

Connect to MySQL

```
<?php // connDB.php

function db_connect ()
{
    $dbh = new
        PDO("mysql:host=64.59.233.246;
            dbname=DBNAME", "USER", "PASSWORD");

    $dbh->setAttribute (PDO::ATTR_ERRMODE,
        PDO::ERRMODE_EXCEPTION);

    return ($dbh);
}
function db_close (& $dbh)
{
    $dbh = null;
}
?>
```

Good Coding

- We want to separate the data access from the presentation as much as we can
 - query files
 - presentation files
 - all are .php files
- Query files: write data access functions.
 - many presentations files can access the same query
 - may have many functions per file
- skeleton.php is an example of a presentation file
 - lots of HTML and PHP function calls to get/present data

Presentation file

skeleton.php

```
<?php
    require_once 'connDB.php';
    session_start();

    $dbh = db_connect();

?>

<html>
    <head>
        <title></title>
    </head>
    <body>
        MIX OF PHP AND HTML
    </body>

</html>

<?php
    db_close($dbh);
```

Rather than **print** every line of HTML, you can inline HTML outside of the `<?php ?>` tags and it is automatically printed

php functions

```
<?php // print.php
```

```
function printData ($data1, $data2)
{
    $lString = $data1 . " " . $data2;
    print $lString;
    return $lString;
}
```

```
?>
```

```
<?php //testPrint.php
```

```
require_once 'print.php';
$result = printData("hello", "World");
print $result;
```

```
?>
```

This code could be in the
<body> of the skeleton.php!

You might collect all the
includes at the top.

php functions

```
<?php // passByReference.php
```

```
function printDataRef (&$data1, &$data2)
{
    $lString = $data1 . " " . $data2;

    print $lString;
    return $lString;
}
?>
```

```
<?php //globalVariables.php
```

```
$gValue = 1;
function printDataGlobal($data)
{
    global $gValue; // this attaches the name
                   // to the global variable.
    print $gValue . ' ' . $data;
}
?>
```

```
?>
```

Query Syntax

```
// already opened the database
$sth = $dbh->prepare(
    "SELECT FName, LName FROM Client");

// run the query
$sth->execute();

printf("Results Count: %d\n", $sth->columnCount());
while( $row = $sth->fetch() )
{
    printf("FName: %s LName %s\n",
        $row[0], $row["LName"]);
}
```

Error Handling

```
$sth = $dbh->prepare( .... );
```

```
try
```

```
{
```

```
    $sth ->execute( );
```

```
}
```

```
catch (PDOException $e)
```

```
{
```

```
    printf ("The statement failed.\n");
```

```
    printf ("getCode: ". $e->getCode () . "\n");
```

```
    printf ("getMessage: ". $e->getMessage () . "\n");
```

```
}
```

queryFunction.php

```
function getAllPeopleNames($dbh)
{
    $rows = array();

    $sth = $dbh->prepare(
        "SELECT FName, LName FROM Client");
    // run the query
    $sth->execute();

    printf("Results Count: %d\n", $sth->columnCount());
    while( $row = $sth->fetch() )
    {
        $rows[] = $row;
    }
    return $rows;
}
```


queryFunctionCall.php

```
<?php
```

```
include 'connDB.php';  
include 'queryFunction.php';
```

```
$dbh = db_connect ();  
$data = getAllPeopleNames($dbh);
```

```
foreach ( $data as $row )  
{  
    print $row['FName'] . ' ' . $row['LName']  
        . ' <br/> ';  
}
```

```
db_close($dbh);
```

```
?>
```

queryFunctionParams.php

```
function getAllPeopleNamesWhereLName  
    ($dbh, $LName)
```

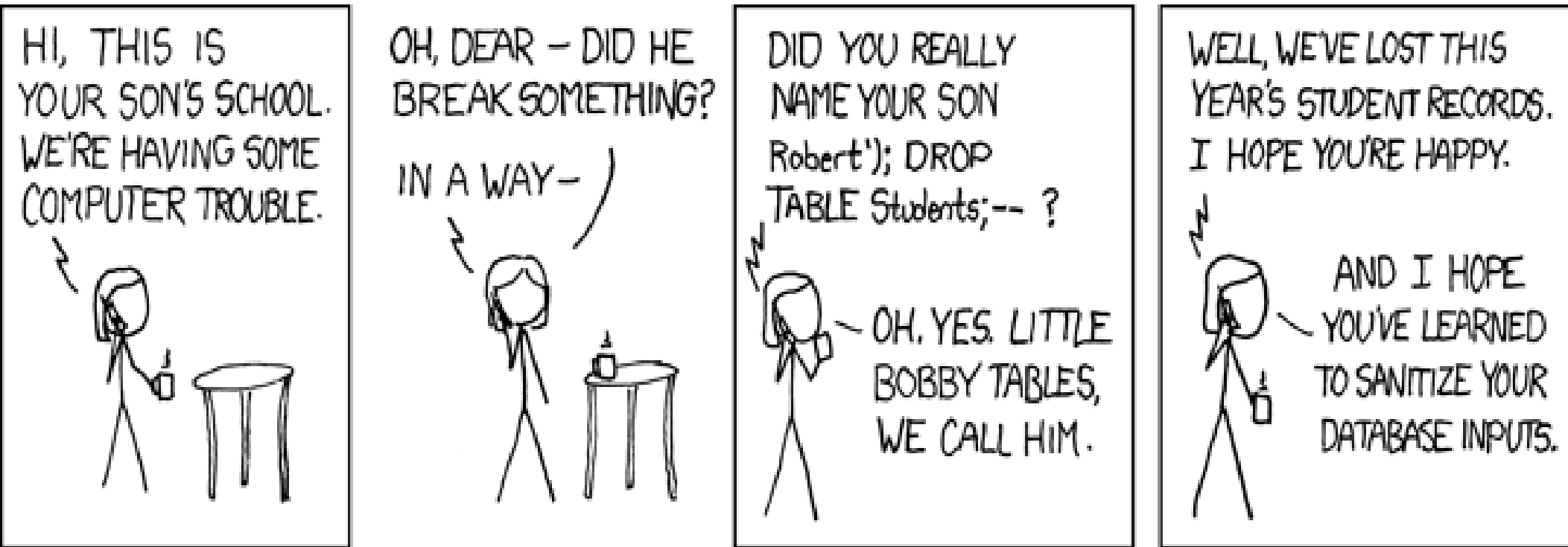
```
{  
    $rows = array();
```

```
    $sth = $dbh->prepare(  
        "SELECT FName, LName FROM Client  
        WHERE LName like :name");
```

```
    $sth->bindValue(":name", $LName);  
    // run the query  
    $sth->execute();
```

```
    // same as getAllPeopleNames.....
```

Why we use prepared statements?



```
SELECT username FROM users WHERE username = '$userId';
```

```
$userId = ' bob' ); Drop Table Students; --"
```

```
mysql_real_escape_string(): $userId = ' bob\\' );  
Drop Table Students; --"
```

SQL Injection

queryFunctionCallParams.php

```
<?php
```

```
include 'connDB.php';
```

```
include 'queryFunctionParams.php';
```

```
$dbh = db_connect();
```

```
$data = getAllPeopleNamesWhereLName($dbh, "Greene");
```

```
foreach ( $data as $row )
```

```
{
```

```
    print $row['FName'] . ' ' . $row['LName'] . '<br/>';
```

```
}
```

```
db_close($dbh);
```

```
?>
```

runQueryTable.php

Software	FName	LName	Email	Salary
Stellar Teller	Aline	Maddox	elementum.purus.accumsan@parturient.edu	153308
Word Precise	Quintessa	Frederick	et@Curae;Donectincidunt.ca	167687
Anodize	Carissa	Ford	Quisque@elitpellentesquea.ca	23308
ATM Buddy	Odette	Espinoza	non.sollicitudin@variusorciin.org	153903
Where Am I? GPS App	Ursula	Stewart	condimentum@a.edu	49855
Speller	Tyrone	Wong	lacus.Quisque@DonecnibhQuisque.edu	31763
SpellerLite	Wyatt	Figueroa	ullamcorper@montesnascetur.ca	73617
StoryTeller	Michael	Atkinson	laoret.lectus@necorciDonec.com	63376
Stone Tablet	Oscar	Cox	Vivamus.rhoncus@Suspendissealiquet.org	61079
Vi	Quail	Crawford	convallis.in.cursus@orci.ca	77551
mauris id	Martin	Mccarthy	ipsum@nec.edu	178511
Vivamus nibh	Palmer	Albert	euismod.in@perconubia.edu	46726

Can you build this web page? This is software, manager, manager's email, and manager's salary.

Exercises

- Build a web page that displays the FName, LName, of each employee and the FName, LName of that employee's Manager.
- Build a web page the displays the total salary earned by all the employees who work on each software product (One row per software product). Display \$0.00 if a product has no one working on it.