Diploma in Web Development – Part I

Lesson 4

What is the Back-End?

Presented by:
Julian Quirke
Web Development Educator

Lesson 3 Recap

What is the Front-End?

- HTML: A Container for Content
- CSS: The Language of Web Design
- JavaScript: For Dynamic Interactivity

- Summary
- Q&A

Today's Lesson

What is the Back-End?

- Back-End Languages
- Persistent Storage
- Introduction to SQL

- Summary
- Q&A

Let's Begin!

Back-End Languages

Back-End Languages

are the programming languages that execute on the server-side of a web application

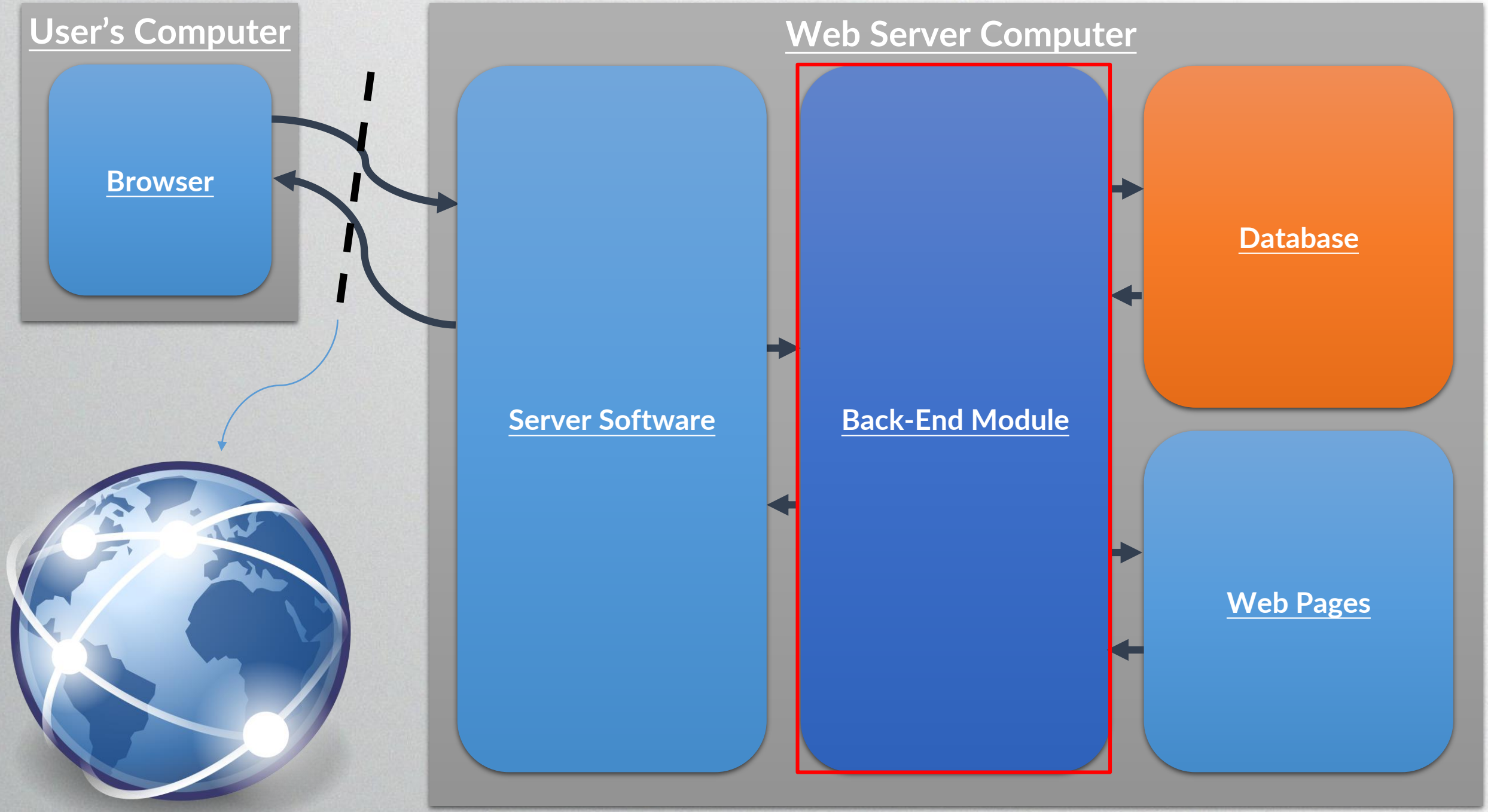
Back-End Languages

Back-End Languages

- Act as the **logic** of a web application
- Responsible for making appropriate database requests
- Can conditionally output HTML to a document before sending

Back-End Languages

- Many server-side programming language options
  - *ASP*
  - *PHP*
  - *Ruby*
- Act as the **logic** of a web application


User's Computer
Browser
Web Server Computer
Server Software
Back-End Module
Database
Web Pages

Back-End Languages

When a Server Receives a Request

1. Data required for long term storage is stored on the database
2. Static templates are retrieved from file system
3. Required content is retrieved from database
4. Content is inserted into template

Back-End Languages

PHP

- **P**HP **H**yperText **P**reprocessor
- Purpose-built as a back-end web application language
- Can be written directly into a HTML template document

Back-End Languages

PHP

- Object-Oriented Language
- Scripting Language
- Turing-Complete

Back-End Languages

PHP

- Declare and call identifiers
- Variables, functions and other objects
- Control systems

Back-End Languages

shawacademy

PHP

```
<html>
<head>
    <title>Lesson 8 - PHP</title>
    <link rel="icon" href="favicon.ico">
</head>
<body>

    <form action="process.php" method="post">
            Name: <input type="text" name="name"><br>
            E-mail: <input type="text" name="email"><br>
        <input type="submit">
    </form>

    <?php
        #Variables are declared using a '$' sign
        $name = "Gary";
        $age = 25;

        #PHP writes to a web page using 'echo'
        echo $name;
        echo "some normal text";
        echo "<h2>Hello from the web server</h2>";
        #Calculations and Comparisons work
        echo 5+2;
        if("" == ""){
            echo "FALSE";
        }

        #Functions exist in PHP but require AJAX to be used from client side
        function doSomething(){
            echo "I have done something";
        }

        #Arrays are declared using the array function
        $arr = array("hello", "goodbye", "Good Day");

        #Loops are the same with dollar signs
        for($i = 0; $i < 3; $i++){
            echo "\n\t<p>The greeting is: '" , $arr[$i] , "' is'nt that nice!</p>";
        }

        #Calling the function doSomething()
        doSomething();
    ?>

</body>
</html>
```

```
<html>
<head>
    <title>Lesson 8 - PHP</title>
    <link rel="icon" href="favicon.ico">
</head>
<body>

    <form action="process.php" method="post">
            Name: <input type="text" name="name"><br>
            E-mail: <input type="text" name="email"><br>
        <input type="submit">
    </form>

    <?php
        #Variables are declared using a '$' sign
        $name = "Gary";
        $age = 25;

        #PHP writes to a web page using 'echo'
        echo $name;
        echo "some normal text";
        echo "<h2>Hello from the web server</h2>";
        #Calculations and Comparisons work
        echo 5+2;
        if("" == ""){
            echo "FALSE";
        }

        #Functions exist in PHP but require AJAX to be used from client side
        function doSomething(){
            echo "I have done something";
        }

        #Arrays are declared using the array function
        $arr = array("hello", "goodbye", "Good Day");

        #Loops are the same with dollar signs
        for($i = 0; $i < 3; $i++){
            echo "\n\t<p>The greeting is: '" , $arr[$i] , "' is'nt that nice!</p>";
        }

        #Calling the function doSomething()
        doSomething();
    ?>

</body>
</html>
```

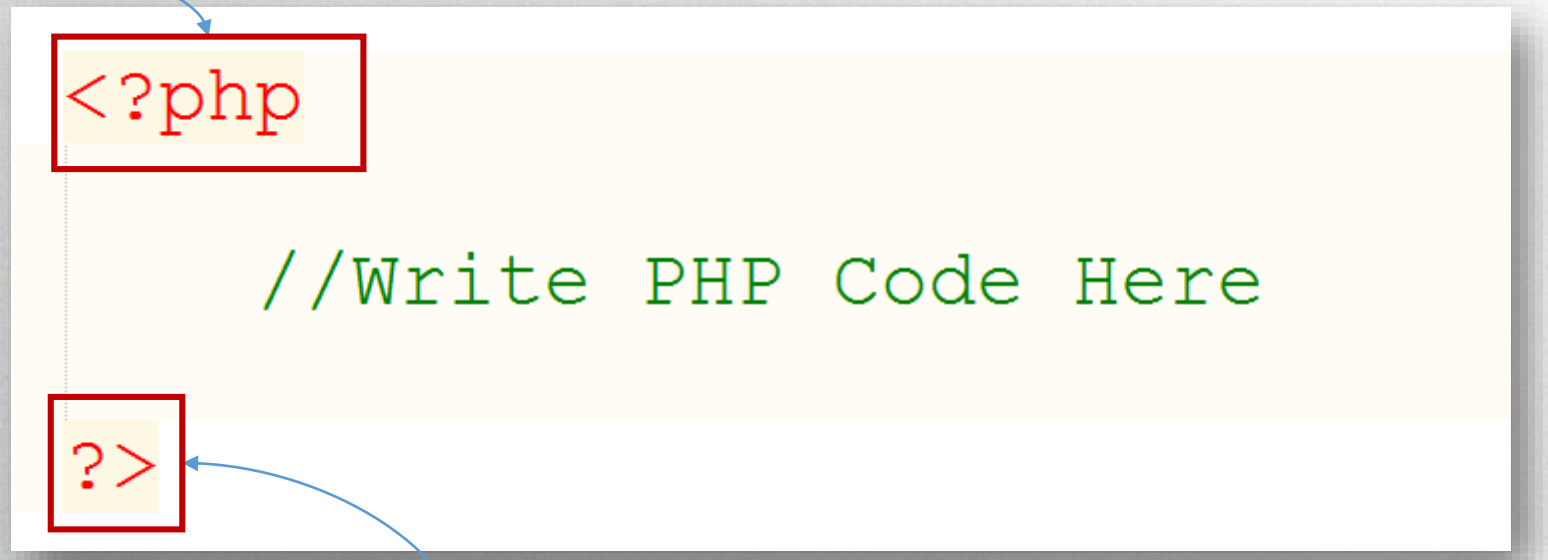
index.html

index.php

Back-End Languages

PHP

Start writing PHP

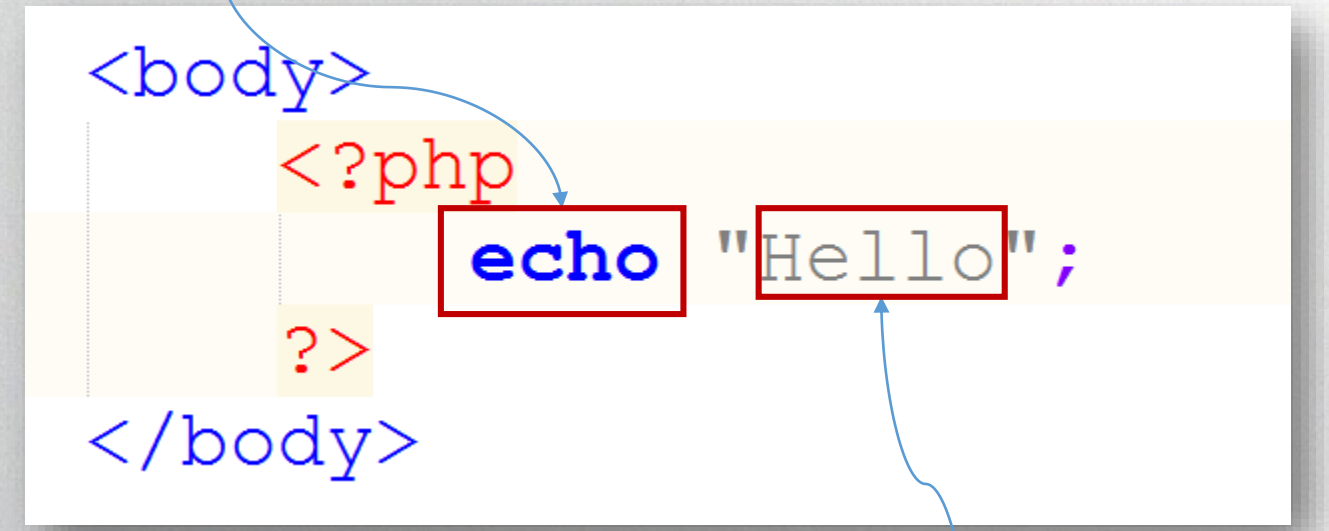


Finish writing PHP

Back-End Languages

PHP

Inserts directly into HTML Document



Text to be inserted

Back-End Languages

PHP

```
<body>
    Hello
</body>
```

Resultant HTML sent to a user

Persistent Data Storage

Julian Quirke
Web Development Educator

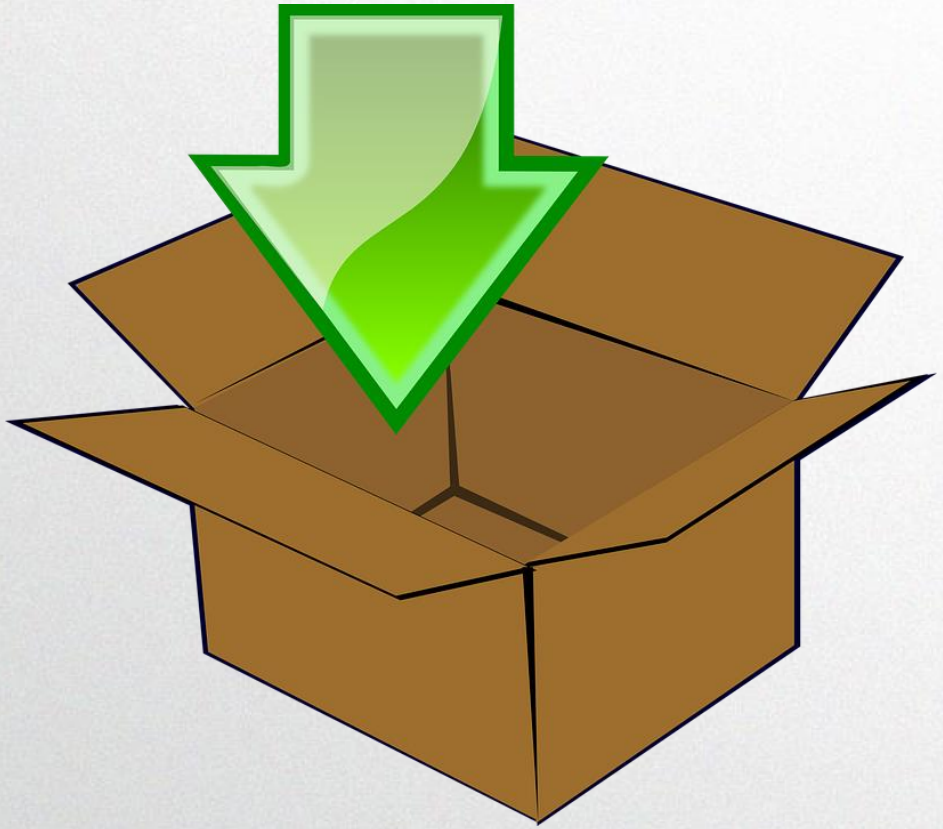
Persistent Data Storage

is the long-term storage of information for a software or web application, even if there is loss of power

Persistent Data Storage

Why do we need persistent storage?

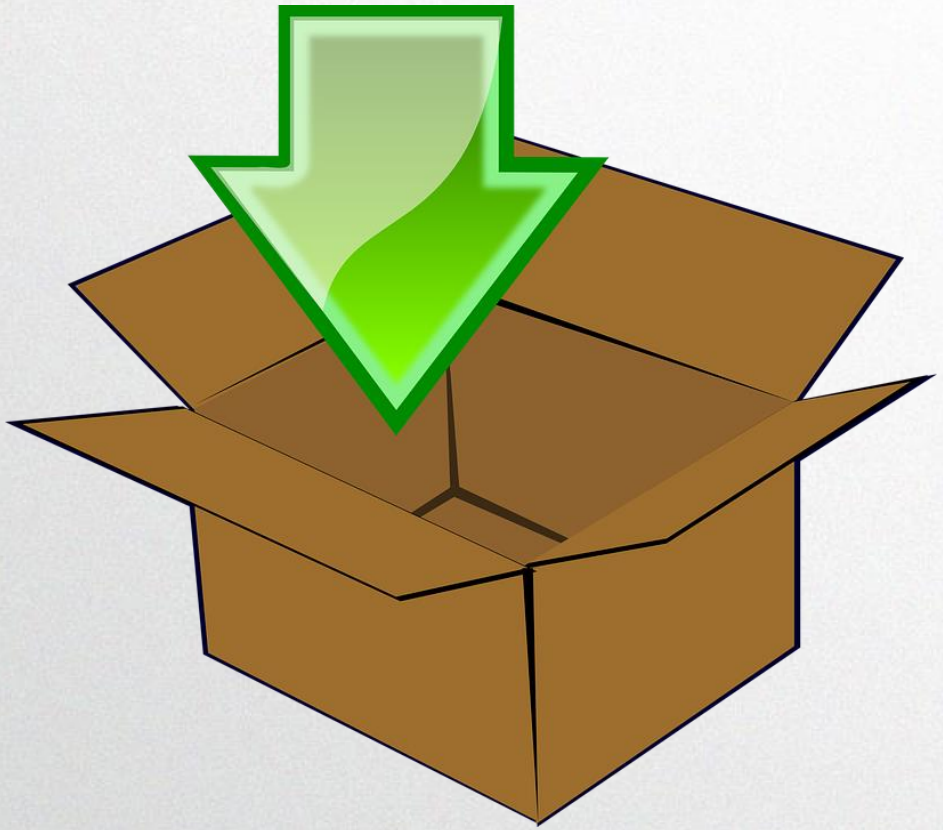
- Store all content related to our web application
- Organised model of data
- Sits **outside** the logic (the back-end code) of the application



Persistent Data Storage

What can we do with persistent storage?

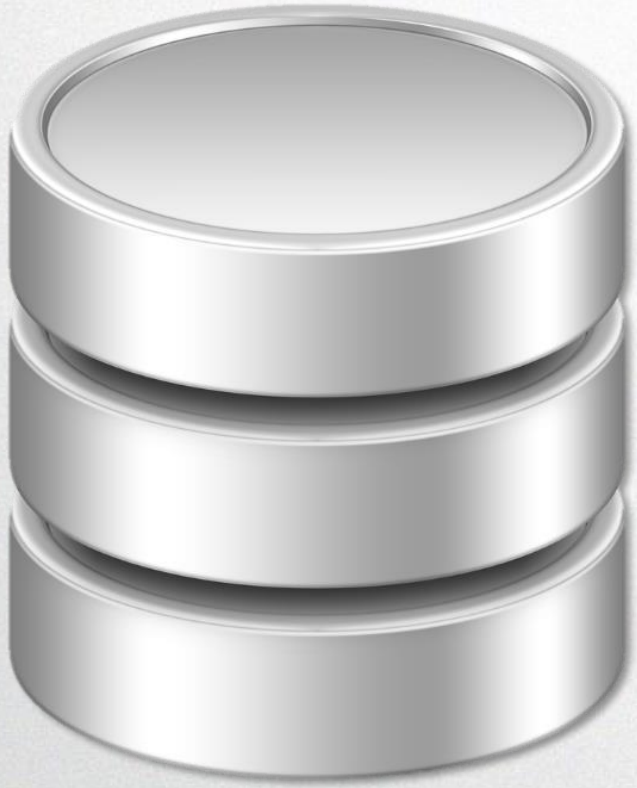
- Add login functionality to a web application
- Store user-generated content for use in a user-driven application
  - e.g. chatbox
- Organise site content in a logical manner



Persistent Data Storage

Persistent Data Storage Options

- **Database! - server-side:**
  - SQL (relational) databases
    - e.g. MySQL
  - NoSQL
    - e.g. MongoDB
- (Local Storage – client-side)



Persistent Data Storage

Database

Persistent Data Storage

Database

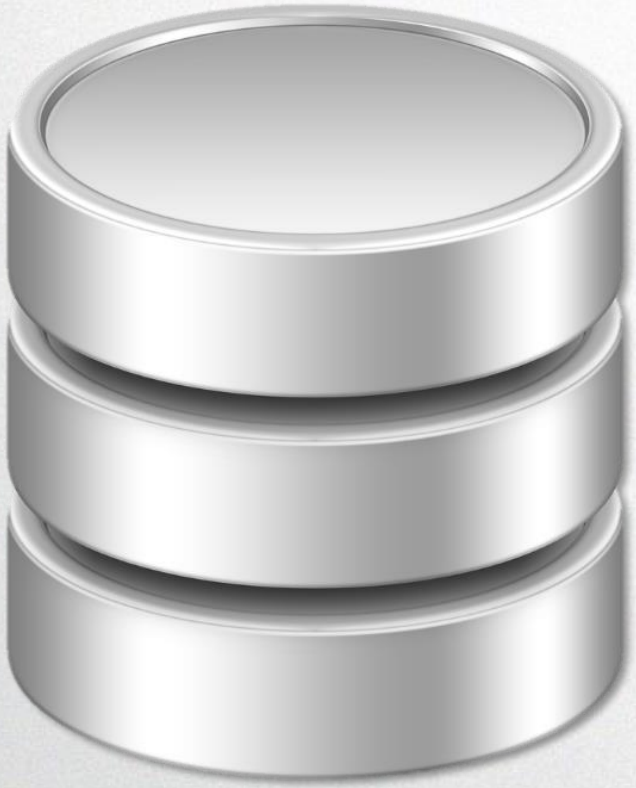
is a **structured set of data** held in a computer,
especially one that is accessible

Persistent Data Storage

Databases

Originally:

- symbol separated values in a text file
- Still sees use today (.csv files)
- (Not very efficient for large sets of data)

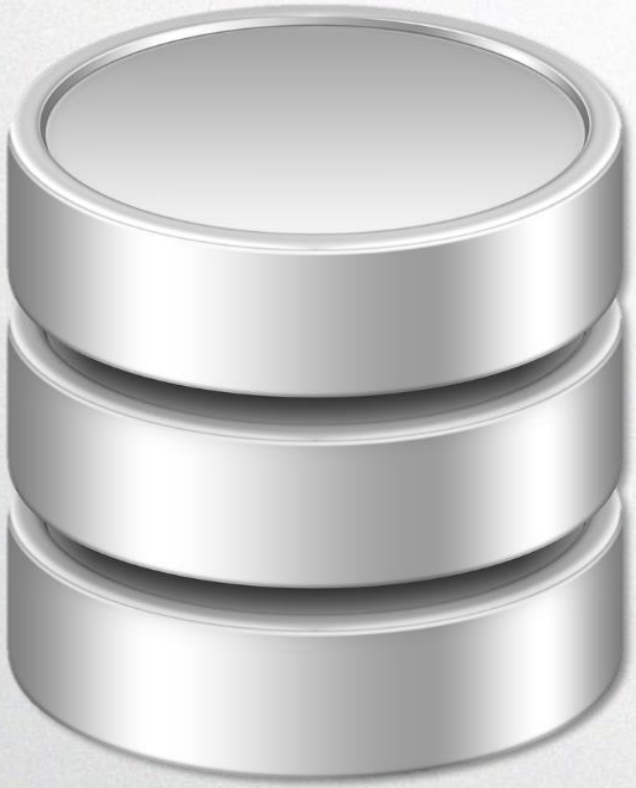


Persistent Data Storage

Relational Databases

Relational Database (RDBMS):

- Organised information
  - Efficient lookup of records
  - Data analysis
- Easy to store and retrieve information
  - SQL



Persistent Data Storage

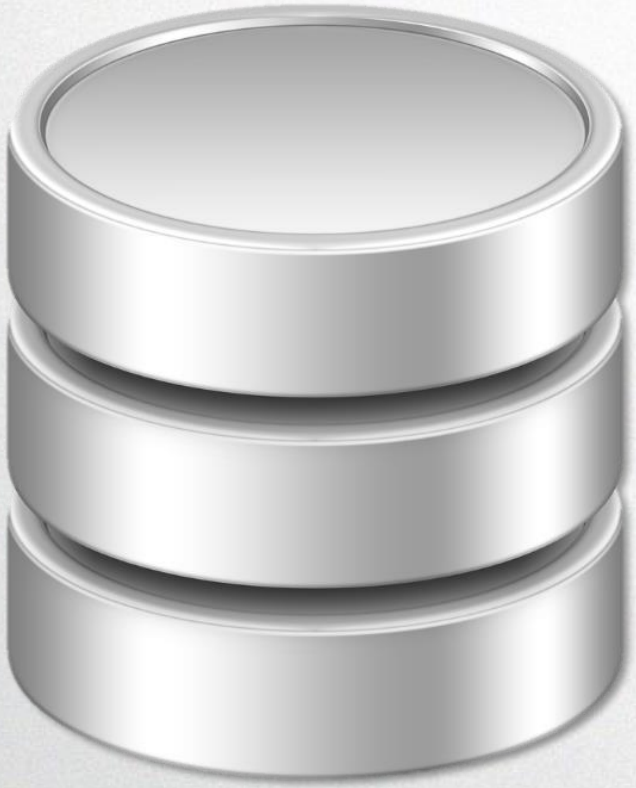
Relational Databases

- Data stored in 2D tables

  (like a spreadsheet)

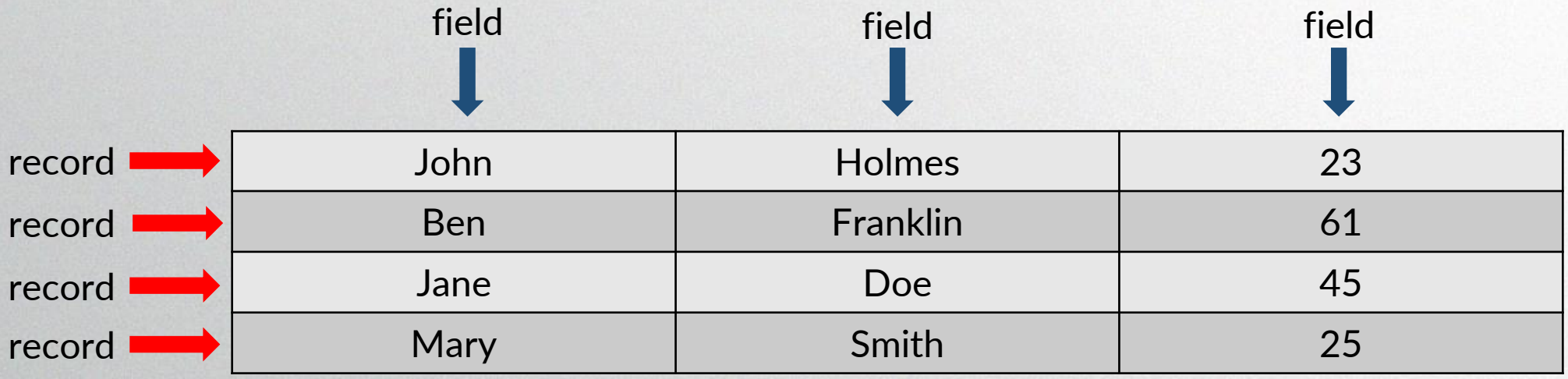
- Each **Column** == **Field**

- Each **Row** == **Record**



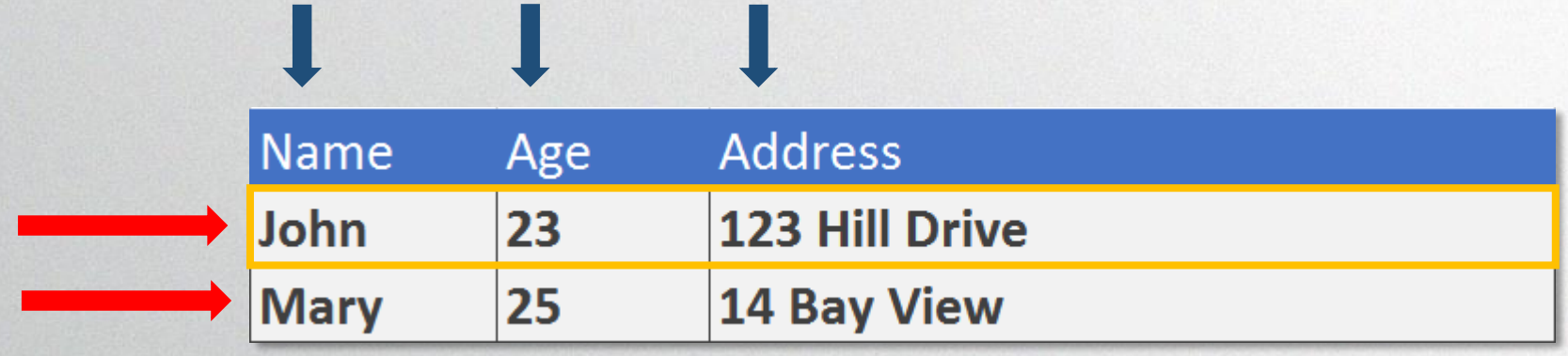
Persistent Data Storage

Relational Databases



Persistent Data Storage

Relational Databases



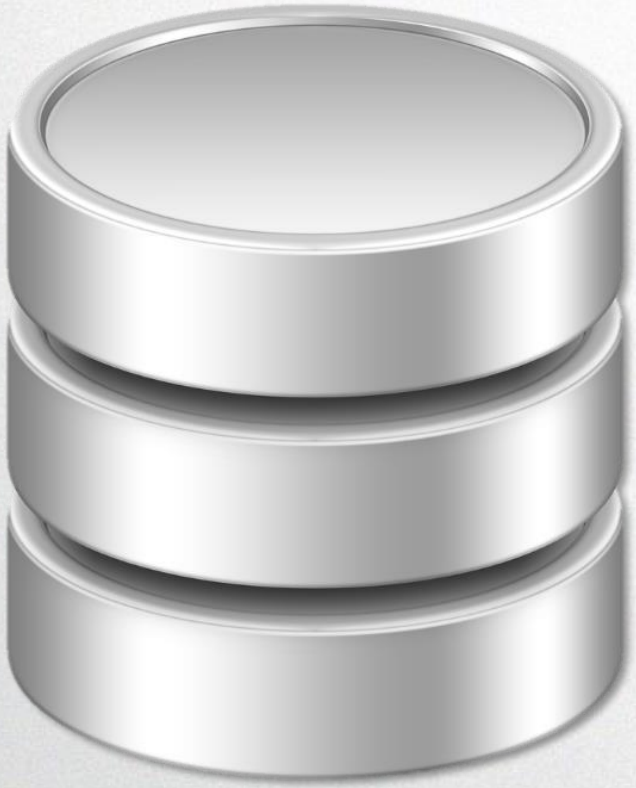
Name	Age	Address
John	23	123 Hill Drive
Mary	25	14 Bay View

Persistent Data Storage

Relational Databases

Relational Database Options

- MySQL
- Oracle
- Microsoft SQL Server



- **MariaDB**
  - Free and open source
  - Follows MySQL interface

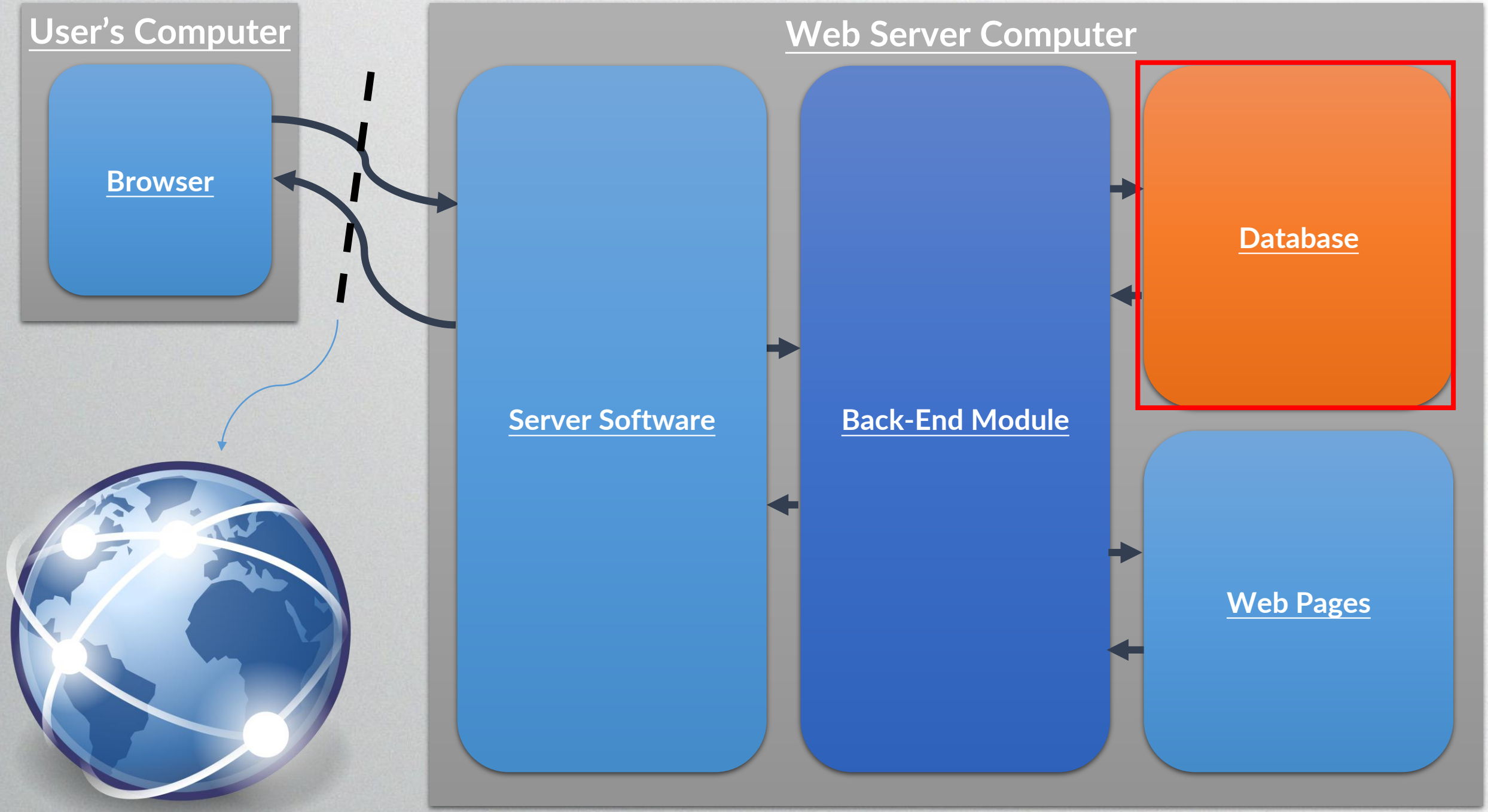
Persistent Data Storage

Relational Databases

WIKIPEDIA
The Free Encyclopedia

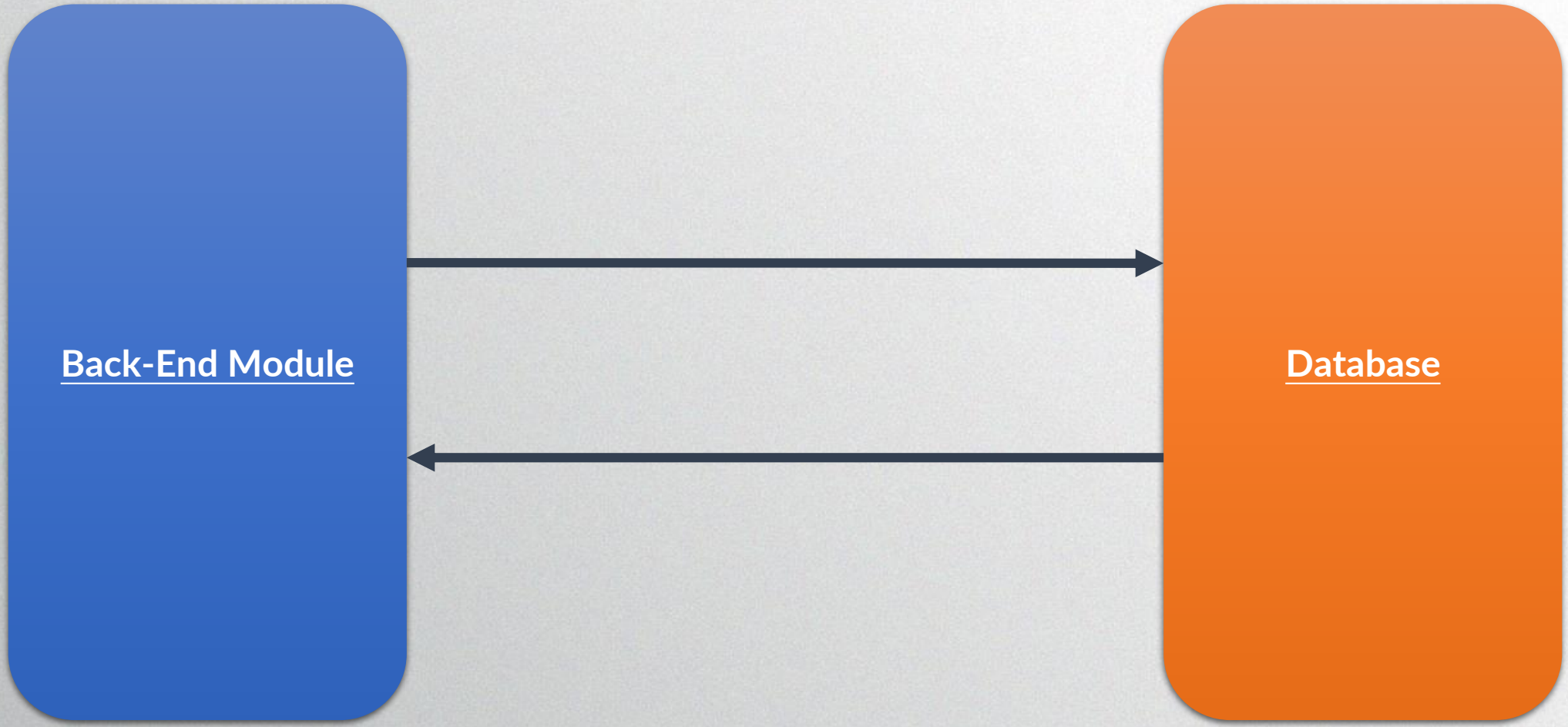
tumblr.

Google


User's Computer
Browser
Web Server Computer
Server Software
Back-End Module
Database
Web Pages

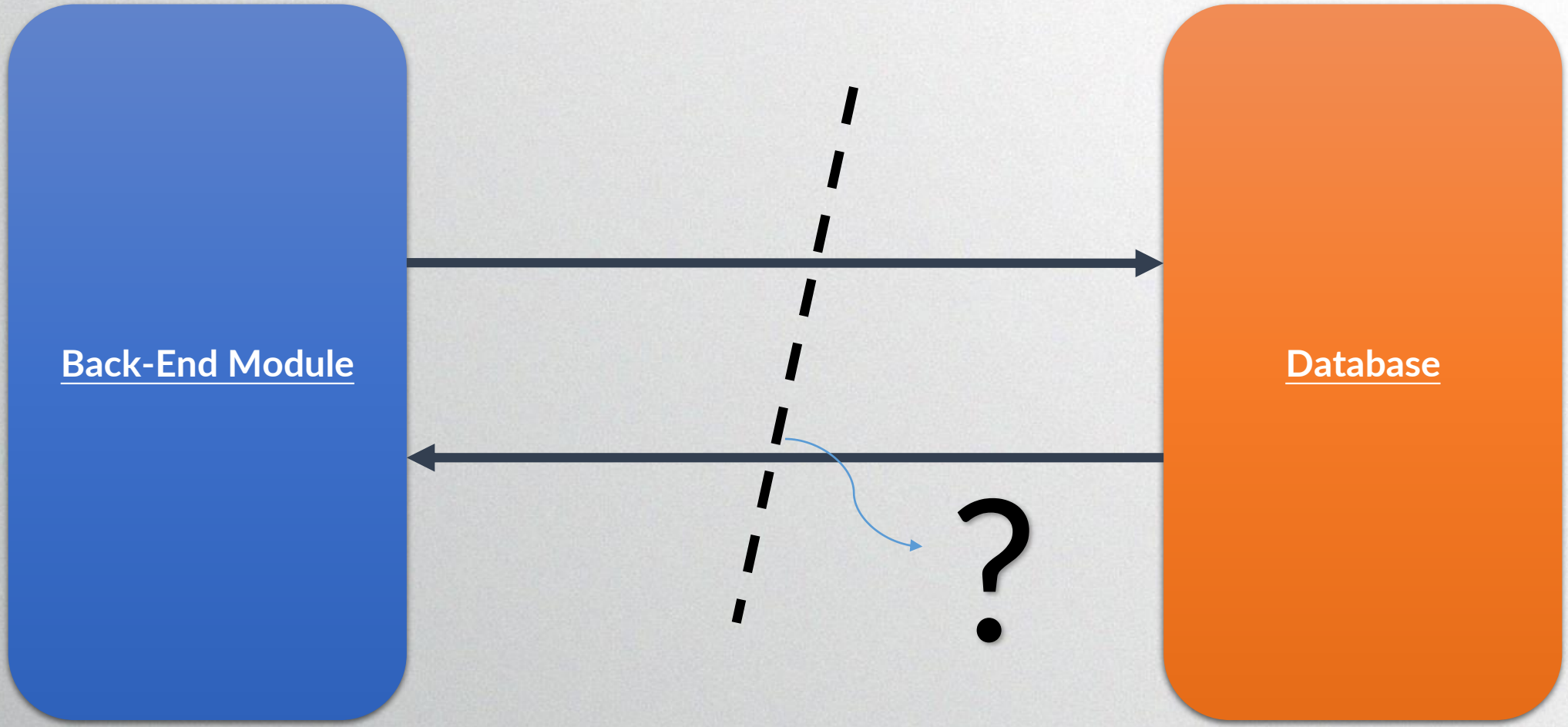
Persistent Data Storage

Relational Databases



Persistent Data Storage

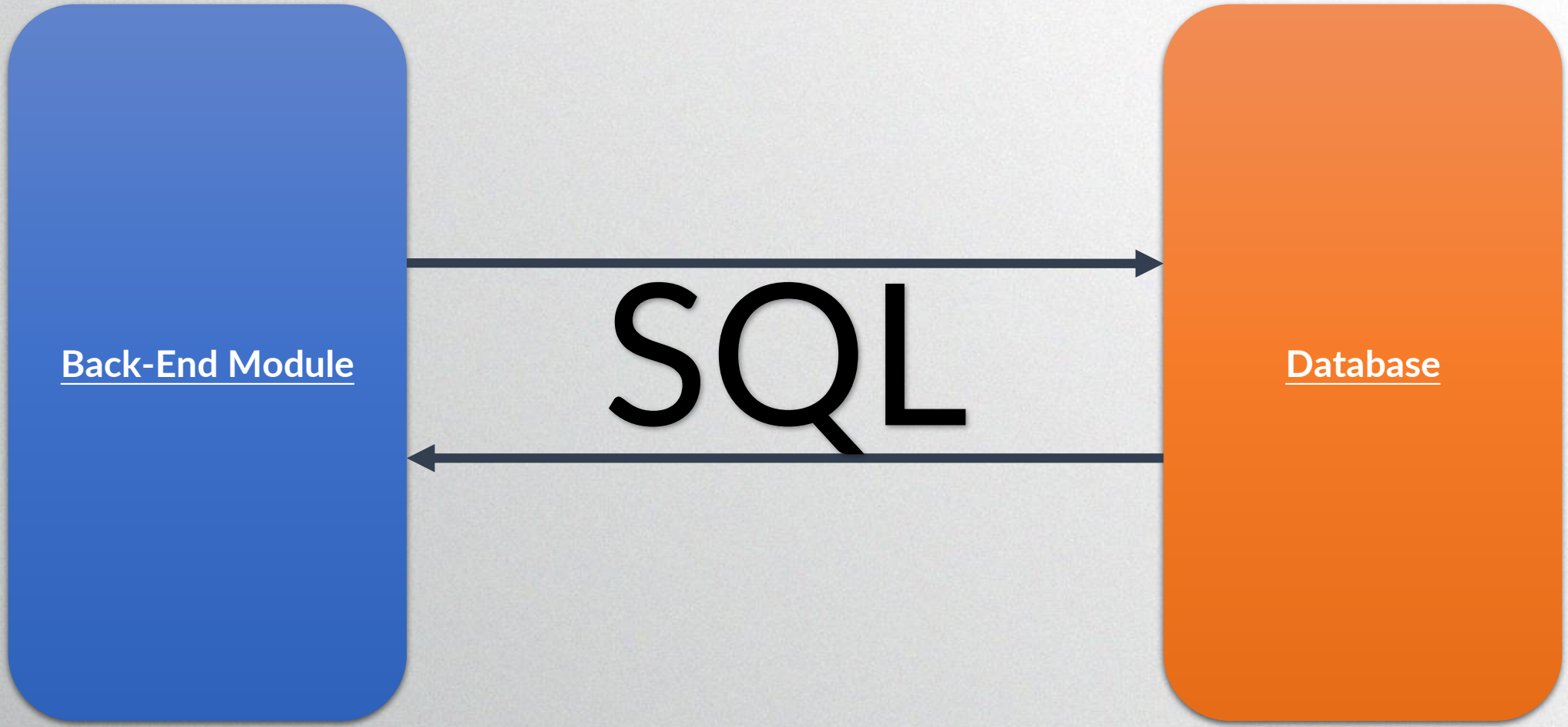
Relational Databases



Persistent Data Storage

shawacademy

Relational Databases



Structured Query Language

Julian Quirke
Web Development Educator

Structured Query Language

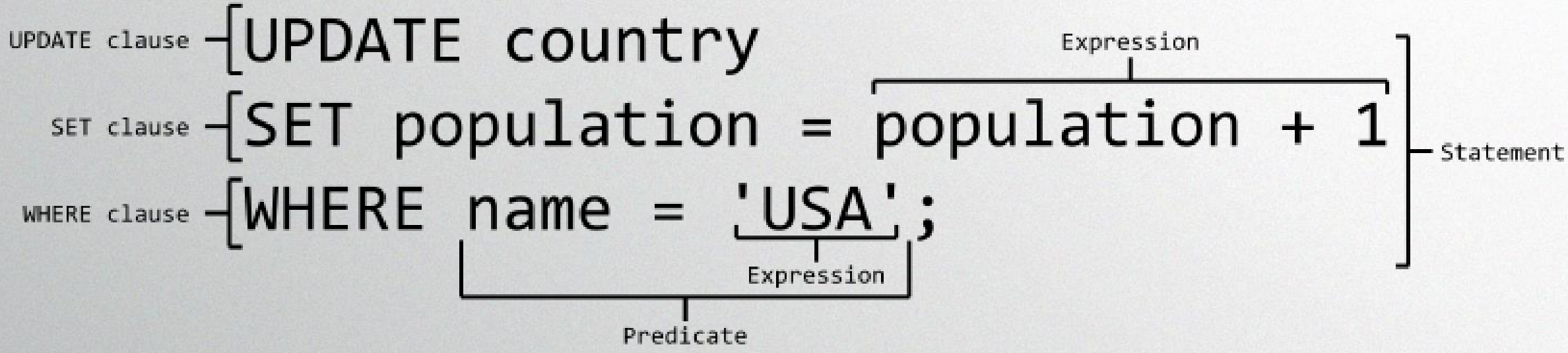
is the language used to communicate with a relational database management system

Julian Quirke
Web Development Educator

Introduction to SQL

- Looks a lot like simplified English
- Keywords signify the beginning of a clause
- Operators allow simple logical operations and conditions

Introduction to SQL



Introduction to SQL

Communicating with a Database:

1. Back-end languages send SQL queries to a RDBMS
2. Database interprets and then executes SQL queries
3. Database “returns” success or failure to sender

shawacademy

Congratulations

You’re Half Way There!!!

Keep going and achieve your Diploma

E-Diploma - €39.95 or FREE with Premium Plus

Hard Copy - €49.95 or €19.95 with Premium Plus

Next Week

Achieving the End Game

- The next session is **"Going Live"**
  - Development vs Live
  - Choosing Your Host
  - Domain Names & The Internet

- Attend all lessons **LIVE** to grow your knowledge
- Shaw Academy **12-Month Membership** Prize during **Lesson 6**
- Recordings are available within 24 hours of the live webinar

Login to www.shawacademy.com using the link at the Top Right Corner – **Members Area**

- (Use your email address and password)

Q&A

Next Lesson is

Going Live

- A look at the similarities and differences between a live server and a development server, and how to set up a good development system
- You will understand how project teams can work together to produce a flexible web application

www.shawacademy.com
www.facebook.com/shawacademy
www.twitter.com/shawacademy
support@shawacademy.com
See local numbers on website