

Prova di Informatica A.s. 2007-2008

Tema di: INFORMATICA GENERALE E APPLICAZIONI TECNICO SCIENTIFICHE

21/06/2007

Prof. Alberto Ferrari

Esame di Stato di Istituto Tecnico Industriale

Informatica

A.S. 2006-2007

L'argomento in esame risulta particolarmente attuale vista la sempre maggior diffusione di "negozi virtuali" che propongono la vendita on line di articoli di varia natura.

Analisi e ipotesi aggiuntive

Viene richiesto di progettare e realizzare un database che contenga le informazioni necessarie per la vendita on line di libri a utenti registrati.

Libri e *Utenti* risultano quindi le entità principali del sistema informatico che si andrà a realizzare.

Il testo è molto preciso nel descrivere le informazioni che dovranno essere associate ai **libri**:

- suddivisione in reparti (architettura e urbanistica, arte, classici greci e latini ...) .
Pur se non esplicitamente evidenziato risulta plausibile ipotizzare che un libro faccia parte di un solo reparto
- ulteriore suddivisione dei Reparti in Categorie, nel testo viene utilizzato il termine "libri" che sostituiamo con "categoria" per evitare ambiguità, ("Da non perdere", "I più venduti" ...); ogni categoria ha una caratterizzazione che può riguardare la tipologia di sconto proposto, il numero di copie vendute o la data di pubblicazione e inserimento in catalogo.
Anche in questo caso non viene esplicitamente definito il fatto che un libro possa appartenere a una sola o a più categorie; nella nostra proposta daremo la possibilità di inserire un libro in più di una categoria pensando che, per esempio, un libro possa appartenere contemporaneamente alla categoria "Da non perdere" e "I più venduti"
- per ogni libro vengono inoltre definite in modo preciso le caratteristiche che dovranno essere inserite nella base di dati
In particolare analizziamo:
 - o *lo sconto che ipotizziamo non strettamente legato alla categoria di appartenenza*
 - o *una serie di dati non particolarmente strutturati (numero di pagine, tipo di rilegatura ...) che rappresentiamo in un unico attributo generico testuale sul quale non sarà prevista nessuna operazione particolare*
 - o *le fotografie delle copertine dei libri saranno memorizzate in file esterni al database all'interno del quale sarà inserito un riferimento a tali file.*

Anche per gli **utenti** vengono definite in modo preciso le caratteristiche che dovranno essere inserite nella base di dati.

- Nessuna caratteristica pone particolari problemi di gestione. Ci sentiamo solo in dovere di precisare che il numero di carta di credito di ogni utente è un dato estremamente sensibile che andrebbe trattato con opportune procedure che ne garantiscano la sicurezza (tale problema esula però dalle richieste formulate nel testo)

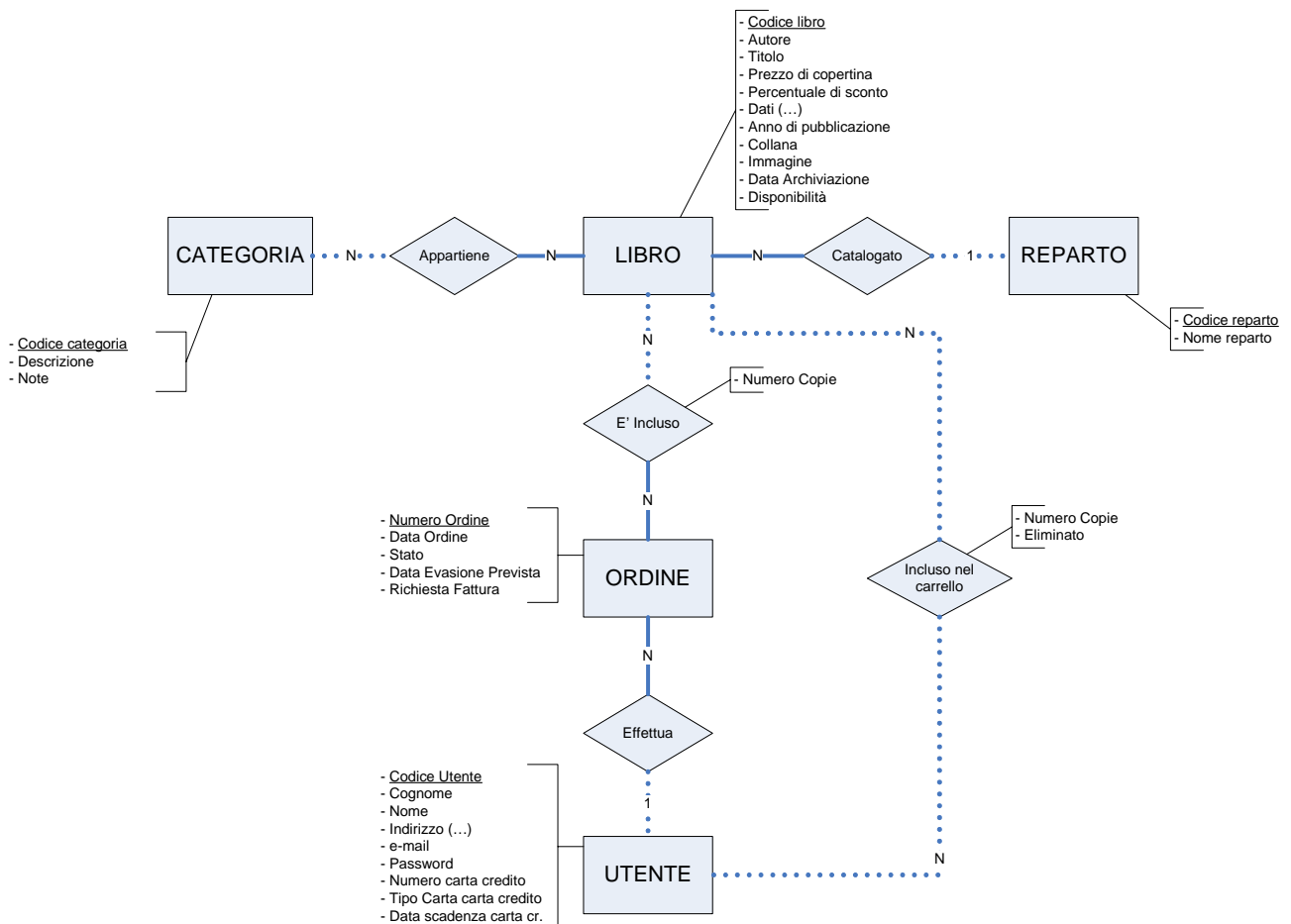
Ad ogni utente, ad ogni nuovo accesso al negozio on line, sarà assegnato un *carrello acquisti* virtuale nel quale inserire i libri selezionati che potranno poi essere confermati (tutti, o in parte) per dar luogo a un ordine di acquisto.

Ulteriori ipotesi

Nella definizione del carrello acquisti il testo parla di "disponibilità" (giorni, settimane ...), dato che intendiamo memorizzato in ogni singolo libro nel database

Schema Concettuale

Realizziamo lo schema concettuale utilizzando i Diagrammi Entità Associazioni



Commento al diagramma:

L'entità **REPARTO** rappresenta la suddivisione in reparti del catalogo della libreria.

Codice reparto è attributo chiave, *Nome reparto* è la descrizione (*architettura e urbanistica, arte ...*)

L'entità **CATEGORIA** rappresenta le categorie a cui appartengono i libri.

Codice categoria è attributo chiave, *Descrizione* è la descrizione della categoria ("Da non perdere", "I più venduti" ...) *Note* permette di inserire un commento ("Sconto del 20%" ...)

L'entità **LIBRO** rappresenta i libri presenti nel catalogo.

Codice libro è attributo chiave, *Dati* è un attributo che raggruppa informazioni varie (numero di pagine, tipo di rilegatura ...), *Disponibilità* è un attributo testuale utilizzato in fase di ordine; gli altri attributi sono di chiara interpretazione.

L'associazione **Catalogato** associa un libro ad un reparto.

L'associazione **Appartiene** associa un libro a varie categorie.

L'entità **UTENTE** rappresenta gli utenti registrati che possono effettuare acquisti.

Codice utente è attributo chiave, l'attributo *Indirizzo* è composto da città, via, numero civico ecc., gli altri attributi sono di chiara interpretazione.

L'entità **ORDINE** rappresenta gli ordini di acquisto effettuati dagli utenti.

Numero Ordine è attributo chiave, l'attributo *Stato* indica se l'ordine è evaso o da evadere, l'attributo *Richiesta Fattura* indica se l'utente ha richiesto fattura in fase di emissione d'ordine.

L'associazione **Incluso nel carrello** associa i libri agli utenti e rappresenta il carrello acquisti che l'utente "riempie" ogni volta che accede al sito del negozio virtuale.

Numero Copie è un attributo dell'associazione e rappresenta il numero di copie di un certo libro che l'utente ha inserito nel carrello, l'attributo *Eliminato* è utilizzato per eliminare il libro dal carrello ed evitare così di inserirlo nell'ordine.

L'associazione **Effettua** associa un utente ai suoi ordini d'acquisto.

L'associazione **E' incluso** associa un libro ad un ordine.

L'attributo *Numero Copie* indica il numero di copie di un libro inserite in un ordine.

Letture del diagramma:

Ad una CATEGORIA *possono appartenere* uno o più LIBRI.

Un LIBRO **appartiene** a una o più CATEGORIE.

In un REPARTO *possono* essere **catalogati** uno o più LIBRI.

Un LIBRO è **catalogato** in un solo REPARTO.

Un LIBRO *può* essere **incluso** nel carrello di uno o più UTENTI.

Un UTENTE può **includere** nel suo carrello uno o più LIBRI.

Un UTENTE *può* **effettuare** uno o più ORDINI.

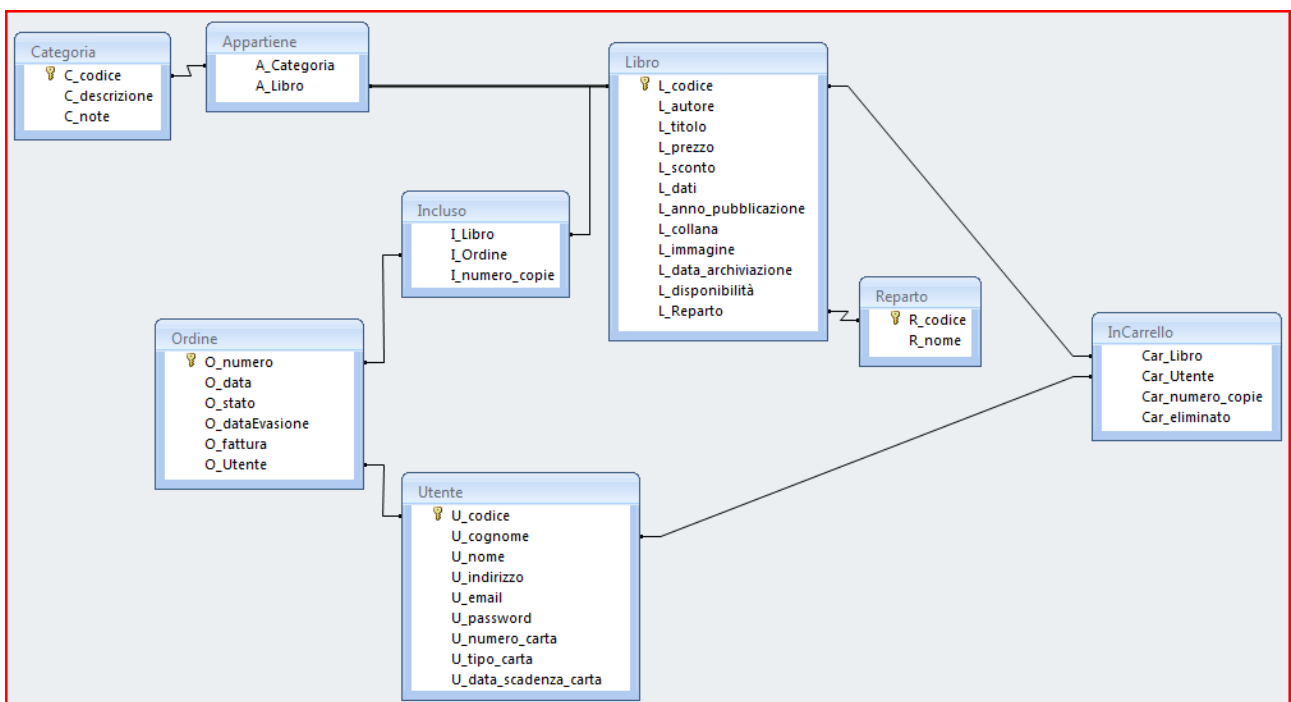
Un ORDINE è **effettuato** da un solo UTENTE.

Un LIBRO *può* essere **incluso** in uno o più ORDINI.

Un ORDINE **include** uno o più LIBRI.

Schema logico fisico

Per la realizzazione del database utilizziamo il modello relazionale e attraverso le regole di derivazione passiamo dallo schema concettuale proposto al seguente schema relazionale:



Lo schema relazionale è creato dai seguenti comandi SQL

```
CREATE TABLE REPARTO
```

```
(  
  R_codice CHAR(20),  
  R_nome CHAR(50) NOT NULL,  
  PRIMARY KEY(R_Codice)  
)
```

```
CREATE TABLE CATEGORIA
```

```
(  
  C_codice CHAR(20),  
  C_descrizione CHAR(50) NOT NULL,  
  C_note CHAR(80),  
  PRIMARY KEY(C_Codice)  
)
```

```
CREATE TABLE LIBRO
```

```
(  
  L_codice CHAR(13),  
  L_autore CHAR(30),  
  L_titolo CHAR(50) NOT NULL,  
  L_prezzo MONEY NOT NULL,  
  L_sconto INTEGER,  
  L_dati CHAR(100),  
  L_anno_pubblicazione INTEGER,  
  L_collana CHAR(30),  
  L_immagine CHAR(30),  
  L_data_archiviazione DATE,  
  L_disponibilita CHAR(20),  
  L_Reparto CHAR(20),  
  PRIMARY KEY(L_Codice),  
  FOREIGN KEY (L_Reparto) REFERENCES REPARTO (R_codice)  
)
```

```
CREATE TABLE APPARTIENE
```

```
(  
  A_Categoria CHAR(20),  
  A_Libro CHAR(13),  
  FOREIGN KEY (A_Categoria) REFERENCES CATEGORIA (C_codice),  
  FOREIGN KEY (A_Libro) REFERENCES LIBRO (L_codice)  
)
```

CREATE TABLE **UTENTE**

```
(
U_codice CHAR(20),
U_cognome CHAR(30) NOT NULL,
U_nome CHAR(30) NOT NULL,
U_indirizzo CHAR(60) NOT NULL,
U_e_mail CHAR(30) NOT NULL,
U_password CHAR(12) NOT NULL,
U_numero_carta CHAR(30) NOT NULL,
U_tipo_carta CHAR(30) NOT NULL,
U_data_scadenza_carta DATE,
PRIMARY KEY(U_Codice)
)
```

CREATE TABLE **INCARRELLO**

```
(
Car_Libro CHAR(13),
Car_Utente CHAR(20),
Car_numero_copie INTEGER,
Car_eliminato BIT,
FOREIGN KEY (Car_Libro) REFERENCES LIBRO (L_codice),
FOREIGN KEY (Car_Utente) REFERENCES UTENTE (U_codice)
)
```

CREATE TABLE **ORDINE**

```
(
O_numero INTEGER,
O_data DATE NOT NULL,
O_stato CHAR(10),
O_data_evasione DATE,
O_fattura BIT,
O_Utente CHAR(20),
PRIMARY KEY(O_numero),
FOREIGN KEY (O_Utente) REFERENCES UTENTE (U_codice)
)
```

CREATE TABLE **INCLUSO**

```
(
I_Libro CHAR(13),
I_Ordine INTEGER,
I_numero_copie INTEGER,
FOREIGN KEY (I_Libro) REFERENCES LIBRO (L_codice),
FOREIGN KEY (I_Ordine) REFERENCES Ordine (O_numero)
)
```

Interrogazioni SQL

1. Ricerca per Reparto: scelto un Reparto, il numero di libri "Da non perdere" ed il numero di libri "I più venduti" con i relativi dettagli

Ipotizziamo che la scelta del reparto avvenga tramite il nome (R_nome). Sono necessarie 4 tabelle: LIBRO, REPARTO, CATEGORIA, APPARTIENE

Con questa prima interrogazione otteniamo il numero di libri "Da non perdere" di un determinato reparto:

```
SELECT count(*) AS [Numero_Libri]
FROM Libro, Reparto, Categoria, Appartiene
WHERE (
L_Reparto=R_codice
AND
C_codice=A_Categoria
AND
A_Libro=L_codice
AND
R_nome=[Nome del reparto]
AND
C_descrizione="Da non perdere"
);
(analogo discorso per i libri "I più venduti")
```

Per ottenere il dettaglio dei libri (in questo caso presentiamo solo autore e titolo:

```
SELECT L_autore, L_titolo
FROM Libro, Reparto, Categoria, Appartiene
WHERE (
L_Reparto=R_codice
AND
C_codice=A_Categoria
AND
A_Libro=L_codice
AND
R_nome=[Nome del reparto]
AND
C_descrizione="Da non perdere"
);
(analogo discorso per i libri "I più venduti")
```


2. con accesso riservato agli utenti registrati, la composizione degli ordini in corso.

L'accesso è riservato agli utenti registrati che quindi possono consultare solo i propri ordini, ipotizziamo quindi che durante la fase di login venga memorizzato il codice utente (COD_UTENTE), l'attributo O_stato della tabella Ordini è utilizzato per individuare gli ordini in corso.

```
SELECT O_numero AS [N. Ordine], L_autore AS [Autore], L_titolo AS [Titolo], I_numero_copie AS [copie]
FROM Ordine, Incluso, Libro
WHERE (
I_Ordine=O_numero
AND
I_Libro=L_codice
AND
O_Utente=COD_UTENTE
AND
O_stato="in corso"
);
```

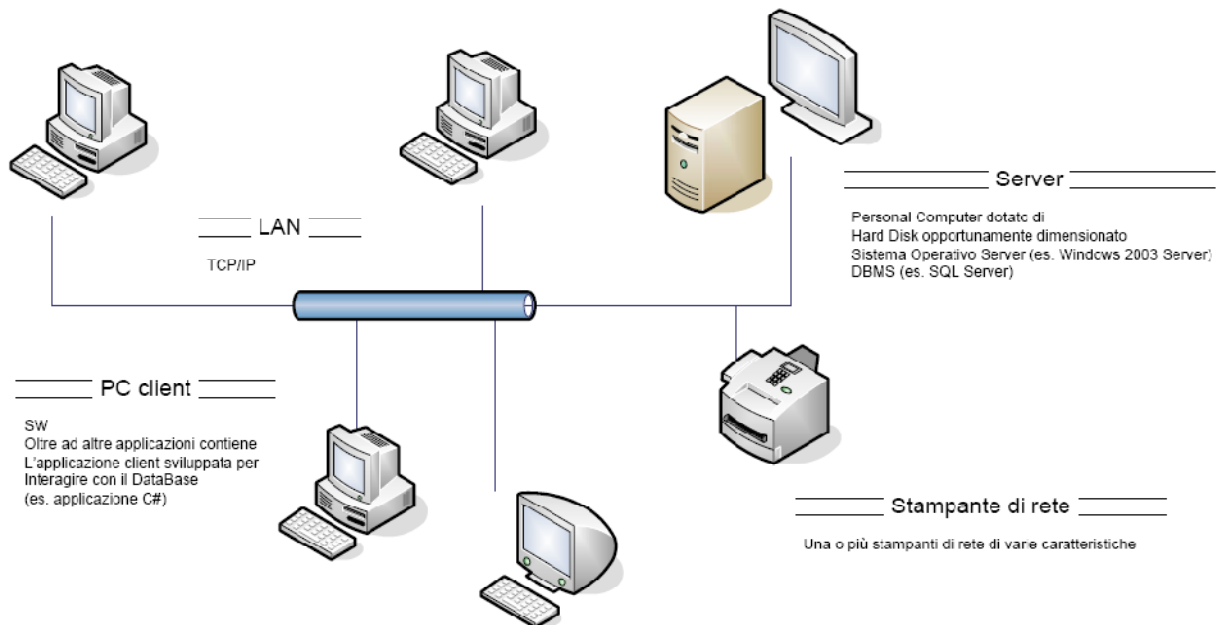
Architettura Hardware e Software

Per poter gestire la base di dati si può ipotizzare un'applicazione software che permetta di interagire con la base di dati per l'inserimento, modifica ed eliminazione delle informazioni oltre alle varie procedure di interrogazione.

Dal punto di vista Hardware si potrebbe ipotizzare una singola postazione di lavoro (Personal Computer) in cui ospitare il database ed il programma di gestione.

Nel caso di realtà di dimensioni maggiori si potrebbe ipotizzare una rete locale (Rete di PC) in cui un nodo avrà la funzione di server database e gli altri di client.

Per il database si potrebbe utilizzare un DBMS (esempio SQL Server) in grado di erogare il servizio anche via LAN. L'applicazione per la gestione e interrogazione dei dati potrebbe essere scritta in linguaggio ad alto livello (es. C#, Java) in cui incapsulare le istruzioni SQL per l'interazione con la base di dati.



Sito Web

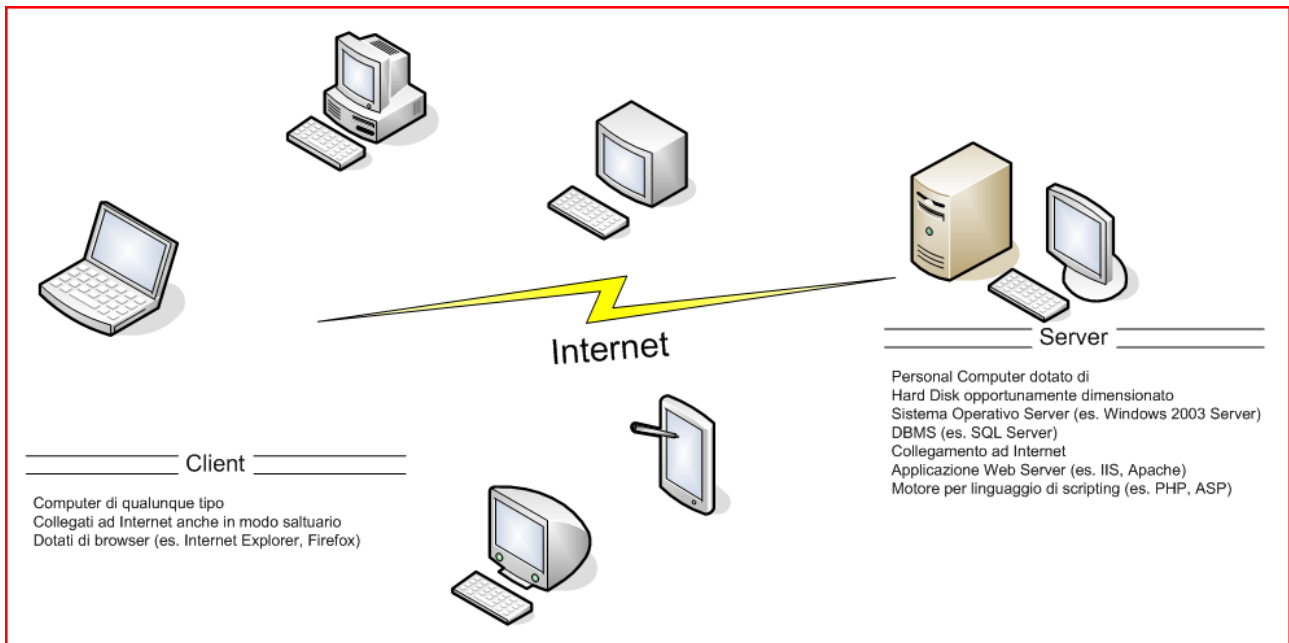
Il negozio on line deve ovviamente essere accessibile via web per la consultazione del catalogo e l'effettuazione degli ordini da parte degli utenti registrati.

Oltre al linguaggio HTML necessario per la realizzazione delle pagine del sito è necessario utilizzare un linguaggio di scripting lato server (es. PHP, ASP, ASP.NET) per poter interagire con il database, recuperare le informazioni e realizzare pagine dinamiche.

Le pagine che formano il sito devono essere memorizzate su un server collegato ad internet che potrebbe essere di proprietà del negozio oppure ospitato da un fornitore di servizi esterno.

Sul server deve essere in esecuzione un applicazione Web Server (es. IIS, Apache) ed un motore in grado di eseguire i comandi del linguaggio di programmazione lato server utilizzato oltre ad un server SQL per l'interazione con il database.

Le pagine possono essere consultate da un qualsiasi computer collegato ad internet e dotato di un browser (es. Internet Explorer, Firefox, Safari)



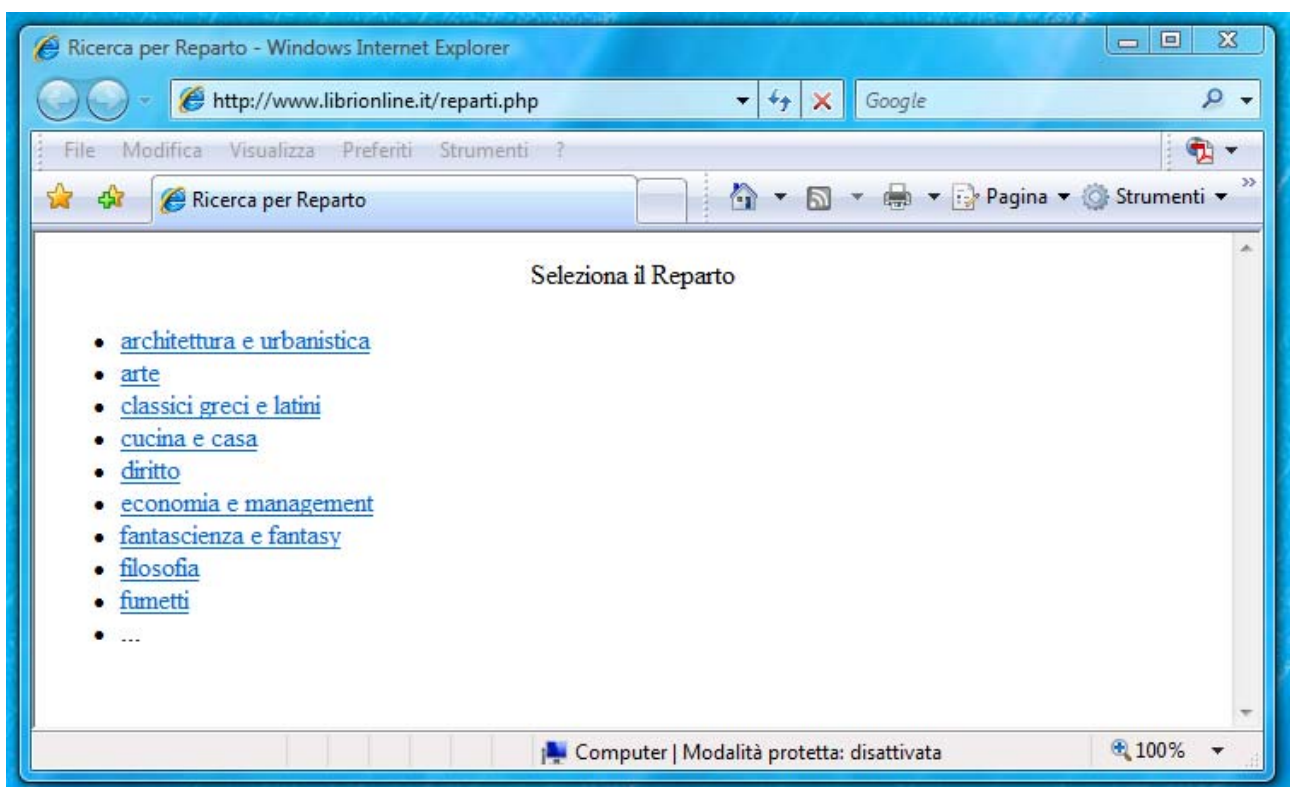
Portale

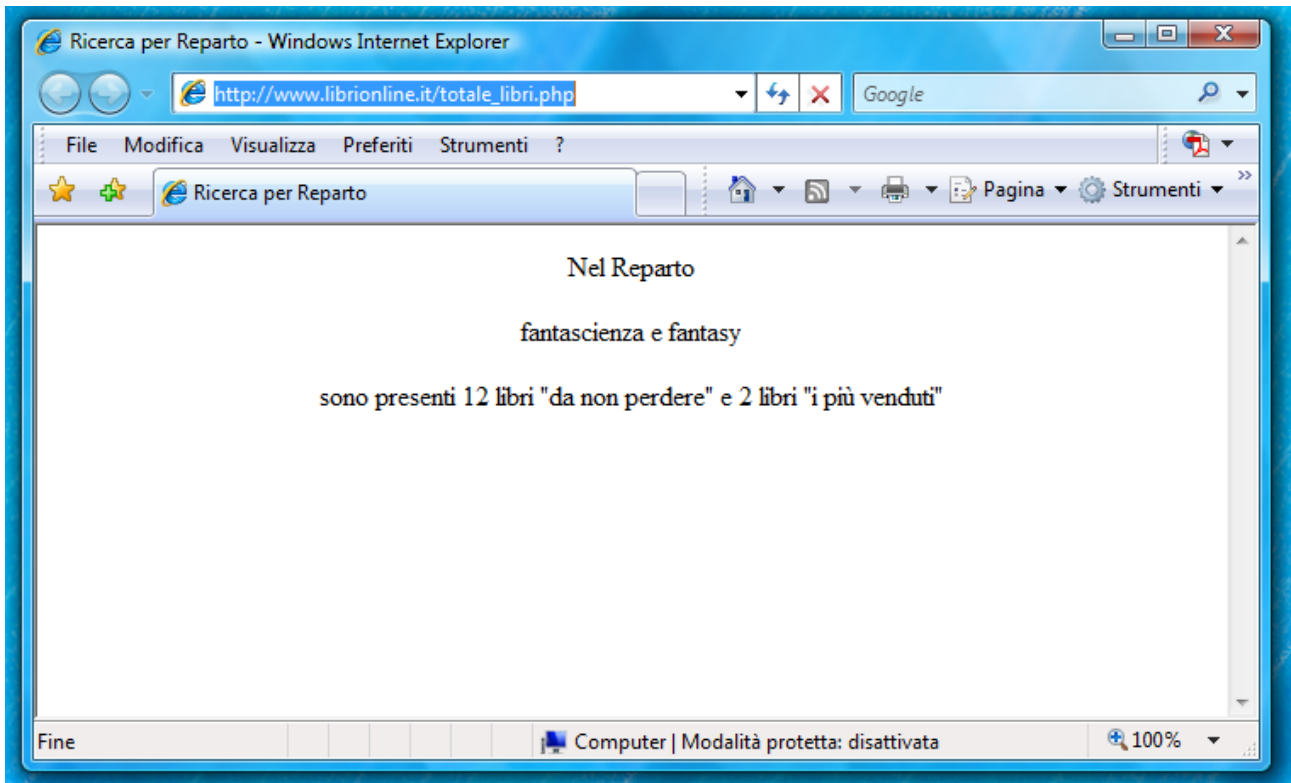
Per la realizzazione del portale è necessario costruire pagine web dinamiche con programmazione lato server; negli esempi proposti si è scelto MYSQL come DBMS e PHP come linguaggio per gli script.

- 1. con accesso libero, la pagina utile a visualizzare i Reparti e, per ciascun reparto, la pagina che implementa la query n. 1;*

La soluzione prevede due pagine, la prima (reparti.php) interroga il database e visualizza tutti i Reparti creando una serie di link alla seconda pagina (totale_libri.php) ognuno dei quali passa come parametro il reparto selezionato; la seconda pagina recupera il parametro che utilizza per l'interrogazione della query 1.

Il layout delle pagine è il seguente:





reparti.php

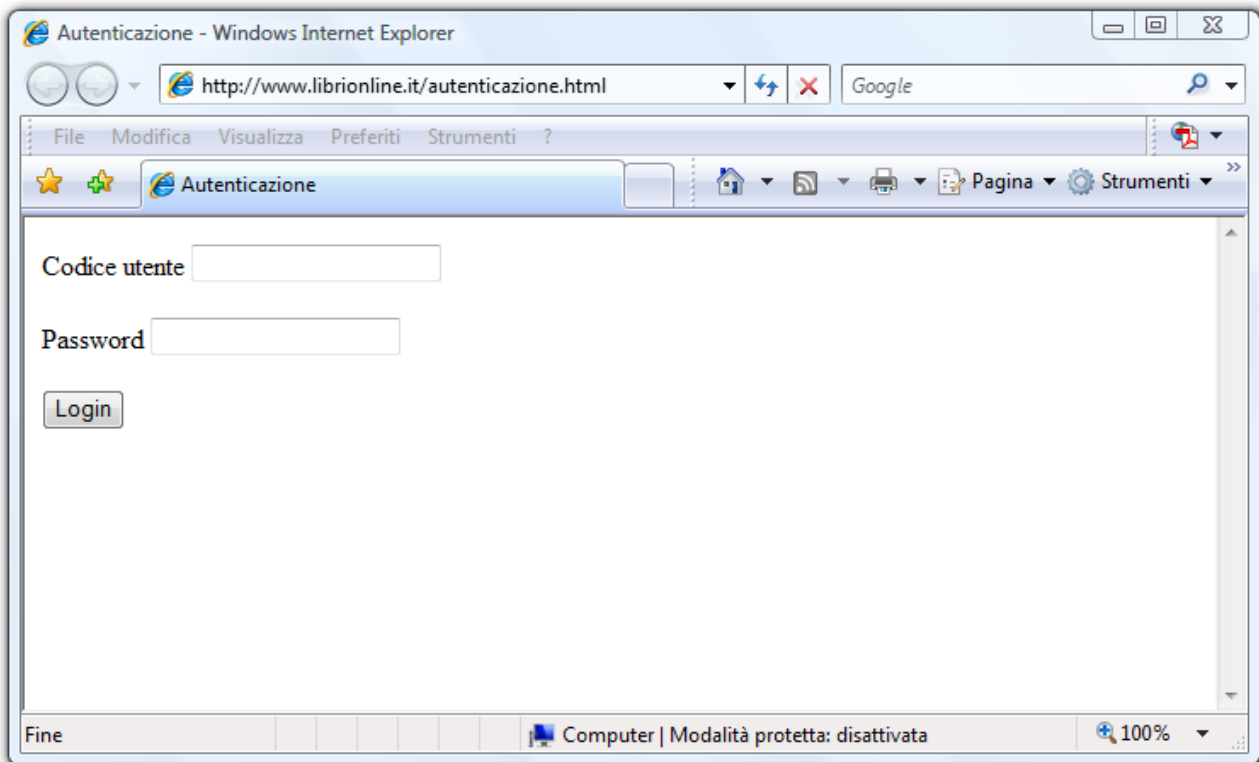
```
<html>
<head>
<title>Ricerca per reparto</title>
</head>
<body>
Seleziona il reparto
<?php
$host='localhost';
$nome_db='libri';
$conn=mysql_connect($host) or die ("errore di connessione");
$query='SELECT R_nome FROM REPARTO';
$elenco=mysql_query($query,$conn);
print " <ul>"
while ($reparto=mysql_fetch_array($elenco)
    print "<li><a href='\"totale_libri.php?REP_SEL=$reparto['R_nome']'\"> $reparto['R_nome']</a></li>";
print " </ul>";
mysql_close($conn);
?>
</body>
</html>
```

totale_libri.php

```
<html>
<head>
<title>Ricerca per reparto</title>
</head>
<body>
Nel reparto
<?php
$scelta=$_GET['REP_SEL'];
print "$scelta <br>"
$host='localhost';
$nome_db='libri';
$conn=mysql_connect($host) or die ("errore di connessione");
$query= 'SELECT count(*) AS [Numero_Libri] FROM Libro, Reparto, Categoria, Appartiene WHERE (
L_Reparto=R_codice AND C_codice=A_Categoria AND A_Libro=L_codice AND R_nome="$scelta" AND
C_descrizione="Da non perdere");
$totale=mysql_query($query,$conn);
print "Sono presenti $totale libri da non perdere";
// analogo discorso per i libri più veduti
mysql_close($conn);
?>
</body>
</html>
```

2. con accesso riservato agli utenti registrati, la composizione degli ordini in corso.

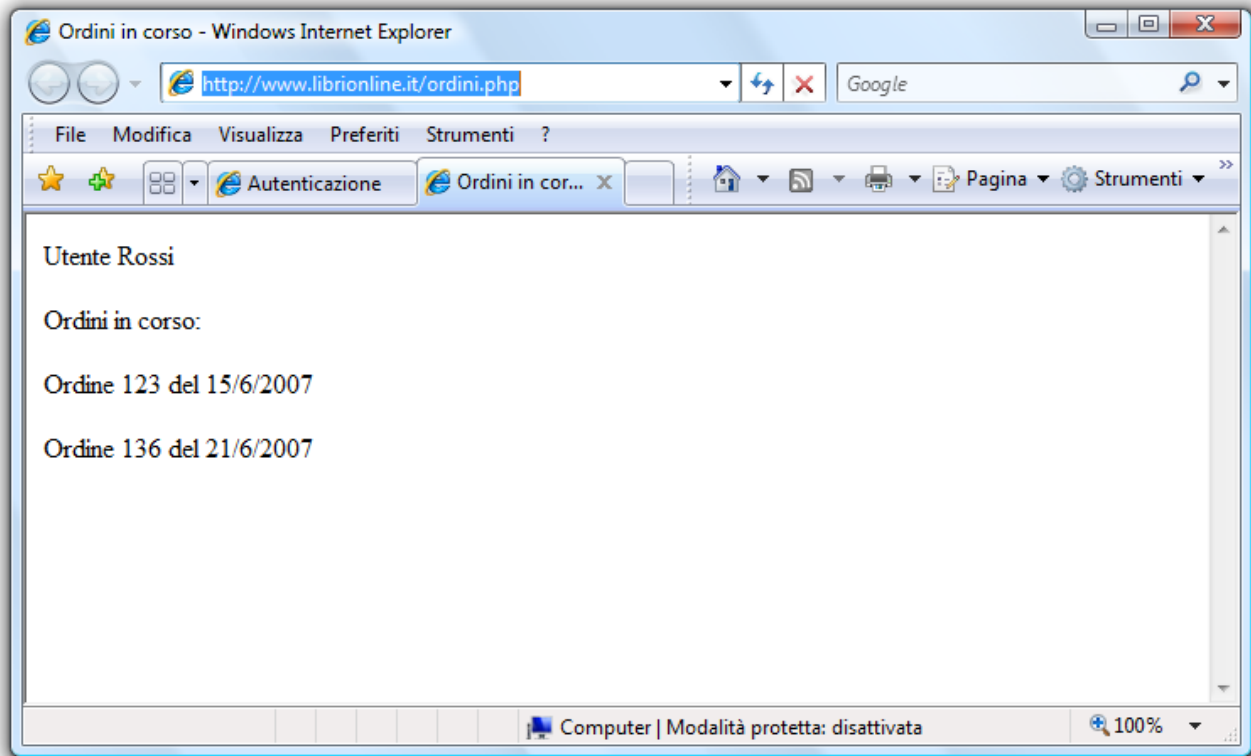
L'autenticazione dell'utente avviene mediante la ricerca di una corrispondenza dei dati ricevuti in input (codice utente e password) nel database.



autenticazione.html

```
<html>
<head>
<title>Autenticazione</title>
</head>
<body>
<form action="ordini.php" method="post" name="autenticazione" id="autenticazione">
  <p>Codice utente
    <input name="cod_utente" type="text" id="cod_utente">
  </p>
  <p>Password
    <input name="codice" type="password" id="codice">
  </p>
  <p>
    <input type="submit" name="Submit" value="Login">
  </p>
</form>
</body>
</html>
```

Si è scelto di utilizzare il metodo POST per ragioni di sicurezza poiché i dati scambiati fra le pagine contengono la password dell'utente



ordini.php

```
html>
<head>
<title>Ordini in corso</title>
</head>
<body>
Utente
<?php
//ricerca utente
$codice=$_REQUEST['codice'];
$password=$_REQUEST['password'];
$host='localhost';
$nome_db='libri';
$conn=mysql_connect($host) or die ("errore di connessione");
$query= 'SELECT U_cognome, U_nome FROM UTENTE WHERE (
U_codice ="$codice" AND U_password ="$password")';
$utente=mysql_query($query,$conn);
$nrec=mysql_affected_rows($conn);
If ($nrec<1)
    die ("utente non trovato");
$dati_utente=mysql_fetch_array($utente);
print "$dati_utente['U_cognome'] $dati_utente['U_nome'];
$query="SELECT O_numero,O_data FROM ORDINE WHERE (O_utente='$codice' AND O_stato='in corso')";
$ordini=mysql_query($query,$conn);
while ($ordine=mysql_fetch_array($ordini);
    print "<p>Ordine $ordine['O_numero'] del $ordine['O_data'] </p>";
mysql_close($conn);
</body>
</html>
```