

Basi di Dati e Web

Sistemi informativi su Web ...

Il Web sta diventando un'interfaccia generale per l'accesso a sistemi informativi di ogni tipo.

Si ipotizza che in futuro diventi l'interfaccia standard o prevalente.

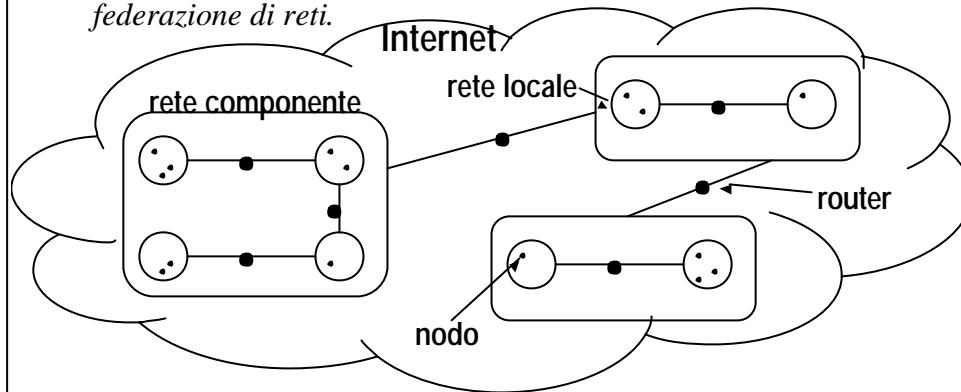
Sicuramente i *sistemi informativi su Web* (**WIS**) cresceranno rapidamente in quantità e funzionalità.

Qual è il ruolo delle basi di dati nei WIS?

Vediamo alcuni esempi di WIS.

Internet

Internet è una rete eterogenea formata da calcolatori dislocati in tutte le parti del mondo, e comunicanti attraverso il protocollo di trasmissione **TCP/IP** (*Transmission Control Protocol/Internet Protocol*). Più precisamente Internet è definita come una *federazione di reti*.



Internet

- Le reti locali sono collegate attraverso un **router** (calcolatore dedicato all'interconnessione di reti), un **modem** e una linea telefonica ad altre reti. I calcolatori di una rete locale sono i nodi della federazione. Le reti possono essere articolate in reti più ampie (di campus, universitaria, delle università italiane, ecc.).

Indirizzi numerici e simbolici

Ciascun nodo (calcolatore) della rete Internet ha un indirizzo numerico, chiamato **IP address**, costituito da una stringa di 32 bit e di solito rappresentato per mezzo di quattro interi (ciascuno compreso fra 0 e 255) separati da punti.

Esempio: 193.204.187.87

A ciascun calcolatore è anche associato un nome simbolico, costituito da identificatori separati da punti.

Esempio: malerba.di.uniba.it

che viene di solito utilizzato nelle applicazioni per fare riferimento ad esso.

Indirizzi numerici e simbolici

Le applicazioni, tuttavia, hanno bisogno di convertire il nome simbolico nell'indirizzo numerico. Per questo esse si rivolgono ad uno specifico calcolatore, chiamato **domain name server** (DNS), che utilizza tabelle locali di corrispondenza fra nomi e indirizzi IP (per i nomi di uso più frequente) e interroga altri server se non dispone della conversione nelle tabelle stesse.

Esempio: 193.204.187.131 → DNS del DIB

Attraverso la struttura ad indirizzi, ciascun nodo Internet può connettersi direttamente ad ogni altro nodo.

La comunicazione

La comunicazione avviene attraverso una organizzazione stratificata:

- Il protocollo IP, utilizzando le funzionalità di interconnessione, trasmette pacchetti ("**IP datagram**") di dimensione relativamente piccola (ad esempio, 500 – 1500 caratteri) scegliendo di volta in volta il percorso ritenuto migliore, senza garantire né verificare la consegna.
- Il protocollo TCP simula una connessione fra nodo mittente e nodo destinatario: fraziona il messaggio da inviare in pacchetti, numerati progressivamente e invia ciascuno separatamente utilizzando il protocollo IP. Poiché i vari pacchetti sono inviati indipendentemente e la loro consegna non è garantita, il destinatario può riceverli in disordine e non tutti, e deve quindi riordinarli e richiedere il nuovo invio per quelli mancanti.

La comunicazione

Sopra TCP è implementato un insieme di protocolli di più alto livello, quali SMTP, per la posta elettronica, FTP, per il trasferimento di file, ed HTTP, per il trasferimento di ipertesti, contenenti eventualmente anche codice eseguibile. Quest'ultimo protocollo è alla base delle cosiddette applicazioni Internet o applicazioni Web, dove il termine Web (World Wide Web, WWW) indica il complesso delle pagine, mutuamente collegate, messe a disposizione dai serveri HTTP collegati ad Internet.

Il World Wide Web

Iper testo: documento testuale con una struttura non sequenziale, costituito da varie porzioni fra loro collegate, al fine di favorire la consultazione a seconda delle esigenze, senza la rigidità della struttura fisica sequenziale.

Ipermedia: ipertesto *multimediale*, dove le porzioni di documento non sono solo di tipo testuale ma anche di tipo suono, immagine, video, ecc.



Le porzioni di documento possono essere distribuite sui nodi di una rete di calcolatori.



Il **World Wide Web** è un ipertesto multimediale distribuito su tutta la rete Internet. Le porzioni di questo mega ipermedia sono prodotte e mantenute da milioni di persone diverse.

HTML

I documenti che costituiscono la struttura ipertestuale del Web sono scritti nel linguaggio **HTML** (*HyperText Markup Language*), che permette di formattare i documenti e di descrivere i legami ipertestuali.

HTML permette di esprimere due livelli di formattazione:

- **Fisica**, come l'allineamento del testo a sinistra, al centro, ecc., oppure l'uso di font in corsivo o in grassetto.
- **Logica**, come l'enfaticizzazione del testo, il titolo del documento, ecc.

HTML

La formattazione logica permette di specificare che una frase deve essere enfatizzata, ma non come l'enfatizzazione dev'essere realizzata. Questa caratteristica è molto importante, perché lo stesso documento potrebbe essere trasmesso per la visualizzazione a client diversi: un client con capacità grafiche e gestione di colori potrebbe utilizzare un carattere corsivo o un colore specifico, mentre un client testuale monocromatico potrebbe usare il lampeggio.

I documenti HTML possono essere visualizzati per mezzo dei **browser Web** (e.g., Netscape Navigator o Microsoft Internet Explorer), che possono accedere sia a documenti locali e sia a documenti remoti (grazie al protocollo HTTP).

La struttura del documento HTML

```
<HTML>
<HEAD>
...
head elements
...
</HEAD>
<BODY>
...
body elements
...
</BODY>
</HTML>
```

La formattazione del documento è espressa mediante un sistema di **tag**.

Le informazioni residenti nella testa descrivono le proprietà del documento come:

- Titolo: ogni documento deve avere un titolo (e.g., <TITLE> Titolo del Documento </TITLE>).
- Link
- Toolbar
- Meta-informazioni

La struttura del documento HTML

```
<HTML>
<HEAD>
...
head elements
...
</HEAD>
<BODY>
...
body elements
...
</BODY>
</HTML>
```

Il testo del documento ne costituisce il corpo. Gli elementi del corpo possono essere:

- **Intestazioni:** <H1> Intestazione di primo livello </H1>
- **Paragrafi:** <P> definisce la fine/inizio di un paragrafo;
- **Collegamenti ipertestuali a file locali o remoti, identificati mediante un *URL* (*Uniform Resource Locator*):**

```
<a href="filename.html"> text that responds to link </a>
```

```
<a href="www.di.uniba.it/index.htm">Home page DIB</a>
```

- **Collegamenti ipertestuali a porzioni diverse dello stesso documento (*ancore*):**

```
<A HREF="#Potomac">Boston</A> is a historic city and
```

...

```
<A NAME="Potomac">The Potomac river flows into ...
```

La struttura del documento HTML

- **Grafici:**
- **Testo preformattato:**

```
<pre>
```

```
We have indented with space.
And used the carriage return to jump to a
New line.
```

```
Here we
use
spaces to
create a
text table.
```

```
</pre>
```

La struttura del documento HTML

- Liste o elenchi:

My Unordered List:

```
<ul>
<lh> Title (Optional) </lh>
<li> Item 1
<li> Item 2
<li> Item 3
</ul>
```

La struttura del documento HTML

- Stili

Grassetto	<code> ... </code>
Corsivo	<code><I> ... </I></code>
Sottolineato	<code><U> ... </U></code>
Macchina da scrivere	<code><TT> ... </TT></code>

- Tabelle:

```
<TABLE ><TR> <TD> A(1,1) </TD> <TD> A(1,2) </TD></TR>
<TR> <TD> A(2,1) </TD> <TD> A(2,2) </TD> </TABLE>
```


Il modello client-server

- Il modello alla base delle applicazioni Web è quello classico *cliente-servente*. Un servente Web, o servente HTTP, è un programma, in esecuzione su una macchina collegata ad Internet, in attesa di rispondere a richieste che pervengono dai clienti HTTP (i browser Web).

Il modello client-server

Lo scambio di informazioni cliente-servente avviene in quattro fasi:

1. **Apertura della connessione:** il browser (nel ruolo di cliente) contatta il servente, con l'indirizzo e il protocollo indicati nella URL, verificandone la correttezza e la disponibilità.
2. **Richiesta:** il cliente invia al servente HTTP un messaggio, con la richiesta di un servizio, i dettagli della risorsa ed eventuali parametri.
3. **Risposta:** il servente comunica al cliente l'esito della richiesta e, in caso positivo, i risultati, preceduti da informazioni sul tipo degli stessi.
4. **Chiusura della connessione:** la connessione si conclude, senza che il servente mantenga memoria alcuna delle operazioni svolte.

Il modello client-server

N.B.: il protocollo HTTP non ha memoria (è *stateless*): nel caso di operazioni complesse, realizzate attraverso operazioni multiple, il server non è in grado di mantenere informazioni sulle operazioni già effettuate e sul loro esito. Questa è una limitazione del protocollo, giustificata dalla semplicità di gestione di richieste concorrenti.

Pubblicazione e consultazione sul Web: comunicazione unilaterale

L'obiettivo originario del Web: mettere a disposizione informazioni, in particolare sotto forma di pagine HTML.

Sin dall'inizio molte organizzazioni (e.g., università) hanno reso disponibili su Web pagine HTML opportunamente predisposte. Gli utenti di questo servizio *consultano* le informazioni *pubblicano* su Web.

Con il crescere delle informazioni da offrire e delle esigenze, è diventato sempre più difficile gestire *manualmente* insiemi di documenti HTML, per varie ragioni:

Pubblicazione e consultazione sul Web: comunicazione unilaterale

L'obiettivo originario del Web: mettere a disposizione informazioni, in particolare sotto forma di pagine HTML.

- Nelle pagine HTML sono presenti contemporaneamente
 - le informazioni in quanto tali (i dati)
 - L'organizzazione ipertestuale
 - La presentazione grafica

l'aggiornamento delle pagine al variare (frequente) dei dati può diventare un'operazione assai difficile; se poi variano anche la struttura ipertestuale o la presentazione, allora la manutenzione si complica ulteriormente.

Pubblicazione e consultazione sul Web: comunicazione unilaterale

- In un sito Web, c'è solitamente molta ridondanza: è necessario infatti ripetere informazioni per favorire la leggibilità e la navigazione.

- In tutte le pagine relative ai professori di un dipartimento universitario comparirà il nome del dipartimento. Se questo dovesse cambiare nome, la modifica riguarderebbe tutte le pagine

Questi problemi vengono alleviati se i dati sono memorizzati in una base di dati: le pagine possono essere costruite in modo automatico

- In modo statico (fuori linea)
- In modo dinamico (in-linea)

Inoltre è possibile separare i vari aspetti ed eliminare i problemi connessi alla ridondanza.

Publicazione e consultazione sul Web: comunicazione unilaterale

Un'altra ragione che spinge a far interagire siti Web e basi di dati si presenta nei casi in cui **le informazioni stesse siano già organizzate in basi di dati.**



Disponendo di un patrimonio informativo, perché non pubblicarlo?

Il costo di pubblicazione del *contenuto* sarebbe basso.



Transazioni sul Web: comunicazione bilaterale

Perché usare il Web solo per pubblicare?

Il caricamento dei dati nel database potrebbe avvenire attraverso il Web, ad esempio, mediante dei form.

Si può anche pensare a distribuire la funzionalità a seconda delle responsabilità dirette

- ogni docente è autorizzato ad aggiornare la propria Home Page



Il browser diventa un'interfaccia standard per sistemi informativi.



Emergono esigenze tipiche del settore delle basi di dati:

- Identificazione utenti e sicurezza
- Affidabilità e gestione delle transazioni

WIS: nuove applicazioni

- Commercio elettronico (attività di vendita attraverso il Web)
 - Se i cataloghi sono la principale applicazione dei siti di consultazione, è naturale pensare alla “vendita per corrispondenza”
 - A differenza della vendita per corrispondenza tradizionale, dove sistema informativo e base di dati sono accessibili solo dal personale interno, viene offerto direttamente all’acquirente l’accesso al sistema.
 - I problemi di sicurezza e affidabilità sono essenziali
 - Si possono avere funzionalità aggiuntive:
 - Commenti
 - Discussioni
 - Negoziazioni



WIS: nuove applicazioni

- Forum di discussione: ad esempio, recensioni di libri
- Sistemi di supporto al lavoro cooperativo
 - Condivisione controllata di documenti
 - Coordinamento delle attività e dei relativi flussi (*workflow management systems*)
 - Internet: infrastruttura utilissima per la realizzazione di sistemi di workflow;
 - I workflow richiedono spesso una certa strutturazione delle attività e il supporto di strumenti SW specifici → **intranet** (rete privata realizzata con la stessa tecnologia di Internet) o **extranet** (reti controllate ma non circoscritte a una singola organizzazione)

WIS: nuove applicazioni

- Le extranet portano a integrare i sistemi di organizzazioni diverse che interagiscono strettamente:
 - Un'azienda e i suoi fornitori
 - Agenzie di viaggio e tour operator
 - Banche diverse ai fini di operazioni interbancarie
- In questi casi si parla di collaborazioni “business to business”.



WIS: nuove applicazioni

- Cooperazione lasca:
 - “lista di link utili” o di “siti correlati” che si trova in molte pagine
- Cooperazione stretta:
 - Ricerca bibliografica su più siti
 - Data warehousing su web → **data farming**
 - Stante la grande disponibilità di informazioni sul Web, una organizzazione può pensare di acquisire informazioni di interesse proprio attraverso il Web
 - Estrazione delle informazioni, integrazione, risoluzione delle ambiguità, qualità dei dati



WIS: nuove applicazioni

- **Sistemi embedded:**

- Hanno come interfacce (verso utente o altri sistemi) dei dispositivi specifici (molto diversi dai PC), quali
 - Telefoni cellulari
 - Calcolatori palmari senza filo
 - Strumenti di controllo di impianti industriali
- Sono caratterizzati da limitatezza delle risorse
 - Potenza di calcolo
 - Memoria
 - Larghezza di banda di collegamento
 - Dimensione del display (se esiste)
- Attualmente pochi aspetti specifici delle BD ma potrebbero crescere.



WIS: nuove applicazioni

- **Sistemi embedded (cont.)**

- Sistema di bordo di un'automobile
 - Possibilità di acquisire, attraverso l'integrazione con un sistema GPS in grado di determinare la posizione in qualunque punto della superficie terrestre, informazioni meteorologiche o turistiche sulla zona in cui si trova.
- Dispositivi domestici
 - Manutenzione e aggiornamento SW via rete



Accedere alle basi di dati attraverso il Web

- Il servizio comunemente richiesti da un browser Web è l'invio di uno specifico *file HTML*, memorizzato in una cartella accessibile al servente HTTP.
- In realtà i serventi Web possono anche richiamare, su richiesta di un cliente, dei *programmi* ad essi accessibili, trasmettendogli anche eventuali parametri specificati nella richiesta pervenuta dal cliente.

Accedere alle basi di dati attraverso il Web

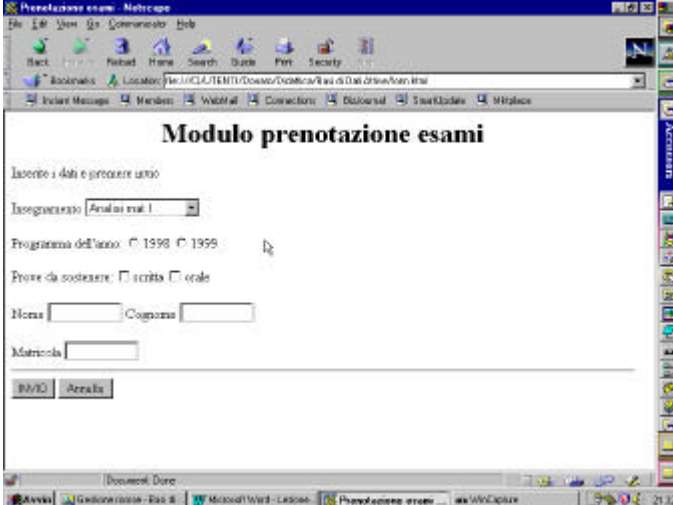


- **Gateway** (entrata, passaggio): un qualunque programma richiamato da un server Web.
- Un gateway può essere scritto o in un linguaggio compilato come C, C++, e Pascal, o in un linguaggio interpretato come Perl e Tcl, o ancora in un linguaggio di comandi di sistema operativo.
- Un gateway viene richiamato per mezzo di una URL analoga a quella per accedere ai file HTML.

Accedere alle basi di dati attraverso il Web

- I parametri passati ad un gateway possono essere specificati mediante un particolare elemento HTML, chiamato *form* (o maschera).
- Un elemento form è racchiuso tra le tag <form> e </form> ed ha un attributo obbligatorio, Action, che specifica la URL cui il contenuto del modulo dev'essere inviato; di solito è un server HTTP, ma potrebbe anche essere un altro server (e.g, di posta elettronica). L'attributo method ha due possibili valori: get (default) e post che specificano la modalità di trasmissione dei parametri.

Accedere alle basi di dati attraverso il Web



The screenshot shows a web browser window with the title 'Prenotazione esami - Netscape'. The address bar shows the URL 'file:///C:/C:\UTENTE\Domini\Tronca\file:///D:/aldris/for.html'. The browser window displays a form titled 'Modulo prenotazione esami'. The form contains the following fields and controls:

- A dropdown menu for 'Insegnamento' with 'Analisi mat. I' selected.
- Radio buttons for 'Programma dell'anno' with '1998' selected and '1999' unselected.
- Radio buttons for 'Prove da sostenere' with 'scritta' selected and 'orale' unselected.
- Text input fields for 'Nome' and 'Cognome'.
- A text input field for 'Matricola'.
- Buttons for 'Inizio' and 'Annulla'.

Accedere alle basi di dati attraverso il Web

```
<html>
<head> <title>Prenotazione esami </title> </head>
<body>
<h1 align=center>Modulo prenotazione esami</h1>
  Inserite i dati e premere invio
<form action="http:// di.uniba.it/cgi-bin/prenota" method=post>
  Insegnamento
  <select>
    <option name="codice" value="Mat001"> Analisi mat I
    <option name="codice" value="Mat003"> Analisi mat II
    <option name="codice" value="Fis001"> Fisica gen I
    <option name="codice" value="Inf001"> Fond Informatica I
    <option name="codice" value="Mat002"> Geometria I
  </select><p>
```

Accedere alle basi di dati attraverso il Web

```
  Programma dell'anno:
  <input type="radio" name="anno" selected value="1998">1998
  <input type="radio" name="anno" value="1999">1999<p>
  Prove da sostenere:
  <input type="checkbox" name="prova" value="scritta">scritta
  <input type="checkbox" name="prova" value="orale">orale<p>
  Nome <input type="text" name="nome" size="10">
  Cognome <input type="text" name="cognome" size="10"><p>
  Matricola <input type="password" name="matr" size="10"><hr>
  <input type="submit" value="INVIO">
  <input type="reset" value="Annulla">
</form>
</body>
</html>
```

Accedere alle basi di dati attraverso il Web

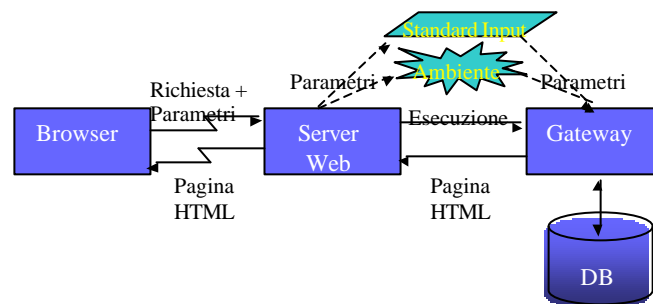
- Vari elementi possono essere inseriti in una form per inserire i dati. L'esempio ne mostra alcuni:
- L'elemento select permette di scegliere un elemento (o più, se viene specificato l'attributo multiple) in una lista di stringhe selezionabili; i valori scelti vengono inviati al server (secondo modalità stabilita dal parametro method), sotto forma di coppie *nome=valore* (e.g., codice=Mat003).

Accedere alle basi di dati attraverso il Web

- L'elemento input permette di specificare campi di vario tipo, sulla base del valore degli attributi, in particolare dell'attributo type. Nell'esempio abbiamo
 - text, che permette di inserire campi testuali di lunghezza definita;
 - password, analogo, ma non visualizza i dati inseriti;
 - radio, che permette di selezionare uno ed un sol valore in un dato insieme.
 - reset, la cui selezione annulla tutti gli inserimenti effettuati;
 - submit, la cui selezione causa l'invio del contenuto del modulo alla URL specificata.

Common Gateway Information

- Il meccanismo di comunicazione fra server Web e gateway è detto *Common Gateway Interface (CGI)*.

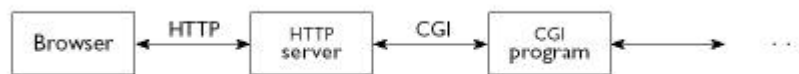


Common Gateway Information

1. L'utente richiede il servizio, selezionando, nel caso specifico, il bottone di submit, dopo aver riempito i campi della form; i parametri vengono trasmessi al server Web in modo diverso a seconda che il metodo scelto sia get o post.
2. Il server richiama il gateway, trasmettendogli i parametri. La trasmissione avviene:
 - sullo standard input se il metodo è post,
 - in una variabile d'ambiente, se il metodo è get.
3. Il gateway esegue il proprio codice, con i parametri ricevuti, interagendo con altre risorse, per esempio una base di dati.

Common Gateway Information

4. Il gateway restituisce i risultati al server Web, preceduti da informazioni di servizio.
5. Il server Web trasmette i risultati al browser.



Accesso ai dati tramite CGI

- La tecnica di comunicazione CGI può essere utilizzata da programmi che svolgono funzioni di ogni genere e quindi in particolare da programmi che accedono a basi di dati, per esempio programmi scritti in un linguaggio di programmazione tradizionale con SQL immerso.
- La form vista precedentemente potrebbe essere utilizzata per trasferire i dati ad un programma CGI che provveda ad effettuare l'inserimento dei dati sulle prenotazioni di esami in una base di dati.
- Attraverso questo meccanismo è possibile scrivere programmi che permettono l'accesso a basi di dati, utilizzando una interazione orientata a maschere (*form-oriented*) fra cliente e servente Web.

Accesso ai dati tramite CGI

L'uso di programmi CGI è relativamente semplice ma presenta una serie di limitazioni:

- Piuttosto oneroso mantenere la **corrispondenza fra campi** della maschera e campi del database.
- La natura **stateless** del protocollo HTTP rende non immediata la gestione di applicazioni che includano transazioni con operazioni multiple sulle basi di dati
- Il programma CGI viene lanciato e termina. Dal punto di vista dell'ambiente di elaborazione, si tratta di un **processo che viene creato, eseguito e terminato**. Pesante nel caso di sistemi con molte richieste e programmi CGI di dimensione significativa.

Accesso ai dati tramite CGI

- Il programma CGI, una volta lanciato, richiede una **nuova connessione** con il DBMS. Anche qui abbiamo un costo di inizializzazione (e terminazione), che è prevalentemente a carico del DBMS, che può risultare sovraccaricato.
- Non è facile realizzare un servizio di ricerca di informazioni sul sito, funzionalità ritenuta estremamente importante in molti contesti. Infatti, gli strumenti standard di solito utilizzati allo scopo operano solo su file memorizzati staticamente sul sito e non sui file **generati dinamicamente**.

Agevolare la generazione dinamica di pagine HTML

Per superare il primo inconveniente, quasi tutti i produttori di DBMS (nonché produttori indipendenti) hanno sviluppato e reso disponibili strumenti di vario tipo:

- **Compileri** (o precompileri) o altri strumenti di sviluppo di gateway in linguaggi ad alto livello o ad/hoc con SQL immerso;
- **Gateway parametrici** che, con un opportuno linguaggio di macro possono descrivere la corrispondenza fra variabili (e quindi campi di form) e istruzioni SQL; i gateway possono essere *proprietary* (cioè legati ad uno specifico DBMS) oppure *indipendenti* (realizzati con interfaccia ODBC) e quindi in grado di interagire con diversi DBMS.

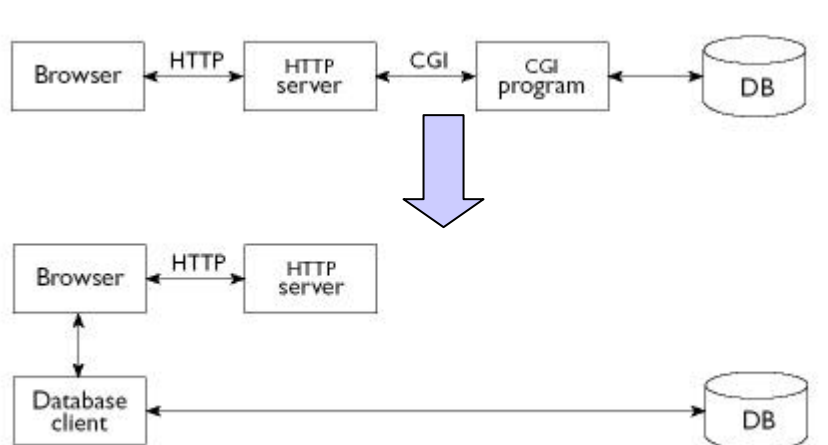
Agevolare la generazione dinamica di pagine HTML

Simulazione di connessioni di lunga durata

La soluzione al secondo problema:

1. Richiedere all'utente di specificare un identificatore (o comunque di costruirlo in qualche modo sulla base della prima richiesta) e di riutilizzare lo stesso dato, opportunamente "nascosto" nel codice HTML nelle schermate successive (*meccanismo di base*).
2. Usare **cookie**, ovvero insiemi di informazioni di dimensione relativamente piccola, che il server può mandare al client in un opportuno campo dell'intestazione di una risposta HTTP. Il contenuto del cookie viene memorizzato come parte della configurazione del browser e viene rinviato al server che lo ha generato ogniqualvolta il browser si connette a tale server.

Soluzioni basate sul client



Soluzioni basate sul client

- **Estensioni del browser:** si tratta di moduli software lanciati dal browser come librerie dinamiche; di solito vengono utilizzati per gestire dati di tipo particolare, ad esempio un formato grafico. Qui possono essere moduli (scritti in Java, Active-X, o altri linguaggi) per accedere a una base di dati remota. Possono risiedere sul client oppure possono essere incorporati nel codice HTML.
- **Helper esterni:** in alternativa alle estensioni, i browser possono lanciare o cedere il controllo a strumenti esterni (appunto helper o viewer). Si può realizzare un'interazione tradizionale client-server, semplicemente lanciata dal browser, per utilizzare in remoto un interprete SQL.

Soluzioni basate sul client

- **Browser proprietari:** alcuni produttori di DBMS hanno valutato la possibilità di realizzare browser specializzati proprio nell'accesso alle basi di dati. L'iniziativa non ha avuto particolare successo, perché gli utenti sono spesso interessati a utilizzare browser generali.

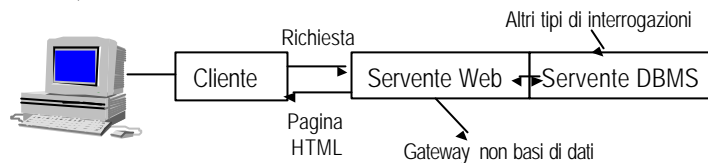
In generale, le soluzioni basate sul client sono adatte quasi esclusivamente ad un contesto noto e predefinito (tipicamente una intranet), a causa della complessità della distribuzione e dell'aggiornamento degli strumenti e della sempre parziale compatibilità dei programmi prelevati via rete con i vari browser.

Soluzioni basate sul server

Specifici server Web che integrano l'accesso alle basi di dati, eliminando la necessità dell'interfaccia CGI e utilizzando la base di dati stessa per memorizzare, ove opportuno, le pagine Web.

In questo modo si può rendere più semplice ed efficiente la programmazione delle operazioni gateway.

Questo approccio, detto del *Servente Universale (Universal Server)* è stato adottato da Oracle e Informix.



Basi di dati e WWW: Differenze e ...

Obiettivo

Web
Diffusione di informazioni

DB
Organizzazione e gestione delle informazioni di interesse per un sistema informativo

Tipo di informazioni

Web
Testuali e multimediali

DB
Dati con struttura relativamente semplice, come insiemi di tuple di valori atomici

Modello dei dati

Web
Orientato alla pagina
Schema di pagina
Attributi semplici e complessi

DB
Relazionale
Schema di relazione
Attributi semplici e talvolta relazioni nidificate

... Convergenze

- La tecnologia delle basi di dati sta cercando, attraverso *i sistemi ad oggetti*, di rispondere ad esigenze di applicazioni che richiedono la gestione di dati complessi (e multimediali).
- I browser Web costituiscono una sorta di *interfaccia generale* per l'accesso a funzioni di vario tipo su sistemi locali o remoti. Ad esempio, si potrebbe pensare di rendere disponibile su Web un catalogo di prodotti, memorizzato su una base di dati, e di poter effettuare transazioni commerciali (*commercio elettronico*).

... Convergenze

- Il Web ha una struttura spesso disordinata, ma è possibile che sue porzioni siano ordinate ed *organizzate in modo regolare*. Ad esempio, i vari dipartimenti di una università, con i relativi corsi e professori, hanno spesso pagine Web simili, collegate fra loro in modo omogeneo. Così come le basi di dati hanno una organizzazione che cerca di semplificare l'accesso riunendo i dati omogenei, può avere senso *progettare porzioni di Web con struttura controllata, con metodologie simili a quelle utilizzate per progettare le basi di dati*. In questo caso si può anche pensare di sfruttare questa struttura per *interrogare il Web con linguaggi analoghi a quelli utilizzati per interrogare la base di dati* (si veda il progetto MURST ARANEUS, <http://www.dia.uniroma3.it/Araneus/>).