

## Lab 4 – CRUD - Insert, Query, Update and Delete

### Aim

The aim of this lab is using “product category” as an example to show the basic operation of data stored in database. We will use the code from last lab and today’s code will be used in future labs.

### Resources

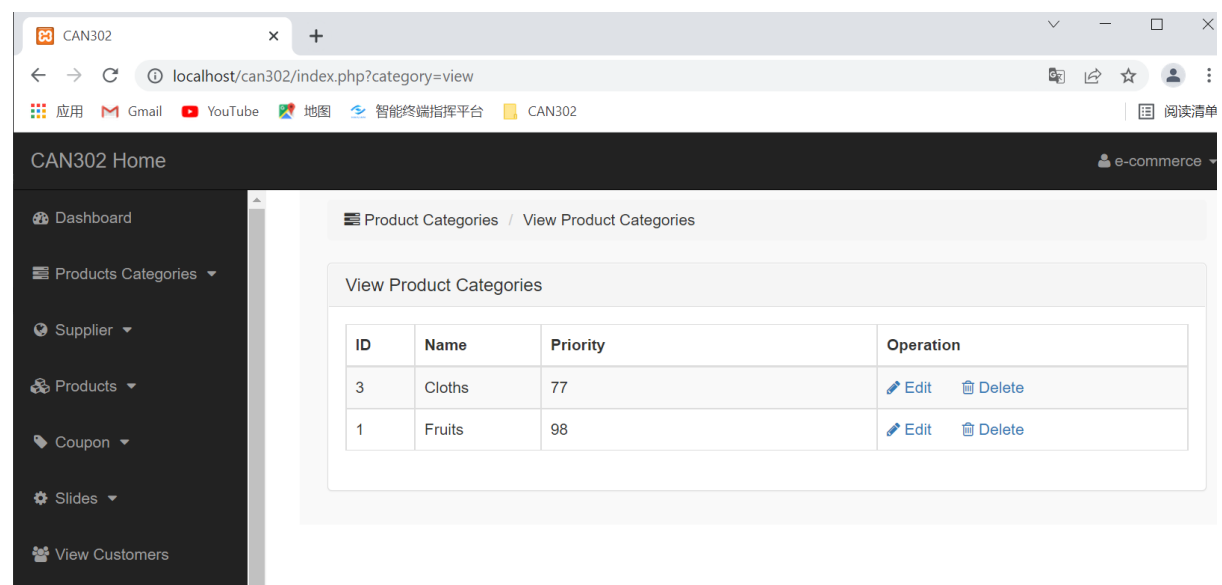
1. You should continuously use the all settings of last lab.

### Tips:

1. If you are not sure why you are doing something, ask a TA. This is what they are here for.
2. Our labs have different focus than the M-Dev-Store online videos. While if you want to be an expert, you are recommended take both labs and on-line videos.
3. The forums are available for questions and discussions.
4. These labs are expected take more than the 2 allocated hours. You should complete them in your own time before the next lab. Practice makes perfect!

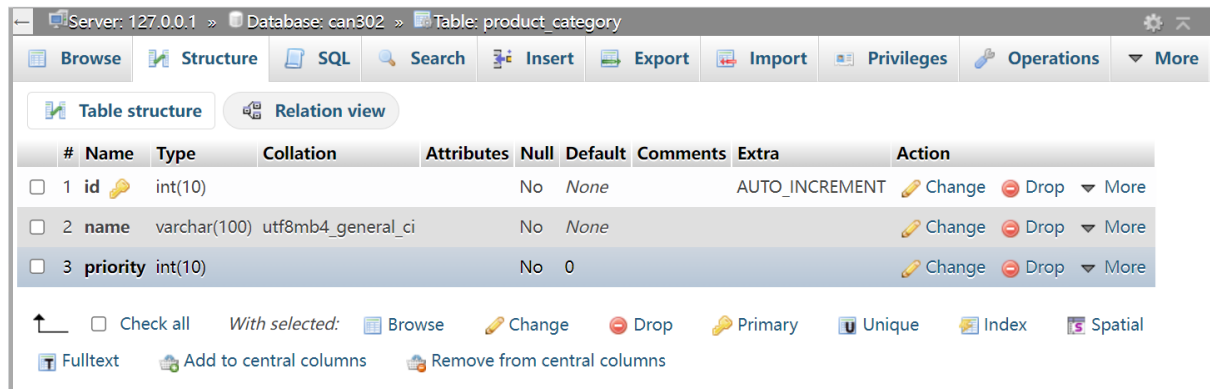
### The final effect of this lab:

Let’s make “product categories” with its full functions. After this lab, you can manage the product categories by webpages. We can view them like this:



## Prepare the table in database:

1. Again, we need to create a table in the database to store the info of product categories. Open <http://localhost/phpmyadmin/> and create a table named “product\_category” as:



The screenshot shows the phpMyAdmin interface for a database named 'can302'. The 'Table structure' tab is selected for the 'product\_category' table. The table has three columns: 'id' (int(10), primary key, auto-increment), 'name' (varchar(100), utf8mb4\_general\_ci), and 'priority' (int(10)).

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(10)			No	None		AUTO_INCREMENT	Change Drop More
2	name	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
3	priority	int(10)			No	0			Change Drop More

Below the table structure, there are options to 'Check all', 'With selected:', and various actions like 'Browse', 'Change', 'Drop', 'Primary', 'Unique', 'Index', 'Spatial', 'Fulltext', 'Add to central columns', and 'Remove from central columns'.

Priority is an int type column, which can be used for ranking the order.

## View the product categories:

2. Create a new folder named “category”, we will put all category related php files here. Create a file named “view.php” under folder “category”. The code to show the data stored in database is similar to that in lab1. Some key code is as:

```
<div class="row"><!-- row begin -->
  <div class="col-lg-12"><!-- col-lg-12 begin -->
    <ol class="breadcrumb">
      <li><i class="fa fa-tasks"></i> Product Categories</li>
      <li>View Product Categories</li>
    </ol>
  </div><!-- col-lg-12 finish -->
</div><!-- row finish -->
```

Here we use the class “row” to have a row. In this row, we declare the current functions in 2<sup>nd</sup> orders by “breadcrumb”.

[illegible]

The code of table was in another row. You may see some php code embedded in html code and very similar to the one in lab1. Such code will output the table stored in database. In the SQL statement, there is a magic word “order by”, it define the sorting rules. Please notice the output.

Add (insert) product category:

3. We need a form to add a new product category. Create a file “add.php” and the key code is as:

```
<?php
if (!isset($_SESSION['can302'])) {
    echo "<script>window.open('login.php','_self')</script>";
}
elseif (isset($_POST['add'])) {
    $p_cat_name = $_POST['p_cat_name'];
    $p_cat_priority = $_POST['p_cat_priority'];
    $insert_p_cat = "insert into product_category(name, priority)values('$p_cat_name','$p_cat_priority')";
    $run_p_cat = mysqli_query($con,$insert_p_cat);
    if ($run_p_cat) {
        echo "<script>alert('One product category successfully added!')</script>";
        echo "<script>window.open('index.php?category=view','_self')</script>";
    }
}
}
```

Once there is a new post, take the value to insert to database.

The code for form is as:

```
<div class="row"><!-- row begin -->|
  <div class="col-lg-12"><!-- col-lg-12 begin -->
    <div class="panel panel-default"><!-- panel panel-default begin -->
      <div class="panel-heading"><!-- panel-heading begin -->
        <h3 class="panel-title"> Add Product Category </h3>
      </div><!-- panel-heading finish -->
      <div class="panel-body"><!-- panel-body begin -->
        <form method="post" class="form-horizontal">
          <div class="form-group"><!-- form-group begin -->
            <label class="control-label col-md-3">Name </label>
            <div class="col-md-6"><!-- col-md-6 begin -->
              <input type="text" name="p_cat_name" class="form-control">
            </div><!-- col-md-6 finish -->
          </div><!-- form-group finish -->
          <div class="form-group"><!-- form-group begin -->
            <label class="control-label col-md-3">Priority</label>
            <div class="col-md-6"><!-- col-md-6 begin -->
              <input type="text" name="p_cat_priority" class="form-control">
            </div><!-- col-md-6 finish -->
          </div><!-- form-group finish -->
          <div class="form-group"><!-- form-group begin -->
            <label class="control-label col-md-3"> </label>
            <div class="col-md-6"><!-- col-md-6 begin -->
              <input type="submit" name="add" class="form-control btn btn-primary" value="Submit">
            </div><!-- col-md-6 finish -->
          </div><!-- form-group finish -->
        </form>
      </div><!-- panel-body finish -->
    </div><!-- panel panel-default finish -->
  </div><!-- col-lg-12 finish -->
</div><!-- row finish -->
```

When you finish the “add.php”, please intergrate it in “index.php” and try to add some data.

### Update the product category:

- At “view.php”, we have two operations for each product category: edit and delete. Create a “edit.php” for the edit purpose. Its webpage looks like:

The screenshot displays the CAN302 web application interface. The browser's address bar shows the URL: `localhost/can302/index.php?category=edit&id=3`. The application has a dark sidebar on the left with navigation links: Dashboard, Products Categories, Supplier, Products, Coupon, and Slides. The main content area has a header 'Product Categories / Edit Product Category'. Below this is a form titled 'Edit Product Category' with two input fields: 'Name' containing the text 'Cloths' and 'Priority' containing the number '77'. At the bottom of the form is a blue 'Update' button.

Basically, it is also a form-post process similar to “add.php”. The main difference is it need to retrieve the current info from database and show them in the form. The key code is as:

```

<?php
if (!isset($_SESSION['can302'])) {
    echo "<script>window.open('login.php','_self')</script>";
}
elseif (isset($_POST['update'])) {
    $p_cat_id = $_POST['p_cat_id'];
    $p_cat_name = $_POST['p_cat_name'];
    $p_cat_priority = $_POST['p_cat_priority'];
    $update_p_cat = "update product_category set name='$p_cat_name', priority='$p_cat_priority' where id='$p_cat_id'";
    $run_p_cat = mysqli_query($con,$update_p_cat);
    if ($run_p_cat) {
        echo "<script>alert('Product Category has been Updated.*)</script>";
        echo "<script>window.open('index.php?category=view','_self')</script>";
    }
}
}

```

Once there is post data, then take the corresponding parameters and use the magic word “update” to update the database. Then redirect to the “view.php”.

The code for form is as:

```

<div class="row"><!-- row begin -->
    <div class="col-lg-12"><!-- col-lg-12 begin -->
        <div class="panel panel-default"><!-- panel panel-default begin -->
            <div class="panel-heading"><!-- panel-heading begin -->
                <h3 class="panel-title">
                    <i class="fa fa-pencil fa-fw"></i> Edit Product Category
                </h3>
            </div><!-- panel-heading finish -->
            <div class="panel-body"><!-- panel-body begin -->
                <form method="post" class="form-horizontal">
                    <div class="form-group"><!-- form-group begin -->
                        <label class="control-label col-md-3"> Name </label>
                        <div class="col-md-6"><!-- col-md-6 begin -->
                            <input type="text" name="p_cat_name" class="form-control" value="<?php echo $p_cat_name; ?>">
                        </div><!-- col-md-6 finish -->
                    </div><!-- form-group finish -->
                    <div class="form-group"><!-- form-group begin -->
                        <label class="control-label col-md-3"> Priority </label>
                        <div class="col-md-6"><!-- col-md-6 begin -->
                            <input name="p_cat_priority" type="text" class="form-control" value="<?php echo $p_cat_priority; ?>">
                        </div><!-- col-md-6 finish -->
                    </div><!-- form-group finish -->
                    <div class="form-group"><!-- form-group begin -->
                        <label class="control-label col-md-3"> </label>
                        <div class="col-md-6"><!-- col-md-6 begin -->
                            <input type="submit" name="update" class="form-control btn btn-primary" value="Update">
                            <input type="hidden" name="p_cat_id" class="form-control" value="<?php echo $p_cat_id; ?>" >
                        </div><!-- col-md-6 finish -->
                    </div><!-- form-group finish -->
                </form>
            </div><!-- panel-body finish -->
        </div><!-- panel panel-default finish -->
    </div><!-- col-lg-12 finish -->
</div><!-- row finish -->

```

Remember to retrieve the data from database before use them.

## Delete product category:

5. Create a “delete.php” and the key code is as:

```

<?php
if (!isset($_SESSION['can302'])) {
    echo "<script>window.open('login.php','_self')</script>";
}
elseif(isset($_GET['id'])) {
    $delete_p_cat_id = $_GET['id'];
    $delete_p_cat = "delete from product_category where id='$delete_p_cat_id'";
    $run_p_cat = mysqli_query($con,$delete_p_cat);
    if ($run_p_cat) {
        echo "<script>alert('Delete one product category successfully.*)</script>";
        echo "<script>window.open('index.php?category=view','_self')</script>";
    }
}
?>

```

This will permanently delete the data in database. Nowadays the storage is relatively cheap and to avoid the potential “silly bugs”, most of systems won’t use true “delete”. People prefer to use a flag for delete and call it “soft delete” or “logical delete”.

Extend questions:

1. Try to add search function in “view.php” to search for certain product category. The code in lab1 will help.
2. Try to add multiple delete function in “view.php”, it means you can choose more than one product category and delete them by one click.