Xi'an Jiaotong-Liverpool University
西交利物浦大学

DTS207TC Database Development and Design

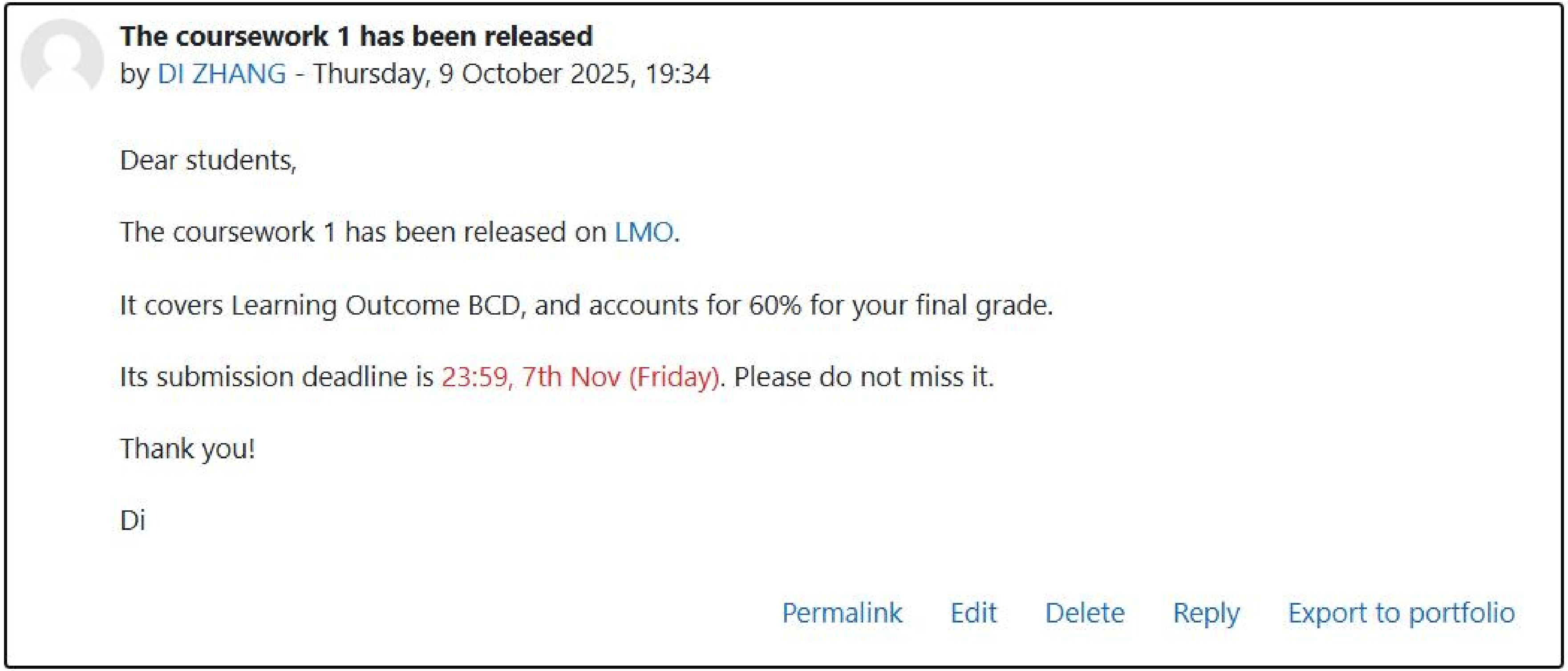
Lecture 5 Web

Chap 8.2 OR-Mapping

Chap 9 App. Development

Di Zhang, Autumn 2025

*Annoucement



The coursework 1 has been released
by DI ZHANG - Thursday, 9 October 2025, 19:34

Dear students,

The coursework 1 has been released on LMO.

It covers Learning Outcome BCD, and accounts for 60% for your final grade.

Its submission deadline is 23:59, 7th Nov (Friday). Please do not miss it.

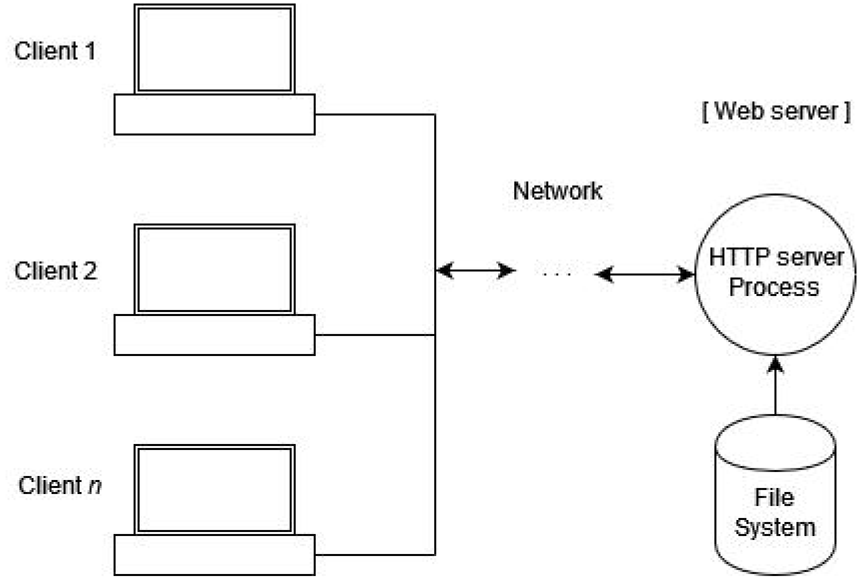
Thank you!

Di

Outline

- Web Architecture
- HTML
- Web Server
- ORM

Overview



- A web server is computer software and underlying hardware that accepts requests via HTTP (the network protocol created to distribute web content).
- A user agent, commonly a web browser or web crawler, initiates communication by making a request for a web page or other resource using HTTP, and the server responds with the content of that resource or an error message.
- A web server can also accept and store resources sent from the user agent if configured to do so.

Web Server

- A Web server can be a physical computer that provides and hosts website content,or it can be a software program that controls network users'access to files.It receives requests from clients such as browsers via the HTTP(or HTTPS)protocol,looks up and returns the requested web pages,images,videos,and other content.
- Two Meanings of Web Server
  - Hardware(Physical Machine):A computer connected to the Internet that stores all the files of a website,such as HTML files,CSS style sheets,JavaScript files,and images.
  - Software:A program running on the hardware,with the most important part being the HTTP server.It is responsible for processing HTTP requests from clients and sending the corresponding website content back in the form of HTTP responses.

*Hardware

- The hardware used to run a web server can vary according to the volume of requests that it needs to handle. At the low end of the range are embedded systems, such as a router that runs a small web server as its configuration interface. A high-traffic Internet website might handle requests with hundreds of servers that run on racks of high-speed computers.

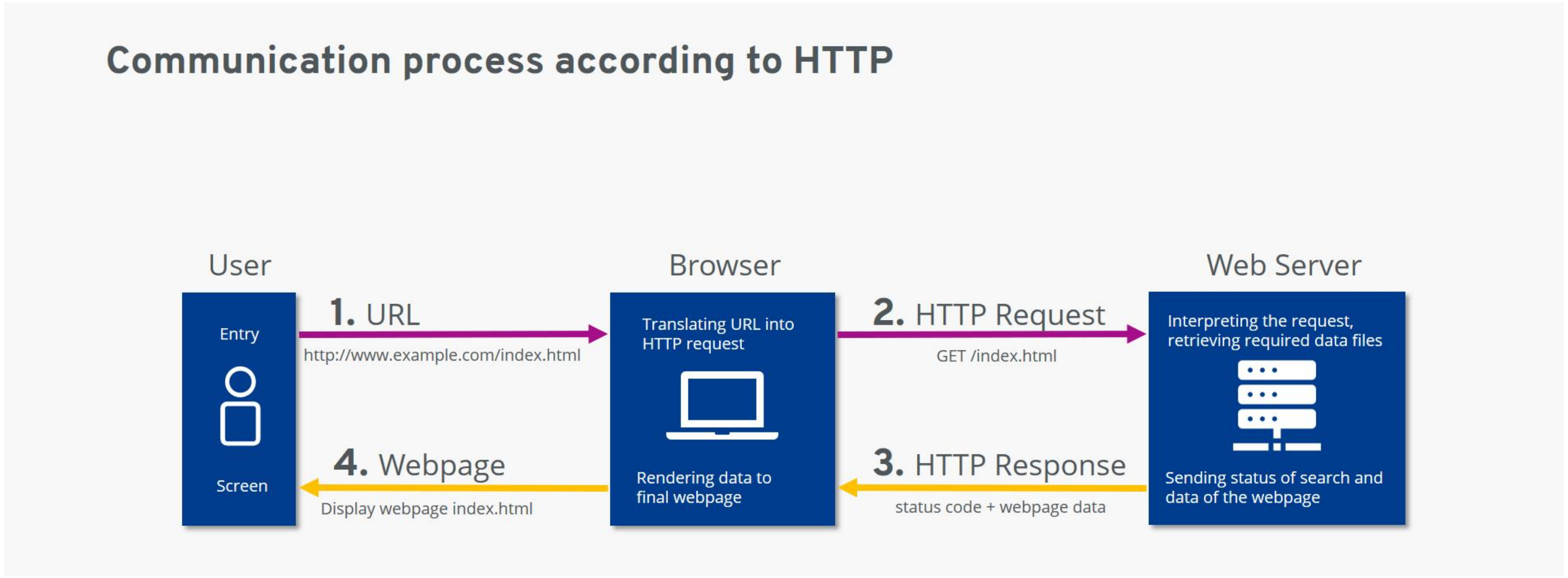
The world's first web server, a NeXT Computer workstation with Ethernet, 1990. The case label reads: "This machine is a server. DO NOT POWER IT DOWN!!"

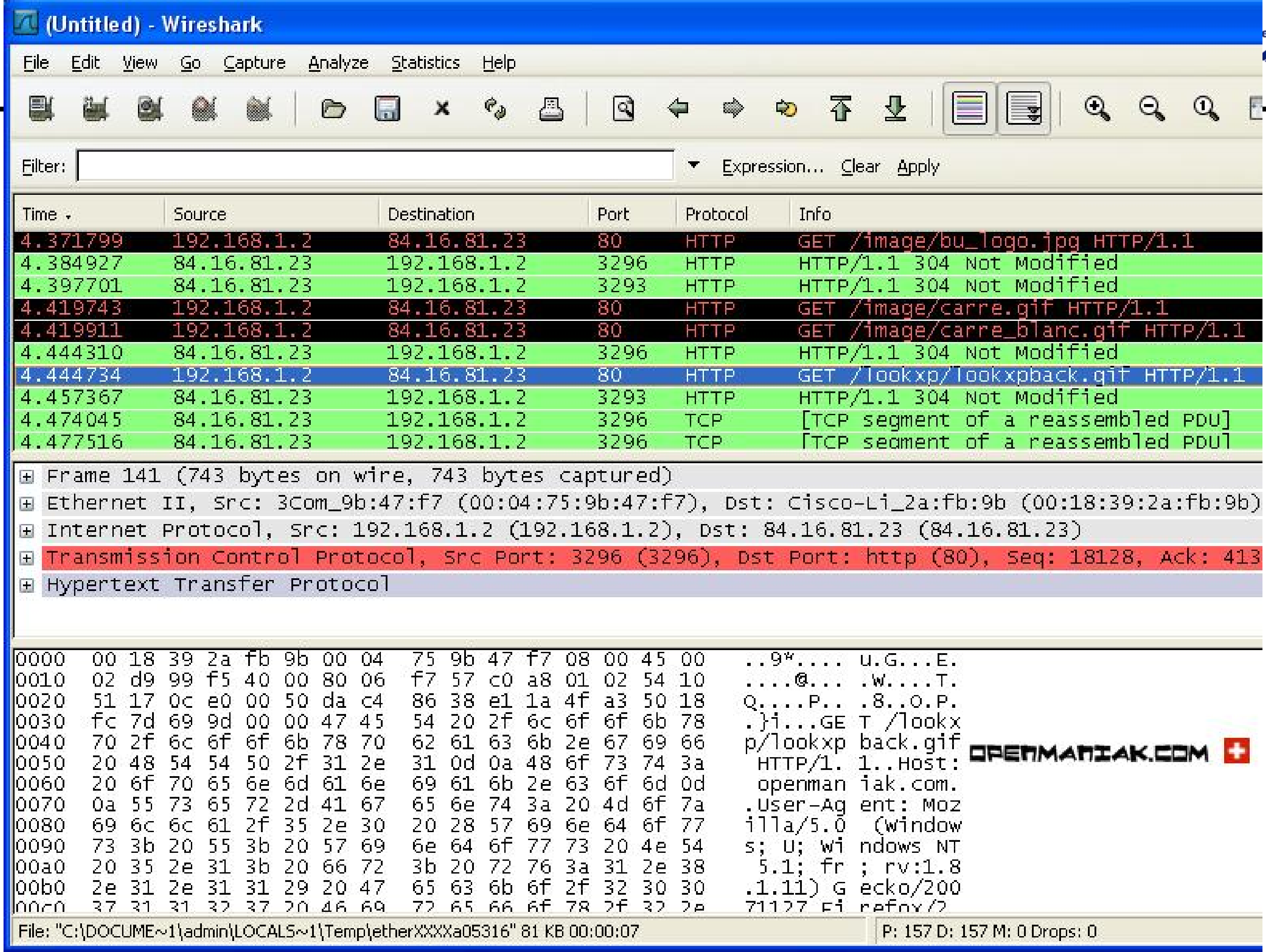
The inside and front of a Dell PowerEdge server, a computer designed to be mounted in a rack mount environment. Servers similar to this one are often used as web servers.

Multiple web servers may be used for a high-traffic website.

Working Principle

- Request:When you type a URL into your browser,the browser sends an HTTP request to the Web server corresponding to that website.
- Processing:After receiving the request,the Web server identifies the URL and looks for the relevant web page files.
- Response:The Web server sends the found web page content(such as HTML,images,etc.)as a response back to the browser via the HTTP protocol.
- Display:After receiving the response,the browser parses and displays the content to you,ultimately showing you the web page.


Communication process according to HTTP
User
Browser
Web Server
Entry
Screen
1. URL
http://www.example.com/index.html
Translating URL into HTTP request
2. HTTP Request
GET /index.html
Interpreting the request, retrieving required data files
4. Webpage
Display webpage index.html
Rendering data to final webpage
3. HTTP Response
status code + webpage data
Sending status of search and data of the webpage


(Untitled) - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter:
Expression... Clear Apply

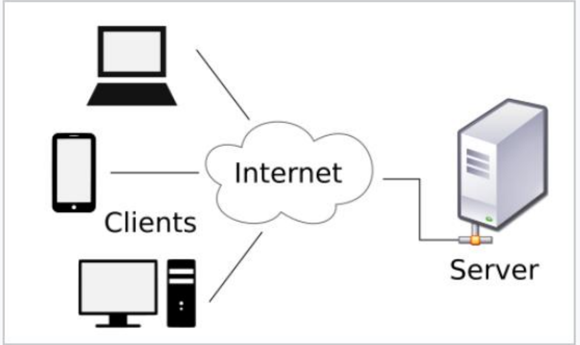
```
Time        Source         Destination    Port   Protocol  Info
4.371799    192.168.1.2    84.16.81.23    80     HTTP      GET /image/bu_logo.jpg HTTP/1.1
4.384927    84.16.81.23    192.168.1.2    3296   HTTP      HTTP/1.1 304 Not Modified
4.397701    84.16.81.23    192.168.1.2    3293   HTTP      HTTP/1.1 304 Not Modified
4.419743    192.168.1.2    84.16.81.23    80     HTTP      GET /image/carre.gif HTTP/1.1
4.419911    192.168.1.2    84.16.81.23    80     HTTP      GET /image/carre_blanc.gif HTTP/1.1
4.444310    84.16.81.23    192.168.1.2    3296   HTTP      HTTP/1.1 304 Not Modified
4.444734    192.168.1.2    84.16.81.23    80     HTTP      GET /lookxp/lookxpback.gif HTTP/1.1
4.457367    84.16.81.23    192.168.1.2    3293   HTTP      HTTP/1.1 304 Not Modified
4.474045    84.16.81.23    192.168.1.2    3296   TCP       [TCP segment of a reassembled PDU]
4.477516    84.16.81.23    192.168.1.2    3296   TCP       [TCP segment of a reassembled PDU]
```

```
⊞ Frame 141 (743 bytes on wire, 743 bytes captured)
⊞ Ethernet II, Src: 3Com_9b:47:f7 (00:04:75:9b:47:f7), Dst: Cisco-Li_2a:fb:9b (00:18:39:2a:fb:9b)
⊞ Internet Protocol, Src: 192.168.1.2 (192.168.1.2), Dst: 84.16.81.23 (84.16.81.23)
⊞ Transmission Control Protocol, Src Port: 3296 (3296), Dst Port: http (80), Seq: 18128, Ack: 413
⊞ Hypertext Transfer Protocol
```

```
0000  00 18 39 2a fb 9b 00 04  75 9b 47 f7 08 00 45 00   ..9*.... u.G...E.
0010  02 d9 99 f5 40 00 80 06  f7 57 c0 a8 01 02 54 10   ....@... .W....T.
0020  51 17 0c e0 00 50 da c4  86 38 e1 1a 4f a3 50 18   Q....P.. .8..O.P.
0030  fc 7d 69 9d 00 00 47 45  54 20 2f 6c 6f 6f 6b 78   .}i...GE T /lookx
0040  70 2f 6c 6f 6f 6b 78 70  62 61 63 6b 2e 67 69 66   p/lookxp back.gif
0050  20 48 54 54 50 2f 31 2e  31 0d 0a 48 6f 73 74 3a    HTTP/1. 1..Host:
0060  20 6f 70 65 6e 6d 61 6e  69 61 6b 2e 63 6f 6d 0d    openman iak.com.
0070  0a 55 73 65 72 2d 41 67  65 6e 74 3a 20 4d 6f 7a   .User-Ag ent: Moz
0080  69 6c 6c 61 2f 35 2e 30  20 28 57 69 6e 64 6f 77   illa/5.0  (Window
0090  73 3b 20 55 3b 20 57 69  6e 64 6f 77 73 20 4e 54   s; U; Wi ndows NT
00a0  20 35 2e 31 3b 20 66 72  3b 20 72 76 3a 31 2e 38    5.1; fr ; rv:1.8
00b0  2e 31 2e 31 31 29 20 47  65 63 6b 6f 2f 32 30 30   .1.11) G ecko/200
00c0  37 31 31 32 37 20 46 69  72 65 66 6f 78 2f 32 2e   71127 Fi refox/2.
```

File: "C:\DOCUME~1\admin\LOCALS~1\Temp\etherXXXXa05316" 81 KB 00:00:07   P: 157 D: 157 M: 0 Drops: 0

Software



PC clients connected to a web server via Internet

- A web server program plays the role of a server in a client–server model by implementing one or more versions of HTTP protocol, often including the HTTPS secure variant and other features and extensions that are considered useful for its planned usage.

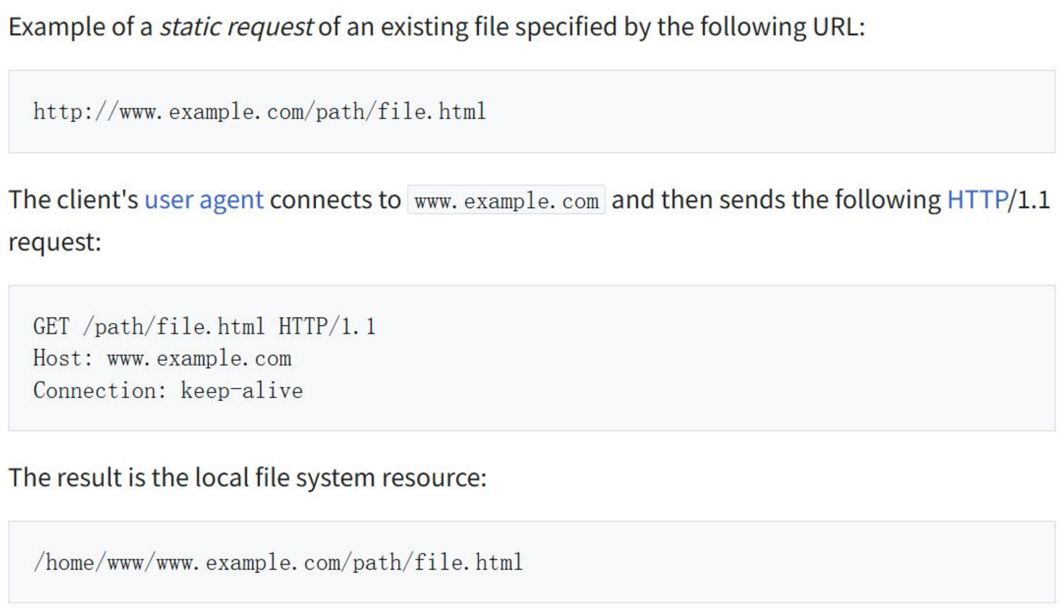
Most Popular Open-Source Web Server

APACHE
HTTP SERVER PROJECT
NGINX
Apache Tomcat
LIGHTTPD
fly light.
Caddy
THE ULTIMATE SERVER

Static content serving

Xi'an Jiaotong-Liverpool University 西交利物浦大学

- Static content serving: to be able to serve static content (web files) to clients via HTTP protocol.



Example of a *static request* of an existing file specified by the following URL:

```
http://www.example.com/path/file.html
```

The client's user agent connects to `www.example.com` and then sends the following HTTP/1.1 request:

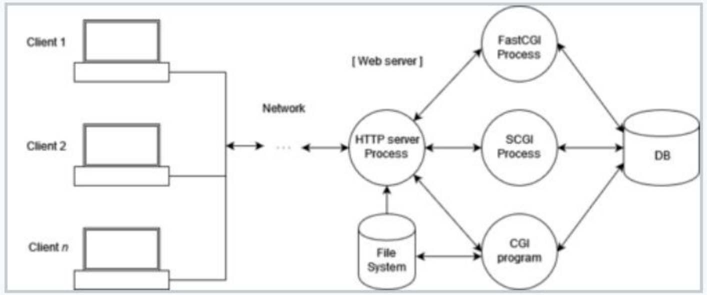
```
GET /path/file.html HTTP/1.1
Host: www.example.com
Connection: keep-alive
```

The result is the local file system resource:

```
/home/www/www.example.com/path/file.html
```

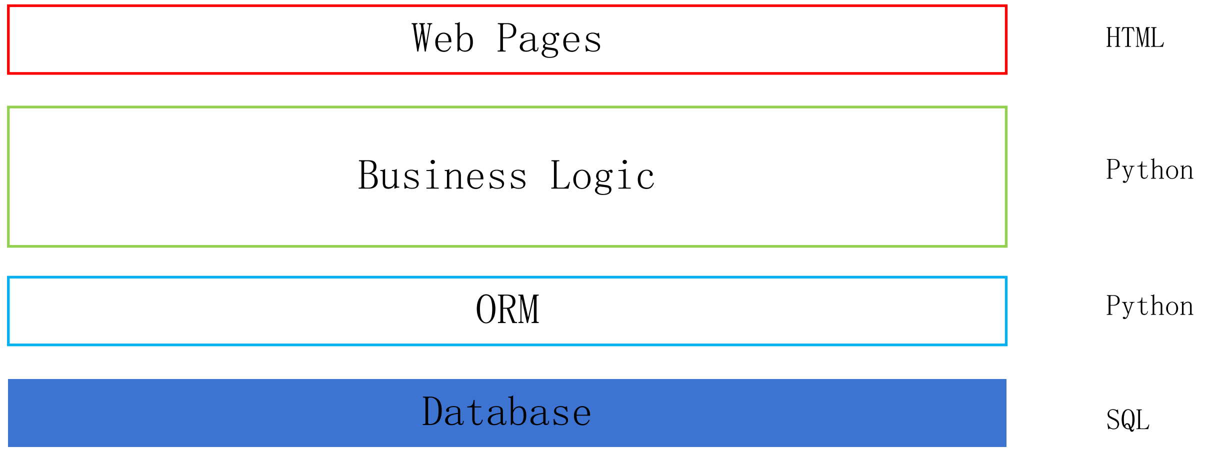
Dynamic content serving

- Dynamic content serving: to be able to serve dynamic content (generated on the fly) to clients via HTTP protocol.



PC clients communicating via network with a web server serving static and dynamic content

Layers

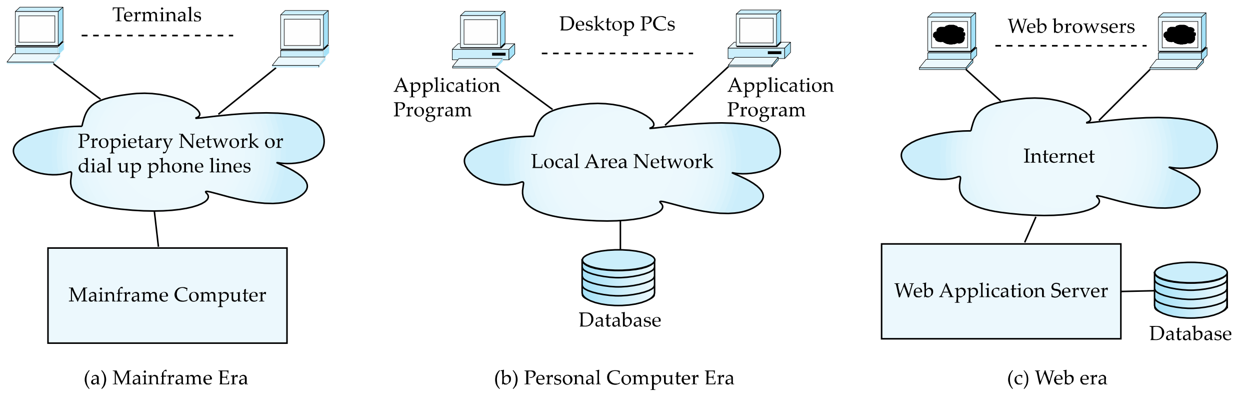

Web Pages
HTML
Business Logic
Python
ORM
Python
Database
SQL

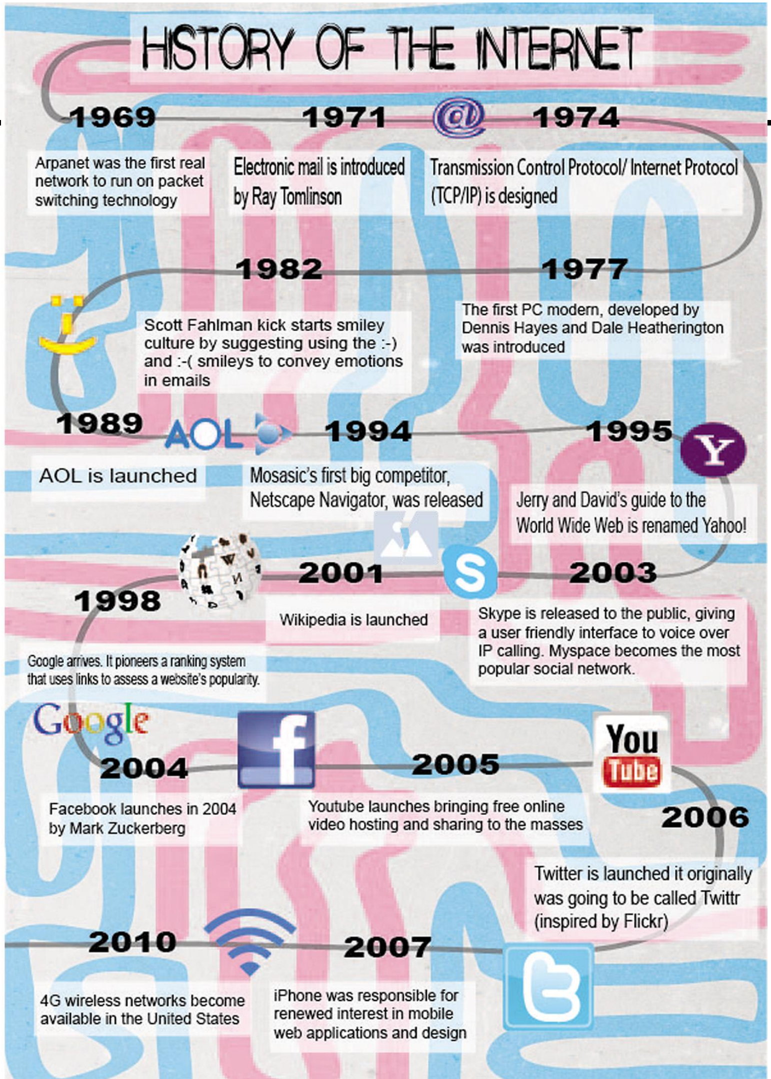
Application Programs and User Interfaces

- Most database users do not use a query language like SQL
- An application program acts as the intermediary between users and the database
  - Applications split into
    - front-end
    - middle layer
    - backend
- Front-end: user interface
  - Forms
  - Graphical user interfaces
  - Many interfaces are Web-based

Application Architecture Evolution

- Three distinct era’s of application architecture
  - Mainframe (1960’s and 70’s)
  - Personal computer era (1980’s)
  - Web era (mid 1990’s onwards)
  - Web and Smartphone era (2010 onwards)





HISTORY OF THE INTERNET

1969
Arpanet was the first real network to run on packet switching technology

1971
Electronic mail is introduced by Ray Tomlinson

1974
Transmission Control Protocol/ Internet Protocol (TCP/IP) is designed

1977
The first PC modern, developed by Dennis Hayes and Dale Heatherington was introduced

1982
Scott Fahlman kick starts smiley culture by suggesting using the :-) and :-(smileys to convey emotions in emails

1989
AOL is launched

1994
Mosasic's first big competitor, Netscape Navigator, was released

1995
Jerry and David's guide to the World Wide Web is renamed Yahoo!

1998
Google arrives. It pioneers a ranking system that uses links to assess a website's popularity.

2001
Wikipedia is launched

2003
Skype is released to the public, giving a user friendly interface to voice over IP calling. Myspace becomes the most popular social network.

2004
Facebook launches in 2004 by Mark Zuckerberg

2005
Youtube launches bringing free online video hosting and sharing to the masses

2006
Twitter is launched it originally was going to be called Twittr (inspired by Flickr)

2007
iPhone was responsible for renewed interest in mobile web applications and design

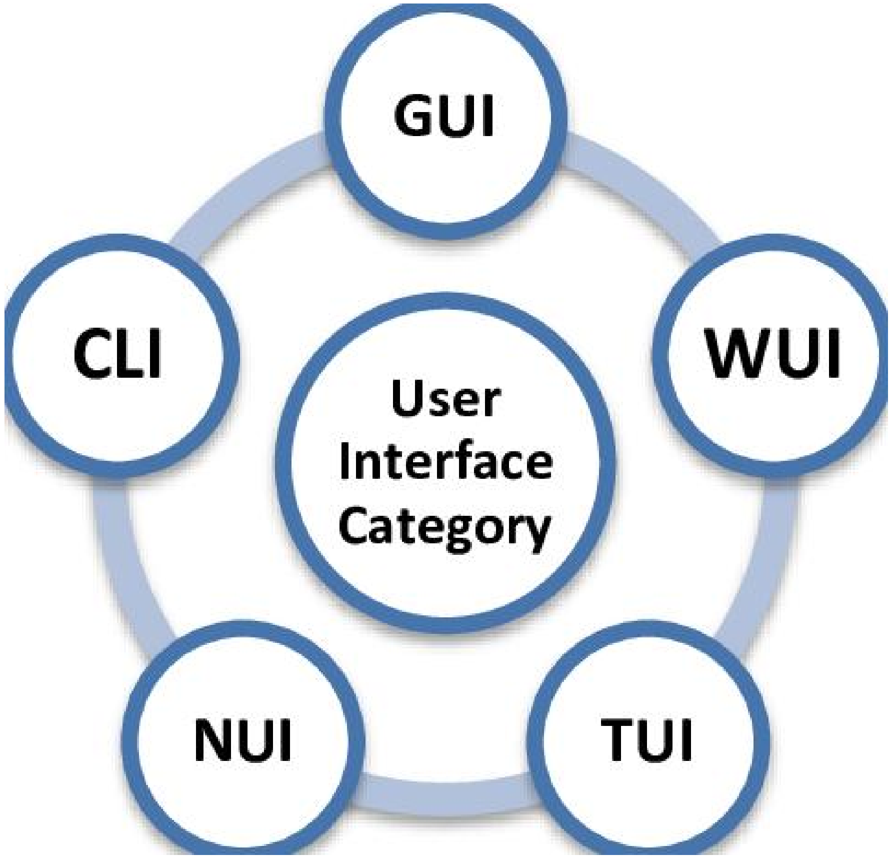
2010
4G wireless networks become available in the United States

Web Interface

Web browsers have become the de-facto standard user interface to databases

- Enable large numbers of users to access databases from anywhere
- Avoid the need for downloading/installing specialized code, while providing a good graphical user interface
  - Javascript, Flash and other scripting languages run in browser, but are downloaded transparently
- Examples: banks, airline and rental car reservations, university course registration and grading, an so on.

*Type



The World Wide Web

- The Web is a distributed information system based on hypertext.
- Most Web documents are hypertext documents formatted via the HyperText Markup Language (HTML)
- HTML documents contain
  - text along with font specifications, and other formatting instructions
  - hypertext links to other documents, which can be associated with regions of the text.
  - forms, enabling users to enter data which can then be sent back to the Web server

Uniform Resources Locators

- In the Web, functionality of pointers is provided by Uniform Resource Locators (URLs).
- URL example:

  http://www.acm.org/sigmod

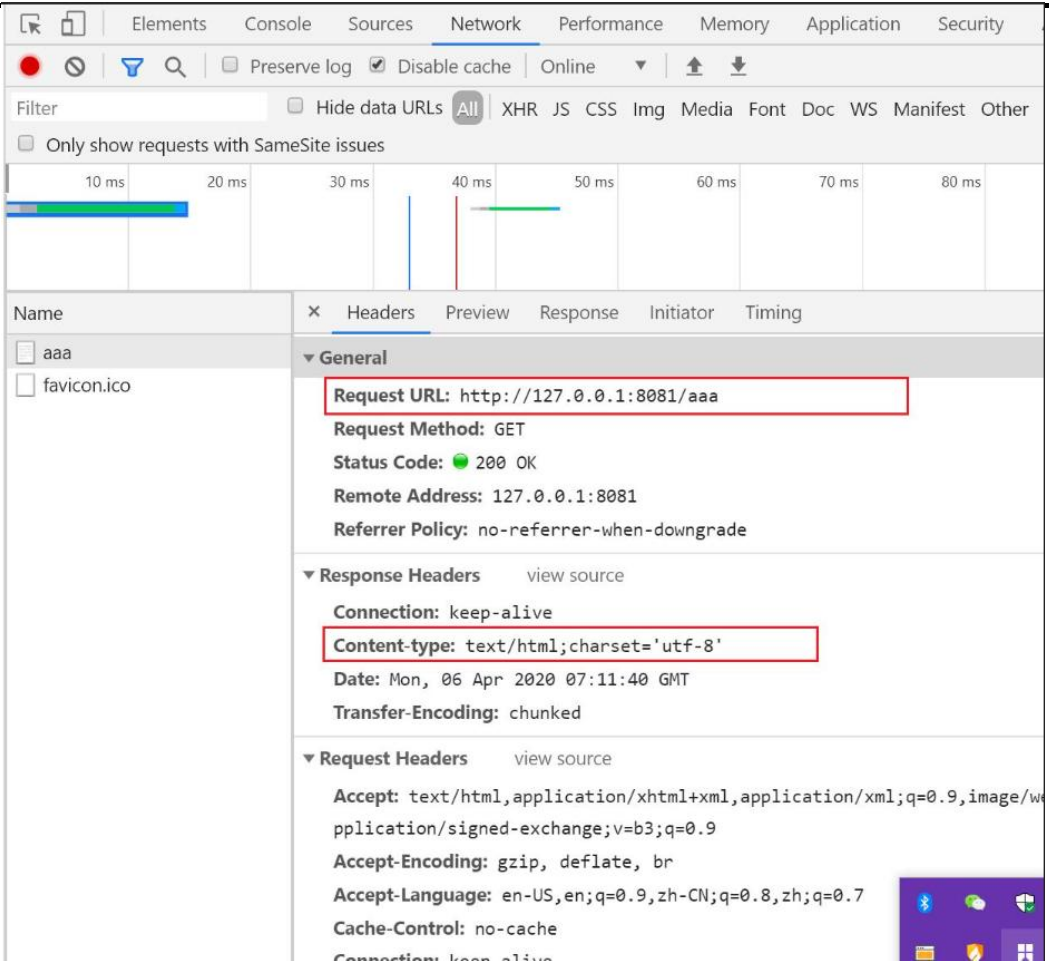
  - The first part indicates how the document is to be accessed
    - "http" indicates that the document is to be accessed using the Hyper Text Transfer Protocol.
  - The second part gives the unique name of a machine on the Internet.
  - The rest of the URL identifies the document within the machine.
- The local identification can be:
  - The path name of a file on the machine, or
  - An identifier (path name) of a program, plus arguments to be passed to the program
    - E.g., http://www.google.com/search?q=silberschatz

Ping

HTML and HTTP

- HTML provides formatting, hypertext link, and image display features
  - including tables, stylesheets (to alter default formatting), etc.
- HTML also provides input features
  - Select from a set of options
    - Pop-up menus, radio buttons, check lists
  - Enter values
    - Text boxes
  - Filled in input sent back to the server, to be acted upon by an executable at the server
- HyperText Transfer Protocol (HTTP) used for communication with the Web server

HTTP Header



Elements Console Sources Network Performance Memory Application Security
Preserve log
Disable cache
Online
Filter
Hide data URLs
All XHR JS CSS Img Media Font Doc WS Manifest Other
Only show requests with SameSite issues
10 ms 20 ms 30 ms 40 ms 50 ms 60 ms 70 ms 80 ms
Name
aaa
favicon.ico
Headers Preview Response Initiator Timing

General

Request URL: http://127.0.0.1:8081/aaa
Request Method: GET
Status Code: 200 OK
Remote Address: 127.0.0.1:8081
Referrer Policy: no-referrer-when-downgrade

Response Headers view source

Connection: keep-alive
Content-type: text/html;charset='utf-8'
Date: Mon, 06 Apr 2020 07:11:40 GMT
Transfer-Encoding: chunked

Request Headers view source

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/we
pplication/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9,zh-CN;q=0.8,zh;q=0.7
Cache-Control: no-cache
Connection: keep-alive

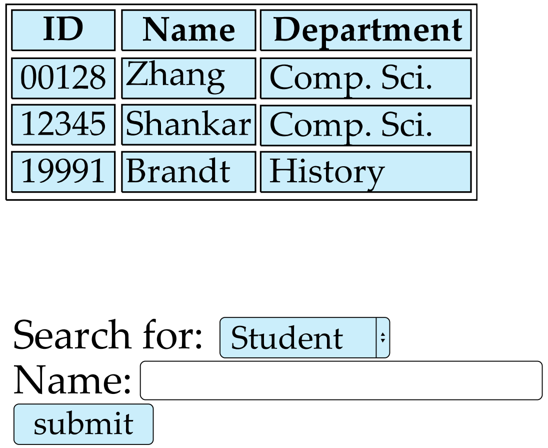
Sample HTML Source Text

Xi'an Jiaotong-Liverpool University 西交利物浦大学

5~11

```
<html>

<body>

  <table border>
     <tr> <th>ID</th> <th>Name</th> <th>Department</th> </tr>
     <tr> <td>00128</td> <td>Zhang</td> <td>Comp. Sci.</td> </tr>
     ….

  </table>

   <form action="PersonQuery" method=get>
     Search for:
        <select name="persontype">
            <option value="student" selected>Student </option>
            <option value="instructor"> Instructor </option>
        </select> <br>
     Name: <input type=text size=20 name="name">
     <input type=submit value="submit">

  </form>

</body> </html>
```

Display of Sample HTML Source



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

Search for: Student

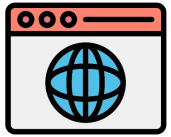
Name:

submit

Javascript

Xi'an Jiaotong-Liverpool University 西交利物浦大学

- What is JavaScript?
  - Client-side scripting language – runs in the browser
  - Dynamic interaction – makes web pages interactive
  - Cross-platform – supported by all modern browsers
- Core Features
  - Lightweight interpreted language
  - Supports object-oriented programming
  - Seamless integration with HTML/CSS
  - Rich ecosystem and libraries
- Main Applications
  - Front-end web development
  - Server-side development (Node.js)
  - Mobile app development
  - Game development



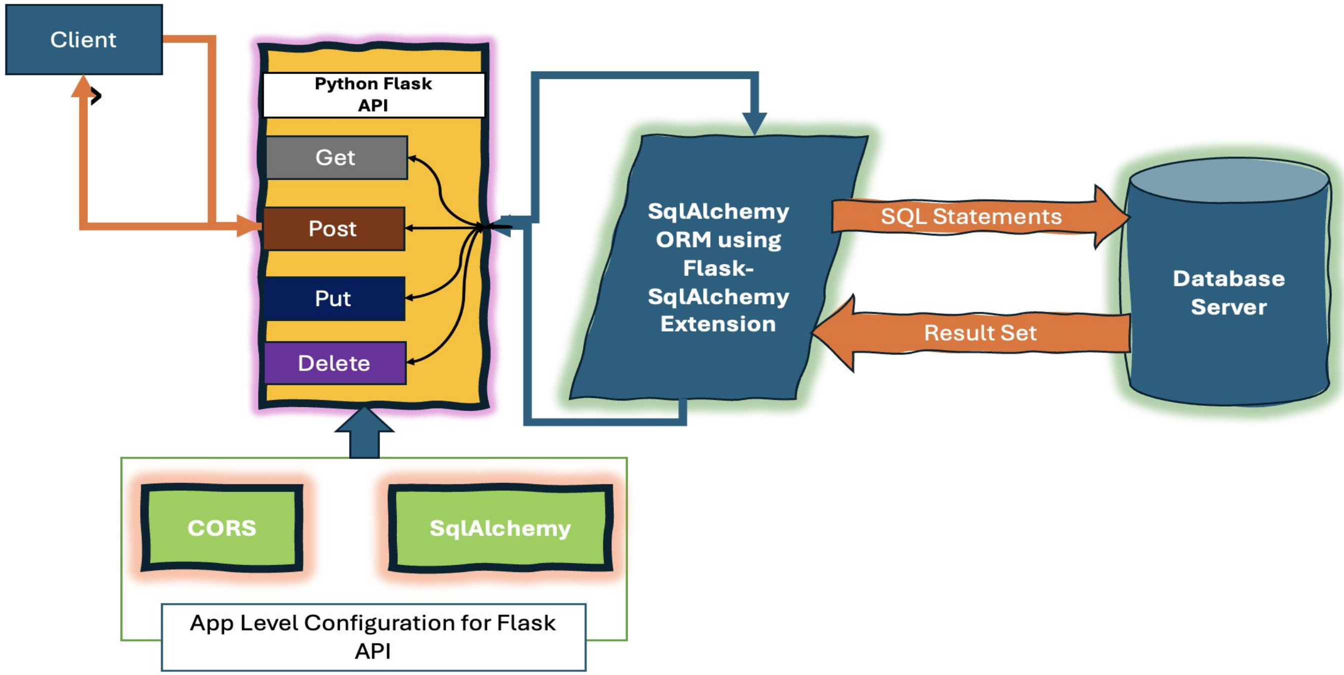
This Will be taught in Lab5!

Web Servers

- A Web server can easily serve as a front end to a variety of information services.
- The document name in a URL may identify an executable program, that, when run, generates a HTML document.
  - When an HTTP server receives a request for such a document, it executes the program, and sends back the HTML document that is generated.
  - The Web client can pass extra arguments with the name of the document.
- To install a new service on the Web, one simply needs to create and install an executable that provides that service.
  - The Web browser provides a graphical user interface to the information service.

Flask

- Flask is a lightweight Python web framework ideal for building small to medium-sized web applications.



Routing

Xi'an Jiaotong-Liverpool University 西交利物浦大学

- Modern web applications use meaningful URLs to help users. Users are more likely to like a page and come back if the page uses a meaningful URL they can remember and use to directly visit a page.

13

Variable Rule

- You can add variable sections to a URL. Your function then receives the as a keyword argument.

Rendering Templates

- Generating HTML from within Python is not fun, and actually pretty cumbersome because you have to do the HTML escaping on your own to keep the application secure.
- Templates can be used to generate any type of text file. For web applications, you’ll primarily be generating HTML pages, but you can also generate markdown, plain text for emails, and anything else.

Receiving User Inputs

- Get Method
  - GET is a request method in the HTTP protocol used to request access to a specific resource on a server. It is one of the most commonly used HTTP request methods, primarily used to retrieve data from a server.
  - Parameters are passed in the URL, which reduces security and imposes a request length limit.
- Post Method
  - POST is also a request method in the HTTP protocol used to submit data to a server to create or update resources. It is commonly used for operations such as submitting form data and uploading files.
  - Parameters are passed in the request body, which provides increased security and an unlimited request length.

Object-oriented Programming

- Object-oriented programming (OOP) is a programming paradigm based on the object – a software entity that encapsulates data and function(s).
- An OOP computer program consists of objects that interact with one another.
- Main features: Encapsulation, Inheritance, and Polymorphism

Object Orientation

- **Object-relational data model** provides richer type system
  - with complex data types and object orientation
- Applications are often written in object-oriented programming languages
  - Type system does not match relational type system
  - Switching between imperative language and SQL is troublesome
- Approaches for integrating object-orientation with databases
  - Build an **object-relational database**, adding object-oriented features to a relational database
  - Automatically convert data between programming language model and relational model; data conversion specified by **object-relational mapping**
  - Build an **object-oriented database** that natively supports object-oriented data and direct access from programming language

Object-Relational Mapping

- Object-relational mapping (ORM) systems allow
  - Specification of mapping between programming language objects and database tuples
  - Automatic creation of database tuples upon creation of objects
  - Automatic update/delete of database tuples when objects are update/deleted
  - Interface to retrieve objects satisfying specified conditions
    - Tuples in database are queried, and object created from the tuples
- Details in Section 9.6.2
  - Hibernate ORM for Java
  - Django ORM for Python

Object-Relational Database Systems

2

- User-defined types

  - **create type** *Person*
      (*ID* **varchar**(20) **primary key**,
       name **varchar**(20),
       address **varchar**(20)) **ref from**(*ID*); /* More on this later */
    create table *people* **of** *Person*;

- Table types

  - **create type** *interest* **as table** (
       topic **varchar**(20),
       degree_of_interest **int**);
    create table *users* (
       ID **varchar**(20),
       name **varchar**(20),
       interests interest);

-

Type and Table Inheritance

3

- Type inheritance
  - **create type** *Student* **under** *Person*
    (*degree* **varchar**(20)) ;
    create type *Teacher* **under** *Person*
    (*salary* **integer**);
- Table inheritance syntax in PostgreSQL and oracle
  - **create table** *students*
        (*degree* **varchar**(20))
        inherits *people*;
    create table *teachers*
        (*salary* **integer**)
        inherits *people*;
  - **create table** *people* **of** *Person*;
    create table *students* **of** *Student*
        under *people*;
    create table *teachers* **of** *Teacher*
        under *people*;

Reference Types

4

- Creating reference types

  -
    ```
    create type Person
        (ID varchar(20) primary key,
         name varchar(20),
         address varchar(20))
         ref from(ID);
    create table people of Person;
    create type Department (
         dept_name varchar(20),
         head ref(Person) scope people);
    create table departments of Department
    insert into departments values ('CS', '12345')
    ```
  - System generated references can be retrieved using subqueries
    - `(select ref(p)   from people as p    where ID = '12345')`
- Using references in **path expressions**

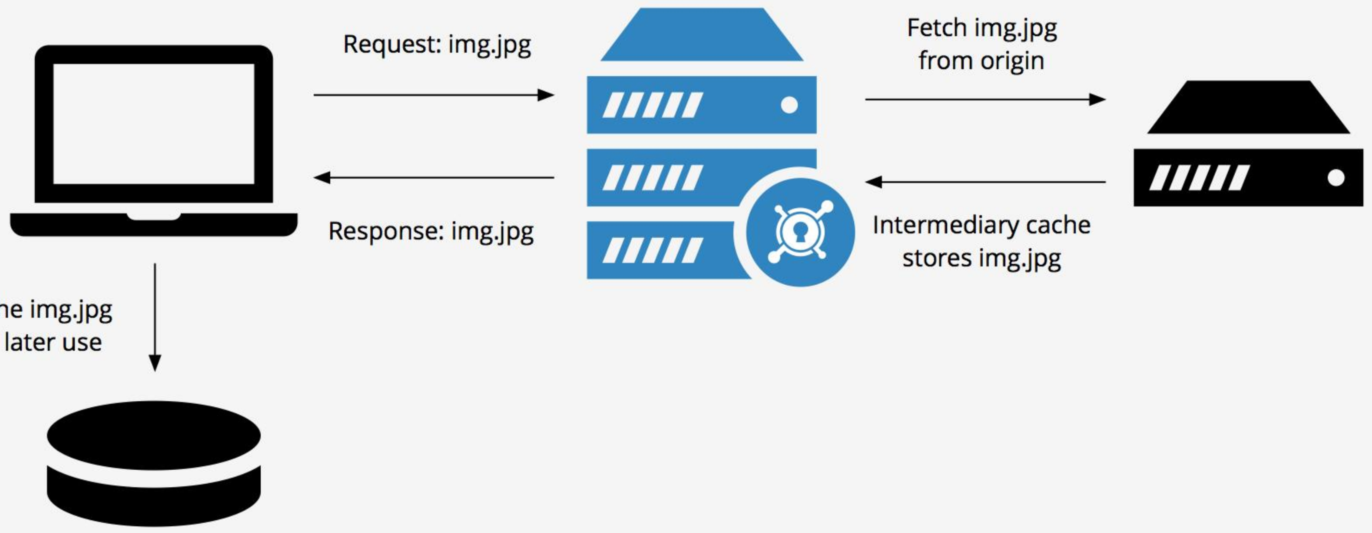
  -
    ```
    select head->name, head->address
    from departments;
    ```

Integration

- A simple web application

17

Performance


Request: img.jpg
Response: img.jpg
Fetch img.jpg from origin
Intermediary cache stores img.jpg
Cache img.jpg for later use

Web Cache

Client
Browser Cache
Intermediary Cache
Origin Server

*Web Services

- Allow data on Web to be accessed using remote procedure call mechanism
- Two approaches are widely used
  - **Representation State Transfer (REST)**: allows use of standard HTTP request to a URL to execute a request and return data
    - Returned data is encoded in XML
  - **Big Web Services**:
    - Uses XML representation for sending request data, as well as for returning results
    - Standard protocol layer built on top of HTTP
    - See Section 23.7.3