**Week#6-7: List of Assignments**

**PHP 1**

**Lecture:**

* Week 6 & 7 Assignments – Review  
   **Elvis: “Create, populate and use a database”**

***REMINDER: Monday, Feb. 20th (Family Day) is a holiday. College is Closed.***

* **Required Assignments: “Elvis app: Connecting to MySQL”**
* **wk6-7\_assign\_A\_php1.doc (createElvisTable.php / displayElvisTable.php)**
* **wk6-7\_assign\_B\_php1.doc (addemail.html / addemail.php)**
* **wk6-7\_assign\_C\_php1.doc (removeemail.html / removeemail.php)**
* **Required Assignments: “Elvis app … continued”**
* **wk6-7\_assign\_D\_php1.doc (sendemail.html / sendemail.php)**

4%

* **wk6-7\_assign\_E\_php1.doc (sendemail\_v2.php with Data Validation)**

Enhance the ‘send email’ assignment:

**- PART ONE:**

* **The 4 test conditions required to ensure data was entered into the FORM.**

**- PART TWO:**

* **Include the FORM code, in the sendemail.php script**
* **Have the sendemail.php page call itself**
* **Check to see if the FORM has been submitted or not**
* **Ensure that any form data that was previously submitted is populated back into the form if/when the form is displayed due to a data validation error.**

**QUIZ : Complete the Online Quizzes posted in the Week#6-7 assignments folder.  
 - Quiz expires Sunday, March 5th**

**REMINDER: The PHP MIDTERM is scheduled for Week#8.  
TEST DATE: MONDAY, Feb. 27th 8-9:30 am Room: 45H**

**WEEKLY REVIEW QUESTIONS: Complete the questions starting on PAGE 2 of this document**

**WEEKLY REVIEW QUESTIONS (WEEK#6-7): 22 questions**

**Answer the following questions. Show your answers to the instructor in-class for making along with your Weekly Assignment solutions.**

1. **Multiple databases may be running on a single MySQL server.**

**TRUE** or **FALSE**

1. MySQL has an INT and a TINYINT data type.   
    **Since it has an INT type, why would you ever use the TINYINT data type?**

|  |  |  |  |
| --- | --- | --- | --- |
| TINYINT | 1 | -128 | 127 |
|  |  | 0 | 255 |
| SMALLINT | 2 | -32768 | 32767 |
|  |  | 0 | 65535 |
| MEDIUMINT | 3 | -8388608 | 8388607 |
|  |  | 0 | 16777215 |
| INT | 4 | -2147483648 | 2147483647 |
|  |  | 0 | 4294967295 |
| BIGINT | 8 | -9223372036854775808 | 9223372036854775807 |
|  |  | 0 | 18446744073709551615 |

The **DECIMAL** data type stores a numeric value and has a fixed number of decimal places.

1. orderTotal DECIMAL(6,2)

**Based on the above definition, what is the largest number that   
 orderTotal can contain?**

1. **The DECIMAL data type is commonly used for storing   
    monetary values.**

**TRUE** or **FALSE**

Example:

salary DECIMAL(5,2)   
- is able to store any value with five digits and two decimals, so values that can be stored in the salary column range from   
-999.99 to 999.99.

STRING data types in MySQL:  
 - Use the **CHAR()** data type – when the values for a column will   
 usually have the same number of characters.  
 - Use **VARCHAR()** – when the value in a column will usually have   
 a varying number of characters.  
 i.e. myname CHAR(30) or myname VARCHAR(30)

1. **When you define a CHAR or VARCHAR type in MySQL,  
    the size refers to**

**(a) The number of characters OR**(b) **The number of bytes ?**

**5(b) Why is this important?**

By default, **MySQL** uses the **Latin-1 character set**, also known as (ISO8859-1), which provides for 256 characters with 1-byte per character, thus it takes 20 bytes to store 20 characters.  
\* If you use the Unicode character set (i.e. UTF-8), it uses 2-bytes per character … therefore 40-bytes to store 20-characters.

1. **The TEXT string data type is used to store variable length   
    characters up to a maximum size of \_\_\_\_\_\_\_\_\_\_\_\_\_ bytes.**

Date and Time data types in MySQL: **DATE** – **TIME** - **DATETIME**

1. **DATE** type – the default format for entry and display is

**YYYY-MM-DD**

1. **TIME** type – the default format for entry and display is

**HH:MM:SS**

1. **DATETIME** type – the default format for entry and display is

**YYYY-MM-DD HH:MM:SS**

1. **If a column is designed with the** CHAR(20) **data type and size,   
    and you insert a value that is less than 20 characters, the end of the value   
    will be padded with spaces.**

**TRUE** or **FALSE**

**Barry**

i.e. firstname char(20)

1. **The SQL statement required to create a table named customers is**

1. **The SQL statement required to drop a table named customers is**

*Use the following database to answer the questions below:* ***Table name: Comics***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *CharacterID* | *Name* | *RealName* | *Abilities* | *FirstAppearance* |
| ***1*** | ***Iron Man*** | ***Tony Stark*** | ***Intelligent*** | ***1963*** |
| ***2*** | ***Spider-man*** | ***Peter Parker*** | ***Scientist*** | ***1962*** |
| ***3*** | ***Hulk*** | ***Robert Bruce Banner*** | ***Physical Strength*** | ***1962*** |
| ***4*** | ***Superman*** | ***Clark Kent*** | ***Super Strength, Flight*** | ***1938*** |
| ***5*** | ***Batman*** | ***Bruce Wayne*** | ***Rich*** | ***1939*** |
| ***6*** | ***Wonder Woman*** | ***Diana*** | ***Super Strength, Flight*** | ***1941*** |

Using the **LIKE** operator in the WHERE clause:   
 The **LIKE** operator allows us to perform basic pattern-matching   
 using wildcard characters.  
 The **%** wildcard character – matches a string of one or more   
 characters.

**13. Write the query to retrieve only those characters whose Abilities contains the text ‘Strength’ anywhere in the Abilities column.   
Display the Name column only.**

**SELECT**

*What Names are returned:* ***Hulk****,* ***Superman****,* ***Wonder Woman***

*Use the following database to answer the questions below:* ***Table name: Comics***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *CharacterID* | *Name* | *RealName* | *Abilities* | *FirstAppearance* |
| ***1*** | ***Iron Man*** | ***Tony Stark*** | ***Intelligent*** | ***1963*** |
| ***2*** | ***Spider-man*** | ***Peter Parker*** | ***Scientist*** | ***1962*** |
| ***3*** | ***Hulk*** | ***Robert Bruce Banner*** | ***Physical Strength*** | ***1962*** |
| ***4*** | ***Superman*** | ***Clark Kent*** | ***Super Strength, Flight*** | ***1938*** |
| ***5*** | ***Batman*** | ***Bruce Wayne*** | ***Rich*** | ***1939*** |
| ***6*** | ***Wonder Woman*** | ***Diana*** | ***Super Strength, Flight*** | ***1941*** |

1. Using the UPDATE SQL statement to modify existing records in a table.   
     
   **Write the query to modify Batman’s Abilities to contain the text ‘Rich, detective, martial arts’.**

**UPDATE**

1. Using the DELETE SQL statement to delete rows in a table.   
     
   **Write the query to delete the Batman row (Row 5).**

**DELETE**

1. The \_\_\_\_\_\_\_\_\_\_\_ SQL statement is used to modify existing records in a table.
2. In the SQL UPDATE statement, the \_\_\_\_\_\_\_\_\_\_ clause specifies which record or records should be updated.
3. If you omit the WHERE clause in the SQL UPDATE statement, all records will be updated. **True** or **False**
4. The \_\_\_\_\_\_\_\_\_\_\_ SQL statement is used to delete rows in a table.

1. If you omit the WHERE clause in the SQL DELETE statement, all records will be   
    deleted. **True** or **False**
2. The PHP \_\_\_\_\_\_\_\_\_ function is used to determine if a variable is set and is not NULL.
3. The \_\_\_\_\_\_\_\_\_\_\_ global variable that returns the current script being executed.   
    Often used in the action of the form tag.

i.e.   
<form method="post" action="<?php echo **$\_SERVER['PHP\_SELF']**; ?>" >

*WEEK#6-7 TASK LIST\_php1.docx*