CCCS 310
Principles of Web Development

MCGILL UNIVERSITY

COMP 307
Principles of Web Development

Lecture 18

Unit 5 – Backend Design

Database-based Websites

CCCS 310
Principles of Web Development

Class Outline

- What is a 3-tiered web application?
- What are databases?
  - About non-DB solutions: matter, CSV, XML, JSON
  - Relational Databases: SQL
  - NoSQL Databases: Mongo
- Applications
  - Dynamic content and state information
  - Examples: registration and login
- Programming Examples: php, python, node.js

Contents

3-Tiered
Databases
Applications

CCCS 310
Principles of Web Development

Contents

3-Tiered Databases Applications

Readings

- MyCourses Readings
  - WWW How to Program:
    - Ch 22 – SQL & Ch 23 - PHP
  - Full Stack Developer:
    - Ch 11 – SQL or NoSQL?
- Internet Resources
    - About relational databases
    - Relational Database Tutorial
    - PHP Programming
    - https://www.w3schools.com/php/php_mysql_intro.asp
    - https://www.w3schools.com/xml/
    - https://www.tutorialspoint.com/json/json_overview.htm
    - SQL or NoSQL ? (a blog)
    - Mongo DB (a quick guide)

What is a database-based website?

A website that stores most of its website content in external files (not HTML, CSS, JS).

This can be specially formatted text files like: matter, CSV, XML, JSON, etc.

Or special applications known as databases: examples are SQL-based and Non-SQL-based.

Contents

3-Tiered
Databases
Applications

CCCS 310
Principles of Web Development

Contents

3-Tiered
Databases
Applications

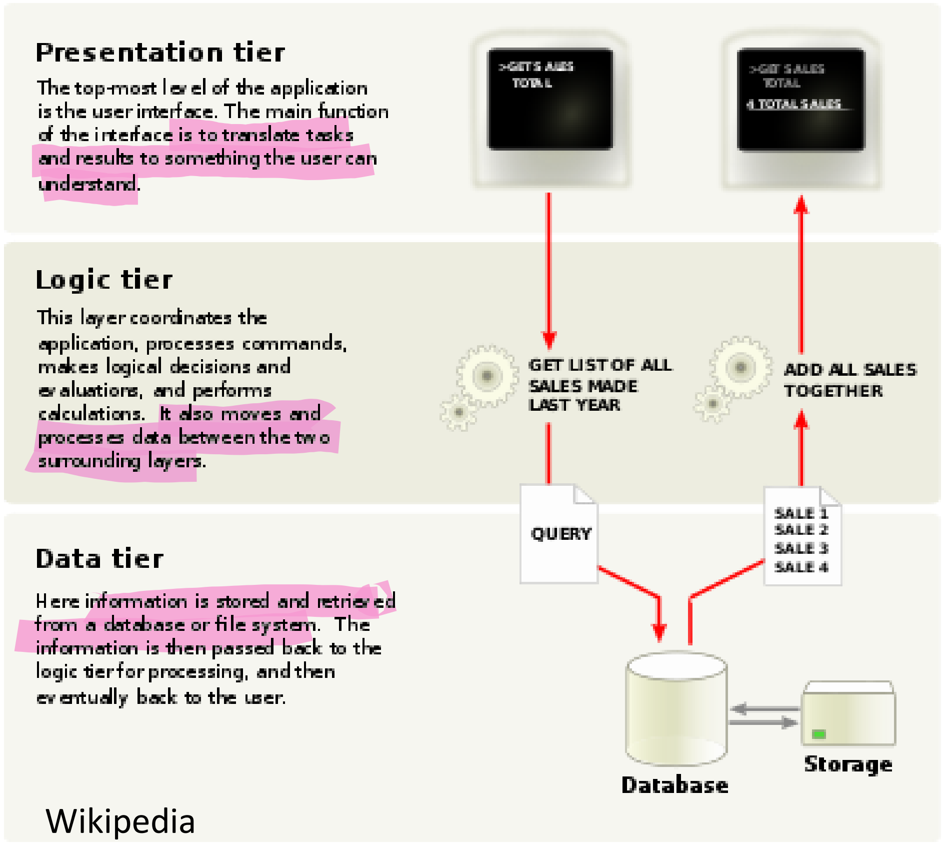
3-tired web application

Database-based Websites

CCCS 310
Principles of Web Development

What is a 3-tired web application?

- Software is developed as 3 separate applications
  - Presentation App
  - Logic App
  - Data App
- Presentation app is downloaded to browser.
- Logic app runs on the server.
- Data app runs on a second server.


Presentation tier
The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.

Logic tier
This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.

Data tier
Here information is stored and retrieved from a database or file system. The information is then passed back to the user.

MVC

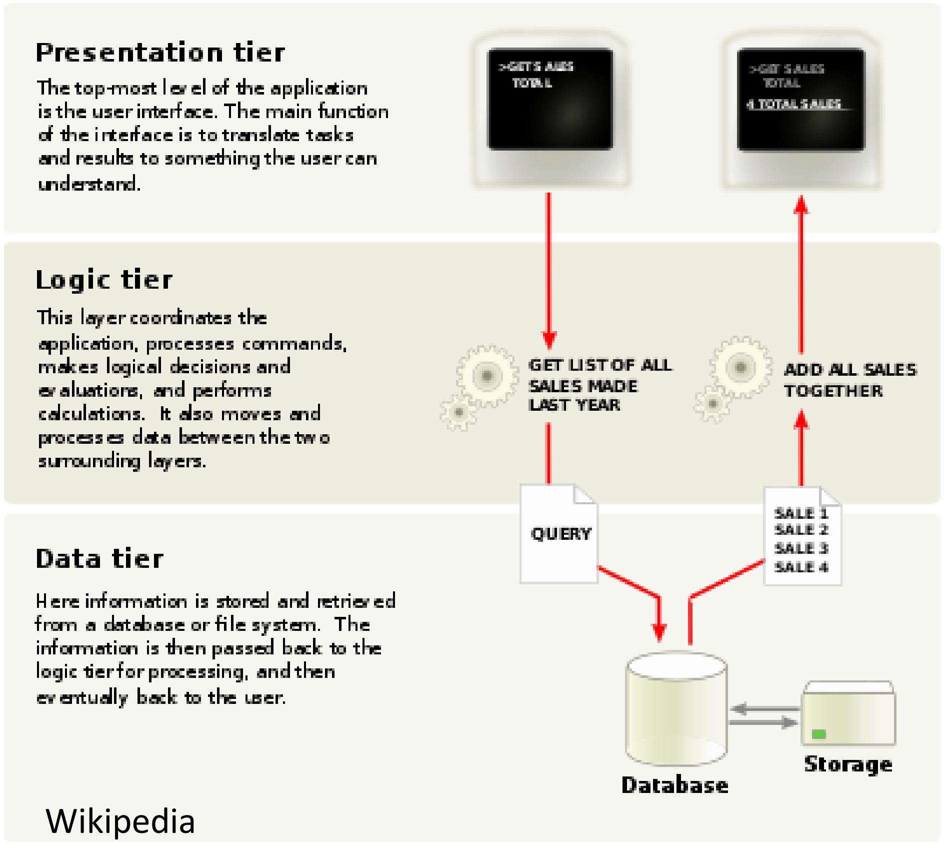
Contents

3-Tiered
Databases
Applications

CCCS 310
Principles of Web Development

What is a 3-tired web application?

- Benefits:
  - Tiers are independent
  - Different teams can work on each tier
    - Don't need to work on the full stack
  - Modifications at a tier do not affect the other tiers as long as the API **signatures** do not change
  - Since tiers are independent
    - Easy to scale a tier
    - Easy to upgrade a tier



Presentation tier

The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.

Logic tier

This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.

Data tier

Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.

Wikipedia

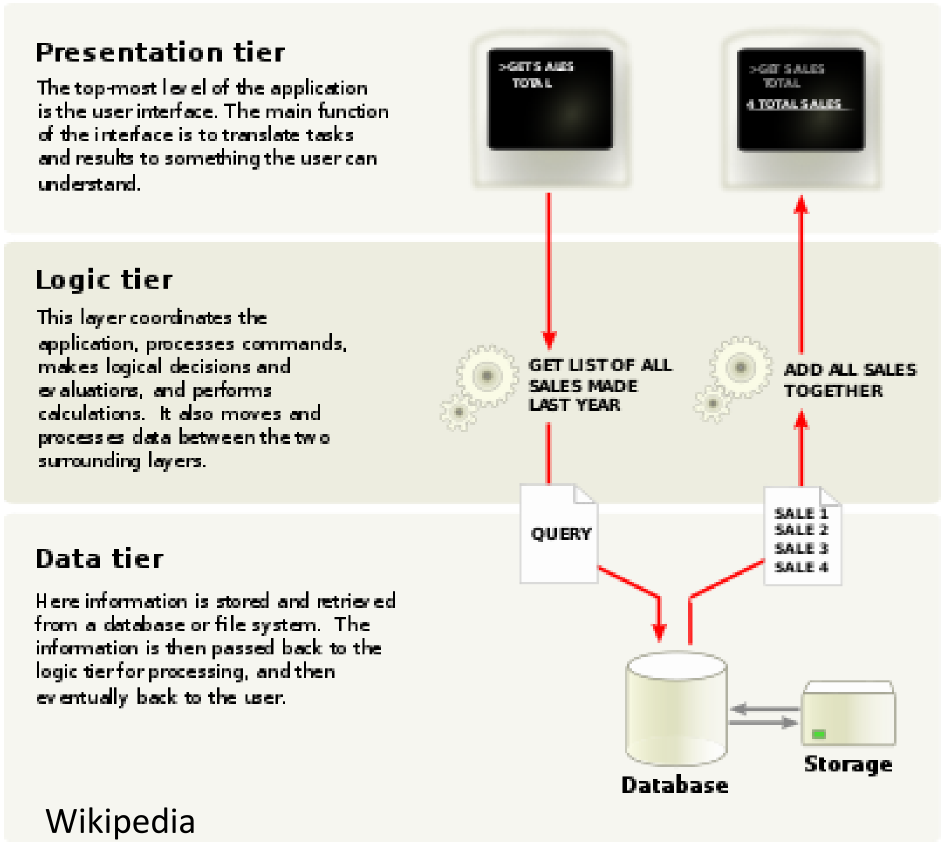
Contents

3-Tiered
Databases
Applications

CCCS 310
Principles of Web Development

What is a 3-tired web application?

- Drawbacks
  - API calls are slower than function/method calls
  - Server preprocessing
  - Network routing delays since Data Tier may run on another server.



Presentation tier
The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.

Logic tier
This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.

Data tier
Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.

Wikipedia

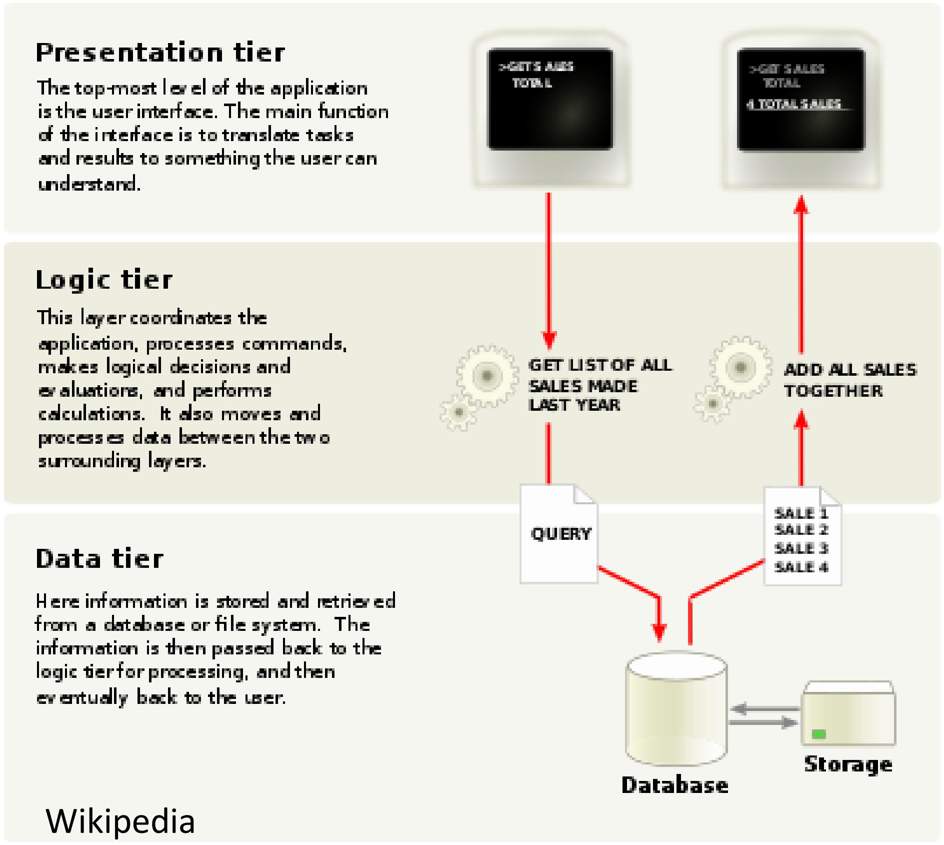
Contents

3-Tiered
Databases
Applications

CCCS 310
Principles of Web Development

What is a 3-tired web application?

- Who uses this?
  - Everybody…
  - Google
  - Facebook
  - Stores
  - Etc.



Presentation tier

The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.

Logic tier

This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.

Data tier

Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.

Wikipedia

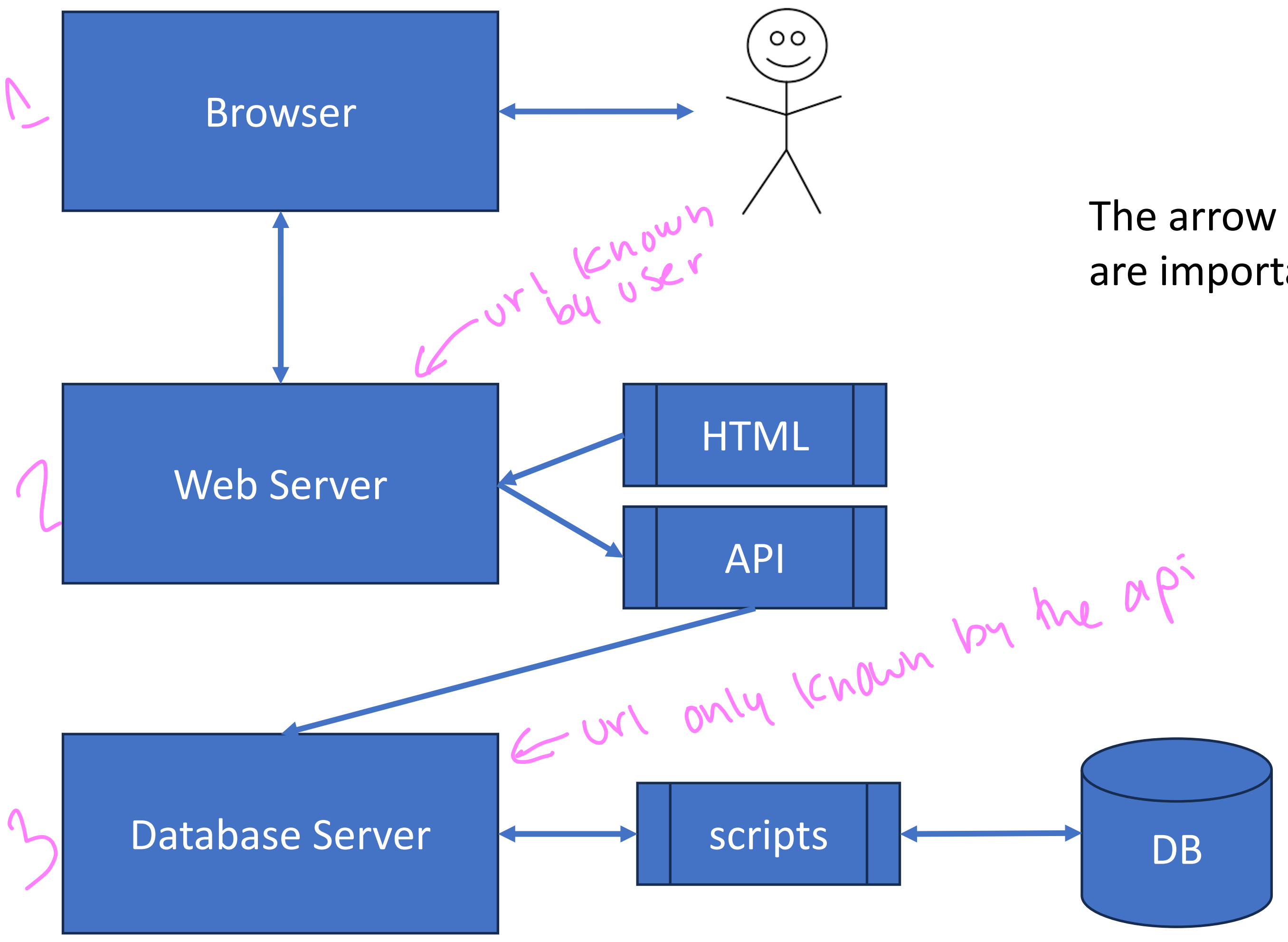
Contents

3-Tiered Databases Applications

CCCS 310 Principles of Web Development

3-tired Architectures

3 machines



The arrow directions are important.

Contents

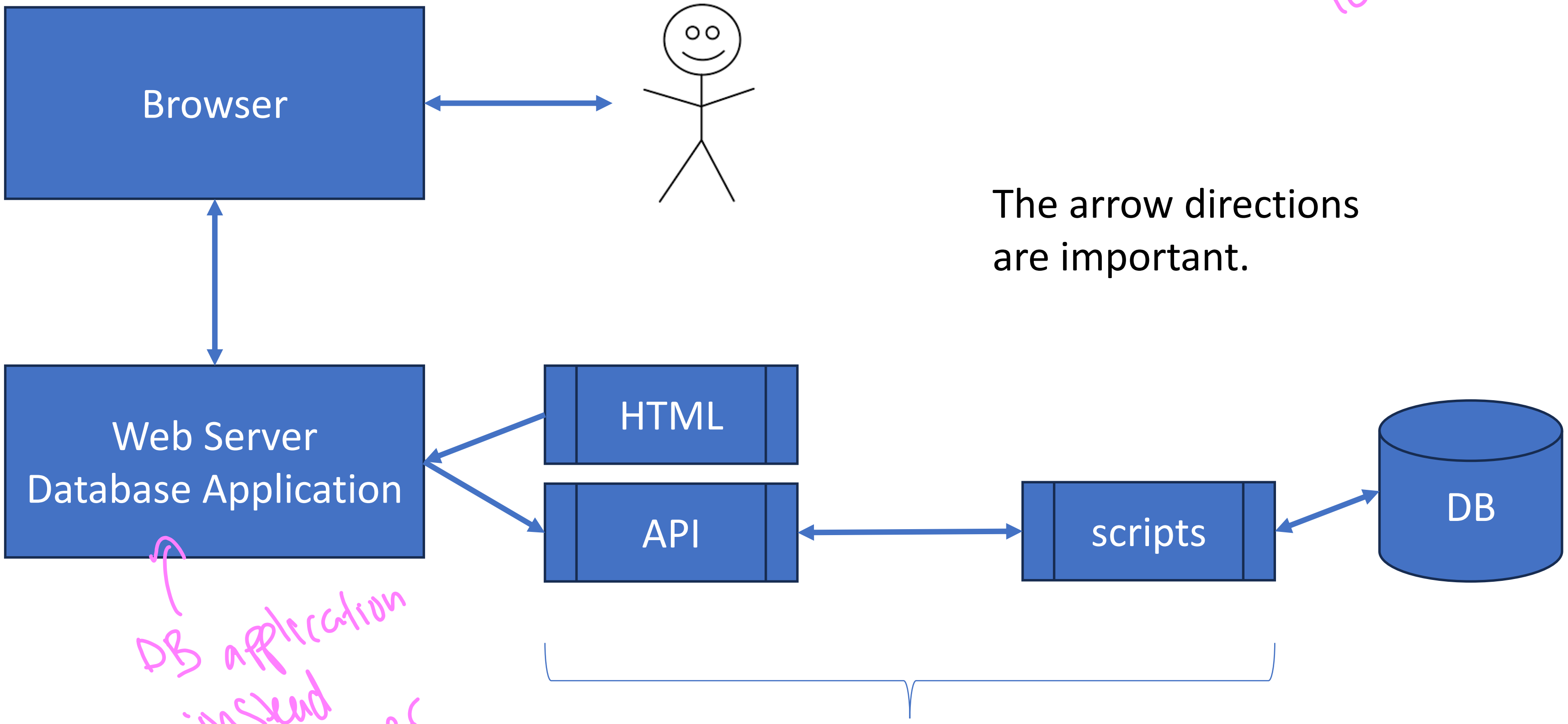
3-Tiered
Databases
Applications

CCCS 310 Principles of Web Development

3-tired Architectures

Xampp is this (when installed locally)

2 machine version



The arrow directions are important.

API and Script can be merged into a single script.

advantage → cheaper

Contents

3-Tiered Databases Applications

CCCS 310
Principles
of Web
Development

What are databases?

Database-based Websites

Contents

3-Tiered
Databases
Applications

CCCS 310
Principles of Web Development

What is a database?

- It is a file
- The file is structured
  - Fields
    - A field is a labeled piece of information – could be keys
    - Example: Name (label) = Bob Smith (information)
  - Records
    - A record is a labeled grouping of related fields
    - Example:
      - Books (group label) :
        - Title (field label): The Lord of the Rings
        - Author (field label): Tolken
        - Price (field label): 20.00
  - Key
    - A specific identified field that will be used for sorting and searching
    - More than one key can be identified for a record

Contents

3-Tiered Databases Applications

CCCS 310 Principles of Web Development

What can be a database?

- Any kind of file
- Text files:
  - CSV, XML, JSON
- Specially designed for database applications
  - Relationally Formated Tables
    - E.g., SQL
    - Products:
      - Maria, mySQL, SQL Light, Prosgress, IBM DB2, MS SQL, JDBC
  - Free Formated Objects
    - E.g., No-SQL
    - Products:
      - Mongo DB, Document DB, Casandra

Contents

3-Tiered Databases Applications

CCCS 310
Principles of Web Development

Persistent Data

Data that stays on the Server or Client machine for the length of the membership to the service.

Persistent data is stored in a file.

Note: local and global variables are dealocated after the program terminates.

Note: REST does not automatically record any information - it is left to the developer

Contents

3-Tiered
Databases
Applications

CCCS 310
Principles of Web Development

Examples: non-DB

Database-based Websites

Why formatted text files?
- better for certain use cases
  ↳ log files
  - interprocess communication
  - sharing data
  - streams
  - specially structured info
  - flexible format cases

→ Supporting needed tools
↳ grep, sed, etc
- spreadsheets
- perl
- CGI

Contents

3-Tiered
Databases
Applications

CCCS 310
Principles of Web Development

Contents

3-Tiered Databases Applications

XML

- Works with the same writing rules as HTML
  - HTML is for formatting text
  - XML is for formatting data

but no predefined tags
↳ you make your own tags

- Naming Rules
  - Names can contain letters, numbers, and other characters
  - Names cannot start with a number or punctuation character
  - Names cannot start with the letters **xml** (or XML, or Xml, etc)
  - Names cannot contain spaces

XML is verbose mode
↳ very english/readable

CCCS 310
Principles of Web Development

XML

- XML is a Tree Structure

```
<root>
  <child>
    <subchild>.....</subchild>
  </child>
</root>
```

Contents

3-Tiered
Databases
Applications

CCCS 310
Principles of Web Development

XML

- Example

```
<?xml version="1.0" encoding="ISO-8859-1" ?>

<note>                                          ← root
  <date>2008-01-10</date>                           }
  <to>Bob</to>                                      }  children
  <from>Jani</from>                                 }
  <heading>Reminder</heading>                       }
  <body>Don't forget me this weekend!</body>
</note>
```

Contents

3-Tiered
Databases
Applications

CCCS 310
Principles of Web Development

XML

- Example

```
<bookstore>
  <book category="CHILDREN">
    <title>Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="WEB">
    <title>Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

← root

← attributes → also up to you

← children

} sub children

Contents

3-Tiered Databases Applications

CCCS 310 Principles of Web Development

XML

- Inserting XML Validation

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE note SYSTEM "Note.dtd">
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <stuff>hello</stuff>
  <body>Don't forget me this weekend!</body>
</note>
```

↑ syntax file, imposes a syntax on how you write your file

with next page this <stuff> would cause an error

Contents

3-Tiered
Databases
Applications

CCCS 310 Principles of Web Development

XML

- XML DTD

```
<!DOCTYPE note
[
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>   ← anything
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]>
```

Syntax

```
<!ELEMENT element-name category>
or
<!ELEMENT element-name (element-content)>
<!ATTLIST element-name
                                        attribute-name
                                        attribute-type
                                        default-value>
```

Example:

```
<!ATTLIST payment type CDATA "check">
```

Contents

3-Tiered
Databases
Applications

CCCS 310
Principles of Web Development

XML

- DTD Elements

The **attribute-type** can be one of the following:

Type	Description
CDATA	The value is character data
(*en1*|*en2*|..)	The value must be one from an enumerated list
ID	The value is a unique id
IDREF	The value is the id of another element
IDREFS	The value is a list of other ids
NMTOKEN	The value is a valid XML name
NMTOKENS	The value is a list of valid XML names
ENTITY	The value is an entity
ENTITIES	The value is a list of entities
NOTATION	The value is a name of a notation
Xml:	The value is a predefined xml value

Contents

3-Tiered Databases Applications

CCCS 310
Principles of Web Development

Contents

3-Tiered Databases Applications

XML

- Default Values

The **default-value** can be one of the following:

Value	Explanation
The value	The default value of the attribute
#REQUIRED	The attribute is required
#IMPLIED	The attribute is not required
#FIXED *value*	The attribute value is fixed

CCCS 310
Principles of Web Development

Contents

3-Tiered Databases Applications

XML

- DTD Entities

Syntax: <!ENTITY entity-name "entity-value">

Example **definition**:
<!ENTITY writer1 SYSTEM "http://www.abc.com/entities.dtd">

Example **definition**:
<!ENTITY writer2 "Donald Duck.">

Example **use** in XML:
<author>&writer2;</author>

CCCS 310
Principles of Web Development

XML

- DTD Example

```
<!DOCTYPE TVSCHEDULE [

<!ELEMENT TVSCHEDULE (CHANNEL+)>
<!ELEMENT CHANNEL (BANNER,DAY+)>
<!ELEMENT BANNER (#PCDATA)>
<!ELEMENT DAY (DATE,(HOLIDAY|PROGRAMSLOT+)+)>
<!ELEMENT HOLIDAY (#PCDATA)>
<!ELEMENT DATE (#PCDATA)>
<!ELEMENT PROGRAMSLOT (TIME,TITLE,DESCRIPTION?)>
<!ELEMENT TIME (#PCDATA)>
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT DESCRIPTION (#PCDATA)>

<!ATTLIST TVSCHEDULE NAME CDATA #REQUIRED>
<!ATTLIST CHANNEL CHAN CDATA #REQUIRED>
<!ATTLIST PROGRAMSLOT VTR CDATA #IMPLIED>
<!ATTLIST TITLE RATING CDATA #IMPLIED>
<!ATTLIST TITLE LANGUAGE CDATA #IMPLIED>
]>
```

Contents

3-Tiered
Databases
Applications

CCCS 310
Principles of Web Development

JSON

- Why JSON?
  - Structured object-like syntax
  - Used as a replacement for the CGI query string in JS apps

- Example:

```
{"employees":[
    {"firstName":"John", "lastName":"Doe"},
    {"firstName":"Anna","lastName":"Smith"},
    {"firstName":"Peter","lastName":"Jones"}

]}
```

Data value

Array value(s)

```
<employees>

    <employee>
        <firstName>John</firstName>
        <lastName>Doe</lastName>
    </employee>

    <employee>
        <firstName>Anna</firstName>
        <lastName>Smith</lastName>
    </employee>

    <employee>
        <firstName>Peter</firstName>
        <lastName>Jones</lastName>
    </employee>

</employees>
```

Contents

3-Tiered Databases Applications

CCCS 310 Principles of Web Development

Contents

3-Tiered
Databases
Applications

Record,
Class,
Struct

JSON

- Format comparison

Field value pairs

```
{"widget": {
    "debug": "on",
    "window": {
        "title": "Sample Konfabulator Widget",
        "name": "main_window",
        "width": 500,
        "height": 500
    },
    "image": {
        "src": "Images/Sun.png",
        "name": "sun1",
        "hOffset": 250,
        "vOffset": 250,
        "alignment": "center"
    },
    "text": {
        "data": "Click Here",
        "size": 36,
        "style": "bold",
        "name": "text1",
        "hOffset": 250,
        "vOffset": 100,
        "alignment": "center",
        "onMouseUp": "sun1.opacity = (sun1.opacity /
100) * 90;"
    }
}}
```

```
<widget>

    <debug>on</debug>

    <window title="Sample Konfabulator Widget">
        <name>main_window</name>
        <width>500</width>
        <height>500</height>
    </window>

    <image src="Images/Sun.png" name="sun1">
        <hOffset>250</hOffset>
        <vOffset>250</vOffset>
        <alignment>center</alignment>
    </image>

    <text data="Click Here" size="36" style="bold">
        <name>text1</name>
        <hOffset>250</hOffset>
        <vOffset>100</vOffset>
        <alignment>center</alignment>
        <onMouseUp>
            sun1.opacity = (sun1.opacity / 100) * 90;
        </onMouseUp>
    </text>

</widget>
```

CCCS 310
Principles of Web Development

JSON

- JS object to packet string

```
var myObj = { "name":"John", "age":31, "city":"New York" };
var myJSON = JSON.stringify(myObj);
```

Can merge with CGI:

```
window.location = "demo_json.php?x=" + myJSON;
```

Contents

3-Tiered
Databases
Applications

CCCS 310
Principles of Web Development

JSON

- JS packet string to JSON

Assume myJSON is '{ "name":"John", "age":31, "city":"New York" }';

```
var myObj = JSON.parse(myJSON);
document.getElementById("demo").innerHTML = myObj.name;
```

Contents

3-Tiered Databases Applications

CCCS 310
Principles of Web Development

JSON

- JSON Types
  - Strings - { "name":"John" }
  - Numbers - { "age":30 }
  - Objects - { "employee":{ "name":"John", "age":30, "city":"New York" }   }
  - Arrays - { "employees":["John", "Anna", "Peter"] }
  - Booleans - { "sale":true }
  - Null - { "middlename":null }

Contents

3-Tiered
Databases
Applications

CCCS 310
Principles
of Web
Development

Examples: databases

Database-based Websites

Contents

3-Tiered
Databases
Applications

CCCS 310
Principles of Web Development

What is a relational DB?

SQL

- A single file with many tables
- Each table is composed of records & fields
- A table's field can be used as a key for sorting, searching, and relations (connecting tables)
- SQL is a language to express database operations like:
  - Make a table with fields
  - Search and sort
  - Delete a record or table
  - Edit fields

need to specify size of every field

table is a fixed size data structure once defined

↳ allows fseek

Loc = start + (rec # * rec size)

↳ jumps to record

Contents

3-Tiered
Databases
Applications

CCCS 310
Principles of Web Development

Term Definitions

Field selected to be the key

Database file

Table 1

Stud ID	First Name	Last Name	Age
1234	Bob	Smith	18
7890	Mary	Legault	20

Field names

A field

One row is a record

Relation / Relationship

Table 2

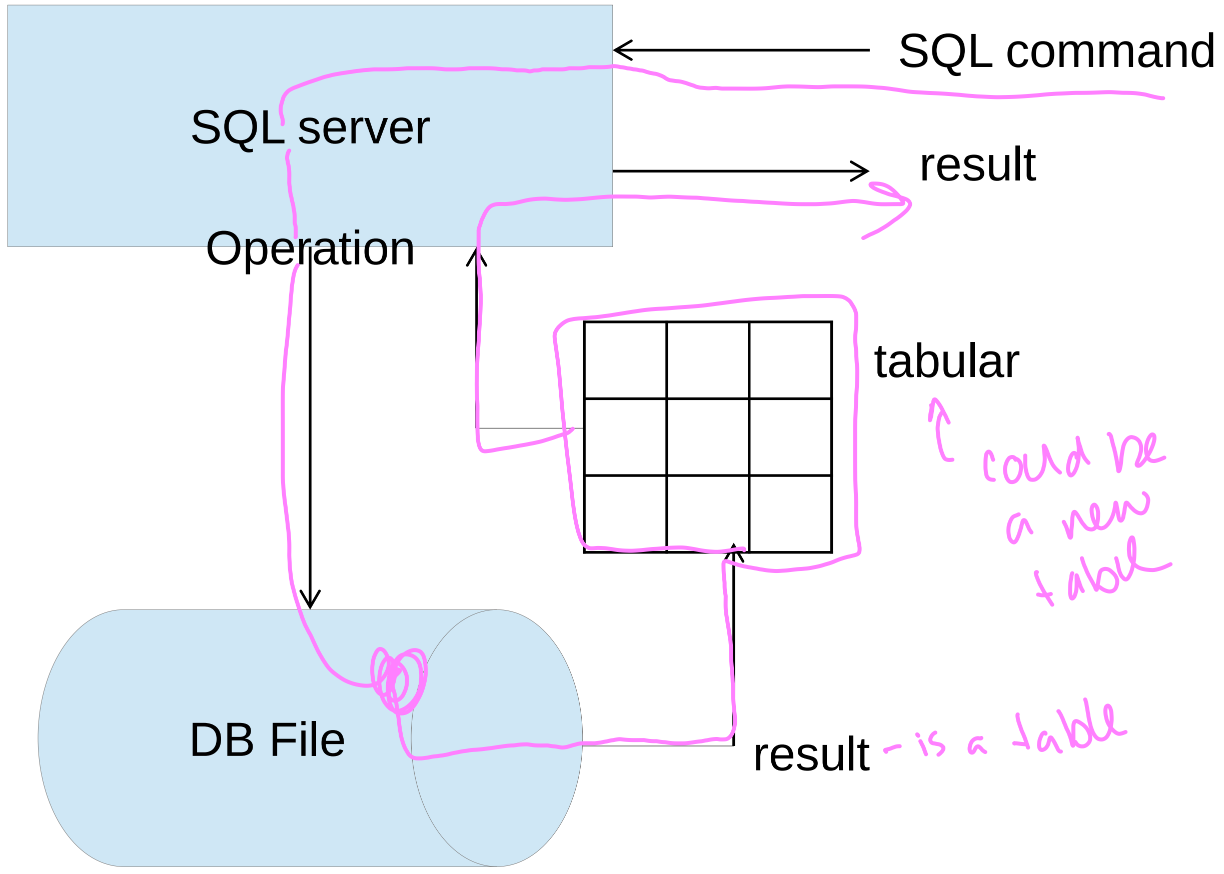
Stud ID	Fee	Pay Status
1234	$100	Paid
1234	$50	Not paid
7890	$100	Not paid

Contents

3-Tiered
Databases
Applications

CCCS 310
Principles of Web Development

mySQL / MariaDB



Contents

3-Tiered
Databases
Applications

CCCS 310
Principles of Web Development

SQL Query

- Definition: asking for information
- Syntax:
  - **select** * **from** table **where** field > value
  - **select** field1, field2 **from** table **where** condition

Contents

3-Tiered
Databases
Applications

CCCS 310
Principles of Web Development

SQL Query Examples

SELECT CustomerName,City FROM Customers;

SELECT * FROM Customers;

SELECT * FROM Customers WHERE Country='Mexico';

SELECT * FROM Customers WHERE Country='Germany' AND City='Berlin';

SELECT * FROM Customers ORDER BY Country DESC;

SELECT * FROM Customers WHERE Country='X' ORDER BY City DESC;

If tables have a relation then:

SELECT ID, City FROM Cust, Payment WHERE City='X'

This creates an imaginary table containing all the fields from both tables organized by the key field.

Contents

3-Tiered
Databases
Applications

CCCS 310
Principles of Web Development

User and program create the SQL query

SQL Injection

Software creating SQL statements:

Safe

```
uName = getRequestString("UserName");
uPass = getRequestString("UserPass");

sql = "SELECT * FROM Users WHERE Name ='" + uName + "' AND Pass ='" + uPass + “'”

db.Execute(sql);
```

Injection could be dangerous because you don’t know what the user will type.

NOT Injection: because the query it fully written beforehand

```
$result = mysqli_query($con,"SELECT * FROM Persons");
```

$\text{Sql = “SELECT * FROM Users WHERE “} \underbrace{\text{+ varFromUser;}}_{\text{user picking query}} \leftarrow$ dangerous

Contents

3-Tiered Databases Applications

CCCS 310
Principles of Web Development

Using the XAMPP SQL Interface

(demo)

PROBLEM: Create a database and query the database

localhost/phpmyadmin

Contents

3-Tiered
Databases
Applications

PHP Example

```
<?php
$conn = new mysqli(“localhost”, “root”, “”, “websitedb”);

if ($conn->connect_error) {
  die(“Internal Server Error: " . $conn->connect_error);
}

$sqlValid     = "SELECT * FROM valid_users WHERE user=’".$_POST["username"].
              "’ and pass=’".$_POST["password"]."’"; # notice single quotes
$sqlContent = “SELECT * FROM content WHERE user=‘”.$_POST[“username”].”’”;

$result = $conn->query($sqlValid);
if ($result->num_rows != 0) {
    $result = $conn->query($sqlContent);
    if ($result->num_rows > 0) {
        while($row = $result->fetch_assoc()) {
             echo “<a href=‘" . $row[“information"] . “’<br>";
        }
    } else {
        echo “No content";
    }
} else { echo “Not a valid user”; }
$conn->close();
?>
```

We need to imagine HTML code that paints a pretty web page

Contents

3-Tiered Databases Applications

CCCS 310 Principles of Web Development

Python Example

```
#!/usr/bin/python

import MySQLdb, cgi
db = MySQLdb.connect("localhost","testuser","test123","TESTDB" )
cursor = db.cursor()

form = cgi.FieldStorage()

sqlValid     = "SELECT * FROM valid_user WHERE user = '%s' and pass='%s'"
                    % (form.getValue("username"), form.getValue("password"))
sqlContent = "SELECT * FROM content where user ='%s'" % (form.getValue("username"))

try:
  cursor.execute(sqlValid)
  results = cursor.fetchall()
  if result:
    cursor.execute(sqlContent)
    result = cursor.fetchall()
    for row in results:
        print "fname=%s,lname=%s,age=%d" % (row[0], row[1], row[2])
except:
  print "Error: unable to fetch data"

db.close()
```

Data returned as a string with \n where each row is:
fname=%s,lname=%s,age=%d\n

To be parsed by JS added to innerHTML.

Notice we are not generating a webpage, just returning data

Contents

3-Tiered Databases Applications

CCCS 310
Principles
of Web
Development

PHP and mySQL

```
<?php

// host, username, password, database name
$con=mysqli_connect("example.com","peter","abc123","my_db");

// Check connection
if (!$con) {
  die("Failed to connect to MySQL: " . mysqli_connect_error());
}

$result = mysqli_query($con,"SELECT * FROM Persons");

while($row = mysqli_fetch_array($result)) {
  echo $row['FirstName'] . " " . $row['LastName'];
  echo "<br>";
}

mysqli_close($con);

?>
```

Contents

3-Tiered
Databases
Applications

CCCS 310
Principles of Web Development

Database Usages:
Dynamic content & State information

Database-based Websites

Contents

3-Tiered
Databases
Applications

CCCS 310 Principles of Web Development

Dynamic Content via Databases

- Public content:
  - **Select * from Content where Page="x" and Div="y"**
    - This assumes that the webpage used AJAX to query the server for a particular <div>
    - This assumes that <div id='y'> matches the database "y"
- Personal content:
  - **Select * from Content where Page="x" and Div="y" and UserID="z"**
    - Notice that this works basically the same way as public content except that the database is also sorted by the userID.
    - We will talk more about security in another lecture, because this query is missing a couple of elements to make it secure.

Contents

3-Tiered
Databases
Applications

CCCS 310
Principles of Web Development

Select * from Table1,Table2 where StudID="1234"

Using Content

Database file

Table 1

Stud ID	First Name	Last Name	Age
1234	Bob	Smith	18
7890	Mary	Legault	20

Table 2

Stud ID	Fee	Pay Status
1234	$100	Paid
1234	$50	Not paid
7890	$100	Not paid

Notice that this could be used to fill in a <table> with the payment history of a user.

Select * from Table1, Table2 where StudID="1234"
Result returned:

Stud ID	First Name	Last Name	Age	Fee	Pay Status
1234	Bob	Smith	18	$100	Paid
1234	Bob	Smith	18	$50	Not paid

Contents

3-Tiered Databases Applications

CCCS 310 Principles of Web Development

Select * from Content where StudID="1234" and Page="Home"

Using Content

Database file

Content

Stud ID	Page	Welcome	Div1
1234	Home	Bob	Owe $
7890	Home	Mary	Fully Paid
1234	Profile	Bob Smith	
7890	Profile	Mary Jane	

Result returned:

← From relation →

Stud ID	Page	Welcome	Div1	Div2	Dive 3
1234	Home	Bob	Owe $		

JS would be used to insert the contents of the fields div1, div2, and div3 into the corresponding **innerhtml** of the actual HTML <div id="div1"> sections.

Contents

3-Tiered Databases Applications

State Information in Databases

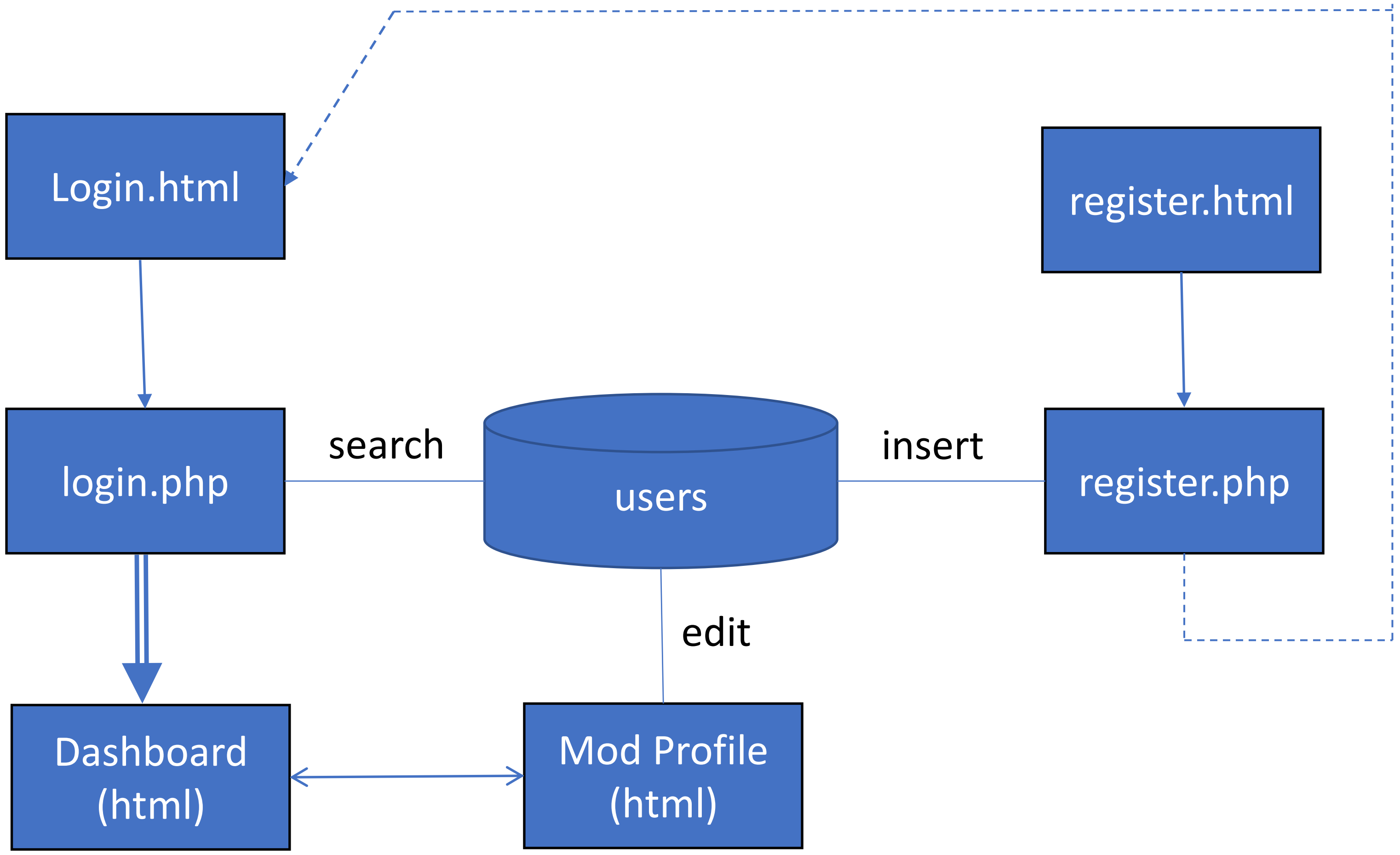
- What is state information?
  - A database (or file) that stores current (recent) information about an object.
  - Objects can be things like users, sessions, and resources.
    - User account information: user ID, password, etc.
    - Room reservations: item given to a user for a period
    - Session: where STDIN and STDOUT are pointing, $HOME, valid API calls, etc.
    - Permissions to resources (files, printers, rooms)
  - State information tends to change over time but has immediate impact on what can be done.
- Use Cases
  - Registration and login
  - Security: encryption keys
  - Permission to invoke an API
  - Permission to access a resource

Contents

3-Tiered
Databases
Applications

CCCS 310
Principles of Web Development

State Information in Databases



Contents

3-Tiered
Databases
Applications

CCCS 310
Principles
of Web
Development

Performance

Database Websites 2

Contents

3-Tiered
Databases
Applications

CCCS 310
Principles of Web Development

Performance comprises…

- API call time
- Query length
- Data structure reply
- Return time
- Rendering

Time = call + query + data structure + return + rendering

From the point of view of databases, we focus on **Query** and **Data Structure** only. We assume Call is similar in all use cases, and Return + Render is proportional to the Data Structure.

depends on DB size

Contents

3-Tiered Databases Applications

CCCS 310
Principles
of Web
Development

Query Length

- Text files (CSV, XML, JSON, Matter)
  - Can only be read in-order, meaning all operations are O(n).

    when record is fixed length, can calc seek & O(1&logn)
- Databases (SQL, Mongo)
  - If not indexed are O(n)
  - If indexed are O(log n) per search
  - If pointer-based O(1)
    - Mongo uses pointers and SQL uses record numbers
- Note:
  - For small numbers O(n) and O(log n) are not that different.
  - O(n) implementations are easier to build and debug.

Contents

3-Tiered
Databases
Applications

CCCS 310
Principles of Web Development

Data Structure Size

- Text files (CSV, XML, JSON, Matter)
  - It is common to download the entire text file, except in CSV.
  - **Packet payload** = file size
- Databases (SQL, Mongo)
  - Returns a table based on the query (subset of original table)
  - If using relations, then returned table includes all relation fields based on query (subset of rows from original table**s**)
  - **Packet payload** = ∑ (rows * columns * bytes)
    - Where sum iterates over tables_in_relation
- Note:
  - For small files, text files run faster with lower overhead
  - For complex queries, database overhead is justified

Contents

3-Tiered Databases Applications

CCCS 310
Principles of Web Development

Mongo DB

Database-based Websites

Contents

3-Tiered
Databases
Applications

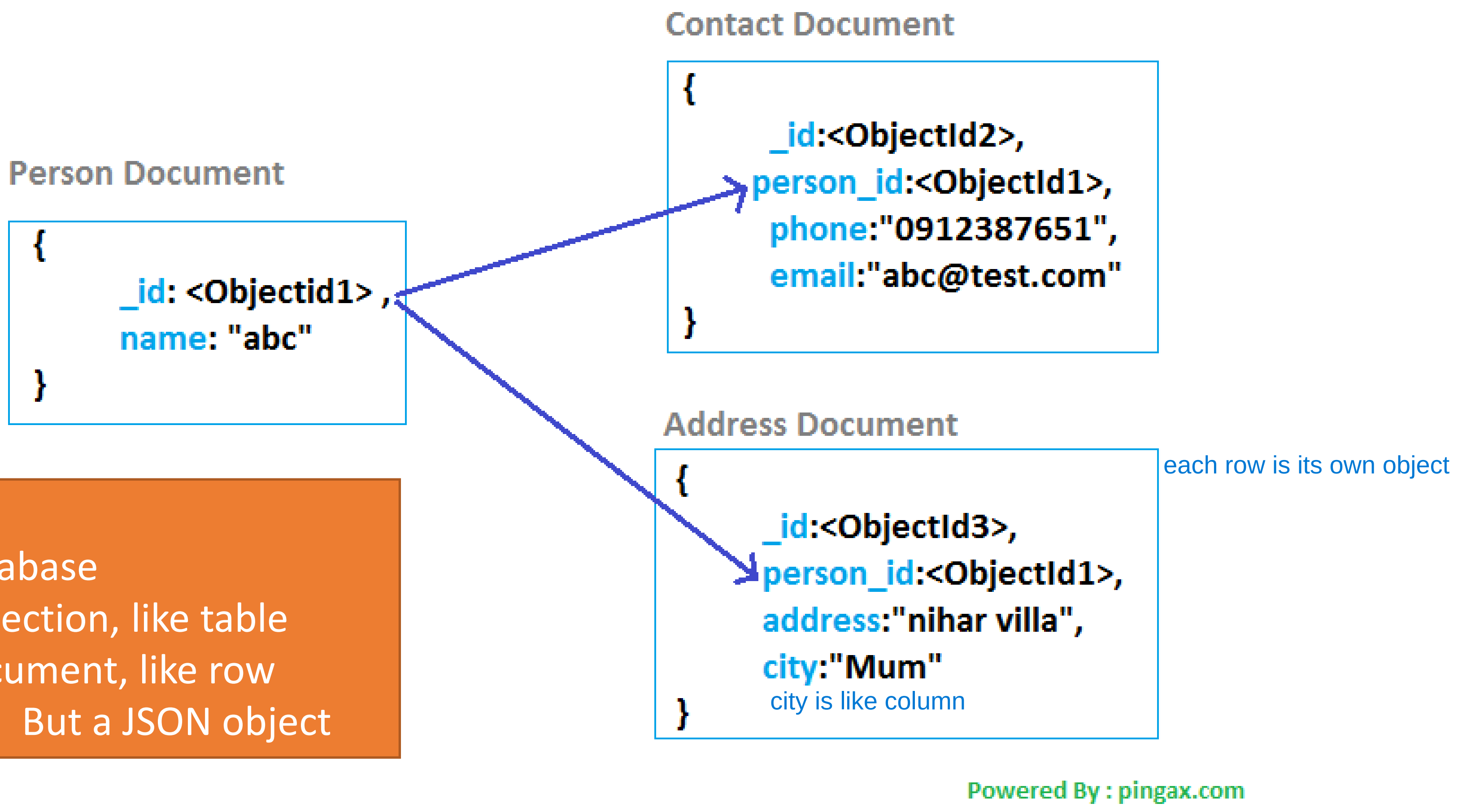
Object-based Databases

- No traditional columns and rows
- Table is a class
- A row is an object
- Fields are value-pairs within the object
  - Uses weak type checking
  - Weak pair set enforcement, like in JSON pairs.
    - Can insert new value-pairs in a specific object that is not in other objects of the same class.

Contents

3-Tiered Databases Applications

Object-based Databases



Person Document

```
{
        _id: <Objectid1> ,
        name: "abc"
}
```

Contact Document

```
{
        _id:<ObjectId2>,
      person_id:<ObjectId1>,
        phone:"0912387651",
        email:"abc@test.com"
}
```

Address Document

```
{
        _id:<ObjectId3>,
      person_id:<ObjectId1>,
      address:"nihar villa",
      city:"Mum"
}
```

Terms:
- Database
- Collection, like table
- Document, like row
    - But a JSON object

* Pointers create fast transitions from record to record
* Easy support for database size increase horizontally (adding fields)
* Non pointer searches have standard run times
* Updates can lead to graph-based balancing operations

Contents

3-Tiered
Databases
Applications

CCCS 310
Principles of Web Development

Installing Mongo DB

- Download community edition:
  - http://www.mongodb.com/try/download/community
  - This will install on your laptop, like XAMPP's MariaDB
  - You can checkmark Compass for the GUI

- Using MongoDB Compass, Create…
  - Table: User
  - **Collection**: Friend
  - Insert **Document**: {"FName":"Bob","LName":"Smith","Age":18}
    - Notice the use of JSON to create the records (documents)
    - Notice further how record structures are created on the fly (**schema**) by just inserting data into the collection.
    - This permits you to change the structure of a record for a particular document.

mongo nice because you can add 'columns' for specific object/'rows' but hard to merge table because of that

sql nice for opposite reason

Contents

3-Tiered Databases Applications

CCCS 310
Principles of Web Development

see the lab E

Mongo DB Programming

- You must first install the drivers, either add them to the same directory of your program or in the run-time's directory
  - PHP = ../ext, php.ini: extension = php_mongo.dll
  - Java = classpath
- PHP
  - https://www.php.net/manual/en/refs.database.vendors.php
- Java
  - Download the jar **mongodb-driver-3.11.2.jar** and its dependency **mongodb-driver-core-3.11.2.jar**. Make sure to download the latest release of these jar files.

Contents

3-Tiered Databases Applications

CCCS 310 Principles of Web Development

https://www.javatpoint.com/php-mongodb

Mongo DB & PHP

```
<?php
        require 'vendor/autoload.php’;  // composer PHP dependency manager

        // Creating Connection
        $con = new MongoDB\Client("mongodb://localhost:27017");

        // Creating Database
        $db = $con->dbname;

        // Creating Document
        $collection = $db->employee;

        // Insering Record
        $collection->insertOne( [ 'name' =>'Peter', 'email' =>'peter@abc.com' ] );

        // Fetching Record
        $record = $collection->find( [ 'name' =>'Peter'] );
        foreach ($record as $employe) {
                echo $employe['name'], ': ', $employe['email']."<br>";
        }
?>
```

Contents

3-Tiered Databases Applications

CCCS 310
Principles of Web Development

Contents

3-Tiered Databases Applications

https://www.javatpoint.com/java-mongodb

Mongo DB & Java

```
import com.mongodb.MongoClient;
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoDatabase;
import org.bson.Document;

public class JavaMongoDemo {
  public static void main(String[] args) {
    try{
        //---------- Connecting DataBase -------------------------//
        MongoClient mongoClient = new MongoClient( "localhost" , 27017 );
        //---------- Creating DataBase ---------------------------//
        MongoDatabase db = mongoClient.getDatabase(“myDB");
        //---------- Creating Collection -------------------------//
        MongoCollection<Document> table = db.getCollection("employee");
        //---------- Creating Document ---------------------------//
        Document doc = new Document("name", "Peter John");
        doc.append("id",12);
        //----------- Inserting Data ----------------------------//
        table.insertOne(doc);
    }catch(Exception e){  System.out.println(e);  }
   }
}
```

https://www.javatpoint.com/python-mongodb

Mongo DB & Python

```
from pymongo import MongoClient     # import mongo client to connect
import pprint

# Creating instance of mongoclient
client = MongoClient()

# Creating database
db = client.dbname
employee = {"id": "101",
              "name": "Peter",
              "profession": "Software Engineer",
}

# Creating document
employees = db.employees

# Inserting data
employees.insert_one(employee)

# Fetching data
pprint.pprint(employees.find_one())
```

Contents

3-Tiered Databases Applications

CCCS 310
Principles of Web Development

https://www.javatpoint.com/nodejs-mongodb-insert

Mongo DB & Node.js & JS

See the server3WithDB.js code

Contents

3-Tiered
Databases
Applications

CCCS 310
Principles of Web Development

Prepare for Next Class

- Assignments
  - Mini 6 due Nov 7
  - Project out Nov 7
- Lab this week
  - Lab E – SQL & Mongo
- Do on your own
  - Get a PHP program to query an SQL XAMPP database
  - Get the Node.JS server database code working

Contents

3-Tiered
Databases
Applications