

La valutazione delle prestazioni di un sito

- Utile per capire perché un sito bellissimo e con contenuti interessantissimi non viene visto

Spiega questi fatti

- “Twenty-eight percent of shoppers who have suffered failed performance attempts said **they stopped shopping at the web site** where they had problems, and six percent said **they stopped buying at that particular company’s off-line store.**”
(Boston Consulting Group, quoted in Infoworld / Computerworld 3/00)
- “It takes only 8 ½ seconds for half of the subjects to [give up]”
(Peter Bickford, “Worth the Wait?” in Netscape/View Source Magazine 10/97)
- “Perhaps as much as \$4.35 billion in e-commerce sales in the U.S. may be lost each year due to **unacceptable download speeds** and resulting user bailout behaviors.”
(Zona Research 4/99)
- “Fifty-eight percent of online customers surveyed indicated **quick download time as a key factor** in determining whether they would return to a web site.”
(Forrester Research 1/99)
- “One of the top three reasons cited by online shoppers for dissatisfaction with a web site is **slow site performance.**”
(Jupiter Communications / NFO Worldwide 1/99)
- “At one site, the **abandonment rate fell from 30% to 6-8%** because of a one second improvement in load time.”
(Zona Research 4/99)

Cosa sono le prestazioni?

- User Experience
 - Quanto è veloce il sito
 - Per quanto tempo è disponibile?
- Web Server
 - Quanti utenti serviti in parallelo
- Network
 - Le prestazioni contribuiscono ad entrambe le prestazioni precedent

Network Performance

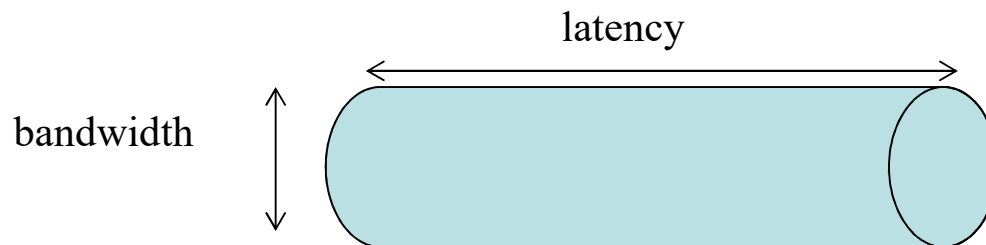
Le prestazioni di una rete possono essere misurate in

Latenza (Latency)

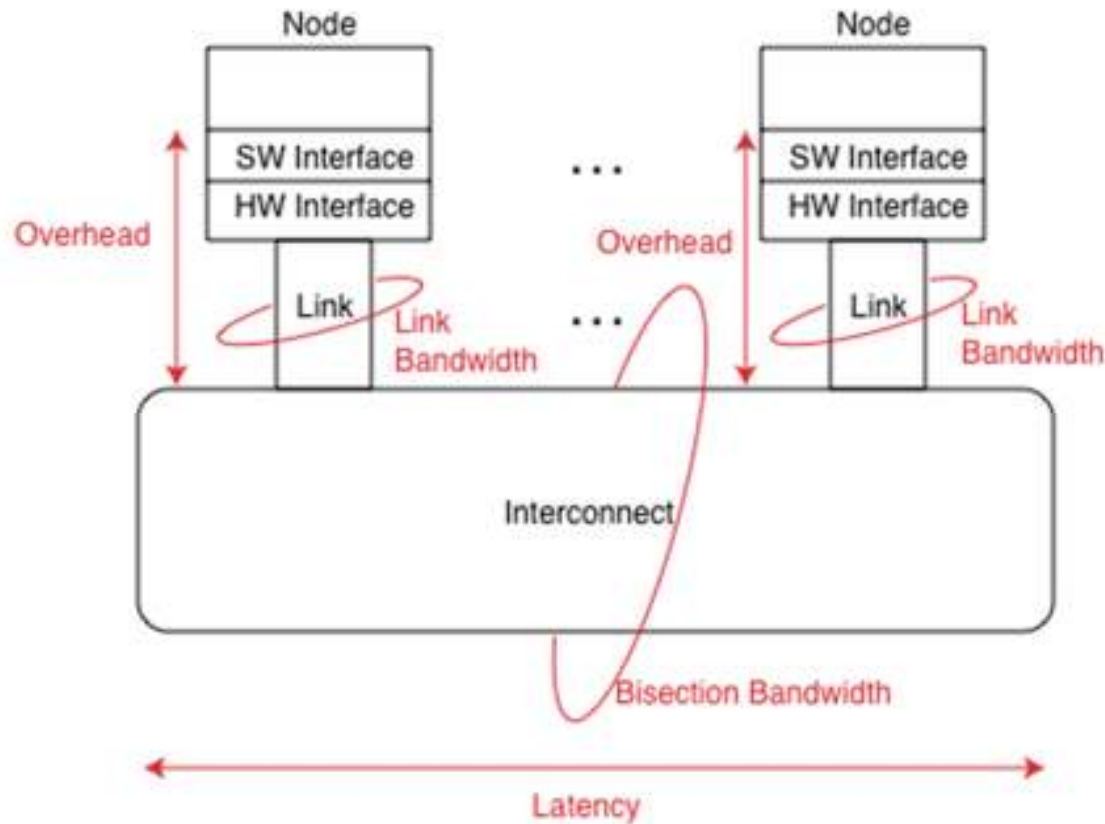
Quanto tempo impiega un messaggio tra i due punti di interesse (ad esempio tra browser dell'utente e il sito)

Banda (Bandwidth)

Quanti bit possono essere trasmessi su un collegamento in una unità di tempo

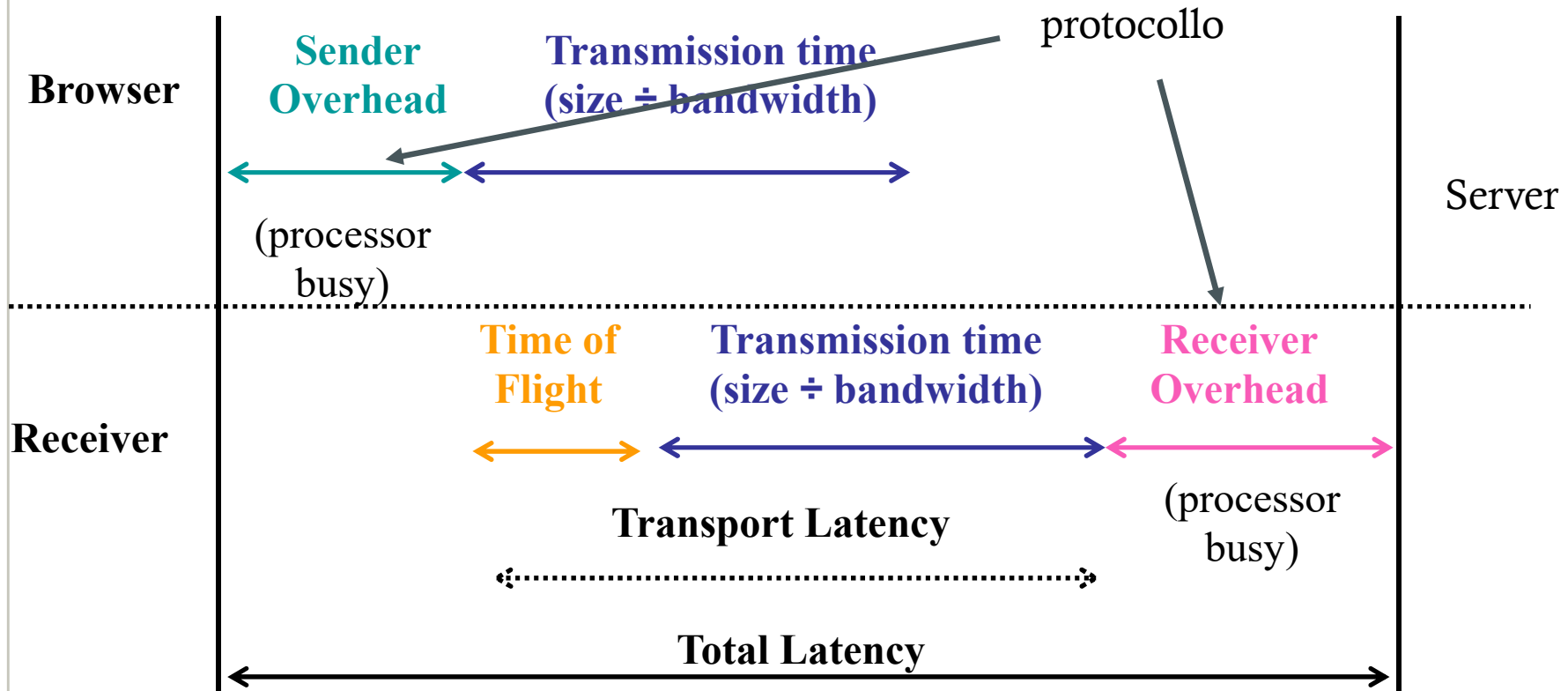


Network Performance Measures



- **Overhead:** latency of interface vs. **Latency:** network
- Non potete intervenire sui parametric fisici quali latenza e ampiezza di banda

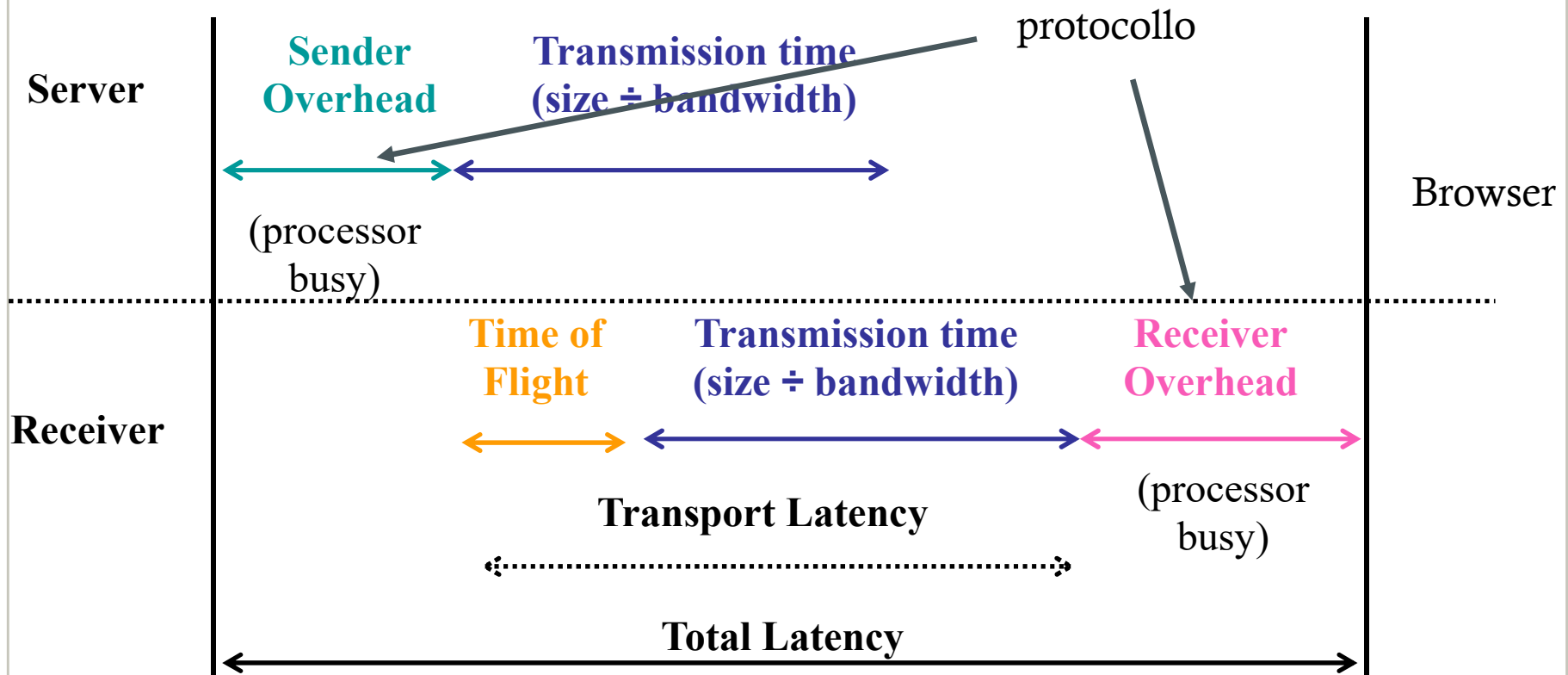
Universal Performance Metrics



$$\text{Total Latency} = \text{Sender Overhead} + \text{Time of Flight} + \text{Message Size} \div \text{BW} + \text{Receiver Overhead}$$

In questo caso
Non potete far nulla
Su parametri server

Ma anche



$$\text{Total Latency} = \text{Sender Overhead} + \text{Time of Flight} + \text{Message Size} \div \text{BW} + \text{Receiver Overhead}$$

In questo caso
il vostro progetto
controlla size

Latenza totale

- 1000 Mbit/sec., overhead invio 80 μ sec & overhead ricezione di 100 μ sec.
- messaggio di 10000 byte (compreso header), posso trasmettere 10000 bytes in un solo messaggio
- 3 casi : distanza 1000 km v. 0.5 km v. 0.01
- Speed of light $\sim 300,000$ km/sec (1/2 nel mezzo media)
- Latency_{0.01km} =
- Latency_{0.5km} =
- Latency_{1000km} =

Latenza totale

- $\text{Latency}_{0.01\text{km}} = 80 + 0.01\text{km} / (50\% \times .3)$
 $+ 10000 \times 8 / 1000 \text{ Mbit/s} + 100 \sim 260 \mu\text{sec}$
- $\text{Latency}_{0.5\text{km}} = 80 + 0.5\text{km} / (50\% \times .3)$
 $+ 10000 \times 8 / 1000 \text{ Mbit/s} + 100 \sim 263 \mu\text{sec}$
- $\text{Latency}_{1000\text{km}} = 80 + 1000 \text{ km} / (50\% \times .3)$
 $+ 10000 \times 8 / 1000 \text{ Mbit/s} + 100 \sim 6926 \mu\text{sec}$
- Minime differenze nonostante distanze molto diverse a causa di un pesante impatto del protocollo

Prestazioni webpagetest.org.

- Per utilizzarlo dovete poter caricare il vostro sito su un nodo con un IP pubblico, ovvero avere un vostro webserver su cui ospitare il sito
- Non sempre questo è possibile in un progetto didattico
- Dovete utilizzarlo nel caso del progetto di un vero sito

Prestazioni webpagetest.org.

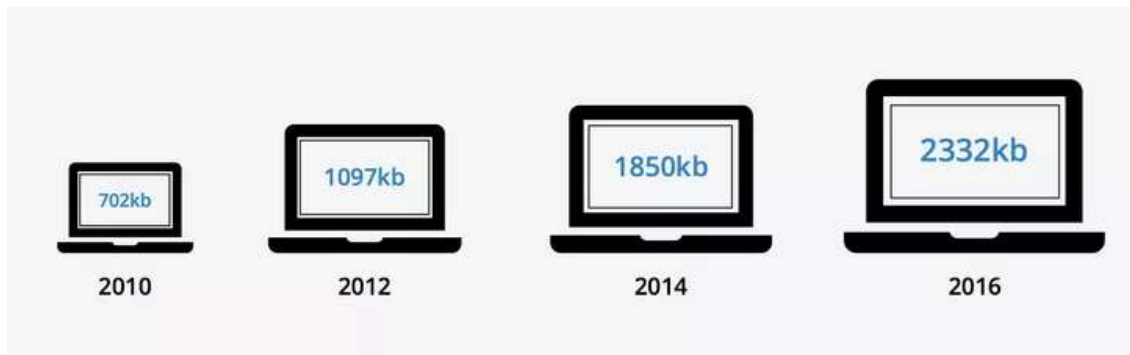
Fornisce alcune misure di tempo

- **Time To Title** = The time between the instant a visitor requests your website and the moment your site's title shows up in their browser tab
- **Time To Start Render** = The time between a user's request and the moment when content appears in their browser
- **Time To Interact** = The time between a request and when a user can click on links, type in fields or scroll the page
- **DNS Lookup Time** = The time of your DNS provider to translate a domain name into an IP address
- **Connection Time** = The time between a request and when a connection is established between browser and your server
- **Time to First Byte** = The time it takes for the very first byte to reach a user's browser after a connection to the server has been established
- **Time to Last Byte** = When the user's browser finally receives each and every byte of your website

Prestazioni webpagetest.org.

Fornisce alcune misure di dimensione

- **Overall Weight** = The total number of bytes the user receives is the overall weight of your website.





Valori medi

- **Overall Asset Count** = It differentiates between asset count and weight. It includes your total JavaScript, CSS and image counts.

Prestazioni [webpagetest.org](https://www.webpagetest.org).

[Login](#) | [Register](#) | [Login with Google](#)



Encryption secures digital transformation

Read the full 2019 Data Threat Report [Ad](#)
[Learn more](#)

HOME **TEST RESULT** TEST HISTORY FORUMS DOCUMENTATION ABOUT

Web Page Performance Test for
pages.di.unipi.it/tonelli/securityteam/publications.php
From: Dulles, VA - Chrome - Cable
6/5/2019, 12:28:00

Need help improving?

A

First Byte Time

A

Keep-alive Enabled

A

Compress Transfer

A

Compress Images

F

Cache static content

X

Effective use of CDN

Summary **Details** Performance Review Content Breakdown Domains Processing Breakdown Screenshot Image Analysis [Request Map](#)

Tester: VM02-08-172.16.20.225
First View only
Test runs: 3

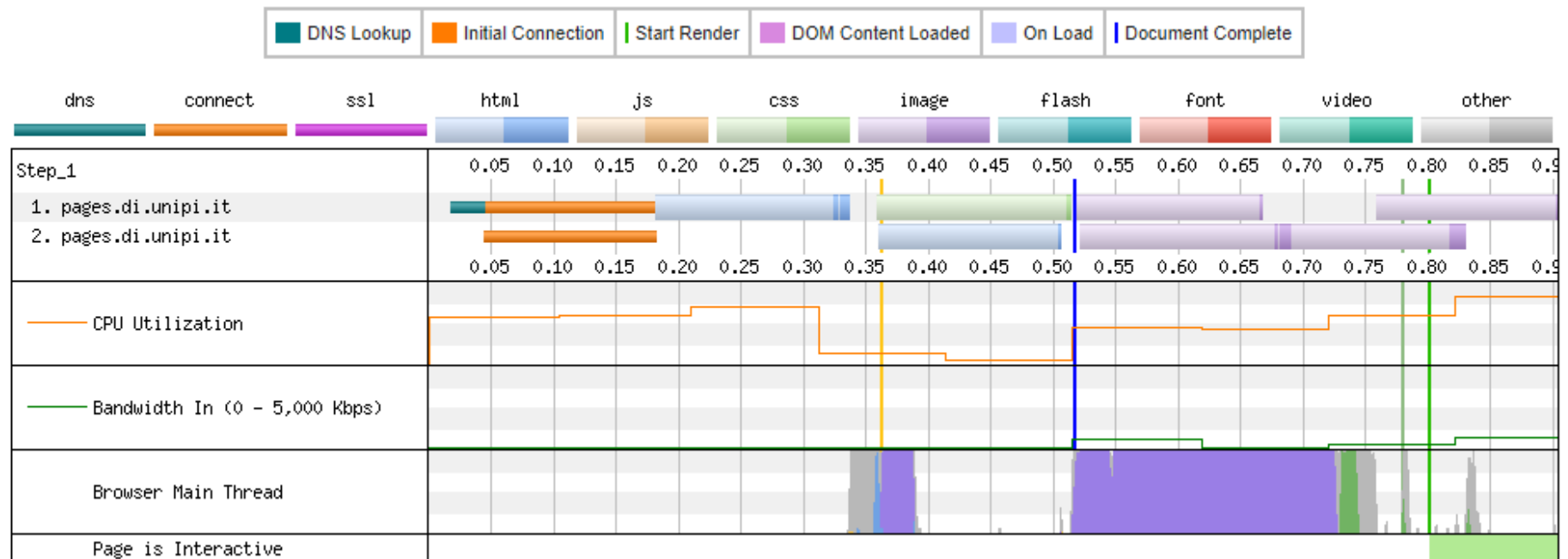
[Export HTTP Archive \(.har\)](#)
[Custom Metrics](#)

	Load Time	First Byte	Start Render	Visually Complete	Speed Index	Last Painted Hero	First Interactive (beta)	Result (error code)	Document Complete			Fully Loaded		
									Time	Requests	Bytes In	Time	Requests	Bytes In
First View (Run 1)	0.516s	0.327s	0.800s	0.800s	0.800s	0.800s	> 0.779s	99999	0.516s	3	9 KB	0.902s	6	25 KB

First Interactive (beta)	Images	Colordepth	RUM First Paint	domInteractive	domContentLoaded	loadEvent
> 0.779s	0	24	0.779s	0.362s	0.362s - 0.362s (0.000s)	0.516s - 0.516s (0.000s)

Prestazioni webpagetest.org.

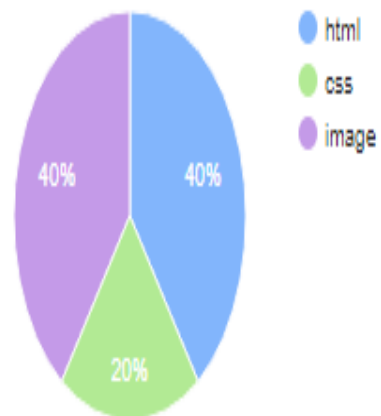
Connection View



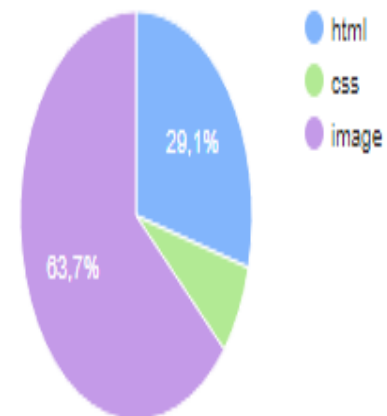
Prestazioni webpagetest.org.

Content Breakdown

Requests



Bytes



Httpperf basics

- Three distinguishing characteristics
 - robustness, which includes the ability to generate and sustain server overload,
 - support for the HTTP/1.1 protocol
 - its extensibility to new workload generators
- performance measurements

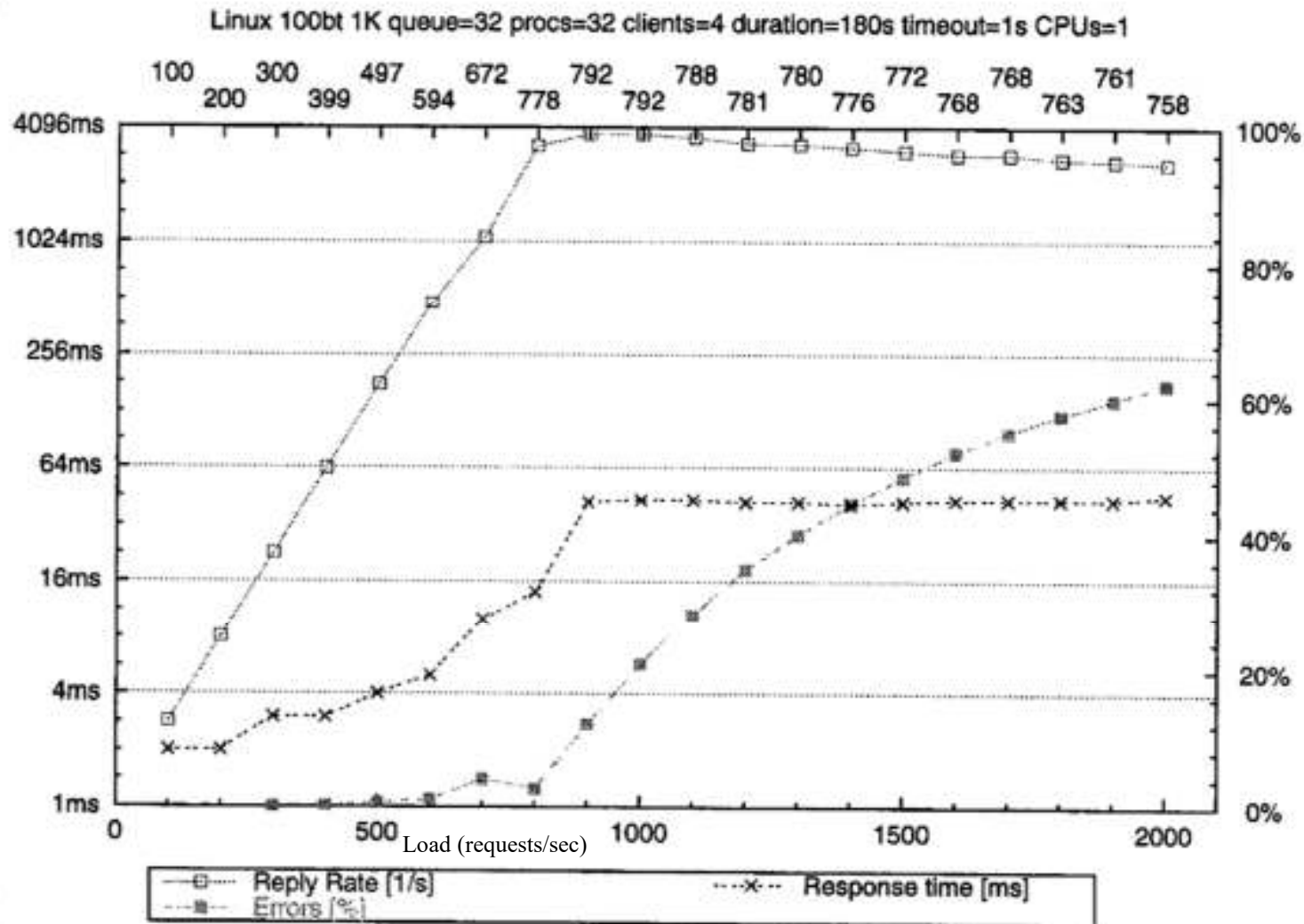
A Simple Example

- `httperf`
 - `--server hostname`
Specify the sever
 - `--port 80`
Specify the port
 - `--uri /test.html`
The file you want to download
 - `--rate 150`
The rate in requests/second
 - `--num-conn 27000`
The total number of TCP Connections
 - `--num-call 1`
The number of requests for each connection
 - `--timeout 5`
The request will fail if it takes longer than this

The output

- `httpperf --server apu --port 5556 --uri /test.html --rate 400 --num-conn 8000 --timeout 20`
- Maximum connect burst length: 1
- Total: connections 8000 requests 8000 replies 8000 test-duration 20.914 s
- Connection rate: 382.5 conn/s (2.6 ms/conn, <=263 concurrent connections)
- Connection time [ms]: min 0.8 avg 125.9 max 3000.5 median 0.5 stddev 594.7
- Connection time [ms]: connect 122.8
- Connection length [replies/conn]: 1.000
- Request rate: 382.5 req/s (2.6 ms/req)
- Request size [B]: 65.0
- Reply rate [replies/s]: min 351.2 avg 395.9 max 432.2 stddev 33.4 (4 samples)
- Reply time [ms]: response 3.0 transfer 0.0
- Reply size [B]: header 64.0 content 49.0 footer 0.0 (total 113.0)
- Reply status: 1xx=0 2xx=8000 3xx=0 4xx=0 5xx=0
- CPU time [s]: user 4.18 system 16.73 (user 20.0% system 80.0% total 100.0%)
- Net I/O: 66.5 KB/s ($0.5 \cdot 10^6$ bps)
- Errors: total 0 client-timo 0 socket-timo 0 connrefused 0 connreset 0
- Errors: fd-unavail 0 addrunavail 0 ftab-full 0 other 0

Sample Graph



Errors

- Errors will occur when the client connection experiences a timeout
- You can reduce the timeout value and increase the file size and rate to see the results:
- `httpperf --server apu --port 5556 --verbose --uri /testmid.html --rate 800 --num-conn 8000 --timeout 2`
- Connection rate: 799.9 conn/s (1.3 ms/conn, <=660 concurrent connections)
- Reply rate [replies/s]: min 671.8 avg 735.8 max 799.8 stddev 90.4 (2 samples)
- Errors: total 641 client-timo 641 socket-timo 0 connrefused 0 connreset 0

Another Example

- **htperf --hog --server apu --port 5556**

This command causes htparf to

- create a connection to host `apu.cs.byu.edu`,
- send a request for the root document (<http://apu:5556/>)
- receive the reply
- close the connection,
- and then print some performance statistics.
- The `--hog` parameter lets htparf use ports outside the normal limits (>5000)

Another Example

- `httpperf --hog --server apu --port 5556 --num-conn 100 --rate 10 --timeout 5`
 - Creates 100 connections
 - connections are created at a fixed rate of 10 per second
 - Connections timeout in 5 seconds

Another Example

- `httpperf --hog --ser=www --wsess=10,5,2 --rate 1 --timeout 5`
- Causes httpperf to generate a total of 10 sessions
- at a rate of 1 session per second.
- Each session consists of 5 calls that are spaced out by 2 seconds.

Using files

- `httpperf --server apu --port 5556 --uri /Pareto --hog --num-conn 100000 --rate 1000 --timeout 2 --verbose --wset 999,1 --period=e2`
- The `--wset` directive indicates that you will access files in the `/Pareto` directory in a round robin fashion.
- The URIs generated are of the form *prefix/path.html*, where *prefix* is the URI prefix specified by option `--wset` and *path* is generated as follows: for the *i*-th file in the working set, write down *i* in decimal, prefixing the number with as many zeroes as necessary to get a string that has as many digits as *N*-1. Then insert a slash character between each digit. For example, the 103rd file in a working set consisting of 1024 files would result in a path of "`0/1/0/3`". Thus, if the URI-prefix is `/wset1024`, then the URI being accessed would be `/wset1024/0/1/0/3.html`. In other words, the files on the server need to be organized as a 10ary tree.

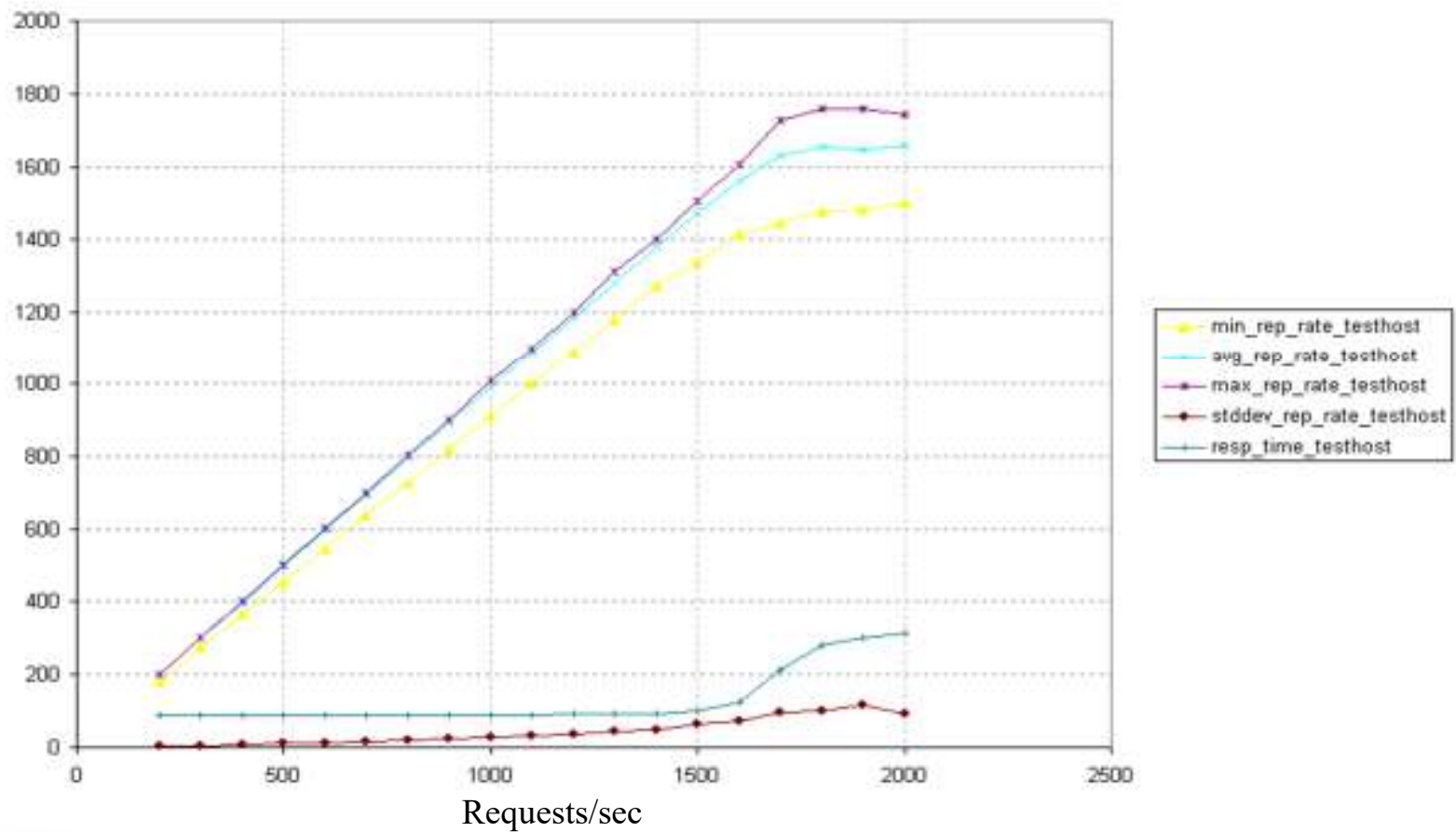
Autobench

- Perl script
- Wrapper for httpperf to make things easier.
- Extracts the data from httpperf output
- Simple mode – benchmark single server

```
autobench --single_host --host1 www.test.com --uri1 /10K --quiet \
  --low_rate 20 --high_rate 200 --rate_step 20 --num_call 10 \
  --num_conn 5000 --timeout 5 --file results.tsv
```

Autobench

Example Benchmark

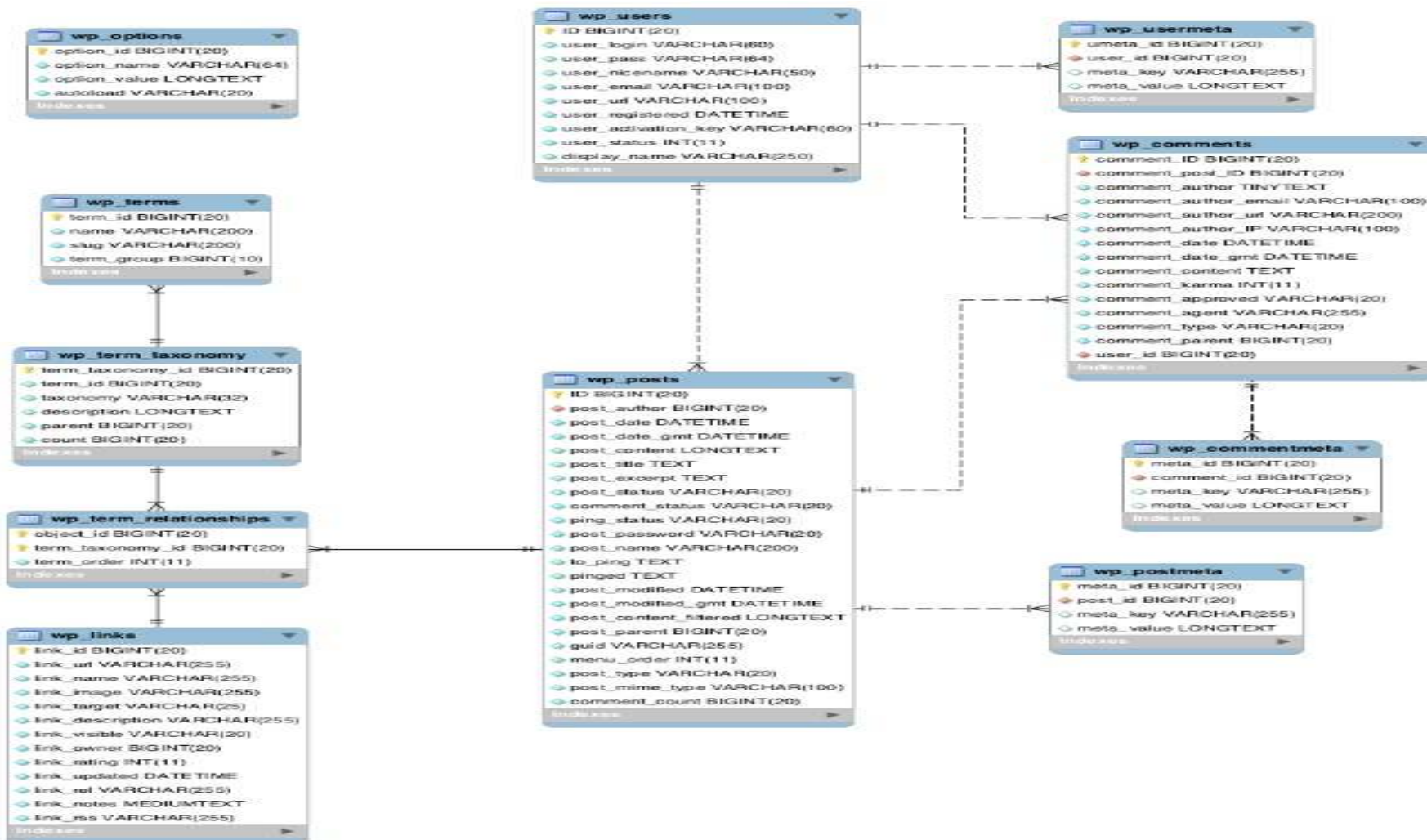


Come fare il progetto -1

1. Decidere il contenuto di un post (testo, file e immagini oppure testo ma non immagini, immagini e filmati etc.)
2. Decidere come ci si autentica (password, quanti caratteri, nome oppure email oppure)
3. Quanti tipi di utenti
4. Quali entità (utenti, blog, post, immagini, testi ...) e quali relazioni = qui può essere utile il modello delle persone
5. Quanti e quali tabelle SQL (forme normali etc)
 1. Blog
 2. Post
 3. Utenti
 4.

A questo punto dovete venire a ricevimento studenti per un primo controllo ...

Cosa portare al ricevimento...



Come fare il progetto -2

5. Numero di pagine
6. Scrittura del codice PHP/html/form
7. Debugging del codice ed eventuali correzioni
8. Valutazione prestazioni
 - Quanto spazio per ogni post
 - Quanto spazio per ogni blog
 - Tempi di operazioni

A questo punto dovete venire a ricevimento studenti per un secondo controllo ...

Cosa portare al ricevimento...

- Codice Php
- Codice Html
- Form
- Esempi di esecuzione
- Info su prestazioni

Come fare il progetto - 3

9. Ottimizzazioni mediante Ajax/jquery

10. Validazione input

11. Miglioramento grafica e presentazione

12. Valutazione prestazioni

- Quanto spazio per ogni post
- Quanto spazio per ogni blog
- Tempi di operazioni

A questo punto dovete solo scrivere la relazione e sostenere l'esame



SQL Injection - I

- È un attacco che sfrutta il passaggio di dati pericolosi nell'input dell'utente

SQL Injection.

User-Id:

Password:

`select * from Users where user_id= 'srinivas'
and password = 'mypassword'`

User-Id:

Password:

`select * from Users where user_id= ' ' OR 1 = 1; /*'
and password = ' */-- '`

SQL Injection - I

- È uno degli attacchi più pericolosi e popolari che sfrutta il passaggio di dati pericolosi nell'input dell'utente
- Consideriamo il caso in cui
 - un utente trasmetta una stringa in input \$name
 - la stringa viene usata nella query

```
$query = "SELECT * FROM customers  
WHERE username = '$name' ";
```

SQL Injection - II

- Utente corretto \$name = "tom";
\$query = "SELECT * FROM customers
WHERE username = "tom"";
- SQL Injection = \$name= " tom " OR 1;
\$query = "SELECT * FROM customers
WHERE username = "tom" OR 1";
sono trasmessi i nomi di tutti gli utenti e non solo di uno

SQL Injection - III

- \$name= ''; DELETE FROM customers WHERE 1 or username = '';

```
$query= SELECT * FROM customers WHERE  
        username = ' ';  
        DELETE FROM customers WHERE 1 or  
        username = ' '
```

Effetto complessivo è di distruggere il database

SQL Injection

- Notare che questo attacco come tutti i tipi di injection inserisce dei comandi dove il sistema si attende un dato
- Ambiguità tra dati e comandi è alla base di molti problemi di sicurezza
- Non è mai completamente risolubile perchè questa ambiguità è alla base di una macchina universale ovvero un dispositivo di calcolo che permetta di risolvere ogni problema

SQL Injection – Prevenzione 1

- Questa soluzione trasforma la stringa dell'utente sostituendo ' e " con le sequenze di escape: \' e \"
- Php che MySQL interpretano questi caratteri come parte della stringa e non come delimitatori, annullando l'attacco.
- Per eseguire la sostituzione si usa la funzione predefinita

`mysql_real_escape_string`

che sostituisce apice, doppio apice, backslash e la stringa nulla con le rispettive sequenze di escape.

```
$user = mysql_real_escape_string($_POST["utente"]);  
$pwd = mysql_real_escape_string($_POST["password"]);
```

Prevenzione di injection

- `mysql_real_escape_string` evita iniezione impedendo che caratteri siano considerati come controllo
- `$name= " OR 1";`
`$name= mysql_real_escape_string($name);`
nella query si cerca un utente che si chiama OR 1
- `$name= "; DELETE FROM customers WHERE 1 or username = ";`
nella query si cerca un utente che si chiama
;DELETE

SQL Injection – Prevenzione 2

- Questa soluzione valida e rilevazione
- Utilizza alcune funzioni stringa predefinite
 - **strlen** restituisce la lunghezza di una stringa;
 - **substr** estrae una sottostringa : substr(stringa, start, len)
restituisce len caratteri a partire dalla posizione start.
 - **strpos** trova la posizione della prima occorrenza di una stringa all'interno di un'altra stringa e restituisce false in caso di esito negativo della ricerca
- Queste funzioni controllano ciascun carattere fornito dall'utente per vedere se corrisponde a un carattere alfabetico (maiuscolo o minuscolo) oppure a una cifra numerica

SQL Injection – Prevenzione 2

```
$user = $_POST["utente"];
$pwd = $_POST["password"];
$ammessi = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ";
$ammessi .= "0123456789";
$userok = true;
for( $pos=0; $pos<strlen($user) && $userok; $pos++ )
    {
        $scar = substr($user, $pos, 1);
        if ( strpos($ammessi, $scar) === false ) $userok = false; }
$pwdok = true;
for( $pos=0; $pos<strlen($pwd) && $pwdok; $pos++ )
    {
        $scar = substr($pwd, $pos, 1); if ( strpos($ammessi, $scar) === false )
        $pwdok = false; }
if ($userok and $pwdok )
    { accesso ai dati }
    else
        echo "Inserimento dati non corretto";
```


SQL Injection – Prevenzione 3

- Questa soluzione controlla che le stringhe fornite dall'utente non contengano comandi SQL oppure il carattere apice, oppure i caratteri – – che in molte versioni del linguaggio SQL indicano l'inizio di frasi di commento; l'inserimento di un commento annulla la rimanente parte del comando SQL previsto dal programmatore nell'applicazione.
- Le parole da controllare sono contenute in un array di 8 elementi. Anche questo codice utilizza strpos per la ricerca di una sottostringa all'interno di una stringa.

SQL Injection – Prevenzione 3

```
$user = $_POST["utente"];  
$pwd = $_POST["password"];  
$vietate = array ("select", "insert", "update", "delete", "drop", "alter", "—", "");  
$userok = true;  
for( $k=0; $k <= 7 && $userok; $k++ )  
if ( strpos($vietate[$k], $user) !== false ) $userok = false;  
$pwdok = true;  
for( $k=0; $k <= 7 && $pwdok; $k++ )  
if ( strpos($vietate[$k], $pwd) !== false ) $pwdok = false;  
// ——come nella soluzione 2
```

- operatore di confronto `!==`, (non identico): il confronto è vero se i due termini differiscono per valore o tipo.
 - `(0 !== false)` è vero,
 - `(0 != false)` è falso.

SQL injection – Prevenzione 4

Prepared statement

```
$query = $mysqli->prepare("SELECT * FROM login  
WHERE user = ? AND password = ?");
```

- La separazione fra la preparazione dell'SQL e i dati ci protegge dalle SQL injection, perché i dati provenienti dall'utente verranno gestiti al di fuori dell'istruzione SQL.
- A livello di sintassi non vi sono delimitatori intorno al parametro: in questo caso pur trattandosi di stringhe non è necessario delimitarle perché l'associazione con il tipo di dato verrà fatta in fase di binding.

SQL injection – Prevenzione 4

Prepared statement

```
$query->bind_param('ss',$user,$password);
```

Bind_param accetta due valori: il primo è il tipo di dato e può avere i seguenti valori

Valore	Descrizione
"s"	Corrisponde a variabili associate al tipo di dato stringa.
"i"	Corrisponde a variabili associate al tipo di dato numeri interi.
"d"	Corrisponde a variabili associate al tipo di dato numeri double.
"b"	Corrisponde a variabili associate al tipo di dato BLOB, formato binario.

SQL injection – Prevenzione 4

Prepared statement

```
$query->bind_param('ss',$user,$password);
```

Il valore della variabile è ripulito se non corrisponde al tipo di dato dichiarato

Se la variabile ha come bind il tipo “i”, numero intero ma contiene “test”, il valore inserito è 0, compatibile con quanto dichiarato

Il secondo valore è la variabile con il dato da utilizzare nella query.

I valori alle variabili possono essere assegnati anche dopo il bind_param, ma prima dell'esecuzione della query.

mysql non supporta binding mediante nomi ma solo il binding posizionale, quindi l'ordine dei parametri. Nel nostro caso la variabile \$user sarà associata al primo parametro della query.

SQL injection – Prevenzione 4

Prepared statement

```
$result = $query->execute();
```

Questo ci permette di eseguire la query ed assegnare \$result per sapere se l'operazione di autenticazione mediante la query ha avuto successo

SQL injection – Prevenzione 4

Prepared statement inserzioni multiple

```
$query = $mysqli->prepare("INSERT INTO login(user, password) values(?,?)");  
$query->bind_param("ss", $user, $password);  
  
....  
$user = $_POST['user'];  
$pass = $_POST['password'];  
$query->execute();  
  
.....  
$user = $_POST['user'];  
$pass = $_POST['password'];  
$query->execute();  
  
....  
$user = $_POST['user'];  
$pass = $_POST['password'];  
$query->execute();
```

Riassunto del corso

Database = **Insieme organizzato di dati** utilizzati per il supporto allo svolgimento delle attività di un ente (azienda, ufficio, persona) = Insieme di dati gestito da un **DBMS**

Nella nostra accezione i database sono:

- un supporto digitale in cui i dati sono immagazzinati
- un software che permetta il recupero dei dati
- una rete che permetta un'accessibilità condivisa ai dati

DEFINIZIONE.

Una base di dati è una raccolta di dati **permanenti** suddivisi in due categorie:

- I METADATI
- I DATI

CCS Ingegneria Informatica Orario - Netscape

File Edit View Go Communicator Help

UNIVERSITA' DEGLI STUDI DI CHISSADOVE

Corso di Studi in Ingegneria Informatica

**ORARIO DELLE LEZIONI PER L'ANNO
ACCADEMICO 1999-2000**

INSEGNAMENTO	Docente	Aula	Orario
Analisi matematica I	Luigi Neri	N1	8:00-9:30
Basi di dati	Piero Rossi	N2	9:45-11:15
Chimica	Nicola Mori	N1	9:45-11:30
Fisica I	Mario Bruni	N1	11:45-13:00
Fisica II	Mario Bruni	N3	9:45-11:15
Sistemi informativi	Piero Rossi	N3	8:00-9:30

Document: Done

I METADATI

i metadati, ovvero lo **schema della base di dati**, sono una raccolta di definizioni che descrivono

- la struttura di alcuni insiemi dati,
- le restrizioni sui valori ammissibili dei dati
- le relazioni esistenti fra gli insiemi.

Lo schema va definito prima di creare i dati ed è indipendente dalle applicazioni che usano la base di dati.

I DATI

i dati, le rappresentazioni dei fatti conformi alle definizioni dello schema, con le seguenti caratteristiche:

I dati sono **strutturati**, cioè hanno un formato predefinito

- a) sono organizzati in **insiemi omogenei**, fra i quali sono definite delle relazioni e sono previsti operatori per estrarre elementi da un insieme e per conoscere quelli che, in altri insiemi, sono in relazione con essi.
- b) **sono molti**, in assoluto e rispetto ai metadati, e non possono essere gestiti in memoria temporanea; sono memorizzati in una memoria permanente.
- c) **sono permanenti**, cioè, una volta creati, continuano ad esistere finché non sono esplicitamente rimossi;
- d) sono accessibili mediante **transazioni**;
- e) **sono protetti** sia da accesso da parte di utenti non autorizzati, sia da corruzione dovuta a malfunzionamenti hardware e software;
- f) sono utilizzabili contemporaneamente da utenti diversi.

TRANSAZIONI L'UTENTE

- Esempi:
 - versamento presso uno sportello bancario
 - emissione di certificato anagrafico
 - dichiarazione presso l'ufficio di stato civile
 - prenotazione aerea
- Due accezioni
 - Per l'utente:
 - programma a disposizione, da eseguire per realizzare una funzione di interesse
 - Per il sistema:
 - sequenza indivisibile di operazioni

ESEMPIO DATI e METADATI

Si consideri la base di dati degli studenti ed esami superati definiti dagli schemi di relazioni:

Studenti (Matricola, Cognome, Città, AnnoNascita)

Esami (Materia, Candidato, Voto, Data)

una base di dati contiene i dati immessi, meno ovvio è il fatto che in una base di dati ma anche informazioni sui dati definiti, i metadati.

Esempi di queste informazioni sono:

1. i nomi delle relazioni definite;
2. il tipo delle ennuple delle relazioni;
3. le chiavi primarie ed esterne definite;
4. i vincoli sui valori ammissibili degli attributi.

Queste informazioni sono memorizzate in tabelle predefinite che sono gestite automaticamente dal sistema.

DATI e METADATI

- Per rappresentare i corsi attivati ed i relativi docenti si può usare uno schema del tipo

(Corso, Nome Docente)

- questo e' uno schema invariante nel tempo,

- mentre le coppie

(Informatica Generale, Pippo)

(Basi di Dati e Sistemi Informativi, Baiardi)

- sono istanze dello schema precedente e possono variare nel tempo

DataBase Management System (DBMS)

Le caratteristiche delle basi di dati sono garantite da un **DBMS, Data Base Management System**, che controlla i dati e li rende accessibili agli utenti autorizzati.

Un DBMS è un sistema centralizzato o distribuito che offre opportuni linguaggi

- a) per definire lo schema della base di dati (lo schema va definito prima di creare dati)
- b) per scegliere le strutture dati per la memorizzazione dei dati,
- c) per usare la base di dati interattivamente o da programmi
- d) memorizzare i dati rispettando i vincoli definiti nello schema;

■ Un **DBMS**

- è un sistema software
- gestire collezioni di dati che siano *grandi, condivise e persistenti*, assicurando la loro *affidabilità e privacy*.
- deve essere *efficiente ed efficace*.

I DATABASE SONO...

- **grandi**
 - dimensioni (molto) maggiori della memoria centrale dei sistemi di calcolo utilizzati
 - il limite deve essere solo quello fisico dei dispositivi
- **persistenti**
 - hanno un tempo di vita indipendente dalle singole esecuzioni dei programmi che le utilizzano
- **condivisi**
 - ogni organizzazione è divisa in settori o comunque svolge diverse attività . Ciascun settore/attività ha un (sotto) sistema informativo (non necessariamente disgiunto)
 - Una base di dati e' una risorsa integrata, condivisa fra applicazioni
 - Conseguenze:
 - Attività diverse su dati condivisi: meccanismi di autorizzazione
 - Accessi di più utenti ai dati condivisi: controllo della concorrenza

DBMS GARANTISCONO...

- **PRIVATEZZA**
 - Si possono definire meccanismi di autorizzazione
 - l'utente A è autorizzato a leggere tutti i dati e a modificare quelli sul ricevimento
 - l'utente B è autorizzato a leggere X e a modificare Y
- **AFFIDABILITA'** (per le basi di dati):
 - resistenza a malfunzionamenti hardware e software
 - una base di dati è una risorsa pregiata e quindi deve essere conservata a lungo termine

I DBMS DEVONO ESSERE

...

- EFFICIENTI

- Cercano di utilizzare al meglio le risorse di spazio di memoria (principale e secondaria) e tempo (di esecuzione e di risposta)
- I DBMS, con tante funzioni, rischiano l'inefficienza e per questo ci sono grandi investimenti e competizione
- L'efficienza è anche il risultato della qualità delle applicazioni

- EFFICACI

- Cercano di rendere produttive le attività dei loro utilizzatori, offrendo funzionalità articolate, potenti e flessibili:
 - il corso è in buona parte dedicato ad illustrare come i DBMS perseguono l'efficacia

Costruzione di DB

MODELLI INFORMATICI

DIAGRAMMI E-R

MODELLI

L'informatica offre metodologie e strumenti per la costruzione di modelli di situazioni reali che ricorrono in ogni campo che richiede un'attività di progettazione. Essi permettono di riprodurre le caratteristiche essenziali di fenomeni reali, omettendo dettagli che costituirebbero inutili complicazioni.

Per gestire un sistema informativo con strumenti informatici dobbiamo realizzare una **descrizione della realtà d'interesse per mezzo di simboli**

Un **modello di dati e' un insieme di concetti utilizzati per organizzare i dati e descriverne la struttura in modo che essa risulti comprensibile a un elaboratore**

Ad es per poter gestire una biblioteca è necessario dare al calcolatore una sua descrizione simbolica

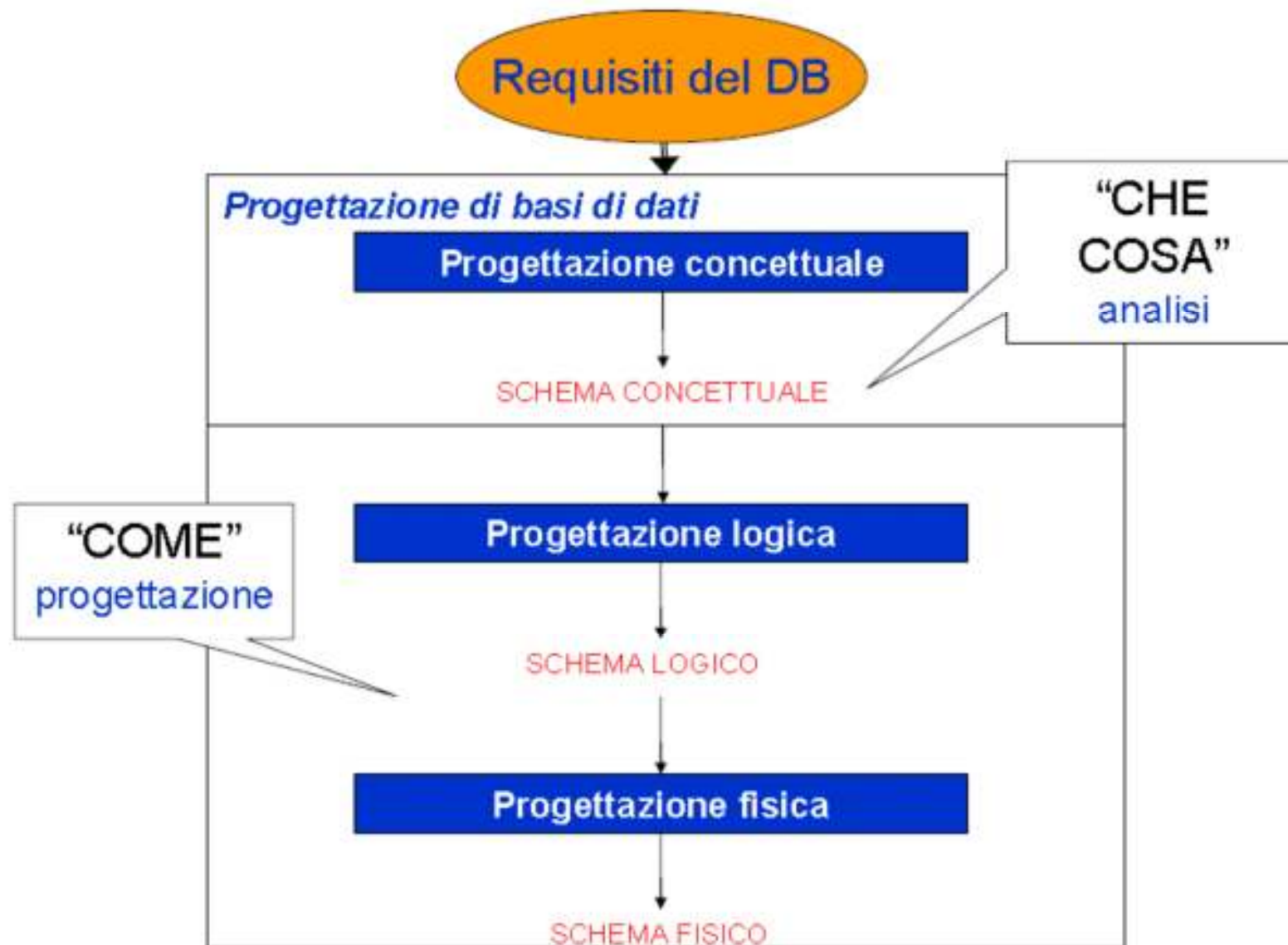
Ogni descrizione costituisce un modello della realtà

MODELLI INFORMATICI

Sono **simbolici**: rappresentazione formale delle idee e conoscenze relative ad un fenomeno

Esistono diverse categorie di **modelli informatici** che differiscono per i tipi di fatti che si considerano, ossia per i diversi livelli di astrazione a cui si opera

- **modelli concettuali**: considerano solo i fatti relativi alla realtà senza riferimento agli strumenti informatici che dovranno usare il modello
- **modelli logici**: sono considerati anche fatti relativi agli strumenti informatici; il modello è adeguato alle caratteristiche del DBMS, in modo che possa essere da questo interpretato
- **modelli fisici**: sono considerate le strutture fisiche usate dal calcolatore per rappresentare i dati



COSTRUZIONE DEL MODELLO

Nella costruzione di un modello informatico,

- ❑ prima si “**definisce**” il modello, descrivendo conoscenza concreta e conoscenza astratta
- ❑ poi si “**costruisce**” la rappresentazione della conoscenza concreta.

Per la definizione del modello si possono usare diversi tipi di formalismi, che si differenziano per il “*modello dei dati*” che supportano, cioè per meccanismi di astrazione offerti per rappresentare la realtà.

Nel corso abbiamo usato il **modello relazionale** come formalismo per la **realizzazione** di una base di dati.

ESEMPIO

Per costruire un modello informatico per la **gestione di informazioni sui libri**, prima si devono definire quelle che interessano ai fini dell'applicazione (titolo, autore, editore ecc.).

Una volta definite le proprietà interessanti comuni a tutti i possibili libri, si passa a costruire per ogni entità "libro" della realtà oggetto di studio una rappresentazione nel modello informatico.

L'argomento di un libro può essere descritto nel modello ma non caratterizza il modello ovvero lo stesso modello informatico può essere usato per descrivere libri che parlano di

- Cucina
- Caccia e pesca

DIAGRAMMI E-R

Nel **formalismo grafico** che si adotta, una **classe = insieme di oggetti che interessa descrivere** si rappresenta con un rettangolo etichettato con il nome della classe.

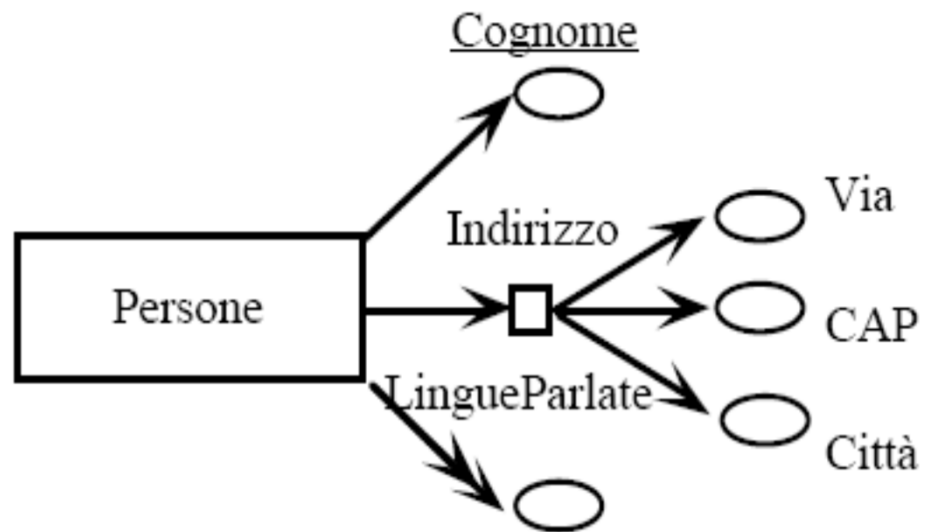
Per gli **attributi** si usano le seguenti convenzioni:

1. attributi con valori atomici sono rappresentati con ovali collegati alla classe da un arco che termina con una freccia singola, se l'attributo è univoco, oppure con una freccia doppia, se l'attributo è multivalore;
2. se l'attributo ha valori strutturati, l'ovale si sostituisce con un quadratino dal quale escono archi verso ovali in numero pari ai campi del valore strutturato.
3. se l'attributo ha valori unione, si procede come nel caso precedente sostituendo il quadratino con un cerchietto;
4. proprietà che possono avere valori non specificati si rappresentano con un taglio sull'arco;
5. gli attributi della chiave primaria si sottolineano.

Esempi

Una classe con gli attributi.

Cognome e' la chiave primaria



Una classe

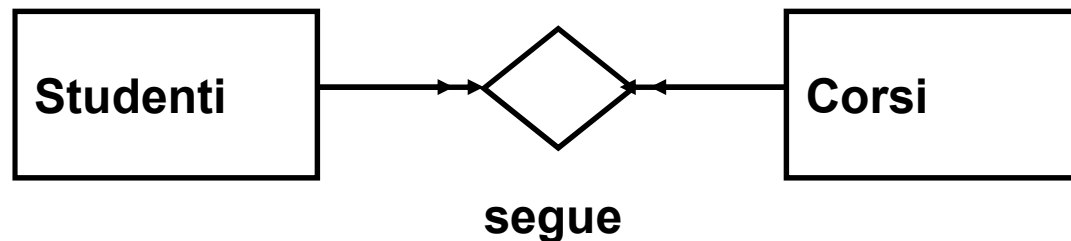
Studenti

Relazioni

Una relazione **binaria** tra classi si rappresenta con un rombo collegato con degli archi alle classi associate.

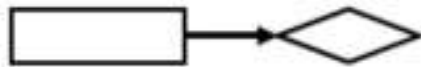
Il rombo è etichettato con il nome della relazione

La relazione non ha attributi

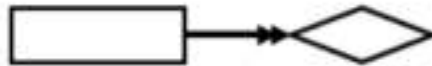


RELAZIONI: caratteristiche

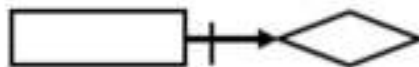
- **univocità**: ad un oggetto di una classe può essere associato un solo oggetto dell'altra classe
- **totale**: ad un oggetto di una classe è sempre associato qualche oggetto dell'altra classe



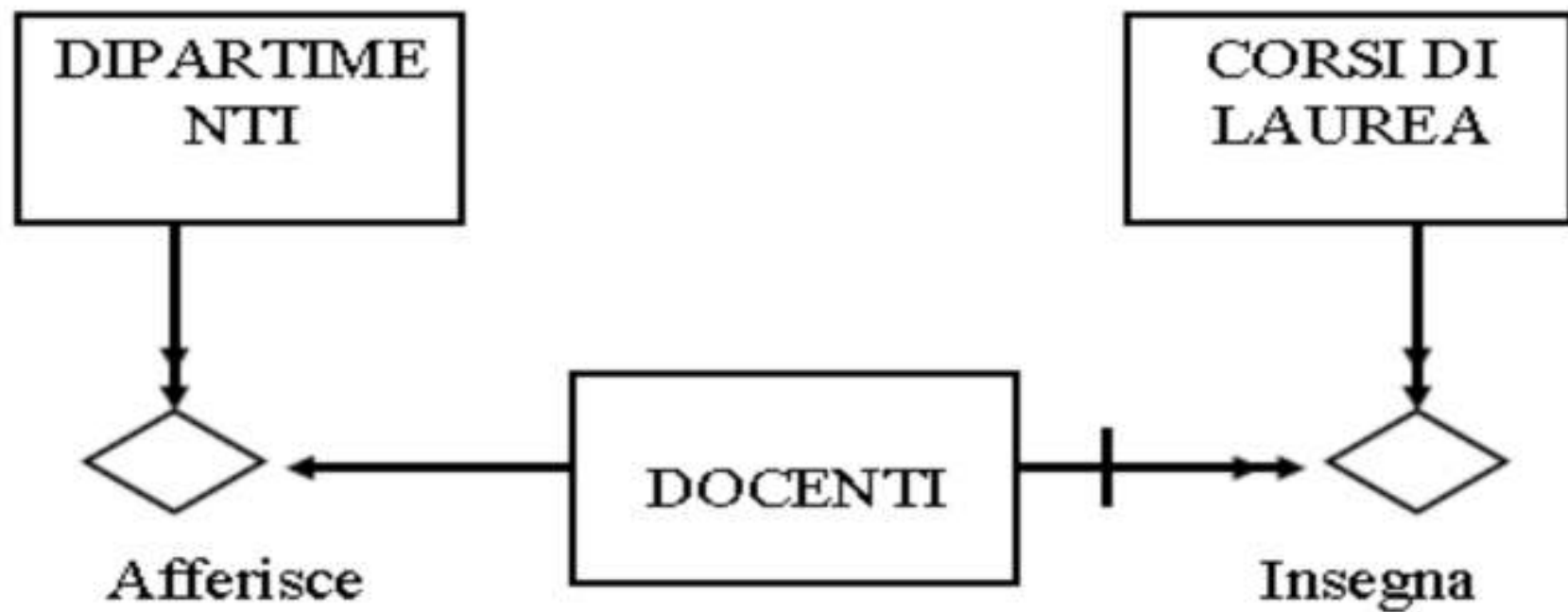
- **multivalore**: ad un oggetto di una classe possono essere associati più oggetti dell'altra classe



- **parziale**: ad un oggetto di una classe non è sempre associato qualche oggetto dell'altra classe



Relazioni: esempio

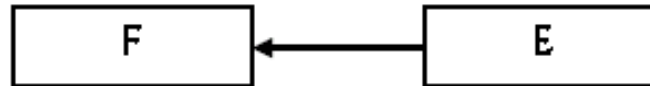


Relazioni: esempio

- **Afferisce** è **univoca** da Docenti a Dipartimenti: ad un oggetto della classe Docenti è associato un solo oggetto della classe Dipartimenti; si modella il fatto che un docente possa afferire ad un solo dipartimento
- **Afferisce** è **multivalore** da Dipartimenti a Docenti: ad un oggetto della classe Dipartimenti possono essere associati più oggetti della classe Docenti; si modella il fatto che ad un dipartimento possano afferire molti docenti
- **Insegna** è **totale** da Corsi di Laurea a Docenti: ad un oggetto della classe Corsi di Laurea è sempre associato qualche oggetto della classe Docenti; si modella il fatto che in un corso di laurea debbano necessariamente insegnare dei docenti
- **Insegna** è **parziale** da Docenti a Corsi di Laurea: ad un oggetto della classe Docenti non è sempre associato qualche oggetto della classe Corsi di Laurea ; si modella il fatto che un docente potrebbe non insegnare

GERARCHIA FRA ENTITA

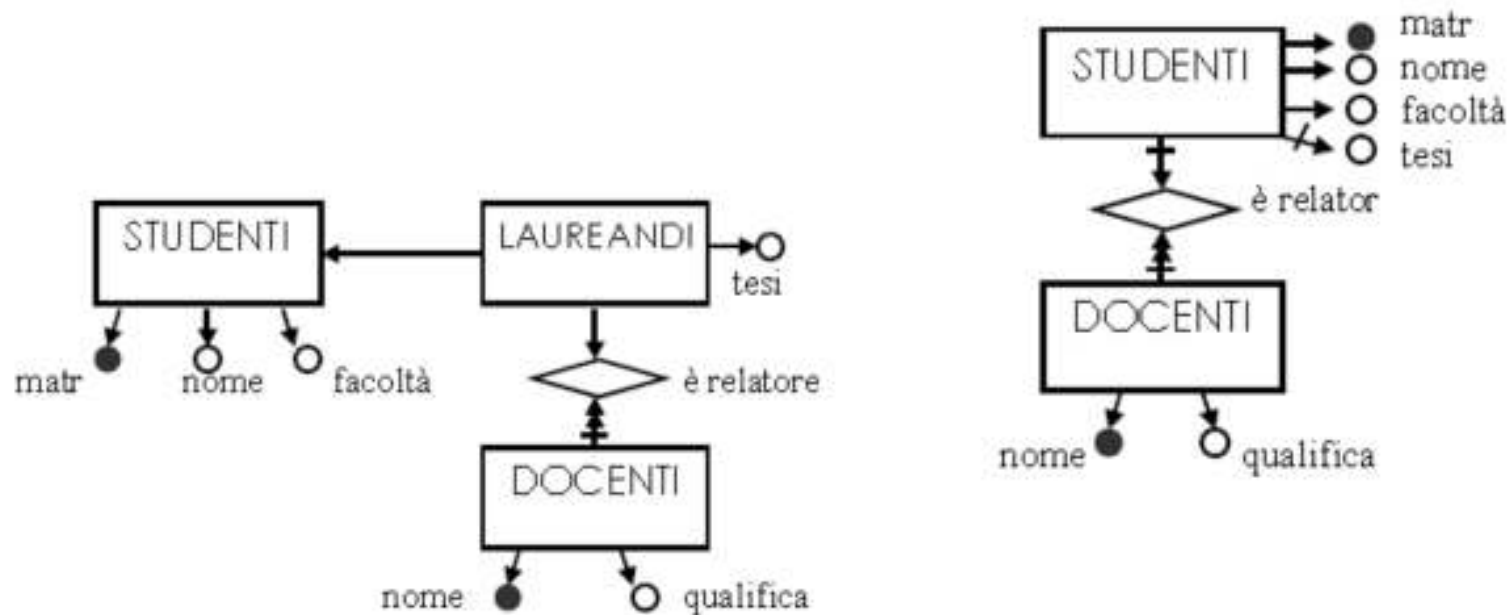
- Tra due entità E ed F può essere stabilita una **gerarchia**



- E è detta **sottoclasse** o specializzazione di F
- F è detta **superclasse** o generalizzazione di E
- Proprietà delle gerarchie
 - Vincolo di struttura**: se E è sottoclasse di F , E ha tutti gli attributi di F , e partecipa a tutte le associazioni cui partecipa F (ereditarietà). E può avere altri attributi, e partecipare ad altre associazioni
 - Vincolo di insieme**: se E è specializzazione di F , ogni oggetto di E è anche un oggetto di F (cioè E è un sottoinsieme di F)

GERARCHIA FRA ENTITA'

: esempio



Lo schema a sinistra e' piu' espressivo di quello di destra

DIAGRAMMI E-R: esempio

A titolo di esempio, mostriamo, ad un primo livello di dettaglio, la rappresentazione con il formalismo grafico di alcuni fatti riguardanti una **biblioteca universitaria**: descrizioni bibliografiche, libri, autori, utenti e prestiti.

Delle entità interessano le seguenti proprietà:

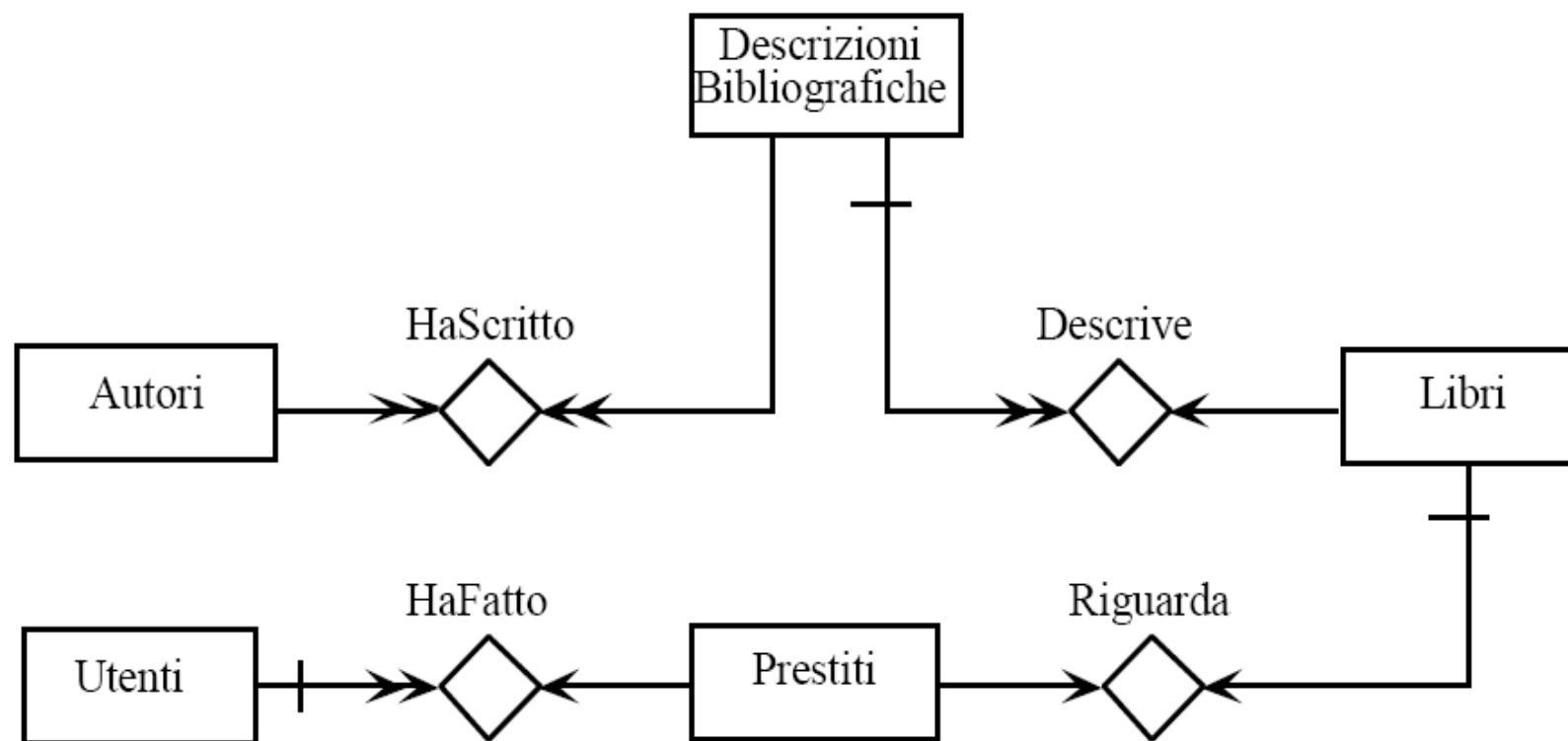
1. Di una **descrizione bibliografica** interessano il codice, il titolo dell'opera, l'editore, l'anno di pubblicazione e un insieme di termini usati per la classificazione del contenuto dell'opera.
2. Di un **libro** interessano la collocazione e il numero della copia.
3. Di un **autore** interessano il nome e cognome, la nazionalità, la data di nascita
4. Di un **utente** interessano il nome, il cognome, l'indirizzo e i recapiti telefonici.
5. Di un **prestito** interessano la data del prestito e la data di restituzione.

DIAGRAMMI E-R: esempio

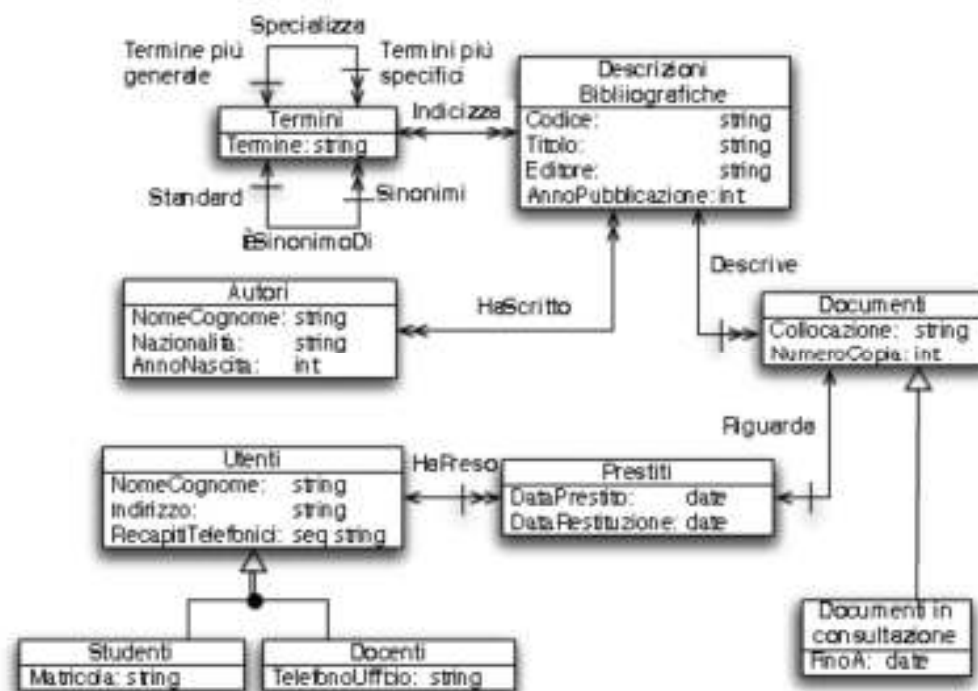
Le relazioni interessanti sono:

1. **HaScritto** (N:M) tra autori e descrizioni bibliografiche, che collega un autore con le descrizioni bibliografiche delle opere che ha scritto. Ogni autore ha scritto almeno un libro e ogni descrizione bibliografica riguarda almeno un autore;
2. **Descrive** (N:1) tra descrizioni bibliografiche e libri, che collega una descrizione bibliografica alle copie dei libri presenti in biblioteca. Ogni libro ha una descrizione bibliografica e ogni descrizione bibliografica descrive una o più copie di libri, ed anche un libro ordinato ma non ancora acquisito dalla biblioteca;
3. **HaFatto** (N:1) tra utenti e prestiti, che collega gli utenti ai prestiti che ha fatto e che non sono ancora scaduti. Ogni utente della biblioteca può avere nessuno, uno o più prestiti, ma un prestito ha sempre associato l'utente che lo ha fatto;
4. **Riguarda** (1:1) tra prestiti e libri, che collega i prestiti alle copie dei libri interessati. Una copia di un libro può essere coinvolta in al più un prestito e un prestito riguarda una copia di un libro.

DIAGRAMMI E-R: esempio



DIAGRAMMI E-R: esempio



INTRODUZIONE

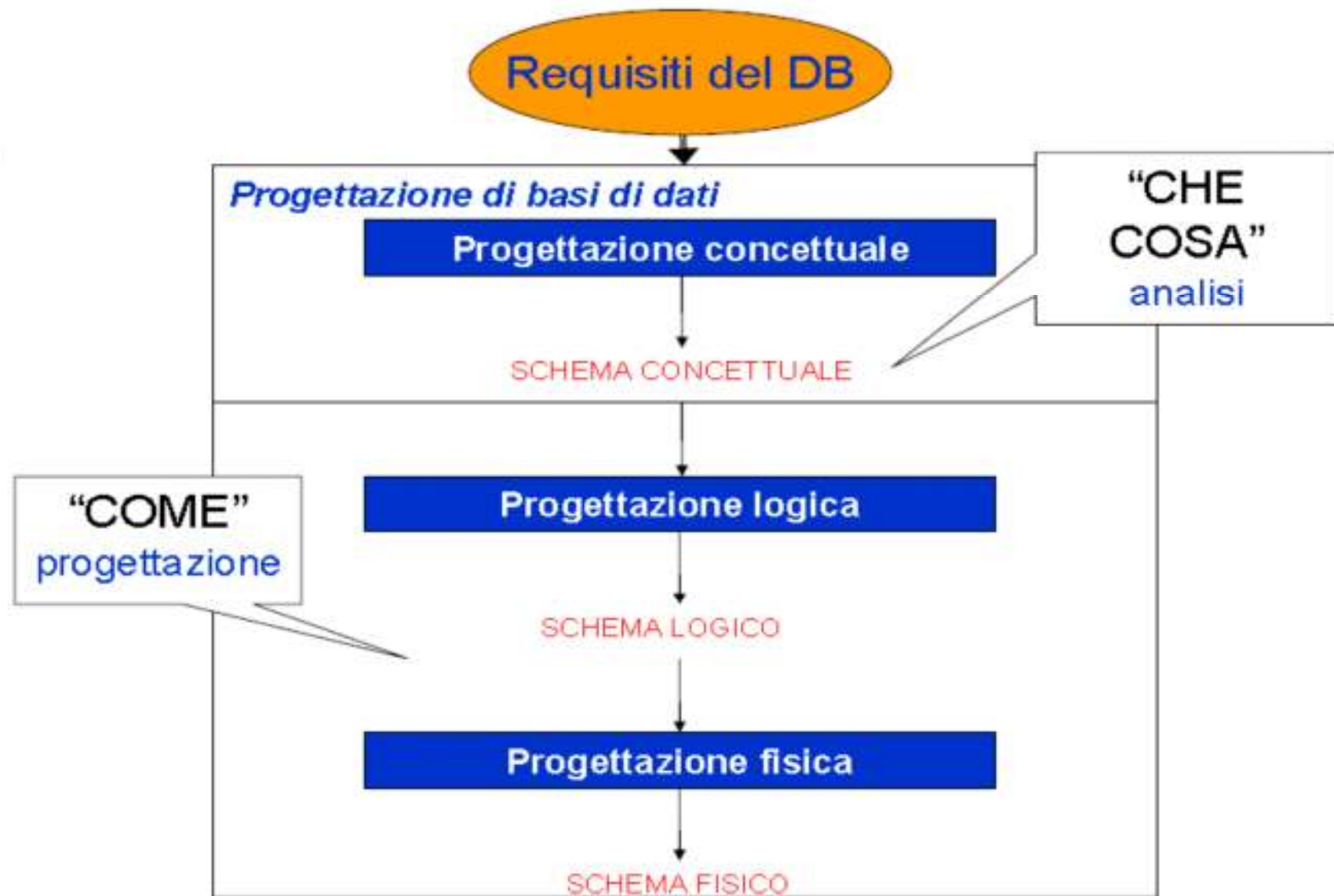
Finora si è visto come definire lo schema concettuale di una base di dati.

Nella pratica non è così semplice ed occorre un lungo procedimento di analisi e studio della situazione da modellare per poter produrre una **progettazione concettuale** della base di dati e poi una sua realizzazione.

Per dare un'idea di come si procede, si considera una tipica metodologia a più fasi in cui gli aspetti del problema vengono considerati gradualmente per ottenere una realizzazione soddisfacente.

Quattro fasi:

1. **analisi dei requisiti**
2. **progettazione concettuale**
3. **progettazione logica**
4. **progettazione fisica**



ANALISI DEI REQUISITI

Scopo dell'analisi dei requisiti è la definizione dei bisogni informativi del committente.

Il progettista deve capire di “cosa si parla”. Poi si passa ad un'analisi del problema per raccogliere una descrizione dei bisogni informativi e formulare la cosiddetta specifica dei requisiti in linguaggio naturale.

Quando il problema è di limitata complessità, e basta interagire con una sola persona, il procedimento può essere relativamente veloce, ma quando il problema è complesso e sono coinvolte persone diverse, il procedimento diventa lungo e comporta un lavoro di unificazione dei concetti coinvolti.

Lo scopo dell'analisi dei requisiti è, in altre parole, proprio quello di chiarire la corretta interpretazione dei fatti descritti, riformulando la specifica in modo chiaro.

Esempio

Si vuole progettare una base di dati per gestire informazioni su musei, le opere conservate, gli artisti che hanno creato tali opere.

Di un museo interessano il nome, che lo identifica, la città, l'indirizzo e il nome del direttore.

Un artista è identificato dal nome; di lui interessano la nazionalità, la data di nascita, la eventuale data di morte.

Di un'opera, identificata da un codice, interessano, l'anno di creazione, il titolo, il nome dei personaggi rappresentati.

Un'opera può essere un dipinto od una scultura; se è un dipinto interessano il tipo di pittura e le dimensioni; se è una scultura interessano il materiale, l'altezza ed il peso.

PROGETTAZIONE CONCETTUALE

La progettazione concettuale traduce la specifica dei requisiti in un progetto della struttura concettuale dei dati descritta utilizzando un formalismo grafico.

Lo schema concettuale si definisce con i seguenti passi:

- 1. identificazione delle entità;**
- 2. descrizione delle relazioni fra le entità;**
- 3. definizione gerarchie;**
- 4. definizione delle proprietà degli elementi delle entità.**

PROGETTAZIONE CONCETTUALE

Identificazione delle entità

Si produce una lista preliminare degli oggetti che interessa modellare e si assegna ad ognuna di esse un nome appropriato. Questo elenco iniziale ha un grado di completezza e di significatività che dipende dal grado di comprensione del problema e, in generale, sarà soggetto a modifiche mano a mano che si procede.

Ad ogni classe di oggetti corrisponde una entità

PROGETTAZIONE CONCETTUALE

Descrizione delle associazioni fra gli oggetti = delle relazioni tra entità

Si individuano le possibili associazioni fra gli oggetti finora definite e le loro proprietà strutturali. L'analisi delle associazioni può portare ad eliminare una classe che può essere rappresentata da un'associazione, o ad aggiungere una nuova classe per rappresentare un'associazione.

Ogni associazione viene modellata con una relazione

PROGETTAZIONE CONCETTUALE

Definizione di gerarchie

Per definire le gerarchie si esaminano tutte gli oggetti e le entità già definite per capire

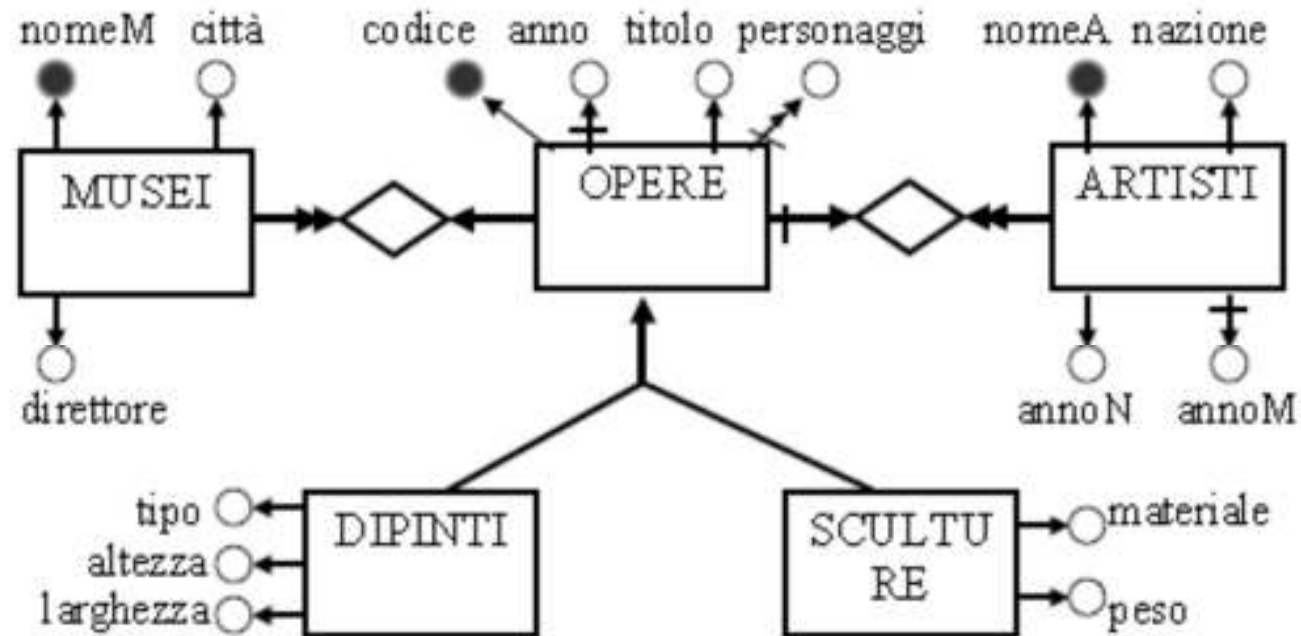
- 1. se può essere utile definirne di nuove per caratterizzare particolari sottoinsiemi di alcune classi**
- 2. se esistono classi che sono un sottoinsieme di altre e quindi possono essere ridefinite**
- 3. se esistono oggetti di classi che possono assumere nel tempo stati significativi per l'applicazione**

PROGETTAZIONE CONCETTUALE

Definizione delle proprietà degli elementi delle classi

Per ogni tipo di oggetto si elencano le proprietà interessanti, specificando, per ognuna di esse, il nome e il tipo. In questo passo va prestata molta attenzione alla possibilità se convenga introdurre nuove classi (che diventeranno poi entità) o viceversa eliminarne alcune sostituendole da attributi.

Esempio



PROGETTAZIONE LOGICA

Scopo della terza fase della metodologia, la **progettazione logica**, è di tradurre lo schema concettuale nello schema logico espresso nel modello dei dati del sistema scelto per la realizzazione della base di dati.

Esempio

MUSEI (NomeM, Città, Indirizzo, Direttore)

ARTISTI (NomeA, Nazionalità, DataN, DataM:optional)

OPERE (Codice, Anno, Titolo, NomeM*, NomeA*)

PERSONAGGI (Personaggio, Codice*)

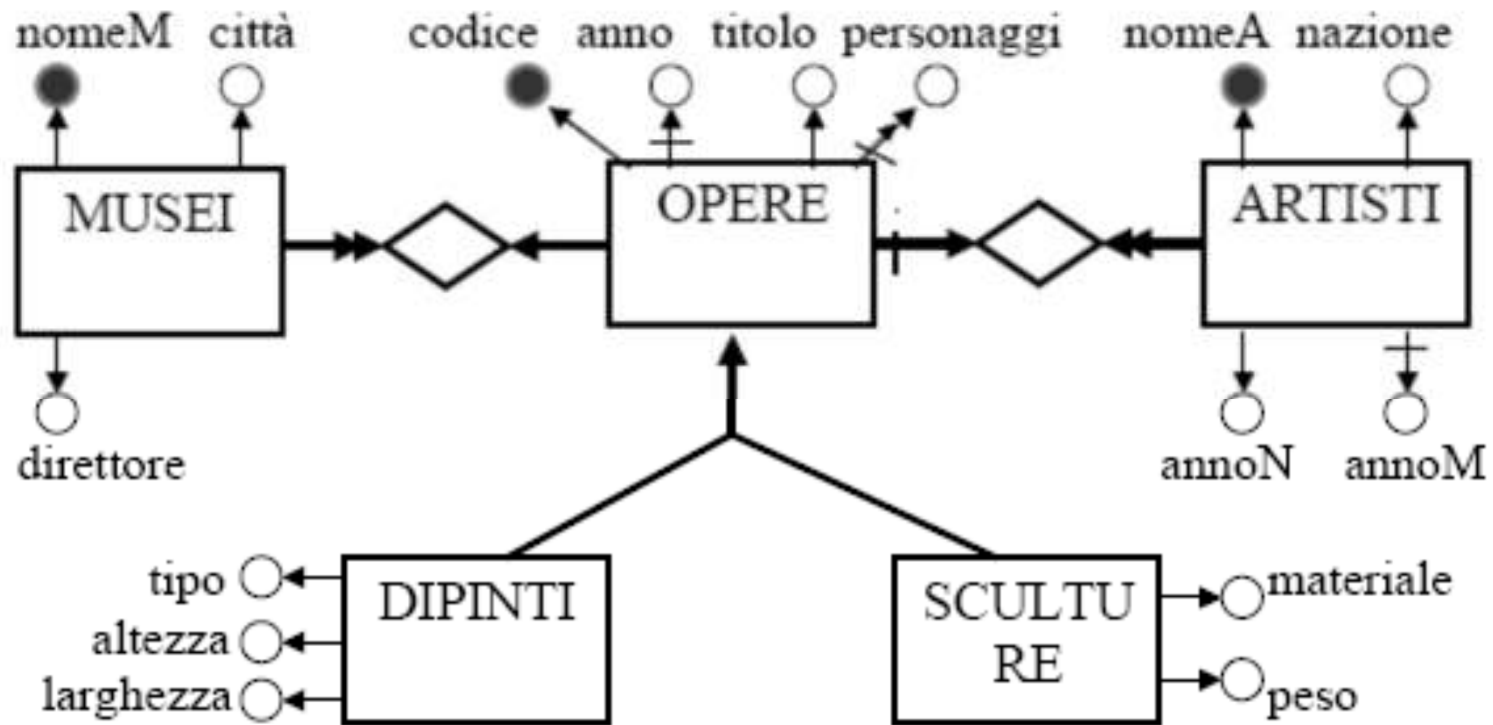
DIPINTI (Codice*, Tipo, Larghezza, Altezza)

SCULTURE (Codice*, Materiale, Altezza, Peso)

PROGETTAZIONE LOGICA

- La descrizione del modello da realizzare è fatta in funzione delle caratteristiche del DBMS che sarà utilizzato (il modello dei dati, il linguaggio per sviluppare le applicazioni).
- Il modello dei dati, detto **modello logico**, è “più vicino” alla rappresentazione informatica dei dati.
- Vedremo l’uso del **Modello dei Dati Relazionale**. Lo schema risultante è detto **schema relazionale**, ed è rappresentato con un formalismo testuale.
 - è un modello dei dati “compreso” dal DBMS
 - si ottiene con una **traduzione dello schema E-R**

ESEMPIO



ESEMPIO

VIENE TRADOTTO NELLO SCHEMA

MUSEI (NomeM, Città, Indirizzo, Direttore)

ARTISTI (NomeA, Nazionalità, DataN, DataM)

OPERE (Codice, Anno, Titolo, NomeM*, NomeA*)

PERSONAGGI (Personaggio, Codice*)

DIPINTI (Codice*, Tipo, Larghezza, Altezza)

SCULTURE (Codice*, Materiale, Altezza, Peso)

INTRODUZIONE

Il modello relazionale dei dati, proposto nel 1970 ed adottato nei sistemi commerciali a partire dal 1978, si è diffuso rapidamente tanto sui sistemi centrali quanto sugli elaboratori personali.

I meccanismi per definire una base di dati con il modello relazionale sono solo due:

- l'**ennupla**
- la **relazione**.

Si basa sul concetto matematico di relazione. Le relazioni hanno una rappresentazione naturale per mezzo di tabelle

ENNUPLA E RELAZIONE

Ennupla. È un insieme finito di coppie (**Attributo, valore atomico**)

Relazione. È un insieme finito (eventualmente vuoto) di ennuple con la stessa struttura.

I campi di un'ennupla sono atomici (numeri, stringhe o il valore NULL).

Si tenga presente che non si possono rappresentare proprietà strutturate, o multivalore e vedremo più avanti come risolvere questi problemi di rappresentazione.

Un'ennupla si usa per rappresentare entità e la relazione si usa per rappresentare classi di entità.

TERMINOLOGIA - I

- ❑ **attributo**: corrisponde (non sempre) ad un attributo del modello E-R. Diversamente dal modello E-R, gli attributi sono sempre **univoci** (ad un sol valore) ed **atomici** (non composti)
 - ❑ L'ordine degli attributi non è significativo
- ❑ **dominio** (di un attributo): è l'insieme dei valori che può assumere un attributo.
- ❑ **chiave primaria** di una relazione: un attributo che identifica univocamente le ennuple della relazione. Gli attributi della chiave primaria vengono sottolineati

Riassumendo

Le regole di traduzione

- **Entità**: diventano tabelle ed i loro identificatori chiavi primarie
- **Associazioni 1-1**: se obbligatorie si procede come per le 1-N scegliendo il lato in cui includere gli attributi e la chiave esterna; se una opzionale si includono gli attributi e la chiave esterna sul lato “obbligatorio”; se entrambe opzionali si costruisce una tabella autonoma come per il caso N-N.
- **Associazioni 1-N**: gli attributi dell’associazione e la chiave primaria della tabella relativa all’entità dal lato “N” sono inclusi nella tabella relativa all’entità dal lato “1”.
- **Associazioni N-N**: diventano tabelle con chiave primaria formata dall’unione delle chiavi delle entità coinvolte

Confronto tra modello E-R e modello Relazionale

Nonostante una evidente analogia tra

Classe e Relazione

Oggetto e Ennupla

Attributo(E-R) e Attributo(Rel)

sussiste in realtà una significativa differenza tra i concetti nei due modelli

- Non sempre una relazione rappresenta una classe: può rappresentare una associazione o un attributo multivalore
- Non sempre un' ennupla rappresenta un oggetto: può rappresentare una coppia di oggetti in associazione o un possibile valore di un attributo multivalore
- Non sempre un attributo Rel rappresenta un attributo E-R: può rappresentare un oggetto di un'altra classe (chiave esterna)

Vincolo di integrità

- Proprietà che deve essere soddisfatta dalle istanze che rappresentano informazioni corrette per l'applicazione
- Un vincolo è una **funzione booleana** (un **predicato**): associa ad ogni istanza il valore **vero** o **falso**
- **Perchè?**
 - descrizione più accurata della realtà
 - contributo alla “qualità dei dati”
 - utili nella progettazione (vedremo)
 - usati dai DBMS nella esecuzione delle interrogazioni

Tipi di vincoli

- Vincoli intrarelazionali: coinvolgono una singola relazione del database.
 - Esempi
 - vincoli di ennupla
 - vincoli di chiave
 - vincoli su valore
- Vincoli interrelazionali: coinvolgono diverse relazioni del database.
 - Esempio: vincoli di integrità referenziale

Vincoli di ennupla

- Esprimono condizioni sui valori di ciascuna ennupla, indipendentemente dalle altre ennuple
- Caso particolare:
 - vincoli di dominio: coinvolgono un solo attributo
- Una possibile sintassi:
 - espressione booleana di atomi che confrontano valori di attributo o espressioni aritmetiche su di essi

`(Voto ≥ 18) AND (Voto ≤ 30)`

`(Voto = 30) OR NOT (Lode = "e lode")`

Importanza delle chiavi

- Chiave: insieme di attributi che identificano univocamente le ennuple di una relazione
- L'esistenza delle chiavi garantisce l'accessibilità a ciascun dato della base di dati
- Le chiavi permettono di correlare i dati in relazioni diverse:
 - il modello relazionale è basato su valori
- In presenza di valori nulli, i valori della chiave non permettono
 - di identificare le ennuple
 - di realizzare facilmente i riferimenti da altre relazioni

Integrità referenziale

- Informazioni in relazioni diverse sono correlate attraverso valori comuni
 - in particolare, valori delle chiavi (primarie)
- le correlazioni debbono essere "coerenti"

Vincolo di integrità referenziale

- Un vincolo di **integrità referenziale** (“foreign key”) fra gli attributi X di una relazione R_1 e un'altra relazione R_2 impone ai valori su X in R_1 di comparire come valori della chiave primaria di R_2
- ES: vincoli di integrità referenziale fra: l'attributo Vigile della relazione INFRAZIONI e la relazione VIGILI

Azioni compensative

- Esempio:
 - Viene eliminata una ennupla causando una violazione
- Comportamento “standard”:
 - Rifiuto dell'operazione
- Azioni compensative:
 - Eliminazione in cascata
 - Introduzione di valori nulli

Forma normale di Boyce e Codd (BCNF)

- Una relazione r è in forma normale di Boyce e Codd se, per ogni dipendenza funzionale (non banale) $X \rightarrow Y$ definita su di essa, X contiene una chiave K di r
- In sostanza: la forma normale BCNF richiede che i concetti in una relazione siano omogenei (solo proprietà direttamente associate alla chiave)

Che facciamo se una relazione non soddisfa la BCNF?

- La rimpiazziamo con altre relazioni che soddisfano la BCNF

Come?

- Decomponendo sulla base delle dipendenze funzionali, al fine di separare i concetti

Decomposizione senza perdita

- Una relazione r si **decompone senza perdita** su X_1 e X_2 se il join delle proiezioni di r su X_1 e X_2 è uguale a r stessa (cioè non contiene ennuple spurie)
- *La decomposizione senza perdita è garantita se gli attributi comuni tra X_1 e X_2 contengono una chiave in almeno una delle relazioni decomposte*

Conservazione delle dipendenze

- Una decomposizione **conserva le dipendenze** se *ciascuna delle dipendenze funzionali dello schema originario coinvolge attributi che compaiono tutti insieme in uno degli schemi decomposti*

Qualità delle decomposizioni

- Una decomposizione dovrebbe sempre soddisfare:
 - la **decomposizione senza perdita**, che garantisce la ricostruzione delle informazioni originarie
 - la **conservazione delle dipendenze**, che garantisce il mantenimento dei vincoli di integrità originali

Un'altra forma normale

- Una relazione r è in **terza forma normale** se, per ogni dipendenza funzionale (non banale) $X \rightarrow Y$ definita su r , è verificata almeno una delle seguenti condizioni:
 - X contiene una chiave K di r
 - ogni attributo in Y è contenuto in almeno una chiave di r

BCNF e terza forma normale

- la terza forma normale è meno restrittiva della forma normale di Boyce e Codd (e ammette relazioni con alcune anomalie)
- ha il vantaggio però di essere sempre “raggiungibile”
 - Cio' si puo' provare

Decomposizione in terza forma normale

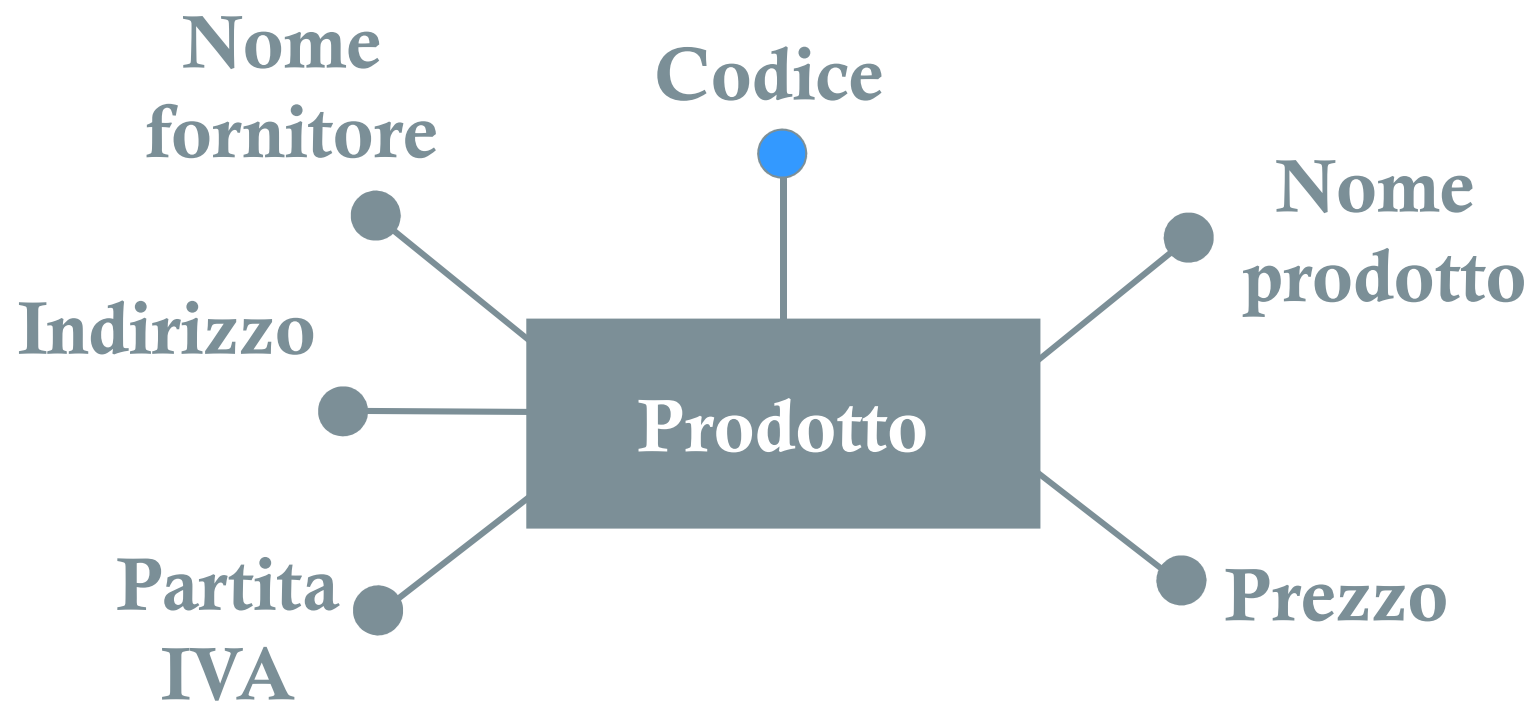
- si crea una relazione per ogni gruppo di attributi coinvolti in una dipendenza funzionale
- si verifica che alla fine una relazione contenga una chiave della relazione originaria
- Dipende dalle dipendenze individuate

Una possibile strategia

- se la relazione non è normalizzata si decompone in terza forma normale
- alla fine si verifica se lo schema ottenuto è anche in BCNF
- Se una relazione ha una sola chiave allora le due forme normali coincidono

Progettazione e normalizzazione

- la teoria della normalizzazione può essere usata nella progettazione logica per verificare lo schema relazionale finale
- si può usare anche durante la progettazione concettuale per verificare la qualità dello schema concettuale



PartitaIVA → NomeFornitore Indirizzo

Analisi dell'entità

- L'entità viola la terza forma normale a causa della dipendenza:

PartitaIVA → NomeFornitore Indirizzo

- Possiamo decomporre sulla base di questa dipendenza

