CodeIgniter - Working with Database

Like any other framework, we need to interact with the database very often and CodeIgniter makes this job easy for us. It provides rich set of functionalities to interact with database.

In this section, we will understand how the CRUD (Create, Read, Update, Delete) functions work with CodeIgniter. We will use **stud** table to select, update, delete, and insert the data in **stud** table.

Table Name: stud	
roll_no	int(11)
Name	varchar(30)

Connecting to a Database

We can connect to database in the following two way −

Automatic Connecting − Automatic connection can be done by using the file application/config/autoload.php. Automatic connection will load the database for each and every page. We just need to add the database library as shown below −

```
$autoload['libraries'] = array('library');
```

Manual Connecting − If you want database connectivity for only some of the pages, then we can go for manual connecting. We can connect to database manually by adding the following line in any class.

```
$this->load->database();
```

Here, we are not passing any argument because everything is set in the database config file application/config/database.php

Inserting a Record

To insert a record in the database, the insert() function is used as shown in the following table −

Syntax	insert([$table = ''[, $set = NULL[, $escape = NULL]]])
Parameters	**$table** (*string*) − Table name **$set** (*array*) − An associative array of field/value pairs **$escape** (*bool*) − Whether to escape values and identifiers
Returns	TRUE on success, FALSE on failure
Return Type	bool

The following example shows how to insert a record in **stud** table. The $data is an array in which we have set the data and to insert this data to the table **stud**, we just need to pass this array to the insert function in the 2nd argument.

```
$data = array(
    'roll_no' => '1',
    'name' => 'Virat'
);

$this->db->insert("stud", $data);
```

Updating a Record

To update a record in the database, the **update()** function is used along with **set()** and **where()** functions as shown in the tables below. The **set()** function will set the data to be updated.

Syntax	set($key[, $value = ''[, $escape = NULL]])
Parameters	**$key** (*mixed*) − Field name, or an array of field/value pairs **$value** (*string*) − Field value, if $key is a single field **$escape** (*bool*) − Whether to escape values and identifiers
Returns	CI_DB_query_builder instance (method chaining)
Return Type	CI_DB_query_builder

The **where()** function will decide which record to update.

Syntax	where($key[, $value = NULL[, $escape = NULL]])
Parameters	**$key** (*mixed*) – Name of field to compare, or associative array **$value** (*mixed*) – If a single key, compared to this value **$escape** (*bool*) – Whether to escape values and identifiers
Returns	DB_query_builder instance
Return Type	object

Finally, the **update()** function will update data in the database.

Syntax	update([$table = ''[, $set = NULL[, $where = NULL[, $limit = NULL]]]])
Parameters	**$table** (*string*) – Table name **$set** (*array*) – An associative array of field/value pairs **$where** (*string*) – The WHERE clause **$limit** (*int*) – The LIMIT clause
Returns	TRUE on success, FALSE on failure
Return Type	bool

```
$data = array(
    'roll_no' => '1',
    'name' => 'Virat'
);

$this->db->set($data);
$this->db->where("roll_no", '1');
$this->db->update("stud", $data);
```

Deleting a Record

To delete a record in the database, the delete() function is used as shown in the following table −

Syntax	delete([$table = ''[, $where = ''[, $limit = NULL[, $reset_data = TRUE]]]])
Parameters	**$table** (*mixed*) − The table(s) to delete from; string or array **$where** (*string*) − The WHERE clause **$limit** (*int*) − The LIMIT clause **$reset_data** (*bool*) − TRUE to reset the query "write" clause
Returns	CI_DB_query_builder instance (method chaining) or FALSE on failure
Return Type	mixed

Use the following code to to delete a record in the **stud** table. The first argument indicates the name of the table to delete record and the second argument decides which record to delete.

```
$this->db->delete("stud", "roll_no = 1");
```

Selecting a Record

To select a record in the database, the **get** function is used, as shown in the following table −

Syntax	get([$table = ''[, $limit = NULL[, $offset = NULL]]])
Parameters	**$table** (*string*) − The table to query array **$limit** (*int*) − The LIMIT clause **$offset** (*int*) − The OFFSET clause
Returns	CI_DB_result instance (method chaining)
Return Type	CI_DB_result

Use the following code to get all the records from the database. The first statement fetches all the records from "stud" table and returns the object, which will be stored in $query object. The second statement calls the **result()** function with $query object to get all the records as array.

```
$query = $this->db->get("stud");
$data['records'] = $query->result();
```

Closing a Connection

Database connection can be closed manually, by executing the following code −

```
$this->db->close();
```

Example

Create a controller class called **Stud_controller.php** and save it at **application/controller/Stud_controller.php**

Here is a complete example, wherein all of the above-mentioned operations are performed. Before executing the following example, create a database and table as instructed at the starting of this chapter and make necessary changes in the database config file stored at **application/config/database.php**

```php
<?php
   class Stud_controller extends CI_Controller {

      function __construct() {
         parent::__construct();
         $this->load->helper('url');
         $this->load->database();
      }

      public function index() {
         $query = $this->db->get("stud");
         $data['records'] = $query->result();

         $this->load->helper('url');
         $this->load->view('Stud_view',$data);
      }

      public function add_student_view() {
         $this->load->helper('form');
         $this->load->view('Stud_add');
      }

      public function add_student() {
         $this->load->model('Stud_Model');

         $data = array(
            'roll_no' => $this->input->post('roll_no'),
            'name' => $this->input->post('name')
         );
```

```php
        $this->Stud_Model->insert($data);

        $query = $this->db->get("stud");
        $data['records'] = $query->result();
        $this->load->view('Stud_view',$data);
    }

    public function update_student_view() {
        $this->load->helper('form');
        $roll_no = $this->uri->segment('3');
        $query = $this->db->get_where("stud",array("roll_no"=>$roll_no));
        $data['records'] = $query->result();
        $data['old_roll_no'] = $roll_no;
        $this->load->view('Stud_edit',$data);
    }

    public function update_student(){
        $this->load->model('Stud_Model');

        $data = array(
            'roll_no' => $this->input->post('roll_no'),
            'name' => $this->input->post('name')
        );

        $old_roll_no = $this->input->post('old_roll_no');
        $this->Stud_Model->update($data,$old_roll_no);

        $query = $this->db->get("stud");
        $data['records'] = $query->result();
        $this->load->view('Stud_view',$data);
    }

    public function delete_student() {
        $this->load->model('Stud_Model');
        $roll_no = $this->uri->segment('3');
        $this->Stud_Model->delete($roll_no);

        $query = $this->db->get("stud");
        $data['records'] = $query->result();
        $this->load->view('Stud_view',$data);
    }
  }
?>
```

Create a model class called **Stud_Model.php** and save it in **application/models/Stud_Model.php**

```php
<?php
   class Stud_Model extends CI_Model {

    function __construct() {
        parent::__construct();
    }

    public function insert($data) {
        if ($this->db->insert("stud", $data)) {
            return true;
        }
    }

    public function delete($roll_no) {
        if ($this->db->delete("stud", "roll_no = ".$roll_no)) {
            return true;
        }
```

```
      }

      public function update($data,$old_roll_no) {
         $this->db->set($data);
         $this->db->where("roll_no", $old_roll_no);
         $this->db->update("stud", $data);
      }
   }
?>
```

Create a view file called **Stud_add.php** and save it in application/views/Stud_add.php

```
<!DOCTYPE html>
<html lang = "en">

   <head>
      <meta charset = "utf-8">
      <title>Students Example</title>
   </head>
   <body>
         <?php
            echo form_open('Stud_controller/add_student');
            echo form_label('Roll No.');
            echo form_input(array('id'=>'roll_no','name'=>'roll_no'));
            echo "<br/>";

            echo form_label('Name');
            echo form_input(array('id'=>'name','name'=>'name'));
            echo "<br/>";

            echo form_submit(array('id'=>'submit','value'=>'Add'));
            echo form_close();
         ?>
   </body>
</html>
```

Create a view file called **Stud_edit.php** and save it in application/views/Stud_edit.php

```
<!DOCTYPE html>
<html lang = "en">

   <head>
      <meta charset = "utf-8">
      <title>Students Example</title>
   </head>

   <body>
      <form method = "" action = "">

         <?php
            echo form_open('Stud_controller/update_student');
            echo form_hidden('old_roll_no',$old_roll_no);
            echo form_label('Roll No.');
            echo form_input(array('id'=>'roll_no',
               'name'=>'roll_no','value'=>$records[0]->roll_no));
            echo "<br/>";

            echo form_label('Name');
            echo form_input(array('id'=>'name','name'=>'name',
               'value'=>$records[0]->name));
```

```
        echo "<br/>";

        echo form_submit(array('id'=>'submit','value'=>'Edit'));
        echo form_close();
      ?>

    </form>
  </body>

</html>
```

Create a view file called **Stud_view.php** and save it in **application/views/Stud_view.php**

```
<!DOCTYPE html>
<html lang = "en">

  <head>
    <meta charset = "utf-8">
    <title>Students Example</title>
  </head>

  <body>
    <a href = "<?php echo base_url(); ?>
      index.php/stud/add_view">Add</a>

    <table border = "1">
      <?php
        $i = 1;
        echo "<tr>";
        echo "<td>Sr#</td>";
        echo "<td>Roll No.</td>";
        echo "<td>Name</td>";
        echo "<td>Edit</td>";
        echo "<td>Delete</td>";
        echo "<tr>";

        foreach($records as $r) {
          echo "<tr>";
          echo "<td>".$i++."</td>";
          echo "<td>".$r->roll_no."</td>";
          echo "<td>".$r->name."</td>";
          echo "<td><a href = '".base_url()."index.php/stud/edit/"
            .$r->roll_no."'>Edit</a></td>";
          echo "<td><a href = '".base_url()."index.php/stud/delete/"
            .$r->roll_no."'>Delete</a></td>";
          echo "<tr>";
        }
      ?>
    </table>

  </body>

</html>
```

Make the following change in the route file at **application/config/routes.php** and add the following line at the end of file.

```
$route['stud'] = "Stud_controller";

$route['stud/add'] = 'Stud_controller/add_student';

$route['stud/add_view'] = 'Stud_controller/add_student_view';
```

```
$route['stud/edit/(\d+)'] = 'Stud_controller/update_student_view/$1';
$route['stud/delete/(\d+)'] = 'Stud_controller/delete_student/$1';
```

Now, let us execute this example by visiting the following URL in the browser. Replace the yoursite.com with your URL.

```
http://yoursite.com/index.php/stud
```

⊕ Previous Page   Next Page ⊕

Write for us   FAQ's   Helping   Contact

Enter email for newsletter   go