ike 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(‘database’);

* **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

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')

);

$this->Stud\_Model->insert($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

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 "

";

echo form\_label('Name');

echo form\_input(array('id'⇒'name','name'⇒'name',

'value'⇒$records[0]→name));

echo "

";

echo form\_submit(array('id'⇒'sub mit','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