

# MSCI 346 – Spring 2019

LAB 4: 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 Lab3!

- Displaying outputs, HTML lists, tables
- GET&POST Methods
- Simple Examples:
  - Body mass calculator
  - Simple Calculator

[https://www.w3schools.com/tags/tag\\_select.asp](https://www.w3schools.com/tags/tag_select.asp)

[https://www.w3schools.com/html/html\\_tables.asp](https://www.w3schools.com/html/html_tables.asp)

[https://www.w3schools.com/html/html\\_lists.asp](https://www.w3schools.com/html/html_lists.asp)

# HTML's Method Get and Post

- Two commonly used methods for a request-response between a client and server are GET and POST

|                               | GET  | POST  |
|-------------------------------|--|---|
| Visibility                    | Data is visible to everyone as it sent in the URL. Less Secure.                                  | Data is not displayed in the URL. The requested data is sent in the HTTP message body of a POST request. More secure. |
| BACK button/Reload            | No Change / Harmless   | Data will be re-submitted   |
| History, Cache and Bookmarked | Parameters remain in browser history. Can be cached and bookmarked.                              | Parameters are not saved in browser history. Cannot be cached or bookmarked.  |
| Restrictions on data length   | The length of a URL and thus the size of data is limited (maximum URL length is 2048 characters) | No restrictions   |

← → ↻ 🔒 https://mansci-db.uwaterloo.ca/~zkorkmaz/index.html

Apps SNAP: Stanford Net... Traversing Relations... java - Understandin...

## Form example using GET method

Enter your value

Send Reset

## Form example using POST method

Enter your value

Send Reset

← → ↻ 🔒 https://mansci-db.uwaterloo.ca/~zkorkmaz/formPost.php

Apps SNAP: Stanford Net... Traversing Relations... java - Understandin... hy/ldbc\_socialnet\_b... hustlijian/ldbc\_snb\_...

POST method

### Form example using POST method

Value passed from HTML-form is korkmaz It sent in HTTP request and cannot be seen in address bar

[Back to form](#)

← → ↻ 🔒 https://mansci-db.uwaterloo.ca/~zkorkmaz/formPost.php

Apps SNAP: Stanford Net... Traversing Relations... java - Understandin... hy/ldbc\_socialnet\_b... hustlijian/ldbc\_snb\_...

POST method

### Form example using POST method

Value passed from HTML-form is korkmaz It sent in HTTP request and cannot be seen in address bar

[Back to form](#)

# More PHP syntax: if-else and while

```
...
<?php

$numerator = 30;
$denominator = 5;
$div_result = $numerator / $denominator;


echo '<p>';
if ($div_result % 2 == 0) { // % is modulo
    echo $div_result . ' is even.';
} else {
    echo $div_result . ' is odd.';
}
echo '</p>';

?>
...
```

 msci-teaching.uwaterloo.ca/~r24mille/

6 is even.







```
...
<?php
$count = 0;
while ($count < 10) {
    print $count. ' <br/>\n';
    $count++; // $count = $count + 1;
}
?>
...
```

 msci-teaching.uwaterloo.ca/~r24mille/

0  
1  
2  
3  
4  
5  
6  
7  
8  
9

# Simple Calculator

← → ↻ <https://mansci-db.uwaterloo.ca/~zkorkmaz/calculator.html>

 Apps  SNAP: Stanford Net...  Traversing Relations...  java - Understandin...  hy/ldbc\_socialnet\_b...  hustlijian/ldbc\_snb\_...

## Simple Calculator

Enter First Number

Enter Second Number

Select Operator

=

[Back to home](#)

← → ↻ <https://mansci-db.uwaterloo.ca/~zkorkmaz/calculator.php>

 Apps  SNAP: Stanford Net...  Traversing Relations...  java - Understandin...

The result for 2 / 4 is:

0.5

[Back to calculator](#)

# BMI Calculator

←

→

↺

https://mansci-db.uwaterloo.ca/~zkorkmaz/bmi.html

Apps

SNAP: Stanford Net...

Traversing Relations...

java - Understandin...

hy/ldbc\_soci

## BMI Calculator

First name

Jill

Last name

Smith

Height in m

1.75

Weight in kg

72

Age

50

Calculate

[Back to home](#)

←

→

↺

https://mansci-db.uwaterloo.ca/~zkorkmaz/bmi.php

Apps

SNAP: Stanford Net...

Traversing Relations...

java - Understandin...

hy/ldbc\_socialnet\_b...

hustlijian/ldbc\_snb\_...

working commit

## BMI Calculator

| BMI Result |          |     |                 |                |
|------------|----------|-----|-----------------|----------------|
| Firstname  | Lastname | Age | BMI             | Category       |
| Jill       | Smith    | 50  | 23.510204081633 | Healthy weight |

[Back to BMI Calculator](#)



# Today...

- Provide more details on phpMyAdmin
- Use phpMyAdmin to:
  - Create a table
  - Insert data into the table
    - Tuple-wise
    - Bulk-load data from a CSV file
  - View Data in tables
  - Run simple SQL queries
- PHP and MySQL

# Log in to phpMyAdmin

- Log on to phpMyAdmin (given in the Lab Manual 1)
  - Go to <https://mansci-db.uwaterloo.ca/phpmyadmin>. In the pop-up, enter your username and password to log on to the server.
    - *This gives you access to the ecdb server*

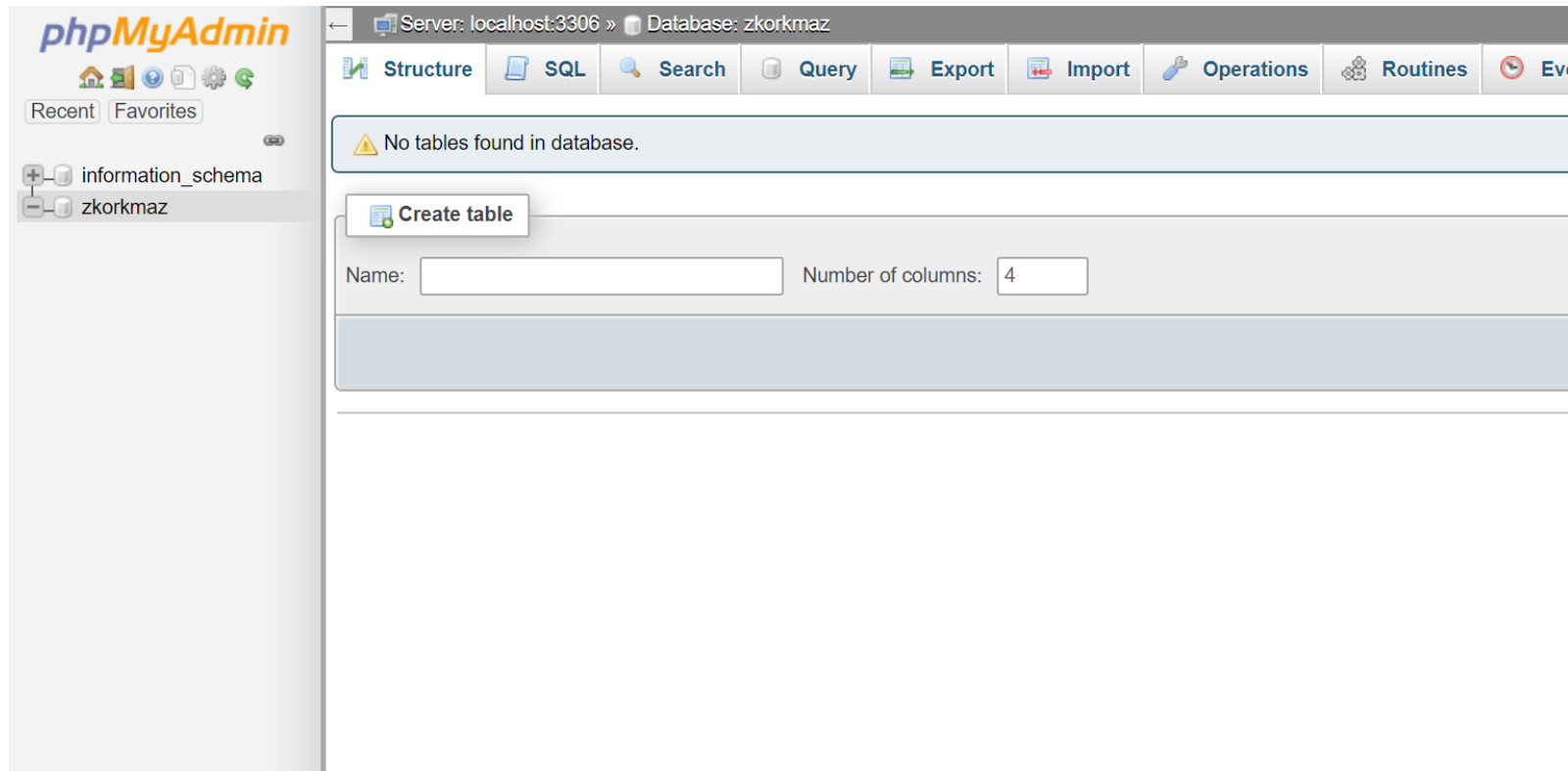
# phpMyAdmin

The screenshot displays the phpMyAdmin web interface for a MySQL server at localhost:3306. The interface is organized into several sections:

- Left Sidebar:** Contains the phpMyAdmin logo, navigation icons (home, databases, SQL, status, export, import, settings, variables, charsets, engines, plugins), and tabs for "Recent" and "Favorites". Below these are tree views for "information\_schema" and "zkorkmaz".
- Top Navigation Bar:** A horizontal bar with icons and labels for "Databases", "SQL", "Status", "Export", "Import", "Settings", "Variables", "Charsets", "Engines", and "Plugins".
- General settings:** A panel with a "Change password" link and a "Server connection collation" dropdown menu currently set to "utf8mb4\_unicode\_ci".
- Appearance settings:** A panel with a "Language" dropdown set to "English", a "Theme" dropdown set to "pmahomme", and a "Font size" dropdown set to "82%". A "More settings" link is also present.
- Database server:** A panel listing server details:
  - Server: Localhost via UNIX socket
  - Server type: MySQL
  - Server connection: SSL is not being used
  - Server version: 8.0.16 - MySQL Community Server - GPL
  - Protocol version: 10
  - User: zkorkmaz@localhost
  - Server charset: UTF-8 Unicode (utf8)
- Web server:** A panel listing web server details:
  - Apache/2.4.39 (Ubuntu)
  - Database client version: libmysql - mysqlnd 5.0.12-dev - 20150407 - \$Id: 3591daad22de08524295e1bd073aceeff11e6579 \$
  - PHP extension: mysqli, curl, mbstring
  - PHP version: 7.2.19-0ubuntu0.18.04.1
- phpMyAdmin:** A panel at the bottom right showing version information (4.8.5, latest stable version: 4.9.0.1) and links to "Documentation" and "Official Homepage".
- Console:** A small "Console" tab is visible at the bottom left.

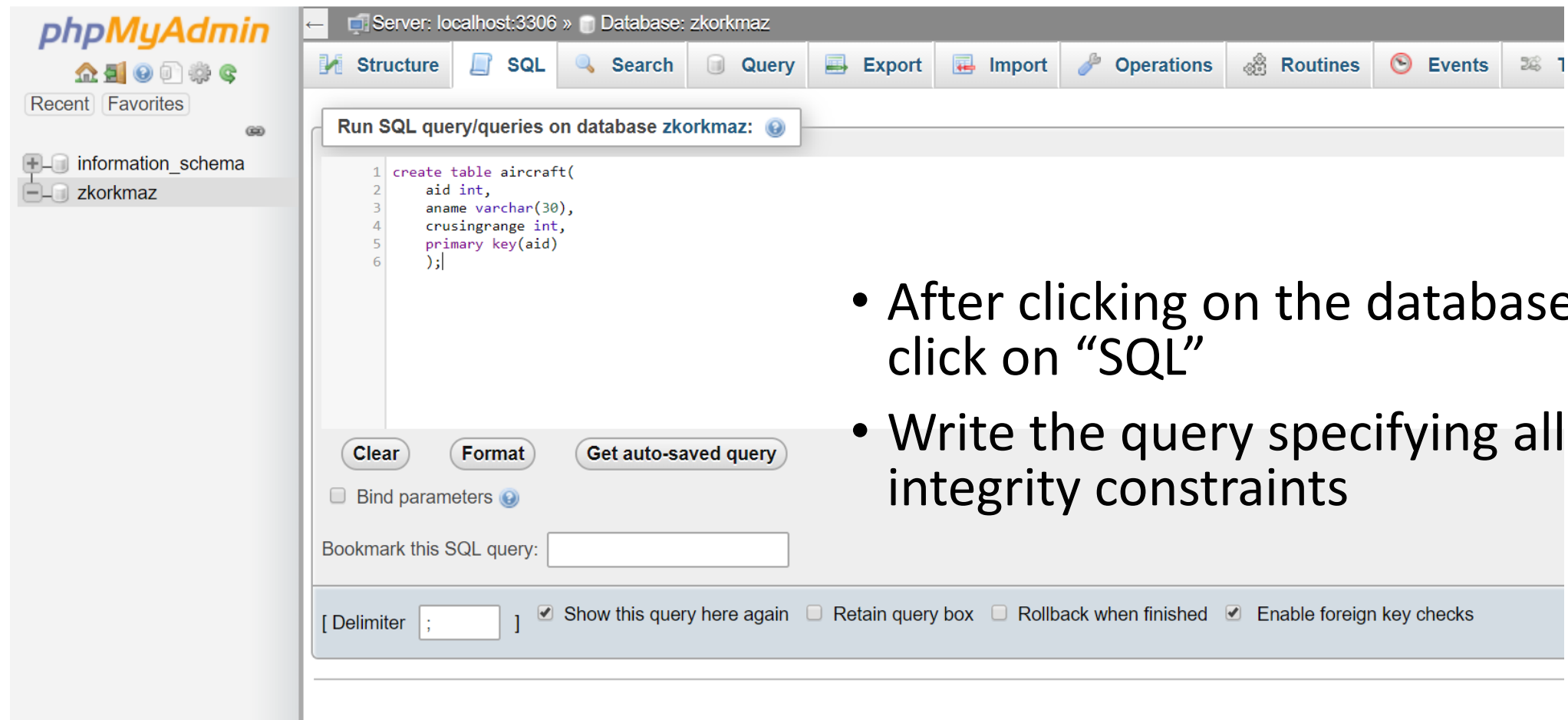
# Create Table

- Click on the name of the database in which you want to create the table (your Quest username)
- **You can create a table using the GUI**



# Create Table

- You can create a table using the GUI or **through the SQL editor**



- After clicking on the database name, click on “SQL”
- Write the query specifying all integrity constraints

# Exercise 1: Create airline db tables using SQL

- We want to create the tables in our own databases (i.e. our username's database).

**Exercise 5.3** The following relations keep track of airline flight information:

`Flights(flno: integer, from: string, to: string, distance: integer,  
departs: time, arrives: time, price: real)`

`Aircraft(aid: integer, aname: string, cruisingrange: integer)`

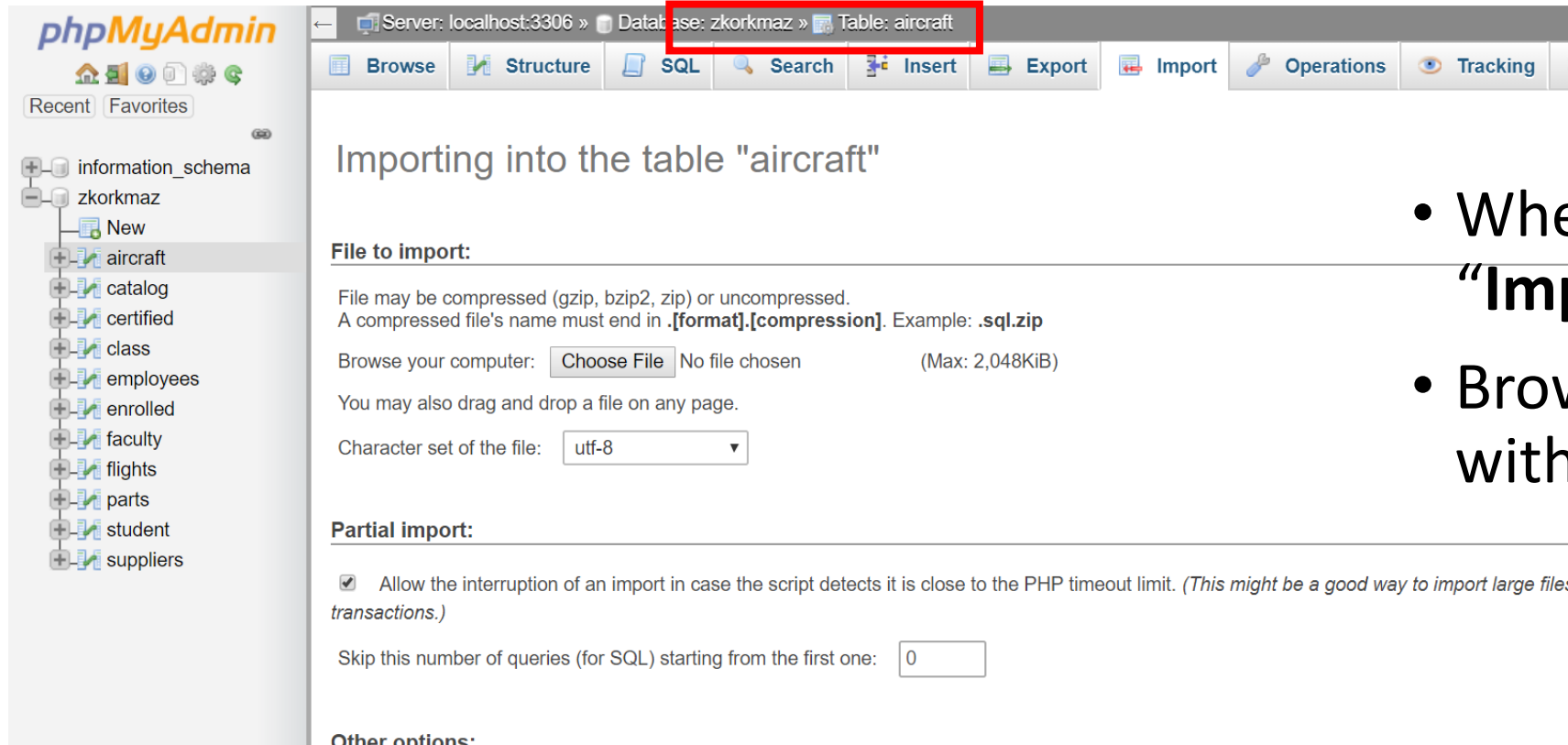
`Certified(eid: integer, aid: integer)`

`Employees(eid: integer, ename: string, salary: integer)`

Note: Will need to create “aircraft” and “employees” before “certified”. If we try to create “certified” before, it will give an error as its dependencies/foreign constraints will not be fulfilled

# Bulk-load data from a file into a table - Import

- Need to be in the table's location in which the data has to be imported
  - Click on the database name, click on the table's name in which you wish to import the data (from the panel on the left or the list on the screen) Make sure you see the correct location displayed in the Import tab



The screenshot shows the phpMyAdmin interface. On the left, the database structure tree is visible, with 'zkorkmaz' selected and 'aircraft' highlighted. The main panel shows the 'Import' tab for the 'aircraft' table. The breadcrumb path at the top is 'Server: localhost:3306 » Database: zkorkmaz » Table: aircraft', with 'Table: aircraft' highlighted in a red box. The 'Import' tab is active, displaying the 'Importing into the table "aircraft"' screen. The 'File to import:' section includes a 'Choose File' button, a file size limit of 2,048KiB, and a character set dropdown set to 'utf-8'. The 'Partial import:' section has a checkbox for 'Allow the interruption of an import...' which is checked, and a text input for 'Skip this number of queries (for SQL) starting from the first one:' with the value '0'.

- When in the table, click **“Import”**
- Browse and select the file with the data

# Bulk-load data from a file into a table

- On success, will see the following screen with the number of rows inserted and the respective SQL queries

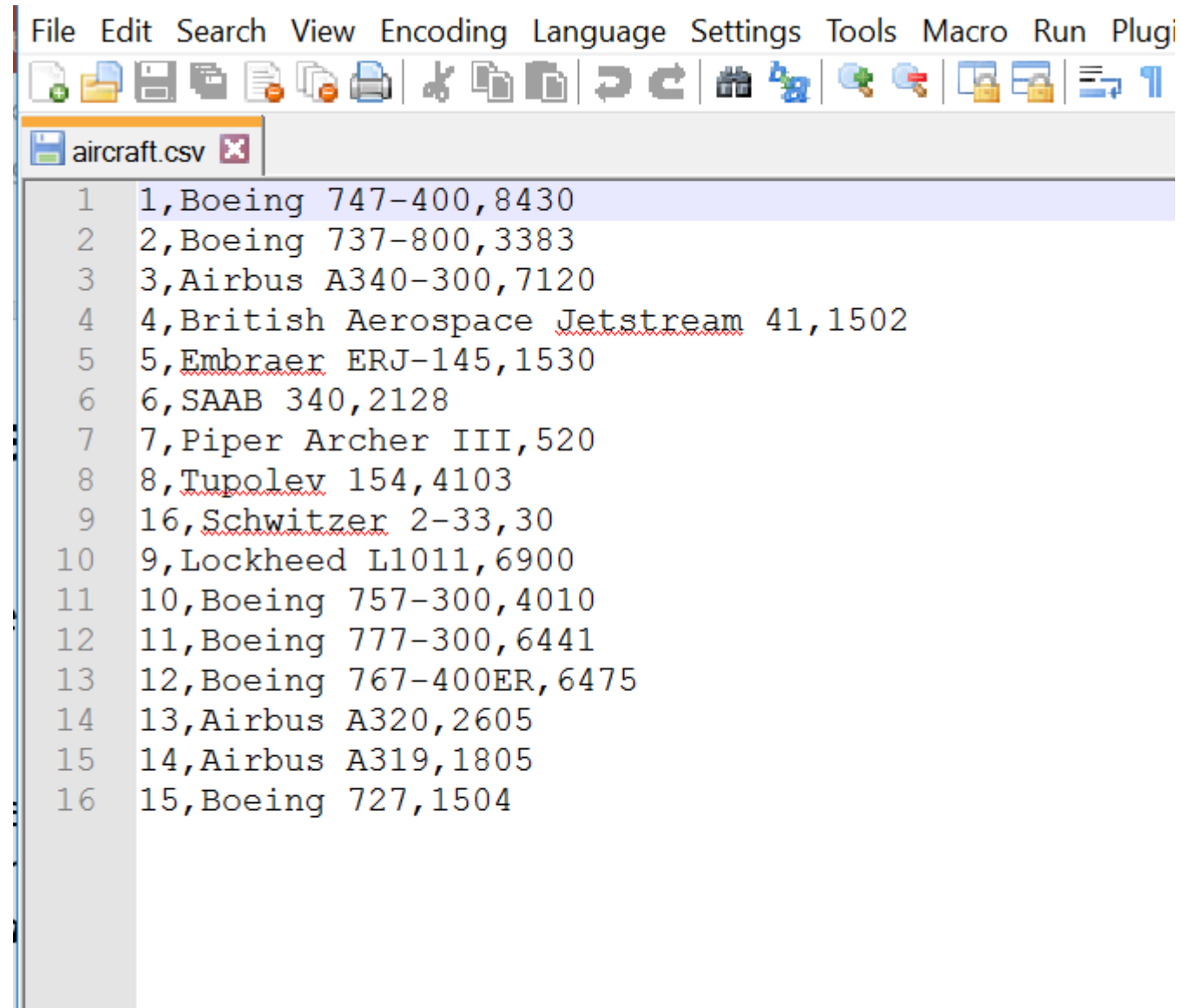
The screenshot displays the phpMyAdmin interface for the 'aircraft' table in the 'zkorkmaz' database. A red box highlights the success message: 'Import has been successfully finished, 16 queries executed. (aircraft.csv)'. Below this, four rows of data are shown, each with a green checkmark, the number of rows inserted, the query execution time, and the SQL query used. The queries are INSERT statements for different aircraft models.

| Row | Status            | Query Execution Time | SQL Query  |
|-----|-------------------|----------------------|--|
| 1   | ✓ 1 row inserted. | 0.0107 seconds.      | <code>INSERT INTO `aircraft` VALUES ('1', 'Boeing 747-400', '8430')</code>                 |
| 2   | ✓ 1 row inserted. | 0.0145 seconds.      | <code>INSERT INTO `aircraft` VALUES ('2', 'Boeing 737-800', '3383')</code>                 |
| 3   | ✓ 1 row inserted. | 0.0049 seconds.      | <code>INSERT INTO `aircraft` VALUES ('3', 'Airbus A340-300', '7120')</code>                |
| 4   | ✓ 1 row inserted. | 0.0066 seconds.      | <code>INSERT INTO `aircraft` VALUES ('4', 'British Aerospace Jetstream 41', '1502')</code> |



# Exercise 2: Bulk-load data into tables

- Insert tuples into the tables
  - Aircraft
  - Flights
  - Employees
  - Certified
- Download .csv files from Learn
- Note: format of the CSV file
  - column values are separated by commas
  - Order of column values is the same as in the table's structure



The screenshot shows a text editor window titled 'aircraft.csv'. The editor contains 16 lines of text, each representing a tuple in a CSV format. The lines are numbered 1 through 16 on the left. The data is as follows:

| Line | CSV Entry                             |
|------|---------------------------------------|
| 1    | 1,Boeing 747-400,8430                 |
| 2    | 2,Boeing 737-800,3383                 |
| 3    | 3,Airbus A340-300,7120                |
| 4    | 4,British Aerospace Jetstream 41,1502 |
| 5    | 5,Embraer ERJ-145,1530                |
| 6    | 6,SAAB 340,2128                       |
| 7    | 7,Piper Archer III,520                |
| 8    | 8,Tupolev 154,4103                    |
| 9    | 16,Schweizer 2-33,30                  |
| 10   | 9,Lockheed L1011,6900                 |
| 11   | 10,Boeing 757-300,4010                |
| 12   | 11,Boeing 777-300,6441                |
| 13   | 12,Boeing 767-400ER,6475              |
| 14   | 13,Airbus A320,2605                   |
| 15   | 14,Airbus A319,1805                   |
| 16   | 15,Boeing 727,1504                    |

# Table-related operations

- All table-related operations can be done using both the GUI of MyPhPAdmin as well as the SQL editor
- Use the buttons specified or the SQL query to perform the required operation

| Table-related operation   | phpMyAdmin GUI                   | SQL query                             |
|---|----------------------------------|---------------------------------------|
| Viewing all rows  | Browse                           | select * from tablename               |
| Viewing and altering the structure of the columns of the tables | Structure<br>Structure -> Change | describe tablename<br>alter table ... |
| Selecting tuples based on criteria                              | Search                           | select ... from ... where ...         |
| Insert tuple into table   | Insert                           | insert into ...                       |
| Delete all the data from a table                                | Empty                            | truncate tablename                    |
| Delete a table  | Drop                             | drop table tablename                  |

# Table-related operations

← → ↻ [https://mansci-db.uwaterloo.ca/phpmyadmin/db\\_structure.php?server=1&db=zkorkmaz](https://mansci-db.uwaterloo.ca/phpmyadmin/db_structure.php?server=1&db=zkorkmaz) 🔑 ☆ 📄 📄 📄 📄 | Z ⋮

📄 Apps 📄 SNAP: Stanford Net... 📄 Traversing Relations... 📄 java - Understandin... 📄 hy/ldbc\_socialnet\_b... 📄 hustlijian/ldbc\_snb\_... 📄 working commit Id... » 📄 Other bookmarks

**phpMyAdmin**

🏠 📄 ⚙️ 📄 📄

Recent Favorites

- information\_schema
- zkorkmaz
  - New
  - aircraft
  - catalog
  - certified
  - class
  - employees
  - enrolled
  - faculty
  - flights
  - parts
  - student
  - suppliers

Server: localhost:3306 » Database: zkorkmaz ⚙️ ↕

Structure SQL Search Query Export Import Operations Routines Events Triggers More

Filters

Containing the word:

|                          | Table     | Action                                      | Rows | Type   | Collation          | Size    | Overhead |
|--------------------------|-----------|---|------|--------|--------------------|---------|----------|
| <input type="checkbox"/> | aircraft  | ★ Browse Structure Search Insert Empty Drop | 16   | InnoDB | utf8mb4_0900_ai_ci | 16 KiB  | -        |
| <input type="checkbox"/> | catalog   | ★ Browse Structure Search Insert Empty Drop | 11   | InnoDB | utf8mb4_0900_ai_ci | 32 KiB  | -        |
| <input type="checkbox"/> | certified | ★ Browse Structure Search Insert Empty Drop | 69   | InnoDB | utf8mb4_0900_ai_ci | 32 KiB  | -        |
| <input type="checkbox"/> | class     | ★ Browse Structure Search Insert Empty Drop | 21   | InnoDB | utf8mb4_0900_ai_ci | 32 KiB  | -        |
| <input type="checkbox"/> | employees | ★ Browse Structure Search Insert Empty Drop | 31   | InnoDB | utf8mb4_0900_ai_ci | 16 KiB  | -        |
| <input type="checkbox"/> | enrolled  | ★ Browse Structure Search Insert Empty Drop | 20   | InnoDB | utf8mb4_0900_ai_ci | 32 KiB  | -        |
| <input type="checkbox"/> | faculty   | ★ Browse Structure Search Insert Empty Drop | 15   | InnoDB | utf8mb4_0900_ai_ci | 16 KiB  | -        |
| <input type="checkbox"/> | flights   | ★ Browse Structure Search Insert Empty Drop | 18   | InnoDB | utf8mb4_0900_ai_ci | 16 KiB  | -        |
| <input type="checkbox"/> | parts     | ★ Browse Structure Search Insert Empty Drop | 9    | InnoDB | utf8mb4_0900_ai_ci | 16 KiB  | -        |
| <input type="checkbox"/> | student   | ★ Browse Structure Search Insert Empty Drop | 24   | InnoDB | utf8mb4_0900_ai_ci | 16 KiB  | -        |
| <input type="checkbox"/> | suppliers | ★ Browse Structure Search Insert Empty Drop | 4    | InnoDB | utf8mb4_0900_ai_ci | 16 KiB  | -        |
|                          | 11 tables | Sum   | 238  | InnoDB | utf8mb4_0900_ai_ci | 240 KiB | 0 B      |

⬆️ ☐ Check all With selected: ▾

🖨️ Print 📖 Data dictionary

📄 Create table

🖨️ Console

# Tuple-related operations

- All tuple-related operations can be done using both the GUI of MyPhPAdmin as well as the SQL editor
- Use the buttons specified or the SQL query to perform the required operation
- Note: The GUI can be used for performing these operations for single as well as many tuples.

| Table-related operation   | phpMyAdmin GUI | SQL query                    |
|---------------------------|----------------|------------------------------|
| Edit values in a tuple    | Edit           | update ... set ... where ... |
| Delete tuple/s in a table | Delete         | delete from ...              |

# Tuple-related operations

phpMyAdmin

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

Browse Structure SQL Search Insert Export Import Operations Tracking Triggers

Showing rows 0 - 15 (16 total, Query took 0.0008 seconds.)

`SELECT * FROM `aircraft``

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

|                          |      |      |        | aid | aname                          | crusingrange |
|--------------------------|------|------|--------|-----|--------------------------------|--------------|
| <input type="checkbox"/> | Edit | Copy | Delete | 1   | Boeing 747-400                 | 8430         |
| <input type="checkbox"/> | Edit | Copy | Delete | 2   | Boeing 737-800                 | 3383         |
| <input type="checkbox"/> | Edit | Copy | Delete | 3   | Airbus A340-300                | 7120         |
| <input type="checkbox"/> | Edit | Copy | Delete | 4   | British Aerospace Jetstream 41 | 1502         |
| <input type="checkbox"/> | Edit | Copy | Delete | 5   | Embraer ERJ-145                | 1530         |
| <input type="checkbox"/> | Edit | Copy | Delete | 6   | SAAB 340                       | 2128         |
| <input type="checkbox"/> | Edit | Copy | Delete | 7   | Piper Archer III               | 520          |
| <input type="checkbox"/> | Edit | Copy | Delete | 8   | Tupolev 154                    | 4103         |
| <input type="checkbox"/> | Edit | Copy | Delete | 9   | Lockheed L1011                 | 6900         |
| <input type="checkbox"/> | Edit | Copy | Delete | 10  | Boeing 757-300                 | 4010         |
| <input type="checkbox"/> | Edit | Copy | Delete | 11  | Boeing 777-300                 | 6441         |
| <input type="checkbox"/> | Edit | Copy | Delete | 12  | Boeing 767-400ER               | 6475         |

Console

# Run SQL queries using the SQL editor

- Click on the database name (*username*)
- Click on “SQL”
- Write the query in the SQL editor & Hit “Go”
  - On success you would see the result page with a “tick” and the number of rows that have been affected by your query
- If there is an error in the query, it will be indicated

# Run SQL queries using the SQL editor

The screenshot displays the phpMyAdmin web interface. The top navigation bar shows the server as 'localhost:3306', the database as 'zkorkmaz', and the selected table as 'aircraft'. The main menu includes options like Browse, Structure, SQL, Search, Insert, Export, Import, Operations, Tracking, and Triggers. On the left sidebar, the database structure is shown with 'information\_schema', 'zkorkmaz', and 'aircraft' listed. The central area is titled 'Run SQL query/queries on table zkorkmaz.aircraft:' and contains a text box with the SQL query: `1 SELECT * FROM aircraft`. Below the query box are buttons for 'SELECT \*', 'SELECT', 'INSERT', 'UPDATE', 'DELETE', 'Clear', and 'Format'. To the right, a 'Columns' panel lists the table's fields: 'aid', 'aname', and 'crusingrange'. At the bottom, there are checkboxes for 'Bind parameters', 'Show this query here again', 'Retain query box', 'Rollback when finished', and 'Enable foreign key checks', along with a 'Go' button. A 'Console' tab is visible at the very bottom left.

phpMyAdmin

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

Browse Structure SQL Search Insert Export Import Operations Tracking Triggers

Run SQL query/queries on table zkorkmaz.aircraft:

```
1 SELECT * FROM aircraft
```

Columns

- aid
- aname
- crusingrange

SELECT \* SELECT INSERT UPDATE DELETE Clear Format

Get auto-saved query

☐ Bind parameters

Bookmark this SQL query:

[ Delimiter ; ] ☒ Show this query here again ☐ Retain query box ☐ Rollback when finished ☒ Enable foreign key checks

Console

# Exercise 4

- Question 1: Show the name of the flight with ID '1'
- Question 2: List all the information about all aircrafts in our fleet that can cruise over 5000 miles
- Question 3: List the name of the employees who has the certification for Airbus A340-300 aircraft
- Note: Table and column names are case-sensitive  
Note: Keep track of your location in the database



# Solutions..

- Question 1: Show the name of the flight with ID '1'
  - `select aircraft.aname from aircraft where aircraft.aid = 1`
- Question 2: List all the information about all aircrafts in our fleet that can cruise over 5000 miles
  - `select * from aircraft where aircraft.cruisingrange > 5000`
- Question 3: List the name of the employees who has the certification for Airbus A340-300 aircraft
  - `select e.ename from employees e, aircraft a, certified c  
where e.eid = c.eid and a.aid = c.aid and a.aname = "Airbus A340-300"`

# Exercise 5

- The schema for Exercise 5 is given below

Student(snum: integer, sname: string, major: string, level: string, age: integer)

Class(name: string, meets\_at: string, room: string, fid: integer)

Enrolled(snum: integer, cname: string)

Faculty(fid: integer, fname: string, deptid: integer)

The meaning of these relations is straightforward; for example, Enrolled has one record per student-class pair such that the student is enrolled in the class.

Write the following queries in SQL. No duplicates should be printed in any of the answers.

| snum      | sname              | major                  | standing | age |
|-----------|--------------------|------------------------|----------|-----|
| 51135593  | Maria White        | English                | SR       | 21  |
| 60839453  | Charles Harris     | Architecture           | SR       | 22  |
| 99354543  | Susan Martin       | Law                    | JR       | 20  |
| 112348546 | Joseph Thompson    | Computer Science       | SO       | 19  |
| 115987938 | Christopher Garcia | Computer Science       | JR       | 20  |
| 132977562 | Angela Martinez    | History                | SR       | 20  |
| 269734834 | Thomas Robinson    | Psychology             | SO       | 18  |
| 280158572 | Margaret Clark     | Animal Science         | FR       | 18  |
| 301221823 | Juan Rodriguez     | Psychology             | JR       | 20  |
| 318548912 | Dorothy Lewis      | Finance                | FR       | 18  |
| 320874981 | Daniel Lee         | Electrical Engineering | FR       | 17  |
| 322654189 | Lisa Walker        | Computer Science       | SO       | 17  |
| 348121549 | Paul Hall          | Computer Science       | JR       | 18  |
| 351565322 | Nancy Allen        | Accounting             | JR       | 19  |
| 451519864 | Mark Young         | Finance                | FR       | 18  |
| 455798411 | Luis Hernandez     | Electrical Engineering | FR       | 17  |
| 462156489 | Donald King        | Mechanical Engineering | SO       | 19  |
| 550156548 | George Wright      | Education              | SR       | 21  |

STUDENT

| fid       | fname            | deptid |
|-----------|------------------|--------|
| 11564812  | John Williams    | 68     |
| 90873519  | Elizabeth Taylor | 11     |
| 141582651 | Mary Johnson     | 20     |
| 142519864 | Ivana Teach      | 20     |
| 159542516 | William Moore    | 33     |
| 242518965 | James Smith      | 68     |
| 248965255 | Barbara Wilson   | 12     |
| 254099823 | Patricia Jones   | 68     |
| 287321212 | Michael Miller   | 12     |
| 356187925 | Robert Brown     | 12     |
| 486512566 | David Anderson   | 20     |
| 489221823 | Richard Jackson  | 33     |
| 489456622 | Linda Davis      | 20     |
| 548977562 | Ulysses Teach    | 20     |
| 619023588 | Jennifer Thomas  | 11     |

FACULTY

| cname                           | meets_at         | room     | fid       |
|---------------------------------|------------------|----------|-----------|
| Air Quality Engineering         | TuTh 10:30-11:45 | R15      | 11564812  |
| American Political Parties      | TuTh 2-3:15      | 20 AVW   | 619023588 |
| Archaeology of the Incas        | MWF 3-4:15       | R128     | 248965255 |
| Aviation Accident Investigation | TuTh 1-2:50      | Q3       | 11564812  |
| Communication Networks          | MW 9:30-10:45    | 20 AVW   | 141582651 |
| Dairy Herd Management           | TuTh 12:30-1:45  | R128     | 356187925 |
| Data Structures                 | MWF 10           | R128     | 489456522 |
| Database Systems                | MWF 12:30-1:45   | 1320 DCL | 142519864 |
| Intoduction to Math             | TuTh 8-9:30      | R128     | 489221823 |
| Introductory Latin              | MWF 3-4:15       | R12      | 248965255 |
| Marketing Research              | MW 10-11:15      | 1320 DCL | 489221823 |
| Multivariate Analysis           | TuTh 2-3:15      | R15      | 90873519  |
| Operating System Design         | TuTh 12-1:20     | 20 AVW   | 489456522 |
| Optical Electronics             | TuTh 12:30-1:45  | R15      | 254099823 |
| Orbital Mechanics               | MWF 8            | 1320 DCL | 11564812  |
| Organic Chemistry               | TuTh 12:30-1:45  | R12      | 489221823 |
| Patent Law                      | F 1-2:50         | R128     | 90873519  |
| Perception                      | MTuWTh 3         | Q3       | 489221823 |

CLASS

| snum      | cname                      |
|-----------|----------------------------|
| 99354543  | Patent Law                 |
| 112348546 | Database Systems           |
| 112348546 | Operating System Design    |
| 115987938 | Database Systems           |
| 115987938 | Operating System Design    |
| 301221823 | American Political Parties |
| 301221823 | Perception                 |
| 301221823 | Social Cognition           |
| 322654189 | Database Systems           |
| 322654189 | Operating System Design    |
| 348121549 | Database Systems           |
| 455798411 | Operating System Design    |
| 455798411 | Optical Electronics        |
| 552455318 | Communication Networks     |
| 552455318 | Database Systems           |
| 552455318 | Operating System Design    |
| 556784565 | Air Quality Engineering    |
| 567354612 | Data Structures            |

ENROLLED

# Exercise 5

Find the names of all Juniors (standing = JR) who are enrolled in a class taught by Ivana Teach

names and ids of students -> sname, snum -> Table STUDENT

standing of students -> standing -> Table STUDENT

students enrolled in a class -> snum, cname -> Table ENROLLED

name of teacher -> fname -> Table FACULTY

name of class taught by Ivana -> cname -> Table CLASS

# Steps to get to Answer for Exercise 5

- To get which junior students are taught by Ivana Teach, we need to find which classes (names) are taught by Ivana Teach
- To get which classes are taught by Ivana Teach, we need her faculty id (fid) as the Class table has only faculty id to identify the faculty who teaches a class (not the faculty's name)
  - Use the Faculty table to find Ivana Teach's faculty id (*Ivana Teach's fid is 142519864*)
  - Look up the Class table to find the classes she takes (*Ivana Teach teaches only Database Systems*)

# Steps to get to Answer for Exercise 5

- Look up the Enrolled table to find the students that take the class
  - 5 students take the Database Systems class taught by Ivana Teach (*their snums are 112348546, 115987938, 322654189, 348121549, 552455318*)
- From the Student table, find out which of these 5 students are Junior.
  - Out of these 5 students, 2 of them with snums 115987938 and 348121549 are junior
- From the Student table, find their names
  - The of the junior students who are enrolled in classes taught by Ivana Teach are Christopher Garcia and Paul Hall

# Answer 5

```
select distinct s.sname
from student s, class c, enrolled e, faculty f
where s.snum = e.snum
      and e.cname = c.cname
      and c.fid = f.fid
      and f.fname = 'Ivana Teach'
      and s.standing = 'JR'
```

✓ Showing rows 0 - 1 (2 total, Query took 0.0012 seconds.)

```
select distinct s.sname from student s, class c, enrolled e, fac
.....
```

☐ Show all | Number of rows: 100 ▼ Filter rows:

+ Options

| sname              |
|--------------------|
| Christopher Garcia |
| Paul Hall          |

There are 2  
junior students  
taught by Ivana



## Other possible query for 5:

```
select distinct s.sname  
from student s JOIN class c JOIN enrolled e JOIN faculty f  
ON s.snum = e.snum and e.cname = c.name and c.fid = f.fid  
where f.fname = 'Ivana Teach' and s.standing = 'JR'
```

# Exercise 6

- The schema for Exercise 5.2 is given below

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

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

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

The Catalog relation lists the prices charged for parts by Suppliers. Write the following queries in SQL:

| sid | sname                 | address                                  |
|-----|-----------------------|--|
| 1   | Acme Widget Suppliers | 1 Grub St., Potemkin Village, IL 61801   |
| 2   | Big Red Tool and Die  | 4 My Way, Bermuda Shorts, OR 90305       |
| 3   | Perfunctory Parts     | 99999 Short Pier, Terra Del Fuego, TX 41 |
| 4   | Alien Aircraft Inc.   | 2 Groom Lake, Rachel NV 51912            |

## SUPPLIERS

| pid | pname                             | color       |
|-----|-----------------------------------|-------------|
| 1   | Left Handed Bacon Stretcher Cover | Red         |
| 2   | Smoke Shifter End                 | Black       |
| 3   | Acme Widget Washer                | Red         |
| 4   | Acme Widget Washer                | Silver      |
| 5   | I Brake for Crop Circles Sticker  | Translucent |
| 6   | Anti-Gravity Turbine Generator    | Cyan        |
| 7   | Anti-Gravity Turbine Generator    | Magenta     |
| 8   | Fire Hydrant Cap                  | Red         |
| 9   | 7 Segment Display                 | Green       |

## PARTS

| sid | pid | cost       |
|-----|-----|------------|
| 1   | 3   | 0.50       |
| 1   | 4   | 0.50       |
| 1   | 8   | 11.70      |
| 2   | 1   | 16.50      |
| 2   | 3   | 0.55       |
| 2   | 8   | 7.95       |
| 3   | 8   | 12.50      |
| 3   | 9   | 1.00       |
| 4   | 5   | 2.20       |
| 4   | 6   | 1247548.23 |
| 4   | 7   | 1247548.23 |

## CATALOG

Note: The foreign keys (sid and pid) of catalog are displayed in a different color. The data of the columns that reference another table are displayed in blue in phpMyAdmin

## Exercise 6

Find the *pnames* of parts for which there is some supplier.

Part names -> pname, pid -> Table PARTS

Supplier for particular parts -> pid -> Table CATALOG

The Catalog Table lists all parts who have suppliers. We want all the pids from this table and find their names from the Table Parts

Note: different colors do not matter

## Exercise 6

select distinct p.pname  
from parts p, catalog c  
where p.pid = c.pid

Out of the 7  
distinct parts listed  
in the Table Parts, 6  
parts have  
suppliers



Showing rows 0 - 5 (6 total, Qu

```
SELECT DISTINCT P.pname FROM parts
```



Show all

Number of rows

+ Options

pname

Left Handed Bacon Stretcher Cover

Acme Widget Washer

I Brake for Crop Circles Sticker

Anti-Gravity Turbine Generator

Fire Hydrant Cap

7 Segment Display

## Other possible queries for Exercise 6

```
select distinct p.pname  
from parts p join catalog c on p.pid = c.pid
```

```
select distinct p.pname from parts p NATURAL JOIN catalog c
```

# To understand Answer 6.1

- Just joined the two tables Parts and Catalog
- Can see that the same part with same or different colour are supplied by many suppliers
- We want only distinct names i.e. parts (irrespective of their color) that are supplied by a supplier

✓ Showing rows 0 - 10 (11 total, Query took 0.0007 seconds.) [pid: 1... - 9...]

```
SELECT * FROM parts P, catalog C WHERE P.pid = C.pid ORDER BY `P`.`pid` ASC
```

☐ Show all | Number of rows: 100 ▾ Filter rows: Search this table

+ Options

| pid | ▲ 1 | pname                             | color       | sid | pid | cost       |
|-----|-----|-----------------------------------|-------------|-----|-----|------------|
| 1   |     | Left Handed Bacon Stretcher Cover | Red         | 2   | 1   | 16.50      |
| 3   |     | Acme Widget Washer                | Red         | 1   | 3   | 0.50       |
| 3   |     | Acme Widget Washer                | Red         | 2   | 3   | 0.55       |
| 4   |     | Acme Widget Washer                | Silver      | 1   | 4   | 0.50       |
| 5   |     | I Brake for Crop Circles Sticker  | Translucent | 4   | 5   | 2.20       |
| 6   |     | Anti-Gravity Turbine Generator    | Cyan        | 4   | 6   | 1247548.23 |
| 7   |     | Anti-Gravity Turbine Generator    | Magenta     | 4   | 7   | 1247548.23 |
| 8   |     | Fire Hydrant Cap                  | Red         | 2   | 8   | 7.95       |
| 8   |     | Fire Hydrant Cap                  | Red         | 1   | 8   | 11.70      |
| 8   |     | Fire Hydrant Cap                  | Red         | 3   | 8   | 12.50      |
| 9   |     | 7 Segment Display                 | Green       | 3   | 9   | 1.00       |

# PHP and MySQLi

- Our PHP application needs to communicate with a database server
- We need to perform activities such as:
  - 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
- To perform these tasks, we will need –
  - HTML to create web forms
  - PHP with MySQLi API to perform all database-related functions



# 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
- It is available in both procedural and object-oriented flavors
  - We will use the object-oriented API and so will need to instantiate classes and then call methods on the resulting objects
- Documentation: <http://php.net/manual/en/book.mysqli.php>

# MySQLi

- Interaction with a database using MySQLi can be categorized into 4 main stages:
  - A. Connecting to a database
  - B. Firing a query
  - C. Collecting its results
  - D. Closing the connection to the database
- These major stages contain 8 steps in total
- Generally, all these 8 steps need to be performed to complete a transaction/interaction with the database using a web application developed in PHP

# Stage A: Connect to a database

- Step 1: Define connection parameters
- Step 2: Define connection variable (*\$mysqli*)

```
<?php
// Function to obtain mysqli connection.
function get_mysqli_conn()
{
    $dbhost = 'localhost';
    $dbuser = 'username';
    $dbpassword = 'password';
    $dbname = 'username';
    $mysqli = new mysqli($dbhost, $dbuser, $dbpassword, $dbname);
    if ($mysqli->connect_errno)
    {
        echo 'Failed to connect to MySQL: (' . $mysqli->connect_errno . ') ' . $mysqli->connect_error;
    }
    return $mysqli;
}
?>
```

## Stage B: Fire a query

- Step 3: Define the SQL Query statement as a variable (*\$sql*)
- The MySQL server supports using anonymous positional placeholder with ?
- Use ? for parameters (later filled in code or user-input)
- Write the SQL query you want to run here (select, update, insert, delete etc.)

```
// (3) SQL statement
$sql = "SELECT a.aid, a.aname "
      . "FROM aircraft a "
      . "WHERE a.aname LIKE CONCAT(?, '%')";
```

## Stage B: Fire a query (continued)

- Step 4a: Prepare statement. Do not need to pass parameters now
- Step 4b: Bind PHP variables to MySQL parameters
- Should be done for all parameters that come from user-inputs
- This sanitizes strings that have to replace ?, thus, combating SQL injection

```
// (4a) Prepared statement, stage 1: prepare
$stmt = $mysqli->prepare($sql);
```

```
// (4b) Bind a PHP variable, $search as a string parameter
$search = 'Boeing';
```

```
// "i" for integer, "d" for double, "s" for string, "b" for blob
$stmt->bind_param('s', $search);
```

## Stage B: Fire a query (continued)

- **Step 5:** Execute the SQL query statement (\$sql)

```
// (5) Execute prepared statement  
$stmt->execute();
```

# Stage C: Collect results of a query

- **Step 6:** Bind selected columns to PHP variables
- **Step 7:** Fetch values by iterating through the executed statement's results

```
// (6) Bind selected columns to PHP variables
$stmt->bind_result($aircraft_id, $aircraft_name);

// (7) fetch values
// <ul> is unordered list
echo '<ul>';
while ($stmt->fetch())
{
    // printf is print format, <li> is list item
    printf ('<li>%s (%s)</li>', $aircraft_name, $aircraft_id);
}
echo '</ul>';
```

%s -> format as string

For every list item, take 1<sup>st</sup> variable and put in 1<sup>st</sup> %s, Take the 2<sup>nd</sup> variable and put in 2<sup>nd</sup> %s which is in the brackets

# Stage D: Close the connection

- **Step 8:** Close the SQL query statement (\$sql). Close the connection (\$mysqli) if there are no further transactions in this PHP file

```
// (9) close statement and mysqli connection  
$stmt->close();  
$mysqli->close();
```

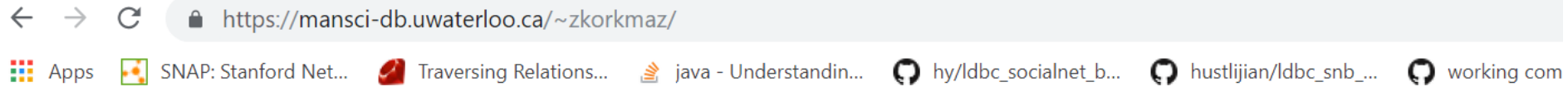
- This completes one regular transaction
- Change the query and follow these steps to perform any action
- Note: For clarity, error-checking has been omitted in these steps; but it is highly recommended. The documentation found in <http://php.net/manual/en/book.mysqli.php> has examples



## Exercise 7

- Download the file “php\_sql.php” from Learn onto your local machine
- The file contains all the code in sections 1 to 4 (an entire transaction)
- Change the credentials for the database to yours
- Transfer the file to your public\_html directory
- Rename the file to index.php
- Go to the link [mansci-db.uwaterloo.ca/~your\\_username](https://mansci-db.uwaterloo.ca/~your_username)

# Exercise 7



- Localhost via UNIX socket
- Boeing 747-400 (1)
  - Boeing 737-800 (2)
  - Boeing 757-300 (10)
  - Boeing 777-300 (11)
  - Boeing 767-400ER (12)
  - Boeing 727 (15)
- If a successful connection to the database was established, the message “Localhost via UNIX socket” is displayed. Otherwise, an error message is displayed
  - The page lists the names (with IDS) of all aircrafts whose names start with “Boeing” in the aircraft table that we created in our database in the last lab