四川大學
SICHUAN UNIVERSITY

Database System Concepts

Application Development

伍元凯
College of Computer Science (Software), Sichuan University
wuyk0@scu.edu.cn

2023/05/06

目录

The RDF model represents data by a set of triples that are in one of these two forms:

1. $(ID, \text{attribute-name}, \text{value})$
2. $(ID1, \text{relationship-name}, ID2)$

where ID, $ID1$ and $ID2$ are identifiers of entities; entities are also referred to as resources in RDF. **The RDF model only supports binary relation.**

The first attribute of a triple is called its subject（主）, the second attribute is called its predicate（谓词）, and the last attribute is called its object（宾）. Thus, a triple has the structure (subject, predicate, object).

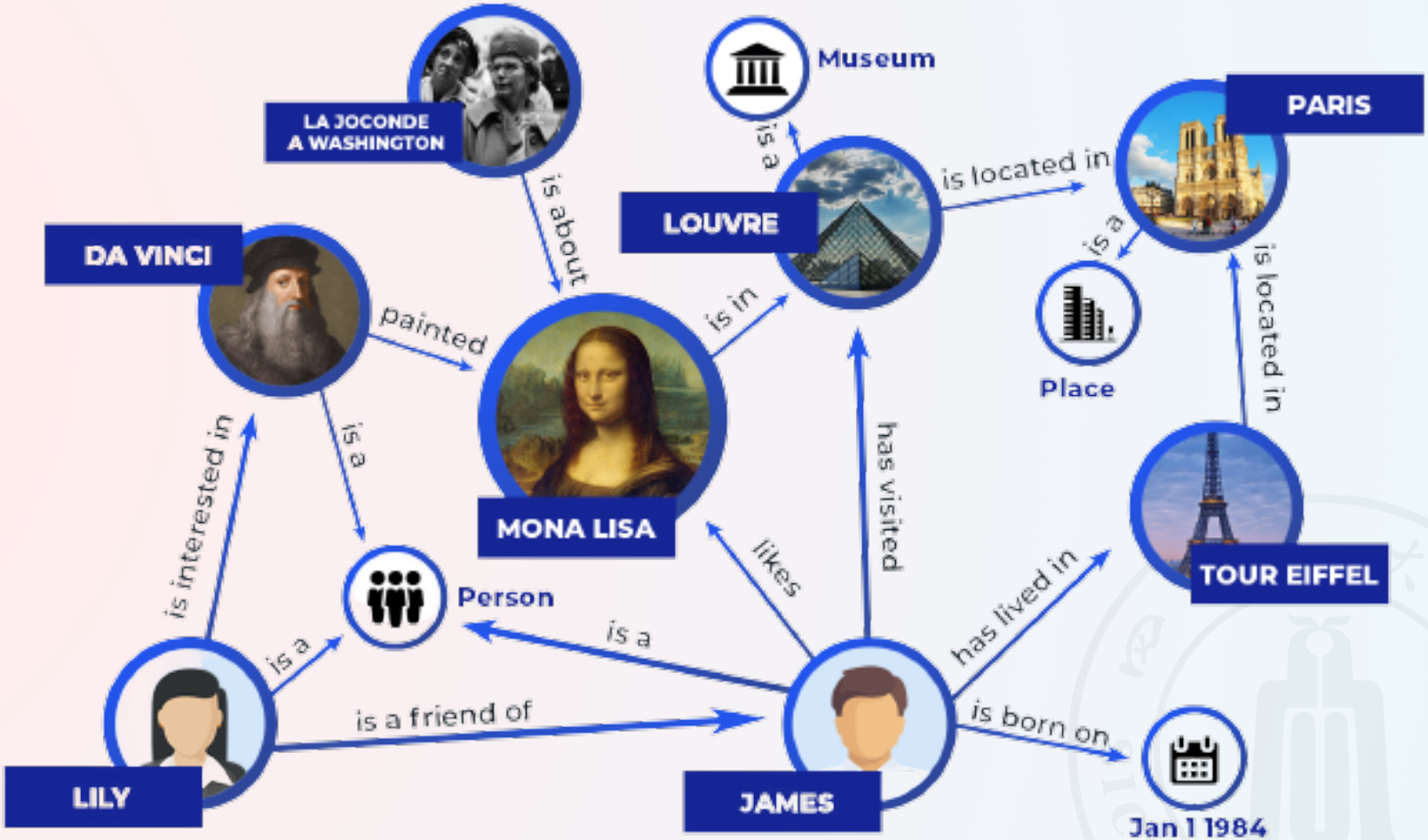
Subject	Predicate	Object
John	has_age	30
John	has_address	123 Main St.
John	has_phone	555-1234
Jane	has_age	25
Jane	has_address	456 Elm St.
Jane	has_phone	555-5678
John	is_friend_with	Jane

Example of an RDF model table

Knowledge graph

The RDF representation has a very natural graph interpretation. Entities and attribute values can be considered as nodes, and attribute names and relationships can be considered as edges between the nodes. The attribute/relationship name can be viewed as the label of the corresponding edge.

A representation of information using the RDF graph model (or its variants and extensions) is referred to as a knowledge graph.


LA JOCONDE A WASHINGTON
is about
Museum
is a
LOUVRE
is located in
PARIS
is a
Place
is located in
DA VINCI
painted
is in
MONA LISA
is interested in
is a
Person
likes
has visited
has lived in
TOUR EIFFEL
is a
is a
is a friend of
LILY
JAMES
is born on
Jan 1 1984

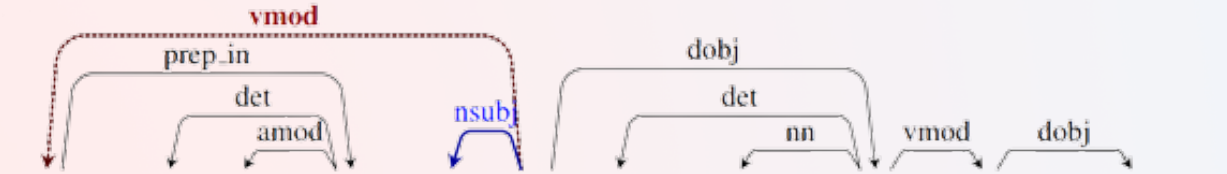
SPARQL | 目录

SPARQL is a query language designed to query RDF data. The language is based on triple patterns.

Finding lemon

```
1 SELECT ?fruit
2 WHERE
3 {
4   ?fruit color yellow.
5   ?fruit taste sour.
6 }
```

目录



(input)

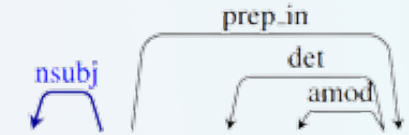
↓

she took the midnight train going anywhere
Born in a small town, she took the midnight train
Born in a town, she took the midnight train

↓

she took the midnight train
she took midnight train
. . .

(she; took; midnight train)



(extracted clause)

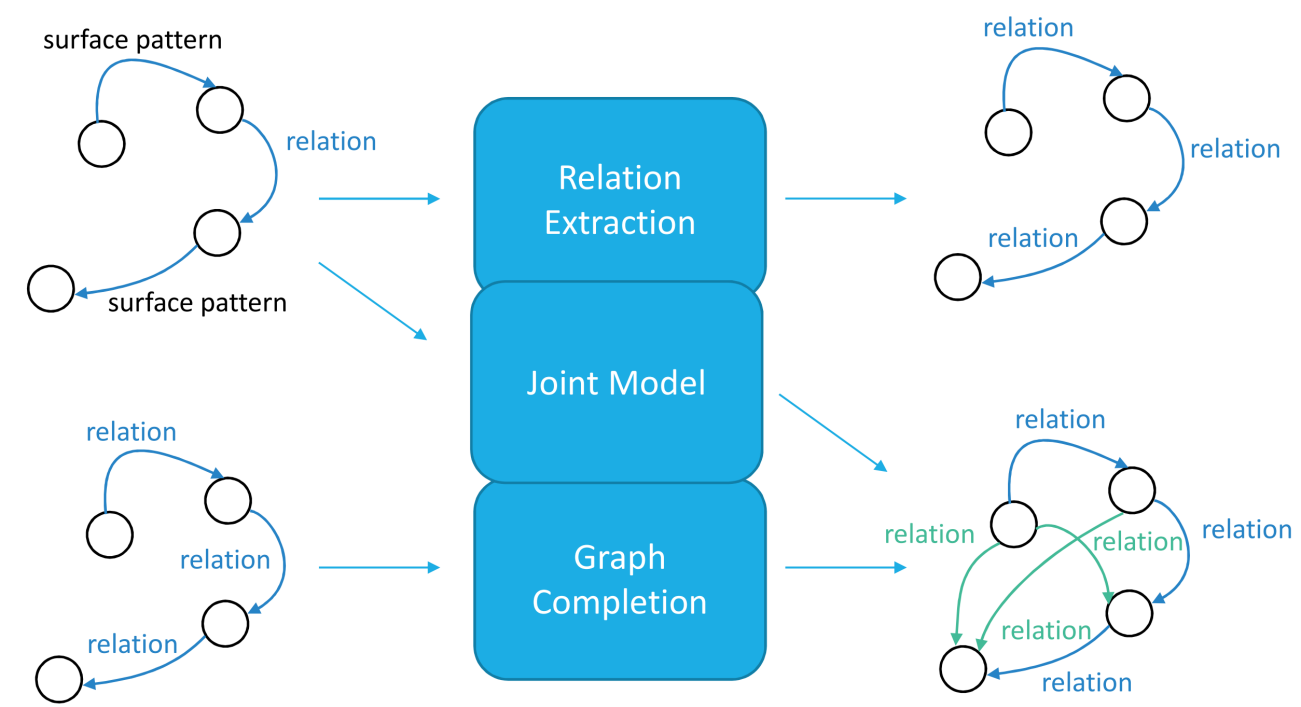
↓

she Born in small town
she Born in a town
she Born in town

↓

(she; born in; small town)
(she; born in; town)

The role of NLP

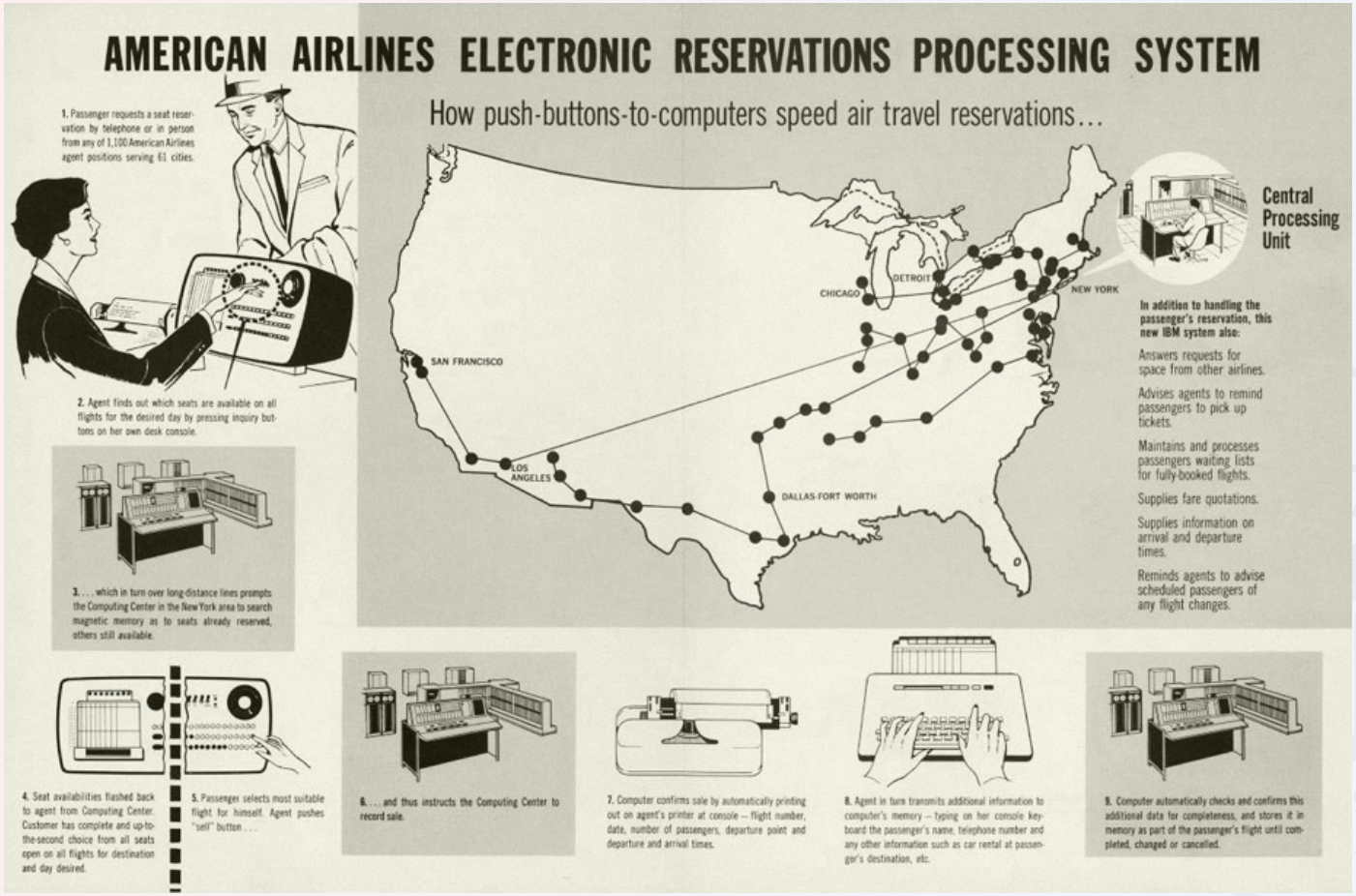


The role of ML

例 1

Some resources for potential final project:
https://github.com/husthuke/awesome-knowledge-graph

The most common way in which users interact with databases is through an application program that provides a user interface at the front end and interfaces with a database at the back end.



A typical application program includes a front-end component, which deals with the user interface.

A backend component, which communicates with a database

A middle layer, which contains "business logic," that is, code that executes specific requests for information or updates, enforcing rules of business such as what actions should be carried out to execute a given task or who can carry out what task.

client–server architecture

The interface depended on code running on a personal computer that directly communicated with a shared database.

Drawbacks:

- User machines had direct access to databases, leading to security risks.
- Any change to the application or the database required all the copies of the application, located on individual computers, to be updated together.

Solutions:

- Web browsers provide a universal front end, used by all kinds of information services. Browsers use a standardized syntax, the HyperText Markup Language (HTML) standard, which supports both formatted display of information and creation of forms-based interfaces. The HTML standard is independent of the operating system or browser, and pretty much every computer today has a web browser installed.
- Application programs are installed on individual devices, which are primarily mobile devices. They communicate with backend applications through an API and do not have direct access to the database. The back end application provides services, including user authentication, and ensures that users can only access services that they are authorized to access

目录

A uniform resource locator (URL) is a **globally unique name** for each document that can be accessed on the web. An example of a URL is: http://www.acm.org/sigmod

The first part of the URL indicates how the document is to be accessed: "http" indicates that the document is to be accessed by the HyperText Transfer Protocol (HTTP)（超文本传输协议）.
"https" would indicate that the secure version of the HTTP protocol must be used, and is the preferred mode today.
The second part gives the name of a machine that has a web server.
The rest of the URL is the path name of the file on the machine, or other unique identifier of a document within the machine

例 2

https://www.google.com/search?q=wu+yuankai

The program search on the server www.google.com should be executed with the argument q=wu+yuankai. On receiving a request for such a URL, the web server executes the program, using the given arguments. The program returns an HTML document to the web server, which sends it back to the front end.

目录

Tabular data in HTML format.

```
<table>
  <thead>
    <tr>
      <th>Product Name</th>
      <th>Price</th>
      <th>Stock</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Apple</td>
      <td>1.00</td>
      <td>100</td>
    </tr>
    <tr>
      <td>Orange</td>
      <td>1.50</td>

      <td>75</td>
    </tr>
    <tr>
      <td>Banana</td>
```

A table is specified by a **table** tag, which contains rows specified by a **tr** tag. The header row of the table has table cells specified by a **th** tag, while regular rows have table cells specified by a **td** tag.

Product Name	Price	Stock
Apple	1.00	100
Orange	1.50	75
Banana	0.50	200

An HTML form.

```
<form>
  <label for="type">Select type:</label>
  <select id="type" name="type">
    <option value="type1">Type 1</option>
    <option value="type2">Type 2</option>
    <option value="type3">Type 3</option>
  </select>
  <br><br>
  <label for="number">Input number:</label>
  <input type="number" id="number" name="number" min="0" max="100">
  <br><br>
  <input type="submit" value="Submit">
</form>
```

- form: Specifies the start of an HTML form
- action: Specifies the URL where the form data will be sent to
- method: Specifies the HTTP method used for submitting the form data (e.g. GET or POST)
- select: Specifies a drop-down list for selecting an option
- option: Specifies an option in a drop-down list
- name: Specifies the name of an input element (used for form data submission)
- type: Specifies the type of an input element (e.g. text, number, submit)
- value: Specifies the default value of an input element
- label: Associates a label with an input element (used for accessibility)
- for: Specifies the ID of the input element associated with a label

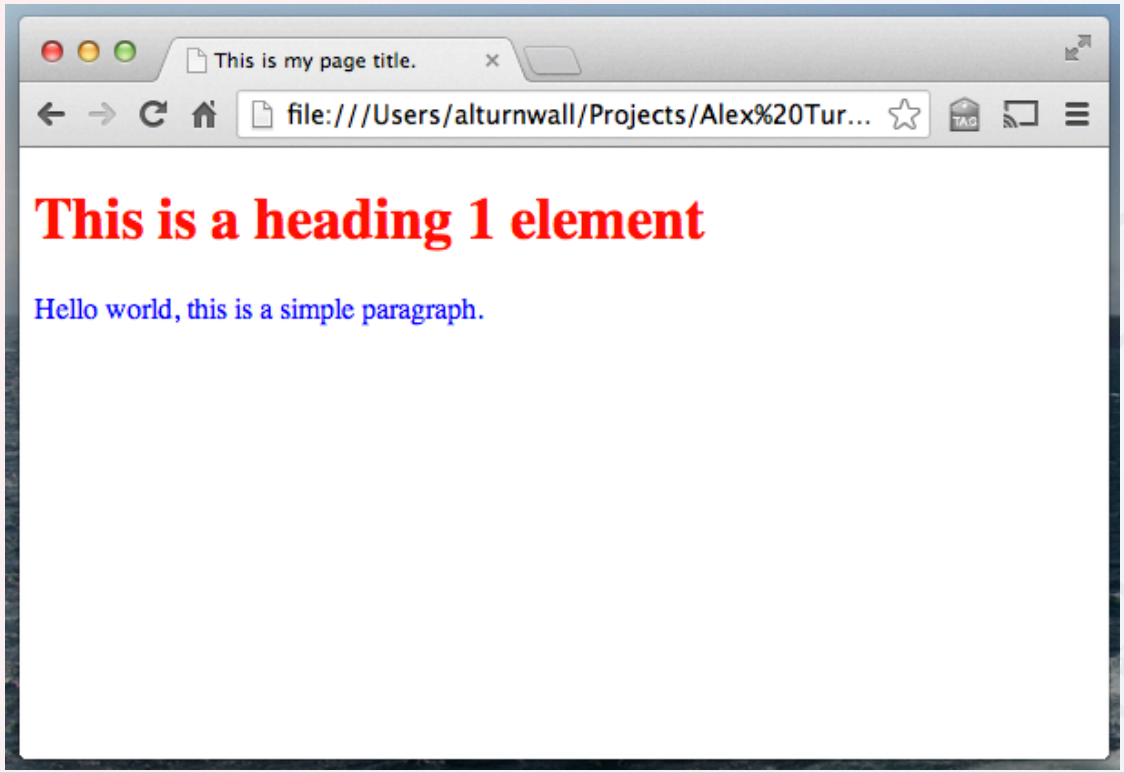
The **cascading stylesheet (CSS)** standard allows the same stylesheet to be used for multiple HTML documents, giving a distinctive but uniform look to all the pages on a web site.

CSS.

```
<!DOCTYPE html>
<html>
 <head>
  <title>This is my page title.</title>
  <link href="style.css" rel="stylesheet" type="text/css" />
 </head>
 <body>
  <h1>This is a heading 1 element</h1>
  <p>Hello world, this is a simple paragraph.</p>
 </body>
<html>
```

CSS.

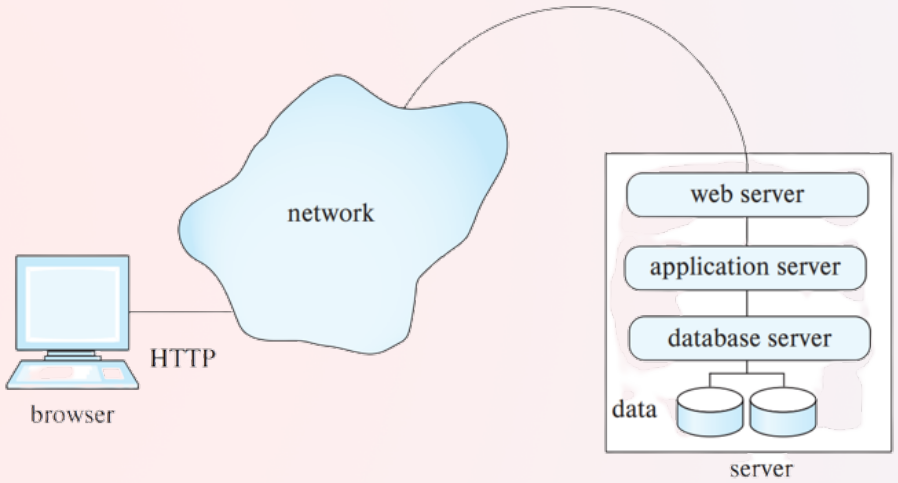
```
1 p {
2   color:blue;
3 }
4 h1 {
5   color:red;
6 }
```



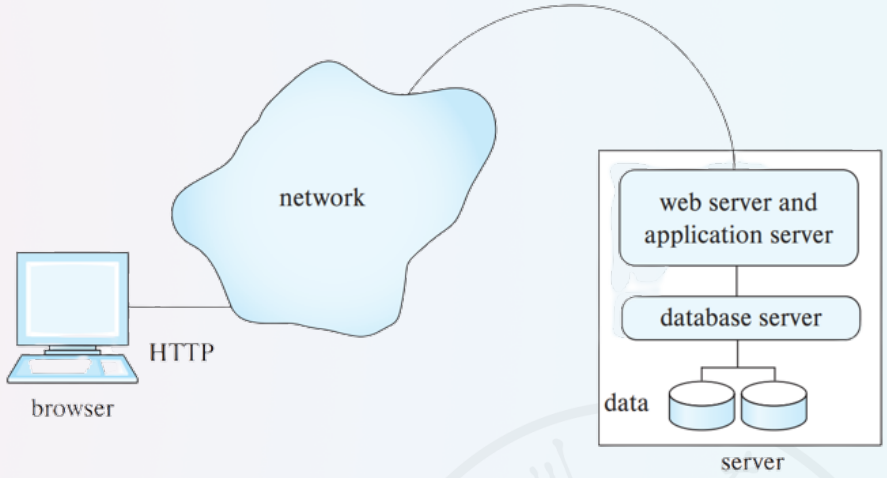
目录

A web server is a program running on the server machine that accepts requests from a web browser and sends back results in the form of HTML documents.

The common gateway interface (CGI) standard defines how the web server communicates with application programs. The application program typically communicates with a database server, through **ODBC, JDBC, or other protocols**, in order to get or store data.



Three-layer web application architecture.



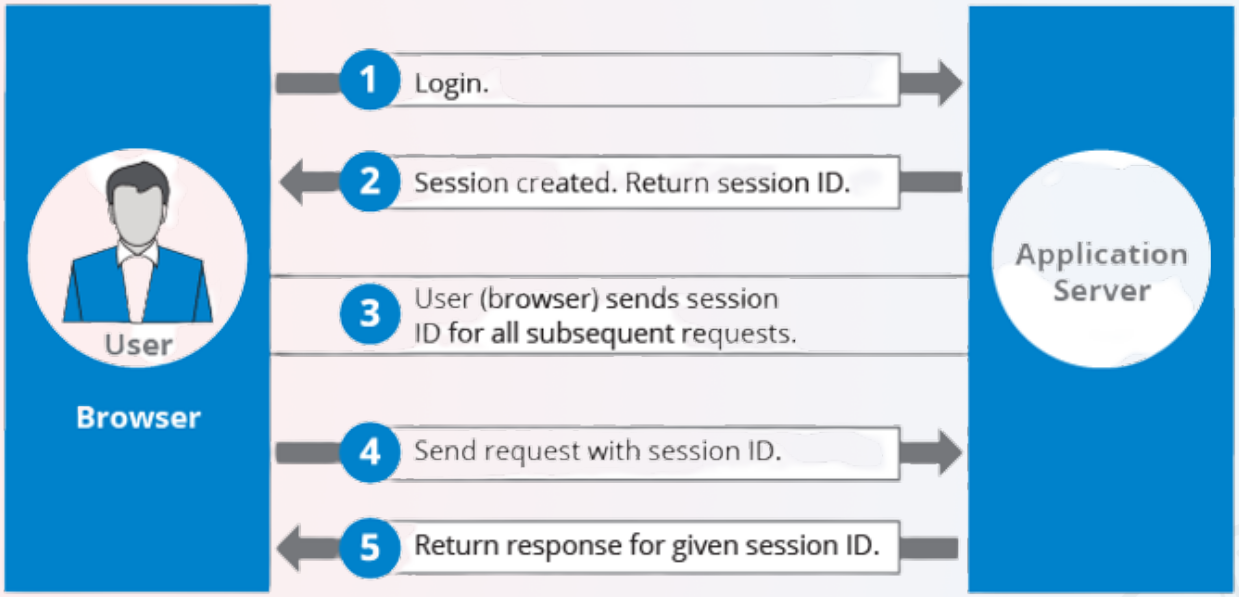
Two-layer web application architecture.

Connectionless

There is no continuous connection between the client and the web server; when a web server receives a request, a connection is temporarily created to send the request and receive the response from the web server. But the connection may then be closed, and the next request could come over a new connection.

Session

When a user logs on to a computer, or connects to a database using ODBC or JDBC, a session is created, and session information is retained at the server and the client until the session is terminatedinformation such as the useridentifier of the user and session options that the user has set.



A web session is a series of contiguous actions by a visitor on an individual website within a given time frame.

Cookies

Cookies are text files with small pieces of data —like a username and password —that are used to identify your computer as you use a computer network. Specific cookies known as HTTP cookies are used to identify specific users and improve your web browsing experience.

1. **You hand over your "coat" to the cloak desk**. In this case, a pocket of data is linked to you on the website server when you connect. This data can be your personal account, your shopping cart, or even just what pages you've visited.
2. **You get a "ticket" to identify you as the "coat" owner**. The cookie for the website is given to you and stored in your web browser. It has a unique ID especially for you.
3. **If you leave and return, you can get the "coat" with your "ticket"** . Your browser gives the website your cookie. It reads the unique ID in the cookie to assemble your activity data and recall your visit just as you left it

The Java servlet specification defines an application programming interface for communication between the web/application server and the application program. The HttpServlet class in Java implements the servlet API specification; servlet classes used to implement specific functions are defined as subclasses of this class

Servlet Example.

```
1 import java.io.*;
2 import javax.servlet.*;
3 import javax.servlet.http.*;
4 @WebServlet("PersonQuery")
5 public class PersonQueryServlet extends HttpServlet {
6     public void doGet(HttpServletRequest request,
7                       HttpServletResponse response)
8         throws ServletException, IOException
9     {
10        response.setContentType("text/html");
11        PrintWriter out = response.getWriter();
12        ... check if user is logged in ...
13        out.println("<HEAD><TITLE> Query Result</TITLE></HEAD>");
14        out.println("<BODY>");
15        String persontype = request.getParameter("persontype");
16        String name = request.getParameter("name");
17        if(persontype.equals("student")) {
```

```
18              ... code to find students with the specified name ...
19              ... using JDBC to communicate with the database ..
20              ... Assume ResultSet rs has been retrieved, and
21              ... contains attributes ID, name, and department name
22              String headers = new String[]{"ID", "Name", "Department
                     Name"};
23              Util::resultSetToHTML(rs, headers, out);
24      }
25      else {
26          ... as above, but for instructors ...
27      }
28      out.println("</BODY>");
29      out.close();
30      }
31  }
```

@WebServlet("PersonQuery") shown in the code. The form specifies that the **HTTP get** mechanism is used for transmitting parameters. So the **doGet()** method of the servlet, as defined in the code, is invoked.

Any values from the form menus and input fields on the web page, as well as cookies, pass through an object of the **HttpServletRequest** class that is created for the request, and the reply to the request passes through an object of the class **HttpServletResponse**.

The **doGet()** method in the example extracts values of the parameters persontype and name by using **request.getParameter()**, and uses these values to run a query against a database.

The servlet code returns the results of the query to the requester by outputting them to the **HttpServletResponse** object response. Outputting the results to response is implemented by first getting a **PrintWriter** object out from response, and then printing the query result in HTML format to out. In our example, the query result is printed by calling the function **Util::resultSetToHTML(resultset, header, out)**,

Utility function to output ResultSet as a table.

```
1  import java.io.*;
2  import javax.servlet.*;
3  import javax.servlet.http.*;
4  public class Util {
5      public static void resultSetToHTML(ResultSet rs,
6                                         String headers[],
                                               PrintWriter out) {
7          ResultSetMetaData rsmd = rs.getMetaData();
8          int numCols = rsmd.getColumnCount();
9          out.println("<table border=1>");
10         out.println("<tr>");
11         for (int i=0; i < numCols; i++)
12             out.println("<th>"+ headers[i] + </th>
13         out.println(" </tr>");
14         while (rs.next()) {
15             out.println("<tr>");
16             for (int i=0; i < numCols; i++)

17                 out.println("<td>"+ rs.getString(i) + </td>
18                 out.println(" </tr>");
19         }
20     }
```

Servlet Sessions | 目录

Track a session.

```
1 import java.io.*;
2 import javax.servlet.*;
3 import javax.servlet.http.*;
4 public class SessionExample extends HttpServlet {
5     public void doGet(HttpServletRequest request,
          HttpServletResponse response)
6             throws ServletException, IOException {
7         // Get the current session or create a new one if none
              exists
8         HttpSession session = request.getSession();
9         // Store a value in the session
10        session.setAttribute("username", "john_doe");
11        // Retrieve a value from the session
12        String username = (String) session.getAttribute("username")
              ;
13        // Print out the value retrieved from the session
14        response.setContentType("text/html");
```

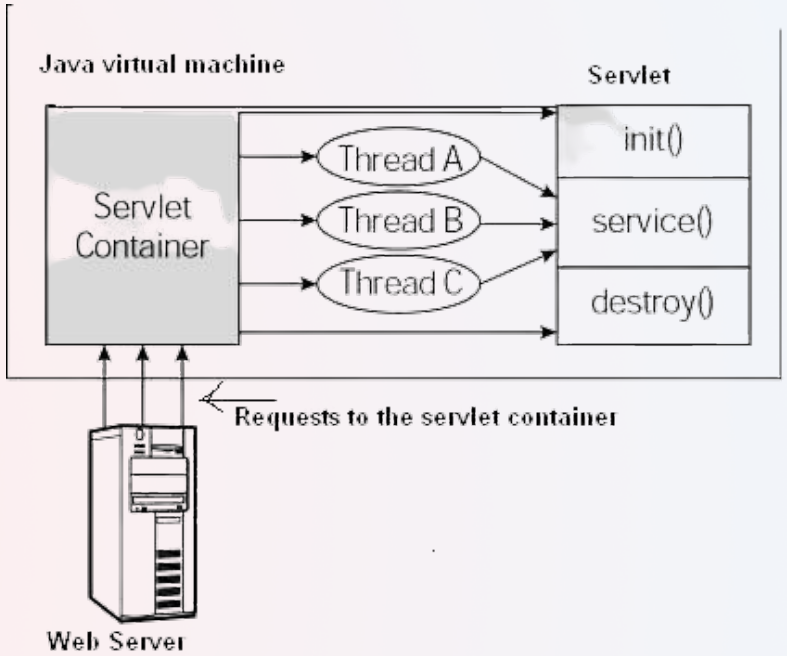
```
15          PrintWriter out = response.getWriter();
16          out.println("<html><body>");
17          out.println("<h1>Username retrieved from session: " +
                username + "</h1>");
18          out.println("</body></html>");
19      }
20  }
```

The **doGet** method of the SessionExample servlet uses the **HttpServletRequest** object to retrieve the current session or create a new one if none exists. It then stores a value ("john_doe") in the session using the **setAttribute** method, and retrieves the same value using the **getAttribute** method.

```
1 Session session = request.getSession(false);
2 if (session == null || session.getAttribute(userid) == null) {
3     out.println("You are not logged in.");
4     response.setHeader("Refresh", "5;url=login.html");
5     return();
6 }
```

例 3

if the user is not logged in, it sends an error message, and after a gap of 5 seconds, redirects the user to the login page.



1. The first HTTP request that reaches the server is delegated to the Servlet container.
2. The Servlet container loads the Servlet before invoking the service() method.
3. The Servlet container handles multiple requests generated by multiple threads, with each thread executing the service() method of a single Servlet instance.

The **init** method is designed to be called only once. It is invoked when the Servlet is first created and is not called again for subsequent user requests. Therefore, it is used for one-time initialization purposes.

```
1 public void init() throws ServletException {
2   // ...
3 }
```

The **service()** method is the main method that performs the actual tasks. The Servlet container (i.e., the web server) calls the **service()** method to handle requests from clients (browsers) and send formatted responses back to the clients.

```
1 public void service(ServletRequest request,
2                     ServletResponse response)
3       throws ServletException, IOException{
4 }
```

A GET request comes from a normal request of a URL or from an HTML form that does not specify a method. It is handled by the **doGet()** method.

```
1 public void doGet(HttpServletRequest request,
2                   HttpServletResponse response)
3     throws ServletException, IOException {
4     // Servlet Code
5 }
```

A POST request comes from an HTML form that specifically specifies the method as POST. It is handled by the **doPost()** method.

```
1 public void doPost(HttpServletRequest request,
2                     HttpServletResponse response)
3     throws ServletException, IOException {
4     // Servlet Code
5 }
```

例 4

In doGet(), the parameters are appended to the URL and sent along with header information. This does not happen in case of doPost(). In doPost(), the parameters are sent separately. Since most of the web servers support only a limited amount of information to be attached to the headers, the size of this header should not exceed 1024 bytes. doPost() does not have this constraint.

doGet() shall be used when small amount of data and insensitive data like a query has to be sent as a request. doPost() shall be used when comparatively large amount of sensitive data has to be sent. Examples are sending data after filling up a form or sending login id and password.

The **destroy()** method is called only once, at the end of the servlet's lifecycle. The **destroy()** method allows you to close database connections, stop background threads, write cookie lists or hit counters to disk, and perform other similar cleanup activities.

```
1  public void destroy() {
2    // ...
3  }
```

目录

Many application servers provide built-in support for servlets.

- Tomcat Server from the Apache Jakarta Project
- Glassfish
- JBoss
- BEA Weblogic Application Server
- Oracle Application Server
- IBM’s WebSphere Application Server

1. Download the latest version of Tomcat from http://tomcat.apache.org/.
2. Once you have downloaded Tomcat, extract it to a convenient location. For example, if you are using Windows, extract it to C:-tomcat-5.5.29. If you are using Linux/Unix, extract it to /usr/local/apache-tomcat-5.5.29. Then, create the CATALINA_HOME environment variable pointing to these locations.

start Tomcat

```
1 %CATALINA_HOME%\bin\startup.bat
2 C:\apache-tomcat-5.5.29\bin\startup.bat
```

set CLASSPATH

```
1 set CATALINA=C:\apache-tomcat-5.5.29
2 set CLASSPATH=%CATALINA%\common\lib\servlet-api.jar;%CLASSPATH%
```

Scripting

目录

Motivation

Writing even a simple web application in a programming language such as Java or C is a **time-consuming task**. An alternative approach, that of serverside scripting, provides a much easier method for creating many applications.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title> </title>
</head>
<body>
<h1>我的 Web 页面</h1>
<p id="myPar">我是一个段落。</p>
<div id="myDiv">我是一个div。</div>
<p>
<button type="button" onclick="myFunction()">点击这里</button>
</p>
<script>
function myFunction(){
^^Idocument.getElementById("myPar").innerHTML="你好世界！ ";
^^Idocument.getElementById("myDiv").innerHTML="你最近怎么样?";

}
</script>
<p>当您点击上面的按钮时，两个元素会改变。</p>
</body>
```

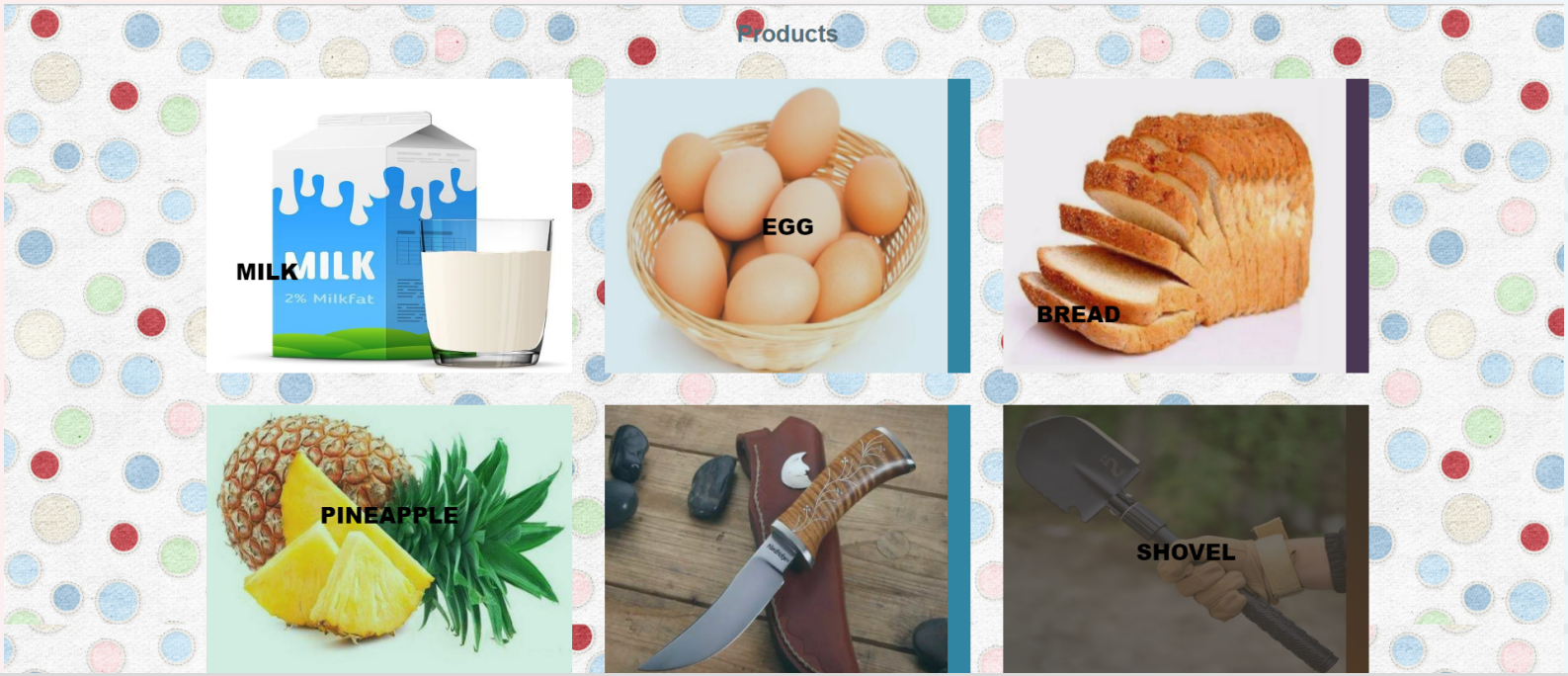
PHP is a scripting language that is widely used for server-side scripting. PHP code can be intermixed with HTML in a manner similar to JSP. The characters "<?php" indicate the start of PHP code, while the characters "?>" indicate the end of PHP code.

```
1 <html>
2 <head> <title> Hello </title> </head>
3 <body>
4 <?php if (!isset($ REQUEST['name']))
5 { echo 'Hello World'; }
6 else { echo 'Hello, ' . $ REQUEST['name']; }
7 ?>
8 </body>
9 </html>
```

- PHP can generate dynamic page content.
- PHP can create, open, read, write, and close files on the server.
- PHP can collect form data.
- PHP can send and receive cookies.
- PHP can add, delete, and modify data in your database.
- PHP can restrict user access to certain pages on your website.
- PHP can encrypt data.
- With PHP, you are not limited to outputting HTML. You can output images, PDF files, and even Flash movies. You can also output any arbitrary text, such as XHTML and XML.

```php
function showProducts()
{
    $figure_class_name = ["effect-lily","effect-sadie","effect-honey","effect-layla","effect-zoe","effect-oscar"];
    $img_addr = ["photos/milk1.jpg","photos/egg.jpg","photos/bread.jpg","photos/pineapple.jpg","photos/knife.jpg","photos/shovel.jpg"];
    $color_name = ["#000","#000","#000","#000","#000","#000"];
    $pn = get_products_names();
    // $length = len($pn);
    for($i = 0;$pn[$i];$i++)
    {
echo <<<EOF
<figure class="$figure_class_name[$i]">
<img src="$img_addr[$i]" style="objec-fit: cover;height: 100%;" alt="img0$i" onclick="showWindow$i()"/>
<figcaption onclick="showWindow$i()">
<h2 style="color: $color_name[$i];"><span onclick="showWindow$i()">$pn[$i]</span></h2>
</figcaption>
</figure>
EOF;
$pa = strtolower($pn[$i]);
```

调用数据库查询



目录

Web Application Frameworks

- A library of functions to support HTML and HTTP features, including sessions.
- A template scripting system.
- A controller that maps user interaction events such as form submits to appropriate functions that handle the event. The controller also manages authentication and sessions. Some frameworks also provide tools for managing authorizations.
- A (relatively) declarative way of specifying a form with validation constraints on user inputs, from which the system generates HTML and Javascript/Ajax code to implement the form.
- An object-oriented model with an object-relational mapping to store data in a relational database.

例 5

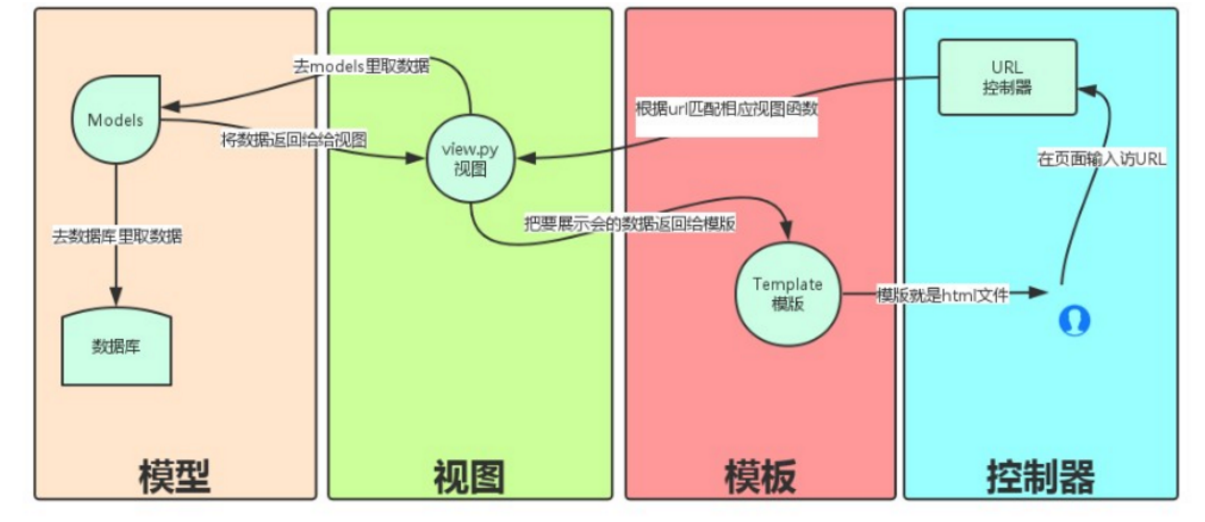
Django framework for the Python language, Ruby on Rails, which supports the Rails framework on the Ruby programming language, Apache Struts, Swing, Tapestry, and WebObjects, all based on Java/JSP.

目录

An example: https://github.com/alonglyn/MIS

```
1 from django.http import HttpResponse
2 from django.db import connection
3 def result set to html(headers, cursor):
4     html = "<table border=1>"
5     html += "<tr>"
6     for header in headers:
7         html += "<th>" + header + "</th>"
8         html += "</tr>"
9     for row in cursor.fetchall():
10        html += "<tr>"
11        for col in row:
12            html += "<td>" + col + "</td>"
13        html += "</tr>"
14    html += "</table>"
15    return html
16 def person query view(request):
17    if "username" not in request.session:
```

```
18          return login view(request)
19      persontype = request.GET.get("persontype")
20      personname = request.GET.get("personname")
21      if persontype == "student":
22          query tmpl = "select id, name, dept name from student where
                name=%s"
23      else:
24          query tmpl = "select id, name, dept name from instructor
                where name=%s"
25      with connection.cursor() as cursor:
26          cursor.execute(query tmpl, [personname])
27          headers = ["ID", "Name", "Department Name"]
28          return HttpResponse(result set to html(headers, cursor))
```



MTV:

- M stands for Model, which represents the data access layer: how to access it, how to validate its integrity, what behaviors it should have, and the relationships between data entities.
- T stands for Template, which represents the presentation layer: how to display the data in web pages or other types of documents.
- V stands for View, which represents the business logic layer. This layer includes the logic for accessing the model and selecting the appropriate template.

Client-side scripting languages are languages designed to be executed on the client's web browser.

JavaScript language is by far the most widely used client-side scripting language. The current generation of web interfaces uses the JavaScript scripting language extensively to construct sophisticated user interfaces.

JavaScript | 目录

Ensures that the input for the parameter “credits” is a number between 1 and 15.

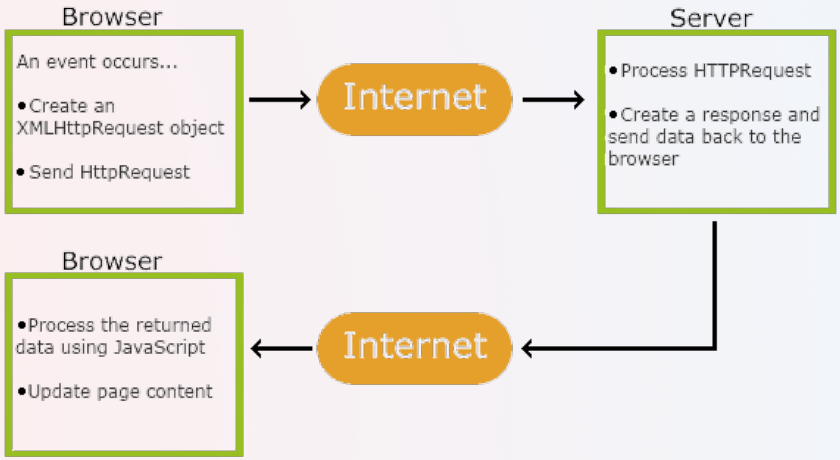
```
1 <input type="number" name="credits" size="2" min="1" max="15"
```

```
\begin{html}
<head>
<script type="text/javascript">
function validate() {
  var startdate = new Date (document.getElementById("start").value)
      ;
  var enddate = new Date (document.getElementById("end").value);
  if(startdate > enddate) {
    alert("Start date is > end date");
    return false;
  }
}
</script>
</head>
<body>
<form action="submitDates" onsubmit="return validate()">
Start Date: <input type="date" id="start"><br />

End Date : <input type="date" id="end"><br />
<input type="submit" value="Submit">
</form>
</body>
```

目录

The key to building a user interface is the ability to dynamically modify the HTML code being displayed by using JavaScript. The browser parses HTML code into an in-memory tree structure defined by a standard called the Document Object Model (DOM).



What is AJAX?

AJAX = Asynchronous JavaScript And XML.

AJAX is not a programming language.

AJAX just uses a combination of:

- A browser built-in XMLHttpRequest object (to request data from a web server)
- JavaScript and HTML DOM (to display or use the data)

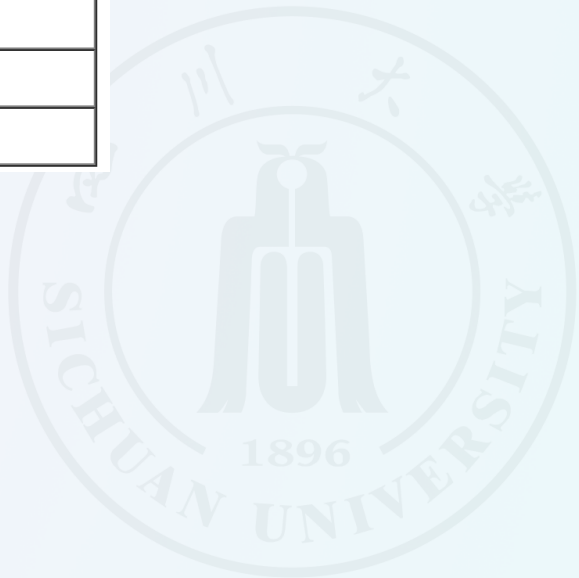
```
1
2 <!DOCTYPE html>
3 <html>
4 <style>
5 th,td {
6   padding: 5px;
7 }
8 </style>
9 <body>
10
11 <h2>The XMLHttpRequest Object</h2>
12
13 <form action="">
14   <select name="customers" onchange="showCustomer(this.value)">
15     <option value="">Select a customer:</option>
16     <option value="ALFKI">Alfreds Futterkiste</option>
17     <option value="NORTS ">North/South</option>
```

```
18      <option value="WOLZA">Wolski Zajazd</option>
19    </select>
20 </form>
21 <br>
22 <div id="txtHint">Customer info will be listed here...</div>
23
24 <script>
25 function showCustomer(str) {
26   if (str == "") {
27     document.getElementById("txtHint").innerHTML = "";
28     return;
29   }
30   const xhttp = new XMLHttpRequest();
31   xhttp.onload = function() {
32     document.getElementById("txtHint").innerHTML = this.
           responseText;
33   }
34   xhttp.open("GET", "getcustomer.php?q="+str);
35   xhttp.send();
36 }
37 </script>
38 </body>
39 </html>
```

The XMLHttpRequest Object

Alfreds Futterkiste

CustomerID	ALFKI
CompanyName	Alfreds Futterkiste
ContactName	Maria Anders
Address	Obere Str. 57
City	Berlin
PostalCode	12209
Country	Germany



web service is an application component that can be invoked over the web and functions, in effect, like an application programming interface. A web service request is sent using the HTTP protocol, it is executed at an application server, and the results are sent back to the calling program

Two approaches are widely used to implement web services. In the simpler approach, called Representation State Transfer (or REST), web service function calls are executed by a standard HTTP request to a URL at an application server, with parameters sent as standard HTTP request parameters.

The term RST was introduced and defined in 2000 by **Roy Fielding** in his doctoral dissertation. It means that a server will respond with the representation of a resource (an HTML, XML or JSON document) and that resource will contain hypermedia links that can be followed to make the state of the system change.

Fielding, R. T. (2000). Representational state transfer (REST). Chapter 5 in Architectural Styles and the Design of Networkbased Software Architectures (Doctoral dissertation, Ph. D. Thesis, University of California, Irvine, CA).

例 6

”A good architecture is not created in a vacuum. All design decisions at the architectural level should be made within the context of the functional, behavioral, and social requirements of the system being designed, which is a principle that applies equally to both software architecture and the traditional field of building architecture. ”

```
{
  "requests": [
    {
      "image": {
        "content": "BASE64_ENCODED_IMAGE"
      },
      "features": [
        {
          "type": "TEXT_DETECTION"
        }
      ]
    }
  ]
}
```

```
POST https://vision.googleapis.com/v1/images:annotate
```


Essential Tools for Cloud Platform
Simplify Your Cloud Management
Collaborative Development on Git


WAITING?
PLEASE
TURN OFF
YOUR
ENGINE

目录

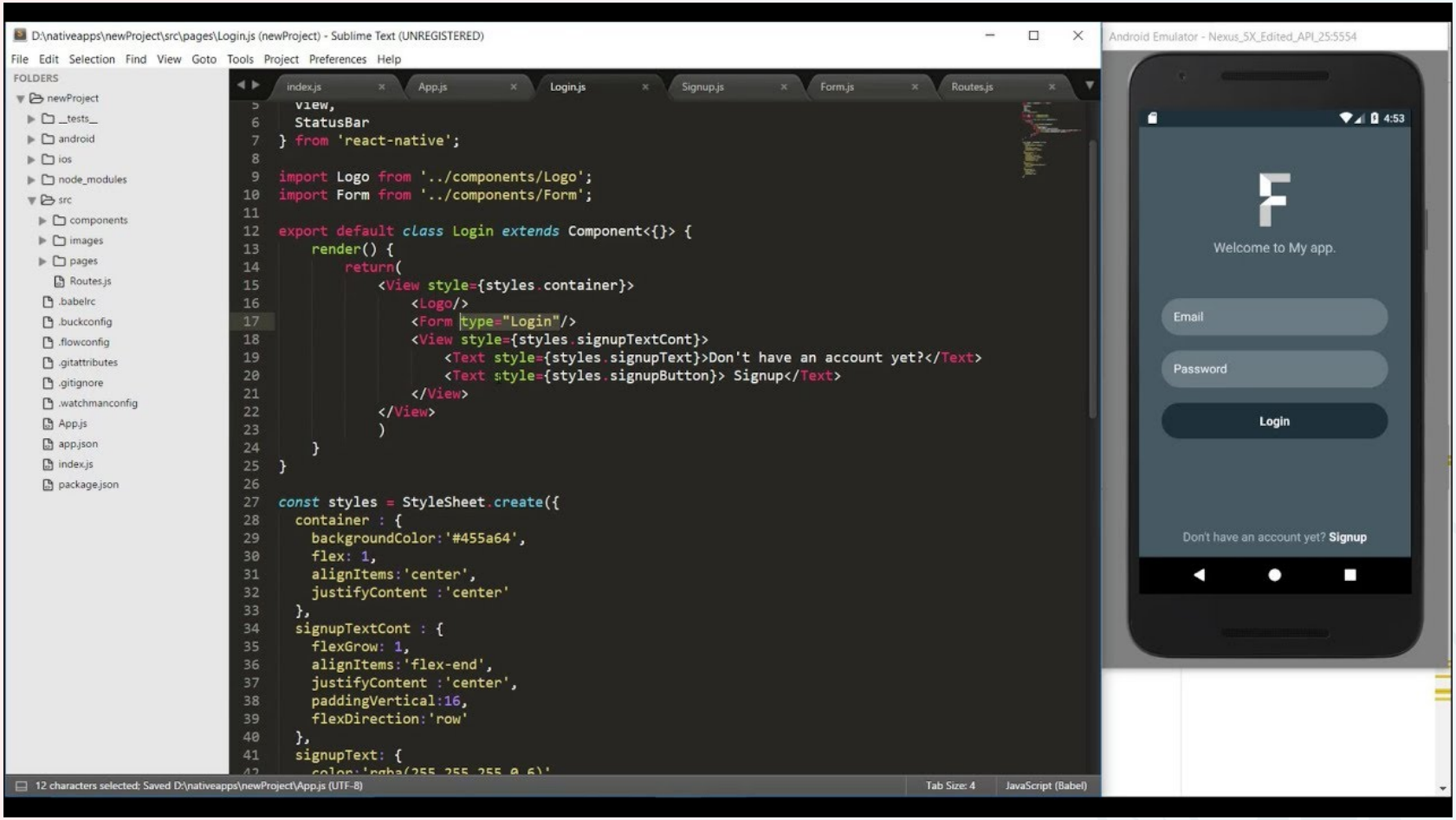
```
1 if (typeof(Storage) !== "undefined") { // browser supports local
      storage
2 ...
3 }
```

```
1 localStorage.setItem(key, value)
2 localStorage.getItem(key)
3 localStorage.deleteItem(key)
```

目录

T

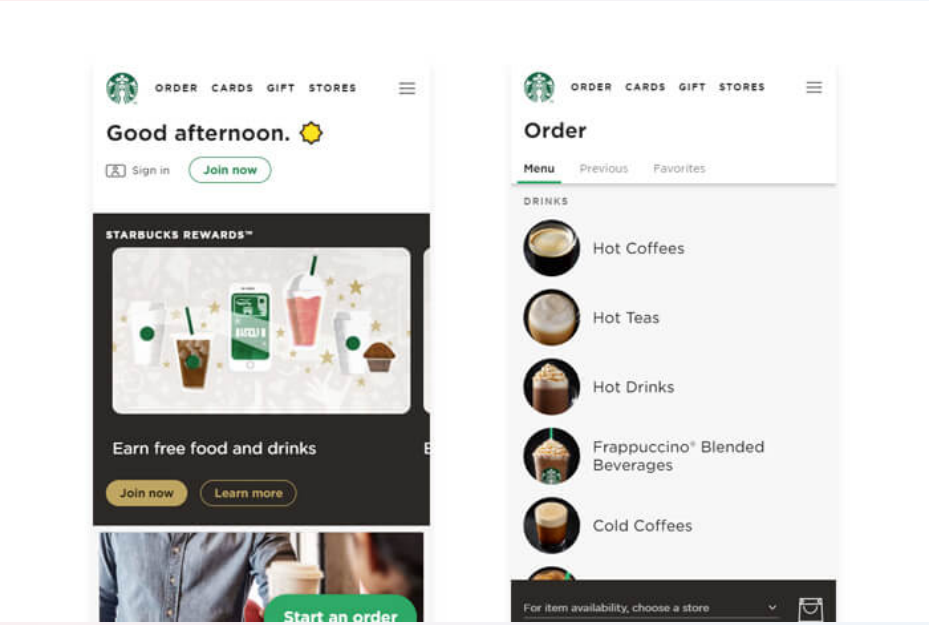
he ability to create applications where the same high-level code can run on either Android or iOS is clearly very important. The React Native framework based on JavaScript, developed by Facebook, and the Flutter framework based on the Dart language developed by Google, are designed to allow cross-platform development


D:\nativeapps\newProject\src\pages\Login.js (newProject) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
newProject
__tests__
android
ios
node_modules
src
components
images
pages
Routes.js
.babelrc
.buckconfig
.flowconfig
.gitattributes
.gitignore
.watchmanconfig
App.js
app.json
index.js
package.json
index.js
App.js
Login.js
Signup.js
Form.js
Routes.js
view,
StatusBar
} from 'react-native';
import Logo from '../components/Logo';
import Form from '../components/Form';
export default class Login extends Component<{}> {
render() {
return(
<View style={styles.container}>
<Logo/>
<Form type="Login"/>
<View style={styles.signupTextCont}>
<Text style={styles.signupText}>Don't have an account yet?</Text>
<Text style={styles.signupButton}> Signup</Text>
</View>
</View>
)
}
}
const styles = StyleSheet.create({
container : {
backgroundColor:'#455a64',
flex: 1,
alignItems:'center',
justifyContent :'center'
},
signupTextCont : {
flexGrow: 1,
alignItems:'flex-end',
justifyContent :'center',
paddingVertical:16,
flexDirection:'row'
},
signupText: {
12 characters selected: Saved D:\nativeapps\newProject\App.js (UTF-8)
Tab Size: 4
JavaScript (Babel)
Android Emulator - Nexus_5X_Edited_API_25:5554
4:53
Welcome to My app.
Email
Password
Login
Don't have an account yet? Signup

A progressive web application (PWA), or progressive web app, is a type of application software delivered through the web, built using common web technologies including HTML, CSS, JavaScript, and WebAssembly. It is intended to work on any platform with a standards-compliant browser, including desktop and mobile devices.

例 7

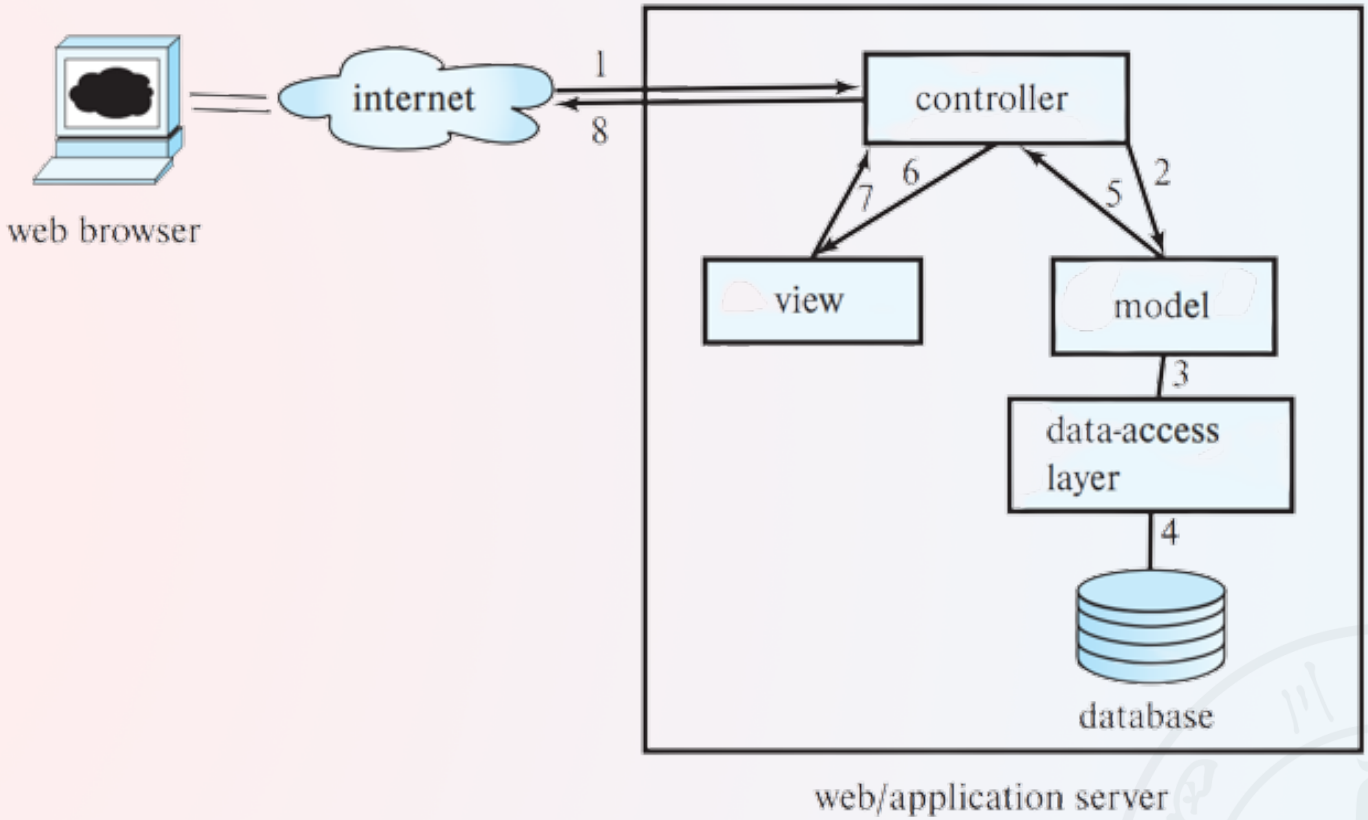
Starbucks provides a PWA that is **99.84%** smaller than its equivalent iOS app. After deploying its PWA, Starbucks **doubled the number of online orders**, with desktop users ordering at about the same rate as mobile app users.



1. For ordinary users, mini-programs make applications easily accessible. Users can simply scan QR codes, search, or receive shared links to directly open the mini-programs.
2. For developers, the mini-program framework itself provides fast loading and rendering capabilities.
3. The mini-program model enables WeChat to expose more data.

To handle their complexity, large applications are often broken into several layers:

- The **presentation or user-interface** layer, which deals with user interaction. A single application may have several different versions of this layer, corresponding to distinct kinds of interfaces such as web browsers and user interfaces of mobile phones, which have much smaller screens.
- The business-logic layer, which provides a high-level view of data and actions on data.
- The data-access layer, which provides the interface between the business-logic layer and the underlying database. Object-oriented $\rightarrow$ relational database.



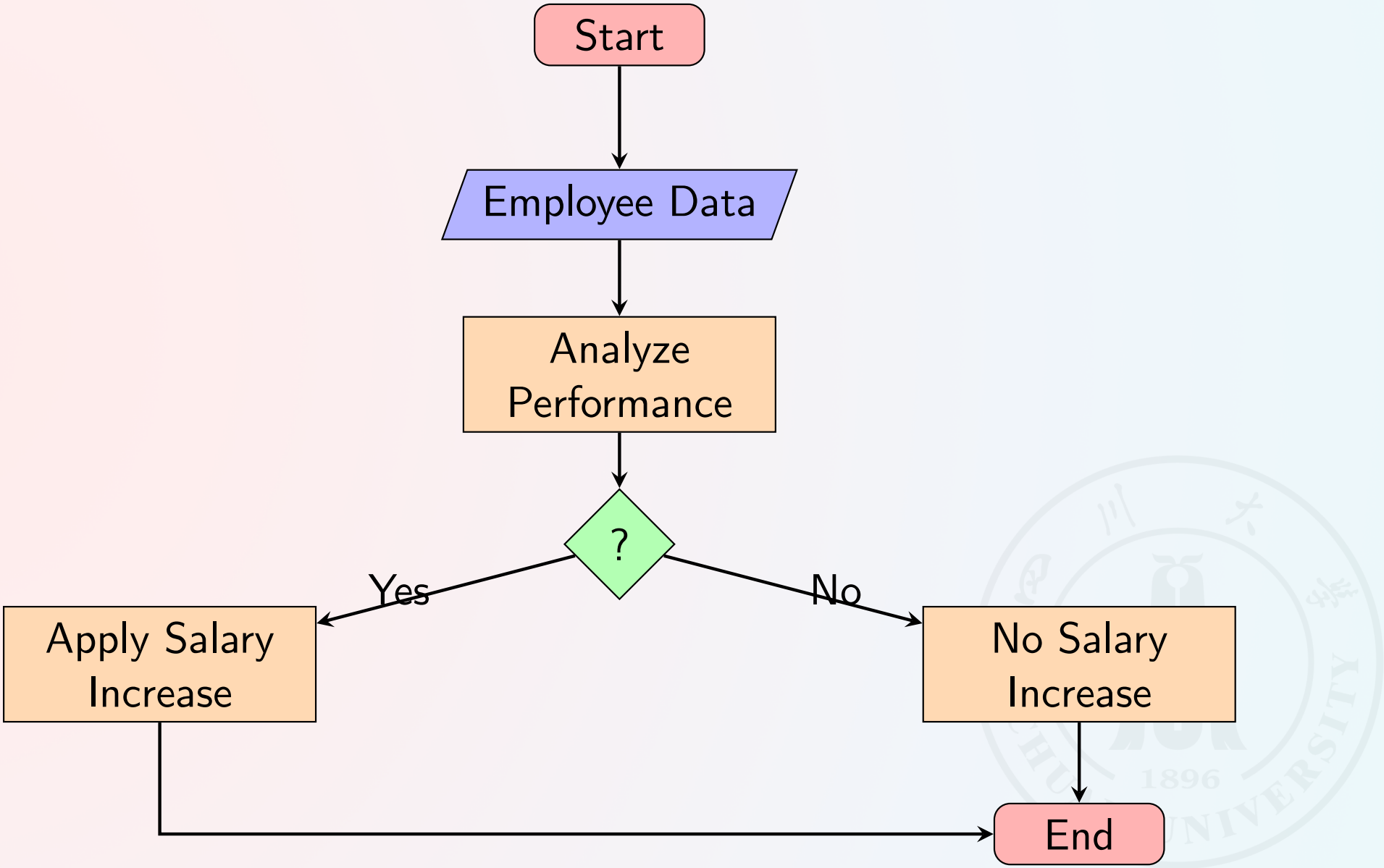
The model processes the request, using business logic. The model in turn uses the data-access layer to update or retrieve information from a database. The result object created by the model is sent to the view module, which creates an HTML view of the result to be displayed on the web browser.

目录

The code implementing these actions ensures that **business rules** are satisfied.

The business logic includes **workflows**, which describe how a particular task that involves multiple participants is handled

A real-world workflow



目录

In early implementations, programmers had to write code for creating objects by fetching data from the database and for storing updated objects back in the database. However, such manual conversions between data models is **cumbersome and error prone**.
Object-oriented databases emerged to meet the need of coupling object-oriented programming languages with a database.
In object-oriented programming (OOP), everything is an object.

Object-oriented databases contain the following foundational elements:

- **Objects** are a real world entities. All objects are assigned a class within data structures for both hierarchy and functional purposes.
- **Attributes and Methods**: An object has state and behaviors.
- **Classes** are a grouping of all objects with the same properties and behaviors.
- **Pointers** are addresses that facilitate both object access and establishing relationships between objects.

An OOP example

```
1 class task
2 {
3     String name;
4     String status;
5     Date create_date;
6
7     public void update_task(String status)
8     {
9     ...
10    }
11 }
```

Object-oriented programming (OOP) concepts:

- **Abstraction（抽象）** is a way of capturing the necessary information to perform desired functionality while excluding unneeded information.
- **Encapsulation（封装）** refers to the ability of each object to maintain a private state within its assigned class.
- **Inheritance（继承）** enables objects to acquire some of the attributes or properties of another object by reusing the existing fields and methods.
- **Polymorphism（多态）** enables a child class to use class attributes like its parent class, while retaining all of its unique methods and attributes.

OOD example: Realm of MongoDB

An OOD example

```
1 class Dog: Object {
2     dynamic var name = ""
3     dynamic var age = 0
4     dynamic var breed: String? = nil
5     dynamic var owner: Person?
6 }
7
8 class Person: Object {
9 ^^Ilet name = ""
10 ^^Ilet dogs = List<Dog>()
11 }
12
13 bob.dogs.append(fido)
```

Simple and easy query

```
1 RealmResults<User> result = realm.where(User.class)
2                                 .equalTo("name", "John")
3                                 .or()
4                                 .equalTo("name", "Peter")
5                                 .findAllAsync();
```

Why aren't Object Oriented databases used as much as Relational Databases?

Inheritance in Oracle

```
1 create table people of Person;
2 create table students of Student
3 under people;
4 create table teachers of Teacher
5 under people;
```

Inheritance in PostgreSQL

```
1 create table students
2 (degree varchar(20))
3 inherits people;
4 create table teachers
5 (salary integer)
6 inherits people;
```

When we declare students and teachers as subtables of people, every tuple present in students or teachers becomes implicitly present in people.

object-relational mappings (ORMs)

Using traditional relational databases to store data, but to automate the mapping of data in relation to **in-memory** objects, which are **created on demand**, as well as the reverse mapping to store updated objects back as relations in the database.

The Hibernate system is widely used for mapping from Java objects to relations.

HQL query

```
1 Session session = getSessionFactory().openSession();
2 Transaction txn = session.beginTransaction();
3
4 // Retrieve student object by identifier
5 Student stud1 = session.get(Student.class, "12328");
6 .. print out the Student information ..
7
8 List students = session.createQuery("from Student as s order by s.
      ID asc").list();
9 for ( Iterator iter = students.iterator(); iter.hasNext(); ) {
10     Student stud = (Student) iter.next();
11     .. print out the Student information ..
12 }
13
14 txn.commit();
15 session.close();
```

The HQL query is automatically translated to SQL by Hibernate and executed, and the results are converted into a list of Student objects. The for loop iterates over the objects in this list.

Class definition in Django

```
1 from django.db import models
2
3 class Student(models.Model):
4     id = models.CharField(primary_key=True, max_length=5)
5     name = models.CharField(max_length=20)
6     dept_name = models.CharField(max_length=20)
7     tot_cred = models.DecimalField(max_digits=3, decimal_places=0)
8
9 class Instructor(models.Model):
10    id = models.CharField(primary_key=True, max_length=5)
11    name = models.CharField(max_length=20)
12    dept_name = models.CharField(max_length=20)
13    salary = models.DecimalField(max_digits=8, decimal_places=2)
14    advisees = models.ManyToManyField(Student, related_name="
          advisors")
```

The reverse relationship from Student to Instructor is created automatically

View definition in Django

```
1 rom models import Student, Instructor
2 def get_names(persons):
3     res = ""
4     for p in persons:
5         res += p.name + ", "
6     return res.rstrip(", ")
7 def person_query_model(request):
8     persontype = request.GET.get('persontype')
9     personname = request.GET.get('personname')
10    html = ""
11    if persontype == 'student':
12        students = Student.objects.filter(name=personname)
13        for student in students:
14            advisors = student.advisors.all()
15            html += "Advisee: " + student.name + "<br>Advisors: " +
                  get_names(advisors) + "<br> \\n"
16    else:
```

```
17          instructors = Instructor.objects.filter(name=personname)
18          for instructor in instructors:
19              advisees = instructor.advisees.all()
20              html += "Advisor: " + instructor.name + "<br>Advisees:
                    " + get_names(advisees) + "<br> \\n"
21      return HttpResponse(html)
```

目录

Suppose that the application code for servicing each user request connects to a database through JDBC. Creating a new JDBC connection may take several milliseconds, so opening a new connection for each user request is not a good idea if very high transaction rates are to be supported.

The connection pooling method is used to reduce this overhead; it works as follows: The connection pool manager creates a pool of open ODBC/JDBC connections. Instead of opening a new connection to the database, the code servicing a user request asks for a connection from the connection pool and returns the connection to the pool when the code completes its processing.

When a connection is first opened, a connection pool is created based on an exact matching algorithm that associates the pool with the **connection string (similar behavior)** in the connection.

Pool Creation

```
1 sing (SqlConnection connection = new SqlConnection(
2   "Integrated Security=SSPI;Initial Catalog=Northwind"))
3     {
4         connection.Open();
5         // Pool A is created.
6     }
7 using (SqlConnection connection = new SqlConnection(
8   "Integrated Security=SSPI;Initial Catalog=pubs"))
9     {
10        connection.Open();
11        // Pool B is created because the connection strings differ.
12    }
13 using (SqlConnection connection = new SqlConnection(
14   "Integrated Security=SSPI;Initial Catalog=Northwind"))
15    {
16        connection.Open();
17        // The connection string matches pool A.
```

Adding Connections

When a pool is created, multiple connection objects are created and added to the pool so that the minimum pool size requirement is satisfied. Connections are added to the pool as needed, up to the maximum pool size specified (100 is the default). Connections are released back into the pool when they are closed or disposed.

Removing Connections

The connection pooler removes a connection from the pool after it has been idle for approximately 4-8 minutes, or if the pooler detects that the connection with the server has been severed.

Clearing the Pool

If there are connections being used at the time of the call, they are marked appropriately. When they are closed, they are discarded instead of being returned to the pool.

Using cached query results

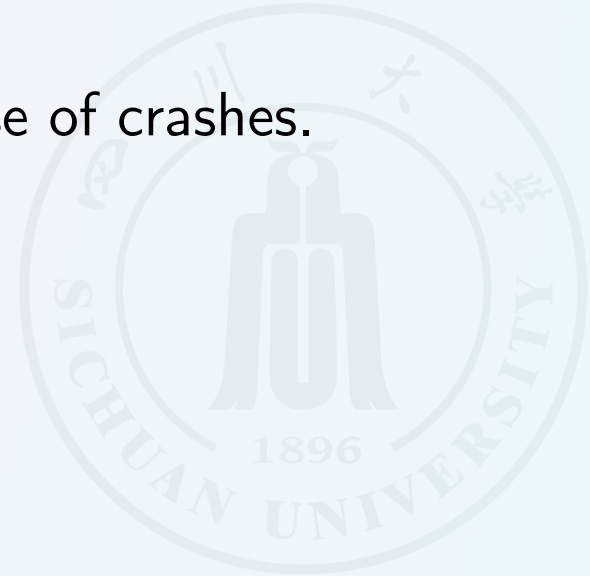
Cached query results and cached web pages are forms of materialized views. If the underlying database data change, the cached results must be discarded, or recomputed, or even incrementally updated, as in materialized-view maintenance

There are several widely used main-memory caching systems; among the more popular ones are memcached and Redis. Both systems allow applications to store data with an associated key and retrieve data for a specified key

Redis **stores data in memory**, it can deliver much faster access times than disk-based databases.

Primary disadvantages:

- Because Redis stores data primarily in memory, it can only store as much data as can fit into memory.
- Redis is susceptible to potential data loss in case of crashes.



In SQL injection attacks, the attacker manages to get an application to execute an SQL query created by the attacker

Vulnerable code

```
1 tring query =  “select * from student where name like ’ %”
2 + name +  “%’  ”
```

Injection attack

```
1 elect * from student where name like '%'; <some SQL statement>;  –
      –  %’
```

The following text inserted by the attacker gets interpreted as a second SQL query.

Solution

Prepared query in Section 5.

Another source of SQL-injection risk comes from applications that create queries dynamically, based on selection conditions and ordering attributes specified in a form.

Injection attack

```
1 tring query = “select * from takes order by ” + orderAttribute;
```

To avoid this kind of SQL injection, the application should ensure that the orderAttribute variable value is one of the allowed values

目录

A web site that allows users to enter text, such as a comment or a name, and then stores it and later displays it to other users, is potentially vulnerable to a kind of attack called a cross-site scripting (XSS) attack.

例 8

Suppose the user happens to be logged into her bank account at the time the script executes. The script could send cookie information related to the bank account login back to the malicious user, who could use the information to connect to the bank's web server, fooling it into believing that the connection is from the original user.

- The simplest technique is to disallow any HTML tags whatsoever in text input by users. There are functions that detect or strip all such tags. These functions can be used to prevent HTML tags, and as a result, any scripts, from being displayed to other users.
- The HTTP protocol allows a server to check the referer of a page access, that is, the URL of the page that had the link that the user clicked on to initiate the page access.
- Instead of using only the cookie to identify a session, the session could also be restricted to the IP address from which it was originally authenticated.
- Never use a GET method to perform any updates.
- use the XSRF/CSRF protection mechanisms provided by the framework.

Password Leakage | 目录

- Store passwords in encrypted form, which the server decrypts before passing it on to the database.
- Allow access to the database to be restricted to a given set of internet addresses, typically, the machines running the application servers.

目录

Authentication refers to the task of verifying the identity of a person/software connecting to an application. The simplest form of authentication consists of a secret password that must be presented when a user connects to the application.

Many applications use two-factor authentication, where two independent factors are used to identify a user. The two factors **should not share a common vulnerability**.

例 9

Even with two-factor authentication, users may still be vulnerable to man-in-the-middle attacks. In such attacks, a user attempting to connect to the application is diverted to a fake web site

A single sign-on system further allows the user to be authenticated once, and multiple applications can then verify the user's identity through an authentication service without requiring reauthentication. (川大一键身份登录)

The Security Assertion Markup Language (SAML) is a protocol for exchanging authentication and authorization information between different security domains, to provide cross-organization single sign-on.

目录

Limited role in managing user authorizations in a typical application.

- **Lack of end-user information**. With the growth in the web, database accesses come primarily from web application servers. The end users typically do not have individual user identifiers on the database itself, and indeed there may only be a single user identifier in the database corresponding to all users of an application server.
- **Lack of fine-grained authorization**. Authorization must be at the level of individual tuples if we are to authorize students to see only their own grades. Such authorization is not possible in the current SQL standard, which permits authorization only on an entire relation or view, or on specified attributes of relations or views.

Oracle VPD

```
1 REATE OR REPLACE FUNCTION employee_access_policy (schema_name
      VARCHAR2, table_name VARCHAR2)
2 RETURN VARCHAR2
3 AS
4   predicate VARCHAR2(4000);
5 BEGIN
6   -- Add your custom logic here to define the security policy
7   -- For example, restricting access based on a user's role
8   IF USER = 'HR_ADMIN' THEN
9     predicate := NULL; -- Allow full access to HR_ADMIN role
10  ELSE
11    predicate := 'employee_id = SYS_CONTEXT(''USERENV'', ''
          SESSION_USERID'')'; -- Allow access to only the employee's
          own record
12  END IF;
13
14  RETURN predicate;

15 END;
16 /
```

Create a policy function:

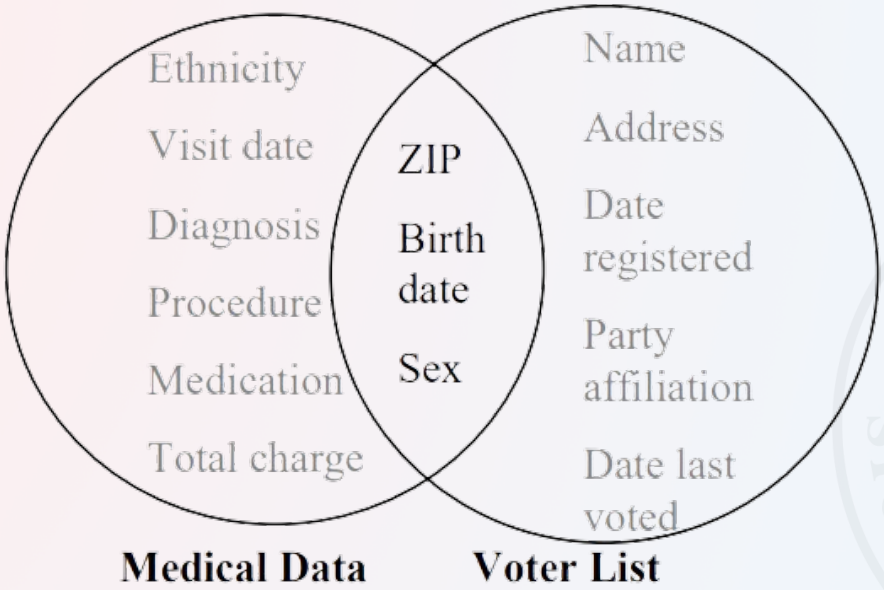
```
1 EGIN
2   DBMS_RLS.ADD_POLICY(
3     object_schema   => 'your_schema',
4     object_name     => 'employees',
5     policy_name     => 'employee_policy',
6     function_schema => 'your_schema',
7     policy_function => 'employee_access_policy',
8     statement_types => 'SELECT',
9     update_check    => FALSE,
10    enable          => TRUE
11  );
12 END;
13 /
```

An audit trail is a log of all changes (inserts, deletes, and updates) to the application data, along with information such as which user performed the change and when the change was performed. (a) help find out what happened, and who may have carried out the actions, and (b) aid in fixing the damage caused by the security breach or erroneous update.

- Copy the audit trail to a different machine, to which the intruder would not have access, with each record in the trail copied as soon as it is generated
- Blockchain

Privacy 目录

In many cases the **survival of the database itself depends on the data holder's ability to produce anonymous data** because not releasing such information at all may diminish the need for the data, while on the other hand, failing to provide proper protection within a release may create circumstances that harm the public or others.



例 10

Re-identification by linking

1. Ambulatory care data from hospitals, physicians offices, clinics, and so forth (Left circle).
2. Voter registration list (Right circle).
3. This information can be linked using ZIP code, birth date and gender to the medical information, thereby This information can be linked using ZIP code, birth date and gender to the medical information, thereby.
4. Six people had his particular birth date; only three of them were men; and, William Weld was the only one in his 5-digit ZIP code.

Quasi-identifiers

Pieces of information that are not of themselves unique identifiers, but are sufficiently well correlated with an entity that they can be combined with other quasi-identifiers to create a unique identifier

k-anonymity

The information for each person contained in the release cannot be distinguished from at least k - 1 individuals whose information also appear in the release

Example of k-anonymity, where $k = 2$ and

$Quasi - identifiers = \{Race, Birth, Gender, ZIP\}$

Race	Birth	Gender	ZIP	Problem
Black	1965	m	0214*	short breath
Black	1965	m	0214*	chest pain
Black	1965	f	0213*	hypertension
Black	1965	f	0213*	hypertension
Black	1964	f	0213*	obesity
Black	1964	f	0213*	chest pain
White	1964	m	0213*	chest pain
White	1964	m	0213*	obesity
White	1964	m	0213*	short breath
White	1967	m	0213*	chest pain
White	1967	m	0213*	chest pain

Potential solution: $1965 \rightarrow 1960 - 1970$

Sweeney, Latanya. "k-anonymity: A model for protecting privacy." International journal of uncertainty, fuzziness and knowledge-based systems 10.05 (2002): 557-570.

Encryption refers to the process of transforming data into a form that is unreadable, **unless the reverse process of decryption is applied**.

Encryption algorithms use an encryption key to perform encryption, and they require a decryption key to perform decryption.

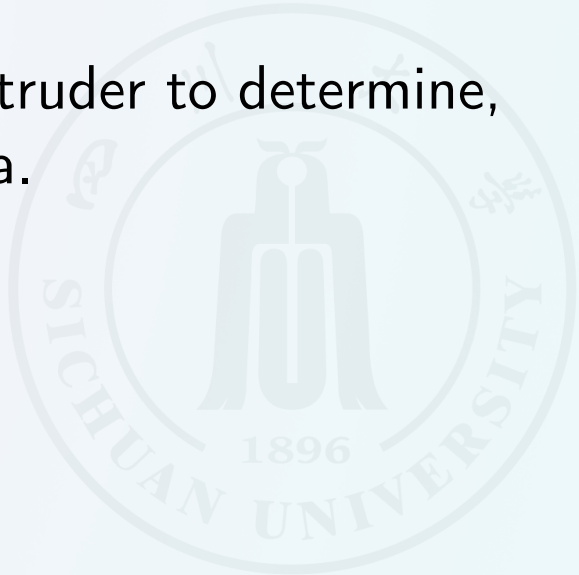
Leakage of personal information allows a criminal to impersonate someone else and get access to service or goods; such impersonation is referred to as identity theft.

Techniques

目录

A good encryption technique has the following properties:

- It is relatively simple for authorized users to encrypt and decrypt data.
- It depends not on the secrecy of the algorithm, but rather on a parameter of the algorithm called the **encryption key**, which is used to encrypt data.
- Its decryption key is extremely difficult for an intruder to determine, even if the intruder has access to encrypted data.



目录

At the lowest level, the disk blocks containing database data can be encrypted, using a key available to the database-system software. When a block is retrieved from disk, it is first decrypted and then used in the usual fashion.

At the next higher level, specified (or all) attributes of a relation can be stored in encrypted form. In this case, each attribute of a relation could have a different encryption key.

Encryption at the database level has the advantage of requiring relatively low time and space overhead and does not require modification of applications.

- Application programs that use **databases as back ends** and interact with users have been around since the 1960s.
- **HTML** provides the ability to define interfaces that combine hyperlinks with forms facilities. Web browsers communicate with web servers by the **HTTP** protocol.
- **Servlets** are a widely used mechanism to write application programs that run as part of the web server process, in order to reduce overhead.
- Complex applications usually have a multilayer architecture, including a model implementing **business logic, a controller, and a view mechanism to display results**.
- Techniques such as **caching** of various forms, including query result caching and connection pooling.

用这节课学习到的知识写一个前端网页，作为大作业的一部分（两周之内完成）
Github 上有大量的可参考的 Project：
https://github.com/topics/dbms-project

Thanks
End of Chapter 9