

# Consulta

El proceso de consulta de datos de una tabla es similar al de listado, la diferencia es que se muestra sólo aquel o aquellos que cumplen la condición por la que buscamos.

Haremos un programa que nos permita consultar los datos de un alumno ingresando su mail para su búsqueda. Tengamos en cuenta que no puede haber dos alumnos con el mismo mail, por lo que la consulta nos puede arrojar uno o ningún registro.

Debemos codificar un formulario para el ingreso del mail a consultar:

pagina1.html

```
<html>

<head>
  <title>Problema</title>
</head>

<body>
  <form action="pagina2.php" method="post">
    Ingrese el mail del alumno a consultar:
    <input type="text" name="mail">
    <br>
    <input type="submit" value="buscar">
  </form>
</body>

</html>
```

Por otro lado tenemos el archivo "pagina2.php" que se encarga de buscar el mail ingresado en el formulario:

```
<html>

<head>
  <title>Problema</title>
</head>

<body>
  <?php
    $conexion = mysqli_connect("localhost", "root", "", "base1") or
      die("Problemas con la conexión");

    $registros = mysqli_query($conexion, "select codigo,nombre,codigocurso
      from alumnos where mail='$_REQUEST[mail]'" ) or
      die("Problemas en el select:" . mysqli_error($conexion));
```

```

if ($reg = mysqli_fetch_array($registros)) {
    echo "Nombre:" . $reg['nombre'] . "<br>";
    echo "Curso:";
    switch ($reg['codigocurso']) {
        case 1:
            echo "PHP";
            break;
        case 2:
            echo "ASP";
            break;
        case 3:
            echo "JSP";
            break;
    }
} else {
    echo "No existe un alumno con ese mail.";
}
mysqli_close($conexion);
?>
</body>

</html>

```

Lo más importante se encuentra en el comando select:

```

$registros = mysqli_query($conexion, "select codigo,nombre,codigocurso
    from alumnos where mail='$_REQUEST[mail]'" ) or
    die("Problemas en el select:" . mysqli_error($conexion));

```

Acá es donde con la clausula where seleccionamos sólo el registro que cumple con la condición que el mail sea igual al que ingresamos.

Como sólo puede haber un registro que cumpla la condición, llamamos a la función `mysqli_fetch_array` en un `if` y no una estructura repetitiva como el listado:

```

if ($reg = mysqli_fetch_array($registros)) {

```

En caso de retornar un vector asociativo la condición del `if` se verifica como verdadera y pasa a mostrar los datos, en caso de retornar `false` se ejecuta el `else`.

# Delete

El objetivo de este punto es el borrado de un registro de una tabla. Para ello, implementaremos un algoritmo que solicite ingresar el mail de un alumno y posteriormente efectúe su borrado.

Para eliminar filas en una tabla debemos utilizar el comando SQL delete.

La primera página es idéntica a la consulta, ya que debemos implementar un formulario que solicite la carga del mail del alumno:

pagina1.html

```
<html>

<head>
  <title>Problema</title>
</head>

<body>
  <form action="pagina2.php" method="post">
    Ingrese el mail del alumno a borrar:
    <input type="text" name="mail">
    <br>
    <input type="submit" value="buscar y borrar">
  </form>
</body>

</html>
```

Por otro lado tenemos el archivo "pagina2.php" que se encarga de buscar el mail ingresado en el formulario y en caso que exista se procede a borrarlo:

```
<html>

<head>
  <title>Problema</title>
</head>

<body>
  <?php
    $conexion = mysqli_connect("localhost", "root", "", "base1") or
      die("Problemas con la conexión");

    $registros = mysqli_query($conexion, "select codigo from alumnos
      where mail='$_REQUEST[mail]'" ) or
      die("Problemas en el select:" . mysqli_error($conexion));
    if ($reg = mysqli_fetch_array($registros)) {
```

```

mysql_query($conexion, "delete from alumnos where mail='$_REQUEST[mail]') or
die("Problemas en el select:" . mysql_error($conexion));
echo "Se efectuó el borrado del alumno con dicho mail.";
} else {
    echo "No existe un alumno con ese mail.";
}
mysql_close($conexion);
?>
</body>

</html>

```

En esta segunda página efectuamos dos llamadas a la función `mysql_query`, una para consultar si existe el mail ingresado y otra para efectuar el borrado del registro respectivo. Si no existe el mail ingresado mostramos un mensaje de tal situación.

## Update

De las actividades con una tabla esta es la más larga. Vamos a resolverlo implementando tres páginas, la primera un formulario de consulta del mail de un alumno, la segunda otro formulario que nos permita cargar su mail modificado y la última registrará el cambio en la tabla.

El formulario de consulta del mail del alumno es similar a problemas anteriores:

```

<html>

<head>
    <title>Problema</title>
</head>

<body>
    <form action="pagina2.php" method="post">
        Ingrese el mail del alumno:
        <input type="text" name="mail"><br>
        <input type="submit" value="buscar">
    </form>
</body>

</html>

```

La segunda página es la más interesante y con conceptos nuevos:

pagina2.php

```

<html>

<head>

```

```

<title>Problema</title>
</head>

<body>

<?php

$conexion = mysqli_connect("localhost", "root", "", "base1") or
die("Problemas con la conexión");

$registros = mysqli_query($conexion, "select * from alumnos
                                where mail=$_REQUEST[mail]") or
die("Problemas en el select:" . mysqli_error($conexion));
if ($reg = mysqli_fetch_array($registros)) {
    ?>

    <form action="pagina3.php" method="post">
    Ingrese nuevo mail:
    <input type="text" name="mailnuevo" value="<?php echo $reg['mail'] ?>">
    <br>
    <input type="hidden" name="mailviejo" value="<?php echo $reg['mail'] ?>">
    <input type="submit" value="Modificar">
    </form>

    <?php
    } else
    echo "No existe alumno con dicho mail";
    ?>
</body>

</html>

```

Lo primero que podemos observar es que si el if se verifica verdadero se ejecuta un bloque que contiene código HTML:

```

if ($reg = mysqli_fetch_array($registros)) {
    ?>

    <form action="pagina3.php" method="post">
    Ingrese nuevo mail:
    <input type="text" name="mailnuevo" value="<?php echo $reg['mail'] ?>">
    <br>
    <input type="hidden" name="mailviejo" value="<?php echo $reg['mail'] ?>">
    <input type="submit" value="Modificar">
    </form>

    <?php

```

Es decir que podemos disponer bloques de PHP dispersos dentro de la página.

Otro concepto importante es como enviar el mail del primer formulario a la tercer página, esto se logra con los controles de tipo "hidden", este tipo de control no se visualiza en el formulario pero se envía al presionar el botón submit.

Si queremos que el control text se cargue con el mail ingresado en el formulario anterior debemos cargar la propiedad value con dicho valor:

```
<input type="hidden" name="mailviejo" value="<?php echo $reg['mail'] ?>">
```

Por último la pagina3.php es la que efectúa la modificación de la tabla propiamente dicha. Con el mail ingresado en la pagina1.php, el mail modificado en la pagina2.php se efectúa el update.

```
<html>
```

```
<head>
```

```
<title>Problema</title>
```

```
</head>
```

```
<body>
```

```
<?php
```

```
$conexion = mysqli_connect("localhost", "root", "", "base1") or  
die("Problemas con la conexión");
```

```
mysqli_query($conexion, "update alumnos  
    set mail='$_REQUEST[mailnuevo]'  
    where mail='$_REQUEST[mailviejo]'" ) or  
die("Problemas en el select:" . mysqli_error($conexion));  
echo "El mail fue modificado con exito";  
?>
```

```
</body>
```

```
</html>
```

Tengamos en cuenta que el segundo formulario nos envía dos datos:

\$\_REQUEST[mailnuevo] y \$\_REQUEST[mailviejo].

Si no llamamos a:

```
mysqli_close($conexion);
```

Cuando finaliza la ejecución del script de PHP se libera automáticamente la conexión con MySQL.