Applications and Security

Anne Denton

Department of Computer Science
North Dakota State University

Outline

Table of Contents

Different Architectures

- Monolithic application
  - Entire application on one machine
  - For example, default installation of MS Access
- Traditional two-tier application
  - Database located on database server
  - Entire application located at client
- Three-tier application
  - Middleware layer with middleware and application program
  - Client with user interface and possibly application program
- Basic web application
  - Client, web-server, and database server separate
  - From a software perspective, it may or may not use a middleware tier

Frontend tier

- User interfaces
- Report generators
- In a two-tier application, the application layer may issue commands written in
  - SQL commands directly
  - Database client interface
- In a three-tier application user layer calls middleware functions

Middleware tier

- Interface between
  - Unprotected and unreliable user layer programs
  - And highly controlled and protected database server
- Web frameworks, like Django, contain middleware functions
- Explicit middleware applications such as web services have the primary purpose separate front- and backend
- Many modern applications have extensive middleware
  - Example: Enterprise resource planning (ERP) systems
  - Front end may not

Backend tier

- Highly protected
- Serves transaction processing functions
  - Concurrency control: Multiple transactions cannot interfere with each other or corrupt the data
  - Recovery from failure: System can recover from small or moderate failures, like interrupted transactions or power outages
  - Note that failures that involve disk storage or entire locations require contingency planning beyond a single DBMS

Question 1 (Multiple answers can be correct)

Which of the following are typical types of frontend applications?

1. Transaction processing
2. Javascript application
3. Web services

Question 2 (Multiple answers can be correct)

Which of the following are typical types of middleware applications?

1. Transaction processing
2. Javascript application
3. Web services

Question 3 (Multiple answers can be correct)

Which of the following are typical types of backend applications?

1. Transaction processing
2. Javascript application
3. Web services

Microsoft Access as Client

- Still play a role in many domains
- MS Access can be used as stand-alone database / interface (not recommended for business applications)
- Can be used as application development tool (much more secure and dependable that stand-alone)
  - Graphical User Interface builder (GUI)
  - Code modules in Visual Basic for Applications (VBA)
- Connection to database
  - Microsoft Jet database engine (allows connecting to Access database files, among others)
  - Other Database Connectivity (ODBC)

ODBC Standard

- Other Database Connectivity
- Microsoft standard for client-server interaction
- Defines types and methods
- Implementation of types depends on database system
- Each database system needs its own driver

Table of Contents

Embedded SQL

- SQL code embedded into C or other complied programming languages
- Requires prepocessor
- Requires runtime library
  - Provided by database system vendors
  - Used by many database applications
- Problems
  - Portability between database system not always guaranteed
  - Debugging tools for host language may not work well
- Uses
  - Is used for CGI (Common Gateway Interface) programs called from HTML pages (security concerns for C)
  - Very commonly used as part of the middleware

SQL integrated in scripting languages

- Scripting languages used very commonly for database interactions
  - No preprocessor needed
  - Database support often intrinsic
- Languages designed for database applications
  - PHP especially for web interactions
  - Perl shell-script-like way of accessing databases
  - Python started as language for scripting but has become general purpose language

SQL integrated in Java

- Java requires both compilation and java virtual machine JVM
- Advantage: Platform and database independence
  - JDBC: Java Database Connectivity
  - Standard Java package: java.sql
  - Implementation of java.sql interfaces depends database system
  - Driver Manager is part of application (no separate installation)
- Web interactions
  - Servlets avoid having to restart java virtual machine JVM
  - Java server pages compile into servlets

Question 4 (Multiple answers can be correct)

Which of the following languages uses compilation into machine language (of the physical machine)

1. C/C++
2. PHP
3. Java

Question 5 (Multiple answers can be correct)

For which of the following languages does an interpreter directly interpret the code that the developer writes

1. C/C++
2. PHP
3. Java

Client-side vs. server side programming

- Database connection should always be set up server side
  - Protects password and other sensitive information
- Server-side programming (suitable)
  - PHP, even when integrated into html page
    - Check source of resulting html page to see that no programmatic content is accidentally visible
  - Java servlets
  - Server side Perl or Python scripts
  - CGI (Common Gateway Interface) programs in C or C++
- Client-side programming (unsuitable)
  - Java Applets
  - Javascript

Question 6 (Multiple answers can be correct)

Which of the following uses server-side programming that is suitable for connecting to a database

1. PHP
2. Javascript
3. Java servlets
4. Java applets

Table of Contents

HTML forms

- Interaction CGI (Common Gateway Interface) applications
- The `<form ...>` tag specifies program or script that is to be called

```
<form action="/cgi-bin/program.cgi" method=
"GET"> ...  </form>
<form action="script.php" method= "GET"> ...
</form>
```

- Different types of input possible
  - Text field `<input type="text" name="name">`
  - Button `<input type="submit">`
- Form content is represented as pairs of names with associated values
- Form data is automatically encoded to satisfy requirements for URLs

Question 7 (Multiple answers can be correct)

With which of the following types of applications can be called from HTML forms

1. PHP script
2. CGI program written in C and compiled
3. Java servlet

Methods of transmitting information

- GET: content appended to URL
  - URL separated from query string by ?
  - Different name-value pairs separated by &
  - Can be typed into browser directly
  - Can be used as a link (active link)

    `http://www.google.com/search?hl=en&lr=&ie=ISO-8859-1&q=database&btnG=Google+Search`
- POST: Information sent in 2 steps
  - First set up connection
  - Then send content

Question 8 (Multiple answers can be correct)

Which of the following statements is correct in regard to using GET rather than POST for transmitting data from a form

1. GET is less secure than post
2. GET is no longer being used due to security concerns
3. GET is faster

Output of a CGI Program

- Must be MIME encoded (Multi Purpose Internet Mail Extension)
- First line of output must be a MIME content-type descriptor
  - E.g. for HTML document

    ```
    Content-type:  text/html
    ```
- Producing HTML programmatically
  - Writing strings that contain HTML tags
  - Watch for special characters
    - Translation of "<" into "&lt" etc.
    - Translation of "\n" into "
"
  - Producing tables from query results

Separating View from Model

- When servlets or cgi scripts are used, code and logic are mixed
- Designs that separate view from logic
  - JSP: Java Server Pages
  - ASP: Microsoft Active Server Pages
  - JSP pages are compiled from servlets
- Allow implementing full Model-View-Controller Architecture
  - PHP was designed such that static content is treated as for html
  - Extension of any html page can be changed to .php if installed
  - If no longer displayed, indicates problem with PHP

Question 9 (Multiple answers can be correct)

How is an HTML page sent out from a CGI program?

1. The CGI program sends out the HTML content and the programmatic content separately
2. The CGI program constructs the HTML page fully and sends it to the browser
3. When PHP is used as CGI program the HTML content is included without explicit need for print statements

Table of Contents

PHP

- PHP was developed specifically with web interactions in mind, so database access is directly integrated into the language
  - Static html content outside PHP blocks
  - Any valid html page is also a valid PHP page
  - PHP typically used through a browser, but command line interface (cli) version exists
  - No need for the printing of html tags for static html code as has to be done for cgi programs
  - Only dynamic portions have to be coded

```
<?php
echo "This portion is active content";
?>
```

- Tutorial at `http://www.w3schools.com/php/default.asp`

Basic PHP syntax

- Similarities to C-based language
  - Comments using `//` or `/* */` although `#` works too
  - Semicolons at end of statements
  - Variable names case sensitive
  - Blocks using `{}`
  - `if/elseif/else` syntax as in C
  - `while` and `for` loops as in C
- Variable names start with $

```
$txt = "Hello world!";
$x = 42;
```

- `echo` or `print` can be used to print to the screen or create HTML content
- Variables do not have to be declared
- Use `.` for string concatenation

Variables and Functions

- PHP is loosely typed
  - For example, you can add strings to numbers
  - You can change the behavior through
    ```
    declare(strict_types=1);
    ```
- Syntax of functions similar to C
  ```
  function myFunction() {
      echo "In function";
  }
  ```
- Variables can be local or global scope
  - Variables declared in functions are considered local
  - Variables declared outside functions are considered global and can only be used in functions if explicitly listed as global in the function

Arrays

- Arrays and indexed arrays
  - Created using the function `array()`
  - `foreach` allows iterating through an array
  - Explicit indexes can also be used
- Associative arrays
  - Created using, e.g., `$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");`
  - Elements addressed using key, e.g. `echo "Peter is " . $age['Peter'] . " years old.";`
  - Iterating through an associative array

```
foreach($age as $x => $x_value) {
    echo "Key=" . $x . ", Value=" . $x_value;
    echo "<br>";}
```

Processing form input

- Consider HTML form

```
<form action="printname.php" method="get">
Name:  <input type="text" name="name"><br>
<input type="submit">
```

- This input could be processed in printname.php

```
The name is <?php echo $_GET["name"]; ?><br>
```

- Note that the rest of the page `printname.php` can be plain HTML
- Because PHP programs are designed so much like HTML pages, it is realistic and not uncommon to send form data to the page itself
  - Can be done by listing the name
  - Or by giving `$_SERVER["PHP_SELF"]` as URL

Question 10 (Multiple answers can be correct)

1. Form content is submitted as key-value pairs that are then processed by a CGI program
2. An HTML form can itself be constructed using information that was submitted by a form
3. PHP can only be used to dynamically modify a form if the key-value pairs originated from the same form

Input sanitization

- Check out `https://www.w3schools.com/php/php_form_validation.asp` for suggestions on sanitizing form input
- A generally helpful function is `htmlspecialchars()` which converts special characters to their HTML representation
- Also useful is `trim()` which removes spaces, tabs, and newline characters
- `stripslashes` removes backslashes (\)
- Do check out form validation more thoroughly!

Question 11 (Multiple answers can be correct)

1. Input sanitization means that all special characters have to be removed and the user, for example, has to type "percent" instead of %
2. The PHP function htmlspecialchars() replaces special characters with their HTML representation
3. The PHP function htmlspecialchars() achieves that all characters are removed that could potentially allow injecting additional SQL queries into a query string

Connecting to a PostgreSQL database

- The connection string looks like this

```
$dbhost = pg_connect("host=hostname
dbname=databasename user=username
password=password");
```

- To avoid having the password in the file, please read host, dbname, user, and password from a file that is not in the public_html directory
- If the command didn't succeed `$dbhost` will be undefined
  - You can check this as `if(!$dbhost)` or by printing `$dbhost`
  - `pg_last_error()` has additional information
  - `die("Error: ".pg_last_error());` will print the error message and exit from the current PHP script

Question 12 (Multiple answers can be correct)

When a PHP page that connects to a database does not properly display the reason could be

1. A coding mistake in the php
2. That php is not enabled on the web server
3. Incorrect information in the connection string
4. That the database server is inaccessible
5. An error in the SQL code

Querying the PostgreSQL database

- Querying is done using the statement
  `$result = pg_query($dbhost, $sql);`
  where `$dbhost` is set above and `$sql` is a text string with the query
- If you are not using the default schema you have to include the schema name in the query
  `$sql = "SELECT * FROM Schema.Tablename";`
- `pg_fetch_array($result)`
  successively extracts records from `$result` as PHP arrays
- You can then iterate through those PHP arrays to get all attributes

Cleanup

- `pg_free_result($result);`
  frees the memory
- `pg_close($dbhost);`
  closes the connection to the database server
- For a complete example that prints out the first column of a table see
  `http://wiki.cs.ndsu.nodak.edu/doku.php?id=classes:general:dbsamples:php_postgres`

Question 13

Differences in how different programming languages connect to databases include

1. How the connection is established
2. How vendor-dependent the SQL syntax is
3. How the result string is processed

Table of Contents

Web application vs. database security

- Classic database security addresses
  - User privileges
  - Security of database for systems users
- Some of the most serious threats are related to web applications
  - Some relate specifically to databases
    e.g. SQL Injection
    `https://xkcd.com/327/`
- Many other concerns that exceed what can be covered in this course
  - Buffer overflow and similar exploits

SQL Injection

- Concept: Code injection
  - Somewhat related to code injection techniques through buffer overflow when C is used in CGI programming
- Malicious SQL segments added after existing statements
- Consider the string from the XKCD comic `https://xkcd.com/327/`
  - `Robert'); DROP TABLE Students; --`
  - When inserted into something like `INSERT INTO Students VALUES ('$name');`
  - The result is `INSERT INTO Students VALUES ('Robert'); DROP TABLE Students;--');`
  - Notice how the insertion statement is correctly closed off by the quote and parenthesis in the supposed name
  - Notice also how the actually closing off quote and parenthesis are commented out

Question 14

1. SQL injection is a database security problem that depends on the security features of the DBMS
2. SQL injection is an example of code injection
3. Unless you do your CGI programming in the C programming language, which is known for buffer overflow code injection vulnerabilities, your code will not be vulnerable to SQL injection

Use of statements that are always true

- Adding statements that are always true, like `1=1` or `' '='` `https://www.w3schools.com/sql/sql_injection.asp`
- Allows listing more entries than were requested
  - If a userid is requested, adding `OR 1=1`
    ```
    SELECT userid, password FROM Users WHERE userid = 105 OR 1=1;
    ```
- Bypassing security checks
  - If a password is requested, adding `'OR ''='`
    ```
    SELECT * from Users WHERE name = ''OR ''=''AND pass = ''OR ''='';
    ```

Tools

- Typically vulnerabilities are detected with tools
  - Tools can scan many combinations
  - The user cannot know what exactly is done with the string that is supplied by the user
- `sqlmap` is a powerful tool `http://sqlmap.org/`
  - Make sure not only use with prior mutual consent!
  - When a vulnerability is found, huge quantities of data can extracted
  - A single vulnerability is enough

Preventing vulnerabilities

- In PHP “Prepared statements" work well
  - They replace problematic characters
  - Number of variables that can be supplied is specified at the time the program is written
- Syntax depends on DBMS
  - PostgreSQL `https://www.php.net/manual/en/function.pg-prepare.php`
  - MySQL `https://www.w3schools.com/php/php_mysql_prepared_statements.asp`
- Do not confuse with database prepared statements!
  - Intended solely for performance and tend not to help against injections
- Make use of PHP prepared statements!

Table of Contents

Reasons for using Python

- PHP is good for web interactions but not as suited for processing data
- Python has libraries that allow anything from basic statistics, plotting, and signal processing to the newest deep learning
- Some of the most important libraries
  - NumPy: Allows array-based processing similar to Matlab
  - Pandas: Allows statistics processing similar to R and SAS
  - Matplotlib: Allows plotting similar to Matlab
  - Tensorflow: Google's deep learning library
- Python's success is largely due to being several languages in one

Properties of Python

- High-level language
  - One statement typically does a lot
  - Instead of using loops with statements that are executed thousands or millions of times, use array-based processing or other libraries programmed in C or other low-level languages
  - Typical applications, like the Geographic Information Systems, use C for low-level programming and Python for the user interface and high-level instructions
- Interpreted / scripting language
  - Run programs using
    `python3 fn.py`
- Open source language

Basic Python syntax

- Does not consistently use C-style syntax (much less so than PHP)
- No semicolons at the end of lines
- Indentation rather than {}s defines blocks
  - Nice for clarity, but you have to be very careful when changing text editors
- It has interactive environments
- IDLE or Jupyter Notebook
  - Similar to Matlab and R

More basics

- Assignment of values to variables does follow C-based standards

```
a = 5
a = a+3
print(a)
```

- Comparisons use (==) as in C-based languages, but syntax of conditions is different

```
if (a == 8):
    print("yes")
```

Python 2 vs. 3

- The language changed substantially between version 2 and 3
  - Syntax for writing to the screen
  - Syntax for division
  - In Python 2 you write `print 5/2` and it returns `2`
  - In Python 3 you write `print (5/2)` and it returns `2.5`
- At this point you should use Python 3
  - The “sunset date" was January 1, 2020
  - That means security holes are no longer fixed for Python 2

Python virtual environments

- Because libraries are so important to Python, it is notorious for inconsistent installations
- It has become common that users of multi-user systems create their own custom installation
- Please refer to the following tutorial `https://realpython.com/python-virtual-environments-a-primer/`
- If you type the following in the lab, a virtual environment will be created for you
  `python3 -m venv /python_virtual_environment`
- For more details check
  `https://docs.python.org/3/library/venv.html`

Data types

- Variables do not have to be declared
- Uses strong typing
  - You cannot add `5 + '5'` as you can, by default, in PHP
- Lists: `squares_list = [0, 1, 4, 16]`
  - `squares_list[2] is 4`
  - You can change individual list elements
  - Counting starts at 0 as in most modern languages!
- Strings: `hello = "Hello!"` (single quotes work too)
  - You extract individual letters
  - You cannot change individual letters (strings are immutable)
- Tuples: thistuple = ("apple", "banana", "cherry")
  - Important for inserting and retrieving from databases!
- Dictionaries: offices = `{"Denton":28, "Ludwig":22}`
  - Key-value pairs
  - Like associative arrays in PHP

- Example factorial:

```
n = 5
factorial = 1
for i in range(1,n+1):
    factorial *= i
    print(factorial)
```

- `range` takes `beginning` and `end+1` as arguments
  - Why n+1?
  - Goes through loop n times
  - `range(0,n)` is 0, 1, 2, ... n-1
- You can iterate over any “iterables" including lists

```
for animal in ["cat", "mouse"]:
    print(animal)
```

Packages

- Some packages are standard, and needed for most of what you do
  - math, sys

```
import math
n = 5
math.factorial(n)
```

- You could also write `import math as m`
  or
  `from math import factorial`
- Numpy arrays are also standard in many scientific applications

Numpy arrays

- Python itself does not have arrays but just lists
- Lists can have different elements but arrays are expected to all have the same datatype
- Numpy arrays fit many scientific applications
- Allow array-based processing

```
import numpy as np
a = 3*np.ones(5)
b = np.array(range(0,5))
print(a+b)
```

Table of Contents

Database connectivity

- We will use the library `psycopg2`
- This library is already installed in the lab
- The following tutorial is quite extensive
  `https://www.postgresqltutorial.com/postgresql-python/connect/`
- The connection string is
  ```
  conn = psycopg2.connect("host=hostname dbname=databasename user=username password=password")
  ```
- Make sure that the file from which you connect to the database does not have read privileges for “all" and that it is not in public_html or a subdirectory thereof

Using psycopg2

- You can find the basic syntax of psycopg2 at `https://www.psycopg.org/docs/usage.html`
- I recommend that you create tables using psql since you only need to do that once
- You probably want to do insertions and queries using psycopg2 since that gives you more flexibility
- Notice that you will not see changes to the database until you commit change!

Security of prepared statements in Python

- The cursor execute statement in psycopg2 automatically implements the more secure prepared statement logic discussed for PHP
- That also means that you do not have to use the single quote string notation in SQL
- Neither do you need to escape single quotes
- It uses the tuple notation that is intrinsic to standard Python!

Working with multiple records at once

- You can insert multiple records at once using psycopg2's executemany
- The tutorial at `https://www.postgresqltutorial.com/postgresql-python/query/` gives you more details
- When you retrieve tables, by default, you will iterate through tuples row by row
- Many data processing libraries, like Pandas, represent data through column-oriented data frames
- Conversion straightforward `https://www.geeksforgeeks.org/creating-a-pandas-dataframe-using-list-of-tuples/`