



CSC309: Introduction to Web Programming

Lecture 13

Wael Aboulsaadat



Web-apps Architecture



N-Tier model + MQ + SP

Presentation Layer
HTML/CSS/JS/JSP/ ...

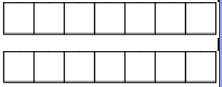
Presentation Layer
HTML/CSS/JS/JSP/...

Presentation Layer
HTML/CSS/JS/JSP/ ...

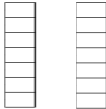
Logic Application 1
Java/Python/...



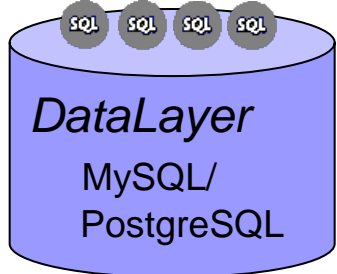
Logic Application 2
Java/Python/...



Logic Application 3
Java/Python/...

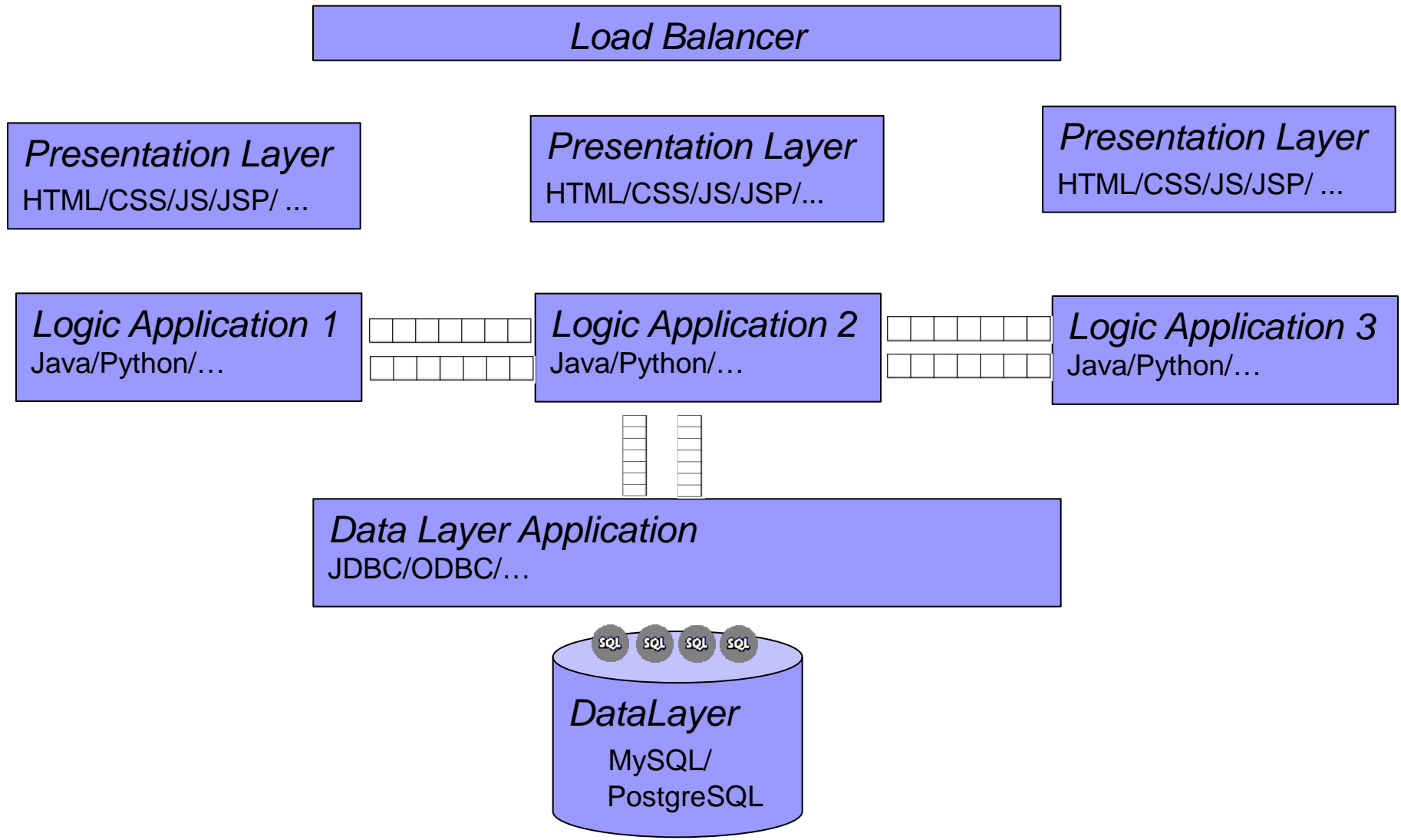


Data Layer Application
JDBC/ODBC/...





N-Tier model + MQ + SP





N-Tier model: cons?

- Complex Architecture requiring administration and maintenance
- Dedicated server or hosting
- Excessive data transformation (XML to XML to Objects to ER to XML...)
- Asynchronous vs. Synchronous problem

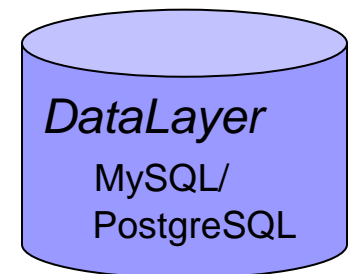
3-Tier model

■ How it works?

- Use a high level language to encode application logic
- Use Scripting language in GUI
- Use a database to store and manipulate data

Presentation Layer
HTML/CSS/JS/JSP/...

Logic Layer
Java/Python/...





3-Tier model – e.g.

■ .jsp

```
<%@page contentType="text/html"%>
<jsp:useBean id="result" class="DBQuery"
            scope="page"/>

<html>
<head> <title> </title> </head>
<body>
<h1>This is a JSP that queries MySQL usingJDBC</h1>
<hr>
<table border=2 cellpadding=5>
<tr>
    <th>Movie Id</th>
    <th>Name</th>
</tr>
```

```
<%
Data data;
while (data= result.nextRow())
{
%>
    <tr>
        <td><%= data.getID()%></td>
        <td><%= data.getMovieName()>
        </td>
    </tr>
<%
}
%>
</table>
</body>
</html>
```

3-Tier model – e.g.

■ .java

```
public class DBQuery {
    private ResultSet rs;

    public DBQuery( ) {
        Class.forName("com.mysql.jdbc.Driver").newInstance();
        Connection conn= DriverManager.getConnection(
            "jdbc:mysql://localhost:3306/jsp","root","");
        Statement stmt = conn.createStatement();
        rs = stmt.executeQuery("select * from MOVIES");
    }

    public Data nextRow( ) {
        Data data;
        if (rs.next() != null )
            data = new Data( rs.getInt("id"),
                rs.getString( "movie_name" ) );
        return data;
    }
}
```

```
public Data {
    private String Movie;
    private int ID;

    public Data( int ID,
        String Movie ){
        this.Movie = Movie;
        this.ID = ID;
    }

    public getMovieName( ) {
        return Movie;
    }

    public getID( ){
        return ID;
    }
}
```

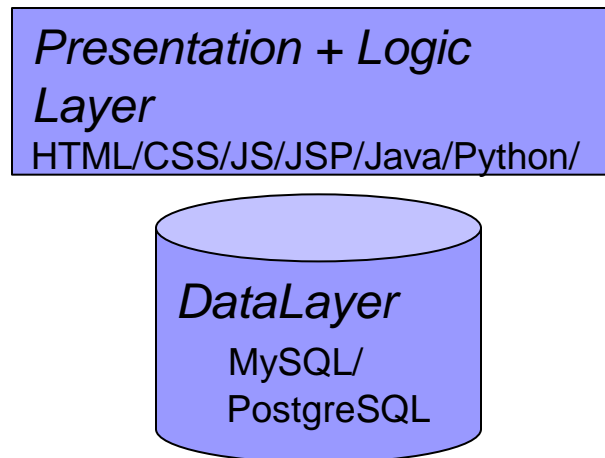



3-Tier model: cons?

- Opportunity for misuse
- Data classes could be exposed at interface level – introducing coupling
- Problems with Scalability

Template-driven model

- How it works?
 - Every .html becomes a template
 - Embed scripting code inside each page template. The scripting code will do all GUI + logic functionality.
 - Upon the page request, each template is processed yielding an HTML page





Template-driven model

■ Example:

```
<%@page contentType="text/html"%>
<html>
<head> <title> </title> </head>
<body>
<%@ page import="java.sql.*" %>
<h1>This is a JSP that queries MySQL usingJDBC</h1>
<hr>
<%
    Class.forName("com.mysql.jdbc.Driver").
        newInstance();
    java.sql.Connection conn;
    conn= DriverManager.getConnection(
        "jdbc:mysql://localhost:3306/jsp","root","");
    Statement stmt = conn.createStatement();
    ResultSet rs = stmt.executeQuery
        ("select * from MOVIES");
%>
```

```
<table border=2 cellpadding=5>
<tr>
<th>Movie Id</th>
<th>Name</th>
</tr>
<%
    while (rs.next())
    {
%>
        <tr>
            <td><%= rs.getInt("id")%></td>
            <td><%= rs.getString(
                "movie_name"); %>
            </td>
        </tr>
    <%
    }
    rs.close();
    stmt.close();
    conn.close();
%>
</table>
</body>
</html>
```



Template driven model: cons?

- One file for both: graphic designers and developers
- Logic code is dependent on specific location in file it is inserted in.
- Tricky to “craft”
- Resource cruncher – because it requires some underpinning applications



Parameters....

- Traffic
- Database Requirements
- Computational Requirements
- Website Content
- Budget

Traffic – e.g.

the amount of data sent and received by visitors to a web site.

- Page-views/page-impressions:
 - Number of times website pages were downloaded by Internet users

- Page-Views/User
 - Average page-views per user

- Time on Site:
 - Average time per user

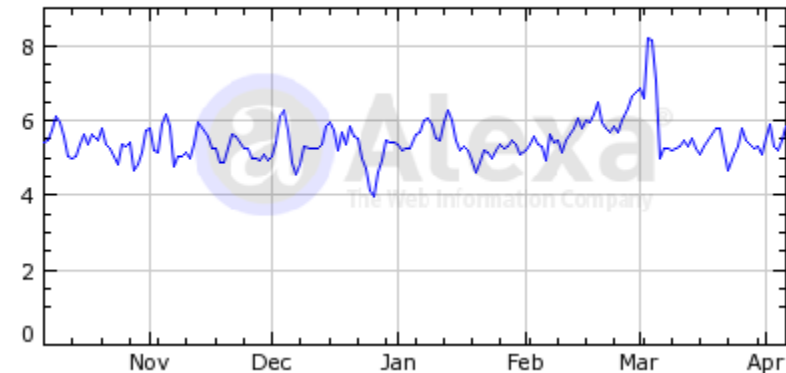
- Reach:
 - Percent of global Internet users who visit your website

Traffic – e.g.: tsn.ca

- Page-views:
 - 1.2 million page-views/day

Time on Site (minutes)

tsn.ca



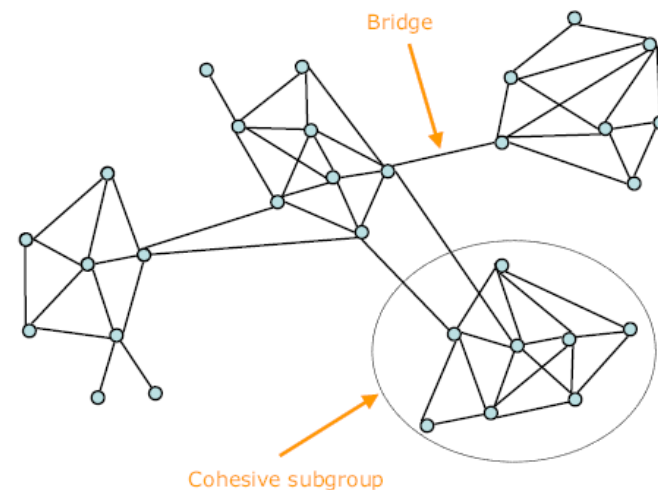
Daily Pageviews (percent)

tsn.ca



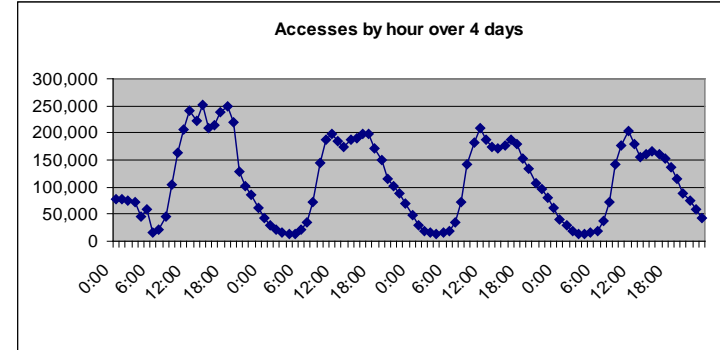
Traffic

- Pattern: long term
- Why?
 - Network effect
 - News item in the media
 - A quickly propagating email
 - Link from a popular site
 - Slashdot effect



Traffic

- Pattern: daily



ManheimAuctions.com web traffic analysis

- Why?
 - Business regulations
 - Major events: natural disasters, 9/11, NBA finals,....



Computational Requirements

- Is the core service offered by the website offline/online?
- Is the website offering several mutually exclusive services?
- Are there computationally expensive operations?



Database Requirements

- Database Performance
 - DBMS benchmarks

- Stored Procedures, Triggers, and Rules
 - SQL-89, SQL-92, and SQL3

- Backup
 - Hot/cold
 - Raw disk partition or OS file-system

- Security



Website Content

- How many static/dynamic pages in the website ?
- How much of the website is public/private space?
- What is the average content size? (web-pages, video, etc...)



Budget: hosting

- Internet infrastructure and Website hosting
 - Backbone maps
 - <http://www.nthelp.com/maps.htm>
 - <http://www.caida.org/tools/visualization/mapnet/Backbones/>

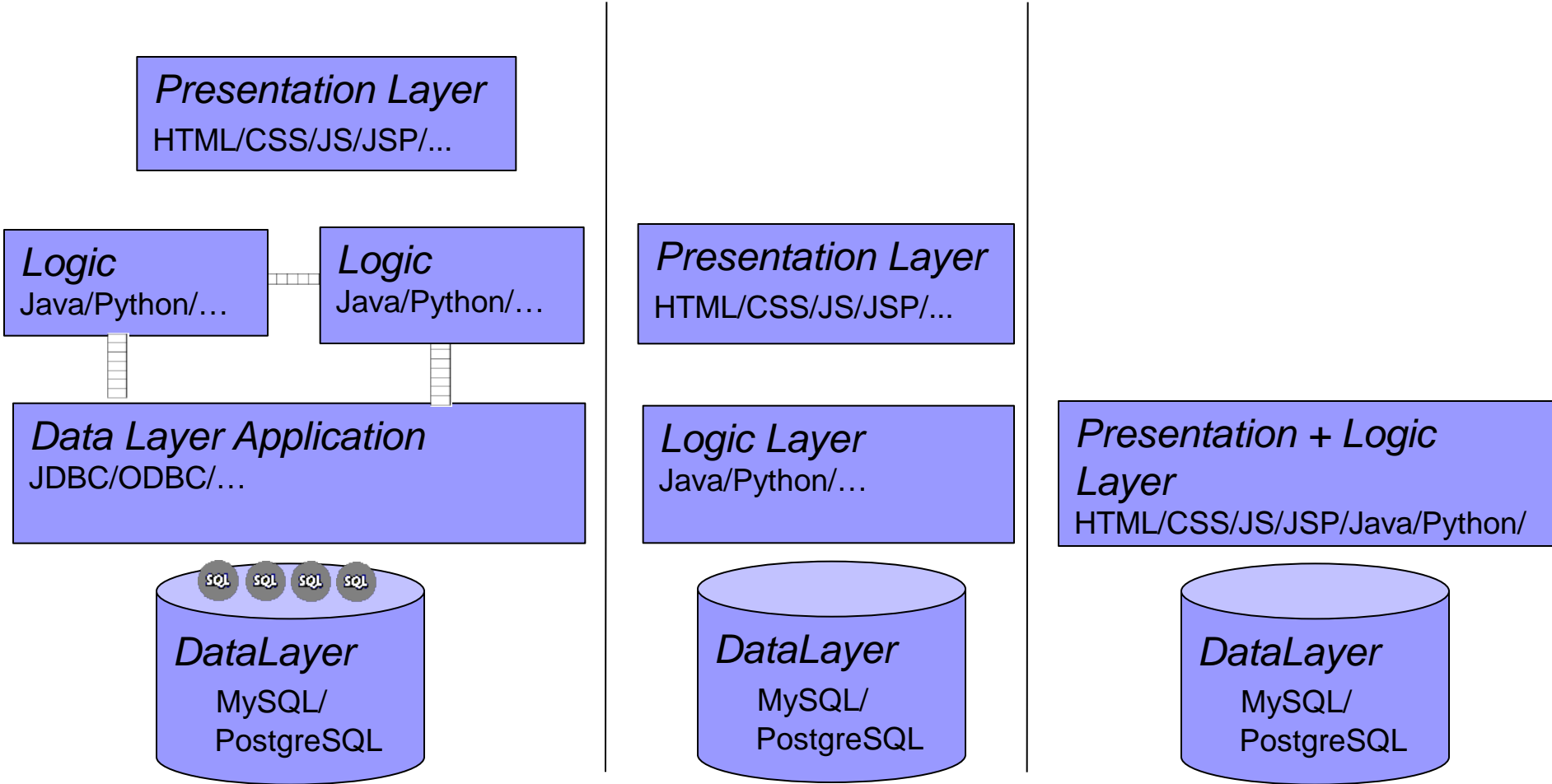
Budget: hosting

- Virtual Hosting:
 - Infrastructure: standby electric power, redundant servers, redundant communication lines,
 - Options: Shared, Dedicated, Backbone hosting

- Onsite:
 - Connection from backbone providers
 - Bandwidth:
 - T1 (1.5 Mb) is \$500 – \$1200/mth
 - T3 (28 T1 - 45 Mb) is ~\$3K/mth
 - Hardware:
 - Servers: 3K – 100K
 - Network Hardware: \$10K – 40K



3 models: which one?





N Tier model + MQ + SP: when?

- Traffic
 - High (millions of page-views/day).
- Database requirements:
 - Large number of tables, complex queries/updates, transactions support
 - DB devs (vs. DB admins).
 - Data handling and security has legal implications
- Processing requirements:
 - Independent applications yielding a consolidated view
 - One computationally intensive application → replicate the application
- Examples:
 - Bank Statement generation (mortgage application, credit card application, checking-account application, savings-account application, etc...)
- Budget:
 - Development budget: \$00,000, Hosting cost: \$Ks/mth



N Tier model + MQ + SP: tools?

■ Typical technologies

- Operating System: Linux, IBM mainframe AIX, Tandem Nonstop
- Application Servers: Tomcat, IBM WebSphere, Apache,...
- Languages: Java/PhP/Python/Perl/C/C++
- Database: Oracle, IBM DB/2, Sybase SQL Server,....
- Message Brokers: Swift MQ, Sonic MQ, FioranoMQ,....
- Load Balancers: F5 BigIP, Foundry ServerIron, Cisco LocalDirector,...
- Routers: Alcatel OBX, Chiaro Enstara,....
- Libraries: XML tools + many other application specific libraries
- Load simulation: Load Runner, SOAPtest, Passler Stress,....
- Profiling: JProbe, Tomcat Probe,....



3-Tier model: when?

- Traffic
 - Moderate to high (few hundred thousands page-views/day).
- Database:
 - Significant number of tables, moderate queries/updates
- Processing requirements:
 - Single application with multiple services
- Examples:
 - Online Auction
- Budget:
 - Development budget: \$00,000
 - Hosting cost: moderate



3-Tier model: tools?

- Typical technologies
 - Operating System: Linux, Windows,...
 - Web Server: Apache, MS IIS,...
 - Application Servers: Tomcat, IBM WebSphere, .Net
 - languages: Java, PHP, Python, Perl, C#, VB,...
 - Database: PostgreSQL, MS-MS SQL Server, Oracle,...
 - Load simulation: LoadSim, WAPT,....
 - Profiling: JProbe, Tomcat Probe,....



Template driven model: when?

- Traffic:
 - Small to moderate

- Database:
 - small number of tables, simple queries/updates,

- Processing requirement:
 - Mostly generating HTML pages resulting from SQL result sets

- E.g.
 - Doctor's Clinic, grocery store, lawyer's office software, etc...
 - Discussion Boards, e-commerce websites, news websites, Blogs,...

- Budget:
 - Development budget: limited, Hosting cost: small to moderate



Template-driven model: tools?

■ Typical technologies

- Operating System:
- Web Server:
- Language:
Perl,...
- Database:

Examples

Windows, Linux

MS IIS, Apache,...

ASP.NET family, PhP, Python,

MySQL, SQL-Server,...

and possibly:

- Content Management System:
- Ecommerce framework:

Joomla/Jaws/Drupal(PhP),
Plone(Python),...

ZenCart/VirtueMart/Ubercart(PhP)..



Exam