Client/Server Programming for Internet Applications

TCSS460

Summer 2020

Lecture 12
- Node.js and Databases
- Web Services

TACOMA

Databases

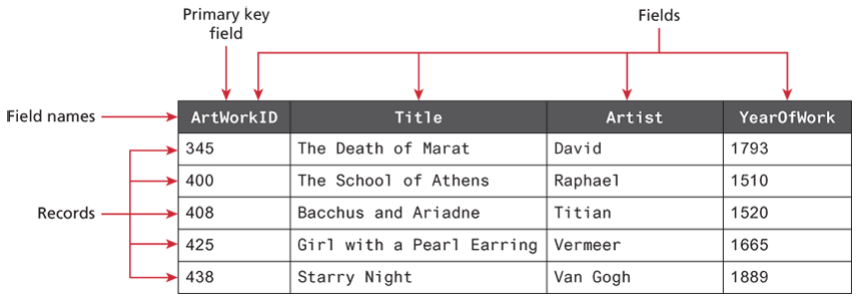
- **Database (DB)**
  - collection of related **data**
    - **data** → known facts that can be recorded and that have implicit meaning
  - DB represents some aspect of the real world
  - changes in the real-world are reflected in the DB
  - many open-source and proprietary relational DBMSs
    - MySQL
    - Oracle Database
    - Microsoft SQL Server
    - IBM DB2
    - PostgreSQL

2

Databases (cont'd)

database structure

- a database is composed of one or more **tables**
  - a **table** is a set of a ***number*** of **rows** (or **records**)
  - a **row** is an ordered list of ***n*** **values** (or **fields**)
- a **primary key** that is used to uniquely identify each row



ArtWorkID	Title	Artist	YearOfWork
345	The Death of Marat	David	1793
400	The School of Athens	Raphael	1510
408	Bacchus and Ariadne	Titian	1520
425	Girl with a Pearl Earring	Vermeer	1665
438	Starry Night	Van Gogh	1889

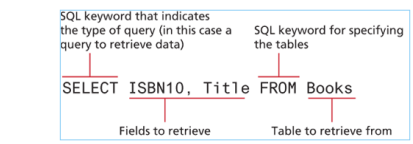
Randy Connolly, Ricardo Hoar, Fundamentals of Web Development (2nd Edition), 2017 3

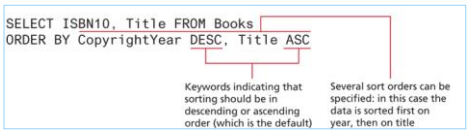
SQL

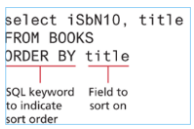
SQL

- **Structured Query Language** has been the de facto standard for interfacing with a database
  - retrieve, store, modify and delete data
  - create, modify or delete tables

SELECT (used to retrieve data from a database**)**







Randy Connolly, Ricardo Hoar, Fundamentals of Web Development (2nd Edition), 2017 4

SQL

SELECT

WHERE clause supply a comparison expression that the data must match in order for a record to be included in the result set

```
SELECT isbn10, title FROM books
WHERE copyrightYear > 2010
```

SQL keyword that indicates to return only those records whose data matches the criteria expression

Expressions take form: field *operator* value

```
SELECT isbn10, title FROM books
WHERE category = 'Math' AND copyrightYear = 2014
```

Comparisons with strings require string literals (single or double quote)

aggregate functions are used perform some type of calculation on multiple records and then return the results

This aggregate function returns a count of the number of records

Defines an alias for the calculated value

```
SELECT Count(ArtWorkID) AS NumPaintings
FROM ArtWorks
WHERE YearOfWork > 1900
```

Count number of paintings after year 1900

Note: This SQL statement returns a single record with a single value in it.

NumPaintings
745

```
SELECT Nationality, Count(ArtistID) AS NumArtists
FROM Artists
GROUP BY Nationality
```

SQL keywords to group output by specified fields

Note: This SQL statement returns as many records as there are unique values in the group-by field.

Nationality	NumArtists
Belgium	4
England	15
France	36
Germany	27
Italy	53

Randy Connolly, Ricardo Hoar, Fundamentals of Web Development (2nd Edition), 2017

5

Node.js and MySQL

```
npm install mysql

var mysql = require('mysql');
var dbcon = mysql.createConnection({
    host: "localhost",
    user: "user",
    password: "pass"
});

dbcon.connect(function(err) {
    if (err) throw err;
    console.log("You are now connected to MySQL!");
});
```

https://www.w3schools.com/nodejs/nodejs_mysql.asp

TCSS 460 - Summer 2020

6

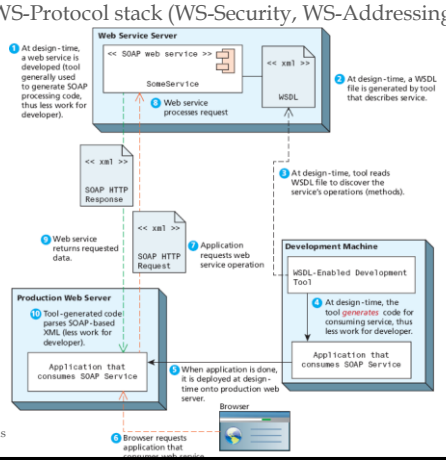
Web Services

Overview of Web Services

- **web services** are the most common example of a computing paradigm commonly referred to as **service-oriented computing (SOC)**
  - **SOC** utilizes **services** as a key element in the development and operation of software applications
  - a **service** is a piece of software with a platform-independent interface that can be dynamically located and invoked
  - a **web service** is a service that uses standardized mechanisms by which one software application can connect to and communicate with another software application using web protocols
    - web services typically use
      - HTTP
      - JSON or XML

SOAP Services

- **first generation**: attention was on a series of related XML vocabularies
  - WSDL, SOAP, WS-Protocol stack (WS-Security, WS-Addressing, etc.)



Randy Connolly, Ricardo Hoar, Fundamentals of Web Development (2nd Edition), 2017

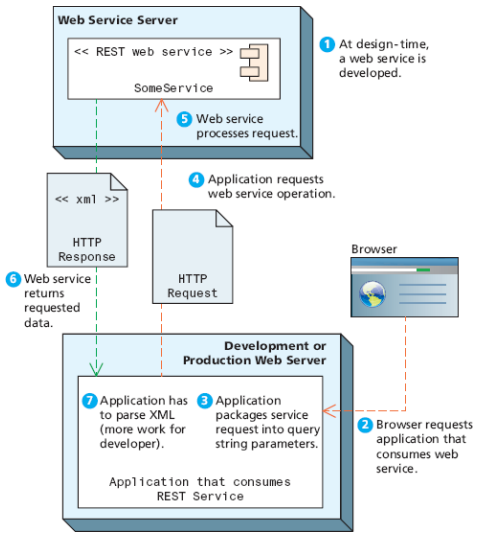
9

REST Services

- **second generation**: REST-based web services
  - **REST** stands for **RE**presentational **S**tate **T**ransfer
  - RESTful web service simply uses HTTP URLS for requesting a resource/object
  - serializing object representation is done via XML or JSON
  - early years, there has been great interest in asynchronous consumption of server data using JavaScript (i.e. AJAX)
  - today, lightweight nature of REST made it significantly easier to consume data in JavaScript versus SOAP
    - e.g. if an object is serialized via JSON, it can be turned into a JavaScript object in one simply line of JavaScript

10

REST Services (cont'd)



Randy Connolly, Ricardo Hoar, Fundamentals of Web Development (2nd Edition), 2017

11

Example of a Web Service (cont'd)

HTTP response from web service

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Date: Sun, 14 Aug 2016 19:15:54 GMT
Expires: Mon, 15 Aug 2016 19:15:54 GMT
Cache-Control: public, max-age=86400
Vary: Accept-Language
Content-Encoding: gzip
Server: mafe
Content-Length: 512
X-XSS-Protection: 1; mode=block
X-Frame-Options: SAMEORIGIN
{ "results" : [
      {
        "address_components" : [
           {
              "long_name" : "Great Russell Street",
              "short_name" : "Great Russell St",
              "types" : [ "route" ]
           },
           {
              "long_name" : "London",
              "short_name" : "London",
              "types" : [ "locality", "political" ]
           },
           …
        ],
```

```
           "geometry" : {
              …
              "location" : {
                 "lat" : 51.5180173,
                 "lng" : -0.1267183
              },
              …
           },
           …
        }
     ],
     "status" : "OK"
}
```

12

Identifying and Authenticating Service Requests

- not all web services are openly available to any request
- typically, web services employ one of the following techniques
  - **identity**
    - each web service request must identify who is making the request.
  - **authentication**
    - each web service request must provide additional evidence that they are who they say they are
- **API key**: necessary for internal record keeping and keep service request volume manageable

```
https://dev.virtualearth.net/REST/v1/Locations?o=json&query=UW%20Tacoma&key=[BING%20API%20KEY%20HERE]
```

13

Extending this Course

- **CSS**
  - Graphics
  - SVG (Scalable Vector Graphics)
  - Canvas
- **HTML**
  - HTML media (video, audio)
  - Geolocation
  - Drag/Drop
  - Web Storage

TCSS 460 - Summer 2020

14

Extending this Course (cont'd)

- **JavaScript**
  - Explore other frameworks/libraries (e.g. AngularJS, React)
- **Node.js**
  - MongoDB
  - Raspberry Pi
- **NoSQL databases**
- **Containerization** (user-space instances: Docker)
  - container-orchestration (e.g. Kubernetes)
- **Serverless Computing**
  - microservices

TCSS 460 - Summer 2020 15

Module Topics

- XML and JSON
- Databases and REST APIs

TCSS 460 - Summer 2020 16