

MSCI 346 – Spring 2019

LAB #5 - PHP/SQL

Introduction to Web Development

- We need to perform activities such as:
 - Connecting to the database server
 - Querying the database
 - Displaying results on a web form
 - Querying the database using user-input as a parameter

1. HTML to create web forms
2. Server Side Scripting (PHP)
3. SQL

Recap of Lab4

- PhpMyAdmin Tutorial
 - Create Tables
 - Bulk-load data
 - Running SQL queries
- PHP-Mysqli Connection

LAB 4: Exercises

- The schema for Exercise 5.2 & 5.4 is given below:

Suppliers(sid: integer, sname: string, address: string)

Parts(pid: integer, pname: string, color: string)

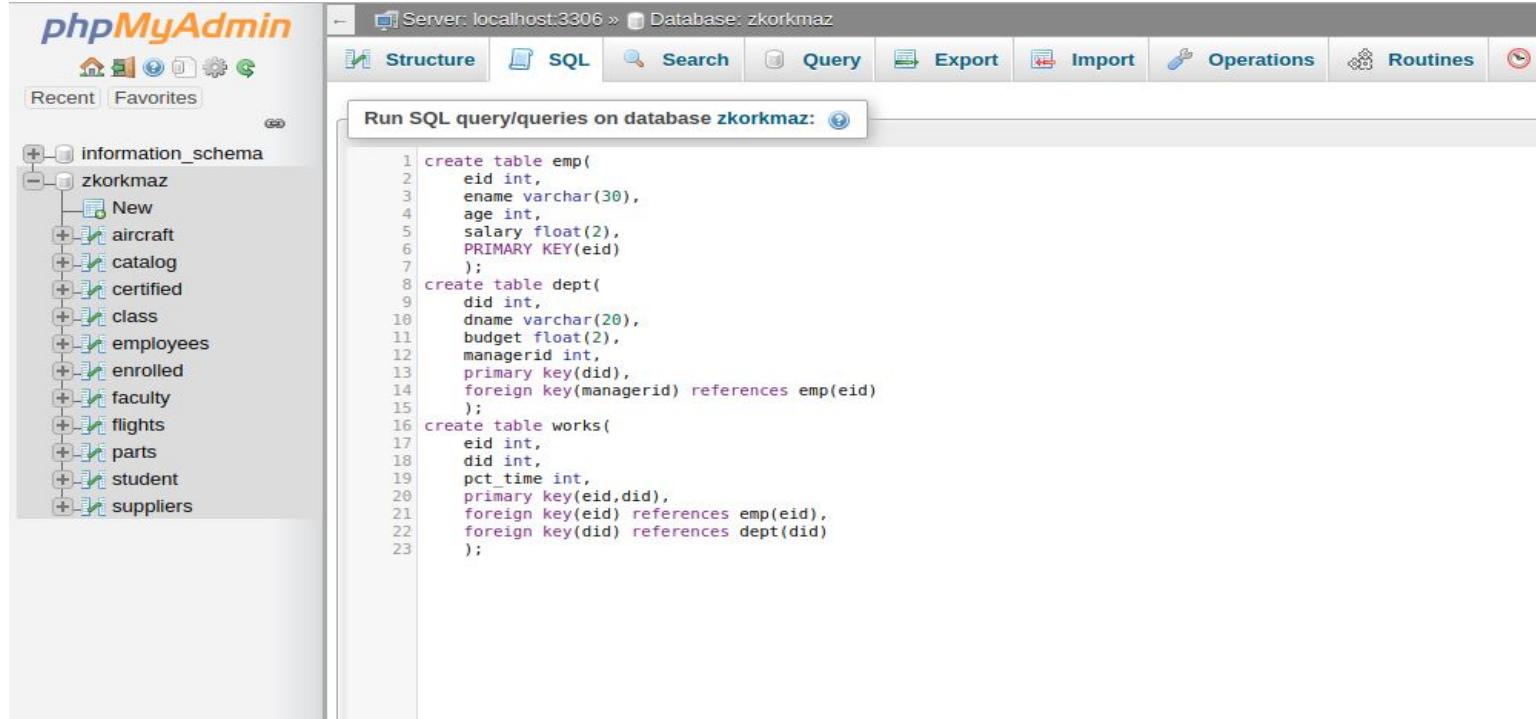
Catalog(sid: integer, pid: integer, cost: real)

Emp(eid: integer, ename: string, age: integer, salary: real)

Works(eid: integer, did: integer, pct_time: integer)

Dept(did: integer, dname: string, budget: real, managerid: integer)

Create company db tables using SQL



The screenshot displays the phpMyAdmin web interface. The top navigation bar includes tabs for Structure, SQL, Search, Query, Export, Import, Operations, and Routines. The left sidebar shows the database structure, with 'zkorkmaz' selected under the 'information_schema' database. The main content area is titled 'Run SQL query/queries on database zkorkmaz:' and contains the following SQL code:

```
1 create table emp(  
2     eid int,  
3     ename varchar(30),  
4     age int,  
5     salary float(2),  
6     PRIMARY KEY(eid)  
7 );  
8 create table dept(  
9     did int,  
10    dname varchar(20),  
11    budget float(2),  
12    managerid int,  
13    primary key(did),  
14    foreign key(managerid) references emp(eid)  
15 );  
16 create table works(  
17     eid int,  
18     did int,  
19     pct_time int,  
20     primary key(eid,did),  
21     foreign key(eid) references emp(eid),  
22     foreign key(did) references dept(did)  
23 );
```

Bulk-load data - import

The screenshot shows the phpMyAdmin interface with the 'Import' tab selected. The left sidebar shows a database structure with 'information_schema' and 'zkorkmaz' databases. Under 'zkorkmaz', there are several tables listed, including 'emp' which is currently selected. The main area displays the 'Import' form for the 'emp' table. The form includes sections for 'File to import', 'Partial import', 'Other options', 'Format', and 'Format-specific options'. The 'File to import' section has a 'Choose file' button and a character set dropdown set to 'utf-8'. The 'Partial import' section has a checkbox for allowing interruption and a skip queries input set to 0. The 'Other options' section has a checkbox for enabling foreign key checks. The 'Format' section has a dropdown set to 'SQL'. The 'Format-specific options' section has a dropdown for SQL compatibility mode set to 'NONE' and a checkbox for not using AUTO_INCREMENT for zero values. A 'Go' button is at the bottom.

Server: localhost:3306 » Database: zkorkmaz » Table: emp

Recent Favorites

information_schema
zkorkmaz
New
aircraft
catalog
certified
class
dept
emp
employees
enrolled
faculty
flights
parts
student
suppliers
works

Importing into the table "emp"

File to import:

File may be compressed (gzip, bzip2, zip) or uncompressed.
A compressed file's name must end in `.[format].[compression]`. Example: `.sql.zip`

Browse your computer: No file chosen (Max: 2,048KiB)

You may also drag and drop a file on any page.

Character set of the file:

Partial import:

☒ Allow the interruption of an import in case the script detects it is close to the PHP timeout limit. *(This might be a good way to import large files, however it can break transactions.)*

Skip this number of queries (for SQL) starting from the first one:

Other options:

☒ Enable foreign key checks

Format:

Format-specific options:

SQL compatibility mode:

☒ Do not use AUTO_INCREMENT for zero values

Today

- PhpMyAdmin
 - Advanced SQL queries
- PHP-MySQLi Connection
 - Running queries with user-supplied parameters and print query output in php

Exercise - 1

Note that the Employees relation describes pilots and other kinds of employees as well; every pilot is certified for some aircraft, and **only pilots are certified to fly**.

Flights(*fno*: integer, [↑]*from*: string, [↑]*to*: string, *distance*: integer,
departs: time, *arrives*: time, *price*: real)
Aircraft(*aid*: integer, *aname*: string, *cruisingrange*: integer)
Certified(*eid*: integer, *aid*: integer)
Employees(*eid*: integer, *ename*: string, *salary*: integer)

Write each of the following queries in SQL.

Exercise-1

- A. Find the names of aircraft such that all pilots certified to operate them have salaries more than \$80,000.
- B. For each pilot who is certified for more than three aircraft, find the eid and the maximum cruisingrange of the aircraft for which she or he is certified.
- C. Identify the routes that can be piloted by every pilot who makes more than \$100,000.
- D. Print the enames of pilots who can operate planes with cruisingrange greater than 3000 miles but are not certified on any Boeing aircraft.
- E. Compute the difference between the average salary of a pilot and the average salary of all employees (including pilots).

Solutions

1-A) Find the names of aircraft such that all pilots certified to operate them have salaries more than \$80,000.

```
SELECT DISTINCT A.aname FROM aircraft A
WHERE A.aid IN
  (SELECT C.aid
   FROM certified C, employees E
   WHERE C.aid = E.aid AND NOT EXISTS
     ( SELECT * FROM employees E1
       WHERE E1.aid = E.aid AND E1.salary < 80000 ))
```

Solutions

1-B) For each pilot who is certified for more than three aircraft, find the eid and the maximum crusingrange of the aircraft for which she or he is certified.

```
SELECT C.eid, MAX(A.crusingrange)
FROM certified C, aircraft A
WHERE C.aid = A.aid
GROUP BY C.eid
HAVING COUNT(*) > 3
```

Solutions

1-C) Identify the routes that can be piloted by every pilot who makes more than \$100,000.

```
SELECT DISTINCT F.origin, F.destination
FROM flights F
WHERE NOT EXISTS
    (SELECT * FROM employees E
     WHERE E.salary > 100000 AND
     NOT EXISTS
        (SELECT * FROM aircraft A, certified C
         WHERE A.cruisingrange > F.distance
         AND E.eid = C.eid
         AND A.aid = C.aid))
```

Solutions

1-D) Print the enames of pilots who can operate planes with cruisingrange greater than 3000 miles but are not certified on any Boeing aircraft.

```
SELECT DISTINCT E.ename FROM employees E
WHERE E.eid IN (
    (SELECT C.eid FROM certified C
    WHERE EXISTS
        (SELECT A.aid FROM aircraft A
        WHERE A.aid = C.aid AND A.cruisingrange > 3000 )
    AND NOT EXISTS
        (SELECT A1.aid
        FROM aircraft A1
        WHERE A1.aid = C.aid
        AND A1.aname LIKE 'Boeing%' )))
```

Solutions

1-E) Compute the difference between the average salary of a pilot and the average salary of all employees (including pilots).

```
SELECT Temp1.avg - Temp2.avg
FROM (SELECT AVG (E.salary) AS avg
      FROM employees E
      WHERE E.eid IN
            (SELECT DISTINCT C.eid
             FROM certified C )
      ) AS Temp1,
      (SELECT AVG (E1.salary) AS avg
      FROM employees E1 ) AS Temp2
```

Exercise - 2

Consider the following relational schema. An employee can work in more than one department; the pct time field of the Works relation shows the percent-age of time that a given employee works in a given department.

`Emp(eid: integer, ename: string, age: integer, salary: real)`

`Works(eid: integer, did: integer, pct_time: integer)`

`Dept(did: integer, dname: string, budget: real, managerid: integer)`

Write each of the following queries in SQL.

Exercise-2

- A. Print the name of each employee whose salary exceeds the budget of all of the departments that he or she works in.
- B. Find the managerids of managers who manage only departments with budgets greater than \$1 million.
- C. Find the enames of managers who manage the departments with the largest budgets.
- D. If a manager manages more than one department, he or she controls the sum of all the budgets for those departments. Find the managerids of managers who control more than \$5 million.

Solutions..

2-C) Find the enames of managers who manage the departments with the largest budgets.

```
SELECT E.ename
FROM emp E
WHERE E.eid IN
      (SELECT D.managerid
       FROM dept D
       WHERE D.budget = (SELECT MAX(D2.budget) FROM dept D2))
```

Solutions..

2-D) If a manager manages more than one department, he or she controls the sum of all the budgets for those departments. Find the managerids of managers who control more than \$5 million.

```
SELECT D.managerid
FROM dept D
WHERE 5000000 < (SELECT SUM (D2.budget)
                  FROM dept D2
                  WHERE D2.managerid = D.managerid )
```

PHP-MySQLi

PHP application needs to communicate with a database server

- **Connecting to the database server**
- **Querying the database**
- **Displaying results on a web form**
- **Taking user input through a web form**
- **Querying the database using user-input as a parameter**
- **Inserting the user-input into the database (next week)**

MySQLi

- MySQLi stands for MySQL improved
- It is an API that provides functions to connect to the database, send queries and parameters to the database and collect results.

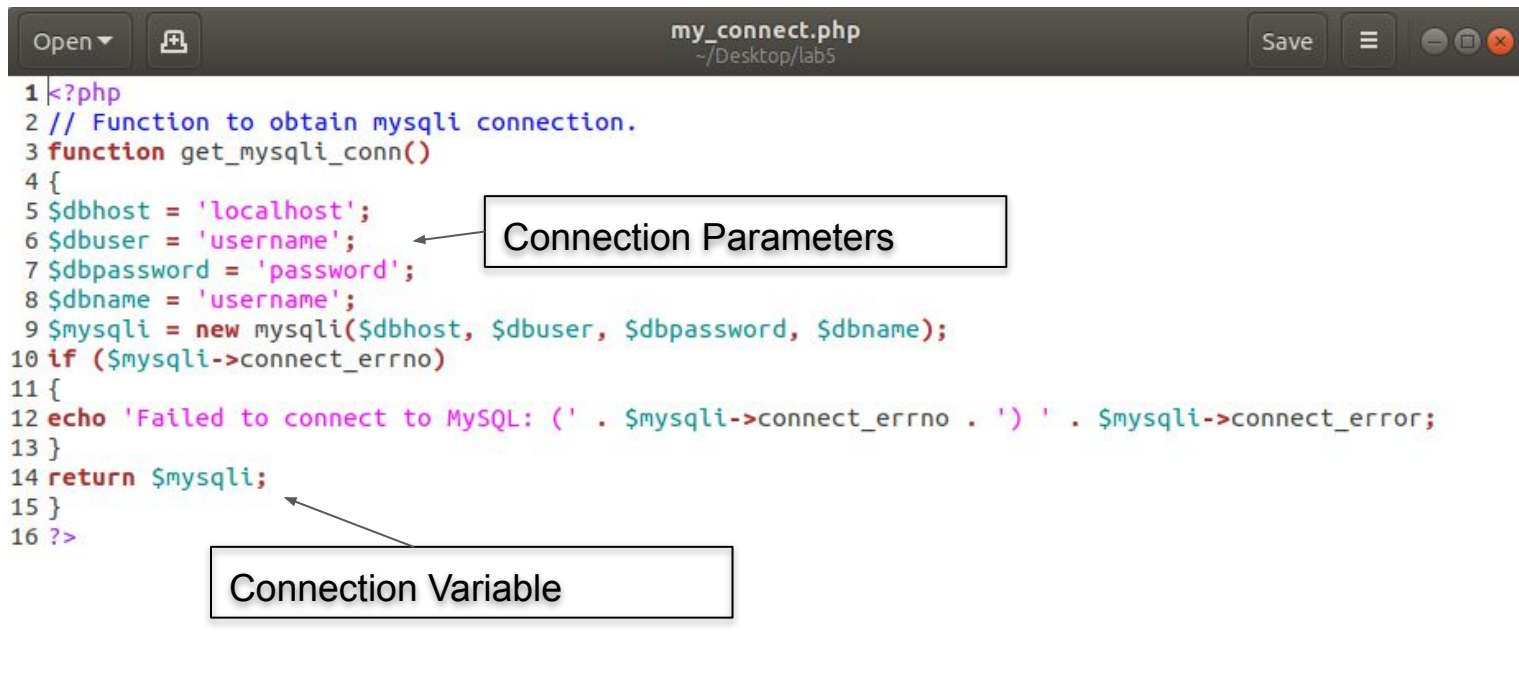
Interaction with a database using MySQLi can be categorized into 4 main stages:

1. Connecting to a database
2. Firing a query
3. Collecting its results
4. Closing the connection to the database

Documentation: <http://php.net/manual/en/book.mysqli.php>

Stage 1: Connect to a database

- Download **my_connect.php** from Learn.



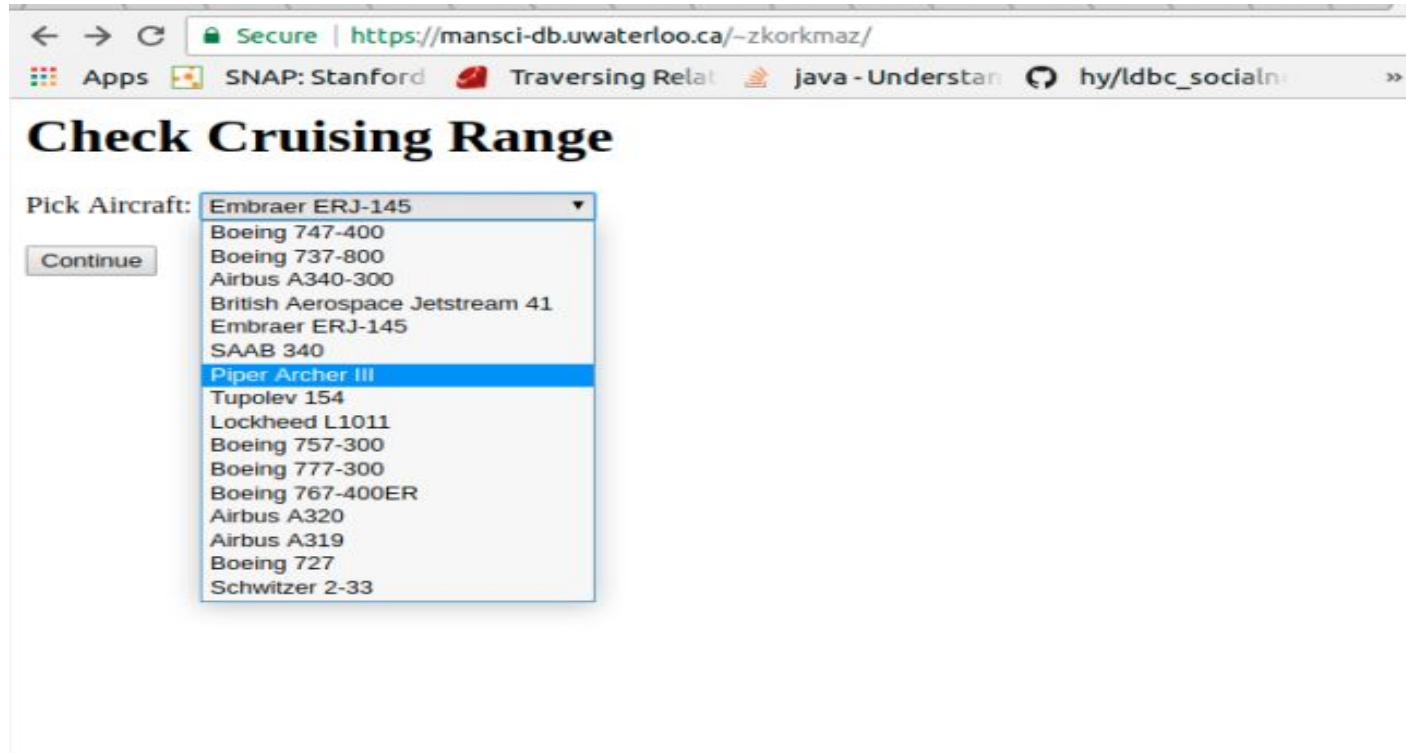
```
1 <?php
2 // Function to obtain mysqli connection.
3 function get_mysqli_conn()
4 {
5     $dbhost = 'localhost';
6     $dbuser = 'username';
7     $dbpassword = 'password';
8     $dbname = 'username';
9     $mysqli = new mysqli($dbhost, $dbuser, $dbpassword, $dbname);
10    if ($mysqli->connect_errno)
11    {
12        echo 'Failed to connect to MySQL: (' . $mysqli->connect_errno . ') ' . $mysqli->connect_error;
13    }
14    return $mysqli;
15 }
16 ?>
```

Connection Parameters

Connection Variable

Exercise-3

Application: Check cruising range of an aircraft



The screenshot shows a web browser window with the address bar displaying <https://mansci-db.uwaterloo.ca/~zkorkmaz/>. The browser's tab bar includes tabs for 'Apps', 'SNAP: Stanford', 'Traversing Relat', 'java - Understan', and 'hy/ldbc_socialn'. The main content area has the title 'Check Cruising Range' in a large, bold, black font. Below the title, there is a label 'Pick Aircraft:' followed by a dropdown menu. The dropdown menu is open, showing a list of aircraft models. The 'Piper Archer III' is currently selected and highlighted in blue. To the left of the dropdown menu is a 'Continue' button.

Check Cruising Range

Pick Aircraft: Embraer ERJ-145 ▼

- Boeing 747-400
- Boeing 737-800
- Airbus A340-300
- British Aerospace Jetstream 41
- Embraer ERJ-145
- SAAB 340
- Piper Archer III**
- Tupolev 154
- Lockheed L1011
- Boeing 757-300
- Boeing 777-300
- Boeing 767-400ER
- Airbus A320
- Airbus A319
- Boeing 727
- Schweizer 2-33


Step 1: Connect to a db

- Download `index.php` from Learn
- Include `my_connect.php` in `index.php`
- Assign connection variable

```
index.php
~/Desktop/lab5
Open
1 <body>
2 <h1>Check Cruising Range</h1>
3
4 <form action="cruisingrange.php" method="get">
5
6 <?php
7 // Enable error logging:
8 error_reporting(E_ALL ^ E_NOTICE);
9 // mysql connection via user-defined function
10 include ('./my_connect.php');
11 $mysqli = get_mysqli_conn();
12 ?>
13
14 <?php
15 // SQL statement
16 $sql = "SELECT a.aid, a.aname "
17       . "FROM aircraft a";
18
19 // Prepared statement, stage 1: prepare
20 $stmt = $mysqli->prepare($sql);
21
22 // Prepared statement, stage 2: execute
23 $stmt->execute();
24
25 // Bind result variables
26 $stmt->bind_result($aircraft_id, $aircraft_name);
27
28 /* fetch values */
29 echo '<label for="aid">Pick Aircraft: </label>';
30 echo '<select name="aid">';
31 while ($stmt->fetch())
32 {
33 printf ('<option value="%s">%s</option>', $aircraft_id, $aircraft_name);
34 }
35 echo '</select><br>';
36
37 /* close statement and connection*/
38 $stmt->close();
39 $mysqli->close();
40 ?>
41
42 <br>
43 <input type="submit" value="Continue"/>
44 </br>
45 </form>
46 </body>
```

Step 2: Fire a query

- Index page needs to prompt the user to select the name of the aircraft to check its cruising range.
 - Populate drop-down list with all aircraft names (fetch from database)
- Prepare statement.
- Execute query.

```
Open ▾  index.php  
~/Desktop/lab5  
1 <body>  
2 <h1>Check Cruising Range</h1>  
3  
4 <form action="cruisingrange.php" method="get">  
5  
6 <?php  
7 // Enable error logging:  
8 error_reporting(E_ALL ^ E_NOTICE);  
9 // mysqli connection via user-defined function  
10 include ('./my_connect.php');  
11 $mysqli = get_mysqli_conn();  
12 ?>  
13  
14 <?php  
15 // SQL statement  
16 $sql = "SELECT a.aid, a.aname "  
17       . "FROM aircraft a";  
18  
19 // Prepared statement, stage 1: prepare  
20 $stmt = $mysqli->prepare($sql);  
21  
22 // Prepared statement, stage 2: execute  
23 $stmt->execute();  
24  
25 // Bind result variables  
26 $stmt->bind_result($aircraft_id, $aircraft_name);  
27  
28 /* fetch values */  
29 echo '<label for="aid">Pick Aircraft: </label>';  
30 echo '<select name="aid">';  
31 while ($stmt->fetch())  
32 {  
33 printf ('<option value="%s">%s</option>', $aircraft_id, $aircraft_name);  
34 }  
35 echo '</select><br>';  
36  
37 /* close statement and connection*/  
38 $stmt->close();  
39 $mysqli->close();  
40 ?>  
41  
42 <br>  
43 <input type="submit" value="Continue"/>  
44 </br>  
45 </form>  
46 </body>
```

Step 3: Collect Results

- Bind selected columns to PHP variables
- Fetch values by iterating through the executed statement's results

```
Open ▾ icon
index.php
~/Desktop/lab5

1 <body>
2 <h1>Check Cruising Range</h1>
3
4 <form action="cruisingrange.php" method="get">
5
6 <?php
7 // Enable error logging:
8 error_reporting(E_ALL ^ E_NOTICE);
9 // mysqli connection via user-defined function
10 include ('./my_connect.php');
11 $mysqli = get_mysqli_conn();
12 ?>
13
14 <?php
15 // SQL statement
16 $sql = "SELECT a.aid, a.aname "
17       . "FROM aircraft a";
18
19 // Prepared statement, stage 1: prepare
20 $stmt = $mysqli->prepare($sql);
21
22 // Prepared statement, stage 2: execute
23 $stmt->execute();
24
25 // Bind result variables
26 $stmt->bind_result($aircraft_id, $aircraft_name);
27
28 /* fetch values */
29 echo '<label for="aid">Pick Aircraft: </label>';
30 echo '<select name="aid">';
31 while ($stmt->fetch())
32 {
33     printf ('<option value="%s">%s</option>', $aircraft_id, $aircraft_name);
34 }
35 echo '</select><br>';
36
37 /* close statement and connection*/
38 $stmt->close();
39 $mysqli->close();
40 ?>
41
42 <br>
43 <input type="submit" value="Continue"/>
44 </br>
45 </form>
46 </body>
```

Step 3: Collect Results

- The name of the drop-down list is “aid”. When the user hits the “continue” button, **its value** will be stored as “aid”.

```
index.php
~/Desktop/lab5

1 <body>
2 <h1>Check Cruising Range</h1>
3
4 <form action="cruisingrange.php" method="get">
5
6 <?php
7 // Enable error logging:
8 error_reporting(E_ALL ^ E_NOTICE);
9 // mysqli connection via user-defined function
10 include ('./my_connect.php');
11 $mysqli = get_mysqli_conn();
12 ?>
13
14 <?php
15 // SQL statement
16 $sql = "SELECT a.aid, a.aname "
17       . "FROM aircraft a";
18
19 // Prepared statement, stage 1: prepare
20 $stmt = $mysqli->prepare($sql);
21
22 // Prepared statement, stage 2: execute
23 $stmt->execute();
24
25 // Bind result variables
26 $stmt->bind_result($aircraft_id, $aircraft_name);
27
28 /* fetch values */
29 echo '<label for="aid">Pick Aircraft:</label>';
30 echo '<select name="aid">';
31 while ($stmt->fetch())
32 {
33 printf ('<option value="%s">%s</option>', $aircraft_id, $aircraft_name);
34 }
35 echo '</select><br>';
36
37 /* close statement and connection*/
38 $stmt->close();
39 $mysqli->close();
40 ?>
41
42 <br>
43 <input type="submit" value="Continue"/>
44 </br>
45 </form>
46 </body>
```

Step 3: Collect Results

- The name of the drop-down list is “aid”. When the user hits the “continue” button, its value will be stored as “aid” and redirect to **cruisingrange.php**

```
Open ▾ icon index.php ~/Desktop/lab5
1 <body>
2 <h1>Check Cruising Range</h1>
3
4 <form action="cruisingrange.php" method="get">
5
6 <?php
7 // Enable error logging:
8 error_reporting(E_ALL ^ E_NOTICE);
9 // mysqli connection via user-defined function
10 include ('./my_connect.php');
11 $mysqli = get_mysqli_conn();
12 ?>
13
14 <?php
15 // SQL statement
16 $sql = "SELECT a.aid, a.aname "
17       . "FROM aircraft a";
18
19 // Prepared statement, stage 1: prepare
20 $stmt = $mysqli->prepare($sql);
21
22 // Prepared statement, stage 2: execute
23 $stmt->execute();
24
25 // Bind result variables
26 $stmt->bind_result($aircraft_id, $aircraft_name);
27
28 /* fetch values */
29 echo '<label for="aid">Pick Aircraft: </label>';
30 echo '<select name="aid">';
31 while ($stmt->fetch())
32 {
33 printf ('<option value="%s">%s</option>', $aircraft_id, $aircraft_name);
34 }
35 echo '</select><br>';
36
37 /* close statement and connection*/
38 $stmt->close();
39 $mysqli->close();
40 ?>
41
42 <br>
43 <input type="submit" value="Continue"/>
44 </br>
45 </form>
46 </body>
```

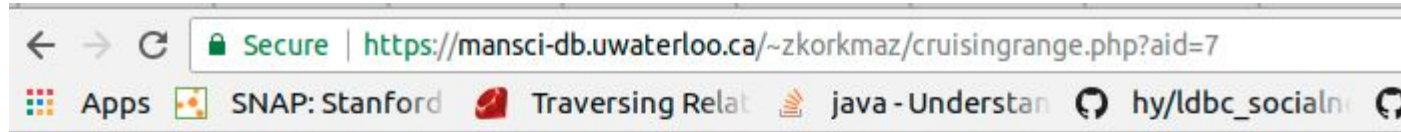
Step 4: Close Connection

- Close the SQL query statement (\$sql).
- Close the connection (\$mysqli) if there are no further transactions in this PHP file

```
Open ▾ icon
index.php
~/Desktop/lab5

1 <body>
2 <h1>Check Cruising Range</h1>
3
4 <form action="cruisingrange.php" method="get">
5
6 <?php
7 // Enable error logging:
8 error_reporting(E_ALL ^ E_NOTICE);
9 // mysqli connection via user-defined function
10 include ('./my_connect.php');
11 $mysqli = get_mysqli_conn();
12 ?>
13
14 <?php
15 // SQL statement
16 $sql = "SELECT a.aid, a.aname "
17       . "FROM aircraft a";
18
19 // Prepared statement, stage 1: prepare
20 $stmt = $mysqli->prepare($sql);
21
22 // Prepared statement, stage 2: execute
23 $stmt->execute();
24
25 // Bind result variables
26 $stmt->bind_result($aircraft_id, $aircraft_name);
27
28 /* fetch values */
29 echo '<label for="aid">Pick Aircraft: </label>';
30 echo '<select name="aid">';
31 while ($stmt->fetch())
32 {
33 printf ('<option value="%s">%s</option>', $aircraft_id, $aircraft_name);
34 }
35 echo '</select><br>';
36
37 /* close statement and connection*/
38 $stmt->close();
39 $mysqli->close();
40 ?>
41
42 <br>
43 <input type="submit" value="Continue"/>
44 </br>
45 </form>
46 </body>
```

How to use selected aircraft (user input) to find its cruising range?





Cruising Range Information

Cruising Range for Aircraft Piper Archer III is 520

Step 1: Connect to a db


- Download cruisingrange.php from Learn.
- Include my_connect.php in index.php
- Assign connection variable

```
Open  cruisingrange.php
mansci-db.uwaterloo.ca ~/public_html
Save 

1 <body>
2 <h1>Cruising Range Information</h1>
3
4 <?php
5 // Enable error logging:
6 error_reporting(E_ALL ^ E_NOTICE);
7 // mysqli connection via user-defined function
8
9 include('./my_connect.php');
10 $mysqli = get_mysqli_conn();
11
12 // SQL statement
13 $sql = "SELECT a.cruisingrange, a.aname "
14 . "FROM aircraft a "
15 . "WHERE a.aid = ?";
16
17 // Prepared statement, stage 1: prepare
18 $stmt = $mysqli->prepare($sql);
19
20 // Prepared statement, stage 2: bind and execute
21 $aid = $_GET['aid'];
22 // "i" for integer, "d" for double, "s" for string, "b" for blob
23 $stmt->bind_param('i', $aid);
24 $stmt->execute();
25
26 // Bind result variables
27 $stmt->bind_result($aircraft_cruisingrange, $aircraft_name);
28
29 /* fetch values */
30 if ($stmt->fetch())
31 {
32     printf ('Cruising Range for Aircraft %s is %s', $aircraft_name, $aircraft_cruisingrange);
33     //echo 'Cruising Range for Aircraft ' . $aircraft_name . ' is: ' . $aircraft_cruisingrange;
34 }
35 //else
36 //{
37 //echo 'Record not found';
38 //}
39
40 /* close statement and connection*/
41 $stmt->close();
42 $mysqli->close();
43 ?>
44 </body>
45
```


Step 2: Fire a query

- The MySQL server supports using anonymous positional placeholder with ?
 - Use ? for parameters (later filled in code or user-input)
- Prepare statement.



```
Open  cruisingrange.php
mansci-db.uwaterloo.ca ~/public_html Save

1 <body>
2 <h1>Cruising Range Information</h1>
3
4 <?php
5 // Enable error logging:
6 error_reporting(E_ALL ^ E_NOTICE);
7 // mysqli connection via user-defined function
8
9 include('./my_connect.php');
10 $mysqli = get_mysqli_conn();
11
12 // SQL statement
13 $sql = "SELECT a.cruisingrange, a.aname "
14 . "FROM aircraft a "
15 . "WHERE a.aid = ?";
16
17 // Prepared statement, stage 1: prepare
18 $stmt = $mysqli->prepare($sql);
19
20 // Prepared statement, stage 2: bind and execute
21 $aid = $_GET['aid'];
22 // "i" for integer, "d" for double, "s" for string, "b" for blob
23 $stmt->bind_param('i', $aid);
24 $stmt->execute();
25
26 // Bind result variables
27 $stmt->bind_result($aircraft_cruisingrange, $aircraft_name);
28
29 /* fetch values */
30 if ($stmt->fetch())
31 {
32 printf ('Cruising Range for Aircraft %s is %s', $aircraft_name, $aircraft_cruisingrange);
33 //echo 'Cruising Range for Aircraft ' . $aircraft_name . ' is: ' . $aircraft_cruisingrange;
34 }
35 //else
36 //{
37 //echo 'Record not found';
38 //}
39
40 /* close statement and connection*/
41 $stmt->close();
42 $mysqli->close();
43 ?>
44 </body>
45
```

“aid” fetched from index.php

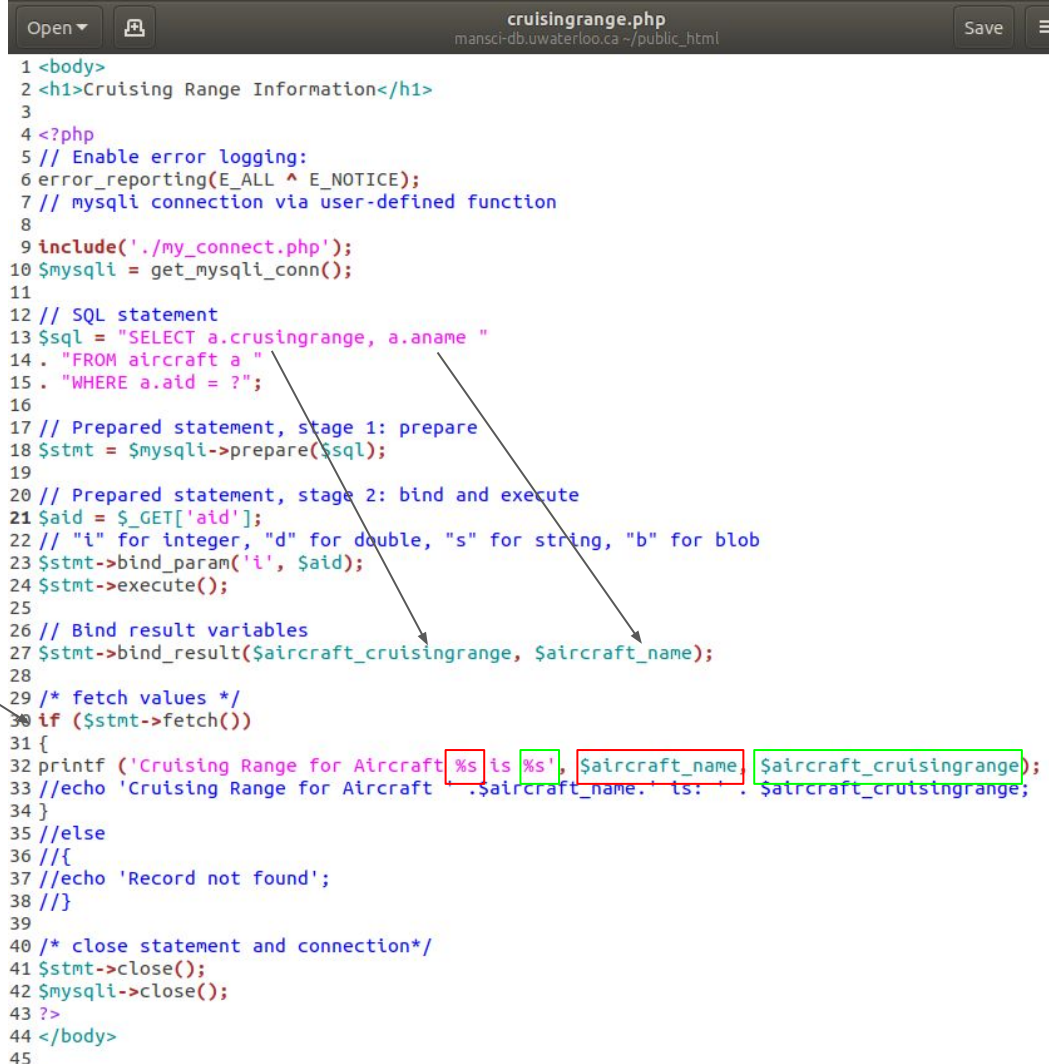
Step 2: Fire a query

- The MySQL server supports using anonymous positional placeholder with ?
 - Use ? for parameters (later filled in code or user-input)
- Prepare statement.
- Bind PHP variables to MySQL parameters
 - Should be done for all parameters that come from user-inputs

```
Open  cruisingrange.php mansci-db.uwaterloo.ca ~/public_html Save   
1 <body>  
2 <h1>Cruising Range Information</h1>  
3  
4 <?php  
5 // Enable error logging:  
6 error_reporting(E_ALL ^ E_NOTICE);  
7 // mysqli connection via user-defined function  
8  
9 include('./my_connect.php');  
10 $mysqli = get_mysqli_conn();  
11  
12 // SQL statement  
13 $sql = "SELECT a.cruisingrange, a.aname "  
14 . "FROM aircraft a "  
15 . "WHERE a.aid = ?";  
16  
17 // Prepared statement, stage 1: prepare  
18 $stmt = $mysqli->prepare($sql);  
19  
20 // Prepared statement, stage 2: bind and execute  
21 $aid = $_GET['aid'];  
22 // "i" for integer, "d" for double, "s" for string, "b" for blob  
23 $stmt->bind_param('i', $aid);  
24 $stmt->execute();  
25  
26 // Bind result variables  
27 $stmt->bind_result($aircraft_cruisingrange, $aircraft_name);  
28  
29 /* fetch values */  
30 if ($stmt->fetch())  
31 {  
32 printf ('Cruising Range for Aircraft %s is %s', $aircraft_name, $aircraft_cruisingrange);  
33 //echo 'Cruising Range for Aircraft ' . $aircraft_name . ' is: ' . $aircraft_cruisingrange;  
34 }  
35 //else  
36 //{  
37 //echo 'Record not found';  
38 //}  
39  
40 /* close statement and connection*/  
41 $stmt->close();  
42 $mysqli->close();  
43 >  
44 </body>  
45
```

Step 3: Collect Results

- Bind selected columns to PHP variables
- Fetch values



```
1 <body>
2 <h1>Cruising Range Information</h1>
3
4 <?php
5 // Enable error logging:
6 error_reporting(E_ALL ^ E_NOTICE);
7 // mysql connection via user-defined function
8
9 include('./my_connect.php');
10 $mysqli = get_mysqli_conn();
11
12 // SQL statement
13 $sql = "SELECT a.cruisingrange, a.aname "
14 . "FROM aircraft a "
15 . "WHERE a.aid = ?";
16
17 // Prepared statement, stage 1: prepare
18 $stmt = $mysqli->prepare($sql);
19
20 // Prepared statement, stage 2: bind and execute
21 $aid = $_GET['aid'];
22 // "i" for integer, "d" for double, "s" for string, "b" for blob
23 $stmt->bind_param('i', $aid);
24 $stmt->execute();
25
26 // Bind result variables
27 $stmt->bind_result($aircraft_cruisingrange, $aircraft_name);
28
29 /* fetch values */
30 if ($stmt->fetch())
31 {
32 printf ('Cruising Range for Aircraft %s is %s', $aircraft_name, $aircraft_cruisingrange);
33 //echo 'Cruising Range for Aircraft ' . $aircraft_name . ' is: ' . $aircraft_cruisingrange;
34 }
35 //else
36 //{
37 //echo 'Record not found';
38 //}
39
40 /* close statement and connection*/
41 $stmt->close();
42 $mysqli->close();
43 ?>
44 </body>
45
```

Step 4: Close Connection

```
Open  cruisingrange.php  Save  mansci-db.uwaterloo.ca ~/public_html
1 <body>
2 <h1>Cruising Range Information</h1>
3
4 <?php
5 // Enable error logging:
6 error_reporting(E_ALL ^ E_NOTICE);
7 // mysqli connection via user-defined function
8
9 include('./my_connect.php');
10 $mysqli = get_mysqli_conn();
11
12 // SQL statement
13 $sql = "SELECT a.cruisingrange, a.aname "
14 . "FROM aircraft a "
15 . "WHERE a.aid = ?";
16
17 // Prepared statement, stage 1: prepare
18 $stmt = $mysqli->prepare($sql);
19
20 // Prepared statement, stage 2: bind and execute
21 $aid = $_GET['aid'];
22 // "i" for integer, "d" for double, "s" for string, "b" for blob
23 $stmt->bind_param('i', $aid);
24 $stmt->execute();
25
26 // Bind result variables
27 $stmt->bind_result($aircraft_cruisingrange, $aircraft_name);
28
29 /* fetch values */
30 if ($stmt->fetch())
31 {
32 printf ('Cruising Range for Aircraft %s is %s', $aircraft_name, $aircraft_cruisingrange);
33 //echo 'Cruising Range for Aircraft ' . $aircraft_name . ' is: ' . $aircraft_cruisingrange;
34 }
35 //else
36 //{
37 //echo 'Record not found';
38 //}
39
40 /* close statement and connection*/
41 $stmt->close();
42 $mysqli->close();
43
44 </body>
45
```

Exercise-4 (Next Week)

Application: Find the aircrafts that a pilot is certified for and have cruising ranges greater than a specified value.

- List all employees in drop-down list
 - Input a cruising range
 - Use the selected pilot and cruising range to list the aircrafts that she or he is certified for.
 - If the result is empty -> print “No record found”
- } You need to bind two parameters!!