



## 第9章:应用程序设计和开发

数据库系统概念, 6<sup>th</sup>编辑。

©Silberschatz, Korth和Sudarshan参见[www.db-book.com](http://www.db-book.com)了解  
重用条件

---



# 第9章:应用程序设计和开发

“用户界面和工具”

“数据库的Web接口” “Web基础”

servlet和JSP

“构建大型Web应用程序”



# 用户界面和工具

大多数数据库用户不使用像SQL这样的查询语言。z形式

z图形用户界面

许多界面都是基于web的

后端(Web服务器)使用z Java servlet等技术

z Java服务器页面(JSP)

z 活动服务器页面(ASP)



# 万维网

Web是基于超文本的分布式信息系统。大多数网络文档是通过超文本标记语言(HTML)格式化的超文本文档

HTML文档包含

z文本连同字体规范，以及其他格式说明z超文本链接到其他文档，这可以与文本的区域相关联。

z表单，允许用户输入数据，然后将其发送回  
Web服务器



# A formatted report

## Acme Supply Company, Inc. Quarterly Sales Report

Period: Jan. 1 to March 31, 2009

Region	Category	Sales	Subtotal
North	Computer Hardware	1,000,000	1,500,000
	Computer Software	500,000	
	All categories		
South	Computer Hardware	200,000	600,000
	Computer Software	400,000	
	All categories		
Total Sales			2,100,000



# 数据库的Web接口

为什么要将数据库连接到Web?

1. Web浏览器已经成为数据库事实上的标准用户界面

- z 使大量用户可以从任何地方访问数据库
- z 避免需要下载/安装专门的代码，同时提供良好的图形用户界面
- z 例子: 银行、航空公司和租车预订、大学课程注册和评分等等。



# 数据库的Web接口(续)

## 2. 动态生成文档

### 静态HTML文档的限制

不能为个人用户定制固定的Web文档。更新Web文档有问题，特别是当多个Web文档复制数据时。

z 解决方案:从存储在数据库中的数据动态生成Web文档。

可以根据数据库中存储的用户信息定制显示。

- 例如，量身定制的广告，量身定制的天气和本地新闻，…显示的信息是最新的，不像静态网页

- 例如，股票市场信息，…



# 统一资源定位器

在Web中，指针的功能由统一资源定位器(url)提供。

URL 示 例 :<http://www.bell-labs.com/topics/book/db-book>

---

z 第一部分表示如何访问文档

" http "表示要使用超文本传输协议访问该文档。

z 第二部分给出了互联网上一台机器的唯一名称。

z URL的其余部分标识机器中的文档。 “本地标识可以是:

机器上文件的路径名, 或者

程序的标识符(路径名), 加上要传递给程序的参数

-例:, <http://www.google.com/search?q=silberschatz>

---





# HTML和HTTP

HTML提供格式、超文本链接和图像显示功能。HTML还提供了输入功能

从一组选项中进行选择

-弹出菜单，单选按钮，检查列表  
输入值

语境的盒子

z填写发送回服务器的输入，由服务器上的可执行文件执行

用于与Web服务器通信的超文本传输协议(HTTP)



# 示例HTML源文本

< html >

身体< >

<表边境>

< tr > < th > ID < / th > < th > 名称 < / th > < th > 部门 < / th > < / tr >

< tr > < td > 00128 < / td > < td > Zhang < / td > < td > Comp。 科学。 < / td > < / tr >

...

表> < /

< form action = "PersonQuery" method = get >

搜索:

< 选择 name = " persontype " >

< option value = " student " 选定 > student < / option >

< option value = " instructor " > instructor < / option >

< 选择 > / < br >

Name: < input type = text size = 20 Name = " Name " >

< input type = submit value = " submit " >

> < / 形式

< / 身体 > < / html >



# 显示HTML源样本

ID	Name	Department
00128	Zhang	Comp. Sci.
12345	Shankar	Comp. Sci.
19991	Brandt	History

Search for:

Name:



# 客户端脚本和小程序

“浏览器可以获取某些脚本(客户端脚本)或程序以及文档，并在客户端站点以“安全模式”执行它们

- z Javascript

- z Macromedia Flash和Shockwave用于动画/游戏
- z VRML

- z 小程序

客户端脚本/程序允许文档处于活动状态，例如，通过在本地站点执行程序来生成动画

- z 例如，确保用户输入的值满足一些正确性检查

允许与用户进行灵活的交互。

在客户端站点执行程序，避免了与服务器的多次往返，从而加快了交互速度



# 客户端脚本和安全

确保恶意脚本不会对客户端机器造成损害所需的安全机制

对于功能有限的脚本语言来说很容易，而对于Java等通用编程语言则比较困难

例如，Java的安全系统确保Java applet代码不直接进行任何系统调用

禁止危险的操作，比如写文件

通知用户潜在的危险操作，并允许选择中止程序或继续执行。



# Web服务器

Web服务器可以很容易地作为各种信息服务的前端。

**URL**中的文档名称可以标识一个可执行程序，该程序在运行时生成一个HTML文档。

当HTTP服务器接收到对这种文档的请求时，它执行该程序，并将生成的HTML文档发回。

Web客户端可以传递带有文档名称的额外参数。

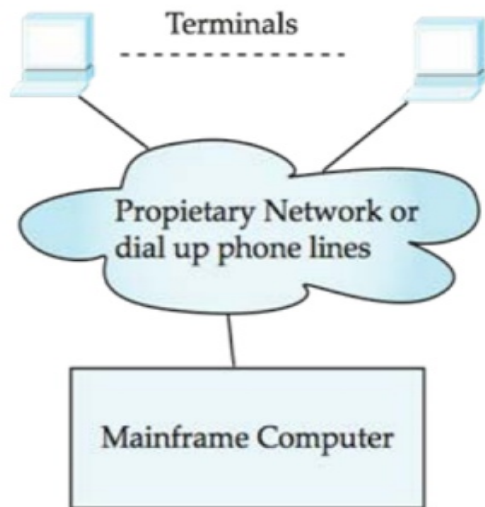
要在Web上安装一个新服务，只需要创建并安装一个提供该服务的可执行文件。

z Web浏览器为信息服务提供图形用户界面。

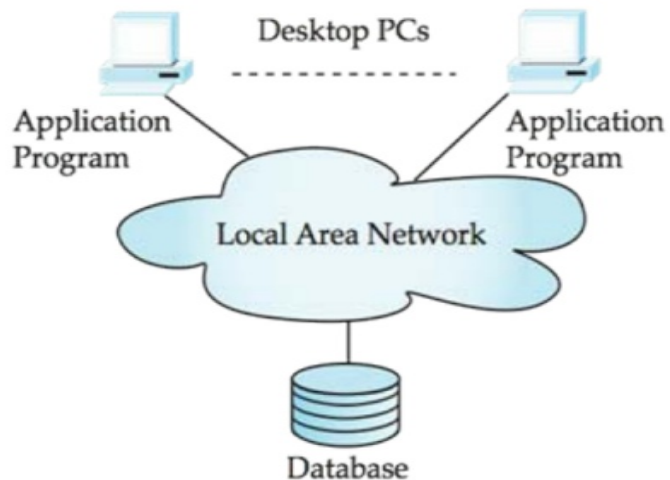
通用网关接口(CGI): web和应用服务器之间的标准接口



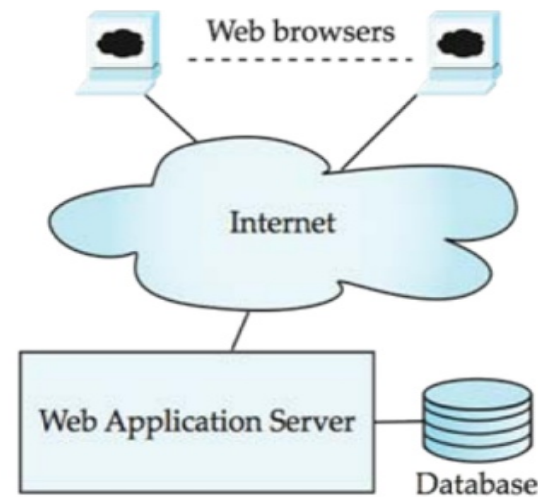
# Three-Tier Web Architecture



(a) Mainframe Era



(b) Personal Computer Era

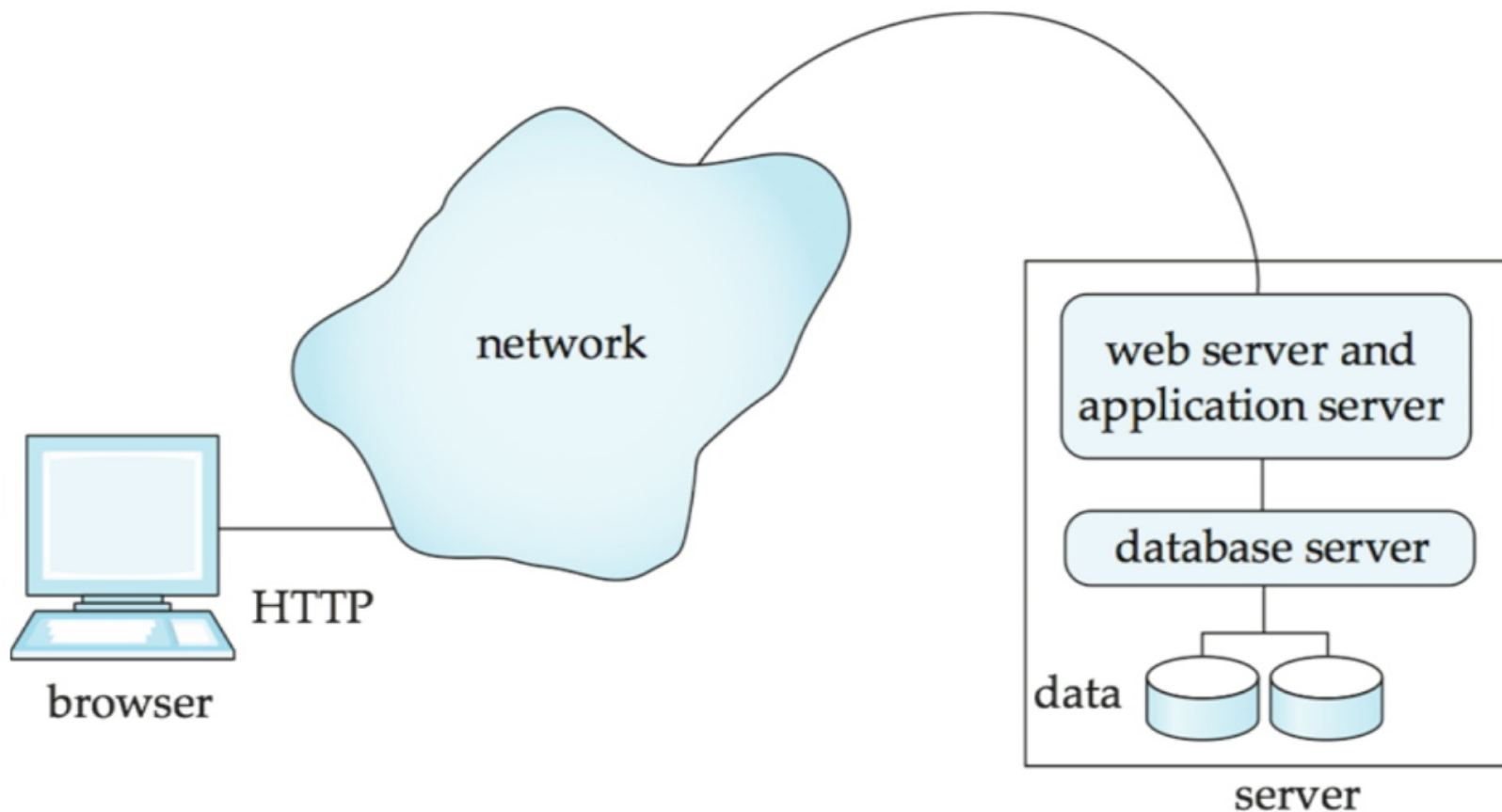


(c) Web era



# 双层Web架构

“多层间接有开销可选:两层架构







# HTTP和会话

☒ HTTP协议是**无连接**的

z也就是说，一旦服务器响应了一个请求，服务器就关闭了与客户端的连接，并且忘记了这个请求

z相反，Unix登录和JDBC/ODBC连接一直保持连接，直到客户端断开连接

保留用户身份验证和其他信息z动机:减少服务器负载

操作系统对机器上打开的连接数量有严格的限制

信息服务需要会话信息

z例如，用户身份验证应该在每个会话中只进行一次

解决方案:使用**cookie**



# 会话和cookie

“cookie” 是包含识别信息的一小段文本

- z 在第一次交互时由服务器发送给浏览器

- z 在进一步交互时由浏览器发送给创建cookie的服务器

HTTP协议的一部分

- z 服务器保存有关它发出的cookie的信息，并且可以在服务请求时使用它

例如，认证信息和用户首选项

“cookie”可以永久存储，也可以在有限的时间内存储



# servlet

**Java Servlet**规范定义了一个用于Web服务器和应用程序之间通信的API

例如，获取参数值和将HTML文本发送回客户端的方法

应用程序(也称为servlet)被加载到Web服务器中

- z 两层模型

- z 每个请求在Web服务器中生成一个新线程线程在请求被服务后关闭

**Servlet** API提供了一个getSession()方法

- z 在与浏览器的第一次交互时设置cookie，并在进一步的交互中使用它来识别会话

- z 提供了存储和查找每个会话信息的方法

例如，用户名，首选项，...



# 示例Servlet代码

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class PersonQueryServlet extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException
    {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<HEAD><TITLE> Query Result</TITLE></HEAD>");
        out.println("<BODY>");
        ..... BODY OF SERVLET (next slide) ...
        out.println("</BODY>");
        out.close();
    }
}
```



## 示例Servlet代码

```
字符串persontype = request.getParameter( "persontype" );
String number = request.getParameter("name");
if(persontype.equals("student")) {
    ... code to find students with the specified name ...
    ... using J DBC to communicate with the database ..
    out.println("<table BORDER COLS=3>");
    out.println(" <tr> <td>ID</td> <td>Name: </td>" + " <td>Department</td> </tr>");
    for(... each result ...){
        ... retrieve ID, name and dept name
        ... into variables ID, name and deptname
        out.println("<tr> <td>" + ID + "</td>" + "<td>" + name + "</td>" + "<td>" + deptname
            + "</td></tr>");
    };
    out.println("</table>");
}
else {
    ... as above, but for instructors ...
}
```



# Servlet会话

要检查会话是否已经激活:

```
z if (request.getSession(false) == true)
```

..然后现有会话

else ..重定向到认证页面

z认证页面

检查登录/密码

request.getSession(true):创建新会话

“存储/检索特定会话z会话的属性值对。setAttribute( “标识” ,用户标识)

```
z session.getAttribute( “标识” )
```



# java服务器页面(JSP)

- ☒ 一个内嵌Java代码的JSP页面

```
< html >
```

```
<head> <title> Hello </title> </head>
```

```
身体< >
```

```
< % if (request.getParameter(" name ") == null)
```

```
{。 println( “Hello World” );}
```

```
Else {out。 println(" Hello,  " + request.getParameter(" name "));} % >
```

```
身体< / >
```

```
< / html >
```



# 服务器端脚本

服务器端脚本简化了将数据库连接到Web的任务

定义一个包含嵌入式可执行代码/SQL查询的HTML文档。

来自HTML表单的输入值可以直接用于嵌入代码/SQL查询。

当文档被请求时，Web服务器执行嵌入的代码/SQL查询来生成实际的HTML文档。

大量的服务器端脚本语言

JSP, 服务器端Javascript, ColdFusion 标记语言 (cfml), PHP, Jscript

z 通用脚本语言: VBScript, Perl, Python





# I提高Web服务器性能

\*性能是流行网站的一个问题

z每天可能有数百万用户访问，高峰时每秒有数千个请求

☒ 缓存技术用于通过利用请求之间的共性来降低服务页面的成本

z在服务器站点:

缓存servlet请求之间的JDBC连接

缓存数据库查询的结果

-如果底层数据库发生变化，缓存的结果必须更新

生成的HTML的缓存

z在客户端的网络

通过Web代理缓存页面



# App 保护

当数据库授权规定不能提供足够的保护时，数据可能被*加密*。

良好的加密技术的特性：

- z 对授权用户加密和解密数据相对简单。
- z 加密方案不依赖于算法的保密性，而是依赖于算法中一个被称为加密密钥的参数的保密性。
- z 对于入侵者来说，确定加密密钥极其困难。



# 加密(续)。

**数据加密标准(DES)**在通过安全机制提供给授权用户的加密密钥的基础上替换字符并重新排列其顺序。方案并不比密钥传输机制更安全，因为密钥必须被共享。

**高级加密标准(AES)**是取代DES的新标准，它基于Rijndael算法，但也依赖于共享密钥。

“**公钥加密**是基于每个用户拥有两个密钥：

- z**公钥——公开发布的密钥，用于加密数据，但不能用于解密数据

- z**私钥——仅为个人用户所知的密钥，用于解密数据。不需要传输到做加密的站点。加密方案是这样的，它是不可能或极难解密的数据只给公钥。



RSA公钥加密方案是基于将一个非常大的数字(100位数字)分解成其质数分量的难度。



# 身份验证

基于密码的身份验证被广泛使用，但在网络上容易被嗅探。



**质询-响应**系统避免密码传输:DB向用户发送一个(随机生成的)质询字符串。

z 用户加密字符串并返回结果。

z DB通过解密结果验证身份

z 可以使用公开密钥加密系统，通过DB发送使用用户的公开密钥加密的消息，然后用户解密并发回消息。

使用**数字签名**来验证数据的真实性

z 例如，使用私钥(反向)对数据进行加密，任何人都可以通过使用公钥(反向)对数据进行解密来验证真实性。只有私钥的持有者才能创建加密的数据。

数字签名还有助于确保**不可抵赖性**:发送方以后不能声称没有创建数据



# 数字证书

- ❑ **数字证书**用于验证公钥的真实性。
- ❑ 问题:当你和一个网站交流时,你怎么知道你是在和一个真正的网站交谈还是一个冒名顶替的网站?
  - z 解决方案:使用网站的公钥
  - z 问题:如何验证公钥本身是否为正品?
- ❑ **解决方案:**
  - z 每个客户端(例如浏览器)都有几个根级**证书颁发机构**的公钥
  - z 网站可以获得由认证机构签名的名称/URL和公钥:签名的文件称为**证书**
  - z 客户端可以使用认证机构的公钥来验证证书
  - z 可以存在多个级别的证书颁发机构。每个证书颁发机构提供其自己的公钥证书,该证书由更高级别的权威机构签名,并且使用其私钥对其他网站/权威机构的证书进行签名



## 章节结束

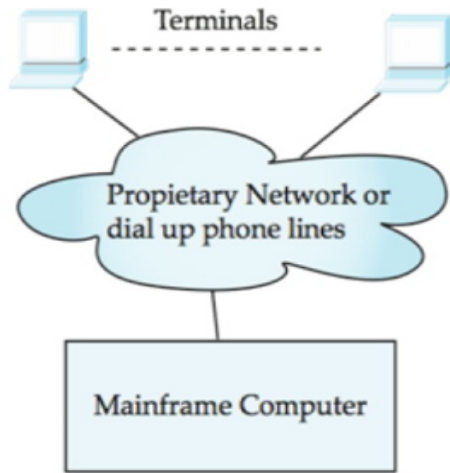
数据库系统概念, 6<sup>th</sup>Ed。

©Silberschatz, Korth和Sudarshan参见[www.db-book.com](http://www.db-book.com)了解  
重用条件

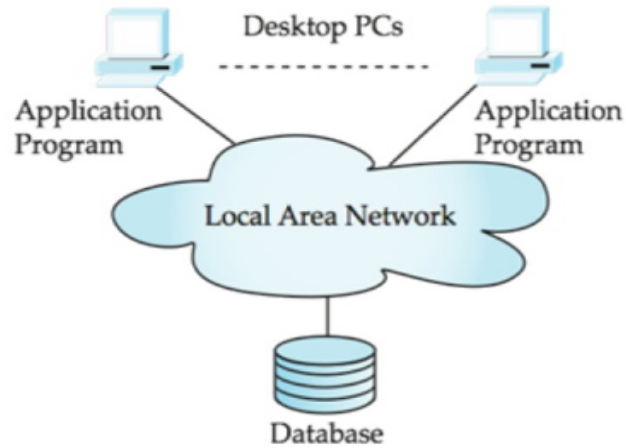
---



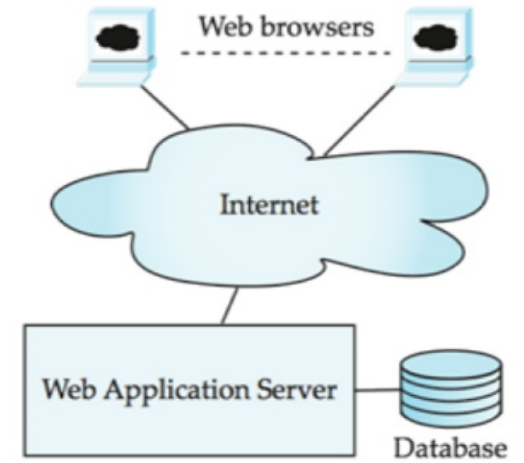
# Figure 9.01



(a) Mainframe Era



(b) Personal Computer Era



(c) Web era



**Figure 9.11**

