SWINBURNE UNIVERSITY OF TECHNOLOGY

COS10026 Computing Technology Inquiry Project

Lecture 11

PHP and MySQL 2 – Database Operations

Topics

- Create and Drop Tables
- Insert, Update, Delete Records
- Select and Display Records

Creating Tables

- The `CREATE TABLE` statement specifies the table and column names and the data type for each column
- The syntax for the `CREATE TABLE` statement is:

```
CREATE TABLE table_name
   (column_name TYPE, ...);
```

- Execute the `USE` statement to select a database before executing the `CREATE TABLE` statement

Creating Tables (continued)

```
...
$sqlString = "CREATE TABLE cars(
   car_id     AUTO_INCREMENT PRIMARY KEY,
   model      VARCHAR(30),
   make       VARCHAR(25),
   price      INT,
   yom        DATE)";

$queryResult = @mysqli_query($dbConnect, $sqlString)
...
```

Use INT if you do not want to store any decimal figures

add NOT NULL if field is required

Note: Usual to check to see if the table exists, and if not, create table.

Creating Tables (continued)

Type	Range	Storage
BOOL	-128 to 127 with 0 considered false	1 byte
INT or INTEGER	-2147483648 to -2147483647	4 bytes
FLOAT	-3.402823466E+38 to -1.175494351E-38, 0, and 1.175494351E+38 to 3.402823466E+38	8 bytes
DOUBLE	-1.7976931348623157E+308 to -2.2250738585072014E+308, 0, and 2.2250738585072014E+308 to 1.7976931348623157E+308	8 bytes
DATE	'1000-01-01' to '9999-12-31'	Varies
TIME	'-838:59:59' to '838:59:59'	Varies
CHAR(n)	Fixed length string between 0 to 255 characters	Number of bytes specified by n
VARCHAR(n)	Variable length string between 0 to 65,535 characters	Varies according to the number of bytes specified by n

Common MySQL field data types

Deleting Tables

- The `DROP TABLE` statement removes all data and the table definition
- The syntax for the `DROP TABLE` statement is:

```
DROP TABLE table_name;
```

Topics

- Create and Drop Tables
- Insert, Update, Delete Records
- Select and Display Records

Structured Query Language (SQL)

Common SQL keywords

Keyword	Description
INSERT	Inserts a new row into a table
UPDATE	Update field value in a record
DELETE	Deletes a row from the table
SELECT	Retrieve records from table(s)
INTO	Specifies the table into which to insert the record(s)
FROM	Specifies the table(s) from which to retrieve or delete record(s)
WHERE	Specifies the condition that must be met
ORDER BY	Sorts the records retrieved (does not affect the table)

e.g. `SELECT * FROM employees`

Adding Records

- Use the INSERT statement to add individual records to a table
- The syntax for the INSERT statement is:

```
INSERT INTO table_name VALUES(value1, value2, ...);
```

OR

```
INSERT INTO table_name (column1,column2,column3,...)
VALUES (value1,value2,value3,...);
```

- In the first case, the values entered in the VALUES list must be in the same order in which you defined the table fields
- Specify NULL in any fields for which you do not have a value e.g. for AUTO_INCREMENT field

Adding record with INSERT: PHP example

Field names and values must be in the same order

Table name

```
<?php
  require_once "settings.php";
  $conn = @mysqli_connect ($host,$user,$pwd,$sql_db);
  if ($conn) {
    $query = "INSERT INTO
            tutors  (userid,  username, password, datejoined)
            VALUES (1,'Alex','8376',curdate())";;
    $result = mysqli_query ($conn, $query);
    if ($result)  { echo "<p>Insert operation successful.</p>";}
    else { echo "<p>Insert operation unsuccessful.</p>";  }
    mysqli_close ($conn);
  } else echo "<p>Unable to connect to the db.</p>";
?>
```

UPDATE record in PHP example

```
<?php
    require_once "settings.php";
    $conn = @mysqli_connect ($host,$user,$pwd,$sql_db);
    if ($conn) {
        $query = "UPDATE tutors
                        SET password='1234'
                        WHERE userid = 1";
        $result = mysqli_query ($conn, $query);
        if ($result) {echo "<p>Update operation successful.</p>";}
        else { echo "<p>Update operation unsuccessful.</p>"; }
        mysqli_close ($conn);
    } else echo "<p>Unable to connect to the db.</p>";
?>
```

What happens if we forget the WHERE clause?

Updating Records

- To update records in a table, use the `UPDATE` statement
- The syntax for the `UPDATE` statement is:

```
UPDATE table_name
SET column_name=value
WHERE condition;
```

  - The `UPDATE` keyword specifies the name of the table to update
  - The `SET` keyword specifies the value to assign to the fields in the records that match the condition in the `WHERE` keyword

Delete record in PHP example

```
<?php
    require_once "settings.php";
    $conn = @mysqli_connect ($host,$user,$pwd,$sql_db);
    if ($conn) {
        $query = "DELETE FROM tutors WHERE userid = 1";
    $result = mysqli_query ($conn, $query);
        if ($result)  { echo "<p>Deleted"
                    .mysqli_affected_rows($dbConnect) . " record(s).</p>"; }
        else { echo "<p>Insert operation unsuccessful.</p>";  }
        mysqli_close ($conn);
    } else echo "<p>Unable to connect to the db.</p>";
?>
```

Deleting Records

To Delete records from a table:

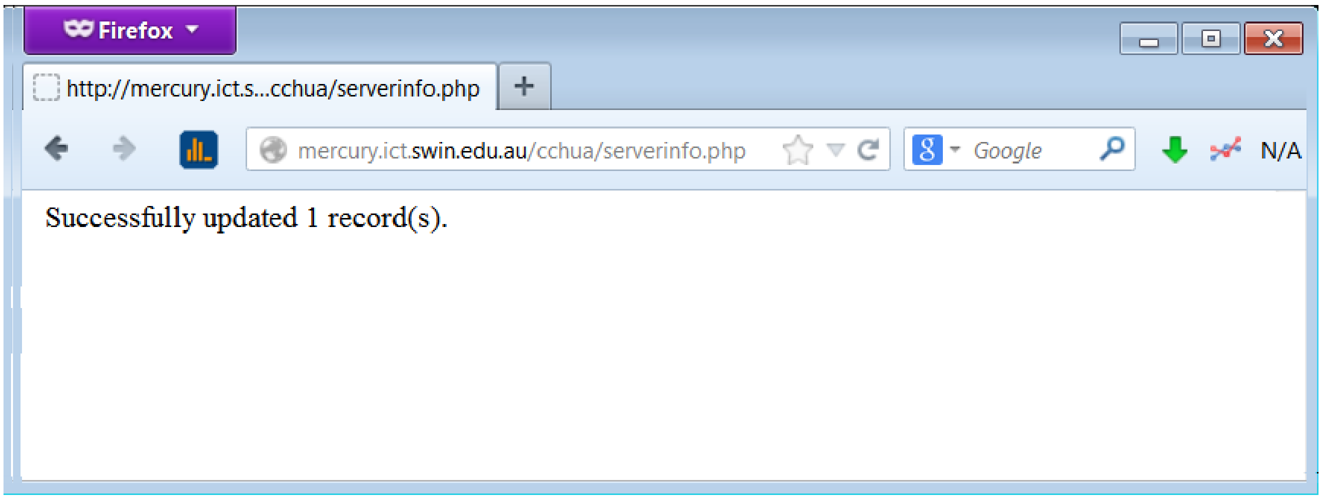
- Use the **`DELETE`** and **`WHERE`** keywords with the **`mysqli_query()`** function
- The **`WHERE`** keyword determines which records to delete in the table
- *Be careful*, if no **`WHERE`** keyword, *all records are deleted !!*

Using the **`mysqli_affected_rows()`** Function

- With queries that modify tables but do not return results (**`INSERT`**, **`UPDATE`**, and **`DELETE`** queries), use the **`mysqli_affected_rows()`** function to determine the *number of affected rows* by the query

```
$sqlString = "UPDATE cars SET price=4500
     WHERE make='Fender' AND model='DG7'";
$queryResult = @mysqli_query($dbConnect, $sqlString);
if ($queryResult){
  echo "<p>Successfully updated "
  . mysqli_affected_rows($dbConnect) . "record(s).</p>";
}
```

Using the **`mysqli_affected_rows()`** Function


Firefox
http://mercury.ict.s...cchua/serverinfo.php
mercury.ict.swin.edu.au/cchua/serverinfo.php
Google
N/A

```
Successfully updated 1 record(s).
```

Output of `mysqli_affected_rows($con)` function for an `UPDATE` query

Topics

- Create and Drop Tables
- Insert, Update, Delete Records
- Select and Display Records

Selecting and Retrieving Records

- Use the `SELECT` statement to retrieve records from a table:

```
SELECT criteria FROM table_name;
```

- Use the asterisk (*) wildcard with the `SELECT` statement to retrieve all fields from a table
- To return multiple fields, separate field names with a comma

```
mysql> SELECT model, quantity FROM inventory;
```

Retrieving Records – Filter

- The **criteria** portion of the `SELECT` statement determines which fields to retrieve from a table
- You can also specify which records to return by using the `WHERE` keyword

```
mysql> SELECT * FROM inventory
    -> WHERE make='Martin';
```

- Use the keywords `AND` and `OR` to specify more detailed conditions about the records you want to return

```
mysql> SELECT * FROM inventory
    -> WHERE make='Washburn' AND price<400;
```

Retrieving Records – Sorting

- Use the `ORDER BY` keyword with the `SELECT` statement to perform an alphanumeric sort of the results returned from a query

```
mysql> SELECT make, model FROM inventory
    -> ORDER BY make, model;
```

- To perform a reverse sort, add the `DESC` keyword after the name of the field by which you want to perform the sort

```
mysql> SELECT make, model FROM inventory
    -> ORDER BY make DESC, model;
```

Selecting Records in PHP

Be careful when constructing query:

```
$make = "Holden";
$dbTable = "inventory";

$sqlString = "SELECT model, quantity FROM
   $dbTable WHERE model = '$make'";
```

Field name not in 'quotes'

Variable name must be in 'quotes' if string

SWINBURNE UNIVERSITY OF TECHNOLOGY

Template 2 – for SQL SELECT queries

```
<?php
  require_once "settings.php";
  $conn = @mysqli_connect ($host,$user,$pwd,$sql_db);
  if ($conn) {
                    $query = "SELECT ……";
    $results = mysqli_query ($conn, $query);
    if ($results) {
    $record = mysqli_fetch_assoc ($results);
      if ($record) {
        echo "<p>At least 1 record was retrieved.</p>";
      } else echo "<p>No records retrieved.</p>";
    } else     echo "<p>MySQL operation unsuccessful.</p>";
    mysqli_close ($conn);
  } else  echo "<p>Unable to connect to the db.</p>";
?>
```

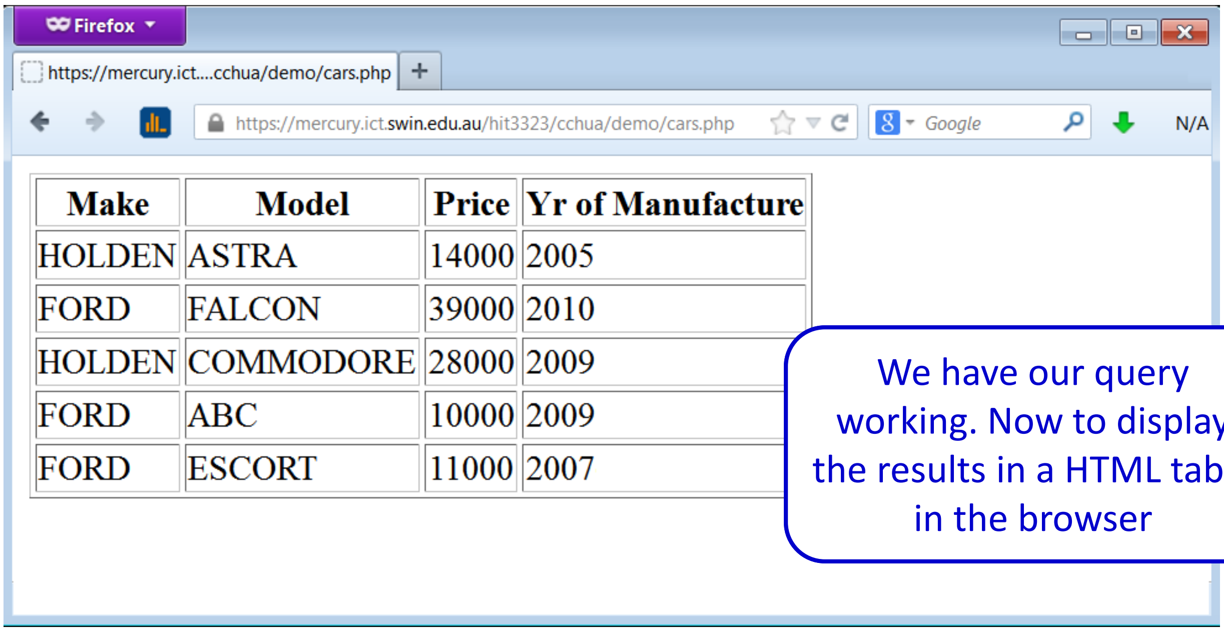
Checks if query successful

Checks if any records exist

Note: we haven't done anything with the records yet

SWINBURNE UNIVERSITY OF TECHNOLOGY

How to put the records in a html table?


Firefox
https://mercury.ict....cchua/demo/cars.php
https://mercury.ict.swin.edu.au/hit3323/cchua/demo/cars.php
Google
N/A

Make	Model	Price	Yr of Manufacture
HOLDEN	ASTRA	14000	2005
FORD	FALCON	39000	2010
HOLDEN	COMMODORE	28000	2009
FORD	ABC	10000	2009
FORD	ESCORT	11000	2007

We have our query working. Now to display the results in a HTML table in the browser

Output of the cars table in a Web browser

Selecting Records (continued)

Retrieving Records into an Associative Array

- The **mysqli_fetch_assoc()** function returns the fields in the current row of a result set into an associative array and moves the result pointer to the next row

```
echo "<table border='1'>";
echo "<tr><th>Make</th><th>Model</th>
  <th>Price</th><th>Yr of Manufacture</th></tr>";
$row = mysqli_fetch_assoc($queryResult);
while ($row) {
       echo "<tr><td>{$row['make']}</td>";
       echo "<td>{$row['model']}</td>";
       echo "<td>{$row['price']}</td>";
       echo "<td>{$row['yom']}</td></tr>";
       $row = mysqli_fetch_assoc($queryResult);
}
echo "</table>";
```

Add \n after the html if you want tidy code. `echo "</table>\n";`

Selecting Records (continued)

- Assignment and comparison can also be combined to reduce the size of the code

```
echo "<table border='1'>";
echo "<tr><th>Make</th><th>Model</th>
  <th>Price</th><th>Yr of Manufacture</th></tr>";

while ($row = mysqli_fetch_assoc($queryResult)) {
      echo "<tr><td>{$row['make']}</td>";
      echo "<td>{$row['model']}</td>";
      echo "<td>{$row['price']}</td>";
      echo "<td>{$row['yom']}</td></tr>";

}
echo "</table>";
```

This is an assignment expression, not a comparison

Selecting Records (continued)

Function	Description
mysqli_data_seek($result, position)	Moves the result pointer to a specific row in the result set
mysqli_fetch_array($result, mysqli_assoc | mysqli_num | mysqli_both)	Returns the fields in the current row of the result set into an associative array, indexed array or both, and moves the result pointer to the next row
mysqli_fetch_assoc($result)	Returns the fields in the current row of the result set into an associative array, and moves the result pointer to the next row
mysqli_fetch_row($result)	Returns the fields in the current row of the result set into an indexed array, and moves the result pointer to the next row
mysqli_fetch_lengths($result)	Returns the field lengths for the current row in a result set into an indexed array

Common PHP functions for accessing database results

Selecting Records (continued)

- The difference between **`mysqli_fetch_assoc()`** and **`mysqli_fetch_row()`** is that instead of returning the fields into an *indexed array*,
- **`mysqli_fetch_assoc()`** function returns the fields into an *associate array* and uses each ***field name*** as the *array key*

Selecting Records (continued)

Retrieving Records into an Indexed Array

- The **`mysqli_fetch_row()`** function returns the fields in the current row of a result set into an indexed array and moves the result pointer to the next row

```
echo "<table border='1'>";
echo "<tr><th>Make</th><th>Model</th>
  <th>Price</th><th>Yr of Manufacture</th></tr>";
$row = mysqli_fetch_row($queryResult);
while ($row) {
       echo "<tr><td>{$row[0]}</td>";
       echo "<td>{$row[1]}</td>";
       echo "<td>{$row[2]}</td>";
       echo "<td>{$row[3]}</td></tr>";
       $row = mysqli_fetch_row($queryResult);
}
echo "</table>";
```

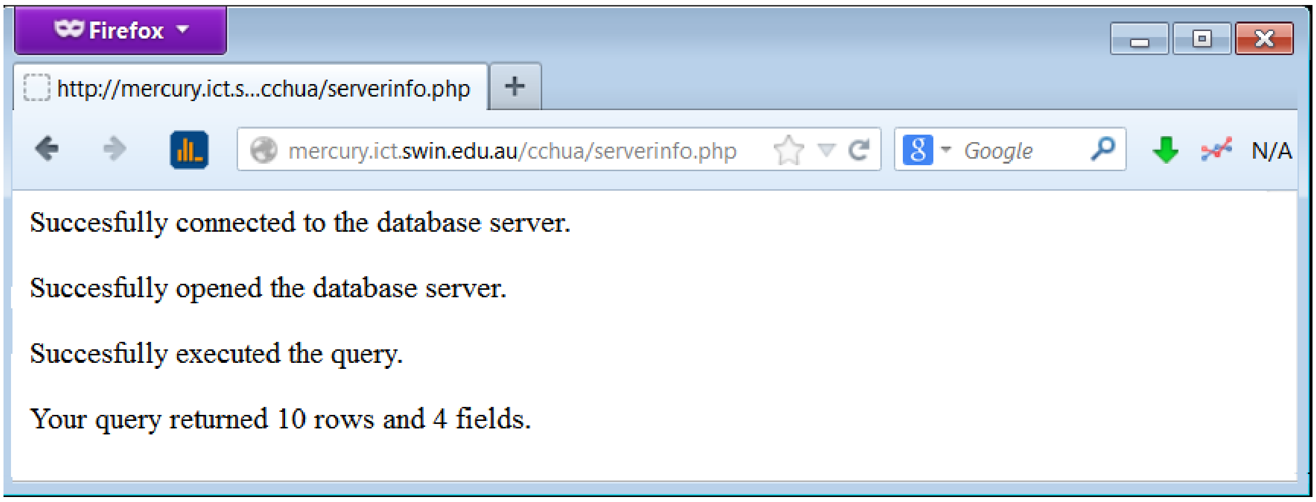
Add \n after the html if you want tidy code. **`echo "</table>\n";`**

Selecting Records (continued)

Accessing Query Result Information for queries that return result sets:

- The `mysqli_num_rows()` function returns the number of rows in a query result
- The `mysqli_num_fields()` function returns the number of fields in a query result
- Both functions accept a database result variable, eg. a query result, as an argument

Selecting Records (continued)


Firefox
http://mercury.ict.s...cchua/serverinfo.php
mercury.ict.swin.edu.au/cchua/serverinfo.php
Google
N/A

```
Succesfully connected to the database server.

Succesfully opened the database server.

Succesfully executed the query.

Your query returned 10 rows and 4 fields.
```

Output of the number of rows and fields returned from a query

Cleaning Up

- When you are finished working with query results retrieved with the `mysqli_query()` function, use the `mysqli_free_result()` function to close the resultset
- To close the resultset, pass to the `mysqli_free_result()` function the variable containing the result pointer from the `mysqli_query()` function

  e.g. `mysqli_free_result(`**`$queryResult`**`);`