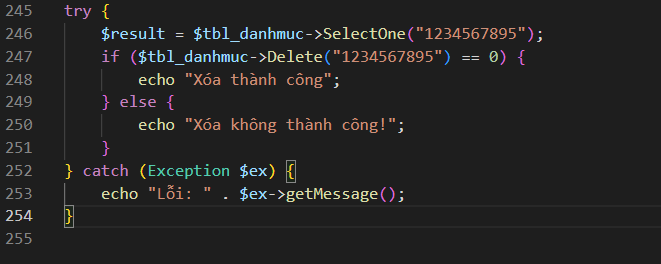
|  |
| --- |
| /\*\*       \* Xóa một danh mục       \*/      public function Delete($id)      {          if ($this->isNullOrEmptyString($id)) {              throw new Exception("Không tồn tại id");              return -1; // không tồn tại id          }          // Việc 1: Tạo câu truy vấn Sql          $sql = self::$connection->prepare("DELETE FROM {$this->table\_name} WHERE {$this->table\_name}.ma = ?/\* '2156465453' \*/");          // Việc 2: Thêm param          $sql->bind\_param("s", $id);          // Việc 3: Thực thi câu truy vấn          if (!$sql->execute()) {              throw new Exception("Lệnh truy vấn lỗi");              return -2; // lệnh truy vấn lỗi          }          // Giải phóng bộ nhớ          $sql->close();          return 0;      } |

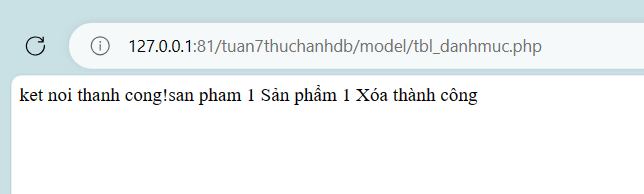
## Test Delete($id)

TH1: Xóa đúng id



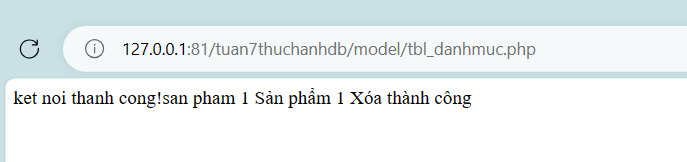


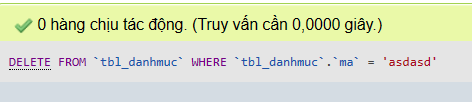




TH2: Xóa sai id

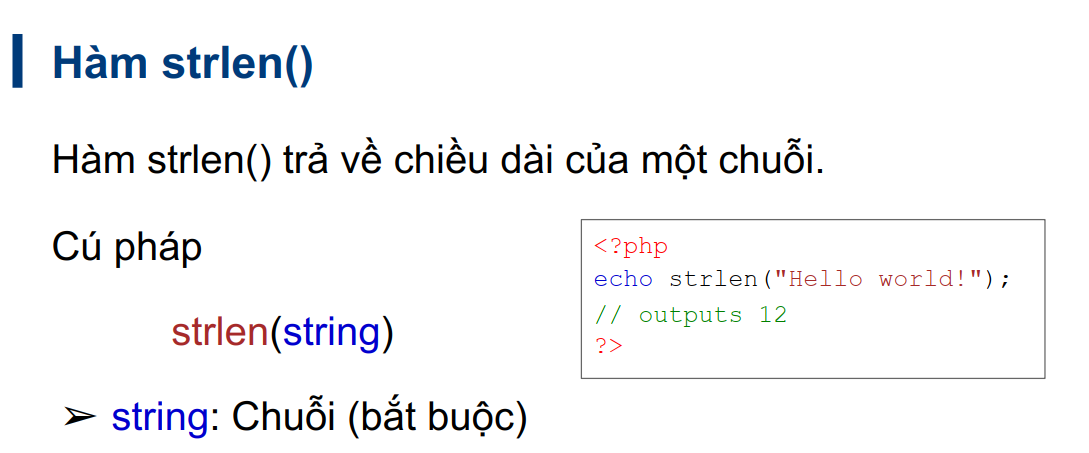
|  |
| --- |
| try {      $result = $tbl\_danhmuc->SelectOne("1234567895");      if ($tbl\_danhmuc->Delete("asdasd") == 0) {          echo "Xóa thành công";      } else {          echo "Xóa không thành công!";      }  } catch (Exception $ex) {      echo "Lỗi: " . $ex->getMessage();  } |

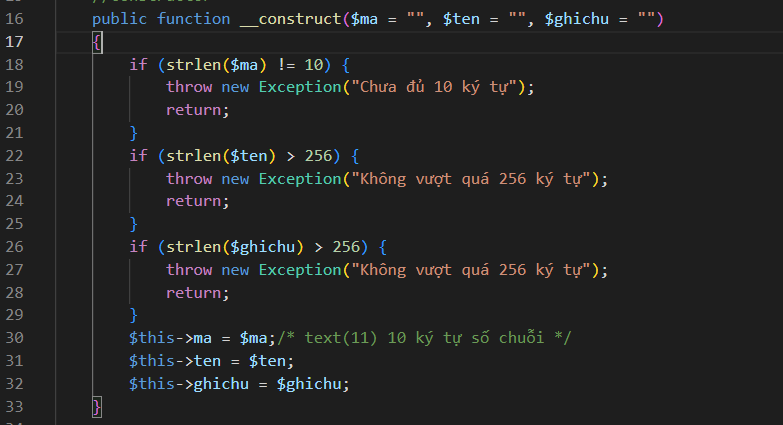




Phần 2: Soạn UI

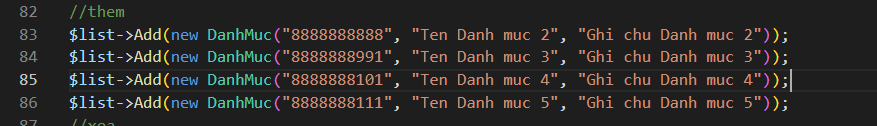
# Controller

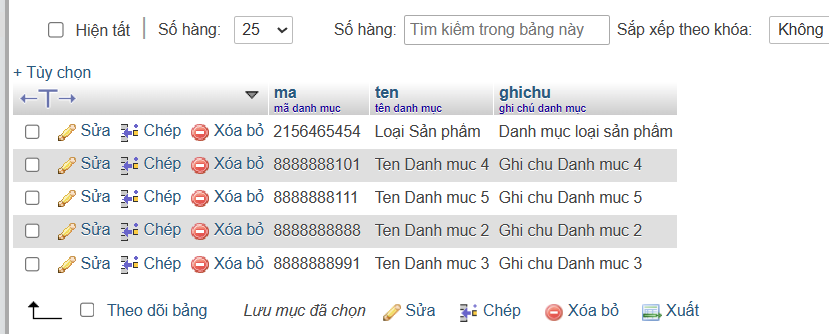




|  |
| --- |
| <?php  /\*\*   \* Sok Kim Thanh 22211tt0063 01/11/2023 8:33 SA   \* Interface: CRUD Controller   \*/  interface CRUDController  {      public function getList()/\* : array \*/;      public function Add($object)/\* : void \*/;      public function Edit($object)/\* : void \*/;      public function DeleteById($id)/\* : void \*/;      public function FindById($id)/\* : object \*/;  } |

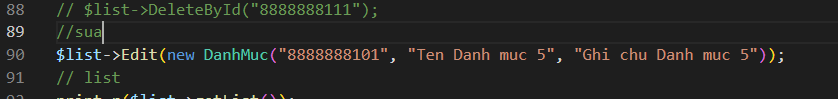
|  |
| --- |
| <?php  /\*\*   \* Sok Kim Thanh 2211tt0063 01/11/2023 8:34 SA   \* Controlller: Danh sách danh mục thêm xóa sửa tìm kiếm   \*/  include\_once '../model/tbl\_danhmuc.php';  include\_once 'CRUDController.php';  class DanhMucArrayAdapter implements CRUDController  {      private $tbl\_danhmuc; /\* db danh muc \*/      private $list;/\* List danh muc \*/      public function \_\_construct()      {          $this->tbl\_danhmuc = new Tbl\_DanhMuc();          $this->list = array();      }      public function getList()      {          try {              $this->list = $this->tbl\_danhmuc->SelectAll();              return $this->list;          } catch (Exception $e) {              echo $e->getMessage();          }      }      public function Add($object)      {          try {              if ($this->tbl\_danhmuc->Insert($object) == 0) {                  echo "Thêm thành công";              } else {                  echo "Thêm không thành công";              }          } catch (Exception $e) {              echo $e->getMessage();          }      }      public function Edit($object)      {          try {              if ($this->tbl\_danhmuc->Update($object) == 0) {                  echo "Sửa thành công";              } else {                  echo "Sửa không thành công";              }          } catch (Exception $e) {              echo $e->getMessage();          }      }      public function DeleteById($id)      {          try {              if ($this->tbl\_danhmuc->Delete($id) == 0) {                  echo "Xóa thành công";              } else {                  echo "Xóa không thành công";              }          } catch (Exception $e) {              echo $e->getMessage();          }      }      public function FindById($id)      {          try {              if ($this->tbl\_danhmuc->Delete($id) == 0) {                  echo "Tìm thành công";              } else {                  echo "Tìm không thành công";              }          } catch (Exception $e) {              echo $e->getMessage();          }      }  }  $list = new DanhMucArrayAdapter();  // test  // try {  //them  // $list->Add(new DanhMuc("8888888888", "Ten Danh muc 2", "Ghi chu Danh muc 2"));  // $list->Add(new DanhMuc("8888888991", "Ten Danh muc 3", "Ghi chu Danh muc 3"));  // $list->Add(new DanhMuc("8888888101", "Ten Danh muc 4", "Ghi chu Danh muc 4"));  // $list->Add(new DanhMuc("8888888111", "Ten Danh muc 5", "Ghi chu Danh muc 5"));  //xoa  // $list->DeleteById("8888888111");  //sua  $list->Edit(new DanhMuc("8888888101", "Ten Danh muc 5", "Ghi chu Danh muc 5"));  // list  print\_r($list->getList());  // } catch (Exception $ex) {      // echo $ex->getMessage();  // } |

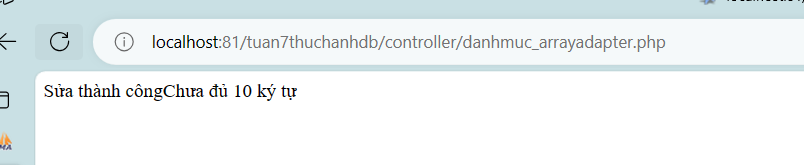














# View

## Bắt lỗi id, ten, ghi chu không hợp lệ

|  |
| --- |
| //constructor      public function \_\_construct($ma = "", $ten = "", $ghichu = "")      {          // xử lý mã tự động          $length = 11;          $characters = '01234567899';          $random\_string = substr(str\_shuffle($characters), 0, $length);          $ma = strlen($ma) != 11 ? $random\_string : $ma;          if (strlen($ma) != 11) {              throw new Exception("MA không hợp lệ.");          }          if (strlen($ten) > 256) {              throw new Exception("TEN không hợp lệ.");          }          if (strlen($ghichu) > 256) {              throw new Exception("GHICHU không hợp lệ.");          }          $this->ma = $ma;/\* text(11) 10 ký tự số chuỗi \*/          $this->ten = $ten;          $this->ghichu = $ghichu;      } |

## Ví dụ: Bắt lỗi nhập trùng id

|  |
| --- |
| $result = mysqli\_query($conn, "SELECT \* FROM table WHERE id = $id");  if (mysqli\_num\_rows($result) > 0) {  die("ID đã tồn tại trong cơ sở dữ liệu.");  }  if ($id\_exists) {  trigger\_error("ID đã tồn tại trong cơ sở dữ liệu.", E\_USER\_ERROR);  }  if ($id\_exists) {  die("ID đã tồn tại trong cơ sở dữ liệu.");  } |

### Chú ý: Dựa vào hàm selectone để kiểm tra id tồn tại trong csdl chưa để bắt trùng id

|  |
| --- |
| // việc 0: bắt lỗi nhập trùng id          $ma = $bienThamChieu->getMa();          if ($this->SelectOne($ma) == -3/\*Không trùng id \*/) {  //thực thi execute code ở đây          } else {              return INSERT\_ERROR;/\*-3 Trùng id \*/          } |

### Hàm insert db()

|  |
| --- |
| /\*\*       \* Thêm 1 danh mục       \* return 0: success -1:null $bienThamChieu; -2 thực thi câu truy van không thành công       \* mấy cái chấm hỏi gọi là biến tham chiếu       \*/      public function Insert(/\* DanhMuc \*/$bienThamChieu)      {          if ($this->isNullOrEmptyString($bienThamChieu)) {              return INPUT\_ERROR;/\* -1 Không tồn tại tham chiếu \*/          }          // việc 0: bắt lỗi nhập trùng id          $ma = $bienThamChieu->getMa();          if ($this->SelectOne($ma) == SELECT\_ERROR/\*Không trùng id \*/) {              // Việc 1: Tạo câu truy vấn Sql              $sql = self::$connection->prepare("INSERT INTO {$this->table\_name} VALUES (/\* '2156465453' \*/?,/\*  'Sản phẩm' \*/?, /\* 'Danh mục sản phẩm' \*/?)");              // Việc 1.1 Tạo biến để gán vào chấm hỏi   (ghi chú: hàm bind\_param chỉ chấp nhận biến tham chiếu)              $ten = $bienThamChieu->getTen();              $ghichu = $bienThamChieu->getGhichu();              // Việc 2: Thêm param              $sql->bind\_param("sss", $ma, $ten, $ghichu);              // Việc 3: Thực thi câu truy vấn              try {                  $sql->execute();              } catch (Exception $e) {                  throw new Exception($e->getMessage());                  // Giải phóng bộ nhớ                  $sql->close();                  return QUERY\_ERROR; //-2 // thực thi truy vấn không thành công              }              $sql->close(); // Giải phóng bộ nhớ              return SUCCESS; // 0          } else {              return INSERT\_ERROR;/\*-3 Trùng id \*/          }      } |

## config.php xử lý mã lỗi

Copy

|  |
| --- |
| <?php  /\*\* The name of the database\*/  define('DB\_NAME', 'db\_quanlybanhang');  /\*\* MySQL database username \*/  define('DB\_USER', 'root');  /\*\* MySQL database password \*/  define('DB\_PASSWORD', 'vertrigo');  /\*\* MySQL hostname \*/  define('DB\_HOST', 'localhost');  /\*\* port number of DB MySQL\*/  define('PORT', 3306);  /\*\* Database Charset to use in creating database tables. \*/  define('DB\_CHARSET', 'utf8');  //CRUD  // Create (Insert)  // Read (Select)  // Update (Update)  // Delete (Delete)  /\*\*Sok Kim Thanh 1/11/2023 6:46CH \*/  /\*\* error\_code lỗi đầu vào\*/  define('INPUT\_ERROR', -1);  /\*\* error\_code thực thi câu truy vấn không thành công\*/  define('QUERY\_ERROR', -2);  define('SUCCESS', 0);  /\*\* error\_code thực thi câu truy vấn select không tìm thấy id tồn tại\*/  define('SELECT\_ERROR', -3);  /\*\* error\_code thực thi câu truy vấn insert tìm thấy id tồn tại\*/  define('INSERT\_ERROR', -3); |

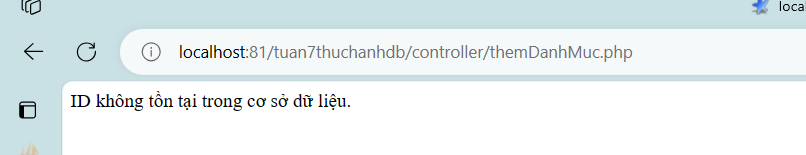
## Add controller

|  |
| --- |
| public function Add($object)      {          if ($this->tbl\_danhmuc->Insert($object) == INPUT\_ERROR) {              throw new Exception("Không tồn tại biến tham chiếu");              return INPUT\_ERROR;          } else if ($this->tbl\_danhmuc->Insert($object) == QUERY\_ERROR) {              throw new Exception("Thực thi truy vấn thất bại");              return QUERY\_ERROR;          } else if ($this->tbl\_danhmuc->Insert($object) == INSERT\_ERROR) {              throw new Exception("ID đã tồn tại trong cơ sở dữ liệu.");              return INSERT\_ERROR;          } else {              echo SUCCESS . ":Thêm thành công";              return SUCCESS;          }      } |

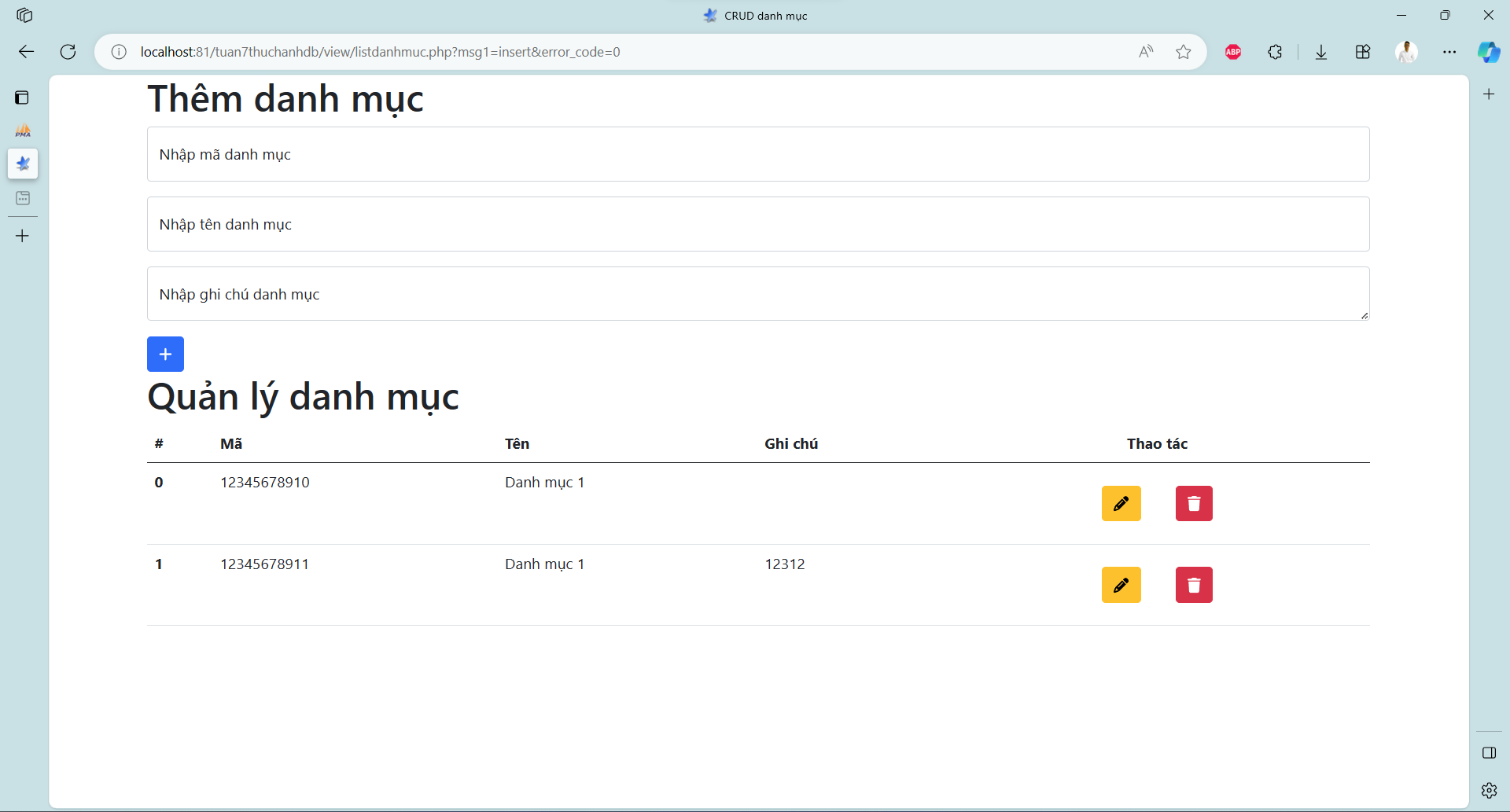
## themdanhmuc.php

|  |
| --- |
| <?php  /\*\*   \* Sok Kim Thanh 7:14 CH   \* Thêm danh mục   \*/  include\_once "DanhMucController.php";  $list = new DanhMucController();  $err\_code = "";  // test  try {      if (isset($\_POST['ma'])) {          if (!isset($\_POST['ten']) && !isset($\_POST['ghichu'])) {              return;          }          $ma = $\_POST['ma'];          $ten = $\_POST['ten'];          $ghichu = $\_POST['ghichu'];          $danhmuc = new DanhMuc($ma, $ten, $ghichu);          $err\_code = $list->Add($danhmuc);          header("Location: ../view/listdanhmuc.php?msg1=insert&error\_code={$err\_code}");      }  } catch (Exception $ex) {      echo $ex->getMessage();  } |

## Test themdanhmuc.php



Id này được phép thêm vào csdl vì không tìm thấy cái nào như vậy



# Message error code

## listdanhmuc.php

|  |
| --- |
| <?php      if (isset($\_GET['msg1']) == "insert") {        echo "<div class='alert alert-success alert-dismissible'>                <button type='button' class='close' data-dismiss='alert'>&times;</button>                Record ADDED successfully              </div>";      }      if (isset($\_GET['msg2']) == "update") {        echo "<div class='alert alert-success alert-dismissible'>                <button type='button' class='close' data-dismiss='alert'>&times;</button>                Record UPDATED successfully              </div>";      }      if (isset($\_GET['msg3']) == "delete") {        echo "<div class='alert alert-success alert-dismissible'>                <button type='button' class='close' data-dismiss='alert'>&times;</button>                Record DELETED successfully              </div>";      }      ?> |

Test thông báo

