



INTRO TO PHP & MYSQL

TJ Nicolaides
Original Curriculum: Izzy Johnston

PART 2:WHAT IS MySQL?

Intro to PHP & MySQL



RELATIONAL DATABASES

- Lots of information
- Content that is always related
- Content that is sometimes related

WHY NOT ONE BIG TABLE?

- Thoroughness
- Consistency
- Human Error
- Dynamic Content
- Performance

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with tabs: Browse, Structure, SQL, Search, Insert, Export, Import, Operations, and Triggers. Below the toolbar, a message bar indicates: "Showing rows 491160 - 491189 (892,917 total, Query took 0.0269 sec)". A red arrow points from this message bar down to the SQL query editor. The SQL query is:

```
SELECT *  
FROM `sensor_data_1sec`  
LIMIT 491160 , 30
```

Below the SQL editor, there are navigation controls: << < 16373 > >> Show : Start row: 491190 Number of rows: 30 Headers every 100 rows. To the right of these controls is a "Sort by key" dropdown set to "None". The main area displays a table with 30 rows of sensor data. The columns are: sensor_id, timestamp, temp, wind_speed, wind_direction, dew_point, humidity, air_density, pressure, and last_update. Each row includes edit, copy, and delete icons.

	sensor_id	timestamp	temp	wind_speed	wind_direction	dew_point	humidity	air_density	pressure	last_update
1	105	1376689792000	300.70	0.00	31	280.91	28.75	1.1598	100110	2013-08-19 10:59:19
2	105	1376689793000	300.73	0.00	31	280.93	28.73	1.1595	100096	2013-08-19 10:59:19
3	105	1376689794000	300.75	0.00	31	280.93	28.70	1.1596	100110	2013-08-19 10:59:19
4	105	1376689795000	300.75	0.00	31	280.93	28.70	1.1594	100095	2013-08-19 10:59:19
5	105	1376689796000	300.75	0.00	31	280.93	28.70	1.1595	100099	2013-08-19 10:59:19
6	105	1376689797000	300.70	0.00	31	280.89	28.70	1.1596	100089	2013-08-19 10:59:19
7	105	1376689798000	300.80	0.00	31	280.97	28.70	1.1592	100093	2013-08-19 10:59:19
8	105	1376689799000	300.73	0.00	31	280.91	28.70	1.1596	100100	2013-08-19 10:59:19
9	105	1376689800000	300.75	0.00	31	280.93	28.70	1.1594	100095	2013-08-19 10:59:19
10	105	1376689801000	300.70	0.00	31	280.89	28.70	1.1596	100093	2013-08-19 10:59:19
11	105	1376689802000	300.75	0.00	31	280.93	28.70	1.1595	100102	2013-08-19 10:59:19
12	105	1376689803000	300.80	0.00	31	280.97	28.70	1.1592	100089	2013-08-19 10:59:19
13	105	1376689804000	300.75	0.00	31	280.93	28.70	1.1594	100093	2013-08-19 10:59:19
14	105	1376689805000	300.70	0.00	31	280.89	28.70	1.1597	100103	2013-08-19 10:59:19
15	105	1376689806000	300.70	0.00	31	280.89	28.70	1.1598	100110	2013-08-19 10:59:19
16	105	1376689807000	300.70	0.00	31	280.89	28.70	1.1597	100104	2013-08-19 10:59:19
17	105	1376689808000	300.75	0.00	31	280.95	28.75	1.1594	100091	2013-08-19 10:59:19
18	105	1376689809000	300.70	0.00	31	280.94	28.80	1.1599	100115	2013-08-19 10:59:19
19	105	1376689810000	300.70	0.00	31	280.94	28.80	1.1598	100106	2013-08-19 10:59:19
20	105	1376689811000	300.70	0.00	31	280.94	28.80	1.1596	100096	2013-08-19 10:59:19
21	105	1376689812000	300.70	0.00	31	280.94	28.80	1.1596	100095	2013-08-19 10:59:19

WHY NOT ONE BIG TABLE?

Date	Book Title	Author First Name	Author Last Name	Patron First Name	Patron Last Name
January 29, 2011	<i>Green Eggs and Ham</i>	Dr.	Seuss	Sally	McBride
January 31, 2011	<i>Stuart Little</i>	E.B.	White	Mario	Batali
February 6, 2011	<i>Winnie the Pooh</i>	A.A.	Milne	Igor	Salbinski
February 8, 2011	<i>The Tale of Peter Rabbit</i>	Beatrix	Potter	Maria	Hernandez
February 9, 2011	<i>Many Moons</i>	James	Thurber	Izzy	Johnston

WHY NOT ONE BIG TABLE?

Authors

Key	Author First Name	Author Last Name
001	Beatrix	Potter
002	James	Thurber
003	E.B.	White
004	A.A.	Milne
005	Dr.	Seuss

Books

Key	Book Title	Author Key
00001	<i>Green Eggs and Ham</i>	005
00002	<i>Many Moons</i>	002
00003	<i>Stuart Little</i>	003
00004	<i>Tale of Peter Rabbit, The</i>	001
00005	<i>Winnie the Pooh</i>	004

Patrons

Key	Patron First Name	Patron Last Name
01	Sally	McBride
02	Izzy	Johnston
03	Igor	Salbinski
04	Mario	Batali
05	Maria	Hernandez

Checkouts

Key	Date	Book Key	Patron Key
1	January 29, 2011	00001	01
2	January 31, 2011	00003	04
3	February 6, 2011	00005	03
4	February 8, 2011	00004	05
5	February 9, 2011	00002	02

WHY NOT ONE BIG TABLE?

Authors

Key	Author First Name	Author Last Name
001	Beatrix	Potter
002	James	Thurber
003	E.B.	White
004	A.A.	Milne
005	Dr.	Seuss

Books

Key	Book Title	Author Key
00001	<i>Green Eggs and Ham</i>	005
00002	<i>Many Moons</i>	002
00003	<i>Stuart Little</i>	003
00004	<i>Tale of Peter Rabbit, The</i>	001
00005	<i>Winnie the Pooh</i>	004

Patrons

Key	Patron First Name	Patron Last Name
01	Sally	McBride
02	Izzy	Johnston
03	Igor	Salbinski
04	Mario	Batali
05	Maria	Hernandez

Checkouts

Key	Date	Book Key	Patron Key
1	January 29, 2011	00001	01
2	January 31, 2011	00003	04
3	February 6, 2011	00005	03
4	February 8, 2011	00004	05
5	February 9, 2011	00002	02

WHO USES RELATIONAL DB'S?

- Wordpress
- Drupal
- Online retailers
- Libraries
- Anyone with lots of content



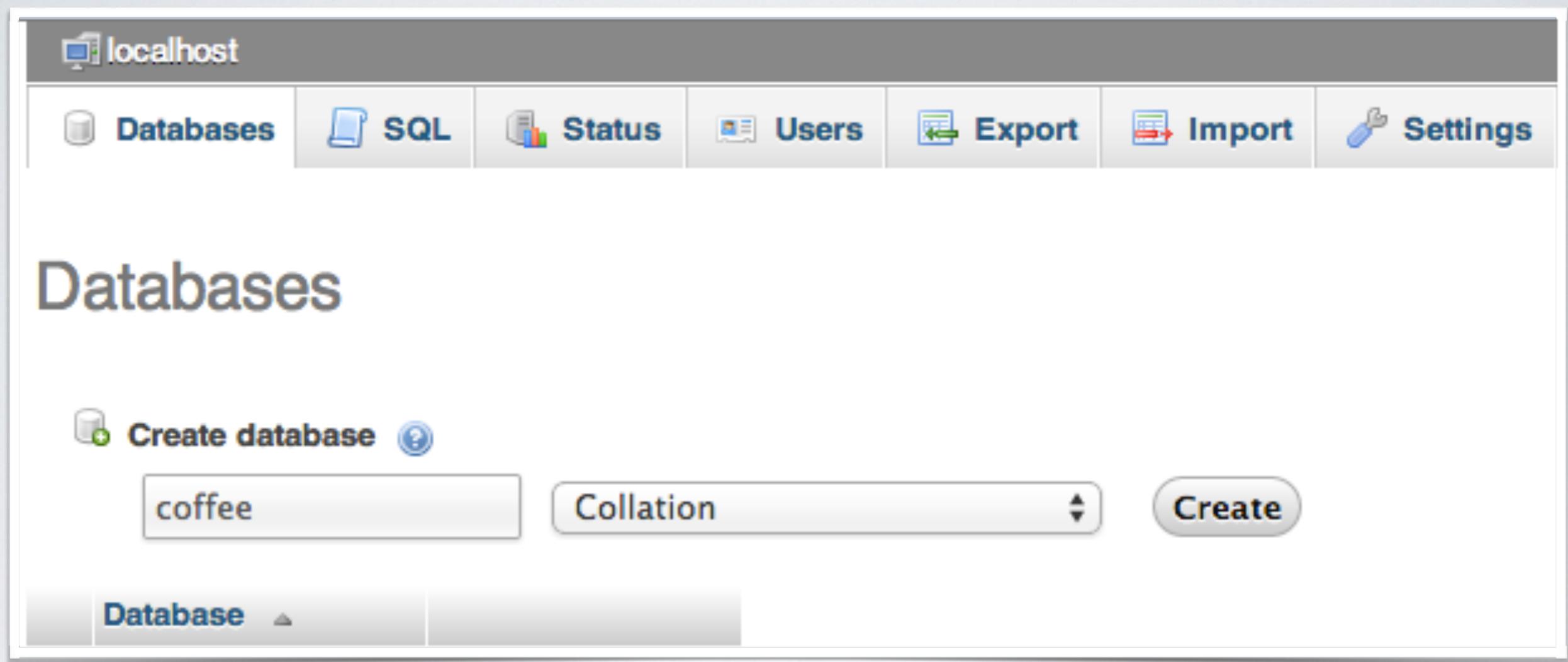
WHY MYSQL?

- MySQL is a popular relational database system
- Provides structure for your data
- Create multiple tables
- “Query” driven
 - Question or request to a database
 - Select a set of data
 - Delete an entry
 - Etc.

CONTENT TYPES

- **VARCHAR** - up to 256 characters
- **INT** - integer up to 11 characters
- **FLOAT** - decimal of any size
- **TEXT** - large amounts of text, no size limit
- **DATE** - date entries, specific format

CREATING A MYSQL DB



CREATING A MYSQL TABLE

The screenshot shows the phpMyAdmin interface for managing a MySQL database named 'coffee'. The left sidebar displays the 'Recent tables' dropdown, which includes 'coffee'. Below it, a message states 'No tables found in database.' A prominent green button labeled 'Create table' is visible. The main workspace is titled 'localhost > coffee' and features a navigation bar with tabs: Structure, SQL, Search, Query, Export, Import, and Operations. The 'Structure' tab is selected. A modal window titled 'Create table' is open, prompting for the table name and number of columns. The table name is set to 'product', and the number of columns is indicated by an empty input field.

phpMyAdmin

localhost > coffee

No tables found in database

Create table

Name: product

Number of columns:

CREATING MYSQL COLUMNS

phpMyAdmin

Table name: product Add 1 column(s) Go

Structure

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A_I	Comments
product_id	INT		None						
company	VARCHAR	256	None						
type	VARCHAR	256	None						
roast	VARCHAR	256	None						
description	TEXT		None						

Table comments:

Storage Engine: InnoDB

Collation:

Save Cancel

(Recent tables) ... coffee No tables found in database. Create table

The screenshot shows the 'Structure' tab of the phpMyAdmin interface for a table named 'product'. The table contains five columns: 'product_id' (INT), 'company' (VARCHAR 256), 'type' (VARCHAR 256), 'roast' (VARCHAR 256), and 'description' (TEXT). The 'Storage Engine' is set to InnoDB and the 'Collation' is not specified. Buttons for 'Save' and 'Cancel' are at the bottom right.

READY TO QUERY

phpMyAdmin

localhost » coffee » product

Browse Structure SQL Search Insert Export Import Operations Triggers

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0003 sec)

```
SELECT *  
FROM `product`  
LIMIT 0 , 30
```

Profiling [Inline] [Edit] [Explain SQL] [Create PHP C]

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	product_id	int(11)			No	None		Change Drop Primary Unique Index Spatial Fulltext
2	company	varchar(256)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext
3	type	varchar(256)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext
4	roast	varchar(256)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext
5	description	text	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext

Check All / Uncheck All With selected: Browse Change Drop Primary Unique Index

Print view Relation view Propose table structure

Add 1 column(s) At End of Table At Beginning of Table After product_id Go

Indexes

PHP + MYSQL = PB & J

- MySQL is a database
- PHP is a language
- MySQL stores the data
- PHP interacts with it
- **Both delicious**



CONNECT PHP TO A DATABASE

- Establish a link with the MySQL server location (here, and usually, localhost) and user + pass credentials
- Set the character encoding
- Choose a database to query

```
1 <?php
2     /*Connect to a server*/
3     $link=mysqli_connect ('localhost', 'root', 'password');
4     /*If you are using a PC, this should be 'Windows XP'*/
5     if (!$link){
6         $output='Unable to connect to the database';
7         echo $output;
8         exit();
9     }
10    /*Check for proper encoding*/
11    if (!mysqli_set_charset($link, 'utf8'))
12        $output = 'Unable to set database encoding';
13        echo $output;
14        exit();
15    }
16    /*Connect to database, named coffee*/
17    if (!mysqli_select_db($link, 'coffee'))
18        $output = 'Unable to locate the database';
19        echo $output;
20        exit();
21    }
22    //echo "You did ok!";
23 ?>
24
```

C. R. U. D.

Create / **R**ead / **U**pdate / **D**elete

CREATE

- HTML form to collect values:
product_insert.html
- Action attribute:
product_insert_result.php
- Method attribute: GET
- Let's collect information about coffee with text inputs, textarea comments, and radio buttons

```
21  
22  
23  
24  
25  
26  
27  
28 <form action="product_insert_result.php" method="g  
29   Company: <input type="text" name="company"/><b  
30   Type: <input type="text" name="type"/><br/>  
31   Roast:  
32     <input type="radio" name="roast" value="light"/>  
33     <input type="radio" name="roast" value="medium"/>  
34     <input type="radio" name="roast" value="dark"/>  
35   <br/>  
36   <textarea name="description" rows="10" cols="40">  
37   <input type="submit" value="Submit"/>  
38 </form>  
39  
40  
41  
42  
43  
44
```

CREATE

- PHP script to receive input from `product_insert.html`:
[product_insert_result.php](#)
- First, include the script we use to connect to the db ([include/db.inc.php](#))
- Use `$_GET` to capture the values passed from the form as URL parameters

```
3  /*Include file to connect to server and database*/
4  include 'include/db.inc.php';
5
6
7
8  /*Get data from form, clean it up for the MySQL database*/
9  $company = mysqli_real_escape_string($link, $_GET['company']);
10 $type = mysqli_real_escape_string($link, $_GET['type']);
11 $roast = mysqli_real_escape_string($link, $_GET['roast']);
12 $description = mysqli_real_escape_string($link, $_GET['description']);
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
```

CREATE

- Next: let's concatenate our MySQL query with these variables
- Basic insert statement:
 - **INSERT INTO table_name SET column_1 = ‘value_1’, column_2 = ‘value_2’**

```
3  /*Include file to connect to server and database*/
4  include 'include/db.inc.php';
5
6
7
8  /*Get data from form, clean it up for the MySQL database, */
9  $company = mysqli_real_escape_string($link, $_GET['company']);
10 $type = mysqli_real_escape_string($link, $_GET['type']);
11 $roast = mysqli_real_escape_string($link, $_GET['roast']);
12 $description = mysqli_real_escape_string($link, $_GET['description']);
13
14
15 /*Include file to connect to server and database*/
16 $sql = "INSERT INTO product SET
17   company='$company',
18   type='$type',
19   roast='$roast',
20   description='$description'";
21
22
23
24
25
26
27
28
29
30
31
```

CREATE

- If there was a problem with the insert statement, let's blow an error here
- Otherwise, let's redirect to another page where we'll **READ** data from our db now

```
3  /*Include file to connect to server and database*/
4  include 'include/db.inc.php';
5
6
7  /*Get data from form, clean it up for the MySQL database*/
8  $company = mysqli_real_escape_string($link, $_GET['company']);
9  $type = mysqli_real_escape_string($link, $_GET['type']);
10 $roast = mysqli_real_escape_string($link, $_GET['roast']);
11 $description = mysqli_real_escape_string($link, $_GET['description']);
12
13 /*Include file to connect to server and database*/
14 $sql = "INSERT INTO product SET
15   company='$company',
16   type='$type',
17   roast='$roast',
18   description='$description'";
19
20 /*Check for errors*/
21 if (!mysqli_query($link, $sql)){
22   $error = 'Error adding submitted data: ' . mysqli_error();
23   echo $error;
24   exit();
25 }
26
27 /*Jump to product_show.php, instead of hanging out here*/
28 header('Location:product_show.php');
29
```

CREATE

- Starting template on Github:
- Finished examples, for reference:
product_insert.html, product_insert_result.php

READ

- Our script to read from our coffee database is product_show.php
- First, include our connection script again (include/db.inc.php)
- Concatenate a SELECT query to fetch rows of data:
 - **SELECT column_1, column_2 FROM tablename ORDER BY column_2 DESC**

```
25  
26  
27  
28  
29  
30  
31 /*Include file to connect to server and  
32 include 'include/db.inc.php';  
33  
34 /*SQL query as variable select given da  
35 $sql='SELECT product_id, company, type,  
36     roast, description  FROM product OR  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47
```

READ

- Next, use a while loop to go through the list of results we get back from our query
- Set some variables to the column values (here they are represented as arrays)
- Handle any special characters with

htmlspecialchars()

```
46
47
48
49
50
51 /*While there are products in our data
52     get and display the company, the
53 $recording=mysqli_fetch_array($re
54
55 $product_id = htmlspecialchars($re
56 $company = htmlspecialchars($recor
57 $type = htmlspecialchars($recordin
58 $roast = htmlspecialchars($recordi
59 $description = htmlspecialchars($r
60 }
61 }
62 }
63 }
64 }
65 }
```

READ

- Echo your variables out with **echo**

```
47
48
49
50
51 /*While there are products in our database, get and
52   display the company, the type, the roast and description
53 while($recording=mysqli_fetch_array($result)){
54   $product_id = htmlspecialchars($recording['product_id'])
55   $company = htmlspecialchars($recording['company'], ENT_QUOTES)
56   $type = htmlspecialchars($recording['type'], ENT_QUOTES)
57   $roast = htmlspecialchars($recording['roast'], ENT_QUOTES)
58   $description = htmlspecialchars($recording['description'])

59   echo $company;
60   echo $type;
61   echo $roast;
62   echo $description;
63 }
64
65
66
67
68
69 |
70
71
```

READ

- Now concatenate some HTML tags with your PHP variables to create an easy-to-read template you can style with CSS

```
47
48
49
50
51
52 /*While there are products in our database, get and
53     display the company, the type, the roast and de-
54 while($recording=mysqli_fetch_array($result)){
55     $product_id = htmlspecialchars($recording['prod
56     $company = htmlspecialchars($recording['company
57     $type = htmlspecialchars($recording['type'], EN
58     $roast = htmlspecialchars($recording['roast'],
59     $description = htmlspecialchars($recording['des
60
61     echo "<div class='product'><div class='company'
62     echo "<span class='type'> " . $type. "</span></
63     echo "<div class='roast'>Roast: " . $roast . "<
64     echo "<div class='description'> " . $descriptio
65     echo "</div>";
66 }
67
68
69
70
71
```

READ

- Starting template on Github:
- Finished example, for reference: product_show.php

HOMEWORK

- Build another table in the coffee database to store companies (fields: company_id, name, country, website)
- Write another input form, **company_insert.html**, based on product_insert.html
- Write another input PHP script, **company_insert_result.php**, based on product_insert_result.php
- Add at least three coffee companies
- BONUS: write another listing page, **company_show.php**, based on product_show.php

NEXT WEEK

- We tackle the **U.D.** in **C.R.U.D.** (Update! Delete!)
- We'll relate tables (companies brew coffee products.
Coffee products are brewed by companies)
- SEARCH
- PHP in the real world