

Introduction and Overview

G51DBI – Databases and Interfaces
Yorgos Tzimiropoulos and Tim Brailsford
yorgos.tzimiropoulos@nottingham.ac.uk

This Lecture

- About us and how to contact us
- Module overview
- Module material
- Reference books
- Lectures and assessment

2

Who are we?

- Yorgos (Tzimiropoulos)
- Assistant Professor in Computer Science
- www.cs.nott.ac.uk/~pszyt/
- yorgos.tzimiropoulos@nottingham.ac.uk
- MSc, PhD, post-doc at Imperial College
- Research on Computer Vision and Deep Learning
- Teach Databases and Interfaces and Computer Vision

3

How to Contact Yorgos

- Before / After lectures
- In the Lab
- By email
Please put "[G51DBI] Query" as subject
- At my office (B81) every Thursday (09:00-10:00)
Please send me an email before coming

4

Who are we?

- Tim Brailsford
- Associate Professor in Computer Science
- <https://www.nottingham.ac.uk/computerscience/people/tim.brailsford>
- tim.brailsford@nottingham.ac.uk
- Teach Databases and Interfaces
- Main interests: hypertext, multimedia and electronic publishing, lot and lots of web scripting

5

How to Contact Tim

- Before / After lectures
- In the Lab
- By email
Please put "[G51DBI] Query" as subject
- At my office (C5) every Tuesday (12:00-13:00) or Thursday (13:30-14:30)
Please send me an email before coming

6

Lectures and Assessment

- Lectures
 - Tue 15:00-16:00 @ JC-EXCHGE- LT2
 - Thu 15:00-17:00 @ JC-EXCHGE- LT3
- Lab sessions
 - Thu 11:00-13:00 @ JC-COMPSCI-A32
(From week 2, i.e. 5th Feb 2018)
 - **Practice is important** for understanding!
 - Coursework 1 is basically (part of) lab work !!
- Assessment
 - 1 Exam, worth 50%, and 2 Courseworks, worth 50%
 - 1st coursework on Databases, part of the lab work (15%)
 - 2nd coursework on Web programming (35%)

7

Module Overview

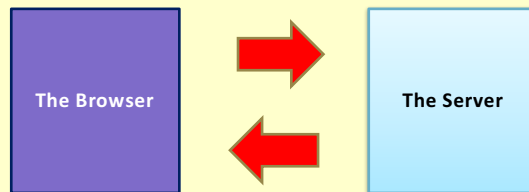
- G51DBI started in 2016
- It's the merging of 2 previously taught first year modules: G51DBS (Databases) and G51WPS (Web programming)
- G51DBS: back-end (server-side programming), G51WPS: front-end (client-side programming)
- We decided to put these together and teach how to build an **end-to-end** system

8

Module Overview

The **Client - Server** paradigm: a widely used form of computer communication

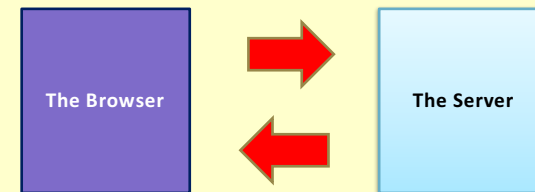
- A server provides specific services (e.g. web-pages, access to a database)
- Server applications wait passively for contact from clients
- Client applications initiate contact with the server
- Server responds and then information flows in both directions
- More than one clients interact with each server



9

Module Overview

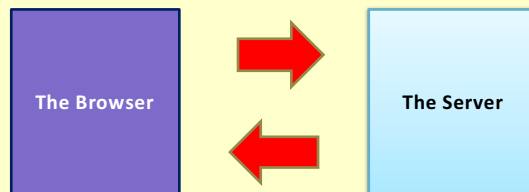
- Web-browsers are clients (Firefox, Chrome, Safari, Internet Explorer)
- Allow user to browse resources available on a server
- These could be existing (e.g. a webpage) or dynamically built documents (e.g. a webpage or the result of a database query)
- E.g. The browser sends a request to the server asking for a web-page such as index.htm, the server responds by sending the requested web-page and/or other data and the browser displays the webpage



10

Module Overview

- Web-servers provide responses to browser requests
- Common servers: Apache, Internet Information Server (IIS), Google Web Server
- They run as background processes in the operating system
- All communications between browsers and servers use Hypertext Transfer Protocol (HTTP). E.g.: browser requests a URL and servers responds with HTML data



11

Module Content

In a nutshell learn SQL, HTML, CSS, Javascript and PHP in one module!

On the server side

- We will learn about Database Management Systems (MySQL) and learn SQL (language for querying Relational Databases)
- We will also learn PHP for server-side programming (a server-side scripting language, great for form processing and database access though the Web)

12

Module Content

On the client side

- We will learn HTML (language describing the general form and layout of web documents)
- We will learn CSS (language for controlling and changing the presentation of HTML documents)
- We also learn JavaScript (HTML-embedded scripting language that provides a way to access the elements of HTML documents and dynamically change them)

13

Module Topics

- Main Topics
 - Intro, module overview, intro to (relational) databases and Relational Algebra (Week 1)
 - Database design and SQL I (Week 2)
 - SQL II (Week 3)
 - SQL III (Week 4)
 - HTML/CSS (Week 5)
 - JavaScript (Week 6)
 - PHP (Week 7)
 - PHP (Week 8)
 - Normalisation (Week 9)
 - Advanced topics in and Interfaces (Week 10)

14

Module Topics – Databases

- Databases and relational databases
 - What is a database
 - What is it useful for?
 - What is a relation?
 - What is an attribute?
 - What is a tuple?
 - More in the next lecture

University		
uName	County	Enr
Notts	Nott/shire	18,000
Cam	Cam/shire	22,000
UCL	Greater Lon	20,000

15

Module Topics – Databases

- Relational Algebra
 - The “maths” of relational databases
 - Can be used to express queries in a mathematical form
 - Various relational algebra operators (e.g. select, project)
 - E.g. “Find all universities with > 20000 students”

$\pi_{uName}(\sigma_{Enr > 20000}(University))$

University		
uName	County	Enr
Notts	Nott/shire	18,000
Cam	Cam/shire	22,000
UCL	Greater Lon	20,000

16

Module Topics – Databases

- Database design
 - Introduction to database design
 - Entities and Attributes, Relationships, E/R Diagrams
 - E.g.

"A university consists of a number of departments. Each department offers several courses. A number of modules make up each course. Students enrol in a particular course and take modules towards the completion of that course. Each module is taught by a lecturer from the appropriate department (several lecturers work in the same department), and each lecturer tutors a group of students. A lecturer can teach more than one module but can work only in one department."

- Entities – Department, Course, Module, Student, Lecturer
- Relationships – Offers, Make Up, Enrol, Take, Taught By

17

Module Topics – Databases

- SQL I
 - Introduction to SQL and to Database Management Systems (i.e. MySQL)
 - Create (Databases and Tables)
 - Data types
 - Primary and Foreign keys
 - Manipulating tables (e.g. INSERT, UPDATE, DELETE, DROP, ALTER)
 - E.g:

```
CREATE TABLE Student (  
    sID INT PRIMARY KEY,  
    sName VARCHAR(50) NOT NULL,  
    sAddress VARCHAR(255) ,  
    sYear INT DEFAULT 1);
```

18

Module Topics – Databases

- SQL II
 - Basic and advanced SELECT
 - Dealing with duplicates
 - Selecting from multiple tables
 - Sub-queries
 - Eg: "Find all universities with > 20000 students"

$\pi_{uName}(\sigma_{enr > 20000}(University))$

```
Select uName  
From University  
Where enr>20000
```

19

Module Topics – Databases

- SQL III
 - Various advanced SQL features
 - Joins (how to combine information from multiple tables)
 - Sorting query results
 - Aggregate functions (e.g. max, min, sum, average, count)
 - Grouping
 - Missing information

20

Module Topics – Databases

- Database Normalisation
 - How to design a database (i.e. define the tables and the columns for each table) so that data redundancy is minimised.
 - Approach: start by a “mega”- table containing all data and then start splitting

21

Module Topics – Databases

- E.g.

Module	Dept	Lecturer	Text
M1	D1	L1	T1
M1	D1	L1	T2
M2	D1	L1	T1
M2	D1	L1	T3
M3	D1	L2	T4



Module	Text
M1	T1
M1	T2
M2	T1
M2	T3
M3	T4

Module	Dept	Lecturer
M1	D1	L1
M2	D1	L1
M3	D1	L2

22

Module Topics – Interfaces

- HTML/CSS
 - Language describing the general form and layout of documents
 - All basic and many advanced HTML elements and features
 - E.g. text, images, lists, links
 - Tables, forms, select menus, check boxes, radio buttons
 - All relevant CSS features and style sheets for displaying HTML documents (e.g. colour, font, positioning, align, borders, margins, padding)

23

Module Topics – Interfaces

- HTML/CSS

E.g. a simple form:

```
<html> <body>
<form
action="www.cs.nott.ac.uk/~pszyt/welcome.p
hp" method="post">
Name: <input type="text" name="name"><br>
E-mail: <input type="text"
name="email"><br>
<input type="submit">
</form>
</body> </html>
```

Name:

E-mail:

24

Module Topics – Interfaces

- JavaScript
 - Basic and many advanced JavaScript features for event programming
 - JS very well known for allowing interaction with user
 - Dialogue boxes (Alert, Confirm, Prompt)
 - Events and event handling (e.g. click, mouse over, mouse move, submit)
 - Dynamic HTML (change attributes, content and style of an html doc while the doc is being displayed through JS)

25

Module Topics – Interfaces

E.g.:

```
<html> <body>
    <p>Click the button to display the date.</p>
    <button onclick="displayDate()">The time
is?</button>
    <p id="demo"></p>
    <script>
        function displayDate() {
            document.getElementById("demo").innerHTML =
Date();
        }
    </script>
</body> </html>
```

Click the button to display the date.

The time is?

Click the button to display the date.

The time is?

Mon Jan 25 2016 08:30:49 GMT+0000

Module Topics – Interfaces

- PHP
 - Like JS but on the server side, used for form handling, file processing and database access (syntax reminds C:)
 - Output of PHP is HTML sent to browser
 - Basic features (e.g. loops, functions, printing)
 - Advanced features (e.g. form handling, files, and database access)

27

Module Topics – Interfaces

- PHP
 - E.g. PHP handling simple form
- ```
<html> <body>
Welcome <?php echo $_POST["name"]; ?>

Your email address is: <?php echo
$_POST["email"]; ?>
</body> </html>
```

Output could be:

Welcome John

Your email address is john.doe@example.com

28

## Module Topics – Interfaces

- Advanced topics in G51DBI
  - AJAX
  - HTML5 (audio, video)
  - Webpage design
  - XML

29

## Learning and Feedback

- Lecture Materials
  - Cover the theory in detail with some examples
  - As self-contained as possible
- Lectures
  - All theory will be explained in the lectures through **live code demonstrations** where possible
  - Will include examples that might not appear in the slides so please do attend!
- Labs
  - **Practice is very important**
  - Great opportunity to ask questions about the taught materials and the coursework
  - Great opportunity to get personal feedback about your progress, understanding of various topics etc.
  - Part of the lab work is Coursework 1

30

## Learning and Feedback

- Coursework 1
  - Will be on databases. Already released.
  - During the labs you will apply the concepts taught in the lectures by building your own database and writing your own queries.
  - Labs of weeks 4 and 5 will be assessed
  - 15%
  - Submission via Moodle (deadline Mon, March 5<sup>th</sup>)
  - In lab assessment (Thu, March 8<sup>th</sup>)
- Coursework 2
  - Already released
  - Aims to cover basic and advanced HTML/CSS, JS, PHP covered in weeks 5-8
  - 35%
  - Submission via Moodle (deadline Monday, April 30<sup>th</sup>)
  - In lab assessment (Thu, May 3<sup>rd</sup>)
- Feedback
  - Feedback and final mark for coursework 1 and 2 within 2 weeks

31

## Module Material

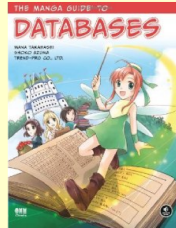
- Moodle page
  - [moodle.nottingham.ac.uk](http://moodle.nottingham.ac.uk)
  - Look for G51DBI
  - Slides for every session will be available
  - Lab session files will be available (sometimes together with solutions)

32



## Books & Resources

- Databases
  - Database Systems - A Practical Approach to Design, Implementation, and Management, Connolly & Begg (*source of some diagrams*)
  - A simple introduction: The Manga Guide To Databases, Mana Takahashi and Shoko Azuma (few copies in the Library now)
- Web programming
  - Programming the World Wide Web, Robert W. Sebesta
- Both
  - [www.w3schools.com](http://www.w3schools.com)
  - [www.stackoverflow.com](http://www.stackoverflow.com)



33

## Next Lecture (Thursday)

- Intro to Databases
- The Relational Model
- Intro to Relational Algebra
  
- Any questions??

34