 [Conéctate \(/login\)](#) o [Regístrate \(/registro\)](#)[Inicio \(/\)](#) > [Artículos \(/articulos\)](#) > [Operaciones CRUD con la extensión MySQLi \(/articulo/operaciones_crud_con_la_extension_mysql_1904\)](#)

Operaciones CRUD con la extensión MySQLi



CRUD significa Create, Read, Update and Delete, es decir, Crear, Leer, Modificar y eliminar. Las operaciones CRUD se suelen utilizar para manipular registros de la base de datos. Casi toda aplicación web que se precie utiliza las funciones de Añadir, Editar, Ver y Eliminar para gestionar su contenido. En este tutorial, veremos cómo implementar operaciones CRUD a una base de datos MySQL para realizar inserciones, lecturas, modificaciones y eliminaciones de registros en una aplicación web.

La extensión MySQLi y las técnicas de la programación orientada a objetos se utilizan para implementar la funcionalidad CRUD en PHP. Para ejemplificarlo, realizaremos en nuestro tutorial las siguientes operaciones en nuestros PHP.

- Obtener y mostrar los datos de un usuario de la base de datos MySQL
- Añadir usuarios en la base de datos
- Editar la información de un usuario
- Eliminar un usuario de la base de datos



Crear base de datos

Para almacenar la información del usuario necesitaremos crear una tabla en nuestra BD de MySQL. El siguiente SQL crea una tabla llamada users con columnas muy, pero que muy básicas.

```
CREATE TABLE `users` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `name` varchar(255) COLLATE utf8_unicode_ci NOT NULL,  
  `email` varchar(255) COLLATE utf8_unicode_ci NOT NULL,  
  `phone` varchar(15) COLLATE utf8_unicode_ci NOT NULL,  
  `created` datetime NOT NULL,  
  `modified` datetime NOT NULL,  
  `status` enum('1','0') COLLATE utf8_unicode_ci NOT NULL DEFAULT '1' COMMENT '1=Active, 0=Inactive'  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

Clase DataBase (DB.class.php)

La clase DB gestionará todas las operaciones relacionadas con la base de datos. Por ejemplo, conectarnos al servidor de la base de datos, insertar, modificar y eliminar registros en la base de datos... Para ello, debemos especificar el host de la base de datos (\$host), el nombre de usuario (\$dbUsername), la contraseña (\$dbPassword) y el nombre de la misma (\$dbName).



```

<?php
/*
 * DB Class
 * This class is used for database related (connect, insert, update, and delete) operations
 * @author CodexWorld.com
 * @url http://www.codexworld.com
 * @license http://www.codexworld.com/License
 */
class DB{
    private $dbHost = "localhost";
    private $dbUsername = "root";
    private $dbPassword = "*****";
    private $dbName = "programacionnet";

    public function __construct(){
        if(!isset($this->db)){
            // Connect to the database
            $conn = new mysqli($this->dbHost, $this->dbUsername, $this->dbPassword, $this->dbName);
            if($conn->connect_error){
                die("Failed to connect with MySQL: " . $conn->connect_error);
            }else{
                $this->db = $conn;
            }
        }
    }

    /*
     * Returns rows from the database based on the conditions
     * @param string name of the table
     * @param array select, where, order_by, limit and return_type conditions
     */
    public function getRows($table,$conditions = array()){
        $sql = 'SELECT ';
        $sql .= array_key_exists("select",$conditions)?$conditions['select']:'';
        $sql .= ' FROM '.$table;
        if(array_key_exists("where",$conditions)){
            $sql .= ' WHERE ';
            $i = 0;
            foreach($conditions['where'] as $key => $value){
                $pre = ($i > 0)?' AND ':'';
                $sql .= $pre.$key." = '".$value."'";
                $i++;
            }
        }

        if(array_key_exists("order_by",$conditions)){
            $sql .= ' ORDER BY '.$conditions['order_by'];
        }

        if(array_key_exists("start",$conditions) && array_key_exists("limit",$conditions)){
            $sql .= ' LIMIT '.$conditions['start'].','.$conditions['limit'];
        }elseif(!array_key_exists("start",$conditions) && array_key_exists("limit",$conditions)){
            $sql .= ' LIMIT '.$conditions['limit'];
        }

        $result = $this->db->query($sql);

        if(array_key_exists("return_type",$conditions) && $conditions['return_type'] != 'all'){

```

```

        switch($conditions['return_type']){
            case 'count':
                $data = $result->num_rows;
                break;
            case 'single':
                $data = $result->fetch_assoc();
                break;
            default:
                $data = '';
        }
    }else{
        if($result->num_rows > 0){
            while($row = $result->fetch_assoc()){
                $data[] = $row;
            }
        }
    }
    return !empty($data)?$data:false;
}

/*
 * Insert data into the database
 * @param string name of the table
 * @param array the data for inserting into the table
 */
public function insert($table,$data){
    if(!empty($data) && is_array($data)){
        $columns = '';
        $values = '';
        $i = 0;
        if(!array_key_exists('created',$data)){
            $data['created'] = date("Y-m-d H:i:s");
        }
        if(!array_key_exists('modified',$data)){
            $data['modified'] = date("Y-m-d H:i:s");
        }
        foreach($data as $key=>$val){
            $pre = ($i > 0)?', ':'';
            $columns .= $pre.$key;
            $values .= $pre."".$val.""";
            $i++;
        }
        $query = "INSERT INTO ".$table." ( ".$columns.") VALUES ( ".$values.")";
        $insert = $this->db->query($query);
        return $insert?$this->db->insert_id:false;
    }else{
        return false;
    }
}

/*
 * Update data into the database
 * @param string name of the table
 * @param array the data for updating into the table
 * @param array where condition on updating data
 */
public function update($table,$data,$conditions){
    if(!empty($data) && is_array($data)){

```


```

        $colvalSet = '';
        $whereSql = '';
        $i = 0;
        if(!array_key_exists('modified',$data)){
            $data['modified'] = date("Y-m-d H:i:s");
        }
        foreach($data as $key=>$val){
            $pre = ($i > 0)?', ':'';
            $colvalSet .= $pre.$key."='".$val."'";
            $i++;
        }
        if(!empty($conditions)&& is_array($conditions)){
            $whereSql .= ' WHERE ';
            $i = 0;
            foreach($conditions as $key => $value){
                $pre = ($i > 0)?' AND ':'';
                $whereSql .= $pre.$key." = '".$value."'";
                $i++;
            }
        }
        $query = "UPDATE ".$table." SET ".$colvalSet.$whereSql;
        $update = $this->db->query($query);
        return $update?$this->db->affected_rows:false;
    }else{
        return false;
    }
}

/*
 * Delete data from the database
 * @param string name of the table
 * @param array where condition on deleting data
 */
public function delete($table,$conditions){
    $whereSql = '';
    if(!empty($conditions)&& is_array($conditions)){
        $whereSql .= ' WHERE ';
        $i = 0;
        foreach($conditions as $key => $value){
            $pre = ($i > 0)?' AND ':'';
            $whereSql .= $pre.$key." = '".$value."'";
            $i++;
        }
    }
    $query = "DELETE FROM ".$table.$whereSql;
    $delete = $this->db->query($query);
    return $delete?true:false;
}
}

```

Operaciones CRUD (userAction.php)

Utilizaremos el fichero userAction.php para realizar las operaciones CRUD con PHP y MySQL (clase DB). El código se ejecutará basándose en dos condiciones: 

- Submit del formulario de edición e inserción: Para insertar o modificar registros de la base de datos

- Query String en la URL: Eliminaremos registros de la base de datos basándonos en el `action_type` y el `id` proporcionados en la query string de la URL

Después de manipular los datos, almacenaremos el estado utilizando sesiones de PHP y redireccionando al usuario a su respectiva página.



```

<?php
//start session
session_start();

//Load and initialize database class
require_once 'DB.class.php';
$db = new DB();

$tblName = 'users';

//set default redirect url
$redirectURL = 'index.php';

if(isset($_POST['userSubmit'])){
    if(!empty($_POST['name']) && !empty($_POST['email']) && !empty($_POST['phone'])){
        if(!empty($_POST['id'])){
            //update data
            $userData = array(
                'name' => $_POST['name'],
                'email' => $_POST['email'],
                'phone' => $_POST['phone']
            );
            $condition = array('id' => $_POST['id']);
            $update = $db->update($tblName, $userData, $condition);
            if($update){
                $sessData['status']['type'] = 'success';
                $sessData['status']['msg'] = 'User data has been updated successfully.';
            }else{
                $sessData['status']['type'] = 'error';
                $sessData['status']['msg'] = 'Some problem occurred, please try again.';

                //set redirect url
                $redirectURL = 'addEdit.php';
            }
        }else{
            //insert data
            $userData = array(
                'name' => $_POST['name'],
                'email' => $_POST['email'],
                'phone' => $_POST['phone']
            );
            $insert = $db->insert($tblName, $userData);
            if($insert){
                $sessData['status']['type'] = 'success';
                $sessData['status']['msg'] = 'User data has been added successfully.';
            }else{
                $sessData['status']['type'] = 'error';
                $sessData['status']['msg'] = 'Some problem occurred, please try again.';

                //set redirect url
                $redirectURL = 'addEdit.php';
            }
        }
    }else{
        $sessData['status']['type'] = 'error';
        $sessData['status']['msg'] = 'All fields are mandatory, please fill all the fields.';

        //set redirect url
    }
}

```

```

        $redirectURL = 'addEdit.php';
    }

    //store status into the session
    $_SESSION['sessData'] = $sessData;

    //redirect to the List page
    header("Location:".$redirectURL);
}elseif(($_REQUEST['action_type'] == 'delete') && !empty($_GET['id'])){
    //delete data
    $condition = array('id' => $_GET['id']);
    $delete = $db->delete($tblName, $condition);
    if($delete){
        $sessData['status']['type'] = 'success';
        $sessData['status']['msg'] = 'User data has been deleted successfully.';
    }else{
        $sessData['status']['type'] = 'error';
        $sessData['status']['msg'] = 'Some problem occurred, please try again.';
    }

    //store status into the session
    $_SESSION['sessData'] = $sessData;

    //redirect to the List page
    header("Location:".$redirectURL);
}
exit();
?>

```

Librería Bootstrap

Utilizaremos la librería Bootstrap para contar con una mejor interfaz. Si no quieres utilizar esta librería, puedes omitir este paso.

```
<link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css"
```

Leer y eliminar registros (index.php)

En el index.php obtendremos los registros de la tabla users mediante la clase DB y los listaremos en formato tabular con los enlaces para Añadir, Editar y Eliminar.

- El enlace de Añadir lo apuntaremos a la página addEdit.php para dar de alta nuevos registros
- El enlace de Editar lo apuntaremos a la página addEdit.php para editar registros
- El enlace de Eliminar lo apuntaremos al fichero userAction.php con sus respectivos parámetros action_type e id. En dicho fichero, eliminaremos el registro de la base de datos basándonos en el id.




```

<?php
//start session
session_start();

//get session data
$sessData = !empty($_SESSION['sessData'])?$_SESSION['sessData']:'';

//load and initialize database class
require_once 'DB.class.php';
$db = new DB();

//get users from database
$users = $db->getRows('users',array('order_by'=>'id DESC'));

//get status message from session
if(!empty($sessData['status']['msg'])){
    $statusMsg = $sessData['status']['msg'];
    $statusMsgType = $sessData['status']['type'];
    unset($_SESSION['sessData']['status']);
}
?>

<div class="container">
    <?php if(!empty($statusMsg) && ($statusMsgType == 'success')){ ?>
    <div class="alert alert-success"><?php echo $statusMsg; ?></div>
    <?php }elseif(!empty($statusMsg) && ($statusMsgType == 'error')){ ?>
    <div class="alert alert-danger"><?php echo $statusMsg; ?></div>
    <?php } ?>
    <div class="row">
        <div class="panel panel-default users-content">
            <div class="panel-heading">Users <a href="addEdit.php" class="glyphicon glyphicon-plus">
            <table class="table table-striped">
                <thead>
                    <tr>
                        <th></th>
                        <th>Name</th>
                        <th>Email</th>
                        <th>Phone</th>
                        <th>Action</th>
                    </tr>
                </thead>
                <tbody id="userData">
                    <?php if(!empty($users)): $count = 0; foreach($users as $user): $count++; ?>
                    <tr>
                        <td><?php echo '#'.$count; ?></td>
                        <td><?php echo $user['name']; ?></td>
                        <td><?php echo $user['email']; ?></td>
                        <td><?php echo $user['phone']; ?></td>
                        <td>
                            <a href="addEdit.php?id=<?php echo $user['id']; ?>" class="glyphicon g
                            <a href="userAction.php?action_type=delete&id=<?php echo $user['id'];
                        </td>
                    </tr>
                    <?php endforeach; else: ?>
                    <tr><td colspan="5">No user(s) found.....</td></tr>
                    <?php endif; ?>
                </tbody>
            </table>

```

```
        </div>
    </div>
</div>
```

Crear y editar registros (addEdit.php)

En el fichero addEdit.php, implementaremos la funcionalidad de crear y editar registros de la tabla users.

- La información del formulario se enviará al fichero userAction.php para insertar registros en la tabla users
- Si el parámetro ID existe en la URL, debemos recuperar la información de dicho usuario y rellenar el formulario con sus datos. El formulario se enviará al fichero userAction.php para modificar los datos de dicho registro.



```

<?php
//start session
session_start();

//get session data
$sessData = !empty($_SESSION['sessData'])?$_SESSION['sessData']:'';

//get user data
if(!empty($_GET['id'])){
    include 'DB.class.php';
    $db = new DB();
    $conditions['where'] = array(
        'id' => $_GET['id'],
    );
    $conditions['return_type'] = 'single';
    $userData = $db->getRows('users', $conditions);
}

$actionLabel = !empty($_GET['id'])? 'Edit': 'Add';

//get status message from session
if(!empty($sessData['status']['msg'])){
    $statusMsg = $sessData['status']['msg'];
    $statusMsgType = $sessData['status']['type'];
    unset($_SESSION['sessData']['status']);
}
?>

<div class="container">
    <?php if(!empty($statusMsg) && ($statusMsgType == 'success')){ ?>
    <div class="alert alert-success"><?php echo $statusMsg; ?></div>
    <?php }elseif(!empty($statusMsg) && ($statusMsgType == 'error')){ ?>
    <div class="alert alert-danger"><?php echo $statusMsg; ?></div>
    <?php } ?>
    <div class="row">
        <div class="panel panel-default users-content">
            <div class="panel-heading"><?php echo $actionLabel; ?> User <a href="index.php" class=
            <div class="panel-body">
                <form method="post" action="userAction.php" class="form">
                    <div class="form-group">
                        <label>Name</label>
                        <input type="text" class="form-control" name="name" value="<?php echo !emp
                    </div>
                    <div class="form-group">
                        <label>Email</label>
                        <input type="text" class="form-control" name="email" value="<?php echo !em
                    </div>
                    <div class="form-group">
                        <label>Phone</label>
                        <input type="text" class="form-control" name="phone" value="<?php echo !em
                    </div>
                    <input type="hidden" name="id" value="<?php echo $userData['id']; ?>">
                    <input type="submit" name="userSubmit" class="btn btn-success" value="SUBMIT" /
                </form>
            </div>
        </div>
    </div>
</div>
</div>

```

Fuente: [codexworld.com](https://www.codexworld.com/php-crud-operations-with-mysqli-extension/) (<https://www.codexworld.com/php-crud-operations-with-mysqli-extension/>)

COMPARTE ESTE ARTÍCULO

☐ ENVIAR A UN AMIGO

COMPARTIR EN GOOGLE +

ARTÍCULO ANTERIOR

[¿Cómo saber si tu teléfono móvil tiene un virus? \(/articulo/como_saber_si_tu_telefono_movil_tiene_un_virus_1903\)](#)

SIGUIENTE ARTÍCULO

[Subida de imágenes múltiple con AngularJS \(/articulo/subida_de_imagenes_multiple_con_angularjs_1905\)](#)

¡SÉ EL PRIMERO EN COMENTAR!

Conéctate (/login) o Regístrate (/registro) para dejar tu comentario.

Secciones

[Artículos \(/articulos\)](#)
[Tutoriales y código fuente \(/codigos\)](#)
[Foros \(/foros\)](#)
[Eventos \(/eventos\)](#)
[Empleo \(/empleo\)](#)

Lenguajes Destacados

[PHP \(/php\)](#)
[Java \(/java\)](#)
[ASP \(/asp\)](#)
[Bases de datos \(/bases-de-datos\)](#)
[C \(/c\)](#)

Información

[Datos Legales \(/datos_legales\)](#)
[Política de privacidad \(/politica_de_privacidad\)](#)
[Publicidad \(/publicidad\)](#)

Contacto

[Contacte con nosotros \(/contacto\)](#)
[Publicidad \(/publicidad\)](#)
<https://www.facebook.com/programacionencastellano> (<https://twitter.com/noprogram>)



Diseño web y desarrollo web (<https://colorvivo.com>). Un proyecto de los hermanos Carrero (<https://carrero.es>).

Alojado en cloud privado StackScale (<https://www.stackscale.es/>)

Más internet: Password (<https://password.es>) | Crear tu Favicon (<https://getfavicon.com>) | material para manualidades (<https://conideade.com>)

Copyright © 1998-2019 Programación en Castellano. Todos los derechos reservados

