# Java, JDBC, Perl, CGI Embedded SQL

November 3rd, 2000

## What will we cover ?

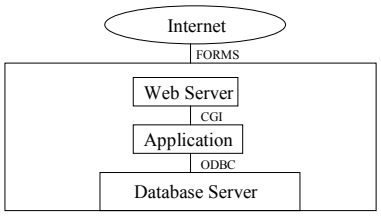
- Assignment 4 – Java and JDBC
- Course project – HTML, PERL, CGI, Embedded SQL in Perl

## How to run a query ?

- Running a query:
  - Query Analyzer (assignment 3)
  - Any application (assignment 4)
  - Web application (project)

## Modern 3-tier architecture

- Web server, application, database




## Application ?

- What is it ?
  - Software that formally separates and coordinates:
    - Data sitting in the database
    - Application logic
    - Web accesses
- What it does ?
  - Connects either a web server or users to the database.

## Why do we need applications ?

- Easier to develop and maintain
- We can reuse components for and application services:
  - Caching, encryption, managing database connections
- Multiple databases
- Security

## Back to Assignment 4

- This is NOT a programming course.
- Tutorial on the web – straightforward
- Steps:
  1. Add an ODBC connection on your machine
  2. Create a new Java Application (Console Application)
  3. Download the example code provided

## JDBC/ODBC

- **Standardized network database APIs**
  - Connect, submit queries, process results, disconnect
  - Abstract application from particular database
- **ODBC from Microsoft, JDBC from Sun**
- **A good driver can make a huge difference**
  - Pre-fetching
  - Data caching
  - SQL translation caching
- **Could spend hours talking about JDBC/ODBC**

## Key Code

```
Class.forName("com.ms.jdbc.odbc.JdbcOdbcDriver");
java.sql.Connection c = DriverManager.getConnection
                        ("jdbc:odbc:444","cse444","WeRock!");
java.sql.Statement s= c.createStatement();
java.sql.ResultSet rs;
rs = s.executeQuery("Select * from movies");
java.sql.ResultSetMetaData md = rs.getMetaData();
while (rs.next()){
        area.append("\nTUPLE: |");
        for (int i = 1; i <= md.getColumnCount();i++){
            area.append(rs.getString(i) + " | ");
         }
}
```

- XML ?

## HTML/CGI

- How does the Web works ?
  - Send a request to a server (cubist)
  - Servers answers back with HTML text
  - Browser parses text and outputs the page
- What if the request involves running an application on the server ?
  - Client must have a FORM in the HTML file
  - FORM indicates what action the server must take

## FORMS

```
<form ACTION=
"http://cubist.cs.washington.edu/
        ~tzoompy/444_2.pl" method=POST>
<P>Input a last name (try 'Gore'):
<INPUT NAME=last_name></P>
<P><INPUT TYPE=SUBMIT></p>
</form>
```

## Perl

- Very successful and popular
- Easy for managing data (files or streams)
- Also popular for using as the CGI language of choice
- Examples:
  - print “Hello World !\n”;
  - while, for, if, very standard….
  - $lname =~ tr/a-z/A-Z/;
  - $lname =~ s/Levy/Halevy/;

## Key Code

```
#!/usr/bin/perl5 -w
use CGI;
use DBI;
$html = new CGI;
print $html->header, "\n",
   $html->start_html();
&searchDatabase;
print $html->end_html;
exit(0);
```

## Key Code (2)

```
sub searchDatabase {
  $sql = "SELECT * FROM Elections2000";
  $db = DBI->connect("dbi:Sybase:server=ISQL01",
        "cse444", "WeRock!");
  $db or die "Sorry we couldn't connect!\n";
  $sth = $db->prepare($sql);
  $sth->execute();
  while ($row = $sth->fetchrow_hashref()) {
            $rec1 = $row->{lname};
            print "Record  = $rec1<p>";
     }
}
```

## User Input

- Check the tutorial and examples available on the web
- Once all this is done you are 75% done with project
- More help ?
  - Google (or your favorite engine)
  - Java ? – I don't know
  - Perl ? – Lama and Camel books (O'Reilly)
  - HTML/CGI ? – Anything, it's easy anyway

## Miscellaneous

- Assignment 4:
  - output – stdout preferred, but anything reasonable
  - queries – do your own or get solutions
  - two programs only
  - code must be reasonable documented
- Project
  - black-box testing (including stress test)
  - why not Java servlets/ASP/Java/Assembly…
    - cse security policy, but…