



CGT 356

Lecture 11

# [ PHP SELECT ]

```
...
$sql      = "SELECT LastName, FirstName FROM Employee";
$result   = mysql_query($sql);

//check to see if the recordset is empty - notice no { or } below:
if(empty($result))
    $num_results = 0;
else
    $num_results = mysql_num_rows($result);
?>
<table border="0" width="50%">
    <?php
    for($i=0; $i < $num_results; $i++)
    {
        $row = mysql_fetch_array($result);
        ?>
        <tr>
            <td> <?php echo $row["LastName"]; ?> </td>
            <td> <?php echo $row["FirstName"]; ?> </td>
        </tr>
        <?php
    } //end for
    ?>
</table>
...
```

# [ SQL INSERT ]

- Syntax

- INSERT INTO <tablename>  
VALUES('v1', 'v2', ..., 'vx');

*or*

- INSERT INTO <tablename>(col1, col2, col3)  
VALUES('v1', 'v2', 'v3');

# [ SQL INSERT Example 1 ]

- INSERT INTO Shippers  
VALUES(25, 'ABC Corp.', '555-2345');
- Notice 25 is not in quotes
  - When the database field is created as a char or a varchar, use single quotes to insert
  - When the database field is created as a decimal (or other number) do NOT use single quotes when inserting or comparing to that field.

# [ SQL INSERT Example 2 ]

- `INSERT INTO Shippers(ShipperID, CompanyName) VALUES(26, 'DEF Corp.');`
- Notice that ShipperID & CompanyName are not in quotes. DB field names do not appear in quotes.
- Notice that each value **MUST** line up with the corresponding field name.
  - i.e.: 26 maps to ShipperID

# [ PHP INSERT ]

```
$shipperID    = $_HTTP_POST_VARS["shipperID"];  
$companyName = $_HTTP_POST_VARS["companyName"];  
$phone       = $_HTTP_POST_VARS["phone"];
```

```
$sql = "INSERT INTO Shippers(ShipperID, CompanyName, Phone)";  
$sql = $sql." VALUES(".$shipperID.", ";  
$sql = $sql."'" . $companyName . "', '" . $phone . "' )";
```

```
//Uncomment this debug echo statement to check your SQL  
//echo $sql;
```

```
//Open DB connection not shown here..
```

```
$result = mysql_query($sql);
```

# [ INSERT Notes ]

- Typically want to use:
  - INSERT INTO  
Shippers(shipperID, companyName, phone)  
VALUES(25, 'ABC Corp.', '555-2345');
  - This verifies that:
    - You insert the correct value into the correct field
    - You have the same number of values as the number of fields in the table

# [ SQL DELETE ]

- Syntax

- DELETE FROM <tablename>  
WHERE <condition>;

- Example

- DELETE FROM Employee  
WHERE EmployeeID = '125775';
  - Deletes the employee having EmployeeID 125775 from the database.

# [ DELETE Notes ]

- Be careful with the DELETE statement!!
- Almost always want to delete by a Primary Key – Why?
  - If you delete by another field, you could delete a large portion of the database!
- DELETE FROM Employee;
  - This statement deletes the **entire** Employee table!

# [ PHP DELETE ]

```
$empID = $_HTTP_POST_VARS["employeeID"];
```

```
$sql = "DELETE FROM Employee WHERE  
EmployeeID='".$empID."'";
```

*//(The above two lines should be only 1 line in your code, I had to wrap)*

```
//Uncomment this debug echo statement to check your SQL  
//echo $sql;
```

```
//Open DB connection not shown here..
```

```
$result = mysql_query($sql);
```

# [ SQL UPDATE ]

- Syntax

- UPDATE <tablename>  
SET <columnname> = <value>

- Example

- UPDATE Employee  
SET HomePhone='555-1234'  
WHERE EmployeeID='125775';

# [ SQL UPDATE Example 1 ]

- UPDATE Employee SET HomePhone='555-1234';
- What is wrong with the above statement?
  - It updates everybody's home phone number!

# [ UPDATE Notes ]

---

- Be careful forming your WHERE clause
  - Proper use of the WHERE clause allows you to update or delete only the rows you really intend to.
  - You will almost ALWAYS need a WHERE clause.

# [ SQL UPDATE Example 2 ]

- An example of setting multiple fields all at one time (separate them by commas):
- UPDATE Employee  
SET HomePhone='555-1234',  
Address='123 N. South St.',  
City='West Lafayette', State='IN'  
WHERE EmployeeID='125775';

# [ SQL UPDATE Example 3 ]

- An example using BETWEEN and AND:
- ```
UPDATE Employee
SET     Region='Europe'
WHERE  City='London'
AND    HireDate BETWEEN '1992-04-01'
AND    '1993-10-17' ;
```
- \* *This is the date syntax for **MySQL***
- \* *The dates used above are pulled from the Northwind database*

# [ PHP UPDATE ]

```
$homePhone = $_HTTP_POST_VARS[ "homePhone" ];  
$address   = $_HTTP_POST_VARS[ "address" ];  
$city      = $_HTTP_POST_VARS[ "city" ];  
$state     = $_HTTP_POST_VARS[ "state" ];  
$empID     = $_HTTP_POST_VARS[ "empID" ];
```

# [ PHP UPDATE (cont.) ]

```
$sql = "UPDATE Employee SET HomePhone=`";  
$sql = $sql.$homePhone.`', Address1=`";  
$sql = $sql.$address.`', City=`";  
$sql = $sql.$city.`', State=`";  
$sql = $sql.$state`' `";  
$sql = $sql."WHERE EmployeeID=`";  
$sql = $sql.$empID.`' "`;
```

```
//Uncomment this debug echo statement to check your SQL  
//echo $sql;
```

```
//Open DB connection not shown here...
```

```
$result = mysql_query($sql);
```